



UNIVERSIDAD DE VALENCIA
FACULTAD DE FILOSOFÍA Y CIENCIAS DE LA EDUCACIÓN
Departamento de Educación Comparada e Historia de la Educación
PROGRAMA DE DOCTORADO EN EDUCACIÓN

**¿PUEDE EL *CODING* MODIFICAR EL PROCESO DE
APRENDIZAJE/ENSEÑANZA EN LA ESCUELA PRIMARIA?**

TESIS DOCTORAL

Presentada por
Annalisa Piazza

Dirigida por:
Santiago Mengual Andrés

VALENCIA
Julio de 2021

A mi hija Carla

Agradecimientos

Mi más profundo y sincero agradecimiento es para mi tutor y director de tesis Santiago Mengual-Andrés pues constantemente me proporcionó su valiosa y competente ayuda durante la elaboración del trabajo. Gracias a sus consejos no me sentí desorientada en este complejo y articulado mundo de la investigación científica.

Además quisiera agradecer a todos los docentes de la Facultad de Filosofía y Ciencias de la Educación de la Universidad de Valencia que contribuyeron a mi formación profesional en el ámbito de la Educación.

Un agradecimiento también para el Director Escolar del Instituto Comprensivo "Don Bosco" de Ribera, Doc. Anna Conti, a las docentes, a los alumnos y a los padres de familia que, cada uno por su parte, permitieron realizar la experimentación presentada en esta tesis de doctorado.

Un agradecimiento especial a mi amiga de toda la vida Leonarda por todo lo que ha hecho, hace y hará.

Por último, un inmenso agradecimiento a quienes compartieron conmigo, casi de forma inconsciente, esta experiencia dentro del hogar: mi hija Carla. No es fácil compaginar los tiempos de estudio con las necesidades de una niña. Sin embargo, tuve la suerte de contar en todo momento con el apoyo de mis padres y de mi esposo, Angelo, a quien agradezco al último, pero no por orden de importancia, él compartió conmigo responsabilidades y tareas que no pueden delegarse a otras personas.

ÍNDICE

Agradecimientos.....	3
Índice	4
Índice de tablas.....	8
Índice de figuras	10
Índice de siglas.....	11
Capítulo 1.....	12
Introducción: Estructura de trabajo y definición del plan de investigación	12
1. Introducción.....	12
1.1 La difusión de la tecnología en contextos escolares	13
1.2 Planificación del trabajo de investigación y su importancia	17
1.3 Organización y estructura de trabajo	23
Capítulo 2.....	29
Revisión de la literatura: pensamiento computacional y uso del <i>coding</i> en las disciplinas del área STEM	29
2. Introducción.....	29
2.1 Desarrollo del pensamiento computacional y su definición.....	39
2.2 Las raíces del pensamiento computacional	43
2.3 Programación y pensamiento computacional	47
2.4 Etimología y orientaciones teóricas	51
2.4.1 En la literatura.....	51
2.4.2 Iniciativas en la Web	55
2.5 El pensamiento computacional en la educación K-12.....	59
2.6 Modelos de desarrollo del pensamiento computacional.....	63
2.6.1 Modelo Scratch: USA	63
2.6.2 Modelo CAS: Computing at School (Reino Unido).....	65
2.7 Ejemplos de difusión del pensamiento computacional en el ámbito educativo	69
2.7.1 El modelo de Barr y Stephenson.....	70
2.7.2 Los recursos de C.S.T.A.	71
2.7.3 Los recursos del modelo CAS.....	73
2.8 Evaluar el pensamiento computacional	75

2.8.1 La evaluación con el modelo Scratch	76
2.8.2 La evaluación con Dr. Scratch	78
2.8.3 La organización <i>Bebras</i>	80
2.9 Contribuciones a la difusión del pensamiento computacional.....	81
2.9.1 La hora del código: una campaña mundial	81
2.9.2 Otros programas e iniciativas	83
2.10 El <i>coding</i> : instrumento de promoción del pensamiento computacional.....	86
2.10.1 El <i>coding</i> de forma online.....	86
2.10.1.1 Scratch	92
2.10.1.2 El sitio code.org	98
2.10.1.3 Robótica educativa	102
2.10.1.4 <i>Game design</i>	110
2.10.2 El <i>coding</i> en modo <i>unplugged</i> (desconectado)	112
2.11 <i>Coding</i> y habilidades sociales	116
2.11.1 École 42: la escuela de <i>coding</i> para todos	118
2.12 <i>Coding</i> en las disciplinas STEM	120
2.12.1 Estudios recientes sobre los efectos del <i>coding</i> en la enseñanza de matemáticas y geometría	125
2.13 Recursos educativos para integrar el <i>coding</i> en la escuela	128
2.14 Políticas educativas que apoyan el pensamiento computacional.....	132
2.14.1 Iniciativas europeas	132
2.14.2 El camino hacia la difusión del pensamiento computacional en Italia	140
2.14.2.1 Campañas de alfabetización informática y de difusión del pensamiento computacional.....	141
2.14.3 Políticas educativas en Italia.....	143
2.15 Conclusiones	150
<i>Capítulo 3</i>	152
<i>Metodología y plan de investigación</i>	152
3. Introducción.....	152
3.1 Enfoque, preguntas y planificación de la investigación	152
3.2 Participantes y contexto	158
3.2.1 Selección de la muestra.....	159
3.3 Instrumentos y variables.....	163

3.3.1 Instrumentos para la recopilación de datos cuantitativos: Prueba de evaluación de los aprendizajes	165
3.3.1.1 Las pruebas INVALSI.....	166
3.3.1.2 Prueba inicial y Prueba Final: características	168
3.3.1.3 Validez del contenido.....	170
3.3.1.4 Versión final de las pruebas de geometría.....	171
3.4.1 Cuestionario.....	172
3.4.1.1 AEQ-ES: Cuadro teórico de referencia.....	177
3.4.2 Instrumentos para la recogida de datos cualitativos.....	181
3.4.2.1 Observación no participante	181
3.4.2.1.1 Método de recopilación de datos de observación.....	186
3.4.2.2 Entrevista.....	190
3.5 Tratamiento: actividad propuesta al grupo experimental	191
3.5.1 Procedimiento.....	195
3.6 Análisis de los datos.....	197
3.7 Conclusiones	199
Capítulo 4.....	201
Resultados	201
4. Introducción.....	201
4.1. Análisis de la validez de contenido y fiabilidad de las pruebas.....	201
4.2. Análisis cuantitativo.....	206
4.2.1. Resultados de las pruebas de geometría.....	206
4.2.2. Cuestionario AEQ-ES.....	214
4.2.2.1. Análisis preliminar cuestionario AEQ-ES.....	214
4.3. Análisis cuali-cuantitativo	230
4.3.1. Observaciones en las aulas.....	230
4.3.1.1. Observación del grupo control.....	233
4.3.1.2. Observación en la fase de tratamiento del grupo experimental.....	241
4.3.1.3. Comparación de resultados.....	252
4.3.2. Entrevista a la docente del grupo experimental.....	255
4.4. Conclusiones	258
Capítulo 5.....	261
Discusión y conclusiones	261

5. Introducción.....	261
5.1. Enseñar y aprender la geometría con Scratch	263
5.2. Discusión	268
5.2.1. Discusión sobre el desempeño y las emociones en geometría.....	269
5.2.2. Discusión sobre las emociones, la motivación y las relaciones sociales en geometría.....	270
5.2.3. Discusión sobre actividades, metodologías y estrategias utilizadas por los docentes	275
5.3. Conclusiones e implicaciones del uso del <i>coding</i> en la didáctica.....	278
5.4. Recomendaciones para futuras investigaciones	281
5.5. Limitaciones y aportaciones de la investigación	283
5.6. Conclusiones	286
Anexos.....	288
Anexo A.....	289
Anexo B.....	290
Anexo C.....	292
Anexo D1 – Cuestionario AEQ_ES para niñas.....	307
Anexo D2 – Cuestionario AEQ_ES para niños.....	320
Anexo D3 – Consentimiento informado.....	332
Anexo E.....	334
Anexo F.....	340
Anexo G.....	348
Anexo H1 - Archivo de la prueba inicial.....	349
Anexo H2 - Archivo de la prueba final.....	361
Referencias.....	373

Índice de tablas

Tabla 1 Distribución de los documentos seleccionados por año y cada tres años en Scopus.....	31
Tabla 2 Tipo de documentos publicados	32
Tabla 3 Tipo de documentos publicados por años.....	33
Tabla 4 Productividad por área temática.....	34
Tabla 5 Productividad por área geográfica.....	35
Tabla 6 Artículos más citados.....	36
Tabla 7 Como define J. Wing el término <i>computational thinking</i> a través de los años.....	44
Tabla 8 Conceptos y prácticas relacionadas al pensamiento computacional.....	50
Tabla 9 Definición del término pensamiento computacional.....	58
Tabla 10 Conceptos y subconceptos del modelo CSTA.....	73
Tabla 11 Localización de las iniciativas que sostienen el pensamiento computacional.....	86
Tabla 12 Herramientas de robótica educativa.....	109
Tabla 13 Método cuasi experimental con grupo de control no equivalente.....	155
Tabla 14 Sistema de la investigación: aplicación del método cuasi experimental	165
Tabla 15 Estructura de la prueba inicial (pretest).....	171
Tabla 16 Estructura de la prueba final (postest)	172
Tabla 17 Distribución de los elementos en función de la categoría de emoción indagada	177
Tabla 18 Distribución de los códigos identificados después de la observación.....	187
Tabla 19 Análisis de los datos.....	197
Tabla 20 Índice de validez de contenido de los elementos del pretest de geometría (G1)	204
Tabla 21 Índice de validez de contenido de los elementos del postest de geometría (G2).....	205
Tabla 22 Nivel de confianza en de las pruebas de geometría (pre y postest).....	206
Tabla 23 Descripción de las características de los grupos por género.....	207
Tabla 24 Puntuaciones mínimas y máximas de las pruebas de geometría.....	207
Tabla 25 Promedio (M) y desviaciones estándar (DS) de las puntuaciones conseguidas en pre y postest en ambos grupos de estudiantes.....	208

Tabla 26 Prueba de normalidad aplicada a pre y postest	209
Tabla 27 Comparación de los resultados obtenidos en el pretest y postest entre los grupos.....	210
Tabla 28 Comparación entre grupos de la prueba final divididos por dimensión	211
Tabla 29 Test t de Student por los niños y las niñas en ambos grupos.....	213
Tabla 30 Nivel de confianza dentro del cuestionario AEQ-ES.....	215
Tabla 31 Nivel de confianza dentro de las tres dimensiones del cuestionario AEQ-ES.....	215
Tabla 32 Puntuaciones mínimas y máximas relacionadas a cada subescala (cuestionario AEQ_	
ES).....	217
Tabla 33 Estadística descriptiva grupo control (cuestionario AEQ-ES).....	219
Tabla 34 Estadística descriptiva grupo experimental (cuestionario AEQ-ES).....	224
Tabla 35 Prueba estadística para la comparación entre los grupos en el postest divididos por	
sub escala.....	229
Tabla 36 Estadísticas descriptivas globales con referencia a cada código identificado para cada	
grupo.....	232
Tabla 37 Correlaciones entre las variables presentes en el conjunto de datos de las observaciones	
(grupo control).....	240
Tabla 38 Distribución de frecuencias de códigos significativos (grupo experimental).....	246
Tabla 39 Correlaciones entre las variables presentes en el conjunto de datos de las observaciones	
durante las actividades de coding (grupo experimental)	251
Tabla 40 Distribución de las frecuencias de códigos: actividades y estrategias (comparación de	
grupos).....	252
Tabla 41 Distribución de las frecuencias de códigos con valencia positiva y negativa (comparación	
de grupos).....	253

Índice de figuras

Figura 1. Interfaz de Scratch.....	92
Figura 2. Espiral del aprendizaje creativo (Resnick, 2007).....	95
Figura 3. Angry bird en code.org.....	99
Figura 4. Dibujar polígonos con Scratch	130
Figura 5. Tarea 1:dibujar un cuadrado con Scratch.....	194
Figura 6. Histogramas de promedio y desviación estándar en el pretest	212
Figura 7. Histogramas de promedio y desviación estándar en el postest.....	213
Figura 8 Modelo de distribución de los códigos (grupo control).....	236
Figura 9 Aula: entorno de clase (grupo control) y puntos de observación.....	237
Figura 10 Categorías de códigos comparadas en Maxqda (grupo control).....	238
Figura 11 Distribución de la frecuencia porcentual de los códigos en el grupo control.....	239
Figura 12 Aula: entorno de clase (grupo experimental) y puntos de observación.....	241
Figura 13 Laboratorio multimedia: entorno de clase (grupo experimental) y puntos de observación.....	242
Figura 14 Modelo de distribución de los códigos (grupo experimental).....	245
Figura 15 Distribución de la frecuencia porcentual de los códigos (Maxqda) en el grupo experimental.....	249
Figura 16 Categorías de códigos comparados en Maxqda (grupo experimental)	250
Figura 17 Categorías de códigos, comparación entre los grupos (distribución de frecuencia).....	254

Índice de siglas

ACM - Asociación de Maquinaria Informática
CIC - Cyber Innovation Center
CINI - Consorcio Interuniversitario Nacional para la Informática
CS – Informática
CSS - Hojas de estilo en cascada
CSTA - Asociación de Profesores de Informática
DS - Director Escolar
DSGA - Director de Servicios Administrativos
EDC - Centro de Desarrollo Educativo
EMMA - Agregador europeo multilingüe Mooc
ESO - Educación Secundaria Obligatoria
GFE – Google para la educación
HTML - Lenguaje de marcas de hipertexto
K-12 - desde el Kindergarten hasta el Grado 12
JRC - Centro común de investigación
ICILS - Estudio internacional de alfabetización informática y de la información
IEA - Asociación Internacional para la Evaluación del Rendimiento Educativo
INTEF - Instituto Nacional de Tecnologías Educativas y de Formación del Profesorado
ISTE - Sociedad para la Tecnología en la Educación
MIT - Massachussets Institute of Technology
MIUR - Ministerio de Educación Universitaria y de Investigación
MOOC – Cursos online masivos y abiertos
NEE - Necesidades Educativas Especiales
NMSI - National Math + Science Initiative
NRC - Consejo Nacional de Investigación
OCDE - Organización para la Cooperación y el Desarrollo Económico
ONG - Organización no gubernamental
PISA - Programa de Evaluación de Estudiantes Internacionales
PSND - Plan Nacional Escuela Digital
SOFIA Sistema Operativo para la Formación y las Iniciativas de Actualización de los docentes
SGD - Diseño escalables de juegos
TIC - Tecnologías de la Información y la Comunicación
TIMSS - Tendencia en el estudio internacional de las matemáticas y las ciencias
STEM - Ciencia, Tecnología, Ingeniería, Matemáticas
UE - Unión Europea.

Capítulo 1

Introducción: Estructura de trabajo y definición del plan de investigación

1. Introducción

En la vida diaria, nos encontramos con un mundo repleto de tecnología; con nuevas interfaces, súperordenadores, televisores y teléfonos inteligentes; sin embargo, no existe un desarrollo análogo en las actitudes y habilidades de los usuarios y ciudadanos en cuanto al logro de una alfabetización digital adecuada (*digital literacy*). La innovación, que avanza con gran velocidad, requiere una inversión no solo en los instrumentos sino, sobre todo, en las habilidades. Los jóvenes del siglo XXI, aun viviendo en este mundo interconectado y de rápida evolución, no parecen estar preparados para los retos y oportunidades que se les presentan, para los requisitos e innovaciones que las Tecnologías de la Información y la Comunicación (TIC) ofrecen (Prensky, 2011). Por lo tanto, es necesario conseguir y perseguir las habilidades específicas que hoy en día requiere el mercado laboral, así como un cierto dominio de la tecnología.

En la actualidad, estamos viviendo un cambio social vinculado a la necesidad de adaptación que exigen las nuevas redes de comunicación ya predominantes en la era digital, que involucra a todos los sectores de la sociedad, de la economía y de la cultura (Castells, 2014). Resulta, por tanto, necesario mirar hacia una sociedad cada vez más tecnológica, en la que sus miembros son ciudadanos digitales a los que se les debe garantizar una alfabetización digital adecuada, constituyéndose como un verdadero instrumento de democracia (Castells, 1999).

El cambio anteriormente citado, sin lugar a duda, también hacon llevado transformaciones a nivel educativo y escolar (Kong & Abelson, 2019). Los progresos en el campo de las TIC han propiciado el desarrollo de aplicaciones, software y interfaces digitales con el fin de ofrecer experiencias efectivas de aprendizaje para los estudiantes de los distintos niveles educativos y en las diversas disciplinas; el crecimiento en este sector ha determinado un aumento significativo en los estudios que buscan medir los efectos del uso de la tecnología sobre la eficiencia del aprendizaje en los estudiantes, a partir de Educación Primaria (Chauhan, 2017).

La escuela ha tenido y sigue teniendo un rol fundamental que debe ser reconocido por la sociedad, por lo que asume una gran responsabilidad en cuanto al desarrollo de habilidades, como instrumento de crecimiento personal, de desarrollo cultural y socioeconómico.

1.1. La difusión de la tecnología en contextos escolares

Desde la introducción de las (TIC) en el ámbito educativo, los gobiernos de todos los países europeos se han esforzado en desarrollar sus sociedades con el objeto de volverlas cada vez más competitivas y dinámicas, fruto de los efectos de la globalización, entre otros.

A partir de los contextos educativos, las investigaciones internacionales del *Programme for International student Assessment* (PISA) promovidas por la Organización para la Cooperación y el Desarrollo Económico (OCDE), han tenido el objetivo de detectar las habilidades de lectura, matemáticas y ciencias en los estudiantes de quince años, con el fin de minimizar la *digital divide*, es decir, la brecha digital existente entre quien usa y tiene acceso a las tecnologías de información (TIC) y quien, por el contrario, no las usa ni tiene acceso a ellas.

Sin embargo, en el panorama escolar, la introducción de lo digital ha creado y sigue causando divisiones y polémicas, especialmente entre los

docentes que están directamente involucrados en el proceso de innovación y renovación de los métodos de enseñanza tradicionales (Livingstone, 2012).

De cualquier forma, la tecnología ha entrado en las escuelas de todo el planeta, no solo para intentar resolver problemas de aprendizaje que se registran entre los estudiantes, sino más bien, con el objetivo principal de aprender, conocer y proporcionar instrumentos de apoyo válidos en el estudio a través de los lenguajes propios del mundo digital, mucho más cercanos a los jóvenes de nuestros tiempos. Se considera que, la tecnología en la escuela contribuye a formar futuros ciudadanos y trabajadores que desarrollen habilidades digitales, es decir, la capacidad de elaborar información, comunicarse con los demás, saber crear nuevos contenidos y resolver los problemas, entre otras competencias, habilidades fundamentales en la vida actual y en el mundo laboral (Ilomäki et al., 2016; Mengual-Andrés et al., 2016).

El sistema educativo, como pilar de crecimiento e inclusión de los ciudadanos de la UE, está llamado a responder a la solicitud de preparar a los ciudadanos para aprovechar al máximo las oportunidades que ofrece un mundo cada vez más interconectado y en continua evolución (Comisión Europea, 2018) donde, de hecho, la tecnología empleada en los diversos sectores laborales, – desde el sector médico hasta el sector industrial, o desde el agroalimentario hasta los transportes–. Hoy en día, por ejemplo, la tecnología se ha enfocado en el desarrollo de la inteligencia artificial como en el *machine learning* (aprendizaje automático) rama de la informática que estudia los mecanismos que permiten a una máquina “inteligente” mejorar sus capacidades y prestaciones–, como en el sector automovilístico (Stilgoe, 2018), o el reconocimiento de los hábitos y gustos de los usuarios de un determinado contenido digital (redes sociales, Netflix, etc.), y dichos mecanismos llevan a modificar y condicionar el estilo de vida de todos.

La Comisión Europea, a partir del 2016, reconociendo que la carencia de habilidades digitales obstaculizaba el crecimiento en Europa, junto con los

Estados miembros, las empresas, las partes sociales, las ONG y el sector de la educación, creó una coalición para el desarrollo de habilidades y ocupaciones digitales con el compromiso de reducir las respectivas brechas en todos los niveles, desde las especializadas de alto nivel hasta las que sirven a los ciudadanos para relacionarse con la sociedad digital y no permanecer excluidos de ella (Comisión Europea, 2016). El impulso proporcionado por la UE ha llevado a los Estados miembros a movilizarse para proporcionar las habilidades requeridas a sus ciudadanos partir del contexto educativo.

Han sido varias las iniciativas que se han llevado a cabo al respecto, en particular, la proyección e integración de prácticas educativas para reforzar las habilidades digitales como el pensamiento computacional (Nouri et al., 2020), las áreas de conocimiento STEM (Ciencia, Tecnología, Ingeniería, Matemáticas) y la promoción de comportamientos positivos para el desarrollo de habilidades sociales de alta calidad en los estudiantes (Caballero-González et al., 2019).

El Instituto Nacional de Tecnologías Educativas y de Formación del Profesorado (INTEF), que forma parte del Ministerio de la Educación y de la Formación Profesional en España, por ejemplo, colabora con las instituciones escolares y administrativas para promover una educación digital competente, tanto dentro de las escuelas a través de la experimentación de instrumentos de autoevaluación online, como entre los docentes y estudiantes. Uno de los proyectos que se han propuesto, en el ámbito del uso y desarrollo de la tecnología es la Escuela de Pensamiento Computacional, que tiene el objetivo de ofrecer recursos educativos, formación y soporte tecnológico a los docentes españoles que quieren consolidar sus propios conceptos de pensamiento computacional a través de actividades de programación y robótica en su práctica docente. Durante el curso escolar 2018/2019 más de 700 docentes y 14.000 estudiantes participaron en este proyecto que preveía que, los estudiantes de primaria, aprendieran matemáticas a través de *Scratch 3.0*, los de primer curso de Educación Secundaria Obligatoria (ESO) el conocimiento y el uso de *Arduino*

en el campo de la técnica creativa y los estudiantes de segundo de bachillerato, la programación de drones y robots.

La idea a partir de la cual toma forma la presente investigación responde a la creciente demanda de incentivar, en contextos escolares, el uso de la tecnología dentro de todas las áreas curriculares y, en particular, en el campo científico y tecnológico típico de las disciplinas STEM, actualmente objeto de interés para los gobiernos que, de hecho, han cambiado la orientación de las políticas de educación hacia dichos argumentos. El objetivo, por tanto, es desarrollar no solo habilidades profesionales del sector, sino también promover entre los estudiantes nuevas identidades de personas capaces y ciudadanos de una sociedad global y en evolución que conduzcan a reducir la brecha de habilidades que hoy en día es aún evidente.

Los resultados de las investigaciones internacionales PISA (*Programme for International Student Assessment*) (INVALSI, 2018) y TIMSS (*Trend in International Mathematics and Science Study*) (INVALSI, 2015) pensados para monitorizar el nivel de preparación de los estudiantes en ciencias y matemáticas; han evidenciado lagunas disciplinarias, que llevan a centrar la atención hacia nuevos enfoques didácticos dentro de los sistemas educativos, estableciendo, como objetivo, la formación de estudiantes más capaces, también en las disciplinas STEM, en línea con las demandas del mercado de trabajo (González & Kuenzi, 2012; Idin, 2020).

Del mismo modo la Recomendación del Consejo del 22 de mayo del 2018 relativa a las competencias clave para el aprendizaje permanente, en el punto 16 de las consideraciones de la introducción, destaca la necesidad de motivar a los jóvenes a emprender carreras en las áreas de Ciencias, Tecnología, Ingeniería y Matemáticas (STEM) y resalta la importancia de mantener el desarrollo de las habilidades asociadas (Consejo de la Unión Europea, 2018).

1.2. Planificación del trabajo de investigación y su importancia

La introducción de las TIC en el ámbito escolar ha llevado a muchos investigadores a realizar estudios enfocados en analizar la eficacia de las experiencias de aprendizaje en los estudiantes con referencia a las diversas disciplinas escolares a través de las tecnologías de la información (Chauhan, 2017) así como para profundizar en cómo estas influyen en las habilidades transversales en la era digital, que son importantes para el ciudadano del siglo XXI. Los resultados de dichos estudios han sido tan distintos como variados y, hasta el día de hoy, la influencia que tiene el uso de la tecnología en los resultados del aprendizaje aún no ha quedado clara.

El presente trabajo pretende contribuir al respecto, buscando conocer el impacto de las TIC, como instrumentos tecnológicos, sobre el proceso de enseñanza y aprendizaje de los jóvenes estudiantes de Educación Primaria en matemáticas, para los cuales, incluso pequeñas mejoras en el aprendizaje pueden generar efectos positivos a lo largo de su futura vida escolar (Heckman, 2006).

En particular, se pretende estudiar los efectos del uso de metodologías didácticas relacionadas con la tecnología, y en particular con el *coding* sobre una disciplina como la geometría, una rama de las matemáticas que, en Educación Primaria italiana y europea en general, a menudo, no encuentra un espacio adecuado dentro del plan de estudios de matemáticas (Clements, 1999); incluso siendo la geometría y el razonamiento espacial fundamentales para aprender las matemáticas (Núñez & Lakoff, 2005).

Como se ha mencionado anteriormente, los resultados de las investigaciones nacionales e internacionales (INVALSI y PISA) han revelado diversas problemáticas relacionadas a las habilidades matemáticas de los estudiantes italianos y, por lo tanto, se cuestiona si dichas problemáticas pueden estar relacionadas de alguna manera, con las actividades didácticas

tradicionales, con un enfoque carente hacia el descubrimiento y llevadas a cabo en contextos poco significativos, en los cuales no hay estímulos. A partir de estas consideraciones toma forma la idea de proponer nuevas experiencias explotando el potencial del pensamiento computacional, traducción del término inglés *computational thinking* y que, en las últimas décadas ha adquirido un rol fundamental en el campo educativo y en las políticas normativas de muchos países, recibiendo gran atención por parte de diversos organismos internacionales que han solicitado expresamente su promoción a partir de la educación. El discurso científico que envuelve el concepto de pensamiento computacional es multidisciplinario e incluye contribuciones del mundo de la informática, de las ciencias cognitivas y de la educación (Kong & Abelson, 2019).

La *International Society for Technology in Education* (ISTE) es, por ejemplo, una comunidad notable de educadores que creen que el uso de la tecnología puede transformar la enseñanza y el aprendizaje, acelerando la innovación y la resolución de diversos problemas en el mundo de la enseñanza; al respecto, desde hace años, sigue proponiendo con referencia a la construcción de habilidades informáticas, la necesidad de desarrollar en los estudiantes el pensamiento computacional como elemento básico, principalmente porque ayuda a comprender los procesos lógicos y algorítmicos que son la base de los proyectos de hardware y software (ISTE, 2019). También, la *Computer Teachers Association* (CSTA), reconociendo las dificultades en la enseñanza de la informática, ha propuesto dar soporte a los educadores K-12 (desde el Kindergarten hasta el Grado 12) en esta difícil tarea, considerando fundamental educar a los estudiantes en habilidades de pensamiento computacional (CSTA, 2019).

El pensamiento computacional, por tanto, va más allá del uso de la tecnología; aun estando estrechamente relacionado con ella, es una habilidad importante que todo niño debería adquirir, no solo como habilidad técnica, sino como construcción mental que promueve el desarrollo del pensamiento lógico en muchas disciplinas. Habiendo reconocido su importancia, el pensamiento

computacional es considerado por muchos autores como la cuarta habilidad básica, además de leer, escribir y el cálculo, una habilidad esencial para el desarrollo de la alfabetización del siglo XXI (Wing, 2008).

Por lo tanto, se ha retomado el interés en un argumento que no es del todo nuevo (ya había sido tratado por Seymour Papert en los años ochenta) y, en poco tiempo, ha existido una unión en los intentos entre gobiernos, Ministerios de Educación de todo el mundo, investigadores y académicos con el objetivo de introducir el pensamiento computacional en los currículos escolares. El contexto educativo ha recogido este argumento para intentar que la informática destaque no tanto como una disciplina en sí, sino como una nueva forma de pensar, tal y como lo hace un especialista en informática—que piensa de manera algorítmica y resuelve los problemas planificando estrategias. Dichos conceptos, según los investigadores del campo, pueden ser trasladados a la vida real (Dennig, 2017; Román-González et al.; 2017 Shute et al.; 2017, Wing, 2008, Yadav et al. 2017).

Las investigaciones llevadas a cabo en el ámbito educativo muestran como punto en común la idea de que los estudiantes, aun no debiendo ser necesariamente expertos en informática, deben guiarse hacia una preparación de base que dé al pensamiento computacional un rol principal (Weintrop et al., 2016).

El pensamiento computacional, por tanto, está estrechamente relacionado con el pensamiento recursivo que permite aplicar una solución ya encontrada en un caso similar al problema que se está analizando, individualizar los aspectos relevantes, descomponerlos en problemas más simples, evaluar la eficacia del proyecto y la eficiencia del procedimiento (Barr et al., 2011). El programador transfiere un algoritmo de su propio cerebro a la máquina y esta habilidad requiere otras múltiples habilidades (Wing, 2006) entre las cuales también están la formalización rigurosa de los procedimientos, la creatividad y la fantasía. Gracias al pensamiento computacional quien aprende se “enseña a pensar”, a representar y resolver problemas que requieren combinar la potencia cognitiva

humana con la capacidad de cálculo (Kafai & Burke 2013; Lye & Koh 2014; Sengupta et al., 2013).

La programación representa, por tanto, un instrumento válido promovedor; que es una habilidad importante no solo en la informática, sino también en la educación de nuestros tiempos, de la que todos los niños pueden obtener un beneficio desde edades tempranas. En particular, el *coding*, término en inglés que traducido significa programación, no se refiere al sentido tradicional de la expresión; en concreto se refiere a la fase de escritura del código de un lenguaje de programación en particular y por lo tanto representa justo la última fase del proceso de desarrollo de un software. Representa una estrategia que pone la programación en el centro de un camino dirigido a la resolución de problemas (Barr & Stephenson, 2011; Grover & Pea, 2013; Kafai & Proctor, 2021; Lye & Koh, 2014).

Bien es sabido que este puede ser practicado también por los niños, además de por técnicos en informática e ingenieros –incluso por artistas–; muchos países han promovido su aprendizaje introduciéndolo en los planes curriculares, con el objetivo de dirimir la brecha evidente entre aquellos que se limitan a utilizar objetos digitales como juegos, software, App, y aquellos que por el contrario, de manera consiente, saben crearlos y compartirlos (Mishra & Yadav, 2013; Yadav et al., 2016). Para conseguir las habilidades relacionadas con la programación es posible recurrir a distintos enfoques y métodos, que pueden aplicarse tanto en contextos formales (ámbito escolar) como en situaciones informales como las clases extracurriculares o los *hackathon*, aprovechando los eventos y el material puesto a disposición por las organizaciones sin ánimo de lucro, gobiernos y sociedad (Tuomi et al., 2017).

La importancia dada al pensamiento computacional y, por ende, a la informática y en particular al *coding*, lleva a cuestionarse, con el fin de entender qué habilidades lo caracterizan, qué relación tiene con la programación y cómo puede ser integrado en los planes de estudio asegurándose de que su enfoque

hacia la resolución de problemas sea enseñado a los estudiantes de todos los niveles educativos (Angeli & Giannakos, 2020; Angeli et al., 2016; Bocconi et al., 2016; Kalelioglu et al., 2016; Pugnali et al., 2017).

El *coding* representa una nueva estrategia didáctica, una práctica que puede utilizarse en clase, y que es de fácil acceso en contextos que no están directamente relacionados con la programación (Rees et al., 2016; Sentance & Csizmadia, 2017;). En los últimos años, ha habido una gran atención mediática alrededor del *coding*, que se ha visto como un vehículo hacia el pensamiento computacional y la informática (Yadav et al., 2018) de manera que ha despertado el interés de los investigadores para analizar sus distintas perspectivas: técnicas, pedagógicas, éticas y económicas. El creciente interés por el *coding*, seguramente, se debe a los cambios sustanciales en los programas escolares europeos que favorecen la introducción y el diseño de planes de estudio que integran temas como el pensamiento computacional, la programación y la informática.

Se ha ido, por tanto, consolidando la idea de que, acercando a los estudiantes hacia el pensamiento computacional a través de instrumentos y métodos de programación visual típicos del *coding*, aprenderán a programar (*learn to code*) y, al mismo tiempo, mientras programan aprenden muchas otras cosas (*code to learn*) (Resnick, 2013).

Por lo tanto, el *coding* representa una de las prácticas didácticas más utilizadas por los docentes en el contexto educativo, desde la educación infantil (Grover & Pea, 2013; Kafai & Burke, 2013) que ya permite poner en práctica los conceptos propios del pensamiento computacional.

En los últimos años se han registrado varios estudios respecto a los beneficios del *coding* (Bishop-Clark et al., 2007; Howland & Good, 2015; Malan & Leitner, 2007; Nikou & Economides, 2014) sobre el desarrollo lógico-cognitivo de quien lo practica con efectos en el logro de habilidades específicas a partir de las

habilidades propias de la alfabetización digital, a las cuales se suman: la motivación, la autorregulación, el desarrollo de capacidades para la resolución de problemas, la superación del miedo al error y a ser juzgado, a realizar experiencias con logros gratificantes en cuestiones de autoeficacia, a desarrollar una actitud desafiante positiva y a conseguir una conciencia de las responsabilidades humanas en el uso de la tecnología.

El *coding* también representa un recurso válido para introducirse en la escuela como una actividad transversal, en todos los ámbitos del saber y, en particular, en las disciplinas del área STEM. De hecho, la recomendación de la UE del 2018 reclama la alfabetización STEM, invitando a explorar nuevos métodos de aprendizaje e inversión en educación de alta calidad y, el *coding*, puede representar un instrumento para lograrlo.

En el contexto italiano, al que se hace referencia en la presente tesis doctoral, el sistema educativo ha estado fuertemente influenciado de manera que, la vigente Ley 107/2015 del Plan Nacional Escuela Digital (PNSD, 2015), ha promovido una serie de iniciativas que sostienen el *coding* y el pensamiento computacional, dirigidas a todos los niveles educativos. Para la Educación Primaria, el Ministerio de Educación y de Investigación (MIUR) ha publicado una actualización de las *Indicaciones Nacionales*, en las que el pensamiento computacional ha sido reconocido como instrumento cultural para el crecimiento personal de la ciudadanía, e invita a los docentes a llevar a cabo actividades relacionadas con el pensamiento computacional que se introduzcan en las diversas disciplinas de enseñanza (MIUR, 2018).

Diversos estudios en investigación educativa han demostrado que los estudiantes aprenden las estrategias propias del pensamiento computacional mientras estudian y que sus profesores logran moldear el pensamiento y aplicar medidas de orientación adecuadas (Brennan & Resnick, 2012). En muchos casos, el *coding* ha sido un elemento clave para la enseñanza de dichas estrategias de pensamiento, gracias a la funcionalidad que ofrecen sus ambientes de

aprendizaje, que son particularmente ideales para promover el pensamiento computacional (Kalelioglu, 2015).

No es casualidad que en el mundo existan iniciativas a favor del desarrollo de habilidades asociadas a su difusión. Por ejemplo, en 2013 en París, nace una escuela de *coding* privada y gratuita, École 42, cuyo modelo educativo de innovación se ha convertido en ejemplo para muchos otros países, porque ha mostrado como la educación puede cambiar y proporcionar nuevos estímulos, incluso desde el punto de vista social, enfocándose en las capacidades y habilidades de cada niño y niña que para tener acceso no requiere algún título específico.

1.3. Organización y estructura de trabajo

Con la presente investigación se pretende entender si es posible aplicar, en el ambiente educativo, el *coding* como herramienta de aprendizaje innovadora e inclusiva para el estudio de las matemáticas, con el fin de transmitir a los estudiantes habilidades que favorezcan el razonamiento, la creatividad, la confianza en sí mismos y la colaboración con los demás.

Los cambios que han ido surgiendo llevan a una lectura crítica de los diversos estudios e investigaciones que se han hecho al respecto hasta el momento. Particularmente en referencia a cómo influyen sobre el currículo escolar, sobre las prácticas didácticas y qué elementos pedagógicos intervienen en todo el proceso de enseñanza y aprendizaje.

Las materias de la rama científica (STEM) bien se prestan a integrar las prácticas y conceptos del pensamiento computacional en el ámbito escolar; aun siendo aplicables en todos los ámbitos del saber. Representan, por tanto, un buen ejercicio que promueve en los alumnos el desarrollo de las habilidades relacionadas con el pensamiento computacional. En particular, disciplinas como las matemáticas ofrecen un contexto significativo para la aplicación de conceptos

relacionados tanto con el pensamiento computacional, como con el *coding* (Jona et al., 2014; Repenning et al. 2010; Wilensky et al. , 2014).

Con referencia a las líneas de pensamiento que ofrece la literatura científica en cuanto al pensamiento computacional en el panorama escolar, y en particular en Educación Primaria, se desea comprender los efectos del uso del *coding* como metodología didáctica en el estudio de una disciplina STEM, en particular: la geometría.

Con el objeto de que los estudiantes puedan desarrollar una comprensión completa de la geometría, deben tomar un rol activo en el aprendizaje (Piaget & Inhelder, 1969; Vygotsky, 1978). Este concepto nos lleva a revisar las metodologías didácticas que han sido llevadas a la práctica por los docentes en la enseñanza de la geometría, implementando buenas prácticas que puedan tener un impacto positivo sobre el proceso de enseñanza y aprendizaje que promueve la adquisición de conocimientos, habilidades y comportamientos que forman a los jóvenes en el contexto escolar.

La decisión de introducir actividades de *coding* dentro del plan curricular de la geometría alude a varios motivos. Aunque esta disciplina ha representado uno de los campos del saber más importantes de las matemáticas, actualmente se aborda y se enseña frecuentemente a los niños con enfoques que tienden a presentar, desde los primeros años de escuela, conceptos abstractos y distantes de su experiencia, por lo que resultan poco significativos. De hecho, se considera que la introducción de actividades de geometría a los niños de Educación Primaria a través de la programación visual por bloques tiene como valor agregado la orientación hacia el desarrollo de habilidades digitales en un contexto más dinámico y lúdico, sin descuidar el estudio de las figuras geométricas y sus propiedades desde un punto de vista más crítico. Estas herramientas podrían predisponer de forma positiva a los estudiantes hacia la geometría, lo cual, conllevaría resultados positivos, no solo sobre el concepto de

rendimiento académico, sino también en lo que respecta al ámbito emotivo y relacional (Benton et al. 2017, Crompton et al., 2018; Foerster, 2017).

El objetivo de la presente investigación, por tanto, es entender si el *coding* puede utilizarse como instrumento didáctico para dar apoyo al aprendizaje y que efectos se pueden manifestar en los estudiantes de una clase de cuarto curso (edad 9-10 años) de una escuela de Educación Primaria italiana.

La presente tesis puede incluirse dentro de la línea de investigación que considera al *coding*, además de un medio para promover y desarrollar el pensamiento computacional; un instrumento didáctico que puede utilizarse en los currículos escolares (Barr & Stephenson, 2011; Grover et al.; 2015; Lye & Koh, 2014). En particular, profundiza en temas relacionados con su integración curricular en matemáticas y, con más concreción, en geometría, con el objetivo de comprender "si" y "cómo" el *coding* puede mejorar y ampliar la percepción de dicha disciplina por parte de los niños y niñas, ya sea en cuanto a los resultados conseguidos como en cuanto a su interés, al grado de satisfacción y a su motivación hacia el aprendizaje de la geometría. Además, trataremos de estudiar los efectos de dicha práctica sobre las emociones de los estudiantes, sobre las dinámicas sociales y las relaciones entre los compañeros y entre el alumno y el docente. Finalmente se evaluará cómo y cuánto incide el rol docente como activador de los procesos de conocimiento y de las dinámicas dentro de las clases.

Por tanto, se tratará de comprender cómo, modificando el enfoque didáctico, cambian las habilidades y competencias que los alumnos alcanzan durante el proceso de aprendizaje.

Además, se cuestionará si las actividades de *coding* tienen efectos positivos sobre la autocomprensión, sobre cómo son vividas ciertas emociones como la ansiedad, la alegría, el aburrimiento, la diversión y sobre las dinámicas de las relaciones activadas con otros y con el mundo de la tecnología.

En síntesis, el objetivo general de la presente investigación es el de comprender si y cómo es posible integrar el *coding* y el pensamiento computacional en la enseñanza de las matemáticas en Educación Primaria a través de actividades funcionales para la enseñanza interdisciplinar. Para tal fin, se empleó una metodología experimental (cuasi experimental con un grupo de control no equivalente) aplicando a una clase de cuarto curso un modelo didáctico basado en el uso del *coding* y de los principios del constructivismo en el desarrollo de una actividad específica de geometría, con una metodología respaldada por recursos tecnológicos. De esta forma, ha sido necesario evaluar el impacto del modelo descrito de la presente investigación, enfocándose en:

- los resultados de aprendizaje conseguidos por los alumnos en geometría;
- las emociones de éxito que surgen en los alumnos;
- la gestión de todo el proceso de enseñanza/aprendizaje por parte del docente.

La presente tesis ha estado dividida en dos partes: una teórica, que se presenta en el capítulo 2 y una parte empírica en los capítulos sucesivos. Tras haber expuesto los motivos de base para esta investigación en el capítulo 1, a partir de la difusión del uso de instrumentos informáticos y, más en general de las tecnologías en el ámbito educativo, se abordará, en el capítulo 2 la revisión de la literatura que parte, como principal referente, en el periodo sucesivo a la publicación del artículo de Wing (2006).

Este trabajo ha sido, para los especialistas en la temática, un punto de inflexión en lo que se refiere al pensamiento computacional. Por lo tanto, el segundo capítulo de la presente tesis continua con la presentación del marco teórico que representa la base de esta investigación, y en la que se definirán los supuestos teóricos y pedagógicos que han favorecido la introducción de

prácticas relacionadas con el pensamiento computacional en el mundo de la educación, a partir de la predisposición de trayectorias y modelos elaborados *ad hoc*. Se abordarán, también, los problemas relacionados con la evaluación de las capacidades y habilidades asociadas con el pensamiento computacional y, posteriormente, se presentarán algunas de las herramientas más utilizadas en el ámbito educativo para favorecer la difusión del *coding*, entre las que encontramos Scratch, el sitio web *code.org*, la robótica educativa (herramientas relacionadas con la práctica del *coding en línea*) y herramientas en papel (*coding unplugged*) de las que se han estudiado límites, ventajas, implicaciones cognitivas y educativas. Posteriormente, se han estudiado los puntos en común entre el *coding* y las disciplinas del área STEM a través de una investigación sobre la aplicación de las prácticas de *coding* en la enseñanza y aprendizaje de la geometría. Una vez encontrados los elementos que relacionan el *coding*, entendido como herramienta de promoción del pensamiento computacional y, la geometría (disciplina STEM), se ha analizado su rol y su importancia en las políticas educativas europeas y, en particular, en las italianas. Del mismo modo se ha abordado, cuál es el estado del arte en la difusión del pensamiento computacional en las escuelas. Por último, el capítulo dos describirá también las estrategias y los instrumentos que pueden utilizarse en la práctica didáctica.

Seguidamente, el capítulo 3 esboza la metodología que se ha seguido en la elaboración del modelo de investigación, para el que se ha preparado un estudio cuasiexperimental y reporta información sobre la selección de la muestra en una escuela de Educación Primaria siciliana, constituida por niños de dos clases de cuarto curso que se han dividido en dos grupos, (experimental y de control). El capítulo describe las principales herramientas utilizadas en la recogida de datos que posteriormente fueron analizados y presentados en el capítulo 4 mediante el uso de estadísticos de tendencia central, dispersión y estadística paramétrica y no paramétrica.

Por último, la discusión de los resultados obtenidos se presenta en el capítulo 5, en el cual se establecen las conclusiones finales tras realizar un

análisis crítico e individuando las fortalezas y las debilidades del estudio y las implicaciones en futuras investigaciones.

Capítulo 2

Revisión de la literatura: pensamiento computacional y uso del *coding* en las disciplinas del área STEM

2. Introducción

Una vez establecido el fin de la presente investigación, el primer objetivo siguiente fue llevar a cabo una revisión sistemática de la literatura para encontrar documentos que hicieran referencia a la investigación que se pretende realizar.

La investigación se focalizó en un periodo de tiempo que comienza principalmente a partir del 2006, año en el que Wing publicó el principal artículo sobre el pensamiento computacional, el cual, dio lugar a diversas investigaciones y proyectos sobre el tema, de manera que actualmente se registran 1850 citas de su publicación en SCOPUS. Los documentos recopilados muestran que los estudios iniciales buscaban principalmente construir una definición compartida del concepto de pensamiento computacional y formar un cuadro teórico de referencia que pudiese sugerir las mejores prácticas didácticas para integrarlo en los contextos educativos. Posteriormente, los investigadores, a través de estudios cualitativos y cuantitativos, pusieron su interés en intervenciones específicas y metodologías didácticas que permitiesen medir y probar las capacidades, las habilidades, el interés y la motivación de los alumnos.

En la primera fase de la presente investigación se realizó una revisión bibliográfica en SCOPUS con el fin de encontrar las investigaciones más importantes en el sector de interés (Ciencias sociales, Matemáticas, Psicología, Informática) posteriormente se continuó con el estudio de la literatura a través de motores de búsqueda más importantes. Como se ha mencionado, la búsqueda de documentos se realizó a través de la base de datos SCOPUS, utilizando operadores booleanos y opciones de búsqueda avanzada. Se decidió

iniciar a buscar a través de las palabras clave: *computational thinking* AND *coding* obteniendo así los primeros 1860 documentos, posteriormente se redujo el campo de investigación ingresando otras palabras clave. La muestra definitiva sobre la cual se trabajó se obtuvo después de la segunda búsqueda efectuada combinando los factores de la siguiente manera:

TITLE-ABS-KEY (***computational thinking***) AND TITLE-ABS-KEY (***coding***) AND TITLE-ABS-KEY (***education***) AND TITLE-ABS-KEY (**K-12**) PUBYEAR > 2005 AND PUBYEAR < 2022

La muestra se redujo a 1055 documentos entre el 2006 y el 2021. Los documentos obtenidos proporcionaron información sobre:

- Fuente;
- Área temática;
- Posición geográfica;
- Artículos más citados.

Se llevaron a cabo análisis descriptivos de los indicadores de producción bibliométrica (productividad de los autores, tipo de documentos, años, citas, etc.) con el paquete estadístico SPSS 22. Es interesante observar cómo cambia la producción de artículos científicos por periodos de cada tres años, como se observa en la tabla 1.

Se observa como la publicación de artículos relacionados con este tema se ha concentrado en los últimos seis años 2021/20/19 y 2018/17/16 y, considerando que el 2021 abarca solo los primeros seis meses se observa, en los últimos tres años, un incremento mayor al 100%. Esto se ha dado gracias al fuerte impulso que el mundo académico y el legislativo han proporcionado aumentando el interés al respecto.

Tabla 1*Distribución de los documentos seleccionados por año y cada tres años en Scopus*

Año	Artículos	Artículos en tres años
2021	154	
2020	331	704
2019	219	
2018	148	
2017	93	291
2016	50	
2015	35	
2014	15	57
2013	7	
2012	2	
2011	1	3
2010	-	
2009	1	
2008	1	2
2007	-	
2006	1	1

Fuente: elaboración propia.

La investigación comenzó gracias a WOS, pero luego se extendió a las principales bases de datos académicas y bibliotecas digitales, en particular a SCOPUS teniendo en cuenta su amplia cobertura nacional e internacional y el hecho de que resulta menos restrictivo que WOS ofreciendo de esta manera, un panorama más amplio y por lo tanto más representativo sobre el tema de estudio (Mengual-Andrés et al., 2017). También se efectuaron búsquedas en:

- ACM Digital Library;
- Springer Link;
- Science Direct;

Para completar la investigación se extendió, además de en el sector específico de la Educación, a la IEE Digital Library que se especializa en literatura

técnica de calidad en el sector de la ingeniería y tecnología, teniendo en cuenta que la presente investigación está relacionada también de manera indirecta al campo de la informática.

Se recopilaron documentos de investigación con relación al pensamiento computacional, entre los cuales publicaciones de revistas científicas, documentos académicos, documentos de política y relacionados con convenios internacionales. Los datos obtenidos de SCOPUS muestran que, en general, la producción científica sobre el tema de estudio se concentra principalmente en dos tipos de documentos (Tabla 2). El 47,5% de estos (n= 501) se presenta como *Article* (artículos publicados en una revista científica) y por el contrario, 476 son los *Conference paper* y representan el 45,1% del total de los documentos. Los capítulos de libros y los *Review* indexados en SCOPUS representan el 5,8%, el resto de los documentos representan el 1,61% de la productividad (*Book, Editorial, Note, Conference Review* y *Undefined*).

Tabla 2

Tipo de documentos publicados

Tipo de documento	N	%
<i>Article</i>	501	47.5
<i>Conference paper</i>	476	45.1
<i>Book Chapter</i>	40	3.80
<i>Review</i>	21	1.99
<i>Book</i>	6	.57
<i>Editorial</i>	6	.57
<i>Note</i>	2	.19
<i>Conference Review</i>	1	.09
<i>Undefined</i>	2	.19
<i>Total</i>	1.055	100.0

Fuente: elaboración propia.

Analizando la distribución de frecuencia de los documentos subdivididos por año y prestando atención principalmente en la productividad de los últimos siete años (Tabla 3) el número de los documentos publicados en revistas científicas

(*Article*) ha aumentado con el paso de los años, y actualmente, ha alcanzado el máximo de producción en el 2020 (n= 174); lo mismo sucede en cuanto a los artículos publicados en conferencias (*Conference paper*); en el año 2020 se llegó a un total de 132 documentos con un porcentaje siempre en aumento a lo largo de los años.

Tabla 3

Tipo de documentos publicados por años

	2015		2016		2017		2018		2019		2020		2021	
	n	%	n	%	n	%	n	%	n	%	n	%	n	%
Article	7	20.6	16	32.0	21	22.6	61	41.5	90	41.1	174	52.6	124	48.8
Conference paper	24	70.6	30	60.0	66	71.0	63	42.8	122	55.7	132	22.6	126	49.6
Book Chapter	2	5.9	3	6.0	2	2.1	17	11.5	1	0.4	14	6.5	-	
Review	-	-	1	2.0	3	3.2	2	1.4	3	1.4	7	-	3	1.2
Book	1	2.9	-	-	1	1.1	1	0.7	-	-	1	-	-	
Editorial	-	-	-	-	-	-	1	0.7	2	1.0	3	6.5	-	
Note	-	-	-	-	-	-	-	-	1	0.4	-	-	1	0.4
Conference review														
Undefined	-	-	-	-	-	-	2	1.4	-	-	-	-		

Fuente: elaboración propia.

También se llevó a cabo un análisis de la productividad de los documentos encontrados en SCOPUS por áreas temáticas. Teniendo en cuenta el hecho de que SCOPUS permite la clasificación múltiple de fuentes, puede ser probable que algunos de los documentos recuperados hayan sido clasificados al mismo tiempo en al menos dos categorías distintas. El análisis arrojó una clasificación de 1940 documentos, que no puede relacionarse con la clasificación que se llevó a cabo en la Tabla 1 tanto porque en el conteo pueden haberse considerado, como ya se ha mencionado, los mismos documentos varias veces, como porque cambia también el periodo de búsqueda que, en este caso, no se limitó a un periodo determinado. Como se observa en la Tabla 4, son dos los campos de conocimiento que resultan particularmente interesados en el fenómeno del pensamiento computacional en contextos educativos; el sector de la informática

(*Computer Science*) con el 37.7% (n=732) y el de las ciencias sociales (*Social Science*) con 728 documentos que representan el 37.5% de la productividad total. Posteriormente, destacan el sector de la ingeniería (*Engineering*) (n= 165), las matemáticas (*Mathematics*) (n= 82) y la psicología (*Psychology*) (n= 65), documentos que cubren respectivamente el 8.5%, 4.2% y 3.3% de la productividad. En menor grado están también presentes otros sectores (Tabla 3) que en total representan el 8.8%.

Tabla 4*Productividad por área temática*

Área temática	N	%
<i>Computer Science</i>	732	37.7
<i>Social Science</i>	728	37.5
<i>Engineering</i>	165	8.5
<i>Mathematics</i>	82	4.2
<i>Psychology</i>	65	3.3
<i>Business, Management and Accounting</i>	34	1.8
<i>Arts and Humanities</i>	31	1.6
<i>Decision Sciences</i>	31	1.6
<i>Material Sciences</i>	13	.7
<i>Energy</i>	12	.6
<i>Environmental Science</i>	11	.6
<i>Physics and Astronomy</i>	11	.6
<i>Medicine</i>	6	.3
<i>Biochemistry, Genetics and Molecular Biology</i>	4	.2
<i>Chemical Engineering</i>	4	.2
<i>Neuroscience</i>	3	.15
<i>Pharmacology, Toxicology and Pharmaceutics</i>	3	.15
<i>Agricultural and Biological Sciences</i>	2	.1
<i>Multidisciplinary</i>	2	.1
<i>Earth and Planetary Sciences</i>	1	.1
<i>Total</i>	1940	100.0

Fuente: elaboración propia.

El análisis de la productividad sobre el tema pensamiento computacional y *coding* en la educación K-12 también se enfocó en la búsqueda de los países del

mundo que muestran, a través de las distintas publicaciones científicas, una sensibilidad o interés especial hacia dichos argumentos. Así pues, observando la Tabla 5 es posible observar que los Estados Unidos representan el porcentaje más alto de producción (43.3%) seguido por España (11.6%) y Italia (7.2%). Sin embargo, también es notable el resultado que se obtiene sumando la productividad de los países miembros de la Unión Europea ya que juntos representan el 38.0% de la productividad global, demostrando así su interés hacia estos argumentos.

Tabla 5*Productividad por área geográfica*

País	N	%
<i>United States</i>	362	38.4
<i>Spain</i>	111	11.8
<i>Italy</i>	52	5.5
<i>Turkey</i>	52	5.5
<i>United Kingdom</i>	41	4.3
<i>Greece</i>	37	3.9
<i>Finland</i>	35	3.7
<i>Sweden</i>	32	3.4
<i>Canada</i>	31	3.3
<i>Taiwan</i>	30	3.2
<i>Australia</i>	28	3.0
<i>Brazil</i>	28	3.0
<i>Hong Kong</i>	27	2.9
<i>China</i>	26	2.8
<i>Germany</i>	26	2.8
<i>Norway</i>	24	2.5

*No se incluyeron países con menos de 20 documentos publicados.

Fuente: elaboración propia.

Finalmente, se seleccionaron las revistas (excluyendo aquellas relacionadas con documentos de convenios y conferencias bajo el título de *Proceeding*) que han tenido el mayor número de citas, evidenciando en particular los ocho títulos más citados en la base de datos de SCOPUS (Tabla 6).

Tabla 6*Artículos más citados*

Título del artículo	Autores	Revista	Año	Citaciones
Review on teaching and learning of computational thinking through programming: What is next for K-12?	Lye, S.Y., Koh, J.H.L.	Computers in Human Behavior 41, pp. 51-61	2014	429
Computer games created by middle school girls: Can they be used to measure understanding of computer science concepts?	Denner, J., Werner, L., Ortiz, E.	Computers and Education 58(1), pp. 240-249	2012	196
Demystifying computational thinking	Shute, V.J., Sun, C., Asbell-Clarke, J.	Educational Research Review 22, pp. 142-158	2017	192
Visual programming languages integrated across the curriculum in elementary school: A two year case study using "scratch" in five schools	Sáez-López, J.-M., Román-González, M., Vázquez-Cano, E.	Computers and Education 97, pp. 129-141	2016	179
Constructionist Gaming: Understanding the Benefits of Making Games for Learning	Kafai, Y.B., Burke, Q.	Educational Psychologist 50(4), pp. 313-334	2015	136
Empirical studies on the Maker Movement, a promising approach to learning: A literature review	Papavlasopoulou, S., Giannakos, M.N., Jaccheri, L.	Entertainment Computing 18, pp. 57-78	2017	120
Assessing elementary students' computational thinking in everyday reasoning and robotics programming	Chen, G., Shen, J., Barth-Cohen, L., Jiang, S., Huang, X., Eltoukhy, M.	Computers and Education, 109, pp. 162-175	2017	114
A crafts-oriented approach to computing in high school: Introducing computational concepts, practices, and perspectives with electronic textiles	Kafai, Y.B., Lee, E., Searle, K., Kaplan, E., Lui, D.	ACM Transactions on Computing Education 14(1),1	2014	111
Computational thinking in K-9 education	Mannila, L., Dagiene, V., Demo, B., Rolandsson, L., Settle, A.	ITiCSE-WGR 2014 - Working Group Reports of the 2014 Innovation and Technology in Computer Science Education Conference pp. 1-29	2014	102
How to learn and how to teach computational thinking: Suggestions based on a review of the literature	Hsu, T.-C., Chang, S.-C., Hung, Y.-T.	Computers and Education, 126, pp. 296-310	2018	88

Fuente: elaboración propia

Con este trabajo inicial, se pretendió tener una visión global sobre el concepto de pensamiento computacional y el uso del *coding* como instrumento didáctico en la educación K-12, demostrando que el interés hacia dichos temas

ha aumentado con el pasar de los años, hoy más que nunca vienen considerados temas de actualidad; a partir del 2006 se incrementó la productividad lo que demuestra la importancia de dichos temas tratados particularmente en el territorio europeo.

Después de esta aproximación inicial, se llevó a cabo una búsqueda más detallada de documentos científicos que trataran los temas de interés. La búsqueda se realizó de la misma manera, a través de una palabra clave, digitando el término *computational thinking* al cual se asociaron otros filtros para afinar la búsqueda entre los cuales, además de las palabras *coding* y *education K-12*, se agregaron los términos *motivation*, *emotion*, *geometry*, *robotic education*, *Scratch* y *code.org*. De todos los documentos obtenidos se eliminaron aquellos que no hacían referencia explícita al pensamiento computacional en el ambiente escolar, ya que se consideran poco aptos para los fines del presente estudio.

A partir de la revisión de la literatura fue posible identificar y catalogar los artículos en cuatro categorías:

- *Computational Thinking;*
- *Computational Thinking and Coding Education in K-12;*
- *Computational Thinking and STEM. Education;*
- *Emotion, motivation and coding.*

Al mismo tiempo, se clasificaron los artículos que introducen al concepto de pensamiento computacional, los estudios empíricos que tienen como objeto el uso del *coding* como instrumento didáctico en el campo de la educación K-12 y aquellos que se refieren a las aplicaciones de las disciplinas STEM, con particular atención a la geometría. Además, la investigación ha comprendido el estudio de los procesos de enseñanza y aprendizaje cuando se practica el *coding*, con referencia a los contenidos, la motivación y el interés de los alumnos, a las

emociones y las experiencias de la autoeficacia, además de la evaluación de las habilidades obtenidas en la resolución de problemas (*problem solving*) y a su desempeño.

Los documentos seleccionados, junto con las fuentes normativas y los textos teóricos de base permitieron la elaboración de la parte teórica de la tesis.

El aumento del interés hacia el pensamiento computacional a través de los años, se ha confirmado también a través de Google Trends, donde se destaca el incremento de las búsquedas realizadas en línea, en todo el mundo sobre el tema y en tiempo real en el motor de búsqueda de Google, con una muestra suficientemente grande y representativa. Los datos muestran que entre el 2006 y hasta el día de hoy (junio de 2021), en el mundo, ha habido un incremento en las investigaciones realizadas en el sector del trabajo y de la educación, ya desde el año 2010 se observaba esta tendencia, y actualmente siguen aumentando. El interés de investigación es particularmente alto en países como Irlanda, Corea del Sur, Malasia, Nueva Zelanda, Indonesia, Países Bajos y Singapur.

En Italia, el interés sobre el argumento, con referencia al sector del trabajo y de la educación, comenzó a dar un giro notable en el año 2014 y la mayor frecuencia de investigación se encontró en las regiones del Sur (Calabria, Puglia, Sicilia y Campania). Sucede lo mismo cuando se inicia la búsqueda tecleando los términos "pensamiento computacional" y "*coding*" o "pensamiento computacional" y Educación Primaria.

En base a la revisión que se llevó a cabo de la literatura, en este capítulo se elaboró y se profundizó el tema del desarrollo y la difusión del pensamiento computacional en el ámbito escolar, centrando la atención en el *coding*, como instrumento de promoción del pensamiento computacional en las disciplinas del área STEM.

2.1 Desarrollo del pensamiento computacional y su definición

Pensamiento computacional (traducción en español del término “computational thinking”) es una expresión que, en los últimos años, se ha difundido y continúa expandiéndose en el mundo educativo y en todos los niveles escolares (Wing, 2017). Se trata de un paradigma de razonamiento real que ofrece la informática, de hecho, en el pasado se sostenía que era una habilidad fundamental para un informático, que puede fomentarse en el mundo escolar y que puede representar una ventaja para los estudiantes. Fue Jeanette Wing, directora del Departamento de Informática de la Universidad Columbia, quien llamó la atención de la comunidad científica mundial al concepto de *computational thinking*, considerándolo una habilidad esencial para el desarrollo de la alfabetización del siglo XXI (Bundy, 2007; Wing, 2008).

Según Wing (2006), el pensamiento computacional es una habilidad que se puede aprender a la par que la literatura, la escritura y la aritmética. Así como la lenguaje escrito y hablado es necesario para la comunicación y las matemáticas son la base para realizar cálculos, el pensamiento computacional permite elaborar correctamente y de manera eficaz la información, obteniendo resultados que las personas por sí mismas no pueden obtener en la resolución de problemas. Afirma que, en la vida cotidiana, además de en el trabajo y las ciencias, es posible utilizar las habilidades conseguidas a través de la educación al pensamiento computacional obteniendo ventajas.

En el mismo año en el que Wing publicó su punto de vista respecto a las prácticas computacionales como activadoras de procesos mentales estrechamente relacionados con la resolución eficaz de problemas, el Parlamento Europeo y el Consejo de la Unión Europea publicaron la Recomendación 2006/962/CE del 18/12/2006 en la cual se definen las habilidades clave para el aprendizaje permanente, que son necesarias para la realización y el desarrollo personal, la ciudadanía activa, la inclusión social y la ocupación. La habilidad digital es una de las ocho habilidades identificadas en el documento. El término

computational thinking es visto por Wing como un atajo que lleva a pensar como un científico informático, centrando la atención no tanto en el uso seguro de la tecnología sino en la comprensión de sus conceptos fundamentales (Bocconi et al., 2016). El hecho de fomentar el desarrollo del pensamiento computacional en los jóvenes parece ir bastante ligado a la creciente demanda de nuevas habilidades en la sociedad tecnológica de hoy en día (Yadav et al., 2017).

Pensar como un informático, de manera algorítmica y en múltiples niveles de abstracción permite mejorar las habilidades que posee un ser humano al planear una estrategia para resolver un problema determinado (National Research Council, 2010; Wing, 2006). Pensar de forma computacional significa lograr descomponer el proceso de la toma de decisiones en etapas individuales, alcanzando el objetivo previsto a través de un razonamiento en pequeños pasos (Aho, 2012; Wing, 2008;).

El pensamiento computacional es un proceso que permite reformular problemas aparentemente más difíciles que otros donde la solución es inmediata, recurriendo de ser posible a una reducción, inclusión, transformación o simulación de estos; significa pensar de manera recursiva, seguir un proceso paralelo, interpretar el código como datos y los datos como código. Con el pensamiento computacional se utilizan los procesos de abstracción y descomposición para abordar tareas complejas o para proyectar sistemas complicados. Además, entran en juego los conceptos de prevención, protección y recuperación haciendo una previsión en los peores casos, a través de la redundancia, la contención de los daños y la corrección de los errores (Cuny, et al., 2010; Curzon et al. 2014; Wing, 2006).

Así como el invento de la imprenta permitió la difusión de la tercera habilidad de base; en el campo de la educación la difusión de la tecnología, según afirma Wing, debe permitir el desarrollo del pensamiento computacional. Desde su punto de vista, Wing sintetiza aquello que caracteriza al pensamiento computacional:

- Nos lleva a pensar por conceptos: porque lleva a pensar como un informático, que raramente conoce a priori cuál es el problema que debe resolver, estudia el contexto en el que este se presenta y las exigencias de los usuarios, trata de comprender cuál es el problema para llegar a la solución. El informático siente la necesidad de comprender bien *por qué* se debe crear cierto programa, *qué cosa* debe hacer el programa y *cómo* se debe proyectar. Además, es consciente de que la solución del problema no es unívoca y puede corregirse. Por lo tanto, la informática posee herramientas generales que pueden usarse en situaciones que no pertenecen al mundo de la informática (Olimpo, 2017).
- Es una habilidad fundamental que cada persona debería dominar para vivir en la sociedad actual, no se trata de un proceso mecánico rutinario que, aun estando estrechamente relacionado con el uso del ordenador, puede llevarse a cabo sin inteligencia humana. Los dispositivos de cálculo son instrumentos que permiten afrontar problemas que nadie hubiera pensado resolver sin la ayuda de un ordenador. Tener un ordenadora disposición ayuda a enfrentar problemas complejos, el único límite es la imaginación del ser humano. Así pues, no se trata de hacer pensar a las personas como lo haría un ordenador sino de usar la inteligencia para enfrentar los problemas que de otra manera el ser humano no sabría resolver.
- Se trata de una idea, no de un artefacto ya que no se encuentra en un software o en un hardware sino en el modo en el que se enfrentan y se resuelven los problemas, como se maneja la vida cotidiana y la eficacia con la que se comunica y se interactúa con los demás.
- El pensamiento computacional afecta a todos, en cualquier lugar pero será una realidad solo cuando se haya integrado en el comportamiento humano de tal manera que no sea necesario referirse a él como una filosofía en particular.

- Es atribuible a un proceso de abstracción que sirve para definir los modelos y parámetros que los describen. A través de la abstracción es posible capturar propiedades esenciales comunes a diversos fenómenos y objetos y esconder las diferencias que son irrelevantes; es posible escalar y enfrentar la complejidad.
- Integra y combina el pensamiento matemático e ingeniero, ya que es evidente la unión entre la informática, y por lo tanto el pensamiento computacional, y las matemáticas, que son la base de todas las ciencias. De hecho, el informático, cuando proyecta y construye sistemas que interactúan con el mundo físico-real, desarrolla el pensamiento de ingeniería y las habilidades típicas de las disciplinas del área STEM (Science Technology Engineering Mathematics). Es fuerte la idea de que el pensamiento computacional es una habilidad central de dichas disciplinas y por este motivo debe transmitirse en las escuelas (Wing, 2006).

Los argumentos de Wing se extienden a todos los campos del saber e abarcan también todo el mundo de la educación lanzando un nuevo reto educativo que involucra la informática, la instrucción y las comunidades educativas (Kong & Abelson, 2019; Wing, 2008) y que requiere la difusión del pensamiento computacional desde los primeros años de la infancia hasta el final de la educación escolar, desde el Kindergarten hasta el Grado 12 (K-12), que en Italia corresponde con el ciclo que va desde la escuela infantil hasta el final de la escuela superior.

A nivel internacional, también el mundo escolar se ha abierto a la posibilidad de difundir los conceptos computacionales entre los alumnos de todas las edades, con buenos resultados, reconociendo el valor educativo. Una educación informática de alta calidad prepara a los alumnos para utilizar el pensamiento computacional y la creatividad para entender y cambiar el mundo.

En una de sus definiciones J. Wing afirma que el conseguir las habilidades

relacionadas a la adquisición del pensamiento computacional permite resolver problemas proporcionando soluciones que pueden ser representadas de manera eficaz por un medio tecnológico (Wing, 2014). Las soluciones se refieren a todos los tipos de problemas, desde aquellos bien estructurados a aquellos más complicados, como los problemas que se pueden presentar en la vida real, que no tienen soluciones definidas ni se pueden medir (Shute et al., 2017). Son problemas de naturaleza multidisciplinaria, sin embargo, su resolución despierta mayor curiosidad intelectual que puede motivar a quienes se preparan para resolverlo mejorando su actitud y rendimiento y manteniendo alto el nivel de interés (Angeli et al., 2016).

En síntesis, con la pregunta: *What is computable?* Wing lanza un reto didáctico a los distintos niveles institucionales, al mundo de la política, al de la educación en general y en particular a los docentes de todos los niveles que pretenden llevar a cabo un proyecto disciplinario o interdisciplinario que dé importancia a las prácticas relacionadas con el pensamiento computacional. Por lo tanto, especialmente en el área de la educación, ha comenzado a difundirse la convicción de que el pensamiento computacional puede dar soporte al aprendizaje en áreas de estudio distintas a la informática, y de esta manera se han iniciado estudios e investigaciones al respecto (Angeli et al., 2016; Barr & Stephenson, 2011; Curzon et al., 2014; Lye & Koh, 2014).

2.2 Las raíces del pensamiento computacional

Todos los artículos científicos que hacen referencia al pensamiento computacional contienen citas de Wing, promotor de la idea de que el pensamiento computacional ofrece beneficios más allá de la disciplina informática. Sin embargo, ya previamente se había hablado de pensamiento computacional en el campo de la educación (Grover & Pea, 2015).

En 1961, Alan Perlis, informático estadounidense, fue uno de los primeros en sostener la necesidad de promover en la escuela prácticas relacionadas con el pensamiento computacional proponiendo un curso cuyo objetivo no era el de

formar programadores de computadoras hábiles en el conocimiento de un lenguaje específico sino más bien, enseñar a construir y analizar los procesos (Greenberger, 1964). Esta tesis ha tomado valor a través del tiempo, por la actitud que muestran los estudiantes al utilizar instrumentos tecnológicos en casa y en la escuela, que puede entenderse como una ventaja para promover la difusión del pensamiento computacional, superando las barreras culturales, económicas, políticas y sociales (Wing, 2008).

En los años setenta, informáticos y educadores sostuvieron que la creación de un programa es fruto de un proceso creativo que prepara a quien lo realiza con algo que va más allá de la informática (Knuth, 1974) y que la programación forma parte de un recorrido más amplio que lleva a la resolución de problemas (Yadav et al., 2018).

El mérito de J. Wing fue poner el pensamiento computacional a la vista del mundo científico en muy poco tiempo, ganando así un lugar de honor en la literatura (Grover & Pea, 2015). La cual, a partir del 2006, enriqueció su definición inicial de pensamiento computacional (Durak & Saritepeci, 2018), como se menciona a continuación (Tabla 7):

Tabla 7

Cómo define J. Wing el término computational thinking a través de los años.

Año	Las definiciones de pensamiento computacional de J. Wing et al.
2006	"El pensamiento computacional representa una actitud y un conjunto de habilidades de aplicación universal que todo el mundo, no sólo los informáticos, debería estar dispuesto a aprender y utilizar".
2008	"El pensamiento computacional implica la resolución de problemas, el diseño de sistemas y la comprensión del comportamiento humano, recurriendo a los conceptos fundamentales de la informática".
2011	"El pensamiento computacional es el proceso de formulación de problemas y de sus soluciones, de manera que éstas se representen de forma que puedan ser llevadas a cabo por un agente procesador de información.".
2017	"El pensamiento computacional es el proceso de formulación de un problema y la expresión de sus soluciones de tal manera que un ordenador -máquina humana- pueda llevar a cabo eficazmente".

Fuente: elaboración propia.

Por su parte Seymour Papert dedicó su actividad de estudio como informático, matemático y pedagogo a la promoción del pensamiento

computacional. Fue pionero en la idea de que los niños, en fase de desarrollo pueden aprender pensamiento computacional, que él llamó *procedural thinking* (Papert, 1980), aunque no proporcionó una definición explícita.

La idea de Papert (1980) tiene sus raíces en la teoría constructivista de Piaget (1954), según la cual el niño construye activamente su conocimiento gracias a la experiencia a través del *learn by doing* (aprender haciendo). Si la teoría de Piaget se desarrolló para demostrar que el conocimiento del niño se construye gracias a la interacción práctica del sujeto con el objeto, Papert, con su construccionismo, afirma que la construcción de los conocimientos es mucho más significativa si el sujeto que aprende está inmerso en la construcción de algo en concreto y que puede compartirse (Papert, 1980).

Así pues, un enfoque cercano a la enseñanza construccionista da a los niños la libertad de explorar sus propios intereses a través de las tecnologías, o a través de objetos o dispositivos que facilitan el aprendizaje y que pueden mostrarse, puestos en discusión, examinados y observados. Para aprender los niños necesitan artefactos cognitivos que los lleven a crear representaciones reales del mundo con el que interactúan. Su forma de aprendizaje y de acción es parecida a la de los programadores que, al inicio de su trabajo, establecen las primeras líneas de códigos y posteriormente enriquecen el programa. En la concepción de Papert, según la cual la mente humana necesita crear artefactos para poder aprender, es posible ver al ordenador como un instrumento didáctico válido que da soporte a la educación y al aprendizaje; a través de su programación se puede llegar a la construcción de nuevos artefactos, y nuevas ideas (Papert, 1980).

Según Papert, la programación favorece el pensamiento procesal, enseña a dividir un problema en componentes más simples y a depurar, detectando los errores que pudieran presentarse y aportando los cambios necesarios si se encuentra algún defecto de funcionamiento. Este procedimiento empleado en la programación puede también aplicarse en todos los aspectos de la vida. Por lo

tanto, el hombre, en su visión construccionista, es un generador de conocimiento y esta construcción es mucho más productiva cuando el sujeto que está aprendiendo se encarga de construir algo en concreto, un proyecto real. Si esta forma de pensamiento concreto era vista por Piaget como un estado intermedio, en el enfoque dado por Papert es la protagonista del aprendizaje, hecho por productos concretos que deben poder mostrarse, puestos en discusión, examinados y observados (Papert, 1980).

Por lo tanto, los malos resultados en el aprendizaje de los estudiantes se deben a que no se propusieron los materiales adecuados que permitiesen hacer el concepto lo más real posible y más fácil de aprender.

Para esto, el uso de las tecnologías juega un rol fundamental en la creación de los ambientes de aprendizaje (Papert, 1980; 1993). A través del lenguaje LOGO, desarrollado por la Massachusetts Institute of Technology de Boston (MIT), Papert propone un ambiente de programación desarrollado a propósito para los niños en el ámbito de la educación matemática y de la geometría, que proporciona los instrumentos necesarios para concretizar el pensamiento abstracto. De esta manera, el niño programa al ordenador y no al contrario, escogiendo realizar arte, ciencias, música, juegos o búsquedas históricas. La enseñanza es privilegiada cuando el aprendizaje activo prevalece sobre la enseñanza (Papert, 1993), donde se le da valor a la resolución de problemas y a la búsqueda de problemas, a un conocimiento de tipo matemático, que da sentido a aquello que se desea aprender. La idea según la cual Papert está convencido de que los niños son capaces de aprender a programar nace de la conciencia de que ellos aprenden y actúan como los programadores informáticos: se enfrentan a los problemas como *bricoleurs* (manitas trabajadoras), es decir se lanzan inmediatamente hacia la resolución de un problema y lo hacen por ensayo y error (Papert, 1987).

De esta manera, adquiere importancia el hecho de construir un conocimiento útil y compartirlo, que se adapte a los estilos de aprendizaje de los

alumnos y que se lleve a cabo en contextos concretos para su uso (Papert & Harel, 1991). En este contexto, el docente se vuelve un facilitador, un promotor de actividades proyectadas por los niños que aprenden, comunican y comparten sus ideas, correctas o equivocadas. En caso de error el docente los guía, resaltando la importancia de la conciencia de los propios errores. Entonces, el conocimiento es el producto de una construcción activa por parte del sujeto que se encuentra en el centro del proceso formativo (Papert, 1986). De esta manera, los profesores y los estudiantes logran aceptar soluciones equivocadas e intentos fallidos, reconociendo que son parte de un recorrido que lleva hacia un resultado positivo. Se prefiere el trabajo en equipo, en el que todos los individuos que se encuentran en el proceso de aprendizaje son motivados a descomponer los problemas en partes más pequeñas que se puedan resolver, a hacer los problemas más simples y a generalizarlos.

Papert, con sus teorías, toca argumentos que llevan las nuevas tecnologías al escenario de la enseñanza y que por lo tanto, tienen implicaciones significativas en el campo de la educación. Su teoría de la educación constructivista no puede reducirse al uso de prácticas de enseñanza deterministas sino también como un enfoque cultural a la educación lejano de tecnicismos ya que aquellos que programan en LOGO rechazan la preocupación escolar de tener respuestas correctas e incorrectas pues el sentido de la vida no es tener la respuesta sino sacar adelante las cosas. El concepto referible a la cultura del LOGO lleva a "hacer que suceda" y es mucho más que un principio educativo o pedagógico: puede describirse como el reflejo de una filosofía de vida en vez de una teoría de educación, también como algo más específico que el constructivismo en el sentido que comúnmente se le atribuye a este término (Papert, 1999)

2.3 Programación y pensamiento computacional

El trabajo de Papert continuó gracias a Mitchel Resnick y al grupo de investigación *Lifelong Kinderkarten* en el Massachusset Institute of Technology

(MIT) de Boston, donde se ha estudiado el desarrollo de instrumentos educativos y de actividades de proyección llevadas a cabo con la ayuda de la tecnología, con el mismo enfoque constructivista. Con su trabajo Resnick muestra interés en los conceptos que hacen referencia al pensamiento computacional y, también él, proporciona su propia visión y definición al respecto. En particular, Resnick sostiene que saber programar abre nuevas oportunidades y desarrolla el pensamiento computacional en los jóvenes; no todos llegarán a ser expertos en informática o programadores pero obtendrán la habilidad de pensar de manera creativa y esquemática, sabrán trabajar en colaboración con los demás en cualquier campo de trabajo en el que se encuentren(Resnick et al., 2009).

Su definición de pensamiento computacional que posteriormente fue retomada por Lye y Koh (2014) se puede explicar haciendo referencia a tres distintas dimensiones:

- Los conceptos computacionales (*computational thinking concepts*) que son los conceptos que los diseñadores utilizan para programar;
- Las prácticas computacionales (*computational thinking practices*) que se refieren a las habilidades que los diseñadores desarrollan cuando programan;
- Las perspectivas computacionales (*computational thinking perspectives*): que los jóvenes logran desarrollar sobre cómo ven la realidad que los rodea cuando programan. Los diseñadores jóvenes observan sus propios progresos, tanto en cuanto a su aprendizaje, como en cuanto a las relaciones que establecen con los demás y con el mundo tecnológico que los rodea.

Desde un punto de vista educativo, aprender los conceptos de la programación y ponerlos en práctica es ciertamente un valor. El pensamiento

computacional permite resolver los problemas en el ámbito científico, pero también ofrece la posibilidad a cada uno de expresarse en el estudio, el trabajo y la vida. Una persona que ha desarrollado el pensamiento computacional utiliza las nuevas tecnologías de forma activa, como instrumentos para expresarse y expresar su creatividad. De esta manera responde a la exigencia de ajustarse a los requerimientos de habilidades específicas en el mundo del trabajo y en la vida cotidiana, a saber comunicar y trabajar en equipo para conseguir un objetivo o una solución en común, a saber enfrentar un problema abierto un paso a la vez, con la conciencia de que un problema puede tener infinitas soluciones, que requieren curiosidad, creatividad y personalidad (Brennan & Resnick, 2012; Resnick et al., 2009;).

Por lo tanto Resnick comparte la idea de Papert, ya que considera que el ordenadores una herramienta óptima, que cuando se programa, puede crear artefactos; en su visión los niños pueden adquirir el pensamiento computacional programando historias interactivas y videojuegos, es decir, utilizando justamente los artefactos de los que Papert hablaba.

Existe una relación entre pensamiento computacional y capacidad de utilizar las herramientas para crear, construir e inventar soluciones a los problemas y dicha unión debe reforzarse en los estudiantes, con el objetivo de mejorar su capacidad de proyección y la cooperación social para darles la posibilidad de enfrentar la realidad compleja e interactiva (NRC, 2011).

Independientemente del lenguaje utilizado, con la programación, los niños interiorizan una serie de conceptos clave que se relacionan con el pensamiento computacional y aprenden prácticas, es decir, nuevos modos de pensar y aprender (Tabla 8).

