

**Universidad Católica de Santa María**  
**Escuela de Postgrado**  
**Maestría en Educación con Mención en Gestión de los Entornos**  
**Virtuales para el Aprendizaje**



**Uso de la plataforma educativa Code.org en la mejora del desarrollo del  
pensamiento computacional en estudiantes del primer grado de educación  
secundaria en Arequipa, Perú**

Tesis presentada por el Bachiller:

**Aymara Alfaro, Raul Barcelon**

**ORCID: 0009-0001-3398-374X**

Para optar el Grado Académico de Maestro en Educación con Mención en Gestión  
de los Entornos Virtuales para el Aprendizaje

Asesor:

**Dr. Gutierrez Aguilar, Olger Albino**

**ORCID: 0000-0002-6657-7529**

**Arequipa - Perú**

**2025**

UCSM-ERP

**UNIVERSIDAD CATÓLICA DE SANTA MARÍA**  
**ESCUELA DE POSTGRADO**  
**DICTAMEN APROBACIÓN DE BORRADOR DE TESIS**

Arequipa, 17 de Octubre del 2025

**Dictamen: 014554-C-EPG-2025**

Visto el borrador del expediente 014554, presentado por:

**2017000421 - AYMARA ALFARO RAUL BARCELON**

Titulado:

**USO DE LA PLATAFORMA EDUCATIVA CODE.ORG EN LA MEJORA DEL DESARROLLO DEL  
PENSAMIENTO COMPUTACIONAL EN ESTUDIANTES DEL PRIMER GRADO DE EDUCACIÓN  
SECUNDARIA EN AREQUIPA, PERÚ**

Nuestro dictamen es:

**APROBADO**

**40230355 - JAIME ZAVALA MILENA KETTY  
DICTAMINADOR**



**30830016 - SIU ANTEZANA ROCIO JACKELINE  
DICTAMINADOR**



**04411473 - BELTRAN MOLINA ROSA PATRICIA  
DICTAMINADOR**



# Uso de la plataforma educativa Code.org en la mejora del desarrollo del pensamiento computacional en estudiantes del primer grado de educación secundaria en Arequipa, Perú

## INFORME DE ORIGINALIDAD

4%

INDICE DE SIMILITUD

5%

FUENTES DE INTERNET

6%

PUBLICACIONES

8%

TRABAJOS DEL ESTUDIANTE

## FUENTES PRIMARIAS

1

Submitted to Universidad Católica de Santa María

Trabajo del estudiante

2%

2

repositorio.pedagogica.edu.co

Fuente de Internet

1%

3

tesis.ucsm.edu.pe

Fuente de Internet

1%

Excluir citas

Apagado

Excluir coincidencias < 1%

Excluir bibliografía

Apagado

## DEDICATORIA

A mis padres, a mi esposa e hijos quienes me apoyan para ser mejor cada día.

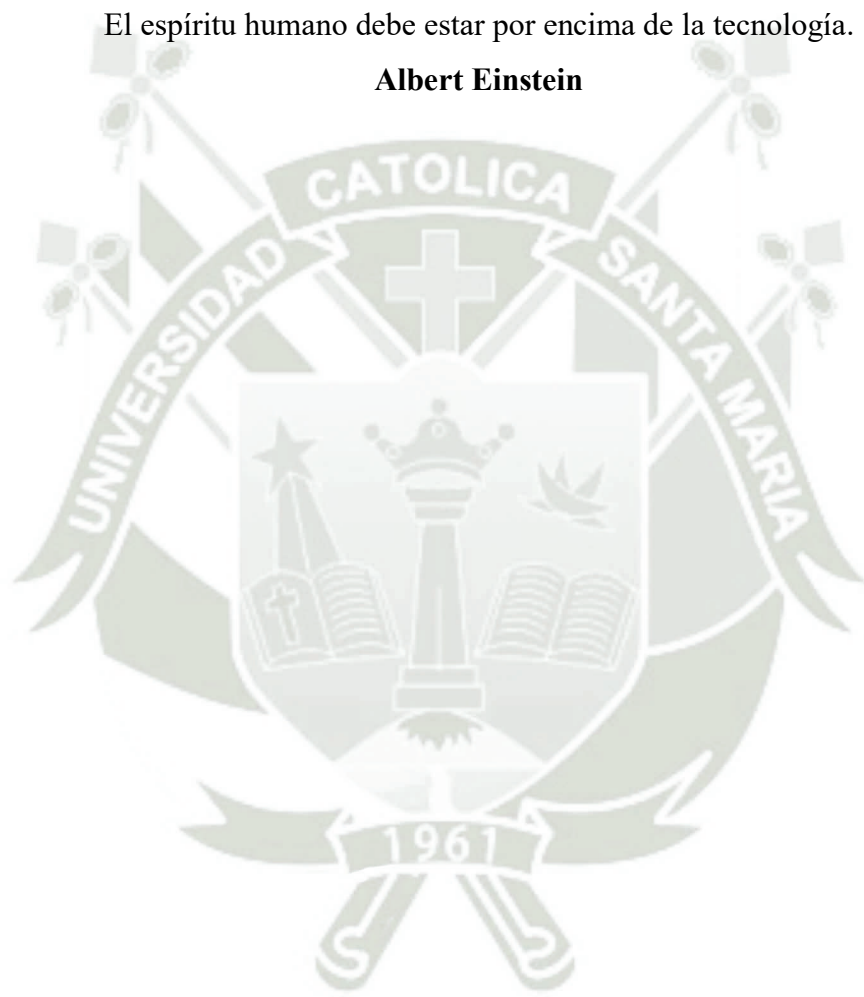
**Raul Barcelon Aymara Alfaro**



## EPÍGRAFE

El espíritu humano debe estar por encima de la tecnología.

**Albert Einstein**



## RESUMEN

El objetivo de este estudio fue analizar el impacto de una estrategia didáctica basada en pensamiento computacional, implementada a través de la plataforma educativa Code.org, en estudiantes de educación básica. Para ello, se empleó un diseño cuasiexperimental con grupo control y grupo experimental, aplicando evaluaciones de entrada y salida para medir el nivel de logro en distintas dimensiones del pensamiento computacional.

Ambos grupos estuvieron conformados por un total de 38 estudiantes. En la evaluación de entrada, los resultados de tipo descriptivo y luego en el análisis inferencial ( $t = -0.683$ ;  $p = .499$ ) confirmaron que no existían diferencias notables entre ambos grupos, lo que garantizó condiciones de partida comparables.

En la evaluación de salida, el grupo experimental mostró una mejora considerable en su desempeño, alcanzando una media de 18.1 frente a 14.1 en el grupo control. La diferencia fue estadísticamente significativa ( $t = 3.3$ ;  $p = .002$ ). El análisis en un mismo grupo también evidenció resultados consistentes: el grupo control no mostró mejoras significativas ( $p = .148$ ), mientras que el grupo experimental presentó una mejora altamente significativa entre el pretest y el posttest ( $t = -7.28$ ;  $p < .001^{***}$ ).

En conclusión, la estrategia implementada con Code.org tuvo un impacto positivo y considerado significativo en el desarrollo del pensamiento computacional, no solo mejorando los niveles de logro en dimensiones básicas y complejas, sino también demostrando su efectividad en comparación con métodos tradicionales. Estos resultados respaldan el uso de plataformas interactivas como recursos pedagógicos eficaces para el fortalecimiento de habilidades digitales en contextos escolares.

**Palabras clave:** Pensamiento computacional, software CODE.ORG, Intervención educativa.

## ABSTRACT

The aim of this study was to analyze the impact of a teaching strategy based on computational thinking, implemented through the educational platform Code.org, on basic education students. A quasi-experimental design with control and experimental groups was used, applying pre- and post-tests to measure achievement levels across different dimensions of computational thinking. Both groups included a total of 38 students. In the pretest, descriptive and inferential analyses ( $t = -0.683$ ;  $p = .499$ ) confirmed that there were no significant differences between the groups, ensuring comparable starting conditions.

In the posttest, the experimental group showed a notable improvement in performance, reaching a mean score of 18.1 compared to 14.1 in the control group. The difference was statistically significant ( $t = 3.3$ ;  $p = .002$ ). Within-group analysis also showed consistent results: the control group did not present meaningful progress ( $p = .148$ ), whereas the experimental group achieved a highly significant improvement between pretest and posttest ( $t = -7.28$ ;  $p < .001^{***}$ ).

In conclusion, the Code.org-based instructional strategy had a positive and meaningful effect on the development of computational thinking. It not only improved achievement levels in both basic and complex dimensions but also demonstrated its superiority over traditional teaching methods. These findings support the use of interactive digital platforms as effective pedagogical tools for strengthening digital skills in school settings.

Keywords: computational thinking, Code.org, didactic intervention.

## ÍNDICE GENERAL

DEDICATORIA	
EPÍGRAFE	
RESUMEN	
ABSTRACT	
INTRODUCCIÓN .....	1
HIPÓTESIS .....	3
OBJETIVOS.....	4
CAPÍTULO I MARCO TEÓRICO.....	5
1.1. PENSAMIENTO COMPUTACIONAL .....	5
1.1.1. Conceptos Computacionales.....	6
1.1.2. Entorno-Interfaz del Ítem.....	6
1.1.3. Estilo de Alternancia de la Respuesta.....	7
1.1.4. Estructuras Anidadas .....	7
1.1.5. Tarea Requerida.....	7
1.2. ASPECTOS PSICOPEDAGÓGICOS DE LA PLATAFORMA CODE.ORG .....	7
1.2.1. Elementos psicopedagógicos de la plataforma Code.Org .....	8
1.2.2. Interés.....	9
1.2.3. Habilidades .....	9
1.2.4. Técnicas .....	9
1.2.5. Creatividad.....	9
1.3 ASPECTOS TÉCNICOS DE LA PLATAFORMA CODE.ORG.....	10
1.3.1. Sesiones Interactivas .....	10
1.3.2. Trabajo en Línea .....	10
1.3.3. Trabajo Fuera de Línea .....	10
1.3.4. Herramientas Digitales .....	10
1.3.5. Foros y Chats .....	10
1.3.6. Wikis / Trabajo Colaborativo .....	11
1.4. ANÁLISIS DE ANTECEDENTES INVESTIGATIVOS. ....	11
1.4.1. Internacional .....	11
1.4.2. Nacional.....	12
1.4.3. Local .....	13

CAPÍTULO II METODOLOGÍA .....	15
2.1. TIPO Y NIVEL DE INVESTIGACIÓN .....	15
2.1.1. Técnicas e instrumentos.....	16
2.1.2. Campo de Verificación .....	18
2.1.3. Estrategia de recolección de datos .....	19
CAPÍTULO III RESULTADOS Y DISCUSIÓN .....	20
3.1. ANÁLISIS DE DATOS: ESTADÍSTICA DESCRIPTIVA .....	20
3.1.1. Evaluación de entrada GC y GE .....	21
3.1.2. Evaluación de salida GC y GE .....	22
3.1.3. Comparativa evaluación de entrada y de salida grupo Control .....	23
3.1.4. Comparativa evaluación de entrada y de salida grupo Experimental .....	24
3.1.5. Prueba de entrada por Dimensiones del Pensamiento Computacional.....	25
3.1.6. Prueba de salida por Dimensiones del Pensamiento Computacional .....	27
3.1.7. Análisis de la prueba de entrada y de salida por Dimensiones en GE.....	29
3.2. ANÁLISIS DE DATOS: ESTADÍSTICA INFERENCIAL .....	30
3.2.1. Evaluación de entrada (grupo control y experimental).....	31
3.2.2. Evaluación de salida (grupo control y experimental) .....	32
3.2.3. Resultados relacionados pre y post test del grupo control.....	34
3.2.4. Resultados relacionados pre y post test del grupo experimental .....	35
3.3. DISCUSIÓN .....	37
CONCLUSIONES .....	40
RECOMENDACIONES .....	42
REFERENCIAS .....	43

## ÍNDICE DE TABLAS

Tabla 1 Cuadro de Coherencia de las variables .....	16
Tabla 2 Baremo de correspondencia .....	18
Tabla 3 Evaluación de entrada para muestras independientes GC y GE .....	21
Tabla 4 Evaluación de salida para muestras independientes GC y GE.....	22
Tabla 5 Comparativa en evaluación de Entrada y salida GC.....	23
Tabla 6 Comparativa en evaluación de entrada y salida GE.....	24
Tabla 7 Prueba de entrada: Dimensiones del Pensamiento Computacional .....	25
Tabla 8 Prueba de salida: Dimensiones del Pensamiento Computacional.....	27
Tabla 9 Pruebas de normalidad .....	30
Tabla 10 Datos descriptivos de grupos en Pre Test.....	31
Tabla 11 Contraste de hipótesis de la evaluación de entrada GC vs.GE.....	31
Tabla 12 Datos descriptivos de grupos en Post Test.....	32
Tabla 13 Contraste de hipótesis de la evaluación de salida GC vs. GE.....	33
Tabla 14 Datos Descriptivos para Muestras Relacionadas de GC .....	34
Tabla 15 Prueba T para Muestras Relacionadas GC.....	35
Tabla 16 Datos Descriptivos para Muestras Relacionadas de GE .....	35
Tabla 17 Prueba T para Muestras Relacionadas GE .....	36

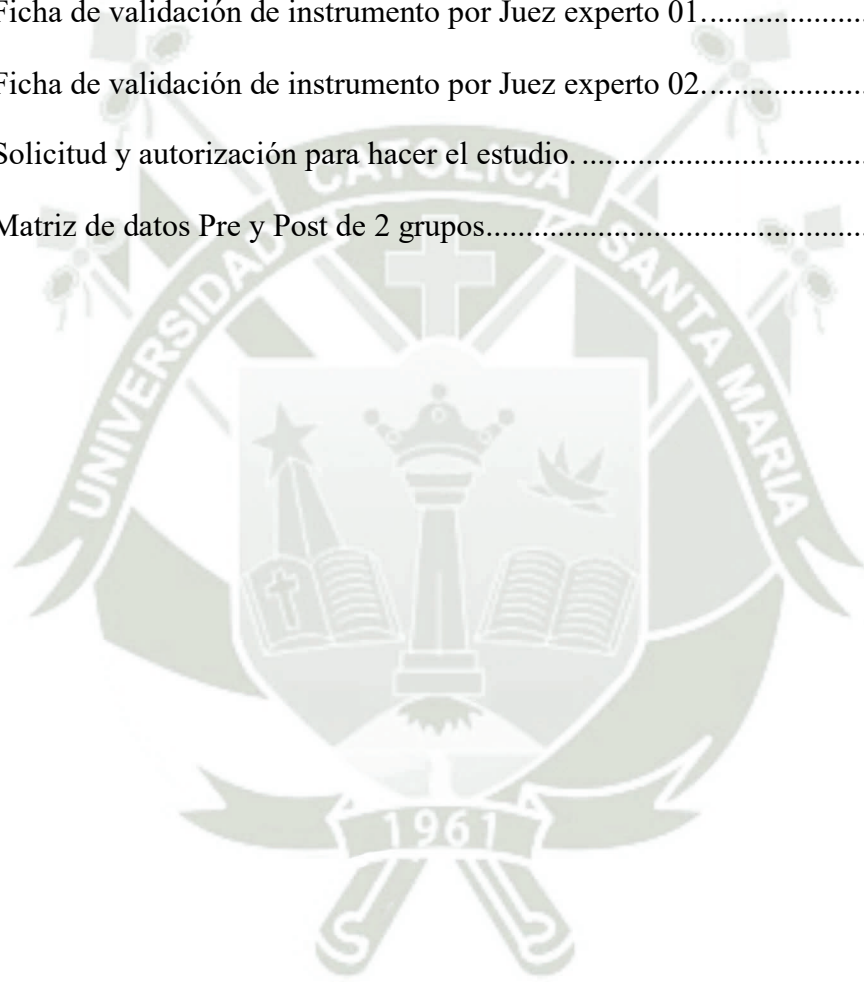
## ÍNDICE DE FIGURAS

Figura 1 Evaluación de entrada para muestras independientes GC y GE .....	21
Figura 2 Evaluación de salida para muestras independientes GC y GE .....	22
Figura 3 Comparativa en evaluación de entrada y salida GC. ....	23
Figura 4 Comparativa en evaluación de entrada y salida GE.....	24
Figura 5 Prueba de entrada: Dimensiones del Pensamiento Computacional .....	25
Figura 6 Prueba de salida: Dimensiones del Pensamiento Computacional .....	27
Figura 7 Prueba de entrada y de salida por Dimensiones en GE .....	29



## ÍNDICE DE ANEXOS

Anexo 1 Matriz de Consistencia .....	46
Anexo 2 Sesiones de aprendizaje. Propuesta que integra el uso de Code.org .....	47
Anexo 3 Instrumento de Evaluación Pre y Post-Test.....	79
Anexo 4 Ficha de validación de instrumento por Juez experto 01.....	86
Anexo 5 Ficha de validación de instrumento por Juez experto 02.....	89
Anexo 6 Solicitud y autorización para hacer el estudio.....	92
Anexo 7 Matriz de datos Pre y Post de 2 grupos.....	94



## INTRODUCCIÓN

La presente investigación parte del interés por comprender cómo La utilización de la plataforma educativa Code.org ha demostrado favorecer el desarrollo de las habilidades de pensamiento computacional en los estudiantes del primer grado de secundaria, promoviendo el razonamiento lógico, la resolución de problemas y la creatividad en sus aprendizajes. A partir de esta inquietud, se formuló como propósito central analizar el efecto de la aplicación de Code.org en la mejora de las habilidades propias del pensamiento computacional, a través de una comparación entre un grupo de estudiantes que trabajó con dicha plataforma y otro que siguió estrategias tradicionales de enseñanza.

El estudio se circunscribe dentro del campo de la Tecnología y la Educación, con un enfoque en la incorporación de recursos digitales al proceso de enseñanza-aprendizaje. Se enmarca en la línea de investigación sobre innovación educativa y recursos digitales para el aprendizaje, con la finalidad de aportar estrategias que fortalezcan la formación de competencias digitales en los estudiantes, un aspecto esencial en la actual era de transformación tecnológica y educativa.

La variable dependiente del estudio fue el pensamiento computacional, entendido como la capacidad de los estudiantes para plantear y resolver problemas aplicando procesos lógicos, estructurados y secuenciales. Esta variable se evaluó a partir de varias dimensiones: conceptos computacionales, interacción con el entorno o la interfaz, estilo de respuesta y tipo de tarea requerida, permitiendo una comprensión más integral de cómo los alumnos aplican el razonamiento propio del pensamiento computacional en distintas situaciones de aprendizaje. Para su medición se utilizó el Test de Pensamiento Computacional adaptado por Román-González (2016), instrumento validado en contextos educativos que aseguró la confiabilidad de los resultados.

La investigación adoptó un diseño cuasiexperimental con pretest y postest, con grupos no equivalentes (control y experimental). Este enfoque permitió observar las diferencias entre ambos grupos antes y después de la intervención pedagógica basada en Code.org, la cual se implementó en diez sesiones con actividades de programación visual y ejercicios de resolución de problemas. Los resultados comparativos obtenidos a través de las mediciones iniciales y finales permitieron identificar cambios atribuibles al uso de la plataforma en el desarrollo del pensamiento computacional.

La importancia del estudio radica en que Code.org se consolida como una herramienta didáctica innovadora y de fácil acceso, capaz de potenciar las competencias digitales y cognitivas que demanda la educación del siglo XXI. Sus resultados ofrecen aportes valiosos para docentes, especialistas en innovación educativa y diseñadores curriculares que buscan integrar la programación y el pensamiento lógico en el ámbito de la educación básica. No obstante, el estudio presenta algunas limitaciones. Parte de los estudiantes del grupo experimental contaba con poca experiencia previa en el uso de la plataforma, lo que hizo necesaria una fase breve de familiarización. Asimismo, factores externos, como las diferencias en el acceso a dispositivos o el nivel de apoyo familiar, pudieron incidir parcialmente en los resultados obtenidos. Por ello, los hallazgos deben interpretarse con cautela, considerando tanto el tamaño reducido de la muestra como las características particulares del contexto educativo en el que se desarrolló la investigación.

En cuanto a los alcances, esta investigación aporta evidencia empírica sobre el valor pedagógico de Code.org como medio para estimular el pensamiento computacional y las habilidades de razonamiento lógico en adolescentes. Los resultados obtenidos pueden servir de base para el diseño de estrategias didácticas innovadoras, acordes con las demandas de la educación digital contemporánea.

El documento se estructura en tres capítulos:

El primer capítulo aborda el marco teórico y los antecedentes relacionados con el pensamiento computacional y la enseñanza mediante plataformas digitales.

El segundo capítulo describe el diseño metodológico, los instrumentos y la población participante.

Finalmente, el tercer capítulo presenta el análisis de los resultados, la discusión, las conclusiones y las recomendaciones derivadas del estudio.

## HIPÓTESIS

Dado que, las plataformas educativas de programación ofrecen recursos didácticos accesibles basados en actividades interactivas, juegos y retos que favorecen la resolución de problemas, la lógica y la programación, condiciones que respaldan su pertinencia para enseñar pensamiento computacional en secundaria. Desde la perspectiva de Wing, dicho pensamiento implica formular y abordar problemas mediante principios computacionales, por lo que entornos guiados y visuales como Code.org deberían potenciar su desarrollo al proporcionar experiencias estructuradas que promueven la práctica progresiva de habilidades cognitivas esenciales para los estudiantes de secundaria. Por lo que:

### *Hipótesis General*

H<sup>1</sup> El uso de la plataforma educativa Code.org mejora significativamente el desarrollo del pensamiento computacional en estudiantes del primer grado de educación secundaria de la I.E. Gran Pachacutec, Arequipa, evidenciándose una diferencia significativa entre el grupo experimental (que utiliza la plataforma) y el grupo control (que no la utiliza).

### *Hipótesis Específicas*

H<sup>2</sup> Los niveles iniciales de pensamiento computacional, medidos mediante una prueba pretest, no presentan diferencias significativas entre el grupo experimental y el grupo control del primer grado de la I.E. Gran Pachacutec, Arequipa.

H<sup>3</sup> Existen diferencias significativas en los niveles de pensamiento computacional entre el grupo experimental y el grupo control después de la intervención con Code.org, mostrando el grupo experimental puntuaciones más altas en la evaluación postest.

H<sup>4</sup> La implementación de una estrategia didáctica basada en pensamiento computacional mediante el uso de la plataforma Code.org genera una mejora significativa en los niveles de logro del grupo experimental en las distintas dimensiones del pensamiento computacional.

## OBJETIVOS

### *General*

Evaluar el efecto del uso de la plataforma educativa Code.org en la mejora del desarrollo del pensamiento computacional en estudiantes del primer grado de educación secundaria de la I.E. Gran Pachacutec, Arequipa, mediante la comparación de un grupo experimental y un grupo control.

### *Específicos*

Establecer, mediante una prueba pretest, la línea base del nivel de pensamiento computacional en los estudiantes de los grupos control y experimental del primer grado de la I.E. Gran Pachacutec, Arequipa.

Comparar, mediante una evaluación postest, las diferencias significativas en los niveles de pensamiento computacional entre el grupo experimental que utiliza Code.org y el grupo control que no lo utiliza.

Analizar el efecto de una estrategia didáctica basada en pensamiento computacional en el grupo experimental, comparando los niveles de logro por dimensiones antes y después de la intervención con Code.org.

## CAPÍTULO I

### MARCO TEÓRICO

#### 1.1. PENSAMIENTO COMPUTACIONAL

El pensamiento computacional puede desarrollarse incluso sin el uso de computadoras, utilizando únicamente papel y lápiz para resolver problemas de forma lógica y estructurada. No obstante, el uso de dispositivos tecnológicos facilita la exploración de situaciones más complejas y el abordaje de diversos tipos de desafíos. Es importante destacar que pensar computacionalmente no equivale a saber programar, ya que implica razonar en distintos niveles de abstracción, analizar patrones y diseñar soluciones, procesos que pueden llevarse a cabo independientemente del uso de herramientas digitales (Valverde-Berrocoso et al., 2015).

El procesamiento que implica la computación es una forma de pensar que ayuda a entender y resolver problemas complicados dividiéndolos en partes más simples. También permite reconocer patrones, enfocarse en lo más importante y encontrar soluciones paso a paso, como si se siguiera una receta. No se trata solo de programar, sino que se puede usar en muchas áreas, como la ciencia o incluso las humanidades. Lo valioso de esta manera de pensar es que permite organizar los problemas y sus soluciones de forma clara, para que tanto las personas como las computadoras los puedan comprender y aplicar (Wing, 2006).

Además, se dice que el pensamiento computacional también incluye saber usar la tecnología y crear soluciones usando simulaciones o análisis de datos. Esta forma de pensar une la creatividad de las personas con la capacidad de las máquinas, por eso se considera una habilidad clave para enfrentar los retos actuales en distintas áreas. Enseñar estas habilidades desde la educación básica ayuda a los estudiantes no solo a resolver problemas técnicos, sino también a pensar con lógica y a desarrollar un pensamiento crítico y el razonamiento lógico que les sirve en muchas asignaturas (Selby & Woollard, 2013).

Aprender a resolver problemas de matemáticas no siempre es fácil, tanto para los profesores como para los estudiantes. Sin embargo, algunos estudios han mostrado que el pensamiento computacional y ciertas estrategias bien diseñadas ayudan bastante a que los estudiantes puedan mejorar en la resolución y la adquisición de habilidades relacionadas con la resolución de problemas aritméticos (Peñaloza-Ochoa & Ortega-Chasi, 2023).

### ***1.1.1. Conceptos Computacionales***

El pensamiento computacional involucra el uso de conceptos fundamentales que permiten resolver problemas de manera eficiente, tales como:

Utiliza la direccionalidad: Implica la capacidad de controlar el flujo de instrucciones en un programa usando direcciones como "arriba", "abajo", "izquierda" o "derecha" para mover objetos o realizar acciones específicas (Grover & Pea, 2013).

Uso del bucle “repetir veces”: Este tipo de instrucción permite ejecutar una misma acción o conjunto de acciones un número específico de veces, facilitando la automatización de tareas repetitivas.

Uso del bucle “repetir hasta”: En este caso, las instrucciones se repiten de manera continua hasta que se cumple una condición establecida, lo que permite un control más dinámico del proceso (Barr & Stephenson, 2011).

Condicional simple (if): Este recurso posibilita que el programa tome decisiones. Si una condición se cumple, se ejecuta una determinada acción o bloque de código (Grover & Pea, 2013).

Condicional compuesto: A diferencia del anterior, este tipo de estructura incluye más de una condición, lo que permite evaluar distintas posibilidades y seguir diferentes caminos según los resultados (Yadav et al., 2016).

Utiliza mientras que: Es un bucle que ejecuta instrucciones mientras se cumpla una condición específica (Barr & Stephenson, 2011).

Funciones simples: Permiten la creación de bloques de código que se ejecutan al ser llamados (Grover & Pea, 2013).

Funciones con parámetros: Funciones que reciben valores de entrada (parámetros) para generar resultados específicos (Yadav et al., 2016).

### ***1.1.2. Entorno-Interfaz del Ítem***

El entorno-interfaz del ítem se refiere a los elementos visuales y las herramientas que los estudiantes utilizan para interactuar con la plataforma de programación. Los entornos más comunes son:

El laberinto: Una interfaz en la que los estudiantes deben encontrar un camino a través de obstáculos, comúnmente utilizada en plataformas de aprendizaje de programación (Yadav et al., 2016).

El lienzo: El área de trabajo donde los estudiantes pueden dibujar, mover objetos o interactuar con el entorno digital de aprendizaje (Grover & Pea, 2013).

### ***1.1.3. Estilo de Alternancia de la Respuesta***

Este concepto se refiere a las diferentes maneras en que los estudiantes pueden proporcionar respuestas dentro de un entorno computacional:

Visual por flechas: Este estilo se caracteriza por el uso de flechas para indicar direcciones y movimientos dentro de un programa (Barr & Stephenson, 2011).

Visual por bloques: Se utiliza un enfoque visual donde los bloques representan instrucciones de programación, facilitando la comprensión de los conceptos (Yadav et al., 2016).

Textual: El estilo textual usa el código tradicional para indicar instrucciones y procesos dentro de un programa (Grover & Pea, 2013).

### ***1.1.4. Estructuras Anidadas***

Las estructuras anidadas hacen referencia a la inclusión de una estructura de control dentro de otra, lo que permite la creación de programas más complejos:

Sin existencia: Hace referencia a la ausencia de estructuras de control anidadas, donde las instrucciones se ejecutan de manera secuencial (Yadav et al., 2016).

Con existencia: Implica la presencia de estructuras anidadas, como bucles dentro de otros bucles o condicionales dentro de otros condicionales (Barr & Stephenson, 2011).

