

**Prototipo de aplicación web como herramienta de apoyo para la evaluación de
Competencias en Pensamiento Computacional**

Sergio Estrada Castro

Juan Camilo Gonzalez

Universidad del Valle

Facultad de Ingeniería

Ingeniería de Sistemas

Tuluá, Valle del Cauca 2025

**Prototipo de aplicación web como herramienta de apoyo para la evaluación de
Competencias en Pensamiento Computacional**

Sergio Estrada Castro

Código 201866145-3743

sergio.estrada@correounivalle.edu.co

Juan Camilo Gonzalez

Código 201958435-3743

juan.gonzalez.g@correounivalle.edu.co

DIRECTORA

Yuri Mercedes Bermudez Mazuera ing.

Ing. Profesora de Ingeniería de Sistemas y Computación

yuri.bermudez@correounivalle.edu.co

Universidad del Valle

Facultad de Ingeniería

Ingeniería de Sistemas

Tuluá, Valle del Cauca 2025

Tabla de Contenido

Resumen	8
1. Introducción.....	11
1.1. Planteamiento del Problema.....	11
1.1.1. Descripción del Problema.....	11
1.1.2 Formulación del Problema.....	12
1.2. Información sobre los capítulos.....	13
2. Alcance del proyecto.....	15
2.1. Línea Base del Alcance.....	15
2.2. Objetivo General:.....	15
2.3. Objetivos Específicos:.....	15
2.4. Entregables Esperados.....	15
3. Marco Referencial.....	17
3.1. Antecedentes.....	17
3.2. Marco Teórico.....	22
3.3. Marco Conceptual.....	24
3.4. Glosario	26
4. Metodología.....	27
4.1. Metodología de Investigación.....	27
5. Métodos de Evaluación de Competencias en Pensamiento Computacional	29
5.1. Métodos de Evaluación del Pensamiento Computacional.....	29
5.2. Comparación de métodos.....	33

5.3. Conclusión.....	35
6. Estructura del Método de Evaluación por Competencias en Pensamiento Computacional.....	36
6.1. Competencia en Pensamiento Computacional que se Evaluarán.....	36
6.2. Criterios de Evaluación.....	37
6.3. Indicadores de Logro.....	41
6.4. Estructura y Formato del Método de Evaluación.....	43
6.5. Conclusión.....	46
7. Desarrollo del Prototipo de Aplicación Web que Incorpora el Método de Evaluación de Competencias de PC.....	48
7.1. Metodología de Desarrollo.....	48
7.2. Requerimientos de la Aplicación.....	48
7.2.1. Requerimientos Funcionales.....	48
7.2.2. Requerimientos no Funcionales.....	49
7.3. Justificación de las Tecnologías.....	51
7.4. Descripción de las Tecnologías de Desarrollo.....	51
7.4.1. Node.js y Express.....	52
7.4.2. EJS.....	52
7.4.3. MySQL.....	52
7.4.4. JWT (JSON Web Token).....	53
7.4.5. Bcrypt y Bcryptjs.....	53
7.4.6. Cookie-parser.....	53
7.4.7. CSV-parser.....	53

7.4.8. Dotenv.....	54
7.4.9. Express-session.....	54
7.4.10. Multer.....	54
7.5. Diseño de la Base de Datos.....	56
7.6. Diseño y Demostración de las vistas de usuario.....	58
7.7. Justificación del diseño visual, enfoque de usabilidad y paleta de colores.....	62
7.8. Descripción técnica del prototipo.....	64
8. Pruebas.....	69
8.1. Pruebas de Usabilidad.....	69
8.2. Objetivo y Planteamiento de las Pruebas.....	69
8.2.1. Objetivos.....	69
8.2.2. Planteamiento de las Pruebas.....	70
8.3. Análisis de Resultados.....	72
8.3.1 Atractivo de la Presentación.....	72
8.3.2. Comprensión de los Niveles de Logro.....	73
8.3.3 Facilidad en la Creación de Evaluaciones.....	74
8.3.4 Utilidad de las Gráficas en la Interpretación del Rendimiento.....	75
8.3.5 Satisfacción General con la Experiencia de Uso.....	76
8.3.6 Evaluación Efectiva de los Resultados de Aprendizaje.....	77
9. Conclusiones del Proyecto.....	81
9.1 Trabajos Futuros.....	82
10. Referencias.....	84
11. Anexos.....	88

Lista de Tablas

Tabla 1: Estructura de Capítulos..	14
Tabla 2: Objetivos Específicos vs Entregables..	16
Tabla 3: Comparación entre proyectos..	22
Tabla 4: Comparación Métodos.	34
Tabla 5: Rúbrica Interiorización...	39
Tabla 6: Rúbrica Coordinación.	40
Tabla 7: Rúbrica Encapsulación.	40
Tabla 8: Rúbrica Generalización..	41

Lista de Figuras

Figura 1: Niveles jerárquicos de la evaluación	44
Figura 2: Diagrama de flujo de la aplicación.....	50
Figura 3: Representación de las tecnologías implementadas.....	55
Figura 4: Diseño de base de datos.	57
Figura 5: Apartado de cursos de la aplicación.	59
Figura 6: Apartado de evaluaciones.	60
Figura 7: Botón de crear nueva evaluación.	60
Figura 8: Apartado de resultados.....	61
Figura 9: pregunta 1.....	76
Figura 10: pregunta 2.	77
Figura 11: resultado manual	78
Figura 12: resultado calculado en la página	79

Resumen

El pensamiento computacional se ha consolidado como una habilidad fundamental para el siglo XXI, ya que permite a los individuos comprender y participar activamente en el mundo digital. En el caso de los jóvenes, su desarrollo aporta beneficios significativos en múltiples dimensiones.

En primer lugar, promueve el razonamiento lógico y el pensamiento crítico, al requerir que los estudiantes aborden problemas de manera sistemática y estructurada, estas habilidades resultan clave tanto para el desempeño académico como para la toma de decisiones en la vida cotidiana. Además, estimula la creatividad y la innovación, al incentivar la búsqueda de soluciones originales a diversos retos, capacidades especialmente valoradas en el contexto actual. Asimismo, el pensamiento computacional prepara a los jóvenes para enfrentar los desafíos del mercado laboral, especialmente en áreas relacionadas con la ciencia, la tecnología, la ingeniería y las matemáticas, donde estas competencias son cada vez más demandadas.

Reconociendo su relevancia, se han propuesto múltiples metodologías para enseñar pensamiento computacional, cada una con distintos enfoques y estrategias de evaluación. Sin embargo, las investigaciones evidencian deficiencias en los métodos de evaluación existentes, lo que dificulta validar de forma efectiva el aprendizaje de estas competencias. Ante esta problemática, el presente trabajo propone el desarrollo de una aplicación web parametrizable, orientada a evaluar las competencias en pensamiento computacional de estudiantes de educación secundaria. La solución se fundamentará en un modelo de evaluación integral que combine enfoques formales e informales, haciendo uso de rúbricas para valorar el desempeño de los estudiantes en tareas específicas relacionadas con el pensamiento computacional.

Palabras clave: pensamiento computacional, aplicación web, jóvenes, habilidades esenciales, metodologías.

Abstract

Computational thinking has become an essential skill for the 21st century, enabling individuals to understand and actively engage with the digital world. For young people, developing this skill offers substantial benefits across various areas.

Firstly, it enhances logical reasoning and critical thinking by encouraging students to approach problems in a structured and systematic way. These abilities are crucial for both academic success and effective decision-making in everyday life. Moreover, computational thinking fosters creativity and innovation by promoting the development of original solutions—skills that are increasingly valued in today's world.

Additionally, it equips young people for future career paths, particularly in fields related to science, technology, engineering, and mathematics (STEM), where such competencies are in high demand.

Given its importance, several teaching methodologies have been proposed to foster computational thinking, each with its own learning strategies and evaluation approaches. However, research has identified shortcomings in current assessment methods, making it difficult to effectively measure students' acquisition of these competencies.

In response to this issue, the present project proposes the development of a customizable web application designed to evaluate computational thinking competencies in secondary school students. The application will be based on a comprehensive evaluation model that combines both formal and informal assessment strategies, using rubrics to assess student performance on computational thinking tasks.

Keywords: computational thinking, web application, young people, essential skills, methodologies.

Capítulo 1

Introducción

El pensamiento computacional es una habilidad que permite a las personas resolver problemas de manera creativa y eficiente, utilizando herramientas y conceptos de la informática. En los últimos años, el pensamiento computacional ha emergido como una habilidad clave para la vida, ya que es cada vez más importante ser capaz de comprender y utilizar la tecnología (Grover & Pea, 2013).

El pensamiento computacional se puede desarrollar a cualquier edad, pero es importante comenzar a enseñarlo a los niños desde una edad temprana. Los estudiantes de primaria pueden comenzar a aprender los conceptos básicos del pensamiento computacional, como la descomposición de problemas, la identificación de patrones y la resolución de problemas. En la secundaria, los estudiantes pueden desarrollar estas habilidades de manera más profunda, aprendiendo a programar y a crear soluciones informáticas complejas (Organización para la Cooperación y el Desarrollo Económicos [OCDE], 2018).

El pensamiento computacional es una habilidad transversal que puede aplicarse a una amplia gama de contextos. Es importante que los estudiantes desarrollen esta habilidad, ya que puede influir en su vida académica, profesional y personal.

Desde el auge de la revolución tecnológica, la necesidad de integrar el pensamiento computacional en la educación se ha vuelto innegable. Múltiples expertos en educación, como Seymour Papert, han subrayado la importancia de incorporar estas habilidades desde las etapas formativas, argumentando que el pensamiento computacional no solo se trata de preparar a los estudiantes para carreras técnicas, sino de equiparlos con herramientas cognitivas cruciales para diversos aspectos de la vida (Pérez & Rodríguez, 2024).

Es en la educación primaria donde se siembran las semillas del pensamiento computacional. Al introducir conceptos fundamentales de lógica y estructuración de

problemas, se establece una base sólida para el desarrollo futuro. Sin embargo, es en la educación secundaria donde esta semilla germina y florece. Aquí, los estudiantes no solo amplían su comprensión de conceptos previos, sino que también se enfrentan a desafíos más complejos que exigen un pensamiento computacional más avanzado (Pérez & Rodríguez, 2024).

En este contexto, se analiza cómo el pensamiento computacional ha emergido como una habilidad clave para la vida, destacando su relevancia desde las etapas iniciales de la educación y su evolución en la educación secundaria. Resaltando la importancia de expertos como Seymour Papert, se busca entender cómo su implementación efectiva en el proceso educativo prepara a los estudiantes no solo para los desafíos tecnológicos, sino también para afrontar creativamente los retos de la sociedad actual. Reconociendo su valor en la educación contemporánea, se exploran métodos y formas de evaluar estas competencias, surgiendo así preguntas como: ¿Cómo se puede evaluar el pensamiento computacional? ¿Qué mecanismos existen para hacerlo?