La idea de Resnick y de sus colaboradores del MIT tomó forma con Scratch, una herramienta que permite, a través de la programación, la creación de artefactos. Desde su visión, el pensamiento computacional tiene una fuerte

relación con el *coding* (Resnick et al., 2009) ya que con la programación es posible describir procedimientos efectivos para la resolución de problemas. El proyecto del *Lifelong Kindergarten Group* se diseñó para dar a los niños, jóvenes y también a los adultos, la posibilidad de expresar su creatividad a través de un lenguaje visual de programación fácil de usar, pero con mucho potencial. Quien usa Scratch logra, de manera divertida y creativa, trabajar en un ambiente de programación de uso fácil, a través de instrucciones dadas al ordenador bajo la forma de elementos gráficos con forma de bloques de colores que pueden encajar, como los ladrillos de Lego (Resnick et al., 2009) y dar vida a animaciones, presentaciones y videojuegos.

La observación que los investigadores llevaron a cabo a lo largo de los años, sobre los niños de la escuela de la infancia ha hecho surgir su idea sobre el proceso de aprendizaje, recursivo, creativo y colaborativo.

Scratch se lanzó en el 2005 y modelado en base a los aspectos del ambiente con logotipo de Papert (Brennan & Resnick, 2012). El actualmente popular ambiente Scratch, "*a low floor and a high ceiling*" se diseñó de manera que, quien lo utilice, se encuentre con un ambiente de programación simple pero, al mismo tiempo, tenga en sus manos un instrumento potente que pueda satisfacer las exigencias de todos, incluso de programadores más expertos.

Tabla 8

Conceptos y prácticas relacionadas con el pensamiento computacional.

Dimensiones clave del pensamiento computacional	Conceptos	Definiciones
Conceptos (Scratch)	Secuencias (Sequences)	Serie consecutiva de pasos ordenados que puede una computadora llevar a cabo para llegar a la solución de un problema
	Ciclos (loops)	Mecanismos que permiten ejecutar varias veces las mismas instrucciones de forma interactiva
	Eventos (<i>events</i>):	Ejecutar una acción en el momento en el que ocurre otra
	Paralelismo (<i>parallelism</i>)	Hacer que distintas acciones se ejecuten al mismo tiempo

	Condiciones (<i>conditionals</i>)	Posibilidad de tomar decisiones en base a determinadas condiciones
	Operadores (<i>operators</i>)	Proporcionan soporte en la escritura de expresiones lógico-matemáticas, cadenas de caracteres, números
	Datos(<i>data</i>)	Valores que pueden ser recogidos, guardados, memorizados y analizados durante la ejecución de un programa
Prácticas computacionales (Scratch)	Experimentar e interactuar (<i>experimenting and iterating</i>)	Cuando se diseña un proyecto se realizan mejoras posteriores a menudo cambiando la idea inicial de enfoque al sistema
	Evaluar y depurar (<i>testing and debugging</i>)	Detectar los problemas y/o errores y resolverlos, corrigiéndolos.
	Reutilizar y remezclar (<i>reusing and remix</i>)	Hacer que partes de soluciones puedan utilizarse para resolver el mismo problema o problemas parecidos utilizándolas como punto de partida para nuevas ideas, incluso más complejas.
	Abstracción y modularización (<i>abstracting and modularizing</i>)	Procesos que permiten reducir la complejidad de un problema; descuidando los detalles inútiles y de esta manera concentrándose en los aspectos importantes para la resolución. Además, dividiendo el problema (modularización) en partes más simples y de fácil solución, se llega a la solución completa.
Perspectivas computacionales (Scratch)	Expresión (<i>expressing</i>)	Los jóvenes se dan cuenta de que el cálculo computacional es un medio para crear.
	Conexión (<i>connecting</i>)	Reconocen la gran fuerza de poder crear para y con los demás.
	Interrogación (<i>questioning</i>)	Se sienten capaces de responder a preguntas relacionadas al mundo real.

Nota: Modelo Brennan & Resnick (2012).

2.4 Etimología y orientaciones teóricas

2.4.1 En la literatura

El artículo de Wing del 2006 representó para muchos investigadores el lógico punto de partida para llevar a cabo un análisis crítico del concepto de pensamiento computacional; de manera particular en el contexto educativo K-12. Su redacción, creada para contrarrestar los problemas del tiempo relacionados con la difusión de la informática en el mundo de la educación capturó la atención de una amplia comunidad académica (Grover & Pea, 2013) que se dedicó bastante a la elaboración de una definición compartida, de tal

forma que incluso Wing ha propuesto, a través de los años, varias actualizaciones en comparación con la dada en el 2006 (Kaleliouglu et al., 2016).

En febrero del 2009, la National Research Council (NRC) en Washington, D.C., organizó un taller sobre el pensamiento computacional en el que participaron informáticos, técnicos de la información, expertos del sector, investigadores en el campo de la educación, que pusieron en discusión la necesidad de evaluar el valor pedagógico del pensamiento computacional para introducirlo dentro los programas de enseñanza. En los talleres organizados tanto en el 2010 como en el 2011, los participantes se concentraron en la revisión y la ampliación de los conceptos de pensamiento procesal y programación, considerando que es necesario ir más allá de los conceptos propios de la programación como tal (National Research Council, 2010; Kong & Abelson, 2019). Se resaltó la diferencia entre los términos relacionados con la informática (Traducción italiana de Computer Science (CS)) como *computational thinking*, *computer literacy*, *computer programming* y *computer application* y se amplió la visión de la noción del pensamiento computacional hasta convertirlo en un contenedor de conceptos propios de la informática, como la abstracción y la descomposición (Angeli et al., 2016).

Otras organizaciones que se ocupan de educación como The International Society for Technology in Education (ISTE) y Computer Science Teachers Association (CSTA) han definido el pensamiento computacional como un conjunto de habilidades que llevan a saber pensar de forma algorítmica, lógica y creativa y a lograr adquirir capacidades para la resolución de problemas (ISTE, 2011). En su definición operativa, el pensamiento computacional es un proceso que permite resolver problemas que incluye, pero no limita, algunas características permitiendo a quien lo practica:

- Formular problemas a través del uso de herramientas particulares, como el ordenador;

- Organizar y analizar datos;
- Utilizar la abstracción para representar datos;
- Usar el pensamiento algorítmico con pasos ordenados para automatizar las soluciones;
- Optimizar el proceso de resolución de un problema;
- Generalizar y transferir la solución de un problema a otros problemas distintos.

Dichas habilidades requieren una serie de habilidades transversales necesarias, como la capacidad de enfrentarse a problemas abiertos y de comunicarse con los demás con el fin de obtener una solución común, la obstinación y la confianza cuando se trabaja en un proyecto complejo (ISTE, 2019).

Otra importante asociación científica, la Royal Society, en 2012, definió el pensamiento computacional como “un proceso de reconocimiento de factores computacionales que se encuentran en el mundo que nos rodea, y la aplicación de instrumentos y técnicas, que derivan de la Computer Science (CS), para comprender y razonar sobre los sistemas y los procesos naturales y artificiales” (Furber, 2012).

La International Association for the Evaluation of Educational Achievement (IEA) -asociación sin ánimo de lucro -, promueve, desde hace varios años, campañas internacionales de investigación en varios continentes para comprender, evaluar y mejorar la educación con el objetivo de promover una mejora en la calidad de la educación en todo el mundo. El objeto de investigación ha sido, en particular, la alfabetización informática, el pensamiento crítico, las habilidades técnicas y las capacidades comunicativas aplicadas a determinados contextos, analizando una muestra de 46.000 estudiantes, más de

26.000 docentes y aproximadamente 2.200 escuelas de etapas diferentes y de 12 distintos países. Algunos de los resultados se presentaron y se pusieron en discusión en el International Computer and Information Literacy Study (ICILS) 2018 del IEA que, después de las indicaciones que surgieron en el precedente ICILS 2013 dirigió la atención al pensamiento computacional, considerándolo un elemento para tener en cuenta para la educación del siglo XXI (Fraillon et al., 2019).

Durante el ICILS 2018 el pensamiento computacional se definió como una capacidad individual para reconocer los problemas del mundo real y formularlos de manera computacional, llegando a soluciones algorítmicas que pueden elaborarse con una computadora (Fraillon et al., 2019; Kong & Abelson, 2019).

Numerosos trabajos (ver Tabla 9) han tratado de elaborar una definición compartida de pensamiento computacional y han llevado a revisiones constantes en la literatura de artículos científicos que trataran dicho argumento, a partir de las palabras clave más usadas, como *algoritmo*, *abstracción*, *descomposición*, *análisis de datos*, *recogida de datos*, *automatización*, *paralelización*, *simulación*.

Si bien se han realizado esfuerzos para definir el término aún no se ha logrado un consenso unánime sobre las distintas definiciones (Denning, 2017; Yadav et al., 2018), sin embargo, existe una aceptación general de que las habilidades del pensamiento computacional abarcan los conceptos de abstracción, pensamiento algorítmico, resolución de problemas, descomposición, generalización y depuración (Durak & Saritepeci, 2018).

Kalelioglu et al. (2016) revisaron la literatura del 2006 al 2014 y proporcionaron una definición adicional de pensamiento computacional como un conjunto de instrumentos mentales clave y de prácticas que van más allá de la informática.

En otras palabras, el pensamiento computacional es una forma de pensar

denotada y útil en casi todas las disciplinas y materias escolares como una intuición de lo que puede y no calcularse. Constantemente surgen cinco habilidades estrechamente relacionadas al término pensamiento computacional, abstracción, pensamiento algorítmico, resolución de problemas, reconocimiento de patrones, pensamiento basado en el diseño. Uno de los elementos comunes en las distintas definiciones que han surgido y que se acepta como elemento clave del pensamiento computacional es la habilidad de abstracción, que es necesaria incluso en los estudiantes más pequeños, a partir del plan de estudios K-6 (Angeli et al., 2016); por lo tanto, es la base para la redacción de los programas relativos. La abstracción se convierte en el elemento principal en la base del pensamiento computacional (Angeli et al., 2016; Wing, 2008) y permite examinar la información para seleccionar aquella que es relevante para la resolución del problema (Aho, 2012). Los conceptos de abstracción y de algoritmo son propios de la informática; escribir programas eficientes requiere abstracción y generalización: por este motivo la programación se usa con frecuencia para promover el pensamiento computacional (Shute et al., 2017) sin embargo no es su única dimensión. Está claro que quien sabe programar tiene una ventaja para conseguir las habilidades del pensamiento computacional, las cuales están estrechamente relacionadas a las de la programación y las de la resolución de problemas (Kong & Abelson, 2019; Shute et al., 2017).

2.4.2 Iniciativas en la Web

Se habla mucho de pensamiento computacional también en la web. Corporaciones como Microsoft o Google han buscado contribuir a la propuesta de prácticas computacionales enfocadas al mundo educativo y a los usuarios del panorama tecnológico. Viendo así la luz diversos proyectos con el fin de difundir el pensamiento computacional.

Desde el 2005, Google ha realizado una serie de inversiones destinadas a cerrar la brecha educativa, ofreciendo las habilidades de los voluntarios *googlers* a un número cada vez mayor de personas con el objetivo de dar acceso a las

ventajas de la tecnología tanto en las aulas escolares como fuera de ellas.

Code with Google, por ejemplo, tiene como objetivo garantizar que cada estudiante pueda adquirir habilidades de colaboración, de codifica y técnicas que abren oportunidades en el contexto escolar y fuera de él, independientemente de los objetivos futuros establecidos.

El empeño de Google continúa también a través de *Grow with Google* que proporciona la certificación *Python Professional* a los estudiantes de todo el mundo que desean comenzar su carrera en informática a través de una oferta formativa que busca proporcionar nuevas habilidades que son requeridas en el mercado del trabajo (Van Kleef Conley, 2020).

Google For Education (GFE) define el pensamiento computacional como un proceso de resolución de problemas, esencial para el desarrollo de aplicaciones informáticas pero útil para resolver problemas que afectan varias materias de estudio, desde las matemáticas hasta las ciencias humanas. El pensamiento computacional es un proceso de resolución de problemas que incluye una serie de características, como ordenar y analizar lógicamente los datos y crear soluciones utilizando una serie de pasos ordenados (o algoritmos), y disposiciones, como la capacidad de afrontar con confianza la complejidad y los problemas abiertos. El pensamiento computacional, fundamental para favorecer el desarrollo de las materias informáticas, encuentra aplicación en la resolución de problemas también en otras disciplinas, desde las matemáticas hasta las ciencias humanas. Los estudiantes que lo aprenden descubren la unión y relación entre las distintas materias y entre lo que es objeto de estudio en clase y lo que es parte del mundo exterior.

GFE dedica una amplia sección al concepto de pensamiento computacional y proporciona lecciones de soporte, video y muchos otros recursos para sostener a quien, como los profesores y los administradores, desean integrarlo en la práctica de la enseñanza y del aprendizaje. Los elementos que lo caracterizan se

pueden sintetizar en tres fases cognitivas:

- la descomposición (*decomposition*) es decir la habilidad de dividir una tarea en problemas más pequeños para volverla comprensible a otra persona o a otra máquina;
- el reconocimiento de patrones (*pattern recognition*) que consiste en saber distinguir las semejanzas y diferencias comunes que permiten hacer previsiones sobre la resolución de un problema; a esto se agrega la habilidad de filtrar; de desechar la información que no es útil y generalizar aquella que es necesaria (abstracción). Si se elabora una idea o un problema generalizándolo, es posible usarlo como base para resolver otros;
- el diseño de algoritmos (*algorithmic design*) desarrolla la habilidad de poner en práctica una estrategia paso a paso para resolver un problema y depende de la adquisición de las habilidades descritas anteriormente.

Como ya se ha mencionado, actualmente, no existe una definición ampliamente compartida, lo cual conlleva discrepancias sobre como el pensamiento computacional debe integrarse en los planes de estudios escolares (Barr & Stephenson, 2011, Brennan & Resnick, 2012, Grover & Pea, 2013) así como una notable dificultad en el análisis de los estudios efectuados al respecto y en la comparación de los resultados obtenidos de los mismos.

En la literatura son muchas las definiciones y frecuentemente son contextualizadas y por lo tanto no se pueden generalizar, como lo muestra la Tabla 9. Para reforzar este punto débil, es necesario que todos los investigadores comiencen a unir sus conocimientos sobre el pensamiento computacional (Shute et al., 2017).

Tabla 9

Definición del término pensamiento computacional “.

Año	Autores	Definición del pensamiento computacional (PC)
2010	Cuny, Snyder, Wing	Es el conjunto de procesos del pensamiento que se involucran en la formulación de los problemas y de su resolución de manera tal que las soluciones sean representadas de forma que puedan realizarse a través de un agente elaborador de la información.
2015	Israel, Pearson, Tapia et al.	Está relacionado con los estudiantes que usan la computadora para dar forma a sus ideas y desarrollar programas.
2015	Berland, Wilensky	Capacidad de pensar con la computadora como instrumento; prefieren hablar de perspectivas computacionales en sustitución del PC para resaltar que el PC puede estar limitado por los contextos.
2016	Kaleliouglu et al.	Compleja capacidad de pensamiento de alto nivel que se involucra en los procesos de resolución de problemas.
2016	Gretter & Jadav	El PC es similar al concepto de Alfabetización en Medios e Información (MIL) como apoyo de las capacidades y de la ciudadanía de los estudiantes.
2017	Shute, Sun, Asbell-Clarke	Es la base conceptual necesaria para resolver los problemas de manera eficaz y eficiente, es decir de manera algorítmica (con o sin el uso de la computadora) con soluciones que pueden utilizarse varias veces en distintos contextos.
2018	ICILS	Es la capacidad individual de reconocer los aspectos de los problemas del mundo real para formularlos de manera computacional y para desarrollarlos con soluciones algorítmicas que pueden ser elaboradas por una computadora.

Fuente: elaboración propia.

En conclusión, actualmente, es posible encontrar en el panorama científico tres tipos de definiciones:

- Aquellas que hacen referencia a los trabajos de Wing (2011) y Aho (2012) que se concentran principalmente en el proceso de resolución de problemas para intentos posteriores mediante el uso de algoritmos. Por lo tanto, estas definiciones pueden relacionarse con habilidades universalmente aplicables como el saber leer, el saber escribir y la contabilidad.

- El segundo tipo de definiciones hace referencia a la habilidad de saber formular y resolver problemas, de organizar, analizar e implementar según lo propuesto por la CSTA (2011).
- El tercer grupo de definiciones es de naturaleza más educativa y se refiere al cuadro propuesto por Brennan y Resnick que han visto el uso de Scratch como un instrumento válido para el desarrollo del pensamiento computacional.

2.5 El pensamiento computacional en la educación K-12

Cuando se habla de incluir el pensamiento computacional en la escuela con frecuencia se piensa que es una habilidad exclusiva que se desarrolla en el campo de la ingeniería informática y del cálculo, sin embargo esto no es así (Roig-Vila & Moreno-Isac, 2020). Se trata de una habilidad compleja ligada al pensamiento abstracto-matemático y a la ingeniería pragmática pero también se aplica en muchos aspectos de la vida cotidiana (Berrocoso et al., 2015) y por lo tanto no debería ser sinónimo de la capacidad de programar una computadora.

Aprender a nivel escolar los elementos principales que son la base del pensamiento computacional puede permitir a los estudiantes, desde la escuela infantil hasta el final de la escuela superior (K-12), prepararse para vivir en un mundo en el cual la tecnología digital condiciona fuertemente el desarrollo de actividades esenciales para la vida cotidiana (Angeli et al., 2016; Kong & Abelson, 2019). Así pues, no es necesario esperar a que los estudiantes inicien carreras universitarias para introducirlos a la comprensión de conceptos de cálculo, de resolución de problemas, a trabajar con algoritmos y con métodos computacionales (Barr & Stephenson, 2011).

El pensamiento computacional debe ser una habilidad base en todo el plan de estudios escolar que debe formar ciudadanos conscientes, que sepan usar una computadora y crear instrumentos computacionales y no ser simplemente consumidores. Introducir dicho concepto en el complejo mundo de la Educación

K-12 es una tarea muy difícil que ha requerido la búsqueda de los mejores y más adecuados enfoques educativos para promoverlo entre los jóvenes estudiantes.

La literatura muestra que existe una relación entre programación informática y pensamiento computacional; en concreto, la informática puede ayudar al alumno a comprender y resolver los problemas a través del uso de algoritmos, cálculo y manipulación de datos (Barr & Stephenson, 2011).

Según diversos autores, la informática, ofrece la posibilidad de crear nuevos artefactos y aplicar sus conceptos a nivel multidisciplinario. A la problemática educativa se suma la económica, aunada al hecho de que los cambios tecnológicos que se han dado en la sociedad no se han verificado con la misma intensidad en el panorama escolar. No obstante, la gran demanda del mundo del trabajo de personal con habilidades y capacidades relacionadas con la informática, los sistemas escolares no le han dado la misma importancia (Angeli, 2016), incluso si, después de treinta años, parece regresar al centro de las políticas educativas. No todos aceptan la idea de que la programación involucre a los estudiantes, a través de la construcción de artefactos, en el aprendizaje de conceptos relacionados con el pensamiento computacional.

La ISTE y el NRC consideran que es posible separar el concepto de pensamiento computacional del concepto de programación y la literatura muestra estos dos enfoques diferentes. Los beneficios del aprendizaje de las habilidades del pensamiento computacional no se limitan únicamente al dominio de la alfabetización digital.

Son muchos quienes han estudiado como definir la naturaleza del pensamiento computacional en el contexto escolar K-12 y como adaptarlo a las distintas actividades de aprendizaje, o en general, como revisar los programas de las distintas disciplinas a la luz de la necesidad de relacionarlos con el pensamiento computacional. Claramente, se trata de un cambio que requiere intervención en varios frentes. Por un lado es necesario involucrar a los

legisladores en lo que se refiere a las políticas educativas sobre la difusión e integración del pensamiento computacional en los programas escolares y por otro lado es fundamental formar a los docentes que deben integrar a su conocimiento estos nuevos conceptos, a partir de su conocimiento pedagógico para luego transferirlo a la práctica en clase (Barr & Stephenson, 2011; Labusch & Eickelmann, 2017; Hsu et al., 2018).

En el pasado, transferir las habilidades relacionadas con el pensamiento computacional era un proceso que se refería a algunas disciplinas específicas, como las del área STEM por ejemplo, y para fases de edad determinadas, ahora su interdisciplinariedad se ha reconocido en todos los niveles escolares (Kalelioglu, 2016). El problema que se ha encontrado es el que aun reconociendo el potencial de los contenidos de los programas escolares que pudiesen abordarse con el pensamiento computacional, el docente tiene pocas posibilidades de adaptar los nuevos modos de pensar a la programación escolar (Shute et al., 2017).

Por este motivo surge la necesidad de discutir algunos elementos clave y de cuestionarse lo siguiente (Angeli et al., 2016; Bocconi et al, 2016; Kalelioglu, 2015; Pugnali et al., 2017; Yadav et al., 2018):

- a) La formación docente y los métodos de enseñanza: ¿qué conocimientos deben tener los docentes para enseñar pensamiento computacional?
- b) Las habilidades y las capacidades en los estudiantes: ¿qué habilidades debe promover un plan de estudios?
- c) La evaluación de las habilidades en la escuela y en la vida real: ¿cómo evaluar la adquisición de las habilidades del pensamiento computacional?y, una vez adquiridas, ¿cómo pueden utilizarse en las situaciones de la vida real?

- d) Saber seleccionar instrumentos y programas escolares: ¿De dónde partir para innovar el plan de estudios?

Surgen también otras preguntas:

- ¿En qué términos se puede definir el pensamiento computacional como habilidad clave del siglo XXI para los estudiantes?
- ¿Cuáles son los conceptos y las habilidades clave principales que caracterizan el pensamiento computacional?
- El pensamiento computacional, ¿en qué relación entra con la programación en la escuela obligatoria?
- ¿Qué relación hay entre el pensamiento computacional y la alfabetización digital/competencia digital?
- ¿Qué tipo de capacitación deben llevar a cabo los docentes para ser capaces de integrar el pensamiento computacional al momento de practicar la enseñanza? ¿Es posible promover el pensamiento computacional desde una disciplina en concreto o integrándolo en materias STEM como argumento extracurricular?
- ¿Qué significa evaluar el pensamiento computacional?
- ¿Qué es necesario hacer para promover el pensamiento computacional en las escuelas?

Desde el 2010 se han buscado las estrategias y los recursos que deben desarrollarse para implementar el pensamiento computacional en el mundo escolar (ISTE, 2019). Actualmente es posible identificar una gran variedad de iniciativas que sirven como estímulo para formular programas educativos

orientados a su difusión en diversas partes del mundo. Sin embargo, sigue abierta la discusión sobre la necesidad de aumentar las investigaciones pedagógicas en cuanto al concepto de pensamiento computacional para ayudar a los estudiantes desde pequeños a convertirse en productores de tecnologías y no solo en consumidores (Kong & Abelson, 2019).

2.6 Modelos de desarrollo del pensamiento computacional

A continuación, se presentan dos modelos de desarrollo del pensamiento computacional, aplicados respectivamente en los Estados Unidos, donde la informática (Computer Science, CS) no es considerada independiente a las demás disciplinas (modelo Scratch) y en el Reino Unido donde, por el contrario, la enseñanza de la informática se ha vuelto obligatoria (modelo CAS).

2.6.1 Modelo Scratch: USA

En los Estados Unidos, a partir de la colaboración entre investigadores de la Harvard Graduate School of Education y el MIT Media Lab de Boston nació un modelo de desarrollo del pensamiento computacional. Dirigieron su interés hacia las posibles modalidades de desarrollo de las actividades de aprendizaje basadas en el diseño, en particular la programación de medios interactivos, dirigida también al mundo K-12.

El modelo se basa en el uso de Scratch, un ambiente de programación que permite crear historias, juegos y simulaciones para compartir en una comunidad online con otros jóvenes programadores (Resnick et al., 2009) y que sigue la definición de pensamiento computacional propuesta por Brennan y Resnick (2012) en la cual se hace referencia a las tres dimensiones clave ya mencionadas anteriormente (conceptos, prácticas computacionales y perspectivas computacionales).

Además, los investigadores estadounidenses se concentraron en identificar las estrategias que pueden consentir la evaluación del desarrollo del

pensamiento computacional, basándose en tres enfoques:

- a través de las Entrevistas basadas en artefactos se proporciona un protocolo de preguntas que se lleva a cabo en distintas fases del aprendizaje, a partir de la primera experiencia con Scratch pasando por el momento principal de la experiencia hasta llegar al final de la actividad. Las preguntas hacen referencia a distintas categorías y tienen el objetivo de evaluar las principales prácticas computacionales (experimentar y reiterar, probar y depurar, reutilizar y remezclar, abstraer y modular) a través de indicadores de nivel de competencia (bajo, medio, alto);
- los escenarios de diseño que proporcionan un enfoque según el cual a los estudiantes se les presentan algunos escenarios de diseño en forma de proyectos creados por otros scratchers; se les pide a los estudiantes seleccionar uno de los proyectos, describirlo, corregir los posibles errores y remezclarlo;
- la documentación del estudiante según la cual se les pide a los estudiantes un trabajo reflexivo guiado (por ejemplo a través de la guía al plan de estudios de *Creative Computing*).

Uno de los principales objetivos que caracterizan el modelo propuesto es el de dar soporte al desarrollo del pensamiento computacional, con la condición de no influenciar la libertad de programación; para tal fin el modelo ofrece algunos recursos a los cuales poder recurrir entre la guía al plan de estudios de *Creative Computing* (Harvard Graduate School of Education, s.f.) que es un conjunto de ideas, estrategias y actividades con las cuales poder usar Scratch, con el objetivo de dar mayor familiaridad y soltura a la creatividad y al pensamiento computacional. Las actividades de esta guía están diseñadas para apoyar la familiaridad y aumentar la fluidez con todo lo relacionado con el pensamiento computacional, fomentan la exploración de conceptos clave como la secuencia, los bucles, el paralelismo, los eventos, los condicionales, los operadores, los

datos y de las prácticas claves del pensamiento computacional (la experimentación, la iteración, las pruebas, la depuración, el reutilizo y la remezcla, la abstracción y la modulación).

El modelo Scratch busca potenciar el pensamiento computacional en los jóvenes a través del uso de la programación, aun limitándose a los conceptos propios del *coding* (Shute et al., 2017) permite introducir a los jóvenes estudiantes al desarrollo de las habilidades requeridas en la actualidad. Por dicho motivo, se eligió como modelo de referencia para la planificación y el diseño de la experimentación a la que se sometió la clase de cuarto curso de Educación Primaria, protagonista de la presente investigación.

2.6.2 Modelo CAS: Computing at School (Reino Unido)

En el Reino Unido, la difusión del pensamiento computacional ha sido promovida en los últimos años, por el *Computing At School (CAS)*, una asociación que ha involucrado al Ministerio de la Educación Pública del Reino Unido, varias Universidades y empresas que se encargan de informática y tecnología, entre las cuales la grande Microsoft.

El CAS nació con motivo de la planificación y realización del nuevo plan de estudio de Computer Science en todas las fases de formación (K-12); el objetivo en común es el de ser un apoyo a los docentes que desean participar en la integración del plan de estudios en el aula, creando también recursos para apoyar la investigación existente, a través de la colaboración y del intercambio de experiencias.

Desde *CAS Barefoot* es posible leer la definición de pensamiento computacional:

El pensamiento computacional enfrenta los problemas de tal manera que para su resolución es posible ser ayudados por la computadora. El pensamiento computacional no piensa a las computadoras sino que piensa como una computadora. Las

computadoras no piensan por sí solas, ial menos por ahora!
(C.A.S. online, 2018).

El modelo conceptual utilizado para la integración del pensamiento computacional en el aula se basa en seis conceptos principales:

- la lógica (*logic*) vinculada a conceptos de predicción y análisis. Las computadoras no tienen poder, cumplen acciones a través de un comando, por lo tanto son predecibles. Por lo tanto, a través del razonamiento lógico es posible establecer que cosa hará una computadora después de un comando;
- los algoritmos (*algorithms*) están relacionados con la atribución de reglas precisas. Son una secuencia de instrucciones necesarias para realizar algo y se escriben de manera que puedan sean comprendidos por los humanos;
- la descomposición (*decomposition*) es el truco para resolver los problemas complejos pues permite fraccionar un problema en pequeñas partes que son mucho más fáciles de abordar. Este concepto es aplicable al trabajo en equipo, en el que cada componente aporta una contribución en términos de conocimientos, experiencias y habilidades durante la ejecución de una tarea;
- los esquemas (*patterns*) permiten detectar y establecer similitudes así como lo hacen los informáticos que buscan un modelo común en un problema para crear una sola solución y poderla utilizar varias veces. De esta manera diseñan el módulo solo una vez y trabajan con una sola versión;
- la abstracción (*abstraction*) es el núcleo fundador del pensamiento computacional y se refiere a la simplificación de las cosas, es decir, decidir

qué detalles es posible ignorar y cuáles considerar para resolver un problema complejo. Sirve en la vida y también en la escuela. Es visto como una potente forma de pensar y por lo tanto se ha impartido a los estudiantes de Educación Primaria;

- la evaluación sistemática (*evaluation*) deriva de la consideración de que diariamente se realizan juicios sobre aquello que se hace y aquello que se piensa en base a una variedad de factores y criterios. La evaluación consiste, cuando es posible, en hacer juicios de forma objetiva y sistemática, y en Informática este proceso permite evaluar la calidad, la eficacia y la eficiencia de soluciones, sistemas, productos y procesos. En la escuela, la práctica de la evaluación es bastante común: los alumnos evalúan su trabajo, los docentes evalúan las lecciones que van a impartir y el aprendizaje.

A los conceptos mencionados anteriormente se agregan también cinco enfoques:

- la experimentación (*tinkering*) que permite aprender explorando y experimentando; es la manera natural de aprender intentando hacer las cosas de diferentes maneras y de forma independiente. El *tinkering* está asociado al razonamiento lógico, a la oportunidad de explorar y cuestionar. En la escuela es necesario alentar a los alumnos a experimentar, dándoles el tiempo de orientar su propia exploración, haciéndoles preguntas abiertas y proporcionándoles estímulos para descubrir algo más;
- la depuración (*debugging*) es el proceso de investigación de los errores en un algoritmo de un programa para hacerlo funcionar, puede llevarse a cabo en la vida real, cuando se buscan los errores que se han cometido y se remedia corrigiéndolos. En la escuela es posible proporcionar algunas estrategias de depuración que los alumnos pueden usar para cualquier error de programación a través del razonamiento lógico que lleva a

pensar que cosa pudiera suceder, descubrir exactamente qué cosa sucede para entender dónde las cosas no están funcionando y corregir los errores;

- la creatividad (*creating*) tiene que ver con la planificación, la realización y la evaluación de las cosas e involucra la originalidad y la creación de algo útil. Crear cosas para los demás como instrumento para aprender, se vuelve un proceso que permite desarrollar en los alumnos confianza, competencia e independencia. En el ámbito escolar, a partir de Educación Primaria, es posible iniciar a los niños en la producción de contenidos de calidad utilizando, por ejemplo, la tecnología digital, con el fin de hacer madurar en ellos las habilidades del pensamiento computacional;
- la perseverancia (*persevering*) cuando se enfrentan problemas complejos es necesario tener mucha paciencia, tolerancia y resistencia. Para llegar a ser expertos en lo que se hace es necesario perseverar y practicar, la habilidad innata por sí sola no es suficiente. Los alumnos que perseveran tienen mayores posibilidades de encontrar alternativas nuevas e insólitas, de utilizar estrategias para resolver las dificultades que se presentan. Por este motivo, en la escuela, es necesario alentar a los alumnos a perseverar a través de, por ejemplo, la resolución de problemas (*problem solving*);
- la colaboración (*collaborating*), es decir, el trabajo compartido con los demás que lleva a mejores resultados. La colaboración estimula la motivación a continuar desarrollando tareas que pueden parecer confusas o incluso imposibles cuando se trabaja en ellas sin ayuda. El trabajo de grupo, actualmente, aplicado en múltiples situaciones incluso en el mundo escolar, permite desarrollar capacidades lógicas y de perseverancia. Los niños que tienen la oportunidad de trabajar en grupo desarrollan confianza y conciencia de sí mismos, aprenden a controlar los sentimientos y las relaciones con los demás.

El CAS tiene la tarea de formar a los profesores ingleses, gracias al apoyo financiero del *Department of Education*. Para este fin CAS ha establecido una red de excelencia para la enseñanza de la Informática que da soporte y capacitación a los docentes (*master teacher*) en la implementación de los programas en sus clases.

2.7 Ejemplos de difusión del pensamiento computacional en el ámbito educativo

La idea de que el pensamiento computacional debe integrarse a la escuela ha estimulado la realización de múltiples estudios que tienen como objetivo identificar las experiencias y los recursos que pueden aplicarse en contextos escolares, a partir de la escuela obligatoria (Grover & Pea, 2015; Angeli & Giannakos, 2020).

En el presente trabajo se pusieron en consideración tres modelos, dos de los cuales fueron utilizados en USA (Barr et al, 2011; Barr & Stephenson, 2011), y el enfoque propuesto en Reino Unido por la comunidad *Computing At School* (CAS) del cual se ha hablado anteriormente. En la literatura, se encuentran varios modelos que han sido propuestos y utilizados en distintas partes del mundo, algunos consideran el pensamiento computacional transversal a muchas disciplinas, en cambio otros, tienen la predisposición de encuadrarlo en el plan de estudios K-12 de informática (CS). En este último caso, las trayectorias diseñadas dentro de la disciplina de la informática surgen de la necesidad de llenar la visión incompleta que tienen los jóvenes al respecto y que los lleva a tomar determinadas decisiones educativas por desinformación. De manera intencional los planes de estudio *K-12 CS* se han establecido para dar mayor conciencia a los jóvenes estudiantes sobre el tema y sobre cómo se refleja en el mundo social (Grover & Pea, 2015).

En Reino Unido, por ejemplo, se formuló un plan de estudios informático para todas las escuelas, de primaria hasta la secundaria, tomando como

referencia la carta política de la Royal Society (2012) que solicita difundir la informática a través de la educación (Grover & Pea, 2015).

2.7.1 El modelo de Barr y Stephenson

Barr y Stephenson (2011) abordan el problema de la búsqueda de la mejor definición del concepto de pensamiento computacional y sobre cómo se debe llevar a la clase. Según su punto de vista, la fortaleza del pensamiento computacional es que se puede aplicar a casi cualquier tipo de razonamiento, pero el proceso que lleva involucrar a los estudiantes

El K-12 es complejo y requiere un cambio en el sistema que incluya a los estudiantes, así como a la comunidad educativa informática (Barr & Stephenson, 2011).

Son muchas las disciplinas que promueven las habilidades del *problem solving*, pensamiento lógico y algorítmico; el informático puede ayudar a transferir los procesos computacionales en la resolución de problemas de otra índole y que involucran otras disciplinas. Los autores proponen su propia definición de pensamiento computacional:

“El pensamiento computacional es un enfoque de resolución de problemas que puede implementarse a través de una computadora” (Barr & Stephenson, 2011).

Los estudiantes no son solo usuarios sino constructores de instrumentos; ellos utilizan un conjunto de conceptos, como la abstracción, recursión e iteración, para procesar y analizar datos creando artefactos reales y virtuales.

“El pensamiento computacional es una metodología de *problem solving* que puede automatizarse, transferirse y aplicarse entre los sujetos” (Barr & Stephenson, 2011).

Barr y Stephenson (2011), por su parte, proponen un modelo de integración transversal de conceptos que son la base del pensamiento computacional en el plan de estudios escolar, con particular referencia a algunas disciplinas como la informática, las matemáticas, las ciencias, las ciencias sociales y las artes lingüísticas.

La implementación de las prácticas computacionales en el aula conlleva el uso de estrategias que puedan facilitar la introducción a los conceptos base. Asume particular importancia la introducción por parte de los educadores de un adecuado léxico computacional y la toma de conciencia de que los errores deben verse como un recurso para llegar a la solución. Además, el trabajo de grupo permite aprender de forma más fácil y mejora el ambiente en clase, la solidaridad grupal en torno a una idea.

Incluso reconociendo la validez del modelo, un punto débil por el que es criticado es que no se muestran claramente qué aspectos relacionados con el pensamiento computacional se encuentran dentro de las disciplinas y los ejemplos proporcionados resultan bastante genéricos para poder ser utilizados por los profesores (Shute et al., 2017).

2.7.2 Los recursos de C.S.T.A.

La Computer Science Teachers Association (CSTA), asociación que da soporte y promueve la enseñanza de la informática a los estudiantes K-12, ya que en USA no existe como materia obligatoria y continua en los planes de estudio. Dicha organización ofrece a los profesores y a los estudiantes la oportunidad de prepararse de manera eficaz, a través de la informática a la enseñanza y al aprendizaje respectivamente. Su misión es responsabilizar e involucrar a los enseñantes K-12 de todo el mundo. Ha diseñado algunos estándares curriculares que pueden adaptarse, de manera transversal, a una buena parte de las disciplinas que el estudiante profundiza durante su carrera escolar y que han sido escritas por los educadores para ser coherentes y

comprensibles para los profesores, administradores y responsables políticos.

Por lo tanto, la CSTA, alienta a las escuelas, distritos y Estados a alinear sus planes de estudio a los estándares *CSTA K-12 Computer Science (CS)* a través de la aprobación de un grupo de trabajo que evalúa si un contenido, producto o plan de estudios está alienado a los estándares que se han organizado y dividido en base al nivel de educación en cuatro niveles divididos por grupos de edad (CSTA, 2017):

- Nivel1A (de los 5 años a 7 años);
- Nivel1B (de los 8 años a los 11 años);
- Nivel2 (de los 11 años a los 14 años);
- Nivel3A (de los 14 años a los 16 años);

Para cada nivel se definen siete prácticas de referencia que se relacionan entre sí y que requieren promover una cultura informática inclusiva donde sea posible saber reconocer los problemas computacionales, desarrollándolos a través de la abstracción y la creación de artefactos computacionales y que se aplican en el desarrollo de actividades asociadas a cinco familias de conceptos a las que se asocian los respectivos subconceptos (Tabla 10).

El modelo propuesto por la CSTA muestra una clara referencia al pensamiento computacional y una evidente diferenciación de la programación informática. En su definición se observa que la esencia del pensamiento computacional consiste en el combinar datos e ideas para resolver problemas, utilizando las tecnologías. Las actividades didácticas deben planearse en modo tal que permitan a los estudiantes descubrir y explicar las relaciones científicas, predecir eventos y adquirir habilidades de procedimiento que les permitan entender mejor los temas propuestos.

Los CSTA K-12 Computer Science Standards definen un conjunto de objetivos de aprendizaje diseñados para proporcionar las bases para un plan de estudios completo de informática y para garantizar su implementación a nivel K-12. Para este fin, los estándares CSTA se proponen introducir los conceptos fundamentales de la informática a todos los estudiantes, a partir del nivel elemental, alentando a las escuelas a ofrecer cursos adicionales de informática a nivel secundario que permitan a los estudiantes interesados estudiar los aspectos de la ciencia informática de manera más profunda y prepararlos para su ingreso en el mundo del trabajo o en la universidad.

Tabla 10

Conceptos y subconceptos del modelo CSTA.

CONCEPTOS	SUB CONCEPTOS
SISTEMAS INFORMÁTICOS	Dispositivos
	Hardware y Software
	Resolución de problemas
REDES E INTERNET	Comunicación de redes y organización
	Cybersecurity
DATOS Y ANÁLISIS	Conservación
	Colección, Visualización y Transformación
	Inferencia y Modelos
ALGORITMOS Y PROGRAMACIÓN	Algoritmos
	Variables
	Control
	Modularidad
IMPACTO DE LA INFORMÁTICA	Desarrollo de programas
	Cultura
	Interacciones sociales
	Seguridad, Leyes y Ética

Fuente: elaboración propia.

2.7.3 Los recursos del modelo CAS

Este modelo se ha elaborado en el Reino Unido donde la informática como ya se ha dicho, se ha incluido en el plan de estudios como materia obligatoria para todos los niveles de educación (K-12) e incluye seis trayectorias distintas que pueden ser propuestas a los estudiantes:

- Algoritmos;
- Programación y Desarrollo;
- Datos y su representación;
- Hardware y su elaboración;
- Comunicaciones y redes;
- Tecnología de la Información.

El concepto de pensamiento computacional se encuentra al interno, ya que cada trayectoria puede asociarse una característica propia del pensamiento computacional, una *oportunidad* que se le puede atribuir (*pensamiento algorítmico, abstracción, descomposición, evaluación, generalización*).

El modelo CAS está relacionado con un proyecto específico que busca desarrollar una estrategia de éxito que pueda inserir la Informática como materia en el plan de estudios nacional inglés. Para este fin se ha preparado una guía que dirige, desarrolla y planifica programas de estudio en el ambiente escolar (Manilla et al., 2014).

Según este modelo "el pensamiento computacional" es la capacidad de:

- Desarrollar abstracciones computacionales de problemas del mundo real;
- Diseñar los artefactos computacionales y posteriormente desarrollarlos y perfeccionarlos a través del razonamiento lógico.

Por lo tanto, en Educación Primaria, los estudiantes deben ser capaces de utilizar el razonamiento lógico para predecir el comportamiento de programas simples. No es importante solo saber escribir un programa, sino también a lograr

ejecutar con la propia mente aquello que está escrito (Brown et al., 2013).

2.8 Evaluar el pensamiento computacional

La búsqueda de las estrategias necesarias para evaluar los efectos de las actividades y de las trayectorias relacionadas al pensamiento computacional está relacionada a la práctica según la cual los docentes, en el contexto escolar, utilizan la evaluación como instrumento de medida, control y adecuación a los procesos de enseñanza y de aprendizaje. Así pues, las estrategias que buscan favorecer el desarrollo del pensamiento computacional y que son aplicadas en los planes de estudio K-12 se pueden juzgar eficientes solo si se lleva a cabo una correcta evaluación (Grover & Pea, 2013).

Al mismo tiempo, como en cualquier área de investigación es necesario tener a disposición instrumentos de medida válidos (Román-González et al., 2017). Son muchas, las investigaciones en los últimos años destinadas a identificar los instrumentos de evaluación más idóneos y una de las preguntas más frecuentes es: ¿Qué evaluar? ¿Cómo evaluar?

Ciertamente los estudiantes deben demostrar de alguna manera que han desarrollado su propio pensamiento computacional y por lo tanto, que han mejorado en la resolución de problemas en distintos contextos, como dar órdenes a las máquinas para realizar tareas similares o saber razonar sobre situaciones que se presentan en la vida cotidiana (Chen et al., 2017).

Se piensa que los alumnos de la era digital tienen buenas habilidades de pensamiento computacional pero aún no se ha encontrado prueba de ello en la literatura respecto a los niveles alcanzados de dichas habilidades (Korkmaz et al., 2017). Más bien, los trabajos que se han llevado a cabo se han concentrado en la evaluación de los artefactos creados por los estudiantes durante las actividades que se relacionan con el desarrollo del pensamiento computacional (Chen et al., 2017).

Hay un creciente interés por parte de muchos investigadores que están trabajando en la formulación de métodos de evaluación del aprendizaje de los estudiantes en el campo del pensamiento computacional en todos los niveles escolares (Korkmaz et al., 2017; Román-González et al., 2017,) pero aún no han sido validados científicamente o se encuentran en fase de validación; en particular, buscan ir más allá de las limitaciones planteadas por algunas plataformas, buscando instrumentos de medición tales como pruebas previas y posteriores aplicables a todas las plataformas de *coding* y de robótica.

Son un ejemplo (Román González et al., 2017):

- Fairy Assessment in Alice;
- Agent Sheets;
- CTP-QUIZ Instrument;
- Robótica.

Para muchos investigadores resultade gran importancia poder verificar la adquisición de determinadas habilidades y modelos de pensamiento que pueden aplicarse en el aprendizaje futuro y en la vida cotidiana. Los estudiantes deben demostrar una competencia mejorada de pensamiento computacional en la resolución de problemas en múltiples contextos (Chen et al., 2017). Actualmente aún se registra una falta de instrumentos que permitan llevar a cabo una adecuada evaluación de las habilidades de pensamiento computacional (Moreno-León et al., 2017; Tang et al., 2020). A continuación, se presentan algunos modelos de evaluación del pensamiento computacional utilizados comúnmente en Educación Primaria.

2.8.1 La evaluación con el modelo Scratch

Brennan y Resnick (2012) describen en cuanto a la evaluación, tres

enfoques útiles para evaluar el desarrollo del pensamiento computacional en los jóvenes que usan Scratch. Sostienen que los proyectos de Scratch pueden dar soporte al aprendizaje a largo plazo; su evaluación incluye análisis de capacitación a través de la evaluación de la cartera de cada usuario, entrevistas y escenarios de diseño. Proponen tres enfoques:

- **Enfoque#1:** análisis de la cartera de proyectos (*project portfolio analysis*)

Cualquiera que esté inscrito a la comunidad online de Scratch puede visualizar desde su propio perfil todas sus creaciones, los comentarios y quién lo sigue. A través de una herramienta Scrape (<http://happyanalyzing.com>) User Analysis desarrollada por investigadores del College of New Jersey, es posible analizar los bloques de programación al interno de los proyectos Scratch, verificando cuáles se han usado con mayor frecuencia y cuáles no, a través de un reporte gráfico. Sin embargo, este enfoque presenta límites ya que, incluso si permite el poder llevar a cabo un análisis formativo de manera veloz, no proporciona información sobre el proceso de desarrollo del proyecto y sobre que práctica de pensamiento computacional en particular se haya utilizado. Actualmente la comunidad de *happyanalyzing* se encuentra cerrada (Brennan & Resnick, 2012).

- **Enfoque#2:** Entrevista basada en artefactos (*artifact based interview*)

Este enfoque implica el uso de entrevistas semiestructuradas: la mayor parte de los *scratchers* vienen seleccionados con un muestreo aleatorio y se entrevistan siguiendo un protocolo, con el objetivo de evaluar tanto los conceptos como las prácticas computacionales. El punto débil de este enfoque se encuentra en el hecho de que la entrevista realizada en un proyecto ya creado con anterioridad y no implementado en tiempo real requiere activar los recuerdos que pueden estar incompletos en lugar de dar la posibilidad de conocer mejor las prácticas computacionales llevadas a

cabo en el proyecto elaborado (Brennan & Resnick, 2012).

- **Enfoque #3:** Escenarios de diseño (*design scenarios*)

En colaboración con los investigadores del Education Development Center (EDC) se probó este enfoque como instrumento de evaluación sobre un grupo de estudiantes de escuelas de distinto orden y grado. A partir de algunos proyectos escogidos ad hoc se les pidió a los estudiantes escoger uno y explicarlo, describir cómo podría extenderse, corregir algún error y remezclarlo. Los escenarios de diseño resultan particularmente útiles porque ofrecen la oportunidad de explorar y requieren fluidez, pero conducen a muchas preguntas relacionadas con la elección de tiempo a invertir para examinar los proyectos, y posibilidad de que estos puedan resultar interesantes para quién los examina (Brennan & Resnick, 2012).

Sin embargo, los mismos autores consideran que es necesario utilizar una combinación de los tres enfoques para llevar a cabo una evaluación más adecuada.

2.8.2 La evaluación con Dr. Scratch

Aplicación web gratuita y de código abierto que permite efectuar una evaluación de los proyectos Scratch en diversas áreas computacionales -el sitio está traducido a varios idiomas-. Herramienta desarrollada por el grupo de investigación GSC/LibreSoft de la Universidad Rey Juan Carlos e inspirado por Scrape (Moreno-León et al., 2017) se encuentra en continua evolución; de hecho, aunque ya ha sido utilizado por millones de estudiantes y docentes, la evaluación automática realizada por Dr. Scratch aún no se ha validado científicamente de forma completa (Moreno-León et al., 2017). A través de esta herramienta se pueden mejorar las capacidades personales de programación de manera divertida además de tener una retroalimentación que sirve para desarrollar el pensamiento computacional.

Para llevar a cabo el análisis de un proyecto realizado con Scratch, basta con buscar en el sitio web de *Dr. Scratch* y cargar un archivo con extensión sb2 o simplemente proporcionar el *url* del proyecto para poder descargarlo en la computadora. A cada proyecto se le asigna una puntuación que va de cero a veintiuno en base a la suma de los puntos que se atribuyen a siete áreas relacionadas a la adquisición de la habilidad de pensamiento computacional, a cada una de ellas se les atribuye una puntuación máxima de tres puntos con referencia a las siete áreas específicas que son la capacidad de abstracción y descomposición de los problemas, el pensamiento lógico, la presencia de procesos coordinados entre sí después de un mismo evento o eventos contemporáneos (sincronización), el paralelismo, las nociones algorítmicas de control de flujo (*flow control*), la posibilidad que tiene el usuario de interactuar y la representación de los datos. (Moreno-León et al., 2017).

Además, *Dr. Scratch* puede detectar los malos hábitos de programación o posibles errores y la cantidad de información que proporciona es distinta dependiendo el nivel de pensamiento computacional alcanzado. Al aumentar la puntuación aumenta la cantidad de información dada como retroalimentación al usuario. En base a la puntuación obtenida el proyecto es clasificado como *Basic Developing o Master*.

La herramienta *Dr. Scratch* es una evaluación de tipo formativa y resulta más bien un divertido instrumento de autoevaluación para los alumnos que es útil también para los docentes pues les ayuda en la difícil tarea de evaluar las habilidades de pensamiento computacional alcanzadas por los estudiantes. Por lo tanto, no puede considerarse como una prueba cognitiva sino como una herramienta para la evaluación formativa de los proyectos Scratch (Román-González et al., 2017) que a nivel global es el lenguaje de programación más utilizado en el campo de la Educación Primaria y Secundaria.

2.8.3 La organización *Bebras*

Bebras es una organización internacional que tiene como objetivo promover “los aspectos científicos de la informática”, se dirige a los estudiantes de las escuelas primarias y secundarias, aunque estas no posean conocimientos específicos. *Bebras* nació en Lituania en el 2003 con el fin de estimular el interés y promover la excelencia en torno a los principios de la ciencia informática desde el punto de vista del pensamiento computacional en las escuelas primarias y secundarias (Dagiene & Stupuriene, 2016).

La comunidad *Bebras* organiza dos grandes eventos:

- un taller internacional durante el cual se crean nuevos juegos que se incluirán en las competencias que se llevan a cabo en el año en curso;
- las competencias nacionales que se llevan a cabo de forma contemporánea en todos los países que participan en la iniciativa (cincuenta y siete Naciones y seis que planean participar).

Los participantes de las competencias en general son supervisados por los profesores que pueden utilizar los juegos propuestos integrándolos en sus actividades para la enseñanza, generalmente a través del uso del ordenador o de dispositivos móviles. Los juegos propuestos por *Bebras* promueven las habilidades para la resolución de problemas, las capacidades de subdividir las tareas complejas en partes más simples, el diseño y el razonamiento algorítmico, el reconocimiento de patrones, la generalización y la abstracción de los modelos.

El *Bebras International Contest* por el momento es solo un evento que tiene como objetivo el promover el pensamiento computacional y por lo tanto no puede considerarse un instrumento de medida de las habilidades asociadas a este. Sin embargo, es un evento en continua expansión y por este motivo, ha suscitado el interés de muchos investigadores de psicometría en varias partes del mundo que han comenzado a estudiarlo como posible instrumento de

medida del pensamiento computacional (Román-González et al., 2017). Así pues, por el momento representa una herramienta popular útil para evaluar el pensamiento computacional, gracias a su enfoque compuesto por desafíos cortos (Dagiene & Stupuriene, 2016).

2.9 Contribuciones a la difusión del pensamiento computacional

A través de Internet encontramos varias propuestas con las cuales es posible iniciar experiencias relacionadas con la adquisición de habilidades computacionales y que pueden ser llevadas a cabo en el mundo escolar y académico. Las herramientas son múltiples y van desde la programación visual por bloques (Scratch, Alice, Snap!) hasta los desafíos de *coding* online (Code.org, Lite-bot) y actividades unplugged (sin utilizar la computadora) (CS Unplugged) (García-Peñalvo et al., 2016). A continuación, se muestra una iniciativa que, a nivel mediático, ha recibido una gran respuesta por parte de los usuarios y en particular por los estudiantes K-12.

2.9.1 La hora del código: una campaña mundial

Para promover la cultura para la enseñanza y el aprendizaje de la programación informática en el mundo escolar, en los Estados Unidos en el 2013 se promovió una iniciativa a nivel global que actualmente registra un gran número de participantes.

La hora del código es un gran evento que involucra a millones de estudiantes en más de 180 naciones: nace como una lección que dura una hora, un evento que dura un día, que introduce a la ciencia informática, con el fin de eliminar la idea generalizada que se tiene sobre que el hecho de programar es algo bastante complicado y no está al alcance de todos.

Esta acción emprendida en USA se ha transformado en una iniciativa mundial que tiene como objetivo difundir la informática en las escuelas y en

otros sitios, a partir de una actividad de programación en el ordenador que dura una hora. Por lo tanto, cada año, durante la semana de educación a la informática, en el mes de diciembre, se lleva a cabo la hora del código, con el soporte de una asociación de socios conocidos, como Microsoft, Apple, Amazon, Boys and Girls Clubs of America y la College Board. El objetivo es que los usuarios se hagan conscientes del hecho de que la informática puede ser divertida y creativa, además de accesible a todos independientemente del contexto. La idea de base viene de la toma de conciencia por parte de los expertos de que existe una gran brecha entre el mundo del trabajo, que requiere cada vez más profesionalidad en el campo de la programación, y el actual número insuficiente de jóvenes graduados con las habilidades específicas que el mercado requiere (Wing, 2014).

Es difícil pensar en un trabajo futuro que no requiera habilidades específicas relacionadas con el pensamiento computacional y los conocimientos informáticos. Además, aprender a programar puede resultar un poderoso instrumento de expresión y de crecimiento personal en un "mundo digital" (Code.org, 2013).

Uno de los lemas utilizados para dar publicidad al evento es una de las frases atribuidas a Steve Jobs:

"Everybody in this country should learn how to program a computer because it teaches you how to think" (Todo el mundo en este país debería aprender a programar un ordenador porque te enseña a pensar)

que anticipa la perspectiva del pensamiento computacional.

A nivel institucional, en el 2014, el presidente de los Estados Unidos Barack Obama participó a la iniciativa escribiendo algunas líneas en código con Java a través de la plataforma de *code.org*. También él contribuyó a la difusión del mensaje dirigido a los estudiantes americanos, pidiéndoles no conformarse con

jugar con su propio Smartphone o con los videojuegos o de descargar las App, sino intentar construirlas y realizarlas con sus propias manos, es decir a programarlas. Así pues, Obama enfocó su discurso en la necesidad de incrementar los niveles de alfabetización digital de los estudiantes de hoy con el fin de no tener únicamente consumidores de productos digitales sino también creadores de estos.

2.9.2 Otros programas e iniciativas

El éxito de la hora del código de *code.org* ha permitido a muchos estudiantes K-12 comenzar a conocer las maravillas de la informática, haciendo aumentar el interés en ella. Se han difundido, especialmente en línea, múltiples recursos que ayudan a quien desea cimentarse en esta nueva realidad. En los últimos años no han faltado las iniciativas que apoyan la promoción del pensamiento computacional a través de campañas de alfabetización a gran escala y de la divulgación del *coding* (Coding Literacy Campaigns) y otros eventos y actividades que no están directamente asociados a instituciones educativas (Manilla et al., 2014). Estas iniciativas han obtenido el apoyo de varias haciendas tecnológicas, organizaciones gubernamentales y personajes públicos alcanzando un amplio público; también inspiraron la planificación de otros eventos. De hecho, buscan activar y extender un proceso de aprendizaje informal que pueda transformar la adquisición de las habilidades de programación en un fenómeno de desarrollo del pensamiento computacional que aproveche la experiencia masiva y su naturaleza colaborativa (Bogliolo et al., 2016).

Un ejemplo son los clubs que, a partir del 2011, son muy populares en varios países (Tabla 11), como los *Coderdojo* y los *Codeclub* (Manilla et al., 2014). Los *Coderdojo* son clubs extraescolares guiados por voluntarios (ninja), gimnasios inspirados en las artes marciales, que llevan a cabo actividades lúdicas y educativas dirigidas a niños de 7 a 17 años; son gratuitos su objetivo es enseñar a programar a los más jóvenes. Se trata de un movimiento abierto, libre

y gratuito organizado en distintos clubs independientes del planeta, el primero de los cuales nació en Irlanda. Los niños aprenden, a través de las actividades propuestas por los voluntarios, a utilizar los lenguajes de programación como Hypertext Markup Language (HTML), Javascript y Cascading Style Sheets (CSS), a crear videojuegos con Scratch y a programar hardware por ejemplo con *Arduino*.

Así pues, niños y jóvenes vienen orientados hacia el uso consiente de las tecnologías, no pasivo, que los pueda volver capaces de adaptar instrumentos y tecnologías a sus propias necesidades. Gracias a la disponibilidad de bibliotecas, coworking, haciendas locales y públicas es posible organizar actividades que, actualmente se están arraigando también en ambientes escolares.

Los *Codeclub* nacen con la idea de que todos los niños de edades entre 9 y 13 años, independientemente de su proveniencia social y de su origen, deben tener la oportunidad de aprender a programar. Proporcionan proyectos de *coding* fáciles de seguir a través de Scratch, HTML, CSS y Python, crean juegos, animaciones y sitios web. El objetivo es el de introducir gradualmente a la creación digital a través de la programación, para dar a todos los niños, la capacidad, la confianza y la oportunidad de plasmar el mundo.

CS First es una de las muchas iniciativas promovidas por Google en el 2014 para fomentar el acceso a la programación informática; accediendo a la plataforma online es posible que los jóvenes realicen actividades que los llevan a descubrir la informática, a través del lenguaje Scratch.

MADE WITH CODE es otra iniciativa promovida por Google dirigida a cerrar la brecha de género en el sector tecnológico. Google busca mejorar las habilidades de programación informática en las mujeres jóvenes de las escuelas medias y superiores para alentarlas al estudio de la Informática.

El objetivo que se proponen los *First Clubs* es el de aumentar la seguridad

en el uso de la computadora, tener el valor de enfrentar nuevos retos y perseverar cuando se enfrentan problemas difíciles. También los concursos de programación se están difundiendo como instrumentos que permiten aumentar la motivación hacia la Informática y el pensamiento computacional, como el ya mencionado concurso internacional *Bebras*, que traducido significa castor, símbolo de un animal inteligente, trabajador y determinado, está dirigido a todos los alumnos y alumnas de las escuelas primarias y secundarias de I° y II° Grado ofreciéndoles la posibilidad de adentrarse en las tecnologías de la información a través de un concurso no competitivo. El concurso toca distintos aspectos informáticos que se relacionan al pensamiento algorítmico, la informática y la sociedad, así como las estructuras, los modelos, las disposiciones y el uso del sistema informático (Manilla et al., 2014).

Otra iniciativa a nivel europeo y que se extendió posteriormente en el resto del mundo fue lanzada durante el European Code Week, a partir del 2013, por un movimiento de base dirigido por voluntarios que promueven el *coding* en sus propios países (39) como embajadores de la semana del código. La idea es transmitir el mensaje de que la programación está al alcance de todos y es importante motivar a las personas a aprenderla. El evento busca contribuir a llevar las habilidades de programación de base y de alfabetización digital a todos de manera divertida y que los involucre. Este evento ha sido sostenido por la Comisión Europea, como parte de la estrategia para un mercado digital único, con el fin de favorecer el incremento de las habilidades digitales.

Las iniciativas de base que tienen los financiamientos públicos cuentan con el apoyo de socios industriales; Google por ejemplo apoya *code.org*, *Bebras* y *Computer Science Unplugged*. Microsoft da soporte a *code.org*, *Coderdojo* y *Computing At School* (CAS) (Bocconi et al., 2016).

Tabla 11

Localización de las iniciativas que sostienen el pensamiento computacional.

Nivel	Iniciativas
Global	Coderdojo
	Code.org
	Bebras
	CS UNPLUGGED
	CODE CLUB
	MADE WITH CODE
Europeo	EU CODE WEEK
	EUROPEAN CODING INITIATIVE
	BAREFOOT COMPUTING
	CAS
	CODE IT LIKE A GIRL
	PROGRAMAMOS
Extra-europeo	Code@SG Movement
	COMPUTHINK
	Code for Chance Myanmar
	COOLTHINK@JC

Fuente: elaboración propia.

2.10 El *coding*: instrumento de promoción del pensamiento computacional

2.10.1 El *coding* de forma online

Las investigaciones científicas han mostrado, en los últimos años, que para sostener las dimensiones del pensamiento computacional es posible usar como recurso el *coding*, uno de los argumentos actualmente más debatidos en los contextos y en las políticas escolares mundiales. Se trata de un enfoque que ve en la participación activa de los estudiantes y en la construcción individual o colectiva de artefactos digitales y no digitales, algunos de los principales puntos de interés para la reflexión pedagógica.

El *coding* indica el uso didáctico de instrumentos y métodos de programación visual por bloques para favorecer el desarrollo del pensamiento

computacional entendido como la habilidad mínima de razonamiento algorítmico que cualquiera debería desarrollar para identificar un proceso de construcción, hecho de pasos simples e inequívocos, que lleven a la solución de problemas complejos (Bogliolo, 2016, p. 13 y 47). Junto con las habilidades de lectura y escritura se considera una de las habilidades necesarias actualmente para desempeñarse de manera consiente en una sociedad tecnológica como la nuestra.

En informática, el término *coding* se refiere a la escritura de un programa, una secuencia de instrucciones que generalmente puede ejecutar una máquina. Hacer *coding* lleva a saber programar, a crear mundos virtuales dictados por la imaginación, a realizar productos a través de las ideas propias y de los propios proyectos mentales. Hacer *coding* presupone que se aprenda el lenguaje de la máquina con la cual se quiere dialogar. Los seres humanos utilizan la facultad del lenguaje para comunicar al interno de la comunidad a la que pertenecen por medio del uso de un código común que permite volver explícitas las propias ideas (García-Peñalvo et al., 2018).

Los lenguajes de programación están dotados de reglas precisas y, por este motivo, hacer *coding* es como aprender un idioma extranjero. Conocer el idioma del ordenador permite mantenerse al día con el futuro que avanza. Crow (2014) afirma que no se trata de equipar a la próxima generación de ingenieros del software, sino de promover el pensamiento computacional. Aunque no todos los jóvenes llegarán a convertirse en informáticos profesionales, seguramente obtendrán ventaja al saber pensar de esta manera. De hecho, comprenderán y sabrán como dominar la tecnología y resolver los problemas en casi cualquier disciplina (Crow, 2014).

Aprender a hacer *coding* y desarrollar el pensamiento computacional son habilidades importantes para el proceso de aprendizaje de los niños, porque les ayuda a enfrentar muchas situaciones que se presentan en la vida y a mejorar la colaboración entre el hombre y la máquina (Rees et al., 2016).

Resnick (2013) sostiene que el *coding* es una extensión de la escritura, ya que si se sabe programar es posible escribir historias interactivas, juegos, animaciones y simulaciones. Y, como en la escritura tradicional, existen poderosas razones por las cuales todos pueden aprender a programar ya que mientras programan, además de aprender las matemáticas y las ideas computacionales, las personas aprenden estrategias para resolver problemas, diseñan y comunican ideas, habilidades que pueden ser útiles para todos, independientemente de la edad, del contexto, de los intereses o de la ocupación

El *coding* es por lo tanto un elemento clave que permite habilitar el pensamiento computacional (Lye & Koh, 2014; Sáez-López et al., 2016) y, en este sentido, resulta una de las prácticas didácticas más utilizadas por los profesores en el panorama K-12 (Grover & Pea, 2013; Kafai & Burke, 2015) siendo objeto de particular interés en el campo de la investigación empírica de los últimos años.

En el panorama político y científico se está consolidando la idea de que los jóvenes de todas las edades deberían poseer algunas capacidades computacionales por lo menos a nivel básico (Kalelioğlu et al., 2016); Incluso si por un lado, hay muchos partidarios de la idea de que aprender a programar no es esencial para el desarrollo del pensamiento computacional y que de hecho, enfatizar el *coding* podría generar en los estudiantes desinterés por la informática (Lu & Fletcher, 2009), en los últimos años, se ha registrado un gran incremento en el número de estudios realizados al respecto y en el interés científico sobre los beneficios que aporta la práctica del *coding* en todos los niveles escolares (K-12) (Grover & Pea, 2013; Israel et al. 2015; Kafai & Burke, 2013).

Muchos estudios afirman que la adquisición de habilidades de pensamiento computacional mejora la capacidad de resolución de problemas y de pensamiento crítico (Durak & Saritepeci, 2018) y se reducen las actitudes negativas respecto a la programación a través de instrumentos idóneos que

materializan el proceso de construcción del algoritmo gracias a la posibilidad que tiene el usuario de interactuar de manera simple (Maloney et al., 2010).

Como sostiene Seymour Papert (1980), el aprendizaje más eficaz se observa cuando los individuos pueden diseñar y crear productos (Harel & Papert, 1991; Peppler & Kafai, 2007). Posteriormente la práctica aumenta la motivación del estudiante y permite un aprendizaje sin lugar a duda mejor, favoreciendo una actitud positiva hacia la programación.

Gracias a varias investigaciones se ha encontrado que las actividades de *coding* llevadas a cabo con herramientas visuales, por ejemplo, Scratch o Alice, han aumentado la comprensión de conceptos de programación de los estudiantes, en su comportamiento y en su motivación para la programación (Bishop-Clark et al., 2007; Howland & Good, 2015; Malan & Leitner, 2007; Nikou & Economides, 2014).

Las habilidades que demuestran ser las más efectivas en la programación están relacionadas con la manera de pensar, que es útil no solo en el ámbito de las actividades cognitivas usadas en la creación de programas y sistemas informáticos. Esto significa que hay un modo específico de pensar y organizar los pensamientos que favorece el desarrollo de las habilidades de cálculo que promueve el análisis y la lógica con la cual se conectan las ideas y se organizan y representan los procedimientos. Estas habilidades se mejoran con actividades y ambientes de aprendizaje relacionados con el pensamiento computacional desde las primeras etapas (Zapata-Ros, 2015).

La programación informática es percibida como una competencia importante para el desarrollo de las habilidades de la resolución de problemas y del razonamiento lógico. Por lo tanto, su integración a todos los niveles educativos, a partir de los primeros años, se considera muy valiosa. Los estudios que se han realizado en estos años han tenido como objetivo explorar el fenómeno de forma más detallada, indagando, por ejemplo, sobre los efectos de

la programación con instrumentos idóneos para las habilidades de la resolución de problemas de los estudiantes y sobre las percepciones que los estudiantes tienen sobre el uso de la práctica del *coding*, desde la confianza en sí mismos hasta el aumento de la motivación (del Olmo-Muñoz et al., 2020) y participación (Kalelioglou et al., 2014).

Las investigaciones llevadas a cabo hasta el día de hoy han tenido el objetivo de analizar los beneficios de las actividades de *coding* a nivel cognitivo y afectivo con el objetivo de proponer una remodelación de los sistemas educativos para que promuevan el pensamiento computacional sin descuidar el aspecto emotivo y social del estudiante (Kalelioglou et al., 2014; Scaffidi & Chambers, 2012; Wilson & Moffat, 2010).

Algunos investigadores, por su parte, han indagado en particular sobre la manera en la que se ha integrado la práctica del *coding* y/o la programación informática en los planes de estudios, que actualmente, resulta insuficiente. Las investigaciones se han enfocado en la necesidad de formar a los profesores desde la escuela infantil para que adquieran las habilidades necesarias para lograr dicho objetivo.

Se ha puesto particular atención a que los individuos involucrados puedan conseguir las habilidades relacionadas al desarrollo del pensamiento lógico y creativo, a la resolución de problemas, a la motivación del aprendizaje, al trabajo en equipo, a la confianza en sí mismos y a la percepción del *coding* en su proceso de aprendizaje (Kalelioglou, 2014; Wong, 2015).

Entre los instrumentos que mejor se prestan para llevar a cabo las actividades de *coding*, tiene particular importancia el lenguaje de programación visual, más que los lenguajes de programación tradicionales como facilitadores de las tres dimensiones del pensamiento computacional (conceptos, prácticas, perspectivas) especialmente en contextos K-12 (Barr & Stephenson, 2011; Grover & Pea, 2013; Lye & Koh, 2014).

Los comandos basados en bloques permiten a los estudiantes operar más fácilmente con el drag and drop, reduciendo la carga cognitiva y permitiéndolos concentrarse en la lógica y en las estructuras involucradas en la programación en vez de que en la mecánica de los programas de escritura (Maloney et al., 2008).

Los lenguajes visuales presentan notables ventajas para introducir el pensamiento computacional a través del *coding* (Lye & Koh, 2014) ya que:

- permiten poner en acción las prácticas computacionales pues los resultados de su programación pueden verse como objetos animados;
- involucran a los estudiantes en la construcción de productos digitales, de manera que las actividades de programación sean utilizadas como un medio para permitir a los estudiantes expresar sus propias ideas;
- favorecen la alfabetización digital de los estudiantes para la creación, el intercambio y la mezcla de los recursos digitales;
- transforman a los estudiantes para que dejen de ser consumidores pasivos de la tecnología (Resnick et al., 2009).

Como ya se ha mencionado, gran parte de la comunidad educativa y científica ha mostrado interés en los beneficios que los niños pueden adquirir aprendiendo a programar independientemente del sector al que dediquen su futura actividad profesional. Bajo este escenario, el *coding* no es un instrumento en sí mismo, pero permite desarrollar otras habilidades y mejorar los resultados en cuanto al aprendizaje y la motivación de los estudiantes. En particular, los nuevos lenguajes de programación, como los visuales, en los que no es necesario aprender la sintaxis como Alice, Kodu y sobre todo Scratch, han

despertado el interés de la comunidad educativa para el *coding* (Moreno-León & Robles, 2016) tanto que la investigación busca comprender, a través de pruebas empíricas, la participación de los estudiantes en las actividades relacionadas (Lye & Koh, 2014).

2.10.1.1 Scratch

Scratch (Resnick et al., 2009) es un lenguaje de programación muy popular que tiene como objetivo el de “aprender a programar”, mientras se aprenden los idiomas (Burke & Kafai, 2012; Lee, 2011), las matemáticas y otras disciplinas escolares. Es un ambiente de programación visual gratuito y accesible desde cualquier navegador, con una interfaz gráfica, que permite programar de forma divertida y creativa, animaciones, videojuegos, música, arte interactiva a través del uso del lenguaje de la computadora.

Se inspira en los lenguajes de programación para jóvenes como LOGO (Resnick, 2007) y se ha diseñado para ser fácil de utilizar e intuitivo: de hecho, las instrucciones se presentan como elementos gráficos en forma de bloques de colores que, encajando entre sí, dan vida al código activando los distintos comandos y acciones (Monroy-Hernández & Resnick, 2008) (Figura 1).

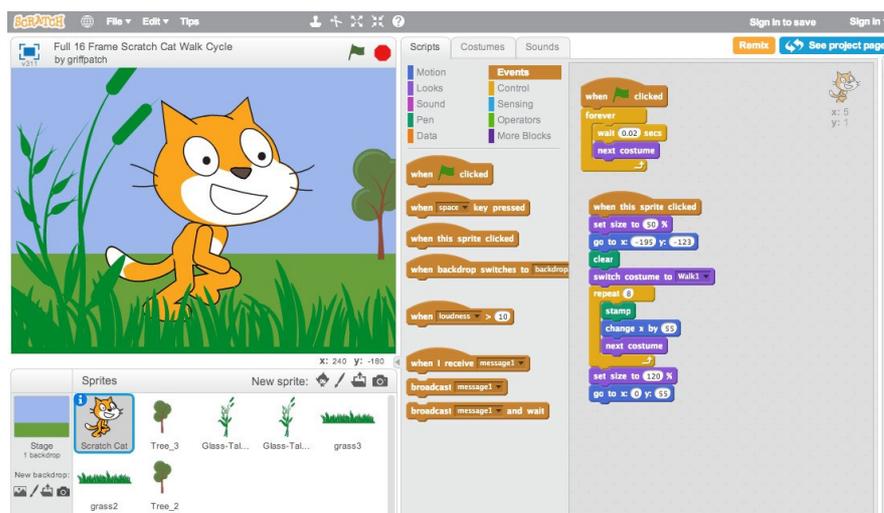


Figura 1. Interfaz de Scratch.

Scratch es capaz de funcionar como una máquina virtual en Linux, Mac OS, UNIX y Windows. Uno de los objetivos clave que se plantearon los diseñadores fue el dar la posibilidad de introducir a la programación a quien no ha tenido previamente ningún tipo de experiencia con ella. De hecho, este objetivo ha guiado muchas decisiones en el diseño del *layout* de la interfaz de usuario, en la constitución de una parte fija, el fondo (escenario), y de un cierto número de *sprite* móviles (Maloney et al., 2008), elementos al alcance de los más jóvenes. En particular, el *sprite* es un elemento fundamental de los programas realizados con Scratch y puede consistir en una o más imágenes (disfraces), con archivos de audio asociados y con todas las instrucciones que lo rigen. En Scratch no hay una sintaxis oscura o la puntuación de los lenguajes de programación tradicionales: sus bloques están moldeados para adaptarse solo cuando tienen sentido sintáctico: las instrucciones (o piezas de código) se pueden representarse como piezas de un rompecabezas que encajan perfectamente si son sintácticamente correctas (Malan & Leitner, 2007).

Las estructuras de control (como *por siempre* y *repite*) hacen la forma de C para sugerir que los bloques deben colocarse dentro de ellas, mientras que los bloques con valores de output se modelan en base a los tipos de valores restituidos: ovalados para los números y hexagonales para los booleanos.

A través de Scratch se pretende ayudar a los niños (de 8 años en adelante) a desarrollar las habilidades de aprendizaje esenciales del siglo XXI y dar soporte al auto aprendizaje a través del *tinkering* y el trabajo en equipo (Resnick et al., 2006). El mismo nombre "Scratch" deriva del término *scratching* que es una técnica utilizada por los disc-jockey hip-hop, que juegan con la música manipulando los discos de vinilo con sus propias manos, mezclando clips musicales y produciendo diversos efectos sonoros de manera creativa (Resnick et al., 2009).

En los últimos años se ha querido incluir a los niños de menor edad en el proyecto inspirado por la teoría constructorista de Papert a través de Scratch

Junior (ScratchJr), que toma como punto de partida el popular lenguaje de Scratch utilizado en el mundo por millones de jóvenes a través de un cambio hecho a la interfaz que ha sido rediseñada con cuidado en todas sus características para adaptarse al nivel de desarrollo cognitivo, personal, social y emocional de los más pequeños (de los 5 a los 7 años). Con ScratchJr es posible crear historias interactivas y juegos uniendo los bloques programables no a través de palabras escritas sino con símbolos (flechas, figuras, etc.) también gracias a la gran cantidad de material pedagógico gratuito disponible en línea (Malan & Leitner, 2007).

Scratch, que puede utilizarse en línea, desde el sitio www.scratch.mit.edu, o de forma offline, después de haber descargado el software, tiene también una retroalimentación social ya que los proyectos se pueden compartir en una comunidad para el aprendizaje creativo en la cual los inscritos pueden publicar, tomar inspiración y compararse con otros usuarios, recibir retroalimentación y alentarse (Resnick et al., 2009). El icono *comparte* permite mostrar aquello que se ha creado, *remix* se puede copiar pero de forma creativa, así no se roba la idea a nadie sino más bien se reconoce el valor de la actividad desarrollada por otro y es posible enriquecerla.

En cambio, ScratchJr se ha desarrollado como aplicación gratuita que puede utilizarse con un dispositivo electrónico (Smartphone, Tablet, etc.).

Scratch y ScratchJr se han diseñado para dar soporte a la llamada *Espiral del aprendizaje creativo* (Figura 2) que sintetiza como se lleva a cabo el proceso de aprendizaje creativo en los niños. En este proceso, ellos imaginan (*imagine*) que cosa desean hacer, crean (*create*) un proyecto basado en sus ideas, juegan (*play*) con sus creaciones, comparten (*share*) sus ideas y creaciones con los demás y reflexionan (*reflect*) sobre sus experiencias: todo esto los lleva a imaginar nuevas ideas y nuevos proyectos, reiniciando nuevamente el proceso. Mientras los estudiantes enfrentan este proceso, de forma repetitiva, aprenden a desarrollar sus propias ideas, probarlas, conocer los límites, experimentar

posibles alternativas, obtener aportes de otros y generar nuevas ideas basadas en sus experiencias. (Resnick, 2013).



Figura 2. Espiral del aprendizaje creativo (Resnick, 2007).

El proyecto Scratch comenzó en el 2003 mientras que el software y el sitio Web se lanzaron públicamente en el 2007. Después de su lanzamiento se propusieron nuevas iniciativas en todo el mundo para enseñar la programación a los niños y jóvenes. Además de las actividades extraescolares, en los campamentos de verano, muchos docentes, desde Educación Primaria hasta la Universidad, iniciaron a introducir el *coding* en sus clases a través de actividades que fomentaran el aprendizaje de diversas disciplinas, como matemáticas, ciencias, artes, música e idiomas; una prueba de esto son los recursos generados por los mismos educadores y compartidos en el sitio web de ScratchEd (Moreno-León & Robles, 2016).

El ambiente de Scratch gusta a los jóvenes por diversos motivos. Por ejemplo, con Scratch no es necesario luchar con los errores de sintaxis porque los estudiantes reciben una retroalimentación inmediata ya que los resultados del código escrito se pueden ver inmediatamente cuando el programa viene ejecutado (Manilla et al., 2014).

En un discurso de la conferencia TED (2012) Mitchel Resnick, director del grupo Lifelong Kinderkarten del laboratorio Medialab del MIT, resume de manera eficaz las razones por las cuales los niños deben aprender a programar (*learn to code*) resaltando la importancia del pensamiento computacional como objetivo educativo innovador (*code to learn*):

Cuando los niños crean un proyecto con Scratch aprenden a programar, pero, algo aún más importante, programan para aprender [...] porque aprendiendo a programar aprenden muchas otras cosas, abriéndose a nuevas oportunidades de aprendizaje [...] Si se aprende a programar entonces se podrá también programar para aprender (Resnick, 2012).

Resnick, en definitiva, sostiene que cuando se aprenden ideas y conceptos a través de Scratch se hace de una forma bastante significativa y estimulante, de manera que se aprende más y se aprende mejor. Las habilidades que se adquieren no están relacionadas únicamente con la programación de códigos sino, como ya se ha dicho, la posibilidad de experimentar nuevas ideas, incluso complejas, dividiéndolas en partes más simples, y utilizando la colaboración con otras personas como una fortaleza para los propios proyectos, junto con la tenacidad y la perseverancia (Brennan & Resnick, 2012).

Son varios los estudios que se han desarrollado en los últimos años en cuanto al uso de Scratch en diversos contextos escolares y con referencia a distintos niveles de educación, tanto en actividades curriculares como extracurriculares, que han puesto en evidencia aspectos relacionados con el aumento en la motivación de los estudiantes para las actividades de programación, trayendo como consecuencia un aumento de su éxito en ellas (Malan & Leitner, 2007, Sáez-López et al., 2017).

Hsu y Hu (2017) han demostrado que el uso de Scratch, en un curso de matemáticas para niños de Educación Primaria, puede aumentar el nivel cognitivo y la autoeficacia de los estudiantes, ya que no conlleva un nivel alto de

ansiedad en el aprendizaje y al mismo tiempo mantiene la motivación y el interés por el aprendizaje de los estudiantes.

Los niños que utilizan Scratch durante las actividades escolares parecen apreciarlo y son más entusiastas con respecto a otras clases (Wilson Moffat, 2010) además del hecho de que los proyectos realizados dan soporte al aprendizaje a largo plazo llevando el aprendizaje significativo a un contexto particular (Brennan & Resnick, 2012).

Moreno-León y Robles (2016) llevaron a cabo una revisión sistemática de la literatura respecto al uso de la programación con Scratch con temas relacionados propiamente con la informática, además de haber analizado los distintos conjuntos de habilidades desarrolladas por los estudiantes mientras aprenden a programar en el ambiente de Scratch.

Si bien los documentos analizados proporcionan resultados prometedores en cuanto al uso de la programación como recurso educativo, esta revisión pone en evidencia la necesidad de llevar a cabo más investigaciones empíricas en las aulas, utilizando muestras más amplias de estudiantes que permitan obtener conclusiones claras sobre los tipos de aprendizaje que pudiesen mejorarse a través de la programación (Moreno-León y Robles, 2016).

