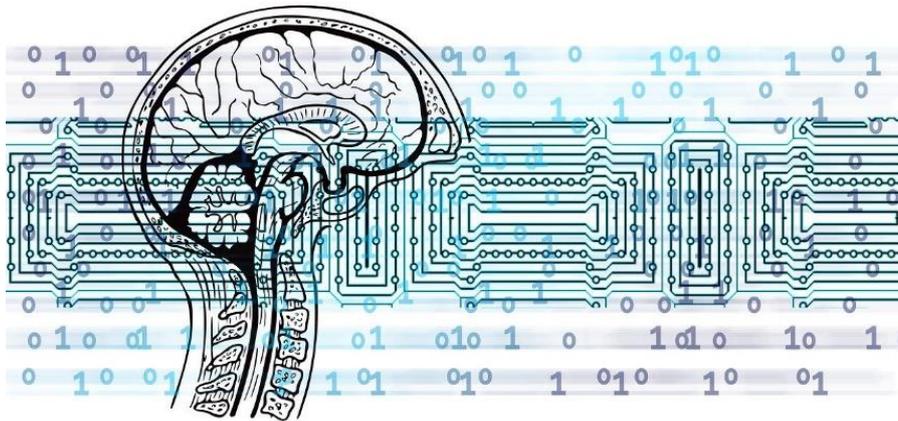


MÁSTER UNIVERSITARIO EN TECNOLOGÍA, APRENDIZAJE Y EDUCACIÓN  
**TRABAJO FIN DE MASTER**

**LA PROGRAMACIÓN EN EL CONTEXTO EDUCATIVO:  
UN ANÁLISIS DEL PENSAMIENTO COMPUTACIONAL**



Fuente: Imagen de Gerd Altmann en Pixabay

**Estudiante:** Gorrochategui Nanclares, Iñigo

---

**Director/Directora:** Tejada Garitano, Eneko

---

**Curso:** 2023-2024

**Fecha:** Leioa, 19 de junio de 2024



## ÍNDICE

|   |    |
|---|----|
| Introducción  | 4  |
| 1. Marco teórico  | 5  |
| 1.1. La programación en el sistema educativo: contexto e importancia      | 5  |
| 1.2. Pensamiento computacional  | 7  |
| 1.2.1. Definición y características del pensamiento computacional         | 7  |
| 1.2.2. Importancia y beneficios en el proceso de enseñanza-aprendizaje    | 9  |
| 1.3. Incorporación de la programación en el currículo vasco (alumnado)    | 10 |
| 1.3.1. Estado de la programación en el currículo vasco                    | 11 |
| 1.3.2. Brecha de género en la programación                                | 12 |
| 1.4. La programación en la competencia digital de enseñanza (profesorado) | 14 |
| 1.5. Tipos de lenguaje de programación                                    | 15 |
| 1.5.1. Programación basada en bloques                                     | 16 |
| 1.5.1.1. Scratch  | 17 |
| 1.5.2. Programación basada en texto                                       | 18 |
| 1.5.2.1. Python   | 19 |
| 1.6. Enfoques y herramientas de evaluación del pensamiento computacional  | 20 |
| 1.6.1. Análisis de herramientas para evaluar el pensamiento computacional | 22 |
| 2. Objetivos y preguntas de la investigación                              | 23 |
| 3. Metodología  | 23 |
| 3.1. Muestra  | 24 |
| 3.2. Instrumentos   | 24 |
| 3.3. Resultados   | 26 |
| 4. Discusión  | 38 |
| 5. Conclusiones   | 40 |
| 6. Futuras líneas de investigación  | 42 |
| Bibliografía  | 43 |

## **La programación en el contexto educativo: un análisis de la evaluación del pensamiento computacional**

**Iñigo Gorrochategui Nanclares**

UPV/EHU

### **Resumen**

En la era digital actual, la programación emerge como una habilidad fundamental que trasciende los límites de la informática para convertirse en una alfabetización esencial. Este trabajo examina la intersección entre la programación, la alfabetización en código y el pensamiento computacional, destacando su relevancia en la sociedad contemporánea. Se destaca la necesidad de desarrollar competencias en programación como parte integral del currículo educativo, reconociendo el papel crucial que desempeña en la formación de ciudadanos capacitados en el mundo tecnológico. Sin embargo, se plantea la cuestión de cómo evaluar y medir adecuadamente el pensamiento computacional, lo que representa un desafío significativo en la integración efectiva de estos conceptos en los programas educativos. Este estudio analiza las dificultades asociadas con la evaluación del pensamiento computacional y sugiere enfoques para su incorporación efectiva en los currículos educativos, con el objetivo de preparar a las generaciones futuras para enfrentar los desafíos y aprovechar las oportunidades en la sociedad digital en constante evolución.

**Palabras clave: programación, código alfabetización, pensamiento computacional, currículo, evaluación.**

### **Abstract**

In the current digital era, programming emerges as a fundamental skill that transcends the boundaries of computer science to become an essential literacy. This work examines the intersection between programming, code literacy, and computational thinking, highlighting its relevance in contemporary society. The need to develop programming competencies as an integral part of the educational curriculum is emphasized, recognizing the crucial role it plays in shaping technologically proficient citizens. However, the question of how to adequately assess and measure computational thinking is raised, posing a significant challenge in the effective integration of these concepts into educational programs. This study analyzes the difficulties associated with the assessment of computational thinking and suggests approaches for its effective incorporation into educational curriculum, aiming to prepare future generations to tackle challenges and seize opportunities in the constantly evolving digital society.

**Key words: programming, code literacy, computational thinking, evaluation, curriculum**

## Introducción

En la sociedad digital actual, la programación emerge como una habilidad fundamental que trasciende los límites tradicionales de la informática, convirtiéndose en una alfabetización esencial para la ciudadanía. A medida que la tecnología se integra cada vez más en todos los aspectos de la vida cotidiana, desde la comunicación y el entretenimiento hasta la educación y el trabajo, la capacidad de comprender y manipular código se vuelve primordial.

Este trabajo se centra en la intersección entre la programación, la alfabetización en código y el pensamiento computacional, destacando su creciente relevancia en la sociedad contemporánea. La programación no solo es una herramienta para desarrolladores de software, sino una competencia clave que empodera a los individuos para participar plenamente en un mundo dominado por la tecnología. Por lo tanto, se reconoce la necesidad de integrar estas competencias en los currículos educativos, preparando a los estudiantes no solo para carreras específicas en tecnología, sino para una ciudadanía versada y capacitada en la era digital.

Sin embargo, la integración de la programación y el pensamiento computacional en la educación enfrenta un desafío significativo: la evaluación adecuada de estas competencias. A diferencia de las materias tradicionales, donde las evaluaciones estandarizadas están bien establecidas, el pensamiento computacional requiere enfoques innovadores para su medición y evaluación. Este estudio se propone explorar estas dificultades y analizar métodos efectivos para la incorporación del pensamiento computacional en los currículos educativos. Para ello, este estudio presenta un análisis sociológico que pretende aunar dos instrumentos de evaluación del pensamiento computacional, ambos validados por estudios previos. Utilizando estos instrumentos, se pretende extraer una imagen completa del nivel de pensamiento computacional en el alumnado. Este enfoque nos permitirá recoger y analizar datos significativos sobre la efectividad de las metodologías actuales y las percepciones de los estudiantes respecto a su aprendizaje y aplicación del pensamiento computacional.

El objetivo es proporcionar un marco que no solo facilite la enseñanza de estas habilidades esenciales, sino que también prepare a las futuras generaciones para enfrentar los desafíos y aprovechar las oportunidades en un ecosistema digital en constante cambio y evolución.

## 1. Marco teórico

### 1.1. La programación en el sistema educativo: contexto e importancia

Es evidente que cada día vivimos en un ecosistema más digital y robótico, la tecnología abarca mucho protagonismo en nuestro ámbito de la vida, es por ello por lo que no podemos obviarla. Esta convivencia con los dispositivos es cada vez tan estrecha que resulta primordial en pleno siglo XXI tener que desarrollar la capacidad de poder comunicarnos con las computadoras. Para ello, dominar los lenguajes de programación se presenta como una habilidad crucial, un nuevo tipo de alfabetismo que nos capacita para participar plenamente y de manera efectiva en el entorno digital que nos rodea. Como Rushkoff (2010) bien explica, tenemos dos opciones: “programar o ser programado”.

Bers (2018) explica que la programación es vista como un nuevo tipo de alfabetización, la alfabetización del siglo XXI. En siglos pasados era necesario que los ciudadanos aprendieran a escribir, no solo para leer, sino para ser productores de información y conocimiento en vez de solo consumidores, actualmente en el siglo XXI, en la sociedad digital de información, es necesario que los ciudadanos/as aprendan a “codificar” para ser productores digitales de información, no solo consumidores digitales de la misma.

Román (2014) acuñó el término "codigoalfabetización" (*code-literacy* en inglés) para describir el proceso de enseñanza y aprendizaje de la lectoescritura utilizando lenguajes de programación. De esta manera, se considera que una persona está codigoalfabetizada cuando puede leer y escribir en el lenguaje utilizado por computadoras y otras máquinas.

El sistema educativo no debe obviar este avance digital y es por eso por lo que Ramos y Carbonell (2021) consideran la programación como una de las dimensiones en las que debe enfocarse la escuela actual, por su importancia en las profesiones del futuro y, por ende, en los trabajos que emergen del desarrollo tecnológico de la sociedad del conocimiento.

En los últimos 10 años, los dispositivos móviles, la *web 2.0*, la industria del videojuego y el nacimiento del movimiento DIY (*Do It Yourself*) han puesto en relieve las necesidades y carencias en la formación sobre pensamiento computacional y programación. Como respuesta a esta demanda tanto gobiernos, empresas y organizaciones o instituciones educativas, han desarrollado proyectos y adoptado

decisiones relacionadas con el fomento de la «codificación» en la educación (Valverde-Berrocoso et al., 2015).

De todos modos, la idea de introducir la programación en el aula no es algo innovador desde que irrumpiera en 1960. El investigador Seymour Papert fue el primero en identificar el potencial de la enseñanza de lenguajes de programación para niños/as como incubadora de ideas poderosas. Declaraba que la programación era una herramienta para involucrar a los niños/as en nuevas formas de pensar, pero mucho más críticas que eso, poniendo al estudiante en el papel de pensar en el proceso del pensamiento (Tadeu y Brigas, 2022).

La codificación informática, una actividad que implica la creación, modificación e implementación de código informático y expone a los estudiantes al pensamiento computacional, es una parte integral de la educación actual en ciencia, tecnología, ingeniería y matemáticas (STEM). Con el avance de la tecnología, la codificación se está convirtiendo en un proceso necesario y una habilidad muy necesaria para resolver problemas científicos complejos de manera eficiente y reproducible, elevando en última instancia las carreras de quienes dominan la habilidad. Muchos países de todo el mundo están lanzando iniciativas de codificación e integrando el pensamiento computacional en los planes de estudio de la educación superior, secundaria, primaria y preescolar (Scherer et al., 2021).

Por ello, la creciente importancia de la programación en la vida en general, ha aumentado la demanda de educación en programación (Graafsmaa et al., 2023). Esto ha cambiado el contexto de aprendizaje actual de tres maneras. Primero, el número de estudiantes que estudian cursos introductorios de programación ha aumentado, lo que ha resultado en más conferencias a gran escala y una necesidad de más aprendizaje independiente. Segundo, ahora están aprendiendo a programar estudiantes con una variedad más amplia de antecedentes y con diferentes fortalezas y debilidades. Finalmente, debido a esta diversificación, los métodos de instrucción se han simplificado y los lenguajes de programación se han desarrollado para asemejarse más a los lenguajes naturales y facilitar así el aprendizaje para los más principiantes (Fedorenko et al., 2019).

## 1.2. Pensamiento computacional

La programación está estrechamente relacionada con el pensamiento computacional, pero no son lo mismo (Bers, 2018). Si la codigoalfabetización se trata como una nueva forma de lectoescritura, el pensamiento computacional es el proceso mental que sustenta esta práctica. Es decir, la codigoalfabetización se refiere a la capacidad de leer y escribir en lenguajes de programación, y el pensamiento computacional al proceso cognitivo subyacente utilizado para resolver problemas de manera eficiente (Román-González, 2014).

En cualquier caso, la relación entre el pensamiento computacional (CT, por sus siglas en inglés) y la programación es confusa. Se asume que la programación informática permite que el CT cobre vida, y es la principal manera de demostrar habilidades de CT. Sin embargo, el CT podría proyectarse en una amplia gama de tareas que no implican programación (Wing, 2008). En otras palabras, es necesario activar habilidades de CT para programar adecuadamente, pero estas habilidades podrían usarse en otros contextos que estén desconectados de la programación informática. Por lo tanto, el CT es un término más amplio que la programación informática (Román-González et al., 2019).

### 1.2.1. Definición y características del pensamiento computacional

Wing (2006) describió el pensamiento computacional como un proceso que implica resolver problemas, diseñar sistemas y comprender el comportamiento humano utilizando conceptos básicos de la informática. Desde entonces, con la evolución y desarrollo de las herramientas tecnológicas, el interés por el pensamiento computacional ha ido en aumento. Wing (2006) sostiene que estas habilidades son beneficiosas para todos y que, hacia mediados del siglo XXI, se convertirán en una destreza esencial, comparable a la lectura o la escritura en la actualidad.

La autora explica que no usa las palabras “problema” y “solución” para referirse solamente a problemas algorítmicos, sino para hablar de problemas reales que pueden resolverse completa o parcialmente con ayuda del ordenador, o incluso de problemas cotidianos (Velázquez Iturbide y Martín, 2021).

Por otra parte, es una competencia básica que toda persona debería conocer para desenvolverse en la sociedad digital, pero no es una habilidad «rutinaria» o «mecánica», ya que es una forma de resolver problemas de manera inteligente e imaginativa (cualidades humanas que no poseen los ordenadores) (Wing, 2006).

Basoagain et al. (2015) añaden que el pensamiento computacional se basa en una metodología implementada por conceptos básicos de las ciencias de la computación para resolver problemas cotidianos, diseñar sistemas domésticos y realizar tareas rutinarias. Esta nueva forma de abordar los problemas nos permite resolver con eficacia y éxito problemas que de otra forma no son tratables por una persona.

En cierto sentido, Román-González et al. (2018) destacan la creación del pensamiento computacional (CT) como un término general extremadamente útil, y su repentino éxito puede explicarse. En primer lugar, el término CT ha ayudado a situar la educación en informática más allá de la programación informática. En segundo lugar, ha ayudado a reducir las barreras de entrada a la programación informática, en paralelo con la aparición y aumento de lenguajes visuales de bloques; de la misma manera, el CT ha proporcionado el marco para centrarse no en la sintaxis de la programación informática, sino en los procesos mentales subyacentes a ella. Como resultado, el CT se percibe como un término amigable y no amenazante que ha contribuido a acercar la informática a los profesores y a fomentar el movimiento de "Informática para todos". En tercer lugar, el CT es un término tan flexible que puede usarse más como un enfoque que como un concepto; entonces, el CT ha mejorado la metáfora de "programar para aprender" en lugar de "aprender a programar".

En otras palabras, el CT ha hecho posible imaginar un enfoque computacional para cualquier materia del currículo. El término CT ha reunido no solo habilidades cognitivas, como la descomposición, el reconocimiento de patrones, la abstracción y el diseño algorítmico, sino también variables no cognitivas (Román-González et al. 2018) y habilidades blandas relacionadas como la persistencia, la autoconfianza, la tolerancia a la ambigüedad, la creatividad y el trabajo en equipo, entre otras.

## 1.2.2. Importancia y beneficios en el proceso de enseñanza-aprendizaje

Tadeu y Brigas (2022) nos indican, que, en la actualidad, las denominadas competencias interpersonales están estrechamente relacionadas con la capacidad de pensamiento computacional. Lo que ha llevado a que el pensamiento computacional se reconozca como una habilidad del siglo XXI necesaria para prosperar en la sociedad actual. El futuro exige personas capaces de trabajar y pensar sistemáticamente. Además, deben saber cómo analizar situaciones y trabajar en diferentes soluciones. El sistema educativo tiene que asumir ese reto dentro de las aulas para lograr ese objetivo. Por eso es crucial definir y comprobar las investigaciones sobre el pensamiento computacional.

Quienes abogan por la integración del pensamiento computacional en los sistemas educativos actuales creen que no es simplemente un proceso para resolver problemas, sino una habilidad intelectual fundamental, equiparable a la lectura, la escritura, el habla y las matemáticas básicas. Estas habilidades son esenciales para comprender y explicar problemas o situaciones complejas, y el pensamiento computacional cumple exactamente esa función. Podríamos decir entonces que el pensamiento computacional es tan esencial como otras habilidades cognitivas básicas que se consideran indispensables en nuestra sociedad moderna. En última instancia, el pensamiento computacional es un nuevo lenguaje que podemos utilizar junto con el lenguaje hablado, escrito y matemático para describir y comprender el universo y sus procesos (Miranda et al., 2017).