1.1. Planteamiento del problema

1.1.1. Descripción del problema

En la sociedad actual, el pensamiento computacional se ha consolidado como una habilidad fundamental, vital para el éxito en un mundo cada vez más digital (Grover & Pea, 2013). Jeannette Wing, en su influyente trabajo, define el pensamiento computacional como la habilidad para abordar problemas de manera lógica, descomponerlos en partes manejables, identificar patrones y diseñar soluciones algorítmicas (Wing, 2006). Por tanto, en la educación secundaria el fomento del pensamiento computacional se ha convertido en un objetivo clave, con programas educativos y currículos que buscan integrar esta habilidad en las aulas (Cuny, 2010). Sin embargo, surge un problema apremiante en este contexto: la evaluación de las competencias en pensamiento computacional.

Ya que la implementación efectiva del pensamiento computacional en la educación secundaria no solo implica la incorporación de conceptos y habilidades, sino también la capacidad de evaluar de manera precisa y las competencias adquiridas por los estudiantes en la actualidad, existe una falta significativa de mecanismos de evaluación estandarizados y efectivos que permitan medir con precisión el nivel de competencia en pensamiento computacional en estudiantes de educación secundaria (Barker & Garvin-Doxas, 2016). Esta deficiencia plantea un desafío sustancial en el proceso educativo, sumado a la falta de un enfoque uniforme y confiable para evaluar estas competencias, lo cual, dificulta tanto la medición del progreso individual de los estudiantes como la toma de decisiones educativas a nivel de currículo y políticas, asimismo limita la capacidad de los educadores para adaptar sus métodos de enseñanza y proporcionar retroalimentación significativa a los estudiantes, lo que podría resultar en un aprendizaje menos efectivo. Además, esta falta de mecanismos de evaluación se debe a que la mayoría de las investigaciones se centran en trabajarlo desde una edad temprana hasta la educación primaria, creando un vacío en la evaluación del pensamiento computacional en la educación secundaria. También, se ha encontrado que en muy pocos estudios se ha demostrado la efectividad de los métodos empleados para evaluar el pensamiento computacional en esta etapa educativa (Barker & Garvin-Doxas, 2016). Por consiguiente, resulta de suma importancia abordar la identificación de métodos y herramientas que permitan evaluar las habilidades de pensamiento computacional en educación secundaria.

1.1.2 Formulación del problema

El problema que se aborda en este trabajo radica en la carencia de mecanismos de evaluación efectivos y estandarizados para medir las competencias en pensamiento computacional de estudiantes en educación secundaria. Específicamente, se busca responder a la siguiente pregunta de investigación:

¿Qué mecanismos se pueden emplear para evaluar y hacer seguimiento a las competencias en pensamiento computacional de estudiantes en nivel de educación secundaria?

Para resolver este problema, es esencial explorar y desarrollar un método de evaluación que permita medir el nivel de competencia en pensamiento computacional de los estudiantes, proporcionar retroalimentación útil y guiar la toma de decisiones en el ámbito educativo, al responder a esta pregunta no solo se beneficiaría a los educadores y estudiantes, sino que también contribuiría al avance de la educación en el siglo XXI, permitiendo que los futuros ciudadanos estén un poco más preparados para enfrentar los desafíos tecnológicos de manera competente y creativa.

1.2. Información sobre los capítulos

En la siguiente tabla se muestra una corta descripción de cada capítulo del documento.

Tabla 1*Estructura de Capítulos*

Capítulo.	Descripción.
Capítulo 2: Alcance de la propuesta.	En este capítulo se define el alcance de la propuesta.
Capítulo 3: Marco Referencial.	Se explican los conceptos necesarios para comprender el problema y su solución, se espera que el lector tenga conocimiento básico en algunos temas relacionados.
Capítulo 4: Metodología.	En este capítulo se detalla la metodología utilizada para llevar a cabo la investigación.
Capítulo 5: Métodos de Evaluación de Competencias en Pensamiento Computacional.	En este capítulo se presenta un análisis comparativo de diversos métodos de evaluación del Pensamiento Computacional (PC) documentados en la literatura reciente.
Capítulo 6: Estructura del Método de Evaluación por Competencias en Pensamiento Computacional.	En este capítulo se expone el diseño estructural del método de evaluación por competencias que se implementó para evaluar el desarrollo del Pensamiento Computacional (PC).
Capítulo 7: Desarrollo del prototipo de aplicación web que incorpora el método de evaluación de competencias de PC.	En este capítulo se muestran algunos aspectos que se han tenido en cuenta para el diseño e implementación de la aplicación.
Capítulo 8: Pruebas.	Se describen las pruebas que se realizaron durante el proyecto, también se explican los resultados obtenidos.
Capítulo 9: Conclusiones y Trabajos futuros.	Las conclusiones del proyecto y los trabajos futuros que se proponen para mejorar la aplicación propuesta en este trabajo de grado.

Nota. Elaboración propia (González & Estrada, 2025).

Capítulo 2

Alcance del proyecto

2.1. Línea base del alcance

Los autores proponen desarrollar en un tiempo estimado de dos semestres, un prototipo de aplicación web que permita evaluar y genere un seguimiento de competencias en pensamiento computacional de estudiantes en el nivel de educación secundaria.

2.2. Objetivo general:

- Implementar un prototipo de aplicación web que permita evaluar y genere un seguimiento de competencias en pensamiento computacional de estudiantes en el nivel de educación secundaria.

2.3. Objetivos específicos:

- Analizar el método de evaluación de competencias en Pensamiento Computacional que se va a incorporar.
- Diseñar la estructura del método de evaluación a implementar.
- Desarrollar el prototipo de aplicación web que incorpore el método de evaluación de las competencias en Pensamiento computacional.
- Realizar pruebas de usabilidad a la aplicación implementada.

2.4 Entregables esperados

Tabla 2

Objetivos Específicos vs Entregables.

Objetivo Específico	Entregable	Indicador	Meta
1	Informe de análisis del método de evaluación, identificando sus componentes clave, criterios de evaluación y fundamentos teóricos.	Cantidad de métodos de evaluación analizados.	100%
	Documento de diseño que		

2	describe la estructura detallada del método de evaluación, incluyendo la definición de la competencia, resultados de aprendizaje y criterios de evaluación asociados.	Competencia de pensamiento computacional que se evaluarán.	100%
3	Código fuente de la aplicación web, con funciones y módulos implementados según el diseño establecido.	Operatividad de la aplicación web.	90%
4	Informe de pruebas de usabilidad que documenta la experiencia del usuario al interactuar con la aplicación.	Número de pruebas de usabilidad realizadas y porcentaje de aceptación.	90%

Nota. Elaboración propia (González & Estrada, 2024).

Capítulo 3

Marco Referencial

3.1 Antecedentes

- **Programa de Tutoría en Pensamiento Computacional para Estudiantes de Primaria en Colombia:** Un importante antecedente que inspira y respalda el presente proyecto de grado se relaciona con el estudio titulado "Diseño y evaluación de un programa de tutoría de pensamiento computacional para estudiantes de primaria en Colombia" llevado a cabo en el año 2023 por Arévalo-Sánchez, J., Suárez-Rodríguez, C., & Suárez-Rodríguez. Este estudio representó un esfuerzo significativo para abordar la creciente importancia del pensamiento computacional en la educación primaria, específicamente en el contexto de una escuela pública en Colombia.

El estudio se centró en la implementación de un programa de tutoría de pensamiento computacional dirigido por tutores capacitados en esta área. Durante un año escolar, los tutores trabajaron con estudiantes de quinto grado, dedicando 30 minutos semanales a sesiones de tutoría. El programa utilizó diversas actividades y recursos diseñados para enseñar a los estudiantes conceptos y habilidades de pensamiento computacional. Los resultados del estudio revelaron un impacto positivo y significativo en las habilidades de pensamiento computacional de los estudiantes que participaron en el programa. Estos estudiantes demostraron mejoras notables en áreas clave, como la representación de problemas, descomposición, algoritmos, pensamiento crítico y resolución de problemas. Además, el programa de tutoría logró despertar una mayor motivación en los estudiantes para aprender sobre programación y tecnología, así como un incremento en su confianza a la hora de abordar desafíos y pensar de forma crítica.

Como parte de sus recomendaciones, los autores del estudio sugirieron la expansión de este programa de tutoría a otras instituciones educativas en Colombia y la realización de investigaciones adicionales para evaluar el impacto a largo plazo del programa en el desarrollo de habilidades de pensamiento computacional y la motivación de los estudiantes.

Este antecedente resalta la efectividad de un programa de tutoría de pensamiento computacional en el contexto de la educación primaria y subraya la importancia de abordar estas habilidades desde edades tempranas. Además, proporciona un enfoque valioso para el presente proyecto de grado, que busca explorar y desarrollar mecanismos de evaluación de competencias en pensamiento computacional en estudiantes de educación secundaria. Las lecciones aprendidas de este antecedente contribuirán a la formulación y ejecución efectiva de este proyecto (Arévalo-Sánchez, Suárez-Rodríguez, & Suárez-Rodríguez, 2023).

- **Iniciativa Internacional Bebras en Promoción del Pensamiento Computacional en Estudiantes:**

La iniciativa internacional Bebras es un antecedente fundamental que respalda la importancia de promover el pensamiento computacional y la informática en estudiantes jóvenes. Creado por un equipo de profesores e investigadores de informática de la Universidad de Vilnius en Lituania, el concurso Bebras tiene como objetivo fomentar el desarrollo de habilidades de pensamiento computacional en niños y jóvenes, abarcando edades desde 5 hasta 19 años.

El concurso Bebras se distingue por su enfoque en la evaluación de las habilidades de pensamiento computacional de los estudiantes, sin requerir ningún conocimiento previo de programación. Esto lo convierte en una herramienta inclusiva que puede ser utilizada en entornos educativos diversos.

El impacto del concurso Bebras se ha extendido a nivel global, con la participación de estudiantes de más de 100 países. Los resultados y beneficios de esta iniciativa son notables. Los estudiantes que participan en el concurso Bebras experimentan mejoras significativas en sus habilidades de pensamiento computacional, resolución de problemas y creatividad. Esto refuerza la idea de que actividades desafiantes y divertidas relacionadas con la informática pueden enriquecer la formación de los jóvenes.

Además, el concurso Bebras se lleva a cabo anualmente en el mes de abril y está disponible en varios idiomas, incluido el español, lo que facilita su implementación en distintos contextos educativos. La apertura del concurso a estudiantes de edades tan tempranas como 5 años subraya la importancia de iniciar la promoción del pensamiento computacional desde edades tempranas (Fedesoft, 2023).