### ***1.1.5. Tarea Requerida***

Las tareas requeridas hacen referencia a las habilidades que los estudiantes deben demostrar al completar una tarea de programación:

Secuenciación: Se refiere a la capacidad de ordenar correctamente las instrucciones para que el programa funcione de manera lógica (Grover & Pea, 2013).

Completamiento: Implica llenar los espacios vacíos o completar partes del código faltante para lograr que el programa funcione correctamente (Yadav et al., 2016).

Depuración: Se refiere al proceso de encontrar y corregir errores dentro del código para asegurar su correcto funcionamiento (Barr & Stephenson, 2011).

## **1.2. ASPECTOS PSICOPEDAGÓGICOS DE LA PLATAFORMA CODE.ORG**

Diversas investigaciones han demostrado que el uso de la plataforma educativa Code.org contribuye significativamente al desarrollo del pensamiento computacional en estudiantes de nivel secundario. Al incorporar actividades de programación en el aula, los estudiantes no solo aprenden conceptos técnicos, sino que también mejoran su capacidad para resolver problemas, razonar de forma lógica y desarrollar la creatividad. Estas habilidades,

fundamentales para la formación integral del alumnado, fortalecen su manera de analizar, planificar y construir soluciones, favoreciendo un aprendizaje más activo y significativo.

Code.org fue creada en Estados Unidos en 2013 y hoy en día es una de las plataformas de enseñanza de programación más grandes del mundo. Ofrece recursos gratuitos en más de 70 idiomas y organiza cada año la "Hora del Código", un evento global que se realiza en más de 70 países al mismo tiempo (Bezos et al., 2013).

Code.Org proporciona un entorno estructurado para que los estudiantes se involucren con la codificación, fomentando una comprensión de los principios computacionales (Barradas et al., 2020).

La plataforma pone un fuerte enfoque en el pensamiento algorítmico, que es muy importante para ayudar a los estudiantes a aprender cómo resolver problemas paso a paso, especialmente en el aprendizaje de programación (Rim, 2017). Las investigaciones indican que los planes de estudio de codificación, como los que ofrece Code.Org, son efectivos para mejorar las habilidades de CT entre los niños y jóvenes que aprenden (Relkin et al., 2021).

El diseño de la plataforma permite una introducción gradual a los conceptos de codificación, haciéndola accesible y atractiva para los estudiantes de secundaria (Mihm, 2021). El uso de Code.org tiene un impacto positivo en los estudiantes, ya que al aprender codificación se sienten más seguros para enfrentar problemas difíciles y se animan a pensar de forma más creativa. Esto les da una mayor confianza en sus propias habilidades (Dwivedi et al., 2024).

La naturaleza interactiva de la plataforma promueve la colaboración y la comunicación entre pares, enriqueciendo aún más la experiencia de aprendizaje programa (Rim, 2017). Por otro lado, aunque Code.org ofrece muchos beneficios, algunos docentes opinan que depender demasiado de plataformas digitales puede hacer que se pierdan experiencias de aprendizaje más prácticas y vivenciales, que también son importantes para desarrollar el pensamiento computacional. Por eso, combinar lo digital con métodos más tradicionales podría dar a los estudiantes una formación más completa.

### ***1.2.1. Elementos psicopedagógicos de la plataforma Code.Org***

El entorno digital de Code.org permite a los docentes personalizar la enseñanza según las necesidades individuales de los estudiantes. Además, los retos gamificados de la plataforma mantienen la motivación y reducen la ansiedad asociada con el aprendizaje de temas complejos (Clark et al., 2016; Yadav et al., 2016).

Code.org impulsa un aprendizaje centrado en proyectos y en la resolución de problemas, lo que ayuda a desarrollar el pensamiento crítico y creativo. Esta forma de enseñar es útil para

estudiantes con diferentes capacidades, ya que ofrece un entorno interactivo que hace más fácil entender conceptos abstractos como la programación y el pensamiento computacional (Barr et al., 2011; Grover & Pea, 2013). El diseño gamificado de Code.org mantiene altos niveles de motivación entre los estudiantes. Los desafíos y recompensas que integra la plataforma fomentan el compromiso, disminuyendo la percepción de dificultad asociada al aprendizaje de habilidades tan complejas como la programación (Clark et al., 2016; Stoilescu, 2008)

Code.org ofrece módulos estructurados que guían el aprendizaje progresivo de los estudiantes, comenzando con conceptos básicos y avanzando hacia tareas más complejas. Estos módulos están diseñados para promover la construcción de conocimientos a través de la práctica activa, facilitando la comprensión del pensamiento computacional (Birds, 2021). Por ejemplo:

### ***1.2.2. Interés***

Code.org incluye actividades divertidas y visuales que llaman la atención de los estudiantes, incluso de los que nunca han tenido experiencia con tecnología. Gracias a este enfoque, se despierta su curiosidad y se animan a explorar por sí mismos conceptos relacionados con la computación (Voogt et al., 2013).

### ***1.2.3. Habilidades***

La plataforma desarrolla en el estudiante habilidades críticas como resolución de problemas, trabajo en equipo y manejo de herramientas tecnológicas. Estas habilidades son esenciales no solo para la programación, sino también para la vida diaria y el trabajo en el siglo XXI (Barr & Stephenson, 2011; Voogt et al., 2013).

### ***1.2.4. Técnicas***

Code.org usa técnicas de enseñanza como la retroalimentación inmediata, que permite a los estudiantes darse cuenta de sus errores y corregirlos al instante. También divide las tareas difíciles en pasos más sencillos, lo que hace que aprender sea más fácil y ayuda a los estudiantes a sentirse más capaces y seguros de lo que pueden lograr (Grover & Pea, 2013; Clark et al., 2016).

### ***1.2.5. Creatividad***

Al proponer tareas abiertas y proyectos que se pueden personalizar, Code.org da a los estudiantes la oportunidad de usar su creatividad mientras resuelven problemas del mundo real. Esta libertad los motiva a innovar y a buscar diferentes formas de resolver un mismo reto (Yadav et al., 2016; Margolis et al., 2008).

## 1.3 ASPECTOS TÉCNICOS DE LA PLATAFORMA CODE.ORG

### 1.3.1. Sesiones Interactivas

Las sesiones interactivas de Code.org forman parte importante de su manera de enseñar. Los estudiantes usan la plataforma realizando ejercicios y retos prácticos, lo que les permite aplicar el pensamiento computacional mientras trabajan. Las actividades usan muchos elementos visuales y se ajustan al ritmo de cada estudiante, lo que ayuda a que aprendan por su cuenta y encuentren soluciones de forma creativa (Grover & Pea, 2013; Yadav et al., 2016).

### 1.3.2. Trabajo en Línea

Code.org permite a los estudiantes trabajar en línea desde cualquier lugar y en cualquier momento, siempre que tengan conexión a Internet. Como funciona directamente en la web, los estudiantes pueden entrar a sus lecciones, ver cómo van avanzando y compartir lo que han hecho. También pueden guardar su trabajo en línea, lo que ayuda a que sigan aprendiendo sin interrupciones y puedan colaborar con otros a distancia (Birds, 2021).

### 1.3.3. Trabajo Fuera de Línea

Aunque Code.org se usa sobre todo en línea, también tiene partes que puedes descargar y trabajar sin internet. Esto resulta muy práctico en escuelas o zonas donde la conexión es inestable. Así, los estudiantes bajan los materiales y continúan con sus lecciones y actividades sin necesidad de estar siempre conectados (Birds, 2021).

### 1.3.4. Herramientas Digitales

La plataforma Code.org ofrece varias herramientas digitales, como un editor visual que permite a los estudiantes escribir, probar y corregir sus propios programas de forma sencilla. También incluye funciones para crear proyectos propios, lo que motiva a los estudiantes a ser creativos y a experimentar con ideas relacionadas con la computación (Margolis et al., 2008; Voogt et al., 2013).

### 1.3.5. Foros y Chats

Code.org también ayuda a que los estudiantes y profesores se comuniquen por medio de foros y chats en línea. Estas herramientas les permiten hacer preguntas, compartir ideas y trabajar juntos al mismo tiempo, creando un ambiente de aprendizaje más participativo. Los foros y chats promueven la colaboración y el intercambio, lo cual es importante para que los estudiantes desarrollen habilidades sociales y de comunicación en el mundo digital (Yadav et al., 2016; Clark et al., 2016).

### ***1.3.6. Wikis / Trabajo Colaborativo***

La plataforma facilita el trabajo en equipo mediante el uso de wikis y proyectos colaborativos, donde los estudiantes pueden unir esfuerzos para resolver problemas y crear propuestas conjuntas. Esta dinámica estimula el desarrollo de competencias comunicativas y cooperativas, al tiempo que fortalece la participación activa y el aprendizaje compartido entre compañeros, favoreciendo un entorno educativo más interactivo y constructivo (Grover & Pea, 2013; Barr & Stephenson, 2011).

## **1.4. ANÁLISIS DE ANTECEDENTES INVESTIGATIVOS.**

### ***1.4.1. Internacional***

El estudio realizado por Escalante y Guerra (2024) tuvo como propósito aplicar una estrategia pedagógica basada en el uso de Code.org con el fin de fortalecer el pensamiento computacional en estudiantes de octavo grado de una institución educativa ubicada en San Alberto, Cesar (Colombia).

Esta investigación, de enfoque cualitativo y alcance descriptivo, se apoyó en el modelo ADDIE, que comprende las fases de Análisis, Diseño, Desarrollo, y también la Implementación y Evaluación del proceso y contó con una muestra de 15 estudiantes seleccionados mediante un muestreo no probabilístico.

Los resultados evidenciaron que la estrategia implementada potenció habilidades cognitivas clave, como el análisis lógico, la abstracción y la identificación de patrones, permitiendo que los estudiantes mejoraran su capacidad para resolver problemas a través del razonamiento lógico y crítico. En conclusión, los autores destacaron que Code.org se consolidó como una herramienta pedagógica eficaz, capaz de promover el aprendizaje del pensamiento computacional y ofrecer a los docentes nuevas oportunidades para innovar en sus prácticas educativas (Escalante Coronel & Guerra Olivella, 2024).

Por su parte, Kale et al. (2023) analizaron los procesos de pensamiento presentes en los planes de lecciones de tercer grado de Code.org a través de un enfoque de tipo relacional. Su estudio examinó las habilidades de pensamiento computacional incorporadas en las actividades y su interconexión mediante un análisis de contenido y un análisis de redes sociales. Los hallazgos dan a conocer que las competencias del pensamiento computacional suelen aparecer de manera conjunta y que los materiales promueven también habilidades socioemocionales como la colaboración, la reflexión y la resiliencia. Los autores concluyeron que los recursos pedagógicos de Code.org poseen un gran potencial para integrar habilidades tecnológicas y

blandas, siempre que se optimice el diseño pedagógico de las sesiones con los estudiantes (Kale et al., 2023). Asimismo, otro estudio investigó la unidad del Curso 2 de la plataforma Code.org utilizando el modelo de evaluación centrado en los objetivos de Tyler. El estudio empleó métodos descriptivos y experimentales con la participación de 22 estudiantes de secundaria, quienes completaron un programa de 22 horas. Los resultados mostraron un incremento significativo en las calificaciones del postest respecto al pretest, aunque no se alcanzó un dominio total en codificación. Los estudiantes manifestaron actitudes positivas hacia las actividades y los docentes consideraron que el contenido era moderadamente suficiente. Los autores recomendaron realizar ajustes al curso para mejorar su efectividad y continuar empleándolo como recurso de apoyo en la enseñanza de la programación (Geliş & Kabul, 2022).

Finalmente, Oluk y Çakir (2021) investigaron cual era el efecto de las actividades de Code.org en el pensamiento computacional y las habilidades de desarrollo de algoritmos en estudiantes de educación secundaria. Se utilizó un diseño cuasi experimental con grupos control y experimental, aplicando pruebas antes y después del experimento. Participaron 67 estudiantes de sexto grado, de los cuales 35 formaron parte del grupo experimental y 32 del grupo control. Los resultados mostraron que, al finalizar el programa, los estudiantes que trabajaron con Code.org presentaron mejoras significativas en comparación con el grupo control. Los autores concluyeron que la intervención tuvo un impacto positivo y comprobable en el fortalecimiento de las destrezas para la creación y estructuración de algoritmos, confirmando el valor pedagógico de la plataforma en la enseñanza del pensamiento computacional (Oluk & Çakir, 2021).

#### **1.4.2. Nacional**

El estudio desarrollado por Pimentel (2024), tuvo como finalidad evaluar la eficacia de la plataforma CODE en el fortalecimiento del pensamiento computacional de los estudiantes de cuarto grado de secundaria. La investigación fue de tipo aplicada y se sustentó en un enfoque experimental, utilizando métodos de observación, medición y análisis para examinar los efectos del uso de la plataforma en el aprendizaje de los estudiantes. Participaron 31 estudiantes, divididos en un grupo experimental de 16 y un grupo control de 15, a quienes se aplicó una prueba pedagógica. Los resultados mostraron que el uso de la plataforma CODE fue efectivo para mejorar el pensamiento computacional general ( $p = 0.019$ ), especialmente en las habilidades de análisis ( $p = 0.010$ ) y abstracción ( $p = 0.036$ ), aunque no se observaron mejoras en la habilidad de secuenciación ( $p = 0.868$ ). En conclusión, la plataforma CODE constituye

una herramienta útil para fortalecer el pensamiento computacional en estudiantes de nivel secundario (Pimentel, 2024).

Por su parte, Siu Antezana (2024) desarrolló un estudio con el objetivo de analizar la relación entre las competencias digitales y el pensamiento computacional en estudiantes del VII ciclo de una institución educativa de la ciudad de Arequipa. La investigación siguió un enfoque cuantitativo con diseño descriptivo-correlacional y contó con la participación de 110 estudiantes. Las competencias digitales se midieron mediante un cuestionario estructurado con alta confiabilidad ( $\alpha = 0.946$ ), mientras que el pensamiento computacional se evaluó con otro instrumento de confiabilidad aceptable ( $\alpha = 0.706$ ). Ambos fueron validados por juicio de expertos. El análisis de los datos se hizo con el estadístico no paramétrico de Spearman para determinar la relación entre las variables. Los resultados indicaron un valor de significancia de  $p = 0.074$ , mayor al nivel de 0.05, lo que llevó a aceptar la hipótesis nula y rechazar la alternativa. Por tanto, no se halló una relación estadísticamente significativa entre las competencias digitales y el pensamiento computacional, lo que evidencia la necesidad de fortalecer ambas dimensiones de forma complementaria en el ámbito educativo (Siu Antezana, 2024).

Finalmente, Díaz Sagástegui y Lozano (2019) desarrollaron un estudio sobre el *Uso de las aplicaciones Code.org y Scratch para el aprendizaje de programación en estudiantes de EBR del C.E. N.º 82099 de la provincia de San Pablo, 2018*. El objetivo fue determinar si el uso de ambas plataformas influye en el aprendizaje de la programación. Se adoptó un enfoque mixto, combinando métodos cuantitativos y cualitativos mediante observaciones directas y encuestas aplicadas a los participantes. El estudio fue de tipo aplicado y utilizó post-tests para analizar la relación entre las variables. Los resultados mostraron una correlación muy alta ( $\rho = 0.875$ ,  $p = 0.009$ ) entre el uso de las plataformas y el aprendizaje de la programación. Asimismo, el 85% de los estudiantes calificó como excelente su experiencia con Scratch al crear videojuegos, mientras que el 77% valoró de igual forma los aprendizajes obtenidos en programación. Se concluyó, con un nivel de confianza del 95%, que el uso de Code.org y Scratch tiene una influencia positiva y significativa en el proceso de enseñanza-aprendizaje de la programación (Díaz Sagástegui & Lozano, 2019).

### **1.4.3. Local**

El estudio de Ballón Aguedo (2023), buscó analizar cómo se relacionan el uso de entornos virtuales apoyados en las TIC, el pensamiento computacional y las competencias matemáticas en estudiantes del nivel avanzado de Educación Básica Alternativa (CEBA) de

Arequipa. El estudio de enfoque cuantitativo empleó un diseño no experimental de tipo descriptivo-correlacional, que permitió observar las relaciones entre las variables sin intervenir directamente en ellas.

Para la recolección de datos se utilizaron tres instrumentos validados por expertos: un cuestionario sobre desenvolvimiento en entornos virtuales, elaborado por la autora a partir de los estándares del Currículo Nacional; el test de pensamiento computacional de Marcos Román González; y una prueba diagnóstica de competencias matemáticas del Ministerio de Educación del Perú. Los resultados mostraron que los estudiantes presentaron un buen nivel de desempeño en entornos virtuales, un pensamiento computacional promedio y un dominio sólido de las competencias matemáticas correspondientes a su grado. En conjunto, los hallazgos reflejan un equilibrio entre las habilidades digitales y académicas, evidenciando que los estudiantes poseen las competencias necesarias para afrontar los retos educativos en entornos mediados por tecnología (Ballón Aguedo, 2023).

Por otro lado, Churata Mayta et al. (2021) realizaron el estudio en estudiantes de séptimo grado en Arequipa”, cuyo objetivo fue determinar si el Programa de Codigofabetización contribuye al fortalecimiento de las habilidades de pensamiento computacional en los estudiantes.

La investigación adoptó un diseño cuasiexperimental, con la participación de un grupo control y un grupo experimental, conformados por un total de 51 estudiantes: 26 pertenecientes al grupo experimental y 25 al grupo control, seleccionados mediante un muestreo no probabilístico.

Para la evaluación de los resultados, se aplicó el Test de Pensamiento Computacional de Román (2016), compuesto por 28 ítems y con un índice de confiabilidad de  $\alpha = 0.80$ , lo que garantiza la consistencia de los resultados.

Los resultados fueron evidentes: el grupo experimental pasó de una mediana de 12 en el pretest a 19 en el postest, lo que evidenció una mejora significativa, mientras que el grupo control no mostró variaciones importantes. El análisis estadístico confirmó que el Programa de Codigofabetización tuvo un efecto positivo, sobre todo en habilidades como direcciones, bucles, condicionales y funciones. En conclusión, se comprobó que la implementación del programa fue eficaz para fortalecer el pensamiento computacional de los estudiantes participantes (Churata Mayta et al., 2021).

## CAPÍTULO II METODOLOGÍA

### 2.1. TIPO Y NIVEL DE INVESTIGACIÓN

Este estudio se realizó con un enfoque numérico (cuantitativo) porque buscaba medir qué tanto ayuda la plataforma Code.org a desarrollar el pensamiento computacional en los estudiantes. Para lograrlo, se aplicaron pruebas antes y después de usar la plataforma, con el fin de comparar los resultados y ver si hubo mejoras.

Se usó un diseño cuasiexperimental, lo que significa que se trabajó con dos grupos que ya estaban formados en el aula: uno que usó Code.org (grupo experimental) y otro que no (grupo control). Como los grupos ya existían, no se eligieron al azar. Al comparar los puntajes antes y después de la intervención, se pudo ver el efecto que tuvo la plataforma.

La figura 1 de coherencia de las variables de estudio presenta una estructura de los elementos principales que conforman el diseño metodológico de la investigación, asegurando la correspondencia entre las variables, indicadores, subindicadores, ítems e instrumentos de medición. A continuación, se describen los componentes clave:

Esta variable considera tanto los aspectos psicopedagógicos (como la motivación o la forma en que los estudiantes aprenden) como los aspectos técnicos (cómo usan la plataforma).

Pensamiento computacional:

Se centra en medir las habilidades que los estudiantes tienen para entender y trabajar con ideas y procesos propios de la computación.

Dimensiones del TPC:

Conceptos computacionales: Evalúa si el estudiante comprende ideas clave como bucles, condicionales o secuencias.

Entorno - Interfaz del ítem: Se refiere a la forma en que se presenta el problema, ya sea en forma visual, textual o simbólica.

Estilo de alternancia de la respuesta: Mide cómo el estudiante cambia o elige entre diferentes opciones para resolver un problema.

Estructuras anidadas: Evalúa si puede manejar estructuras dentro de otras, como bucles dentro de bucles.

Tarea requerida: Analiza si entiende lo que se le pide hacer y cómo debe aplicar sus conocimientos para resolverlo.

**Tabla 1**  
*Cuadro de Coherencia de las variables*

Variables	Indicadores	Subindicadores	Ítems	Instrumento
<b>Uso de la plataforma educativa Code.org</b>	Aspectos psicopedagógicos	Módulos de instrucción, motivación, interés, habilidades técnicas y creatividad.	1 y 2	Sesiones de aprendizaje. Propuesta experimental que integra el uso de Code.org.
	Aspectos técnicos	Sesiones interactivas, trabajo en línea, herramientas digitales y trabajo colaborativo.	3–10	
<b>Pensamiento computacional</b>	Conceptos computacionales	Direccionalidad, bucles, condicionales, funciones simples y compuestas.	4 ítems cada uno	Test de pensamiento computacional (Román-González, 2016).
	Entorno–Interfaz del ítem	El laberinto, el lienzo.	5 ítems cada uno	
	Estilo de alternancia de la respuesta	Visual por flechas, visual por bloques.	8 ítems cada uno	
	Estructuras anidadas	Sin existencia, con existencia.	9 y 19 ítems	
	Tarea requerida	Secuenciación, completamiento, depuración.	14, 9 y 5 ítems	

*Nota.* Las 10 Sesiones de aprendizaje son propuestas del investigador y están basadas en el TPC de (Román-González, 2016). Fuente: Elaboración propia.

### **2.1.1. Técnicas e instrumentos.**

La investigación utilizó como técnica principal la observación tanto directa como participante. Esto permitió al investigador ver de cerca cómo los estudiantes usaban la plataforma CODE.ORG, cómo resolvían las tareas y cómo aplicaban los conceptos del pensamiento computacional dentro del aula. Gracias a esta técnica, se pudo registrar con detalle el comportamiento, la participación y el desarrollo de habilidades en un contexto real de aprendizaje.

Para evaluar el impacto del uso de la plataforma Code.org tanto en el pensamiento computacional como en los aspectos psicopedagógicos de los estudiantes, se emplearon los siguientes instrumentos:

### Sesiones de aprendizaje (propuesta del investigador)

El investigador elaboró diez sesiones de aprendizaje diseñadas específicamente para este estudio, las cuales se desarrollaron en el aula y se encuentran detalladas en los anexos. Estas actividades permitieron observar la interacción de los estudiantes con la plataforma, así como analizar variables relacionadas con la motivación, el interés, las habilidades y la creatividad.

Cada sesión se orientó al trabajo de los principales conceptos del pensamiento computacional, junto con las destrezas técnicas necesarias para aprovechar adecuadamente las herramientas que ofrece Code.org.

### Test de Pensamiento Computacional

Este instrumento se aplicó en dos momentos del estudio: al inicio, como evaluación de entrada (pretest), y al finalizar la intervención, como evaluación de salida (postest).

El test tuvo como propósito valorar la comprensión de conceptos básicos del pensamiento computacional, entre ellos la direccionalidad, los bucles, las estructuras condicionales y las funciones. Contenía preguntas de opción múltiple y de respuesta abierta, adaptadas al nivel de avance de los estudiantes durante el proceso formativo.

### Ficha técnica del instrumento

Título: Test de Pensamiento Computacional (Román-González, 2016).

Autor: Marcos Román González

Año: 2016

Institución: Universidad Nacional de Educación a Distancia (UNED), Escuela Internacional de Doctorado, Programa de Doctorado en Educación (España).

Resumen del instrumento. El Test fue diseñado para medir las habilidades de pensamiento computacional en estudiantes de Educación Secundaria Obligatoria, principalmente en los primeros grados. Su validación se llevó a cabo mediante el juicio de 20 expertos y la aplicación en una muestra de 1,251 estudiantes de 24 instituciones educativas.

El análisis psicométrico incluyó evaluaciones de fiabilidad y validez. Entre las características psicométricas; la Fiabilidad, presenta una alta consistencia interna y estabilidad temporal. Su validez fue confirmada a través de distintos procedimientos estadísticos (criterial, discriminante, convergente y factorial), lo que respalda su uso en contextos educativos diversos. Adicionalmente, se validó el instrumento para su adaptación local con 2 expertos especialistas del medio (ver en Anexos).

Para interpretar los datos correctamente, se empleó el siguiente baremo de correspondencia:

**Tabla 2**  
*Baremo de correspondencia*

Letra	Nivel de logro	Rango de puntaje
<b>AD</b>	Destacado	22 a 28
<b>A</b>	Logro previsto	15 a 21
<b>B</b>	En proceso	8 a 14
<b>C</b>	En inicio	1 a 7

*Nota.* Elaboración propia basado en fuentes oficiales (Minedu, 2016).

### 2.1.2. Campo de Verificación

#### Ubicación espacial

Institución Educativa Gran Pachacutec en Arequipa, Perú.

#### Ubicación temporal

Marzo y abril del 2025.

#### Población

La población del estudio estuvo integrada por los estudiantes de primer grado de secundaria de la I. E. Gran Pachacutec, situada en Arequipa, Perú. Este grupo estaba compuesto por alumnos que poseían nociones básicas de informática y que ya estaban familiarizados con el uso de plataformas digitales educativas, lo que facilitó su participación en las actividades del estudio.

#### Criterios de Inclusión

Estudiantes de primer grado de secundaria de la I.E. con acceso a dispositivos electrónicos (computadoras, laptop o teléfonos) para usar la plataforma Code.org.

Participantes voluntarios con autorización de padres o tutores.

Conocimientos básicos de informática y manejo de herramientas digitales educativas.

#### Criterios de Exclusión

Estudiantes fuera del primer grado o sin autorización parental.

Quienes tuvieran experiencia avanzada en programación o uso previo de plataformas similares a Code.org.

#### Muestra

La muestra estuvo conformada por 38 estudiantes seleccionados de forma intencional, según los propósitos del estudio. Los participantes fueron organizados en dos grupos:

Grupo Experimental: fueron 19 estudiantes que usaron la plataforma Code.Org durante las sesiones de aprendizaje.

Grupo Control: 19 estudiantes que no utilizaron la plataforma y siguieron el plan de aprendizaje convencional con plumones y pizarra acrílica.

El muestreo utilizado fue de tipo *por conveniencia*, dado que los participantes fueron seleccionados dentro de la misma institución educativa, tomando en cuenta su disponibilidad, accesibilidad y disposición para participar activamente en las actividades propuestas durante el desarrollo del estudio.

En el grupo experimental (1ro A) participaron 10 mujeres (52.6%) y 9 varones (47.4%), mientras que en el grupo control (1ro B) hubo 11 mujeres (57.9%) y 8 varones (42.1%).

### **2.1.3. Estrategia de recolección de datos**

Evaluaciones inicial y final:

Se aplicó una evaluación de entrada utilizando el Test de Pensamiento Computacional, con el propósito de medir el nivel de habilidades de los estudiantes antes de la intervención. Posteriormente, al terminar el estudio, se realizó una evaluación de salida repitiendo el test, lo que permitió medir y evaluar el progreso en el desarrollo del pensamiento computacional.

Intervención: Durante el proceso, los estudiantes del grupo experimental trabajaron con la plataforma en sesiones de aprendizaje que incluyeron actividades interactivas y tareas autónomas, proporcionando una experiencia de aprendizaje integral. Los recursos utilizados incluyeron:

La plataforma Code.Org como herramienta principal.

El Test de Pensamiento Computacional como instrumento de evaluación y equipos tecnológicos adecuados para facilitar la interacción efectiva de los estudiantes con la plataforma.

Fiabilidad y validación: La confiabilidad de los instrumentos se hizo con un grupo piloto de 15 estudiantes y mediante el cálculo del alfa de Cronbach para el Test arrojó un valor de .749 el cual se considera aceptable.

Para el análisis de datos, se realizó un análisis descriptivo de los resultados obtenidos en las evaluaciones inicial y final. Adicionalmente, se emplearon análisis inferenciales, para evaluar la significancia del impacto de la intervención en los estudiantes. Los resultados permitieron comparar los niveles de pensamiento computacional previo y después de la intervención, asegurando que los datos recogidos fueran confiables y estuvieran alineados con los objetivos del estudio.

## CAPÍTULO III

### RESULTADOS Y DISCUSIÓN

Este capítulo se estructura en tres apartados principales. En el primero se presentan los resultados del análisis descriptivo, donde se muestran los niveles de logro del pensamiento computacional alcanzados por los grupos control y experimental. Los datos se exponen mediante frecuencias y porcentajes, apoyados con tablas y figuras que facilitan su interpretación.

El segundo apartado aborda los resultados del análisis inferencial, empleando la prueba t de Student para la contrastación de hipótesis, tanto en la comparación entre los grupos control y experimental como en la evaluación inicial y final del grupo experimental, con el fin de determinar los cambios producidos tras la intervención.