González (2019) recalca que las habilidades propias del pensamiento computacional son muchas y Gamito et al. (2022) en la **tabla 1** nos extraen las siguientes habilidades y actitudes que desarrollamos con el PC, obtenidas de las organizaciones la Sociedad Internacional de Tecnología en Educación (ISTE) y la Asociación de Maestros de Ciencias de la Computación (CSTA):

Tabla 1. Habilidades y actitudes del PC.

| Habilidades   | Actitudes   |
|---|---|
| Formulación de problemas que permiten usar un ordenador y otras herramientas para ayudar a resolverlos. | Confianza en el manejo de la complejidad.           |
| Organización y análisis lógico de datos.  | Persistencia en el trabajo con problemas difíciles. |

|   |   |
|---|---|
| Representación de datos a través de abstracciones tales como modelos y simulaciones.  | Tolerancia a la ambigüedad.   |
| Automatización de soluciones a través del pensamiento algorítmico (una serie de pasos ordenados).   | Capacidad de lidiar con problemas abiertos.   |
| Identificación, análisis e implementación de las posibles soluciones con el objetivo de lograr la combinación más eficiente y efectiva de pasos y recursos. | Capacidad de comunicarse y trabajar con otros para lograr un objetivo o solución común. |
| Generalización y transferencia de este proceso de resolución del problema a una amplia variedad de problemas.   |   |

Por este motivo, visto el potencial, son muchas las iniciativas que buscan la integración del PC en los procesos de enseñanza-aprendizaje.

### 1.3. Incorporación de la programación en el currículo vasco (alumnado)

Antes de comenzar a hablar del estado de la programación en el currículo vasco, es necesario mencionar cuál es la situación global de la misma y es que existen iniciativas en países que han incorporado la enseñanza de la programación a nivel formal en sus currículums escolares, tales como Estonia, Suiza, Finlandia, EEUU, Israel, Singapur o Reino Unido entre otros (González, 2019).

Reino Unido es uno de los pioneros en Europa implantando el pensamiento computacional y la programación de manera obligatoria en el currículo de primaria y secundaria desde el año 2004 y Finlandia, uno de los primeros países europeos en implantar de manera obligatoria en sus currículos el pensamiento algorítmico y la programación (año 2016).

En el entorno digital actual, es crucial implementar estrategias que modernicen los procesos de enseñanza-aprendizaje, lo que incluye iniciativas para que todas las personas adquieran competencias digitales que les permitan desenvolverse de manera eficaz en una sociedad altamente tecnológica. En este contexto, está surgiendo una tendencia que defiende enseñar habilidades de programación desde una edad temprana en los centros

escolares. El objetivo es capacitar a las personas para que sean activas y creativas en el uso de la tecnología, al dominar nuevas habilidades cognitivas y prácticas, como la códigoalfabetización. (García y Caballero, 2019).

En este sentido, la competencia digital se ha convertido en competencia clave para desenvolverse en el siglo XXI (2006/962/CE) (Gamito et al., 2022). DigComp 2.1, el Marco Europeo de Competencias Digitales para la Ciudadanía, recoge 21 competencias, divididas en cinco áreas: alfabetización informacional, comunicación y colaboración, creación de contenidos digitales, seguridad y resolución de problemas.

Carretero et al. (2017) explican que la programación se integra perfectamente en este contexto, ya que, dentro de la competencia del área de creación de contenidos digitales, DigComp abarca la planificación y desarrollo de secuencias de instrucciones para resolver problemas utilizando sistemas informáticos. Por ello, se ha fortalecido el énfasis en enseñar programación en todas las etapas educativas.

### **1.3.1. Estado de la programación en el currículo vasco**

Si analizamos el decreto Decreto 77/2023, de 30 de mayo, de establecimiento del currículo de Educación Básica e implantación en la Comunidad Autónoma de Euskadi podemos destacar que el pensamiento computacional con la ley educativa LOMLOE ha adquirido un protagonismo significativo.

#### **Educación primaria**

Una vez completada la educación primaria, el alumnado se habrá iniciado «en el desarrollo de soluciones digitales sencillas y sostenibles para resolver problemas concretos o retos propuestos de manera creativa, solicitando ayuda en caso necesario».

Para ello se han incluido competencias específicas y saberes básicos relacionados con el pensamiento computacional tanto en el área Conocimiento del medio natural, social y cultural como en el área de Matemáticas. A lo largo de los tres ciclos de primaria, los estudiantes aprenderán a abordar problemas a través de proyectos interdisciplinarios, utilizando el pensamiento de diseño y el pensamiento computacional. Esto les permitirá colaborar en la creación conjunta de productos creativos e innovadores que satisfagan necesidades específicas. Además, aprenderán a utilizar el pensamiento computacional para organizar datos, descomponer problemas en partes más pequeñas, identificar

patrones, generalizar e interpretar información, así como modificar y crear algoritmos de manera guiada. Todo esto les permitirá modelar y automatizar situaciones de la vida cotidiana de manera efectiva.

### **Educación secundaria obligatoria**

Una vez completada la educación secundaria obligatoria el alumnado sabrá cómo «desarrollar aplicaciones informáticas sencillas y soluciones tecnológicas creativas y sostenibles para resolver problemas concretos o responder a retos propuestos, mostrando interés y curiosidad por la evolución de las tecnologías digitales y por su desarrollo sostenible y uso ético.»

Para obtener dichos conocimientos, se ha creado la asignatura Tecnología y Digitalización, que se oferta 2º y 3º de ESO. Además, en 4º de ESO se oferta por un lado la asignatura de Tecnología y por otro lado Digitalización, ambas con bloques de saberes sobre pensamiento computacional. Y se incluyen competencias específicas y criterios de evaluación específicos sobre pensamiento computacional en otras materias, como Biología y Geología (3º y 4º ESO), Matemáticas (de 1º a 4º), y a la optativa Iniciación a la Tecnología y a la Digitalización.

#### **1.3.2. Brecha de género en la programación**

En disciplinas como la tecnología, ciencias y matemáticas la brecha de género no es algo novedoso. La UNESCO y la OCDE llevan años trabajando y luchando este problema para poder obtener esa parte fundamental de la Agenda Mundial de Educación 2030, alcanzar el cuarto Objetivo de Desarrollo Sostenible: "garantizar una educación inclusiva, equitativa y de calidad, y promover oportunidades de aprendizaje durante toda la vida para todos".

Si profundizamos en el rol de la mujer en la ciencia, las diferencias se vuelven evidentes, ya que las profesiones en áreas como informática, telecomunicaciones, arquitectura y tecnología siguen estando dominadas por hombres. Esto ocurre a pesar de que las mujeres han mostrado un rendimiento ascendente en estas materias tanto en la educación obligatoria como en los estudios superiores. Esta situación ilustra problemas significativos para las mujeres, como el techo de cristal o el suelo pegajoso, metáforas para describir cómo pocas mujeres logran avanzar en sus carreras científicas, a pesar de su gran esfuerzo. Como resultado, muy pocas mujeres alcanzan posiciones de

reconocimiento o poder, lo que perpetúa el modelo tradicional y mantiene una discriminación jerárquica donde se valora más a los hombres que a las mujeres (Espino y González, 2016).

Los factores que influyen en la brecha de género en STEM, según la UNESCO (2019), incluyen:

- **Factores individuales:** La autopercepción, los estereotipos y las identidades STEM. Los estudiantes adoptan estereotipos de género a una edad temprana, afectando la autoeficacia de las niñas y su interés en campos STEM.
- **Factores familiares y de pares:** Las expectativas y creencias de los padres, así como su nivel educativo, especialmente el de la madre, influyen en el rendimiento de los estudiantes en STEM.
- **Factores escolares:** La presencia de profesoras en STEM y el entorno de los pares afectan la confianza y el interés de las niñas en estas áreas.
- **Factores sociales:** La igualdad de género, normas culturales, políticas y los medios de comunicación influyen en la brecha de género.

Para poder reducir esta diferencia existente es fundamental estudiar cómo comienza la enseñanza de la programación para los estudiantes, identificar posibles brechas de género en esta etapa inicial y, si existen, tomar medidas adecuadas para abordarlas. Esto garantizaría una alfabetización en programación equitativa, promoviendo una mayor presencia femenina desde las primeras etapas de aprendizaje hasta el desarrollo profesional. Es especialmente importante centrarse en el colectivo femenino, desde edades tempranas hasta la adultez (Torres et. al, 2022).

Un colectivo femenino que demuestra en estudios nacionales e internacionales como el informe PISA 2014 (PISA, 2014) o la encuesta de Competencias de la Población Adulta de la OCDE de 2012 (OCDE, 2012), el ascendente rendimiento que experimentan en materias como matemáticas, física o tecnología.

También se ha demostrado que los chicos superan a las chicas en tareas relacionadas con la robótica, aunque esto está influenciado por la edad en la que comienzan a trabajar en este campo y por estereotipos culturales que afectan la representación femenina. Sin embargo, con una intervención educativa adecuada, tanto niños como niñas pueden obtener resultados muy similares (Sullivan y Bers, 2013).

Paucar-Curasma et. al (2023), en cambio, no extrajeron diferencias significativas entre estudiantes varones y mujeres en ninguna de las habilidades del PC. Destacan que

la causa de ello es la estrategia educativa que aplicaron generando el mismo entusiasmo en estudiantes mujeres y hombres en las actividades STEM que se llevaron a cabo.

Niousha et. al (2022) y Funke y Geldreich (2017) realizaron estudios donde encontraron algunos hallazgos diferenciales en cuanto al género en proyectos de Scratch. Ambos estudios recalcan que los proyectos de los chicos eran más complejos y requerían de una mayor destreza computacional frente a los proyectos de las chicas donde se apreciaba más creatividad y contenido multimedia.

#### **1.4. La programación en la competencia digital de enseñanza (profesorado)**

En el apartado anterior, hablábamos de las competencias que deben adquirir los estudiantes durante su proceso de formación y el profesorado, en este caso, tiene un papel fundamental. Castañeda *et al.* (2018) mencionan que uno de los principales desafíos en el desarrollo del pensamiento computacional y las competencias para la resolución de problemas en el ámbito educativo es la formación inicial del futuro profesorado.

Debemos de formar mejor a los docentes en el conjunto de conocimientos, habilidades y actitudes necesarios para promover el aprendizaje del alumno en un mundo crecientemente tecnológico. Los propios docentes no han sido formados en PC y sus experiencias en programación en sus respectivas formaciones iniciales ha sido escasa o nula (Bustillo Bayón, 2015); así pues, es preciso planificar la formación del profesorado para que no solo sea competente desde la perspectiva del PC, sino también como diseñador e implementador de estrategias que ayudan a adquirirlo.

Por otro lado, están emergiendo modelos de formación en didáctica del pensamiento computacional (PC) que, en líneas generales, buscan adaptar el modelo TPACK (Mishra y Koehler, 2006) al contexto específico que nos concierne. Estos modelos sugieren de manera concisa que cualquier estrategia de formación dirigida a los docentes debe asegurar: (1) que el docente pueda identificar con claridad qué proyectos son adecuados para integrar situaciones de aprendizaje del PC; (2) que sea capaz de identificar qué tecnologías son las más apropiadas para estos proyectos; y (3) que reconozca las oportunidades para transformar la combinación de proyectos y tecnologías en oportunidades de aprendizaje del PC. Aunque aún está en proceso de desarrollo y operativización, este enfoque ofrece un marco conceptual que destaca la necesidad de

formar a los maestros generales o profesores de informática en habilidades específicas que garanticen su competencia como docentes de PC (González et al., 2018).

En definitiva, la capacitación de los docentes en tecnologías de la información y la comunicación sigue siendo un desafío tanto en su formación inicial como en su desarrollo profesional. Para abordar este desafío, ha surgido en los últimos años el concepto de competencia digital docente (CDD) (Castañeda et al., 2018).

Castañeda et al. (2018) mencionan, por un lado, el Joint Research Centre (JRC) de la Comisión Europea, en el que elaboraron el Marco Europeo para la Competencia Digital del Profesorado (DigCompEdu). Un modelo que abarca aspectos relacionados con el compromiso profesional del profesorado y su entorno laboral, el uso de las tecnologías de la información y la comunicación (TIC) en el proceso de enseñanza-aprendizaje, así como el empoderamiento de los estudiantes y el desarrollo de la competencia digital del profesorado como ciudadanos. Por otro lado, estatal destacan, el Marco Común de Competencia Digital Docente del Instituto Nacional de Tecnologías Educativas y de Formación del Profesorado (INTEF), basándose en el modelo de competencia digital genérica o ciudadana DigComp de la Comisión Europea.

### **1.5. Tipos de lenguaje de programación**

La enseñanza sobre el pensamiento computacional y programación necesita herramientas que disminuyan las dificultades iniciales y hagan que las actividades sean accesibles, estimulantes y motivacionales para los estudiantes (Piedade et al., 2019). Seymour Papert estuvo a la vanguardia en este ámbito en los años 60 al desarrollar Logo (Papadakis et al., 2017), un lenguaje de programación diseñado para niños. Inicialmente basado en texto, tuvo un desarrollo hacia interfaces basadas en bloques en la década de los 90. Lenguajes más contemporáneos como Alice o Scratch han seguido este enfoque de bloques, permitiendo a los alumnos enfocarse en la creación y la experimentación sin preocuparse por la sintaxis u otros aspectos complejos y técnicos del lenguaje.

Piedade et al. (2019) destacan la importancia de los lenguajes basados en bloques y textos como herramientas esenciales que permiten iniciar a los alumnos en el pensamiento computacional y la programación.

### 1.5.1. Programación basada en bloques

Muchas de las propuestas para fomentar el desarrollo del pensamiento computacional hacen uso del lenguaje de programación de tipo visual. López (2014) explica que este lenguaje además de destacar por su sencillez proporciona retroalimentación visual en forma de objetos animados, permitiendo al alumnado la creación de medios interactivos, como animaciones y juegos.

Actualmente existen varias plataformas que permiten la codificación mediante procesos de ludificación y herramientas de programación visual diseñadas para usuarios noveles. Weintrop et al. (2015) destacan el lenguaje de Scratch como uno de los entornos gráficos de programación más completos, pero no es el único. A continuación, en la **tabla 2** detallamos los más significativos:

Tabla 2. Lenguajes de programación basados en bloques.

| Nombre    | Descripción   |
|-----------|---|
| Scratch   | Lenguaje visual de programación por bloques de diferentes colores. Probablemente el lenguaje de programación visual más apropiado.  |
| Alice     | Entorno de programación 3D con bloques gráficos y animaciones para crear historias y juegos.  |
| mBlock    | Entorno gráfico de programación que se basa en Scratch 2.0. Útil para programar robot con entorno Arduino.  |
| Snap!     | Una versión parecida a Scratch, pero más desarrollada.  |
| Tynker    | Entorno de programación gráfico de bloques parecido a Scratch. Sencilla e intuitiva. Permite la conexión con herramientas externas como Minecraft y la programación de drones.                    |
| Code.org  | Organización sin ánimo de lucro dedicada a acercar la programación y la computación a los estudiantes. Ofrece varios proyectos para iniciarse en el mundo de la programación.                     |
| Blockly   | Herramienta que convierte en tiempo real una aplicación por bloques en texto. Es aconsejable visitar Blockly Games, juegos que ayudarán a los alumnos a familiarizarse con el entorno de Blockly. |
| Game blox | Lenguaje de programación gráfico para desarrollar juegos en dispositivos móviles.   |

|                   |  |
|-------------------|--|
| App Inventor      | Permite crear una aplicación Android con un entorno sencillo similar a Scratch.  |
| Thunkable         | Basada en App Inventor, familiariza al alumnado en el desarrollo de apps para Android.   |
| Swift playgrounds | Lenguaje creado por Apple para dispositivos iOS que permite iniciarse en el mundo de la programación de manera sencilla e intuitiva. |
| Stencyl           | Similar a Scratch que permite crear videojuegos con gráficos 2D.   |
| Hopscotch         | Lenguaje de programación visual para móviles iOS.  |

### 1.5.1.1. Scratch

Logo fue el primer lenguaje de programación basado en un enfoque pedagógico que permitió el aprendizaje de matemáticas, cibernética y ciencias de una forma innovadora. Se denomina el precursor de Scratch, un entorno de programación con una gran potencialidad educativa. La novedad de Logo fue la introducción de un objeto gráfico (la tortuga) sobre el que se aplicaban las instrucciones que los alumnos escribían en un código de fácil comprensión. Este concepto lo introdujo Scratch creando un gato movable, capaz de emitir sonidos, dibujar y otra serie de acciones a través de bloques preconfigurados. En definitiva, Logo introdujo la programación orientada a objetos dentro de la educación, un método muy usado en la mayoría de los lenguajes profesionales de programación gracias a su sencillez y carácter intuitivo (Valverde-Berrocoso et al., 2015).

Scratch es un lenguaje visual de programación por bloques creado por el grupo Lifelong Kindergarten del Instituto Tecnológico de Massachusetts (MIT, s. f.) que permite la iniciación en el mundo de la programación con la creación de proyectos interactivos, juegos, animaciones y un largo etc. Es probablemente el lenguaje de programación visual más apropiado para desarrollar el pensamiento computacional a través de la programación en el ámbito de la educación primaria y secundaria (Brennan & Resnick, 2012). Y uno de los principales motivos es por su interfaz, un entorno que se presenta a modo de “piezas de puzle” (figura 1) que se deben arrastrar, soltar y unir de manera lógica para crear el conjunto de instrucciones o *script* encargado de controlar las acciones que deben ejecutar las escenas y/o los objetos (Gamito et al., 2022).