En general, la mayor parte de los estudiantes que utilizan Scratch, afirman tener una influencia positiva en sus experiencias posteriores con lenguajes especializados como Java (Resnick et al., 2009).

En los primeros días de enero 2019, ante el evidente aumento de interés en el mundo educativo, se hizo pública la nueva versión de Scratch 3.0, proyectada con nuevas extensiones para funcionar en cualquier navegador y en varios tipos de dispositivos, incluidas las *Tablet* (Scratch, 2019).

2.10.1.2 El sitio code.org

“Cada estudiante de cada escuela debería tener la oportunidad de aprender informática”.

Esta frase aparece en la página de inicio de la versión italiana del sitio web de *code.org* y sintetiza de manera muy clara la filosofía de sus creadores, convencidos de que cada estudiante de cada escuela debería tener la oportunidad de aprender a programar la computadora.

Su objetivo es extender la educación informática a todas las escuelas, aumentando la participación de mujeres y estudiantes de color. En su visión, la informática debería ser parte del plan de estudios de base de la educación, junto con otros cursos STEM (Ciencias, Tecnología, Ingeniería y Matemáticas) como biología, física, química y álgebra.

Code.org es una organización sin fines de lucro que se ha dedicado a sensibilizar a la población sobre este tema y lo sigue haciendo (Du et al., 2018) empezando en los Estados Unidos hasta llegar a diversas partes del mundo. De hecho, su plataforma, se presenta en 54 idiomas distintos dando así testimonio de su gran difusión. Está estructurada como un conjunto de juegos que frecuentemente se describen y se explican a través de carteles y videos de personajes famosos como Bill Gates y Mark Zuckerberg con el fin de proporcionar un motivo más a los estudiantes para dedicarse a la informática (Kalelioglu, 2015) y en particular al *coding*.

Las actividades de juego no requieren prerrequisitos específicos en las materias escolares del plan de estudios o habilidades de programación precedentes. Como en Scratch, los estudiantes utilizan los bloques visuales, a través del drag & drop, para escribir los programas, con la consiguiente facilidad de uso y de aprendizaje respecto a un lenguaje basado en el texto, ya que no se debe pensar a la sintaxis de los programas de escritura, sino concentrarse solo en la lógica de cómo escribir dichos programas (Corradini et al., 2017;

Theodoropoulos et al., 2017). Con mucha frecuencia los protagonistas de los juegos son personajes de dibujos animados de las generaciones más jóvenes a los cuales el usuario debe hacerles realizar tareas específicas, a diferencia de Scratch en donde se puede proporcionar libremente el conjunto de instrucciones (Figura 3).



Figura 3. Angry bird en code.org

Code.org busca proporcionar los primeros elementos de la educación de base para la informática a todos los estudiantes de manera gradual; de hecho, los ejercicios iniciales son muy triviales y el conjunto de instrucciones disponibles es reducido (por ejemplo "adelante", "gira a la derecha/izquierda). El nivel de dificultad aumenta de manera gradual de un ejercicio a otro, las instrucciones y las estructuras de programación se agregan gradualmente al todo. Si el estudiante no es capaz de completar un ejercicio con éxito, el sistema proporciona una retroalimentación, útil para la autocorrección (Corradini et al., 2017). Los estudiantes pueden observar su propio proceso de aprendizaje, monitorear los niveles que han completado, aquellos que un no han completado y los trofeos que han ganado y en qué tema (Kalelioglu, 2015).

Entre una actividad y otra los tutoriales web enseñan y refuerzan importantes conceptos informáticos útiles para el desarrollo de la tarea/juego sucesivo. Es posible descargar como adjunto material didáctico que se puede imprimir en el que se describen de manera detallada las lecciones que se pueden proponer a los alumnos tanto de manera online como *unplugged* (del Olmo-Muñoz et al., 2020). Con la ayuda de este material didáctico, los estudiantes pueden aprender la lógica del algoritmo, de las condiciones, de las variables, de los bucles y de las funciones (Kim & Kim, 2017). El material y las actividades fueron pensados haciendo referencia al “*K-12 Computer Science Framework*” desarrollado por CSTA (Computer Science Teacher Association), en colaboración con otras organizaciones como ACM (Association for Computer Machinery), CIC (Cyber Innovation Center), NMSI (National Math + Science Initiative) y el mismo *code.org* (K–12 Computer Science Framework, 2016).

Desde el sitio es posible acceder a la sección *cursos* (con duración de 20 horas cada uno) que, divididos por edades, permiten aprender las bases de la informática tanto a través del uso directo del ordenador que sin el uso de esta. Dichos cursos siguen un modelo de enseñanza/aprendizaje informático basados en el *blended-learning*, que significa que los estudiantes aprenden de una combinación de actividades de *coding on-line* auto guiadas y de actividades que no requieren el uso del ordenador (*unplugged*), que son dirigidas de manera más tradicional por el docente (Kalelioglu, 2015).

También los cursos gratuitos se han desarrollado de acuerdo con los estándares de la filosofía educativa de *code.org* y CSTA (Theodoropoulos et al., 2017) y buscan mostrar a los estudiantes de todas las edades que su aprendizaje puede ser divertido, colaborativo y creativo; están diseñados para motivar a los estudiantes y a los educadores a continuar en futuro respectivamente con el aprendizaje y la enseñanza de la informática, mejorando las relaciones entre la escuela y el mundo real, cada vez más tecnológico y digital. A través de la perseverancia los estudiantes aprenden a resolver los problemas, incluso aquellos reales y al mismo tiempo aprenden la informática

desarrollando el pensamiento computacional además de las habilidades de programación de base.

Además de los diversos cursos de la hora del código (*hour of code*) el sitio proporciona recursos para escuelas medias y superiores que permiten crear aplicaciones, animaciones y juegos en Javascript. Con sus materiales didácticos útiles tiene características motivadoras ya que pueden ser un buen recurso para docentes y estudiantes (Kim & Kim, 2017), independientemente de su habilidad, medios familiares o estado cognitivo (Corradini et al., 2017).

Los cursos propuestos en el sitio de *code.org* y las iniciativas relacionadas han servido como punto de partida para muchos investigadores del área educativa que, en los últimos años, de manera particular, han analizado los múltiples aspectos relacionados con el uso de la plataforma en sí, y por consecuencia al *coding*, al desarrollo del pensamiento computacional y a la alfabetización digital, tanto que se ha convertido en una de las plataformas más citadas, junto con *Bebras*, en artículos y trabajos científicos (Roig-Vila & Moreno-Isac, 2020). Por ejemplo, los juegos de programación que requieren una forma de pensar algorítmica se están volviendo la principal fuente de educación a la programación a partir de Educación Primaria (Popovic, 2017).

Por su parte, Kalelioglu (2015) ha explorado los efectos que tiene la programación de *code.org* en los estudiantes de una clase de cuarto curso de una escuela de Educación Primaria privada en Turquía en cuanto a las habilidades de pensamiento reflexivo y a la resolución de problemas. Este estudio cualitativo y cuantitativo se llevó a cabo con una muestra de 32 estudiantes de 10 años, 17 de sexo femenino y 15 de sexo masculino, se trató de actividades realizadas una hora por semana durante la hora de informática en un total de 5 semanas. De los resultados de este estudio Kalelioglu sostiene que el sitio *code.org* posee características motivadoras, material didáctico útil y programas de lecciones que pueden ser un buen recurso tanto para los profesores de informática como para los estudiantes que la aprenden (Kim & Kim, 2017).

Entre los entrevistados hubo estudiantes que declararon haber aprendido la programación, haber mejorado, a través del sitio, sus conocimientos de matemáticas y geometría, haber aprendido a encontrar soluciones a través de un menor número de pasos, y haber mejorado su alfabetización informática.

Los beneficios registrados no solo son cuantitativos sino también cualitativos, ya que, además de apreciar las actividades de programación, los usuarios tienen la oportunidad de desarrollar conceptos matemáticos, pensamiento lógico y creativo, habilidades para la resolución de problemas, habilidades sociales. El uso de la plataforma de *code.org*, mejora la motivación para el aprendizaje, la participación, la reflexión y el interés de los estudiantes y profesores, y la cooperación. Los aspectos metodológicos también surgen con éxito gracias a la idea de aprender jugando (Corradini et al, 2017; Kalelioglu, 2015; Popovic, 2017; Theodoropoulos et al., 2017).

2.10.1.3 **Robótica educativa**

La robótica educativa permite el diseño, la construcción y el desarrollo de ambientes de aprendizaje activos, pasando efectivamente de lo abstracto a lo tangible de manera que quien participa obtiene un rol principal en la generación y en la construcción del conocimiento del objeto en estudio (Caballero-González, 2019). Comprende una amplia gama de tecnologías robotizadas utilizadas en el ámbito de la enseñanza y el aprendizaje en el contexto escolar. En los últimos años se ha presenciado la difusión de la robótica como nueva tecnología que puede utilizarse en los distintos niveles escolares como ambiente de aprendizaje multidisciplinario motivante que promueve el desarrollo de las habilidades de pensamiento computacional y algorítmico y de las habilidades de programación.

El hecho de que la robótica educativa sea considerada por muchos como una herramienta valiosa para el desarrollo de las habilidades cognitivas y sociales de los estudiantes ha atraído fuertemente el interés de los profesores, desde la escuela de la infancia hasta la universidad, y de los investigadores

(Kucuk & Sisman, 2017). El enfoque a la robótica educativa parece ser particularmente adecuado para favorecer el proceso de aprendizaje de todos los alumnos, incluso aquellos con necesidades educativas especiales (NEE), ya que el uso de un robot favorece experiencias de autoeficacia y de autocontrol (*coping*) y un acercamiento hacia la tarea menos estresante, en un ambiente colaborativo y de juego (Damiani et al., 2013).

Por lo tanto, la robótica educativa se convierte en una herramienta versátil, muy interesante en las áreas de la educación, formal e informal, pues permite contextualizar la informática como materia, aumentando y mejorando las habilidades de programación, favoreciendo el crecimiento del pensamiento computacional y, al mismo tiempo, mejorando los recursos emotivos, cognitivos y motivacionales de cada alumno y de todo el grupo (García-Peñalvo, 2016; Jaipal-Jamani & Angeli, 2017).

Seymour Papert ya había expresado su idea respecto al valor de la robótica en los ambientes educativos, siguiendo la teoría constructivista según la cual los estudiantes deben asumir un rol más activo durante el proceso de aprendizaje convirtiéndose en diseñadores y creadores de sus artefactos (Papert, 1999).

Algunos estudios recientes muestran que el uso de la robótica educativa como interfaz tangible facilita el aprendizaje de los estudiantes por ejemplo en el desarrollo de los conocimientos en las áreas STEM determinando también el fortalecimiento de las habilidades sociales como la creatividad (López & Gutiérrez, 2017), la comunicación y la colaboración (García-Peñalvo & Mendes, 2018).

Si bien el uso educativo de la robótica es reconocido desde hace muchos años, es solo recientemente que comienza a ser ampliamente utilizada en la educación. En este contexto han surgido diversas tecnologías acompañadas por material y recursos de aprendizaje relevantes (Yiannoutsou et al. , 2017).

Los primeros enfoques de la robótica educativa tenían como objetivo desarrollar actividades interdisciplinarias relacionadas principalmente con las materias STEM pero, con el paso del tiempo, se han extendido a todos los niveles ofreciendo beneficios dentro de todo el mundo de la educación (Atmatzidou & Demetriadis, 2016; Kim et al., 2013;). La robótica está estrechamente relacionada a la programación (Shute et al., 2017): los estudiantes, con mucha facilidad, logran construir y controlar robots utilizando lenguajes específicos de programación, generalmente de tipo visual a bloques, mucho más fáciles de entender.

Las experiencias de robótica educativa, estrechamente relacionadas con la filosofía constructorista de Papert, estimulan a los alumnos a construir y descubrir las cosas por sí mismos, convirtiéndose ellos mismos en constructores de nuevos conocimientos (Papert, 1980). La literatura examinada muestra como las actividades de robótica educativa tienen un impacto positivo sobre el desarrollo del pensamiento crítico, de las habilidades para la resolución de problemas y las metacognitivas (Atmatzidou & Demetriadis, 2016; Bers et al., 2014; Shute et al., 2017) apoyan al pensamiento creativo y a la motivación en los estudiantes que crean sus propios productos.

Algunos estudios han demostrado que este tipo de experiencia mejora también la comprensión, por parte de los estudiantes, de los conceptos de Ciencia, Tecnología, Ingeniería, Matemáticas (materias del área STEM) (Barker et al., 2008; Kim et al., 2013) y enriquece la capacidad de pensamiento computacional relacionada a la descomposición de los problemas, la abstracción de informaciones esenciales, la generalización de la solución de varios problemas, la creación de algoritmos y procedimientos de automatización (Shute et al., 2017). Aún proporcionando un ambiente de aprendizaje rico y atractivo para la educación STEM, el logro de resultados depende en gran parte de una planificación cuidadosa de la metodología con la cual se aplica en clase (Barak & Assal, 2018).

La robótica, a través de la programación, proporciona experiencias táctiles para resolver problemas a través de las habilidades de pensamiento computacional. Los estudiantes, después de haber identificado el problema general, deben descomponerlo determinando el número de pasos necesarios para alcanzar el objetivo y de esta manera desarrollan algoritmos para el robot de manera que pueda seguir las instrucciones y actuar en consecuencia. Y si el robot no funciona como se esperaba, entra en juego la depuración (*debugging*), una forma de resolución de problemas propia de la ingeniería informática que incluso los niños pueden poner en práctica (Bers et al., 2014). Hacer depuración (*debug*) lleva a detectar y corregir los eventuales errores de programación y requiere procesos interactivos sistemáticos a través de pruebas y cambios. (Shute et al., 2017) así como la adopción de estrategias utilizadas anteriormente, para resolver problemas nuevos. De hecho, la descomposición de los problemas, las pruebas sistemáticas y la depuración, la generalización y la iteración son habilidades que se requieren en los juegos y forman parte importante del pensamiento computacional.

Cuando se habla de robótica educativa se hace referencia a las actividades con distintos tipos de robot que los estudiantes instruyen y dirigen mediante programación que pueden ser tanto *unplugged* cuando los estudiantes deben ayudar a un robot a ejecutar determinadas tareas, dándoles instrucciones claras y precisas en un orden determinado (Manilla et al., 2014) como tecnológicas.

Los robots "físicos" son aquellos que deben programarse a través de un lenguaje informático, actualmente, existe una amplia gama de posibles alternativas entre las cuales escoger a partir de los robots dirigidos a la primera infancia hasta llegar a aquellos para edades más avanzadas (Manilla et al., 2014). Por ejemplo *Bee-Bot* y *Blue-Bot* son robots de piso en forma de abeja y son usados como herramientas didácticas para alumnos desde la escuela materna hasta la primaria, perfectos para enseñar a contar, secuenciar, estimar, resolver problemas y simplemente divertirse. Desde los controles presentes en la parte posterior es posible programarlos para hacerlos moverse, incluso a través de

tecnología bluetooth, según un recorrido establecido después de haber impostado el algoritmo con las instrucciones. Como soporte para estos robots, se creó un software que, mediante la simulación en 3D, ofrece la posibilidad de moverlos de manera virtual, esto resulta un óptimo punto de partida para enseñar a los niños el lenguaje y la programación de las direcciones. Los desafíos que presenta el programa se resuelven mediante la programación de secuencias de movimiento y al final del recorrido se proporciona una retroalimentación.

Estos robots, así como muchos otros en comercio, son herramientas potentes para motivar a los niños a sintetizar el conocimiento, volviéndolo tangible y significativo. Porejemplo, cuando los niños codifican un robot, hay un resultado inmediato físico de la tarea abstracta: el robot se exhibe delante de los niños gracias a sus instrucciones (Lammer et al., 2017). Los Bee-Bot por ejemplo utilizan material didáctico que tiene el fin de motivar a los niños a la resolución de los problemas gracias también a una variedad de tapetes que sirven como superficie de trabajo sobre la cual funcionan los Bee-Bot, que estimulan las habilidades de cálculo, secuencia y orientación en el espacio.

Así pues, es importante que los profesores aprendan a utilizar dichas herramientas pedagógicas de modo que puedan maximizar sus efectos en el desarrollo de las habilidades de pensamiento computacional de los niños (Burke & Kafai, 2012; Fessakis et al., 2013; Angeli & Valanides, 2019)

Para enseñar la robótica e integrarla en otras disciplinas han sido bastante útiles los kits de robótica educativa. Generalmente, un kit educativo de robótica contiene un microcontrolador, varios sensores, pequeños ladrillos de plástico, baterías, cables, ruedas y engranes gracias a los cuales es posible construir un robot capaz de detectar objetos y moverse sin superar obstáculos, transportar objetos a través de los brazos del robot, detectar sonidos y alcanzar la fuente del sonido (Kucuk & Sisman, 2017).

Un ejemplo de kit de robótica educativa utilizado frecuentemente en Educación Primaria y objeto de educación empírica en el campo de la educación es *LEGO® Education WeDo 2.0*. Se trata de una solución práctica que desarrolla las actividades científicas en clase a través de un sistema de aprendizaje basado en robots programables incluso con bluetooth. Consiste en ladrillos, un motor y sensores de movimiento de inclinación: al ensamblar las piezas se pueden obtener robots de distintas formas y tamaños o realizar proyectos de robótica gracias a un software, fácil de utilizar que es un componente esencial de todos los proyectos. El software está dotado por una interfaz *drag and drop* de tipo gráfico compatible con Windows, IOS, Mac y Android. La participación en las actividades de construcción y programación con los kits de robótica LEGO *WeDo* ha traído resultados positivos, que hay que tener en cuenta con mucho cuidado, respecto a un aumento en el interés mostrado en el aprendizaje, en la comprensión de los conceptos de base de la informática y en la adquisición de las habilidades de pensamiento computacional (Jaipal-Jamani & Angeli, 2017; Mayerová & Veselovská, 2017).

Los conceptos abstractos se concretizan usando elementos tangibles y los materiales prácticos alientan a los estudiantes a volverse parte activa del proceso de aprendizaje siempre con referencia a las áreas clave de los estándares curriculares.

La literatura muestra que los trabajos con los kits de robótica y los enfoques de *black box* generalmente son utilizados con los niños más pequeños, mientras que para edades más avanzadas (escuelas secundarias de primero y segundo curso) se cuenta con el uso de plataformas de robótica (*white-box*) como Arduino, una plataforma hardware de bajo costo programable, con la cual es posible crear circuitos para muchas aplicaciones, sobre todo en ámbito de robótica y automatización.

Por ejemplo, los LEGO© Mindstorms (Manilla et al., 2014; Buitrago et al., 2017) son robots que se utilizan junto con un lenguaje de programación visual

que controla el comportamiento del robot en base a una serie de instrucciones o eventos basados en los inputs recibidos por los sensores (Rinderknecht, 2013). Una vez que un robot ha sido construido y un programa escrito se puede descargar del ladrillo (Samuels & Poppa, 2017). El set LEGO © Mindstorms permite a los estudiantes crear de manera fácil sistemas robóticos que pueden actuar y percibirse como pequeñas entidades sociales (Kipp & Schneider, 2017): usando estos robots los estudiantes aprenden jugando, creando nuevos conocimientos (Buitrago et al., 2017).

Makey es un producto hardware de código abierto que consiste en una placa electrónica propuesta por algunos investigadores del grupo *Lifelong Kindergarten* del MIT de Boston. En el contexto didáctico la placa *Makey* se presta para ser utilizada, por su simplicidad, al conectarla al entorno de Scratch; existen experiencias en el ámbito de la enseñanza de diversas disciplinas a partir de los cursos escolares inferiores que permiten producir diversos proyectos; en la enseñanza de la música y de manera transversal también de la tecnología el uso de *Makey* muestra su potencial en la reproducción de sonidos existentes o registrados por los estudiantes que manifiestan una mayor motivación gracias a la facilidad de uso de este hardware (Calvillo A., 2014).

Como ya se ha dicho, existen diversas formas con las que los profesores pueden aplicar los principios de la robótica educativa, sin embargo, no es la tecnología la que puede realmente hacer la diferencia sino más bien el modelo pedagógico y didáctico con el que viene concretamente utilizada. No todos los productos son igual de accesibles y simples para ser llevados a cabo en clase o pedagógicamente adecuados para los diferentes grupos de edad, por lo que es necesario recurrir a criterios para la selección que busquen la simplicidad y el uso de un idioma y la visualización gráfica eficaz y simplificada (Damiani et al, 2013).

En la tabla que se muestra a continuación se muestra una síntesis de algunos de los tipos de herramientas de robótica educativa más utilizados en el

ámbito escolar y que son objeto de diversos estudios, subdivididos por las habilidades proporcionadas y por curso escolar, a partir de la escuela de la infancia hasta llegar a la escuela secundaria de segundo curso (García-Peñalvo et al, 2016; Kanbul & Uzunboylu, 2017; Pugnali et al, 2017).

Tabla 12

Herramientas de robótica educativa.

Escuela	Herramientas de robótica	Habilidades adquiridas
Infancia	Bee-Bots, Blue-Bots, KIBO	Lograr poner en orden, subdividir un problema en pequeñas partes.
Primaria	Sphero, Dash &Dot, LEGO WeDo Makey	Saber usar herramientas de programación visual. Saber programar usando distintas instrucciones.
Secundaria de 1º grado	Lego Mindstorms Makey	Saber usar herramientas de programación visual. Saber programar usando distintas instrucciones de nivel avanzado.
Secundaria de 1º grado	Arduino, Raspberry Pi Makey	Saber usar herramientas de programación textuales. Saber resolver con la programación problemas de la vida real.

Fuente: elaboración propia.

En los últimos años el número de los estudios científicos sobre robótica educativa ha aumentado de manera considerable confirmando así el creciente interés en el ámbito educativo hacia herramientas que, como estas, permitan el desarrollo de una verdadera cultura digital en la sociedad (Zapata-Ros, 2015).

Angeli & Valanides (2019) han examinado los efectos del aprendizaje con el Bee-Bot sobre el pensamiento computacional de los niños (n=24) y de las niñas (n= 26). El estudio reporta ganancias de aprendizaje estadísticamente significativas entre la evaluación inicial y final respecto a sus habilidades de pensamiento computacional. Los resultados muestran una ganancia en términos de aprendizaje entre la evaluación inicial y la final y no reportan diferencias estadísticamente significativas por género o por grupo. Los autores invitan a integrar los dispositivos de robótica educativa en los planes de estudios desde la

infancia. Además, estos resultados demuestran que la robótica puede utilizarse para enseñar a los jóvenes estudiantes los conceptos propios de las relaciones espaciales, que constituyen un componente importante del programa de matemáticas (geometría) en la educación preescolar.

Caballero-Gonzales et al. (2019) llevaron a cabo un estudio *cuasi experimental* en una escuela de Educación Primaria con niños de edades entre 6 y 7 años, con el fin de evaluar el desempeño de los estudiantes que participaron en una experiencia formativa para reforzar las habilidades relacionadas con el pensamiento computacional y con la interacción social. Encontraron diferencias significativas entre los miembros de los grupos de experimentación y de control en cuanto al desarrollo del pensamiento computacional y se registraron valores de frecuencia positiva a favor de los miembros del grupo experimental con respecto a los comportamientos.

En un estudio *cuasi experimental* llevado a cabo en una escuela en China con ocasión del Robot Summer Camp en la que participaron 38 estudiantes de edades entre 15 y 16 años en actividades de robótica educativa con el fin de verificar el aspecto colaborativo, de crecimiento personal y de mejora en el aprendizaje en particular en cuanto a la resolución de problemas (Zhong & Li, 2020); comparando los dos grupos, experimental y de control, se encontró que no siempre el trabajo junto a otra persona lleva a resultados mejores respecto al trabajo individual, este aspecto debe ser profundizado en estudios futuros.

2.10.1.4 ***Game design***

Los estudiantes de estos tiempos se muestran bastante disponibles cuando, en la práctica didáctica, se utiliza el juego especialmente de forma digital; de hecho, la adquisición de conocimiento a través de prácticas lúdicas no es una práctica nueva.

El *game design* se propone con el fin de aumentar el interés y la motivación de los estudiantes, a partir de la escuela de la infancia, hacia la

Informática (Webb et al., 2012). Representa una herramienta válida para que los estudiantes se familiaricen y comiencen a conocer los conceptos relacionados con el pensamiento computacional (Jenson & Droumeva, 2015). El uso de plataformas como Scratch, *AgentSheets* y *AgentCubes* permite a los estudiantes construir juegos diseñándolos personalmente (Israel et al., 2015; Repenning et al., 2010). Son varios los estudios realizados con el fin de analizar los efectos de la introducción de las prácticas de *game design* en un contexto escolar.

El grupo de investigación *Scalable game Design* (SGD) del Departamento de la Universidad de Colorado Boulder en USA ha publicado varios trabajos en los que se centra la atención en el uso de juegos en el contexto escolar ya que esto lleva a los estudiantes a aprender los conceptos de Informática, de lógica y de pensamiento computacional

Repenning et al. (2010) proporcionan una lista de control que puede utilizarse en el campo K-12 para verificar las características de una herramienta de promoción del pensamiento computacional. Ellos opinan que, por ejemplo, *AgentSheets* y *AgentCubes* responden bien a los requisitos indicados ya que permiten a quien los usa crear juegos y simulaciones y publicarlos en la web a través de una simple interfaz.

Un estudio llevado a cabo en una escuela del Sur de Italia (Chiazzese et al., 2018) precisamente en Sicilia, proporciona indicaciones preliminares sobre como el *coding* para los estudiantes (N=51) es una herramienta importante pues mejora las habilidades de pensamiento computacional y en particular el uso de un ambiente de programación visual como Kodu Game Lab estimula una percepción positiva de la programación informática que viene considerada, por los resultados antes y después de la prueba, divertida, estimulante, creativa y altamente accesible. Por lo tanto, a través de experiencias de *game design* es posible crear oportunidades interesantes de diseño de actividades didácticas que pueden garantizar la promoción del pensamiento computacional para los estudiantes de ambos géneros (Touretzky, 2014).

Así pues, las experiencias de *game design* llevadas a cabo a partir de los primeros cursos escolares ponen en evidencia cambios en el comportamiento de las jóvenes en cuanto a su percepción sobre la informática, además del aumento de sus habilidades, su confianza en sí mismas y en el uso que hacen de las computadoras (Carbonaro et al., 2010).

Por lo tanto proponer experiencias de juego a través del *game design* a los estudiantes puede ser una estrategia válida para estimular el interés de las jóvenes hacia la informática y aumentar en todos los estudiantes las percepciones positivas sobre sus propias capacidades de aprendizaje, sobre la creatividad y los conocimientos sobre el uso del *coding* (Çakir et al., 2017).

El diseño y la realización de juegos electrónicos lleva a crear artefactos digitales y a activar los procesos propios del pensamiento procesal, del cálculo que pone en acción al pensamiento computacional como elemento a introducir en los planes de estudio principalmente relacionados con las disciplinas STEM (Jenson & Droumeva, 2015).

En los últimos años el *game design* ha resultado ser un enfoque interesante en el campo educativo K-12 justo porque, algunos tipos de juegos están relacionados con los modelos del pensamiento computacional (Repenning et al., 2010). De hecho, el diseño de los juegos puede focalizarse en el desarrollo de habilidades de pensamiento computacional y puede facilitar el aprendizaje de la programación informática y la adquisición de las habilidades para la resolución de problemas gracias a que generan motivación y participación por parte de los estudiantes (Kazimoglu et al., 2012).

2.10.2 El *coding* en modo *unplugged* (desconectado)

El término *coding unplugged* se refiere a aquellas actividades que utilizan herramientas no digitales para la realización de proyectos que introducen a los conceptos fundamentales de la informática y de la lógica de la programación. El elemento común que caracteriza las actividades *unplugged* es el hecho de no

utilizar computadora, tablet ni otros dispositivos digitales: es el mundo análogo al papel, los colores, los juegos, las cuadrículas.

Entonces, la adquisición de las habilidades del pensamiento computacional puede ocurrir también en ausencia de herramientas tecnológicas (Shute et al., 2017). En los últimos años han sido cada vez más las prácticas didácticas que involucran a niños y jóvenes en la realización de actividades analógicas (Manilla et al., 2014), a través de las cuales se adquiere experiencia en cuanto a la lógica y a los conceptos base de la programación que son necesarios para el desarrollo del pensamiento computacional (del Olmo-Muñoz et al., 2020). A continuación, se mencionan brevemente algunas iniciativas relacionadas a actividades *unplugged*.

Un primer ejemplo es el proyecto promovido por la Universidad de Canterbury, en Nueva Zelanda, y apoyado por la hacienda americana Google, Computer Science Unplugged (CS Unplugged) que contribuye a la divulgación y a la enseñanza de la informática a través de un enfoque insólito dirigido a los estudiantes de todas las edades precisamente sin el uso de la computadora. Desde el sitio es posible acceder a material completamente gratuito que puede ser usado por quién lo desee. Estos instrumentos son elaborados, basándose en un enfoque constructorista, que renueva y rediseña las bases de la programación. Prácticamente se trata de una propuesta de enseñanza del pensamiento computacional a través de juegos y rompecabezas que involucran a quienes los llevan a cabo a través de uso de papeles, cuerdas, colores y mucho movimiento físico. Los estudiantes aprenden de esta forma a conocer los números binarios y algoritmos sin utilizar la programación del ordenador (Angeli et al., 2016; Bell et al., 2009).

El llevar a cabo las actividades *unplugged* lleva hacia la resolución de problemas relacionados con conceptos fundamentales de la informática, de esta forma, los niños ven el ordenador como una herramienta en vez de encuadrarla como el objeto de estudio. Argumentos con cierta complejidad como los

algoritmos, los datos, el diseño de interfaces y modelos de cálculo pueden abordarse sin requerir ninguna experiencia técnica previa. En muchos casos, los niños, se apasionan a los temas propuestos que de otra forma serían para ellos complejos y abstrusos (Bell et al., 2009). Las actividades propuestas por *CS Unplugged* se encuentran en un libro, traducido en varios idiomas, en el que se pueden encontrar actividades fáciles y divertidas, diseñadas para los estudiantes de todas las edades, que introducen al aprendizaje de la Informática, sin el uso de una computadora. El proyecto nace de la idea de que frecuentemente la informática se enseña a través de la programación, pero no todos los estudiantes la encuentran particularmente motivadora, convirtiéndose así en un obstáculo importante para su difusión (Bell et al., 2015).

Las actividades *unplugged* son aconsejadas para los planes de estudio ACM K-12 (Association for Computing Machinery) (Bell et al, 2009); de hecho, se prestan para utilizarse en la didáctica, tanto en los programas de enriquecimiento y profundización como en las actividades normales. Muchos argumentos se pueden perfectamente sobreponer a las disciplinas del plan de estudios, por ejemplo a los conceptos propios de las matemáticas y de la tecnología, en los que se estudian los números binarios, gráficas y mapas, modelos, problemas de clasificación y criptografía, y se aprende como está hecha una computadora.

Los estudiantes se involucran activamente en situaciones que promueven la resolución de problemas, y que desarrollan habilidades de comunicación, la creatividad y las habilidades del pensamiento computacional en un ambiente significativo y divertido (Bell et al, 2015).

Así como lo ha hecho *CS Unplugged*, también *Computer Science for fun* (CS4FN) promueve la difusión de los conocimientos de las bases de la informática de manera divertida y apoya la enseñanza y la difusión del pensamiento computacional (Curzon, 2013). *CS4FN* es una página web, con una revista digital asociada, que proporciona recursos que ilustran el lado divertido

de la informática y contribuyen a resolver problemas y crear artefactos en el mundo real. El proyecto se elaboró en la Queen Mary University di Londra en el 2005 para responder a la necesidad de aumentar el interés en la Informática, más allá de los planes de estudios, para alentar a los estudiantes a ver la informática como un "hobby" (Curzon, 2013).

CS4FN a través de su sitio, apoya a los profesores de todo el mundo proporcionándoles hojas de actividades y otros recursos que pueden inserir en su didáctica cotidiana con el objetivo de mejorar y apoyar el desarrollo de habilidades de pensamiento racional, de pensamiento lógico y la capacidad de utilizar las diversas formas de abstracción. CS4FN posee una revista impresa que trata, de forma divertida y accesible, argumentos relacionados con la informática; las revistas se distribuyen de forma gratuita en las escuelas del Reino Unido además de publicarse en el sitio web para poder descargarlas en formato *.pdf*.

La experiencia muestra que la enseñanza de las habilidades de pensamiento computacional no requiere necesariamente instrumentos digitales. Son muchos los ejemplos de actividades didácticas llevadas a cabo en modo "desconectado" que se encuentran en la red y, muchos de estos, han sido objeto de experimentación, como el sitio *code.org* que, como ya se ha mencionado, propone lecciones tradicionales de forma *unplugged* pensadas para transferir los conceptos de la informática y desarrollar el pensamiento computacional sin computadora ni conexión a internet.

Existen también ejemplos de didáctica *unplugged* que utilizan el diseño, como *pixel art*, para practicar sobre una cuadrícula para colorear con la cual los alumnos descubren un dibujo escondido dentro de una secuencia de comandos, simulando los pixeles de la pantalla de la computadora.

Kim et al (2013) han estudiado los efectos de la estrategia de programación en papel y lápiz (Paper and pencil strategy) que se refiere a un

modo de representar de manera lógica una idea con cualquier representación que pueda ser creada usando papel y lápiz con el fin de mejorar la comprensión, el uso del pensamiento computacional y aumentar el interés por el aprendizaje de la informática. Los resultados de dicha investigación han demostrado la eficacia de la enseñanza de la informática sin el uso del ordenador (Shute, 2017).

El estudio *cuasi experimental* llevado a cabo por del Olmo-Muñoz, Cózar-Gutiérrez & González-Calero (2020) se dirigió a estudiantes de las clases de segundo de Educación Primaria (n= 84) con el objetivo de explorar los eventuales beneficios sobre el desarrollo del pensamiento computacional con un enfoque mixto que combina tanto actividades online como *unplugged*. El grupo experimental, respecto al de control, llevó a cabo actividades online y *unplugged* demostrando que la inclusión de estas últimas en la enseñanza aporta beneficios tanto en el desarrollo de las habilidades del pensamiento computacional como en cuanto a la motivación y diferencias de género.

Gracias a estas investigaciones se está reforzando la idea de que el *coding unplugged* representa una válida oportunidad en el campo de la didáctica para favorecer el desarrollo del pensamiento computacional a través de un enfoque informal a los argumentos y de manera puramente lúdica, incluso sin el uso de instrumentos tecnológicos (Bocconi et al, 2016). En los libros de texto, en los últimos años, en los distintos cursos escolares, se han comenzado a encontrar propuestas de actividades *unplugged* a partir de la escuela de la Infancia.

2.11 Coding y habilidades sociales

Son varias las investigaciones que analizan los resultados asociados a la adquisición de las habilidades informáticas o relacionadas al desarrollo del pensamiento computacional cuando se aprende a través del *coding* (Grover & Pea, 2013; Lye & Koh, 2014).

Existen trabajos de investigación que, además de evaluar el alcance de las

habilidades antes mencionadas, examinan otras, preguntándose si, cuando los estudiantes programan, además del pensamiento computacional, adquieren otras habilidades y si su promoción puede servir como estímulo para favorecer la participación computacional clave de un cambio social muy interesante (Kafai & Burke, 2015). Se ha encontrado que, a través del *coding*, se adquieren habilidades que van más allá de saber programar y estas habilidades se desarrollan aún más cuando se sigue un adecuado proyecto pedagógico a través de la inclusión de prácticas de *coding* en el plan de estudios (Popat & Starkey, 2019).

La nueva Recomendación sobre las habilidades clave para el aprendizaje permanente adoptada el 22 de mayo del 2018 en sustitución de la anterior Recomendación del Parlamento Europeo y del Consejo, del 18 de diciembre del 2006 se enfoca en las habilidades personales y sociales, considerando necesaria entre los jóvenes el crecimiento de más habilidades empresariales, sociales y cívicas, necesarias para mejorar la capacidad de control de la propia vida, de adaptarse a los cambios, de resiliencia, de reflexión sobre uno mismo y, resaltando que, dichas habilidades están relacionadas al uso del *coding*.

Resnick et al. (2009) consideran el *coding* una herramienta que permite expresarse y compartir las ideas, trabajar de forma colaborativa con el consecuente logro de habilidades sociales. Sin embargo, hoy en día, el consenso sobre la importancia de dichas habilidades, relacionadas principalmente a la capacidad de comunicar y trabajar no es unánime y en algunos estudios parece ser fruto de una situación casual que se da durante la fase de aprendizaje de los estudiantes que usan el *coding* (Falloon, 2016). También Kaleliouglu (2015) y Fessakis et al. (2013) hablan de habilidades sociales (*social skills*) definiéndolas como la capacidad de comunicar de manera eficaz con otras personas y especifican que no era un resultado de aprendizaje planeado en su trabajo, sino que se dio en el contexto de la clase.

Sin embargo, en la mayor parte de los casos, prevalece la convicción de que los estudiantes que aprenden a programar logran obtener resultados educativos más amplios en cuanto al pensamiento crítico, la resolución de problemas, las habilidades sociales, el autocontrol en presencia de un programa didáctico eficaz y completo (Popat & Starkey, 2018).

2.11.1 École 42: la escuela de *coding* para todos

Como ya se ha descrito antes en esta tesis, el caso de *École 42*, una escuela de *coding* privada y gratuita, abierta en París en el 2013 con un modelo educativo innovador que está siendo tomado como ejemplo en muchos países del Mundo, muestra como la educación puede cambiar y proporcionar nuevos estímulos, también desde el punto de vista social. Esta escuela se enfoca en las habilidades y capacidades de cada chico que para ingresar a ella no requiere ningún título en específico (Peris-Ortiz et al., 2017). De hecho, el proceso de admisión se basa exclusivamente en el mérito y no toma en cuenta los antecedentes, de los créditos escolares y extraescolares.

La escuela fundada por Xavier Niel, un millonario emprendedor francés, se enfoca dar importancia a los méritos, en el reconocimiento del derecho de todos a tener la oportunidad de expresarse y de tener una educación de calidad. La falta de programadores en un País como Francia, quinta potencia económica mundial, dominado por la tecnología, y la presencia de escuelas de alto costo que impiden a los estudiantes más pobres obtener los mejores trabajos, han motivado a Xavier Niel y sus colaboradores a crear una escuela gratuita en la cual cuentan competencia, resistencia e igualdad social.

La escuela tradicional incluso siendo buena según sus fundadores ofrece una formación de bajo rendimiento, poco creativa y muy técnico-científica dando espacio a hábiles ejecutores, pero a pocos talentos.

École 42 utiliza un enfoque didáctico distinto al de las otras instituciones escolares ya que los progresos en el aprendizaje se evalúan gracias a los

procesos de revisión y de crítica interna entre los mismos estudiantes que tienen una edad entre los 18 y 30 años. En la base de la formación está el *coding* y las habilidades técnicas que se adquieren abren las puertas hacia todas aquellas profesiones que utilizan la programación como instrumento de trabajo. A estas habilidades imprescindibles se suman otras aún más innovadoras, como la capacidad de saberse adaptar, de pensar de manera diferente, de elaborar nuevas soluciones, de hacer equipo, y de utilizar la creatividad.

Su formación dura entre tres y cinco años y solo después de haber superado una primera selección online y posteriormente la llamada *Piscine*, una inmersión completa en programación hecha de pruebas, ejercicios y proyectos en lenguaje C con duración de cuatro semanas que los estudiantes deben superar para ser seleccionados y admitidos a estudiar gratuitamente en el Instituto.

El método de enseñanza propuesto no contempla la presencia de docentes sino de un equipo pedagógico que propone proyectos sobre los cuales trabajar cuando y cuando se desea aprendiendo de igual a igual. Es tarea de los mismos estudiantes llegar a la mejor solución de forma completamente autodidacta y evaluar entre ellos el trabajo realizado.

La escuela no otorga ni diplomas ni títulos, pero ofrece oportunidades de trabajo imperdibles ya desde antes de completar el curso. En el *École 42*, a través de la experiencia se construye el conocimiento y se educa para la vida, justo porque dentro de la escuela se consigue un estilo de vida, hecho de independencia y, al mismo tiempo, de reglas de conducta que deben ser respetadas por todos.

El modelo de *École 42* que promueve la igualdad social de los estudiantes que son admitidos en la escuela, sin importar sus orígenes ni condiciones, ha sido replicado en los Estados Unidos en la Silicon Valley, con el objetivo de involucrar a muchos estudiantes americanos en la *School 42* que, al igual que su homónimo en francés, ofrece una serie de oportunidades que van desde la

Educación hasta la posibilidad de aprender las habilidades necesarias para la vida real y el mundo del trabajo, y actualmente, es un ejemplo para muchos países, como Italia donde ya se proyecta abrir otra *École 42*.

Hoy en la actualidad, *École 42* se ha convertido en una red internacional con campus en veinte ciudades de todo el mundo y también en Italia, en Roma, con el nombre de *42 Roma Luiss*. Todas las escuelas satélite comparten el mismo proceso de selección que, como ya se ha mencionado, se denomina *Piscine* e independientemente de la ubicación geográfica del campus, todos los estudiantes tienen garantizada una educación de alta calidad y la posibilidad de completar su formación en uno o varios lugares del extranjero.

En cada sede, los estudiantes siguen un camino común, al tiempo que desarrollan proyectos específicamente adaptados al contexto, la cultura y el ecosistema de cada campus/zona geográfica y económica en la que se encuentra.

2.12 Coding en las disciplinas STEM

Las disciplinas del área STEM, en los últimos años, han adquirido cada vez más importancia en los procesos de enseñanza ya que se consideran fundamentales para el desarrollo social de un País; representan los argumentos claves de una educación que mira al futuro, orientada a formar ciudadanos capaces de competir, actuar y controlar su propio futuro. La adquisición de dichas habilidades inicia en los contextos escolares que deben direccionarse hacia experiencias positivas y gratificantes ya desde la escuela de la infancia y continuando en la adolescencia cuando se empiezan a definir las pasiones y los intereses. Con el acrónimo STEM no solo se refiere a cuatro disciplinas autónomas sino también a su integración en una nueva visión educativa basada en aplicaciones auténticas.

La experimentación de las trayectorias didácticas orientadas al desarrollo del pensamiento computacional actualmente requerida y fuertemente deseada

en el mundo lleva a los docentes a dedicar mayor tiempo y atención a las disciplinas STEM consideradas indispensables para quien se enfrentará en algunos años al mundo del trabajo (Tang et al., 2020).

De hecho, el *coding*, representa uno de los primeros recursos que pueden introducirse en la escuela como actividades transversales, en todos los ámbitos del saber, y, sin duda alguna, en las disciplinas del área STEM (Sengupta et al., 2018).

Así pues, uno de los enfoques para proponer actividades transversales en el ámbito de la educación STEM consiste en introducir a los estudiantes a la informática a través de la programación o el *coding* para desarrollar también habilidades y conceptos de alfabetización de base para todos los estudiantes (García-Peñalvo, 2016; Prensky, 2011). Existe una idea generalizada de que el pensamiento computacional y el desarrollo de los conceptos matemáticos están relacionados entre sí, en la perspectiva de la resolución de problemas, sin embargo, las investigaciones al respecto son limitadas en cuanto a los métodos y herramientas que deben aplicarse en las clases de las escuelas primaria (Miller, 2019).

Por lo tanto, con el presente trabajo se busca contribuir a dar valor a las actividades de *coding* al interno de los planes de estudios escolares de disciplinas como las matemáticas y la geometría en Educación Primaria, con el fin de proponer soluciones y trayectorias que respondan a los requerimientos de estos tiempos, buenas habilidades en las áreas STEM y, al mismo tiempo, habilidades relacionadas al pensamiento computacional.

La enseñanza y el aprendizaje de las matemáticas a través de la programación (Clements et al. , 2001) y el uso de robots programables (Highfield, 2014), han mostrado como dichos instrumentos como apoyo de la enseñanza de la disciplina promueven el aprendizaje, además de estimular las habilidades para la resolución de problemas, los conceptos geométricos

espaciales y de medida (Savard & Highfield, 2015). De esto estaba completamente convencido Seymour Papert pero, incluso en tiempos más recientes, con estudios de tipo cuantitativo como el que llevó a cabo Lewis y Shah (2012) quedó demostrada la relación entre las actividades de *coding* llevadas a cabo con la ayuda de Scratch y las puntuaciones obtenidas en las pruebas de matemáticas para los estudiantes de edades entre 11 y 12 años (n=47).

Las matemáticas, por ejemplo, pueden ser un válido entrenamiento para guiar a los alumnos hacia la comprensión de sus conceptos contextualizándolos a través del *coding*; frecuentemente, las lecciones de matemáticas resultan aburridas y enfocadas a la memorización de hechos y reglas; también la geometría viene presentada bajo el aspecto epistemológico descuidando la comprensión de los conceptos relacionales y el rol activo del estudiante en el proceso de aprendizaje (Crompton et al., 2019).

La competencia matemática incluye también la geométrica, que, en Educación Primaria, está relacionada principalmente a las habilidades visuales-espaciales y que a veces, es poco estimulada. Dentro del plan de estudios de matemáticas de Educación Primaria, que incluye también la geometría, es posible integrar los conceptos de la programación y experimentar las ventajas del *coding*, como enfoque metodológico (Foerster, 2017).

La idea de utilizar el *coding* como instrumento didáctico no es del todo nueva; como ya se ha dicho, en los años ochenta se llevaron a cabo muchos estudios sobre la introducción de la programación informática en las escuelas a través de lenguajes como LOGO (Papert, 1980), diseñado, además de como un lenguaje de programación real, como herramienta de aprendizaje y de promoción de la creatividad de los más pequeños.

En el ámbito matemático, los estudios han demostrado mejoras en los niños asociadas al uso de LOGO en la escuela, en particular en la clasificación de

las figuras, en el desarrollo del pensamiento geométrico, en el conocimiento de los ángulos, pero también en las habilidades metacognitivas (Clements, 1986). A pesar del resultado inicial, se registraron en general, resultados contradictorios tanto que en los años noventa las investigaciones en el campo educativo K-12 tuvieron un retroceso a causa principalmente de los problemas de sintaxis que presentaba el lenguaje *LOGO* (Resnick, 2013); sin embargo, cuando se introdujeron en clase nuevos lenguajes de programación visual, se asistió al renacimiento del *coding* como práctica de enseñanza para desarrollar nuevas habilidades y mejorar el aprendizaje y la motivación, superando los límites de *LOGO* pero no sus principios que sirvieron de inspiración.

Muchos investigadores, inspirados por el trabajo de Seymour Papert, han reiniciado sus trabajos destinados a investigar los posibles efectos positivos del *coding* en el desarrollo de las actividades didácticas, haciendo propias las teorías constructivistas. Algunos han diseñado actividades matemáticas aprovechando el potencial de los ambientes de programación como Scratch para explorar y expresar una gama de ideas matemáticas a través del *coding* (Benton et al., 2017).

La enseñanza de las matemáticas, según Papert y sus seguidores, a menudo ocurre mediante el uso de un modelo disociado, que ofrece pocas ocasiones a los alumnos que deben dar un sentido a lo que están aprendiendo: "Las verdaderas matemáticas para los niños, que sean dignas de este nombre, no pueden ser algo que queramos infligirles, como una medicina desagradable, sin encontrar ninguna razón para tomarla nosotros mismos" (Papert, 1986).

La geometría, uno de los campos más importantes del conocimiento de las matemáticas, se basa en conceptos abstractos, como el punto y la línea, que son muy difíciles de comprender para los niños.

La geometría de la tortuga de Papert se presenta al niño de manera alternativa respecto a las abstracciones geométricas y matemáticas; tiene un estilo distinto al de Euclides o de Cartesio pero de cualquier manera es efectivo.

La tortuga, a diferencia del punto, no es estática: puede moverse y orientarse. Es como un ser humano, un animal o un medio de transporte: para los niños la tortuga representa su alter-ego, se identifican con ella gracias al conocimiento que tienen sobre su cuerpo. Trabajando con la tortuga ponen en práctica los conocimientos adquiridos sobre la *geometría del cuerpo* para abordar la geometría formal, a través de un *estilo informático* (Papert, 1986). La estrategia del enfoque propuesto por Papert lleva a pasar de situaciones familiares a las desconocidas, poniendo al niño en contacto con ideas importantes, como la organización jerárquica, la escritura de un plan de trabajo, la corrección de los errores y los alienta al uso consciente de las estrategias matemáticas y de la resolución de problemas.

La geometría de la tortuga enseña a pensar de forma heurística, es decir a encontrar la mejor estrategia para la resolución de un problema, aprovechando de manera eficaz los propios recursos, se ha diseñado para que los niños puedan dar un sentido a aquello que están aprendiendo.

Concretamente, el programa *LOGO*, haciendo que la tortuga se mueva en la pantalla, estimula al niño a resolver problemas geométricos a través de una programación intrínseca y una visualización inmediata global que permite el verificar si el algoritmo en movimiento es correcto.

A través de *LOGO* es posible consolidar los conceptos propios de la geometría como: rotación, desplazamiento, ángulo, polígono, plano cartesiano, transformación isométrica y aquellos que son conceptos típicos de informática como el concepto de procedimiento y de programación como tal.

La integración de la programación y de los algoritmos en el plan de estudios de la geometría abre nuevos escenarios de aplicación, también en cuanto a la mejora significativa que se observa en el aprendizaje de los estudiantes y en su conciencia con respecto a la geometría (Crompton et al., 2018).

Describir una figura geométrica significa proporcionar un algoritmo apropiado y verificar de manera algorítmica si cada fase de construcción es correcta (Foerster, 2016); la práctica del *coding* dentro del plan de estudios de matemáticas y geometría, en Educación Primaria, aunque fue objeto de estudio en el pasado con Papert, ha vuelto a la vanguardia de la investigación educativa con el fin de estudiar su influencia sobre la adquisición de los conocimientos geométricos pero también sobre la motivación y la conciencia de los estudiantes. En algunos estudios se ha buscado entender si el uso del *coding* durante la práctica de actividades curriculares de matemáticas o geometría pueda satisfacer los requisitos de un curso de introducción de programación en ámbito informático (Foerster et al., 2018) ya que, en muchas naciones europeas no se ha incluido la informática como materia de estudio.

El lenguaje de programación Scratch puede verse como una continuación de los ideales del LOGO. De igual manera, también las herramientas de robótica educativa vienen actualmente utilizadas en el contexto educativo para hacer, por ejemplo, actividades de geometría y desarrollar habilidades visuales-espaciales.

2.12.1 Estudios recientes sobre los efectos del *coding* en la enseñanza de matemáticas y geometría

A continuación describiremos algunos estudios efectuados en el campo matemático, a partir de la introducción de los nuevos lenguajes de programación visual con referencia al impacto educativo del *coding* asociado a las actividades de aprendizaje.

Zavala, Gallardo y García-Ruiz (2013) han demostrado, con un estudio cuasi experimental, que el uso del *coding*, y en particular de Scratch, durante el desarrollo de una actividad de matemáticas que tiene que ver con la construcción de secuencias numéricas ha mejorado el desempeño de los niños de la tercera elemental comparando los resultados entre el grupo experimental (n= 27) compuesto por niños de ocho y nueve años y el grupo de control (n= 22) compuesto por niños de la misma edad de la misma Educación Primaria mexicana, en la cual seguían el mismo plan de estudios.

Calao, Moreno-León, Correa, & Robles (2015) llevaron a cabo una investigación cuasi experimental con el fin de verificar si el uso del *coding* en las lecciones de matemáticas, en una clase de sexto curso (n= 42), pudiera tener un impacto positivo sobre el aprendizaje, sobre la capacidad de formular los problemas y resolverlos, sobre el razonamiento lógico y sobre la ejecución de procedimientos y algoritmos. Se encontraron cambios significativos en la comprensión de los procesos matemáticos después del uso de Scratch.

De igual manera, Ke (2014) reportó resultados positivos para n= 64 niños de una clase de la escuela secundaria de primer curso que participaron en actividades de realización de juegos matemáticos con Scratch.

Los datos se recogieron a través de las actividades, la observación y la conversación, el análisis de los artefactos, las entrevistas y los sondeos; los resultados del estudio, aunque no se pueden generalizar a una población más amplia, muestran que los participantes desarrollaron una disposición mucho más positiva hacia las matemáticas después de haber llevado a cabo la actividad.

Foerster (2017) estudió la integración de Scratch en lecciones de geometría para los estudiantes de 12/13 años de Alemania, mostrando que los objetivos de conocimiento matemático pueden alcanzarse de una mejor manera con respecto a los grupos de control que utilizan los métodos tradicionales. Ha encontrado

efectos positivos a largo plazo con respecto a la experiencia de combinar las lecciones de geometría con las actividades de programación.

Psycharis y Kallia (2017), con un estudio cuasi experimental han demostrado que la enseñanza de las matemáticas en combinación con el *coding* ha mejorado de manera significativa la resolución de los problemas matemáticos, aunque dicha mejora no es estadísticamente significativa con respecto al grupo de control que aprendió las matemáticas sin el uso de la programación informática.

Benton, Saunders, Kalas, Hoyles & Noss (2018) han querido aprovechar el potencial que ofrece Scratch para estudiar el aprendizaje de las matemáticas y de la geometría en las escuelas primaria de Inglaterra junto a las habilidades de programación, enfrentando temas como los números positivos y negativos, las expresiones algebraicas, el plano cartesiano, observando un resultado positivo en los niveles de empeño y de entusiasmo entre los alumnos.

Foerster et al. (2018) han llevado a cabo su investigación en una clase de la escuela secundaria de primer curso que utiliza el *coding* durante el desarrollo de las actividades del plan de estudios de matemáticas, evaluando las mejoras en cuanto a las habilidades geométricas y de programación.

Miller (2019) realizó una investigación experimental a través de la aplicación de un enfoque didáctico sobre un pequeño grupo (n= 40) que realizó actividades de *coding* por un periodo de seis semanas. Los datos recogidos en la prueba previa y la prueba final confirman que a través del *coding* es posible alcanzar buenos resultados en matemáticas especialmente en cuanto a la identificación de los modelos y de las estructuras matemáticas.

En resumen, la reciente colección de estudios sobre los efectos del *coding* con los nuevos lenguajes de programación visual en términos de aprendizaje y motivación muestra efectos positivos en el área de las matemáticas, así como en

el ámbito lingüístico, en el campo de la modelación-simulación de complejos sistemas físicos o biológicos.

Ahora nos enfrentamos a la pregunta de si es posible incorporar el *coding* en el sistema educativo italiano, preguntándose si el proceso de aprendizaje propuesto a través del uso de lenguajes de programación puede abordarse mediante el conjunto de recursos, herramientas, sistemas y metodologías actualmente disponibles.

2.13 Recursos educativos para integrar el *coding* en la escuela

A continuación, se mencionan algunos de los recursos para la enseñanza-aprendizaje de materias como las matemáticas o la geometría a través de la práctica del *coding* que se pueden utilizar en la práctica didáctica. En particular se han seleccionado aquellas que se tradujeron al italiano y que pueden facilitar la comprensión inmediata de comandos e instrucciones a los docentes y niños. Dichos recursos, por comodidad, se clasifican en base a los tipos y las características principales del lenguaje de programación que contienen:

- Los lenguajes visuales con flechas, como *ScratchJr* y *Bee-Bot App*, guían al estudiante a programar de manera visual una secuencia de comandos de flechas e iconos similares intuitivos. Estos lenguajes de programación están proyectados para los pre-lectores y pueden utilizarse al final de la escuela de la infancia o al inicio de Educación Primaria. Por ejemplo, *ScratchJr* propone módulos curriculares que refuerza los estándares de alfabetización y matemáticas.
- A través de los lenguajes visuales por bloques el estudiante programa de forma visual una secuencia de comandos usando bloques o partes de códigos mediante el establecimiento de diversos parámetros. Estos lenguajes requieren la capacidad de lectura y por lo tanto están diseñados para las escuelas primarias y secundarias de primer curso. Algunos

ejemplos de recursos que utilizan lenguajes visuales de bloques son *Scratch*, *Blockly*, *code.org*.

Las actividades que se pueden llevar a cabo a través de estas plataformas pueden tener el fin de resolver problemas cerrados, en lo que el alumno escribe un programa que sirve a resolver un problema que le ha sido dado, incluso de tipo geométrico. Por ejemplo, desde la plataforma de *Blockly* es posible hacer diseñar a una tortuga figuras geométricas regulares utilizando los bloques de instrucciones propuestos.

En la actividad de la hora del código de nombre *El Artista*, se enseña como componer los bloques (las instrucciones) que el personaje, el artista en este caso, debe realizar en la pantalla para trazar figuras geométricas regulares.

El resultado es una descripción constructiva inusual de imágenes que los niños generalmente están acostumbrados a considerar en su totalidad. Llevando a cabo los ejercicios, se le toma confianza al concepto de pixel y a los nuevos comandos para hacer mover al personaje y hacerlo girar un cierto número de grados. Sucesivamente se aprende a utilizar el bloque "repetir" para hacer que el ordenador repita las instrucciones varias veces, sin necesidad de que estas sean escritas cada vez. El comando es muy útil en el caso de figuras geométricas regulares, cuya construcción presenta esquemas que se repiten. Una vez comprendido el concepto de repetición, el estudiante puede aplicarlo no solo a las partes que constituyen a la figura, sino a las figuras mismas, logrando crear figuras complejas basadas en la repetición de formas geométricas de base.

Además es posible crear proyectos abiertos, es decir nuevos objetos digitales ya sea escribiendo el programa entero desde cero o re mezclando parte de códigos escritos por otros usuarios.

Por ejemplo, al trabajar en el diseño generativo es posible experimentar el uso de la programación y del pensamiento computacional para la generación de

patrones y diseños geométricos que surgen de la aplicación y de la variación de reglas lógico-matemáticas.

En este sentido, Scratch resulta un recurso eficaz, ya que puede generar todo tipo de objetos digitales, gracias también a su versatilidad de aplicación en el contexto escolar. Se pueden explorar las leyes de la geometría que llevan al diseño de una figura geométrica o de una progresión de polígonos (Fig. 4) o reproducir con el *coding* e estructuras e imágenes que se basan en las propiedades de la simetría, todas las actividades que se integran perfectamente con los temas del plan de estudios del área lógico-matemática.

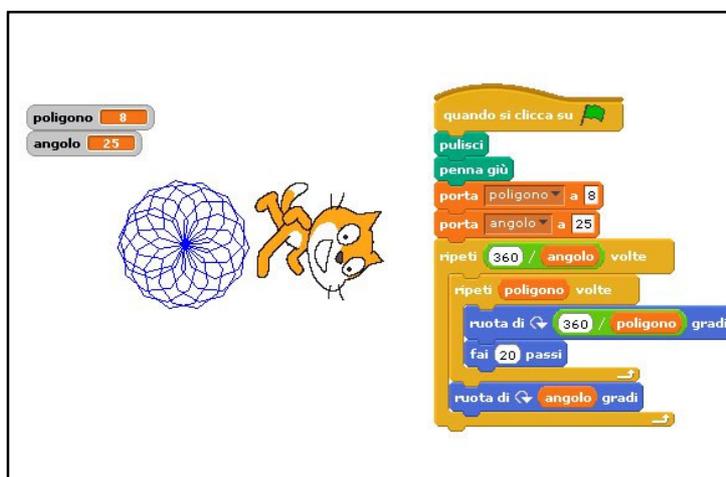


Figura 4. Dibujar polígonos con Scratch

Otro recurso que es posible integrar en las actividades didácticas del área lógico-matemática de Educación Primaria es el *pixel art*, que puede proporcionar al profesor ideas didácticas relacionadas al pensamiento computacional y a las habilidades digitales.

A través del *pixel art*, es posible descomponer una imagen en píxeles: dicha práctica vuelve posible el dictado gráfico que presupone una convención rigurosa entre quien dicta y quien ejecuta el diseño, completamente análoga a la convención entre el programador y ejecutor típica del *coding*. Aunque es difícil encontrar un diseño hecho a mano por un niño que sea igual al de otro, todos

los diseños parecen ajustarse de manera convencional al original. Las actividades de *pixel art*, tanto de forma *unplugged* como digital, guían a los alumnos a través de un recorrido que los lleva a la escritura de algoritmos que ejecutan diseños geométricos o abstractos, o diseños simétricos o pueden ser el inicio para abordar los conceptos de cálculo y del plano cartesiano. Aunque el ambiente de trabajo inicial es visual y gráfico, los ejercicios que deben realizarse involucran también habilidades lingüísticas y lógico-matemáticas.

Un ejemplo es la aplicación del concepto de orto-convexidad a través de los conceptos de la llamada *geometría del taxi* de Minkowsky en un ambiente no euclidiano.

Con este capítulo se cierra la primera parte de la presente tesis doctoral, en la cual se ha buscado profundizar y analizar el argumento con respecto a los métodos de difusión del pensamiento computacional en el panorama educativo K-12, desde el punto de vista de la investigación.

Actualmente, no existe una definición concordada del concepto de pensamiento computacional lo cual lleva a una notable dispersión y confusión en cuanto a la terminología relacionada con este concepto.

De igual manera, se está difundiendo cada vez más rápido, gracias también a peticiones explícitas que provienen del mundo de la política, se busca introducir en el campo de las intervenciones educativas actividades de promoción del pensamiento computacional para integrarlas a los planes de estudios. Existe una notable variedad de programas, metodologías y recursos. En este contexto, el *coding* parece representar una herramienta válida para utilizarse en el proceso de enseñanza/aprendizaje de todos los cursos escolares integrándolo en las actividades curriculares y en particular en las disciplinas del área STEM.

2.14 Políticas educativas que apoyan el pensamiento computacional

2.14.1 Iniciativas europeas

Son muchas las preguntas que surgen entre los investigadores científicos en cuanto a las estrategias que hay que llevar a cabo para favorecer el desarrollo del pensamiento computacional desde temprana edad en el mundo escolar.

El cambio educativo debe pasar necesariamente a través del mundo de las políticas educativas, que debe estar informado sobre la naturaleza y la importancia del pensamiento computacional, sobre las formas para su integración en los planes de estudios, en la definición de los objetivos de aprendizaje (Barr & Stephenson, 2011). Los mensajes que el mundo científico ha lanzado han sido acogidos por Europa, América y Asia.

La escuela tradicional se ha considerado siempre un sector estable, a la cual raramente se le aplican cambios en los planes de estudio. Sin embargo, en aproximadamente diez años, el pensamiento computacional se ha convertido en la palabra clave para la modificación de los planes de estudios de muchos países, que han adoptado las propuestas del mundo académico. De hecho, los responsables políticos han aceptado el mensaje lanzado por el mundo de la investigación con respecto a la introducción del pensamiento computacional en las escuelas.

En el 2006 el Consejo y el Parlamento Europeo publicaron la Recomendación sobre las "Habilidades Clave para el Aprendizaje Permanente". Entre las ocho habilidades clave está la habilidad digital, definida como la capacidad de saber usar con destreza y espíritu crítico las tecnologías de la sociedad de la Información.

En el 2013 la Comisión Europea publica el documento *Digicomp: a*

framework for developing and understanding Digital Competence in Europe en el que se encuentra el marco de referencia para las habilidades digitales consideradas necesarias para vivir como ciudadanos activos y conscientes.

Las veintiuna habilidades digitales están distribuidas en cinco áreas distintas. Es posible encontrar dos habilidades afines a los conceptos de pensamiento computacional:

- en el área 3 en el punto 4 se encuentra la habilidad de programación “aplicar configuraciones, modificar programas, aplicaciones, software, herramientas” y “comprender los principios de la programación y que hay dentro de un programa”;
- el área 5 abarca la habilidad de resolución de problemas (problem solving).

En el 2014 *European Schoolnet*, una organización sin fines de lucro compuesta por treinta y cuatro ministros europeos de la Educación, que tiene el objetivo de aportar innovación en la enseñanza y el aprendizaje de todos los involucrados en el proceso educativo, llevó a cabo un sondeo entre sus miembros para conocer el estado del arte en el mundo escolar sobre el *coding*, dicha investigación fue llevada a cabo nuevamente en el 2010, cuando los esfuerzos y el compromiso de apoyar la iniciativa comenzaron a intensificarse, con el fin de introducir el *coding* como habilidad clave para el desarrollo de la economía y de la sociedad.

El documento elaborado *Computing our future – Computer programming and coding priorities, school curricula and initiatives across Europe*, presenta los resultados de la investigación realizada con veintiún ministros de la Educación (20 países de la U.E. e Israel) (Balanskat & Engelhardt, 2014).

En el documento se alude de forma explícita al *coding*, a la programación

informática y al pensamiento computacional como término típicamente relacionado con ellos y se resalta el deseo de que dicho concepto se vuelva parte integrante de los planes de estudios escolares.

En el reporte claramente se indica la importancia de hacer adquirir a los estudiantes en la sociedad actual, las habilidades relacionadas con el pensamiento computacional, como la capacidad de resolución de problemas y de pensamiento lógico y, en menor medida, para cubrir la ausencia de mano de obra calificada en Europa en el campo de la tecnología informática y alentar a los estudiantes de ambos géneros a emprender las carreras relacionadas a dichos argumentos.

Posteriormente surge la necesidad de dar respuesta a algunas preguntas de naturaleza pedagógica, ya destacadas en los artículos científicos que tratan el tema.

La investigación llevada a cabo a nivel europeo pone en evidencia las distintas causas que han llevado a los gobiernos locales a recurrir a la planificación de las actividades de *coding*. Los países participantes a este estudio fueron: Austria, Bélgica/Flandes, Bélgica/Valonia, Bulgaria, Dinamarca, Eslovaquia, España, Estonia, Finlandia, Francia, Hungría, Irlanda, Israel, Lituania, Malta, Noruega, países Bajos, Polonia, Portugal, Reino Unido, República Checa.

Para la mayor parte de los países es fundamental el desarrollo del pensamiento lógico (15 países) y de las capacidades para la resolución de problemas (14 países) entendidas como habilidades importantes que son necesarias en el siglo XXI.

Son once los países que prefieren el desarrollo de las habilidades clave y las habilidades de programación, once países que consideran esencial lograr impulsar a más jóvenes al estudio de las Ciencias de la Información y, finalmente ocho países que consideran importante lograr promover la creación de empleos

en el sector informático.

En el informe se usan frecuentemente los términos *computing* y *coding* de forma intercambiable, refiriéndose principalmente a aquellas actividades que permiten a los niños aprender a usar softwares específicos y aprender la programación de muchos dispositivos electrónicos. El término pensamiento computacional viene, en algunos países, usado en remplazo de los dos términos anteriores aunque tendencialmente es visto como una habilidad esencial para el desarrollo de la alfabetización del siglo XXI.

En base a la información obtenida, es posible detectar los países en los que ya se ha implementado la práctica del *coding* dentro los planes de estudio y a qué nivel (nacional, regional o local).

Son dieciséis los países que incluyen el *coding* en los planes de estudio en todos los niveles (Austria, Bulgaria, Dinamarca, Eslovaquia, España, Estonia, Francia, Hungría, Irlanda, Israel, Lituania, Malta, Polonia, Portugal, Reino Unido, república Checa; dos países (Bélgica/Flandes y Finlandia) han programado incluir el *coding*, en el 2016/17, en la formación básica; en cambio, tres países (Bélgica/Valonia, Noruega y países Bajos) aún no han integrado el *coding* en sus planes de estudio y no han indicado cuando pretenden integrarlo.

La investigación encargada al European Schoolnet responde a la invitación propuesta por la Agenda Digital para Europa presentada por la Comisión Europea que, en el 2010, lanzó 101 acciones agrupadas en torno a siete áreas prioritarias con el objetivo de promover condiciones para el crecimiento y el empleo en Europa, a través del uso de las tecnologías digitales. En el punto 3 se propone iniciar una gran coalición para las habilidades digitales y para la ocupación (Parlamento y Consejo de la Unión Europea, 2006), señalando la necesidad de que cada ciudadano tenga por lo menos las habilidades digitales básicas para vivir, trabajar, aprender y participar en la sociedad moderna.

La Agenda Digital es uno de los siete pilares de la estrategia EUROPA 2020 que pone los objetivos para el crecimiento de la Unión Europea (UE) para el 2020. Propone aprovechar lo mejor posible el potencial de las Tecnologías de la Información y la Comunicación (TIC) con el fin de promover el crecimiento económico y el progreso.

En el 2016 el Joint Research Centre (JRC), servicio de ciencia y conocimiento de la Comisión Europea, elaboró un documento que proporciona una visión general de los resultados más recientes de la investigación y de las iniciativas políticas llevadas a cabo, en Europa, para promover el desarrollo del pensamiento computacional como habilidad esencial de los estudiantes del siglo XXI.

Los datos obtenidos permiten delinear algunas recomendaciones sobre cómo abordar, desde el punto de vista político, la integración del pensamiento computacional en la educación obligatoria.

En particular, una definición compartida de lo que es el pensamiento computacional seguramente puede favorecer su integración a los planes de estudio, así como la experiencia que se ha adquirido hasta el día de hoy puede representar una valiosa contribución a la discusión política.

El documento muestra que todas las partes interesadas deben estar debidamente formadas e informadas y debe garantizarse la medida del impacto y de la sostenibilidad de las acciones que se ponen en práctica. Asume gran importancia la posibilidad de intercambiar información y experiencias, que no se repitan o que se conviertan en buenas prácticas, entre todos los individuos involucrados (responsables políticos, iniciativas de base, centros de investigación y otros sujetos interesados) que pueden agregar valor a las acciones políticas llevadas establecidas.

El término pensamiento computacional, según esta nueva investigación, se

ha convertido, en los últimos años, en un tema central para muchos investigadores, profesionistas y formuladores de políticas en el campo educativo.

La Agenda para las nuevas habilidades para la Europa (Comisión Europea, 2016) invita de manera explícita a los Estados miembros a promover la Computer Science (CS) en la educación, pidiéndoles invertir en la formación de las habilidades digitales.

Para este fin se busca entender cuál es el potencial del pensamiento computacional para la educación obligatoria documentando las implicaciones para la política y la práctica que provienen de informes técnicos y políticos, artículos científicos, sitios web y blogs, artículos de periódico, videoclip, presentaciones y MOOC.

La investigación tiene que ver principalmente con algunas áreas relacionadas al pensamiento computacional:

- terminología;
- integración curricular;
- preparación de docentes;
- estrategias de evaluación;
- relación con la alfabetización digital/habilidades digitales;
- relación con el *coding*/programación informática.

Por ejemplo, en cuanto a la terminología usada normalmente cuando se hace referencia al concepto de pensamiento computacional, se ha visto que el contexto influye en la elección de un término en específico.

Así pues, algunos países utilizan expresamente el término pensamiento computacional en la traducción del idioma en uso, otros consideran más

apropiado referirse a *coding* o problem solving o incluso a CS o habilidad digital.

Por su parte, el reporte *Developing Computational Thinking in compulsory education* (Bocconi et al., 2016), proporciona entonces un panorama de la situación hasta el 2016 en Europa incluyendo a Israel y Turquía, con particular referencia a las iniciativas políticas que tienen que ver con la reforma de los programas de estudio nacionales y las líneas guidas oficiales en las cuales se incluye la integración del pensamiento computacional en la escuela obligatoria.

Nuevamente se confirma la motivación que debería alentar a los gobiernos, tomando como ejemplo USA, a favorecer dicha integración, tanto para garantizar a los jóvenes y niños el pensar de forma diversa, resolviendo y analizando los problemas del mundo real desde distintas perspectivas como para estimular el crecimiento económico, cerrando la brecha que se ha creado en el mundo del trabajo por la falta de personas calificadas.

Existe una gran variedad en cuanto a la terminología utilizada y también en cuanto a los aspectos motivacionales y las estrategias a implementar; para este fin se establecieron tres grupos dentro de los cuales identificar los países europeos según el enfoque que utilizan.

El primer grupo está formado por Inglaterra, Francia, Finlandia, Polonia, Italia, Turquía, Dinamarca, Portugal, Malta, Croacia, Escocia. De este grupo forman parte los países (11) que en los últimos años (2011-2016) independientemente de las razones, han evaluado los planes de estudio, mejorando la enseñanza del pensamiento computacional y de los conceptos relacionados con este en la escuela obligatoria a nivel nacional. Inglaterra fue pionera de la conceptualización del pensamiento computacional que se ha inserido en el programa educativo de la educación obligatoria.

Del segundo grupo forman parte República Checa, Irlanda, Noruega, Gales, Grecia, países Bajos, Suecia. Este comprende los países que no han iniciado aún

a introducir el pensamiento computacional en la escuela obligatoria pero están trabajando para hacerlo, a través del apoyo de estudios oficiales presentados a los responsables políticos.

El tercer grupo formado por Austria, Chipre, Israel, Lituania, Hungría, Eslovaquia incluye a los países europeos que tradicionalmente han promovido el estudio de la CS en las escuelas secundarias superiores y que, tienden a buscar que se incluya el estudio de la Informática también en niveles inferiores (primaria y secundaria de primer curso).

Existen países en los que la integración y la difusión del pensamiento computacional suceden a nivel regional. Por ejemplo, en España a nivel político nacional no existe algún documento que haga referencia al pensamiento computacional sin embargo es posible encontrar similitudes con algunas comunidades autónomas.

Lo mismo se describe para Alemania, Bélgica y Suiza.

El trabajo muestra, como elemento más significativo, que resulta necesario que los responsables políticos comprendan que es ineludible incluir el pensamiento computacional no solo en la educación informal de los jóvenes sino también en la educación formal, planeando las actividades que se desean insertar, a través de objetivos específicos y nuevos enfoques. El intercambio de experiencias puede ser particularmente útil para futuros desafíos y nuevas iniciativas.

La *Escuela de Pensamiento Computacional* es, por ejemplo, un proyecto del Ministerio de Educación y de la Formación primaria, secundaria, bachillerato, profesional del Gobierno de España, que tiene como objetivo principal ofrecer recursos educativos abiertos, formación y soluciones tecnológicas que proporcionen a los docentes españoles los instrumentos útiles para integrar las habilidades asociadas al pensamiento computacional en la enseñanza a través de

la programación y la robótica, con las mayores garantías y con el mínimo esfuerzo. La iniciativa está coordinada por el Instituto Nacional de Tecnologías Educativas y de Formación del Profesorado (INTEF) en colaboración con los Departamentos de Educación de las Comunidades y de las Ciudades Autónomas.

Después de un periodo inicial de formación para los profesores que participan en el proyecto les será posible poner en práctica en sus clases lo que aprendieron.

El proyecto responde completamente a las indicaciones europeas y en este sentido se basa en el plan de acción para la educación digital de la Comisión Europea y se pretende contribuir a alcanzar algunos objetivos de Agenda 2030 para el desarrollo sostenible, como la educación de calidad, la igualdad de género, tener un trabajo digno, promover el crecimiento económico reduciendo la desigualdad dentro de los países y entre ellos.

2.14.2 El camino hacia la difusión del pensamiento computacional en Italia

Las iniciativas políticas emprendidas en Italia con respecto a la difusión del pensamiento computacional en la escuela están motivadas por la tendencia europea de proponer una apertura hacia la introducción de las habilidades directa o indirectamente relacionadas al mundo digital.

Ya en las *Indicaciones Nacionales para la escuela de la Infancia y 1º ciclo* (Cerini, 2012) se hace referencia explícita a las habilidades digitales y a la Informática como disciplina que lleva en sí misma un cierto rol cultural y transversal.

Come en muchos otros países, en Italia, el MIUR (Ministero dell'Istruzione, Università e Ricerca), en los últimos años, se ha vuelto promotor de iniciativas destinadas a innovar los procesos de producción del conocimiento, a través de una serie de nuevas normas e inversiones.

2.14.2.1 **Campañas de alfabetización informática y de difusión del pensamiento computacional**

La importancia de la promoción del *coding* como herramienta para desarrollar el pensamiento computacional se resaltó a través de campañas de alfabetización a gran escala como *Hour of Code*, *Europe Code Week*, *África Code Week*, iniciativas que han tenido la aprobación y la colaboración de empresas y de organizaciones gubernamentales.

El 2013 es el año que marca el inicio de las campañas de alfabetización, a nivel internacional, a partir de USA, y de divulgación del *coding* para el desarrollo del pensamiento computacional de manera organizada. De hecho, antes del 2013, estas iniciativas eran espontáneas, tenían distintos objetivos y no estaban legitimadas por las instituciones, por consecuencia, no eran promovidas a nivel gubernamental pues no se consideraban una prioridad. Las herramientas utilizadas eran las mismas.

Por ejemplo, las iniciativas, promovidas con Scratch y Coderdojo se iniciaron antes del 2013 y tenían como objetivo principal volver el *coding* accesible para todos.

Posteriormente, la finalidad de dichas campañas se asoció a la necesidad de compensar la falta de habilidades (skill gap), dando a conocer el potencial de la programación; se estimó que en Europa faltarían para el 2020, novecientas mil unidades de trabajo.

Aeste fin, contribuyó la campaña de la Comisión Europea *Europe Code Week* una iniciativa institucionalizada conducida por voluntarios, embajadores, de los Estados Miembros y otros. La primera campaña se llevó a cabo del 25 al 30 de noviembre del 2013. En USA entre el 9 y el 15 de diciembre del 2013 inició "la hora del código" que involucró también otros países del mundo. En Italia, en durante el mismo periodo, inició la iniciativa *Code's Cool* con el intento de reunir

profesores, alumnos y familias para crear una comunidad de aprendizaje destinada a la resolución de problemas de programación. Las nuevas campañas fueron anuales y se han organizado con criterios bien definidos.

En el 2014 el Consorcio Interuniversitario Nacional para la Informática (CINI), compuesto por todas las Universidades de investigación italianas activas en el campo de la Informática, trajo a Italia *code.org* a través de la creación de un sitio web en idioma italiano que ofrece, como red de Universidad, un soporte a quien estuviese interesado a utilizar los recursos presentes. El proyecto involucró el MIUR con el fin de sensibilizar el mundo escolar a través de comunicaciones oficiales provenientes directamente del Ministerio. En realidad, se dirige también al mundo de los adultos (centros de formación para los adultos, iniciativas de educación extraescolar, autoaprendizaje) partiendo de la idea de que aprender los conceptos fundamentales de la informática tiene un valor intrínseco y un rol práctico, pues ayuda a entender mejor las bases informáticas de los mecanismos sociales de la sociedad actual. El proyecto *Programa el Futuro* fue estructurado y presentado con el objetivo de favorecer el aprendizaje y el desarrollo del pensamiento computacional como habilidad clave para la educación moderna, resaltando la importancia del valor cultural de la informática más allá de sus aspectos técnicos y tecnológicos. Es necesario precisar que en Italia, antes de esta iniciativa, la enseñanza de la informática en Educación Primaria tenía la finalidad de enseñar a usar el procesador *Word* y las hojas de cálculo. Entonces, el objetivo del proyecto es estimular el desarrollo de una adecuada educación informática a todos los niveles escolares en Italia (Corradini et al., 2017).

En el 2015 se agregaron a las ya existentes, nuevas campañas de sensibilización que se inspiraron en la idea de que las habilidades relacionadas al pensamiento computacional tenían estereotipos y por lo tanto era necesario cerrar la brecha de la diversidad (*diversity gap*) entre las personas que tienen habilidades de programación, enriqueciendo la comunidad de programadores con nuevos integrantes.

Se piensa, por ejemplo, en la diversidad de género (Campagna Rosa Digitale) o a las distintas condiciones socioeconómicas.

Por otro lado, *Rosa Digitale* es un movimiento nacional para la igualdad de oportunidades en el ámbito tecnológico, que organiza manifestaciones que nacen como punto de partida hacia la igualdad de género en el campo de la tecnología y de la informática. El lema del movimiento es *The conquest is information* y tiene el objetivo de sensibilizar para la igualdad de oportunidades a través de la información.

Se difunde también en Italia la exigencia de desarrollar nuevas habilidades no solo relacionadas al mundo del trabajo sino que también favorezcan el crecimiento personal, a través del pensamiento computacional. La programación se convierte en una herramienta y para aprovecharla; se materializa la necesidad de promover su difusión en las escuelas para desarrollar habilidades para razonar de forma algorítmica y para la resolución de problemas complejos. Al mismo tiempo se cierra la brecha de la diversidad, ya que las escuelas acogen a todos los alumnos con distintas características; de forma análoga se cierra la brecha de habilidades (skill gap), dado que los jóvenes crecen desarrollando una mentalidad adecuada para conocer los temas que pueden ser parte integrante de su formación profesional.