- **Fostering computational thinking in primary school through a lego®-based music notation:**

Una investigación publicada en la revista IEEE Explore examinó una actividad de aprendizaje basada en LEGO® que utiliza esta combinación. La actividad, llamada "LEGO® Musical Notation", ayuda a los alumnos a comprender los conceptos básicos del pensamiento computacional, como la secuencia, la selección y la repetición.

Para abordar este desafío, un equipo de investigadores de la Universidad de Tampere, Finlandia, desarrolló una nueva metodología de enseñanza del PC basada en la música. La metodología utiliza LEGO® bricks para crear una notación musical multimodal.

La notación musical multimodal combina aspectos de la música y la informática. Por ejemplo, los LEGO® bricks se utilizan para representar notas musicales, pero también se pueden utilizar para representar conceptos computacionales como la secuenciación y la abstracción.

La actividad "LEGO® Musical Notation" utiliza LEGO® ladrillos para representar notas musicales. Los alumnos pueden utilizar los ladrillos para crear sus propias composiciones musicales. La investigación encontró que la actividad "LEGO® Musical Notation" es una forma efectiva de enseñar pensamiento computacional a los niños. Los alumnos que participaron en la actividad mostraron una mejora significativa en su comprensión de los conceptos básicos del pensamiento computacional (Vlahović & Biškupić, 2023).

- **sCool--Game-Based Learning in Computer Science Class: A Case Study in Secondary Education:**

El aprendizaje basado en juegos (ABJ) es una metodología educativa que utiliza juegos para enseñar conceptos y habilidades. El ABJ ha demostrado ser una herramienta eficaz para el aprendizaje en una variedad de contextos, incluidos los entornos educativos formales e informales.

En el contexto de la educación en ciencias de la computación, el ABJ ha sido utilizado para enseñar conceptos básicos de programación, como la descomposición, la secuenciación, la abstracción y la evaluación. Los resultados de estudios realizados sobre el uso del ABJ en la educación en ciencias de la computación han sido positivos, mostrando que los estudiantes que participan en actividades de ABJ obtienen mejores resultados en las evaluaciones de ciencias de la computación que los estudiantes que no participan.

El artículo "sCool – Game-Based Learning in Computer Science Class: A Case Study in Secondary Education" presenta un estudio de caso sobre el uso del juego educativo sCool para enseñar ciencias de la computación en clases de secundaria. El juego está diseñado para ayudar a los estudiantes a aprender conceptos básicos de programación de una manera divertida y engaging. Los resultados del estudio mostraron que los estudiantes que utilizaron el juego sCool obtuvieron mejores resultados en las evaluaciones de ciencias de la computación que los estudiantes que no lo utilizaron. Los estudiantes que utilizaron el juego también mostraron una mayor motivación por aprender ciencias de la computación (Teinmaurer, Pirker, & Gütl, 2019).

A continuación se presenta una tabla que compara las características entre los proyectos antes vistos y el presente.

Tabla 3*Comparación entre proyectos*

Características	Prototipo de aplicación web como herramienta de apoyo para la evaluación de Competencias en Pensamiento Computacional	Programa de Tutoría en Pensamiento Computacional	Iniciativa Internacional Bebras	LEGO®-Based Music Notation	sCool - Game-Based Learning in Computer Science Class
Evaluación de Competencias en Pensamiento Computacional	✓	✓	✓	✓	✓
Desarrollo de una Aplicación Web	✓	X	✓	X	X
Retroalimentación a Estudiantes	✓	✓	X	X	X
Personalización de Evaluaciones	✓	X	X	X	X
Seguimiento y Reportes de Progreso	✓	✓	✓	X	✓

Nota. Elaboración propia (González & Estrada, 2024).

3.2 Marco teórico

- Competencias en educación:** Las competencias se definen como "un conjunto de conocimientos, habilidades, actitudes y valores que permiten a las personas enfrentar y resolver problemas en contextos diversos" (Tobón, 2005). En el ámbito educativo, las competencias se consideran un objetivo fundamental de la enseñanza, ya que permiten a los estudiantes desarrollar las habilidades necesarias para ser ciudadanos autónomos y responsables. La educación centrada en competencias busca no solo la adquisición de conocimientos teóricos, sino también el desarrollo práctico de

habilidades aplicables a situaciones del mundo real, fomentando así una preparación integral para los desafíos actuales (Tobón, 2005).

- **Pensamiento computacional:** En este contexto, el pensamiento computacional se convierte en una habilidad esencial para el siglo XXI y se define como "la capacidad de pensar y resolver problemas de manera lógica y sistemática, utilizando herramientas y técnicas de la informática" (Wing, 2006). El pensamiento computacional se compone de una serie de competencias que pueden agruparse en tres categorías: competencias de pensamiento creativo, competencias de pensamiento crítico y competencias de pensamiento algorítmico (Wing, 2006). Fomentar el pensamiento computacional en la educación no solo prepara a los estudiantes para futuras carreras en tecnología, sino que también fortalece sus habilidades cognitivas, mejorando su capacidad para abordar desafíos complejos de manera estructurada y creativa.
- **Evaluación de competencias en pensamiento computacional:** La evaluación desempeña un papel crucial en la medición de estas competencias por lo que es un proceso continuo que permite medir el progreso de los estudiantes y proporcionar retroalimentación para mejorar su aprendizaje, específicamente en el contexto del pensamiento computacional, es vital emplear modelos de evaluación que permitan medir las competencias de manera integral. Existen diversos modelos de evaluación que pueden aplicarse para medir las competencias en pensamiento computacional, siendo la evaluación formativa y la evaluación sumativa los más comunes (Black & Wiliam, 1998). La evaluación formativa se utiliza para proporcionar retroalimentación a lo largo del proceso educativo y ayudar a los estudiantes a mejorar sus habilidades, mientras que la evaluación sumativa se enfoca en medir el nivel de competencia alcanzado al final de un período de aprendizaje.

- **Rúbricas de evaluación:** Para medir de manera objetiva las competencias en pensamiento computacional, las rúbricas de evaluación son herramientas esenciales ya que estas rúbricas describen los criterios y los niveles de desempeño esperados en una evaluación (González-Calero, 2023). Además de proporcionar una evaluación cuantitativa, las rúbricas también ofrecen una retroalimentación detallada sobre áreas específicas de mejora, facilitando un enfoque personalizado para el desarrollo de habilidades en pensamiento computacional (González-Calero, 2023).
- **Aplicaciones web:** El desarrollo de una aplicación web parametrizable es una vía para abordar la evaluación de estas competencias. La aplicación requiere de una metodología de desarrollo que permita diseñar, desarrollar e implementar una aplicación que pueda adaptarse a diferentes necesidades (Pressman, 2019). La versatilidad de las aplicaciones web no solo facilita la evaluación, sino que también promueve la accesibilidad y la participación activa de los estudiantes, creando un entorno interactivo que mejora la experiencia de aprendizaje en el pensamiento computacional.

3.3 Marco conceptual

- **Abstracción:** La abstracción es un proceso mental que nos permite centrarnos en los aspectos esenciales de un problema, dejando de lado los detalles irrelevantes. Esto nos permite comprender mejor el problema y encontrar soluciones más eficientes (García & Sánchez, 2024).
- **Descomposición:** La descomposición es un proceso mental que nos permite dividir un problema en partes más pequeñas, que son más fáciles de comprender y resolver. Podemos pensar en la descomposición como un proceso de "partir" un problema en "pedazos" (Pérez & Rodríguez, 2024).

- **Algoritmos:** Una serie de pasos ordenados que se utilizan para resolver un problema (Ferrovial, s. f.).
- **Programación:** El proceso de escribir instrucciones para que una computadora realice una tarea. (Concepto.de, s. f.).
- **Pensamiento crítico:** El pensamiento crítico es un conjunto de habilidades y estrategias que nos permiten evaluar el razonamiento y las afirmaciones de forma objetiva. Es una habilidad importante en el mundo actual, ya que nos permite tomar decisiones informadas y evitar ser engañados (García & Sánchez, 2024).
- **Resolución de problemas:** La capacidad de identificar y resolver problemas.
- **Creatividad:** La capacidad de generar ideas nuevas y originales.
- **Competencia de abstracción:** Los estudiantes pueden identificar los elementos esenciales de un problema y representarlos de forma simplificada (Pérez & Rodríguez, 2024).
- **Competencia de descomposición:** Los estudiantes pueden dividir un problema complejo en problemas más pequeños y manejables.
- **Rúbrica de resolución de problemas:** Esta rúbrica se puede utilizar para evaluar la capacidad de los estudiantes para identificar y resolver problemas.
- **Rúbrica de pensamiento crítico:** Esta rúbrica se puede utilizar para evaluar la capacidad de los estudiantes para analizar información y llegar a conclusiones basadas en evidencia.

3.4 Glosario

- **Pensamiento computacional:** Habilidad cognitiva que implica abordar problemas de manera lógica, descomponer tareas en partes manejables, identificar patrones y diseñar soluciones algorítmicas.
- **Reconocimiento de patrones:** Habilidad para identificar regularidades o tendencias en datos y problemas, facilitando la creación de soluciones eficientes.
- **Secuenciación:** Organización de pasos o instrucciones en una secuencia lógica para lograr un resultado deseado.
- **Condicionales:** Instrucciones que se ejecutan sólo si se cumple una condición determinada, permitiendo la toma de decisiones en el proceso algorítmico.
- **Educación primaria y secundaria:** Etapas del sistema educativo donde se busca introducir y desarrollar las competencias de pensamiento computacional de manera progresiva y adaptada a la edad.
- **Programación:** Habilidad para escribir y entender código informático, que es una manifestación práctica del pensamiento computacional.
- **Evaluación formativa:** Proceso continuo de evaluación que proporciona retroalimentación en tiempo real, permitiendo a los estudiantes mejorar constantemente sus habilidades de pensamiento computacional.
- **K-12:** El sistema educativo K-12 es un sistema educativo que comprende los grados de prejardín a doceavo grado.

Capítulo 4

Metodología

4.1. Metodología de investigación

La metodología de investigación para este proyecto de grado se fundamenta en enfoques cualitativos y cuantitativos. Dado que el objetivo central es implementar una aplicación web para evaluar y dar seguimiento a las competencias en pensamiento computacional de estudiantes en el nivel de educación secundaria, se busca establecer una metodología que garantice un desarrollo efectivo y una evaluación integral (Godoy, 2023).