Finalmente, la tercera parte está dedicada a la discusión de los resultados, donde se contrastan los hallazgos del estudio con investigaciones previas y con los fundamentos teóricos que sustentan el problema de investigación.

#### 3.1. ANÁLISIS DE DATOS: ESTADÍSTICA DESCRIPTIVA

En esta sección se analizan los datos obtenidos en las evaluaciones de entrada y salida de los grupos control y experimental. A partir de un enfoque descriptivo, los resultados se presentan en tablas y gráficos que permiten una comprensión clara de la información recogida. Estos datos representan un aporte fundamental para interpretar y valorar el comportamiento de la variable dependiente dentro del estudio.

### 3.1.1. Evaluación de entrada GC y GE

**Tabla 3**

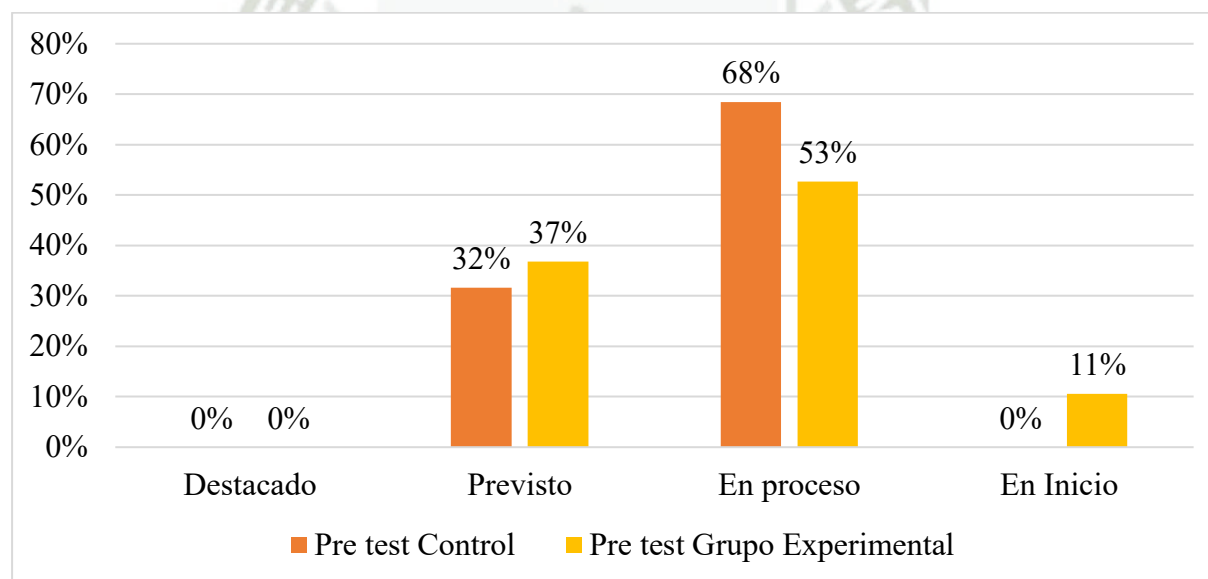
*Evaluación de entrada para muestras independientes GC y GE*

Nivel de logro	Grupo control		Grupo experimental	
	f	%	f	%
Destacado	0	0	0	0
Previsto	6	32	7	37
En proceso	13	68	10	53
En Inicio	0	0	2	11
<b>Total</b>	<b>19</b>	<b>100%</b>	<b>19</b>	<b>100%</b>

*Nota:* Se muestra las diferencias entre los dos grupos. Fuente: elaboración propia (2025).

**Figura 1**

*Evaluación de entrada para muestras independientes GC y GE*



*Nota.* Elaboración propia basada en los resultados obtenidos con Code.Org (2025).

La tabla y figura muestran a ambos grupos eran similares al comienzo: la mayoría se ubicaba en el nivel “En proceso” (68% en el GC y 53% en el GE), lo que indica que contaban con nociones básicas, aunque sin dominio completo. El nivel “Logro previsto” fue mayor en el grupo experimental (37% frente al 32% del GC), lo que evidencia un ligero avance previo. En cambio, el nivel “En inicio” apareció solo en el GE (11%), mostrando una mayor diversidad en ese grupo. Ningún estudiante alcanzó el nivel “Logro destacado”, lo que confirma que ambos grupos partían de niveles básicos o medios, ofreciendo una base equitativa para evaluar luego los efectos de la intervención.

### 3.1.2. Evaluación de salida GC y GE

**Tabla 4**

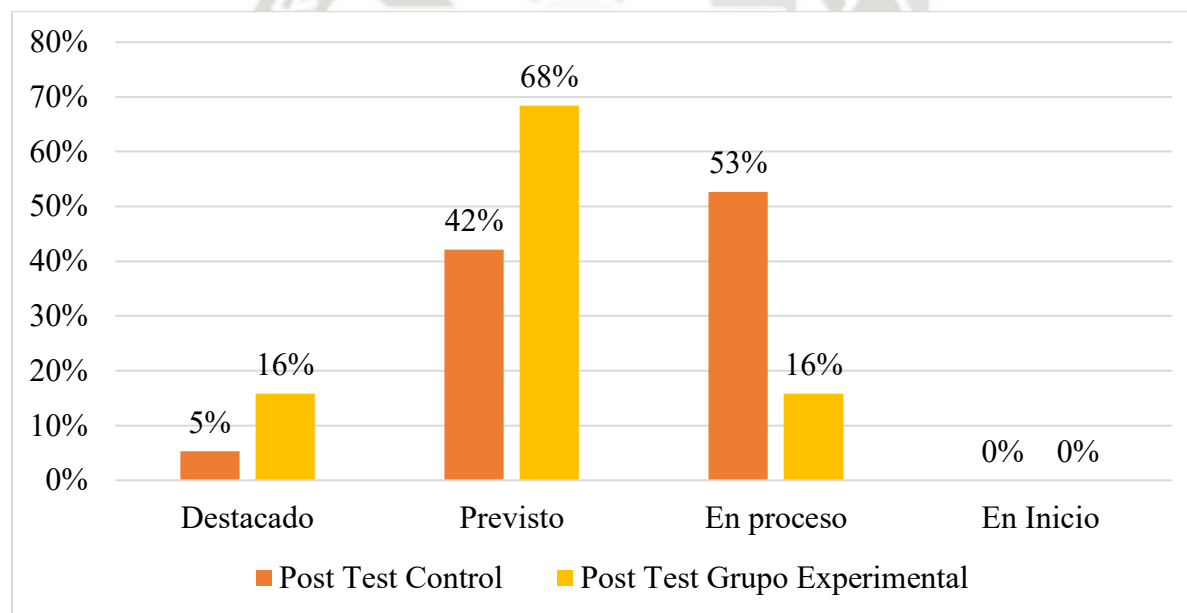
*Evaluación de salida para muestras independientes GC y GE*

Nivel de logro	Grupo control		Grupo experimental	
	f	%	f	%
Destacado	1	5	3	16
Previsto	8	42	13	68
En proceso	10	53	3	16
En Inicio	0	0	0	0
<b>Total</b>	<b>19</b>	<b>100%</b>	<b>19</b>	<b>100%</b>

*Nota:* Esta tabla muestra las diferencias entre los dos grupos de estudio en salida. Fuente: elaboración propia (2025).

**Figura 2**

*Evaluación de salida para muestras independientes GC y GE*



*Nota.* Elaboración propia basada en los resultados obtenidos con Code.Org (2025).

La Tabla 4 evidencia los niveles de logro alcanzados en pensamiento computacional por los grupos control (GC) y experimental (GE) tras la intervención con Code.org. Se observan diferencias notables, con un mayor impacto en el grupo experimental. El 68 % del GE alcanzó el nivel “Previsto”, frente al 42 % del GC, y un 16 % logró el nivel “Destacado”, frente al 5 % del control. El nivel “En proceso” fue menor en el GE (16 %) que en el GC (53 %), y ningún estudiante se ubicó en “En inicio”. Estos resultados muestran un progreso donde el uso pedagógico de Code.org generó mejoras sostenidas en el grupo experimental, mientras el grupo control mantuvo un rendimiento estable.

### 3.1.3. Comparativa evaluación de entrada y de salida grupo Control

**Tabla 5**

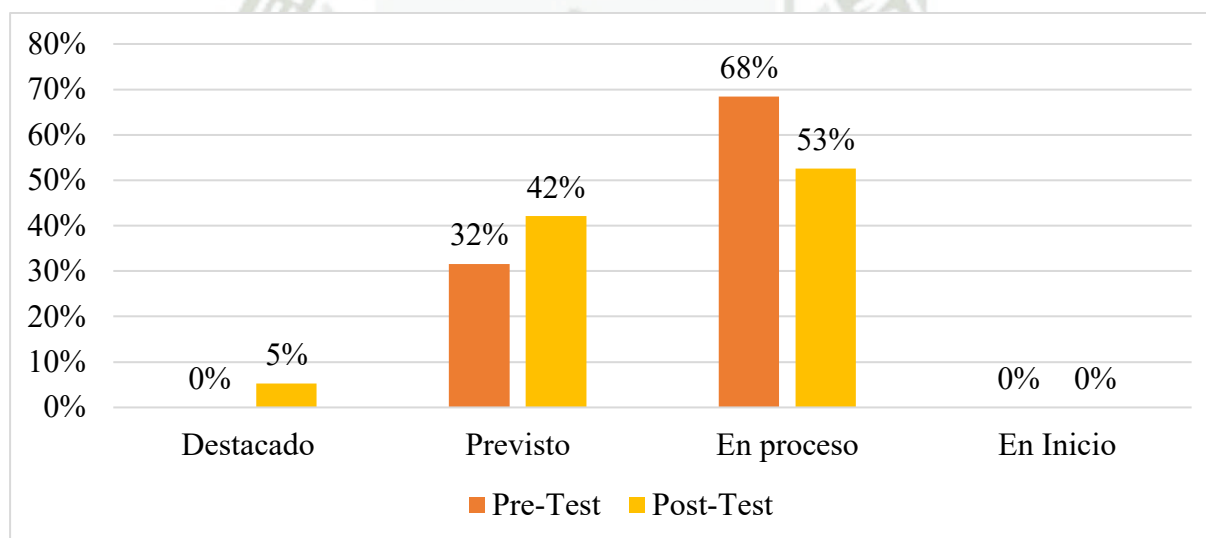
*Comparativa en evaluación de Entrada y salida GC.*

Nivel de logro	Evaluación de entrada		Evaluación de salida	
	f	%	f	%
Destacado	0	0	1	5
Previsto	6	32	8	42
En proceso	13	68	10	53
En Inicio	0	0	0	0
<b>Total</b>	<b>19</b>	<b>100%</b>	<b>19</b>	<b>100%</b>

*Nota:* Diferencias en entrada y salida del grupo Control. Fuente: elaboración propia (2025).

**Figura 3**

*Comparativa en evaluación de entrada y salida GC.*



*Nota.* Elaboración propia basada en los resultados obtenidos con Code.Org (2025).

La Tabla 5 muestra los cambios en los niveles de logro del grupo control entre la evaluación inicial y la final, sin intervención didáctica. Los avances fueron leves: el nivel “Logro previsto” subió del 32 % al 42 %, mientras que “En proceso” bajó del 68 % al 53 %, reflejando una mejora moderada. Solo un estudiante (5 %) alcanzó el nivel “Destacado”, inexistente al inicio, y el nivel “En inicio” se mantuvo en 0 %. En conjunto, los resultados evidencian una mejora mínima y poco consistente, atribuible a la práctica escolar habitual, lo que sugiere que sin una intervención estructurada el pensamiento computacional progresa lentamente.

### 3.1.4. Comparativa evaluación de entrada y de salida grupo Experimental

**Tabla 6**

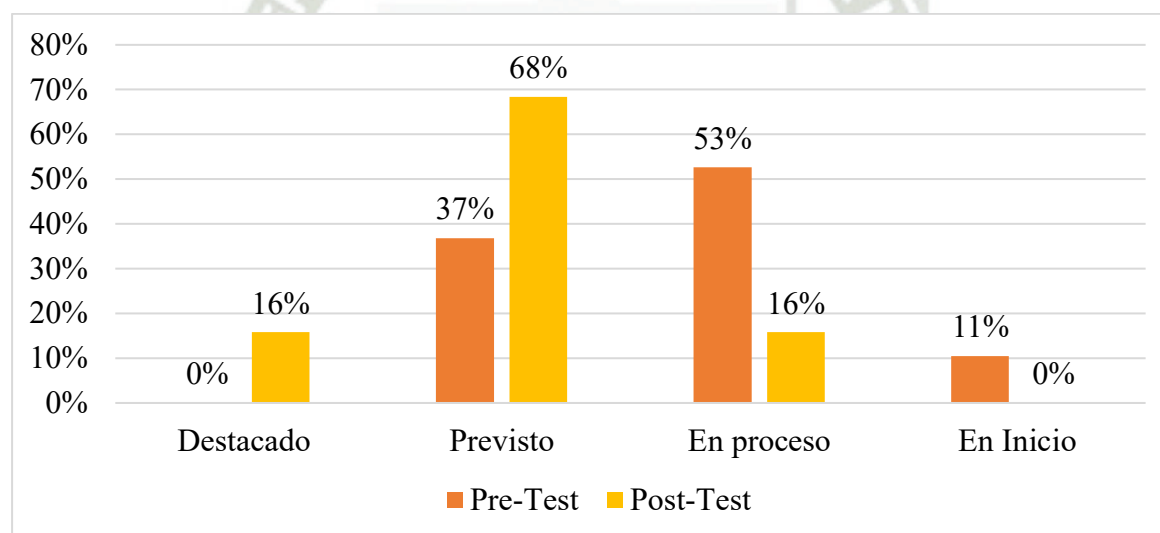
Comparativa en evaluación de entrada y salida GE.

Nivel de logro	Evaluación de entrada		Evaluación de salida	
	f	%	f	%
Destacado	0	0	3	16
Previsto	7	37	13	68
En proceso	10	53	3	16
En Inicio	2	11	0	0
<b>Total</b>	<b>19</b>	<b>100%</b>	<b>19</b>	<b>100%</b>

*Nota:* Diferencias en entrada y salida del grupo experimental. Elaboración propia (2025).

**Figura 4**

Comparativa en evaluación de entrada y salida GE.



*Nota.* Elaboración propia basada en los resultados obtenidos con Code.Org (2025).

La Tabla 6 muestra la evolución del grupo experimental antes y después de la intervención con la plataforma Code.org. Al inicio, la mayoría se ubicaba en “En proceso” (53 %) y un 11 % en “En inicio”, reflejando carencias iniciales. Tras la intervención, ningún estudiante permaneció en “En inicio” y “En proceso” bajó al 16 %, evidenciando un avance general. El nivel “Logro previsto” aumentó de 37 % a 68 %, y un 16 % alcanzó “Logro destacado”, inexistente al inicio. Estos resultados confirman una mejora sustancial atribuida al uso pedagógico de Code.org, que fortaleció conocimientos y habilidades, a diferencia del grupo control, cuyo progreso fue mínimo.

### 3.1.5. Prueba de entrada por Dimensiones del Pensamiento Computacional

**Tabla 7**

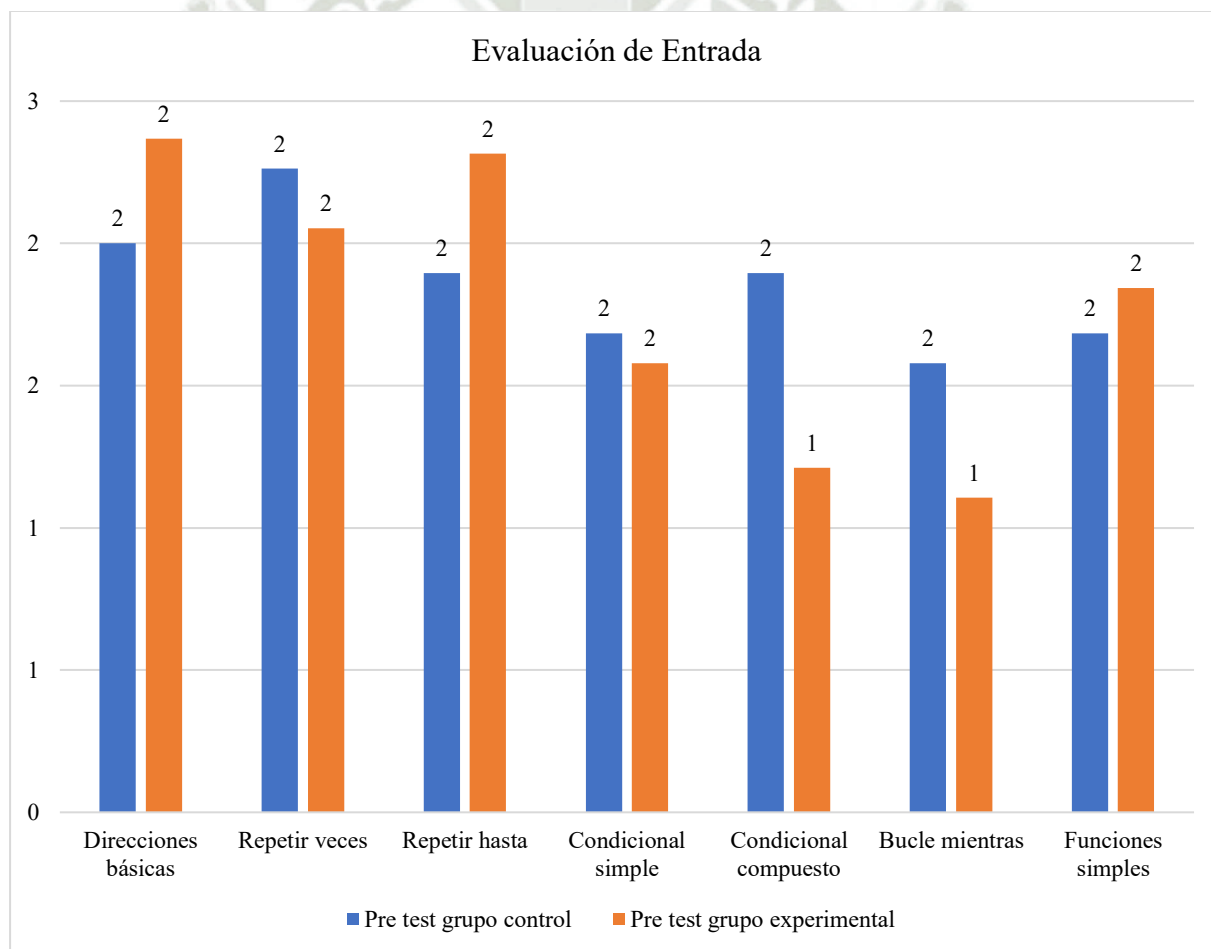
*Prueba de entrada: Dimensiones del Pensamiento Computacional*

<b>Evaluación de Salida</b>	Direcciones básicas	Repetir veces	Repetir hasta	Condicional simple	Condicional compuesto	Bucle mientras	Funciones simples
<b>Grupo control</b>	B	B	B	B	B	B	B
<b>Grupo experimental</b>	B	B	B	B	C	C	B

*Nota.* AD= Destacado; A= Previsto; B= En proceso; C= En inicio. Fuente: Elaboración propia.

**Figura 5**

*Prueba de entrada: Dimensiones del Pensamiento Computacional*



*Nota.* Elaboración propia basada en los resultados obtenidos con Code.Org (2025).

La evaluación comparativa inicial entre el grupo control y el grupo experimental permite establecer una línea base clara para ambos antes de aplicar la intervención. Los resultados

evidencian que, en general, los dos grupos presentan niveles parecidos en la mayoría de las dimensiones del pensamiento computacional, ubicándose principalmente en el nivel “en proceso” (nivel 2). Esto indica que los estudiantes ya contaban con conocimientos básicos sobre los conceptos evaluados, aunque aún no los habían consolidado completamente.

En dimensiones como “Direcciones básicas”, “Repetir veces”, “Condicional simple”, “Condicional compuesto” y “Funciones simples”, ambos grupos alcanzaron puntajes iguales, lo que refleja una situación inicial homogénea respecto al dominio de estas habilidades. Esta similitud resulta clave para asegurar que las condiciones de partida eran comparables y poder analizar con mayor claridad el impacto posterior de la intervención educativa.

Sin embargo, también se identifican algunas diferencias puntuales entre los grupos. Por ejemplo, en la dimensión “Repetir hasta”, el grupo experimental logró ubicarse en el nivel “logro previsto” (nivel 3), mientras que el grupo control permaneció en “En proceso” (nivel 2), lo cual podría indicar que los estudiantes del grupo experimental tenían mayor experiencia previa con estructuras repetitivas. En cambio, en las dimensiones “Condicional compuesto” y “Bucle mientras”, el grupo experimental obtuvo un nivel más bajo (“en inicio”, nivel 1), a diferencia del grupo control, que se mantuvo en el nivel “en proceso” (nivel 2). Esto sugiere que el grupo experimental presentaba más dificultades al inicio en la comprensión de estructuras condicionales y de iteración más complejas.

En general, estos hallazgos permiten concluir que, aunque ambos grupos pueden considerarse similares en términos generales, existen pequeñas diferencias que deben tenerse en cuenta al momento de analizar el impacto de la intervención. En particular, el grupo experimental partía con ciertas desventajas en aspectos más avanzados del pensamiento computacional, lo cual podría influir en el nivel de mejora observado tras aplicar estrategias pedagógicas específicas.

### 3.1.6. Prueba de salida por Dimensiones del Pensamiento Computacional

**Tabla 8**

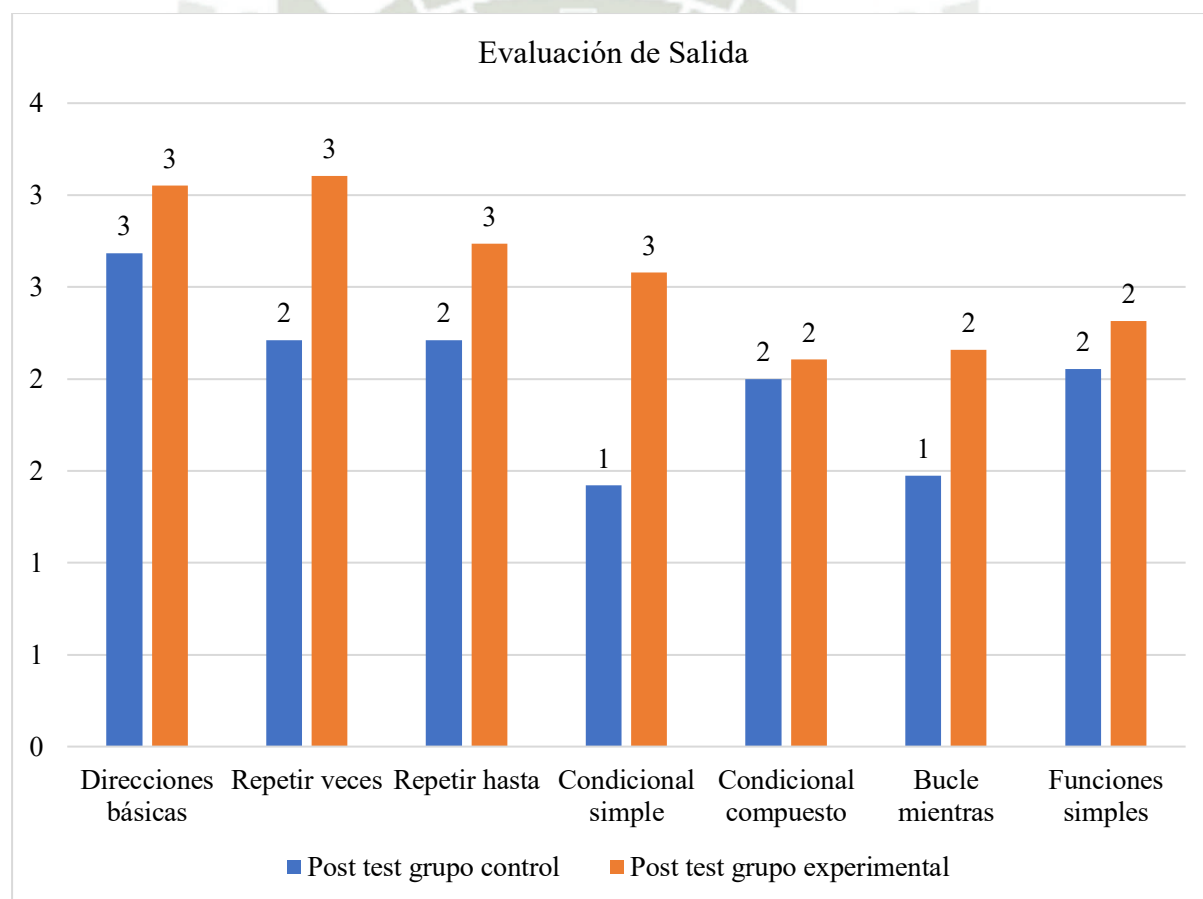
*Prueba de salida: Dimensiones del Pensamiento Computacional*

<b>Evaluación de Salida</b>	Direcciones básicas	Repetir veces	Repetir hasta	Condicional simple	Condicional compuesto	Bucle mientras	Funciones simples
<b>Grupo control</b>	A	B	B	C	B	C	B
<b>Grupo experimental</b>	A	A	A	A	B	B	B

*Nota.* AD= destacado; A= previsto; B= en proceso; C= en inicio. Fuente: elaboración propia.

**Figura 6**

*Prueba de salida: Dimensiones del Pensamiento Computacional*



*Nota.* 4 = destacado; 3= previsto; 2= en proceso; 1= en inicio. Fuente: elaboración propia.

En la evaluación final se advierte una distinción clara entre los niveles de rendimiento alcanzados por el grupo experimental comparado con el grupo control, lo que permite apreciar la efectividad de la propuesta pedagógica aplicada. En términos generales, el grupo experimental evidencia avances notorios en todas las dimensiones del pensamiento

computacional, alcanzando en la mayoría de ellas el nivel de “logro previsto” (3). En cambio, el grupo control se mantiene predominantemente en el nivel “en proceso” (2).

Las dimensiones “Direcciones básicas”, “Repetir veces” y “Repetir hasta” destacan por su buen desempeño dentro del grupo experimental, donde todos los estudiantes alcanzaron el nivel 3. En contraste, el grupo control se ubicó en niveles inferiores, especialmente en las dos últimas dimensiones mencionadas, donde la mayoría no superó el nivel 2. Este resultado sugiere que la intervención permitió fortalecer con éxito las habilidades iniciales y de repetición asociadas al pensamiento computacional.

La diferencia más notoria se presenta en la dimensión “Condicional simple”, en la cual el grupo experimental se posicionó en el nivel 3, mientras que el grupo control descendió al nivel 1. Esto revela no solo un progreso importante en el grupo experimental, sino también una pérdida de desempeño en el grupo control. Un patrón similar se observa en “Bucle mientras”, donde el grupo experimental avanza al nivel 2 y el control se mantiene en el nivel 1, lo que refuerza la idea de que, sin una intervención específica, los estudiantes pueden estancarse o retroceder en el dominio de estructuras lógicas más complejas.