2.14.3 Políticas educativas en Italia

El sistema educativo italiano se ha hecho cargo de promover la comprensión y el uso de las tecnologías digitales para que los estudiantes superen el rol de consumidores pasivos y se conviertan en ciudadanos capaces de aplicar el pensamiento computacional para resolver problemas y tomar las oportunidades que la sociedad ofrece (MIUR, 2017, mar, 3).

Los resultados de las pruebas internacionales muestran que aún hay mucho por hacer para mejorar las habilidades y capacidades de los estudiantes en matemáticas, aunque, en el 2018, Italia obtuvo una puntuación alineada con el

promedio OCSE en matemáticas: el rendimiento promedio en matemáticas se mantuvo estable (OECD, 2019).

Siguiendo las Recomendaciones Europeas, Italia ha iniciado un recorrido de innovación y digitalización, así como se prevé en el documento de la Buona Scuola (2014) que se convirtió en ley en el 2015 (Ley n.107 del 13/07/2015) (MIUR, 2015).

Gracias a la colaboración entre MIUR y CINI, se lanzaron algunos proyectos, de manera que se experimentó la introducción estructural en los institutos escolares italianos de los contenidos propios de la informática y del pensamiento computacional (MIUR, 2014, sept, 23).

El documento de la Buona Scuola antes de llegar a ser Ley, en el punto 4.2 señala:

“la próxima alfabetización: idiomas extranjeros, *coding*, economía”.

Surge la necesidad de introducir el *coding* y el pensamiento computacional a partir de Educación Primaria a través de un Plan Nacional y planea hacerlo en los siguientes tres años escolares (2014/15, 2015/16, 2016/17) con el objetivo de que “los alumnos aprendan a resolver problemas complejos aplicando la lógica del paradigma informático” (MIUR, 2014)

Al final del documento en la página 131 en síntesis al punto 10 se señala: “Las nuevas alfabetizaciones: fortalecimiento del plan de formación para los idiomas extranjeros a partir de los seis años. Habilidades digitales: *coding* y pensamiento computacional en la primaria y plan *Digital Movers* en la secundaria.”

Del documento se desprende:

“...este es el siglo de la alfabetización digital: la escuela tiene el

deber de motivar a los jóvenes para entender el mundo digital más allá de la superficie. A no conformarse con utilizar un sitio web, una App, un videojuego, sino a diseñar alguno.”

La Ley n.107 del 13/07/2015 “Reforma del sistema nacional de educación y formación y delega para la reorganización de las disposiciones legislativas vigentes (G.U. 15/07/2015 Serie Generale – n.162) hace referencia explícita al pensamiento computacional a pág.2 art.1 coma 7 punto h *desarrollo de las habilidades digitales de los estudiantes, con particular atención al pensamiento computacional, al uso crítico y consiente de las redes sociales y de los medios así como a la producción y a lo relacionado con el mundo del trabajo.*

La trayectoria de innovación y digitalización prevista por la reforma de la Buona Scuola se materializa en el “Piano Nazionale per la Scuola Digitale (Plan Nacional para la Escuela Digital)” (PSND, 2015) presentado el 27 de octubre del 2015, con la idea de innovar el sistema educativo italiano y centrarlo en la era digital. Uno de los objetivos es introducir en la escuela italiana las nuevas tecnologías y difundir el concepto de aprendizaje permanente (*lifelong learning*).

El Plan, con validez multianual, busca utilizar más fuentes de recursos a favor de la innovación digital, partiendo de los recursos puestos a disposición por los Fondos Estructurales Europeos (PON Instrucción 2014/2020) y de los fondos de la Ley 107/2015. En el párrafo con título *Habilidades y contenidos* (punto 4.2), se observa que el nuevo desafío formativo está relacionado con la “capacidad de encontrar, comprender, describir, utilizar, producir información compleja y estructurada, tanto en el ámbito científico y tecnológico como en el humanístico y social”. Las habilidades requeridas para ello son de tipo lógico, computacional, tecnológicas, operativas, argumentativas, semánticas e interpretativas (PSND, 2015). Son varias las dimensiones de las habilidades digitales ya que, además de servir como herramienta didáctica, deben transmitir el desarrollo de habilidades transversales y actitudes, y al mismo tiempo, favorecer la nueva alfabetización a través del pensamiento computacional.

Con una acción específica, *acción#17*, se desea llevar el *pensamiento lógico computacional a toda Educación Primaria*. Las causas son debidas a la evidente cercanía que se tiene con la tecnología ya desde pequeños en familia y a la convicción de que es necesario preparar a los estudiantes al desarrollo de habilidades que serán el centro de sus vidas y sus carreras.

Se toma como referencia el proyecto realizado por MIUR-CINI *Programa el futuro* para la introducción del pensamiento computacional en la escuela, incluso en la escuela de la Infancia, que deriva de una asociación con varias empresas en el territorio italiano.

En los primeros meses del 2017 el MIUR publica un aviso con el objetivo del desarrollo del pensamiento computacional, de la creatividad digital y de las habilidades de "ciudadanía digital", como apoyo a la oferta de capacitación que se encuentra en el marco de las acciones destinadas a aumentar las habilidades de base según el Programa Operativo Nacional *Per la scuola – Competenze e ambienti per l'apprendimento (Para la escuela – Habilidades y ambientes para el aprendizaje)* para el periodo de programación 2014-2020" (MIUR, 2017, mar, 3).

En este documento, el mundo digital es definido como l'alfabeto de nuestro tiempo y en su centro se encuentra el pensamiento computacional, una nueva sintaxis, entre pensamiento lógico y creativo, que forma el lenguaje que hablamos cada vez con más frecuencia en la actualidad (MIUR, 2017, p.1). El objetivo es el de "promover los elementos fundamentales para la introducción a las bases de la programación, también con el fin de desarrollar las habilidades relacionadas a la informática; uno de estos elementos es justamente el desarrollo del pensamiento computacional para reforzar la capacidad de análisis y de resolución de problemas y el uso de sus herramientas y métodos, tanto a través de las tecnologías digitales como a través de actividades unplugged, para estimular una interacción creativa entre digital y manual, incluso a través de experiencias de fabricación, robótica educativa e internet de las cosas".

Al mismo tiempo, la reforma de la Educación de la escuela en Italia hace que surja, a través de la Ley 107/15, la importancia de la formación inicial y en servicio de los profesores que resulta fundamental para su desarrollo profesional y para la mejora de las instituciones escolares (MIUR, 2015).

Así pues, la formación de profesores italianos forma parte del objetivo político para el aumento de la calidad de la educación. De hecho, la eficacia de la acción educativa pasa también por una formación de alta calidad que pueda proporcionar a los docentes las habilidades necesarias para ser eficaces en la escuela actual (PSND, 2015).

En particular el PSND dirige su atención a la formación de docentes calificados en la innovación didáctica y en la promoción de la cultura digital en el campo de la enseñanza, del aprendizaje y de la formación de las habilidades para el trabajo, cognitivas y sociales de los estudiantes.

El MIUR, siguiendo esta nueva visión, ha abierto para todos los profesores la plataforma digital SOFIA (Sistema Operativo para la Formación y las Iniciativas de Actualización de los docentes) con el fin de gestionar su formación y responder a uno de los objetivos establecidos en la Agenda 2030 sobre el desarrollo sostenible de la Educación de calidad (MIUR, 2017, mayo, 19).

Con la Ley 107/15 se introdujo en el mundo de la escuela el rol del animador digital, docente experto en metodologías y tecnologías didácticas con buenas habilidades de organización y liderazgo (acción#28) (PNSD, 2015) que trabaja junto al Director Escolar (DS) y el Director de Servicios Administrativos (DSGA) en el diseño y la realización de los proyectos de innovación digital que se encuentran en el PSND.

El campo operativo del animador digital tiene que ver principalmente con el ámbito de la educación metodológica y tecnológica de sus colegas, fomentar la participación de la comunidad escolar y diseñar soluciones metodológicas y

tecnológicas sostenibles, para difundir al interno del ambiente escolar. Queda clara la referencia al uso de herramientas y métodos relacionados con la robótica educativa, al *coding*, es decir, a prácticas que favorecen el desarrollo del pensamiento computacional (Chen et al, 2017).

Es evidente la necesidad de replantear y revisar la formación de todos los profesores para poder proporcionarles herramientas y conocimientos actualizados al siglo XXI (Manilla & Dagiene, 2014). Los docentes italianos se encuentran ante una gran demanda de innovación que ve más allá del uso lasTIC como dimensión fundamental del proyecto formativo de la escuela, también en el marco de una educación a la ciudadanía, la adquisición de habilidades personales transmisibles a través de un innovador proceso de enseñanza en el que el proceso de aprendizaje esta hecho de experiencias multidimensionales, es más constructivo y reticular, puede compartirse socialmente, incluso a través del juego, la imaginación, las emociones y es alimentado por sucesos de comunicación informal (Falcinelli et al., 2017).

Las propuestas dirigidas a los docentes se materializan con ofertas formativas que provienen de la web a través de webinars, cursos online, hakathons. En particular, con referencia al pensamiento computacional, muchos profesores se actualizan a través de los MOOC (Massive Online Open Courses), cursos online a gran escala abiertos, o dirigidos a un gran número de usuarios, impartidos por Universidades de prestigio, y por este motivo, acreditables (Taranto et al, 2018).

Un ejemplo es CODEMOOC, un MOOC impartido gratuitamente por la Universidad de Urbino, antes en la plataforma europea European Multilingual Mocc Aggregator, EMMA, y a partir del 2018 en el portal de la enseñanza abierta de la Universidad de Urbino Carlo Bo; es un curso de formación de 48 horas que se puede certificar y usarse libremente. El curso se llama *Coding in your classroom, now!* tiene como objetivo ayudar a los docentes de cualquier nivel escolar, desde la Infancia hasta la Universidad, y de cualquier disciplina a

“introducir el pensamiento computacional en clase a través del *coding*, usando solo actividades fáciles y divertidas para proponerlas directamente a los alumnos”. A dicho curso está relacionada una gran comunidad de aprendizaje, de la que forman parte docentes (más de 6.500 docentes) y alumnos (más de 100.000) en ella todos son facilitadores y pioneros.

En los primeros meses del 2018 el MIUR publica una actualización de las *Indicaciones Nacionales para el Plan de Estudios de la Escuela de la Infancia y del Primer Ciclo de Educación*, titulado *Indicaciones Nacionales y Nuevos Escenarios* (MIUR, 2018). El documento propone una revisión de las Indicaciones Nacionales con el fin de garantizar a todos los estudiantes las habilidades clave para enfrentar los últimos cambios que requiere el desarrollo hacia la sostenibilidad en todas sus dimensiones y que garantice la adquisición de los contenidos de la 2030.

El capítulo 5.4 del documento está dedicado al pensamiento computacional, que de hecho se reconoce como *herramienta cultural para el crecimiento personal y la ciudadanía*:

“En los contextos actuales, en lo que la tecnología de la información es tan generalizada, el dominio del *coding* y del pensamiento computacional puede ayudar a las personas a gobernar las máquinas y a comprender mejor su funcionamiento, sin ser por el contrario dominados y esclavizados de manera acrítica” (MIUR, 2018).

El pensamiento computacional se ha visto como el elemento común de disciplinas aparentemente divididas entre ellas, como idiomas y matemáticas, y es un *aspecto de aprendizaje que las recientes normativas, la ley 107/2015 y el decreto legislativo 62/2017 solicitan desarrollar*.

Se invita a los docentes a diseñar actividades relacionadas al pensamiento computacional no solo en el proceso de enseñanza de la tecnología, sino

también en todos los demás ámbitos del saber (MIUR, 2018), incluyendo al sector matemático.

2.15 Conclusiones

En este capítulo se presentó el marco teórico en base al cual se desarrolla el presente estudio. La exigencia de adecuarse a la tecnología que está cada vez más presente en todos los contextos lleva a reflexionar sobre la formación escolar que reciben los estudiantes, futuros ciudadanos digitales y, en particular, a poner atención a dos de las habilidades fundamentales que se requieren, la digital y la matemática, unidas por el desarrollo de habilidades relacionadas al pensamiento computacional.

En los primeros párrafos del capítulo, se analizaron las distintas definiciones que se han dado en el panorama científico al concepto de pensamiento computacional divididas en tres grupos. Aho y Wing (2012) proporcionan una definición más general que se puede poner en relación con habilidades universales como saber leer, escribir y hacer una cuenta. El segundo tipo de definiciones está directamente relacionado con la capacidad de resolución de problemas, a organizarlos, analizarlos e implementarlos y fue proporcionada por la Computer Science Teacher Association (CSTA, 2011). Finalmente, forman parte del tercer grupo, las definiciones que toman como referencia el marco teórico de Brennan y Resnick (2012) que consideran el uso de Scratch, una herramienta para el desarrollo de las habilidades de pensamiento computacional. Este trabajo utiliza como punto de partida justamente la idea de poder implementar en el ámbito educativo y de la enseñanza la definición que se refiere al tercer grupo.

A partir de estas definiciones, se analizaron distintos modelos de desarrollo del pensamiento computacional en el ámbito educativo y se examinaron sus aplicaciones a partir de la escuela de la Infancia, profundizando también los estudios sobre la definición de este concepto que en los últimos años ha

despertado el interés de investigadores y legisladores. Se encontró una gran variedad y heterogeneidad de estudios, por lo que se concentró la atención en el uso de herramientas que promueven el pensamiento computacional como el *coding* en los planes de estudios de disciplinas del área STEM, como las matemáticas y la geometría. Por otro lado, se encontraron pocos estudios relacionados con el desarrollo de habilidades del pensamiento computacional en el campo de las matemáticas en las escuelas primarias, por lo que se considera que sería interesante aplicar metodologías no tradicionales en proyectos cuasi experimentales, en contextos escolares para estudiar los efectos.

En los siguientes capítulos se expodrán las características del presente estudio con referencia a la metodología aplicada y a los datos obtenidos y sometidos a discusión.

Capítulo 3

Metodología y plan de investigación

3. Introducción

Esta tesis doctoralse ha diseñado para estudiar las posibles repercusiones en el ámbito escolar del uso del *coding* como práctica que promueve el pensamiento computacional y como herramienta transversal que favorece el aprendizaje de conceptos propios de la geometría.

Después de haber identificado los aspectos a investigar, a través de la literatura científica, se procedió a delimitar el campo de investigación, definiendo el enfoque metodológico que se utilizó en la elaboración del trabajo.

De hecho, en esta sección de la tesis, se describe la metodología utilizada para responder a las preguntas que surgen a partir de la revisión de la literatura, con particular referencia al tipo de metodología que se consideró pertinente aplicar, a los objetivos que se desean conseguir y a los individuos que han participado. También se describe el contexto en el que se llevó a cabo la experimentación, una escuela de Educación Primaria de la ciudad de Ribera, en provincia de Agrigento así como los instrumentos que se utilizaron para medir las variables a estudiar, y los procedimientos llevados a cabo.

3.1 Enfoque, preguntas y planificación de la investigación

La planificación de la investigación es una de las fases más importantes en una investigación empírica ya que tiene la función de guiar las acciones del investigador que debe tener con referencia a los métodos operativos, los individuos involucrados, así como las condiciones en las cuales se recogen los datos que se analizarán. La elección del método a aplicar más oportuno depende

de la naturaleza de las preguntas que el investigador se ha hecho (McMillan et al., 2005).

La decisión de abordar como argumento de esta tesis doctoral el estudio de la aplicación del *coding* en las disciplinas STEM se deriva también de la consideración de que, en estos últimos años, muchos docentes, de distintos cursos escolares, lo han utilizado como una herramienta de mediación didáctica mientras realizan su trabajo. Se ha pensado que medir el uso del *coding* y su influencia en las actividades escolares a través de técnicas cuantitativas podría ser una manera válida para proporcionar confirmaciones interesantes sobre su difusión. Pero para poder tomar en cuenta los diversos aspectos que intervienen dentro de una clase, se ha considerado interesante, también, combinar las metodologías cuantitativas con las cualitativas para así recopilar más datos (Cohen et al., 2013).

Las investigaciones en este sector, claramente en aumento, se están convirtiendo en objeto de interés para la comunidad científica, de hecho, en estos años, varios trabajos han desarrollado enfoques cualitativos y cuantitativos para estudiar y evaluar los resultados del aprendizaje en los alumnos cuando se implementan prácticas de desarrollo del pensamiento computacional (Grover & Pea, 2013; Kafai & Burke, 2013, Israel et al, 2015).

En cuanto a los estudios experimentales, en general, se recurre a la ejecución de un *experimento* cuando se piensa que se puede establecer una posible relación de causa-efecto entre las variables dependientes e independientes. En este sentido, un estudio cuantitativo se presta para responder a estas exigencias ya que permite revisar bien los factores que intervienen en la investigación (Creswell, 2002). Aunque en el enfoque más riguroso se deben asignar aleatoriamente individuos a los grupos en un experimento, en este caso no fue posible hacerlo por motivos de logística propios de la investigación educativa en un contexto real.

No se crearon grupos artificiales puesto que, tratándose de grupos existentes en un contexto educativo, la formación de nuevos grupos habría sido un elemento interferenciador para el aprendizaje en una clase que, de cualquier forma, habría tenido que realizar la actividad planeada en horario escolar; precisamente tratándose de grupos "naturales" no fue posible llevar a cabo un control experimental completo.

Así pues, el presente trabajo, puede catalogarse dentro de los estudios *cuasi experimentales con grupo de control no equivalente*, procedimientos aproximados a diseños experimentales (Campbell & Stanley, 1963; Kerlinger, 1970;). En dichos diseños, quien conduce la investigación no tiene un control completo de todas las variables y no tiene la posibilidad de asignar de forma casual a los sujetos a las distintas condiciones experimentales, sino únicamente puede seleccionarlos en base a reagrupaciones ya existentes: en estas condiciones, la investigación se lleva a cabo después de que los grupos se hayan formado en base a criterios que el investigador no decide de manera total. Se recurre a este tipo de diseño cuando no es posible tener a disposición dos grupos de sujetos equivalentes, ya que es difícil encontrar un grupo de control realmente equivalente al grupo experimental y como no hay asignación aleatoria de los sujetos, para verificar la equivalencia de los grupos, es necesario formar el grupo de control haciendo que sea lo más similar posible al grupo experimental (Creswell, 2002, p.312).

Es por ello que el presente trabajo no posee las características de un *verdadero* experimento (Cohen et al., 2013): de hecho, se llevó a cabo en una situación de vida real en donde es posible manipular las variables independientes, pero no se puede realizar una asignación aleatoria de los sujetos que forman parte del grupo experimental y de control.

La comparación se realiza entre dos grupos no equivalentes, que no se han compuesto a través de un procedimiento rigurosamente casual, sino que estaban ya constituidos. Tras la fase inicial en la que ambos grupos se someten a una

prueba inicial, se ha llevado a cabo el tratamiento en el grupo experimental. Si la diferencia entre los grupos no resulta significativa estos se consideran equivalentes con respecto a la variable dependiente. En la segunda y última fase, se realiza una prueba final a ambos grupos y se analizan las diferencias midiendo el efecto del tratamiento a partir de la comparación de las pruebas finales.

La Tabla 13 muestra, en resumen, los pasos con relación a la aplicación del método *cuasi experimental* con grupo de control no equivalente descrito anteriormente; el procedimiento implica, por lo tanto, la aplicación de una prueba inicial (G1) a ambos grupos, la implementación de un tratamiento que consiste en el desarrollo de una actividad, específicamente proyectada en base a los objetivos de la investigación, con el grupo experimental, y la finalmente la aplicación de una prueba final (G2) nuevamente a ambos grupos.

Tabla 13

Método cuasi experimental con grupo de control no equivalente

Grupo	Prueba inicial	Tratamiento	Prueba final
Experimental	G1	Geometría con <i>coding</i>	G2
De control	G1	Geometría con método tradicional	G2

Fuente: elaboración propia.

Una vez establecidas las preguntas de investigación y los objetivos que se pretenden alcanzar, se consideró interesante, como ya se ha mencionado, aplicar un enfoque integral (método mixto) en el que se utilizan los datos analizados a través de un método cuantitativo y los resultados del análisis cualitativo para lograr una mejor comprensión del problema y proporcionar las respuestas a las preguntas formuladas (Creswell, 2002, p.534). El objetivo del análisis cualitativo es comprender y mejorar la calidad del trabajo y dar una explicación más profunda de los resultados que surgen de la fase cuantitativa (Cetin, 2016).

Esta decisión representa un compromiso válido cuando se lleva a cabo una investigación educativa como esta en la que es casi imposible efectuar una selección aleatoria de escuelas o de clases. El contexto en cuestión proporciona las condiciones para recurrir a optar por este procedimiento, ya que se refiere a una situación bastante típica de la investigación escolar en la que los grupos de individuos son generalmente clases ya constituidas en las que se llevan a cabo con frecuencia proyectos similares.

La investigación cualitativa representó el punto de partida de una parte del trabajo, a partir de ella se establecieron los objetivos y las preguntas de la investigación. También, se consideró interesante sustentar los datos cualitativos, precisamente obtenidos como resultado de la observación en el aula y a partir de las entrevistas, con otro tipo de información de tipo numérica que puede procesarse mediante procedimientos estadísticos (análisis cuantitativo).

Los estudios científicos realizados en los últimos años muestran una tendencia acentuada a combinar los métodos cualitativos con los cuantitativos para poder aprovechar un conjunto de técnicas más amplias y complejas que pueden enriquecer bastante los datos obtenidos. Así pues, es posible catalogar este trabajo dentro de los métodos mixtos, procedimientos de recogida, análisis y mezcla de métodos cuantitativos y cualitativos en el mismo estudio con el fin de comprender mejor un problema de investigación (Creswell, 2002, p.535). En el presente estudio se llevó a cabo una especie de diseño integral (*embedded design*) en el que los datos cualitativos y cuantitativos se recopilaron en parte en secuencias y en parte de manera simultánea (Creswell, 2002, p.544).

En síntesis, después de haber definido el plan de investigación se procedió a elegir la metodología que se adaptara mejor al problema y a definir las variables educativas que se desean medir y que constituyen los datos tanto cualitativos como cuantitativos.

El objetivo principal de la presente tesis de doctorado ha sido verificar si el uso del *coding*, entendido como herramienta de mediación didáctica transversal al aprendizaje disciplinario del área STEM, como la geometría, contribuye a mantener alto el aspecto motivacional intrínseco, el interés y el sentido de autoeficacia, generando un mayor esfuerzo por parte de los niños hacia una materia considerada difícil y accesible solo para quienes se consideran especialmente buenos en el tema.

También se pretendió verificar si un aumento del grado de satisfacción permite un mayor control de las emociones negativas, como el ansiaedad, y logra mejorar o hace surgir las emociones positivas además de favorecer las dinámicas sociales y de grupo.

En paralelo, se cuestiona si el enfoque a través del uso de instrumentos de soporte a la programación, como Scratch, puede facilitar el aprendizaje de la geometría a un número mayor de niños mejorando sus resultados de aprendizaje en términos de habilidades y competencias.

Por último se desea verificar si existe una relación entre las distintas variables que entran en juego y si, en definitiva, resulta factible para los docentes introducir prácticas de *coding* que desarrollen el pensamiento computacional dentro de los programas de enseñanza de los planes de estudios de matemáticas/geometría.

De acuerdo con esto, se plantearon las siguientes preguntas de investigación:

D1. ¿Puede la codificación facilitar el aprendizaje de los alumnos en geometría mejorando los resultados finales?

D2. ¿Logran los alumnos, a través de las actividades de *coding*, experiencias positivas y gratificantes?

D2.1. ¿Pueden los alumnos controlar las emociones negativas para dar paso a las otras positivas?

D3. ¿Ayuda el *coding* a aumentar el aspecto motivacional, el interés y el nivel de participación en el aprendizaje de la geometría?

D4. ¿La actividad de *coding* mejora la relación social y la dinámica de grupo?

D5. ¿Qué relevancia tiene el rol del docente en el proceso de aprendizaje de los alumnos?

3.2 Participantes y contexto

Uno de los momentos más importantes del proceso de investigación es cuando se selecciona el grupo de individuos que participaran en el estudio. La elección del número y de las características de los participantes para la investigación se basa en criterios científicos con relación a los objetivos planteados, a los métodos para la recogida de datos y al tipo de metodología que se ha escogido. Por lo tanto, el tipo de muestreo que se implementa en una investigación está en función de un análisis crítico llevado a cabo durante la planeación de esta (Cohen et al., 2013, p.100).

En cuanto a la elección del número de participante para la muestra, está claro que mientras más grande sea la muestra mayor será la fiabilidad del estudio y también la posibilidad de generalizar sus resultados. Sin embargo, en algunos casos, el investigador se ve en la necesidad de trabajar con datos de pequeños grupos o subconjuntos de la población que sustituyen a la muestra.

La necesidad de llevar a cabo el muestreo nace de la dificultad de examinar directamente toda la población y, al mismo tiempo, de la exigencia de efectuar una generalización de un estudio. La elección de la población de referencia obviamente define los límites que el investigador desea poner a su estudio.

3.2.1 Selección de la muestra

Con referencia al presente estudio, se decidió llevar a cabo un muestreo no probabilístico con decisión razonada (McMillan, 2005) en el que la elección de las unidades estadísticas incluidas en la investigación no es casual, sino que se efectuó considerando los objetivos de la investigación y las características particulares de la población de referencia (McMillan, 2005). Teniendo en cuenta que, al inicio del trabajo, la información disponible sobre el número de niños que se educan en pensamiento computacional con la práctica del *coding* en las clases italianas es insuficiente, se decidió efectuar un muestreo razonado para extraer de manera intencional y no casual a los individuos que, según el investigador, responderían mejor a los objetivos cognitivos establecidos en el proyecto de investigación.

A partir de la población constituida por todos los estudiantes de Educación Primaria italiana, se decidió trabajar con una muestra reducida ($n=47$) formada por niños de dos clases de cuarto curso de edades entre los 8 y 9 años de manera no casual; se trata de un pequeño grupo que, por sus características, no resulta ser representativo y no permite por lo tanto efectuar una generalización pero puede llevar a analizar aspectos a pequeña escala que de otra manera no habrían sido detectados (Cohen et al., 2013).

La elección de la zona geográfica en la cual buscar la muestra no fue del todo aleatoria sino guiada por la evaluación razonada con respecto a los resultados de las pruebas INVALSI que se realizan en diversas áreas distribuidas en el territorio italiano. La puntuación promedio de los estudiantes en dichas pruebas muestra un rendimiento ligeramente mejor en los estudiantes del Norte de Italia en comparación con los estudiantes del Sur y de las Islas. Por lo tanto, se decidió dirigir la atención a las áreas del Sur de Italia y de manera específica al territorio siciliano.

La investigación involucró a los estudiantes de Educación Primaria del Instituto Comprensivo "Don Bosco" de Ribera, ciudad siciliana que se encuentra en provincia de Agrigento. El Instituto en cuestión está compuesto por un alumnado heterogéneo, en cuanto a los niveles económicos, sociales y culturales. Las familias se muestran generalmente disponibles al diálogo y a la colaboración, atentas a los elementos que caracterizan la oferta formativa en todas sus implicaciones culturales.

La muestra fue seleccionada tras haber detectado las clases que tenían conocimiento en cuanto a prácticas de *coding* en el ambiente escolar con respecto a quienes lo desconocen, en el Instituto Comprensivo Don Bosco. Así pues se involucraron al proyecto dos clases de cuarto curso con un total de 47 participantes. Los estudiantes tenían entre 9 y 10 años.

La clase que formó el grupo experimental la constituyeron 25 estudiantes de los cuales 12 eran del sexo masculino (48%) y 13 del sexo femenino (52%) con una edad media de 9,12 años ($DS = ,33$); la clase identificada en el grupo de control estuvo compuesta por 22 estudiantes de los cuales 13 fueron del sexo masculino (59,1%) y 9 del sexo femenino (40,9%) con una edad media de 9,09 años ($DS = ,29$); en este grupo finalmente participaron 21 a causa de la falta de autorización para participar por parte de los padres de una niña.

Solo la clase del grupo experimental poseía una Pizarra Digital Interactiva (PDI) o *Lavagna Interattiva Multimediale* (LIM) y ambos grupos podían utilizar un laboratorio de Informática con ordenador y conexión a internet. El consejo de clase o el docente es distinto en cada grupo, sin embargo, la programación de las materias, los objetivos disciplinarios a alcanzar y las pruebas adoptadas son las mismas. Ambos docentes de los dos grupos se unieron de manera espontánea a la experimentación.

En cuanto a la dimensión de la muestra de referencia, algunos autores sugieren que, cuando se llevan a cabo estudios relacionados con una aplicación

de un tratamiento a un grupo experimental que se compara con un grupo control (que no recibe tratamiento), es recomendable tener treinta participantes por grupo sin embargo, especialmente en el ámbito escolar, resulta difícil que pueda ser así (Cohen et al., 2013; Gay et al., 2009;). Para esta investigación *cuasi experimental*, se consideró suficiente poder utilizar muestras de tamaño un poco más pequeño con respecto al mínimo recomendado por los estudios experimentales, con enfoque, no tanto a los resultados del estudio único sino a los resultados similares de varios estudios (Corsi et al., 2019).

En base a los estudios análogos, encontrados en la literatura, en los que se llevaron a cabo investigaciones del tipo *cuasi experimental* en muestras reducidas constituidas por grupos de clase, en el ambiente escolar (K-12), se tomó la decisión de llevar a cabo la experimentación sobre una muestra constituida en total por 46 participantes. De hecho, son varias las investigaciones del tipo *cuasi experimental* realizadas, que utilizando una muestra poco numerosa, han estudiado varios aspectos relacionados con el uso del *coding* en la enseñanza y los beneficios que de manera transversal adquiere el estudiante y el grupo de la clase. A continuación, se mencionan, a continuación, algunos ejemplos.

En su trabajo, Wilson y Moffat (2010) describen las experiencias didácticas hechas por un grupo de niños escoceses de Educación Primaria (n=21) con Scratchcomparándolas con las lecciones tradicionales; utilizaron como instrumentos para la recogida de datos los cuestionarios, las observaciones y una entrevista a la docente que participó en las actividades para describir los efectos del *coding* sobre los progresos cognitivos y emotivos.

Bers, Flannery, Kazakoff y Sullivan (2014) llevaron a cabo un estudio con niños de la escuela de la Infancia (n=53) con respecto al aprendizaje de conceptos y habilidades propias del pensamiento computacional, de la robótica, de la programación y de la resolución de problemas. Grover, Pea y Cooper (2015) realizaron un estudio con una muestra de 54 estudiantes (n= 54) con el

fin de profundizar el conocimiento de algunos aspectos relacionados con el pensamiento computacional a través del uso de Scratch; para esto efectuaron un *cuasi experimento*, a través de la aplicación de pruebas iniciales y finales destinadas a medir las posibles mejorías en cuanto a las habilidades de programación.

De igual manera, Cetin (2016), en el ámbito de la enseñanza de la informática, estudia los efectos sobre el pensamiento computacional, en el desarrollo de actividades con Scratch para el grupo experimental (n=28) y con el Lenguaje C para el grupo de control (n=28), sometiéndolo a algunos grupos de estudiantes universitarios a una entrevista semiestructurada, para obtener datos cuantitativos y cualitativos al mismo tiempo.

Kalelioglu (2015) condujo un estudio cuali-cuantitativo con 32 estudiantes (n=32) de Educación Primaria con respecto al análisis del desempeño de estos que utilizan el sitio *code.org*. La parte cuantitativa se realizó efectuando una comparación entre los resultados de la prueba inicial y la prueba final con una visión cuasi experimental. En la parte cualitativa de la investigación se realizaron entrevistas y grupos focales con diez estudiantes (n=10).

En su proyecto de investigación, Seo y Kim (2016) examinaron los efectos del *coding* en la enseñanza de la geometría sobre el pensamiento computacional y la creatividad, en Educación Primaria, aplicando una prueba inicial y una prueba final a los dos grupos, uno experimental y el otro de control constituidos por un total de veintidós cuarenta y cuatro (n=44).

Corsi et al. (2019) estudiaron la posibilidad de alentar el desarrollo de la habilidad emotiva y la asimilación de los contenidos a través del desarrollo de los juegos en clase utilizando la programación inspirada al constructivismo de Papert, trabajando con dos grupos experimentales compuestos de 25 participantes cada uno (n=25) y un grupo de control de veintisiete estudiantes (n=27).

Como ya se ha mencionado, los resultados que derivan de la aplicación de la metodología a la muestra, por sus características, poseen claramente límites pues no permiten una generalización de los resultados obtenidos justo porque la elección de la muestra se basó en una selección subjetivamente establecida; por otro lado, pueden transferirse a determinados contextos comparables al contexto en el que se realizó la investigación, teniendo en consideración siempre las condiciones de recogida de datos y las características del muestreo utilizado. Además, el número reducido de los sujetos que participaron en la investigación hizo posible la planeación de una observación no participante de los sujetos involucrados para poder llevar a cabo un análisis cualitativo de los datos, además de los cuantitativos, como soporte del estudio.

3.3 Instrumentos y variables

En el párrafo siguiente se definen y se describen las características de las principales variables utilizadas en el estudio y los instrumentos que se utilizaron para recogerlos datos (Tabla 10).

Para determinar las variables se hizo referencia a las preguntas y a las hipótesis de la investigación. Una variable es una característica que se refiere a una variación entre los miembros de un grupo en particular y es necesario definirla para poderla medir de manera significativa (Fraenkel et al., 2011). Teniendo en cuenta que, en una investigación, es fundamental conocer las variables que intervienen, en el presente estudio se especificaron tanto las variables dependientes como las independientes partiendo de su definición. Una variable independiente es una variable de entrada que determina el resultado, la respuesta a un estímulo; una variable dependiente está relacionada al resultado que se obtiene después de haber introducido la variable independiente (Cohen et al., 2013) y es aquello que se mide a través de las pruebas, cuestionarios u observaciones (Vanderstoep & Johnston, 2009).

Una vez definidas las variables del presente estudio, en base a sus características, se procedió a seleccionar los instrumentos considerados más significativos para la recogida de los respectivos datos y para su análisis. En cuanto a los instrumentos utilizados para medir, observar y documentar los datos cuantitativos y cualitativos una parte se construyó a propósito mientras que la otra parte se identificó gracias a la contribución de estudios científicos previos.

Para analizar la variable relacionada con el rendimiento de los alumnos en geometría se utilizaron dos pruebas específicas de evaluación de los aprendizajes para proponer a los alumnos. A través de estos dos instrumentos, utilizados en el ámbito de la investigación como prueba inicial y prueba final, se obtuvieron los datos que permitieron cuantificar los resultados de aprendizaje alcanzados en geometría por parte de cada alumno de cada grupo.

Con el fin de recopilar los datos respectivos a las actitudes individuales, relacionadas por ejemplo a la esfera emocional y motivacional, se consideró más eficaz utilizar un cuestionario que se aplicó a los estudiantes de ambos grupos basado en una escala de satisfacción. En la presente investigación, se midieron las emociones de logro de los niños a través de la versión adaptada en idioma italiano (Raccanello et al., 2019) del cuestionario con título *The Achievement Emotions Questionnaire-Elementary School* identificado con la sigla *AEQ-ES* (Lichtenfeld et al., 2012).

Para recopilar los datos sobre los comportamientos, las actitudes, las emociones, el esfuerzo y el interés que mostraron los niños durante el proceso de aprendizaje y enseñanza se procedió a observarlos a través de un protocolo de observación preparado antes del inicio de las observaciones en el que fue posible registrar y recopilar los datos para su posterior análisis, cualitativo y cuantitativo. La actividad de observación fue de tipo no participativa y la llevó a cabo directamente el investigador en ambas clases, con el fin de examinar algunos aspectos relacionados con el proceso de aprendizaje, entre los cuales la

motivación, el interés y las emociones. De esta manera, no estuvimos directamente involucrados y nos limitamos a registrar las notas en el formulario de observación debidamente preparado.

Todo el trabajo se llevó a cabo durante el año escolar 2018/2019, en el periodo de marzo a mayo.

Finalmente para completar la recopilación, análisis y elaboración de los datos cualitativos útiles para responder a las preguntas de la investigación se preparó una entrevista semi estructurada para la docente que condujo el tratamiento en la clase que conformó el grupo experimental, utilizando el *coding* como metodología didáctica en el estudio de la geometría. La entrevista se realizó preparando una serie de preguntas para las cuales se el contenido se estableció a priori mas no la forma. También en este caso, se informó a la entrevistada en forma oral y escrita sobre los propósitos de la entrevista misma.

Tabla 14

Sistema de la investigación: aplicación del método cuasi experimental

Preguntas de la investigación	Métodos para la recogida de datos
D1 Desempeño en geometría	Prueba inicial y final
D2 Emociones de logro	Cuestionario AEQ-ES
D3 Esfuerzo cognitivo y motivacional	Observación no participativa/Entrevista
D4 Relaciones sociales	Observación no participativa/Entrevista
D5 Función del docente	Observación no participativa/Entrevista

Fuente: elaboración propia.

3.3.1 Instrumentos para la recopilación de datos cuantitativos: Prueba de evaluación de los aprendizajes

La metodología utilizada requirió el uso de dos pruebas de geometría elaboradas durante la planeación del presente trabajo como instrumentos de

evaluación de los aprendizajes relacionados; ambos están compuestos por preguntas con respuesta de opción múltiple que tratan sobre el conocimiento de elementos atribuibles a habilidades relacionadas a la disciplina y se aplicaron tanto en la clase del grupo experimental como en la del grupo de control.

3.3.1.1 **Las pruebas INVALSI**

Para construir las dos herramientas se hizo referencia a las pruebas nacionales propuestas anualmente por el Instituto Nacional de Evaluación del Sistema de Educación (Istituto Nazionale di Valutazione del Sistema di Istruzione INVALSI) para Educación Primaria que tiene el objetivo de verificar los niveles de aprendizaje alcanzados por los estudiantes en matemáticas y medir la capacidad de expresar y utilizar el pensamiento matemático (Capperucci, 2017). Se trata de pruebas a gran escala con características objetivas ya que la corrección se realiza a través de un protocolo establecido a priori sin dar la posibilidad de cometer errores a quien la realiza.

Las pruebas estandarizadas de matemáticas tienen el objetivo de evaluar el nivel de aprendizaje y de preparación de los estudiantes a escala nacional al final de la segunda y quinta clase de primaria y están dirigidas a estudiantes que asisten a la misma clase en todo el territorio italiano. Su objetivo es detectar el aprendizaje de los estudiantes en los distintos niveles de escuela para poder evaluar el valor agregado que proporciona cada escuela en la enseñanza de los alumnos, es decir, la eficacia de las acciones de la escuela misma para proporcionar a los docentes un estímulo para una planeación didáctica más realista.

Las pruebas INVALSI son en general para las disciplinas de italiano, matemáticas e idioma inglés (Corsini, 2013) y para producirlas se activa una trayectoria que dura entre dos y tres años y comienza con la formulación de las preguntas. Estas deben construirse para ser comprensibles, con formas lingüísticas simples y compatibles desde el punto de vista técnico-terminológico

con el tipo de clase a la que son destinadas, con el fin de reducir la ambigüedad a través de estímulos claros y circunscritos. Las opciones de respuesta se formulan a través de distractores plausibles para estimular un proceso significativo de discriminación en la selección de la respuesta correcta.

El proceso utilizado para la construcción de las preguntas, desde el punto de vista metodológico, presenta muchas similitudes con el que se utiliza en las investigaciones *Trends in Mathematics and Science Study* (TIMMS-IEA) y *Programme for International Student Assessment* (PISA - OECD) (Capperucci, 2017). Para construir las pruebas TIMMS se toman en cuenta dos dimensiones:

- los dominios de contenido que son temas evaluados en matemáticas como números, geometría, álgebra, datos y probabilidad;
- los dominios cognitivos que son los procesos de pensamiento que los alumnos deben poner en marcha al trabajar con las matemáticas, como el conocimiento, la aplicación y el razonamiento.

Para las pruebas PISA se hace referencia a cuatro aspectos diversos:

- el contenido matemático;
- el contexto en el que se realiza la prueba;
- los procesos matemáticos (*formulating – employing – interpreting*);
- las habilidades del siglo XXI a partir de las cuales toma forma la alfabetización matemática.

En este trabajo, se hace referencia a las pruebas de matemáticas para el primer ciclo de educación cuya construcción ha tenido en cuenta los planes de estudios nacionales del sistema escolar (Ministero della pubblica Istruzione, 2007, Cerini, 2012)

Como ya se ha mencionado, el Cuadro de Referencia de INVALSI (QdR) se ha realizado haciendo referencia explícita a los trabajos de TIMMS y PISA. De las pruebas TIMMS comparte en la mayor parte los aspectos relacionados con el dominio del contenido y de las pruebas PISA los procesos matemáticos y de resolución de problemas de la vida real. Define en forma detallada el objeto de la prueba y la forma de medición para así representar la base para quien debe construir las pruebas y un elemento de aclaración para las partes interesadas con respecto a los contenidos y aspectos que se pretenden medir con las pruebas propuestas (INVALSI, 2018).

Por lo tanto, las pruebas INVALSI de matemáticas, refiriéndose a los procesos anteriormente especificados, tienen en cuenta a detalle dos dimensiones:

- los contenidos, subdivididos a su vez en cuatro áreas (números, espacio y figuras, datos y previsiones, relaciones y funciones);
- los procesos, que se refieren a los instrumentos cognitivos utilizados para la resolución del estímulo matemático.

3.3.1.2 Prueba inicial y Prueba Final: características

Para detectar los niveles de aprendizaje alcanzados por los niños en geometría se utilizaron dos pruebas construidas *ad hoc* para esta investigación, una para aplicar antes del inicio de la actividad experimental tanto al grupo experimental como al de control (prueba inicial) y otra utilizada como prueba final, después de haber llevado a cabo la actividad.

Reconociendo en las pruebas INVALSI el valor de los instrumentos de verificación formativa así como de auto verificación y reflexión para el estudiante, las pruebas de aprendizaje utilizadas en la investigación se prepararon extrapolando parte de las preguntas de las pruebas INVALSI relacionadas con los conceptos de geometría y en particular con el reconocimiento de las formas del plano y de

espacio, a la clasificación de las figuras en base a características geométricas así como al conocimiento de los conceptos de medida, proyecto y construcción de modelos.

El objetivo de las pruebas fue analizar dos dimensiones:

- Los contenidos matemáticos que comprenden el ámbito de geometría en lo que se refiere a espacio y figuras y, más a detalle, mapas, plantillas y orientación, las principales figuras del plano y del espacio, los objetos y las figuras en el plano y en el espacio, las reproducciones a escala;
- Los procesos cognitivos útiles para resolver los problemas propuestos.

Las preguntas que se encuentran en las pruebas se seleccionaron con atención para detectar aquellas que pudieran ser adecuadas para los conocimientos adquiridos y representativas del ámbito de contenido que se pretende medir. En particular, se buscaron todas aquellas preguntas relacionadas a la adquisición de conocimientos y habilidades relacionadas al tipo de actividad planificada que se refiere al estudio de las figuras planas, las definiciones de las relaciones entre los diversos elementos, las propiedades y sus representaciones. Los elementos identificados son atribuibles a los tipos de preguntas que poseen respuestas de opción múltiple y respuestas abiertas únicas.

Se seleccionaron, tras haber validado su contenido, 15 elementos para la prueba inicial y 18 para la prueba final que tratan los temas de geometría afines a la unidad didáctica de aprendizaje a realizar. Una vez realizadas las pruebas, los docentes de cada clase procedieron a su corrección presentando una votación en función del número de respuestas correctas y utilizando una cuadrícula de corrección.

3.3.1.3 Validez del contenido

Verificar la validez de un instrumento es un proceso importante para una investigación eficaz (Cohen et al., 2013). Para este fin, tras haber seleccionado y preparado las preguntas de las pruebas de geometría, se volvió necesaria la demostración de la validez de los dos instrumentos a aplicar a los alumnos. En particular, nos se preguntado si los elementos cubrían todos los dominios a investigar y si eran representativos. Se consideró útil medir la validez de contenido de las pruebas preparadas incluso cuando estas ya pertenecían a pruebas estandarizadas anteriormente analizadas en cuanto a su validez; para esto participaron 20 expertos, docentes del área científico-matemática en Educación Primaria, con la tarea de evaluar la calidad de las preguntas y que tan adecuadas eran estas tanto en para la prueba inicial como para la final.

La prueba inicial, en la versión original, contiene 16 preguntas divididas en tres áreas:

1. reconocer las formas en el espacio y utilizarlas para la resolución de problemas (9 elementos);
2. conocer y dominar algoritmos y procedimientos en el ámbito geométrico (4 elementos);
3. reconocer en distintos contextos el carácter medible de los objetos y fenómenos; utilizar instrumentos de medida; medir tamaños (3 elementos).

La prueba final, en la versión original presenta 14 preguntas subdivididas también estas en tres áreas:

1. reconocer y representar formas del plano y del espacio, relaciones y estructuras que se encuentran en la naturaleza o que han sido creadas por el hombre (7 elementos);
2. describir, denominar y clasificar figuras en base a características geométricas, determinar las medidas, diseñar y construir modelos concretos de varios tipos (4 elementos);

3. conocer y dominar los contenidos específicos de la geometría con objetos matemáticos, propiedades y estructuras (3 elementos).

El análisis realizado por los docentes expertos proporcionó datos útiles para poder calcular el *Content Validity Index (CVI)* de cada una de las preguntas de las pruebas, este indicador proporciona un juicio con respecto a los contenidos de las preguntas que deben ser coherentes con aquellos que se desea medir y representativas de los objetivos de la investigación (Singh, 2007).

El cálculo del *CVI* se lleva a cabo a través de un procedimiento cuantitativo que permite determinar, según una escala graduada, como un instrumento (prueba, cuestionario, etc.) mide un concepto o una variable, así como lo sugiere Lawshe (1975).

Los resultados de la elaboración de dicho índice se presentan en el siguiente capítulo (Capítulo 4: Resultados).

3.3.1.4 Versión final de las pruebas de geometría

La prueba inicial, indicada con la sigla G1, en su versión final está compuesta por 15 elementos subdivididos por dimensión prevalente. A continuación, en la Tabla 15, los elementos se agrupan en función a la dimensión prevalente que representan.

Tabla 15

Estructura de la prueba inicial (pretest)

Dimensión	Elementos	Total. Elementos
1. Reconocer las formas en el espacio	1, 2a, 2b, 5b, 9, 10a, 10b, 12	8
2. Conocer y dominar algoritmos y procedimientos	4, 5a, 8a, 8b	4
3. Reconocer en distintos contextos el carácter medible de los objetos	3, 6, 7	3

Fuente: elaboración propia.

La prueba final indicada con la sigla G2 está compuesta, en su versión final, por 18 elementos también subdivididos por lastres dimensiones prevalentes como se observa en la Tabla 16.

Tabla 16

Estructura de la prueba final (postest)

Dimensión	Elementos	Total. Elementos
1. Reconocer y representar formas del plano y del espacio	1, 2, 3, 4, 12a1, 12a2, 12a3, 12b	8
2. Describir, denominar y clasificar figuras	6, 7, 10, 11	4
3. Conocer y dominar los contenidos específicos de las matemáticas	5, 8, 9a, 9b, 9c, 9d	6

Fuente: elaboración propia.

La aplicación de ambas pruebas se realizó a través de una explicación previa por parte del docente que proporcionó instrucciones claras y precisas necesarias para que los niños pudieran llevarlas a cabo: este trabajo puede delegarse a los administradores que pueden ser los mismos docentes de la clase. El tiempo estimado para completarlo es de 30 minutos aproximadamente; la puntuación general se puede calcular considerando el número de respuestas correctas con respecto al total.

La prueba G1 se aplicó a los estudiantes en el mes de marzo del 2019 mientras que en mayo del 2019 (después de aproximadamente 12 horas de realizar la actividad relacionada) ambos grupos de estudiantes realizaron la prueba de aprendizaje G2 y el cuestionario.

3.4.1 Cuestionario

El cuestionario representa una herramienta interesante para recopilar información cuando se realizan encuestas con muestras de población; en poco tiempo, a través de este, es posible tener una gran cantidad de información de los participantes, aunque frecuentemente, no se permite, como en este caso, expresar respuestas libremente. El uso de preguntas y procedimientos uniformes

ofrece la posibilidad de simplificar el proceso de análisis e interpretación de datos (Cohen et al., 2013).

En la presente investigación, como herramienta de recogida de datos se eligió un cuestionario autocompletado, para aplicar a ambos grupos después del desarrollo de la actividad didáctica planificada con el fin de indagar respecto a las emociones que surgieron después de las actividades relacionadas con el estudio de la geometría y posteriormente verificar, si existe relación entre dichas emociones y los otros factores que surgieron a través del uso de otras herramientas.

Se trata pues de un cuestionario estructurado que invitaba al estudiante a reflexionar sobre la experiencia vivida en el proceso de aprendizaje y sobre su impacto en la esfera emocional y motivacional.

Se aplicó a ambos grupos en presencia del investigador, que lo distribuyó, estando presente mientras fue completado; posteriormente lo retiró con el fin de reducir los riesgos relacionados (Corbetta, 2003). La presencia del investigador es útil, particularmente en las clases de una escuela, ya que puede responder inmediatamente a cualquier pregunta o incerteza que pueda surgir cuando se está completando (Cohen et al., 2013).

Con dicha herramienta se deseó evaluar algunos aspectos que surgen de la actividad de estudio realizada en clase durante el proceso de aprendizaje de una rama de las matemáticas, la geometría, para muchos considerada "difícil" y emocionalmente poco interesante. El objetivo del cuestionario fue conocer las percepciones de los estudiantes durante la fase de aprendizaje y como logran manejar sus emociones.

En el contexto educativo italiano la enseñanza de las matemáticas en Educación Primaria se distribuye en cinco años e involucra una serie de actividades relacionadas sustancialmente con dos líneas: aritmética y geometría

que siguen las Indicaciones Nacionales (MIUR, 2018). Las clases se imparten en el aula y se asignan generalmente, excepto en aquellas de tiempo completo, tareas para realizar en casa por parte del profesor lo cual implica un trabajo individual para el alumno al que se le da una fecha de vencimiento para entregar las tareas. Además, se programan en clase pruebas orales y escritas con opción de respuestas abiertas y cerradas.

Una vez definidos los objetivos que se desean alcanzar se identifican las áreas del contenido que se busca explorar, a través de la aplicación del cuestionario que trata principalmente sobre el control de las emociones, la motivación y el interés, el esfuerzo escolar y las dinámicas sociales.

A partir de las investigaciones de estudios previos y recientes, se realizó una revisión de la literatura y los argumentos de interés por parte de la comunidad científica, y fue posible seleccionar como instrumento de medida el cuestionario con título *The Achievement Emotions Questionnaire-Elementary School (AEQ-ES)* (Pekrun et al., 2011) que tiene el objetivo de evaluar las emociones de logro en los estudiantes de Educación Primaria.

En particular, la intención es medir una variedad de distintas emociones que también están presentes en los niños de Educación Primaria asumiendo como enfoque teórico de control-valor (Pekrun, 2006).

AEQ-ES es un cuestionario de autoevaluación que se realizó sobre la base del cuestionario "*Achievement Emotions questionnaire*" diseñado para los estudiantes más grandes (Lichtenfeld et al., 2012, Pekrun et al., 2011).

Los resultados del estudio principal llevado a cabo en 30 escuelas primaria de Alemania con una muestra de niños de segundo (n=678) y tercer curso (n=687) demuestran que el instrumento se puede aplicar en las culturas occidentales. Además de considerarse un instrumento válido para la evaluación de las emociones en matemáticas, para los autores es posible aplicarlo en otras

materias escolares sustituyendo simplemente el sustantivo en inglés *math* con otros términos para poder así utilizarlo también en otras áreas temáticas (Lichtenfeld, et al., 2012).

Este instrumento posee 28 elementos que toman en consideración los componentes afectivos, cognitivos, motivacionales y/o comportamentales de las emociones relacionadas a las matemáticas en los niños de Educación Primaria, con referencia en específico a tres distintos momentos del aprendizaje: las lecciones llevadas a cabo en el aula, las tareas asignadas para la casa y las evaluaciones en el aula, con referencia a algunas emociones en particular: la alegría y la diversión (9 elementos), la ansiedad (12 elementos) y el aburrimiento durante las clases y al realizar las tareas (7 elementos).

Los elementos se subdividen en tres bloques para evaluar las experiencias emotivas antes, durante y después de la actividad escolar y exactamente cuándo se toma la clase, cuando se estudia y cuando, al final, se lleva a cabo la prueba de evaluación (Lichtenfeld et al., 2012). Tiene como soporte visual caras que corresponden a niveles crecientes de intensidad. Las etiquetas verbales y los dibujos corresponden al género de los niños y están agrupadas en función del dominio (lecciones, tareas o evaluación). Cada sección está precedida por una etiqueta escrita que especifica justamente el dominio y por un dibujo correspondiente que utiliza un personaje de dibujos animados para expresar el concepto.

El cuestionario identifica elementos individuales a los que es posible responder a través de una escala de evaluación (Likert) cuya simplicidad y aplicabilidad es bien conocida; el entrevistado debe responder a las preguntas expresando en sí y en qué medida está de acuerdo con respecto a una determinada pregunta. Las escalas de evaluación, como la del tipo Likert, son ampliamente utilizadas en la investigación pues permiten combinar la oportunidad de una respuesta léxica de la cual es posible conocer frecuencias y correlaciones, ofreciendo así a quien la utiliza la libertad de medir opiniones,

cantidad y calidad (Cohen et al., 2013).

En este caso, las opciones de respuesta son cinco y van desde “en absoluto” a “muchísimo” (1= en absoluto; 2= poco; 3= bastante; 4= mucho; 5= muchísimo): a los estudiantes se les pide plasmar su propia experiencia emotiva con referencia a esto cinco elementos de la escala Likert con el fin de medir las emociones que surgieron durante el aprendizaje y con relación a los resultados de la geometría.

La duración de la aplicación del cuestionario es de 45 minutos y es posible dar la posibilidad a los niños de hacer una breve pausa y comenzar nuevamente.

Antes de distribuir el cuestionario se garantizó a los niños la privacidad de las respuestas y se les explicó el tipo de cuestionario, leyendo las preguntas y explicándoles que no habría respuestas correctas o incorrectas.

El *AEQ-ES* es un instrumento relativamente nuevo que, como aconsejan algunos autores, puede ser utilizado no solo para la investigación científica, sino también en las actividades promovidas por los docentes y educadores. Además de haber sido aplicado para demostrar la invariancia de la medición entre las escuelas americanas (Estados Unidos) y alemanas, se probó también en las escuelas primarias italianas (Raccanello et al., 2019) abriendo camino a la validación del instrumento traducido al italiano. La versión italiana fue validada y aceptada para las disciplinas de italiano y matemáticas; está constituida por 32 elementos (Tabla 17) en vez de los 28 elementos que contiene la original, en el que:

- 9 elementos se refieren a la esfera relacionada con la diversión de tipo *Me divierto en las lecciones de Geometría*.
- 12 elementos se refieren a la ansiedad de tipo *La geometría me asusta o Cuando pienso a las lecciones de geometría me pongo nervioso*;

- 11 elementos se refieren al aburrimiento (4 elementos más con respecto a la versión original) de tipo *La geometría me aburre o Las evaluaciones de geometría me aburren*.

Tabla 17

Distribución de los elementos en función de la categoría de emoción indagada

Emociones (AE)	Elementos		
	<i>Lecciones</i>	<i>Tareas</i>	<i>Evaluaciones</i>
Alegría-Diversión	1,4,8,12	1,6	1,5,12
Ansiedad	3,5,7,10	3,5,7	2,4,6,9,11
Aburrimiento	2,6,9,11	2,4,8	3,7,8,10

Fuente: elaboración propia.

El estudio conducido por Raccanello et al. (2018) ha examinado las propiedades psicométricas del instrumento, verificando que l'AEQ-ES puede ser utilizado para evaluar diversión, aburrimiento y ansiedad de los estudiantes de Educación Primaria italiana, tanto para lengua como para las matemáticas.

En este trabajo se quiso aplicar l'AEQ-ES para medir las emociones de los estudiantes de cuatro años de Educación Primaria con edades de entre 9 y 10 años, en geometría; para esto, así como lo aconsejan los autores, el término matemáticas (math) fue sustituido por el sustantivo geometría.

3.4.1.1 **AEQ-ES: Cuadro teórico de referencia**

La investigación teórica sobre los efectos de las actividades de *coding* sobre el aprendizaje ha puesto en evidencia una serie de estudios sobre las emociones en los entornos escolares que representan un elemento central, especialmente cuando el estudiante vive el momento del aprendizaje como un proceso pesado y agotador que tiene consecuencias a nivel emocional (Pekrun et al., 2009).

Así pues, la decisión de investigar las emociones dentro del proceso de aprendizaje de los niños deriva de la convicción de que estas están siempre presentes en la vida escolar de los estudiantes, desde temprana edad. Ellos

están orgullosos cuando consiguen buenas calificaciones, preocupados si no comprenden, enojados si piensan que los profesores los tratan mal, aburridos si el tema no les interesa.

En realidad, son pocos los estudios realizados en el ámbito científico con respecto a las emociones de los niños, aun teniendo estas un gran impacto en la motivación, el aprendizaje, la salud y el bienestar físico y psicológico (Lichtenfeld et al., 2012). De hecho, la mayor parte de los trabajos, abordan la medida de la ansiedad, en particular aquella asociada a las matemáticas, además, en niños de edades más avanzadas con respecto a los que frecuentan la Educación Primaria y principalmente en el ámbito académico (Lichtenfeld et al., 2012, Pekrun et al., 2011; Raccanello et al., 2018). El modelo de Pekrun sobre las emociones de logro (Pekrun, 2006) ha dado un impulso positivo al tema delineando tanto la definición como el complejo modelo de estudios previos y relacionados.

Pekrun trata las emociones en los contextos escolares y académicos. Según su teoría del control-valor (Pekrun, 2006) durante la fase del aprendizaje los estudiantes experimentan una serie de emociones que dependen de las diferencias individuales, de la disciplina que están estudiando, del contexto en el que se encuentran y de muchas otras características relacionadas en el que dichas emociones se presentan (Pekrun et al., 2002). Los efectos de estas emociones sobre el aprendizaje y el rendimiento académico están mediados por mecanismos cognitivos y motivacionales y están relacionados con un conjunto de factores antecedentes, entre los cuales construcciones debidas a la creencia sobre el control y sobre el valor de una tarea pero también a la naturaleza emocional que incluye la regulación de las emociones. A su vez, las emociones resultan de alguna manera relacionadas al éxito académico a través de mecanismos de tipo cognitivo y metacognitivo y dichas teorías, en los últimos años, han sido apoyadas por una cierta cantidad de datos aun siendo, dichas investigaciones dirigidas a estudiantes universitarios o de escuela secundaria y solo en menor medida a estudiantes de Educación Primaria (Lichtenfeld et al., 2012).

Las emociones de logro son consideradas por la psicología de la educación contemporánea, componentes importantes para la trayectoria de los estudiantes, para su bienestar y para los resultados alcanzados. El modo en el que se perciben dichos resultados, las acciones realizadas y la importancia atribuida condiciona definitivamente las emociones de logro (Pekrun & Perry, 2014). Por consecuencia a distintos niveles de control y valor corresponden emociones diversas que pueden relacionarse con la diversión (alto control, alto valor), el aburrimiento (bajo valor) y la ansiedad (bajo valor, bajo control).

Según la taxonomía propuesta por Pekrun las emociones se ven afectadas por el contexto dependiendo si es evaluativo o no evaluativo y pueden subdividirse según dos dimensiones:

- según su importancia o valor se pueden dividir en positivas y negativas;
- en base al nivel de activación (activación/desactivación) que se refiere al efecto que tienen sobre el aprendizaje se dividen en emociones activadoras y no activadoras;

La teoría del control-valor asume por lo tanto dos tipos de evaluación que se refieren, precisamente, al grado de control que el estudiante cree poder ejercer sobre las actividades y sobre las posibilidades de logro y el valor o la percepción de la importancia que se le atribuye a la actividad que se realiza y a los resultados que se obtienen.

Las emociones son, según Pekrun, el resultado de la combinación de un conjunto de procesos psicológicos que tienen en cuenta también la parte motivacional y no solamente están relacionadas con los resultados alcanzados. A la esperanza o a la ansiedad por superar o fracasar una prueba se agregan otras emociones, como la satisfacción de haber comprendido la lección o el aburrimiento que se experimenta durante la misma o incluso el enojo si es que esta no se ha comprendido. Un estudiante se divierte estudiando cuando considera la materia de estudio interesante y cuando se siente competente y seguro de poder dominarla; sus emociones están vinculadas al posible éxito o

fracaso que cree que logrará. Por lo tanto, el rendimiento y el aprendizaje de los estudiantes están influenciados por las emociones, positivas o negativas, que tienen también efectos sobre la motivación, el uso de distintas estrategias de aprendizaje y la aceptación de las reglas.

Las emociones positivas (placer, esperanza, orgullo) estimulan la motivación extrínseca vinculada al logro de resultados positivos o a la prevención de aquellos negativos, y la intrínseca relacionada con el interés y la curiosidad de aprender; en cambio, aquellas negativas (desesperación y aburrimiento), reducen la motivación y los efectos sobre el rendimiento. En particular, enojo, ansiedad y vergüenza pueden condicionar la motivación intrínseca pero pueden inducir una fuerte motivación extrínseca que lleva a evitar el fracaso y a promover el uso de estrategias de aprendizaje más eficaces.

Algunos estudios de tipo exploratorio muestran que emociones como diversión, esperanza, orgullo, satisfacción, enojo, ansiedad, desesperación, vergüenza y aburrimiento se presentan de manera frecuente en los estudiantes (Pekrun et al., 2011).

El cuestionario *AEQ*, elaborado, utilizando como referencia la teoría propuesta por Pekrun (2006) y por Pekrun et al. (2009), considera cuatro categorías de emociones:

- *emociones de activación positiva* que incluyen el placer al aprender, la esperanza y el orgullo;
- *emociones de desactivación positiva* como la satisfacción, la relajación después de haber alcanzado un objetivo con éxito;
- *emociones de activación negativa* como el anisa, el enojo y la vergüenza;
- *emociones de desactivación negativa* que incluyen la decepción, la desesperación y el aburrimiento.

Como ya se ha comentado anteriormente, los efectos de las emociones

sobre el aprendizaje académico están mediadas por mecanismos cognitivos y motivacionales (Pekrun et al., 2002). Las emociones de activación positiva pueden aumentar el interés y la motivación, aquellas negativas pueden disminuirlo mientras las emociones de desactivación positiva, como el alivio que se siente, pueden facilitar el desapego. En cuanto a las emociones negativas, como la ansiedad y la vergüenza, se registra que pueden tener efectos ambivalentes reduciendo la motivación intrínseca y aumentando la extrínseca.

En apoyo a la teoría del control valor de Pekrun, estudios empíricos muestran que las emociones de activación positiva están asociadas de forma positiva a los resultados obtenidos, a la mejora del pensamiento creativo, al uso de estrategias para el aprendizaje; mientras aquellas negativas como desesperación, orgullo y rabia resultan influenciar de manera negativa el rendimiento (Pekrun et al., 2009; Pekrun et al., 2011).

La presente tesis utiliza esta herramienta para medir las emociones de logro en el contexto italiano, y explora la especificidad de dichas emociones con respecto a entornos evaluativos (verificación) y no evaluativos (estudio en la escuela y en casa).

3.4.2 Instrumentos para la recogida de datos cualitativos

3.4.2.1 Observación no participante

Para recopilar los datos cualitativos y una parte de los cuantitativos - considerados útiles para los fines de la presente investigación- se realizaron varias observaciones dentro de ambas clases (grupo experimental y de control) durante todo el tiempo en el que se realizó la actividad didáctica relacionada con el estudio de las figuras planas en geometría.

Teniendo la convicción de que una de las mejores formas de responder a algunos interrogantes de la investigación es observando cómo se comportan las personas y/o cómo se ven las cosas, se procedió a activar la capacidad de

escucha y atención (Creswell, 2012). De hecho, la observación no es simplemente una mirada, general, no selectiva sino que consiste en mirar y anotar sistemáticamente sujetos en situaciones particulares de vida cotidiana con el fin de comprender emociones, actitudes y dinámicas relacionales (Cohen et al., 2013). Es un proceso intencional en el que el investigador tiene la oportunidad de recoger datos directamente de situaciones sociales naturales, analizando directamente aquello que sucede *in situ*. Al observar es posible recopilar datos sobre el ambiente físico en el que se trabaja (en el presente caso el aula), el ambiente humano (cómo está formado el grupo de clase), el interactivo (los tipos de interacciones en progreso) y la configuración del programa (la organización del trabajo).

En la presente investigación se llevó a cabo una observación no participante en la que el investigador decide no participar de forma activa durante el desarrollo de la lección y por lo tanto no se involucra directamente en la situación que pretende observar. Como investigador externo (Creswell, 2002) se “mantiene al margen” a mirar aquello que sucede influenciando lo menos posible todo aquello que sucede y tomando una posición externa a la realidad que se desea estudiar, sin interactuar con esta; los participantes no saben que están siendo observados durante todo el tiempo en el que se lleva a cabo la actividad (Fraenkel et al., 2011).

De hecho, para evitar que la presencia del investigador pueda tener efectos sobre el comportamiento de quienes están siendo observados, como lo aconsejan los expertos, nos presentamos previamente con los niños para habituarlos a la presencia de una persona extraña para ellos (Creswell, 2002). Queriendo entonces mantener la presencia en el contexto observado, se adoptó como estrategia el introducirse en clase, durante el desarrollo de las actividades académicas, para ser aceptado gradualmente por el grupo y no ser visto por los niños como un individuo extraño al cual temer.

Antes de realizar la observación real, se examinó con atención y con antelación todo el proceso de observación que se pretendía implementar; tras haber identificado los grupos de clases a observar se buscó entrar en el contexto lentamente, con breves observaciones iniciales, realizadas a partir del año escolar precedente, que permitieron un primer contacto con el ambiente y, al mismo tiempo, darse a conocer y ser aceptado por los niños. Después de haber confirmado la idea de conducir una observación en el rol de no participante y de haber tomado confianza con el ambiente, se procedió a realizar la observación real a través de la construcción de las notas de campo para identificar los aspectos considerados significativos para el trabajo, como actitudes, comportamientos, relaciones, emociones (Creswell, 2002).

Así pues, la observación realizada se guió por las hipótesis, inherentes a la estructura misma del instrumento de registro y clasificación que, obviamente, quedó relacionado con el cuadro teórico que guía al observador en la detección. El primer paso llevado a cabo fue el de definir qué aspectos anotar durante el proceso de observación con referencia obvia a las hipótesis y a los objetivos que guían la investigación. El campo cognitivo se redujo enfocándose en los aspectos específicos en los cuales poner la atención, a través de la identificación de las categorías a observar y a los indicadores relacionados que pueden ver surgir información útil en cuanto a las variables de interés.

Entre los distintos enfoques propuestos por varios autores, para conducir la observación dentro de una clase, se consideró útil, para los fines del presente trabajo de investigación, poner una mayor atención a la descripción cualitativa de los fenómenos que ocurren durante la lección además de medir sistemáticamente la frecuencia con la cual ocurren ciertas situaciones (Poliandri et al, 2014) a través de un análisis cuantitativo. Por lo tanto se decidió integrar las técnicas de detección cualitativas con las cuantitativas, y junto con la detección de las frecuencias de determinados comportamientos y emociones contemporáneamente se recogió información de tipo cualitativo, con referencia, por ejemplo, a las interacciones entre los niños, las acciones que se desarrollan

durante las actividades académicas, el comportamiento social y no verbal (Yin, 2015). Con respecto a este último factor, tiene una cierta importancia el tono de la voz, la mímica, las actitudes, los gestos, señales que, sin embargo, dada su singularidad, pueden dar espacio a la interpretación pero al mismo tiempo proporcionar información útil.

En particular se dirigió la atención hacia algunos aspectos considerados significativos y relacionados de alguna forma al proceso de aprendizaje que se lleva a cabo en clase, poniendo en evidencia ciertas situaciones de comportamiento con referencia explícita al esfuerzo o a la falta de este, al interés o al desinterés, a la atención o la desatención que muestran los estudiantes. Se observó el esfuerzo cognitivo que mostraron los niños con referencia al propio trabajo y al trabajo en pequeños grupos en cuanto a la concentración y la participación. Para los componentes respectivos a la parte del proceso de enseñanza y aprendizaje relacionados con el esfuerzo emotivo y motivacional y a la participación de los estudiantes de geometría se utilizaron algunos indicadores propuestos por Helme & Clarke (2001).

Helme y Clarke (2001) proporcionan un análisis detallado del esfuerzo de los estudiantes bajo la forma de indicadores que pueden detectarse durante la interacción en clase, mostrando como se manifiesta su participación durante los distintos tipos de trabajo (individual, de grupo con y sin profesor y con toda la clase). Dichos elementos surgen a partir de observaciones y entrevistas. Los indicadores señalan situaciones visibles de comportamiento, con particular referencia a la atención, la concentración y la participación a través de preguntas o respuestas dadas, a la comunicación a través de la escucha, el habla o los gestos. Por lo tanto, la participación de los estudiantes es, para los autores, un indicador del pensamiento deliberado y específico de una tarea mientras participan a una actividad, durante la cual los estudiantes se esfuerzan realmente (Helme & Clarke, 2001). Parece que el esfuerzo cognitivo puede ser constantemente reconocido por indicadores lingüísticos y comportamentales

específicos y es promovido por aspectos particulares de la situación de la clase, de las actividades y del individuo.

Se ha buscado registrar también el componente del esfuerzo emocional analizando las distintas facetas del entusiasmo, la felicidad, el interés, la diversión, la alegría o el aburrimiento, la ansiedad, la infelicidad, el enojo, escogiendo como indicadores, aquellos propuestos en la teoría del control-valor (Pekrun, 2006; Pekrun et al., 2009)

Para responder a las preguntas planteadas se consideró realizar una observación de tipo semi estructurada en la cual, con antelación, se conoce que cosa observar. Así pues, en la presente investigación, se utilizaron herramientas de recopilación y clasificación de la información, como las cuadrículas de observación, que contienen en parte elencos de los comportamientos esperados que se desean estudiar y en parte dan la posibilidad, a quien registra en la hoja, de escribir libremente aquello que considere interesante. Los elementos considerados útiles para los fines del trabajo fueron, por lo tanto, reportados en una hoja de observación que se preparó de tal manera que se pudieran registrar eventos, circunstancias y dinámicas definidas, de acuerdo con una cuadrícula semiestructurada.

Se trata de una cuadrícula de observación sintética subdividida en dos partes: en la primera se registra la información general sobre la clase (fecha y hora, duración de la observación, alumnos presentes y entorno de clase) mientras en la segunda hay un diagrama dividido en tres columnas para poder registrar y llevar a cabo el análisis de la actividad realizada de forma más práctica y rápida. El esquema incluye descriptores específicos como *los procesos, las actividades, las estrategias y las relaciones* (Anexo G). Junto a los elementos individuales es posible además anotar las posibles observaciones personales, para permitir a quien observa precisar aspectos, circunstancias y reflexiones propias con el fin de integrar lo más posible la información recogida.

Como ya se ha mencionado, la observación se llevó a cabo durante las actividades programadas y ocupó, en términos de cuantificación del tiempo, doce horas totales de lecciones para el grupo experimental y once horas de lecciones en el aula para el grupo de control. Después de haber recogido los datos a través de los instrumentos indicados anteriormente, a partir de la observación realizada, se realizó una primera codificación de los datos, transcribiendo en forma discursiva lo que surgió y registrado en manera más secuencial y anotando, también a parte, los aspectos considerados más significativos para los fines de la investigación. Las imágenes fotográficas se modificaron con el fin de cubrir los rostros de los niños para proteger su privacidad.

3.4.2.1.1 **Método de recopilación de datos de observación**

Tras haber recogido y digitalizado todos los documentos provenientes de la observación realizada en las clases, a través de las notas de campo y las imágenes fotográficas, se identificaron los dominios dentro de los cuales poder categorizar los temas objeto de interés. La técnica de codificación utilizada permitió la atribución de lo observado a través de la evaluación de los hechos y los comportamientos que surgieron durante el proceso de enseñanza y de aprendizaje a los términos de prevalencia positiva, negativa o neutral. Después de haber identificado estos tres temas principales, se revisaron todas las transcripciones de las notas de campo con el fin de asegurarse de que el objeto de interés estuviese presente y se relacionara a estos dominios. En el complejo surgieron catorce códigos de los cuales cinco con referencia a la categoría de los factores con prevalencia positiva, siete en aquella con referencia a los de prevalencia negativa y cuatro códigos que comprenden los aspectos más generales del proceso de enseñanza y aprendizaje y por este motivo incluidos en el dominio con valor neutral (Tabla 18).

Tabla 18*Distribución de los códigos identificados después de la observación*

Dominio	Código
Prevalencia positiva	Participación - Satisfacción alegría y diversión - Motivación Interés y curiosidad
Prevalencia negativa	Preocupación y ansiedad - Desilusión Enojo – Aburrimiento - Distracción Desinterés - Aburrimiento/Cansancio
Prevalencia neutral	Relaciones entre pares - Relaciones maestra- alumno Estrategias–Actividades

Fuente: elaboración propia.

En el dominio que corresponde a los factores que se catalogan con prevalencia positiva se codificaron las principales actitudes, comportamientos y emociones que surgieron en el aula y que se relacionan precisamente con situaciones clasificables de manera objetiva como positivas en un contexto escolar; como ya se ha mencionado, en total, se identificaron cinco códigos. En particular, se buscaron y se identificaron elementos relacionados con la presencia o intervención por parte de los niños durante el desarrollo de la actividad didáctica propuesta a través de lecciones o tareas y se resaltaron con el código de participación, refiriéndose a todos aquellos casos en los que los niños alzaban la mano o buscaban interactuar de alguna manera durante el proceso de construcción del conocimiento, incluso a través del asentimiento de la cabeza, con movimientos del cuerpo o respondiendo si el profesor lo solicitaba. El código satisfacción se utilizó para identificar los casos en los que los estudiantes se sintieron gratificados y se mostraron, a través de comentarios, expresiones visuales o movimientos corporales, claramente satisfechos y complacidos. Con los términos alegría y diversión se codificaron todas aquellas actitudes a través de las cuales los niños mostraron claramente un impulso irreprimible para trabajar a través de una participación total, alegre, lejos de la ansiedad y la preocupación, y con admiración por lo visto y lo escuchado, seguido por manifestaciones de dichas emociones con expresiones verbales, movimientos físicos y tono de la voz.

El código denominado motivación busca comportamientos específicos de los alumnos que muestran un cierto esfuerzo durante el desarrollo de la tarea en clase, relacionado con las ganas de aprender más, de querer sentirse más competente. No es una simple participación sino una real motivación hacia el aprendizaje que lleva a los alumnos a efectuar intervenciones espontáneas, a tener un mayor control sobre aquello que hace, sintiéndose involucrado de forma activa en la ejecución de una tarea o de un trabajo en general. Curiosidad e interés es el código que engloba dos términos que se refieren a los casos en los que los niños muestran una cierta curiosidad y deseo por conocer, a través de preguntas pertinentes y curiosas dirigidas al profesor o con comentarios que sirven como puntos interesantes para proseguir con la lección.

Con respecto a los factores con características de prevalencia negativa, se codificaron dentro de esta categoría emociones, comportamientos y actitudes que se registran en un contexto académico y se pueden evaluar según un significado prevalentemente negativo o atribuible a este. Se consideraron siete códigos en total. Con el código preocupación se hace referencia a todas aquellas situaciones en las que los niños han mostrado un estado de ánimo caracterizado por temores, dudas, aprensión al momento de realizar una tarea, todo esto señalado a través de expresiones verbales o con solicitudes de ayuda. El término *desilusión* codifica todas aquellas circunstancias en las que los alumnos han mostrado un malestar moral provocado por un resultado contrario al que ellos se esperaban como una calificación negativa, no lograr llevar a cabo una tarea, no comprender una lección. Durante la observación en clase se tomó también nota de los episodios durante los cuales surgieron manifestaciones de ira o cólera entre los niños a través de reacciones verbales o físicas, activadas por peleas y malentendidos con los compañeros. Dichos comportamientos, codificados con el término enojo, se reconocieron en los momentos en que surgieron las típicas expresiones faciales de quién está enojado o por el notorio aumento en el tono de la voz que se vuelve más estridente y amenazante. Con el término distracción se codificaron todos aquellos comportamientos en los que los niños mostraron poca atención y concentración durante todas las fases del aprendizaje escolar,

manifestadas a través de expresiones ausentes de la realidad, intervenciones no pertinentes, negligencia al llevar a cabo una tarea, superficialidad o cuando se les veía entablar una conversación con un compañero, incluso con voz baja. Los casos en los que los niños mostraron poco interés a la propuesta formativa ofrecida a través de actitudes de molestia, de rechazo a participar, de indiferencia a través de acciones y comportamientos inadecuados al contexto y al momento, como por ejemplo pedir frecuentemente ir al baño, se catalogaron con el código desinterés. Posteriormente se atribuyó la prevalencia negativa a las situaciones en las cuales los niños parecían aburridos, insatisfechos o fastidiados con respecto a lo que estaban haciendo. El código aburrimiento comprende también las actitudes de aquellos niños que manifestaron una sensación de molestia ante algo que para ellos resultaba monótono o inútil a través de expresiones faciales, frases o incluso simples acciones, como bostezar. Se incluyó también el código aburrimiento/cansancio para tener en cuenta todos aquellos casos en los que los niños parecieron aburridos o bostezaron cerca del final de la ora de geometría (en los últimos 10/15 minutos), momento en el que pudiera presentarse la sensación de cansancio; ante la duda de la objetividad se prefirió agregar esta codificación adicional, diferenciándola del aburrimiento.

Para completar la recogida de datos en campo, se decidió considerar una categoría que abarcase todas las situaciones identificadas en clase relacionadas con las relaciones sociales que surgieron entre los compañeros (relaciones c_c) y entre el profesor y el alumno (relaciones m_a), a las estrategias puestas en capo por el docente durante el proceso de enseñanza y aprendizaje (estrategias) y a las actividades relacionadas realizadas por el docente (actividad). A estos códigos se les atribuyó una valencia natural en la fase inicial de codificación, ya que no es posible lograr clasificarlos de manera objetiva como positivos o negativos.

La codificación de los textos y de las imágenes se llevó a cabo agrupando datos similares y tomando en cuenta, además de las situaciones que surgían de forma explícita también las consideraciones y participaciones que se

interpretaron como un intento de parafrasear una emoción o una sensación, como por ejemplo "Es muy guay" se codificó con el código alegría y diversión.

3.4.2.2 **Entrevista**

La entrevista, así como la observación no participante, es una herramienta de investigación cualitativa y se lleva a cabo a través de preguntas generales abiertas que plantea el investigador a uno (entrevista individual) o más sujetos cuyas respuestas se registran y se transcriben en un medio digital, para su posterior análisis (Creswell, 2002, p. 217). Utilizar la entrevista en la investigación educativa como herramienta lleva a tener una visión de los seres humanos no como elementos manipulables por el investigador sino más bien como participantes activos, con sus interpretaciones de los hechos y con la manifestación de su punto de vista, pero sin olvidar que pueden estar condicionados por el prejuicio obstaculizando la investigación. (Cohen et al, 2013). Cuando se decide realizar una entrevista, en una investigación cualitativa, se deben seguir algunos pasos fundamentales para su éxito, entre los cuales la identificación de los individuos a entrevistar, a través de estrategias de muestreo focalizadas, la decisión del tipo de entrevista (individual, grupo focal, entrevista por correo electrónico, etc.), el uso de herramientas de registración adecuadas, la elección del lugar oportuno (Creswell, 2002, p. 221).