Etapas de Investigación:

- **Indagación inicial:** Para construir una base sólida de conocimiento y comprender a fondo el contexto educativo en el nivel de educación secundaria, se llevó a cabo una investigación exhaustiva. Esto implicó estudiar las necesidades específicas de los estudiantes, analizar los enfoques actuales en pensamiento computacional y comprender los desafíos y oportunidades que se presentan en este entorno.
- **Recopilación de información:** Con un entendimiento completo del contexto, la fase de recopilación de datos comenzó utilizando diversas fuentes disponibles. Además, se buscó la colaboración de expertos en educación y pensamiento computacional para enriquecer la información recopilada y ajustarla a las necesidades específicas del proyecto.
- **Planteamiento de herramientas:** En esta etapa, se seleccionó cuidadosamente las herramientas tecnológicas apropiadas para el desarrollo de la aplicación web. Esto incluye la elección del lenguaje de programación, frameworks y otras tecnologías relevantes que permitan una implementación efectiva y eficiente de la aplicación.
- **Diseño e implementación del prototipo:** Con las herramientas definidas, se procederá al diseño detallado de la aplicación web. La implementación del prototipo

se realizó priorizando la ejecución de las funcionalidades clave para evaluar las competencias en pensamiento computacional. El seguimiento constante garantizó que el desarrollo cumpla con los estándares de calidad y los requisitos establecidos.

- **Evaluación y ajustes:** Finalmente, se llevó a cabo una evaluación exhaustiva de la aplicación implementada. Se definió un grupo de muestra de usuarios reales para realizar pruebas y recopilar retroalimentación. La información obtenida se utilizó para ajustar la aplicación, asegurando que responda de manera efectiva a las necesidades de evaluación y seguimiento de competencias en pensamiento computacional.

Capítulo 5

Métodos de Evaluación de Competencias en Pensamiento Computacional

En este capítulo se presentan diversos métodos de evaluación del Pensamiento Computacional (PC) encontrados en la literatura reciente. El análisis se centra en cómo estos métodos abordan la evaluación de competencias clave en PC, como la abstracción, la descomposición y el pensamiento algorítmico, y qué herramientas utilizan como rúbricas, actividades prácticas o estrategias de enseñanza para validar el proceso evaluativo.

5.1. Métodos de evaluación del pensamiento computacional

- **Método de evaluación del pensamiento computacional y alfabetización en código:**

El método propuesto por Diana Carolina López-Gómez, José Luis Sierra-Rodríguez y María del Carmen Suárez-Castañón, en el artículo titulado *“El método de evaluación del pensamiento computacional y alfabetización en código”* (López-Gómez, Sierra-Rodríguez, & Suárez-Castañón, 2021). Este estudio plantea una alternativa innovadora para medir habilidades asociadas tanto al PC como a la alfabetización en código (AC). Específicamente, define el pensamiento computacional como un conjunto de destrezas que involucran la descomposición de problemas, el reconocimiento de patrones, la abstracción, el diseño de algoritmos y su expresión mediante lenguajes de programación.

Cabe señalar que, aunque el estudio no presenta una metodología estructurada en fases, sí se describen detalladamente las actividades aplicadas a cada dimensión evaluada, a diferencia de otros enfoques que incorporan dinámicas lúdicas, este método se centra exclusivamente en la resolución de tareas como principal estrategia evaluativa.

En cuanto al objetivo del estudio, este se orienta a valorar la capacidad de los participantes para aplicar sus conocimientos en la resolución de problemas. Los resultados, por su parte, respaldan la validez y confiabilidad del método, al evidenciar diferencias notables en el rendimiento de los evaluados según las dimensiones del PC y la AC.

Si bien el artículo no especifica de forma explícita el tipo de aprendizaje utilizado, se puede inferir que la base conceptual de la metáfora corresponde a la computación como proceso de resolución de problemas. No obstante, el estudio reconoce ciertas limitaciones, tales como el tamaño reducido de la muestra y su aplicación restringida a población adulta.

- **Evaluación del pensamiento computacional en educación: un marco de referencia y una propuesta:** M.^a Ángeles Martínez-Sánchez, Pedro J. Muñoz-Merino y M.^a Teresa Sancho-Vinuesa, presentan una propuesta metodológica en el artículo titulado “*Evaluación del Pensamiento Computacional en Educación: Un Marco de Referencia y una Propuesta de Evaluación*” (Martínez-Sánchez, Muñoz-Merino, & Sancho-Vinuesa, 2017). en el cual se realiza un análisis profundo sobre instrumentos que permiten evaluar el PC en contextos escolares, especialmente en educación primaria.

A diferencia del estudio anterior, este se basa en el modelo TPACK (Technological Pedagogical and Content Knowledge), el cual integra tres dimensiones del conocimiento: disciplinar, pedagógico y tecnológico, por lo tanto esta integración resulta clave para garantizar una evaluación holística del pensamiento computacional.

El PC es concebido aquí como un proceso mental que permite resolver problemas de manera eficiente, y se aborda desde tres dimensiones fundamentales: conceptos, habilidades y actitudes, el estudio se aplicó a estudiantes de educación primaria, quienes participaron en un curso de 12 sesiones utilizando el entorno de programación Scratch. En este marco, la evaluación se estructuró en tres fases claramente definidas: diagnóstico, intervención y evaluación final.

Para medir el progreso de los estudiantes, se utilizaron dos rúbricas específicas: una para evaluar conceptos y otra para habilidades. Ambas se organizaron en cuatro niveles de logro, que iban desde “insuficiente” hasta “excelente”. Además, se empleó una estrategia pedagógica basada en el aprendizaje por proyectos, donde los estudiantes aplican sus conocimientos a través del desarrollo de productos en Scratch. Como resultado, se evidenció una mejora significativa en las tres dimensiones evaluadas al finalizar el curso. En cuanto a las reflexiones finales, los autores destacan tanto la integralidad del modelo como la utilidad de las rúbricas para lograr una evaluación objetiva y fiable. Sin embargo, también se reconocen desafíos, entre ellos la necesidad de más estudios que refuercen la validez del enfoque y la complejidad inherente a su implementación.

- **Evaluación del pensamiento computacional mediante resolución de problemas:**

Se presenta el estudio desarrollado por Beatriz Ortega-Ruipérez, María Jesús Moreno-León y Fernando Sancho-Vinuesa, titulado *“Evaluar el Pensamiento Computacional mediante Resolución de Problemas: Validación de un Instrumento de Evaluación”* (Ortega-Ruipérez, Moreno-León, & Sancho-Vinuesa, 2021). Este trabajo constituye una propuesta concreta para evaluar el PC en estudiantes de Educación Primaria (EP) y Educación Secundaria Obligatoria (ESO), mediante una batería de ítems orientados a la resolución de problemas.

El PC, en este contexto, se define como una competencia transversal que implica descomponer problemas, identificar patrones, abstraer elementos y diseñar algoritmos. La población del estudio estuvo conformada por 400 estudiantes de 4º y 6º de primaria, así como de 1º de secundaria. Para su evaluación, se empleó un conjunto de 15 ítems divididos en dos bloques: uno centrado en la descomposición y el reconocimiento de patrones, y otro enfocado en la abstracción y los algoritmos.

Cada ítem fue valorado utilizando rúbricas que incluían criterios como claridad, precisión, organización, creatividad y corrección, estos fueron clasificados en distintos niveles de desempeño, que permitieron observar el progreso de los estudiantes en cada una de las dimensiones.

Los resultados evidenciaron un mejor desempeño en los estudiantes de secundaria en comparación con los de primaria. Además, las competencias asociadas a la descomposición y al reconocimiento de patrones mostraron resultados superiores, mientras que los ítems de algoritmos representaron una mayor dificultad. Aunque el artículo no menciona expresamente el tipo de aprendizaje implementado, se deduce que la estrategia general también se basa en la resolución de problemas como eje estructural.

- **Método de evaluación de competencias de pensamiento computacional:** Por último, se encuentra el *Método Clasificador de Competencias en Pensamiento Computacional*, desarrollado como parte de la tesis doctoral de Yuri Mercedes Bermúdez (Bermúdez Mazuera, s. f.). A diferencia de los métodos anteriormente mencionados, este enfoque propone una estructura sistemática basada en la noción de competencia, estableciendo una relación directa entre el nivel de desempeño y los resultados de aprendizaje (RA).

El método identifica tres competencias principales: abstracción, descomposición y pensamiento algorítmico. Cada una se desglosa en cuatro RA, que a su vez contienen criterios de evaluación (CE) específicos. Para evaluar cada RA, se utiliza una rúbrica general de tipo cuantitativo-cualitativo, la cual permite asignar niveles de logro (básico, intermedio o avanzado) en función del puntaje obtenido en actividades previamente diseñadas.

Posteriormente, los puntajes se ponderan para determinar el nivel global de competencia alcanzado por el estudiante. Además, se dispone de rúbricas específicas para cada competencia, como por ejemplo la rúbrica de abstracción, lo que facilita una evaluación más detallada, donde se desagregan los CE en Indicadores de logro. Como valor agregado, el método ofrece retroalimentación personalizada al finalizar el proceso, orientando al estudiante según su desempeño en cada dimensión.

5.2. Comparación de métodos

Tabla 4

Comparación Métodos.

			Evaluación del Pensamiento Computacional en Educación:	Evaluar el Pensamiento Computacional mediante Resolución de
		El método de		

CARACTERÍSTICA	Método de evaluación de competencias de PC	evaluación del pensamiento computacional y alfabetización en código	Un Marco de Referencia y una Propuesta de Evaluación	Problemas: Validación de un Instrumento de Evaluación
POBLACIÓN OBJETIVO	Estudiantes.	Adultos participantes en curso binacional de formación en PC y AC.	Estudiantes de Educación Primaria y Secundaria Obligatoria.	Estudiantes de Educación Primaria y Secundaria Obligatoria.
QUE SE EMPLEÓ PARA LA EVALUACIÓN	Competencia de PC (Abstracción, Descomposición, Pensamiento Algorítmico).	Tareas de descomposición de problemas, escritura y comprensión de código en Python.	Modelo TPACK, Curso de Scratch.	Resolución de problemas de algoritmos.
COMPETENCIAS EVALUADAS	Abstracción.	Pensamiento computacional y alfabetización en código.	Conceptos, Habilidades, Actitudes.	Descomposición, Reconocimiento de Patrones, Abstracción, Algoritmos.
RÚBRICAS	Rúbricas generales y específicas.	Rúbricas para evaluar dimensiones del PC y AC.	Rúbricas para conceptos y habilidades.	Rúbricas para cada ítem de evaluación.
RETROALIMENTACIÓN	Ofrece retroalimentación personalizada.	No ofrece.	No ofrece.	Ofrece retroalimentación general.
FLEXIBILIDAD	Puede adaptarse a diferentes contextos educativos.	Enfoque específico para adultos.	Enfoque específico en la educación primaria.	Enfoque específico para estudiantes de EP y ESO.
ESTRATEGIA	Rúbricas para evaluar dimensiones del PC.	Resolución de tareas como estrategia principal.	Aprendizaje basado en proyectos.	Evaluación basada en la resolución de problemas.