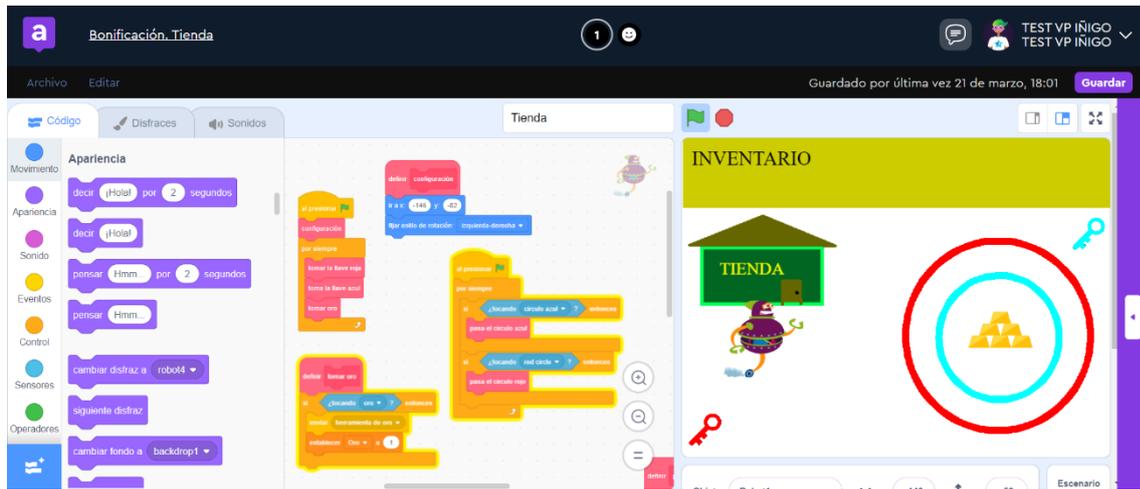


Figura 1 Interfaz de programación de Scratch. Fuente: elaboración propia en <https://learn.alg.academy/>

### 1.5.2. Programación basada en texto

La programación basada en texto implica escribir una secuencia de instrucciones en un lenguaje de programación específico, siguiendo las reglas establecidas por ese lenguaje. Estas instrucciones son analizadas e interpretadas por el procesador para llevar a cabo las tareas asignadas. A diferencia de la programación visual, donde se utilizan bloques para representar las instrucciones y órdenes, los lenguajes de programación basados en texto requieren un mayor conocimiento y capacidad de abstracción. Poder dominar un lenguaje de programación textual requiere comprender su “sintaxis” y “gramática” específicas, lo que puede resultar complejo. No obstante, desde la década de los 60 se han desarrollado diversos entornos de programación, y en la actualidad existen numerosos lenguajes y entornos visuales más intuitivos y accesibles para los usuarios (INTEF, s.f.). A continuación, en la **tabla 3** detallamos los más significativos:

Tabla 3. Lenguajes de programación basados en texto.

| Nombre | Descripción   |
|--------|---|
| Python | Lenguaje de texto de sintaxis sencillo y fácil de comprender. Versátil, flexible y con una alta legibilidad |
| JAVA   | Lenguaje para codificar aplicaciones web.   |
| C      | Lenguaje eficiente y útil generalmente para el desarrollo de sistemas operativos.                           |
| C++    | Basado en C y ofrece programación orientada a objetos.  |
| C#     | Basado en C y creado por MICROSOFT.   |

|                     |  |
|---------------------|--|
| PSEINT Pseudocódigo | Herramienta para familiarizar a los estudiantes con los lenguajes basados en texto.  |
| Processing          | Basado en JAVA, ideal para la producción de proyectos de diseño digital, dando resultados muy vistosos, que estimulan mucho a los estudiantes. |

### 1.5.2.1. Python

Python es un lenguaje de programación de un nivel de código abierto alto, orientado a objetos, simple y con una sintaxis fácil de entender (Figura 2). Además, la semántica de este lenguaje se basa en módulos que pueden ser importados y exportados para ser utilizados en otros proyectos. Es por eso por lo que Python se ha convertido en uno de los lenguajes más versátiles y óptimos para programar garantizando flexibilidad, legibilidad y compatibilidad en distintos proyectos educativos (INTEF, s.f.).

Existen multitud de experiencias y estudios que avalan las ventajas de Python como candidato bien posicionado (Monsálvez, 2017). Peña (2015) explica cuáles son las principales características por las que destacan a Python:

- Sintaxis simple.
- Alta legibilidad (sangrado obligatorio).
- Entorno amigable de desarrollo (intérprete interactivo).
- Abstracciones de más alto nivel (mayor nivel de expresividad).
- Librería estándar de gran potencial y gran cantidad de módulos de terceros (actualmente son más de 100.000).
- Multi-paradigma (imperativo, POO y funcional).
- Disponibilidad de recursos educativos abiertos.
- Software libre y comunidad entusiasta

```

1 #Los primeros 3 clientes obtienen un descuento
2 count = 0
3 while count < 3:
4     card = int(input("Ingrese su número de tarjeta:"))
5     print("Felicitaciones")
6     count = count + 1
7
8 print("No más descuentos.")

Ingreso su número de tarjeta:
>>> 4
Felicitaciones
Ingreso su número de tarjeta:
>>> 5
Felicitaciones
Ingreso su número de tarjeta:
>>> 6
Felicitaciones
No más descuentos.
    
```

Figura 2 Interfaz de programación de Python. Fuente: elaboración propia en <https://learn.alg.academy/>

## 1.6. Enfoques y herramientas de evaluación del pensamiento computacional

Aún no existe una idea clara sobre cómo incorporar el PC a los sistemas educativos en sus distintas etapas. Existen una enorme variedad y heterogeneidad de intervenciones educativas al respecto (Lye & Koh, 2014), pero a su vez, existe una gran carencia en cuanto a cómo medir y evaluar la competencia del pensamiento computacional, un aspecto relevante que debe ser abordado y estudiado. Sin una atención adecuada hacia la medición y evaluación del pensamiento computacional, será complejo integrarla exitosamente en cualquier plan de estudios curricular. (Grover & Pea, 2013).

Centrándonos en la educación K-12, podemos encontrar las siguientes herramientas de evaluación de pensamiento computacional (Román-González et al., 2019), las cuales pueden clasificarse según su enfoque evaluativo:

- **Instrumentos de diagnóstico de PC:** Están diseñadas para medir el nivel de aptitud en PC del sujeto. Su ventaja principal es que pueden administrarse en condiciones de pretest puro (por ejemplo, sujetos sin experiencia previa en programación). y postest (es decir, después de una intervención educativa) para verificar si la capacidad de PC ha aumentado. Román-González et al. (2019) señalan algunas de las herramientas de diagnóstico de PC:
  - Test de Pensamiento Computacional (Román-González et al., 2017).
  - Test para Medir Habilidades Básicas de Programación (Mühling, R., y Hubwieser, 2015).
  - Test de Evaluación Conmutativa (Weintrop et al., 2015).
- **Instrumentos sumativos de PC:** El objetivo es evaluar si el alumno ha adquirido suficiente conocimiento de contenido después de recibir alguna instrucción en habilidades de PC. El principal uso de las herramientas sumativas de PC se encuentra en la condición de postest. Podemos distinguir entre varias de estas herramientas según el entorno de aprendizaje utilizado:
  - Las herramientas sumativas de Meerbaum-Salant et al. (2013) en el contexto de Scratch.
  - Quizly1 (Maiorana et al., 2015) para evaluar el conocimiento de contenido en el contexto de App Inventor.
  - Evaluación Fairy (Werner et al., 2012), una herramienta basada en el rendimiento que se ejecuta en el entorno de Alice.

- **Herramientas iterativas-formativas de PC:** Están diseñadas para proporcionar retroalimentación al alumnado, generalmente de manera automática, con el fin de desarrollar y mejorar sus habilidades de PC. Estas herramientas no evalúan a los individuos, sino sus productos de aprendizaje, generalmente proyectos de programación. Por lo tanto, estas herramientas se utilizan principalmente durante el proceso de aprendizaje y están diseñadas específicamente para un entorno de programación particular. Entre estas herramientas podemos encontrar:
  - Dr. Scratch (Moreno-León et al., 2015) o Ninja Code Village (Ota et al., 2016) para el lenguaje de programación Scratch.
- **Herramientas de minería de datos PC:** Estas herramientas, al igual que las anteriores, están enfocadas en el proceso de aprendizaje. Sin embargo, mientras que las herramientas formativas-iterativas analizan estáticamente el código fuente de los proyectos de programación, las herramientas de minería de datos recuperan y registran la actividad del educando en tiempo real. Estas últimas herramientas proporcionan datos valiosos y análisis del aprendizaje a partir de los cuales se pueden inferir los procesos cognitivos del sujeto.
- **Herramientas de transferencia de habilidades PC:** El objetivo es evaluar en qué medida los estudiantes son capaces de transferir sus habilidades de CT a diferentes tipos de problemas, contextos y situaciones. Destacan las siguientes:
  - The Bebras Tasks (Dagiene y Futschek, 2008), que se centran en medir la transferencia de habilidades de PC a problemas de la vida real.
  - CTP-Quiz (Basawapatna et al., 2011), que evalúa cómo se transfieren las habilidades de CT al contexto de problemas científicos y simulaciones.
- **Escalas de percepciones-actitudes de PC:** Están diseñadas para evaluar las percepciones (por ejemplo, percepciones de autoeficacia) y actitudes de los sujetos no solo sobre PC, sino también sobre temas relacionados como computadoras, ciencias de la computación, programación informática o incluso alfabetización digital. Destacan las siguientes:
  - Escalas de Pensamiento Computacional (CTS) (Korkmaz et al., 2017).
  - Escala de Habilidades de Pensamiento Computacional (CTSS) (Durak y Saritepeci, 2018).
- **Evaluación de vocabulario PC:** estas herramientas tienen la intención de medir varios elementos y dimensiones de PC cuando son expresados verbalmente por los sujetos.

### 1.6.1. Análisis de herramientas para evaluar el pensamiento computacional

Siguiendo la clasificación anterior, las herramientas de diagnóstico que tienen como objetivo capturar la competencia en CT de los estudiantes podrían ser la base más adecuada para formar niveles de competencia. Las herramientas de diagnóstico son pruebas de rendimiento y no requieren conocimientos previos específicos, por ejemplo, un lenguaje de programación específico. Por lo tanto, se pueden utilizar para evaluar los avances en el aprendizaje mediante la comparación de los resultados de las pruebas previas y posteriores, es decir, si los estudiantes han alcanzado un nivel de competencia más alto después de la instrucción (Guggemos et al., 2022).

Una desventaja de muchas herramientas de diagnóstico es que no están disponibles de forma gratuita. Sin embargo, podemos destacar el Test De Pensamiento Computacional (Román-González, 2015) primeramente por su libre accesibilidad y segundo porque puede ser especialmente adecuado para el propósito de formar niveles de competencia.

El CTt (Test Pensamiento Computacional) es una prueba de rendimiento para estudiantes de secundaria que utiliza el marco teórico de Brennan y Resnick (2012) como base teórica. Define el CT como "la capacidad de formular y resolver problemas confiando en los conceptos fundamentales de la informática y utilizando la lógica-sintaxis de los lenguajes de programación: secuencias básicas, bucles, iteraciones, condicionales, funciones y variables" (Román-González et al., 2017).

A pesar de su idoneidad para evaluar conceptos computacionales también tiene desventajas: descuida las perspectivas computacionales (Román-González et al., 2017). Para capturar tales perspectivas, las escalas de percepción-actitud pueden ser adecuadas (Román-González et al., 2019). Complementar una prueba de rendimiento con un instrumento de autoevaluación puede ser beneficioso para obtener una imagen integral de un sujeto (Rosman et al., 2015).

Polat et al. (2021) destacan que el Test Pensamiento Computacional (CTt) y las Escalas de Pensamiento Computacional (CTS) pueden ser herramientas de evaluación complementarias que pueden proporcionar una imagen integral de la habilidad de PC de los estudiantes. El CTt, como una prueba de rendimiento unidimensional, parece ser adecuado para formar niveles de competencia, especialmente porque las operaciones

cognitivas están documentadas. El CTS, como un instrumento de autoevaluación multidimensional, puede utilizarse para identificar perfiles de PC.

## 2. Objetivos y preguntas de la investigación

Este estudio tiene como objeto analizar y evaluar las competencias y actitudes en pensamiento computacional entre estudiantes, centrándose en la variación a lo largo de diferentes cursos académicos y las diferencias de género.

Esta finalidad se concreta con los siguientes objetivos específicos:

- Identificar las preguntas con la tasa más alta de respuestas correctas e incorrectas en el CTt.
- Analizar las diferencias entre hombres y mujeres en las tasas de respuestas correctas e incorrectas en el CTt.
- Conocer cómo varía el desempeño en el CTt a lo largo de los diferentes cursos académicos.
- Explorar las percepciones de los estudiantes sobre las competencias y actitudes en pensamiento computacional.
- Determinar si existen diferencias significativas de género en las competencias y actitudes relacionadas con el pensamiento computacional.
- Analizar si existen diferencias significativas desde la experiencia académica sobre pensamiento computacional.

## 3. Metodología

La investigación que se ha llevado a cabo durante el transcurso del trabajo es de carácter cuantitativa, basada en la recopilación y el análisis de datos numéricos. Dicha recopilación se produjo a través de un cuestionario de 45 minutos, el segundo cuatrimestre del año lectivo de 2024 y el análisis se realizó mediante técnicas estadísticas.

Los participantes del cuestionario son alumnos/as de la Academia Internacional de Programación Algorithmics San Sebastián, una escuela de programación de educandos entre 6 y 17 años. El objetivo educativo de este centro consiste en desarrollar el pensamiento algorítmico, lógico y creativo, así como la gestión de proyectos y las habilidades de trabajo en equipo. Los lenguajes de programación más trabajados son Scratch y Python.

### 3.1. Muestra

La muestra está compuesta por 155 sujetos ( $n=155$ ) de la población objetivo y el procedimiento de muestreo utilizado es no probabilístico, de conveniencia e intencional. De conveniencia dada la facilidad de acceso de los sujetos, e intencional, por la intención de seleccionar un grupo de población que haya desarrollado procesos de aprendizaje relacionados con el lenguaje de programación y pensamiento computacional.

La muestra se distribuye de la siguiente manera:

- Género: Masculino ( $n = 122$ ), Femenino ( $n = 33$ )
- Edad: 8 años ( $n = 4$ ), 9 años ( $n = 8$ ), 10 años ( $n = 17$ ), 11 años ( $n = 27$ ), 12 años ( $n = 29$ ), 13 años ( $n = 37$ ), 14 años ( $n = 17$ ), 15 años ( $n = 8$ ), 16 años ( $n = 3$ )
- Curso escolar: LH4 ( $n = 13$ ), LH5 ( $n = 22$ ), LH6 ( $n = 37$ ), DBH1 ( $n = 30$ ), DBH2 ( $n = 29$ ), DBH3 ( $n = 12$ ), DBH4 ( $n = 7$ ), BATX 1 ( $n = 5$ )
- Experiencia académica: 1 curso ( $n = 79$ ), 2 cursos ( $n = 31$ ), 3 cursos ( $n = 18$ ), más de 3 cursos ( $n = 27$ )
- Lenguajes de programación aprendidos: Scratch ( $n = 67$ ), Python ( $n = 32$ ), ambos ( $n = 56$ )

### 3.2. Instrumentos

Para la recopilación de datos se ha utilizado un cuestionario mediante la herramienta *Google Forms* (Gorrochategui, 2024) que está distribuido en dos secciones. La primera de ellas estaba compuesta por 28 ítems de opción múltiples (4 opciones de respuesta y una sola correcta) del CTt (Test de Pensamiento Computacional) de Román (2015), con el objetivo de medir el desarrollo de pensamiento computacional del alumnado (**anexo 1**).

El Test de Pensamiento Computacional está diseñado y destinado principalmente a estudiantes entre 12 y 14 años (1º y 2º de ESO); aunque también se puede utilizar en grados inferiores (5º y 6º Primaria) y superiores (3º y 4º ESO).

Cada ítem del CTt está diseñado y caracterizado de acuerdo con las siguientes cinco dimensiones del dominio de muestreo (Román et al., 2017):

- **Concepto computacional abordado:** cada ítem aborda uno o más de los siguientes siete conceptos computacionales, ordenados en dificultad creciente: Direcciones y secuencias básicas (4 ítems); Bucles: (8 ítems); Si – condicional

simple (4 ítems); If/else-condicional complejo (4 ítems); Mientras condicional (4 ítems); Funciones simples (4 ítems).

- **Entorno-Interfaz del ítem:** Los ítems CTt se presentan en cualquiera de los dos entornos-interfaces siguientes: 'The Maze' (23 ítems) o 'The Canvas' (5 ítems). Estilo de alternativas de respuesta: Flechas visuales (8 ítems) o Bloques visuales (20 ítems).
- **Existencia o no existencia de anidamiento:** dependiendo de si la solución del ítem involucra un script con (19 ítems) o sin (9 ítems) conceptos computacionales anidados (un concepto incrustado en otro a un nivel jerárquico superior).
- **Tarea requerida:** dependiendo de cuál de las siguientes tareas cognitivas se requiere para la resolución del ítem: Secuenciación: el estudiante debe secuenciar, expresando de manera ordenada, un conjunto de comandos (14 ítems); Finalización: el estudiante debe completar un conjunto de comandos incompleto (9 ítems); Depuración: el estudiante debe depurar un conjunto de comandos incorrecto (5 elementos).