En la presente investigación, se buscó efectuar una entrevista individual a uno de los actores que participaron en la investigación: el docente de la clase que recibió el tratamiento, que llevó a cabo la actividad experimental en el aula. El objetivo fue completar los datos recogidos durante el desarrollo de la actividad de investigación, dando voz al docente que contribuyó a realizar la experimentación en la clase en la que enseña las matemáticas. La entrevista se realizó dejando que el docente expresara de manera libre sus propias opiniones e impresiones al final de la actividad desarrollada.

Para dar fluidez a la entrevista, se preparó una lista de control constituida por una lista que contenía las preguntas planteadas, elaborados en base al marco teórico de referencia y las preguntas de investigación; la entrevista individual duró aproximadamente quince minutos con un total de veinte preguntas y se registró con un registrador vocal cuyo archivo de cargo en MAXQDA Analytcs Pro (Release 18.2.0) para realizar el análisis de contenido.

El objetivo principal de la entrevista fue el obtener el punto de vista del docente en cuanto al uso del *coding* como herramienta didáctica, sus fortalezas y debilidades, para los efectos que el docente pudo percibir dentro del proceso de enseñanza y aprendizaje en cuanto a los aspectos cognitivos, comportamentales, sociales, motivacionales y emotivos. Los datos recogidos se triangularon con aquellos obtenidos a través de la observación.

3.5 Tratamiento: actividad propuesta al grupo experimental

Hacer *coding* o programar, escribir un código manipulando el lenguaje del ordenador durante las actividades de geometría: esta es, en resumen, la actividad que se realizó en la clase de cuarto curso de Educación Primaria "Don Bosco" de Ribera (Ag) por la docente de matemáticas y tecnología y se puede catalogar dentro del género de dibujos geométricos y generativos a través de los cuales es posible experimentar el uso de la programación y del pensamiento computacional para la creación de figuras geométricas. A dicha actividad se le llamó "*Geometría con Scratch*".

La propuesta operativa lleva a enfrentar el estudio de las leyes de la geometría de una forma distinta, más creativa, partiendo de la construcción de un simple algoritmo, para diseñar en Scratch que, como se ha demostrado en varios estudios, resulta ser un ambiente válido de aprendizaje integral.

Los objetivos que se pretenden alcanzar en términos de habilidades y competencias son los expresados en el plan de estudios de geometría para las clases de cuarto curso, así como lo programan los docentes mismos. Los

alumnos podrán aprender, después del desarrollo de las actividades con y sin *coding*, a clasificar las figuras planas, analizando las características principales de los polígonos entre las cuales ángulos, lados, ejes de simetría, diagonales y alturas; sabrán construir las figuras simétricas, identificar traslaciones y rotaciones adquiriendo habilidades relacionadas al saber representar, comparar y analizar las figuras geométricas a partir de situaciones reales.

Específicamente, el objetivo del tratamiento que se efectuó en el ámbito de la experimentación es el de construir, a través de Scratch, secuencias lógicas en la creación de figuras geométricas con el fin de lograr encontrar una regla general para poder generarlas.

La actividad preparada para los estudiantes hace referencia, en primer lugar, al marco teórico definido por Papert y Harel (1991), según el cual el aprendizaje se alcanza a través de la construcción del conocimiento con experiencias en las que los estudiantes están directamente involucrados gracias al uso de artefactos externos. De hecho, el construccionismo, representa claramente una referencia válida para este estudio, ya que permite explorar de qué forma los niños construyen los conceptos propios del pensamiento matemático y geométrico mientras que utilizan un artefacto externo como Scratch y en qué forma interactúan con los demás (Miller, 2019). Además se hace referencia al modelo Scratch propuesto a través del marco teórico de Brennan y Resnick (2012), descrito en el capítulo 2, cuyas bases son de naturaleza construccionista, y en particular al aspecto relacionado con la posibilidad de crear un ambiente didáctico simplificado en el cual llevar a cabo diversas prácticas computacionales (experimentar e iterar, hacer pruebas y depuración, reutilizar y re mezclar, abstraer e modular). La decisión de tomar como referencia dicho modelo se debió al hecho de que Scratch es usado con más frecuencia que cualquier otro software con características similares (Roig-Vila & Moreno-Isac, 2020) tanto que en enero del 2018 estaba en el primer lugar de popularidad entre todos los lenguajes de programación con fines educativos con un alto número de usuarios a partir de los cuatro años de edad (Montiel &

Gomez-Zermeño, 2021; Zhang & Nouri, 2019). El experimento didáctico consistió en nueve lecciones en total, de las cuales seis tuvieron duración de 60 minutos y los niños estuvieron expuestos al *coding* online y tres con duración de 120 minutos, en las que se realizaron actividades de *coding* unplugged y lecciones constructivistas (Anexo E).

Para poder trabajar con Scratch, los niños fueron llevados al laboratorio multimedia y colocados frente a la computadora, después de haberlos dividido en pequeños grupos de dos o tres personas por estación, mediante una selección mayoritariamente aleatoria.

El docente, durante la primera sesión de *coding*, asigna un proyecto para realizar ilustrando las funciones de Scratch relacionadas con la actividad; les presenta el bloque *bolígrafo hacia abajo* necesario para hacer que un *sprite* siga una línea sobre el *stage*. El primer paso que los niños deben realizar es crear un proyecto en el que el personaje que eligieron (*sprite*), a través de bloques del grupo *bolígrafo*, debe moverse dejando huella de su recorrido. Con este primer estímulo se busca invitar a los alumnos a la experimentación de diferentes trayectorias manipulando los comandos de movimiento. Una de las posibles respuestas al estímulo es la creación de figuras que *parecen* figuras geométricas reales.

Así pues, la primera tarea se debe materializar con la realización de un *script* que permite dibujar un cuadrado para continuar con las siguientes solicitudes con el fin de hacer que el *sprite* se mueva para diseñar triángulos y otros polígonos (pentágono, hexágono, decágono etc.) (Figura 5). El objetivo es razonar sobre la longitud de los lados y la amplitud de los ángulos y lograr encontrar una regla general que permita dibujar un polígono con cualquier número de lados. A esto se le suma el descubrimiento del uso de los ciclos, de las variables, la investigación, la identificación y corrección de los *bug* (errores), elementos que caracterizan la programación informática. A los niños se les da la

posibilidad de trabajar escogiendo, de común acuerdo, escenarios (*stage*) y personajes (*sprite*).

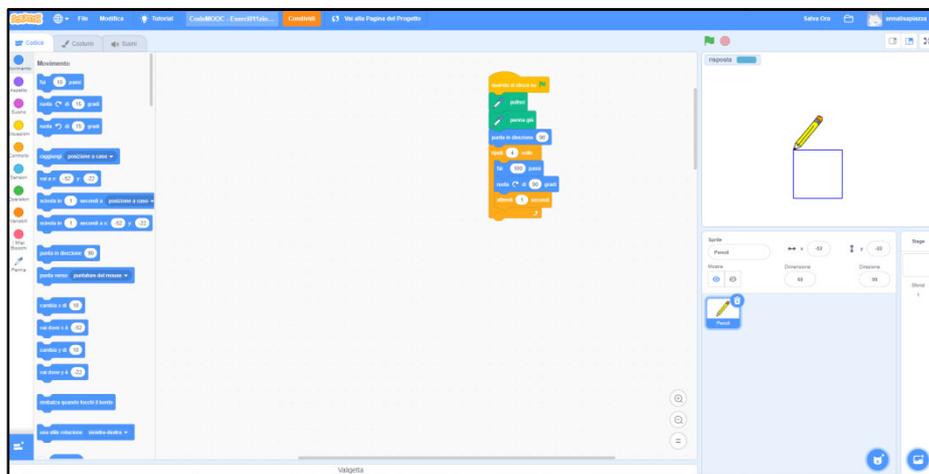


Figura 5. Tarea 1: dibujar un cuadrado con Scratch

La actividad de *coding* se concluye proponiendo realizar un diseño generativo a partir de la función escrita, utilizando los principios de repetición, aleatoriedad, variación de tamaño y trazo, a través de la cual obtener composiciones y/o progresiones de polígonos (Giordano & Moschetti, 2016).

Las actividades descritas son principalmente de laboratorio, requieren el uso del ordenador y se alternan por el docente con lecciones que profundizan y amplían los conceptos de geometría que se desean transmitir. De hecho, entre una y otra sesión de *coding* el proceso de construcción del conocimiento se potencializa a través de actividades, ejercicios, discusiones en clase durante las que se abordan los temas a través de un enfoque constructivista que prevé el estudio y el análisis crítico de todas las propiedades geométricas de las figuras planas.

Al final de cada sesión, tanto en el laboratorio como en el aula, el docente invita a los niños a reconstruir con palabras aquello que han aprendido estimulando así la generalización, la descontextualización de los conceptos que han surgido y el trabajo compartido.

El proceso de aprendizaje, además de alcanzar el conocimiento de los conceptos de geometría, brinda la posibilidad al alumno de aprender los conceptos base de la programación como las secuencias, los ciclos, los eventos, los operadores y el tratamiento de los datos; enfrentando este tipo de problemas computacionales ellos experimentan las prácticas del pensamiento computacional como la reutilización, la remezcla, el ser incremental e iterativo, probar y depurar, etc.; desarrollan también las perspectivas propias del pensamiento computacional como expresarse, discutir y conectarse con el mundo digital (Brennan & Resnick, 2012).

A través de la plataforma Scratchse busca ofrecer a los estudiantes la posibilidad de personalizar su experiencia de aprendizaje de la geometría, permitiéndoles ampliar sus conocimientos de manera distinta a la tradicional. Los niños realizan una actividad con características lúdicas y, mientras juegan, se les proporciona una retroalimentación visual instantánea, que es una parte fundamental del *debug* (depuración) de su proceso de aprendizaje, claramente distinta con respecto al aprendizaje tradicional en clase; de hecho, de esta manera pueden verificar inmediatamente el trabajo que han realizado, reconsiderar la situación y corregir sus movimientos (Tan et al., 2019).

Se adjunta el detalle de todo el diseño de la actividad de *coding* como soporte de la geometría aplicada durante el tratamiento en el grupo experimental.

3.5.1 Procedimiento

El plan de la investigación empírica preveía la identificación de los grupos de control y experimental dentro del Instituto Comprensivo "Don Bosco" de Ribera. En particular, queriendo utilizar la geometría como la disciplina sobre la cual trabajar a través del *coding* se consideró estratégico dirigir el estudio a las clases de cuarto curso de primaria, cuando los temas que se tratan comienzan a ser más complejos y abstractos. Para este fin se pidió la disponibilidad para

colaborar, bajo la indicación de Director Escolar del Instituto, de los docentes de matemáticas de las clases de cuarto curso en el año escolar 2018/2019.

Adquirió una cierta importancia, en la selección de la clase experimental, la formación de la docente de matemáticas que mostró haber tenido experiencias didácticas previas en cuanto al uso del *coding* y actualizaciones sobre el tema en estudio.

El plan consistió en llevar a cabo, en un periodo determinado el año escolar, entre marzo y mayo, una parte del plan de estudios de geometría a través del uso del *coding*, en la clase identificada como el grupo experimental. A ambos grupos se les aplicaron las dos pruebas G1 y G2 respectivamente al inicio y al final cuando completaron las actividades programadas.

A diferencia del grupo experimental, el grupo control desarrollo el programa preestablecido abordando los mismos temas, pero a través del método que tradicionalmente utiliza el docente (condiciones ordinarias).

En todas las actividades realizadas por las dos docentes tanto de la clase del grupo experimental como la de la clase del grupo control, así como se describe en el párrafo anterior, estuvo presente el investigador que realizó una observación de tipo no participante, observando las actitudes, comportamientos y dinámicas dentro de las clases en fase de trabajo, usando una cuadrícula de observación semiestructurada como herramienta de apoyo.

Tras haber realizado la actividad y la prueba final G2 los niños de ambos grupos se invitaron a responder al cuestionario *AEQ-ES* en versión italiana. A cada cuestionario se le atribuyó un código de identificación para garantizar la privacidad.

3.6 Análisis de los datos

El trabajo de investigación en campo requirió obtener algunos permisos para llevarlo a cabo. Para realizar el trabajo y la recogida de los datos se solicitó el permiso a la Directora Escolar del Instituto Escolar la cual autorizó al investigador para tener acceso, en calidad de observador, dentro de las aulas de la escuela. Para la aplicación de las herramientas utilizadas, como las pruebas de geometría y cuestionario, se solicitó la autorización escrita de los padres de familia (módulo de consentimiento informado) con el pleno respeto de la ley sobre la privacidad tomando en cuenta la edad de los sujetos que participaron en la investigación. Ambos padres y/o tutores firmaron un módulo que se les entregó y que contenía todo el procedimiento y la finalidad del tratamiento de los datos como lo prevé el Reglamento UE 2016/679 en materia de tratamiento de datos personales y de privacidad. El estudio fue aprobado y autorizado por el Director Escolar del Instituto, después de haber obtenido las autorizaciones necesarias por parte de los padres de familia.

Una vez obtenidos los permisos necesarios se llevó a cabo la actividad planeada y la rde datos para posteriormente procesarlos.

Antes de examinar los datos recogidos se planificó el trabajo a desarrollar teniendo en cuenta dos distintos tipos de análisis para efectuar, el cuantitativo y el cualitativo (Tabla 19).

Tabla 19

Análisis de los datos

	Herramientas	Análisis
Análisis cuantitativo	Prueba inicial G1 Prueba final G2 Cuestionario AEQ_ES	Estadístico-descriptiva Inferencial
Análisis cualitativo	Observación no participante Entrevista al docente	Estadístico-descriptiva Codificación y triangulación de datos

Fuente: elaboración propia.

En cuanto al análisis de los datos cuantitativos se decidió elaborar los datos recogidos con el programa de estadística aplicado en el campo de la Educación y de las Ciencias Sociales, IBM SPSS versión 22. Se realizó un análisis descriptivo calculando las medidas de posición y tendencia central, relaciones y características. Posteriormente, para comparar el grupo de control con el grupo experimental, se realizó un análisis inferencial, que permitió verificar si la distribución de los datos era normal o no, y por consecuencia, aplicar el tipo de prueba estadística más idónea (Creswell, 2002).

Para ambas pruebas aplicadas, con el fin de describir las características de base de los datos recopilados, se calculó, en particular, la frecuencia de las respuestas correctas dadas tanto en el grupo control como en el experimental. Después de haber realizado el control de la normalidad de la distribución se procedió a comparar las varianzas obtenidas en ambos grupos, con relación a la prueba inicial y a la final.

Se realizó un análisis estadístico descriptivo del cuestionario calculando el promedio y la desviación estándar para cada elemento después de haberlas clasificado en función de las emociones a investigar en tres grupos: alegría/diversión, ansiedad y aburrimiento. Además de realizó el cálculo de la frecuencia de respuestas y su relativo porcentaje para cada grado de acuerdo con la escala Likert. De la misma forma, para las pruebas de aprendizaje, se efectuó el control de la distribución de los datos y se verificó la presencia eventual de diferencias significativas en la consecución de las emociones entre el grupo de niños que se expusieron al tratamiento con el *coding* y el grupo de niños que trabajó siguiendo el modelo tradicional.

Con referencia al análisis cualitativo de los datos, en relación a la observación realizada, se procedió a recoger los datos disponibles que se obtuvieron principalmente de las notas de campo y de las consiguientes transcripciones, de las imágenes fotográficas que se pudieron tomar sin condicionar los comportamientos y las emociones de los niños, transfiriéndolas

posteriormente en el software MAXQDA Analytcs Pro (Release 18.2.0) con el fin de realizar una cuidadosa elaboración y un análisis estadístico-descriptivo, del que se calcularon promedio, desviación estándar, frecuencias, tendencias. El mismo procedimiento se realizó después de haber recabado y codificado la entrevista a la docente del grupo experimental.

Posteriormente, los resultados obtenidos se compararon con aquellos encontrados en la literatura, revisando los límites del estudio y terminando con sugerencias para las futuras investigaciones.

3.7 Conclusiones

En este capítulo se describió la metodología que se siguió para responder a las preguntas y a los objetivos establecidos en la presente tesis doctoral.

Nos preguntamos como punto de partida, cómo los niños se involucran en el estudio de la geometría y qué emociones surgen en el campo de la enseñanza durante el desarrollo de las lecciones, la ejecución de las tareas para la casa y cuando se llevan a cabo las evaluaciones. Se desea evaluar si en el grupo al que se le aplicó el tratamiento a través de las actividades de *coding* se registra algún cambio en cuanto al aprendizaje, el esfuerzo, los resultados y en la percepción de las emociones.

Como ya se ha mencionado, se trata de una investigación *cuasi experimental* que utiliza como herramienta para la recogida de los datos: el cuestionario *AEQ-ES*, dos pruebas de geometría denominadas con las siglas G1 (prueba inicial) y G2 (prueba final), los resultados de la observación no participativa realizada en ambos grupos y de la entrevista al docente de la clase experimental.

La selección de la muestra tuvo lugar en la localidad de Ribera, en provincia de Agrigento, una ciudad de aproximadamente 19.000 habitantes, cuya zona de influencia gira en torno a tres instituciones integrales. Teniendo la necesidad de identificar una muestra constituida por dos grupos de clase y habiendo podido contar con la colaboración y disponibilidad de la Directora Escolar de uno de estos tres Institutos presentes en el territorio se seleccionaron dos clases de cuarto curso de Educación Primaria, cuyas características resultaron responder a aquellas necesarias para la investigación. La muestra está constituida por 46 participantes subdivididos entre grupo experimental ($n=25$) y grupo control ($n=21$).

El capítulo también muestra los procedimientos implementados para la construcción de las pruebas de geometría mientras que su validación se reporta en el capítulo siguiente, la descripción del procedimiento utilizado durante el tratamiento, los resultados de los estudios relacionados con la validación y adaptación en el idioma italiano del cuestionario *AEQ-ES* aplicado a los alumnos participantes a la investigación y los métodos de recogida y codificación de los datos provenientes de la observación.

Capítulo 4

Resultados

4. Introducción

Uno de los objetivos principales de la presentetesis doctoral fue el verificar si el uso del *coding*, como método pedagógico y didáctico, además de promover el pensamiento computacional también puede producir cambios en el proceso de enseñanza y aprendizaje de la geometría principalmente en cuanto a los resultados obtenidos, a las emociones de logro y a los factores que directa o indirectamente tienen que ver con dicho proceso.

Los resultados obtenidos se dividieron en tres secciones. En la primera se realizó el análisis de validez de contenido de la prueba inicial G1 y final G2 de geometría; en la segunda se presentan los resultados relacionados con la aplicación de la prueba inicial y la prueba final y los resultados del cuestionario aplicado después de las actividades a los dos grupos y por último, en la tercer sección se presentan los resultados relacionados con las observaciones que se llevaron a cabo en las clases durante las cuales surgieron algunos factores interesantes, de esta manera se logró analizar, también desde el punto de vista cualitativo, los posibles efectos de la aplicación del *coding* como metodología didáctica.

4.1. Análisis de la validez de contenido y fiabilidad de las pruebas

De forma general, la validez de una herramienta se valora en función de cómo la medición tomada a partir de las puntuaciones que se le atribuyen resulta adecuada, significativa y útil para el estudio que se está realizando (McMillan et al., 2005) y es una forma de dar mayor fuerza a las conclusiones

obtenidas. En este trabajo, se examinó el índice de validez de contenido de las dos pruebas de geometría G1 y G2 (prueba inicial y prueba final) que se aplicaron a los niños, tomando como referencia el procedimiento descrito por Lawshe (1975).

La validez de contenido mide el grado con el que una herramienta de medición determinada resulta ser adecuada en cuanto a los contenidos que trata, en ausencia de omisiones y partes desequilibradas (Kerlinger & Lee, 2002). Por lo tanto, los contenidos expresados por los indicadores deben ser coherentes con lo que se pretende medir y representativos de los objetivos establecidos en la investigación. Uno de los métodos que se suele utilizar para examinar la validez de contenido es solicitar el juicio de sujetos, externos a la investigación, expertos en el tema, los cuales tienen la tarea de definir el grado de relevancia de los elementos según el dominio de los contenidos que se les presentan.

Lawshe (1975) propuso un procedimiento muy utilizado para examinar la validez de contenido de una prueba, que consiste en calcular, a partir de la opinión de cada experto, un índice de validez de contenido definido con el término en inglés *Content Validity Index (CVI)* para cada elemento y para la prueba completa. Como lo propone este autor, el nivel de validez de contenido depende del número de expertos que evalúan el elemento individual como relevante y del hecho de que al menos la mitad de los expertos consideren relevante el elemento, obteniendo así un elevado valor del índice de validez.

Así pues, en la fase relacionada con la validez de contenido, el conjunto de elementos es revisado por un grupo de personas bien informadas sobre el tema que pueden confirmar o invalidar la definición del constructo de interés. Los expertos pueden evaluar si los elementos realmente capturan el constructo (o los constructos) y analizar la claridad en la formulación de las preguntas. En esta fase se eliminan los elementos considerados inadecuados o no relevantes,

se formulan nuevamente los elementos poco claros y se incluyen los aspectos del tema que se hayan ignorado (Cooper et al., 2012).

Además, para ser válido, el conjunto de preguntas no solo debe investigar el constructor de interés, sino que debe poder abarcar todo el dominio, representando todos los aspectos. Por tanto, la validez del contenido resulta es más sencilla cuando el dominio está bien definido.

En el presente trabajo, para obtener la validez de contenido de las pruebas de geometría G1(prueba inicial) y G2 (prueba final) se solicitó la opinión de 20 expertos, docentes del área científico-matemática en Educación Primaria, a los que se les solicitó indicar para cada elemento de cada prueba cuanto era relevante para ellos en una escala constituida por cuatro posibles alternativas (NR = *no es relevante*, PR = *poco relevante*, R = *relevante* y MR = *muy relevante*). Una vez recogidos todos los datos se procedió a aplicar la fórmula de Lawshe (1975) para el cálculo del índice de validez de contenido según la cual:

$$CVI = [n_e - (N/2)] / (N/2)$$

En la que n_e es el número total de expertos que han definido el elemento como relevante o muy relevante y N el número total de expertos que participaron en la validación.

En el presente caso, se calculó un índice para cada elemento de cada prueba para después calcular el *CVI* global de ambas pruebas. De acuerdo con Lawshe (1975), con la participación de 20 expertos, es necesario obtener un valor del *CVI* de por lo menos .42 para considerar el elemento como significativo.

La tabla 20 muestra que se registraron valores elevados en el índice de validez de contenido para todos los elementos de la prueba inicial. Los

elementos 10, 11a y 11b consiguieron una validez de contenido perfecta y los demás oscilan entre .73 (elementos 2, 3a, 3b, 4, 6b) y .87 (elementos 5, 6a, 7, 8, 9a,9b,12). El elemento 1 que obtuvo la puntuación de .6 más bajo que los demás, aun entrando en los límites, se desechó. El índice de validez de contenido global para la prueba inicial es de .85 lo que permite confirmar la validez de contenido de la prueba.

Tabla 20

Índice de validez de contenido de los elementos del pretest de geometría (G1)

Elemento	No relevante	Poco relevante	Relevante	Muy relevante	R+MR	CVI
1.	0	3	10	2	12	.60
2.	0	2	10	3	13	.73
3a.	0	3	8	4	12	.73
3b.	0	2	7	6	13	.73
4.	0	2	5	8	13	.73
5.	0	1	6	8	14	.87
6a.	0	1	8	6	14	.87
6b.	0	2	7	6	13	.73
7.	0	1	11	3	14	.87
8	0	1	7	7	14	.87
9a.	1	0	11	3	14	.87
9b.	1	0	9	5	14	.87
10.	0	0	10	5	15	1
11a.	0	0	10	5	15	1
11b.	0	0	10	5	15	1
12	0	1	7	7	14	.87
Total						.85

Fuente: elaboración propia.

La tabla 21 reporta los resultados obtenidos en la validación de contenido del postest de geometría (G2). Se observa que, también en este caso, los valores del índice son bastante altos y, como se hizo en el pretest, se decidió desechar el elemento 8 con el índice más bajo igual a .6. Los elementos 2, 3, 6, 7, 9, 13a y 13 b obtuvieron una validez de contenido perfecta y los demás oscilan entre .73 (elemento 10 y 12) y .87 (elemento 1, 4, 5,11). El índice de

validez de contenido global para la prueba inicial es de .92 y permite confirmar la validez de contenido de la prueba.

Tabla 21

Índice de validez de contenido de los elementos del posttest de geometría (G2)

Elemento	No relevante	Poco relevante	Relevante	Muy relevante	R+MR	CVI
1.	0	1	6	8	14	.87
2.	0	0	5	10	15	1
3.	0	0	3	12	15	1
4.	0	1	4	10	14	.87
5.	0	1	7	7	14	.87
6.	0	0	4	11	15	1
7.	0	0	6	9	15	1
8.	0	3	9	3	12	.60
9.	0	0	7	8	15	1
10	0	2	7	6	13	.73
11.	1	0	8	6	14	.87
12.	0	2	3	10	13	.73
13a.	0	0	9	6	15	1
13b.	0	0	5	10	15	1
Total						.92

Fuente: elaboración propia.

A partir de los resultados del análisis realizado se procedió a definir las pruebas de geometría en su versión definitiva.

Con referencia a las dos pruebas de geometría también se realizó el cálculo del nivel de confianza estadística, es decir, la reproducibilidad de los resultados en el tiempo, en las mismas condiciones.

Para dar una interpretación se utilizó como indicador estadístico el coeficiente de Cronbach:

$$\alpha = \frac{k}{k-1} \left(1 - \frac{\sum_{i=1}^k \sigma_{Y_i}^2}{\sigma_X^2} \right)$$

Donde k representa el número total de los elementos de la prueba, $\sigma_{Y_i}^2$ la varianza de cada elemento y σ_X^2 la varianza total de la prueba.

Los resultados se muestran en la tabla 21 donde el nivel de confianza es de .66 para la prueba inicial y de .50 para la prueba final.

Tabla 22

Nivel de confianza en de las pruebas de geometría (pre y postest)

Instrumento	Fuente	Alfa de Cronbach
Prueba inicial	Invalsi, 2001-2018	.66
Prueba final	Invalsi, 2001-2018	.50

Los resultados parecen condicionados por el número reducido de elementos que componen las dos pruebas.

4.2. Análisis cuantitativo

4.2.1. Resultados de las pruebas de geometría

Para obtener una evaluación significativa del desempeño de los estudiantes que realizaron ambas pruebas, se realizó un análisis cuantitativo aplicando algunas pruebas estadísticas con el software SPSS, versión 22.

A través de la aplicación de las pruebas de geometría denominadas con las iniciales G1 (prueba inicial) y G2 (prueba final) se buscó analizar los resultados de aprendizaje conseguidos por los alumnos de ambas clases de cuarto curso que participaron en las actividades preparadas por las docentes con relación a la descripción y clasificación de las figuras geométricas y al análisis de sus propiedades.

Las pruebas se realizaron a 21 estudiantes del grupo control, de los cuales 61,9% de sexo masculino y 38,1% de sexo femenino y a 25 niños del grupo experimental, en el que la brecha entre el sexo masculino (48%) y el sexo femenino (52%) era menor. Los datos se muestran en la tabla 23.

Tabla 23

Descripción de las características de los grupos por género

Grupo	Género	n	%
Experimental	Masculino	12	48.0
	Femenino	13	52.0
Control	Masculino	13	61.9
	Femenino	8	38.1

Los datos reportados en el software SPSS para ser elaborados, corresponden al total de respuestas exactas proporcionadas por los alumnos de cada grupo en ambas pruebas de geometría tanto la inicial, llevada a cabo antes de la intervención didáctica, como la final, llevada a cabo después de dicha intervención.

La idea de base es muy clara: una diferencia significativa de las respuestas correctas dadas por los alumnos, en la prueba final aplicada tanto al grupo experimental como al grupo control, puede poner en evidencia la eficacia del método de aprendizaje *coding*. Los cálculos corresponden a los resultados de las pruebas inicial y final. En la Tabla 24 se muestran las puntuaciones mínimas y máximas que se pueden conseguir de ambas pruebas.

Tabla 24

Puntuaciones mínimas y máximas de las pruebas de geometría

Sub escala	N. elementos	Puntuación mínima	Puntuación máxima
Prueba inicial	14	0	14
Prueba final	18	0	18

Fuente: elaboración propia.

Como primer paso, se realizó una prueba de normalidad para establecer si los datos relativos a las respuestas correctas dadas por cada sujeto de ambos grupos, en la prueba inicial y en la final, podían considerarse extraídos de una distribución normal.

En particular, para la verificación de la normalidad, se hizo referencia a la prueba de Shapiro-Wilk, considerada en la literatura una prueba potente para muestras pequeñas.

La distribución del desempeño con respecto a la prueba inicial del grupo control, con un promedio $M = 12.14$ ($DS = 1.93$) (Tabla 25), muestra valores de *p-value* más altos con respecto al nivel de significancia ($t(21) = .921$, $p > .05$) al que se hace referencia, esto no es válido para el grupo experimental por lo que, en base a la lectura del valor p , la hipótesis nula no es aceptable ($t(25) = .829$, $p < .005$).

Tabla 25

Promedio (M) y desviaciones estándar (DS) de las puntuaciones conseguidas en pre y posttest en ambos grupos de estudiantes

Grupo	N	Puntuación prueba inicial		Puntuación prueba final	
		M	DS	M	DS
Control	21	12.14	1.93	11.67	2.42
Experimental	25	12.64	2.16	13.40	2.16

Fuente: elaboración propia.

En cuanto a la prueba final, a partir de los resultados obtenidos, se observa que las frecuencias de las respuestas correctas se distribuyen normalmente con valores de p mayores respecto al nivel de significancia tanto en el grupo de control ($t(21) = .933$, $p > .05$) como en el grupo experimental ($t(25) = .932$, $p > .05$).

Analizando los datos reportados en la Tabla 26 se puede afirmar que han sido extraídos de una variable aleatoria normal (distribución normal) y que los resultados obtenidos para el grupo experimental en la prueba inicial se deben

atribuir al caso para lo cual, no obstante se haya registrado un p más bajo del nivel designificación ($p < .005$), se puede considerar válida la hipótesis de la normalidad de la distribución de los errores en ambas pruebas G1 y G2 y para ambos grupos de estudiantes.

Tabla 26*Prueba de normalidad aplicada a pre y postest*

Shapiro Wilk test				
<i>Grupo</i>		<i>n</i>	<i>T</i>	<i>p</i>
Prueba inicial	Experimental	25	.829	.001
	Control	21	.921	.090
Prueba final	Experimental	25	.932	.096
	Control	21	.933	.157

Fuente: elaboración propia.

Dado que los datos del pretest y del postest se distribuyeron normalmente, se aplicaron pruebas paramétricas, como el t-Test y el test de Levene, con el fin de verificar la presencia de diferencias significativas en los resultados de las pruebas entre el grupo control y el grupo experimental. Así pues, se compararon las puntuaciones de los dos grupos antes del tratamiento y después, con el fin de verificar si el uso del *coding* afectó de alguna manera los resultados finales obtenidos por los estudiantes del grupo experimental o, por el contrario, no provocó cambios significativos en el aprendizaje.

Se realizaron, en particular, dos test t de Student a muestras independientes para comparar las puntuaciones conseguidas por los estudiantes del grupo control con los del grupo experimental antes y después de la intervención. En cuanto al pretest, no se detectó una diferencia estadísticamente significativa entre las puntuaciones del grupo control (M= 12.14, DS=1.93) y del grupo experimental (M=12.64, DS=2.46), $t(44) = 1.066$, $p = .292$). El efecto de la diferencia en promedio (diferencia promedio = .19, 95% IC: -.168 a .545) es bajo ($R^2 = .013$). La falta de significación estadística en la prueba inicial, que se aplicó a ambos grupos antes de llevar a cabo el tratamiento, pone en evidencia la homogeneidad en el aprendizaje de los niños

de ambas clases, independientemente del método *coding*.

En cuanto al postest, a partir del t-Test se observa una diferencia estadísticamente significativa entre las puntuaciones del grupo de control (M=11.67, DS=2.42) y del grupo experimental (M=13.40, DS=2.16, $t(44)=-2.569$, $p=.014$) El efecto de la diferencia en promedio (diferencia promedio= -1.73, 95% IC: -3.09 a -.373) es bastante alto ($R^2= .13$) (Tabla 27).

Tabla 27

Comparación de los resultados obtenidos en el pretest y postest entre los grupos

	t-Test muestras independientes		
	T	P	η^2
Puntuación pretest	.751	.457	.013
Puntuación postest	2.569	.014	.13

Fuente: elaboración propia.

En cuanto al postest, es posible verificar que el promedio de las respuestas dadas por ambos grupos presenta valores significativamente distintos entre ellos con respecto a las respuestas dadas antes de la aplicación del tratamiento (Tabla 27).

Para el postest, se decidió aplicar una prueba estadística adicional (t-Test) después de haber subdividido la prueba de geometría en áreas, en las que se agruparon los elementos con referencia a una determinada dimensión común; el objeto fue identificar las áreas de conocimiento dentro de las que surgen las eventuales diferencias entre el grupo experimental y el de control con referencia a un dominio específico. Como se menciona en el Capítulo 3 (par. 3.4.1.3) dentro de la prueba final G2 se identificaron tres áreas distintas:

1. reconocer y representar formas del plano y del espacio, relaciones y estructuras que se encuentran en la naturaleza o que han sido creadas por el hombre (n.7 elementos);

2. describir, denominar y clasificar figuras en base a características geométricas, determinar las medidas, diseñar y construir modelos concretos de varios tipos (n.4 elementos);
3. conocer y dominar los contenidos específicos de las matemáticas como objetos matemáticos, propiedades y estructuras (n.3 elementos);

Para cada una de estas áreas se realizó el t-Test que demostró que no hay una diferencia estadísticamente significativa en cuanto a los resultados conseguidos por ambos grupos con referencia a las áreas investigadas, excepto para el grupo de elementos de la dimensión 1 (Tabla 28).

Tabla 28

Comparación entre grupos de la prueba final divididos por dimensión

	t-Test muestras independientes		
	<i>T</i>	<i>P</i>	η^2
Puntuación para dimensión 1	-2.38	.021	.11
Puntuación para dimensión 2	-.210	.833	.00
Puntuación para dimensión 3	-1.12	.270	.03

Fuente: elaboración propia.

El test se repitió considerando la puntuación obtenida para cada elemento por los niños del grupo experimental y del grupo control. Ambos grupos se sometieron a una comparación adicional a través del test t de Student, en el que se tomaron en cuenta los elementos individuales de la prueba final.

Se observa que todos los niños del grupo control respondieron correctamente a la pregunta 2, todos los niños del grupo experimental respondieron correctamente a la pregunta 12a1. Para los elementos 3, 5, 12a1, 12a2, 12b se registraron diferencias estadísticamente significativas entre los dos grupos con valores de p que entran en la zona de no aceptación de la hipótesis nula (Anexo A) y con valores de η^2 relativamente altos; en este caso se confirma que, aun siendo un número limitado de preguntas, existe una

diferencia entre las respuestas dadas por los dos grupos, en beneficio del grupo experimental.

A partir del análisis realizado, se considera que la aplicación de la metodología *coding* no produjo grandes efectos sobre la mejora en los resultados del aprendizaje en los niños que formaron parte del grupo experimental en comparación con aquellos que trabajaron sin el *coding*, sin embargo, el tratamiento tuvo efectos positivos. Habiendo abordado, en el capítulo 2, el tema de la diferencia de género en la enseñanza de las disciplinas del área STEM, se buscó tener en cuenta también este aspecto; para este fin, se integró a la investigación el estudio y análisis de las posibles diferencias en cada grupo comparando la puntuación obtenida por los niños y aquella obtenida por las niñas. Las estadísticas descriptivas presentadas en la Figura 6 y 7 muestran que los niños proporcionaron más respuestas correctas con respecto a las niñas para cada una de las pruebas y en ambas clases, sin embargo, la diferencia resulta realmente mínima.

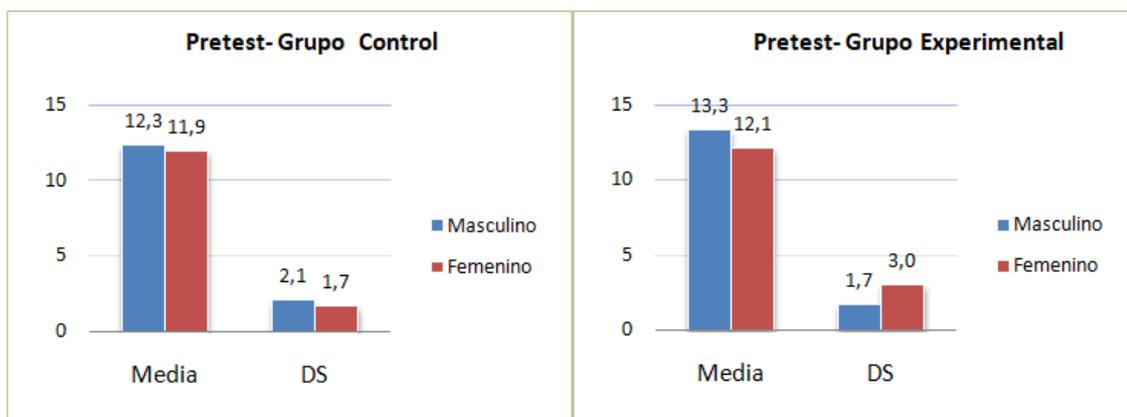


Figura 6. Histogramas de promedio y desviación estándar en el pretest

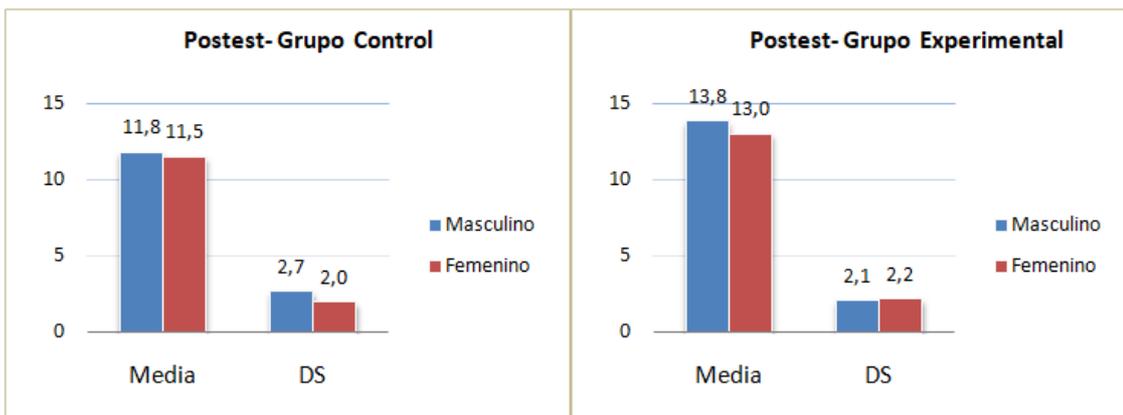


Figura 7. Histogramas de promedio y desviación estándar en el posttest

Como se observa en la tabla 29, para el grupo experimental, no se encontró una diferencia estadísticamente significativa entre las puntuaciones de los niños ($M= 13.83$, $DS=2.13$) y de las niñas ($M=13.00$, $DS=2.20$), $t(23)=.962$, $p=.346$) y el efecto de la diferencia en promedio es bajo ($R^2= .04$). Con respecto al grupo control, tampoco se encuentra una diferencia estadísticamente significativa entre las puntuaciones de los niños ($M= 11.77$, $DS=2.71$) y de las niñas ($M=11.50$, $DS=2.00$, $t(19)=.242$, $p=.811$, a dos colas) y el efecto de la diferencia promedio es bajo ($R^2= .003$).

Tabla 29

Test t de Student por los niños y las niñas en ambos grupos

Grupo	T-test de Student		
	T	P	η^2
Control	.242	.811	.003
Experimental	.962	.346	.04

Fuente: elaboración propia.

Es importante mencionar que las pruebas estadísticas aplicadas permitieron obtener resultados que no pueden extenderse a la totalidad de la población de sujetos con las características de aquellos que participaron en la experimentación, principalmente debido a la elección no aleatoria de la muestra y al reducido número de participantes. Sin embargo, en este caso en particular, es posible hacer algunas consideraciones que se describirán en el capítulo

siguiente.

4.2.2. Cuestionario AEQ-ES

A través del cuestionario *AEQ-ES*, aplicado al final de las actividades realizadas, se pretendió evaluar las emociones de logro de los alumnos con referencia a las fases del aprendizaje de la geometría. El objetivo fue medir, en particular, las emociones relacionadas a la actividad realizada y a los resultados obtenidos en tres distintos momentos del proceso de aprendizaje, durante las lecciones, durante las tareas en casa y durante las evaluaciones en clase.

Antes de efectuar los análisis estadísticos necesarios para dar respuesta a las preguntas de la investigación, se realizó un análisis preliminar para cada categoría en las que se puede dividir el cuestionario. El fin de esta operación fue identificar las principales dimensiones que lo constituyen y calcular su nivel de confianza; también se pretendió, después de haber identificado las variables medidas por las categorías individuales, explorar la distribución de los datos recogidos para definir cuál es mejor técnica estadística para aplicar.

A continuación, se describen los procedimientos de análisis preliminar que se realizaron para el cuestionario.

Los treinta y ocho elementos que forman parte del cuestionario se dividieron en 3 escalas separadas, cada una compuesta por los elementos que se refieren a la misma categoría distinguiéndose según la situación de aprendizaje (lecciones, tarea en casa, evaluación). Para las 3 escalas se calcularon la suma de las puntuaciones de los elementos correspondientes a cada escala y el nivel de confianza alfa de Cronbach; de este modo fue posible identificar, de forma más clara, las tres variables continuas de interés.

4.2.2.1. Análisis preliminar cuestionario AEQ-ES

Los participantes respondieron al cuestionario *AEQ-ES* en versión italiana

que investiga tres tipos de emociones que pueden a su vez dividirse en tres subescalas (a) alegría/diversión, (b) ansiedad, (c) aburrimiento, evaluando todos los elementos que lo constituyen según la escala de Likert que va desde la puntuación 1 (en absoluto) hasta 5 (muchísimo). Se calcularon los valores de las estadísticas descriptivas y de las frecuencias con referencia a cada elemento, calculando la puntuación total para cada grupo haciendo referencia a tres distintas situaciones de aprendizaje (lecciones, tareas, evaluaciones) y para toda la escala.

No fue necesario recodificar el cuestionario ya que no hay elementos formulados negativamente, por lo que se calcularon directamente los valores del nivel de confianza interna de toda la escala en ambos grupos. Se registraron valores de alfa de Cronbach que muestran un buen grado de confianza interna (Tabla 30) tanto con referencia al grupo control como al grupo experimental.

Tabla 30

Nivel de confianza dentro del cuestionario AEQ-ES

	<i>N</i>	<i>Alfa de Cronbach</i>
Grupo di control	32	.775
Grupo experimental	32	.812

Fuente: elaboración propia.

Puede decirse lo mismo del cálculo del coeficiente alfa de Cronbach para cada subescala como se reporta en la Tabla 31.

Tabla 31

Nivel de confianza dentro de las tres dimensiones del cuestionario AEQ-ES

<i>Sub Escala</i>	<i>N. elementos</i>	<i>Grupo de control</i>	<i>Grupo experimental</i>
Alegría-diversión	9	.88	.79
Ansiedad	12	.91	.89
Aburrimiento	11	.93	.90

Fuente: elaboración propia.

En cambio, las tablas 33 y 34 reportan los valores de promedio y desviación estándar que se refieren a cada elemento y las frecuencias de respuesta en función de la escala de Likert del cuestionario.

Sumando las puntuaciones de los elementos 1, 4, 8, 12, 1(a), 6(a), 1(b), 5(b), 12(b), de la subescala *alegría/diversión* se calculó el total que representa la variable global que indica las percepciones de emociones como el placer, la diversión, la alegría, que se experimentaron durante las fases del proceso de aprendizaje de las actividades de geometría.

De igual forma, se calculó el total de la puntuación relacionada a la subescala *ansiedad* sumando las puntuaciones de los elementos 3, 5, 7, 10, 3(a), 5(a), 7(a), 2(b), 4(b), 6(b), 9(b), 11(b) con el fin de medir la variable que indica las emociones de valencia negativa, como el miedo, tristeza y ansiedad que se experimentan durante el desarrollo de las actividades escolares.

Finalmente de la suma de las puntuaciones de los elementos 2, 6, 9, 11, 2(a), 4(a), 8(a), 3(b), 7(b), 8(b), 10(b) se pudo definir la variable que indica la percepción de los estudiantes con respecto a emociones relacionadas a la subescala *aburrimiento*.

Con referencia al factor correspondiente a las emociones de valencia positiva (subescala *alegría/diversión*) la puntuación total puede oscilar entre 9 y 45 (Tabla 24) a la cual corresponde un valor central de 27; la puntuación promedio total correspondiente fue 33.48 (DS= 7.81) para el grupo control y 29.52 (DS= 7.46) para el grupo experimental, valores que indican una buena activación de emociones positivas en clase para ambos grupos.

La puntuación total promedio con respecto al factor *ansiedad* fue 19.52 (DS= 7.27) para el grupo control y 19.76 (DS= 8.09) para el grupo experimental; teniendo en cuenta que dicha puntuación puede oscilar entre 12 y 60 (Tabla 24), con un valor promedio de 36, los resultados en ambos casos

muestran una actitud relativamente baja en cuanto a la activación de emociones relacionadas con la ansiedad.

Lo mismo se puede decir para el grupo de elementos relacionados con el factor aburrimiento, para los que los estudiantes mostraron en complejo una baja percepción de dicha emoción, teniendo en cuenta que la puntuación puede oscilar entre un mínimo de 11 y un máximo de 55 (Tabla 32), con un valor promedio de la escala de 33, los resultados muestran, para ambas clases, valores promedio respectivamente de 17.67 ($DS= 9.28$) y de 20.16 ($DS= 8.73$) que muestran una medida de niveles bajos de emociones relacionadas con el aburrimiento.

Tabla 32

Puntuaciones mínimas y máximas relacionadas a cada subescala (cuestionario AEQ_ES)

Subescala	N. elemento	Puntuación mínima	Puntuación máxima
Alegría/Diversión	9	9	45
Ansiedad	12	12	60
Aburrimiento	11	11	55

Fuente: elaboración propia.

Analizando las respuestas de los participantes para cada elementorelacionado con el factor alegría/diversión, se registra un grado de acuerdo mayor en el grupo de control con respecto al elemento 8 (*La geometría es divertida para mí*) con un promedio de 4.00 ($DS= 1.38$) y en el grupo experimental en el elemento 1 (*Me divierto en las lecciones de geometría*) a la que corresponde un valor promedio de 4.00 ($DS= 1.12$).

A partir de las respuestas dadas a los elementos del grupo ansiedad, el grupo control mostró mayor grado de desacuerdo para el elemento 3 (*La geometría me asusta*) a la que corresponde un valor promedio de 1.19 ($DS= .68$), para el elemento 5 (*Cuando pienso en las lecciones de geometría, me pongo nervioso*) ($M= 1.52 DS= .75$) y el elemento 3(a) (*Me preocupo tanto que no logro terminar las tareas para casa de geometría y comienzo a sudar*) a la

que corresponde un valor promedio de 1.24 ($DS= .54$). El grupo experimental muestra un mayor desacuerdo en el elemento 11(b) (*Me pongo nervioso durante las evaluaciones de geometría*) con un promedio de 1.64 ($DS= .91$).

En cuanto a las respuestas relacionadas al factor *aburrimiento* el grupo control mostró un mayor desacuerdo en la pregunta 2 (*Las lecciones de geometría me aburren*) para la cual se registra un valor promedio de 1.48 ($DS= .75$) y en el elemento 6 (*La geometría me aburre*) para la cual el promedio es 1.38 ($DS= .74$). Los niños del grupo experimental han mostrado el mayor grado de desacuerdo en el elemento 2 (*Las lecciones de geometría me aburren*) ($M= 1.56 DS = .58$).

Como se muestra en la tabla 33, la mayor parte de los niños (57.1%) sostiene que la geometría es divertida (elemento 8) y el 95% de ellos no se asusta en absoluto con esta disciplina en la cual no se experimentan emociones negativas excesivas (elemento 3). Se registran porcentajes más bajos con respecto al aburrimiento durante lecciones, tareas y evaluaciones: el 76.2% de los participantes sostiene que la geometría no es aburrida (elemento 2).

Tabla 33
Estadística descriptiva grupo control (cuestionario AEQ-ES)

	M	DS	Para nada		Poco		Suficiente		Mucho		Muchísimo	
			f	%	f	%	f	%	f	%	f	%
Sub escala												
Alegría-Diversión												
<i>Lecciones</i>												
1. Me divierto en las lecciones de geometría.	3.67	1.11	1	4.8	1	4.8	8	38.1	5	23.8	6	28.6
4 no puedo esperar para las lecciones de geometría.	3.76	1.14	1	4.8	1	4.8	7	33.3	5	23.8	7	33.3
8. La geometría es divertida para mí.	4.00	1.38	2	9.5	1	4.8	4	19.0	2	9.5	12	57.1
12. Me divierto estudiando geometría.	3.81	1.40	3	14.3	1	4.8	1	4.8	8	38.1	8	38.1
<i>Tareas</i>												
1(a). Cuando hago las tareas de geometría en casa, estoy contento.	3.57	1.40	2	9.5	3	14.3	5	23.8	3	14.3	8	38.1
6(a). Me divierten tanto las tareas en casa de geometría que no quisiera terminar jamás.	3.33	1.43	3	14.3	3	14.3	5	23.8	4	19.0	6	28.6
<i>Evaluaciones</i>												
1(b). Me divierto en las evaluaciones de geometría.	3.86	1.10	1	4.8	1	4.8	5	23.8	7	33.3	7	33.3
5(b). Non puedo esperar para hacer las evaluaciones de geometría.	3.62	1.20	2	9.5	1	4.8	5	23.8	8	38.1	5	23.8
12(b). Durante las evaluaciones de geometría, pienso "¡Voy bien!".	3.67	1.16	-	-	4	19.0	6	28.6	4	19.0	7	33.3

Tabla 33
Estadística descriptiva grupo control (cuestionario AEQ-ES)

	M	DS	Para nada		Poco		Suficiente		Mucho		Muchísimo	
			f	%	f	%	f	%	f	%	f	%
Sub escala												
Ansiedad												
<i>Lecciones</i>												
3. La geometría me asusta.	1.19	.68	19	95	1	4.8	-	-	1	4.8	-	-
5. Cuando pienso a las lecciones de geometría, me pongo nervioso.	1.52	.75	12	57.1	8	38.1	-	-	1	4.8	-	-
7. Cuando pienso en las lecciones de geometría, me duele de estómago.	1.24	.63	18	85.7	1	4.8	2	9.5	-	-	-	-
10. Durante las lecciones de geometría me preocupo, pues me parece todo demasiado difícil.	2.00	1.05	7	33.3	10	47.6	2	9.5	1	4.8	1	4.8
<i>Tareas</i>												
3(a). Me preocupo tanto de no lograr terminar las tareas de geometría para la casa que comienzo a sudar.	1.24	.54	17	81.0	3	14.3	1	4.8	-	-	-	-
5(a). Cuando hago las tareas para la casa de geometría, me preocupo porque me pregunto si podré entenderlas.	1.29	.64	7	33.3	10	47.6	2	9.5	1	4.8	1	4.8
7(a). Las tareas de geometría para la casa me asustan tanto que no quisiera iniciar a hacerlas.	3.86	1.11	17	81.0	2	9.5	2	9.5	-	-	-	-
<i>Evaluaciones</i>												
2(b). Cuando hago las evaluaciones de geometría, tengo miedo de obtener una mala calificación.	1.67	1.11	3	14.3	8	38.1	4	19.0	3	14.3	3	14.3
4(b). Estoy tan nervioso durante las evaluaciones de geometría que no logro recordar bien lo que he aprendido.	2.05	1.28	8	38.1	5	23.8	7	33.3	-	-	1	4.8
6(b). Las evaluaciones de geometría me asustan tanto que preferiría no hacerlas.	1.62	1.12	15	71.4	3	14.3	1	4.8	1	4.8	1	4.8

Tabla 33
Estadística descriptiva grupo control (cuestionario AEQ-ES)

	<i>M</i>	<i>DS</i>	Para nada		Poco		Suficiente		Mucho		Muchísimo	
			<i>f</i>	%	<i>f</i>	%	<i>f</i>	%	<i>f</i>	%	<i>f</i>	%
9(b). Estoy tan nervioso durante las evaluaciones de geometría que no logro concentrarme bien.	2.05	1.28	9	42.9	7	33.3	2	9.5	1	4.8	2	9.5
11(b). Me pongo nervioso durante las evaluaciones de geometría.	1.86	1.28	11	52.4	7	33.3	-	-	1	4.8	2	9.5
Sub escala												
Aburrimiento												
<i>Lecciones</i>												
2. Las lecciones de geometría me aburren.	1.48	.75	13	61.9	7	33.3	-	-	1	4.8	2	9.5
6. La geometría me aburre.	1.38	.74	15	71.4	5	23.8	-	-	1	4.8	-	-
9. Para mí estudiar geometría es aburrido.	1.48	1.03	16	76.2	2	9.5	2	9.5	-	-	1	4.8
11. Para mí las lecciones de geometría son tan aburridas que preferiría hacer otra cosa.	1.62	1.25	15	71.4	3	14.3	1	4.8	-	-	2	9.5
<i>Tareas</i>												
2(a). Las tareas de geometría para la casa me aburren demasiado.	1.81	1.25	13	61.9	3	14.3	2	9.5	2	9.5	1	4.8
4(a). Las tareas de geometría para la casa me aburren tanto que quisiera no hacerlas.	1.57	1.12	15	71.4	3	14.3	1	4.8	2	9.5	1	4.8
8(a). Cuando hago las tareas de geometría en casa, me canso rápidamente porque me aburro.	1.71	1.01	12	57.1	5	23.8	2	9.5	2	9.5	-	-
<i>Evaluaciones</i>												
3(b). Las evaluaciones de geometría me aburren.	1.57	1.17	16	76.2	1	4.8	2	9.5	1	4.8	1	4.8

Tabla 33
Estadística descriptiva grupo control (cuestionario AEQ-ES)

	<i>M</i>	<i>DS</i>	Para nada		Poco		Suficiente		Mucho		Muchísimo	
			<i>f</i>	%	<i>f</i>	%	<i>f</i>	%	<i>f</i>	%	<i>f</i>	%
7(b). Para mi hacer las evaluaciones de geometría son aburridas.	1.76	1.34	14	66.7	3	14.3	1	4.8	1	4.8	2	9.5
8(b). Las evaluaciones de geometría me aburren tanto que quisiera no hacerlas.	1.62	1.12	14	66.7	4	19	1	4.8	1	4.8	1	4.8
10(b). Cuando hago las evaluaciones de geometría me canso rápidamente porque me aburro.	1.67	1.11	13	61.9	5	23.8	1	4.8	1	4.8	1	4.8

Fuente: elaboración propia.

La tabla 34 muestra los resultados respecto a las emociones de logro que experimentan los alumnos del grupo experimental. Emociones como alegría y diversión registran el porcentaje más alto en los elementos 1 y 14 con el 40% de respuestas con alto grado de acuerdo (*muchísimo*), el 92% de las respuestas muestra que los niños no tienen miedo de la geometría (elemento 3) mientras el 72% de ellos se declaran aburridos al realizar las evaluaciones de geometría (elemento 8(b)).

Tabla 34

Estadística descriptiva grupo experimental (cuestionario AEQ-ES)

	<i>M</i>	<i>DS</i>	<i>Para nada</i>		<i>Poco</i>		<i>Suficiente</i>		<i>Mucho</i>		<i>Muchísimo</i>	
			<i>f</i>	<i>%</i>	<i>f</i>	<i>%</i>	<i>f</i>	<i>%</i>	<i>f</i>	<i>%</i>	<i>f</i>	<i>%</i>
Alegría-Diversión												
<i>Lecciones</i>												
1. Me divierto en las lecciones de geometría.	4.00	1.12	1	48.0	2	8.0	3	12.0	9	36.0	10	40.0
4 .No puedo esperar para las lecciones de geometría.	3.56	1.39	3	12.0	3	12.0	4	16.0	7	28.0	8	32.0
8. La geometría es divertida para mí.	3.40	1.50	4	16.0	3	12.0	6	24.0	3	12.0	9	36.0
12. Me divierto estudiando geometría.	3.76	1.30	2	8.0	2	8.0	6	24.0	5	20.0	10	40.0
<i>Tareas</i>												
1(a). Cuando hago las tareas de geometría para la casa, estoy contento.	2.64	1.32	6	24.0	6	24.0	7	28.0	3	12.0	3	12.0
6(a). Me divierten tanto las tareas en casa de geometría que no quisiera terminar jamás.	2.64	1.35	6	24.0	7	28.0	5	20.0	4	16.0	3	12.0
<i>Evaluaciones</i>												
1(b). Me divierto en las evaluaciones de geometría.	2.88	1.45	6	24.0	4	16.0	7	28.0	3	12.0	5	20.0
5(b). No puedo esperar para hacer las evaluaciones de geometría.	3.36	1.41	4	16.0	2	8.0	7	28.0	5	20.0	7	28.0
12(b). Durante las evaluaciones de geometría, pienso "¡Voy bien!".	3.28	1.34	2	8.0	6	24.0	7	28.0	3	12.0	7	28.0
Ansiedad												

Tabla 34

Estadística descriptiva grupo experimental (cuestionario AEQ-ES)

	<i>M</i>	<i>DS</i>	<i>Para nada</i>		<i>Poco</i>		<i>Suficiente</i>		<i>Mucho</i>		<i>Muchísimo</i>	
			<i>f</i>	<i>%</i>	<i>f</i>	<i>%</i>	<i>f</i>	<i>%</i>	<i>f</i>	<i>%</i>	<i>f</i>	<i>%</i>
<i>Lecciones</i>												
3. La geometría me asusta.	1.20	.82	23	92.0	1	4.0	-	-	-	-	1	4.0
5. Cuando pienso en las lecciones de geometría, me pongo nervioso.	1.36	.64	18	72.0	5	20.0	2	8.0	-	-	-	-
7. Cuando pienso en las lecciones de geometría, me duele el estómago.	1.36	.70	19	76.0	3	12.0	3	12.0	-	-	-	-
10. Durante las lecciones de geometría me preocupó, pues me parece todo demasiado difícil.	1.48	.59	14	56.0	10	40.0	1	4.0	-	-	-	-
<i>Tareas</i>												
3(a). Me preocupó tanto de no lograr terminar las tareas de geometría para la casa que comienzo a sudar.	1.48	1.05	19	76.0	3	12.0	1	4.0	1	4.0	1	4.0
5(a). Cuando hago las tareas para la casa de geometría, me preocupó porque me pregunto si podré entenderlas.	1.76	1.09	14	56.0	6	24.0	3	12.0	1	4.0	1	4.0
7(a). Las tareas de geometría para la casa me asustan tanto que no quisiera iniciar a hacerlas.	1.60	1.32	19	76.0	3	12.0	-	-	-	-	3	12.0
<i>Evaluaciones</i>												
2(b). Cuando hago las evaluaciones de geometría, tengo miedo de obtener una mala calificación.	2.64	1.38	6	24.0	8	32.0	3	12.0	1	4.0	3	12.0
4(b). Estoy tan nervioso durante las evaluaciones de geometría que no logro recordar bien lo que he aprendido.	1.88	1.09	10	40.0	12	48.0	1	4.0	-	-	2	8.0
6(b). Las evaluaciones de geometría me asustan	1.60	1.08	17	68.0	4	16.0	2	8.0	1	4.0	1	4.0

Tabla 34

Estadística descriptiva grupo experimental (cuestionario AEQ-ES)

	<i>M</i>	<i>DS</i>	<i>Para nada</i>		<i>Poco</i>		<i>Suficiente</i>		<i>Mucho</i>		<i>Muchísimo</i>	
			<i>f</i>	<i>%</i>	<i>f</i>	<i>%</i>	<i>f</i>	<i>%</i>	<i>f</i>	<i>%</i>	<i>f</i>	<i>%</i>
tanto que preferiría no hacerlas.												
9(b). Estoy tan nervioso durante las evaluaciones de geometría que no logro concentrarme bien.	1.76	.93	11	44.0	11	44.0	2	8.0	-	-	1	4.0
11(b). Me pongo nervioso durante las evaluaciones de geometría.	1.64	.91	15	60.0	5	20.0	4	16.0	1	4.0	-	-
Aburrimiento												
<i>Lecciones</i>												
2. Las lecciones de geometría me aburren.	1.56	.58	12	48.0	12	48.0	1	4.0	-	-	-	-
6. La geometría me aburre.	1.68	.85	13	52.0	8	32.0	3	12.0	1	4.0	-	-
9. Para mí estudiar geometría es aburrido.	1.68	.85	13	52.0	8	32.0	3	12.0	1	4.0	-	-
11. Para mí las lecciones de geometría son tan aburridas que preferiría hacer otra cosa.	2.20	1.38	11	44.0	5	20.0	5	20.0	1	4.0	3	12.0
<i>Tareas</i>												
2(a). Las tareas de geometría para la casa me aburren demasiado.	2.24	1.48	11	44.0	6	24.0	3	12.0	1	4.0	4	16.0
4(a). Las tareas para la casa de geometría me aburren tanto que preferiría no hacerlas.	2.16	1.55	13	52.0	5	20.0	1	4.0	2	8.0	4	16.0
8(a). Cuando hago las tareas de geometría en casa, me canso rápidamente porque me aburro.	1.88	1.05	11	44.0	10	40.0	-	-	4	16.0	-	-

Tabla 34

Estadística descriptiva grupo experimental (cuestionario AEQ-ES)

	<i>M</i>	<i>DS</i>	<i>Para nada</i>		<i>Poco</i>		<i>Suficiente</i>		<i>Mucho</i>		<i>Muchísimo</i>	
			<i>f</i>	<i>%</i>	<i>f</i>	<i>%</i>	<i>f</i>	<i>%</i>	<i>f</i>	<i>%</i>	<i>f</i>	<i>%</i>
<i>Evaluaciones</i>												
3(b). Las evaluaciones de geometría me aburren.	1.64	.95	14	56.0	8	32.0	2	8.0	-	-	1	4.0
7(b). Para mi hacer las evaluaciones de geometría son aburridas.	1.92	1.22	13	52.0	6	24.0	2	8.0	3	12.0	1	4.0
8(b). Las evaluaciones de geometría me aburren tanto que quisiera no hacerlas.	1.56	1.04	18	72.0	2	8.0	4	16.0	-	-	1	4.0
10(b). Cuando hago las evaluaciones de geometría me canso rápidamente porque me aburro.	1.64	1.08	16	64.0	5	20.0	2	8.0	1	4.0	1	4.0

Fuente: elaboración propia.

Las variables descritas anteriormente se sometieron al control de la normalidad de la distribución con el fin de verificar que se respetaran los parámetros para poder aplicar las técnicas estadísticas paramétricas. La distribución de los datos no resultó normal para todos los elementos tanto en el grupo control como en el grupo experimental ($t(46) = .862$, $p < .001$); habiendo registrado valores p más bajos que el nivel de significancia, se puede concluir que, las variables no están distribuidas de forma normal.

Por dicho motivo se decidió utilizar técnicas estadísticas no paramétricas que permitieran trabajar con los datos disponibles e igualmente condujeran a conclusiones válidas.

Con el fin de verificar la presencia de diferencias significativas en las respuestas dadas al cuestionario por los estudiantes del grupo control y del grupo experimental, se realizó la prueba no paramétrica U de Mann-Whitney, que puede aplicarse en situaciones como esta en las que, teniendo una muestra pequeña ($n=46$), se comprobó que la distribución de las variables en cuestión no es normal.

Primero se hizo una comparación entre los grupos con respecto a la puntuación total registrada por cada alumno.

Con referencia a la puntuación total obtenida por los estudiantes en el cuestionario, la prueba U de Mann-Whitney no muestra alguna diferencia estadísticamente significativa en las puntuaciones del grupo control ($Md=69$, $n=21$) y del grupo experimental ($Md=66$, $n=25$, $U=225$, $z=-.828$, $p=.408$, $r=.12$).

Posteriormente se compararon, para cada grupo de elementos, relacionados respectivamente con las tres subescalas alegría/diversión, ansiedad y aburrimiento, el total de las puntuaciones de los dos grupos, con el fin de verificar, específicamente, si existían diferencias significativas en la

experimentación de dichas emociones, entre el grupo de niños sometidos al tratamiento con el *coding* y el grupo que trabajo siguiendo el método tradicional.

En cuanto a las subescalas identificadas en el cuestionario la prueba U de Mann-Whitney no muestra alguna diferencia estadísticamente significativa en las puntuaciones obtenidas por los estudiantes del grupo de control (n=21) y del grupo experimental (n=25), cuyos resultados se pueden leer en la tabla 35.

Tabla 35

Prueba estadística para la comparación entre los grupos en el cuestionario divididos por subescala

Sub escala	Prueba U Mann-Whitney			
	<i>N</i>	<i>Z</i>	<i>p</i>	<i>r</i>
Alegría/Diversión ^a	46	-1.867	.06	.28
Ansiedad ^b	46	-.232	.82	.03
Aburrimiento ^c	46	-1.504	.13	.22

Nota.

^aPosible intervalo de variación de puntuación: 9-45.

^bPosible intervalo di variación de puntuación: 12-60.

^cPosible intervalo di variación de puntuación: 11-55.

Fuente: elaboración propia.

La prueba de Mann-Whitney se repitió considerando también la puntuación obtenida para cada actividad (lecciones, tareas, evaluaciones) y para cada elemento haciendo la comparación entre los dos grupos. De esta manera, pasando del resultado general a un análisis más detallado surgen diferencias estadísticamente significativas con respecto a la variable alegría-diversión para las actividades de ejecución de las tareas asignadas (U= 5.28, z= -2.207, p =.027, r= .33) y durante las evaluaciones (U= 3.08, z= -2.11, p =.035, r= .31) (Anexo B). Las mismas consideraciones pueden hacerse para el elemento 1a *Cuando hago las tareas de geometría para la casa, estoy contento* (U= 165.00, z= -2.201, p =.028, r= .33) y el elemento 1b *Me divierto en las*

evaluaciones de geometría ($U= 159.50$, $z= -2.331$, $p=.020$, $r= .34$) (Anexo B) en los que se registraron diferencias estadísticamente significativas entre los dos grupos con valores de p -value que caen en la zona de no aceptación de la hipótesis nula, confirmando en este caso que, al acotar el campo de análisis, existe una diferencia en las respuestas dadas entre los dos grupos, a favor del grupo experimental.

4.3. Análisis cuali-cuantitativo

Se consideró necesario agregar, a la investigación cuantitativa, una estrategia de investigación de tipo cualitativa, enfocada en analizar y comprender aquello que solo con datos numéricos no era posible.

4.3.1. Observaciones en las aulas

La estrategia de recogida de datos para la investigación cualitativa a través de la observación no participativa requería buenas habilidades de escucha, saber leer entre líneas y captar ciertas señales para enriquecer el trabajo (Yin, 2011). Las observaciones permitieron estudiar múltiples aspectos asociados al proceso de enseñanza y de aprendizaje de la geometría, para después examinarlos relacionándolos entre sí (comparación entre los grupos) y dentro de su entorno.

Se anotaron los comportamientos de los niños en el aula durante las fases del aprendizaje, verificando su participación en varios niveles, no solo como una simple participación sino también como un posible interés que surge con intervenciones en la discusión, con sugerencias de ideas y propuestas de soluciones. También se registraron todos aquellos indicadores relacionados con las emociones: por ejemplo, para comprender como los niños viven el desarrollo del trabajo en clase se anotaron sus expresiones verbales y expresivas relacionadas a emociones como la ansiedad, la alegría, el entusiasmo, la diversión o también la satisfacción que muestran cuando logran

el objetivo preestablecido.

También se observaron las habilidades socio-relacionales de los niños y niñas, así como las actividades realizadas y las estrategias puestas en práctica por las docentes durante las horas de clase.

Como ya se ha mencionado anteriormente, los datos cualitativos se analizaron subdividiéndolos esencialmente en tres categorías para poder definir mejor el dominio de investigación; los códigos identificados para cada categoría se transformaron en variables para poder estudiarlas más a fondo analizando su contenido.

Para esto, se eligió realizar un análisis cualitativo de los datos utilizando los métodos y los criterios estadísticos aplicados en las investigaciones de tipo cuantitativo, para determinar la distribución de frecuencias y las diversas relaciones (Cohen et al., 2013).

La investigación cualitativa en cuestión fue respaldada por el software *MAXQDAAnalytcs Pro* (Release 18.2.0) que sigue los criterios generales de los CAQDAS (*Computer Assisted Qualitative Data Analysis Software*) que, desde un punto de vista constructivista, permite al investigador facilitar el proceso de encontrar, descomponer, analizar y recomponer el fenómeno que se está estudiando (Yin, 2015). *MAXQDA*, en particular, representa una herramienta válida que se ha implementado siguiendo el enfoque teórico y metodológico de la *grounded theory* con referencia explícita a las elecciones terminológicas y a la lógica de la recursividad que representa la base del enfoque (De Gregorio & Lattanzi, 2012).

Así pues, se guardaron en el software los documentos derivados de las reelaboraciones de las notas de capo y se activó la codificación de los datos, que consiste en la atribución de una etiqueta verbal (código) con el fin de sintetizar su contenido y facilitar su recuperación en momentos posteriores

cuando se realizan las comparaciones, evaluaciones y mapeos de los textos reportados en los siguientes párrafos.

Las notas de campo junto con todos los demás elementos útiles obtenidos a partir de las observaciones, imágenes fotográficas, entrevista se analizaron después de haber definido una cierta estructura de codificación (Glaser & Strauss, 2017).

Al principio, los datos se codificaron de forma separada; posteriormente se realizó un análisis cruzado que se concentró principalmente en analizar semejanzas y/o diferencias entre los dos grupos de trabajo.

En el grupo control se realizaron y codificaron n=6 observaciones semiestructuradas no participantes con duración promedio de 111,67 minutos (DS=20,41 min= 70, máx.=120 min), el número promedio de estudiantes en clase resultó ser 20 (DS= 1,26 min= 19, máx.= 22).

En la clase del grupo experimental se realizaron y codificaron n.9 observaciones semiestructuradas no participantes con duración promedio de 80 minutos (DS= 30 min= 60 máx.= 120), el número promedio de estudiantes en clase resultó ser 22,89 (DS= 2,57 min= 17, máx.= 25).

Las estadísticas descriptivas calculadas con referencia a cada código se reportan en la Tabla 36.

Tabla 36

Estadísticas descriptivas globales con referencia a cada código identificado para cada grupo

	Grupo control			Grupo experimental		
	<i>f (%)</i>	<i>M</i>	<i>DS</i>	<i>f (%)</i>	<i>M</i>	<i>DS</i>
Códigos con prevalencia negativa						
Distracción	148(54.6)	15.56	14.81	140 (63.6)	24.67	8.07
Desinterés	50 (18.5)	3.78	3.11	29 (13.2)	8.33	4.68

	Grupo control			Grupo experimental		
	<i>f (%)</i>	<i>M</i>	<i>DS</i>	<i>f (%)</i>	<i>M</i>	<i>DS</i>
Aburrimiento	40 (14.8)	2.89	3.95	27 (12.3)	6.67	6.53
Aburrimiento/cansancio	12 (4.4)	1.00	1.12	9 (4.1)	2.00	1.79
Preocupación y ansiedad	13 (4.8)	1.22	1.30	11 (5.0)	2.17	1.83
Enojo	4 (1.5)	.33	.71	3 (1.4)	.67	1.21
Desilusión	4 (1.5)	.44	1.01	1 (0.5)	.44	.82
Códigos con prevalencia positiva						
Motivación	58 (28.7)	14.44	10.47	130 (34.4)	9.67	7.76
Curiosidad e interés	25 (12.4)	7.56	3.32	69 (18.3)	4.17	2.32
Alegría y diversión	0 (.0)	4.22	5.04	38 (10.1)	.00	.00
Participación	110 (54.5)	11.00	9.68	99 (26.2)	18.33	3.98
Satisfacción	9 (4.5)	4.67	4.36	42 (11.1)	1.50	1.97
Códigos con prevalencia neutral						
Actividades	65 (40.4)	8.33	4.42	75 (40.3)	10.83	4.07
Relaciones m_a	65 (40.4)	6.44	4.36	58 (31.2)	10.83	4.36
Relaciones c_c	17 (10.6)	3.00	2.24	27 (14.5)	2.83	2.24
Estrategias	14 (8.7)	2.78	4.36	26 (14.0)	2.33	4.36

Para verificar la fiabilidad del análisis de datos se realizó una triangulación de las distintas fuentes; por ejemplo, la docente habló, durante la entrevista, del entusiasmo que manifestaron los niños durante las sesiones de geometría con Scratch, esto pudo confirmarse tanto con el repertorio fotográfico en donde se evidencian las actitudes alegres y entusiastas de los niños (se abrazan, aplauden, sonrían) (Anexo C), como también al leer las transcripciones de las notas de campo. Esto, en comparación con la participación y la motivación que los estudiantes mostraron durante una lección tradicional durante la cual se notan más distraídos, aburridos y desinteresados, las fuentes, también en este caso, se confirmaron entre sí.

4.3.1.1. Observación del grupo control

El plan de la investigación preveía la realización, para la clase del grupo

control (n=22), de una observación no participante, utilizando los mismos criterios de investigación que se usaron para el grupo experimental. Como ya se ha mencionado, la clase del grupo control no se sometió al tratamiento, es decir no realizaron actividades de *coding*, sino que trabajaron bajo las mismas condiciones de siempre con respecto a cómo se condujo la lección, la asignación de tareas y la realización de evaluaciones.

Por tanto, la docente de la clase llevó a cabo las actividades programadas sin aportar cambios a la metodología didáctica que aplica usualmente para enseñar la geometría. El trabajo se realizó en varios días y se dividió en actividades de 120 minutos cada una con un total de 12 horas. La docente utilizó, durante todo el proceso de enseñanza y aprendizaje, herramientas tradicionales para la transmisión del conocimiento (libros de texto, pizarra, hojas de papel, herramientas de dibujo); en el aula no estuvo presente la LIM únicamente una computadora portátil que no se utilizó para fines didácticos.

Generalmente, al inicio de cada lección, la docente pregunta a los niños para verificar si los conceptos ya tratados están consolidados. El proceso de construcción del conocimiento generalmente se basa en la lectura de la lección descrita en el libro y posteriormente interviene la docente que, con otros ejemplos y otras palabras, completa la presentación del tema que se estudia; continúa proponiendo ejercicios para realizar, siempre extraídos del libro, o con la transcripción de esquemas y apuntes en el cuaderno. Con frecuencia, los niños son llamados y se les pide pasar a la pizarra para realizar los ejercicios indicados por la maestra.

Durante todo este proceso, se tomó nota de los comportamientos, las emociones y las dinámicas relacionales observadas. En general, aproximadamente seis niños, se mostraron atentos, curiosos e interesados a través de una participación constante. Un grupo de trece niños interviene si es cuestionado y realiza de manera regular el trabajo asignado, especialmente si se trata de copiar esquemas y apuntes en el cuaderno. Uno de ellos se

distingue por sus intervenciones poco pertinentes y por interrupciones continuas que se asocian a su deseo por hacerse notar y ser el centro de atención, como lo confirma la docente; en clase, tres niños se mostraron particularmente desinteresados, dos de ellos intentaron de cualquier forma realizar el trabajo, aunque se distraían continuamente, y el otro se negó expresamente a trabajar repitiendo que se estaba aburriendo (*¡Maestra me aburro!*). La maestra, frecuentemente, intervino invitándolo a trabajar, pero tuvo pocos resultados; el niño raramente aceptó el consejo, pero cuando lo hizo, mostró buenas habilidades lógicas. El ambiente relacional es positivo, los niños respetan las reglas, realizan lo que se les pide, están en silencio y los que se distraen no perturban la lección. Alguna vez se registró algún desacuerdo entre los compañeros que se resolvió rápidamente gracias a la maestra hacia la que mostraron una actitud respetuosa y correcta, como también hacia las reglas básicas que regulan la vida escolar cotidiana.

La docente guía a los niños en la construcción del conocimiento poniendo particular atención, durante la realización de los ejercicios, en el uso de herramientas de diseño, a menudo recorre los pupitres para comprobar lo que hace cada alumno, estimulando a quien no trabaja o a quien se atrasa. En cuanto a quienes se muestran inseguros, la maestra busca transmitir valor y confianza.

Las dos horas consecutivas de geometría, generalmente las primeras dos de la mañana, son vividas con ritmos bastante lentos por una buena parte de los componentes de la clase que a menudo se muestran cansados y aburridos. Son pocos los niños que parecen motivados para trabajar especialmente cuando la maestra asigna las tareas para realizar en clase que requieren mayor autonomía.

El modelo visual de la Figura 8 presenta las conexiones entre códigos y subcódigos referentes a las anotaciones realizadas durante las horas de observación del grupo control; el grosor de los segmentos que conectan códigos

y subcódigos es proporcional a la frecuencia de detección y proporciona una visión de todo el proceso de observación realizada en clase.

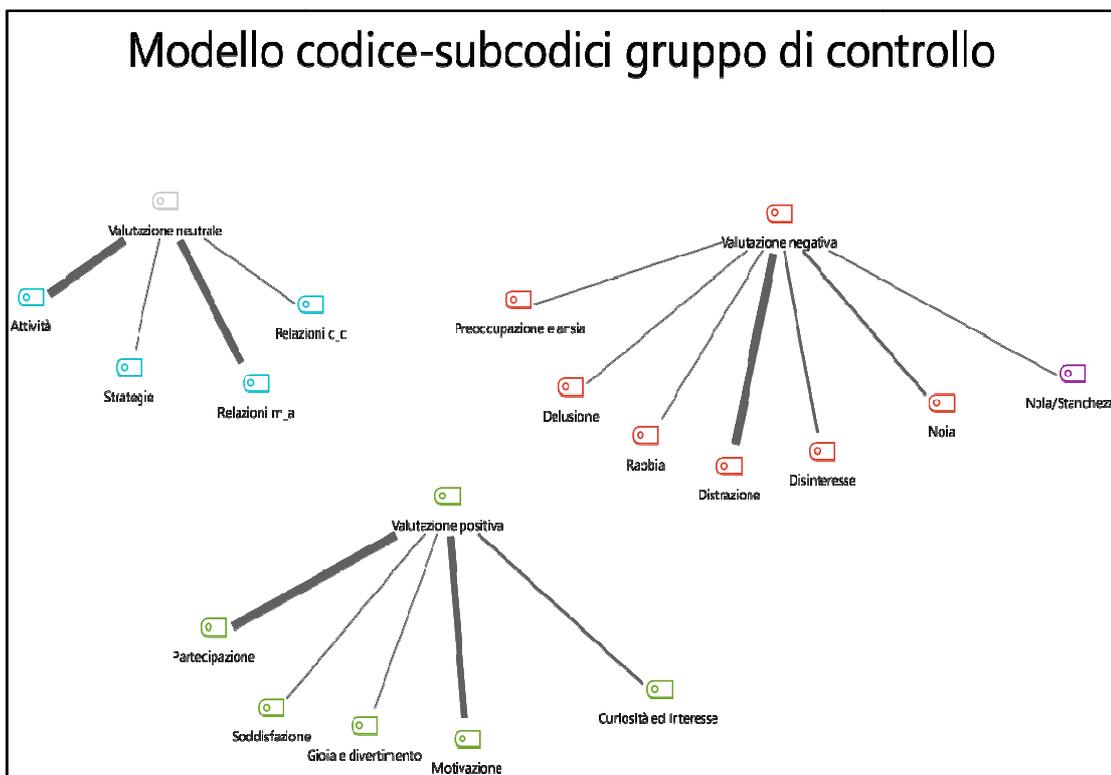


Figura 8. Modelo de distribución de los códigos (grupo control)

La Figura 9 muestra la disposición del aula en la que es posible observar el entorno de la clase que se mantuvo así durante todo el proceso, y la indicación de los puntos de vista desde los que, de hecho, se realizó la observación.