Nota. Elaboración propia (González 2025).

5.3. Conclusión

Como se puede observar en la Tabla 3 el **Método clasificador de Competencias en Pensamiento Computacional**, se destaca por su claridad, estructura, y flexibilidad, lo que permite adaptarse a diferentes contextos educativos, esto sumado a su enfoque integral y objetivo, junto con la validación rigurosa proporcionada por las rúbricas, lo convierten en la mejor opción para evaluar el Pensamiento Computacional de manera precisa y confiable, contando además con una retroalimentación personalizada que la diferencia de las otras metodologías expuestas.

Capítulo 6

Estructura del Método de Evaluación por Competencias en Pensamiento

Computacional

En este capítulo se presenta el diseño estructural del método de evaluación por competencias que se implementó para evaluar el desarrollo del Pensamiento Computacional (PC). Esta propuesta se fundamenta en el *Método de Evaluación de Competencias de Pensamiento Computacional*, desarrollado por Bermúdez (Bermúdez Mazuera, s. f.), el cual contempla exclusivamente la competencia de abstracción dentro de su estructura evaluativa.

A partir de dicha competencia, el diseño se organiza en cuatro Resultados de Aprendizaje (RA): interiorización, coordinación, encapsulación y generalización. En consecuencia, se definen los criterios de evaluación específicos y la rúbrica cuantitativa correspondiente, permitiendo establecer un sistema de evaluación objetivo y alineado a través de Resultados de Aprendizaje claramente definidos en el Marco conceptual de competencias de PC definido.

6.1 Competencia de pensamiento computacional que se evaluó

En primer lugar, se especifica que la competencia a evaluar será la **abstracción**, entendida como la capacidad para modelar problemas, identificar lo esencial y transitar entre distintos niveles de detalle con el fin de encontrar soluciones computacionales. Esta competencia se estructura en cuatro RA:

- **Interiorización**, que implica la comprensión profunda del problema y su expresión simbólica.
- **Coordinación**, que permite articular distintas abstracciones para construir soluciones válidas.

- **Encapsulación**, orientada a transformar procesos dinámicos en representaciones estáticas.
- **Generalización**, que busca extrapolar soluciones específicas a problemas similares.

Cada una de estas RA contribuyen al desarrollo de esta competencia del pensamiento computacional, esencial en la resolución de problemas dentro del ámbito computacional.

6.2 Criterios de evaluación

A continuación, se presenta el Marco conceptual de la competencia de Abstracción, en el cual cada Resultado de Aprendizaje (RA) se desagrega en Criterios de Evaluación (CE). Cada criterio, a su vez, posee un nivel de logro asignado (Básico, Intermedio o Avanzado), permitiendo medir el grado de desempeño del estudiante frente a la competencia.

- **Interiorización:**
 - **CE1: Identificación de características fundamentales.** El estudiante debe reconocer los elementos esenciales del problema, justificando su importancia en la solución.
 - **CE2: Uso de lenguaje simbólico.** Se espera la capacidad de representar dichas características mediante un lenguaje formal y simbólico.
- **Coordinación**
 - **CE1: Generación de abstracciones a través de representaciones simbólicas.** El estudiante debe crear representaciones simbólicas que reflejan estados válidos del problema.
 - **CE2: Combinación de abstracciones para estrategias de solución.** Se debe evidenciar la capacidad de combinar abstracciones generadas para llegar a un estado final resolutivo.
- **Encapsulación**

- **CE1: Transformación de procesos dinámicos en estados estáticos.** Se evalúa la habilidad de representar procesos complejos como estados fijos y válidos.
- **CE2: Construcción de nuevas abstracciones para estrategias de solución.** El estudiante debe desarrollar abstracciones coherentes para cada estado, empleando un lenguaje simbólico.

- **Generalización**

- **CE1: Identificación de formas simbólicas para generalización.** El estudiante debe reconocer estructuras que le permitan transferir soluciones a otros problemas.
- **CE2: Comprensión y justificación de estrategias de generalización.** Se requiere la capacidad de explicar la aplicabilidad y adaptabilidad de la solución a diversos contextos.

Con esto se expone las tablas que desagrega cada Criterio de Evaluación (CE) por Resultado de Aprendizaje (RA), utilizando los niveles de logro **Básico, Intermedio y Avanzado**. Estos niveles corresponden a los propuestos en el **Método de Evaluación de Competencias de Pensamiento Computacional**.

Tabla 5

Rúbrica interiorización

criterio de evaluación (CE)	nivel básico	nivel intermedio	nivel alto
CE1	Identifica con dificultad algunas de las características fundamentales de un problema y cuando logra hacerlo, se le dificulta distinguir las razones válidas que justifican que un elemento del problema sea una característica fundamental	Identifica la mayoría de las características fundamentales de un problema; pero aún se le dificulta distinguir las razones válidas que justifican que un elemento del problema sea una característica fundamental	Identifica las características fundamentales de un problema y distingue con total claridad las razones válidas que justifican que un elemento del problema sea una característica fundamental del mismo.

	del mismo. puntaje: 8	del mismo. puntaje: 15	puntaje: 30
CE2	Tiene dificultades para representar las características de un problema haciendo uso de un lenguaje simbólico y cuando logra hacerlo, se le dificulta representar estados válidos de una solución descritos de un problema de manera gráfica o en lenguaje natural y/o simbólico. puntaje: 25	Logra representar algunas de las características de un problema haciendo uso de un lenguaje simbólico; pero aún se le dificulta representar estados válidos de una solución descritos de un problema en manera gráfica o en lenguaje natural y/o simbólico. puntaje: 40	Logra representar las características de un problema haciendo uso de un lenguaje simbólico y representa estados válidos de solución descritos de un problema en manera gráfica o en lenguaje natural y/o simbólico. puntaje: 70

Nota. Elaboración propia (Estrada, 2025).

Tabla 6

Rúbrica Coordinación

criterio de evaluación (CE)	nivel básico	nivel intermedio	nivel alto
CE1	Tiene dificultades para representar nuevos estados de solución de un problema a partir de un estado válido específico representado por medio de un lenguaje simbólico y no logra identificar las razones para justificar la validez de un estado de un problema. puntaje: 8	Logra representar algunos estados nuevos de solución de un problema a partir de un estado válido específico representado por medio de un lenguaje simbólico, pero aún se le dificulta identificar las razones para justificar la validez de un estado de un problema. puntaje: 20	Logra representar todos los nuevos estados de solución de un problema a partir de un estado válido específico representado por medio de un lenguaje simbólico y logra identificar las razones para justificar la validez de un estado de un problema. puntaje: 40
CE2	Tiene dificultades para construir nuevos estados válidos del problema utilizando un lenguaje simbólico para su representación, partiendo de un estado inicial y no logra llegar a un estado final del mismo. puntaje: 20	Logra construir la mayoría de los estados nuevos válidos del problema utilizando un lenguaje simbólico para su representación, partiendo de un estado inicial pero no logra llegar a un estado final del mismo. puntaje: 40	Logra construir todos los estados válidos del problema utilizando un lenguaje simbólico para su representación, partiendo de un estado inicial hasta llegar a un estado final del mismo, construyendo una estrategia de solución del problema. puntaje: 60

Nota. Elaboración propia (González 2025).

Tabla 7

Rúbrica Encapsulación

criterio de	nivel básico	nivel intermedio	nivel alto
--------------------	---------------------	-------------------------	-------------------

evaluación (CE)			
CE1	Tiene dificultades para construir representaciones estáticas de estados válidos del problema que representan estados dinámicos proporcionados por medio de un lenguaje simbólico, y no logra identificar las razones para justificar la validez de una representación estática de un estado del mismo. puntaje: 8	Logra construir algunas representaciones estáticas de estados válidos del problema que representan estados dinámicos proporcionados por medio de un lenguaje simbólico, pero aún se le dificulta identificar las razones para justificar la validez de una representación estática de un estado del mismo. puntaje: 20	Logra construir todas las nuevas representaciones estáticas de estados válidos de un problema que representan estados dinámicos proporcionados por medio de un lenguaje simbólico, y logra identificar las razones para justificar la validez de una representación estática de un estado del mismo. puntaje: 40
CE2	Tiene dificultades para construir nuevas representaciones estáticas de estados válidos de un problema a partir de representaciones dinámicas proporcionadas por medio de un lenguaje simbólico definido, partiendo de un estado inicial y no logra llegar a un estado final del mismo. puntaje: 20	Logra construir la mayoría de las representaciones estáticas de estados válidos de un problema a partir de representaciones dinámicas proporcionadas por medio de un lenguaje simbólico definido; partiendo de un estado inicial, pero no logra llegar a un estado final del mismo. puntaje: 40	Logra construir todas las representaciones estáticas de distintos estados válidos de un problema a partir de representaciones dinámicas proporcionadas por medio de un lenguaje simbólico definido; partiendo de un estado inicial hasta llegar a un estado final del mismo, construyendo una estrategia de solución del problema. puntaje: 60

Nota. Elaboración propia (Estrada 2025).

Tabla 8

Rúbrica Generalización

criterio de evaluación (CE)	nivel básico	nivel intermedio	nivel alto
CE1	No logra identificar la forma de generalizar simbólicamente la relación entre las características de un problema particular para aplicar a problemas similares. puntaje: 10	Tiene dificultades para identificar formas de generalizar simbólicamente la relación entre las características de un problema particular para aplicar a problemas similares. puntaje: 20	Logra identificar formas de generalizar simbólicamente la relación entre las características de un problema particular para aplicar a problemas similares. puntaje: 50
CE2	Tiene dificultades para identificar la estrategia que le permite resolver un problema particular y no logra identificar la razón por la cual la	Logra identificar una estrategia que le permite resolver un problema particular, pero se le dificulta identificar la razón por la	Logra identificar la estrategia que le permite resolver un problema particular; así mismo, identifica la razón por la cual la generalización

	generalización de un problema en particular representa una estrategia general del problema. puntaje: 7	cual la generalización de un problema particular representa una estrategia general del problema. puntaje: 12	de un problema particular representa una estrategia general del problema. puntaje: 50
--	---	---	--

Nota. Elaboración propia (Gonzalez, 2025).

6.3 Indicadores de logro

Posteriormente, se definen los Indicadores de Logro (IL) que permiten observar y medir de forma concreta el desempeño del estudiante frente a cada Criterio de Evaluación (CE), dentro de un Resultado de Aprendizaje (RA). En este modelo, los indicadores constituyen la unidad más específica del proceso evaluativo, ya que describen comportamientos observables o evidencias puntuales que reflejan el nivel de apropiación de una competencia.