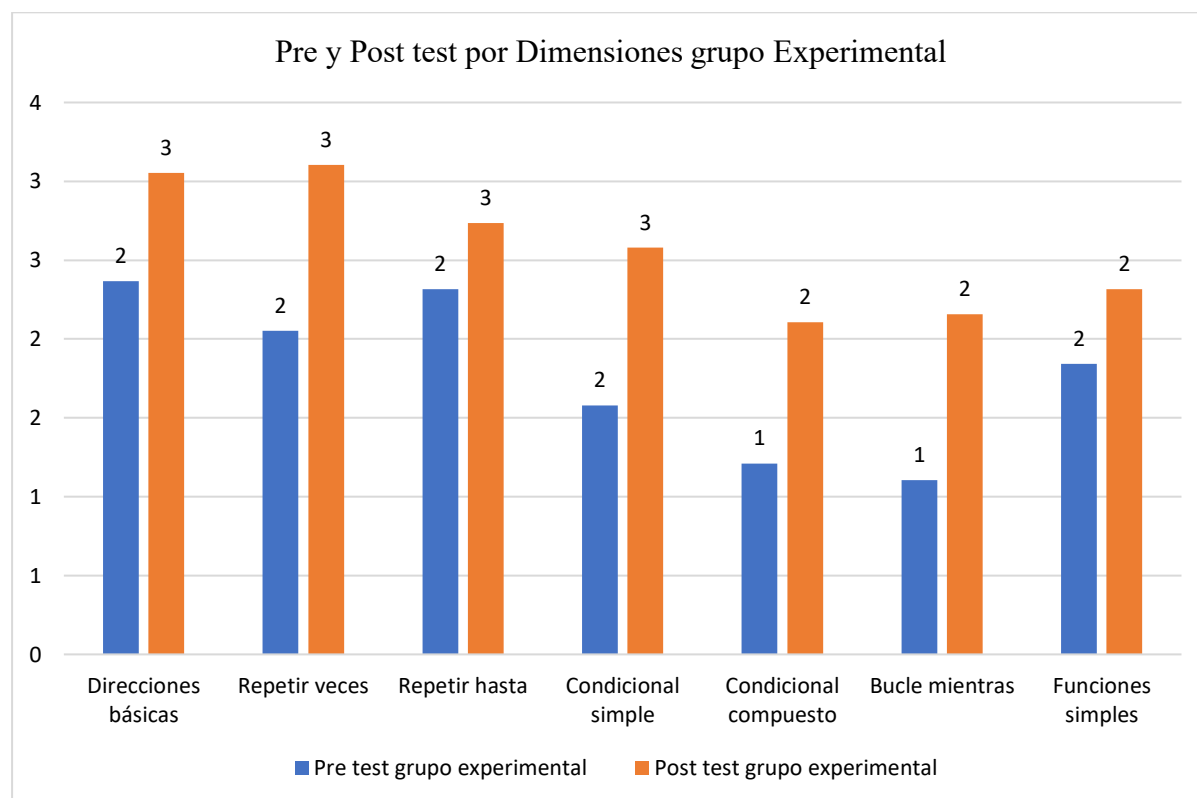
Por otra parte, en las dimensiones “Condicional compuesto” y “Funciones simples”, ambos grupos mantienen resultados similares (nivel 2), lo que podría indicar que estas competencias demandan mayor tiempo de práctica o estrategias didácticas más focalizadas para consolidarse.

En conjunto, los resultados del postest confirman que el grupo experimental alcanzó mejoras sostenidas y consistentes en comparación con el grupo control. Esta evolución permite inferir que la metodología implementada contribuyó de manera efectiva al desarrollo del pensamiento computacional, ampliando las diferencias entre los grupos en la evaluación final, especialmente en las dimensiones relacionadas con la lógica y el razonamiento condicional. En consecuencia, se pone de relieve la relevancia de incorporar enfoques pedagógicos activos y estructurados que impulsen la adquisición de estas competencias en el aula.

### 3.1.7. Análisis de la prueba de entrada y de salida por Dimensiones en GE.

**Figura 7**

*Prueba de entrada y de salida por Dimensiones en GE*



*Nota.* 4 = destacado; 3= previsto; 2= en proceso; 1= en inicio. Fuente: elaboración propia.

Los resultados del grupo experimental evidenciaron una mejora significativa en todas las dimensiones del pensamiento computacional evaluadas. Al comparar los puntajes obtenidos en el pretest y el postest, se observó un avance notable en los niveles de desempeño, lo que indica que el uso de la plataforma tuvo un impacto positivo y efectivo en el proceso de enseñanza-aprendizaje.

En dimensiones como “Direcciones básicas”, “Repetir veces” y “Repetir hasta”, los estudiantes pasaron del nivel “En proceso” (nivel 2) al nivel “Logro previsto” (nivel 3), lo que indica una apropiación más sólida de conceptos clave relacionados con la secuencia y repetición de instrucciones. Estos resultados sugieren que las actividades visuales, dinámicas e interactivas de Code.org ayudaron considerablemente a que los estudiantes comprendieran mejor estos elementos.

Los avances más notables se dieron en las dimensiones “Condicional simple”, “Condicional compuesto” y “Bucle mientras”. Estas empezaron con niveles bajos en el pretest

(niveles 1 o 2) y lograron alcanzar niveles medios o esperados (niveles 2 o 3) en el postest. Este progreso es especialmente importante, ya que dichas estructuras requieren habilidades como el razonamiento lógico, la toma de decisiones y el pensamiento algorítmico, todas ellas claramente fortalecidas gracias al enfoque progresivo y guiado que propone Code.org.

Por último, en la dimensión “Funciones simples” también se registró un avance del nivel 2 al nivel 3, lo que demuestra una mejor comprensión del concepto de modularidad en la programación, favorecida por el uso de bloques reutilizables que la plataforma ofrece.

### 3.2. ANÁLISIS DE DATOS: ESTADÍSTICA INFERENCIAL

Para verificar la hipótesis estadística planteada se utilizó el paquete estadístico Jamovi versión 2.3.2 para analizar la misma. Previamente se tuvo que determinar la normalidad en la distribución de los datos (notas) para seleccionar el tipo de estadístico a utilizar para hacer la inferencia.

La Tabla 9 muestra los resultados de tres pruebas estadísticas de normalidad aplicadas a los datos del Pre Test las cuales evalúan si los datos siguen una distribución normal.

**Tabla 9**  
*Pruebas de normalidad*

Pruebas	Momento	Estadístico	p
Shapiro Wilk	Pre Test	0.962	0.225
		0.135	0.492
		0.445	0.270
Anderson Darling	Post Test	0.979	0.665
		0.109	0.757
		0.338	0.484

*Nota.* Elaboración propia basado en resultados proporcionados por *Moretest* de Jamovi 2.7.

Dado que en las tres pruebas el valor p es mayor que 0.05, se concluye que los datos del PRE TEST y el POST TEST no violan el supuesto de normalidad. Dado que los datos cumplen con la suposición de normalidad, es adecuado utilizar pruebas de tipo paramétricas, como la T de Student, para analizar y contrastar las evaluaciones iniciales y finales. Esto garantiza un análisis robusto y confiable en la comparación de los grupos de estudio.

### 3.2.1. Evaluación de entrada (grupo control y experimental)

**Tabla 10**

*Datos descriptivos de grupos en Pre Test*

	Grupo	N	Media	Mediana	DE	EE
Pre Test	Experimental	19	12.5	12	3.37	0.774
	Control	19	13.3	13	3.74	0.858

*Nota.* Elaboración propia basado en resultados proporcionados por Jamovi 2.7.

La Tabla 10 presenta los resultados del pretest de pensamiento computacional de los grupos experimental y control. En general, ambos grupos parten con niveles similares, aunque se notan ligeras diferencias en los promedios y en la variación de los puntajes.

El grupo control alcanzó una media de 13.3 y una mediana de 13, mientras que el grupo experimental obtuvo una media de 12.5 y una mediana de 12. Esta pequeña diferencia sugiere que los dos grupos iniciaron el estudio con condiciones bastante parecidas.

Respecto a la dispersión de los puntajes, el grupo control mostró una desviación estándar de 3.74, mientras que el grupo experimental registró una de 3.37. Esto significa que las calificaciones del grupo control fueron un poco más variadas, con diferencias más amplias entre estudiantes. Además, el error estándar fue ligeramente superior en el grupo control (0.858) en comparación con el experimental (0.774), lo que indica que la media del grupo experimental representa de forma un poco más precisa su rendimiento general.

**Tabla 11**

*Contraste de hipótesis de la evaluación de entrada GC vs. GE.*

Prueba T para Muestras Independientes	Estadístico	gl	p	Diferencia de medias	EE de la diferencia	Tamaño del Efecto	
Pre Test	T de Student	-0.683	36	0.499	-0.789	1.16	-0.222

*Nota.*  $p < .05$  (\*),  $p < .01$  (\*\*),  $p < .001$  (\*\*\*). Valores de  $p$  mayores a .05 indican ausencia de significancia estadística. Elaboración propia basado en resultados proporcionados por Jamovi 2.7.

La Tabla 11 presenta los resultados de la prueba t de Student para muestras independientes, aplicada para comparar los puntajes del pretest entre el grupo control (GC) y el

grupo experimental (GE). El objetivo fue comprobar si existían diferencias previas antes de implementar la intervención.

El análisis mostró un valor de  $t = -0.683$  y un  $p = 0.499$ , lo que indica que no hubo diferencias estadísticamente relevantes entre ambos grupos al inicio del estudio. Dado que el valor de  $p$  supera  $0.05$ , se mantiene la hipótesis nula, confirmando que los dos grupos presentaban un nivel de desempeño similar en pensamiento computacional antes de la aplicación del programa.

La diferencia de medias fue de  $-0.789$  puntos, con un error estándar de  $1.16$ , lo que demuestra que la distancia entre los promedios es mínima. Asimismo, el tamaño del efecto, calculado con la  $d$  de Cohen ( $-0.222$ ), refleja una influencia muy pequeña, prácticamente sin importancia práctica.

En resumen, los resultados confirman que el grupo experimental y el grupo control iniciaron el estudio en condiciones comparables, lo que garantiza una base sólida y equitativa para analizar con validez el efecto real de la intervención educativa.

### 3.2.2. Evaluación de salida (grupo control y experimental)

**Tabla 12**

*Datos descriptivos de grupos en Post Test*

	Grupo	N	Media	Mediana	DE	EE
Post Test	Experimental	19	18.1	18	3.47	0.796
	Control	19	14.1	14	3.98	0.913

*Nota.* Elaboración propia basado en resultados proporcionados por Jamovi 2.7.

La Tabla 12 muestra los resultados que reflejan una diferencia clara entre los dos grupos evaluados. El grupo experimental obtuvo una media de  $18.1$  puntos y una mediana de  $18$ , mientras que el grupo de control alcanzó una media de  $14.1$  puntos y una mediana de  $14$ . Esta diferencia de  $4$  puntos nos da a entender que el grupo del experimento tuvo un mejor rendimiento en general al terminar la intervención educativa.

En cuanto a la variación de los puntajes, ambos grupos presentaron un nivel de dispersión parecido: la desviación estándar fue de  $3.47$  en el grupo experimental y de  $3.98$  en el grupo control. Aunque el grupo control mostró una variación ligeramente mayor, en ambos casos los resultados se distribuyen de forma moderadamente uniforme entre los estudiantes. Respecto al error estándar, este fue de  $0.796$  en el grupo experimental y de  $0.913$  en el grupo control, lo cual indica que la media del grupo experimental es una estimación un poco más precisa del rendimiento general de sus estudiantes. Estos datos sugieren que la intervención

pedagógica con Code.org ayudó al grupo experimental a alcanzar un nivel más alto y constante de pensamiento computacional. Este primer resultado será puesto a prueba en el análisis de hipótesis que se presenta más adelante.

**Tabla 13**

*Contraste de hipótesis de la evaluación de salida GC vs. GE.*

Prueba T para Muestras Independientes		Estadístico	gl	p	Diferencia de medias	EE	Tamaño Efecto
Post Test	T de Student	3.3	36	0.002**	4	1.21	La d de Cohen 1.07

*Nota.*  $p < .05$  (\*),  $p < .01$  (\*\*),  $p < .001$  (\*\*\*). Valores de  $p$  mayores a .05 indican ausencia de significancia estadística. Elaboración propia basado en resultados proporcionados por Jamovi 2.7.

La Tabla 13 presenta los resultados de la prueba t de Student en los dos grupos de estudio, y compara los puntajes del postest de pensamiento computacional entre el grupo del experimento y el grupo control, con el fin de determinar el impacto de la intervención educativa basada en el uso de Code.org.

El valor obtenido del estadístico fue  $t = 3.3$ , y un valor  $p = 0.002$ , el cual es estadísticamente significativo ( $p < 0.05$ ). Esto indica que existen diferencias que son significativas en los niveles de pensamiento computacional entre ambos grupos tras la intervención. En consecuencia, se rechaza la hipótesis nula, lo que permite afirmar que la estrategia didáctica aplicada al grupo experimental tuvo un efecto positivo. La diferencia de medias entre los grupos fue de 4 puntos, con un error estándar de la diferencia de 1.21, lo que refuerza la magnitud del cambio observado.

Además, el tamaño del efecto, fue de 1.07, lo que se considera un efecto grande según los criterios convencionales (Cohen, 1988). Este valor indica que la intervención no solo tuvo un impacto estadísticamente significativo, sino también relevante desde el punto de vista práctico y educativo.

En resumen, los resultados confirman que el uso de la plataforma Code.org produjo mejoras sustanciales en el desempeño del grupo experimental, superando significativamente al grupo control en la evaluación de salida del pensamiento computacional.

### 3.2.3. Resultados relacionados pre y post test del grupo control

**Tabla 14**

*Datos Descriptivos para Muestras Relacionadas de GC*

<b>Descriptivas</b>	<b>N</b>	<b>Media</b>	<b>Mediana</b>	<b>DE</b>	<b>EE</b>
<b>Pre_control</b>	19	13.3	13	3.74	0.858
<b>Post_control</b>	19	14.1	14	3.98	0.913

*Nota.* Elaboración propia basado en resultados proporcionados por Jamovi 2.7.

La Tabla 14 nos muestra los resultados de tipo descriptivo del grupo control en las pruebas de pensamiento computacional realizadas antes y después del periodo de intervención. Como se puede apreciar, las diferencias entre la evaluación al inicio y la de salida fueron muy pequeñas. El promedio del grupo control pasó de 13.3 puntos en el pretest a 14.1 puntos en el postest, lo que representa un aumento leve de apenas 0.8 puntos. Las medianas también muestran una ligera variación, pasando de 13 a 14, lo que reafirma la estabilidad general del desempeño del grupo.

En cuanto a la dispersión de los puntajes, la desviación estándar creció levemente de 3.74 a 3.98. Esto sugiere que, aunque algunos estudiantes pudieron haber mejorado individualmente, esas mejoras no se reflejaron de manera uniforme en todo el grupo. El error estándar (EE) se mantuvo en valores cercanos, pasando de 0.858 a 0.913, lo que indica que no hubo cambios relevantes en la precisión con la que se estimó la media del grupo.

A partir de estos datos, se puede deducir que el grupo control no experimentó una mejora significativa en su desempeño, ya que no recibió una intervención pedagógica específica centrada en el desarrollo del pensamiento computacional. Aunque hubo un leve aumento en la media, este cambio probablemente se deba a factores externos o al aprendizaje habitual dentro del entorno escolar, más que a un enfoque sistemático de enseñanza de estas habilidades.

**Tabla 15**  
*Prueba T para Muestras Relacionadas GC*

Grupo Control	Estadístico	gl	p	Diferencia de medias	EE de la diferencia	Tamaño del Efecto
Pre Test Post Test	T de Student	-1.51	18	0.148	-0.789	0.522
						La d de Cohen -0.347

*Nota.*  $p < .05$  (\*),  $p < .01$  (\*\*),  $p < .001$  (\*\*\*)). Valores de  $p$  mayores a .05 indican ausencia de significancia estadística. Elaboración propia basado en resultados proporcionados por Jamovi 2.7.

La Tabla 15 resume los resultados para muestras relacionadas, aplicada al grupo control con el propósito de comprobar si existieron cambios entre los puntajes del pretest y posttest de pensamiento computacional.

El análisis estadístico mostró un valor de  $t = -1.51$ , con 18 grados de libertad y un  $p = 0.148$ . Dado que este valor supera el límite de 0.05, se concluye que no hay evidencia suficiente para considerar una diferencia real entre ambas mediciones. En consecuencia, se mantiene la hipótesis nula, lo que sugiere que el rendimiento del grupo control se mantuvo estable a lo largo del estudio.

La diferencia de medias, de -0.789 puntos, y el error estándar de 0.522, reflejan una variación mínima y sin relevancia estadística. Asimismo, el tamaño del efecto obtenido mediante la  $d$  de Cohen (-0.347) se clasifica como pequeño, lo que implica que la variación detectada carece de impacto práctico.

En síntesis, los datos evidencian que el grupo control no experimentó mejoras notorias en su pensamiento computacional durante el periodo analizado. La ausencia de una intervención pedagógica estructurada, como la que incluyó el uso de Code.org en el grupo experimental, podría explicar la escasa evolución observada en su desempeño.

### 3.2.4. Resultados relacionados pre y post test del grupo experimental

**Tabla 16**

*Datos Descriptivos para Muestras Relacionadas de GE*

Descriptivas	N	Media	Mediana	DE	EE
Pre Test	19	12.5	12	3.37	0.774
Post Test	19	18.1	18	3.47	0.796

*Nota.* Elaboración propia basado en resultados proporcionados por Jamovi 2.7.

La Tabla 16 muestra los resultados descriptivos obtenidos en las evaluaciones aplicadas al grupo experimental antes y después de la intervención educativa con la plataforma Code.org, con el objetivo de evaluar su impacto en el desarrollo del pensamiento computacional. Los datos reflejan una mejora significativa en el desempeño de los estudiantes del grupo experimental. La media se incrementó de 12.5 en el pretest a 18.1 en el postest, lo que supone una diferencia de 5.6 puntos. Asimismo, la mediana pasó de 12 a 18, evidenciando que el progreso fue consistente en la mayoría de los participantes, confirmando la efectividad de la intervención.

Respecto a la dispersión de los puntajes, la desviación estándar (DE) se mantuvo prácticamente igual, variando ligeramente de 3.37 a 3.47. Esto indica que, a pesar del incremento en el rendimiento, los resultados entre los estudiantes continuaron siendo similares entre sí. Esta estabilidad es un aspecto positivo, ya que demuestra que la mejora no se concentró en unos pocos estudiantes, sino que fue generalizada.

El error estándar (EE) también se mantuvo casi sin cambios, siendo de 0.774 en el pretest y de 0.796 en el postest. Esta consistencia sugiere que las medias obtenidas son confiables y reflejan adecuadamente el rendimiento del grupo.

En resumen, los resultados reflejan un avance significativo en el grupo experimental, el cual se atribuye a la intervención pedagógica con el uso de Code.org. Este progreso será analizado con mayor detalle en la siguiente tabla, mediante el contraste estadístico de hipótesis para muestras relacionadas.

**Tabla 17**

*Prueba T para Muestras Relacionadas GE*

Grupo Experimental	Estadístico	gl	p	Diferencia de medias	EE	Tamaño del Efecto
Pre Test Post Test	T de Student -7.28	18	< .001***	-5.58	0.766	La d de Cohen -1.67

*Nota.*  $p < .05$  (\*),  $p < .01$  (\*\*),  $p < .001$  (\*\*\*)). Valores de p mayores a .05 indican ausencia de significancia estadística. Elaboración propia basado en resultados proporcionados por Jamovi 2.7.

La Tabla 17 presenta los resultados de la prueba t de Student para muestras relacionadas, aplicada al grupo experimental con el propósito de determinar si la intervención educativa basada en Code.org produjo cambios relevantes en el desempeño de los estudiantes respecto al pensamiento computacional.

El análisis estadístico arrojó un valor de  $t = -7.28$ , con 18 grados de libertad y un  $p < .001$ , lo que demuestra una diferencia altamente significativa entre los puntajes obtenidos antes y después de usar la plataforma. Este hallazgo permite descartar la hipótesis nula y confirma que la mejora registrada en el grupo experimental no responde al azar, sino al efecto directo del uso pedagógico de la plataforma.

La diferencia media entre las dos mediciones fue de  $-5.58$  puntos, con un error estándar de  $0.766$ , lo que evidencia un avance claro y sostenido en el rendimiento. Además, el tamaño del efecto ( $-1.67$ ), corresponde a un efecto muy grande, superando ampliamente el umbral de  $0.8$  establecido por Cohen, (1988) para este tipo de comparación.

Estos resultados muestran que la intervención no solo generó resultados estadísticamente relevantes, sino que también tuvo una incidencia real y perceptible en el aprendizaje, manifestada en una mejora amplia y constante entre los participantes del grupo experimental. En síntesis, la evidencia confirma que la implementación de Code.org favoreció de manera contundente el desarrollo del pensamiento computacional, consolidando su valor como herramienta eficaz dentro del proceso educativo.

### 3.3. DISCUSIÓN

Este estudio ha demostrado que usar la plataforma educativa de nombre Code.org tiene un impacto claro y positivo en el desarrollo del pensamiento computacional en estudiantes de secundaria. El grupo experimental que trabajó con esta herramienta mostró una mejora importante en sus puntajes entre el pretest y el postest, con un efecto muy fuerte ( $d = -1.67$ ). En comparación, el grupo que no recibió esta intervención (grupo control) tuvo una mejora leve y sin significado estadístico. Además, al comparar ambos grupos, se observó una diferencia significativa a favor del grupo que nos sirvió de experimento, con un efecto considerable ( $d = 1.07$ ).

Estos hallazgos coinciden con estudios realizados en otros países. Por ejemplo, Oluk y Çakir (2021) también comprobaron que los estudiantes de secundaria mejoraron sus habilidades en pensamiento computacional y en desarrollo de algoritmos tras usar Code.org, mientras que los del grupo control no mostraron cambios importantes. Ambos estudios tienen un enfoque similar, ya que utilizaron un diseño cuasiexperimental, y confirman que Code.org funciona bien en contextos reales dentro de las escuelas.

De igual manera, Geliş y Kabul (2022) encontraron un aumento importante en los resultados del postest después de aplicar el Curso 2 de Code.org, aunque mencionaron que aún hay margen para mejorar en el área de codificación. A diferencia de ellos, este estudio no solo

muestra mejoras importantes desde el punto de vista de la estadística, sino también desde lo práctico: muchos estudiantes pasaron del nivel "En proceso" a niveles más altos como "Logro previsto" o incluso "Logro destacado".

Desde una mirada más cualitativa, Escalante Coronel y Guerra Olivella (2024) destacaron que Code.org no solo mejora aspectos técnicos como bucles o lógica booleana, sino que también ayuda a desarrollar el pensamiento considerado crítico y a aumentar la motivación de los estudiantes. Esto coincide con lo encontrado en esta investigación, donde el uso de bloques visuales y la participación activa de los alumnos permitieron avances importantes en áreas como condicionales y funciones simples.

Por otro lado, Kale y sus colegas (2023) resaltaron que Code.org también contribuye al desarrollo de habilidades blandas, como la reflexión y el trabajo en equipo, además de mejorar las competencias tecnológicas. Aunque este estudio no evaluó directamente ese tipo de habilidades, el hecho de que los puntajes del grupo experimental mejoraron de forma pareja y sin aumentar la diferencia entre estudiantes (la desviación estándar se mantuvo similar), sugiere que Code.org también promueve una experiencia de aprendizaje equilibrada para todos.

En el contexto nacional, los resultados coinciden con los del estudio de Pimentel (2024), quien también encontró mejoras significativas en el pensamiento computacional, especialmente en aspectos como el análisis y la abstracción, tras aplicar la plataforma CODE. En este estudio se observó algo similar, ya que las dimensiones de repetición, bucles y condicionales mostraron avances importantes después de la intervención, lo que refuerza la idea de que estas herramientas digitales ayudan a fortalecer habilidades clave en los estudiantes.

Por otro lado, Siu Antezana (2024) no encontró una relación que sea significativa entre competencias digitales y el pensamiento computacional. Esta diferencia con nuestro estudio puede deberse a los métodos utilizados: mientras su investigación fue de tipo correlacional, en este caso se aplicó una intervención específica, lo que permite observar resultados más claros y medibles sobre el impacto en las habilidades trabajadas.

El trabajo de Díaz Sagástegui y Lozano (2019) también es relevante, ya que muestra una relación positiva entre el uso de plataformas como Code.org y Scratch y el aprendizaje de programación en estudiantes de primaria. Aunque este estudio se enfocó en secundaria, los resultados van en la misma línea: las herramientas visuales ayudan a los estudiantes a desarrollar habilidades lógicas y creativas desde temprana edad, además de generar una buena percepción entre ellos.

A nivel local, el estudio de Ballón Aguedo (2023) reveló que los estudiantes de educación básica alternativa en Arequipa tienen un nivel adecuado del pensamiento

computacional y las habilidades matemáticas, en parte gracias al uso de entornos virtuales. Esto sugiere que plataformas como Code.org pueden mejorar las capacidades cognitivas en distintos niveles educativos, algo que también se confirma en esta investigación con estudiantes de secundaria.

Asimismo, el estudio de Churata Mayta y colaboradores (2021) en Arequipa demostró que el Programa de Codigoalfabetización ayudó a mejorar significativamente las áreas de direcciones, bucles, condicionales y funciones. Estas mismas dimensiones fueron reforzadas en este estudio, donde los estudiantes del grupo que fueron situados como experimental pasaron de niveles bajos a logros destacados. Además, los métodos de ambos estudios son similares, lo que refuerza la validez de los resultados obtenidos.

En general, los resultados respaldan la efectividad de Code.org no solo como una herramienta útil para la enseñanza de la programación, sino también como un recurso educativo integral que favorece el desarrollo de habilidades esenciales, como el pensamiento lógico, la resolución de problemas, el razonamiento algorítmico y otras competencias cognitivas de alto nivel.

A partir de los datos obtenidos y de la revisión de investigaciones previas, se concluye que la implementación de plataformas como Code.org en la educación secundaria constituye una estrategia pedagógica que puede ser considerada viable, motivadora y altamente efectiva para fortalecer el pensamiento computacional y fomentar un aprendizaje más dinámico y significativo.

Finalmente, se recomienda seguir explorando el uso de esta plataforma en combinación con otras metodologías activas, como el del aprendizaje basado en proyectos, la gamificación o el trabajo colaborativo. También sería útil estudiar cómo adaptarla a contextos con menor acceso a tecnología, para ampliar su alcance e impacto educativo.

## CONCLUSIONES

### PRIMERA:

El estudio permitió analizar cómo el uso de la plataforma Code.org influyó en el desarrollo del pensamiento computacional de los estudiantes de primer grado de secundaria de la I.E. Gran Pachacutec, en Arequipa. Los resultados mostraron que el grupo que trabajó con la plataforma mejoró notablemente su desempeño en comparación con el grupo que siguió la metodología tradicional. Esto demuestra que la integración de herramientas digitales interactivas puede convertirse en una estrategia eficaz para fortalecer las habilidades propias del pensamiento computacional, aportando una alternativa innovadora y pertinente para el contexto educativo peruano.

### SEGUNDA:

En la evaluación inicial del pensamiento computacional (pretest), se observó que tanto el grupo control como el grupo experimental partían de niveles similares de desempeño, sin diferencias relevantes desde el punto de vista estadístico. Esta equivalencia inicial permitió contar con una base de comparación justa y confiable, indispensable para valorar con precisión los cambios obtenidos tras la implementación de la plataforma Code.org.

### TERCERA:

Los resultados posteriores a la intervención muestran que el grupo experimental que empleó Code.org obtuvo un rendimiento significativamente mayor que el grupo control. En la evaluación final, la media del grupo experimental alcanzó 18.1, mientras que el grupo control registró 14.1, evidenciando una diferencia de 4.0 puntos. Esta variación, estadísticamente significativa ( $p < .002$ ), confirma que la plataforma incidió positivamente en el desarrollo del pensamiento computacional. En consecuencia, se acepta la hipótesis alterna y se rechaza la hipótesis nula al demostrarse un efecto real atribuible al uso de Code.org.

### CUARTA:

El análisis de los resultados que compara las evaluaciones de entrada y de salida del grupo experimental, se demostró una mejora clara y sostenida en las principales dimensiones del pensamiento computacional, como el razonamiento lógico del *entorno-interfaz*, la *alternancia de respuestas*, las *estructuras anidadas* y las *tareas requeridas*. Estos avances confirman la eficacia de la plataforma Code.org como recurso pedagógico, gracias a su carácter

visual, interactivo y basado en desafíos graduales, que facilita la comprensión y aplicación de conceptos desde los más simples hasta los más complejos. En total, los resultados demuestran que Code.org no solo despierta el interés y la motivación del estudiante, sino que también fortalece de manera notable sus habilidades en pensamiento computacional.



## RECOMENDACIONES

### **Primera:**

A las direcciones de las instituciones educativas y unidades de gestión educativa local (UGEL) se recomienda implementar programas permanentes de capacitación docente orientados al uso pedagógico de herramientas tecnológicas como Code.org. Estos programas deben incluir talleres prácticos y acompañamiento técnico para que los docentes desarrollen competencias digitales aplicables al aula y promuevan el pensamiento computacional de los estudiantes.

### **Segunda:**

Al Ministerio de Educación y a los equipos de innovación pedagógica de las instituciones educativas, se sugiere integrar Code.org en el currículo escolar, estableciendo horas específicas dentro del plan de estudios para su implementación. Asimismo, se recomienda definir objetivos de aprendizaje claros y medibles relacionados con la resolución de problemas, la creatividad y la lógica computacional.

### **Tercera:**

Se recomienda que investigadores y centros de formación amplíen futuras líneas de estudio a distintos contextos educativos y empleen muestras mayores con intervenciones más prolongadas. Esto permitirá comprender mejor el impacto de Code.org y fortalecer la validez y la generalización de los hallazgos a nivel nacional.