En la segunda parte encontramos las Escalas de Pensamiento Computacional (CTS) desarrolladas por Korkmaz et al. (2017) con la intención de capturar las perspectivas computacionales. Este instrumento estandarizado utiliza el marco de trabajo del Pensamiento Computacional de la Sociedad Internacional para la Tecnología en la Educación (ISTE, 2015), es decir, las cinco dimensiones de creatividad, pensamiento algorítmico, cooperatividad, pensamiento crítico y resolución de problemas.

El CTS consta de 29 preguntas de autoevaluación distribuidas en 5 bloques (dimensiones), en nuestro caso hemos hecho una adaptación al perfil académico de la muestra, simplificando el lenguaje de las preguntas puesto que este estudio se llevó a cabo con alumnado de grados superiores. A su vez, hemos reducido las escalas en 24 afirmaciones (**tabla 4**), evaluadas mediante una escala Likert de 1 a 5, donde 1 representa "nunca" y 5 representa "siempre".

Tabla 4. Adaptación de las escalas de PC (CTS).

|                                |  |
|--------------------------------|--|
| <b>Creatividad</b>             | 1. Me gustan las personas que están seguras de la mayoría de sus decisiones.   |
|                                | 2. Me gustan las personas que son realistas y justas.  |
|                                | 3. Creo que puedo resolver la mayoría de los problemas a los que me enfrente si tengo suficiente tiempo y si muestro esfuerzo.         |
|                                | 4. Pienso y creo que puedo resolver los problemas que puedan producirse cuando me encuentro con una situación nueva.                   |
|                                | 5. Cuando sueño y tengo ideas creativas, mis proyectos más importantes salen a la luz.   |
| <b>Pensamiento algorítmico</b> | 6. Cuando me encuentro con un problema, me detengo antes de pasar a otro tema y reflexiono sobre ese problema.                         |
|                                | 7. Creo que tengo un interés especial en las matemáticas.  |
|                                | 8. Utilizo las matemáticas para solucionar los problemas que tengo en la vida diaria.  |
|                                | 9. Puedo decidir rápidamente qué es justo para resolver un problema.   |
|                                | 10. Puedo convertir un problema matemático que se explica con palabras en un formato digital que pueda ser utilizado por un ordenador. |
| <b>Cooperación</b>             | 11. Me gusta trabajar y experimentar el aprendizaje cooperativo junto con mi grupo de amigos.  |
|                                | 12. En el aprendizaje cooperativo, creo que logro/lograré resultados más exitosos porque estoy trabajando en grupo.                    |
|                                | 13. Me gusta resolver problemas relacionados con proyectos grupales junto con mis amigos en el aprendizaje cooperativo.                |
|                                | 14. En el aprendizaje cooperativo se producen más ideas.   |
| <b>Pensamiento crítico</b>     | 15. Soy bueno/a preparando planes y estrategias para resolver problemas complicados.   |
|                                | 16. Es divertido intentar resolver problemas difíciles.  |
|                                | 17. Estoy dispuesto/a a aprender cosas nuevas y desafiantes.   |
|                                | 18. Estoy orgulloso/a de poder pensar con gran precisión.  |
|                                | 19. Sigues un plan o una serie de pasos para asegurarte de elegir la mejor opción posible cuando quieres resolver un problema.         |
| <b>Resolución de problemas</b> | 20. Tengo dificultades para demostrar la solución de un problema en mi mente.  |
|                                | 21. No puedo desarrollar mis propias ideas en el ambiente de aprendizaje cooperativo.  |
|                                | 22. Me cansa intentar aprender algo junto con mi grupo de amigos en el aprendizaje cooperativo.  |
|                                | 23. Tienes dificultades para pensar en muchas opciones cuando intentas resolver un problema.   |
|                                | 24. Tienes dificultades para aplicar paso a paso una solución a algún problema que puedas tener.                                       |

### 3.3. Resultados

En el siguiente apartado, presentamos los resultados de la muestra en relación con el pensamiento computacional.

#### Tasas de aciertos por ítem en el CTt

En la **tabla 5** se muestra la tasa de acierto por ítem de las 155 personas que realizaron el Test de Pensamiento Computacional. Los resultados más destacados muestran que las preguntas 1 (98,06% de aciertos), 5 (96,13% de aciertos), y las preguntas 2, 3 y 9 (todas con 92,90% de aciertos) fueron las que tuvieron los mayores porcentajes de respuestas correctas. Además, las preguntas 6 (90,97% de aciertos), 13 (85,16% de aciertos), 10 (86,45% de aciertos) y 28 (81,29% de aciertos) también reflejan un buen desempeño.

Sin embargo, hay áreas que presentan dificultades. La pregunta 23 tuvo solo un 25,81% de aciertos, mientras que las preguntas 12 y 15 también mostraron bajos porcentajes de aciertos, con 36,77% y 37,42% respectivamente. Otras preguntas que evidencian un menor rendimiento incluyen la pregunta 19 (52,26% de aciertos) y la pregunta 17 (51,61% de aciertos).

Tabla 5. Tasa acierto por ítem.

|             | Total   |       |       |       |
|-------------|---------|-------|-------|-------|
|             | Acierto |       | Error |       |
|             | N       | %     | N     | %     |
| PREGUNTA 1  | 152     | 98,06 | 3     | 1,94  |
| PREGUNTA 2  | 144     | 92,90 | 11    | 7,10  |
| PREGUNTA 3  | 144     | 92,90 | 11    | 7,10  |
| PREGUNTA 4  | 113     | 72,90 | 42    | 27,10 |
| PREGUNTA 5  | 149     | 96,13 | 6     | 3,87  |
| PREGUNTA 6  | 141     | 90,97 | 14    | 9,03  |
| PREGUNTA 7  | 128     | 82,58 | 27    | 17,42 |
| PREGUNTA 8  | 112     | 72,26 | 43    | 27,74 |
| PREGUNTA 9  | 144     | 92,90 | 11    | 7,10  |
| PREGUNTA 10 | 134     | 86,45 | 21    | 13,55 |
| PREGUNTA 11 | 125     | 80,65 | 30    | 19,35 |
| PREGUNTA 12 | 57      | 36,77 | 98    | 63,23 |
| PREGUNTA 13 | 132     | 85,16 | 23    | 14,84 |
| PREGUNTA 14 | 114     | 73,55 | 41    | 26,45 |
| PREGUNTA 15 | 58      | 37,42 | 97    | 62,58 |
| PREGUNTA 16 | 108     | 69,68 | 47    | 30,32 |
| PREGUNTA 17 | 80      | 51,61 | 75    | 48,39 |
| PREGUNTA 18 | 104     | 67,10 | 51    | 32,90 |
| PREGUNTA 19 | 81      | 52,26 | 74    | 47,74 |
| PREGUNTA 20 | 113     | 72,90 | 42    | 27,10 |
| PREGUNTA 21 | 111     | 71,61 | 44    | 28,39 |
| PREGUNTA 22 | 88      | 56,77 | 67    | 43,23 |
| PREGUNTA 23 | 40      | 25,81 | 115   | 74,19 |
| PREGUNTA 24 | 125     | 80,65 | 30    | 19,35 |
| PREGUNTA 25 | 81      | 52,26 | 74    | 47,74 |
| PREGUNTA 26 | 119     | 76,77 | 36    | 23,23 |
| PREGUNTA 27 | 110     | 70,97 | 45    | 29,03 |
| PREGUNTA 28 | 126     | 81,29 | 29    | 18,71 |

## Diferencias de género en tasas de respuestas en el CTt

Por otro lado, en la **tabla 6** se observa el rendimiento del alumnado diferenciando el género y pone de manifiesto que en general, ambos géneros demostraron un alto nivel de competencia en varias preguntas. Las mujeres obtuvieron un 100% de aciertos en las preguntas 1 y 5, superando ligeramente a los hombres en estas preguntas (97,54% y 95,08% respectivamente). Sin embargo, en las preguntas como 12 se evidencia una notable diferencia: solo el 24,24% de las mujeres respondieron correctamente frente al 40,16% de los hombres.

Tabla 6. Tasa de acierto por ítem en base al género.

|             | Hombres |       |         |       | Mujeres |        |         |       |
|-------------|---------|-------|---------|-------|---------|--------|---------|-------|
|             | Acierto |       | Errores |       | Acierto |        | Errores |       |
|             | N       | %     | N       | %     | N       | %      | N       | %     |
| PREGUNTA 1  | 119     | 97,54 | 3       | 2,46  | 33      | 100,00 | 0       | 0,00  |
| PREGUNTA 2  | 115     | 94,26 | 7       | 5,74  | 29      | 87,88  | 4       | 12,12 |
| PREGUNTA 3  | 113     | 92,62 | 9       | 7,38  | 31      | 93,94  | 2       | 6,06  |
| PREGUNTA 4  | 90      | 73,77 | 32      | 26,23 | 23      | 69,70  | 10      | 30,30 |
| PREGUNTA 5  | 116     | 95,08 | 6       | 4,92  | 33      | 100,00 | 0       | 0,00  |
| PREGUNTA 6  | 109     | 89,34 | 13      | 10,66 | 32      | 96,97  | 1       | 3,03  |
| PREGUNTA 7  | 102     | 83,61 | 20      | 16,39 | 26      | 78,79  | 7       | 21,21 |
| PREGUNTA 8  | 89      | 72,95 | 33      | 27,05 | 23      | 69,70  | 10      | 30,30 |
| PREGUNTA 9  | 114     | 93,44 | 8       | 6,56  | 30      | 90,91  | 3       | 9,09  |
| PREGUNTA 10 | 106     | 86,89 | 16      | 13,11 | 28      | 84,85  | 5       | 15,15 |
| PREGUNTA 11 | 100     | 81,97 | 22      | 18,03 | 25      | 75,76  | 8       | 24,24 |
| PREGUNTA 12 | 49      | 40,16 | 73      | 59,84 | 8       | 24,24  | 25      | 75,76 |
| PREGUNTA 13 | 104     | 85,25 | 18      | 14,75 | 28      | 84,85  | 5       | 15,15 |
| PREGUNTA 14 | 94      | 77,05 | 28      | 22,95 | 20      | 60,61  | 13      | 39,39 |
| PREGUNTA 15 | 44      | 36,07 | 78      | 63,93 | 14      | 42,42  | 19      | 57,58 |
| PREGUNTA 16 | 87      | 71,31 | 35      | 28,69 | 21      | 63,64  | 12      | 36,36 |
| PREGUNTA 17 | 63      | 51,64 | 59      | 48,36 | 17      | 51,52  | 16      | 48,48 |
| PREGUNTA 18 | 88      | 72,13 | 34      | 27,87 | 16      | 48,48  | 17      | 51,52 |
| PREGUNTA 19 | 67      | 54,92 | 55      | 45,08 | 14      | 42,42  | 19      | 57,58 |
| PREGUNTA 20 | 92      | 75,41 | 30      | 24,59 | 21      | 63,64  | 12      | 36,36 |
| PREGUNTA 21 | 90      | 73,77 | 32      | 26,23 | 21      | 63,64  | 12      | 36,36 |
| PREGUNTA 22 | 74      | 60,66 | 48      | 39,34 | 14      | 42,42  | 19      | 57,58 |
| PREGUNTA 23 | 33      | 27,05 | 89      | 72,95 | 7       | 21,21  | 26      | 78,79 |
| PREGUNTA 24 | 99      | 81,15 | 23      | 18,85 | 26      | 78,79  | 7       | 21,21 |
| PREGUNTA 25 | 66      | 54,10 | 56      | 45,90 | 15      | 45,45  | 18      | 54,55 |
| PREGUNTA 26 | 95      | 77,87 | 27      | 22,13 | 24      | 72,73  | 9       | 27,27 |
| PREGUNTA 27 | 87      | 71,31 | 35      | 28,69 | 23      | 69,70  | 10      | 30,30 |
| PREGUNTA 28 | 97      | 79,51 | 25      | 20,49 | 29      | 87,88  | 4       | 12,12 |

Otras preguntas donde las mujeres tuvieron un rendimiento notable incluyen la 6, 96,97% de aciertos comparado con 89,34% de los hombres, y la 28, con un 87,88% frente a 79,51%. En cambio, las preguntas 14 y 19 muestran una brecha significativa en los resultados, con las mujeres obteniendo 60,61% y 42,42% de aciertos frente al 77,05% y 54,92% de los hombres, respectivamente. La pregunta 18, sin embargo, es un caso interesante donde los hombres (72,13%) superaron claramente a las mujeres (48,48%).

### **Variabilidad del desempeño en el CTt por cursos académicos**

En la **tabla 7** se aporta una comparativa entre la tasa de acierto por ítem y la experiencia académica del discente. En general, los estudiantes de tercer y cuarto curso demostraron un rendimiento superior en comparación con los de primer y segundo curso. Por ejemplo, en preguntas como la 1 y la 5, los estudiantes de tercer y cuarto curso lograron un 100% de aciertos, mientras que los de primer curso obtuvieron un 96,25% y 95%, y los de segundo curso un 96,77% y 90,32% respectivamente.

En la pregunta 12 los de primer curso, solo obtuvieron un 30% de aciertos, en comparación con el 44,44% en tercer y cuarto curso. La pregunta 23 también resultó difícil, con un 11,25% de aciertos en primer curso y un 44,44% en tercer y cuarto curso, mostrando una mejora significativa con la experiencia académica.

En preguntas específicas, los de tercer y cuarto curso mostraron un rendimiento consistente y elevado. Por ejemplo, en la pregunta 10, el segundo curso obtuvo un 93,55% de aciertos, el tercer curso un 94,44%, y el cuarto curso un 92,59%, comparado con el 78,75% de primer curso.

Tabla 7. Tasa de acierto y experiencia académica.

|             | 1 curso  |       |         |       | 2 curso  |       |         |       | 3 curso  |        |         |       | 4 curso  |        |         |       |
|-------------|----------|-------|---------|-------|----------|-------|---------|-------|----------|--------|---------|-------|----------|--------|---------|-------|
|             | Aciertos |       | Errores |       | Aciertos |       | Errores |       | Aciertos |        | Errores |       | Aciertos |        | Errores |       |
|             | N        | %     | N       | %     | N        | %     | N       | %     | N        | %      | N       | %     | N        | %      | N       | %     |
| PREGUNTA 1  | 77       | 96,25 | 3       | 3,75  | 30       | 96,77 | 1       | 3,23  | 18       | 100,00 | 0       | 0,00  | 27       | 100,00 | 0       | 0,00  |
| PREGUNTA 2  | 71       | 88,75 | 9       | 11,25 | 30       | 96,77 | 1       | 3,23  | 18       | 100,00 | 0       | 0,00  | 25       | 92,59  | 2       | 7,41  |
| PREGUNTA 3  | 69       | 86,25 | 11      | 13,75 | 30       | 96,77 | 1       | 3,23  | 18       | 100,00 | 0       | 0,00  | 27       | 100,00 | 0       | 0,00  |
| PREGUNTA 4  | 54       | 67,5  | 26      | 32,5  | 20       | 64,52 | 11      | 35,48 | 16       | 88,89  | 2       | 11,11 | 23       | 85,19  | 4       | 14,81 |
| PREGUNTA 5  | 76       | 95    | 4       | 5     | 28       | 90,32 | 3       | 9,68  | 18       | 100,00 | 0       | 0,00  | 27       | 100,00 | 0       | 0,00  |
| PREGUNTA 6  | 67       | 83,75 | 13      | 16,25 | 30       | 96,77 | 1       | 3,23  | 18       | 100,00 | 0       | 0,00  | 26       | 96,30  | 1       | 3,70  |
| PREGUNTA 7  | 59       | 73,75 | 21      | 26,25 | 27       | 87,10 | 4       | 12,90 | 18       | 100,00 | 0       | 0,00  | 24       | 88,89  | 3       | 11,11 |
| PREGUNTA 8  | 45       | 56,25 | 35      | 43,75 | 27       | 87,10 | 4       | 12,90 | 15       | 83,33  | 3       | 16,67 | 25       | 92,59  | 2       | 7,41  |
| PREGUNTA 9  | 71       | 88,75 | 9       | 11,25 | 30       | 96,77 | 1       | 3,23  | 17       | 94,44  | 1       | 5,56  | 26       | 96,30  | 1       | 3,70  |
| PREGUNTA 10 | 63       | 78,75 | 17      | 21,25 | 29       | 93,55 | 2       | 6,45  | 17       | 94,44  | 1       | 5,56  | 25       | 92,59  | 2       | 7,41  |
| PREGUNTA 11 | 59       | 73,75 | 21      | 26,25 | 27       | 87,10 | 4       | 12,90 | 17       | 94,44  | 1       | 5,56  | 22       | 81,48  | 5       | 18,52 |
| PREGUNTA 12 | 24       | 30    | 56      | 70    | 13       | 41,94 | 18      | 58,06 | 8        | 44,44  | 10      | 55,56 | 12       | 44,44  | 15      | 55,56 |
| PREGUNTA 13 | 62       | 77,5  | 18      | 22,5  | 27       | 87,10 | 4       | 12,90 | 17       | 94,44  | 1       | 5,56  | 26       | 96,30  | 1       | 3,70  |
| PREGUNTA 14 | 52       | 65    | 28      | 35    | 25       | 80,65 | 6       | 19,35 | 15       | 83,33  | 3       | 16,67 | 22       | 81,48  | 5       | 18,52 |
| PREGUNTA 15 | 28       | 35    | 52      | 65    | 7        | 22,58 | 24      | 77,42 | 11       | 61,11  | 7       | 38,89 | 12       | 44,44  | 15      | 55,56 |
| PREGUNTA 16 | 48       | 60    | 32      | 40    | 23       | 74,19 | 8       | 25,81 | 17       | 94,44  | 1       | 5,56  | 20       | 74,07  | 7       | 25,93 |
| PREGUNTA 17 | 35       | 43,75 | 45      | 56,25 | 20       | 64,52 | 11      | 35,48 | 10       | 55,56  | 8       | 44,44 | 15       | 55,56  | 12      | 44,44 |
| PREGUNTA 18 | 43       | 53,75 | 37      | 46,25 | 23       | 74,19 | 8       | 25,81 | 16       | 88,89  | 2       | 11,11 | 22       | 81,48  | 5       | 18,52 |
| PREGUNTA 19 | 37       | 46,25 | 43      | 53,75 | 12       | 38,71 | 19      | 61,29 | 16       | 88,89  | 2       | 11,11 | 16       | 59,26  | 11      | 40,74 |
| PREGUNTA 20 | 48       | 60    | 32      | 40    | 26       | 83,87 | 5       | 16,13 | 16       | 88,89  | 2       | 11,11 | 23       | 85,19  | 4       | 14,81 |
| PREGUNTA 21 | 47       | 58,75 | 33      | 41,25 | 25       | 80,65 | 6       | 19,35 | 17       | 94,44  | 1       | 5,56  | 22       | 81,48  | 5       | 18,52 |
| PREGUNTA 22 | 32       | 40    | 48      | 60    | 21       | 67,74 | 10      | 32,26 | 12       | 66,67  | 6       | 33,33 | 23       | 85,19  | 4       | 14,81 |
| PREGUNTA 23 | 9        | 11,25 | 71      | 88,75 | 11       | 35,48 | 20      | 64,52 | 8        | 44,44  | 10      | 55,56 | 12       | 44,44  | 15      | 55,56 |
| PREGUNTA 24 | 58       | 72,5  | 22      | 27,5  | 26       | 83,87 | 5       | 16,13 | 18       | 100,00 | 0       | 0,00  | 23       | 85,19  | 4       | 14,81 |
| PREGUNTA 25 | 40       | 50    | 40      | 50    | 12       | 38,71 | 19      | 61,29 | 10       | 55,56  | 8       | 44,44 | 19       | 70,37  | 8       | 29,63 |
| PREGUNTA 26 | 56       | 70    | 24      | 30    | 25       | 80,65 | 6       | 19,35 | 16       | 88,89  | 2       | 11,11 | 22       | 81,48  | 5       | 18,52 |
| PREGUNTA 27 | 48       | 60    | 32      | 40    | 26       | 83,87 | 5       | 16,13 | 15       | 83,33  | 3       | 16,67 | 21       | 77,78  | 6       | 22,22 |
| PREGUNTA 28 | 60       | 75    | 20      | 25    | 28       | 90,32 | 3       | 9,68  | 14       | 77,78  | 4       | 22,22 | 24       | 88,89  | 3       | 11,11 |