Cada IL se valora también en niveles de Básico, Intermedio y Avanzado, lo cual permite precisar aún más el grado de dominio demostrado por el estudiante en las actividades diseñadas para su evaluación.

- **Interiorización.**

- **CE1:** Identificación de características fundamentales.
 - **IL1:** Selecciona de un listado de opciones las características más representativas del juego.
 - **IL2 :** Identifica las razones por las cuales un elemento del juego es una característica fundamental del mismo.
- **CE2:** Uso de lenguaje simbólico.
 - **IL1:** Comprende el lenguaje simbólico definido para expresar características representativas del juego
 - **IL2 :** Comprende qué es un estado válido del juego de acuerdo a las reglas del mismo.

- **IL3** : Emplea el lenguaje simbólico definido para representar estados válidos del juego.
- **Coordinación.**
 - **CE1:** Generación de abstracciones a través de representaciones simbólicas.
 - **IL1:** Identifica estados válidos y no válidos del juego utilizando el lenguaje simbólico definido para expresar características representativas del mismo, además identifica las razones que permiten justificar la validez de un estado del juego.
 - **IL2** : Construye una secuencia lógica de nuevos estados válidos del juego representados por medio de un lenguaje simbólico definido.
 - **CE2:** Combinación de abstracciones para estrategias de solución.
 - **IL1:** Utiliza el lenguaje simbólico definido para generar a partir de un estado inicial válido del juego, varios estados dinámicos hasta llegar al estado final del mismo, cumpliendo cada una de las reglas del juego.
- **Encapsulación.**
 - **CE1:** Transformación de procesos dinámicos en estados estáticos.
 - **IL1:** Identifica representaciones estáticas válidas y no válidas del juego a partir de representaciones dinámicas, utilizando para ello, el lenguaje simbólico definido para expresar características representativas del mismo, además identifica las razones que permiten justificar la validez de una representación estática de un estado del juego.
 - **IL2** : Construye una secuencia lógica de representaciones estáticas de estados válidos del juego a partir de representaciones dinámicas expresadas por medio de un lenguaje simbólico definido.

- **CE2:** Construcción de nuevas abstracciones para estrategias de solución.
 - **IL1:** Genera nuevas representaciones estáticas válidas de cada uno de los estados válidos del juego, partiendo de un estado inicial hasta un estado final, utilizando el lenguaje simbólico definido para expresar una estrategia de solución del juego.
- **Generalización.**
 - **CE1:** Identificación de formas simbólicas para generalización.
 - **IL1:** Generaliza de manera simbólica la relación entre la cantidad de discos y el menor número de movimientos del juego, después de hallar el menor número de movimientos para mover $n \geq 1$ discos (mayor o igual) siguiendo las reglas del juego.
 - **CE2:** Comprensión y justificación de estrategias de generalización.
 - **IL1:** Comprende la estrategia identificada para ganar el juego de las Torres de Hanoi en la menor cantidad de movimientos posibles de acuerdo a una cantidad n de discos y tres torres.

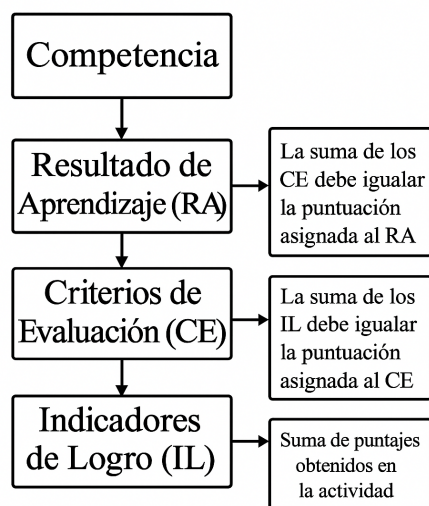
6.4 Estructura y formato del método de evaluación

Finalmente, se presenta la rúbrica cuantitativa estructurada jerárquicamente que permite valorar el nivel de desempeño del estudiante a partir de criterios objetivos y medibles. Este método ofrece una ruta sistemática que comienza desde la actividad evaluativa más concreta hasta alcanzar el Resultado de Aprendizaje (RA), garantizando coherencia interna, trazabilidad y precisión en la asignación de niveles de logro.

La evaluación se organiza en los siguientes niveles jerárquicos:

Figura 1

Niveles jerárquicos de la evaluación



Nota. La figura detalla los niveles jerárquicos en que se divide la evaluación. Fuente:

Elaboración propia (González 2025).

Cada Indicador de Logro (IL) representa una evidencia observable asociada a un CE, y se evalúa mediante actividades específicas, como pruebas, ejercicios o simulaciones. Los puntajes obtenidos en estas actividades se suman para asignar un nivel de logro al IL, de acuerdo con tres rangos porcentuales establecidos en la rúbrica:

- **Nivel Básico:** cuando el puntaje obtenido es menor o igual al 40 % de la ponderación asignada al IL.
- **Nivel Intermedio:** si el puntaje se encuentra entre el 41 % y el 80 %.
- **Nivel Avanzado:** cuando el puntaje supera el 81% y hasta el 100 % de la ponderación.

Es importante resaltar que los indicadores de logro pueden ser definidos o adaptados por el evaluador, siempre y cuando se cumpla con la condición esencial de que la suma de sus

valores represente exactamente el puntaje total asignado al criterio de evaluación, garantizando así consistencia interna.

Una vez determinado el nivel de cada IL, se procede a la asignación del nivel cualitativo del Criterio de Evaluación (CE) mediante la suma de los valores cuantitativos alcanzados por cada IL que lo compone. La valoración del CE se realiza comparando dicha suma con un umbral de referencia denominado “Umbral de Básico”, el cual se construye tomando como base el valor cuantitativo asignado al nivel Básico del IL con mayor ponderación en la rúbrica específica, más la suma de los valores correspondientes al nivel Intermedio de los demás IL que integran dicho criterio. Este procedimiento debe ser aplicado individualmente a cada CE, ya que cada uno posee una configuración distinta de indicadores y ponderaciones.

La asignación del nivel al CE se hace con base en los siguientes criterios:

- Si la suma total de los IL es menor o igual al umbral de básico calculado, se asigna el nivel básico al CE.
- Si la suma es igual al valor máximo posible asignado al CE (correspondiente al nivel Avanzado según la rúbrica), se otorga el nivel Avanzado.
- Si la suma no cumple ninguna de las dos condiciones anteriores, se asigna el nivel Intermedio.

Posteriormente, con los niveles asignados a cada CE, se determina el nivel del Resultado de Aprendizaje (RA). Para ello, se suman los valores cuantitativos de todos los criterios asociados al RA y se compara el resultado con el Umbral de Básico para la Clasificación del RA, así como con el valor máximo de 100 puntos, definido en la rúbrica general:

- Si la suma es menor o igual al umbral de básico, se asigna al RA el nivel Básico
- Si la suma es menor o igual a 80 puntos, se otorga el nivel intermedio.

- En los demás casos, se asigna el nivel Alto.

En resumen, el proceso de evaluación sigue esta secuencia:

1. Asignación de puntajes a las actividades que evalúan cada IL.
2. Determinación del nivel del IL con base en su ponderación.
3. Suma de valores cuantitativos para determinar el nivel del Criterio de Evaluación (CE), comparando el resultado con el umbral de básico.
4. Acumulación de los valores de los CE para establecer el nivel del Resultado de Aprendizaje (RA), contrastando con su umbral de básico y el puntaje máximo.

Cabe destacar que tanto los IL como los CE y los RA se clasifican utilizando los mismos umbrales de nivel (básico, intermedio, avanzado), lo que permite mantener una coherencia estructural en todo el proceso evaluativo.

6.5 Conclusión

De esta manera, se establece una estructura sólida y detallada para la evaluación de la competencia de abstracción en Pensamiento Computacional, basada en el *Método de evaluación de Competencias* de PC y adaptada a través de rúbricas cuantitativas, criterios de evaluación, indicadores de logro y Resultados de aprendizaje. Esta estructura no solo organiza el proceso evaluativo de manera objetiva y gradual, sino que también sirvió como base para el desarrollo funcional del prototipo web propuesto en este trabajo.

A partir de esta rúbrica, se diseñó una aplicación web dirigida al docente evaluador, que le permita gestionar su proceso de evaluación de manera eficiente, el sistema contempla la posibilidad de que cada docente cree un perfil, registre cursos, y construya evaluaciones personalizadas con base en las plantillas predeterminadas derivadas de esta investigación. Estas evaluaciones podrán ser editadas, guardadas y reutilizadas en diferentes cursos.

Una vez aplicadas, el sistema procesa automáticamente los puntajes obtenidos por los estudiantes en los indicadores de logro, asignando niveles por criterio y resultado de aprendizaje, de acuerdo con los umbrales establecidos. Además, el docente podrá visualizar los resultados de forma clara y comprensible, mediante gráficos estadísticos interactivos, que facilitarán la interpretación del desempeño individual y grupal.

En resumen, la estructura metodológica desarrollada en este capítulo constituye el modelo base para la implementación del prototipo web.

Capítulo 7

Desarrollo del prototipo de aplicación web que incorpora el método de evaluación de competencias de PC

7.1. Metodología de Desarrollo

El prototipo se desarrolló siguiendo la metodología ágil Scrum, reconocida por su capacidad para mantener una supervisión constante del progreso del proyecto. Se optó por organizar sprints semanales con plazos cortos, durante los cuales se definen tareas que permanecen fijas. Debido a la disponibilidad limitada de los integrantes del proyecto, se decidió prescindir de las reuniones diarias. En su lugar, la evaluación retrospectiva del sprint se llevaba a cabo justo antes de planificar el siguiente sprint (Evidencia de product backlog en los anexos del documento).

7.2 Requerimientos de la aplicación

7.2.1 Requerimientos funcionales

- El sistema deberá permitir el acceso a través de un formulario de login con usuario y contraseña, restringiendo el acceso a usuarios no autenticados.
- El sistema deberá permitir el registro de nuevos usuarios mediante un formulario de inscripción que incluya nombre completo, correo, usuario y contraseña.
- El sistema deberá permitir la creación y edición de cursos, incluyendo el nombre del curso y el nombre de la institución educativa.
- El sistema deberá permitir la gestión de estudiantes por curso: agregar, listar y eliminar estudiantes manualmente o mediante archivos CSV.
- El sistema deberá permitir la creación de evaluaciones personalizadas, basadas en criterios, indicadores y preguntas, con distribución de puntajes y niveles de logro definidos.