## REFERENCIAS

- Ballón Aguedo, S. (2023). *Desarrollo de entornos virtuales generados por las TIC, pensamiento computacional y las competencias del área de matemática en estudiantes de 4° grado del CEBA CIDECH Perú - Arequipa, 2022* [Universidad Católica de Santa María]. En *Repositorio Institucional de la Universidad Católica de Santa María*. <https://repositorio.ucsm.edu.pe/items/67430543-2cb8-409f-8e5b-de48f7a859c0>
- Barr, V., & Stephenson, C. (2011). Bringing computational thinking to K-12: What is involved and what is the role of the computer science education community? *ACM Inroads*, 2(1), 48-54. <https://doi.org/10.1145/1929887.1929905>
- Barradas, R., Lencastre, J., Soares, S., & Valente, A. (2020). Developing Computational Thinking in Early Ages: A Review of the code.org Platform. *Proceedings of the 12th International Conference on Computer Supported Education (CSEDU 2020)*, 2, 157-168. <https://typeset.io/pdf/developing-computational-thinking-in-early-ages-a-review-of-3erp3xvwyl.pdf>
- Bezoz, J., Zuckerberg, M., & Gates, B. (2013). *Code.org en Italia*. 1-5.
- Birds, A. (2021). *Express Course (2021) Standards*. Code.org. <https://studio.code.org/s/express-2021/standards>
- Churata Mayta, E. B., Coila Cárdenas, T. E., & Tintaya Quispe, L. M. (2021). Influencia del scrath en el desarrollo del pensamiento computacional en los estudiantes del quinto de primaria de la institución educativa 40028 A lto Selva Alegre-Arequipa-2019 [Tesis. Universidad Católica de Santa María]. En *Repositorio Institucional de la UCSM*. <http://tesis.ucsm.edu.pe/repositorio/bitstream/handle/UCSM/9944/P1.2047.MG.pdf?sequence=1&isAllowed=y>
- Clark, D. B., Tanner-Smith, E. E., & Killingsworth, S. S. (2016). Digital Games, Design, and Learning. *Review of Educational Research*, 86(1), 79-122. <https://doi.org/10.3102/0034654315582065>
- Cohen, J. (1988). *Statistical power analysis for the behavioral sciences* (2.<sup>a</sup> ed.). Lawrence Erlbaum Associates, Publishers.
- Diaz Sagástegui, G., & Lozano, R. (2019). *Uso de las aplicaciones Code.org y Scratch para el aprendizaje de programación en los estudiantes del 5° y 6° grado de EBR del C.E. N° 82099 de la Provincia de San Pablo, 2018* [Universidad Privada Antonio Guillermo Urrel]. <http://repositorio.upagu.edu.pe/handle/UPAGU/1125>
- Dwivedi, R. K., Bisen, S., Yadav, M., & Yadav, A. (2024). *Coding and Computational*

- Thinking* (pp. 10-24). <https://doi.org/10.4018/979-8-3693-5370-7.ch002>
- Escalante Coronel, W., & Guerra Olivella, R. (2024). Estrategia didáctica mediada por la plataforma code.org para la enseñanza del pensamiento computacional en estudiantes de la institución educativa técnica Andrés Bello. *Actualidades Pedagógicas*, 1(82), e1816. <https://doi.org/10.19052/ap.vol1.iss82.6>
- Geliş, T., & Kabul. (2022). Evaluation of the Code. Org Online Coding Platform Curriculum. *Milli Egitim*, February, 689-715. <https://doi.org/10.37669/milliegitim.799492>
- Grover, S., & Pea, R. (2013). Computational Thinking in K–12. *Educational Researcher*, 42(1), 38-43. <https://doi.org/10.3102/0013189X12463051>
- Kale, U., Yuan, J., & Roy, A. (2023). Thinking processes in code.org: A relational analysis approach to computational thinking. *Computer Science Education*, 33(4), 545-566. <https://doi.org/10.1080/08993408.2022.2145549>
- Mihm, C. M. (2021). Why Teach Coding to Early Elementary Learners. En *Teaching Computational Thinking and Coding to Young Children* (pp. 21-40). <https://doi.org/10.4018/978-1-7998-7308-2.ch002>
- Oluk, A., & Çakir, R. (2021). The Effect of Code. Org Activities on Computational Thinking and Algorithm Development Skills. *Journal of Teacher Education and Lifelong Learning*, 3(2), 32-40. <https://doi.org/10.51535/tell.960476>
- Peñaloza-Ochoa, S., & Ortega-Chasi, P. (2023). Computational Thinking as a Didactic Strategy for the Development of Mathematical Competencies in Problem-Solving. En *Actas de la XVIII Conferencia Latinoamericana de Tecnologías de Aprendizaje (LACLO 2023)* (pp. 86-100). [https://doi.org/10.1007/978-981-99-7353-8\\_8](https://doi.org/10.1007/978-981-99-7353-8_8)
- Pimentel, J. (2024). *Plataforma CODE en el desarrollo del pensamiento computacional en estudiantes de secundaria de la Institución Educativa San Juan de Huasahuasi, Tarma* [Universidad Nacional del Centro]. <https://repositorio.uncp.edu.pe/handle/20.500.12894/11471>
- Relkin, E., de Ruiter, L. E., & Bers, M. U. (2021). Learning to code and the acquisition of computational thinking by young children. *Computers & Education*, 169(April), 104222. <https://doi.org/10.1016/j.compedu.2021.104222>
- Rim, H. (2017). A Study on Teaching using Website «Code.org» in Programming Education based on Computational Thinking. *Journal of Korea Multimedia Society*, 20(2), 382-395. <https://doi.org/10.9717/kmms.2017.20.2.382>
- Román-González, M. (2016). *Codigoalfabetización y pensamiento computacional en educación primaria y secundaria: validación de un instrumento y evaluación de*

- programas* [Universidad Nacional de Educación a Distancia]. <https://e-spacio.uned.es/entities/publication/d359acd5-5706-4996-b4f0-dd98e2ff8736>
- Selby, C., & Woollard, J. (2013). Computational Thinking: The Developing Definition. *ITiCSE Conference* 2013, 5-8. [https://eprints.soton.ac.uk/356481/1/Selby\\_Woollard\\_bg\\_soton\\_eprints.pdf](https://eprints.soton.ac.uk/356481/1/Selby_Woollard_bg_soton_eprints.pdf)
- Siu Antezana, R. (2024). Competencias digitales y pensamiento computacional en estudiantes del VII ciclo de una institución educativa jornada escolar completa, Arequipa 2023. [Universidad Católica de Trujillo Benedicto XVI]. En *Repositorio de tesis de la UCT*. [http://repositorio.uct.edu.pe/bitstream/123456789/346/1/0061220211\\_0001193711\\_T\\_2018.pdf](http://repositorio.uct.edu.pe/bitstream/123456789/346/1/0061220211_0001193711_T_2018.pdf)
- Stoilescu, D. (2008). Stuck in the Shallow End: Education, race and computing. En *McGill Journal of Education* (Vol. 45, Número 3). Cambridge MA: MIT Press. <https://doi.org/10.7202/1003580ar>
- Valverde-Berrocoso, J., Fernández-Sánchez, M. R., & Garrido-Arroyo, M. C. (2015). El pensamiento computacional y las nuevas ecologías del aprendizaje. *Revista de Educación a Distancia (RED)*, 46(46). <https://doi.org/10.6018/red/46/3>
- Voogt, J., Erstad, O., Dede, C., & Mishra, P. (2013). Challenges to learning and schooling in the digital networked world of the 21st century. *Journal of Computer Assisted Learning*, 29(5), 403-413. <https://doi.org/10.1111/jcal.12029>
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33-35. <https://doi.org/10.1145/1118178.1118215>
- Yadav, A., Hong, H., & Stephenson, C. (2016). Computational Thinking for All: Pedagogical Approaches to Embedding 21st Century Problem Solving in K-12 Classrooms. *TechTrends*, 60(6), 565-568. <https://doi.org/10.1007/s11528-016-0087-7>

## Anexo 1 Matriz de Consistencia

TÍTULO	FORMULACIÓN DEL PROBLEMA	HIPÓTESIS	OBJETIVOS	VARIABLES	DIMENSIONES	METODOLOGÍA
Uso de la plataforma educativa Code.org en la mejora del desarrollo del pensamiento computacional en estudiantes del primer grado de educación secundaria en Arequipa, Perú	<p><b>General</b> ¿Qué efecto tiene el uso de la plataforma educativa Code.org en el desarrollo del pensamiento computacional de los estudiantes del primer grado de la I.E. Gran Pachacutec, Arequipa?</p> <p><b>Específicas</b> ¿Cuál es el nivel inicial de pensamiento computacional en los estudiantes del primer grado del grupo experimental y el grupo control de la I.E. Gran Pachacutec, Arequipa, antes de la intervención con Code.org? ¿Existen diferencias significativas en el nivel de pensamiento computacional entre el grupo experimental (que usa Code.org) y el grupo control (que no lo usa) después de la intervención, según los resultados de la evaluación posttest? ¿El uso de Code.org mejora de manera significativa las dimensiones específicas del pensamiento computacional en los estudiantes del primer grado de la I.E. Gran Pachacutec, Arequipa.?</p>	<p><b>Hipótesis General</b> 1. El uso de la plataforma educativa Code.org mejora significativamente el desarrollo del pensamiento computacional en estudiantes del primer grado de educación secundaria de la I.E. Gran Pachacutec, Arequipa.</p> <p><b>Hipótesis Específicas</b> 2. Los niveles iniciales de pensamiento computacional, medidos mediante una prueba pretest, no presentan diferencias significativas entre el grupo experimental y el grupo control del primer grado de la I.E. Gran Pachacutec, Arequipa. 3. Existen diferencias significativas en los niveles de pensamiento computacional entre el grupo experimental y el grupo control después de la intervención con Code.org, mostrando el grupo experimental puntuaciones más altas en la evaluación posttest. 4. La implementación de una estrategia didáctica basada en pensamiento computacional mediante el uso de la plataforma Code.org genera una mejora significativa en los niveles de logro del grupo experimental en las distintas dimensiones del pensamiento computacional.</p>	<p><b>General</b> 1. Evaluar el efecto del uso de la plataforma educativa Code.org en la mejora del desarrollo del pensamiento computacional en estudiantes del primer grado de educación secundaria de la I.E. Gran Pachacutec, Arequipa, mediante la comparación de un grupo experimental y un grupo control.</p> <p><b>Específicos</b> 2. Establecer, mediante una prueba pretest, la línea base del nivel de pensamiento computacional en los estudiantes de los grupos control y experimental del primer grado de la I.E. Gran Pachacutec, Arequipa. 3. Comparar, mediante una evaluación posttest, las diferencias significativas en los niveles de pensamiento computacional entre el grupo experimental que utiliza Code.org y el grupo control que no lo utiliza. 4. Analizar el efecto de una estrategia didáctica basada en pensamiento computacional en el grupo experimental, comparando los niveles de logro por dimensiones antes y después de la intervención con Code.org.</p>	<p><b>Uso de la plataforma educativa CODE.ORG</b></p> <p><b>Pensamiento computacional</b></p>	<ol style="list-style-type: none"> <li>Aspectos psicopedagógicos</li> <li>Aspectos técnicos</li> </ol> <ol style="list-style-type: none"> <li>Conceptos computacionales</li> <li>Entorno-Interfaz del Ítem</li> <li>Estilo de alternancia de la respuesta</li> <li>Estructuras anidadas</li> <li>Tarea Requerida</li> </ol>	<p><b>Tipo de investigación:</b> Explicativo (experimental).</p> <p><b>Nivel:</b> Aplicado.</p> <p><b>Diseño:</b> Cuasiexperimental con pretest-postest. GC y GE.</p> <p><b>Enfoque temporal:</b> Longitudinal (mediciones en dos momentos: inicio y final de la intervención).</p> <p><b>Técnicas:</b> 1. Observación sistemática (para monitorear la implementación de las sesiones con Code.org). 2. Evaluación estandarizada (para medir el pensamiento computacional).</p> <p><b>Instrumentos:</b> 1. Sesiones de aprendizaje (registro de la aplicación de las 5 sesiones experimentales). 2. Test de pensamiento computacional de Román-González (2016) (pretest y posttest).</p> <p><b>Variables:</b> - Independiente (VI): Intervención con Code.org (5 sesiones diseñadas). - Dependiente (VD): Nivel de pensamiento computacional (puntuación en el test).</p> <p><b>Unidades de estudio:</b> 48 estudiantes de primer grado de secundaria de la I.E. Gran Pachacútec (Arequipa, Perú), divididos en: - Grupo experimental: 25 estudiantes (intervención con Code.org). - Grupo control: 23 estudiantes (metodología convencional).</p>

**Anexo 2**

*Sesiones de aprendizaje. Propuesta que integra el uso de Code.org*





## Sesión de Aprendizaje N° 01 Primero

### I unidad de aprendizaje

#### Título de sesión: "Introducción al pensamiento computacional"

#### I. DATOS INFORMATIVOS

Institución Educativa:	Gran Pachacútec	Área:	Educación para el trabajo
Director:	César Vásquez Meza	Docente:	Raúl Aymara Alfaro
Grado:	Primer grado	Duración:	02 horas
Sección:	A	Fecha:	25/03/2025

#### II. PROPÓSITO DE LA SESIÓN

"Los estudiantes identifican los conceptos fundamentales del pensamiento computacional."

#### III. PROPÓSITOS DE APRENDIZAJE

COMPETENCIAS	CAPACIDADES	DESEMPEÑO PRECISADO	CRITERIOS DE EVALUACIÓN	INSTRUMENTOS DE EVALUACIÓN	EVIDENCIAS DE APRENDIZAJE
<b>Gestiona proyectos de emprendimiento o económico o social.</b>	Trabaja cooperativamente para lograr objetivos y metas.	Propone acciones que debe realizar el equipo explicando cómo integra el pensamiento computacional a las actividades de la vida cotidiana. Promueve la perseverancia por lograr el objetivo común a pesar de las dificultades y cumple con responsabilidad las tareas asignadas a su rol.	Identificamos los principios básicos del pensamiento computacional aplicando la descomposición del problema, identificación de patrones, usando la abstracción para simplificar las tareas y creando algoritmos para seguir en ejemplos de la vida cotidiana.	Lista de cotejo	Elaboración de algoritmos de la vida cotidiana por cada grupo
<b>Gestiona su aprendizaje de manera autónoma</b>	Organiza acciones estratégicas para alcanzar sus metas de aprendizaje	Organiza un conjunto de estrategias y acciones en función del tiempo y de los recursos de que dispone, para responder en equipo a los retos de los principios básicos del pensamiento computacional.	En equipo plantea ejemplos de la vida cotidiana que se relacionen con la descomposición del problema, identificación de patrones, usando la abstracción para simplificar las tareas y creando algoritmos.		

#### IV. SECUENCIA DIDÁCTICA

##### INICIO

**MOTIVACIÓN:** ¿Cuáles son las actividades diarias que realizamos en casa antes de venir al colegio?, Al menos tres estudiantes participan y cuentan sus casos particulares. ¿Podría alterar esa secuencia como, por ejemplo: cambiarme de ropa y luego ducharme? ¿Por qué?

**EXPLORACIÓN DE SABERES PREVIOS:**

- ¿Qué es un problema?
- ¿Cómo resolvemos las dificultades o problemas?
- ¿Habrán estrategias que nos permitan mejorar la capacidad de resolución de problemas?

**PROBLEMATIZACIÓN O CONFLICTO:** Desequilibrio cognitivo.

- ¿Cómo crees que una calculadora resuelve una suma o multiplicación?

##### DESARROLLO

**GESTIÓN Y ACOMPAÑAMIENTO:** Se activan los procesos didácticos y auxiliares de acuerdo con el enfoque del área.

**ACTIVIDAD 1: Introducción al Pensamiento Computacional**



**Presentación:** El docente explica que el pensamiento computacional es una forma de resolver problemas y entender el mundo a través de conceptos como descomposición, patrones, abstracción y algoritmos.



**Relación con la Vida Cotidiana:** El docente realiza analogías con tareas diarias, como hacer una tarea doméstica, preparar un desayuno.

**ACTIVIDAD 2: Descomposición**

- Explica que la descomposición es el proceso de dividir una tarea grande en tareas más pequeñas y manejables. Pide a los estudiantes que piensen en cómo preparar un sándwich de hamburguesa de pollo. Luego, los guía a descomponer el proceso en pasos simples (ejemplo, colocar en el zarten con poco de aceite la hamburguesa y freír, cortar los tomar el pan, poner la hamburguesa frita, poner las papitas al hilo, etc.).

- Dividir a los estudiantes en grupos y pedirles que descompongan una tarea familiar, como lavar los platos o realizar una compra. Cada grupo debe presentar sus pasos al resto de la clase.

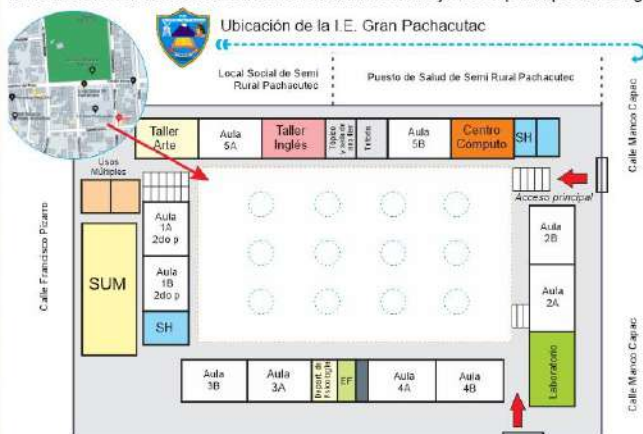
**ACTIVIDAD 3: Reconocimiento de Patrones**

- El docente explica que el reconocimiento de patrones involucra identificar similitudes y diferencias en situaciones o objetos. Muestra imágenes de diferentes animales y pide a los estudiantes que identifiquen patrones comunes (ejemplo, todos tienen patas, ojos, etc.).

- Pide a los estudiantes que busquen patrones en su entorno escolar (ejemplo, el horario de clases, el diseño de la sala, etc.) y compartan sus observaciones.

**ACTIVIDAD 4: Abstracción**

- El docente explica que la abstracción es el proceso de enfocarse en las características esenciales de un problema o tarea, ignorando los detalles no relevantes. Muestra a los estudiantes el dibujo del mapa simple del colegio, destacando solo los elementos esenciales.





I.E. GRAN PACHACUTEC  
Arequipa - Perú

Pide a los estudiantes que describan cómo llegar a un lugar específico del colegio, usando solo las características esenciales (ejemplo, "Empieza en la entrada, gira a la derecha, camina cruzando el patio, sube las gradas")

**ACTIVIDAD 5: Algoritmos**

- Explica que un algoritmo es una secuencia de pasos para resolver un problema o realizar una tarea. Pide a los estudiantes que escriban un algoritmo para cepillarse los dientes, incluyendo cada paso detallado (ejemplo, "Tomar el cepillo dental", "Poner pasta en el cepillo", "frotar con el cepillo en los dientes", etc.).
- Divide a los estudiantes en parejas. Uno escribe un algoritmo para realizar una tarea simple (ejemplo, hacer una tarea de matemáticas), y el otro sigue los pasos exactamente como están escritos. Luego, discuten si el algoritmo funcionó correctamente.

**CIERRE**

**EVALUACIÓN Y RETROALIMENTACIÓN** (metacognición) 15 m

Retroalimentación hacemos algunas preguntas

- ¿Qué aprendimos el día de hoy?
- ¿Cómo podemos aplicar el pensamiento computacional en la vida diaria?
- ¿Qué dificultades hemos tenido?

V. ESPACIO EDUCATIVO

- Aula de clases

VI. MATERIALES Y RECURSOS PARA LA SESIÓN

- Proyector multimedia
- Pizarra
- Plumones

VII. OBSERVACIONES

Arequipa, 25 marzo del 2025



Prof. Raúl Aymara Alfaro





I.E. GRAN PACHACUTEC  
Arequipa - Perú

## Sesión de Aprendizaje N° 02 Primero

### I unidad de aprendizaje

Título de sesión: "Navegando el Entorno Virtual: Mi Primera Interacción en Code.org"

#### I. DATOS INFORMATIVOS

Institución Educativa:	Gran Pachacútec	Área:	Educación para el trabajo
Director:	César Vásquez Meza	Docente:	Raúl Aymara Alfaro
Grado:	Primer grado	Duración:	02 horas
Sección:	A	Fecha:	01/04/2025

#### II. PROPÓSITO DE LA SESIÓN

"Nos familiarizamos con el entorno Code.org (interfaz del ítem, uso de comandos por bloques gráficos).  
Desarrollar autonomía y responsabilidad al navegar en un entorno virtual."

#### III. PROPÓSITOS DE APRENDIZAJE

COMPETENCIAS	CAPACIDADES	DESEMPEÑO PRECISADO	CRITERIOS DE EVALUACIÓN	INSTRUMENTOS DE EVALUACIÓN	EVIDENCIAS DE APRENDIZAJE
<b>Gestiona proyectos de emprendimiento o económico o social.</b>	Aplica habilidades técnicas.	Utiliza de manera básica la plataforma Code.org para resolver retos de programación por bloques, aplicando habilidades técnicas digitales que contribuyen al desarrollo de propuestas innovadoras.	Supera retos iniciales aplicando comandos gráficos, demostrando comprensión de las instrucciones.	Lista de cotejo	Evidencia de avance en los niveles iniciales de Code.org (ej. primeras secciones de El Laberinto o El Lienzo)
<b>Gestiona su aprendizaje de manera autónoma</b>	Interactúa en entornos virtuales de acuerdo con sus necesidades.	Reconoce la interfaz de Code.org y sus elementos básicos para orientarse y participar en el entorno virtual. Configura su perfil en la plataforma de forma autónoma y responsable, siguiendo pautas de seguridad digital. Resuelve los primeros retos de programación por bloques (ej. El Laberinto), aplicando instrucciones de movimiento en un entorno visual.	Accede y configura su cuenta en Code.org, gestionando su perfil con responsabilidad digital. Navega en el entorno de programación visual por bloques, identificando las funciones básicas de la interfaz.		

#### IV. SECUENCIA DIDÁCTICA

##### INICIO

##### MOTIVACIÓN:

Se formulan preguntas generadoras:

¿Qué ventajas tiene aprender en un entorno virtual?

¿Cómo podemos cuidarnos y trabajar de manera responsable en estas plataformas?

##### EXPLORACIÓN DE SABERES PREVIOS:

a. ¿Qué sitios web visitaste estos últimos días?

b. ¿Qué complicaciones o dificultades notaste?

##### PROBLEMATIZACIÓN O CONFLICTO: Desequilibrio cognitivo.

a. ¿Alguna vez creaste una cuenta en algún sitio web que visitas con frecuencia?

b. ¿Qué beneficios tiene registrarse en un sitio web?



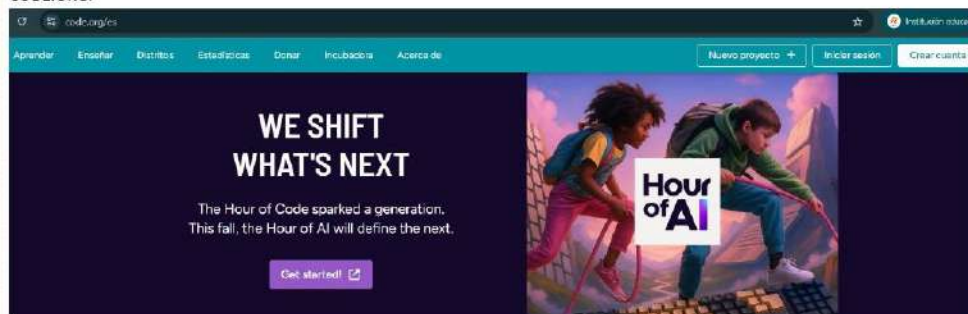
I.E. GRAN PACHACUTEC  
Arequipa - Perú

DESARROLLO

**GESTIÓN Y ACOMPAÑAMIENTO:** El docente presenta Code.org como una herramienta gratuita, accesible y gamificada para aprender programación. Dialoga con los estudiantes sobre la importancia de la competencia “Se desenvuelve en entornos virtuales” en la vida diaria y profesional.

**ACTIVIDAD 1: Ingresamos a nuestra cuenta institucional**

**Presentación:** El docente pide a los estudiantes accedan a su cuenta institucional @gp.edu.pe y luego se dirijan a la plataforma CODE.ORG.



Se pregunta: ¿Qué puedes observar en este entorno? ¿Dónde crees que podemos registrarnos?

**ACTIVIDAD 2: Registro en el aula virtual de CODE.ORG**

- El docente guía a los estudiantes para que puedan registrarse en el sitio web code.org, usando un enlace compartido mediante el aula virtual de Google classroom.



**Vídeo: Introducción a Code Studio**



- Reconocen el entorno de trabajo: identificando la parte gráfica, reto y área de programación. El docente resalta que deben comprender el reto, descomponerlo en partes más pequeñas, abstraerse y generar el algoritmo que solucionará el reto:





I.E. GRAN PACHACUTEC  
Arequipa - Perú

- Demostración breve del primer reto de El Laberinto de los primeros retos consientes a Programación con Angry Bird

**ACTIVIDAD 3: Práctica autónoma**

- Los estudiantes resuelven los retos iniciales aplicando comandos básicos de movimiento.
- El docente monitorea el progreso desde la plataforma y brinda retroalimentación oportuna.

**CIERRE**

**EVALUACIÓN Y RETROALIMENTACIÓN** (metacognición) 15 m

Retroalimentación hacemos algunas preguntas

- ¿qué dificultades y logros tuvieron al interactuar en Code.org?
- ¿Cómo me ayuda el trabajo por bloques a entender la lógica de la programación?
- ¿Qué dificultades hemos tenido?

V. ESPACIO EDUCATIVO

- Centro de cómputo

VI. MATERIALES Y RECURSOS PARA LA SESIÓN

- Proyector multimedia
- Pizarra
- Plumones
- Laptops por cada estudiante con acceso a Internet
- Plataforma CODE.ORG – Curso Express (2022)

VII. OBSERVACIONES

Arequipa, 01 abril del 2025



Prof. Raul Aymara Alfaro



I.E. GRAN PACHACUTEC  
Arequipa - Perú

## Sesión de Aprendizaje N° 03 Primer Grado

### I unidad de aprendizaje

**Título de sesión:** "Secuencia lógica: de comandos simples a la corrección de errores (debugging)"

#### I. DATOS INFORMATIVOS

Institución Educativa:	Gran Pachacútec	Área:	Educación para el trabajo
Director:	César Vásquez Meza	Docente:	Raúl Aymara Alfaro
Grado:	Primer grado	Duración:	02 horas
Sección:	A	Fecha:	08/04/2025

#### II. PROPÓSITO DE LA SESIÓN

"Desarrollamos la secuenciación (capacidad de ordenar comandos), que es la base para todo programa. Introducir la Depuración (Debugging) para identificar y corregir errores"

#### III. PROPÓSITOS DE APRENDIZAJE

COMPETENCIAS	CAPACIDADES	DESEMPEÑO PRECISADO	CRITERIOS DE EVALUACIÓN	INSTRUMENTOS DE EVALUACIÓN	EVIDENCIAS DE APRENDIZAJE
<b>Gestiona proyectos de emprendimiento o económico o social.</b>	Trabaja cooperativamente para lograr objetivos y metas	Promueve la perseverancia al enfrentar errores de código (bugs) y cumple su rol en la identificación y solución de fallos dentro de la secuencia de comandos, demostrando adaptabilidad.	1. Ordena los comandos de movimiento y dirección de manera lógica (Secuenciación). 2. Identifica y corrige errores (Depuración) en el código propio o de un par, asegurando su correcto funcionamiento.	Lista de cotejo	Secuencia Funcional en Code.org (resolución de retos) que resuelve un laberinto complejo. Lección 01 y Lección 02 de code.org.
<b>Gestiona su aprendizaje de manera autónoma</b>	Crea objetos virtuales en diversos formatos	Desarrolla procedimientos lógicos y secuenciales en la plataforma Code.org, aplicando la técnica de depuración para corregir programas de bloques gráficos y lograr la tarea requerida.	3. Muestra perseverancia, resolviendo los retos prácticos que implican fallos en la secuencia.		

#### IV. SECUENCIA DIDÁCTICA

##### INICIO

##### MOTIVACIÓN:

Se saluda cordialmente a los estudiantes y se menciona que el docente actuará como un robot y los estudiantes, agrupados en 4 grupos deben formular las órdenes precisas para que el robot llegue a la ubicación de su equipo. Se da un ejemplo "Dos pasos adelante, girar a la izquierda, avanzar 5 pasos, girar a la derecha" para llegar al pupitre.