## Percepciones estudiantiles sobre competencias en PC

Centrándonos en la segunda parte del cuestionario, en la **tabla 8** medimos mediante medias y DT las escalas del PC. Las respuestas que muestran medias superiores a los 4 puntos indican una alta concordancia con la mayoría de las afirmaciones positivas; entre ellas, se destaca la preferencia por las personas que son realistas y justas, con una media de 4,32 y una desviación estándar de 0,820. Esta tendencia sugiere que los participantes valoran significativamente estas cualidades en los demás.

De manera similar, la afirmación "Creo que puedo resolver la mayoría de los problemas a los que me enfrento si tengo suficiente tiempo y si muestro esfuerzo" obtuvo una media de 4,26 y una desviación estándar de 0,883, lo que refleja una alta confianza en las propias capacidades para resolver problemas. Además, la disposición a aprender cosas nuevas y desafiantes también se destaca con una media de 4,37 y una desviación estándar de 0,799, indicando un fuerte interés por el aprendizaje continuo y la automejora.

Por otro lado, las afirmaciones que presentan medias por debajo de los 3 puntos no evidencian dificultades o desinterés en ciertas áreas, sino que indican que los participantes rara vez tienen problemas en esos aspectos. Por ejemplo, la afirmación "Tengo dificultades para demostrar la solución de un problema en mi mente" tiene una media de 2,39 y una desviación estándar de 1,229, lo que indica que muchos participantes en contadas ocasiones encuentran complicado visualizar mentalmente las soluciones.

Otro caso parecido como, "No puedo desarrollar mis propias ideas en el ambiente de aprendizaje cooperativo" y "Me cansa intentar aprender algo junto con mi grupo de amigos en el aprendizaje cooperativo" presentan medias de 2,41 y 2,21 respectivamente, con desviaciones estándar de 1,268 y 1,243, sugiriendo que los participantes rara vez perciben estas dificultades en el aprendizaje cooperativo. Además, la afirmación "Tienes dificultades para pensar en muchas opciones cuando intentas resolver un problema" obtuvo una media de 2,27 y una desviación estándar de 1,234, reflejando que los participantes rara vez tienen limitaciones en la generación de alternativas para la resolución de problemas.

Tabla 8. Medias en escalas PC.

|  | Media | DT    |
|--|-------|-------|
| 1. Me gustan las personas que están seguras de la mayoría de sus decisiones.   | 4,00  | ,781  |
| 2. Me gustan las personas que son realistas y justas.  | 4,32  | ,820  |
| 3. Creo que puedo resolver la mayoría de los problemas a los que me enfrento si tengo suficiente tiempo y si muestro esfuerzo.         | 4,26  | ,883  |
| 4. Pienso y creo que puedo resolver los problemas que puedan producirse cuando me encuentro con una situación nueva.                   | 3,88  | ,926  |
| 5. Cuando sueño y tengo ideas creativas, mis proyectos más importantes salen a la luz.   | 3,50  | 1,261 |
| 6. Cuando me encuentro con un problema, me detengo antes de pasar a otro tema y reflexiono sobre ese problema.                         | 3,55  | 1,146 |
| 7. Creo que tengo un interés especial en las matemáticas.  | 3,70  | 1,276 |
| 8. Utilizo las matemáticas para solucionar los problemas que tengo en la vida diaria.  | 3,39  | 1,136 |
| 9. Puedo decidir rápidamente qué es justo para resolver un problema.   | 3,76  | ,961  |
| 10. Puedo convertir un problema matemático que se explica con palabras en un formato digital que pueda ser utilizado por un ordenador. | 3,35  | 1,177 |
| 11. Me gusta trabajar y experimentar el aprendizaje cooperativo junto con mi grupo de amigos.  | 4,14  | 1,119 |
| 12. En el aprendizaje cooperativo, creo que logro/lograré resultados más exitosos porque estoy trabajando en grupo.                    | 3,72  | 1,131 |
| 13. Me gusta resolver problemas relacionados con proyectos grupales junto con mis amigos en el aprendizaje cooperativo.                | 3,81  | 1,105 |
| 14. En el aprendizaje cooperativo se producen más ideas.   | 4,28  | ,923  |
| 15. Soy bueno/a preparando planes y estrategias para resolver problemas complicados.   | 3,87  | ,895  |
| 16. Es divertido intentar resolver problemas difíciles.  | 3,65  | 1,312 |
| 17. Estoy dispuesto/a a aprender cosas nuevas y desafiantes.   | 4,37  | ,799  |
| 18. Estoy orgulloso/a de poder pensar con gran precisión.  | 4,35  | ,887  |
| 19. Sigues un plan o una serie de pasos para asegurarte de elegir la mejor opción posible cuando quieres resolver un problema.         | 3,79  | 1,157 |
| 20. Tengo dificultades para demostrar la solución de un problema en mi mente.  | 2,39  | 1,229 |
| 21. No puedo desarrollar mis propias ideas en el ambiente de aprendizaje cooperativo.  | 2,41  | 1,268 |
| 22. Me cansa intentar aprender algo junto con mi grupo de amigos en el aprendizaje cooperativo.  | 2,21  | 1,243 |
| 23. Tienes dificultades para pensar en muchas opciones cuando intentas resolver un problema.   | 2,27  | 1,234 |
| 24. Tienes dificultades para aplicar paso a paso una solución a algún problema que puedas tener.                                       | 2,40  | 1,292 |

## Diferencias de género en competencias y actitudes relacionadas con el PC

En la **tabla 9** se muestran datos de las medias de las percepciones del PC en base al género. En varios aspectos clave, tanto hombres como mujeres superan los 4 puntos de media en los siguientes ítems de la encuesta. En primer lugar, tanto hombres como mujeres muestran una inclinación positiva hacia las personas que están seguras de la mayoría de sus decisiones, con puntuaciones medias de 4.12 (DT = 0.781) y 4.36 (DT = 0.781) respectivamente.

Asimismo, ambos géneros valoran la justicia y el realismo en las relaciones interpersonales, con puntuaciones medias de 4.30 (DT = 0.832) para hombres y 4.36 (Dt = 0.783) para mujeres en este ítem específico.

Además, tanto hombres como mujeres comparten la creencia en su capacidad para resolver problemas con suficiente tiempo y esfuerzo, con medias de 4.25 (DT = 0.884) y 4.33 (Dt = 0.890) respectivamente.

La predisposición hacia el aprendizaje cooperativo también es evidente, con ambos géneros mostrando una fuerte preferencia por trabajar en equipo y experimentar el aprendizaje colaborativo con sus amigos, con puntuaciones medias de 4.11 (Dt = 1.107) para hombres y 4.24 (Dt = 1.173) para mujeres en este ítem específico.

Los resultados revelan diferencias significativas de género en el uso de las matemáticas para resolver problemas cotidianos ( $p = .037$ ). Los participantes masculinos reportaron una media de 3.48 (DT = 1.166), mientras que las participantes femeninas mostraron una media ligeramente menor de 3.06 (DT = 0.966).

Los resultados, también revelan una diferencia significativa de género en la preferencia por resolver problemas en proyectos grupales dentro del contexto del aprendizaje cooperativo ( $p = .046$ ). Los participantes masculinos muestran una media de 3.76 (DT = 1.143), mientras que las participantes femeninas exhiben una media ligeramente más alta de 3.97 (Dt = 0.951).

Los datos exhiben una significativa disparidad de género en la capacidad para generar ideas dentro del entorno de aprendizaje cooperativo ( $p = .008$ ). Los participantes

masculinos revelaron una media de 2.48 (DT = 1.325), en contraste con las participantes femeninas que reportaron una media notablemente más baja de 2.18 (DT = 1.014).

Los datos demuestran una significativa disparidad de género en la percepción del esfuerzo asociado al aprendizaje cooperativo en grupo ( $p = .027$ ). Los participantes masculinos reportan una media de 2.27 (DT = 1.292), mientras que las participantes femeninas exhiben una media ligeramente menor de 2.00 (DT = 1.031).

Tabla 9. Percepciones sobre el PC según el género.

|  | Género | Media | DT    | F     | Sig  |
|--|--------|-------|-------|-------|------|
| 1. Me gustan las personas que están seguras de la mayoría de sus decisiones.   | 1      | 3,97  | ,781  | ,171  | ,680 |
|  | 2      | 4,12  | ,781  |       |      |
| 2. Me gustan las personas que son realistas y justas.  | 1      | 4,30  | ,832  | ,051  | ,821 |
|  | 2      | 4,36  | ,783  |       |      |
| 3. Creo que puedo resolver la mayoría de los problemas a los que me enfrente si tengo suficiente tiempo y si muestro esfuerzo.         | 1      | 4,25  | ,884  | ,087  | ,769 |
|  | 2      | 4,33  | ,890  |       |      |
| 4. Pienso y creo que puedo resolver los problemas que puedan producirse cuando me encuentro con una situación nueva.                   | 1      | 3,88  | ,905  | 2,080 | ,151 |
|  | 2      | 3,91  | 1,011 |       |      |
| 5. Cuando sueño y tengo ideas creativas, mis proyectos más importantes salen a la luz.   | 1      | 3,45  | 1,260 | ,015  | ,902 |
|  | 2      | 3,70  | 1,262 |       |      |
| 6. Cuando me encuentro con un problema, me detengo antes de pasar a otro tema y reflexiono sobre ese problema.                         | 1      | 3,54  | 1,151 | ,040  | ,841 |
|  | 2      | 3,61  | 1,144 |       |      |
| 7. Creo que tengo un interés especial en las matemáticas.  | 1      | 3,87  | 1,192 | 2,763 | ,099 |
|  | 2      | 3,06  | 1,391 |       |      |
| 8. Utilizo las matemáticas para solucionar los problemas que tengo en la vida diaria.  | 1      | 3,48  | 1,166 | 4,427 | ,037 |
|  | 2      | 3,06  | ,966  |       |      |
| 9. Puedo decidir rápidamente qué es justo para resolver un problema.   | 1      | 3,80  | ,906  | 3,448 | ,065 |
|  | 2      | 3,61  | 1,144 |       |      |
| 10. Puedo convertir un problema matemático que se explica con palabras en un formato digital que pueda ser utilizado por un ordenador. | 1      | 3,46  | 1,144 | ,138  | ,711 |
|  | 2      | 2,97  | 1,237 |       |      |
| 11. Me gusta trabajar y experimentar el aprendizaje cooperativo junto con mi grupo de amigos.  | 1      | 4,11  | 1,107 | ,204  | ,653 |
|  | 2      | 4,24  | 1,173 |       |      |
| 12. En el aprendizaje cooperativo, creo que logro/lograré resultados más exitosos porque estoy trabajando en grupo.                    | 1      | 3,66  | 1,133 | 1,782 | ,184 |
|  | 2      | 3,94  | 1,116 |       |      |
| 13. Me gusta resolver problemas relacionados con proyectos grupales junto con mis amigos en el aprendizaje cooperativo.                | 1      | 3,76  | 1,143 | 4,048 | ,046 |
|  | 2      | 3,97  | ,951  |       |      |
| 14. En el aprendizaje cooperativo se producen más ideas.   | 1      | 4,26  | ,943  | ,112  | ,739 |
|  | 2      | 4,33  | ,854  |       |      |
| 15. Soy bueno/a preparando planes y estrategias para resolver problemas complicados.   | 1      | 3,89  | ,855  | 2,710 | ,102 |
|  | 2      | 3,82  | 1,044 |       |      |
| 16. Es divertido intentar resolver problemas difíciles.  | 1      | 3,74  | 1,284 | ,971  | ,326 |
|  | 2      | 3,33  | 1,384 |       |      |
| 17. Estoy dispuesto/a a aprender cosas nuevas y desafiantes.   | 1      | 4,43  | ,727  | 2,582 | ,110 |
|  | 2      | 4,15  | 1,004 |       |      |
| 18. Estoy orgulloso/a de poder pensar con gran precisión.  | 1      | 4,36  | ,824  | 3,882 | ,051 |
|  | 2      | 4,30  | 1,104 |       |      |
| 19. Sigues un plan o una serie de pasos para asegurarte de elegir la mejor opción posible cuando quieres resolver un problema.         | 1      | 3,76  | 1,143 | ,081  | ,776 |
|  | 2      | 3,88  | 1,219 |       |      |

|  |   |      |       |       |      |
|--|---|------|-------|-------|------|
| 20. Tengo dificultades para demostrar la solución de un problema en mi mente.                    | 1 | 2,35 | 1,192 | 2,597 | ,109 |
|  | 2 | 2,52 | 1,372 |       |      |
| 21. No puedo desarrollar mis propias ideas en el ambiente de aprendizaje cooperativo.            | 1 | 2,48 | 1,325 | 7,181 | ,008 |
|  | 2 | 2,18 | 1,014 |       |      |
| 22. Me cansa intentar aprender algo junto con mi grupo de amigos en el aprendizaje cooperativo.  | 1 | 2,27 | 1,292 | 4,960 | ,027 |
|  | 2 | 2,00 | 1,031 |       |      |
| 23. Tienes dificultades para pensar en muchas opciones cuando intentas resolver un problema.     | 1 | 2,31 | 1,227 | ,065  | ,800 |
|  | 2 | 2,12 | 1,269 |       |      |
| 24. Tienes dificultades para aplicar paso a paso una solución a algún problema que puedas tener. | 1 | 2,43 | 1,298 | ,355  | ,552 |
|  | 2 | 2,27 | 1,281 |       |      |

## Impacto de la experiencia académica en PC

Finalmente, la **tabla 10** enseña las percepciones en base a la experiencia académica del alumnado. En el grupo con un curso completado, destaca la preferencia por trabajar y experimentar el aprendizaje cooperativo junto con amigos, con una media de 4,18 y una desviación estándar de 1,152. Además, la afirmación "En el aprendizaje cooperativo se producen más ideas" obtuvo una media de 4,18 y una desviación estándar de 0,902, lo que refleja una valoración positiva del trabajo en equipo. Los participantes con un curso también muestran una fuerte disposición a aprender cosas nuevas y desafiantes, con una media de 4,38 y una desviación estándar de 0,773.