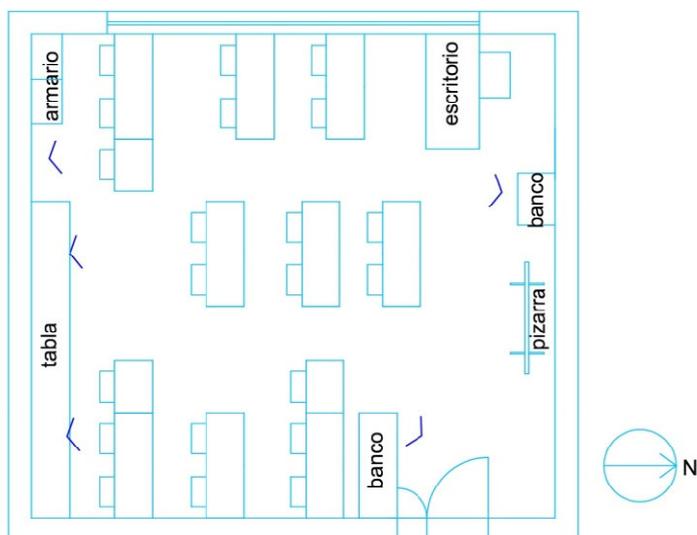


Figura 9. Aula: entorno de clase (grupo control) y puntos de observación

A partir de una primera revisión de las transcripciones de las notas de campo, es posible verificar visualmente en base a la matriz de códigos elaborada con *MAXQDA* (Figura 10), para cada jornada de observación, cómo se distribuyen todos los factores del proceso de enseñanza/aprendizaje primero registrados y después codificados.

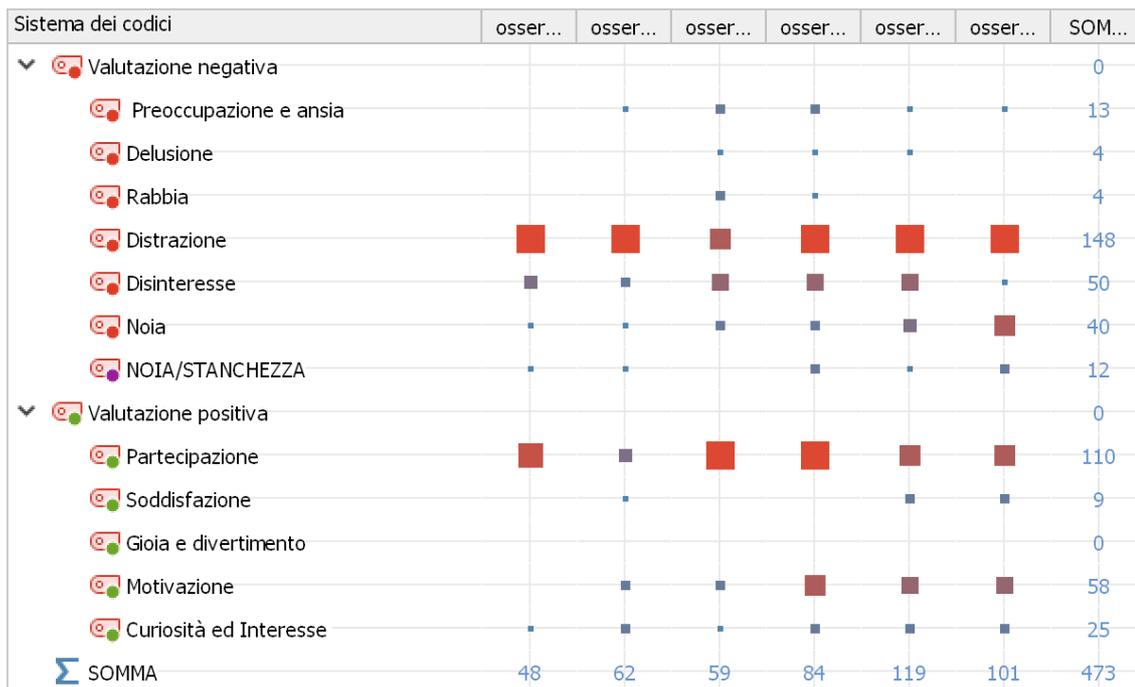


Figura 10. Categorías de códigos comparadas en MAXQDA (grupo control)

A partir de la tabla de comparación de los documentos fue posible hacer una evaluación global de la distribución de los códigos imperantes que se encuentran repartidos, en la mayor parte de las observaciones, durante toda la hora. Solamente en muy pocos casos prevalece una categoría de códigos sobre las demás, como en la observación del 16 de mayo en la que, en la parte final de la lección se observaron elementos relacionados con la categoría de códigos de prevalencia positiva (MAXQDA).

Analizando los datos recogidos, desde el punto de vista cuantitativo surge una presencia predominante de casos relacionados con las variables distracción y desinterés, para las que se registra una frecuencia porcentual respectivamente del 31.30% y del 10.60%. Durante la realización del trabajo en clase, en el 23.30% de los casos, se registró una buena participación por parte de los niños que sumándose a los códigos motivación y curiosidad e interés alcanza un valor del 40.90%. El 11.00% de los casos señalados y codificados con el término aburrimiento y aburrimiento/cansancio reflejan expresiones verbales como “estoy aburrido”, bostezos y miradas ausentes. No

se registraron hechos relacionados con comportamientos con referencia al código llamado alegría y diversión (Figura 11).

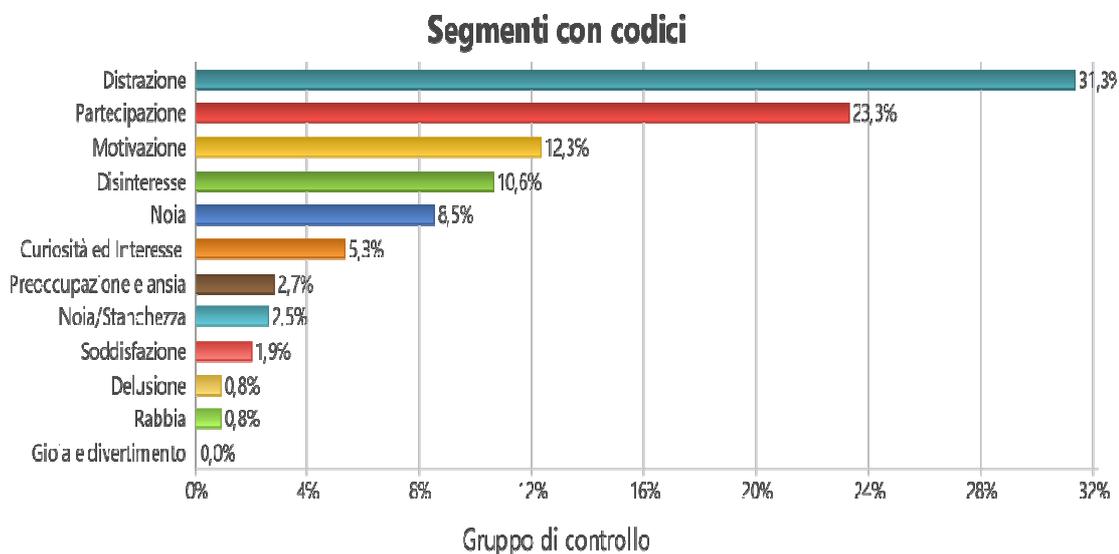


Figura 11. Distribución de la frecuencia porcentual de los códigos (MAXQDA) en el grupo control

Considerando las variables relacionadas al ámbito emocional y motivacional dentro de una matriz de correlación, se puede notar en la Tabla 37 que algunas variables relacionadas a emociones con valencia negativa, como enojo y desilusión, se correlacionan positivamente entre sí ($r = .876, p < .05$). Existe también una buena correlación entre las variables que pertenecen al grupo emociones con prevalencia positiva, como satisfacción y curiosidad e interés ($r = .852, p < .05$), también para participación y motivación refiriéndose a todos aquellos casos en los que los niños alzaron la mano o buscaron interactuar de alguna manera durante todo el proceso de construcción del conocimiento, incluso a través del asentimiento con la cabeza, con movimientos del cuerpo o respondiendo cuando se lo solicitó la maestra.

También surgieron relaciones entre variables entendidas en el sentido positivo con variables negativas relacionadas a la conducta de los individuos y, en particular, de un niño como se mencionó anteriormente que se mostró siempre distraído.

Tabla 37*Correlaciones entre las variables presentes en el conjunto de datos de las observaciones (grupo control)*

	1	2	3	4	5	6	7	8	9	10	11	12
1 Preocupación y ansia	1											
2 Desilusión	.712	1										
3 Enojo	.480	.876*	1									
4 Distracción	-.006	-.350	-.668	1								
5 Desinterés	.598	.559	.129	.254	1							
6 Aburrimiento	-.074	-.147	-.273	.684	-.174	1						
7 Aburrimiento/cansancio	-.122	-.548	-.462	.319	-.598	.605	1					
8 Participación	.635	.156	-.175	.521	.371	.525	.497	1				
9 Satisfacción	-.138	-.248	-.502	.897*	.087	.905*	.396	.483	1			
10 Alegría y diversión	a	a	a	a	a	a	a	a	a	a		
11 Motivación	.512	.450	-.216	.771	.397	.739	.396	.897*	.762	a	1	
12 Curiosidad e interés	.039	-.388	-.689	.991**	.215	.655	.386	.558	.852*	a	.777	1

*correlación significativa al 5% ($p < .05$)**correlación significativa al 1% ($p < .01$)^a cálculo imposible para variable constante.*Fuente:* elaboración propia.

4.3.1.2. Observación en la fase de tratamiento del grupo experimental

El grupo de la clase que se sometió al tratamiento a través del uso del *codingse* observó durante el desarrollo de toda la actividad didáctica, dividida en lecciones de 60 o 120 minutos, que consistió, como ya se ha mencionado, en el estudio de las figuras geométricas planas y en el análisis de sus elementos significativos (vértices, ángulos, lados, ejes de simetría, diagonales).

La observación duró en total 13 horas divididas entre el trabajo en clase y las actividades de trabajo multimedia, en el periodo comprendido entre marzo y mayo del 2019.

Las Figuras 12 y 13 que se muestran a continuación muestran el entorno de clase en los dos ambientes de aprendizaje en los que se trabajó con los respectivos puntos de observación.

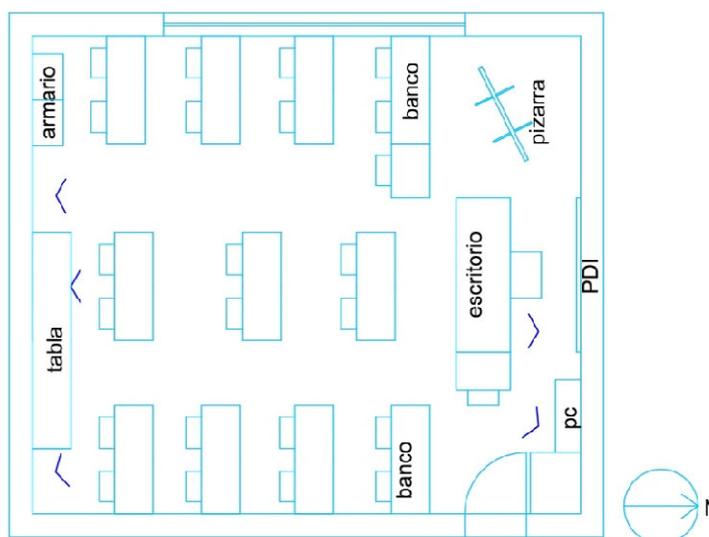


Figura 12. Aula: entorno de clase (grupo experimental) y puntos de observación

En el laboratorio, como en el aula, se instaló una LIM y a lo largo de tres paredes se distribuyeron las computadoras (N=12); la maestra puso a disposición de un grupo de niños otra computadora que se encuentra en el escritorio (Anexo C).

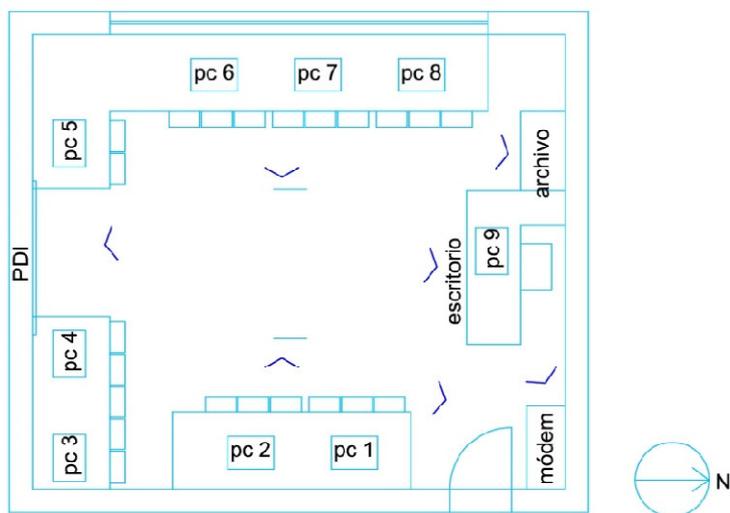


Figura 13. Laboratorio multimedia: entorno de clase (grupo experimental) y puntos de observación

La docente propuso una actividad de *coding* a la clase como herramienta pedagógica tanto para introducir a los niños a los nuevos conceptos como para consolidar los conceptos ya vistos en clase, sin descuidar la idea de que haciendo *coding* los niños se educan al pensamiento computacional. Durante su trabajo, a menudo se comportó como guía para sus alumnos haciéndolos reflexionar e interrogarse con respecto a las tareas asignadas, estimulando el uso de los procesos cognitivos que involucraron análisis, síntesis y capacidad de resolución de problemas. También trabajó como observadora, interviniendo en las interacciones de grupo a través de retroalimentaciones positivas hacia quien se mostró disponible y facilitó la participación de los demás compañeros; y viceversa, en presencia de niños menos dispuestos a interactuar con sus

compañeros, intentó infundirles confianza involucrándolos cada vez más en las actividades a desarrollar. Dirigió las actividades entre los dos ambientes físicos de aprendizaje buscando de dar continuidad y conexión lógica a las diversas actividades diseñadas, relacionándolas entre sí.

Los datos recogidos durante la observación en clase reflejaron una buena participación de los niños que, ciertamente, trabajan con mayor constancia y atención en la primera mitad de la lección. Un grupo de aproximadamente diez niños, en general, participa con interés, interactúa con la docente facilitándole el trabajo de construcción del conocimiento. Hubo dos niños que se mostraron entusiastas y particularmente motivados, tanto que la docente se vio obligada a frenarlos para dar espacio a los demás compañeros. Hubo otro grupo de niños, aproximadamente diez, que solo intervenía si se les pedía, siguieron las actividades, trabajaron silenciosamente, pero, en los momentos de dificultad, no dudaron en solicitar ayuda a la maestra. Entre estos niños, alguno se distrajo con facilidad y, a menudo, esto lo nota el docente que busca involucrarlos a través de preguntas o simplemente llamándoles la atención. Los niños menos interesados son dos y, en particular, uno de estos raramente se mostró disponible para realizar algunas tareas o para participar en la lección.

La realización de la actividad en el aula sigue un protocolo bien definido: el docente inicia la lección retomando los conceptos y los temas tratados en la lección anterior, facilitando de esta forma la participación activa de los estudiantes a través de un intercambio y una discusión de grupo. Después introduce los nuevos conceptos, a través de un diálogo continuo, propone ejercicios y esquemas que los niños deben realizar en su cuaderno. A menudo, durante la lección, busca, de todas las maneras posibles, estimular a quien se muestra distraído o poco interesado llamándole la atención, sin descuidar a quienes muestran una curiosidad cognitiva. Las herramientas que utiliza como soporte de su trabajo (LIM, pizarra, material en papel, herramientas de dibujo) sirven como estímulo para el aprendizaje.

La simple idea de ir al laboratorio de multimedia entusiasma y estimula a los niños particularmente. Las actividades realizadas en el laboratorio hicieron registrar un levantamiento en el tono de la voz y murmullos de fondo que se justifica parcialmente por el hecho de que el tipo de actividad requiere el trabajo de grupo. Después de haber dividido a los niños en pequeños grupos, la docente propone el trabajo que deben realizar en la plataforma Scratch que trata de aplicaciones prácticas a través de ejercicios y tareas relacionadas con los temas de geometría (figuras planas, perímetro, área) presentadas en clase.

Generalmente, el nivel de participación es muy alto, los niños se mostraron interesados y motivados; la ejecución del ejercicio los lleva a preguntar, a confrontarse con los compañeros con el fin de alcanzar el objetivo establecido; una vez que logran realizar la tarea asignada muestran entusiasmo y satisfacción. En algunos casos surgieron malentendidos entre los componentes del mismo grupo debidos al hecho de compartir el ordenador y fueron resueltos por la docente.

En el modelo visual de la Figura 14 se puede observar las relaciones entre códigos y respectivos subcódigos en función de las detecciones realizadas durante las horas de observación del grupo experimental; el esquema muestra también el grosor de los segmentos de conexión que es proporcional a la frecuencia de detección de los subcódigos detectados después del proceso de observación realizado en clase.

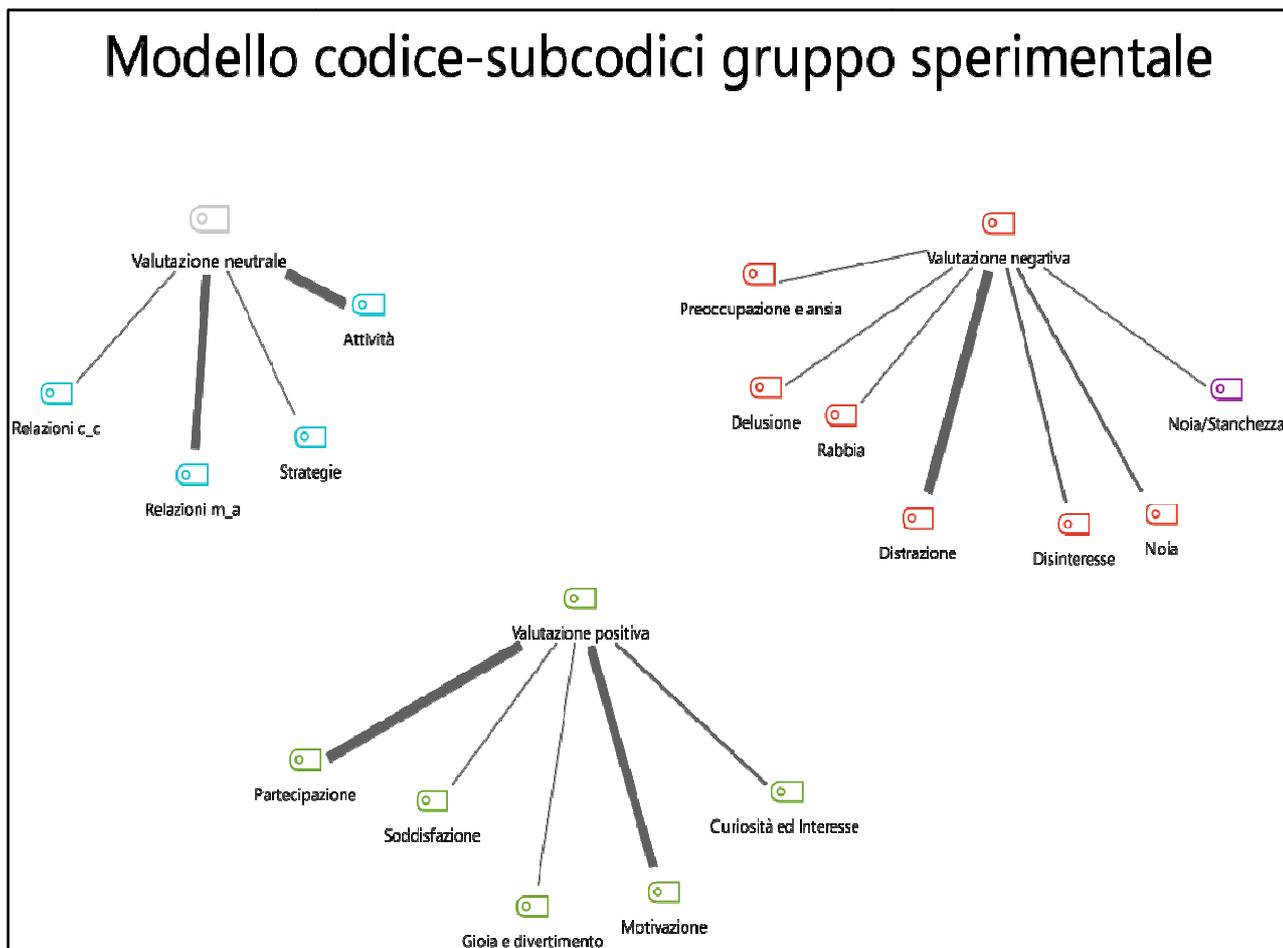


Figura 14. Modelo de distribución de los códigos (grupo experimental)

El porcentaje de niños participantes es alto, solo un niño no muestra interés ni participa. Como se espera de un ambiente de aprendizaje de laboratorio, surgen claramente los aspectos relacionados con la colaboración entre estudiantes y a las habilidades socio-relacionales. Los dos niños que se destacaron en clase se destacaron también en el laboratorio, se mostraron siempre curiosos, propositivos y entusiastas. Incluso la docente lo confirma esta en la entrevista diciendo que *“Los buenos niños son siempre buenos”* en cualquier situación didáctica en la que se encuentren.

La realización de la actividad de *coding* con Scratch facilitó la experiencia directa de los estudiantes que trabajaron en un proyecto personalizado, discutiendo el procedimiento a aplicar para lograr la solución y, al mismo

tiempo, cuidando el aspecto gráfico (*sprite* y *stage*), lo que destacó la parte lúdica del trabajo. Los estudiantes, desde el inicio de la actividad, se mostraron interesados y predispuestos favorablemente.

Esto se demuestra con los datos elaborados que indican las diferencias en las frecuencias porcentuales con las que se codificaron las variables de interés en el mismo grupo de clase entre el momento en el que se realizan las actividades con Scratch y el momento en el que, incluso a través de un sistema didáctico de tipo constructivista, la lección se desarrolla sin el uso del *coding*. Esta consideración puede extenderse tanto al grupo de códigos que se refieren a la categoría de los factores con prevalencia positiva como a aquellos con prevalencia negativa.

La tabla 38 muestra cómo cambia la frecuencia porcentual de las variables en observación, cuando, precisamente, dentro de la misma clase, se proponen actividades de geometría con herramientas didácticas distintas.

Tabla 38

Distribución de frecuencias de códigos significativos (grupo experimental)

	Actividades de <i>coding</i>		Actividades tradicionales	
	<i>f</i>	%	<i>F</i>	%
Códigos con prevalencia negativa				
Distracción	41	14.90	99	30.70
Desinterés	10	3.60	19	5.90
Aburrimiento	8	2.90	19	5.90
Aburrimiento/cansancio	4	1.40	5	1.60
Preocupación y ansiedad	4	1.40	7	2.20
Enojo	1	.40	2	.60
Desilusión	0	.00	1	.30
Total	68	24.60	152	47.2
Códigos con prevalencia positiva				
Motivación	55	19.90	75	23.30
Curiosidad e interés	49	17.80	20	6.20
Alegría y diversión	37	13.40	1	.30

Participación	34	12.30	65	20.20
Satisfacción	33	12.00	9	2.80
Total	208	75.40	170	52.80

Fuente: elaboración propia.

A la propuesta del uso del *coding* durante las actividades de geometría parece corresponder un aumento de códigos y sus respectivas frecuencias atribuibles particularmente a la esfera emocional, en el contexto de elementos evaluables positivamente. Los datos se confirman precisamente en la observación de distintas circunstancias en las que los niños manifiestan satisfacción en la ejecución y resolución de una tarea, felicidad y entusiasmo; después de verificar que la han realizado correctamente, aplauden como señal de aprobación e intervienen en voz alta diciendo “¡Es muy guay!”, “¡Lo logramos!”.

La actividad realizada con Scratch los estimula para relacionarse: los niños trabajan en grupos de tres, interactúan entre sí y, cuando alcanzan el objetivo, se muestran contentos y lo celebran con sus compañeros con un abrazo o ayudan a otros compañeros con alguna sugerencia para la resolución de la tarea. En algunos pocos casos, los niños no lograron encontrar ponerse de acuerdo, en particular con respecto al uso de la computadora, y comenzaron una discusión que la maestra resolvió rápidamente.

Los estudiantes parecen apreciar las actividades de *coding* tanto que se mostraron eufóricos cada vez que la maestra les propuso trabajar con Scratch. Lo confirma la docente en la entrevista:

“Noté una mayor participación de los niños, un mayor entusiasmo con respecto a la lección tradicional”.

Los retos propuestos sirvieron como estímulo para la mayor parte de los

niños, incluso a muchos de los que, durante las actividades escolares más tradicionales, se muestran poco interesados y casi no participan. De hecho, algunos de ellos, se mostraron más motivados, más independientes, más constantes en el trabajo a realizar con respecto a cuándo no aprendieron a través del *coding*. Esto los lleva a aumentar su motivación para aprender y a tener mayor confianza en sus propias capacidades. Incluso la docente lo resaltó diciendo:

“Noté diferencias en los niños con mayores dificultades. Quizá durante la lección tradicional ellos no levantaban la mano jamás, en cambio durante las actividades de coding realizaron observaciones que a veces eran incluso relevantes”.

Los episodios observados codificados con el código aburrimiento /cansancio y aburrimiento, en total, disminuyeron durante las lecciones con el *coding* con respecto a las lecciones más tradicionales.

En cuanto a los eventos codificados con el término distracción los datos reflejan una propensión por parte de los niños a distraerse menos durante las sesiones con Scratch (Tabla 29). De hecho, con el mismo número de horas de observación, se registraron más casos codificados con el término distracción cuando no se realizó el *coding* en cambio el código motivación presenta frecuencias más altas en las horas de laboratorio. En particular, con referencia a los eventos registrados en el laboratorio, es necesario especificar que se observaron situaciones en las que algunos niños terminaron su trabajo en muy poco tiempo y con una cierta autonomía con respecto a otros que solicitaron más tiempo. Este hecho, en mi opinión, favoreció los momentos de distracción registrados. También es importante mencionar que el entusiasmo que los niños mostraron en el laboratorio, en algunos casos, provocó un aumento en el tono de la voz y confusión, elementos que misma docente destacó en la entrevista.

Los datos relevantes se refieren a las variables satisfacción y alegría y

diversión que muestran un aumento porcentual significativo (de 2.80% a 12.00% para satisfacción y de 0.30 % a 13.40% para alegría y diversión) dentro del mismo grupo de niños (Tabla 30).

A partir del análisis también se sugiere que, durante el desarrollo de las actividades de *coding*, aumentaron, de manera general, las relaciones sociales entre los compañeros pues hubo más oportunidades de interactuar y de compartir el trabajo realizado. De hecho, se nota que, si se encuentran en dificultad, no dudan en pedir ayuda a sus compañeros y buscan confrontarse con ellos, por otro lado, en algunas circunstancias, la posibilidad de trabajar en grupo por un proyecto en común se transforma en una ocasión para iniciar una discusión que, la mayor parte de las veces requiere la intervención por parte de la docente.

Al englobar los datos en un único cálculo, sin subdividirlos especificando la actividad desarrollada, para el grupo experimental es posible leer el gráfico de barras de la Figura 15, donde el código predominante es la distracción (23,4%), este es seguido por cinco códigos clasificados como positivos.

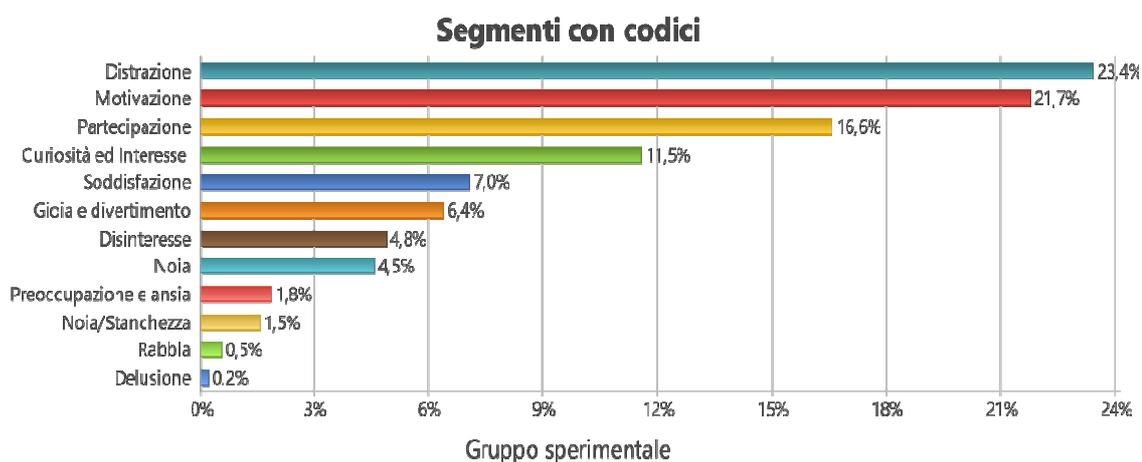


Figura 15. Distribución de la frecuencia porcentual de los códigos (MAXQDA) en el grupo experimental

En cambio, a partir de la Figura 16 se pueden extrapolar algunos datos relacionados con aquello que surgió en cada una de las jornadas de

observación, los cuadros más grandes muestran la prevalencia de comportamientos y actitudes que pueden asociarse a la variable indicada.

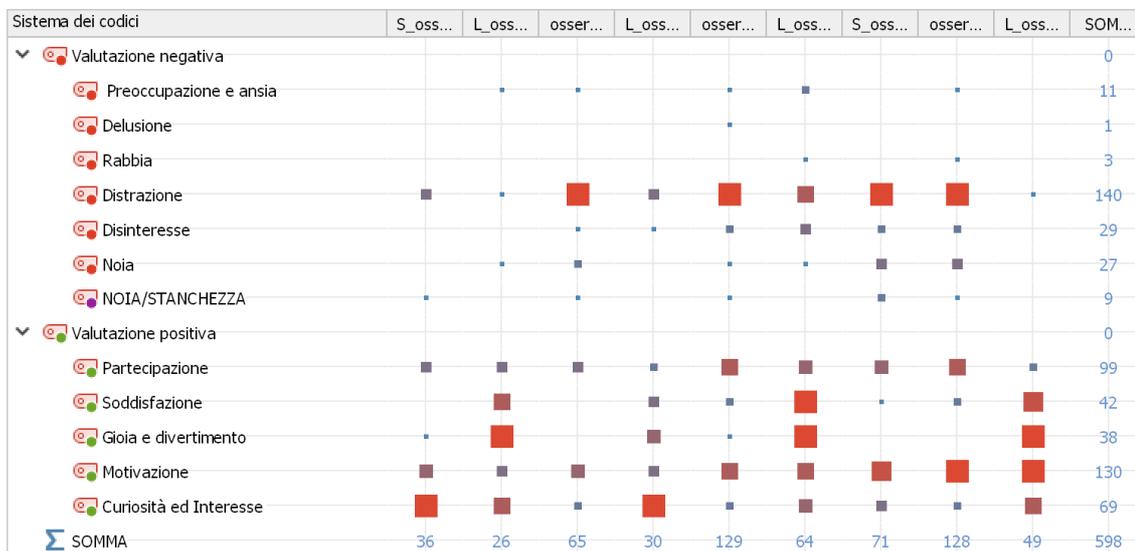


Figura 16. Categorías de códigos comparados en MAXQDA (grupo experimental)

La matriz triangular que se puede observar en la Tabla 39 finalmente muestra los valores del índice de correlación de Pearson con referencia a las variables que surgieron a partir de la observación, como ya se reportó para el grupo control. En este caso, se pueden identificar buenas correlaciones entre elementos de un mismo grupo, como por ejemplo entre aburrimiento y distracción ($r=.934, p<.001$), entre preocupación y rabia ($r=.944, p<.001$), motivación y participación ($r=.830, p<.05$) y alegría y satisfacción ($r=.966, p<.001$). Se observan también correlaciones positivas entre variables que pertenecen a categorías diferentes y, en particular, para el código participación con respecto a distracción, aburrimiento y aburrimiento/cansancio: dicho resultado se puede explicar tomando en cuenta el significado de la variable participación (Capítulo 3), que precisamente incluye el comportamiento de los niños sobre todo en respuesta a las llamadas de atención individuales o colectivas por parte de la maestra cuando detectó una reducción en el umbral de atención de los alumnos.

Tabla 39*Correlaciones entre las variables presentes en el conjunto de datos de las observaciones durante las actividades de coding (grupo experimental)*

		1	2	3	4	5	6	7	8	9	10	11	12
1	Preocupación y ansiedad	1											
2	Desilusión	^a	^a										
3	Enojo	.944**	^a	1									
4	Distracción	-.052	^a	.078	1								
5	Desinterés	.448	^a	.581	.769	1							
6	Aburrimiento	-.094	^a	-.070	.934**	.638	1						
7	Aburrimiento/cansancio	-.364	^a	-.270	.913*	.448	.895*	1					
8	Participación	.094	^a	.210	.972**	.783	.927**	.848*	1				
9	Satisfacción	.578	^a	.583	-.381	.057	-.371	-.636	-.192	1			
10	Alegría y diversión	.614	^a	.568	-.459	-.087	-.412	-.655	-.250	.966**	1		
11	Motivación	-.042	^a	.099	.753	.607	.748	.610	.830*	.189	.080	1	
12	Curiosidad e interés	.091	^a	.135	-.187	.112	.377	-.364	-.188	.509	.312	.209	1

*correlación significativa al 5% (a dos colas)

**correlación significativa al 1% (a dos colas)

^a cálculo imposible por variable constante.*Fuente:* elaboración propia.

4.3.1.3. Comparación de resultados

Las observaciones realizadas en ambas clases permitieron recoger los datos necesarios para poder realizar un análisis que pudiera evidenciar, dentro de los dos procesos de enseñanza y de aprendizaje, respectivamente, analogías o diferencias desde un punto de vista cualitativo.

Deteniéndonos con atención a analizar el contenido de cada observación realizada, que tuvo la misma duración total (12 horas) y en el mismo periodo (marzo-mayo del 2019), en ambos grupos los casos de distracción superan en porcentaje todas las demás situaciones reportadas en las notas de campo.

La motivación para aprender además de la curiosidad y el interés por las actividades propuestas tuvieron una mayor frecuencia en el grupo experimental con respecto al grupo control en donde las intervenciones de la docente típicas de una lección frontal, fueron más frecuentes con respecto a las lecciones de laboratorio, en las que la docente interviene proponiendo estrategias de trabajo alternativas y dando mayor libertad para el desarrollo de las mismas, a lo que se suma una mayor relación entre los compañeros (Tabla 40).

Tabla 40

Distribución de las frecuencias de códigos: actividades y estrategias (comparación de grupos)

	Grupo control		Grupo experimental	
	<i>F</i>	%	<i>f</i>	%
Estrategias	14	8.70	26	14.00
Actividades	65	40.40	75	40.30
Relaciones m_a	65	40.40	58	31.20
Relaciones c_c	14	10.60	27	14.50
Total	161	100.00	186	100.00

Fuente: elaboración propia.

En el grupo control se observaron más casos de aburrimiento (8.50%) con

respecto al grupo experimental (4.50%) y la frecuencia porcentual disminuye justo durante las sesiones de *coding*.

En cuanto a los eventos codificados con el término satisfacción el 1,90% del grupo control manifiestan experiencias que se relacionan a dicha variable, mientras que en el grupo experimental el porcentaje es 7,00%, en este grupo surgen niveles altos de satisfacción relacionados principalmente con los momentos en los que los niños verificaban que el trabajo realizado era correcto a través de retroalimentaciones positivas por parte de la maestra, o con mayor frecuencia, gracias a las retroalimentaciones de Scratch.

Los episodios de alegría y diversión fueron ausentes en el grupo control (.00%) mientras que en el grupo experimental claramente están relacionados con las actividades de *coding* y al uso de Scratch (6.40%) (Fig. 15).

Así, analizando los resultados relacionados con los dos dominios dentro de los cuales se codificaron las observaciones y se definieron las variables (Tabla 41) surge una mayor presencia de elementos relacionados con el dominio de prevalencia negativa en la clase del grupo control (57.20%). En cambio, las variables relacionadas a los aspectos del aprendizaje que pueden evaluarse como positivos están más presentes en el grupo al que se le aplicó el tratamiento (56.20%).

Tabla 41

Distribución de las frecuencias de códigos con valor positivo y negativo (comparación de grupos)

	Grupo control		Grupo experimental	
	<i>f</i>	%	<i>f</i>	%
Códigos con prevalencia negativa				
Distracción	148	31.30	140	23.40
Desinterés	50	10.60	27	4.80
Aburrimiento	40	8.50	27	4.50
Aburrimiento/cansancio	12	2.50	9	1.50

Preocupación y ansiedad	13	2.70	11	1.80
Enojo	4	.80	3	.50
Desilusión	4	.80	1	.20
Total	271	57.20	218	36.70
Códigos con prevalencia positiva				
Motivación	58	12.30	130	21.70
Curiosidad e interés	25	5.30	69	11.50
Alegría y diversión	0	.00	38	6.40
Participación	110	23.30	99	16.60
Satisfacción	9	1.90	42	7.00
Total	202	42.80	378	56.20

Fuente: elaboración propia.

En general, se puede observar, en el diagrama de barras de la Figura 17, la frecuencia porcentual con la que se distribuyen los tres grupos de códigos respectivamente en las dos clases, con una presencia predominante de códigos con valor positivo en el grupo experimental y de códigos con prevalencia negativa en el grupo control.

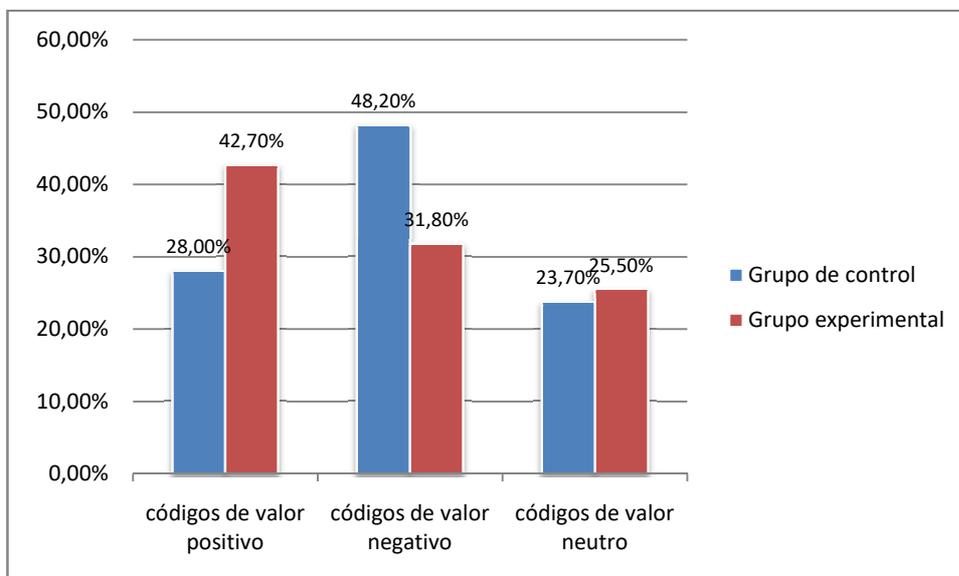


Figura 17. Categorías de códigos, comparación entre los grupos (distribución de frecuencia)

4.3.2. Entrevista a la docente del grupo experimental

El análisis cualitativo se extendió también al estudio del fenómeno desde la percepción de la docente quien fue parte activa del proceso de enseñanza/aprendizaje en la clase del grupo experimental. El objetivo era completar el estudio en cuestión. Como ya se ha mencionado, la docente de matemáticas y tecnología se puso a disposición para realizar la actividad en la clase, habiendo realizado ella misma en el año escolar anterior, sesiones de *coding* unplugged y lecciones de Scratch, gracias a las cuales los niños aprendieron a conocer algunas funciones de la plataforma.

Según lo informado, los niños aprendieron las actividades de la hora del código y como se trabaja en el ambiente Scratch, a través de experiencias de *game design*. La docente no había integrado jamás el *coding* como instrumento metodológico transversal a la enseñanza de una disciplina como las matemáticas, por lo que esta fue su primera experiencia (Anexo F).

Después de haber realizado la entrevista esta se transcribió en *MAXQDA* y, una parte de los datos recogidos, se triangularon con los provenientes de otros instrumentos de recogida (observaciones, imágenes fotográficas).

La entrevista semi estructurada se basó en algunas preguntas principales, dirigidas a obtener datos sobre la percepción de la importancia dada por la docente a la estrategia de enseñanza/aprendizaje utilizada, sobre las impresiones con respecto a los conocimientos adquiridos durante la aplicación de esta metodología y la percepción de las emociones de los estudiantes. Las respuestas proporcionadas confirmaron también los datos recogidos en el formulario que la misma docente completó antes del inicio del trabajo, sobre su formación profesional y la experiencia en *coding*.

La docente refirió que sus conocimientos sobre pensamiento computacional y *coding* se deben principalmente al uso de las plataformas Code.org y Scratch, las cuales ha utilizado, durante la enseñanza de la disciplina tecnología, en los temas relacionados con el conocimiento de la informática. Conoció estas plataformas gracias a un curso MOOC que realizó en los años anteriores, considerando muy importante la formación de los docentes antes de implementar nuevas actividades con los alumnos. Mostró interés y participación hacia estos temas, introduciéndolos inicialmente en el plan de estudios de tecnología; por otro lado sostiene que sus colegas no son propensas a acercarse a este mundo, quizá demasiado tecnológico. La docente valoró positivamente la experiencia realizada declarando que:

“...fue útil porque traté temas que, de lo contrario, con una lección tradicional de geometría, probablemente no habría abordado, así que desarrollé los conceptos con más profundidad.”

A la pregunta “¿Hubo algún momento en el que pensaste que las actividades que estabas realizando pudiesen haber sido, por ejemplo, inútiles?”, la docente responde:

“...nunca pensé que las actividades fueran inútiles pero a veces tuve algunas perplejidades en cuanto a cómo se desarrollaba la lección pues se creaba un poco de confusión ya que los niños estaban muy entusiasmados y por lo tanto, a veces, pensaba que la situación se salía de control.”

Luego expresó cierta preocupación afirmando que:

“con el manejo del pequeño grupo a menudo surgían pequeños problemas pues había siempre algún niño que asfixiaba a otro por lo que no siempre las dinámicas eran manejables.”

Con referencia a sus impresiones personales sobre el proceso de aprendizaje de los alumnos y sobre las emociones que surgieron, las respuestas proporcionadas resaltan las fortalezas del método *coding* en cuanto al aumento del esfuerzo, la motivación y el interés por parte de todos los niños, y en particular, de aquellos que en el aula se mostraron más débiles:

“Noté diferencias en los niños que tienen más dificultades. Quizá durante la lección tradicional ellos no alzaban jamás la mano, en cambio, durante las actividades de coding realizaron observaciones a veces incluso pertinentes.”

Y también:

“...durante la lección tradicional, muchos niños se aburren y pierden el sentido del tema, en cambio, durante las actividades de coding los niños se sentían siempre atraídos a la actividad y, a menudo, me daba cuenta, analizando la situación, que aquellos que en clase se distraían y no entendían el tema ahí lograban retomar en hilo y entender el tema.”

Consideraciones similares con respecto a las dinámicas relacionales que surgieron:

“en términos sociales vi que cuando los niños terminaban un trabajo tenían el placer de compartirlo con sus compañeros, cosa que en cambio en clase, durante la lección tradicional, difícilmente sucede a menos que se realice una actividad especial.”

Y al entusiasmo de los niños:

“Noté una mayor participación de los niños, un mayor entusiasmo con respecto a la lección tradicional; los niños lo

tomaban casi como un juego, sin embargo, después notaba que los conceptos que desarrollábamos eran realmente conceptos importantes aunque ellos pensaban solamente en jugar.”

Durante la entrevista también se abordó el tema de los resultados de la prueba de geometría realizada al final del curso (prueba final). Según la opinión de la docente, los resultados no reflejan la cantidad de trabajo realizado, el tiempo invertido y las expectativas al respecto. Sostiene que probablemente:

“Realizaron esta prueba con superficialidad y por lo tanto no alcanzaron los resultados que yo esperaba de ellos.”

La docente, en la conclusión de la entrevista, resalta que las actividades de *coding* requirieron mayor cantidad de tiempo con respecto al tiempo que se necesita para realizar una actividad de tipo tradicional, tanto que, en comparación con la programación realizada por la docente del grupo control, no logró introducir un tema que estaba en el programa. Por otro lado, mencionó y profundizó, como ya se ha dicho, en aspectos que diversamente no habría considerado. En el Anexo F se encuentra el formulario completado por la docente y la transcripción de la entrevista.

4.4. Conclusiones

En este capítulo se han recogido y analizado los datos cuantitativos y cualitativos derivados de la metodología utilizada: con referencia al proceso de enseñanza/aprendizaje activado en los dos grupos de clases, en este apartado se resumen los resultados del estudio realizado, en base a las herramientas aplicadas.

Los datos extrapolados de las pruebas de geometría aplicadas a todos los participantes de las dos clases se sometieron a análisis estadístico descriptivo e inferencial. Los niños del grupo experimental, sometidos al tratamiento que

utilizó el *coding* como herramienta de mediación didáctica, obtuvieron puntuaciones promedio que presentan diferencias estadísticamente significativas con respecto a las puntuaciones promedio conseguidas por los estudiantes del grupo control. El coeficiente R^2 permite medir el efecto de la intervención didáctica que, con respecto a la clasificación propuesta por Cohen (1988) se puede considerar alta en valores promedio.

Las pruebas estadísticas que se realizaron en los grupos de elementos subdivididos a nivel de dimensión o simplemente con referencia a los elementos individuales muestran resultados análogos que llevan a concluir que a través del análisis cuantitativo fue posible registrar efectos generalmente positivos en beneficio del grupo experimental.

Con respecto a las emociones de logro que surgieron de los datos del cuestionario, al comparar los grupos, no se encontraron diferencias significativas al analizar los datos en su conjunto. Algunas diferencias son más evidentes en pequeños grupos de elementos y con referencia a los resultados de la prueba final.

A través del análisis cualitativo, realizado gracias a la observación no participante y a la información recopilada con la entrevista, se buscó completar la investigación respondiendo también a las preguntas relacionadas con el esfuerzo cognitivo y motivacional en geometría, al desarrollo de las relaciones sociales dentro del grupo de clase, al rol de la docente y a su formación, otras piezas importantes en el proceso de enseñanza/aprendizaje.

Los datos cualitativos recopilados, con respecto a los cuantitativos, y con referencia también al factor emotivo, muestran una diferencia más evidente entre los grupos a favor de la experiencia de *coding* realizada que da mayor espacio a factores como el sentimiento de satisfacción, la alegría y la diversión, la motivación para aprender y el interés. Aun registrando, a través del cuestionario, una buena activación de emociones positivas en ambas clases, la

observación muestra que la diferencia aumenta a favor del grupo experimental. También el rol del docente en el proceso didáctico resulta determinante con respecto a las relaciones que de este se derivan y a su formación en el tema del *coding*.

Por lo tanto, el trabajo de análisis de datos tuvo como objetivo la recogida de datos para su posterior discusión para así poder tener una visión general de todo el proceso de enseñanza/aprendizaje llevado a cabo en los dos grupos.

En el siguiente capítulo se detalla el análisis crítico y sus respectivas observaciones sobre los datos recogidos especificando las fortalezas y debilidades, los límites e implicaciones, y las perspectivas futuras del presente trabajo.

Capítulo 5

Discusión y conclusiones

5. Introducción

En este trabajo se utilizó un enfoque tipo mixto (complementariedad metodológica cuantitativo-cualitativo) (Pérez, 2011) y que puede catalogarse dentro de los estudios cuasi-experimental con grupo control no equivalente. El objetivo fue explorar los efectos del uso del *coding* como herramienta metodológica en la enseñanza de la geometría y en el aprendizaje de los estudiantes. En el primer capítulo se definió la problemática objeto de estudio, describiendo el contexto de referencia; en el segundo capítulo se delimitó el marco teórico con referencia al uso del *coding* como herramienta para la promoción del pensamiento computacional y de las habilidades digitales así como el estado actual de los estudios realizados; en el tercer capítulo se describió la metodología elegida para abordar el problema tomando como referencia estudios similares previos; en el cuarto capítulo se presentaron los resultados obtenidos a partir de la aplicación del método y, finalmente, en este capítulo, en base a los estudios teóricos y empíricos previamente descritos, se procederá a interpretar los principales resultados con el fin de llegar a las conclusiones identificando límites, implicaciones y recomendaciones para futuras investigaciones.

La presente tesis buscó abordar la temática del uso del *coding* dentro de las actividades educativas y didácticas. El trabajo se basó en la exigencia de comprender cómo es posible trabajar con dicha metodología integrándola en los procesos de enseñanza de las disciplinas de Educación Primaria. La pregunta general de investigación ha sido si el uso del *coding* puede influenciar, por un lado la experiencia educativa y escolar de los alumnos y, por el otro, el proceso de enseñanza de los docentes. Además de a nivel estrictamente didáctico, se

investigaron los efectos del uso de esta metodología sobre el aprendizaje, sobre el esfuerzo cognitivo y sobre los resultados alcanzados, teniendo en cuenta también el aspecto emotivo, motivacional y relacional.

La discusión sobre los resultados obtenidos del presente estudio se estructura en base a algunas preguntas principales que han guiado la investigación y buscan verificar si el *coding* mejora los resultados del aprendizaje en geometría y cómo modifica las dinámicas sociales, las relaciones entre los compañeros y entre el alumno y el docente. Se investigaron, también, las emociones que surgieron durante el proceso de enseñanza/aprendizaje en los estudiantes y si ellos se sintieron involucrados en el desarrollo de las actividades, si aumenta su interés, su grado de satisfacción, su motivación para aprender la geometría. Se cuestionó también, cuál es el rol del docente como activador del proceso de enseñanza/aprendizaje.

5.1. Enseñar y aprender la geometría con Scratch

Las actividades didácticas relacionadas con el estudio de las matemáticas en general, y por lo tanto también de la geometría, pueden tener algunos momentos críticos; especialmente cuando se les pide a los estudiantes que desarrollen conceptos espaciales que incluyen las operaciones de traslación, rotación y vuelco, y que hagan una abstracción inmediata. En realidad, según Papert (1980) y sus seguidores, las matemáticas del espacio, del movimiento, de los modelos repetitivos de acción, es lo más natural para los niños. Por lo tanto, el abordar dichos temas a través de actividades de *coding*, correctamente moduladas, permite construir un ambiente didáctico que tiene el potencial para responder a las características típicas de la disciplina (Benton et al., 2017).

Scratch es un software didáctico muy utilizado para enseñar el *coding* y fue elegido en este estudio, como ambiente de aprendizaje para los niños de Educación Primaria. Gracias a su potencial (Resnick et al, 2009) que permite tratar conceptos matemáticos y computacionales, da espacio a la creatividad, al razonamiento matemático y al trabajo en equipo (Benton et al., 2017). Después de haber estudiado y diseñado las características de la intervención curricular a través del *coding* se procedió a aplicarla de forma experimental.

La intervención didáctica que se puso en marcha en una de las dos clases seleccionadas (grupo experimental), en buena parte, se llevó a cabo en un ambiente físico de aprendizaje distinto al aula donde los niños normalmente realizan la mayor parte de sus actividades didácticas; se trata del laboratorio de informática, que es fácilmente accesible para los estudiantes, pues se encuentra en la sala contigua al aula.

A través de la plataforma Scratch se ofreció a los estudiantes la posibilidad de personalizar su experiencia de aprendizaje en geometría, permitiéndoles

ampliar sus propios conocimientos en una forma distinta a la tradicional. Los niños realizaron una actividad con características lúdicas y, mientras jugaban, se les dio una retroalimentación visual instantánea, que es parte fundamental del *debug* de su proceso de aprendizaje, ciertamente distinto al tradicional que se realiza en clase; de hecho, de esta forma, pudieron realizar una verificación inmediata del trabajo realizado para reconsiderar la situación y corregir sus errores (Resnick et al., 2009; Manilla, Dagiene et al, 2014; Kong & Abelson, 2019).

Las observaciones realizadas mostraron que, durante el trabajo con Scratch, el ambiente en la clase fué muy positivo y el porcentaje de niños concentrados en la tarea fué muy alto; en general, todos contribuyeron al trabajo en los pequeños grupos, incluso los estudiantes con más dificultades y aquellos que, en las actividades en el aula, se mostraron desinteresados o particularmente distraídos (Wilson & Moffat, 2010).

El impacto que tuvieron los alumnos con Scratch, durante la hora de geometría fue, en general, positivo con respecto a las emociones de logro que los estudiantes manifestaron de forma evidente; tanto antes de iniciar las actividades de *coding* como al completar el trabajo: lograr resolver los problemas planteados verificando en la pantalla que las instrucciones dadas fuesen correctas, los llevó a animarse, alegrarse e interactuar aplaudiendo y abrazándose; satisfechos de trabajo realizado (Wilson & Moffat, 2010; Zavala et al., 2013). También recibieron las felicitaciones de la maestra que los estimuló a hacerlo cada vez mejor. Además de la satisfacción, surgieron, a partir del análisis cualitativo, otros aspectos individuales como la motivación para aprender (Calao et al., 2015) y la curiosidad y el interés que mostraron al lograr realizar el trabajo con Scratch. Se registró, también, una buena participación activa, describiéndose colaboración tanto entre los miembros individuales de un grupo como entre los propios grupos que interactuaron de manera positiva logrando así, ante los ojos del observador, un ambiente positivo para los componentes sociales y relacionales, factores relacionados con las habilidades

emotivas (Fessakis et al., 2013; Kalelioglu & Gülbahar, 2014; Benton et al., 2018; Corsi et al., 2019).

La clase, durante el desarrollo de la hora de geometría con Scratch, se dividió en grupos de 2/3 niños; esta situación se derivó de la disponibilidad inicial de dispositivos en el laboratorio, pero resultó una ventaja pues estimuló la comunicación entre los niños y la colaboración típica de la didáctica cooperativa. En muy pocos casos, la división del trabajo en pequeños grupos resultó poco eficaz, esto sucedió cuando los estudiantes no lograron gestionar el uso del ratón a través de un acuerdo interno ("Maestra así me aburro, ino me deja usar el ratón!") así que fue necesaria la intervención de la docente. El hecho de haber trabajado en pequeños grupos, incluso con algunas dificultades iniciales en cuanto a la aceptación de la división de los roles, permitió a los niños aprender, colaborando entre sí, y dar su propia contribución si alguien solicitaba ayuda.

Por lo tanto, se puede deducir, que el proyecto desarrollado en torno a la geometría resultó particularmente eficaz en cuanto al crecimiento de las relaciones entre los estudiantes, a la generación de emociones con prevalencia positiva, así como de factores que favorecieron la motivación y la participación a las actividades escolares propuestas. La experiencia realizada a través del *coding* aumentó el nivel de satisfacción de los alumnos y de la misma docente (Wilson & Moffat, 2010) quien reiteró que la metodología favoreció la comprensión de todos los alumnos, especialmente de aquellos más "débiles" que aprendieron más cosas de forma útil, divertida e interesante (Brennan & Resnick, 2012; Kazimoglu et al., 2012).

Hubo un caso en el que se registró un aumento significativo del interés por parte de un niño que, generalmente, en clase no se mostraba muy dispuesto a aprender; al mismo tiempo se percibió una mejora en las relaciones sociales de este mismo niño con sus demás compañeros. Sin embargo, la mejora cognitiva no pudo verificarse con la misma evidencia.

También, hubo solo un niño de la clase que no mostró signos de cambio en cuanto al interés y atención con respecto a las actividades realizadas sin la ayuda de Scratch, se mostró poco disponible para aprender a usar herramientas tecnológicas. Lo expresó claramente con una respuesta a la maestra que lo invitó a trabajar "¡A mí no me interesa, prefiero el campo!".

En cuanto al manejo de la clase y, en particular, el manejo de los tiempos y de la participación, durante las actividades de *coding*, se observó una buena comunicación incentivada por la docente quien guio a los estudiantes para realizar intervenciones espontáneas a través de consideraciones y/o preguntas pertinentes. La maestra trabajó con mayor proximidad hacia los alumnos, caminando entre las estaciones de trabajo y logrando dar a todos la información necesaria para continuar con el trabajo los tranquilizó y les mostró que el error debe verse como una oportunidad de crecimiento y reflexión, como sucede para los informáticos cuando programan (Wing, 2006). El haberles confiado la tarea para trabajar en el ordenador concediéndoles un cierto grado de autonomía suscitó un notable entusiasmo que se manifestó a través del aumento del tono de las voces, un molesto murmullo de fondo, lo cual hizo que la maestra interviniera varias veces ("¡Entiendo el entusiasmo, pero estáis gritando demasiado!"). Estos factores junto con otros relacionados con la esfera de la *diversión* muestran como los mismos niños reconocieron el carácter lúdico de las actividades de *coding* (Brennan & Resnick, 2012).

La maestra relata que el entusiasmo y la euforia de los niños, algunas veces, le hicieron sospechar que estas actividades podrían de alguna manera, haber reducido la calidad de la atención y de escucha. Estas dudas se superaron cuando posteriormente, a pesar de las constantes llamadas para contener el excesivo tono de la voz, los niños mostraron una gran motivación que los llevó a intervenir y a pedir ayuda sin dudar.

De hecho, con la investigación se buscó comprender si el *coding*, además de estimular el desarrollo del pensamiento computacional, permite enseñar la

geometría de forma innovadora, divertida y amena, a través de una metodología más apreciada por los niños, usuarios tecnológicos a los que se les piden cada vez más las habilidades digitales. Se quisieron estudiar los efectos de este método sobre los distintos factores que caracterizan el proceso de aprendizaje de los niños, con referencia a los conocimientos y las habilidades adquiridas tanto en el ámbito matemático como en cuanto a la capacidad de relacionarse con los demás, de pensar y de comunicar.

A partir de estas consideraciones, se intentó combinar el uso de herramientas atribuibles al pensamiento computacional con las metodologías de enseñanza de la geometría, rama y disciplina del área STEM, con el objetivo de comprender si es posible mejorar el proceso de enseñanza/aprendizaje de las matemáticas. Para este fin, se propuso a los niños el uso de Scratch, considerada una herramienta apropiada para su edad y para los fines para los que fue diseñada (Brennan & Resnick, 2012; Kong, 2019), respaldado por los resultados de estudios precedentes que muestran mejoras en los resultados obtenidos en matemáticas después de haber realizado actividades de *coding* (Lye & Koh, 2014; Yadav et al., 2014; Ke, 2014; Foerster, 2016; Hsu & Hu, 2017). Haber utilizado Scratch para aprender geometría lleva a pensar que para los niños hubo una cierta ganancia; tanto en los aspectos motivacionales como en las reacciones emotivas y cognitivas pues se observaron comportamientos positivos y constructivos, intervenciones espontáneas y pertinentes particularmente por parte de quien en general, durante las actividades didácticas normales, interactúa menos. Dicho aspecto se confirmó durante la entrevista a la docente que afirma que vio a los niños “más motivados e involucrados” con respecto a las actividades de rutina.

Con este trabajo, se quiso profundizar la investigación, estudiando los efectos del *coding* en la escuela no solo sobre el rendimiento, sino también sobre los efectos que se producen en todo el proceso que se activa en clase, cuestionando si el uso del *coding* y de Scratch, en particular, puede aportar beneficios al aprendizaje de las matemáticas en Educación Primaria.

Los resultados de las pruebas estadísticas muestran que las puntuaciones en geometría obtenidas por los estudiantes que trabajaron con el método *coding* son mejores con respecto a aquellas obtenidas por el grupo que no llevo a cabo el tratamiento. Por lo tanto, las actividades de *coding*, en apoyo de la geometría, determinaron mejoras significativas en el rendimiento de los alumnos.

Por otro lado, el cuestionario que respondieron los estudiantes de ambas clases, al final de las actividades didácticas, proporciona resultados que no evidencian diferencias estadísticamente significativas entre las emociones manifestadas por los niños que trabajaron con Scratch y aquellos que no trabajaron con esta plataforma. Sentimientos como alegría, diversión, ansiedad, aburrimiento, a partir de un análisis global, parecen surgir de la misma manera entre los estudiantes, independientemente del *coding*. Dichas consideraciones se refieren a los resultados de un análisis cuantitativo global que, en una visión holística, proporciona los resultados descritos. Sin embargo, a partir de un análisis más detallado, surgen diferencias no despreciables, incluso desde el punto de vista estadístico que, en el presente caso, hicieron necesario un análisis cualitativo para completar el estudio.

5.2. Discusión

Los resultados del análisis de los documentos provenientes de las observaciones realizadas, junto a los datos recabados a través de las pruebas y el cuestionario, conducen a centrar la atención en múltiples factores con cierta relevancia para el fenómeno que se desea estudiar.

5.2.1. **Discusión sobre el desempeño y las emociones en geometría**

En cuanto a la parte cuantitativa de la investigación, las respuestas que proporcionaron los estudiantes en la prueba de aprendizaje de geometría (prueba final) permiten deducir que haber integrado el *coding* a una disciplina escolar y haberlo utilizado como enfoque metodológico (Foerster, 2017) tuvo retroalimentaciones positivas para los niños del grupo experimental. No es posible, llegar a las mismas conclusiones en cuanto a los resultados del cuestionario (*AEQ-ES*) ya que no se observaron diferencias estadísticamente significativas entre los dos grupos de niños; las herramientas de medición utilizadas dan como resultado una cierta *resistencia* a la metodología *coding* en cuanto a las emociones alcanzadas por el grupo experimental que parecen estar en consonancia con las respuestas dadas por los niños del grupo control.

Queriendo profundizar en el estudio de las respuestas proporcionadas a través de un procedimiento deductivo, fue posible verificar que, en realidad, para algunos elementos, se registran diferencias no despreciables que requirieron mayores investigaciones y análisis. Esta consideración puede extenderse en particular a los resultados del cuestionario *AEQ-ES*; para algunos elementos se registraron diferencias estadísticamente significativas, las cuales, confirman que existe una relación entre la capacidad de programar y el aprendizaje de conceptos matemáticos y geométricos (Benton et al., 2017; Miller, 2019) a través de experiencias agradables y colaborativas (Sáez-López et al., 2016). Las habilidades que se consiguen no están relacionadas exclusivamente al desarrollo del pensamiento computacional y muestran como los estudiantes pueden realizar abstracciones trabajando con Scratch, y por lo tanto, con dispositivos tecnológicos para explorar conceptos matemáticos ya desde Educación Primaria.

5.2.2. **Discusión sobre las emociones, la motivación y las relaciones sociales en geometría**

Las emociones, la motivación y las relaciones sociales son factores que intervienen en el proceso de aprendizaje de los estudiantes y han sido objeto de análisis cualitativo en esta tesis. Estudios previos, aunque no han sido numerosos, describen experiencias positivas y gratificantes para grupos de alumnos que parecen apreciar las actividades de *coding* y, en particular, de Scratch durante las actividades didácticas curriculares. Como ya se ha mencionado anteriormente, se buscó analizar el impacto de la integración del *coding* a una disciplina del área STEM en una escuela de Educación Primaria del sur de Italia, donde los datos INVALSI muestran en matemáticas, puntuaciones promedio más bajas con respecto a las escuelas del norte.

Los resultados del análisis cuantitativo realizado a partir de los datos provenientes del cuestionario *AEQ-ES* aplicado a todos los estudiantes ($n=46$) al final de las actividades didácticas, no muestran una influencia estadísticamente significativa sobre las emociones de logro; por lo tanto, el *coding* no hizo surgir emociones distintas con respecto a las que experimentaron los niños que no recibieron el tratamiento. Pero, incluso si los resultados del cuestionario muestran para ambos grupos de niños, en general, diferencias no significativas en cuanto a las emociones de logro, un análisis más profundo de las categorías y grupos de elementos hace surgir resultados diferentes que necesitaron una mayor profundización.

Así pues, si se extrapolan los datos relativos a las emociones individuales surgen diferencias estadísticamente significativas ($p<.05$) con exclusiva referencia a las emociones alegría-diversión. Este dato se confirma con el análisis de las frecuencias de los códigos utilizados en el examen de los documentos de observación donde justamente emociones como *alegría* y *diversión* se registraron con cierta regularidad en la clase experimental y en cambio resultaron poco presentes o incluso ausentes en el grupo control.

A través del análisis cualitativo fue posible obtener los datos que surgieron durante el desarrollo de las actividades educativas en ambas clases. En dicho análisis, realizado con el software *MAXQDA* y utilizando los datos de la observación, se destacó la presencia constante, durante las sesiones de *coding*, de componentes relacionados con las emociones de valencia positiva como la satisfacción, la alegría, la diversión, las cuales se encontraron con menor frecuencia en el grupo que no trabajó con Scratch. De esta manera, fue posible conocer y describir, de la forma más objetiva posible, aspectos particulares y específicos del proceso de enseñanza y aprendizaje en las clases en las que se realizó la observación, como se describe detalladamente en el capítulo 3. El proceso de codificación permitió primero detectar y después medir las características y la frecuencia con la que determinados comportamientos y situaciones se registraron, permitiendo así efectuar una cuantificación y asociarla a un análisis cualitativo.

Durante el desarrollo de las actividades de *coding* se registró un alto nivel de participación emotiva, cognitiva y motivacional, lo cual confirma lo que se ha registrado en trabajos previos (Benton et al., 2018). De hecho, los participantes se mostraron bastante motivados y entusiasmados para programar y realizar las actividades de geometría; intercambiaron ideas, colaboraron para llegar a la solución, compartieron los errores y los corrigieron; disminuyó el nivel de falta de atención, aunque el tipo de actividad hizo que se registrara un menor orden en el aula. Varias veces se observaron reacciones relacionadas con la alegría y la satisfacción de haber realizado un buen trabajo, estos factores permitieron iniciar discusiones sobre las decisiones que debían tomarse. Los estudiantes trabajaron con esfuerzo en pequeños grupos, y para alcanzar sus objetivos, a menudo buscaron la ayuda de la maestra.

En el laboratorio, disminuyeron los episodios de interacción que surgieron por iniciativa de la docente, tanto que presentaron una frecuencia menor con respecto a aquellos que se registraron en el aula. También aumentaron los episodios de alegría, diversión y de satisfacción. Los indicadores relacionados

con las emociones negativas de ansiedad y preocupación, desilusión, aburrimiento y enojo se encontraron con mayor frecuencia en el grupo control.

Las experiencias de aprendizaje conseguidas por los estudiantes de la clase experimental resultaron ser en su conjunto, cualitativamente muy positivas, lo que permite deducir que, desde el punto de vista emocional, prevalecen los factores relacionados con la categoría de emociones que pueden evaluarse como positivas con referencia particular a las actividades de *coding*. Además, estas sesiones hicieron surgir de forma evidente, aspectos relacionados con la colaboración entre estudiantes y a sus habilidades socio-relacionales.

Los efectos de las emociones con prevalencia negativa parecen reducirse a la transición de las actividades tradicionales al *coding*, dentro del mismo grupo, y también en la comparación entre ambos grupos. Las dificultades que se presentaron se enfrentaron con un mayor espíritu de iniciativa, activando estrategias diferentes como pedir ayuda a la maestra o a los compañeros de clase, realizar varios intentos y aceptar el error como momento de crecimiento.

Los resultados obtenidos están en línea con los obtenidos en otros trabajos (Wilson & Moffat, 2010; Fessakis, 2013; Kalelioglu, 2014; Benton et al., 2018; Corsi et al., 2019; Miller, 2019) en los cuales se reportan experiencias gratificantes, placenteras y divertidas que han estimulado el aprendizaje y la motivación de los estudiantes, exaltando el potencial educativo de la intervención también en lo que respecta a el desarrollo de las relaciones sociales y al conseguimiento de las habilidades correspondientes.

En concreto, la actividad de trabajo propuesta, en línea con las conclusiones de Fessakis (2013) y Wilson & Moffat (2010), ha implicado a los niños que han aprendido los conceptos de la geometría, han participado activamente divirtiéndose ante los problemas planteados; han desplegado

estrategias de comunicación y colaboración gracias a la capacidad del docente de orquestar el aprendizaje.

Los resultados obtenidos están en consonancia con la afirmación de Miller (2019) y de Benton et al. (2018) de que las actividades de *coding* tienen un alto potencial para ofrecer a los alumnos un entorno en el que desarrollar y conseguir las estructuras matemáticas básicas.

Al mismo tiempo, el interés por el uso de Scratch ha hecho que el aprendizaje de la geometría resulte una experiencia positiva y diferente de la lección tradicional habitual, como confirman en los estudios Wilson y Moffat (2010) y Kalelioglu (2014).

La docente tuvo un rol determinante en todo el proceso ya que implementó el *coding* en la disciplina escolar cuidando los aspectos didácticos, pero sin descuidar la idea de inculcar estrategias, pensamiento crítico, capacidad para la resolución de problemas complejos, elementos típicos del pensamiento computacional. Ella misma sostiene, en la entrevista, que abordó temas y conceptos que de otra manera no habría tenido manera de proponerlos y notó no solo un cierto entusiasmo entre los niños sino también una mejor respuesta oportuna y una mayor participación:

...los niños en clase necesitan varias llamadas de atención en cambio ahí(laboratorio)estaban seguramente más involucrados

y

durante la lección tradicional, muchos niños se aburren y pierden el sentido del tema, en cambio durante las actividades de *coding* los niños se sentían siempre atraídos a la actividad y, a menudo, me daba cuenta, analizando la situación, que aquellos que en clase se distraían y no entendían el tema, ahí lograban retomar el hilo y entender el tema.

Así pues, los estudiantes que generalmente presentan más dificultad en el aprendizaje y se esfuerzan menos a nivel emotivo y motivacional mostraron, durante las actividades de *coding*, mayor esfuerzo e interés; los datos elaborados muestran como algunos estudiantes que en clase se muestran más débiles, participaron con intervenciones pertinentes y se involucraron demostrando ser conscientes de aquello que habían aprendido.

Es evidente que las emociones positivas están principalmente asociadas a las experiencias en laboratorio con Scratch, por lo que también se registró un aumento en el interés y en la curiosidad y una disminución de los episodios de aburrimiento y distracción, todos estos elementos exaltan el potencial del *coding*.

En resumen, se puede deducir que los niños (de edades entre 9 y 10 años) muestran una buena predisposición y aceptación de las actividades didácticas que utilizan la programación visual por bloques, celebran su carácter lúdico y divertido que genera una mayor participación incluso por parte de aquellos niños que, durante las lecciones tradicionales, presentan mayores dificultades. Además, el *coding* representa una especie de enfoque integral que une las matemáticas y la tecnología (Miller, 2019) dos campos del saber que teniendo tantas cosas en común, deberían caminar juntos incluso en la planeación didáctica de las actividades curriculares. Por lo tanto, la eficacia del *coding*, como estrategia pedagógica, depende de una adecuada planeación didáctica y de la capacidad de los docentes para integrarla en su proceso de enseñanza.

En conclusión, se considera interesante, no descuidar en el proceso de aprendizaje de los estudiantes, el desarrollo de emociones con prevalencia positiva, como la alegría, la diversión y la diversión, pues ciertamente no son menos importantes que los resultados del desempeño alcanzados por un niño en la disciplina de estudio.

5.2.3. Discusión sobre actividades, metodologías y estrategias utilizadas por los docentes

Parte del análisis cualitativo se centró en el estudio del proceso de enseñanza que implementaron las docentes de ambas clases, con referencia a las actividades realizadas, a la metodología utilizada, al estilo de enseñanza y a las estrategias escogidas. El análisis tuvo como objetivo no tanto el realizar una comparación entre los métodos utilizados y establecer cuál era el más eficaz sino más bien observar directamente en el campo las formas de implementación del método *coding* y conocer la perspectiva de la docente frente a esta experiencia.

A partir de las observaciones realizadas surgió que la maestra del grupo control utilizó una método de enseñanza principalmente de transmisión aunque hubo ocasiones en las que la lección se volvió participativa justamente porque, especialmente durante la explicación, se logró establecer una relación con los estudiantes presentes a través de preguntas y solicitudes de intervención desde sus pupitres o en la pizarra; esto sucedió en distintos momentos, tanto al inicio de la hora como punto de partida para las actividades, como durante toda la lección. La docente decidió los tiempos, las herramientas y el trabajo a realizar; el libro de texto representó una guía de referencia tanto para la explicación de la clase como para la asignación y el desarrollo de las tareas en clase y para la casa. La docente supervisó todo el proceso de aprendizaje, reprendió a quien estaba distraído, estimuló a quien no quería trabajar, felicitó a quien realizó correctamente el trabajo, alentó a quien se mostró inseguro, se molestó cuando alguien se comportó mal (Anexo C).

La docente del grupo experimental trabajó implementando las actividades de *coding* dentro de un ambiente más tradicional, a través de estrategias de organización y llevando la clase con un enfoque constructivista; de esta

manera, los niños construyeron el conocimiento entrelazando la información proveniente de un entorno familiar para ellos con aquella obtenida a través de la experiencia de *coding*. La docente comenzó la lección a través de recordatorios y preguntas de estímulo, estableciendo los objetivos a alcanzar al final de la hora y presentando los nuevos temas en pequeños pasos con retroalimentaciones continuas que acompañaron el proceso de aprendizaje hasta el final de la lección cuando se destacan los elementos más importantes a considerar en el estudio. Se constató un dialogo continuo entre profesor y alumno, un intercambio de conocimientos en el que el docente buscó estimular la discusión y la participación de todos los miembros del grupo. Para lograr esto, a menudo repitió los conceptos, los puntualizó, estimuló a los alumnos utilizando algunas estrategias para la resolución de problemas, ayudó a quien se encontraba en dificultad, alzó el tono de la voz para reprender a quién se comportaba mal, intervino para resolver problemas entre los compañeros (Anexo C).

Como ya se ha especificado anteriormente, para el grupo experimental, la construcción de los conocimientos se llevó a cabo en dos ambientes de aprendizaje distintos, ya sea en cuanto a lugares físicos (disposición de los pupitres, ubicación de los niños), que en cuanto a situaciones mentales donde se requirió un esfuerzo cognitivo, emocional y relacional distinto; los mismos niños fueron conscientes de ello y se mostraron totalmente de acuerdo cuando se les pidió que se trasladaran al laboratorio de multimedia. La docente, para el trabajo en clase utilizó herramientas tecnológicas como la LIM y herramientas visuales para comunicar de forma más eficaz con sus estudiantes. Utilizó el libro de texto raramente y solo para la asignación de tareas para la casa. Propuso experiencias en el laboratorio a través del *coding*, alentando el aprendizaje colaborativo y basado en la experiencia, activando procesos y proporcionando estímulos que involucraron un alto número de alumnos en el aprendizaje de la geometría.

Los datos recogidos confirman una cierta diferencia entre las dos actividades propuestas, en particular con referencia a las relaciones entre docente y alumno, a las actividades y a las estrategias utilizadas. Mientras que en el grupo control prevalecen las intervenciones de la maestra típicas de una lección frontal, en el grupo experimental la docente propone estrategias de trabajo alternativas y deja una mayor independencia para el desarrollo del trabajo asignado. Estudios previos han analizado el rol del docente dentro del grupo de clase que realiza actividades de *coding* estudiando cómo se lleva a cabo la implementación de la intervención (Benton et al., 2017) y cómo influye sobre su experiencia el incluir el *coding* en la enseñanza de una disciplina escolar (Benton et al., 2018) mostrando como una correcta planeación didáctica y pedagógica puede llevar a los estudiantes a expresar ideas matemáticas importantes a través del *coding*. De hecho, como sostienen Kalelioglu & Gulbahar (2014) no es suficiente proporcionar a los estudiantes ambientes adecuados para el aprendizaje, sino que además se deben incluir los enfoques correctos, las buenas prácticas, los recursos oportunos estrechamente relacionados con una experiencia en el campo.

La docente del grupo experimental, al llevar a la práctica las actividades de geometría a través del *coding*, tuvo la tarea de presentar los conceptos geométricos a través de Scratch creando una conexión entre la programación informática, tecnológica y geometría. Fue necesario guiar a la mayor parte de los alumnos a través de un apoyo constante buscando el razonamiento y las mejores estrategias para llegar a la solución del problema ya que solo algunos niños trabajaron en autonomía. La docente utilizó varias técnicas y herramientas, haciendo uso también del *coding* unplugged, por ejemplo, a través del uso del movimiento del cuerpo que simula los pasos del *sprite* de Scratch. Una situación similar se dio en el estudio de Benton et al. (2017) en el que el docente que adoptó esta misma estrategia era quien tenía una menor experiencia en el uso de Scratch en comparación con aquellos que ya lo habían usado por más tiempo. Se deduce que el docente adaptó las actividades a su estilo de enseñanza y a su experiencia, así como a las exigencias de sus

alumnos manteniéndose fiel a los objetivos establecidos. Las observaciones y la entrevista confirman la idea de Miller (2019) según la cual el rol del docente juega un papel fundamental en el experimento didáctico, es la guía hacia la comprensión de los conceptos geométricos y de las instrucciones necesarias para la programación visual, alienta a los estudiantes a hacer razonamientos, a identificar los errores y a realizar las debidas correcciones y a compartir el trabajo con sus compañeros.

Como en el estudio de Wilson & Moffat (2010) la entrevista a la docente que aplicó el *coding* como intervención didáctica fue útil para estudiar el punto de vista de quien conoce bien su clase, ya sea con respecto a los aprendizajes como con respecto al aspecto emotivo y relacional. Por este motivo, se le pidió emitir un juicio sobre la experiencia realizada. A partir de las respuestas proporcionadas surgen varias similitudes con respecto al entusiasmo y a la diversión de los niños al trabajar con Scratch, así como al interés que mostraron los niños con mayores dificultades.

5.3. Conclusiones e implicaciones del uso del *coding* en la didáctica

El análisis de la efectividad de una intervención didáctica mediada por el *coding* fue el objeto principal del presente trabajo. Se identificó y se adaptó un plan curricular de matemáticas en Educación Primaria, a través del cual poder analizar las habilidades transversales que surgieran (emotivas, sociales, de pensamiento computacional y digitales), así como habilidades propias de la disciplina, respondiendo a la creciente demanda de los gobiernos nacionales y de la política europea, que, en los últimos años, han hecho hincapié en el uso del *coding* y en el estudio de las materias STEM.

Se decidió realizar este estudio en una escuela de Educación Primaria del sur de Italia, donde los resultados conseguidos por los estudiantes en las

pruebas INVALSI en matemáticas, muestran puntuaciones promedio inferiores con respecto a las escuelas del norte, disminuyendo así el promedio nacional (INVALSI, 2018). Así pues, se consideró interesante conocer el efecto del *coding* sobre la percepción de estudiantes y profesores y su impacto en el proceso de enseñanza y de aprendizaje de la geometría.

Con respecto a los resultados obtenidos se puede concluir que, analizando la parte del proceso de aprendizaje relacionada con los resultados obtenidos, se encontraron repercusiones positivas en los niños que estudiaron la geometría a través del *coding*; y no se encontraron diferencias en los resultados al comparar entre niños y niñas en ambos grupos.

Con referencia a los componentes emotivos, motivacionales y sociales asociados al proceso de aprendizaje, el uso de Scratch y de la programación visual por bloques, en el estudio de la geometría, no produjo de manera general diferencias estadísticamente significativas sobre las emociones de logro para los niños que tuvieron la intervención en comparación con aquellos que siguieron las lecciones tradicionales. Las diferencias entre ambos grupos surgieron al restringir los campos de investigación y analizando las subescalas y los elementos individuales del cuestionario. En cambio, el análisis cualitativo muestra como la experiencia de enseñanza con Scratch generó un mayor esfuerzo especialmente entre los niños que, normalmente, están más distraídos y menos interesados. Se registraron niveles más altos de esfuerzo y de entusiasmo, experiencias placenteras y divertidas que estimularon el aprendizaje y la motivación.

La docente, a través de su estilo personal de enseñanza y su práctica didáctica, integró el *coding* en el estudio de la geometría, poniendo en evidencia la importancia de su rol al organizar dicha integración; la experiencia de enseñanza muestra cuán importante es la capacidad de saber equilibrar el componente lúdico de Scratch con el aprendizaje de los conceptos matemáticos, el completar el proceso de enseñanza y de aprendizaje

estimulando a los alumnos para la adquisición de las *life skill* que los acerquen cada vez más a la realidad social de nuestro siglo. Por lo tanto, se deduce que la presencia de una figura profesional y con experiencia en el campo del *coding* además de en el ámbito de la enseñanza de la disciplina, es de vital importancia. De hecho, estudios previos han resaltado su importancia ya que gracias a los docentes se generan las conexiones entre los conocimientos matemáticos y las respectivas experiencias de los estudiantes; son ellos quienes facilitan el descubrimiento y la comprensión de las ideas que ofrecen los ambientes de programación como Scratch (Benton et al., 2017; Rich et al., 2021).