- El sistema deberá permitir aplicar una evaluación a un grupo, ingresando los puntajes de los estudiantes por cada pregunta generada según el resultado de aprendizaje.
- El sistema deberá validar automáticamente que los puntajes ingresados no excedan los máximos establecidos para cada pregunta e indicador.
- El sistema deberá calcular y almacenar los niveles de logro por Resultado de Aprendizaje (RA), a partir de los puntajes registrados, según criterios cualitativos preestablecidos.
- El sistema deberá presentar los resultados por estudiante y RA.
- El sistema deberá permitir generar gráficos que representen la distribución de puntajes por RA para el análisis visual de resultados grupales.
- El sistema deberá ofrecer una sección de configuración donde el usuario pueda editar su información personal, cambiar la contraseña y actualizar su foto de perfil.

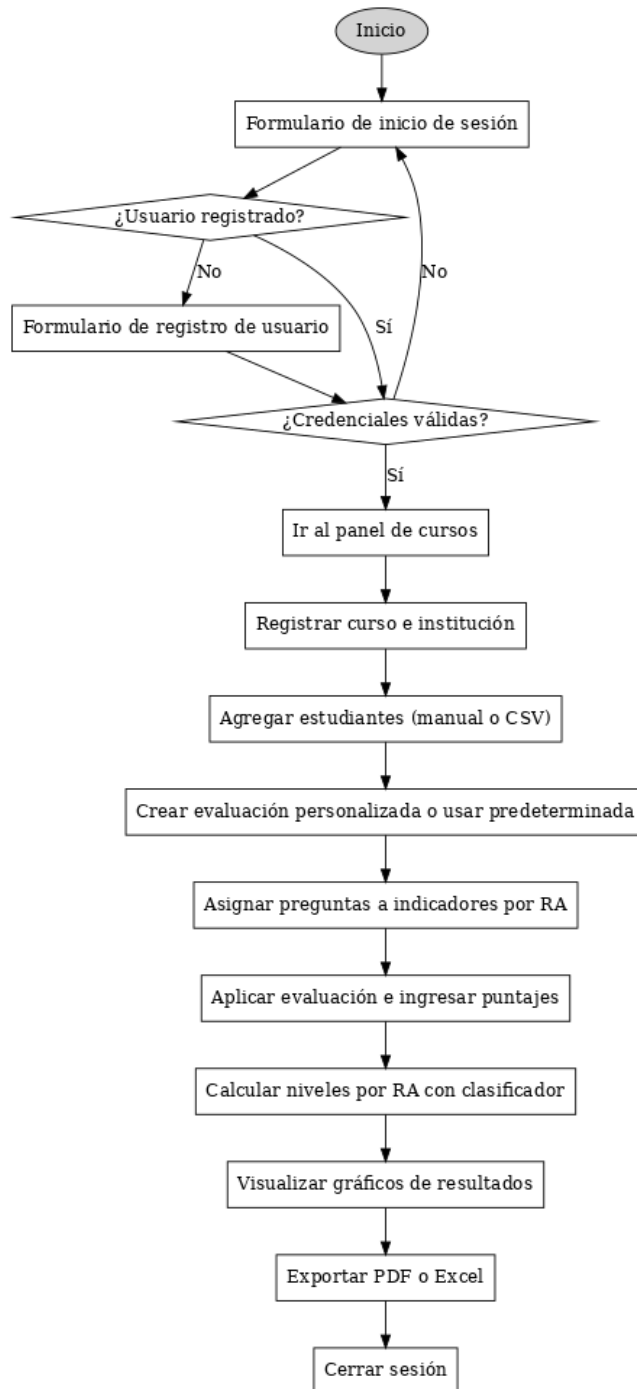
7.2.2 Requerimientos no funcionales

- El sistema no garantiza que los niveles asignados representen de forma absoluta el nivel real de competencia del estudiante, ya que se basa en una rúbrica cualitativa y criterios definidos por el usuario.
- El sistema debe ser completamente utilizable mediante una interfaz web accesible desde navegadores modernos.
- La aplicación debe mantener la coherencia visual y de navegación a lo largo de todas sus vistas, haciendo uso de estilos consistentes y elementos interactivos intuitivos.
- El sistema debe procesar correctamente archivos en formato CSV para la carga de estudiantes y puntajes, siguiendo una estructura predefinida que será validada por la interfaz.

A continuación, se presenta el diagrama de flujo general del funcionamiento del sistema, el cual resume las etapas clave que sigue el usuario dentro de la plataforma:

Figura 2

Diagrama de flujo de la aplicación



Nota. La figura detalla el diagrama de flujo de la aplicación. Fuente: Elaboración propia
(González & Estrada, 2024).

7.3. Justificación de las tecnologías

Para el desarrollo de la aplicación web orientada a la evaluación de competencias en pensamiento computacional, se consideraron diversas tecnologías y lenguajes de programación ampliamente utilizados en el desarrollo web moderno. Las decisiones tecnológicas fueron tomadas con base en criterios como la facilidad de aprendizaje, escalabilidad, robustez, comunidad de soporte y compatibilidad con los objetivos del proyecto.

Entre las posibles alternativas se evaluaron frameworks como Django (Python), Laravel (PHP) y Ruby on Rails, sin embargo, se optó por una pila tecnológica basada en **Node.js** con **Express**, por ser una opción ligera, altamente modular, de rápido desarrollo y ampliamente adoptada en la industria. Esta elección permite una mayor flexibilidad al momento de estructurar la aplicación, así como una integración fluida con herramientas de frontend y bases de datos.

Para el motor de plantillas, se seleccionó **EJS (Embedded JavaScript Templates)**, el cual facilita la generación dinámica de vistas HTML desde el servidor utilizando JavaScript. Asimismo, se decidió utilizar **MySQL** como sistema de gestión de bases de datos relacional por su robustez, escalabilidad y compatibilidad con el entorno de desarrollo local y de producción.

Para la autenticación y protección de rutas, se integraron herramientas como **JWT (JSON Web Token)** y **bcrypt**, las cuales permiten una gestión segura de sesiones de usuario y el cifrado de contraseñas.

7.4. Descripción de las tecnologías de desarrollo

A continuación, se describen las tecnologías y dependencias principales utilizadas en el desarrollo de la aplicación:

7.4.1. Node.js y Express.

Node.js es un entorno de ejecución para JavaScript en el servidor. Permite construir aplicaciones escalables y de alto rendimiento (Kinsta, 2024). Sobre este se construyó la aplicación utilizando **Express**, un framework minimalista y flexible que facilita la creación de rutas, middlewares y lógica de servidor.

- **Arquitectura modular:** Permite organizar el código en rutas, controladores y servicios, mejorando el mantenimiento del sistema.
- **Manejo de solicitudes HTTP:** Express facilita el manejo de rutas, peticiones POST, GET, PUT y DELETE.
- **Middlewares personalizados:** Posibilita agregar funciones intermedias como validaciones, logs o manejo de errores.

7.4.2. EJS.

EJS (Embedded JavaScript Templates) es un motor de plantillas que permite insertar lógica de JavaScript directamente en el HTML (EJS, 2024). Se utilizó para renderizar páginas dinámicas del lado del servidor, como el panel administrativo, formularios, y resultados de evaluaciones.

- **Integración con Express:** Funciona de manera natural con Express, permitiendo pasar datos del backend al frontend de manera sencilla.
- **Reutilización de vistas:** Facilita el uso de plantillas parciales y estructuras repetitivas como cabeceras, pies de página y menús.

7.4.3. MySQL.

MySQL es un sistema de gestión de bases de datos relacional ampliamente utilizado por su estabilidad, eficiencia y soporte en múltiples plataformas (Kinsta, 2024).

- **Modelo relacional:** Permite estructurar la información en tablas relacionadas, lo cual es ideal para representar entidades como usuarios, actividades y resultados.
- **Consultas eficientes:** Su compatibilidad con SQL estándar facilita la implementación de consultas complejas y operaciones CRUD.

7.4.4. JWT (JSON Web Token).

JWT se utilizó para la gestión de autenticación basada en tokens. Los usuarios autenticados reciben un token cifrado, el cual es utilizado para verificar su identidad en peticiones posteriores (OpenWebinars, 2024).

- **Autenticación sin sesión:** Evita la necesidad de mantener sesiones en el servidor.
- **Transporte seguro:** Los tokens se envían como cabeceras HTTP, y pueden ser verificados en cada petición protegida.

7.4.5. Bcrypt y Bcryptjs

Bcrypt y **Bcryptjs** son bibliotecas utilizadas para el cifrado de contraseñas (Crenshaw, 2022). Ambas implementan el algoritmo bcrypt, diseñado para proteger contraseñas mediante hashing seguro.

- **Hashing con salt:** Protege contra ataques de diccionario y fuerza bruta.
- **Comparación segura:** Permite verificar contraseñas ingresadas sin revelar las originales.
- **Compatibilidad:** bcryptjs se utiliza como alternativa en entornos donde las dependencias nativas de bcrypt pueden presentar problemas de instalación.

7.4.6. Cookie-parser.

Cookie-parser es un middleware que permite analizar cookies enviadas por el cliente (Express.js, 2024). Se utilizó para gestionar información como tokens o preferencias del usuario de manera sencilla.

- **Lectura de cookies:** Extrae automáticamente las cookies presentes en la petición.

- **Integración con sesiones:** Trabaja junto con `express-session` para mantener la persistencia de estado del usuario.

7.4.7. CSV-parser.

CSV-parser es un módulo para procesar archivos `.csv` (Python Software Foundation, 2024). Se utilizó principalmente para importar datos estructurados, como resultados o registros masivos, desde hojas de cálculo o sistemas externos.

- **Procesamiento eficiente:** Permite leer grandes volúmenes de datos sin bloquear el hilo principal.
- **Fácil integración:** Compatible con flujos de lectura nativos de `Node.js`.

7.4.8. Dotenv.

Dotenv es una herramienta para manejar variables de entorno. Permite definir configuraciones sensibles como claves de API, puertos o credenciales en un archivo `.env` separado del código fuente (npm, 2024).

- **Separación de configuración:** Mejora la seguridad y flexibilidad del proyecto.
- **Carga automática:** Al iniciar el servidor, las variables son cargadas en `process.env`.

7.4.9. Express-session.

Express-session permite implementar sesiones de usuario persistentes en la aplicación (Express.js, 2024).

- **Almacenamiento en memoria o persistente:** Las sesiones pueden ser almacenadas localmente o en bases de datos como Redis.
- **Seguimiento de usuarios:** Útil para recordar el estado de autenticación o preferencias.

7.4.10. Multer.

Multer es un middleware para manejar la carga de archivos enviados desde formularios web (Nkubito, 2023).

- **Soporte para multipart/form-data:** Necesario para subir imágenes, documentos u otros archivos.
- **Configuración flexible:** Permite definir límites de tamaño, ubicación de almacenamiento y tipos de archivo aceptados.

Se presenta el gráfico de representación de las tecnologías implementadas.