En el grupo con dos cursos completados, la valoración más alta se encuentra en la afirmación "En el aprendizaje cooperativo se producen más ideas", con una media de 4,45 y una desviación estándar de 0,888. La disposición a aprender cosas nuevas y desafiantes también es destacable, con una media de 4,35 y una desviación estándar de 0,877. El gusto por trabajar en equipo es igualmente alto, con una media de 4,52 y una desviación estándar de 0,626.

Para los participantes con tres cursos completados, la media más alta se observa en la afirmación "Estoy dispuesto/a a aprender cosas nuevas y desafiantes", con una media de 4,44 y una desviación estándar de 0,705. Además, "Me gustan las personas que son realistas y justas" obtuvo una media de 4,44 y una desviación estándar de 0,705, indicando una fuerte valoración de estas cualidades. La afirmación "En el aprendizaje cooperativo se producen más ideas" también se destaca con una media de 4,22 y una desviación estándar de 0,732.

En el grupo con más de tres cursos completados, la afirmación "Estoy dispuesto/a a aprender cosas nuevas y desafiantes" obtuvo una media de 4,33 y una desviación estándar de 0,877, mientras que "En el aprendizaje cooperativo se producen más ideas" registró una media de 4,41 y una desviación estándar de 1,118. Estas tendencias reflejan una alta valoración del aprendizaje continuo y del trabajo cooperativo.

Por otro lado, las afirmaciones que presentan medias por debajo de los 3 puntos no evidencian dificultades o desinterés en ciertas áreas, sino que indican que los participantes rara vez tienen problemas en esos aspectos. En el grupo con un curso completado, la afirmación "Me cansa intentar aprender algo junto con mi grupo de amigos en el aprendizaje cooperativo" tiene una media de 2,34 y una desviación estándar de 1,260, sugiriendo una percepción mayormente positiva del aprendizaje en grupo. En los participantes con dos cursos completados, la media más baja se observa en la afirmación "Me cansa intentar aprender algo junto con mi grupo de amigos en el aprendizaje cooperativo", con una media de 1,61 y una desviación estándar de 0,882.

Para los participantes con tres cursos completados, la afirmación "Tienes dificultades para pensar en muchas opciones cuando intentas resolver un problema" obtuvo una media de 2,11 y una desviación estándar de 1,079, indicando que rara vez encuentran dificultades en la generación de alternativas. En el grupo con más de tres cursos completados, las medias más bajas se encontraron en las afirmaciones relacionadas con la dificultad para pensar en opciones y aplicar soluciones paso a paso, ambas con una media de 2,11 y desviaciones estándar de 1,121 y 1,013 respectivamente.

El estudio de comparación de medias evidencia que en la afirmación "Me cansa intentar aprender algo junto con mi grupo de amigos en el aprendizaje cooperativo", se encontraron diferencias significativas entre los grupos con distintos números de cursos completados. En particular, la comparación entre el grupo con un curso completado y el grupo con dos cursos completados mostró una diferencia significativa con un valor p de 0,029. Esta diferencia sugiere que los participantes con un curso completado encuentran más agotador el aprendizaje cooperativo en comparación con aquellos que han completado dos cursos.

Tabla 10. Percepciones sobre el PC según la experiencia.

|  | 1     |       | 2     |       | 3     |       | 4     |       | F     | Sig  |
|--|-------|-------|-------|-------|-------|-------|-------|-------|-------|------|
|  | Media | DT    | Media | DT    | Media | DT    | Media | DT    |       |      |
| 1.Me gustan las personas que están seguras de la mayoría de sus decisiones.  | 3,99  | ,809  | 4,06  | ,680  | 3,89  | ,676  | 4,04  | ,898  | ,216  | ,885 |
| 2.Me gustan las personas que son realistas y justas.   | 4,38  | ,867  | 4,13  | ,885  | 4,44  | ,705  | 4,26  | ,656  | ,885  | ,450 |
| 3.Creo que puedo resolver la mayoría de los problemas a los que me enfrento si tengo suficiente tiempo y si muestro esfuerzo.          | 4,33  | ,796  | 4,29  | 1,071 | 4,11  | ,832  | 4,15  | ,949  | ,482  | ,695 |
| 4.Pienso y creo que puedo resolver los problemas que puedan producirse cuando me encuentro con una situación nueva.                    | 3,92  | ,859  | 4,00  | ,894  | 4,06  | ,938  | 3,52  | 1,087 | 1,852 | ,140 |
| 5.Cuando sueño y tengo ideas creativas, mis proyectos más importantes salen a la luz.  | 3,42  | 1,317 | 3,77  | 1,175 | 3,33  | 1,085 | 3,56  | 1,311 | ,719  | ,542 |
| 6.Cuando me encuentro con un problema, me detengo antes de pasar a otro tema y reflexiono sobre ese problema.                          | 3,46  | 1,238 | 3,61  | 1,086 | 3,67  | 1,029 | 3,70  | 1,031 | ,428  | ,733 |
| 7.Creo que tengo un interés especial en las matemáticas.   | 3,53  | 1,348 | 3,87  | 1,204 | 3,89  | 1,132 | 3,85  | 1,231 | ,901  | ,442 |
| 8.Utilizo las matemáticas para solucionar los problemas que tengo en la vida diaria.   | 3,28  | 1,176 | 3,52  | 1,180 | 3,50  | 1,043 | 3,48  | 1,051 | ,490  | ,689 |
| 9.Puedo decidir rápidamente qué es justo para resolver un problema.  | 3,68  | 1,020 | 3,84  | ,860  | 3,89  | ,832  | 3,81  | 1,001 | ,369  | ,776 |
| 10. Puedo convertir un problema matemático que se explica con palabras en un formato digital que pueda ser utilizado por un ordenador. | 3,16  | 1,148 | 3,68  | 1,301 | 3,56  | 1,199 | 3,41  | 1,047 | 1,678 | ,174 |
| 11. Me gusta trabajar y experimentar el aprendizaje cooperativo junto con mi grupo de amigos.  | 4,18  | 1,152 | 4,52  | ,626  | 3,72  | 1,320 | 3,89  | 1,219 | 2,561 | ,057 |
| 12. En el aprendizaje cooperativo, creo que logro/lograré resultados más exitosos porque estoy trabajando en grupo.                    | 3,77  | 1,085 | 4,13  | ,922  | 3,33  | 1,283 | 3,37  | 1,245 | 3,088 | ,029 |
| 13. Me gusta resolver problemas relacionados con proyectos grupales junto con mis amigos en el aprendizaje cooperativo.                | 3,77  | 1,109 | 4,00  | ,931  | 3,83  | 1,200 | 3,67  | 1,240 | ,485  | ,693 |
| 14. En el aprendizaje cooperativo se producen más ideas.   | 4,18  | ,902  | 4,45  | ,888  | 4,22  | ,732  | 4,41  | 1,118 | ,877  | ,454 |
| 15. Soy bueno/a preparando planes y estrategias para resolver problemas complicados.   | 3,89  | ,920  | 3,90  | ,790  | 3,83  | ,924  | 3,81  | ,962  | ,066  | ,978 |
| 16. Es divertido intentar resolver problemas difíciles.  | 3,63  | 1,322 | 3,77  | 1,230 | 3,56  | 1,423 | 3,63  | 1,363 | ,128  | ,943 |
| 17. Estoy dispuesto/a a aprender cosas nuevas y desafiantes.   | 4,38  | ,773  | 4,35  | ,877  | 4,44  | ,705  | 4,33  | ,877  | ,076  | ,973 |
| 18. Estoy orgulloso/a de poder pensar con gran precisión.  | 4,35  | ,848  | 4,29  | 1,006 | 4,56  | ,616  | 4,26  | 1,023 | ,459  | ,712 |
| 19. Sigues un plan o una serie de pasos para asegurarte de elegir la mejor opción posible cuando quieres resolver un problema.         | 3,89  | 1,086 | 4,00  | 1,125 | 3,50  | 1,200 | 3,44  | 1,311 | 1,727 | ,164 |
| 20. Tengo dificultades para demostrar la solución de un problema en mi mente.  | 2,59  | 1,345 | 2,16  | 1,098 | 2,28  | 1,274 | 2,11  | ,892  | 1,622 | ,187 |
| 21. No puedo desarrollar mis propias ideas en el ambiente de aprendizaje cooperativo.  | 2,54  | 1,289 | 1,84  | ,969  | 2,50  | 1,581 | 2,63  | 1,149 | 2,787 | ,043 |
| 22. Me cansa intentar aprender algo junto con mi grupo de amigos en el aprendizaje cooperativo.  | 2,34  | 1,260 | 1,61  | ,882  | 2,83  | 1,465 | 2,11  | 1,155 | 4,538 | ,004 |
| 23. Tienes dificultades para pensar en muchas opciones cuando intentas resolver un problema.   | 2,41  | 1,354 | 2,16  | 1,098 | 2,11  | 1,079 | 2,11  | 1,121 | ,639  | ,591 |
| 24. Tienes dificultades para aplicar paso a paso una solución a algún problema que puedas tener.                                       | 2,61  | 1,363 | 1,97  | 1,197 | 2,67  | 1,328 | 2,11  | 1,013 | 2,621 | ,053 |

#### 4. Discusión

Los primeros resultados concluyen que el rendimiento en el Test de PC se incrementa con la experiencia académica, algo lógico con la percepción de que el PC es una aptitud cognitiva que está vinculada con el desarrollo madurativo-evolutivo de los individuos (Román, 2015).

Por otro lado, no existen demasiadas diferencias en el rendimiento si diferenciamos los resultados en base al género. Es verdad que los hombres han obtenido una tasa de acierto más elevada que las mujeres y eso puede ser motivo de la autoeficacia, una percepción que puede afectar de forma negativa a las chicas al tener una menor confianza para responder a la prueba. La falta de autoeficacia puede generar obstáculos en las mujeres a la hora de afrontar el aprendizaje de la programación y podría disuadirlas de futuros estudios académicos y profesionales relacionados con la materia. Estudios como el de Román-González et al. (2018), subrayan la correlación existente entre el PC, autoeficacia y las diversas dimensiones de la personalidad humana (“cinco grandes”), donde obtuvieron diferencias de género en autoeficacia para tareas de codificación a favor de los hombres.

Los resultados de las escalas de PC también nos dejan hallazgos destacables como las medias obtenidas en las dimensiones de “resolución de problemas” y “creatividad”, dos áreas esenciales e interrelacionadas que potencian el desarrollo del PC. Según Wing (2006), el PC es una forma de resolver problemas de manera inteligente e imaginativa, y los datos reflejan que los estudiantes poseen una buena disposición para enfrentar desafíos de forma creativa y efectiva.

El PC es un proceso de resolución de problemas (ISTE y CSTA, 2011) que incluye habilidades como organizar y analizar datos de forma lógica, automatizar soluciones (secuenciar) e identificar, analizar e implementar posibles soluciones para obtener la solución más eficiente, dichas habilidades las encontramos en las afirmaciones 20, 23 y 24, donde hemos obtenidos resultados destacables.

Siguiendo la misma línea, los resultados reflejan una alta confianza en la capacidad para resolver problemas (media de 4,26) y un fuerte interés por el aprendizaje continuo (media de 4,37), lo cual se relaciona directamente con el desarrollo del pensamiento computacional. La persistencia en resolver problemas y la confianza en manejar

dificultades son actitudes clave en este campo (mencionadas en la **tabla 1**), donde la capacidad para enfrentar desafíos complejos y la disposición a aprender constantemente son esenciales para encontrar soluciones efectivas.

Además, destacamos los resultados obtenidos en la escala del aprendizaje cooperativo en el desarrollo del Pensamiento Computacional (PC), alineándose con las disposiciones esenciales identificadas por el ISTE y CSTA (2011), donde remarca una de las actitudes que potencia el desarrollo del PC: La capacidad de comunicarse y trabajar con otros para lograr un objetivo o solución común. El área de aprendizaje cooperativo presentaba unas medias bajas, señal de que rara vez presentaban dificultades. Estas habilidades son fundamentales para la adquisición y aplicación efectiva del PC, como se evidenció en los resultados donde ambos géneros demostraron una fuerte preferencia por este enfoque colaborativo. Tomando en cuenta los resultados específicos del estudio sobre el aprendizaje cooperativo y las percepciones variadas según la experiencia académica de los participantes, hemos observado una mayor predisposición al aprendizaje colaborativo entre los más veteranos.

Huang y Parker (2022), examinan cómo el pensamiento computacional se desarrolla de manera efectiva en pequeños grupos de trabajo colaborativo. La interacción y cooperación en equipos reducidos facilita la adquisición de habilidades de pensamiento computacional, destacando la importancia de las prácticas computacionales en un entorno colaborativo (Chowdhury et. al 2018).

Por último y no menos importante, los resultados de las medias en las escalas basándonos en el género no presentan grandes diferencias, al igual que en la literatura, donde apenas se han encontrado hallazgos relevantes. Una de las diferencias más significativa se ha mostrado en la afirmación del uso de las matemáticas para la resolución de problemas cotidianos, donde los hombres han destacado casi medio punto por encima de las mujeres. Los estudios de Niousha et. al (2022) y Funke y Geldreich (2017) encontraron diferencias en los proyectos de Scratch elaborados por hombres y mujeres. Los hombres programaban proyectos con bloques más complejos y técnicos, frente al de las mujeres donde prevalecía más el arte y la creatividad. Estos primeros eran más propensos a crear juegos y hacían un mayor uso de bloques relacionados con conceptos lógico-matemáticos.

Por otro lado, Jansen et. al (2016) realizaron un estudio donde correlacionaban los factores afectivos y motivacionales entre las habilidades matemáticas y el uso de ellas en la vida cotidiana. Los resultados mostraron que las mujeres reportaron mayor ansiedad matemática, menor competencia matemática percibida y menor uso de las matemáticas en la vida cotidiana, en comparación con los hombres. Y es que la motivación, junto con la ya mencionada autoeficacia, es otro de los aspectos cognitivos causantes de la brecha de género en la iniciación a la programación (Torres et. al, 2022).

Armoni et al. (2015) comentan que los lenguajes de programación visuales (bloques) aumentan la motivación en estudiantes de primaria y secundaria, promoviendo la codificación entre las mujeres.

## 5. Conclusiones

Concluimos este trabajo planteando lo crucial que es la alfabetización en la sociedad digital actual. En este trabajo se han disipado diferentes conceptos relacionados con la codificación, programación y pensamiento computacional presentes en la literatura, donde el consenso es ambiguo. Además, se ha revisado la literatura existente para resaltar los beneficios de estos términos, como el desarrollo del pensamiento lógico, la mejora en la resolución de problemas, el fomento de la creatividad y la capacidad de trabajar de manera colaborativa. Estos beneficios subrayan el potencial significativo de aplicar la codificación, la programación y el pensamiento computacional en los procesos de enseñanza-aprendizaje, mejorando la competencia digital de los estudiantes y preparándolos para enfrentar los desafíos de la era digital.

Los resultados nos muestran que los estudiantes con formación en programación muestran habilidades avanzadas en la resolución de problemas, pensamiento creativo y aprendizaje cooperativo entre otros. Esto subraya la importancia de incorporar la programación en los currículos educativos.

Un currículo en el que después de analizarlo muestra una presencia desproporcionada del pensamiento computacional en todos los niveles educativos. La implementación es más común y abundante en niveles superiores y en centros educativos con una apuesta fuerte en la innovación tecnopedagógica.

Además, hemos resaltado la brecha de género existente en la enseñanza de la programación, siendo fundamental estudiar cómo comienza la enseñanza de la programación para los estudiantes, identificar posibles brechas de género en esta etapa inicial y, si existen, tomar medidas adecuadas para abordarlas y de esta manera garantizar una alfabetización en programación equitativa.

Asimismo, hemos llevado a cabo una revisión, clasificación y análisis de los instrumentos de evaluación del pensamiento computacional según los distintos enfoques evaluativos, lo que nos ha permitido entender mejor cómo se mide esta habilidad y su alineación con las necesidades actuales de la sociedad.

Centrándonos en el objetivo principal de la investigación, se realizó un estudio sociológico de carácter cuantitativo haciendo uso de dos instrumentos cuya complementariedad nos proporcionaba una visión íntegra de la habilidad del pensamiento computacional del sujeto. El Test de Pensamiento Computacional, como una prueba de rendimiento, para identificar el nivel de competencia computacional y las escalas, como un instrumento de autoevaluación, para identificar perfiles de pensamiento computacional.

Los resultados obtenidos confirman que los instrumentos utilizados en el estudio son complementarios, efectivos, válidos y confiables para proporcionar una visión completa del pensamiento computacional de los estudiantes, demostrando una mejoría en los datos de los educandos con más experiencia y conocimiento en códigoalfabetización.

El Test De Pensamiento Computacional resulta una herramienta muy útil para identificar el nivel de competencia en PC (nivel inicial sobretodo) y una detección temprana en altas capacidades, pero también tiene sus puntos a mejorar. Por ejemplo, la limitación de respuesta de los ítems, donde solo tenemos la opción multirrespuesta. Abriendo el abanico de tipos de respuesta podríamos potenciar más habilidades en el pensamiento como la opción de poder codificar, fomentar la creatividad, problemas complejos de distinta índole etc.