Los resultados de esta investigación contribuyen al conocimiento de las prácticas didácticas que utilizan el *coding* y Scratch en Educación Primaria y, por consecuencia, abren una discusión con respecto a algunos factores:

- Es importante tener en cuenta todo el proceso de aprendizaje de los estudiantes, además del proceso cognitivo, también el relacionado con el desarrollo de las habilidades relacionales y sociales, a la generación de emociones con valencia positiva que pueden hacer placentera y gratificante la experiencia didáctica;
- Asume importancia el uso de las TIC no tanto como medios de transmisión del saber sino como herramientas metodológicas que pueden facilitar el proceso de aprendizaje y favorecer el desarrollo de las habilidades digitales, en particular en las disciplinas STEM;
- Es necesario proporcionar a todos los docentes una adecuada formación profesional sobre los métodos, herramientas y recursos que pueden consentirles alcanzar buenos niveles de preparación para la gestión de alumnos con actitudes diferentes y para contribuir en la innovación de los planes curriculares didácticos.

5.4. Recomendaciones para futuras investigaciones

A partir de los resultados obtenidos en esta tesis doctoral se proporcionan algunas recomendaciones útiles para los posibles trabajos futuros que tengan como objeto la integración del *coding* como instrumento de mediación didáctica para el aprendizaje disciplinar tanto en las escuelas primarias como en otros contextos.

La investigación en este campo debería profundizar en el estudio de los efectos del *coding* como herramienta metodológica para el aprendizaje de las matemáticas y de la geometría con particular referencia tanto al rendimiento de los estudiantes como a su potencial de inclusión en las clases. Los resultados obtenidos a partir de los estudios realizados sobre los beneficios en el desempeño de los estudiantes no son suficientes para poder llegar a conclusiones inequívocas (Wilson & Moffat, 2010; Zavala et al., 2013; Calao et al., 2015; Psycharis & Kallia, 2017; Miller, 2019).

Futuras investigaciones deberían tener como objeto comprender mejor cómo implementar el *coding* como estrategia didáctica dentro de las disciplinas escolares en los distintos niveles de educación (K-12) (Angeli & Giannakos, 2020).

También los posibles estudios futuros deberían tener en cuenta el muestreo de otros grupos pertenecientes a diversas escuelas extendiendo así la investigación a un mayor grupo de participantes. Así pues, se considera necesario profundizar en cuanto a la duración del tratamiento con el *coding*, para comprender si el factor tiempo puede influir de forma determinante sobre el rendimiento de los estudiantes y tenerlo en cuenta en las investigaciones futuras. De esta manera, se espera poder analizar los resultados obtenidos por los estudiantes en las pruebas de aprendizaje después de haber trabajado con el *coding* durante un largo periodo, por ejemplo, todo el curso escolar.

Trabajar la geometría a través del *coding* permite desarrollar el pensamiento matemático a través de métodos diferentes a los utilizados en las lecciones tradicionales (Miller & Larkin, 2017) que conducen a experiencias más gratificantes y potencialmente más completas; permite el desarrollo de habilidades transversales como la digital, que es fundamental en la formación del ciudadano del siglo XXI (Miller, 2019). Al respecto, no fue posible medir cuánto aumentaron las habilidades digitales con el uso de Scratch, pero es evidente que muchos niños tuvieron la posibilidad de aprender sobre funciones que antes no conocían. En este sentido, las futuras investigaciones, podrían abordar este tema.

Reconocida, en el panorama político y de la investigación científica, la importancia del pensamiento computacional en el ambiente educativo, sería oportuno investigar con detalle los métodos de formación que reciben los docentes para poner en práctica actividades de promoción (cómo, cuándo y por qué) y de desarrollo de las habilidades asociadas para verificar si estas iniciativas son eficaces. Esta consideración también se encuentra en los estudios de Tang et al. (2020) y Angeli & Giannakos (2020), que consideran esencial una formación adecuada de los profesores para integrar las prácticas de pensamiento computacional en sus respectivos contextos de enseñanza. La experiencia de trabajo del docente, directamente en campo, también puede ser una variable interesante para tomar en cuenta con respecto a, por ejemplo, los resultados alcanzados por los alumnos.

El estudio también se abre a la promoción del uso de las herramientas tecnológicas en la didáctica (TIC) en modo tal de proponer la informática a los estudiantes no en contextos separados sino de forma paralela al estudio de una disciplina escolar. Por lo tanto, teniendo en cuenta que los argumentos que se refieren al pensamiento computacional, a la práctica del *coding* su aplicación a través de herramientas tecnológicas (TIC), han comenzado a recibir atención solo en los últimos años; es necesario, para comprender mejor este fenómeno, desarrollar y mejorar los estudios al respecto tomando en cuenta los múltiples

aspectos que intervienen. Por este motivo es importante ampliar el conocimiento teórico y empírico en cuanto a la investigación del tema del desarrollo del pensamiento computacional en los contextos escolares a través de las TIC y fuera de ellas, por ejemplo, evaluando la percepción que los mismos padres de familia tienen del *coding* y de su uso en la escuela, pues también son ellos protagonistas del proceso educativo de sus hijos.

Muchos investigadores han tratado este tema a través de enfoques metodológicos distintos, de tipo cuantitativo, cualitativo o estudios de caso, y han presentado sus resultados contribuyendo a la investigación para comprender mejor el fenómeno que se estudia. Los instrumentos utilizados como entrevistas, cuestionarios y observaciones han permitido recoger los datos que, en algunos casos, han resaltado como las respuestas de un cuestionario pueden no corresponder a lo observado en la realidad. En este sentido, otros proyectos *cuasi experimentales* podrán contribuir a dar validez y credibilidad en cuanto a la efectividad de las prácticas didácticas asociadas a la enseñanza de las disciplinas escolares a través del *coding*.

5.5. Limitaciones y aportaciones de la investigación

Los resultados del presente estudio deben interpretarse con cierta cautela. En primer lugar, es necesario tener en cuenta el tipo de muestreo realizado y las características de la muestra que, así como en investigaciones similares (véase capítulo 3), presenta un tamaño reducido por lo que la interpretación de resultados debe realizarse teniendo en cuenta esta limitación, así como los distintos factores que intervinieron en todo el proceso de la investigación.

Los resultados obtenidos a partir de la aplicación de la metodología a la muestra tal como fue definida, por sus características, no permiten una generalización, pero ciertamente pueden trasladarse a contextos comparables al contexto en el que se llevó a cabo el presente estudio, sin descuidar las condiciones para la recogida de datos y las características del muestreo

utilizado. Tratándose de una investigación en el campo educativo, la generalización de los resultados puede realizarse cuando las características de la muestra de referencia sean similares a las de toda la población; en este caso, la muestra se refiere a dos clases seleccionadas en una escuela de Educación Primaria siciliana de la provincia de Agrigento, un área geográfica pequeña y no necesariamente representativa de todo el territorio nacional italiano. Por lo tanto, como para los estudios similares, se trata de una intervención que no permite analizar cómo se gestiona el tiempo, cómo se abordan los distintos aspectos y conceptos geométricos y cómo, con el tiempo, influyen sobre el nivel de aprendizaje de los alumnos (Benton et al., 2018). Para aumentar la credibilidad y la fiabilidad de los resultados es necesario realizar otros estudios con muestras que sean representativas de la población de estudiantes de Educación Primaria.

Como se ha mencionado anteriormente, la aplicación de la metodología se realizó con referencia a una sola actividad didáctica durante un periodo de tiempo no excesivamente largo. El breve tiempo de exposición al tratamiento, muy probablemente, condicionó los resultados de las pruebas (Zavala et al., 2013) y en el caso concreto del cuestionario. Sin embargo, si se analizan con detenimiento los resultados de la prueba final para los dos grupos y más a detalle los resultados del cuestionario, es posible notar algunas variaciones a favor del grupo experimental gracias al uso del *coding*. Para las futuras investigaciones, es recomendable considerar un tiempo de exposición más largo, para poder evaluar los posibles cambios y/o beneficios a largo plazo y tener más datos de observación para poder analizarlos cualitativamente.

Una limitación de la investigación está ligada a las características autorreferenciales del cuestionario, por lo que no es posible establecer con certeza absoluta cuán sinceras son las respuestas dadas; los encuestados, siendo niños de entre 9 y 10 años, pueden haber proporcionado la respuesta que se espera de ellos o la más apreciada, por ejemplo, por la maestra, incluso si no corresponde completamente a la verdad. El cuestionario, en el presente

caso, se completó en horas escolares en presencia de la docente y del investigador por lo que no se excluye una cierta influencia sobre los niños incluso cuando ellos estaban conscientes de su anonimato.

Otro límite de la investigación está relacionado con el hecho de que la codificación de las distintas situaciones de aprendizaje y, en particular, de las emociones, se realizó tomando en cuenta solo una parte de los factores que pueden intervenir en el proceso de aprendizaje, sin tener en cuenta la amplia gama de variaciones y las distintas facetas que pueden intervenir. En trabajos futuros, pudieran tenerse en cuenta otras variables relacionadas, por ejemplo, a la exposición de los niños a las TIC tanto en ámbito escolar como familiar o al conocimiento/resistencia sobre el tema *coding* por parte de los padres.

De cualquier forma, aun teniendo en cuenta las consideraciones hechas anteriormente, los resultados de esta tesis doctoral representan una aportación al estudio de los efectos de las prácticas relacionadas con el pensamiento computacional en el proceso de enseñanza y de aprendizaje de una disciplina escolar con respecto al rendimiento, las emociones de logro, las habilidades sociales, la motivación y el interés. Si por un lado los resultados no pueden generalizarse ni son representativos de toda la población de estudiantes, por otro lado, el número reducido de participantes permitió completar la investigación con un análisis cualitativo complementario, a través del cual fue posible individuar aspectos y relaciones que, de otra manera, solo con el análisis cuantitativo, no habría sido posible.

La investigación hizo que surgiera un dato importante con respecto al rol del docente en el aula (Kafai & Proctor, 2021), a las formas en las que presenta la actividad didáctica, las estrategias que utiliza y a su práctica educativa en general, verificando que la experiencia y el conocimiento profundo de un argumento o un tema pueden influenciar todo el proceso de enseñanza y de aprendizaje y representan condiciones *sine qua non* para poder activar cualquier proyecto innovador dentro de un aula escolar.

5.6. Conclusiones

Aunque la investigación en el campo del conocimiento investigado (pensamiento computacional, codificación, áreas STEM) está todavía en sus inicios, ha habido un considerable interés por parte del mundo de la investigación científica en los temas tratados, que están creciendo rápidamente. Ciertamente, la pandemia de COVID-19 no ha favorecido este tipo de investigación, que requiere una intervención y una acción sobre el terreno, pero a pesar de todo, se ha comprobado que la atención del mundo educativo y de la formación sigue siendo muy alta.

Se ha entendido que las experiencias de los años setenta y ochenta han sido reconsideradas desde una nueva perspectiva que ahora es dominante, a saber, como un componente esencial del pensamiento computacional con importantes repercusiones en los procesos educativos.

Resumiendo, la presente tesis doctoral y los resultados obtenidos han demostrado el potencial que ofrece el *coding* como herramienta para promover y desarrollar el pensamiento computacional y el uso de la plataforma Scratch en la enseñanza de la geometría con atención particular a algunos factores del proceso de enseñanza y aprendizaje como el rendimiento, la colaboración entre los estudiantes, la motivación, la activación de emociones con prevalencia positiva (Benton et al., 2018; Ke, 2014; Psycaris & Kallia, 2017).

El *coding* permite desarrollar propuestas didácticas innovadoras e inclusivas, orientadas a la adquisición de varias habilidades requeridas en el mundo escolar y por la sociedad en general. En estos últimos años se ha consolidado cada vez más su valor didáctico reconociendo su potencial para el aprendizaje y para la adquisición de habilidades de resolución de problemas, de creatividad y de colaboración. El *coding* muestra entonces un gran potencial no sólo de aprendizaje, sino también de desarrollo emocional positivo en los alumnos que la han practicado. Las experiencias registradas se prestan a ser

transferidas a otros contextos educativos con el objetivo de fomentar formas de aprendizaje más activas y motivadoras.

El rol de la docente que integró el ambiente de programación visual de Scratch como herramienta pedagógica, fue fundamental ya que contribuyó tanto a la enseñanza de las disciplinas STEM como a la familiarización con la tecnología (TIC) y con la programación, aspecto que, aunque no fue directamente objeto de la presente investigación, surgió durante la realización del estudio. Se reitera la importancia de que los profesores que van a utilizar la codificación en su práctica docente estén adecuadamente formados y preparados para que puedan desarrollar prácticas innovadoras de calidad en el aula. La implementación de un plan de formación para los docentes dirigido a las distintas dimensiones del proceso de enseñanza como la humanística, la expresiva, la creativa, puede considerarse un elemento indispensable para la práctica del pensamiento computacional, incluso si actualmente, aún no se ha tomado debidamente en cuenta.

En definitiva, a pesar de tener un marco teórico de referencia que no es homogéneo en cuanto a los temas relacionados con el pensamiento computacional y con el *coding*, se considera que, el mundo de la educación necesita intervenciones que pueden favorecer el desarrollo de iniciativas orientadas a complementar las experiencias de aprendizaje en clase.

Teniendo en cuenta que el *coding*, forma parte cada vez más del plan de estudios de las escuelas primarias en todo el mundo, se reconoce la necesidad de seguir estudiando sus efectos sobre todo el proceso de enseñanza y aprendizaje de las disciplinas escolares (Arfè et al., 2020) en un mayor número de clases y por periodos de tiempo más largos; para hacer esto, es necesario contar con docentes formados en *coding*, en su uso en la enseñanza de los distintos grados escolares y, en particular, en Educación Primaria.

Anexos

Anexo A

Tabla A1

Comparación entre grupos del posttest divididos por dimensión

	T-test de Student (post test) por dimensión						
	Grupo de control(n=21)		Grupo experimental(n=25)		<i>t</i>	<i>p</i>	η^2
	<i>M</i>	<i>DS</i>	<i>M</i>	<i>DS</i>			
Puntuación para dimensión 1	5.10	1.480	6.08	1.320	-2.38	.021	.11
Puntuación para dimensión 2	2.67	.730	2.72	.936	-.21	.833	.00
Puntuación para dimensión 3	.390	1.221	4.32	1.282	-1.12	.270	.03

Fuente: elaboración propia.

Tabla A2

Comparación entre grupos del posttest divididos por ítem

	T-test de Student						
	Grupo de control (n=21)		Grupo experimental (n=25)		<i>t</i>	<i>p</i>	η^2
	<i>M</i>	<i>DS</i>	<i>M</i>	<i>DS</i>			
item 1	.48	.512	.64	.490	1.107	.217	.03
item 2	1.00	.000	.96	.200	-.915	.063	.02
item 3	.19	.402	.64	.490	3.358	.012	.20
item 4	.43	.507	.36	.490	-.465	.388	.06
item 5	.62	.498	.96	.200	3.142	.000	.18
item 6	.48	.512	.56	.507	.556	.670	.01
item 7	.90	.301	.96	.200	.744	.137	.01
item 8	.76	.436	.80	.408	.305	.546	.00
item 9a	.67	.483	.76	.436	.689	.184	.01
item 9b	.62	.498	.52	.510	.663	.258	.01
item 9c	.38	.498	.48	.510	.663	.258	.01
item 9d	.81	.402	.88	.332	.651	.197	.01
item 10	.38	.498	.32	.476	-.424	.415	.00
item 11	.90	.301	.88	.332	-.263	.599	.00
item 12a1	.90	.301	1.00	.000	1.587	.001	.05
item 12a2	.71	.463	.96	.200	2.404	.000	.12
item 12a3	.76	.436	.80	.408	.305	.546	.00
item 12b	.62	.498	.92	.277	2.588	.000	.13

Fuente: elaboración propia.

Anexo B

Tabla B1

Prueba estadística para la comparación entre los grupos en el postest divididos por subescala

<i>Sub escala</i>	U-Mann Whitney					
	Grupo de control (n=21)	Grupo experiemetal (n=25)	<i>U</i>	<i>z</i>	<i>p</i>	<i>r</i>
	<i>Md</i>	<i>Md</i>				
Alegría-Diversión^a	33.48	29.00	178.00	-1.867	.06	.28
Ansiedad^b	19.52	18.00	252.00	-.232	.82	.03
Aburrimiento^c	17.67	19.00	195.00	-1.504	.13	.22

Nota.

^a Posible intervalo de variación de puntuación: 9-45.

^b Posible intervalo di variación de puntuación: 12-60.

^c Posible intervalo di variación de puntuación: 11-55.

Fuente: elaboración propia.

Tabla B2

Comparación entre grupos del cuestionario AEQ-ES divididos por actividad

<i>Attività</i>	U-Mann Whitney					
	Grupo de control (n=21)	Grupo experiemetal (n=25)	<i>U</i>	<i>z</i>	<i>p</i>	<i>r</i>
	<i>Md</i>	<i>Md</i>				
Lecciones						
Alegría-Diversión ^c	16	16	233.50	-.643	.520	.09
Ansiedad ^c	5	5	213.00	-1.138	.255	.17
Aburrimiento ^c	4	7	192.50	-1.616	.106	.24
Tareas						
Alegría-Diversión ^a	7	5	163.50	-2.207	.027	.33
Ansiedad ^b	4	4	260.50	-.046	.963	.01
Aburrimiento ^b	3	5	192.00	-1.608	.108	.24
Evaluaciones						
Alegría-Diversión ^b	12	9	167.50	-2.108	.035	.31
Ansiedad ^d	9	8	237.00	-.566	.571	.08
Aburrimiento ^c	4	6	230.00	-.750	.453	.11

Nota.

^a Posible intervalo de variación de puntuación: 2-10.

^b Posible intervalo di variación de puntuación: 3-15.

^c Posible intervalo de variación de puntuación: 4-20.

^d Posible intervalo de variación de puntuación: 5-25.

Fuente: elaboración propia.

Tabla B3

Comparación entre grupos del cuestionario AEQ-ES divididos por ítem

	U-Mann Whitney							
	Grupo de control (n=21)		Grupo experimental (n=25)		U	z	p	r
	Md	Md						
Lecciones								
1	4.00	4.00	211.00	-1.187	.235	.18		
2	1.00	2.00	229.00	-.842	.400	.12		
3	1.00	1.00	259.00	-.158	.875	.02		
4	4.00	4.00	248.50	.320	.749	.05		
5	1.00	1.00	228.00	-.909	.363	.13		
6	1.00	1.00	207.00	-1.411	.158	.21		
7	1.00	1.00	238.50	-.766	.444	.11		
8	5.00	3.00	200.50	-1.449	.147	.21		
9	1.00	1.00	208.00	-1.399	.162	.21		
10	2.00	1.00	187.00	-1.836	.066	.27		
11	1.00	2.00	188.50	-1.812	.070	.27		
12	4.00	4.00	252.00	-.242	.809	.04		
Tareas								
1(a)	4.00	3.00	165.00	-.201	.028	.03		
2(a)	1.00	2.00	214.50	-1.149	.250	.02		
3(a)	1.00	1.00	245.50	-.521	.603	.08		
4(a)	1.00	1.00	205.50	-1.434	.151	.21		
5(a)	2.00	1.00	215.00	-1.128	.259	.17		
6(a)	3.00	2.00	189.00	-1.655	.098	.24		
7(a)	1.00	1.00	246.50	-.490	.624	.07		
8(a)	1.00	2.00	234.00	-.686	.492	.10		
Evaluaciones								
1(b)	4.00	3.00	159.50	-2.331	.020	.34		
2(b)	2.00	2.00	245.00	-.398	.691	.06		
3(b)	1.00	1.00	224.50	-.991	.322	.15		
4(b)	2.00	2.00	223.50	-.914	.360	.13		
5(b)	4.00	3.00	237.00	-.580	.562	.09		
6(b)	1.00	1.00	254.50	-.217	.828	.03		
7(b)	1.00	1.00	230.50	-.794	.427	.12		
8(b)	1.00	1.00	252.50	-.271	.786	.04		
9(b)	2.00	2.00	241.00	-.512	.609	.08		
10(b)	1.00	1.00	258.50	-.103	.918	.02		
11(b)	1.00	1.00	245.50	-.419	.675	.06		
12(b)	4.00	3.00	219.50	-.980	.327	.14		

Nota: Posible intervalo de variación de puntuación: 1-5.

Fuente: elaboración propia.

Anexo C

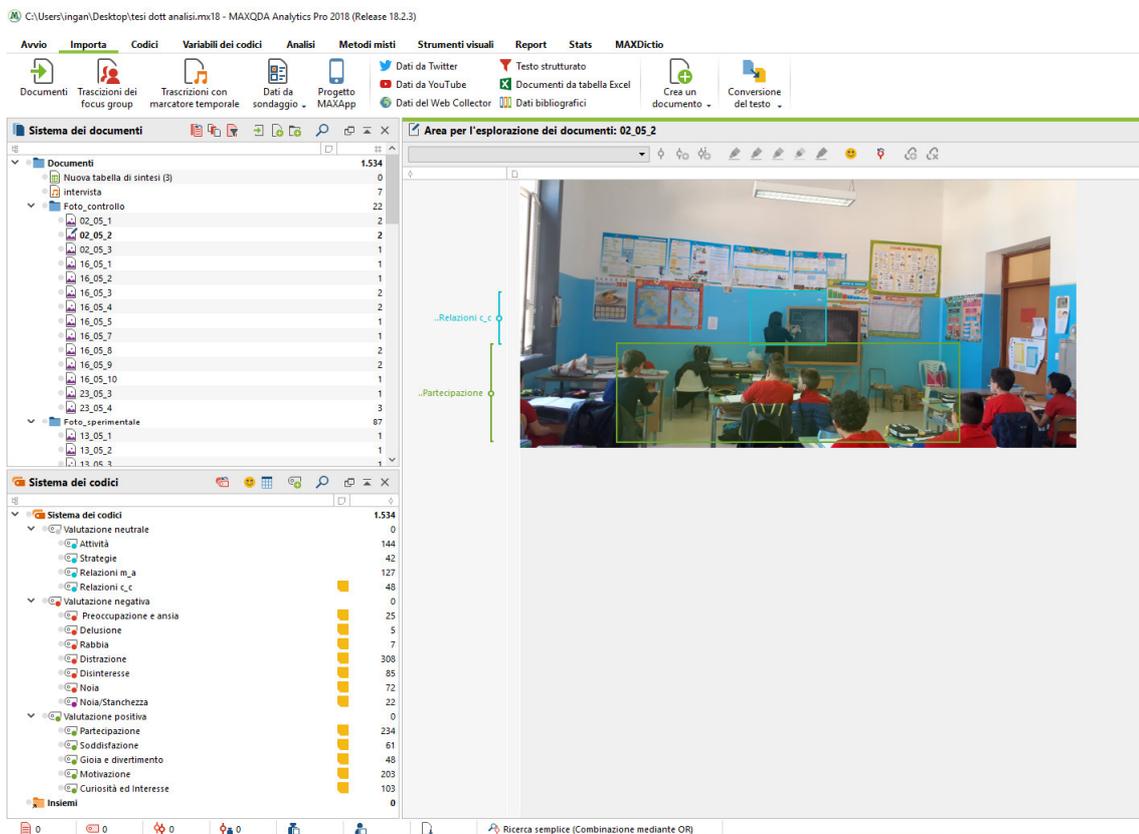


Figura 4.1. Procesamiento de imágenes con MAXQDA (02-05-2019)



Figura 4.2. Observación del grupo de control (02-05-2019).

Figuras 4.3 Observación del grupo de control (02-05-2019, 16-05-2019).











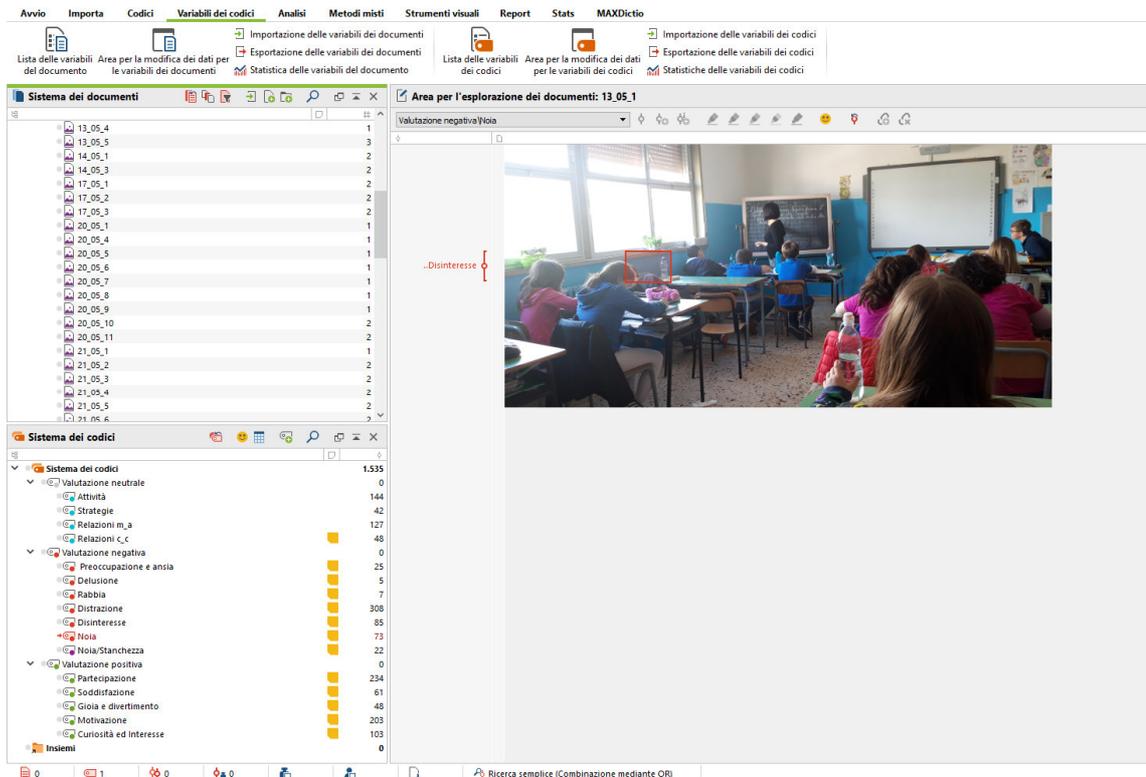


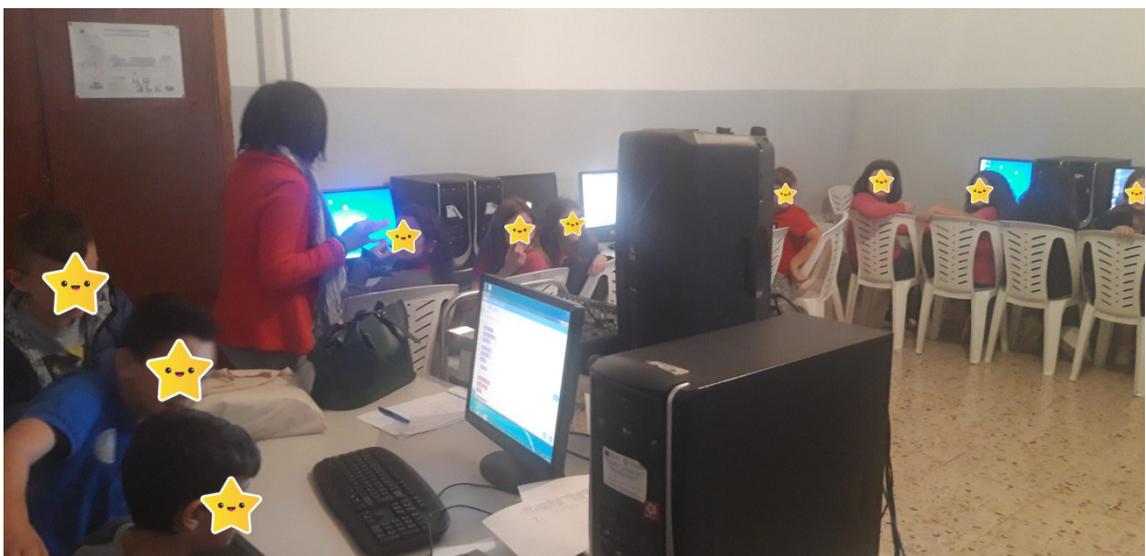
Figura 4.4 Procesamiento de imágenes con MAXQDA (grupo experimental) (13-05-2019)

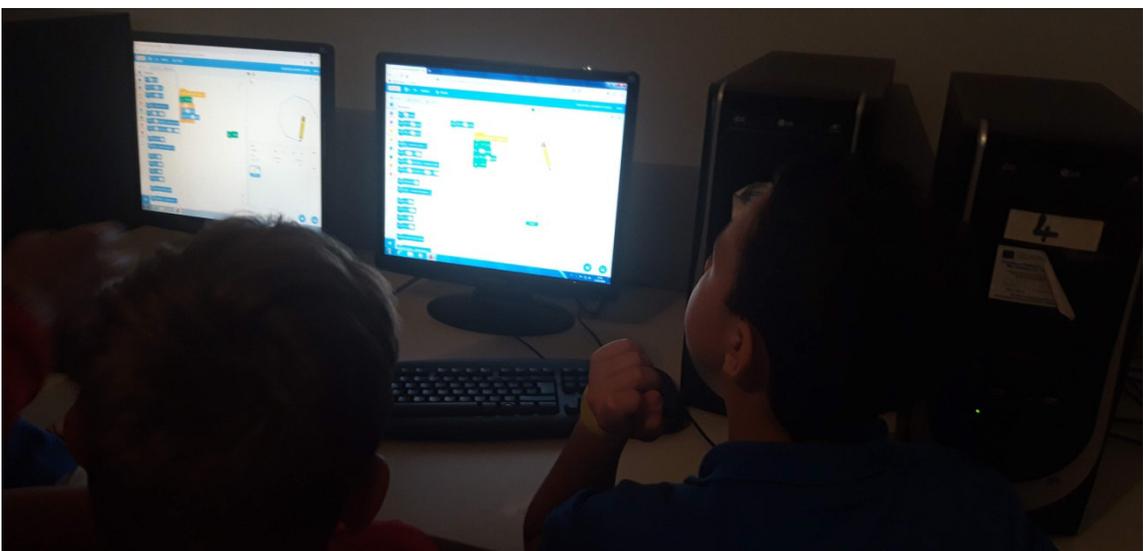
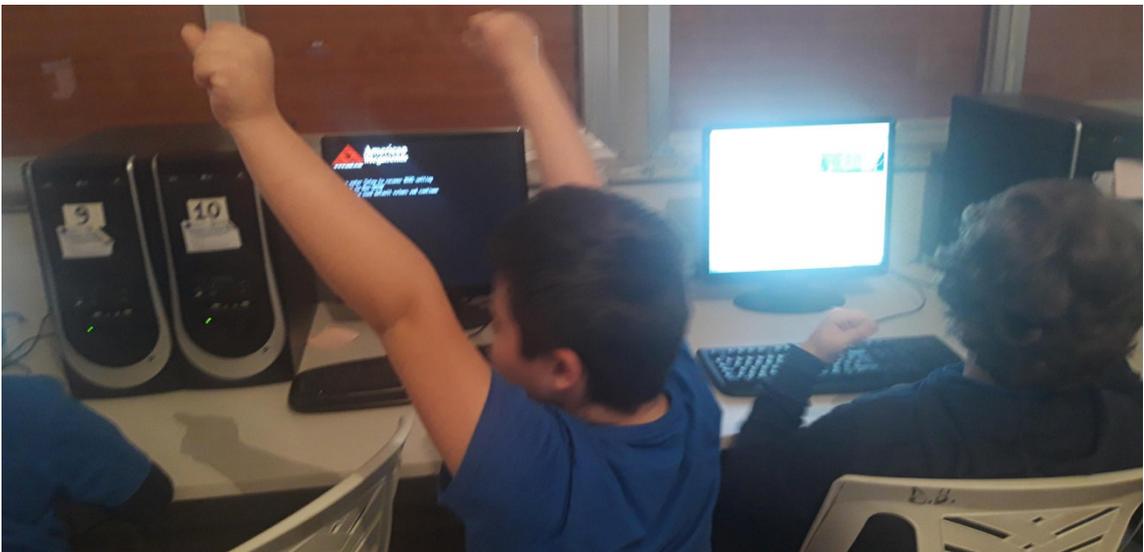
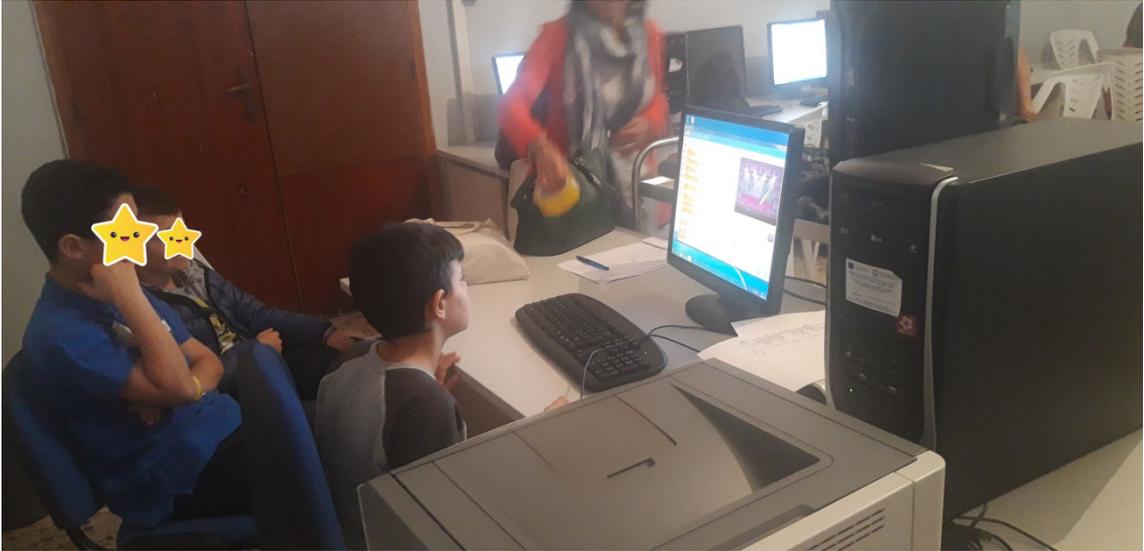


Figura 4.5 Observación del grupo experimental(13-05-2019).

Figuras 4.5 Observación del grupo experimental (02-05-2019, 16-05-2019).

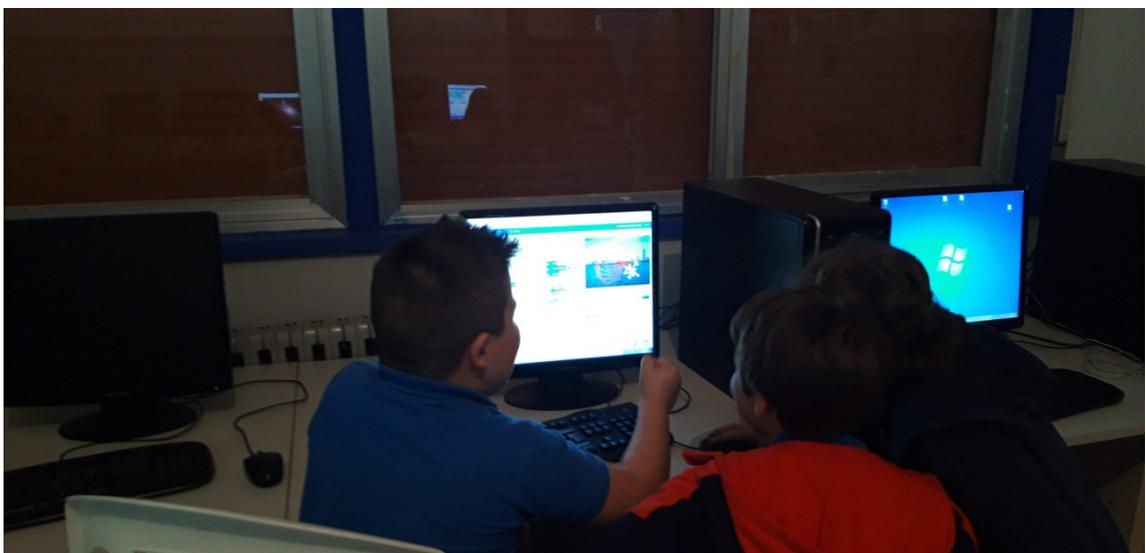
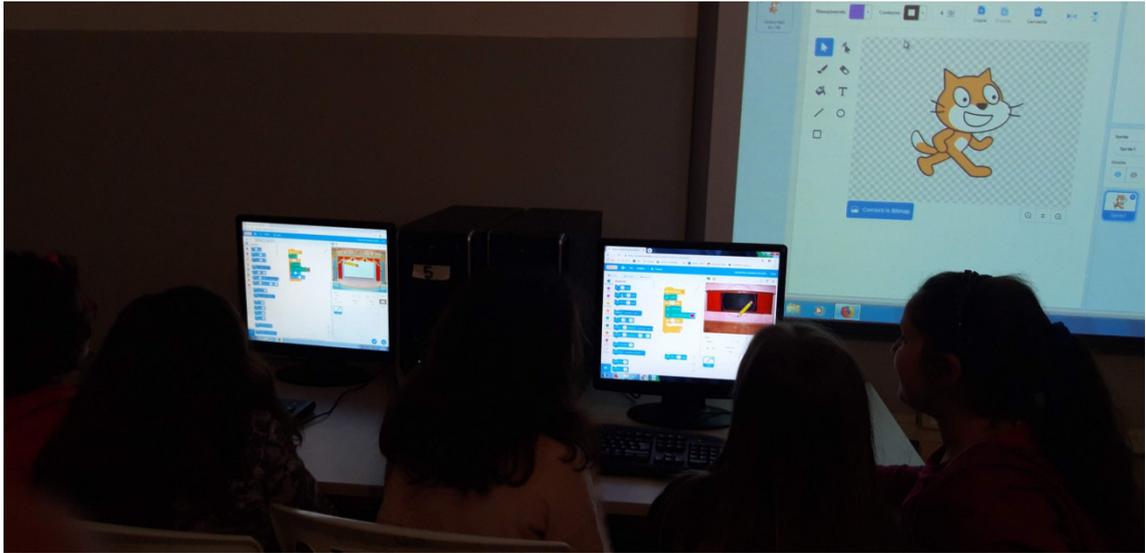


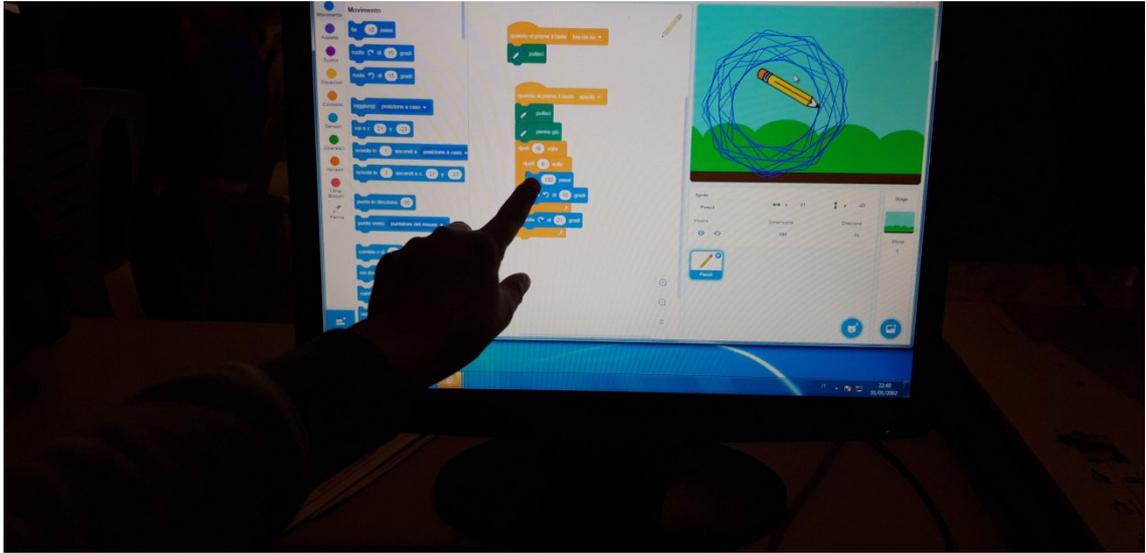














Anexo D1 – Cuestionario AEQ_ES para niñas

QUESTIONARIO

ESEMPIO

Fai una croce sul pallino che corrisponde alla tua risposta.

1					
Mi piace nuotare.					
					
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
1	2	3	4	5	
per niente	poco	abbastanza	molto	moltissimo	
2					
Andare in montagna mi annoia.					
					
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
1	2	3	4	5	
per niente	poco	abbastanza	molto	moltissimo	
3					
Nuotare nell'acqua fonda mi spaventa.					
					
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
1	2	3	4	5	
per niente	poco	abbastanza	molto	moltissimo	

Geometria-Lezioni

Fai una croce sul pallino che corrisponde alla tua risposta.



1	Mi diverto nelle lezioni di geometria.				
					
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
1 per niente	2 poco	3 abbastanza	4 molto	5 moltissimo	
2	Le lezioni di geometria mi annoiano.				
					
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
1 per niente	2 poco	3 abbastanza	4 molto	5 moltissimo	
3	La geometria mi spaventa.				
					
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
1 per niente	2 poco	3 abbastanza	4 molto	5 moltissimo	

Geometria-Lezioni

Fai una croce sul pallino che corrisponde alla tua risposta.



4	Non vedo l'ora di fare le lezioni di geometria.				
					
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
1 per niente	2 poco	3 abbastanza	4 molto	5 moltissimo	
5	Quando penso alle lezioni di geometria, divento nervosa.				
					
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
1 per niente	2 poco	3 abbastanza	4 molto	5 moltissimo	
6	La geometria mi annoia.				
					
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
1 per niente	2 poco	3 abbastanza	4 molto	5 moltissimo	

Geometria-Lezioni

**Fai una croce sul pallino
che corrisponde alla tua risposta.**



7	Quando penso alle lezioni di geometria, mi viene mal di pancia.				
					
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
1 per niente	2 poco	3 abbastanza	4 molto	5 moltissimo	

8	La geometria è divertente per me.				
					
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
1 per niente	2 poco	3 abbastanza	4 molto	5 moltissimo	

9	Per me fare geometria è noioso.				
					
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
1 per niente	2 poco	3 abbastanza	4 molto	5 moltissimo	

Geometria-Lezioni

Fai una croce sul pallino che corrisponde alla tua risposta.



10 Durante le lezioni di geometria mi preoccupo, perchè mi sembra tutto troppo difficile.				
				
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
1 per niente	2 poco	3 abbastanza	4 molto	5 moltissimo
11 Per me le lezioni di geometria sono così noiose che preferirei fare qualcos'altro.				
				
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
1 per niente	2 poco	3 abbastanza	4 molto	5 moltissimo
12 Mi diverto a fare geometria.				
				
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
1 per niente	2 poco	3 abbastanza	4 molto	5 moltissimo

Geometria-Compiti per casa
Fai una croce sul pallino
che corrisponde alla tua risposta.



1	Quando faccio i compiti per casa di geometria, sono contenta.				
					
○	○	○	○	○	
1 per niente	2 poco	3 abbastanza	4 molto	5 moltissimo	
2	I compiti per casa di geometria mi annoiano da morire.				
					
○	○	○	○	○	
1 per niente	2 poco	3 abbastanza	4 molto	5 moltissimo	
3	Mi preoccupo così tanto di non riuscire a finire i compiti per casa di geometria che comincio a sudare.				
					
○	○	○	○	○	
1 per niente	2 poco	3 abbastanza	4 molto	5 moltissimo	

Geometria-Compiti per casa
Fai una croce sul pallino
che corrisponde alla tua risposta.



4	I compiti per casa di geometria mi annoiano così tanto che non vorrei continuare a farli.				
					
○	○	○	○	○	
1 per niente	2 poco	3 abbastanza	4 molto	5 moltissimo	
5	Quando faccio i compiti per casa di geometria, mi preoccupò perchè mi chiedo se mai li capirò.				
					
○	○	○	○	○	
1 per niente	2 poco	3 abbastanza	4 molto	5 moltissimo	
6	Mi diverto così tanto nei compiti per casa di geometria che non vorrei fermarmi mai.				
					
○	○	○	○	○	
1 per niente	2 poco	3 abbastanza	4 molto	5 moltissimo	

Geometria-Compiti per casa
Fai una croce sul pallino
che corrisponde alla tua risposta.



7	I compiti per casa di geometria mi spaventano così tanto che non vorrei mai iniziare a farli.				
					
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
1 per niente	2 poco	3 abbastanza	4 molto	5 moltissimo	
8	Quando faccio i compiti per casa di geometria, mi stanco presto perchè mi annoio.				
					
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
1 per niente	2 poco	3 abbastanza	4 molto	5 moltissimo	

Geometria-Verifiche

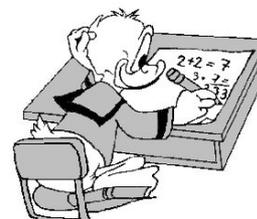
Fai una croce sul pallino che corrisponde alla tua risposta.



1 Mi diverto nelle verifiche di geometria.				
				
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
1 per niente	2 poco	3 abbastanza	4 molto	5 moltissimo
2 Quando faccio le verifiche di geometria, ho paura di prendere un brutto voto.				
				
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
1 per niente	2 poco	3 abbastanza	4 molto	5 moltissimo
3 Le verifiche di geometria mi annoiano.				
				
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
1 per niente	2 poco	3 abbastanza	4 molto	5 moltissimo

Geometria-Verifiche

Fai una croce sul pallino che corrisponde alla tua risposta.



4	Sono così nervosa durante le verifiche di geometria che non riesco a ricordarmi bene quello che ho imparato.				
					
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
1 per niente	2 poco	3 abbastanza	4 molto	5 moltissimo	
5	Non vedo l'ora di fare le verifiche di geometria.				
					
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
1 per niente	2 poco	3 abbastanza	4 molto	5 moltissimo	
6	Le verifiche di geometria mi spaventano così tanto che preferirei non farle.				
					
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
1 per niente	2 poco	3 abbastanza	4 molto	5 moltissimo	

Geometria-Verifiche

Fai una croce sul pallino che corrisponde alla tua risposta.



7	Per me fare le verifiche di geometria è noioso.				
					
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
1 per niente	2 poco	3 abbastanza	4 molto	5 moltissimo	
8	Le verifiche di geometria mi annoiano così tanto che non vorrei continuare a farle.				
					
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
1 per niente	2 poco	3 abbastanza	4 molto	5 moltissimo	
9	Sono così nervosa durante le verifiche di geometria che non riesco a concentrarmi bene.				
					
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
1 per niente	2 poco	3 abbastanza	4 molto	5 moltissimo	

Geometria-Verifiche

Fai una croce sul pallino che corrisponde alla tua risposta.



10	Quando faccio le verifiche di geometria, mi stanco presto perchè mi annoio.				
					
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
1 per niente	2 poco	3 abbastanza	4 molto	5 moltissimo	
11	Divento nervosa durante le verifiche di geometria.				
					
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
1 per niente	2 poco	3 abbastanza	4 molto	5 moltissimo	
12	Durante le verifiche di geometria, penso "Sto andando molto bene!"				
					
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
1 per niente	2 poco	3 abbastanza	4 molto	5 moltissimo	

Anexo D2 – Cuestionario *AEQ_ES* para niños

ESEMPIO
 Fai una croce sul pallino
 che corrisponde alla tua risposta.

1	Mi piace nuotare.				
					
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
1 per niente	2 poco	3 abbastanza	4 molto	5 moltissimo	
2	Andare in montagna mi annoia.				
					
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
1 per niente	2 poco	3 abbastanza	4 molto	5 moltissimo	
3	Nuotare nell'acqua fonda mi spaventa.				
					
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
1 per niente	2 poco	3 abbastanza	4 molto	5 moltissimo	

Geometria-Lezioni

Fai una croce sul pallino che corrisponde alla tua risposta.



1	Mi diverto nelle lezioni di geometria.				
					
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
1 per niente	2 poco	3 abbastanza	4 molto	5 moltissimo	
2	Le lezioni di geometria mi annoiano.				
					
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
1 per niente	2 poco	3 abbastanza	4 molto	5 moltissimo	
3	La geometria mi spaventa.				
					
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
1 per niente	2 poco	3 abbastanza	4 molto	5 moltissimo	

Geometria-Lezioni

Fai una croce sul pallino che corrisponde alla tua risposta.



4 Non vedo l'ora di fare le lezioni di geometria.				
				
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
1 per niente	2 poco	3 abbastanza	4 molto	5 moltissimo

5 Quando penso alle lezioni di geometria, divento nervoso.				
				
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
1 per niente	2 poco	3 abbastanza	4 molto	5 moltissimo

6 La geometria mi annoia.				
				
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
1 per niente	2 poco	3 abbastanza	4 molto	5 moltissimo

Geometria-Lezioni

Fai una croce sul pallino che corrisponde alla tua risposta.



7	Quando penso alle lezioni di geometria, mi viene mal di pancia.				
					
○	○	○	○	○	
1 per niente	2 poco	3 abbastanza	4 molto	5 moltissimo	
8	La geometria è divertente per me.				
					
○	○	○	○	○	
1 per niente	2 poco	3 abbastanza	4 molto	5 moltissimo	
9	Per me fare geometria è noioso.				
					
○	○	○	○	○	
1 per niente	2 poco	3 abbastanza	4 molto	5 moltissimo	

Geometria-Lezioni

Fai una croce sul pallino che corrisponde alla tua risposta.



10 Durante le lezioni di geometria mi preoccupò, perchè mi sembra tutto troppo difficile.				
				
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
1 per niente	2 poco	3 abbastanza	4 molto	5 moltissimo
11 Per me le lezioni di geometria sono così noiose che preferirei fare qualcos'altro.				
				
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
1 per niente	2 poco	3 abbastanza	4 molto	5 moltissimo
12 Mi diverto a fare geometria.				
				
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
1 per niente	2 poco	3 abbastanza	4 molto	5 moltissimo

Geometria-Compiti per casa

Fai una croce sul pallino che corrisponde alla tua risposta.



1	Quando faccio i compiti per casa di geometria, sono contento.				
					
○	○	○	○	○	
1 per niente	2 poco	3 abbastanza	4 molto	5 moltissimo	
2	I compiti per casa di geometria mi annoiano da morire.				
					
○	○	○	○	○	
1 per niente	2 poco	3 abbastanza	4 molto	5 moltissimo	
3	Mi preoccupo così tanto di non riuscire a finire i compiti per casa di geometria che comincio a sudare.				
					
○	○	○	○	○	
1 per niente	2 poco	3 abbastanza	4 molto	5 moltissimo	

Geometria-Compiti per casa

Fai una croce sul pallino che corrisponde alla tua risposta.



4	I compiti per casa di geometria mi annoiano così tanto che non vorrei continuare a farli.				
					
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
1 per niente	2 poco	3 abbastanza	4 molto	5 moltissimo	

5	Quando faccio i compiti per casa di geometria, mi preoccupò perchè mi chiedo se mai li capirò.				
					
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
1 per niente	2 poco	3 abbastanza	4 molto	5 moltissimo	

6	Mi diverto così tanto nei compiti per casa di geometria che non vorrei fermarmi mai.				
					
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
1 per niente	2 poco	3 abbastanza	4 molto	5 moltissimo	

Geometria-Compiti per casa
Fai una croce sul pallino
che corrisponde alla tua risposta.

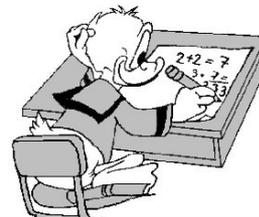


7	I compiti per casa di geometria mi spaventano così tanto che non vorrei mai iniziare a farli.				
					
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
1 per niente	2 poco	3 abbastanza	4 molto	5 moltissimo	

8	Quando faccio i compiti per casa di geometria, mi stanco presto perchè mi annoio.				
					
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
1 per niente	2 poco	3 abbastanza	4 molto	5 moltissimo	

Geometria-Verifiche

Fai una croce sul pallino che corrisponde alla tua risposta.



1 Mi diverto nelle verifiche di geometria.				
				
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
1 per niente	2 poco	3 abbastanza	4 molto	5 moltissimo
2 Quando faccio le verifiche di geometria, ho paura di prendere un brutto voto.				
				
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
1 per niente	2 poco	3 abbastanza	4 molto	5 moltissimo
3 Le verifiche di geometria mi annoiano.				
				
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
1 per niente	2 poco	3 abbastanza	4 molto	5 moltissimo

Geometria-Verifiche

Fai una croce sul pallino che corrisponde alla tua risposta.



4	Sono così nervoso durante le verifiche di geometria che non riesco a ricordarmi bene quello che ho imparato.				
					
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
1 per niente	2 poco	3 abbastanza	4 molto	5 moltissimo	
5	Non vedo l'ora di fare le verifiche di geometria.				
					
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
1 per niente	2 poco	3 abbastanza	4 molto	5 moltissimo	
6	Le verifiche di geometria mi spaventano così tanto che preferirei non farle.				
					
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
1 per niente	2 poco	3 abbastanza	4 molto	5 moltissimo	

Geometria-Verifiche

Fai una croce sul pallino che corrisponde alla tua risposta.



7	Per me fare le verifiche di geometria è noioso.				
					
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
1 per niente	2 poco	3 abbastanza	4 molto	5 moltissimo	
8	Le verifiche di geometria mi annoiano così tanto che non vorrei continuare a farle.				
					
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
1 per niente	2 poco	3 abbastanza	4 molto	5 moltissimo	
9	Sono così nervoso durante le verifiche di geometria che non riesco a concentrarmi bene.				
					
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
1 per niente	2 poco	3 abbastanza	4 molto	5 moltissimo	

Geometria-Verifiche

Fai una croce sul pallino che corrisponde alla tua risposta.



10	Quando faccio le verifiche di geometria, mi stanco presto perchè mi annoio.				
					
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
1 per niente	2 poco	3 abbastanza	4 molto	5 moltissimo	
11	Divento nervoso durante le verifiche di geometria.				
					
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
1 per niente	2 poco	3 abbastanza	4 molto	5 moltissimo	
12	Durante le verifiche di geometria, penso "Sto andando molto bene!"				
					
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
1 per niente	2 poco	3 abbastanza	4 molto	5 moltissimo	

Anexo D3 – Consentimiento informado



VNIVERSITAT ID VALÈNCIA



MODULO PER IL CONSENSO INFORMATO DEI GENITORI DEGLI STUDENTI DELLE CLASSI QUARTE DELL'I.C. DON BOSCO

I dati personali dell'alunno sono utilizzati dall'I.C. Don Bosco, che ne è titolare per il trattamento, nel rispetto dei principi di protezione dei dati personali stabiliti dal Regolamento GDPR 2016/679.

MODALITÀ E FINALITÀ DEL TRATTAMENTO DATI

1. La informiamo che i dati verranno trattati con il supporto dei seguenti mezzi:
 - cartaceo

con le seguenti finalità: scopo di ricerca.

Il presente modulo di consenso informato permette di far partecipare Vostro figlio/a alla compilazione di due tests di Geometria e un questionario del tipo Scala Likert a 5 punti* dal titolo "*Emozioni, interesse e motivazione in Geometria*", tutti ANONIMI, che verranno utilizzati per compiere un'indagine sul processo di insegnamento/apprendimento della Geometria nella scuola primaria, le cui informazioni verranno elaborate dalla ricercatrice Dott.ssa Annalisa Piazza dell'Università di Valencia (Spagna).

La compilazione non prevede alcun onere a carico delle famiglie.

Tutte le informazioni saranno utilizzate a soli scopi di ricerca e conservate nel più completo rispetto della privacy [ai sensi dell'art. 13 del Regolamento UE 2016/679](#).

E' importante che il presente modulo sia firmato da entrambi i genitori e compilato in ogni sua parte.

2. Titolare del trattamento dei Suoi dati personali è l' I.C. Don Bosco c.f. 83002130843
 - Email: agic85600x@istruzione.it
 - PEC: agic85600x@pec.istruzione.it
 - Telefono: 092561162
3. Responsabile del trattamento dei suoi dati è il Direttore SGA dott.ssa Rosalia Lazzara
4. La scrivente Società ha nominato quali DPO (Data Protection Officer) il Sig. Furnari Calogero

Io sottoscritta (madre) _____
 nata il ___/___/___ residente a _____ via/piazza
 _____ Tel. _____ domicilio (se
 diverso dalla residenza) _____

Io sottoscritto (padre) _____
 nato il ___/___/_____ residente a _____ via/piazza
 _____ Tel. _____ domicilio (se diverso
 dalla residenza) _____

dell'alunno _____
 nato il ___/___/_____ residente a _____ via/piazza

ai sensi [dell'art. 13 del Regolamento UE 2016/679](#), dichiariamo di essere stati adeguatamente informati sulle finalità della ricerca e sulle modalità di svolgimento della stessa ed in virtù della potestà genitoriale. Diamo pertanto il consenso alla partecipazione di nostro figlio / a alla compilazione dei **TEST DI GEOMETRIA E DEL QUESTIONARIO (ANONIMI) ed al trattamento dei dati che da esso deriveranno, che saranno utilizzati solo ai fini della ricerca stessa. Siamo a conoscenza inoltre che la compilazione è facoltativa e non obbligatoria.**

** viene chiesto di esprimere il grado di accordo/disaccordo con ciascuna affermazione scegliendo tra cinque modalità.*

Acconsentiamo con la sottoscrizione della presente alla somministrazione del questionario (anonimo) secondo le modalità e le finalità sopra riportate.

NON acconsentiamo

Nome per esteso del genitore	Data	Firma
_____	___/___/_____	_____

Nome per esteso del genitore	Data	Firma
_____	___/___/_____	_____

Anexo E

Diseño de la actividad experimental

El trabajo de investigación implica la planificación y el diseño de una actividad de geometría desarrollada a través del *coding*, los estudiantes tienen una experiencia en la que llevan a cabo lecciones de clase y actividades de programación gracias a las cuales pueden lograr y profundizar las habilidades requeridas por el estudio de la geometría y las habilidades transversales relacionadas. El experimento didáctico consiste en un total de nueve lecciones, con una duración total de seis horas, divididas en sesiones de 60 o 120 minutos, que abarcan actividades del *coding* desenchufadas o *unplugged* y lecciones constructivistas divididas por igual en términos de tiempo.

Las lecciones de *coding* son sobre el uso de Scratch y las imparte el profesor de matemáticas y tecnología. Se centran principalmente en conceptos matemáticos relacionados con el reconocimiento de las propiedades de las principales figuras geométricas, desde los triángulos hasta los polígonos en general (Tabla 1). Se decide que los estudiantes se dividan, durante las sesiones de *coding*, en pequeños grupos de dos o tres por estación, tanto por la falta de disponibilidad de computadoras como para estimular la colaboración entre pares.

Al principio, el docente prepara algunas actividades desenchufadas para desarrollar la capacidad de observar y analizar un gesto compuesto pero muy familiar, pidiendo a los niños que con su movimiento dibujen en el suelo. Propone al pizarrón un juego que consiste en dibujar un cuadrado de ciertos tamaños en una mitad; luego, después de haber verificado que la propuesta se cumple con bastante facilidad, le pide que dibuje el mismo cuadrado en la otra mitad del pizarrón pero manteniendo los ojos cerrados. La comparación de los

dos dibujos da lugar a una discusión sobre por qué era más difícil dibujar con los ojos cerrados y qué ha cambiado desde el primer dibujo. La idea es discutir las características fundamentales del cuadrado y los gestos que se hacen para dibujarlo, con los movimientos de las manos o del cuerpo.

Después de este momento colectivo y verificada la conciencia alcanzada del gesto gráfico, se podrá proceder a la solicitud de elaboración de un algoritmo. La articulación del trabajo prevé algunas fases y se distribuye en diferentes sesiones.

El docente asigna un proyecto a realizar ilustrando las funcionalidades de Scratch relacionadas con la actividad; les presenta el bloque "bajar lápiz" necesario para que un *objeto* dibuje una línea en el *escenario*. El primer paso que se pide a los niños es crear un proyecto en el que el personaje de su elección (*objeto*), a través de los bloques del grupo "lápiz", debe moverse dejando un rastro de su camino. Con este primer estímulo se quiere invitar a los alumnos a experimentar con varios caminos posibles manipulando comandos de movimiento. Una de las posibles respuestas al estímulo es la creación de caminos que se "parecen" a las formas geométricas reales.

Por lo tanto, la primera tarea debe materializarse con la creación de un script que permita dibujar un cuadrado y luego proceder a las siguientes tareas con el objetivo de hacer que los objetos se muevan para dibujar triángulos y otros polígonos (pentágono, hexágono, decágono, etc.). El objetivo es pensar en la longitud de los lados y la anchura de los ángulos y poder derivar una regla general que permita dibujar un polígono dado de cualquier número de lados. A esto se añade el descubrimiento del uso de ciclos, de variables, la búsqueda, identificación y corrección de errores (bugs), elementos que caracterizan la programación informática. Los niños tienen la oportunidad de trabajar eligiendo, de mutuo acuerdo, los escenarios y los personajes. La figura 1 muestra un resumen de las tarjetas divididas en tareas a realizar por los estudiantes durante las sesiones con Scratch.

La actividad se concluye con la propuesta de crear un diseño generativo utilizando los principios de repetición, aleatoriedad, variación de tamaño y trazo, a través del cual se obtendrán composiciones y/o progresiones de polígonos. (Giordano, Moscetti, 2016).

Las actividades descritas son principalmente actividades de laboratorio, requieren el uso de computadoras y se alternan por el profesor con lecciones constructivistas realizadas en el aula que profundizan y amplían los conceptos de geometría que deben transmitirse de acuerdo con los *objetivos de desarrollo de habilidades* relacionadas con la disciplina matemáticas y los *objetivos de aprendizaje* relacionados (Tabla 1). Entre una sesión del taller y otra, el docente invita a los niños a adquirir conocimientos mediante ejemplos y debates que incluyen la medición de los ángulos y sus características, la descripción, la denominación y la clasificación de las figuras geométricas (triángulos, cuadriláteros, polígonos) y el estudio de sus propiedades en función de los ángulos, los lados y las diagonales, el trazado de los ejes de simetría, el cálculo del perímetro mediante unidades de medida fijas y el reconocimiento de las figuras isoperimétricas.

Durante las sesiones de *coding*, el docente, de paso por las estaciones, comprueba la ejecución de la tarea haciendo preguntas a los alumnos para verificar su comprensión de los conceptos. "¿Cómo programaste el objeto?" o "¿Qué instrucciones le diste para llevarlo a la posición final?".

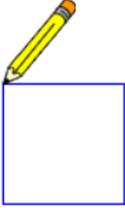
Al final de cada sesión, en el laboratorio y en el aula, el maestro lleva a los estudiantes a hacer un balance de la situación a través de un momento de intercambio, invitándolos a reconstruir en palabras lo que han aprendido, estimulando la discusión sobre las estrategias utilizadas para resolver los problemas, la generalización, la descontextualización de los conceptos aprendidos. La actividad termina con la administración a los estudiantes de la segunda prueba de geometría (post test) y su posterior corrección.

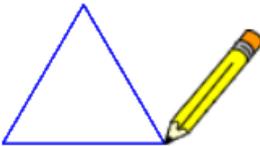
Tabla 1*Scheda attività sperimentale*

Sesión	Contenido
1	Administración y corrección del pretest G1.
2	Unplugged Coding: discusión de las características y propiedades de un cuadrado.
3	Clasificación de los cuadriláteros: definiciones y propiedades. Simetría. Ejercicio práctico.
4	Coding: diseña un cuadrado con Scratch.
5	Clasificación de los triángulos: definiciones y propiedades. Ejercicio práctico.
6	Coding: diseña un triángulo equilátero con Scratch.
7	Clasificación del cuadrilátero. Resolver problemas geométricos en el cuadrilátero y el triángulo.
8	Polígonos: características y propiedades. Coding unplugged: perímetro de figuras geométricas.
9	Coding: dibujar con Scratch un hexágono y un decágono. Encuentra una fórmula general para dibujar polígonos.
10	Coding: Ejecutar un diseño generativo con polígonos con Scratch.
11	Administración y corrección del posttest G2.
12	Administración del cuestionario AEQ_ES.

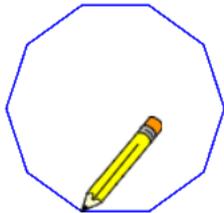
Nota: Fuente: elaboración propia.

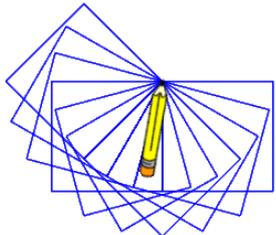
Figura E1 Schede per sessioni di coding con Scratch

Tarea 1	Conceptos geométricos	Conceptos <i>coding</i>
<p>¿Puedes ayudar al objeto a dibujar un cuadrado?</p> 	<p>Discutir las propiedades del cuadrado.</p> <p>Mide lados y ángulos.</p> <p>Calcula la suma de los ángulos interiores.</p> <p>Mide el perímetro.</p> <p>Identificar patrones de repetición.</p>	<p>Escribir una instrucción.</p> <p>Llevar a cabo una instrucción.</p> <p>Debugging.</p> <p>Uso de las funciones cíclicas (repetir).</p>

Tarea 2	Conceptos geométrici	Conceptos <i>coding</i>
<p>¿Puedes ayudar al objeto a dibujar un triángulo equilátero?</p> 	<p>Discutiendo las propiedades de los triángulos.</p> <p>Identificar ángulos externos e internos.</p> <p>Calcula la suma de los ángulos internos.</p> <p>Identificar las alturas.</p> <p>Mide el perímetro.</p> <p>Identificar patrones de repetición.</p>	<p>Escribir una instrucción.</p> <p>Llevar a cabo una instrucción.</p> <p>Debugging.</p> <p>Uso de las funciones cíclicas (repetir).</p>

Tarea 3	Conceptos geométrici	Conceptos <i>coding</i>
<p>¿Puedes ayudar al objeto a dibujar un hexágono/decágono?</p> 	<p>Discutiendo las propiedades de los polígonos.</p> <p>Mide lados y ángulos.</p> <p>Calcula la suma de los ángulos internos.</p> <p>Mide el perímetro.</p> <p>Identificar patrones de repetición.</p>	<p>Escribir una instrucción.</p> <p>Llevar a cabo una instrucción.</p> <p>Debugging.</p> <p>Uso de las funciones cíclicas (repetir).</p>

Tarea 4	Conceptos geométrici	Conceptos <i>coding</i>
<p>¿Puedes ayudar al objeto a dibujar cualquier polígono?</p> 	<p>Buscar una fórmula para dibujar todos los polígonos.</p> <p>Calcula la suma de los ángulos interiores.</p> <p>Identificar los patrones que se repiten.</p>	<p>Escribir una instrucción.</p> <p>Llevar a cabo una instrucción.</p> <p>Debugging.</p> <p>Uso de las funciones cíclicas (repetir).</p> <p>Usar una variable.</p>

Tarea 5	Conceptos geométrici	Conceptos <i>coding</i>
<p>Puoi aiutare lo sprite a generare un disegno utilizzando un quadrato?</p> 	<p>Pensar en las unidades.</p> <p>Razonar en ángulos y rotaciones.</p> <p>Identificar los patrones que se repiten.</p>	<p>Escribir una instrucción.</p> <p>Llevar a cabo una instrucción.</p> <p>Debugging.</p> <p>Uso de las funciones cíclicas (repetir).</p> <p>Usar una variable.</p> <p>Usar la casualidad.</p>

Anexo F

Tarjeta de maestra del grupo experimental

SCHEDA DOCENTE

INFORMAZIONI GENERALI

Nombre y apellidos: Rosalinda Di Stefano
Formación: título universitario en Lenguas y Literaturas Extranjeras
Centro educativo: Istituto comprensivo "Don Bosco" Ribera (Ag)
Años de docencia: 23
Asignaturas impartidas: Matemáticas y tecnología

CONOSCENZA SUL TEMA CODING

Cursos de actualización profesional realizados (indicar si se realizan de forma independiente o a propuesta de la escuela):

- Code.org (scuola)
- Coding in your classroom now! (autonomia)
- Umano digitale (autonomia)

Proyectos de *coding* en las escuelas:

- Progetto Pon "A scuola di *coding*!" con funzione di tutor

Plataformas de *coding* conoce (marque una X):

Code.org

Alice.org

Scratch

Codeacademy.com

Zaply code

Google CS First

Education.minecraft.net

Csunplugged.org

Altro (specificare).....

Prácticas de *coding* en la escuela: Si No

Materias: Matemáticas y tecnología

Consenso, per l'intervistatore, all'intervista e all'uso delle informazioni date

Titolo del progetto di ricerca :

Può il coding modificare il processo di apprendimento/insegnamento della geometria nella scuola primaria?

Istituzione di ricerca : Universitat de València

Nome dell'intervistatore e del curatore della raccolta

Annalisa Piazza

Sono stato informato circa il progetto di ricerca e il modo in cui il mio contributo potrà essere usato. Mi è stato spiegato come la trascrizione dell'intervista, e/o l'ascolto della registrazione audio relativa, sarà mantenuta riservata, salvo il mio assenso a renderla pubblica.

Il mio contributo sarà conservato in luogo sicuro, e ne sarà consentito l'accesso solo a coloro che ne ricevano il permesso dall'intervistatore.

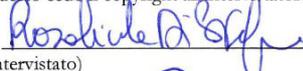
Sono a conoscenza che posso ritirare il mio assenso in qualunque momento informando l'intervistatore.

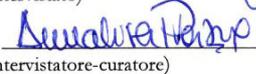
Barrare una delle due :

Dò il mio assenso all'intervista che sto per concedere all'interno del progetto di cui sopra, intervista che potrà essere usata solo per i fini della ricerca (includere pubblicazioni e rapporti di ricerca), e in forma strettamente anonima, dall'intervistatore*.

Dò il mio assenso a pubblicare la parte dell'intervista che ho concesso all'interno del progetto di cui sopra, intervista che potrà essere usata solo per i fini della ricerca dall'intervistatore (includere pubblicazioni e rapporti di ricerca) con la possibilità di rendere noto il mio nome.

Con questo cedo il copyright all'intervistatore e curatore D.ssa Annalisa Piazza

Firma  Data 01/06/2019
(dell'intervistato)

Firma  Data 01/06/2019
(dell'intervistatore-curatore)

* La riservatezza dei dati personali è comunque tutelata dal DL n. 196 del 30 gennaio 2003, e dal "Codice di deontologia dei dati personali per scopi storici" Provvedimento n. 8/P/2001) del garante per la protezione dei dati personali.

Transcripción de la entrevista

E: El objetivo de esta entrevista es escuchar el punto de vista de la profesora Rosalinda Di Stefano, promotora del proceso de enseñanza/aprendizaje en la cuarta clase B del plexo Imbornone del Istituto Comprensivo Don Bosco. Esta clase constituyó el grupo experimental del proyecto de investigación: en ella se realizaron actividades de *coding* durante las clases de geometría. Nos gustaría conocer sus impresiones, ideas y sugerencias tras la experiencia en el aula. Hola Rosalinda, la primera pregunta que me gustaría hacerte es la siguiente: - ¿Crees que existe cierta relación entre la promoción del desarrollo del pensamiento computacional y el uso del *coding* en las escuelas?

R: Hola Annalisa, sí, creo que hay una cierta relación entre la práctica de la codificación y el desarrollo del pensamiento computacional porque creo que el coding es la herramienta adecuada para desarrollar el pensamiento computacional.

E: ¿Has realizado alguna actividad de *coding* en tus clases?

R: Sí, realicé actividades de coding en mis tres clases, dos de cuarto y una de segundo: empecé en modo desconectado y luego pasé gradualmente a modo online utilizando Code.org y la plataforma Scratch..

E: ¿En qué contexto utilizó estas actividades, de forma extraescolar o durante actividades específicas de matemáticas u otras?

R: Hice estas actividades principalmente durante la tecnología, pero también (ahora) utilicé la geometría, así que digamos que en general hice el *coding* tanto en la tecnología como en la geometría.

E: ¿Su escuela ha promovido y estimulado a los profesores en la promoción de prácticas que puedan relacionarse con el pensamiento computacional?

R: Sí, mi director nos anima constantemente a los profesores a practicar el *coding*, a utilizarla en nuestras clases..

E: ¿Ha realizado, por ejemplo, cursos de perfeccionamiento o participado en proyectos nacionales, como PON?

R: Sí, en nuestra escuela se imparten cursos PON sobre el *coding* incluso en dos niveles, uno básico y otro más avanzado.

E: ¿Cómo conoció el *coding* y su posible uso como metodología de enseñanza?

R: Oí hablar mucho de ello en las redes sociales, me intrigó este *coding* y luego hice un curso MOOC en la Universidad de Urbino, con Alessandro Bogliolo.

E: ¿Ha realizado alguna vez actividades relacionadas con el *coding*?

R: No, es la primera vez.

E: Es decir... ¿cuánto tiempo llevas utilizando esta metodología?

R: Unos tres años.

E: ¿Te ha parecido útil trabajar en actividades de geometría a través del *coding*?

R: Sí, en mi opinión fue útil porque traté temas que probablemente no habría tratado de otro modo en una clase de geometría tradicional, por lo que desarrollé los conceptos con mayor profundidad.

E: ¿Hubo algún momento en el que pensó que las actividades que estaba realizando podrían ser, por ejemplo, inútiles?

R: Nunca pensé que las actividades fueran inútiles, pero a veces tenía dudas sobre cómo iba la clase porque había un poco de confusión porque los niños estaban muy entusiasmados y entonces a veces pensaba que la situación se me iba de las manos.

E: ¿Y cómo ha afrontado estas situaciones?

R: Siempre intentaba hacer un punto, es decir, hacerles entender que estábamos jugando pero, al mismo tiempo, estábamos desarrollando un concepto, así que intentaba llamar su atención y preguntarles qué estábamos haciendo... Siempre les hacía discutir.

E: Todas estas actividades de codificación que dices que hiciste en clase, ¿las planificaste antes de entrar en el aula?

R: Sí, siempre. Siempre he planificado la actividad antes de presentarla a los niños.

E: Durante las actividades, ¿qué pensaste, qué notaste, qué pasaba en el aula?

R: Mientras tanto, noté una mayor implicación de los niños, un mayor entusiasmo en comparación con la clase tradicional; los niños se lo tomaron casi como un juego, aunque luego me di cuenta de que los conceptos que desarrollamos eran realmente importantes, pero ellos sólo pensaban que estaban jugando.

E: ¿Le ha gustado el trabajo realizado?

R: Sí, me gustó mucho el trabajo porque era innovador en comparación con la lección tradicional... lo único era contener su entusiasmo.

E: ¿Crees que los niños han disfrutado de esta actividad?

R: Sí, creo que han disfrutado de esta actividad y estoy seguro de que me pedirán que continúe.

E: ¿Ha compartido esta experiencia y sus conocimientos de coding, por ejemplo, con otros compañeros que hacen el mismo trabajo que usted?

R: Por ello, sólo compartía mis conocimientos y experiencia con quienes me pedían opinión o consejo. En general, no hay una buena disposición hacia esta codificación. El coding como tal da miedo porque no se conoce y por lo tanto no todo el mundo está dispuesto a involucrarse.

E: ¿Cree que la formación es, por tanto, necesaria?

R: Sí, creo que la formación es necesaria.

E: ¿Recomendarías este método de enseñanza a otros colegas?

R: Sí, ya lo he recomendado. Es, sin duda, una metodología innovadora, especial y todavía atractiva para los niños.

E: Y que usted sepa, ¿ha habido algún colega que se haya embarcado en esta experiencia hasta cierto punto?

R: Sí, alguien está empezando a practicar el *coding*.

E: Sí, hace un rato has dicho que has notado algunas diferencias en comparación con la clase tradicional. Más concretamente, ¿podría mencionar algunas otras o detallarlas?

R: ¿A qué se refiere?

E: Ya que ha observado diferencias entre las clases tradicionales y las actividades de *coding*, ¿qué diferencia en particular a estas dos actividades?

R: El hecho de que, durante la clase tradicional, muchos niños se aburren y pierden el sentido del discurso, mientras que durante las actividades de *coding* los niños siempre se sentían atraídos y, a menudo, observé, haciendo un balance de la situación, que los que se perdían en el aula eran capaces de recuperar el hilo del discurso.

E: ¿Cree que el compromiso y el interés de sus alumnos fue diferente al de las actividades tradicionales?

R: Sí, ciertamente porque en el aula los niños necesitan más estimulación, pero allí estaban definitivamente más involucrados.

E: Sí, ¿ha notado alguna diferencia, por ejemplo, entre los niños que generalmente se comportan mejor o se implican más y los que suelen estar menos comprometidos?

R: Por lo tanto, los niños buenos son siempre buenos, pero noté diferencias en los niños con más dificultades. ¿Quizás durante la clase tradicional nunca levantaron la mano, pero durante las actividades de *coding* se lanzaron a hacer comentarios a

veces pertinentes?

E: ¿Ha notado un cambio en el compromiso, la motivación y el interés?

R: Definitivamente sí. Más motivados, más implicados.

E: ¿Piensa seguir utilizando el *coding* en el aula, por ejemplo, en el próximo curso escolar?

R: Sí, definitivamente seguiré utilizando el *coding*.

E: ¿Qué opina, en cambio, de los resultados obtenidos en cuanto al rendimiento de los alumnos? Como sabes, hicieron una prueba final al término de las actividades previstas y te mostré los resultados. ¿Qué opinas, qué comentarios tienes al respecto?

R: Ciertamente esperaba más, tanto por todo el trabajo que había realizado como por el tiempo que había invertido en estas actividades, pero evidentemente realizaron esta prueba de forma superficial y, por tanto, no obtuvieron los resultados que yo esperaba de ellos.

E: En lo que respecta, por ejemplo, a otros factores, como las relaciones sociales y la participación e interés, que ya hemos mencionado, ¿qué cree que ha conseguido en términos de resultados?

R: Eh, en términos sociales, vi que cuando terminaron su trabajo tuvieron el placer de compartirlo con sus compañeros, algo que casi no sucede en clase, durante la lección tradicional, excepto cuando hacen una actividad particular y entonces sí, pero...

E: ¿Tiene alguna sugerencia a la luz de lo que ha hecho? ¿Cambiarías algo?

R: Sí, en mi opinión hubiera estado bien que cada niño hubiera tenido un ordenador y por lo tanto un trabajo más particular y mejor porque con la gestión del grupo pequeño a menudo había pequeños problemas porque siempre había un niño que sofocaba al otro por lo que la dinámica no siempre era manejable.

E: Bien, pero la última pregunta que quiero hacerte es sobre una comparación con la clase paralela que, digamos, realizó la misma programación que tú hiciste en tu clase, pero sin usar esta metodología, sin usar el *coding*, y por lo tanto realizando actividades tradicionales; ¿crees que, comparando tu trabajo con el de tu compañero, hiciste todo lo previsto en la programación, o hiciste menos o más?

R: Seguro que, comparándome con mi colega, he hecho algunos aspectos menos que tú, he profundizado menos en algunos aspectos porque no tenía tiempo y tú, en cambio, has hecho más la programación...

E: Bien, pero ¿se retomarán estos aspectos de la programación al año siguiente?

R: Sí, por supuesto, el año que viene pienso continuar donde lo dejé este año.

R: Bien, la entrevista ha finalizado, agradezco a la maestra Rosalinda la disposición que ha mostrado a lo largo del proceso de investigación y aprovecho para precisar que las respuestas dadas en esta entrevista serán utilizadas únicamente con fines de investigación y almacenadas con pleno respeto a la privacidad según la legislación vigente. Gracias.