¿En qué casos de la vida cotidiana usamos instrucciones?

##### EXPLORACIÓN DE SABERES PREVIOS:

a. Si tuvieran que dar instrucciones para hacer una receta, ¿qué pasaría si intentan hornear el pastel antes de mezclar los ingredientes?

b. ¿Por qué es tan importante el orden en el que se ejecutan las instrucciones?

##### PROBLEMATIZACIÓN O CONFLICTO: Desequilibrio cognitivo.

a. ¿Qué pasaría si intentamos hacer una receta de cocina sin seguir el orden de los pasos?

b. ¿Cómo sabríamos dónde nos equivocamos al seguir una secuencia de instrucciones?

c. ¿el ordenador ejecuta las instrucciones al azar o sigue un orden predeterminado de ejecución?

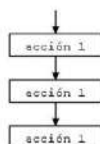
##### DESARROLLO

**GESTIÓN Y ACOMPAÑAMIENTO:** Revisión y Profundización en la Secuenciación (Conceptos y Direcciones Básicas).

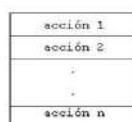


## Estructura Secuencial

### ■ Diagrama de Flujo:



### ■ Diagrama de N - S:



La secuenciación se refiere a la capacidad de ordenar correctamente las instrucciones para que el programa funcione de manera lógica. El orden de las instrucciones es crucial, ya que si se ejecutan de manera incorrecta, el resultado se ve afectado

#### ACTIVIDAD 1: Práctica con CODE.ORG

El Camino Preciso. Los estudiantes acceden a la plataforma Code.org (Lección 01. Programación con Angry Birds). Se les asigna una serie de desafíos iniciales que requieren únicamente comandos de direcciones básicas (como avanzar, girar a la izquierda, girar a la derecha).

Se pregunta: ¿Qué puedes observar en este entorno? ¿Dónde crees que podemos registrarnos?

#### ACTIVIDAD 2: Registro en el aula virtual de CODE.ORG

- El docente guía a los estudiantes para que puedan registrarse en el sitio web code.org, usando un enlace compartido mediante el aula virtual de Google classroom.

- Se enfatiza que las computadoras requieren instrucciones precisas y completas al escribir las instrucciones de forma anticipada. Los estudiantes deben usar la menor cantidad de bloques posible para lograr la ruta objetivo, reforzando el concepto de que el código debe ser eficiente.

- El docente dirige una breve reflexión sobre cómo, si se cambia el orden de dos bloques de comando (por ejemplo, "girar" y luego "avanzar" vs. "avanzar" y luego "girar"), el resultado final es distinto, demostrando que el orden en que se ejecutan las instrucciones afecta el resultado

- Los estudiantes dan las órdenes por bloques para cumplir con el reto:

#### ACTIVIDAD 3: Introducción a la Depuración (Debugging)

- La Depuración es la tarea requerida que implica depurar un conjunto incorrecto de comandos previamente dado. Esta habilidad es fundamental para la resolución de problemas.

- El Código Fallido (Modelado Guiado). docente proyecta un ejercicio sencillo de Code.org resuelto con un código que contiene un error evidente de secuenciación (un bug). Los estudiantes identifican donde se encuentra el error.



Resuelven los ejercicios de la **Lección 02 Depuración en el laberinto**: Depuran las instrucciones en la secuencia de bloques

**CIERRE**

**EVALUACIÓN Y RETROALIMENTACIÓN** (metacognición) 15 m

Retroalimentación hacemos algunas preguntas

- ¿Qué dificultades y logros tuvieron al resolver los retos de secuencialidad y depuración?
- ¿Cómo me ayuda el trabajo por bloques a entender la lógica de la programación?
- ¿Qué dificultades hemos tenido?

V. ESPACIO EDUCATIVO

- Centro de cómputo

VI. MATERIALES Y RECURSOS PARA LA SESIÓN

- Proyector multimedia
- Pizarra
- Plumones
- Plataforma CODE.ORG – Curso Express (2022)
- Laptops por cada estudiante con acceso a Internet

VII. OBSERVACIONES

Arequipa, 08 abril del 2025



I.E. GRAN PACHACUTEC  
Arequipa - Perú

## Sesión de Aprendizaje N° 04 Primer Grado

### I unidad de aprendizaje

Título de sesión: "Automatizando Pasos: Creación de Bucles por Repetición Fija"

#### I. DATOS INFORMATIVOS

Institución Educativa:	Gran Pachacútec	Área:	Educación para el trabajo
Director:	César Vásquez Meza	Docente:	Raúl Aymara Alfaro
Grado:	Primer grado	Duración:	02 horas
Sección:	A	Fecha:	15/04/2025

#### II. PROPÓSITO DE LA SESIÓN

"Aplicamos el concepto de Bucles fijos y Reconocimiento de Patrones para automatizar soluciones y reducir la redundancia en el código"

#### III. PROPÓSITOS DE APRENDIZAJE

COMPETENCIAS	CAPACIDADES	DESEMPEÑO PRECISADO	CRITERIOS DE EVALUACIÓN	INSTRUMENTOS DE EVALUACIÓN	EVIDENCIAS DE APRENDIZAJE
<b>Gestiona proyectos de emprendimiento o económico o social.</b>	Crea propuestas de valor	Plantea alternativas de propuesta de valor creativas al optimizar la eficiencia del código mediante el reconocimiento de patrones y el uso de la iteración fija (Bucles Repetir veces).	1.- Identifica correctamente los patrones o bloques de instrucciones que se repiten. 2. Utiliza el bucle 'repetir veces' de forma precisa para lograr el número exacto de repeticiones necesario. 3. Reduce la redundancia del código, demostrando un algoritmo optimizado.	Lista de cotejo	Resolución de Bucles en Code.org (resolución de retos de la lección 10).
<b>Gestiona su aprendizaje de manera autónoma</b>	Interactúa en entornos virtuales	Desarrolla secuencias lógicas, utilizando el bloque 'repetir veces' para representar y automatizar tareas repetitivas de forma eficiente en la plataforma de Code.org			

#### IV. SECUENCIA DIDÁCTICA

##### INICIO

##### MOTIVACIÓN:

Se saluda cordialmente a los estudiantes y establece los acuerdos de convivencia.

Muestra una breve animación o video del desfile de la escolta escolar por el aniversario del colegio Gran Pachacútec. Pregunta:

- ¿Cuántos pasos dan los estudiantes desde el inicio hasta el estrado?

- ¿Qué pasaría si tuviéramos que escribir cada paso uno por uno para programar un robot que marche igual?

e invita a pensar cómo podríamos hacer que el robot repita el mismo paso varias veces sin escribirlo tantas veces.

##### EXPLORACIÓN DE SABERES PREVIOS:

a. ¿Qué recuerdan del uso de comandos por bloques en Code.org en las sesiones anteriores?

b. ¿Qué sucede cuando repetimos varias veces el mismo bloque para programar una repetición de instrucciones?

c. ¿Por qué creen que en programación es importante reconocer cuando una acción se repite muchas veces?

##### PROBLEMATIZACIÓN O CONFLICTO: Desequilibrio cognitivo.

a. Si nuestro personaje robot debe avanzar 20 pasos, ¿cómo podríamos hacerlo sin escribir 20 veces "avanzar"?

b. Si en un proyecto robótico de la escuela, el robot debe regar las plantas de los jardines del colegio en 5 puntos distintos, ¿cómo podríamos programar esa tarea sin repetir el mismo comando una y otra vez?

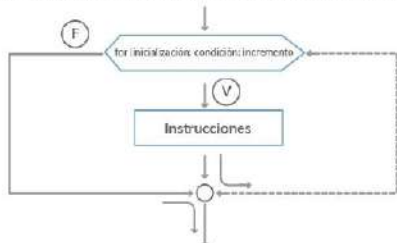
El docente presenta el título y propósito de aprendizaje: Hoy aprenderemos cómo los programadores y emprendedores usan una herramienta llamada 'bucle' para automatizar acciones que se repiten y ahorrar tiempo, esfuerzo y errores. Así como en la vida diaria, cuando hacemos algo muchas veces, también podemos automatizar tareas en nuestros proyectos digitales



DESARROLLO

**GESTIÓN Y ACOMPAÑAMIENTO: Comprensión de la estructura de un Bucle.**

El docente explica mediante un diagrama la estructura repetir (for). Presenta una forma gráfica como se puede representar un Bucle.



Los estudiantes, con el uso de su cuenta institucional, ingresan al aula virtual de classroom y se redirigen mediante el link compartido al sitio virtual de code.org. lección 10: Bucles con Rey y BB-8.

**ACTIVIDAD 1: Práctica con CODE.ORG**

Los estudiantes resuelven los 09 retos planteados en la lección 10 de Bucles:

Se pregunta: ¿Qué nos plantea el desafío del ejercicio 10? ¿Cuántos bloques podemos usar para resolverlo?

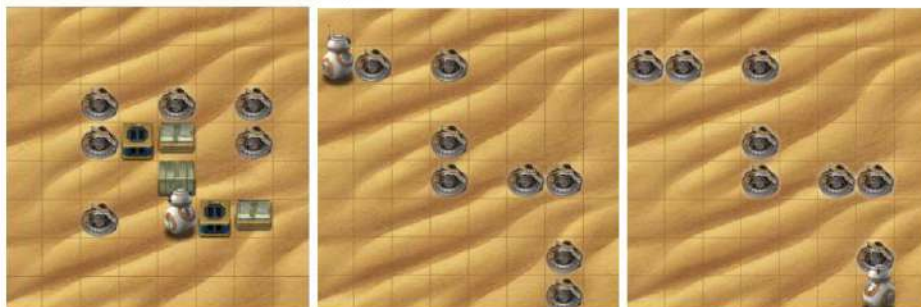
Los estudiantes con el uso de su creatividad e interacción de sus compañeros, organizan los bloques de instrucciones, usando Bucles para resolver el desafío con el uso de solo 10 bloques.



I.E. GRAN PACHACUTEC  
Arequipa - Perú

**ACTIVIDAD 2: Aftanzan la habilidad del uso de Bucles en CODE.ORG**

- Los estudiantes resuelven los retos 11,12,13 y 14.



El docente monitorea y apoya a los estudiantes que muestran dificultades.

**ACTIVIDAD 3: Verificación del avance de los estudiantes y el nivel de logro**

- Se verifica el desarrollo de los retos de manera general y se identifica los ejercicios que tuvieron más dificultades, los que se resuelven de manera colaborativa.

**CIERRE**

**EVALUACIÓN Y RETROALIMENTACIÓN** (metacognición) 15 m

Retroalimentación hacemos algunas preguntas

- ¿Qué dificultades y logros tuvieron al resolver los retos de Bucle?
- ¿Cómo podríamos estas habilidades para identificar patrones y programar acciones repetitivas?
- ¿Cómo hemos aprendido? ¿En que situaciones podemos usar estas habilidades?

V. ESPACIO EDUCATIVO

- Centro de cómputo

VI. MATERIALES Y RECURSOS PARA LA SESIÓN

- Proyector multimedia
- Pizarra
- Plumones
- Plataforma CODE.ORG – Curso Express (2022)
- Laptops por cada estudiante con acceso a Internet

VII. OBSERVACIONES

Arequipa, 15 abril del 2025

Prof. Raúl Aymara Alfaro



I.E. GRAN PACHACUTEC  
Arequipa - Perú

## Sesión de Aprendizaje N° 05 Primer Grado

### I unidad de aprendizaje

Título de sesión: "Aplicamos estructuras anidadas de Bucles"

#### I. DATOS INFORMATIVOS

Institución Educativa:	Gran Pachacútec	Área:	Educación para el trabajo
Director:	César Vásquez Meza	Docente:	Raúl Aymara Alfaro
Grado:	Primer grado	Duración:	02 horas
Sección:	A	Fecha:	22/04/2025

#### II. PROPÓSITO DE LA SESIÓN

"Comprendemos el funcionamiento y la utilidad de las estructuras anidadas (bucles dentro de bucles) en la optimización de algoritmos y aprendan a completar códigos incompletos dentro de un entorno virtual como Code.org, fortaleciendo así sus habilidades técnicas, lógicas y de resolución de problemas"

#### III. PROPÓSITOS DE APRENDIZAJE

COMPETENCIAS	CAPACIDADES	DESEMPEÑO PRECISADO	CRITERIOS DE EVALUACIÓN	INSTRUMENTOS DE EVALUACIÓN	EVIDENCIAS DE APRENDIZAJE
<b>Gestiona proyectos de emprendimiento o económico o social.</b>	Crea propuestas de valor	Emplea habilidades técnicas complejas al diseñar y construir estructuras anidadas (bucles dentro de otros bucles) para optimizar la solución algorítmica de problemas con existencia de anidamiento.	1. Implementa correctamente bucles anidados para resolver secuencias complejas. 2. Integra bloques faltantes en estructuras preexistentes, demostrando habilidad de completamiento.	Lista de cotejo	Resolución de Bucles anidados en la plataforma virtual code.org, Lección 12 Bucles anidados en Laberinto.
<b>Gestiona su aprendizaje de manera autónoma</b>	Gestiona información del entorno virtual	Desarrolla secuencias lógicas que incluyen el uso de estructuras anidadas y demuestra la capacidad de completamiento de código en un entorno virtual para generar soluciones modulares.	3. Justifica la necesidad del anidamiento para optimizar la resolución de tareas.		

#### IV. SECUENCIA DIDÁCTICA

##### INICIO

Se saluda cordialmente a los estudiantes y establece los acuerdos de convivencia.

##### MOTIVACIÓN:

El docente muestra en la pantalla una imagen de un tablero de ajedrez. Se plantea una breve historia:

"Imagina que los estudiantes de la I.E. Gran Pachacútec están diseñando un robot para pintar un tablero de ajedrez en el patio del colegio para el aniversario. Si el robot puede pintar solo un cuadro por vez, ¿cuántas veces tendría que repetir los mismos movimientos?" "¿Qué pasaría si pudiéramos decirle al robot que repita un grupo de repeticiones?", estas preguntas buscan introducir visualmente y de forma lúdica la idea de repeticiones dentro de repeticiones, base del anidamiento

##### EXPLORACIÓN DE SABERES PREVIOS:

- ¿Recuerdan cómo usamos los bucles simples ("repetir veces") en la sesión anterior?
- ¿Qué ventajas tenía usar un bucle en lugar de repetir instrucciones manualmente?
- ¿En qué tipos de actividades diarias ustedes repiten una misma acción varias veces dentro de otra (por ejemplo, al limpiar, armar algo o practicar un deporte)?

##### PROBLEMATIZACIÓN O CONFLICTO: Desequilibrio cognitivo.

- Si el robot debe pintar 10 filas de 10 cuadros, ¿cómo podríamos hacerlo sin escribir 100 comandos?



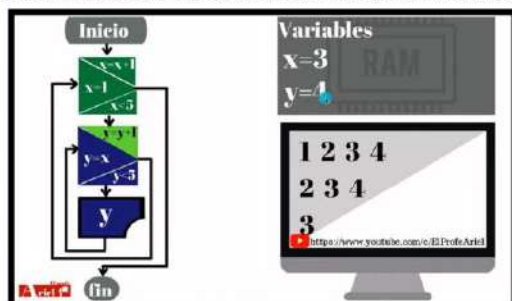
I.E. GRAN PACHACUTEC  
Arequipa - Perú

El docente presenta el título y propósito de aprendizaje: Hoy aprenderemos a que una repetición puede contener otra dentro, y cómo podemos completar códigos preexistentes para resolver problemas más complejos y eficientes, igual que cuando organizamos un trabajo grande en tareas más pequeñas.

**DESARROLLO**

**GESTIÓN Y ACOMPAÑAMIENTO:** Comprensión de la estructura de bucles anidados.

El docente muestra un video donde se representa gráficamente la ejecución de los ciclos anidados:.



<https://www.youtube.com/watch?v=FVQmMRV78qU>

Los estudiantes, con el uso de su cuenta institucional, ingresan al aula virtual de classroom y se redirigen mediante el link compartido al sitio virtual de code.org. lección 12: Bucles anidados en Laberinto.

**ACTIVIDAD 1: Práctica con CODE.ORG de Actividad de modelado de anidamiento: simular tareas que implican repetición de repeticiones**

Los estudiantes resuelven los 09 retos planteados en la lección 12 de Bucles anidados:

Se pregunta: ¿Qué nos plantea el desafío del ejercicio 10? ¿Cuántos bloques podemos usar para resolverlo?



I.E. GRAN PACHACUTEC  
Arequipa - Perú

Los estudiantes con el uso de su creatividad e interacción de sus compañeros, organizan los bloques de instrucciones, usando Bucles para resolver el desafío con el uso de solo 7 bloques.

**ACTIVIDAD 2: Afianzan la habilidad del uso de Bucles anidados en CODE.ORG**

- Los estudiantes resuelven los retos 11, 12 y 13.



El docente monitorea y apoya a los estudiantes que muestran dificultades.

**ACTIVIDAD 3: Verificación del avance de los estudiantes y el nivel de logro**

- Se verifica el desarrollo de los retos de manera general y se identifica los ejercicios que tuvieron más dificultades, los que se resuelven de manera colaborativa.

**CIERRE**

**EVALUACIÓN Y RETROALIMENTACIÓN** (metacognición) 15 m

Retroalimentación hacemos algunas preguntas

- ¿Dónde aplicaron el bucle anidado?
- ¿Cómo redujeron el número de comandos?
- ¿Qué errores se presentaron al anidar los bucles?
- ¿Cómo el uso de bucles anidados me ayudó a resolver el problema de forma más rápida y ordenada?
- ¿Qué diferencia encuentro entre usar un bucle simple y usar bucles dentro de otros?

V. ESPACIO EDUCATIVO

- Centro de cómputo

VI. MATERIALES Y RECURSOS PARA LA SESIÓN

- Proyector multimedia
- Pizarra
- Plumones
- Plataforma CODE.ORG – Curso Express (2022)
- Laptops por cada estudiante con acceso a Internet

VII. OBSERVACIONES

Arequipa, 22 abril del 2025



Prof. Raul Aymara Alfaro



I.E. GRAN PACHACUTEC  
Arequipa - Perú

## Sesión de Aprendizaje N° 06 Primer Grado

### I unidad de aprendizaje

Título de sesión: "Aplicamos decisiones Lógicas, implementando del Condicional Simple (If)"

#### I. DATOS INFORMATIVOS

Institución Educativa:	Gran Pachacútec	Área:	Educación para el trabajo
Director:	César Vásquez Meza	Docente:	Raúl Aymara Alfaro
Grado:	Primer grado	Duración:	02 horas
Sección:	A	Fecha:	29/04/2025

#### II. PROPÓSITO DE LA SESIÓN

"Aplicamos la Lógica Condicional Simple para que el programa ejecute comandos solo si una condición es verdadera, introduciendo el concepto binario de una condición (verdadero/falso) y su relación con la toma de decisiones en algoritmos."

#### III. PROPÓSITOS DE APRENDIZAJE

COMPETENCIAS	CAPACIDADES	DESEMPEÑO PRECISADO	CRITERIOS DE EVALUACIÓN	INSTRUMENTOS DE EVALUACIÓN	EVIDENCIAS DE APRENDIZAJE
<b>Gestiona proyectos de emprendimiento o económico o social.</b>	Crea propuestas de valor	Plantea alternativas de propuesta de valor al diseñar un algoritmo que incluye la lógica condicional simple (If) para programar decisiones basadas en principios de causa y efecto, reflexionando sobre cómo la toma de decisiones oportunas mejora la funcionalidad de una propuesta tecnológica o de emprendimiento.	<ol style="list-style-type: none"> <li>1. Implementa correctamente el condicional simple (If) para que una acción se ejecute solo si la condición es verdadera.</li> <li>2. Demuestra comprensión del concepto binario de una condición (verdadero o falso).</li> <li>3. Resuelve un problema aplicando la lógica condicional para la toma de decisiones del programa.</li> <li>4. Explica cómo la toma de decisiones programada aporta valor a su propuesta o solución tecnológica.</li> </ol>	Lista de cotejo	Resolución de condicional simple (if) code.org, <b>Lección 14</b> Mirando hacia el Futuro con Minecraft y <b>Lección 15</b> Condicionales (Si/sino) con Abejas.
<b>Gestiona su aprendizaje de manera autónoma</b>	Gestiona información del entorno virtual	Desarrolla secuencias lógicas implementando el condicional simple (If) para que el programa ejecute una acción únicamente si se cumple la condición establecida.			

#### IV. SECUENCIA DIDÁCTICA

##### INICIO

Se saluda cordialmente a los estudiantes y establece los acuerdos de convivencia.

##### MOTIVACIÓN:

- El docente pregunta ¿Cómo le decimos a un programa que haga una cosa, pero solo si se cumple una regla específica? ¿Cómo se asemeja esto a nuestras decisiones diarias, como decir: ¿hace sol?, entonces llevo lentes" o "¿está lloviendo?, entonces debo llevar paraguas", con ello se busca introducir el pensamiento lógico binario (condición verdadera/falsa) y su relación con la toma de decisiones humanas.

- El docente plantea una historia cercana: "Imagina que tenemos un robot en el colegio Gran Pachacutec que ayuda a mantener limpio el patio. Debe recoger papeles del suelo, pero solo cuando detecte uno. ¿Cómo podríamos decirle que recoja el papel solo si realmente hay uno frente a él?".

##### EXPLORACIÓN DE SABERES PREVIOS:

a. ¿Qué significa para ustedes "tomar una decisión"?

b. ¿En qué momentos del día ustedes toman decisiones dependiendo de algo que pasa (por ejemplo: ¿si llueve, si llega el bus)?



c. ¿Recuerdan si en sesiones anteriores el personaje del programa repetía siempre lo mismo o si alguna vez debía “decidir” ¿qué hacer?

**PROBLEMATIZACIÓN O CONFLICTO:** Desequilibrio cognitivo.

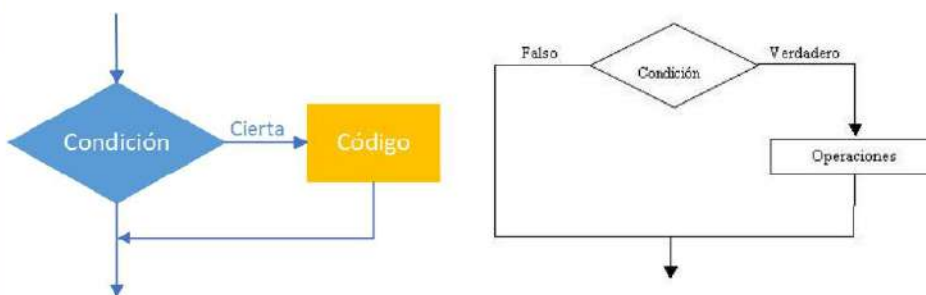
a. Si construimos un robot que recoge objetos sin verificar si existen, ¿qué pasaría? ¿Haría su trabajo correctamente o cometería errores?

**DESARROLLO**

**GESTIÓN Y ACOMPAÑAMIENTO:** Comprensión de la estructura condicional simple.

El docente proyecta una presentación multimedia donde se representa de manera gráfica ejemplos de la estructura condicional:

Estructura Condicional simple



Los estudiantes, con el uso de su cuenta institucional, ingresan al aula virtual de classroom y se redirigen mediante el link compartido al sitio virtual de code.org. lección 15: Bucles anidados en Laberinto.

**ACTIVIDAD 1: Práctica con CODE.ORG de Actividad de modelado de las estructuras de condicional**

Los estudiantes resuelven los 09 retos planteados en la lección 14 de condicionales.



El docente explica el concepto de condición lógica (verdadero/falso) y su relación con la toma de decisiones.

Se pregunta: ¿Qué nos plantea el desafío del ejercicio 11? ¿Cuántos bloques podemos usar para resolverlo?

Los estudiantes resuelven los retos 12 y 13

**ACTIVIDAD 2: Resuelven el desafío planteado en CODE.ORG**

Los estudiantes con el uso de su creatividad e interacción de sus compañeros, resuelven el desafío planteado en las actividades 3 y 4 de la Lección 15: Condicionales (Sí/no) con abeja:



I.E. GRAN PACHACUTEC  
Arequipa - Perú



**ACTIVIDAD 3: Afianzan la habilidad del uso de condicionales en CODE.ORG**

- Los estudiantes resuelven los retos 5, 6 y 7.



El docente monitorea y apoya a los estudiantes que muestran dificultades.

**ACTIVIDAD 4: Verificación del avance de los estudiantes y el nivel de logro en la clase de code.org**

- Se verifica el desarrollo de los retos de manera general y se identifica los ejercicios que tuvieron más dificultades, los que se resuelven de manera colaborativa.

**CIERRE**

**EVALUACIÓN Y RETROALIMENTACIÓN** (metacognición) 15 m

Retroalimentación hacemos algunas preguntas

- ¿Por qué es importante que un programa pueda decidir qué hacer según las condiciones?
- ¿Qué pasaría si todos los programas hicieran siempre lo mismo sin condiciones?
- ¿Cómo se relaciona este tipo de decisiones con las que tomamos en la vida real o en un proyecto de emprendimiento??

**V. ESPACIO EDUCATIVO**

- Centro de cómputo

**VI. MATERIALES Y RECURSOS PARA LA SESIÓN**

- Proyector multimedia
- Pizarra
- Plumones
- Plataforma CODE.ORG – Curso Express (2022)
- Laptops por cada estudiante con acceso a Internet

**VII. OBSERVACIONES**

Arequipa, 29 abril del 2025

Prof. Raúl Aymara Alfaro



I.E. GRAN PACHACUTEC  
Arequipa - Perú

## Sesión de Aprendizaje N° 07 Primer Grado

### I unidad de aprendizaje

Título de sesión: "Resolvemos situaciones diversas con condicionales compuestas (if/else)"

#### I. DATOS INFORMATIVOS

Institución Educativa:	Gran Pachacútec	Área:	Educación para el trabajo
Director:	César Vásquez Meza	Docente:	Raúl Aymara Alfaro
Grado:	Primer grado	Duración:	02 horas
Sección:	A	Fecha:	06/05/2025

#### II. PROPÓSITO DE LA SESIÓN

"Comprendemos y aplicamos la estructura If/Else para crear bifurcaciones explícitas en el flujo del algoritmo, de modo que el programa ejecute una acción cuando la condición sea verdadera y otra cuando sea falsa, garantizando decisiones lógicas en todo momento."

#### III. PROPÓSITOS DE APRENDIZAJE

COMPETENCIAS	CAPACIDADES	DESEMPEÑO PRECISADO	CRITERIOS DE EVALUACIÓN	INSTRUMENTOS DE EVALUACIÓN	EVIDENCIAS DE APRENDIZAJE
<b>Gestiona proyectos de emprendimiento o económico o social.</b>	Trabaja cooperativamente para lograr objetivos y metas	Trabaja cooperativamente para determinar los posibles casos o alternativas de una situación problema y diseña, mediante programación, un algoritmo con bifurcación (If/Else) que responda de forma adecuada ante cada posibilidad, asegurando la eficacia del proyecto o tarea.	1. Utiliza correctamente el condicional compuesto (If/Else) para gestionar las dos posibles salidas de una condición. 2. Demuestra comprensión de la lógica de la bifurcación, garantizando que el programa siempre ejecute una acción, sea la condición verdadera o falsa. 3. Identifica y representa todos los casos posibles del problema, asegurando una solución completa y funcional.	Lista de cotejo	Resolución de Condicionales compuestos en la plataforma virtual code.org, Lección 15 Condicionales (Si/sino) con Abejas.
<b>Gestiona su aprendizaje de manera autónoma</b>	Gestiona información del entorno virtual	Desarrolla secuencias lógicas implementando el Condicional Compuesto (If/Else) para generar bifurcaciones en el flujo del programa, manejando adecuadamente las alternativas posibles (verdadero/falso).			

#### IV. SECUENCIA DIDÁCTICA

##### INICIO

Se saluda cordialmente a los estudiantes y establece los acuerdos de convivencia.

##### MOTIVACIÓN:

El docente inicia mostrando un breve video donde un robot aspiradora "decide" si girar o seguir recto según detecte un obstáculo. Luego pregunta: "¿Cómo logra el robot decidir entre dos caminos diferentes? ¿Creen que hay una orden especial que le dice qué hacer si algo no sucede como esperaba?"

##### EXPLORACIÓN DE SABERES PREVIOS:

- ¿Qué recuerdan del condicional simple If que aprendimos antes?
- ¿Qué hacía el programa cuando la condición no se cumplía?
- ¿Qué pasaba si necesitábamos que el programa haga algo en ese caso?