Por otro lado, las escalas nos permiten identificar perfiles en PC del estudiante, cubriendo la medición de las perspectivas computacionales que no es capaz de evaluar el test. Sin embargo, se debe realizar un estudio de adaptación en caso de que se desee aplicar este estudio a los estudiantes de secundaria o de niveles inferiores. Además, el pensamiento computacional está limitado a las sub-habilidades expresadas en la definición

realizada por ISTE (2015) en esta escala y a los ítems básicos que se encuentran en las escalas desarrolladas por separado para estas habilidades.

## 6. Futuras líneas de investigación

Marcos Román González y Özgen Korkmaz se destacan por sus numerosas publicaciones y la adaptación internacional de sus instrumentos. La Escala de Pensamiento Computacional (CTS) ha sido adaptada y validada psicométricamente en Europa y Asia, mientras que la Prueba de Pensamiento Computacional (CTt) ha sido adaptada y su validez ha sido establecida en las mismas regiones.

Sin embargo, los instrumentos se aplicaron principalmente a estudiantes de secundaria y universitarios, futuras investigaciones deberían dirigirse a otras poblaciones, como etapas educativas infantiles y primarias, donde cada día florecen más proyectos e iniciativas pedagógicas de codigoalfabetización.

Por otro lado, una línea interesante de investigación se puede centrar en analizar cómo la inclusión de la programación en el currículo educativo puede influir en el desempeño académico general de los estudiantes, especialmente en áreas científicas. Es primordial estudiar la transferencia de las habilidades de pensamiento computacional, adquiridas a través de la programación, a otras áreas del conocimiento y su aplicación en la vida cotidiana. Investigaciones en este ámbito pueden revelar la magnitud del impacto de la programación no solo en el aprendizaje de conceptos específicos, sino también en el desarrollo de competencias clave que faciliten el entendimiento y la resolución de problemas complejos en diversas disciplinas. Comprender estas dinámicas puede aportar evidencias sólidas sobre el valor de integrar la programación en los currículos, destacando su potencial para mejorar el rendimiento académico.

## Bibliografía

- Armoni, M., Meerbaum-Salant, O., & Ben-Ari, M. (2015). From Scratch to «Real» Programming. *ACM Transactions on Computing Education*, 14(4), 1-15. <https://dl.acm.org/doi/10.1145/2677087>
- Basawapatna, A., Koh, K. H., Repenning, A., Webb, D. C., & Marshall, K. S. (2011). Recognizing computational thinking patterns. In *Proceedings of the 42nd ACM Technical Symposium on Computer Science Education* (pp. 245–250). <https://doi.org/10.1145/1953163.1953241>.
- Basogain, X., Olabe, M.A. & Olabe, J.C., (2015). Pensamiento Computacional a través de la Programación: Paradigma de Aprendizaje. *RED. Revista de Educación a Distancia*. Número 46. 30 de Septiembre de 2015. Consultado el (dd/mm/aa) en <http://www.um.es/ead/red/46>
- Bers, M. U. (2018). *Coding as a playground: Programming and computational thinking in the early childhood classroom*. Routledge. doi:<https://doi.org/10.4324/9781315398945>
- Brennan, K. y Resnick, M. (2012). *Proceedings of the Annual American Educational Research Association Meeting: New frameworks for studying and assessing the development of computational thinking*. Vancouver, Canada.
- Bustillo Bayón, J. (2015). Formación del profesorado con scratch: análisis de la escasa incidencia en el aula. *Opción*, 31(1), 164–182. Retrieved from <http://www.redalyc.org/pdf/310/31043005010.pdf>
- Carretero, S., Vuorikari, R., & Punie, Y. (2017). *DigComp 2.1: The Digital Competence Framework for Citizens with eight proficiency levels and examples of use*. Publications Office of the European Union.
- Castañeda, L., Esteve, F., & Adell, J. (2018). ¿Por qué es necesario repensar la competencia docente para el mundo digital? *RED*, 56(6), 1-20. <http://dx.doi.org/10.6018/red/56/6>

- Chowdhury, B., Bart, A., & Kafura, D. (2018). Analysis of Collaborative Learning in a Computational Thinking Class. *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*. <https://doi.org/10.1145/3159450.3159470>.
- Dagiene, V., y Futschek, G. (2008). Bebras international contest on informatics and computer literacy: Criteria for good tasks. In *International Conference on Informatics in Secondary Schools Evolution and Perspectives* (pp. 19–30). Berlin, Germany: Springer. [https://doi.org/10.1007/978-3-540-69924-8\\_2](https://doi.org/10.1007/978-3-540-69924-8_2).
- DECRETO 77/2023, de 30 de mayo, de establecimiento del currículo de Educación Básica e implantación en la Comunidad Autónoma de Euskadi.
- 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>.
- Espino Espino, Elisenda & González González, Carina. (2016). Estudio sobre Pensamiento Computacional y Género. *Revista Iberoamericana de Tecnologías del Aprendizaje*. 4. 119-128.
- Fedorenko, E., Ivanova, A., Dhamala, R., & Bers, M. U. (2019). The language of programming: A cognitive per-spective. *Trends in Cognitive Sciences*, 23(7), 525–528. <https://doi.org/10.1016/j.tics.2019.04.010>
- Funke, A., & Geldreich, K. (2017). Gender Differences in Scratch Programs of Primary School Children. *Proceedings of the 12th Workshop on Primary and Secondary Computing Education*. <https://doi.org/10.1145/3137065.3137067>.
- Gamito, R., Aristizabal, P., Basasoro, M., & León, I. (2022). «El desarrollo del pensamiento computacional en educación: valoración basada en una experiencia con Scratch». Innoeduca. *International Journal of Technology and Educational Innovation*, vol. 8, n.o 1, junio de 2022, pp. 59-74. [revistas.uma.es, https://doi.org/10.24310/innoeduca.2022.v8i1.12093](https://doi.org/10.24310/innoeduca.2022.v8i1.12093).

- González Martínez, J., Estebanell Minguell, M., & Peracaula Bosch, M. (2018). ¿Robots o programación? El concepto de Pensamiento Computacional y los futuros maestros. *Education in the Knowledge Society (EKS)*, 19(2), 29–45. <https://doi.org/10.14201/eks20181922945>
- González-González, C. S. (2019). Estado del arte en la enseñanza del pensamiento computacional y la programación en la etapa infantil. *Education in the Knowledge Society (EKS)*, 20, 15. [https://doi.org/10.14201/eks2019\\_20\\_a17](https://doi.org/10.14201/eks2019_20_a17)
- Gorrochategui, I. (2024). *Test de Pensamiento Computacional*. [Test en línea]. Recuperado de <https://forms.gle/2PtdxhjJQtYeiqcH7>
- Graafsmaa, I. L., Robidoux, S., Nickels, L., Robertse, M., Polito, V., Zhuband, J.D. & Marinus, E. (2023). The cognition of programming: logical reasoning, algebra and vocabulary skills predict programming performance following an introductory computing course. *JOURNAL OF COGNITIVE PSYCHOLOGY*, VOL. 35, NO. 3, 364–381 <https://doi.org/10.1080/20445911.2023.2166054>
- Grover, S., & Pea, R. (2013). Computational thinking in K– 12. *A review of the state of the field. Educational Researcher*, 42(1), 38-43
- Guggemos, J., Seufert, S. & Román-González, M. (2022). Computational Thinking Assessment – Towards More Vivid Interpretations. *Technology, Knowledge and Learning*. 28. 10.1007/s10758-021-09587-2.
- Huang, J., & Parker, M. (2022). Developing computational thinking collaboratively: the nexus of computational practices within small groups. *Computer Science Education*, 33, 342 - 374. <https://doi.org/10.1080/08993408.2022.2039488>.
- INTEF (s.f.). *Programación y Robótica: Programación textual*. <https://formacion.intef.es/mod/book/view.php?id=2625&chapterid=2400>.
- Informes PISA (2015). Accesible en: <http://bit.ly/1cXFBAh>
- Informe OCDE (2012). Accesible en: <http://bit.ly/1C38VBA>

- ISTE, & CSTA (2011). Operational Definition of Computational Thinking for K–12 Education. *National Science Foundation*. <https://bit.ly/31D0fbH>
- ISTE. (2015). Computational thinking: leadership toolkit. <https://www.iste.org/computational-thinking>
- Jansen, B., Schmitz, E., & Maas, H. (2016). Affective and Motivational Factors Mediate the Relation between Math Skills and Use of Math in Everyday Life. *Frontiers in Psychology*, 7. <https://doi.org/10.3389/fpsyg.2016.00513>.
- Korkmaz, Ö., Çakir, R., & Özden, M. Y. (2017). A validity and reliability study of the computational thinking scales (CTS). *Computers in Human Behavior*. <https://doi.org/10.1016/j.chb.2017.01.005>.
- 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. doi: <http://dx.doi.org/10.1016/j.chb.2014.09.012>
- López, J. C. (2014). *Impacto de Scratch en el desarrollo del pensamiento algorítmico*. Master Thesis. Universidad ICESI.
- Maiorana, F., Giordano, D., & Morelli, R. (2015). Quizly: A live coding assessment platform for App Inventor. In *2015 IEEE Blocks and Beyond Workshop (Blocks and Beyond)* (pp. 25–30). <https://doi.org/10.1109/BLOCKS.2015.7368995>.
- Meerbaum-Salant, O., Armoni, M., & Ben-Ari, M. (2013). Learning computer science concepts with scratch. *Computer Science Education*, 23(3), 239–264.
- Miranda, G., León, C. y Rodríguez, C. (2017). La programación, ¿una herramienta para desarrollar la capacidad de resolución de problemas?. *Imaginar y Comprender la Innovación en la Universidad*, 119-130.
- MIT (s. f.). *Media Lab*. <https://bit.ly/2ZvhyJc>
- Mishra, P., & Koehler, M. J. (2006). Technological Pedagogical Content Knowledge: A Framework for Teacher Knowledge. *Teachers College Record*, 108(6), 1017–1054. doi:<https://doi.org/10.1111/j.1467-9620.2006.00684.x>

- Monsálvez, J. (2017). Python como primer lenguaje de programación textual en la Enseñanza Secundaria. *Education in the Knowledge Society (EKS)*. 18. 147. 10.14201/eks2017182147162.
- Moreno-León, J., Robles, G., & Román-González, M. (2015). Dr. Scratch: Automatic analysis of scratch projects to assess and foster computational thinking. *RED. Revista de Educación a Distancia*, 15(46), 1–23. Retrieved from [http://www.um.es/ead/red/46/moreno\\_robles.pdf](http://www.um.es/ead/red/46/moreno_robles.pdf)
- Mühling, A., Ruf, A., y Hubwieser, P. (2015). Design and first results of a psychometric test for measuring basic programming abilities. *In Proceedings of the Workshop in Primary and Secondary Computing Education* (pp. 2–10). <https://doi.org/10.1145/2818314.2818320>.
- Niousha, R., Saito, D., Washizaki, H., & Fukazawa, Y. (2022). Gender Characteristics and Computational Thinking in Scratch. *Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 2*. <https://doi.org/10.1145/3545947.3576290>.
- Ota, G., Morimoto, Y., & Kato, H. (2016). Ninja code village for scratch: Function samples/function analyser and automatic assessment of computational thinking concepts. *In 2016 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)* (pp. 238–239).
- Papadakis, S. J., Kalogiannakis, M., Orfanakis, V. & Zaranis, N. (2017) The appropriateness of scratch and app inventor as educational environments for teaching introductory programming in primary and secondary education. *International Journal of Web-Based Learning and Teaching Technologies*, vol. 12(4), páginas 58-77.
- Paucar-Curasma, R., Cerna-Ruiz, L., Acra-Despradel, C., Villalba-Condori, K., Massa-Palacios, L., Olivera-Chura, A., & Esteban-Roblado, I. (2023). Desarrollo del Pensamiento Computacional a través de Actividades STEM para la Promoción de la Igualdad de Género. *Sostenibilidad* . <https://doi.org/10.3390/su151612335>.
- Peña Ros, R. (2015). Python como primera aproximación a la programación. *ReVisión*, 8(2), 17-29.

- Piedade, J., Dorotea, N., Ferrentini, F. S. & Pedro, A. (2019) *A cross-analysis of block-based and visual programming apps with computer science student-teachers*. Disponible en <https://www.mdpi.com/2227-7102/9/3/181>
- Polat, E., Hopcan, S., Kucuk, S., & Sisman, B. (2021). A comprehensive assessment of secondary school students' computational thinking skills. *British Journal of Educational Technology*, 52(5), 1965– 1980. <https://doi.org/10.1111/bjet.13092>
- Ramos, Y., y Carbonell, Y. (2021). ¿Por qué no estudiar matemáticas? *EduSol*, 21(74), 218-229.
- Román-González, M. (2014). Aprender a programar 'apps' como enriquecimiento curricular en alumnado de alta capacidad. *Bordón*. 66. 135-155. 10.13042/Bordon.2014.66401.
- Román-González, M. (2015). Computational thinking test: Design guidelines and content validation. <https://doi.org/10.13140/RG.2.1.4203.4329>
- Román-González, M., Pérez-González, J.-C., & 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>.
- Román-González, M., Pérez-González, J.-C., Moreno-León, J., & Robles, G. (2018). Extending the nomological network of computational thinking with non-cognitive factors. *Computers in Human Behavior*, 80, 441–459. <https://doi.org/10.1016/j.chb.2017.09.030>
- Román-González, M., Moreno-León, J. & Robles, G. (2019). *Combining Assessment Tools for a Comprehensive Evaluation of Computational Thinking Interventions*. 10.1007/978-981-13-6528-7.
- Rosman, T., Mayer, A.-K., & Krampen, G. (2015). Combining self-assessments and achievement tests in information literacy assessment: Empirical results and recommendations for practice. *Assessment & Evaluation in Higher Education*, 40(5), 740–754. <https://doi.org/10.1080/02602938.2014.950554>

- Rushkoff, D. (2010). *Program or be programmed*. New York: OR Books.
- Scherer, R., Siddiq, F. & Sánchez-Scherer, B. (2021). *Some Evidence on the Cognitive Benefits of Learning to Code*. *Front. Psychol.* 12:559424. doi: 10.3389/fpsyg.2021.559424
- Sullivan, A., & Bers, M. U. (2013). Gender differences in kindergarteners' robotics and programming achievement. *International Journal of Technology and Design Education*, 23 (3), 691-702.
- Tadeu, P. y Brigas, C. (2022). Computational thinking in early childhood education: an análisis through the Computer Science Unplugged. *Revista Interuniversitaria de Formación del Profesorado*, 98,149-166. <https://doi.org/10.47553/rifop.v98i36.2.94881>
- Torres-Torres, Y. D.; Román-González, M. & Pérez-González, J. C. (2022). Brechas de Género en la iniciación a la Programación Informática en Educación Secundaria en España. *Revista Complutense de Educación*,33(4), 701-712.
- UNESCO. (2019). *Descifrar el código: La educación de las niñas y las mujeres en Ciencias, Ingeniería, Tecnología y Matemáticas*. Paris: UNESCO.
- Valverde-Berrocoso, J., Fernández-Sánchez, M.R.,& Garrido-Arroyo, M.C. (2015). El pensamiento computacional y las nuevas ecologías del aprendizaje. *RED, Revista de Educación a Distancia*. Número 46. Número monográfico sobre «Pensamiento Computacional». Septiembre de 2015. Consultado el (dd/mm/aa) en <http://www.um.es/ead/red/46>
- Velázquez Iturbide, J. A. y Martín, M. (2021). Análisis del “pensamiento computacional”: una perspectiva educativa. *RED. Revista de Educación a Distancia*. Núm. 68, Vol. 21. Artíc. 6, 30-Nov-2021 DOI: <http://dx.doi.org/10.6018/red.484811>
- Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L. & Wilensky, U. (2015). Defining Computational Thinking for Mathematics and Science Classrooms. *Journal of Science Education and Technology*, 10. <http://dx.doi.org/10.1007/s10956-015-9581-5>

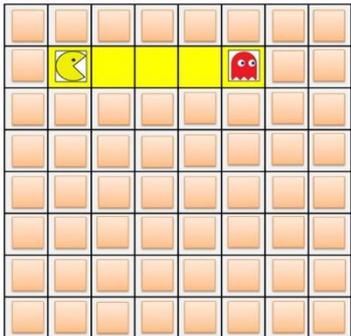
Werner, L., Denner, J., Campe, S., & Kawamoto, D. C. (2012). The fairy performance assessment: Measuring computational thinking in middle school. In Proceedings of the 43rd ACM Technical Symposium on Computer Science Education (pp. 215–220). Wing, J. M. (2006). *Computational thinking*. *Communications of the ACM*, 49(3), 33–35. <https://doi.org/10.1145/1118178.1118215>.

Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33–35. <https://doi.org/10.1145/1118178.1118215>.

Wing, J. M. (2008). Computational thinking and thinking about computing. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 366(1881), 3717–3725. <https://doi.org/10.1098/rsta.2008.0118>.

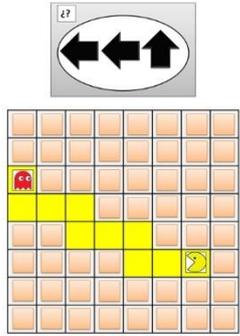


¿Qué órdenes llevan a 'Pac-Man' hasta el fantasma por el camino señalado?