Anexos H1 - Archivo de la prueba inicial

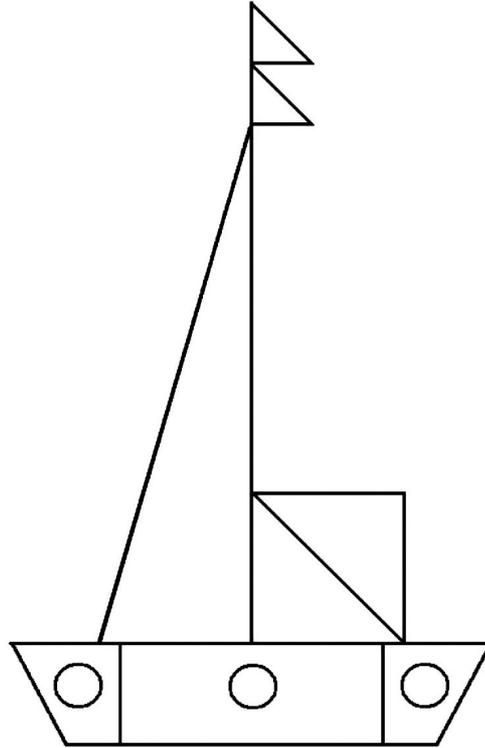
TEST DI GEOMETRIA G1

Nome _____ Cognome _____

Classe _____

Data _____

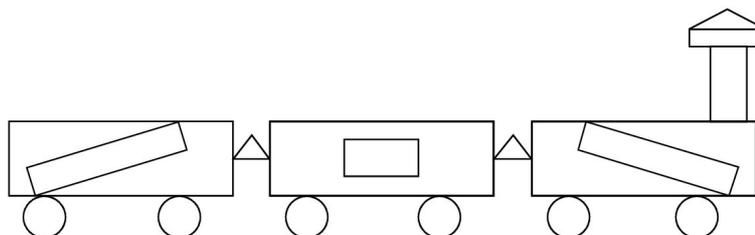
D1. Osserva la barchetta.



In tutto i triangoli sono:

- A. 3
- B. 5
- C. 7

D2. Osserva questa figura



a. Quanti rettangoli ci sono nella figura?

- A. 6
- B. 8
- C. 3

b. Quanti triangoli ci sono nella figura?

Risposta: triangoli

D3. Quale, tra queste figure, corrisponde alla seguente descrizione?

“È UN QUADRATO DENTRO UN CERCHIO”

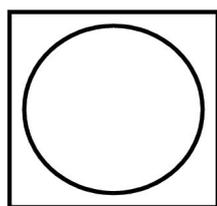


Figura 1

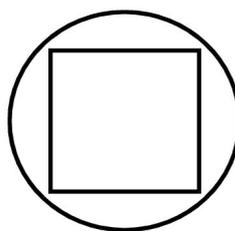


Figura 2

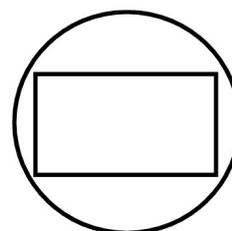
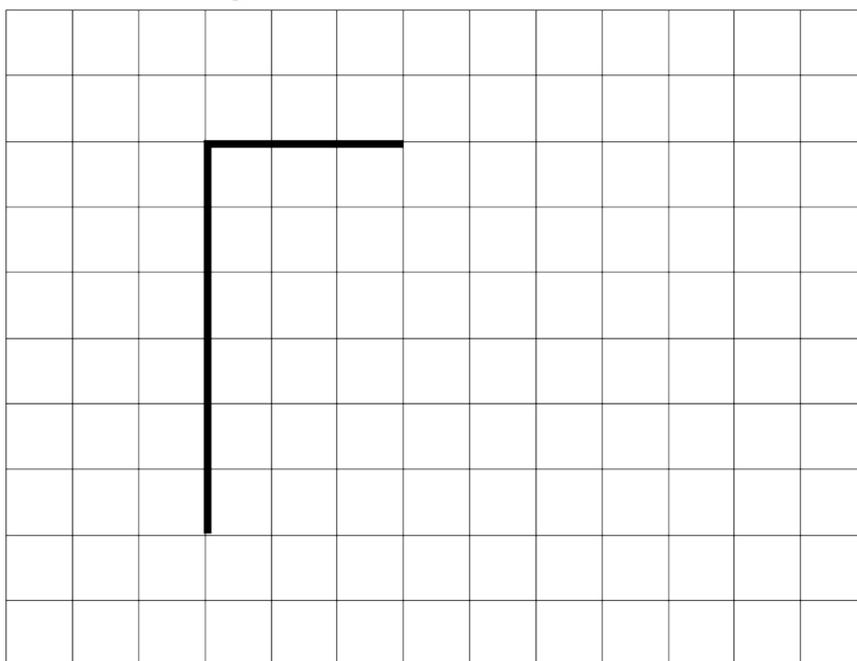


Figura 3

- A. Figura 1
- B. Figura 2
- C. Figura 3

D4. Osserva il disegno.

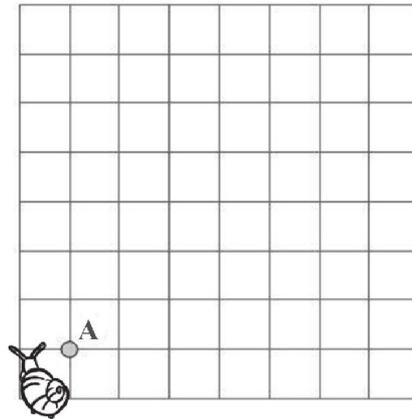


Completa il disegno in modo che la figura che ottieni sia un QUADRATO.

D5. La chiocciola parte dal punto A e segue il percorso indicato dai numeri e dalle frecce.

a. Disegna nella griglia il suo percorso.

6↑ 4→ 6↓ 4←

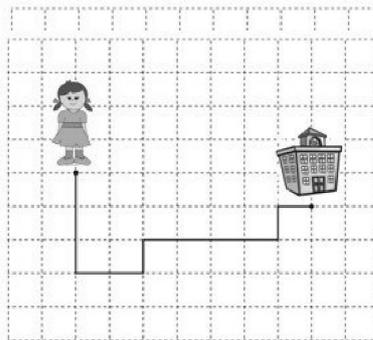


b. Quale forma ha il percorso della chiocciola?

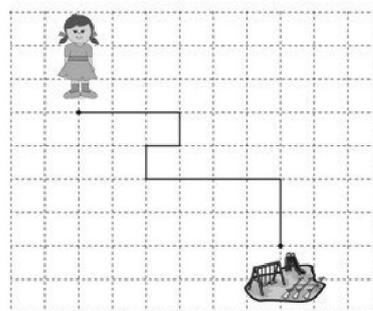
- A. Triangolo
- B. Rettangolo
- C. Quadrato

D6. Durante la giornata Anna fa questi percorsi.

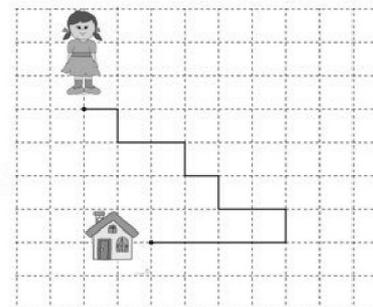
Percorso A



Percorso B



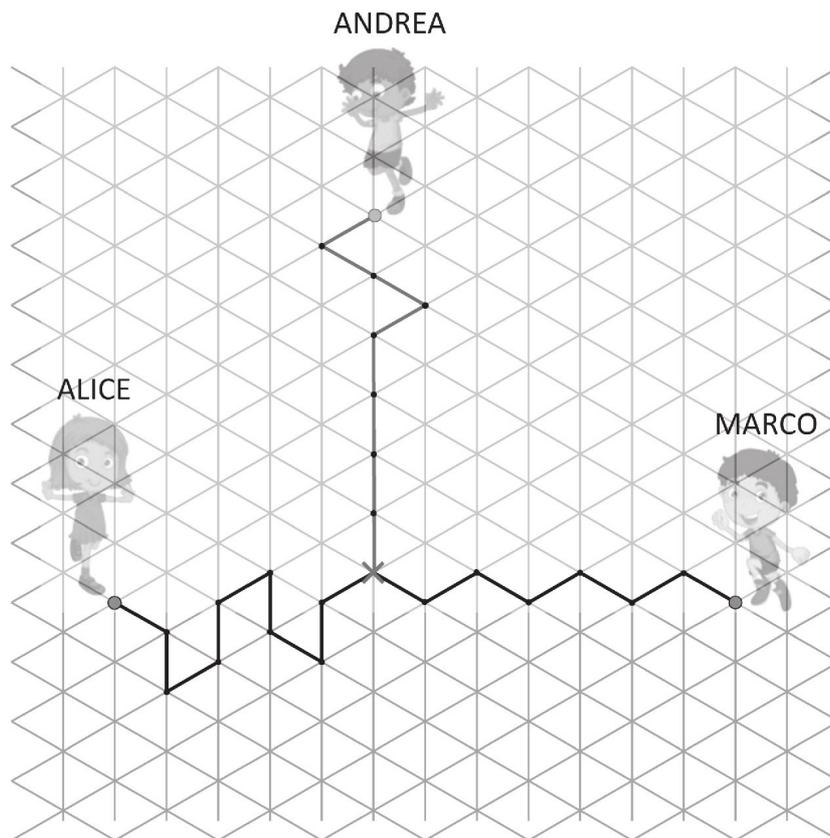
Percorso C



Qual è il percorso più lungo?

Risposta: il percorso

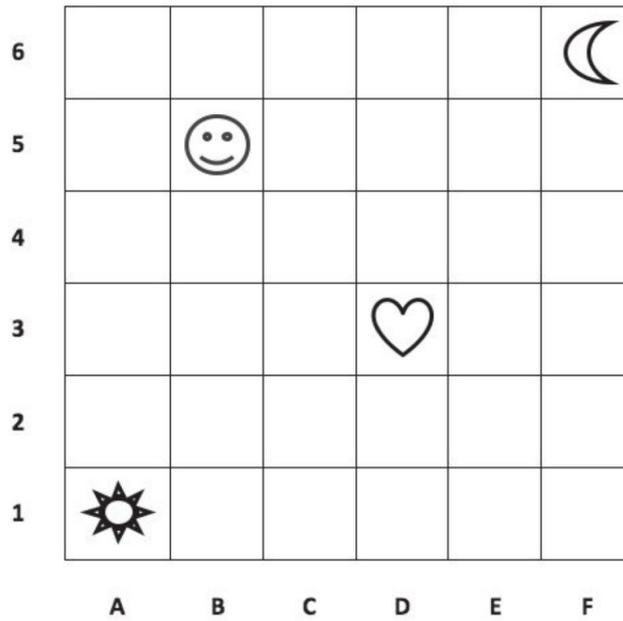
D7. Alice, Marco e Andrea giocano alla caccia al tesoro.



Chi fa il percorso più lungo per arrivare al tesoro (indicato con una X)?

- A. Alice
- B. Marco
- C. Andrea

D8. Osserva la seguente figura



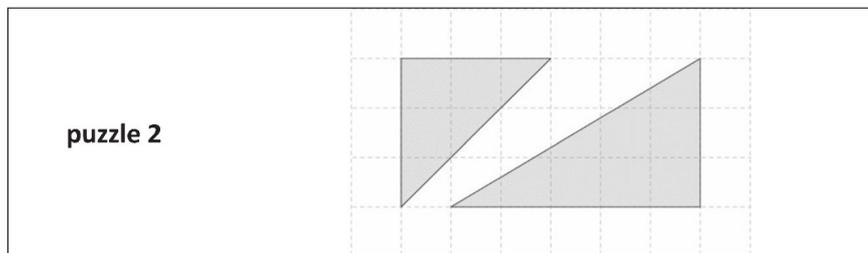
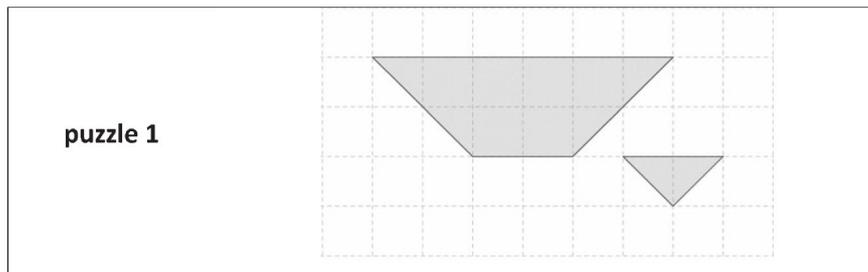
La  è nella posizione (F, 6).

a. In che posizione è il  ?

Risposta: (..... ,)

b. Disegna un  nella posizione (B, 3).

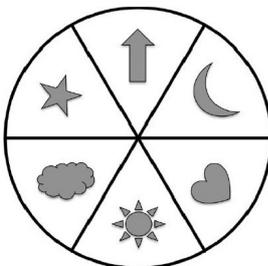
D9. Fabio ha disposizione questi tre puzzle, ognuno fatto da due pezzi.



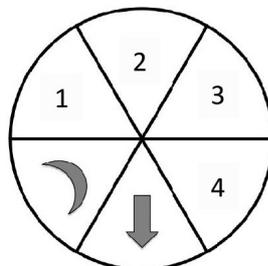
Con quale puzzle costruisce un rettangolo?

- A. Con il puzzle 1
- B. Con il puzzle 2
- C. Con il puzzle 3

D10. Osserva questa ruota.

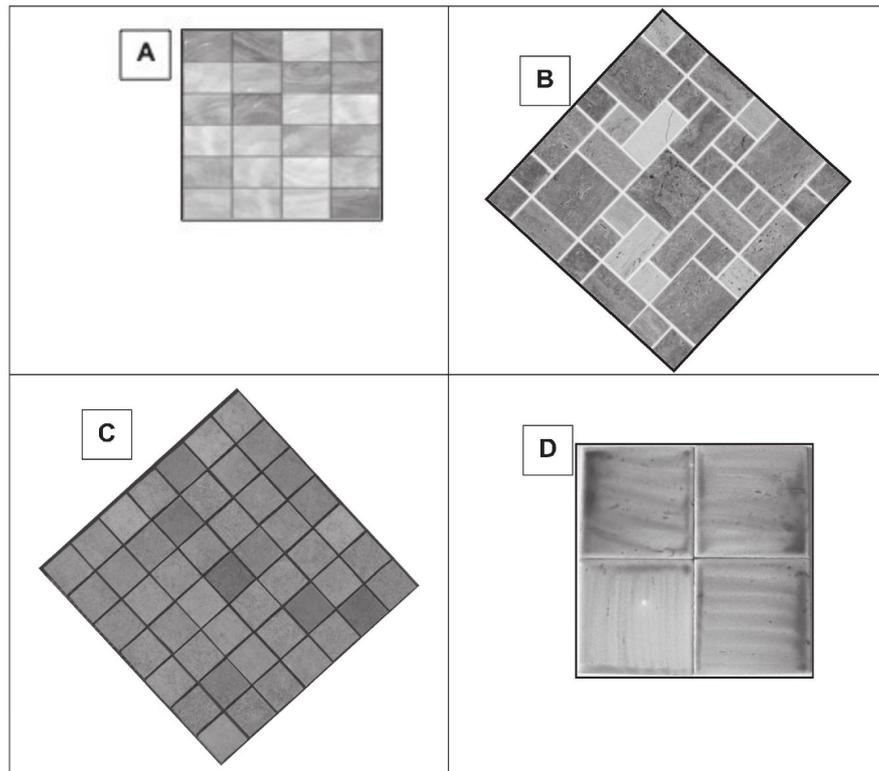


Matteo la fa girare e ora la freccia e la luna si trovano in questa posizione.



- a. Cosa c'è adesso nella posizione 4?
- A. Il simbolo 
 - B. Il simbolo 
 - C. Il simbolo 
- b. Dove si trova adesso il simbolo  ?
- A. Nella posizione 1
 - B. Nella posizione 2
 - C. Nella posizione 4

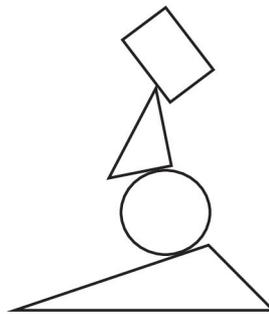
D11. Marta vuole fare una decorazione in cucina usando solo piastrelle di forma quadrata. Per decidere, Marta guarda questi tipi di decorazione.



Quali decorazioni sono formate solo da piastrelle di forma quadrata?

- A. Solo la decorazione D
- B. Le decorazioni C e D
- C. Le decorazioni A e B

D12. Osserva la figura.



Questa figura è formata da

- A. un cerchio, due triangoli e un quadrato
- B. un cerchio, due quadrati e un triangolo
- C. un cerchio, due triangoli e un rettangolo

Anexos H2 - Archivo de la prueba final

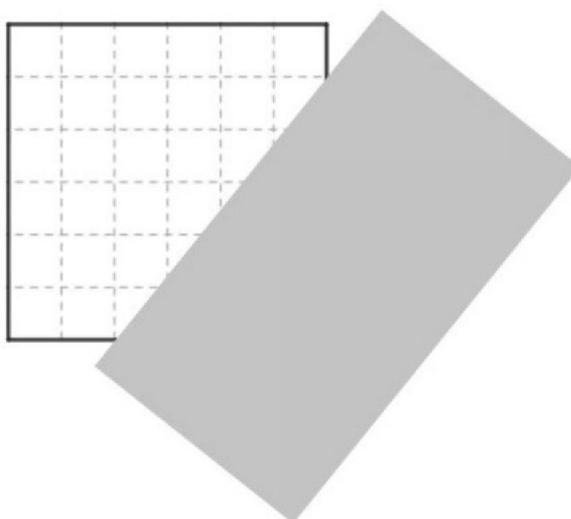
TEST DI GEOMETRIA G2

Nome _____ Cognome _____

Classe _____

Data _____

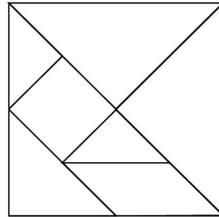
D1. Un quadrato e un rettangolo sono sovrapposti, come vedi in figura.



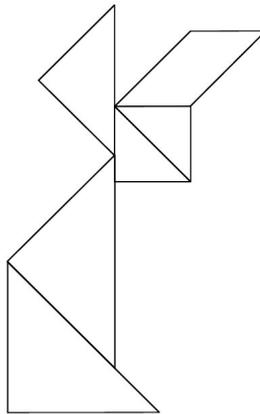
La parte del quadrato nascosta dal rettangolo ha la forma di un

- A. rettangolo
- B. trapezio rettangolo
- C. triangolo isoscele
- D. triangolo rettangolo

D2. Questo è il Tangram, un gioco cinese formato da sette pezzi.



a. Con i pezzi del Tangram, Giorgio ha costruito questo coniglietto, ma non ha usato uno dei pezzi.



Quale pezzo non ha usato Giorgio?

- A.
- B.
- C.

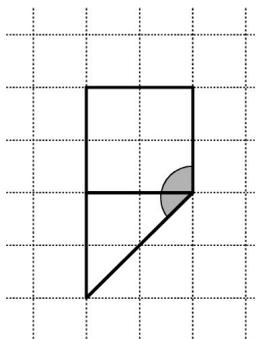
D3. Giorgio inizia gli allenamenti di pallavolo alle 16:30.



Quando finisce il riscaldamento l'orologio segna le 16:45.
Durante questo intervallo di tempo la lancetta dei minuti ha ruotato descrivendo un angolo

- A. acuto
- B. retto
- C. ottuso
- D. piatto

D4. Osserva la figura, formata da un quadrato e da un triangolo rettangolo isoscele.

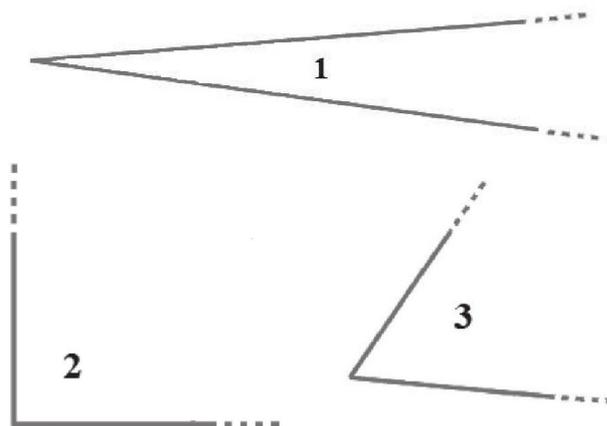


Quanto misura l'angolo evidenziato in grigio ?

Risposta: gradi

3

D5. Osserva gli angoli rappresentati qui sotto.



Qual è l'ordinamento corretto degli angoli dal più ampio al meno ampio?

- A. Angolo 2 – Angolo 3 – Angolo 1
- B. Angolo 1 – Angolo 3 – Angolo 2
- C. Angolo 3 – Angolo 1 – Angolo 2
- D. Angolo 3 – Angolo 2 – Angolo 1

D6. Osserva le seguenti figure.



Figura 1

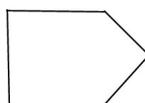


Figura 2

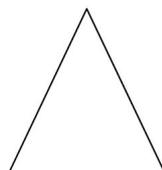


Figura 3

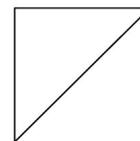


Figura 4

Quale corrisponde alla descrizione: *poligono con un solo angolo retto e con due soli lati uguali?*

- A. Solo la 4.
- B. Solo la 1.
- C. Solo la 1 e la 4.
- D. Solo la 2 e la 3.

D7. Osserva le seguenti figure

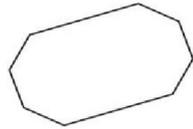


Figura 1

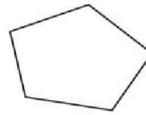


Figura 2

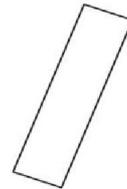


Figura 3

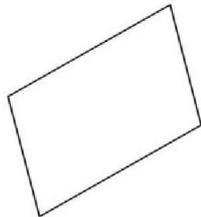


Figura 4

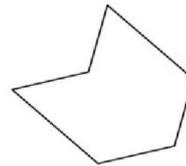


Figura 5

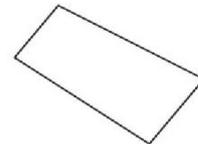


Figura 6

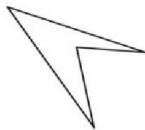


Figura 7

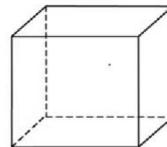
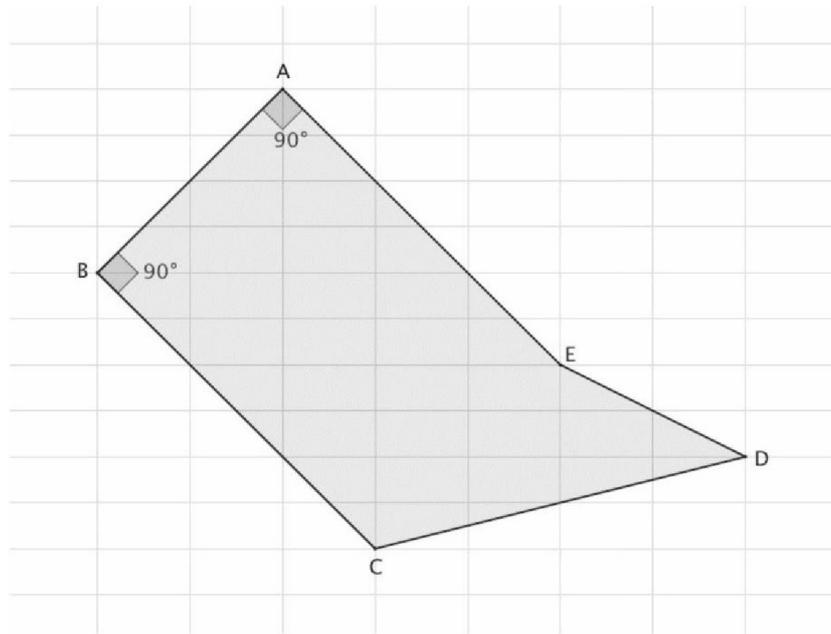


Figura 8

Quali sono i quadrilateri?

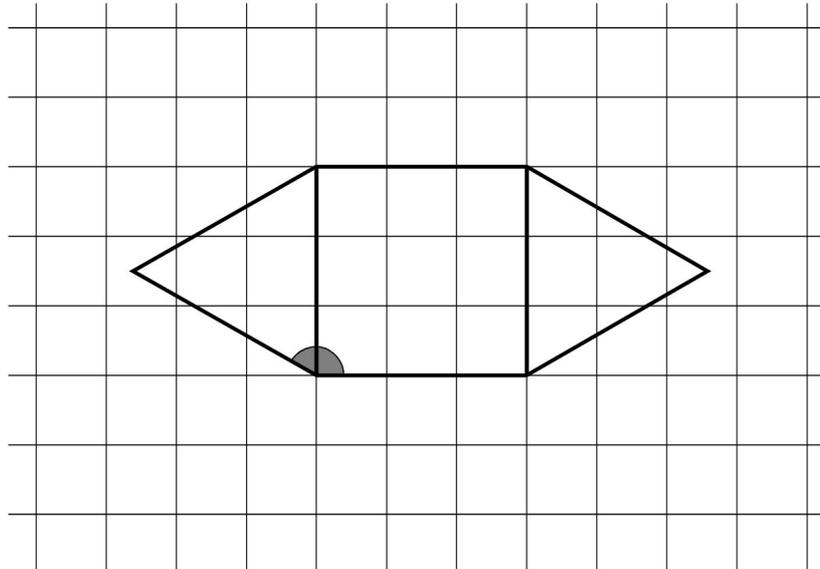
- A. Le figure: 3, 4, 6, 7.
- B. Le figure: 3, 4, 7, 8.
- C. Le figure: 1, 2, 3, 8.
- D. Le figure: 3, 4, 5, 6.

D8. Osserva la figura ABCDE.



Traccia un segmento sulla figura per scomporla in modo da ottenere un rettangolo e un triangolo.

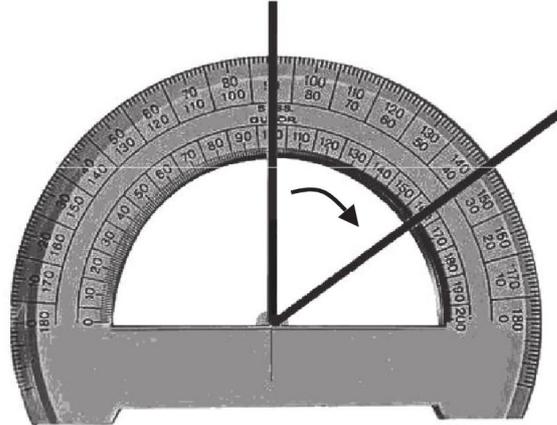
D9. Questa figura è formata da due triangoli equilateri e un quadrato.



Indica se ciascuna delle seguenti affermazioni è vera (V) o falsa (F).

		V	F
a.	La figura è un esagono	<input type="checkbox"/>	<input type="checkbox"/>
b.	La figura ha tutti i lati uguali	<input type="checkbox"/>	<input type="checkbox"/>
c.	L'angolo evidenziato in grigio misura 150°	<input type="checkbox"/>	<input type="checkbox"/>
d.	La figura ha un solo asse di simmetria	<input type="checkbox"/>	<input type="checkbox"/>

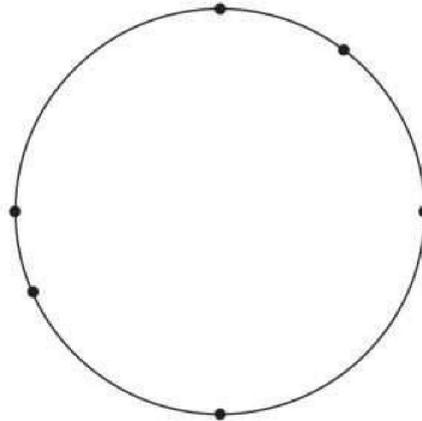
D10. Per misurare l'angolo rappresentato qui sotto, Francesco posiziona il goniometro nel modo che vedi.



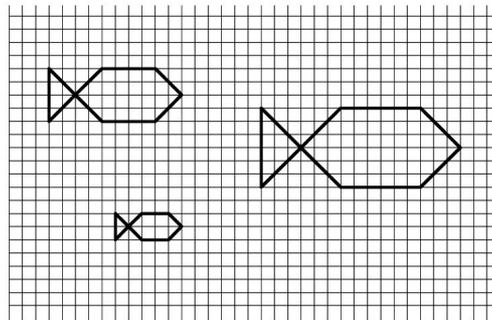
Quanto misura l'angolo?

- A. 35°
- B. 55°
- C. 90°
- D. 145°

D11. Disegna un quadrato che abbia come vertici quattro dei punti indicati sulla circonferenza.



D12. Alice ha disegnato tre pesciolini sul suo quaderno a quadretti.



a. Indica se ciascuna delle seguenti affermazioni è vera (V) o falsa (F).

	V	F
1. Le tre figure hanno la stessa forma		
2. Le tre figure hanno gli angoli corrispondenti della stessa ampiezza		
3. Il pesce più piccolo e il pesce più grande sono in scala 1 : 4		

b. Disegna sulla figura del pesce più grande il suo asse di simmetria.

Referencias

Referencias

- Aho, A. V. (2012). Computation and computational thinking. *The Computer Journal*, 55(7), 832-835. <https://doi.org/10.1093/comjnl/bxs074>
- Angeli, C., & Giannakos, M. (2020). Computational thinking education: Issues and challenges. *Computers in Human Behavior*, 105, 106185. <https://doi.org/10.1016/j.chb.2019.106185>
- Angeli, C., & Valanides, N. (2020). Developing young children's computational thinking with educational robotics: An interaction effect between gender and scaffolding strategy. *Computers in Human Behavior*, 105, 105954. <https://doi.org/10.1016/j.chb.2019.03.018>
- Angeli, C., Voogt, J., Fluck, A., Webb, M., Cox, M., Malyn-Smith, J., & Zagami, J. (2016). A K-6 computational thinking curriculum framework: Implications for teacher knowledge. *Journal of Educational Technology & Society*, 19(3), 47.
- Arfé, B., Vardanega, T., & Ronconi, L. (2020). The effects of coding on children's planning and inhibition skills. *Computers & Education*, 148, 103807. <https://doi.org/10.1016/j.compedu.2020.103807>
- Atmatzidou, S., & Demetriadis, S. (2016). Advancing students' computational thinking skills through educational robotics: A study on age and gender relevant differences. *Robotics and Autonomous Systems*, 75, 661-670. <https://doi.org/10.1016/j.robot.2015.10.008>
- Balanskat, A., & Engelhardt, K. (2014). *Computing our future: Computer programming and coding-priorities, school curricula and initiatives across europe*. European Schoolnet.
- Barak, M., & Assal, M. (2018). Robotics and STEM learning: Students' achievements in assignments according to the P3 task taxonomy—practice, problem solving, and projects. *International Journal of Technology and Design Education*, 28(1), 121-144. <https://doi.org/10.1007/s10798-016-9385-9>
- Barker, B. S., Nugent, G., & Grandgenett, N. (2008). Examining 4-H robotics and geospatial technologies in the learning of science, technology, engineering, and mathematics topics. *Journal of Extension*, 46(3), 1.

- Barr, D., Harrison, J., & Conery, L. (2011). Computational thinking: A digital age skill for everyone. *Learning & Leading with Technology*, 38(6), 20-23.
- Barr, V., & Stephenson, C. (2011). Bringing computational thinking to K-12: What is involved and what is the role of the computer science education community? *Acm Inroads*, 2(1), 48-54.
- Bell, T., Alexander, J., Freeman, I., & Grimley, M. (2009). Computer science unplugged: School students doing real computing without computers. *The New Zealand Journal of Applied Computing and Information Technology*, 13(1), 20-29.
- Bell, T., Witten, I., & Fellows, M. (2015). CS unplugged: An enrichment and extension programme for primary-aged students.
- Benton, L., Hoyles, C., Kalas, I., & Noss, R. (2017). Bridging primary programming and mathematics: Some findings of design research in England. *Digital Experiences in Mathematics Education*, 3(2), 115-138. <https://doi.org/10.1007/s40751-017-0028-x>
- Benton, L., Saunders, P., Kalas, I., Hoyles, C., & Noss, R. (2018). Designing for learning mathematics through programming: A case study of pupils engaging with place value. *International Journal of Child-Computer Interaction*, 16, 68-76. <https://doi.org/10.1016/j.ijcci.2017.12.004>
- Berrocso, J. V., Sánchez, M. R. F., & Arroyo, María del Carmen Garrido. (2015). El pensamiento computacional y las nuevas ecologías del aprendizaje. *Revista De Educación a Distancia*, (46)
- Bers, M. U., Flannery, L., Kazakoff, E. R., & Sullivan, A. (2014). Computational thinking and tinkering: Exploration of an early childhood robotics curriculum. *Computers & Education*, 72, 145-157. <https://doi.org/10.1016/j.compedu.2013.10.020>
- Bishop-Clark, C., Courte, J., Evans, D., & Howard, E. V. (2007). A quantitative and qualitative investigation of using Alice programming to improve confidence, enjoyment and achievement among non-majors. *Journal of Educational Computing Research*, 37(2), 193-207. <https://doi.org/10.2190/J8W3-74U6-Q064-12J5>.

- Bocconi, S., Chiocciariello, A., Dettori, G., Ferrari, A., & Engelhardt, K. (2016). *Developing Computational Thinking in Compulsory Education-Implications for Policy and Practice*, JRC Working Papers JRC104188, Joint Research Centre (Seville site).
- Bogliolo, A. (2016). *Coding in your classroom, now!*. Giunti editore.
- Bogliolo, A., Delpriori, S., Klopfenstein, L. C., & Paolini, B. D. (2016). Immersive coding: Innovative tools and formats for large-scale coding events. *EDULEARN16 Proceedings*.
- Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. Paper presented at the *Proceedings of the 2012 Annual Meeting of the American Educational Research Association, Vancouver, Canada*, 1-25.
- Brown, N. C. C., Kölling, M., Crick, T., Peyton Jones, S., Humphreys, S., & Sentance, S. (2013). Bringing computer science back into schools: Lessons from the UK. Ponencia presentada en el *Proceeding of the 44th ACM Technical Symposium on Computer Science Education*, 269-274. <https://doi.org/10.1145/2445196.2445277>
- Buitrago Flórez, F., Casallas, R., Hernández, M., Reyes, A., Restrepo, S., & Danies, G. (2017). *Changing a generation's way of thinking: Teaching computational thinking through programming*. Sage Publications Sage CA: Los Angeles, CA. <https://doi.org/10.3102/0034654317710096>
- Bundy, A. (2007). Computational thinking is pervasive. *Journal of Scientific and Practical Computing*, 1(2), 67-69.
- Burke, Q., & Kafai, Y. B. (2012). The writers' workshop for youth programmers: Digital storytelling with scratch in middle school classrooms. En *Proceedings of the 43rd ACM Technical Symposium on Computer Science Education*, 433-438. <https://doi.org/10.1145/2157136.2157264>
- Caballero-Gonzalez, Y., Muñoz-Repiso, A. G., & García-Holgado, A. (2019). Learning computational thinking and social skills development in young children through problem solving with educational robotics. En

- Proceedings of the Seventh International Conference on Technological Ecosystems for Enhancing Multiculturality*, 19-23. <https://doi.org/10.1145/3362789.3362874>
- Çakır, N. A., Gass, A., Foster, A., & Lee, F. J. (2017). *Development of a game-design workshop to promote young girls' interest towards computing through identity exploration*. Elsevier. <https://doi.org/10.1016/j.compedu.2017.02.002>
- Calao, L. A., Moreno-León, J., Correa, H. E., & Robles, G. (2015). Developing mathematical thinking with scratch. *Design for teaching and learning in a networked world* (pp. 17-27). Springer. https://doi.org/10.1007/978-3-319-24258-3_2
- Calvillo, A. (2014). Makey makey y scratch en el aula de música. *España: Instituto Nacional De Tecnologías Educativas Y De Formación Del Profesorado, Observatorio De Tecnología Educativa*. Recuperado de <https://Intef.Es/Wp-Content/Uploads/2019/02/MakeyMakey-1.Pdf>.
- Campbell, D. T., & Stanley, J. C. (1963). Experimental and quasi-experimental designs for research. *Handbook of Research on Teaching*, 5, 171-246.
- Capperucci, D. (2017). Valutazione degli apprendimenti e calcolo del valore aggiunto nelle rilevazioni nazionali per il primo ciclo d'istruzione. *Form@ Re*, 17(3).
- Carbonaro, M., Szafron, D., Cutumisu, M., & Schaeffer, J. (2010). Computer-game construction: A gender-neutral attractor to computing science. *Computers & Education*, 55(3), 1098-1111. <https://doi.org/10.1016/j.compedu.2010.05.007>
- Castells, M. (1999). *Information technology, globalization and social development*. UNRISD Geneva.
- Castells, M. (2014). *La nascita della società in rete*. EGEA spa.
- Cerini, G. (2012). Indicazioni nazionali per il curricolo della scuola dell'infanzia e del primo ciclo d'istruzione.
- Cetin, I. (2016). Preservice teachers' introduction to computing: Exploring utilization of scratch. *Journal of Educational Computing Research*, 54(7), 997-1021. <https://doi.org/10.1177/0735633116642774>.

- Chauhan, S. (2017). A meta-analysis of the impact of technology on learning effectiveness of elementary students. *Computers & Education*, 105, 14-30. <https://doi.org/10.1016/j.compedu.2016.11.005>
- Chen, G., Shen, J., Barth-Cohen, L., Jiang, S., Huang, X., & Eltoukhy, M. (2017). Assessing elementary students' computational thinking in everyday reasoning and robotics programming. *Computers & Education*, 109, 162-175. <https://doi.org/10.1016/j.compedu.2017.03.001>
- Chiazese, G., Fulantelli, G., Pipitone, V., & Taibi, D. (2018). Involucrando a los niños de educación primaria en el pensamiento computacional: Diseñando y desarrollando videojuegos. *Education in the Knowledge Society (EKS)*, 19(2), 63-81.
- Clements, D. H. (1986). Effects of logo and CAI environments on cognition and creativity. *Journal of Educational Psychology*, 78(4), 309-318.
- Clements, D. H., Battista, M. T., & Sarama, J. (2001). Logo and geometry. *Journal for Research in Mathematics Education. Monograph*, 10, i-177.
- Code.org. (2013, feb, 26). *What Most Schools Don't Teach* [Video] <https://www.youtube.com/watch?v=nKlu9yen5nc>.
- Cohen, J. (1988). *Statistical power analysis* (2nd ed.). Hillsdale NJ: Erlbaum.
- Cohen, L., Manion, L., & Morrison, K. (2013). *Research methods in education*. routledge.
- Comisión Europea (2016). *Comunicato stampa*. https://ec.europa.eu/commission/presscorner/detail/it/IP_16_4081.
- Comisión Europea (2018). *Comunicazione della Commissione al Parlamento Europeo, al Consiglio, al Comitato delle Regioni sul piano d'azione per l'istruzione digitale*. <https://eur-lex.europa.eu/legal-content/IT/TXT/PDF/?uri=CELEX:52018DC0022&from=EN>.
- Consejo de la Unión Europea (2018). *Recomendación del Consejo de 22 de mayo de 2018 sobre las competencias clave para el aprendizaje permanente* [https://eur-lex.europa.eu/legalcontent/ES/TXT/PDF/?uri=CELEX:32018H0604\(01\)&from=SV](https://eur-lex.europa.eu/legalcontent/ES/TXT/PDF/?uri=CELEX:32018H0604(01)&from=SV).

- Cooper, H., Camic, P. M., Long, D., Panter, A., Rindskof, D., & Sher, K. (2012). *The APA handbook of research methods in psychology (volumes 1-3)*. American Psychological Association.
- Corbetta, P. (2003). *La ricerca sociale: Metodologia e tecniche*. Il mulino.
- Corradini, I., Lodi, M., & Nardelli, E. (2017). Computational thinking in italian schools: Quantitative data and teachers' sentiment analysis after two years of. En *Proceedings of the 2017 ACM Conference on Innovation and Technology in Computer Science Education*, 224-229.
- Corsi, D., Domínguez, F. I. R., & Rodríguez, M. I. P. Adquisición de competencias emocionales mediante el desarrollo y uso de serious games en educación superior. *Píxel-Bit. Revista De Medios Y Educación*, 56, 95-112. <https://doi.org/10.12795/pixelbit.2019.i56.05>.
- Corsini, C. (2013). La validità di contenuto delle prove INVALSI di comprensione della lettura. *Italian Journal of Educational Research*, (10), 46-61.
- Creswell, J. W. (2002). *Educational research: Planning, conducting, and evaluating quantitative*. Prentice Hall Upper Saddle River, NJ.
- Crompton, H., Burke, D., & Lin, Y. (2019). Mobile learning and student cognition: A systematic review of PK-12 research using bloom's taxonomy. *British Journal of Educational Technology*, 50(2), 684-701. <https://doi.org/10.1111/bjet.12674>.
- Crompton, H., Grant, M. R., & Shraim, K. Y. (2018). Technologies to enhance and extend children's understanding of geometry: A configurative thematic synthesis of the literature. *Journal of Educational Technology & Society*, 21(1), 59-69.
- Crow, D. (2014). Why every child should learn to code. *The Guardian*, 7.
- CSTA (2017). *Progression of CSTA K-12 Computer Science Standards*. <https://www.esd105.org/cms/lib/WA01920102/Centricity/Domain/175/2017%20CSTA%20K-12%20Standards%20Progression%20Chart%20CT.pdf>.

- CSTA (2019). *School Counselors Prepare Students for 21st Century Computational Thinking Skills*.
<https://www.csteachers.org/Stories/school-counselors-prepare-students-for-21st-century-computational-thinking-skills>.
- Cuny, J., Snyder, L., & Wing, J. (2010). Demystifying computational thinking for non-computer scientists, work in progress.
- Curzon, P. (2013). Cs4fn and computational thinking unplugged. En *Proceedings of the 8th Workshop in Primary and Secondary Computing Education*, 47-50. <https://doi.org/10.1145/2532748.2611263>
- Curzon, P., McOwan, P. W., Plant, N., & Meagher, L. R. (2014, November). Introducing teachers to computational thinking using unplugged storytelling. En *Proceedings of the 9th workshop in primary and secondary computing education* (pp. 89-92). <https://doi.org/10.1145/2670757.2670767>
- Dagiene, V., & Stupuriene, G. (2016). Bebras—a sustainable community building model for the concept based learning of informatics and computational thinking. *Informatics in Education*, 15(1), 25-44.
- Damiani, P., Grimaldi, R., & Palmieri, S. (2013). Robotica educativa e aspetti non verbali nei disturbi specifici di apprendimento. Paper presented at the *Didamatica 2013. Tecnologie E Metodi Per La Didattica Del Futuro*, 1, 1211-1220. AICA-Scuola Superiore Sant'Anna-CNR Pisa.
- De Gregorio, E., & Lattanzi, P. F. (2012). *Programmi per la ricerca qualitativa. guida pratica all'uso di ATLAS. ti e MAXQDA: Guida pratica all'uso di ATLAS. ti e MAXQDA*. FrancoAngeli.
- del Olmo-Muñoz, J., Cózar-Gutiérrez, R., & González-Calero, J. A. (2020). Computational thinking through unplugged activities in early years of primary education. *Computers & Education*, 150, 103832. <https://doi.org/10.1016/j.compedu.2020.103832>
- Denner, J., Werner, L., & Ortiz, E. (2012). Computer games created by middle school girls: Can they be used to measure understanding of computer science concepts?. *Computers & Education*, 58(1), 240-249.
- Denning, P. J. (2017). Remaining trouble spots with computational thinking. *Communications of the ACM*, 60(6), 33-39. <https://doi.org/10.1016/j.compedu.2011.08.006>

- Denning, P. J. (2017). Remaining trouble spots with computational thinking. *Communications of the ACM*, 60(6), 33-39.
- Du, J., Wimmer, H., & Rada, R. (2016). "Hour of code": Can it change students' attitudes toward programming? *Journal of Information Technology Education: Innovations in Practice*, 15, 52-73.
<https://doi.org/10.28945/3421>
- Durak, H. Y., & Saritepeci, M. (2018). Analysis of the relation between computational thinking skills and various variables with the structural equation model. *Computers & Education*, 116, 191-202.
<https://doi.org/10.1016/j.compedu.2017.09.004>
- Falcinelli, F., Filomia, M., & Sabatini, M. (2017). Formare i futuri docenti al pensiero computazionale attraverso un approccio laboratoriale. *Direttore Scientifico: Pierpaolo Limone (Università Di Foggia) Comitato Scientifico E Di Referaggio: José Luis Rodríguez Illera (Università Di Barcellona)*, 297.
- Falloon, G. (2016). An analysis of young students' thinking when completing basic coding tasks using Scratch Jr. On the iPad. *Journal of Computer Assisted Learning*, 32(6), 576-593.
<https://doi.org/10.1111/jcal.12155>.
- Fessakis, G., Gouli, E., & Mavroudi, E. (2013). *Problem solving by 5–6 years old kindergarten children in a computer programming environment: A case study*. Elsevier.
<https://doi.org/10.1016/j.compedu.2012.11.016>
- Foerster, E., Foerster, K., & Loewe, T. (2018). Teaching programming skills in primary school mathematics classes: An evaluation using game programming. <https://doi.org/10.1109/EDUCON.2018.8363411>
- Foerster, K. (2016). Integrating programming into the mathematics curriculum: Combining scratch and geometry in grades 6 and 7. En *Proceedings of the 17th Annual Conference on Information Technology Education*, 91-96. <https://doi.org/10.1145/2978192.2978222>

- Foerster, K. (2017). Teaching spatial geometry in a virtual world: Using minecraft in mathematics in grade 5/6. Ponencia presentada en el *Global Engineering Education Conference (EDUCON), 2017 IEEE*, 1411-1418. <https://doi.org/10.1109/EDUCON.2017.7943032>
- Fraenkel, J. R., Wallen, N. E., & Hyun, H. H. (2011). *How to design and evaluate research in education*. New York: McGraw-Hill Humanities/Social Sciences/Languages.
- Fraillon, J., Ainley, J., Schulz, W., Duckworth, D., & Friedman, T. (2019). *IEA international computer and information literacy study 2018 assessment framework (p.74)*. Springer Nature. <https://doi.org/10.1007/978-3-030-19389-8>.
- Furber, S. (2012). Shut down or restart? the way forward for computing in UK schools. *The Royal Society, London*, Royal Society.
- García-Peñalvo, F. J., Rees, A. M., Hughes, J., Jormanainen, I., Toivonen, T., & Vermeersch, J. (2016). A survey of resources for introducing coding into schools. En *Proceedings of the Fourth International Conference on Technological Ecosystems for Enhancing Multiculturality*, 19-26. <https://doi.org/10.1145/3012430.3012491>
- García- Peñalvo, F. J. (2016). A brief introduction to TACCLE 3—coding european project. En *2016 International Symposium on Computers in Education (SIIE)*, 1-4. <https://doi.org/10.1109/SIIE.2016.7751876>
- García-Peñalvo, F. J., & Mendes, A. J. (2018). No title. *Exploring the Computational Thinking Effects in Pre-University Education*, (SIIE) (pp. 1-4). IEEE. . <https://doi.org/10.1109/SIIE.2016.7751876>
- García-Peñalvo, F. J., Reimann, D., & Maday, C. (2018). *Introducing coding and computational thinking in the schools: The TACCLE 3—coding project experience*. Springer. https://doi.org/10.1007/978-3-319-93566-9_11
- Gay, L. R., Mills, G. E., & Airasian, P. W. (2009). *Educational research: Competencies for analysis and applications*. Merrill/Pearson.

- Glaser, B. G., & Strauss, A. L. (2017). *Discovery of grounded theory: Strategies for qualitative research*. Routledge.
- Gonzalez, H. B., & Kuenzi, J. (2012). *Science, technology, engineering, and mathematics (STEM) education: A primer*. Washington, DC: Congressional Research Service, Library of Congress.
- Greenberger, M. (1964). *The computers of tomorrow*. Atlantic Monthly.
- Grover, S., Cooper, S., & Pea, R. (Jun 21, 2014). Assessing computational learning in K-12.. <https://doi.org/10.1145/2591708.2591713> <http://dl.acm.org/citation.cfm?id=2591713>
- Grover, S., & Pea, R. (2013). Computational thinking in K-12. *Educational Researcher*, 42(1), 38-43. <https://doi.org/10.3102/0013189X12463051>
- Grover, S., Pea, R., & Cooper, S. (2015). Designing for deeper learning in a blended computer science course for middle school students. *Computer Science Education*, 25(2), 199-237. <https://doi.org/10.1080/08993408.2015.1033142>
- Harvard Graduate School of Education (s.f.). *Creative Computing curriculum*. https://docs.google.com/presentation/d/e/2PACX-1vTejqFgcPqVm6X5Kbi54b-otyJps8gg43W6Lhk5036ZYvROPX_ywq35tgLkjhkVUv2b3RPGzBVL-4jg/pub?start=false&loop=false&delayms=3000&slide=id.g4b50e4d41c_0_0.
- Heckman, J. J. (2006). Skill formation and the economics of investing in disadvantaged children. *Science*, 312(5782), 1900-1902. <https://doi.org/10.1126/science.1128898>
- Helme, S., & Clarke, D. (2001). Identifying cognitive engagement in the mathematics classroom. *Mathematics Education Research Journal*, 13(2), 133-153.
- Highfield, K. (2014). Stepping into STEM with young children: Simple robotics and programming as catalysts for early learning. *Technology and digital media in the early years* (pp. 182-193). Routledge.
- Howland, K., & Good, J. (2015). Learning to communicate computationally with Flip: A bi-modal programming language for game creation. *Computers & Education*, 80, 224-240. <https://doi.org/10.1016/j.compedu.2014.08.014>

- Hsu, T., & Hu, H. (2017). Application of the four phases of computational thinking and integration of blocky programming in a sixth-grade mathematics course. *Siu-Cheung KONG the Education University of Hong Kong, Hong Kong, 73*.
- Hsu, T. C., Chang, S. C., & Hung, Y. T. (2018). How to learn and how to teach computational thinking: Suggestions based on a review of the literature. *Computers & Education, 126*, 296-310. <https://doi.org/10.1016/j.compedu.2018.07.004>
- Idin, S. (2020). New trends in science education within the 21st century skills perspective. *Education Research Highlights in Mathematics, Science and Technology*, 150-159.
- Illomäki, L., Paavola, S., Lakkala, M., & Kantosalo, A. (2016). Digital competence—an emergent boundary concept for policy and educational research. *Education and Information Technologies, 21*(3), 655-679.
- INVALSI (2015). *Indagini IEA 2015 Timms: i risultati degli studenti italiani in matematica e scienze*. https://www.invalsi.it/invalsi/ri/timss2015/documenti/Rapporto_nazionale_TIMSS_2015.pdf.
- INVALSI (2018). *Quadro di riferimento delle prove di INVALSI matematica*. https://invalsi-areaprove.cineca.it/docs/file/QdR_MATEMATICA.pdf.
- Israel, M., Pearson, J. N., Tapia, T., Wherfel, Q. M., & Reese, G. (2015). Supporting all learners in school-wide computational thinking: A cross-case qualitative analysis. *Computers & Education, 82*, 263-279. <https://doi.org/10.1016/j.compedu.2014.11.022>.
- ISTE, C. (2011). No title. *Computational Thinking in K–12 Education Leadership Toolkit*, Computer Science Teacher Association: http://csta.acm.org/Curriculum/sub/CurrFiles/471.11_CTLeadershipToolkit-SP-vF.pdf adresinden alındı.
- ISTE (2019). *CT Competencies: Embrace integration across the curriculum*. <https://www.iste.org/explore/empowered-learner/ct-competencies-embrace-integration-across-curriculum>.

- Jaipal-Jamani, K., & Angeli, C. (2017). Effect of robotics on elementary preservice teachers' self-efficacy, science learning, and computational thinking. *Journal of Science Education and Technology*, 26(2), 175-192.
- Jenson, J., & Droumeva, M. (2015). Making games with game maker: A computational thinking curriculum case study. Ponencia presentada en el *ECGBL2015-9th European Conference on Games Based Learning: ECGBL2015*, 260.
- Jona, K., Wilensky, U., Trouille, L., Horn, M. S., Orton, K., Weintrop, D., & Beheshti, E. (2014). Embedding computational thinking in science, technology, engineering, and math (CT-STEM). Ponencia presentada en el *Future Directions in Computer Science Education Summit Meeting, Orlando, FL*.
- K-12 Computer Science Framework. (2016). <http://www.k12cs.org>.
- Kafai, Y. B., & Burke, Q. (2013, March). The social turn in K-12 programming: moving from computational thinking to computational participation. In *Proceeding of the 44th ACM technical symposium on computer science education* (pp. 603-608).
- Kafai, Y. B., & Burke, Q. (2015). Constructionist gaming: Understanding the benefits of making games for learning. *Educational Psychologist*, 50(4), 313-334. [:https://doi.org/10.1080/00461520.2015.1124022](https://doi.org/10.1080/00461520.2015.1124022).
- Kafai, Y. B., Lee, E., Searle, K., Fields, D., Kaplan, E., & Lui, D. (2014). A crafts-oriented approach to computing in high school: Introducing computational concepts, practices, and perspectives with electronic textiles. *ACM Transactions on Computing Education (TOCE)*, 14(1), 1-20. <https://doi.org/10.1145/2576874>.
- Kafai, Y. B., & Proctor, C. (2021). A Reevaluation of Computational Thinking in K-12 Education: Moving Toward Computational Literacies. *Educational Researcher*, <https://doi.org/10.3102/0013189X211057904>.
- Kalelioğlu, F. (2015). A new way of teaching programming skills to K-12 students: Code. org. *Computers in Human Behavior*, 52, 200-210. <https://doi.org/10.1016/j.chb.2015.05.047>.
- Kalelioglu, F., Gülbahar, Y., & Kukul, V. (2016). A framework for computational thinking based on a systematic research review. *Baltic Journal of Modern Computing*, 4(3), 583.

- Kalelioglu, F., & Gülbahar, Y. (2014). The effects of teaching programming via scratch on problem solving skills: A discussion from learners' perspective. *Informatics in Education, 13*(1), 33-50.
- Kalelioğlu, F., Gülbahar, Y., Akçay, S., & Doğan, D. (2014). Curriculum integration ideas for improving the computational thinking skills of learners through programming via scratch. Ponencia presentada en el *Local Proceedings of the 7th International Conference on Informatics in Schools: Situation, Evolution and Perspectives*, 101-112.
- Kanbul, S., & Uzunboylu, H. (2017). Importance of coding education and robotic applications for achieving 21st-century skills in north cyprus. *International Journal of Emerging Technologies in Learning (IJET)*, 12(01), 130-140.
- Kazakoff, E., & Bers, M. (2012). Programming in a robotics context in the kindergarten classroom: The impact on sequencing skills. *Journal of Educational Multimedia and Hypermedia, 21*(4), 371-391. <https://www.learntechlib.org/p/39512>.
- Kazimoglu, C., Kiernan, M., Bacon, L., & Mackinnon, L. (2012). A serious game for developing computational thinking and learning introductory computer programming. *Procedia-Social and Behavioural Sciences, 47*, 1991-1999. <https://doi.org/10.1016/j.sbspro.2012.06.938>.
- Ke, F. (2014). An implementation of design-based learning through creating educational computer games: A case study on mathematics learning during design and computing. *Computers & Education, 73*, 26-39. <https://doi.org/10.1016/j.compedu.2013.12.010>.
- Kerlinger, F. N. (1970). A social attitude scale: Evidence on reliability and validity. *Psychological Reports, 26*(2), 379-383.
- Kerlinger, F.N. & Lee, H.B. (2002) *Investigación del Comportamiento: Métodos de Investigación en ciencias sociales*, México: McGraw-Hill Interamericana Editores.

- Kim, B., Kim, T., & Kim, J. (2013). And-pencil programming strategy toward computational thinking for non-majors: Design your solution. *Journal of Educational Computing Research*, 49(4), 437-459. <https://doi.org/10.2190/EC.49.4.b>.
- Kim, J. A., & Kim, H. J. (2017). Flipped learning of scratch programming with code. org. Ponencia presentada en el *Proceedings of the 2017 9th International Conference on Education Technology and Computers*, 68-72. <https://doi.org/10.1145/3175536.3175542>.
- Kipp, A., & Schneider, S. (2017). *Applied social Robotics—Building interactive robots with LEGO mindstorms*. Springer.
- Knuth, D. E. (1974). Computer programming as an art. *Cacm*, <http://www.paulgraham.com/knuth.html>.
- Kong, S., & Abelson, H. (2019). *Computational thinking education*. Springer. <https://doi.org/10.1007/978-981-13-6528-7>.
- Korkmaz, akir, R., & Özden, M. Y. (2017). A validity and reliability study of the computational thinking scales (CTS). *Computers in Human Behavior*, 72, 558-569. <https://doi.org/10.1016/j.chb.2017.01.005>.
- Kucuk, S., & Sisman, B. (2017). *Behavioral patterns of elementary students and teachers in one-to-one robotics instruction*. Elsevier. <https://doi.org/10.1016/j.compedu.2017.04.002>.
- Labusch, A., & Eickelmann, B. (2017). Computational thinking as a key competence-a research concept. Paper Ponencia presentada en el *Conference Proceedings of International Conference on Computational Thinking Education*, 103-106.
- Lammer, L., Lepuschitz, W., Kynigos, C., Giuliano, A., & Girvan, C. (2017). ER4STEM educational robotics for science, technology, engineering and mathematics. *Robotics in education* (pp. 95-101). Springer. https://doi.org/10.1007/978-3-319-42975-5_9.
- Lawshe, C. H. (1975). A quantitative approach to content validity 1. *Personnel Psychology*, 28(4), 563-575.
- Lee, Y. (2011). Scratch: Multimedia programming environment for young gifted learners. *Gifted Child Today*, 34(2), 26-31.

- Lewis, C. M., & Shah, N. (2012). Building upon and enriching grade four mathematics standards with programming curriculum. Paper presented at the *Proceedings of the 43rd ACM Technical Symposium on Computer Science Education*, 57-62. <https://doi.org/10.1145/2157136.2157156>
- Lichtenfeld, S., Pekrun, R., Stupnisky, R. H., Reiss, K., & Murayama, K. (2012). Measuring students' emotions in the early years: The achievement emotions questionnaire-elementary school (AEQ-ES). *Learning and Individual Differences*, 22(2), 190-201. <https://doi.org/10.1016/j.lindif.2011.04.009>.
- Livingstone, S. (2012). Critical reflections on the benefits of ICT in education. *Oxford Review of Education*, 38(1), 9-24.
- López, J. M. S., & Gutiérrez, R. C. (2017). Pensamiento computacional y programación visual por bloques en el aula de primaria. *Educar*, 53(1), 129-146. <https://raco.cat/index.php/Educar/article/view/317274>.
- Lu, J. J., & Fletcher, G. H. (2009). Thinking about computational thinking. *ACM SIGCSE Bulletin*, 41(1), 260-264.
- Lye, S. Y., & Koh, J. H. L. (2014). Review on teaching and learning of computational thinking through programming: What is next for K-12? *Computers in Human Behavior*, 41, 51-61. <https://doi.org/10.1016/j.chb.2014.09.012>.
- Malan, D. J., & Leitner, H. H. (2007). Scratch for budding computer scientists. Paper presented at the *ACM Sigcse Bulletin*, 39(1) 223-227. <https://doi.org/10.1145/1227504.1227388>
- Maloney, J. H., Peppler, K., Kafai, Y., Resnick, M., & Rusk, N. (2008). *Programming by choice: Urban youth learning programming with scratch*. ACM. <https://doi.org/10.1145/1352135.1352260>.
- Maloney, J., Resnick, M., Rusk, N., Silverman, B., & Eastmond, E. (2010). The scratch programming language and environment. *ACM Transactions on Computing Education (TOCE)*, 10(4), 16. <https://doi.org/10.1145/1868358.1868363>.
- Mannila, L., Dagiene, V., Demo, B., Grgurina, N., Mirolo, C., Rolandsson, L., & Settle, A. (2014). Computational thinking in K-9 education. Ponencia presentada en el *Proceedings of the Working Group Reports of the*

- 2014 on Innovation & Technology in Computer Science Education Conference, 1-29.
<https://doi.org/10.1145/2713609.2713610>.
- Mayerová, K., & Veselovská, M. (2017). How to teach with LEGO WeDo at primary school. *Robotics in education* (pp. 55-62). Springer. https://doi.org/10.1007/978-3-319-42975-5_5
- McMillan, J. H., Schumacher, S., & Sánchez Baidés, J. (2005). *Investigación educativa: una introducción conceptual* (5a ed.). Madrid [etc.]: Pearson. Recuperado de http://uves.summon.serialssolutions.com/2.0.0/link/0/eLvHCXMwY2AQntIz0EUrE0AaoRZ-kSYOYOuVIKfiQTe8IKsV1qml1qMGJUwszA3sjDVAoY0M-hmJtBCPW8DP1gqAVZoJhYGiE4M6KoI5N2VoK2dwHodLm9saAY6hwp8OZ-JEbCFAmyBQl_egfMNwWd-phYjVS1uggwsoO0GQgxMqcUiDBLwYy-SMw9vzIOAnKuaWZYoyiDr5hri7KEL0h8PHXOJh3miwkiMgTcRtFY9rwS8py1FgkEh1djMONUYWFGnmFiYpBqYJxobJicaGqYYpSWaJieapkyiGE3TAqXhDQDF_hIUfDQgAwDaxow6abKQvwbBw49AJz2aLI.
- Mengual-Andrés, S., Roig-Vila, R., & Mira, J. B. (2016). Delphi study for the design and validation of a questionnaire about digital competences in higher education. *International Journal of Educational Technology in Higher Education*, 13(1), 12. <https://doi.org/10.1186/s41239-016-0009-y>.
- Mengual-Andrés, S., Vázquez-Cano, E., & Meneses, E. L. (2017). La productividad científica sobre MOOC: Aproximación bibliométrica 2012-2016 a través de SCOPUS. *RIED.Revista Iberoamericana De Educación a Distancia*, 20(1), 39-58.
- Miller, J. (2019). STEM education in the primary years to support mathematical thinking: Using coding to identify mathematical structures and patterns. *Zdm*, 51(6), 915-927. <https://doi.org/10.1007/s11858-019-01096-y>
- Miller, J., & Larkin, K. (2017). Using Coding to Promote Mathematical Thinking with Year 2 Students: Alignment with the Australian Curriculum. *Mathematics Education Research Group of Australasia*.

- Ministero della pubblica istruzione. (2007). *Indicazioni per il curricolo: Per la scuola dell'infanzia e per il primo ciclo d'istruzione*. Ministero della pubblica istruzione. Italia. https://archivio.pubblica.istruzione.it/normativa/2007/allegati/dir_310707.pdf.
- Mishra, P., Yadav, A., & Deep-Play Research Group. (2013). Rethinking technology & creativity in the 21st century. *TechTrends*, 57(3), 10-14.
- MIUR (2014, sept, 23). *Nota prot. 2937 del 23/09/2014*. https://www.istruzione.it/allegati/2014/prot2937_14.pdf.
- MIUR (2014). *La buona scuola*. https://www.miur.gov.it/documents/20182/254335/La_Buona_Scuola_Approfondimenti.pdf/c52f6685-5ed2-41c0-9320-68fa0e548f51?version=1.0&t=1495630889159.
- MIUR (2015, julio, 2). *Legge n. 107 del 13/07/2015*. <https://www.gazzettaufficiale.it/eli/id/2015/07/15/15G00122/sg>.
- MIUR (2017, mar, 3). *Avviso pubblico 2669 del 03 marzo 2017*. https://www.istruzione.it/pon/avviso_cittadinanza-creativita.html.
- MIUR (2017, mayo, 19). *Nota 22272 del 19 maggio 2017 – Piano triennale di formazione docenti 2016*. https://www.istruzione.it/allegati/2017/CIRCOLARE_PIATTAFORMA_SOFIA.pdf.
- MIUR (2018, feb, 22). *Indicazioni nazionali e nuovi scenari*. <http://www.indicazioninazionali.it/wp-content/uploads/2018/08/Indicazioni-nazionali-e-nuovi-scenari.pdf>.
- Montiel, H., & Gomez-Zermeño, M. G. (2021). Educational challenges for computational thinking in K–12 education: A systematic literature review of “Scratch” as an innovative programming tool. *Computers*, 10(6), 69. <https://doi.org/10.3390/computers10060069>.
- Monroy-Hernández, A., & Resnick, M. (2008). FEATURE empowering kids to create and share programmable media. *Interactions*, 15(2), 50-53.

- Moreno-León, J., & Robles, G. (2016, April). Code to learn with Scratch? A systematic literature review. In *2016 IEEE Global Engineering Education Conference (EDUCON)* (pp. 150-156). IEEE. <https://doi.org/10.1109/EDUCON.2016.7474546>.
- Moreno-León, J., Román-González, M., Hartevelde, C., & Robles, G. (2017). On the automatic assessment of computational thinking skills: A comparison with human experts. En *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, 2788-2795. <https://doi.org/10.1145/3027063.3053216>.
- Núñez, R., & Lakoff, G. (2005). The cognitive foundations of mathematics. *Handbook of Mathematical Cognition*, 109-124.
- National Research Council. (2010). *Report of a workshop on the scope and nature of computational thinking*. National Academies Press.
- Nikou, S. A., & Economides, A. A. (2014). Transition in student motivation during a scratch and an app inventor course. Paper presented at the *Global Engineering Education Conference (EDUCON), 2014 IEEE*, 1042-1045. <https://doi.org/10.1109/EDUCON.2014.6826234>.
- Nouri, J., Zhang, L., Mannila, L., & Norén, E. (2020). Development of computational thinking, digital competence and 21st century skills when learning programming in K-9. *Education Inquiry*, 11(1), 1-17. <https://doi.org/10.1080/20004508.2019.1627844>.
- OECD (2019), *PISA 2018 Results (Volume I-III)*. https://www.oecd.org/pisa/publications/PISA2018_CN_ITA_IT.pdf.
- Olimpo, G. (2017). Dal mestiere dell'informatico al pensiero computazionale. *Italian Journal of Educational Technology*, 25(2), 15-26.
- Papavlasopoulou, S., Giannakos, M. N., & Jaccheri, L. (2017). Empirical studies on the maker movement, a promising approach to learning: A literature review. *Entertainment Computing*, 18, 57-78. <https://doi.org/10.1016/j.entcom.2016.09.002>

- Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. Basic Books, Inc.
- Papert, S. (1986). *Constructionism: A new opportunity for elementary science education*. Massachusetts Institute of Technology, Media Laboratory, Epistemology and Learning Group.
- Papert, S. (1987). Information technology and education: Computer criticism vs. technocentric thinking. *Educational Researcher*, 16(1), 22-30.
- Papert, S. (1993). *The children's machine: Rethinking school in the age of the computer*.
- Papert, S. (1999). Logo philosophy and implementation. *Logo Computer Systems Inc*.
- Papert, S., & Harel, I. (1991). Situating constructionism. *Constructionism*, 36(2), 1-11.
- Parlamento y Consejo de la Unión Europea (2006). *Recomendación del Consejo de 19 de diciembre de 2006 sobre las competencias clave para el aprendizaje permanente*. <https://eur-lex.europa.eu/legal-content/ES/TXT/HTML/?uri=CELEX:32006H0962&from=LV>.
- Pekrun, R. (2006). The control-value theory of achievement emotions: Assumptions, corollaries, and implications for educational research and practice. *Educational Psychology Review*, 18(4), 315-341. <https://doi.org/10.1007/s10648-006-9029-9>.
- Pekrun, R., Elliot, A. J., & Maier, M. A. (2009). Achievement goals and achievement emotions: Testing a model of their joint relations with academic performance. *Journal of Educational Psychology*, 101(1), 115. <https://doi.org/10.1037/a0013383>.
- Pekrun, R., Goetz, T., Frenzel, A. C., Barchfeld, P., & Perry, R. P. (2011). Measuring emotions in students' learning and performance: The achievement emotions questionnaire (AEQ). *Contemporary Educational Psychology*, 36(1), 36-48. <https://doi.org/10.1016/j.cedpsych.2010.10.002>.
- Pekrun, R., Goetz, T., Titz, W., & Perry, R. P. (2002). Academic emotions in students' self-regulated learning and achievement: A program of qualitative and quantitative research. *Educational Psychologist*, 37(2), 91-105. https://doi.org/10.1207/S15326985EP3702_4.

- Peppler, K., & Kafai, Y. (2007). *What videogame making can teach us about literacy and learning: Alternative pathways into participatory culture.*
- Pérez, Z. P. (2011). Los diseños de método mixto en la investigación en educación: Una experiencia concreta. *Revista Electrónica Educare*, 15(1), 15-29.
- Peris-Ortiz, M., Llera, J. J. A., & Rueda-Armengot, C. (2017). Entrepreneurship and innovation in a revolutionary educational model: École, 42. *Social entrepreneurship in non-profit and profit sectors* (pp. 85-97). Springer. https://doi.org/10.1007/978-3-319-50850-4_7.
- Piaget, J., Inhelder, B., Fraise, P., & Piaget, J. (1969). Intellectual operations and their development. *P. Fraise & J. Piaget*, 144-205.
- PNSD (2015). *Piano Nazionale Scuola Digitale*. https://www.istruzione.it/scuola_digitale/index.shtml.
- Poliandri, D., Muzzioli, P., Quadrelli, I., & Romiti, S. (2014). La scheda di osservazione in classe: Uno strumento per esplorare le opportunità di apprendimento. *Italian Journal of Educational Research*, 173-187.
- Popat, S., & Starkey, L. (2019). Learning to code or coding to learn? A systematic review. *Computers & Education*, 128, 365-376. <https://doi.org/10.1016/j.compedu.2018.10.005>.
- Popović, J., Korolija, N., Marković, Ž, & Bojić, D. (2017). Developing algorithmic skills of pupils in serbian schools using code. org materials. En *Telecommunication Forum (TELFOR)*, 2017 25th, 1-4. <https://doi.org/10.1109/TELFOR.2017.8249483>.
- Prensky, M. (2011). The reformers are leaving our schools in the 20th century. *On the Horizon*, 1-18.
- Psycharis, S., & Kallia, M. (2017). The effects of computer programming on high school students' reasoning skills and mathematical self-efficacy and problem solving. *Instructional science*, 45(5), 583-602. <https://doi.org/10.1007/s11251-017-9421-5>.
- Pugnali, A., Sullivan, A., & Bers, M. U. (2017). The impact of user interface on young children's computational thinking. *Journal of Information Technology Education: Innovations in Practice*, 16, 171-193.

- Raccanello, D., Brondino, M., Moè, A., Stupnisky, R., & Lichtenfeld, S. (2019). Enjoyment, boredom, anxiety in elementary schools in two domains: Relations with achievement. *The Journal of Experimental Education*, 87(3), 449-469. <https://doi.org/10.1080/00220973.2018.1448747>.
- Rees, A., García-Peñalvo, F. J., Jormanainen, I., Tuul, M., & Reimann, D. (2016). An overview of the most relevant literature on coding and computational thinking with emphasis on the relevant issues for teachers.
- Repenning, A., Webb, D., & Ioannidou, A. (2010). Scalable game design and the development of a checklist for getting computational thinking into public schools. En *Proceedings of the 41st ACM Technical Symposium on Computer Science Education*, 265-269. <https://doi.org/10.1145/1734263.1734357>
- Resnick, M. (2012). Let's teach kids to code [Video]. Ted. https://www.ted.com/talks/mitch_resnick_let_s_teach_kids_to_code#t-17358.
- Resnick, M. (2013). Learn to code, code to learn. *EdSurge*, May, 54.
- Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., & Silverman, B. (2009). Scratch: Programming for all. *Communications of the ACM*, 52(11), 60-67.
- Rich, P. J., Mason, S. L., & O'Leary, J. (2021). Measuring the effect of continuous professional development on elementary teachers' self-efficacy to teach coding and computational thinking. *Computers & Education*, 168, 104196. <https://doi.org/https://doi.org/10.1016/j.compedu.2021.104196>.
- Rinderknecht, M. (2013). *Tutorial for programming the LEGO® MINDSTORMS™ NXT*.
- Roig-Vila, R., & Moreno-Isac, V. (2020). El pensamiento computacional en educación. análisis bibliométrico y temático. *Revista De Educación a Distancia*, 20(63). <https://doi.org/10.6018/red.402621>
<https://doi.org/10.6018/red.402621>

- Román-González, M., Pérez-González, J., & Jiménez-Fernández, C. (2017). Which cognitive abilities underlie computational thinking? criterion validity of the computational thinking test. *Computers in Human Behavior*, 72, 678-691. <https://doi.org/10.1016/j.chb.2016.08.047>
- Sáez-López, J., Román-González, M., & Vázquez-Cano, E. (2016). Visual programming languages integrated across the curriculum in elementary school: A two year case study using “Scratch” in five schools. *Computers & Education*, 97, 129-141. <https://doi.org/10.1016/j.compedu.2016.03.003>.
- Samuels, P., & Poppa, S. (2017). *Developing extended real and virtual robotics enhancement classes with years 10–13*. Springer.
- Savard, A., & Highfield, K. (2015). Teachers' talk about robotics: Where is the mathematics?. *Mathematics Education Research Group of Australasia*.
- Scaffidi, C., & Chambers, C. (2012). Skill progression demonstrated by users in the scratch animation environment. *International Journal of Human-Computer Interaction*, 28(6), 383-398. <https://doi.org/10.1080/10447318.2011.595621>.
- Scratch (2019, En 2). *Introducing Scratch 3.0: Expanding the Creative Possibilities of Coding*. <https://medium.com/scratchteam-blog/introducing-scratch-3-0-expanding-the-creative-possibilities-of-coding-ad4cd9eda9ef>.
- Sengupta, P., Kinnebrew, J. S., Basu, S., Biswas, G., & Clark, D. (2013). Integrating computational thinking with K-12 science education using agent-based computation: A theoretical framework. *Education and Information Technologies*, 18(2), 351-380. <https://doi.org/10.1007/s10639-012-9240-x>.
- Sengupta, P., Dickes, A., & Farris, A. (2018). Toward a phenomenology of computational thinking in STEM education. *arXiv Preprint arXiv:1801.09258*.
- Sentance, S., & Csizmadia, A. (2017). Computing in the curriculum: Challenges and strategies from a teacher’s perspective. *Education and Information Technologies*, 22(2), 469-495. <https://doi.org/10.1007/s10639-016-9482-0>.

- Seo, Y., & Kim, J. (2016). Analyzing the effects of coding education through pair programming for the computational thinking and creativity of elementary school students. *Indian Journal of Science and Technology*, 9(46).
- Shute, V. J., Sun, C., & Asbell-Clarke, J. (2017). Demystifying computational thinking. *Educational Research Review*, 22, 142-158. <https://doi.org/10.1016/j.edurev.2017.09.003>.
- Singh, K. (2007). *Quantitative social research methods*. Sage.
- Stilgoe, J. (2018). Machine learning, social learning and the governance of self-driving cars. *Social Studies of Science*, 48(1), 25-56. <https://doi.org/10.1177/0306312717741687>.
- Tan, C. W., Yu, P., & Lin, L. (2019). Teaching computational thinking using mathematics gamification in computer science game tournaments. *Computational thinking education* (pp. 167-181). Springer.
- Tang, X., Yin, Y., Lin, Q., Hadad, R., & Zhai, X. (2020). Assessing computational thinking: A systematic review of empirical studies. *Computers & Education*, 148, 103798. <https://doi.org/10.1016/j.compedu.2019.103798>.
- Taranto, E., Arzarello, F., & Robutti, O. (2018). MOOC: Repository di strategie e metodologie didattiche in matematica. *Annali Online Della Didattica E Della Formazione Docente*, 9(14), 257-279.
- Theodoropoulos, A., Antoniou, A., & Lepouras, G. (2017). How do different cognitive styles affect learning programming? insights from a game-based approach in greek schools. *ACM Transactions on Computing Education (TOCE)*, 17(1), 3. <https://doi.org/10.1145/2157136.2157191>.
- Touretzky, D. S. (2014). Teaching kodu with physical manipulatives. *ACM Inroads*, 5(4), 44-51.
- Tuomi, P., Multisilta, J., Saarikoski, P., & Suominen, J. (2017). Coding skills as a success factor for a society. *Education and Information Technologies*, 1-16. <https://doi.org/10.1007/s10639-017-9611-4>.
- Van Kleef Conley (2020). *A new certificate to help people grow careers in IT.*
<https://www.blog.google/outreach-initiatives/grow-with-google/new-certificate-help-people-grow-careers/>.

- Vanderstoep, S.W., Johnston, D, D., 2009. *Research Methods for Everyday Life: Blending Qualitative and Quantitative Approaches*,
- Vygotsky, L. (1978). Interaction between learning and development. *Readings on the Development of Children*, 23(3), 34-41.
- Webb, D. C., Repenning, A., & Koh, K. H. (2012, February). Toward an emergent theory of broadening participation in computer science education. In *Proceedings of the 43rd ACM technical symposium on Computer Science Education* (pp. 173-178). <https://doi.org/10.1145/2157136.2157191>.
- Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2016). Defining computational thinking for mathematics and science classrooms. *Journal of Science Education and Technology*, 25(1), 127-147. <https://doi.org/10.1007/s10956-015-9581-5>.
- Wilensky, U., Brady, C. E., & Horn, M. S. (2014). Fostering computational literacy in science classrooms. *Communications of the ACM*, 57(8), 24-28.
- Wilson, A., & Moffat, D. C. (2010). Evaluating scratch to introduce younger schoolchildren to programming. *Proceedings of the 22nd Annual Psychology of Programming Interest Group. Universidad Carlos III De Madrid, Leganás, Spain*.
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33-35.
- Wing, J. M. (2008). Computational thinking and thinking about computing. *Philosophical Transactions: Mathematical, Physical and Engineering Sciences*, 366(1881), 3717-3725.
- Wing, J. M. (2014). Computational thinking benefits society. *40th Anniversary Blog of Social Issues in Computing*, 2014, 26.
- Wong, K., Ching, C. C., Mark, K. P., Tang, J. K., Lei, C. U., Cheung, H. Y., & Chui, H. L. (2015). Impact of computational thinking through coding in K-12 education: A pilot study in hong kong. *General Studies*, 85(88.01), 2.08.

- Yadav, A., Good, J., Voogt, J., & Fisser, P. (2017). Computational thinking as an emerging competence domain. *Competence-based vocational and professional education* (pp. 1051-1067). Springer.
- Yadav, A., Hong, H., & Stephenson, C. (2016). Computational thinking for all: Pedagogical approaches to embedding 21st century problem solving in K-12 classrooms. *TechTrends*, 60(6), 565-568. <https://doi.org/10.1007/s11528-016-0087-7>.
- Yadav, A., Krist, C., Good, J., & Caeli, E. N. (2018). Computational thinking in elementary classrooms: Measuring teacher understanding of computational ideas for teaching science. *Computer Science Education*, 28(4), 371-400.
- Yiannoutsou, N., Nikitopoulou, S., Kynigos, C., Gueorguiev, I., & Fernandez, J. A. (2017). Activity plan template: A mediating tool for supporting learning design with robotics. *Robotics in education* (pp. 3-13). Springer. https://doi.org/10.1007/978-3-319-42975-5_1.
- Yin, R. K. (2015). *Qualitative research from start to finish*. Guilford Publications.
- Zapata-Ros, M. (2015). Pensamiento computacional: Una nueva alfabetización digital. *Revista De Educación a Distancia*, (46).
- Zavala, L. A., Gallardo, S. C. H., & García-Ruíz, M. Á. (2013). Designing interactive activities within scratch 2.0 for improving abilities to identify numerical sequences. En *Proceedings of the 12th International Conference on Interaction Design and Children*, 423-426. <https://doi.org/10.1145/2485760.2485831>.
- Zhang, L., & Nouri, J. (2019). A systematic review of learning computational thinking through Scratch in K-9. *Computers & Education*, 141, 103607. <https://doi.org/10.1016/j.compedu.2019.103607>.
- Zhong, B., & Li, T. (2020). *Can pair learning improve students' troubleshooting performance in robotics education?*. SAGE Publications Sage CA: Los Angeles, CA. <https://doi.org/10.1177/0735633119829191>.