**PROBLEMATIZACIÓN O CONFLICTO:** Desequilibrio cognitivo.



I.E. GRAN PACHACUTEC  
Arequipa - Perú

a. ¿Qué pasa si un robot del programa no cumple la condición y no tiene ninguna instrucción alternativa? ¿Se queda sin hacer nada?  
¿Cómo podríamos decirle qué hacer **cuando no se cumple** la condición?

**DESARROLLO**

**GESTIÓN Y ACOMPAÑAMIENTO:** Comprensión de la estructura de condicionales compuestos.  
El docente muestra una presentación multimedia para ilustrar la estructura condicional doble.



Los estudiantes, con el uso de su cuenta institucional, ingresan al aula virtual de classroom y se redirigen mediante el link compartido al sitio virtual de code.org. lección 15: Condicionales (Si/sino) con Abejas

**ACTIVIDAD 1:** Práctica con CODE.ORG de Actividad de estructuras de condicional compuesto:

Los estudiantes se le presenta el reto del ejercicio 09 y 10 de la Lección 15: Condicionales (Si/sino) con Abejas:



Se pregunta: ¿Qué nos plantea el desafío del ejercicio 09? ¿Cuántos bloques podemos usar para resolverlo? ¿Cuál es la diferencia de la condicional SI/SINO con la condicional simple?

Los estudiantes con el uso de su creatividad e interacción de sus compañeros, organizan los bloques de instrucciones, usando Bucles Mientras para resolver el desafío con el uso de solo 7 bloques.



**ACTIVIDAD 2:** Afianzan la habilidad del uso de Bucles anidados en CODE.ORG

- Los estudiantes resuelven los retos 12 y 13.



I.E. GRAN PACHACUTEC  
Arequipa - Perú



El docente monitorea y apoya a los estudiantes que muestran dificultades.

**ACTIVIDAD 3: Verificación del avance de los estudiantes y el nivel de logro**

- Se verifica el desarrollo de los retos de manera general y se identifica los ejercicios que tuvieron más dificultades, los que se resuelven de manera colaborativa.

**CIERRE**

**EVALUACIÓN Y RETROALIMENTACIÓN** (metacognición) 15 m

Retroalimentación hacemos algunas preguntas

- ¿Por qué es importante que un programa tenga una respuesta incluso cuando la condición no se cumple?
- ¿Cómo se parece el If/Else a las decisiones que tomamos en la vida real?
- En un proyecto de emprendimiento, ¿qué beneficios tiene pensar en alternativas o "planes B"?

V. ESPACIO EDUCATIVO

- Centro de cómputo

VI. MATERIALES Y RECURSOS PARA LA SESIÓN

- Proyector multimedia
- Pizarra
- Plumones
- Plataforma CODE.ORG – Curso Express (2022)
- Laptops por cada estudiante con acceso a Internet

VII. OBSERVACIONES

Arequipa, 06 mayo del 2025.

Prof. Raul Aymara Alfaro



## Sesión de Aprendizaje N° 08 Primer Grado

### I unidad de aprendizaje

Título de sesión: "Programando con la Estructura "Mientras Que" (While)"

#### I. DATOS INFORMATIVOS

Institución Educativa:	Gran Pachacútec	Área:	Educación para el trabajo
Director:	César Vásquez Meza	Docente:	Raúl Aymara Alfaro
Grado:	Primer grado	Duración:	02 horas
Sección:	A	Fecha:	13/05/2025

#### II. PROPÓSITO DE LA SESIÓN

"Usamos la estructura "Mientras que" (While) para programar repeticiones continuas controladas por una condición, diferenciándola de la bifurcación (If/Else) y reforzando el dominio del control de flujo en los algoritmos."

#### III. PROPÓSITOS DE APRENDIZAJE

COMPETENCIAS	CAPACIDADES	DESEMPEÑO PRECISADO	CRITERIOS DE EVALUACIÓN	INSTRUMENTOS DE EVALUACIÓN	EVIDENCIAS DE APRENDIZAJE
<b>Gestiona proyectos de emprendimiento o económico o social.</b>	Aplica habilidades técnicas	Emplea habilidades técnicas de control de flujo al diseñar un algoritmo que utiliza la estructura "Mientras que" (While) para lograr una repetición continua hasta que se rompa la condición, asegurando que el programa mantenga su funcionamiento eficiente y controlado.	1. Implementa correctamente el bucle "Mientras que" (While) en su programa. 2. Diferencia el bucle de iteración continua (While) de las estructuras condicionales (If/Else). 3. Garantiza que el programa se repita solo mientras la condición sea verdadera y se detenga adecuadamente.	Lista de cotejo	Utiliza el bloque <i>Mientras</i> para resolver los retos en la plataforma virtual code.org, lección 16: "Bucles "mientras" en Granjera".
<b>Gestiona su aprendizaje de manera autónoma</b>	Gestiona información del entorno virtual	Desarrolla secuencias lógicas implementando la estructura "Mientras que" (While) para la iteración continua, reconociendo y ajustando la condición de término para mantener el control del flujo y evitar repeticiones infinitas.			

#### IV. SECUENCIA DIDÁCTICA

##### INICIO

Se saluda cordialmente a los estudiantes y establece los acuerdos de convivencia.

##### MOTIVACIÓN:

El docente plantea una situación simple: "Cuando lavamos la ropa, seguimos enjuagando **mientras** el agua esté sucia. En cuanto el agua sale limpia, detenemos el enjuague. ¿Qué pasaría si no observáramos la condición del agua y repitiéramos la acción sin parar?" Estas situaciones introducen la idea de repetición controlada mediante una condición, base del bucle **While**.

##### EXPLORACIÓN DE SABERES PREVIOS:

- ¿Recuerdan qué hacía el bloque "Repetir veces"?
- ¿Y las "condicionales compuestas"? ¿Cuál era la diferencia entre ambos?

##### PROBLEMATIZACIÓN O CONFLICTO: Desequilibrio cognitivo.

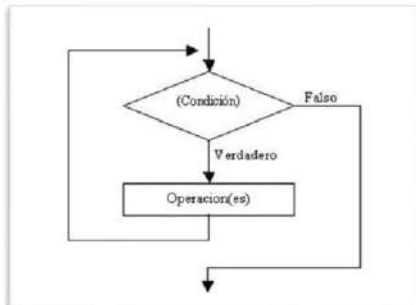
- "¿Cómo podríamos programar una acción que se repita constantemente, pero solo mientras se cumpla una condición, sin que el programa se vuelva infinito? ¿Qué pasaría si olvidamos decirle cuándo detenerse?"



I.E. GRAN PACHACUTEC  
Arequipa - Perú

**DESARROLLO**

**GESTIÓN Y ACOMPAÑAMIENTO:** Comprensión de la estructura de bucles mientras se cumpla una condición.  
El docente muestra un video donde se representa gráficamente la ejecución de la estructura mientras:



Los estudiantes, con el uso de su cuenta institucional, ingresan al aula virtual de classroom y se redirigen mediante el link compartido al sitio virtual de code.org. lección 16: "Bucles "mientras" en Granjera".

**ACTIVIDAD 1: Práctica con CODE.ORG de Actividad de estructuras repetitivas mientras:**

Los estudiantes se le presenta el reto del ejercicio 06 de la lección 16: "Bucles "mientras" en Granjera":



Seguidamente los estudiantes combinan sus conocimientos de bucles fijos con los bloques mientras en el ejercicio 07, 08 y 09



Se pregunta: ¿Qué nos plantea el desafío del ejercicio 09? ¿Cuántos bloques podemos usar para resolverlo?

Los estudiantes con el uso de su creatividad e interacción de sus compañeros, organizan los bloques de instrucciones, usando Bucles Mientras para resolver el desafío con el uso de solo 7 bloques.



I.E. GRAN PACHACUTEC  
Arequipa - Perú

cuando se ejecuta

- repetir 6 veces
- hacer mientras haya camino adelante
  - hacer avanzar
- mientras hay un agujero
  - hacer llenar 1
- girar a la derecha

**ACTIVIDAD 2: Afianzan la habilidad del uso de Bucles anidados en CODE.ORG**

- Los estudiantes resuelven los retos 11,12 y 13.



El docente monitorea y apoya a los estudiantes que muestran dificultades.

**ACTIVIDAD 3: Verificación del avance de los estudiantes y el nivel de logro**

- Se verifica el desarrollo de los retos de manera general y se identifica los ejercicios que tuvieron más dificultades , los que se resuelven de manera colaborativa.

**CIERRE**

**EVALUACIÓN Y RETROALIMENTACIÓN** (metacognición) 15 m

- ¿Qué pasaría si olvidamos definir una condición de salida en el bucle "Mientras que"?
- ¿Por qué es importante controlar cuándo termina una acción repetitiva?
- En la vida real o en un emprendimiento, ¿en qué situaciones repetimos una acción mientras algo siga sucediendo? (ej. "Mientras haya demanda, seguimos produciendo").

V. ESPACIO EDUCATIVO

- Centro de cómputo

VI. MATERIALES Y RECURSOS PARA LA SESIÓN

- Proyector multimedia
- Plataforma CODE.ORG – Curso Express (2022)
- Laptops por cada estudiante con acceso a Internet

VII. OBSERVACIONES

Arequipa, 13 mayo del 2025

Prof. Raúl Aymara Alfaro



I.E. GRAN PACHACUTEC  
Arequipa - Perú

## Sesión de Aprendizaje N° 09 Primer Grado

### I unidad de aprendizaje

Título de sesión: "Aplicamos Iteración y Bucles con Condición de Término"

#### I. DATOS INFORMATIVOS

Institución Educativa:	Gran Pachacútec	Área:	Educación para el trabajo
Director:	César Vásquez Meza	Docente:	Raúl Aymara Alfaro
Grado:	Primer grado	Duración:	02 horas
Sección:	A	Fecha:	20/05/2025

#### II. PROPÓSITO DE LA SESIÓN

"Exploramos y aplicamos los bucles controlados por condición Mientras, comprendiendo que la iteración se detiene al cumplirse una condición específica. Desarrollar pensamiento lógico y trabajo cooperativo para crear algoritmos eficientes que respondan dinámicamente a los cambios del entorno."

#### III. PROPÓSITOS DE APRENDIZAJE

COMPETENCIAS	CAPACIDADES	DESEMPEÑO PRECISADO	CRITERIOS DE EVALUACIÓN	INSTRUMENTOS DE EVALUACIÓN	EVIDENCIAS DE APRENDIZAJE
<b>Gestiona proyectos de emprendimiento o económico o social.</b>	Trabaja cooperativamente para lograr objetivos y metas	Propone acciones que debe realizar el equipo basadas en el diseño de un algoritmo que utiliza un bucle controlado por condición (Repetir hasta) para gestionar la finalización de una tarea de forma automática, demostrando cooperación y comunicación en la resolución de problemas.	1. Diferencia correctamente el bucle fijo del bucle condicional. 2. Implementa el bloque <i>Repetir hasta</i> para detener el programa al alcanzar una condición de término. 3. Diseña un algoritmo que gestiona la iteración mediante condiciones. 4. Participa activamente en el trabajo en equipo para depurar y optimizar su código.	Lista de cotejo	Utiliza el bloque <i>repetir hasta</i> para resolver los retos en la plataforma virtual code.org, lección 18: "Bucles "hasta" en Laberinto".
<b>Gestiona su aprendizaje de manera autónoma</b>	Gestiona información del entorno virtual	Desarrolla secuencias lógicas implementando el bucle Repetir hasta (iteración condicional) para controlar la ejecución del flujo hasta que se cumpla o deje de cumplirse una condición preestablecida.			

#### IV. SECUENCIA DIDÁCTICA

##### INICIO

Se saluda cordialmente a los estudiantes y establece los acuerdos de convivencia.

##### MOTIVACIÓN:

El docente presenta un caso cercano: "Imagina que programamos un robot para limpiar los pasillos del colegio Gran Pachacutec. No sabemos cuántos metros tiene que recorrer, porque cada día el recorrido cambia. ¿Cómo podríamos decirle que siga avanzando solo mientras haya camino libre?"

##### EXPLORACIÓN DE SABERES PREVIOS:

- ¿Recuerdan qué hacía el bloque "Mientras"?
- ¿Y el "mientras"? ¿Cuál era la diferencia entre ambos?
- ¿Qué pasaría si no sabemos cuántas veces debemos repetir una acción? ¿Cómo podríamos resolverlo?

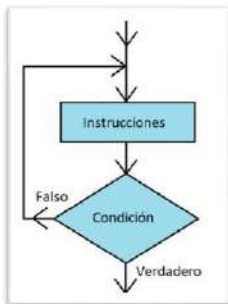
##### PROBLEMATIZACIÓN O CONFLICTO: Desequilibrio cognitivo.

- Si no sabemos cuántos pasos debe dar un robot, ¿podemos usar el bucle Repetir X veces? ¿Por qué sí o por qué no?
- ¿Cómo podríamos hacer que el robot siga moviéndose hasta que ya no haya camino sin decirle cuántas veces debe repetir?



DESARROLLO

**GESTIÓN Y ACOMPAÑAMIENTO:** Comprensión de la estructura de bucles hasta que no se cumpla una condición.  
El docente muestra un video donde se representa gráficamente la ejecución de la estructura *repetir hasta*:



Los estudiantes, con el uso de su cuenta institucional, ingresan al aula virtual de classroom y se redirigen mediante el link compartido al sitio virtual de code.org. lección 18: "Bucles "hasta" en Laberinto"

**ACTIVIDAD 1: Práctica con CODE.ORG de Actividad de estructuras repetitivas hasta:**

Los estudiantes acceden a los retos 4,5,6,7 y 8 de la lección 18: "Bucles "hasta" en Laberinto" y resuelven con los bloques "repetir hasta"



```

cuando se ejecuta
  repetir hasta [0]
  hacer
    avanzar
    girar a la derecha
    avanzar
    girar a la izquierda
  
```



Se pregunta: ¿Qué nos plantea el desafío del ejercicio 09? ¿Cuántos bloques podemos usar para resolverlo?



```

cuando se ejecuta
  girar a la derecha
  girar a la derecha
  avanzar
  girar a la derecha
  repetir hasta
  hacer
    mientras haya camino adelante
    hacer
      avanzar
    girar a la izquierda
  
```

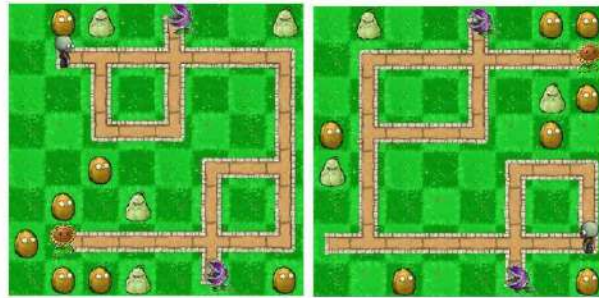
Los estudiantes con el uso de su creatividad e interacción de sus compañeros, organizan los bloques de instrucciones, usando Bucles "hasta" para resolver el desafío con el uso de solo 9 bloques de instrucciones como máximo.

**ACTIVIDAD 2: Afianzan la habilidad del uso de Bucles anidados en CODE.ORG**

- Los estudiantes resuelven los retos 10 y 11.



I.E. GRAN PACHACUTEC  
Arequipa - Perú



El docente monitorea y apoya a los estudiantes que muestran dificultades.

**ACTIVIDAD 3: Verificación del avance de los estudiantes y el nivel de logro**

- Se verifica el desarrollo de los retos de manera general y se identifica los ejercicios que tuvieron más dificultades, los que se resuelven de manera colaborativa.

**CIERRE**

**EVALUACIÓN Y RETROALIMENTACIÓN** (metacognición) 15 m

Retroalimentación hacemos algunas preguntas

- ¿Qué recuerdan de los bucles que aprendimos en sesiones anteriores?
- ¿Cómo funcionaba el bloque Repetir X veces? ¿Qué controlaba la cantidad de repeticiones?
- ¿Qué pasaría si no sabemos cuántas veces debemos repetir una acción? ¿Cómo podríamos resolverlo?

V. ESPACIO EDUCATIVO

- Centro de cómputo

VI. MATERIALES Y RECURSOS PARA LA SESIÓN

- Proyector multimedia
- Pizarra
- Plumones
- Plataforma CODE.ORG – Curso Express (2022)
- Laptops por cada estudiante con acceso a Internet

VII. OBSERVACIONES

Arequipa, 20 de mayo del 2025



Prof. Raúl Aymara Alfaro



I.E. GRAN PACHACUTEC  
Arequipa - Perú

## Sesión de Aprendizaje N° 10 Primer Grado

### I unidad de aprendizaje

#### Título de sesión: "Creando Funciones Reutilizables"

##### I. DATOS INFORMATIVOS

Institución Educativa:	Gran Pachacútec	Área:	Educación para el trabajo
Director:	César Vásquez Meza	Docente:	Raúl Aymara Alfaro
Grado:	Primer grado	Duración:	02 horas
Sección:	A	Fecha:	27/05/2025

##### II. PROPÓSITO DE LA SESIÓN

"Fomentamos la abstracción y la modularidad mediante la creación de funciones simples, que permitan agrupar comandos y reutilizar código para resolver problemas de manera más eficiente, ordenada y profesional."

##### III. PROPÓSITOS DE APRENDIZAJE

COMPETENCIAS	CAPACIDADES	DESEMPEÑO PRECISADO	CRITERIOS DE EVALUACIÓN	INSTRUMENTOS DE EVALUACIÓN	EVIDENCIAS DE APRENDIZAJE
<b>Gestiona proyectos de emprendimiento o económico o social.</b>	Evalúa los resultados del proyecto de emprendimiento.	Evalúa los resultados del proyecto al comparar la eficiencia del código modularizado con el código inicial, justificando cómo la abstracción y la reutilización de funciones optimizan el diseño algorítmico y facilitan el mantenimiento del programa.	1. Define y utiliza correctamente funciones simples (modularización). 2. Aplica la abstracción al nombrar las funciones según su propósito específico. 3. Reutiliza el código mediante el llamado de funciones, logrando un programa más corto, ordenado y comprensible.	Lista de cotejo	Utiliza el bloque <i>Hacer algo</i> para resolver los retos en la plataforma virtual code.org, lección 21: "Funciones en cosechadora".
<b>Gestiona su aprendizaje de manera autónoma</b>	Crea objetos virtuales en diversos formatos	Desarrolla secuencias lógicas y crea funciones simples que encapsulan fragmentos de código, aplicando los principios de modularización y abstracción para reutilizar código y simplificar programas complejos.			

##### IV. SECUENCIA DIDÁCTICA

###### INICIO

Se saluda cordialmente a los estudiantes y establece los acuerdos de convivencia.

###### MOTIVACIÓN:

El docente muestra un video breve o ejemplo en vivo donde un robot realiza una "coreografía" repetitiva (saludar, girar, avanzar y detenerse). Luego comenta: "Cada saludo tiene 15 comandos... y el robot repite ese saludo 5 veces. ¿Qué pasaría si tuviéramos que escribir 75 comandos uno por uno? ¿Habría una manera más inteligente de hacerlo?"

###### EXPLORACIÓN DE SABERES PREVIOS:

- ¿Qué recuerdan de las estructuras que hemos usado para repetir acciones (bucles) o tomar decisiones (condicionales)?
- ¿Cómo organizaban su código cuando necesitaban repetir una misma secuencia en distintas partes?
- ¿Qué problemas surgen cuando copiamos muchas veces las mismas instrucciones?

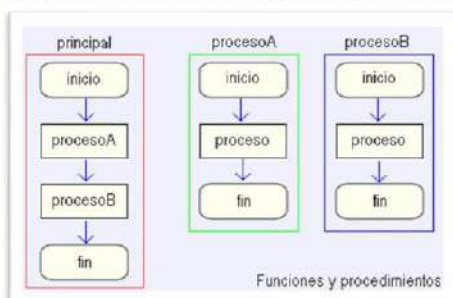
###### PROBLEMATIZACIÓN O CONFLICTO: Desequilibrio cognitivo.

- "Si tuviéramos que usar un mismo conjunto de 15 comandos varias veces, ¿es eficiente escribirlo cada vez? ¿Existe una forma de agruparlos bajo un solo nombre para llamarlos cuando los necesitemos?"



DESARROLLO

GESTIÓN Y ACOMPAÑAMIENTO: Comprensión de la estructura de la representación de funciones.



Los estudiantes, con el uso de su cuenta institucional, ingresan al aula virtual de classroom y se redirigen mediante el link compartido al sitio virtual de code.org. lección 18: "Bucles "hasta" en Laberinto"

**ACTIVIDAD 1: Práctica con CODE.ORG de Actividad de estructuras repetitivas hasta:**

Los estudiantes acceden a los retos de la 1 hasta las 10 de la lección 21: "Funciones en cosechadora" y resuelven con la creación de funciones en cada caso que lo requiera:



Se pregunta: ¿Qué nos plantea el desafío del ejercicio 11? ¿Cuántos bloques podemos usar para resolverlo?



Los estudiantes con el uso de su creatividad e interacción de sus compañeros, organizan los bloques de instrucciones, usando funciones y demás estructuras para resolver el desafío con el uso de 19 bloques de instrucciones como máximo.



I.E. GRAN PACHACUTEC  
Arequipa - Perú

**ACTIVIDAD 2: Afianzan la habilidad del uso de Bucles anidados en CODE.ORG**

- Los estudiantes resuelven los retos 12 y 13.



El docente monitorea y apoya a los estudiantes que muestran dificultades.

**ACTIVIDAD 3: Verificación del avance de los estudiantes y el nivel de logro**

- Se verifica el desarrollo de los retos de manera general y se identifica los ejercicios que tuvieron más dificultades, los que se resuelven de manera colaborativa.

**CIERRE**

**EVALUACIÓN Y RETROALIMENTACIÓN (metacognición) 15 m**

- ¿Por qué la abstracción es una de las habilidades más importantes del pensamiento computacional?
- ¿Cómo te ayuda la modularización a resolver problemas complejos, tanto en la programación como en la vida diaria?
- ¿Qué ventajas tiene reutilizar funciones en lugar de repetir código?
- ¿Qué relación encuentras entre dividir el trabajo en partes y trabajar cooperativamente en un proyecto de emprendimiento?

**V. ESPACIO EDUCATIVO**

- Centro de cómputo

**VI. MATERIALES Y RECURSOS PARA LA SESIÓN**

- Proyector multimedia
- Pizarra
- Plumones
- Laptops por cada estudiante con acceso a Internet
- Plataforma CODE.ORG – Curso Express (2022)

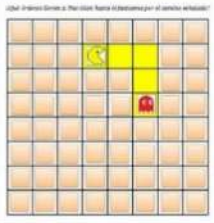
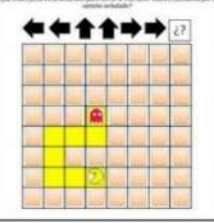
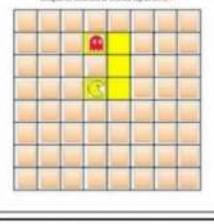
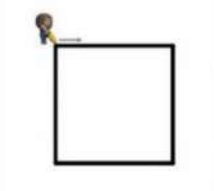
**VII. OBSERVACIONES**

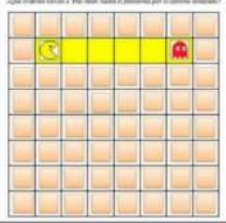









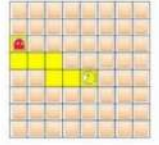
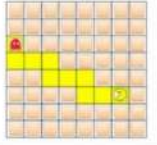


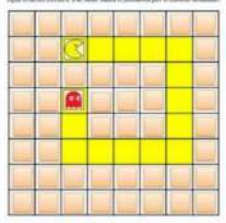


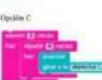

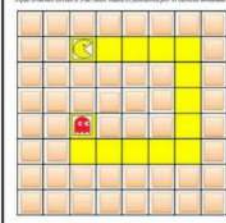


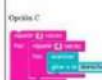

Arequipa, 27 de mayo del 2025

Prof. Raúl Aymara Alfaro

### Anexo 3

#### Instrumento de Evaluación Pre y Post-Test.

I-1 PRE TEST		Opc I-2 POST TEST		Opc	CONCEPTOS	ENTORNO/ INTERFAZ	ESTILO DE RESPUESTA	ANIDAM IENTO	TAREA REQUERIDA
1	<p>¿Qué camino debes ir? ¿Por dónde tienes el camino para el camino correcto?</p>  <p>Opción A → → → ↓</p> <p>Opción B → → ↓ ↓</p> <p>Opción C → → → ↓ ↓</p> <p>Opción D ↓ ↓ ↓ →</p>	c	1	b	Direcciones Básicas	El Laberinto	Visual por flechas	No	Secuenciación
2	<p>¿Qué camino debes ir? ¿Por dónde tienes el camino para el camino correcto?</p>  <p>Opción A →</p> <p>Opción B ←</p> <p>Opción C ↑</p> <p>Opción D ↓</p>	c	2	c	Direcciones Básicas	El Laberinto	Visual por flechas	No	Completamiento
3	<p>¿Qué camino debes ir? ¿Por dónde tienes el camino para el camino correcto?</p>  <p>avanzar → Fase A →</p> <p>girar a la derecha ↻ → Fase B →</p> <p>avanzar → Fase C →</p> <p>girar a la izquierda ↻ → Fase D →</p> <p>avanzar →</p>	b	3	d	Direcciones Básicas	El Laberinto	Visual por bloques	No	Depuración (debug)
4	<p>¿Qué camino debes ir? ¿Por dónde tienes el camino para el camino correcto?</p>  <p>Opción A → → → ↓</p> <p>Opción B → → ↓ ↓</p> <p>Opción C → → → ↓ ↓</p> <p>Opción D ↓ ↓ ↓ →</p>	c	4	d	Direcciones Básicas	El Lienzo	Visual por bloques	No	Secuenciación









I-1 PRE TEST		Opc I-2 POS TEST		Opc	CONCEPTOS	ENTORNO/ INTERFAZ	ESTILO DE RESPUESTA	ANIDAM IENTO	TAREA REQUERIDA	
5	<p>¿Qué dirección debería ir el robot para llegar al destino por el camino señalado?</p>  <p>Opción A: </p> <p>Opción B: </p> <p>Opción C: </p> <p>Opción D: </p>	a	5	<p>¿Qué dirección debería ir el robot para llegar al destino por el camino señalado?</p>  <p>Opción A: </p> <p>Opción B: </p> <p>Opción C: </p> <p>Opción D: </p>	c	Bucles; Repetir veces	El Laberinto	Visual por flechas	No	Secuenciación
6	<p>¿Cuántas veces se debe repetir la secuencia para llegar al destino por el camino señalado?</p>  <p>Opción A: <math>\times 2</math></p> <p>Opción B: <math>\times 1</math></p> <p>Opción C: <math>\times 4</math></p> <p>Opción D: <math>\times 3</math></p>	a	6	<p>¿Cuántas veces se debe repetir la secuencia para llegar al destino por el camino señalado?</p>  <p>Opción A: <math>\times 2</math></p> <p>Opción B: <math>\times 1</math></p> <p>Opción C: <math>\times 4</math></p> <p>Opción D: <math>\times 3</math></p>	d	Bucles; Repetir veces	El Laberinto	Visual por flechas	No	Completamiento
7	<p>Para que el robot alcance una vez al objetivo necesitamos un programa de control que permita de inicio, con qué parte de la siguiente secuencia de órdenes hay un error?</p>  <p>Para A:</p> <pre> repetir 2 veces   mover hacia [Derecha] 2 veces   girar a la [Derecha] por 90° grados → Fase B   mover hacia [Derecha] 2 veces → Fase C   girar a la [Derecha] por 90° grados → Fase D </pre>	c	7	<p>Para que el robot alcance una vez al objetivo necesitamos un programa de control que permita de inicio, con qué parte de la siguiente secuencia de órdenes hay un error?</p>  <p>Para A:</p> <pre> repetir 2 veces   mover hacia [Derecha] 2 veces   girar a la [Derecha] por 90° grados → Fase B   mover hacia [Derecha] 2 veces → Fase C   girar a la [Derecha] por 90° grados → Fase D </pre>	a	Bucles; Repetir veces	El Lienzo	Visual por bloques	No	Depuración (debug)
8	<p>¿Qué dirección debería ir el robot para llegar al destino por el camino señalado?</p>  <p>Opción A: </p> <p>Opción B: </p> <p>Opción C: </p> <p>Opción D: </p>	d	8	<p>¿Qué dirección debería ir el robot para llegar al destino por el camino señalado?</p>  <p>Opción A: </p> <p>Opción B: </p> <p>Opción C: </p> <p>Opción D: </p>	b	Bucles; Repetir veces	El Laberinto	Visual por bloques	Si	Secuenciación