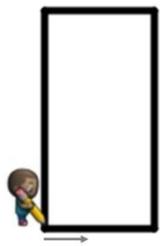
|   |  |
|---|--|
| <p>Opción A</p>  | <p>Opción B</p>  |
| <p>Opción C</p>  | <p>Opción D</p>  |

¿Cuántas veces se debe repetir la secuencia para llevar a 'Pac-Man' hasta el fantasma por el camino señalado?



|                            |
|----------------------------|
| <p>Opción A</p> <p>× 2</p> |
| <p>Opción B</p> <p>× 1</p> |
| <p>Opción C</p> <p>× 4</p> |
| <p>Opción D</p> <p>× 3</p> |

Para que el artista dibuje una vez el siguiente rectángulo (50 píxeles de ancho y 100 píxeles de alto), ¿en qué paso de la siguiente secuencia de órdenes hay un error?



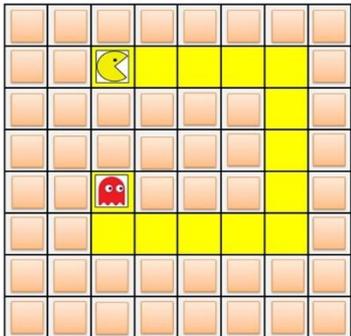
Paso A

```

repetir 4 veces
hacer
  mover hacia adelante 50 píxeles
  girar a la izquierda por 90 grados
  mover hacia adelante 100 píxeles
  girar a la izquierda por 90 grados
  
```

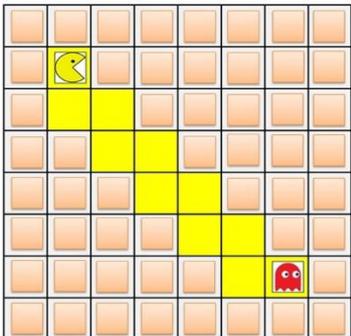
Paso B →  
 Paso C →  
 Paso D →

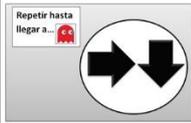
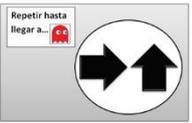
¿Qué órdenes llevan a 'Pac-Man' hasta el fantasma por el camino señalado?



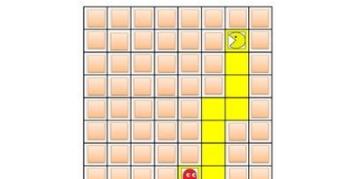
|  |  |
|--|--|
| <p>Opción A</p> <pre> repetir 4 veces haz   repetir 3 veces   haz     avanzar     girar a la derecha 90   avanzar   </pre> | <p>Opción B</p> <pre> repetir 3 veces haz   repetir 4 veces   haz     avanzar     girar a la derecha 90   avanzar   </pre> |
| <p>Opción C</p> <pre> repetir 3 veces haz   repetir 4 veces   haz     avanzar     girar a la derecha 90   avanzar   </pre> | <p>Opción D</p> <pre> repetir 4 veces haz   avanzar   repetir 3 veces   haz     girar a la derecha 90   avanzar   </pre>   |

¿Qué órdenes llevan a 'Pac-Man' hasta el fantasma por el camino señalado?



|   |  |
|---|--|
| <p>Opción A</p>  | <p>Opción B</p>  |
| <p>Opción C</p>  | <p>Opción D</p>  |

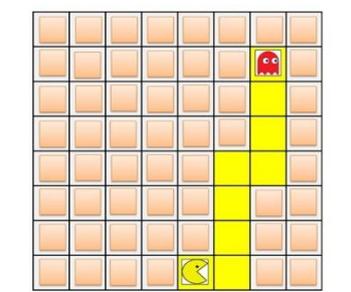
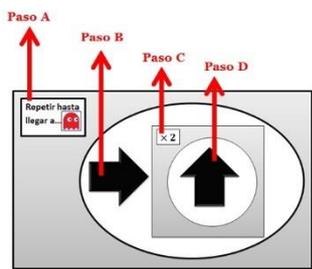
¿Qué bloque falta en la siguiente secuencia de órdenes para que 'Pac-Man' llegue hasta el fantasma por el camino señalado?



Repetir hasta llegar a...   
 hacer  
 girar a la izquierda   
 avanzar   
 avanzar   
 girar a la derecha   
 avanzar 

|   |  |
|---|--|
| <p>Opción A</p>  | <p>Opción B</p>  |
| <p>Opción C</p>  | <p>Opción D</p> <p>No falta ningún bloque</p>  |

Para que 'Pac-Man' llegue hasta el fantasma por el camino señalado, ¿en qué paso de la siguiente secuencia de órdenes hay un error?

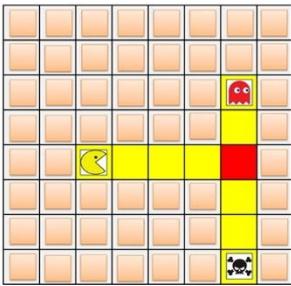



¿Qué secuencia de órdenes debe ejecutar el artista para dibujar la escalera que llegue hasta la flor? Cada peldaño sube 30 píxeles

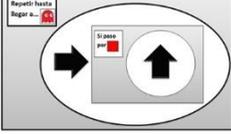


|   |  |
|---|--|
| <p>Opción A</p>  | <p>Opción B</p>  |
| <p>Opción C</p>  | <p>Opción D</p>  |

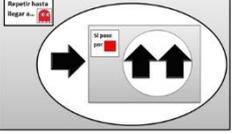
¿Qué órdenes llevan a 'Pac-Man' hasta el fantasma por el camino señalado?



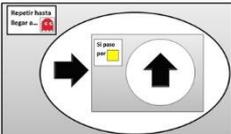
Opción A



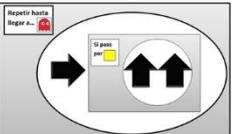
Opción B



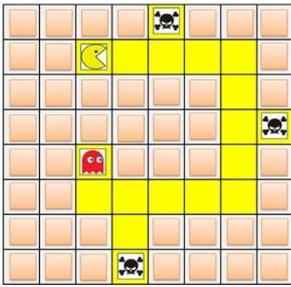
Opción C



Opción D



¿Qué órdenes llevan a 'Pac-Man' hasta el fantasma por el camino señalado?



Opción A



Opción B



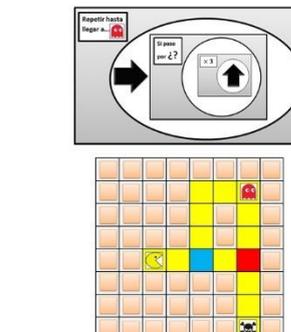
Opción C



Opción D



¿Qué falta en la siguiente secuencia de órdenes para llevar a 'Pac-Man' hasta el fantasma por el camino señalado?



Opción A



Opción B



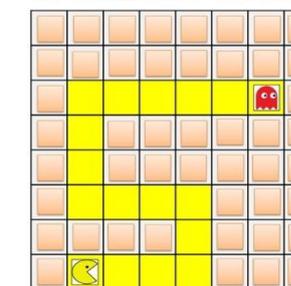
Opción C



Opción D

Tanto la opción A como la opción C son correctas

Para que 'Pac-Man' llegue hasta el fantasma por el camino señalado, ¿en qué paso de la siguiente secuencia de órdenes hay un error?



Repetir hasta llegar a...

hacer avanzar → Paso A

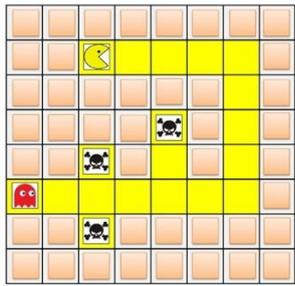
si hay camino a la izquierda

hacer girar a la izquierda → Paso B

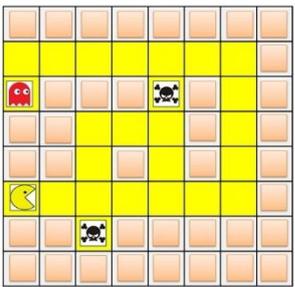
si hay camino a la derecha → Paso C

hacer avanzar → Paso D

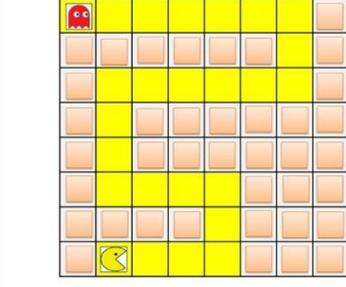
¿Qué órdenes llevan a 'Pac-Man' hasta el fantasma por el camino señalado?

|   |   |   |
|---|---|---|
|  | <p>Opción A</p> <pre> Repetir hasta llegar a... hacer   si hay un camino delante   hacer avanzar   sino girar a la izquierda         </pre> | <p>Opción B</p> <pre> Repetir hasta llegar a... hacer   si hay un camino delante   hacer avanzar   sino girar a la derecha         </pre>       |
|  | <p>Opción C</p> <pre> Repetir hasta llegar a... hacer   si hay camino a la derecha   hacer girar a la derecha   sino avanzar         </pre> | <p>Opción D</p> <pre> Repetir hasta llegar a... hacer   si hay camino a la izquierda   hacer girar a la izquierda   sino avanzar         </pre> |

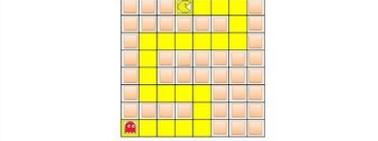
¿Qué órdenes llevan a 'Pac-Man' hasta el fantasma por el camino señalado?

|   |   |   |
|---|---|---|
|  | <p>Opción A</p> <pre> Repetir hasta llegar a... hacer   si hay un camino delante   hacer avanzar   sino girar a la izquierda         </pre> | <p>Opción B</p> <pre> Repetir hasta llegar a... hacer   si hay un camino delante   hacer avanzar   sino girar a la derecha         </pre>       |
|  | <p>Opción C</p> <pre> Repetir hasta llegar a... hacer   si hay camino a la derecha   hacer girar a la derecha   sino avanzar         </pre> | <p>Opción D</p> <pre> Repetir hasta llegar a... hacer   si hay camino a la izquierda   hacer girar a la izquierda   sino avanzar         </pre> |

Para que 'Pac-Man' llegue hasta el fantasma por el camino señalado, ¿en qué paso de la siguiente secuencia de órdenes hay un error?

|   |  |
|---|--|
|  | <pre> repetir hasta haz   si hay un camino delante   haz avanzar → Paso A   sino     si hay camino a la derecha     haz girar a la izquierda → Paso B     sino girar a la derecha → Paso C     sino girar a la izquierda → Paso D         </pre> |
|---|--|

¿Qué bloque falta en la siguiente secuencia de órdenes para que 'Pac-Man' llegue hasta el fantasma por el camino señalado?

|   |   |   |
|---|---|---|
| <pre> Repetir hasta llegar a... hacer   si hay un camino delante   hacer avanzar   sino     si hay camino a la derecha     hacer girar a la derecha     sino [?]         </pre> | <p>Opción A</p> <pre> avanzar         </pre>              | <p>Opción B</p> <pre> girar a la derecha         </pre> |
|    | <p>Opción C</p> <pre> girar a la izquierda         </pre> | <p>Opción D</p> <p>No falta ningún bloque</p>           |

¿Qué órdenes llevan a 'Pac-Man' por el camino señalado hasta las fresas e indican a 'Pac-Man' que se coma el número de fresas indicado?

|   |   |
|---|---|
| <p>Opción A</p> <pre> mientras haya camino delante hacer avanzar repetir 3 veces haz Comer 1 fresa                     </pre> | <p>Opción B</p> <pre> mientras haya camino delante hacer avanzar repetir 4 veces haz Comer 1 fresa                     </pre> |
| <p>Opción C</p> <pre> mientras haya camino delante hacer avanzar repetir 5 veces haz Comer 1 fresa                     </pre> | <p>Opción D</p> <pre> mientras haya camino delante hacer avanzar repetir 3 veces haz Comer 1 fresa                     </pre> |

¿Qué órdenes van llevando a 'Pac-Man' por el camino señalado e indicándole que se coma el número de fresas correspondiente?

|   |   |
|---|---|
| <p>Opción A</p> <pre> mientras haya camino delante haz repetir 5 veces hacer avanzar repetir 3 veces hacer Comer 1 fresa                     </pre> | <p>Opción B</p> <pre> mientras haya camino delante hacer avanzar repetir 3 veces haz Comer 1 fresa                     </pre> |
| <p>Opción C</p> <pre> mientras haya camino delante haz repetir 3 veces hacer avanzar repetir 5 veces hacer Comer 1 fresa                     </pre> | <p>Opción D</p> <pre> mientras haya camino delante hacer avanzar repetir 3 veces haz Comer 1 fresa                     </pre> |

¿Qué falta en la siguiente secuencia de órdenes para que 'Pac-Man' avance por el camino señalado comiendo el número de fresas indicadas?

```

mientras haya camino delante
haz repetir 2 veces
hacer avanzar
si hay alguna fresa
haz Comer 1 fresa
                    
```

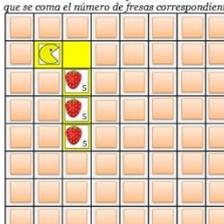
|                                |
|--------------------------------|
| <p>Opción A</p> <p>1 vez</p>   |
| <p>Opción B</p> <p>2 veces</p> |
| <p>Opción C</p> <p>3 veces</p> |
| <p>Opción D</p> <p>5 veces</p> |

¿Qué bloque falta en la siguiente secuencia de órdenes para que 'Pac-Man' avance por el camino señalado comiendo el número de fresas indicadas (número desconocido)?

```

mientras haya camino delante
hacer avanzar
si hay alguna fresa
hacer
hacer Comer 1 fresa
                    
```

|  |
|--|
| <p>Opción A</p> <p>Mientras haya camino delante</p>    |
| <p>Opción B</p> <p>Mientras no haya camino delante</p> |
| <p>Opción C</p> <p>Mientras haya alguna fresa</p>      |
| <p>Opción D</p> <p>Mientras no haya ninguna fresa</p>  |

|   |   |  |
|---|---|--|
| <p>Si tenemos el siguiente conjunto de órdenes, al que llamamos 'my function', y que dibuja un cuadrado de 100 píxeles de lado:</p> <pre> Función my function   repetir 4 veces   haz mover hacia adelante 100 píxeles   girar a la derecha 90 grados </pre> <p>¿Qué secuencia debe ejecutar el artista para dibujar el siguiente diseño? Cada uno de los lados de cada cuadrado mide 100 píxeles.</p>   | <p>Opción A</p> <pre> repetir 3 veces   haz my function   girar a la derecha 90 grados </pre> | <p>Opción B</p> <pre> repetir 3 veces   haz my function   girar a la derecha 90 grados </pre>                      |
|   | <p>Opción C</p> <pre> repetir 4 veces   haz my function   girar a la derecha 90 grados </pre> | <p>Opción D</p> <pre> repetir 4 veces   haz my function   girar a la derecha 90 grados </pre>                      |
| <p>Si tenemos el siguiente conjunto de órdenes, al que llamamos 'my function', y que dibuja un triángulo de 50 píxeles de lado:</p> <pre> Función my function   repetir 3 veces   haz mover hacia adelante 50 píxeles   girar a la izquierda 120 grados </pre> <p>¿Qué le falta a la siguiente secuencia para que el artista dibuje el siguiente diseño? Cada uno de los lados de cada triángulo mide 50 píxeles.</p> <pre> repetir ??? veces   haz my function   saltar hacia adelante 50 píxeles </pre>  | <p>Opción A</p> <p>15</p>   | <p>Opción B</p> <p>5</p>   |
|   | <p>Opción C</p> <p>4</p>  | <p>Opción D</p> <p>3</p>   |
| <p>Si tenemos el siguiente conjunto de órdenes, al que llamamos 'get 5':</p> <pre> Función get 5   repetir 5 veces   haz Comer 1 fresa </pre> <p>¿Qué órdenes van llevando a 'Pac-Man' por el camino señalado e indicándole que se coma el número de fresas correspondiente?</p>   | <p>Opción A</p> <pre> avanzar girar a la derecha 90 repetir 3 veces   avanzar   get 5 </pre>  | <p>Opción B</p> <pre> avanzar girar a la derecha 90 repetir 3 veces   repetir 3 veces   haz get 5   avanzar </pre> |
|   | <p>Opción C</p> <pre> avanzar girar a la derecha 90 repetir 5 veces   avanzar   get 5 </pre>  | <p>Opción D</p> <pre> avanzar girar a la derecha 90 repetir 5 veces   haz get 5   avanzar </pre>                   |
| <p>Si tenemos el siguiente conjunto de órdenes, llamado 'move and get 4':</p> <pre> Función move and get 4   avanzar   girar a la derecha 90   avanzar   repetir 4 veces   haz Comer 1 fresa   girar a la izquierda 90 </pre> <p>¿Qué falta en la siguiente secuencia para llevar a 'Pac-Man' por el camino señalado hasta las fresas, comiendo el número de fresas indicado?</p> <pre> repetir ??? veces   haz move and get 4 </pre>    | <p>Opción A</p> <p>3</p>  | <p>Opción B</p> <p>4</p>   |
|   | <p>Opción C</p> <p>5</p>  | <p>Opción D</p> <p>6</p>   |