I-1	PRE TEST	Opc	I-2	POS TEST	Opc	CONCEPTOS	ENTORNO/ INTERFAZ	ESTILO DE RESPUESTA	ANIDAM IENTO	TAREA REQUERIDA	
25	<p>El terreno de ajedrez compuesto de casillas, al que llamamos "my board", y que dibujo en pantalla de un punto de vista...</p>  <p>¿Qué terreno de ajedrez se obtiene al moverse el jugador "black" desde cada uno de los lados de cada casilla hacia sus vecinos...</p> <p>Opción A: <code>my board</code> <code>repeat 3 veces</code> <code>hacer move and get 1</code> <code>repeat 3 veces</code> <code>hacer move and get 1</code></p> <p>Opción B: <code>my board</code> <code>repeat 3 veces</code> <code>hacer move and get 1</code> <code>repeat 3 veces</code> <code>hacer move and get 1</code></p> <p>Opción C: <code>my board</code> <code>repeat 3 veces</code> <code>hacer move and get 1</code> <code>repeat 3 veces</code> <code>hacer move and get 1</code></p> <p>Opción D: <code>my board</code> <code>repeat 3 veces</code> <code>hacer move and get 1</code> <code>repeat 3 veces</code> <code>hacer move and get 1</code></p>		c	25	<p>El terreno de ajedrez compuesto de casillas, al que llamamos "my board", y que dibujo en pantalla de un punto de vista...</p>  <p>¿Qué terreno de ajedrez se obtiene al moverse el jugador "black" desde cada uno de los lados de cada casilla hacia sus vecinos...</p> <p>Opción A: <code>my board</code> <code>repeat 3 veces</code> <code>hacer move and get 1</code> <code>repeat 3 veces</code> <code>hacer move and get 1</code></p> <p>Opción B: <code>my board</code> <code>repeat 3 veces</code> <code>hacer move and get 1</code> <code>repeat 3 veces</code> <code>hacer move and get 1</code></p> <p>Opción C: <code>my board</code> <code>repeat 3 veces</code> <code>hacer move and get 1</code> <code>repeat 3 veces</code> <code>hacer move and get 1</code></p> <p>Opción D: <code>my board</code> <code>repeat 3 veces</code> <code>hacer move and get 1</code> <code>repeat 3 veces</code> <code>hacer move and get 1</code></p>	b	Funciones simples	El Lienzo	Visual por bloques	No	Secuenciación
26	<p>El terreno de ajedrez compuesto de casillas, al que llamamos "my board", y que dibujo en pantalla de un punto de vista...</p>  <p>¿Qué terreno de ajedrez se obtiene al moverse el jugador "black" desde cada uno de los lados de cada casilla hacia sus vecinos...</p> <p>Opción A: 15</p> <p>Opción B: 5</p> <p>Opción C: 4</p> <p>Opción D: 3</p>		d	26	<p>El terreno de ajedrez compuesto de casillas, al que llamamos "my board", y que dibujo en pantalla de un punto de vista...</p>  <p>¿Qué terreno de ajedrez se obtiene al moverse el jugador "black" desde cada uno de los lados de cada casilla hacia sus vecinos...</p> <p>Opción A: 15</p> <p>Opción B: 5</p> <p>Opción C: 4</p> <p>Opción D: 3</p>	b	Funciones simples	El Lienzo	Visual por bloques	No	Completamiento
27	<p>El terreno de ajedrez compuesto de casillas, al que llamamos "my board", y que dibujo en pantalla de un punto de vista...</p>  <p>¿Qué terreno de ajedrez se obtiene al moverse el jugador "black" desde cada uno de los lados de cada casilla hacia sus vecinos...</p> <p>Opción A: <code>my board</code> <code>repeat 3 veces</code> <code>hacer move and get 1</code> <code>repeat 3 veces</code> <code>hacer move and get 1</code></p> <p>Opción B: <code>my board</code> <code>repeat 3 veces</code> <code>hacer move and get 1</code> <code>repeat 3 veces</code> <code>hacer move and get 1</code></p> <p>Opción C: <code>my board</code> <code>repeat 3 veces</code> <code>hacer move and get 1</code> <code>repeat 3 veces</code> <code>hacer move and get 1</code></p> <p>Opción D: <code>my board</code> <code>repeat 3 veces</code> <code>hacer move and get 1</code> <code>repeat 3 veces</code> <code>hacer move and get 1</code></p>		c	27	<p>El terreno de ajedrez compuesto de casillas, al que llamamos "my board", y que dibujo en pantalla de un punto de vista...</p>  <p>¿Qué terreno de ajedrez se obtiene al moverse el jugador "black" desde cada uno de los lados de cada casilla hacia sus vecinos...</p> <p>Opción A: <code>my board</code> <code>repeat 3 veces</code> <code>hacer move and get 1</code> <code>repeat 3 veces</code> <code>hacer move and get 1</code></p> <p>Opción B: <code>my board</code> <code>repeat 3 veces</code> <code>hacer move and get 1</code> <code>repeat 3 veces</code> <code>hacer move and get 1</code></p> <p>Opción C: <code>my board</code> <code>repeat 3 veces</code> <code>hacer move and get 1</code> <code>repeat 3 veces</code> <code>hacer move and get 1</code></p> <p>Opción D: <code>my board</code> <code>repeat 3 veces</code> <code>hacer move and get 1</code> <code>repeat 3 veces</code> <code>hacer move and get 1</code></p>	a	Funciones simples	El Laberinto	Visual por bloques	No	Secuenciación
28	<p>El terreno de ajedrez compuesto de casillas, al que llamamos "my board", y que dibujo en pantalla de un punto de vista...</p>  <p>¿Qué terreno de ajedrez se obtiene al moverse el jugador "black" desde cada uno de los lados de cada casilla hacia sus vecinos...</p> <p>Opción A: 3</p> <p>Opción B: 4</p> <p>Opción C: 5</p> <p>Opción D: 6</p>		d	28	<p>El terreno de ajedrez compuesto de casillas, al que llamamos "my board", y que dibujo en pantalla de un punto de vista...</p>  <p>¿Qué terreno de ajedrez se obtiene al moverse el jugador "black" desde cada uno de los lados de cada casilla hacia sus vecinos...</p> <p>Opción A: 3</p> <p>Opción B: 4</p> <p>Opción C: 5</p> <p>Opción D: 6</p>	c	Funciones simples	El Laberinto	Visual por bloques	No	Completamiento

## Anexo 4

### Ficha de validación de instrumento por Juez experto 01.

#### VALIDACIÓN DE EXPERTOS.

Estimado y respetado Juez.

Ud. ha sido seleccionado para evaluar y validar el instrumento sobre Pensamiento Computacional que hace parte de la investigación “Uso de la plataforma educativa Code.org en la mejora del desarrollo del pensamiento computacional en estudiantes del primer grado de educación secundaria en Arequipa, Perú”.

La evaluación del instrumento es de gran importancia para lograr que los resultados obtenidos sean válidos y a partir de éstos sean utilizados eficazmente en el área de investigación educativa.

Agradecemos su valiosa colaboración.

De acuerdo con los siguientes indicadores califique cada uno de los ítems según corresponda:

CATEGORIA	CALIFICACIÓN	INDICADOR
<b>SUFICIENCIA</b>	1. No cumple con el criterio	Los ítems no son suficientes para medir la dimensión.
	2. Bajo Nivel	Los ítems miden algún aspecto de la dimensión, pero no corresponden con la dimensión total.
	3. Moderado nivel	Se deben incrementar algunos ítems para poder evaluar la dimensión completamente.
	4. Alto nivel	Los ítems son suficientes.
<b>CLARIDAD</b>	1. No cumple con el criterio	El ítem no es claro.
	2. Bajo Nivel	El ítem requiere bastantes modificaciones o una modificación muy grande en el uso de las palabras de acuerdo con su significado o por el orden de las mismas.
	3. Moderado nivel	Se requiere una modificación muy específica de algunos de los términos del ítem.
	4. Alto nivel	El ítem es claro, tiene semántica y sintaxis adecuada.
<b>COHERENCIA</b>	1. No cumple con el criterio	El ítem no tiene relación lógica con la dimensión.
	2. Bajo Nivel	El ítem tiene una relación tangencial con la dimensión.
	3. Moderado nivel	El ítem tiene una relación moderada con la dimensión que está midiendo.
	4. Alto nivel	El ítem se encuentra completamente relacionado con la dimensión que está midiendo.
<b>RELEVANCIA</b>	1. No cumple con el criterio	El ítem puede ser eliminado sin que se vea afectada la medición de la dimensión.
	2. Bajo Nivel	El ítem tiene alguna relevancia, pero otro ítem puede estar incluyendo lo que mide éste.
	3. Moderado nivel	El ítem es relativamente importante.
	4. Alto nivel	El ítem es muy relevante y debe ser incluido.


**Listado de Ítems**

Ítems	Suficiencia	Claridad	Coherencia	Relevancia
Pregunta 1	4	4	4	4
Pregunta 2	4	4	4	4
Pregunta 3	4	4	4	3
Pregunta 4	4	4	4	4
Pregunta 5	4	4	4	4
Pregunta 6	4	4	4	4
Pregunta 7	4	4	4	3
Pregunta 8	4	3	4	4
Pregunta 9	4	4	4	4
Pregunta 10	4	4	4	4
Pregunta 11	4	4	4	3
Pregunta 12	3	4	3	3
Pregunta 13	4	3	4	4
Pregunta 14	4	4	4	4
Pregunta 15	4	4	4	4
Pregunta 16	4	4	4	4
Pregunta 17	4	4	4	4
Pregunta 18	4	3	4	4
Pregunta 19	4	4	4	4
Pregunta 20	4	4	4	4
Pregunta 21	4	3	4	4
Pregunta 22	4	3	4	4
Pregunta 23	4	4	4	4
Pregunta 24	4	4	4	4
Pregunta 25	4	4	4	4
Pregunta 26	4	3	4	4
Pregunta 27	4	3	4	4
Pregunta 28	4	4	4	4

**ASPECTOS GENERALES**

ASPECTO	SI	NO	OBSERVACIONES
El instrumento contiene instrucciones claras y precisas para llenar la ficha.	x		Ninguna
Los ítems permiten el logro del objetivo de la investigación.	x		Ninguna
Los ítems están distribuidos en forma lógica y secuencial.	x		Ninguna
El número de ítems es suficiente para recoger la información. En caso de ser negativa su respuesta, sugiera los ítems a añadir.	x		Ninguna
<b>CONSIDERACIONES FINALES</b>			
Ninguna			

**DECISIÓN**

Aplicable	Si	No Aplicable	Aplicable atendiendo a las observaciones
Instrumento validado por:	<i>Dr. Ronald Paucar Curasma</i>		
Teléfono:	993248663		
Correo electrónico:	<a href="mailto:rpaucarc@gmail.com">rpaucarc@gmail.com</a>		Firma

## Anexo 5

### Ficha de validación de instrumento por Juez experto 02.

#### VALIDACIÓN DE EXPERTOS.

Estimado y respetado Juez.

Ud. ha sido seleccionado para evaluar y validar el instrumento sobre Pensamiento Computacional que hace parte de la investigación “Uso de la plataforma educativa Code.org en la mejora del desarrollo del pensamiento computacional en estudiantes del primer grado de educación secundaria en Arequipa, Perú”.

La evaluación del instrumento es de gran importancia para lograr que los resultados obtenidos sean válidos y a partir de éstos sean utilizados eficazmente en el área de investigación educativa.

Agradecemos su valiosa colaboración.

De acuerdo con los siguientes indicadores califique cada uno de los ítems según corresponda:

CATEGORIA	CALIFICACIÓN	INDICADOR
<b>SUFICIENCIA</b>	1. No cumple con el criterio	Los ítems no son suficientes para medir la dimensión.
	2. Bajo Nivel	Los ítems miden algún aspecto de la dimensión, pero no corresponden con la dimensión total.
	3. Moderado nivel	Se deben incrementar algunos ítems para poder evaluar la dimensión completamente.
	4. Alto nivel	Los ítems son suficientes.
<b>CLARIDAD</b>	1. No cumple con el criterio	El ítem no es claro.
	2. Bajo Nivel	El ítem requiere bastantes modificaciones o una modificación muy grande en el uso de las palabras de acuerdo con su significado o por el orden de las mismas.
	3. Moderado nivel	Se requiere una modificación muy específica de algunos de los términos del ítem.
	4. Alto nivel	El ítem es claro, tiene semántica y sintaxis adecuada.
<b>COHERENCIA</b>	1. No cumple con el criterio	El ítem no tiene relación lógica con la dimensión.
	2. Bajo Nivel	El ítem tiene una relación tangencial con la dimensión.
	3. Moderado nivel	El ítem tiene una relación moderada con la dimensión que está midiendo.
	4. Alto nivel	El ítem se encuentra completamente relacionado con la dimensión que está midiendo.
<b>RELEVANCIA</b>	1. No cumple con el criterio	El ítem puede ser eliminado sin que se vea afectada la medición de la dimensión.
	2. Bajo Nivel	El ítem tiene alguna relevancia, pero otro ítem puede estar incluyendo lo que mide éste.
	3. Moderado nivel	El ítem es relativamente importante.
	4. Alto nivel	El ítem es muy relevante y debe ser incluido.

Listado de Ítems

Ítems	Suficiencia	Claridad	Coherencia	Relevancia
Pregunta 1	4	4	4	4
Pregunta 2	4	4	4	4
Pregunta 3	4	4	4	3
Pregunta 4	4	4	4	4
Pregunta 5	4	4	4	4
Pregunta 6	4	4	4	4
Pregunta 7	4	4	4	3
Pregunta 8	4	3	4	4
Pregunta 9	4	4	4	4
Pregunta 10	4	4	4	4
Pregunta 11	4	4	4	3
Pregunta 12	3	4	3	3
Pregunta 13	4	3	4	4
Pregunta 14	4	4	4	4
Pregunta 15	4	4	4	4
Pregunta 16	4	4	4	4
Pregunta 17	4	4	4	4
Pregunta 18	4	3	4	4
Pregunta 19	4	4	4	4
Pregunta 20	4	4	4	4
Pregunta 21	4	3	4	4
Pregunta 22	4	3	4	4
Pregunta 23	4	4	4	4
Pregunta 24	4	4	4	4
Pregunta 25	4	2	4	4
Pregunta 26	4	2	4	4
Pregunta 27	4	2	4	4
Pregunta 28	4	4	4	4

**ASPECTOS GENERALES**

ASPECTO	SI	NO	OBSERVACIONES
El instrumento contiene instrucciones claras y precisas para llenar la ficha.	x		Ninguna
Los ítems permiten el logro del objetivo de la investigación.	x		Ninguna
Los ítems están distribuidos en forma lógica y secuencial.	x		Ninguna
El número de ítems es suficiente para recoger la información. En caso de ser negativa su respuesta, sugiera los ítems a añadir.	x		Ninguna
<b>CONSIDERACIONES FINALES</b>			
Ninguna			

**DECISIÓN**

Aplicable	Si	No Aplicable	Aplicable atendiendo a las observaciones
Instrumento validado por:	<i>Dra. Nombre Apellido Apellido</i> Elizabeth Vidal Duarte		 Firma
Teléfono:	986451412		
Correo electrónico:	<a href="mailto:evidadl@unsa.edu.pe">evidadl@unsa.edu.pe</a>		

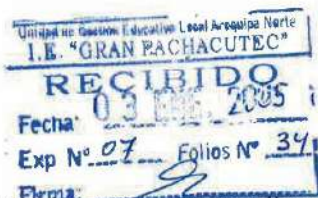
## Anexo 6

*Solicitud y autorización para hacer el estudio.*

### SOLICITO AUTORIZACIÓN PARA APLICAR INSTRUMENTO DE INVESTIGACIÓN

**SEÑOR: CÉSAR VÁSQUEZ MEZA**

**DIRECTOR DE LA INSTITUCIÓN EDUCATIVA GRAN PACHACÚTEC**



RAÚL BARCELON AYMARA ALFARO identificado con DNI N° 29658711, con domicilio real en la Urb. Sudamericana Mz. E, Lote 4, distrito de Cerro Colorado, provincia y departamento de Arequipa; ante usted con el debido respeto me presento y digo:

Que, mi persona viene realizando un trabajo de investigación sobre **“Uso de la plataforma educativa Code.org en la mejora del desarrollo del pensamiento computacional en estudiantes del primer grado de educación secundaria en Arequipa, Perú”**; para cuya investigación es necesario el recojo de información, por lo que **solicito** a usted la correspondiente **autorización** de aplicación del instrumento de investigación, la que consta de una prueba pre y postest de evaluación estandarizada que medirá el pensamiento computacional en os estudiantes del primer grado de las secciones A y B, durante el primer bimestre educativo del presente año lectivo.

Por lo antes expuesto: ruego acceder a mi petición.

Arequipa, 02 de enero del 2025.



.....  
Raúl Aymara Alfaro  
DNI 29658711

*Adjunto: Proyecto de investigación.*



**I.E. "GRAN PACHACUTEC"**

FUNDADO EL 19 DE MAYO DE 1982

**NIVEL SECUNDARIO - JORNADA ESCOLAR COMPLETA**



*Decreto Directoral N° 001-2025- D.IE "GP"-UGEL-AN*

*Arequipa, 2025 marzo 14*

**VISTO:**

La solicitud presentada por don RAÚL BARCELON AYMARA ALFARO, con expediente N° 72 -2025, de 58 folios, quien a través del cual solicita autorización para aplicar instrumento de investigación

**CONSIDERANDO:**

Que, don RAUL BARCELON AYMARA ALFARO , solicita autorización para aplicar instrumento de investigación, para el trabajo de investigación sobre “ **Uso de la plataforma educativa Code.org en la mejora del desarrollo del pensamiento computacional en estudiantes del primer grado de educación secundaria en Arequipa, Perú**”, en la que consta de una prueba pre y postest de evaluación estandarizada en los estudiantes del Primer Grado de las Secciones A y B, durante el Primer Bimestre del este año lectivo.

Que, en mérito a la normativa vigente Ley N° 28044 Ley General de Educación, Decreto Supremo N° 006-2021-MINEDU, que aprueba los lineamientos para la Gestión Escolar en las Instituciones Públicas, la RVM N° 556-2025 sobre la norma Técnica para el año escolar de Instituciones y programas educativos y privados de la Educación Básica para el año 2025.

Estando a lo dispuesto por Ley 28044 y demás normas conexas.

**SE DECRETA:**

**1. AUTORIZAR** al Prof. Raúl Barcelon Aymara Alfaro, la aplicación del instrumento de investigación, sobre “**Uso de la plataforma educativa Code.org en la mejora del desarrollo del pensamiento computacional en estudiantes del primer grado de educación secundaria en Arequipa, Perú**”, en la que consta de una prueba pre y postest de evaluación estandarizada en los estudiantes del Primer Grado de las Secciones A y B, durante el Primer Bimestre del este año lectivo.

**2. DISPONER** que el presente documento sea remitido a la instancia correspondiente para su conocimiento y fines del caso.

**REGISTRESE, COMUNÍQUESE Y ARCHÍVESE.**



# Anexo 7

GRUPO CONTROL PRE TEST (04-ABRIL - 2025)

1A	Direcciones básicas				Repetir veces				Repetir hasta				Condicional simple				Condicional compuesto				Bucle mientras				Funciones simples				NIVEL DE	
Nro	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	PTS	LOGRO
1		1											1		1		1	1					1	1		1		8	En proceso	
2	1	1			1	1			1	1	1						1		1					1				10	En proceso	
3	1	1	1		1	1	1		1	1	1		1	1			1	1		1	1			1	1			17	Previsto	
4					1		1		1						1	1		1							1		1	8	En proceso	
5	1	1	1		1				1		1		1			1	1						1			1		11	En proceso	
6	1	1	1		1	1	1	1	1	1	1		1	1			1	1	1		1	1	1	1		1	1	21	Previsto	
7	1				1				1		1		1	1				1	1	1				1		1		10	En proceso	
8	1	1	1		1	1	1	1	1	1	1		1			1			1	1			1	1				16	Previsto	
9	1	1	1	1	1	1			1		1		1	1			1	1					1					12	En proceso	
10	1	1	1		1	1	1		1	1	1	1	1			1		1	1	1			1		1	1	1	19	Previsto	
11	1	1			1	1	1			1			1		1			1	1	1							1	12	En proceso	
12	1	1	1		1	1							1	1			1		1								1	10	En proceso	
13	1	1	1		1		1		1	1	1		1				1		1				1	1				13	En proceso	
14	1	1			1	1	1		1		1		1	1			1	1	1		1				1			14	En proceso	
15			1		1	1			1		1		1		1	1	1	1					1		1	1		13	En proceso	
16	1	1	1		1	1	1		1	1	1		1		1			1		1	1			1	1	1	1	19	Previsto	
17			1		1				1		1		1	1	1								1		1	1	1	11	En proceso	
18	1	1			1	1			1				1		1		1				1			1	1	1	1	13	En proceso	
19	1	1	1		1	1	1		1	1	1				1				1	1	1		1		1			15	Previsto	
	<b>15</b>	<b>15</b>	<b>12</b>	<b>1</b>	<b>18</b>	<b>13</b>	<b>10</b>	<b>2</b>	<b>15</b>	<b>9</b>	<b>11</b>	<b>1</b>	<b>12</b>	<b>9</b>	<b>4</b>	<b>7</b>	<b>9</b>	<b>10</b>	<b>10</b>	<b>7</b>	<b>12</b>	<b>3</b>	<b>1</b>	<b>14</b>	<b>7</b>	<b>10</b>	<b>7</b>	<b>8</b>	<b>13.3</b>	
	10.75				10.75				9				8				9				7.5				8					

GRUPO CONTROL POST TEST (30 DE JUNIO - 2025)

1A	Direcciones básicas				Repetir veces				Repetir hasta				Condicional simple				Condicional compuesto				Bucle mientras				Funciones simples				NIVEL DE	
Nro	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	PTS	LOGRO
1		1	1		1						1		1		1	1	1						1		1		11	En proceso		
2		1	1		1				1	1	1	1	1							1			1			1		10	En proceso	
3	1	1	1		1				1	1		1	1	1	1	1			1				1	1	1			16	Previsto	
4	1						1		1				1					1					1	1		1	1	10	En proceso	
5	1	1			1	1			1	1						1						1			1			10	En proceso	
6	1	1	1		1	1	1	1	1	1	1		1	1	1		1	1	1	1	1	1	1			1	1	23	Destacado	
7	1	1	1		1	1			1	1	1		1				1											10	En proceso	
8	1	1	1		1	1	1	1	1	1			1	1			1	1		1			1		1	1	1	18	Previsto	
9	1		1	1	1	1		1	1				1			1			1				1	1		1		13	En proceso	
10	1	1	1	1		1	1	1	1		1		1			1	1	1					1	1	1	1	1	18	Previsto	
11	1	1	1	1				1		1			1					1			1						1	12	En proceso	
12		1	1		1	1					1			1			1							1		1	1	11	En proceso	
13	1	1	1		1	1			1	1	1		1				1	1	1	1			1		1			15	Previsto	
14	1	1	1	1	1	1	1		1	1	1	1	1	1			1	1			1			1	1	1		19	Previsto	
15			1			1							1		1						1	1				1	1	8	En proceso	
16	1	1	1		1	1	1		1	1	1					1	1		1	1			1		1	1	1	17	Previsto	
17	1	1	1		1	1			1		1					1	1	1	1				1	1		1	1	15	Previsto	
18	1	1	1		1	1	1	1		1	1		1				1	1	1		1			1				14	En proceso	
19	1	1	1		1	1			1	1	1		1			1	1	1	1				1	1		1	1	17	Previsto	
	<b>15</b>	<b>15</b>	<b>17</b>	<b>4</b>	<b>15</b>	<b>14</b>	<b>8</b>	<b>5</b>	<b>15</b>	<b>11</b>	<b>13</b>	<b>3</b>	<b>10</b>	<b>10</b>	<b>3</b>	<b>4</b>	<b>10</b>	<b>11</b>	<b>9</b>	<b>8</b>	<b>9</b>	<b>5</b>	<b>1</b>	<b>13</b>	<b>9</b>	<b>7</b>	<b>12</b>	<b>11</b>	<b>14.1</b>	
	12.75				10.5				10.5				6.75				9.5				7				9.75					

Matriz de datos Pre y Post de 2 grupos

GRUPO EXPERIMENTAL PRE TEST (04-ABRIL - 2025)

1B	Direcciones básicas				Repetir veces				Repetir hasta				Condicional simple				Condicional compuesto				Bucle mientras				Funciones simples				PTS	NIVEL DE LOGRO
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28		
1	1	1	1		1	1	1		1		1		1	1	1		1	1		1				1	1	1	1	17	Previsto	
2	1				1	1	1		1		1		1	1						1			1					9	En proceso	
3		1	1	1	1	1	1		1	1	1		1	1				1		1		1				1	1	16	Previsto	
4			1	1		1	1		1											1							1	7	En Inicio	
5	1				1				1				1	1			1	1		1						1		9	En proceso	
6	1	1	1		1	1		1	1	1	1	1	1	1	1		1	1		1	1		1	1				19	Previsto	
7	1	1	1	1	1		1		1		1		1				1					1					1	12	En proceso	
8	1	1	1	1	1	1			1			1		1	1			1	1		1	1		1		1	1	16	Previsto	
9	1	1			1	1			1	1			1	1		1	1										1	11	En proceso	
10	1			1		1				1	1				1	1			1						1	1		10	En proceso	
11	1	1			1	1	1		1	1	1		1												1	1		11	En proceso	
12		1		1	1		1		1		1	1	1				1		1				1	1	1			13	En proceso	
13		1	1					1	1	1																1	1	7	En Inicio	
14	1	1			1					1	1		1	1		1	1		1	1			1				1	12	En proceso	
15		1		1	1	1				1	1		1		1				1			1				1	1	12	En proceso	
16		1			1			1		1	1	1	1		1		1						1	1			1	11	En proceso	
17	1	1	1		1	1		1	1				1					1				1		1	1	1	1	15	Previsto	
18	1	1	1		1		1	1	1	1	1	1	1	1					1		1				1	1		15	Previsto	
19	1	1		1	1		1		1	1		1	1					1	1				1	1		1	1	15	Previsto	
	13	15	9	8	16	11	9	3	14	10	11	9	13	6	7	4	8	6	4	5	9	2	3	7	7	8	9	11	12.47	
	11.25				9.75				11				7.5				5.75				5.25				8.75					

GRUPO EXPERIMENTAL POST TEST (30 DE JUNIO - 2025)

1B	Direcciones básicas				Repetir veces				Repetir hasta				Condicional simple				Condicional compuesto				Bucle mientras				Funciones simples				PTS	NIVEL DE LOGRO
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28		
1	1	1			1	1	1	1	1	1	1		1	1	1				1			1		1		1	1	17	Previsto	
2	1			1			1	1	1	1	1		1					1		1	1		1		1		1	11	En proceso	
3	1	1	1		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	24	Destacado
4	1	1	1			1	1	1	1	1	1		1	1				1	1	1			1					14	En proceso	
5		1				1			1		1		1	1							1	1		1	1	1	1	12	En proceso	
6	1	1	1	1	1	1	1		1	1	1		1	1	1				1			1	1	1	1	1	1	19	Previsto	
7	1	1	1		1	1	1		1	1	1		1	1	1		1	1	1		1	1		1	1	1	1	18	Previsto	
8	1		1	1	1	1	1	1	1	1	1		1	1	1	1	1	1	1	1	1	1		1	1		1	19	Previsto	
9	1	1	1		1	1	1		1	1	1		1		1		1	1	1	1	1	1		1		1	1	19	Previsto	
10	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1			1			1	1		1	1		1	18	Previsto	
11	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	24	Destacado	
12	1	1	1		1	1	1		1	1	1		1		1	1				1	1	1		1		1	1	19	Previsto	
13	1	1	1	1	1	1	1		1	1	1		1	1	1				1				1			1	1	16	Previsto	
14	1	1	1		1	1	1		1	1	1		1	1	1				1		1	1	1	1	1	1	1	21	Previsto	
15	1	1			1	1	1	1	1	1	1		1	1	1		1		1		1		1	1			1	18	Previsto	
16	1	1	1	1	1	1	1	1	1	1	1								1	1			1			1	15	Previsto		
17	1	1	1	1	1	1	1		1	1	1		1		1	1	1	1	1	1	1	1		1	1		1	18	Previsto	
18	1	1	1		1	1	1		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	22	Destacado	
19	1			1	1	1	1		1	1	1		1	1	1	1				1	1	1		1	1		1	19	Previsto	
	18	16	14	10	16	18	15	10	15	18	14	5	14	15	7	13	8	11	7	14	12	10	0	19	10	9	11	14	18.05	
	14.5				14.75				13				12.25				10				10.25				11					