Figura 3

Representación de las tecnologías implementadas



Nota. La figura detalla el diagrama de integración tecnológica. Fuente: Elaboración propia (Estrada, 2024).

7.5. Diseño de la base de datos

El sistema de evaluación fue desarrollado utilizando una base de datos relacional en MySQL, gestionada a través de la interfaz phpMyAdmin, ampliamente empleada en entornos con servidores locales como XAMPP.

La base de datos lleva por nombre compensamiento y su diseño responde a una estructura normalizada que facilita la gestión eficiente de datos relacionados con usuarios, cursos, estudiantes, evaluaciones, rúbricas y resultados. Su propósito principal es permitir el almacenamiento estructurado y seguro de la información necesaria para evaluar competencias mediante criterios e indicadores.

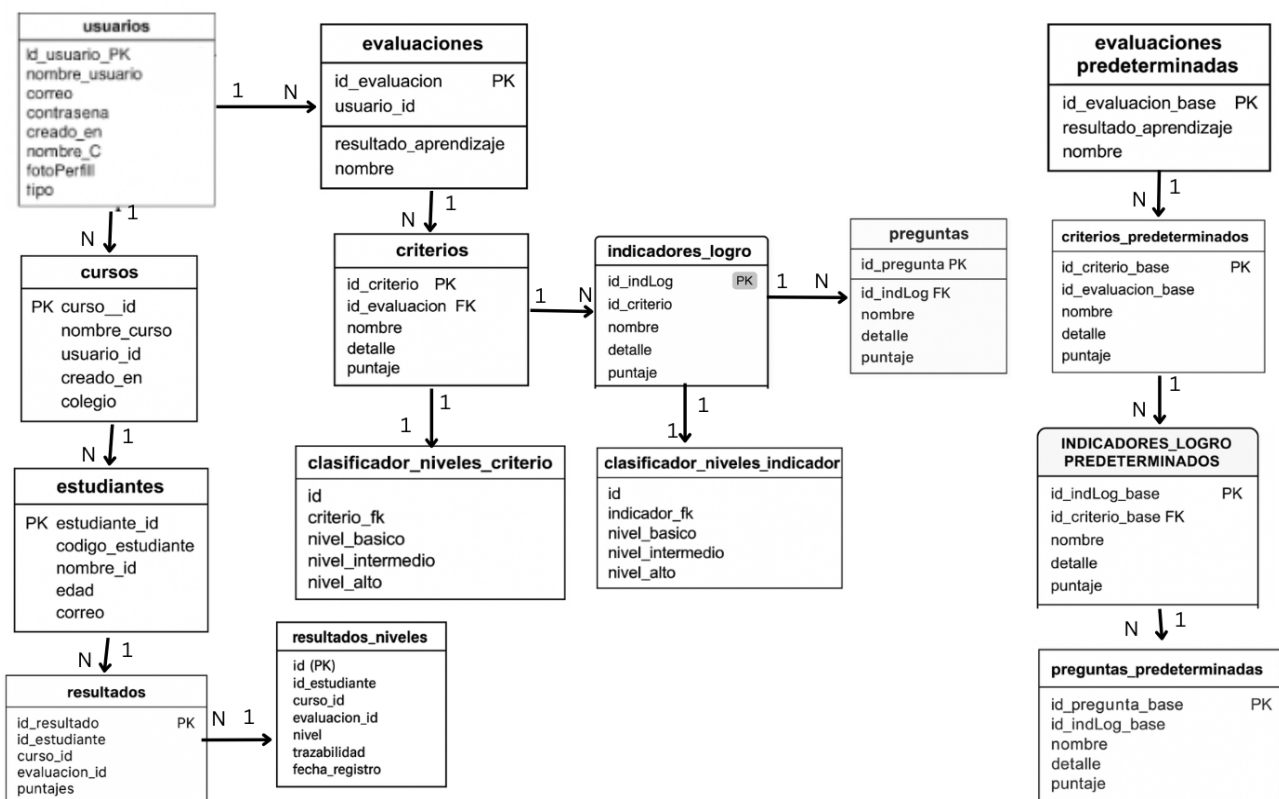
Las tablas más representativas de este modelo incluyen:

- **Usuarios:** contiene la información de registro e inicio de sesión de los profesores que acceden al sistema.
- **Cursos y estudiantes:** almacenan los cursos creados por los usuarios y los estudiantes asociados a cada uno, permitiendo su gestión individual o masiva (por carga CSV).
- **Evaluaciones y evaluaciones_predeterminadas:** distinguen entre evaluaciones creadas por el usuario y aquellas establecidas como base común para múltiples cursos.
- **Criterios, indicadores, preguntas:** forman la estructura jerárquica que representa la rúbrica evaluativa, permitiendo una descomposición pedagógica del aprendizaje.
- **Clasificador_niveles_criterio y clasificador_niveles_indicador:** permiten configurar los umbrales de logro (básico, intermedio, avanzado) en función del puntaje máximo definido.

- **Resultados_niveles:** registra los resultados de cada estudiante luego de aplicar una evaluación, incluyendo una trazabilidad detallada en formato JSON que asocia los puntajes obtenidos con los niveles alcanzados.

La estructura fue implementada mediante sentencias CREATE TABLE con claves primarias y foráneas para garantizar integridad referencial, tipos de datos apropiados (como DECIMAL para puntajes y TEXT para trazabilidad) y operaciones de inserción y consulta optimizadas. Este diseño asegura la trazabilidad educativa, la generación de estadísticas, y la exportación de resultados en formatos PDF y Excel, cumpliendo con los objetivos funcionales y analíticos del sistema. A continuación, se presenta la Figura 4, que ilustra de manera estructurada el diseño de la base de datos implementada.

Figura 4 *Diseño de base de datos.*



Nota. La figura detalla el diseño de base de datos de la aplicación. Fuente: Elaboración propia (Gonzalez, 2024).

7.6 Diseño y demostración de las vistas de usuario

Inicio y registro de la aplicación.

Como se observa en el Anexo A. Imagen 1, donde se encuentran varias evidencias del diseño de las vistas, la aplicación cuenta con dos apartados principales en su interfaz de inicio: uno destinado al registro de nuevos usuarios y otro para el inicio de sesión. Estos formularios permiten gestionar el acceso seguro a la plataforma y son el punto de entrada para todas las funcionalidades del sistema. La organización visual de ambos formularios en una misma vista facilita la experiencia del usuario y optimiza la navegación desde el primer momento de uso.

Para el diseño del registro y del login de la aplicación se muestran mensajes claros para que sea adaptable a cualquier tipo de usuario, es un diseño con descripciones detalladas y claras para que los usuarios puedan entender de manera intuitiva qué información es solicitada para cada caso. La imagen 1 del anexo A está compuesta por un título claro, dos campos el primero donde se solicita el correo registrado, y el segundo la contraseña, por último un botón para iniciar la sesión. La imagen 2 del anexo A muestra un título claro y un formulario de registro, donde se solicitan varios campos de información el primero el nombre completo del usuario a registrar, segundo el correo electrónico, tercero un nombre de usuario, cuarto una contraseña correspondiente y por último un botón de registro.

La imagen 3 del anexo A muestra el inicio de la aplicación y el menú de la misma, en el apartado de inicio se pueden ver tres botones que muestra información de la aplicación el

primero un botón con un interrogante ¿quiénes somos?, el segundo un botón “Nuestro Objetivo”, y el tercero un botón de funcionamiento.

En la Figura 5 se puede observar el apartado de cursos el cual muestra un botón de añadir cursos y muestra los cursos añadidos previamente que si oprimes sobre ellos se despliega un menú de opciones primero añadir estudiantes al curso, mostrar estudiantes del curso, editar el curso, eliminar el curso, duplicar el curso.

Figura 5

Apartado de cursos de la aplicación.



Nota. La figura muestra el diseño del apartado de cursos de la aplicación. Fuente:

Elaboración propia (Gonzalez, 2025).

En la Figura 6 se muestra el menú de gestión de evaluaciones donde se puede observar tres selectores el primero de competencia el cual está predefinido como abstracción, el segundo de seleccionar evaluación, y el tercero de seleccionar curso. También se puede observar varios botones uno de crear evaluación, otro de ver la evaluación otro para agregar estudiantes, y uno de cargar puntajes desde un .csv también se puede observar una tabla

donde se mostrarán los estudiantes del curso seleccionado y los puntos de la evaluación seleccionada.

En la Figura 7 se muestra el menú desplegado del botón crear evaluación, en el cual se ven dos selectores uno de resultado de aprendizaje y el otro de nombre de la evaluación; también el apartado de modificación de los puntajes de cada uno de los criterios de evaluación.

Figura 6 *Apartado de evaluaciones*

Nota. La figura muestra el diseño del apartado de evaluaciones de la aplicación. Fuente:

Elaboración propia (Gonzalez, 2025).

Figura 7

Botón de crear nueva evaluación

Nota. La figura muestra el diseño del panel para crear una evaluación en la aplicación.

Fuente: Elaboración propia (Estrada, 2025).

En la Figura 8 se muestra el apartado de resultados donde se puede observar varios selectores, el primer selector de curso luego un mensaje claro para realizar el cálculo completo de la competencia se debe seleccionar en todos los selectores, el segundo selector es el de interiorización, el tercero de coordinación, el cuarto de encapsulación y el quinto de generalización, luego un botón de ver los resultados y una tabla donde se mostrarán los resultados y por último un botón donde se verán los gráficos del análisis estadístico de los resultados.

Figura 8

Apartado de resultados

Nota. La figura muestra el diseño del apartado de resultados de la aplicación. Fuente:

Elaboración propia (Gonzalez, 2025).

En la imagen 4 del anexo A se puede observar el apartado de configuraciones, en el cual se muestra la información del perfil del usuario, un botón para editar el perfil y un botón para cambiar la contraseña.

7.7 Justificación del diseño visual, enfoque de usabilidad y paleta de colores

El diseño del prototipo fue cuidadosamente planeado para ser intuitivo, funcional y centrado en el usuario, teniendo como público principal a los docentes encargados de gestionar procesos de evaluación y seguimiento de competencias. En cuanto al manejo de archivos .CSV, se procuró que este aspecto también respondiera al enfoque de usabilidad. La aplicación presenta de forma clara el formato esperado, incluyendo una imagen de ejemplo que facilita la comprensión por parte del usuario. Esta decisión busca reducir posibles errores en la carga de información, permitiendo que el docente tenga una referencia visual que guíe adecuadamente la preparación del archivo.

El objetivo central fue garantizar que cualquier usuario, sin necesidad de formación técnica avanzada, pudiera interactuar con la aplicación de manera sencilla y eficiente, accediendo rápidamente a la información y funcionalidades más relevantes.

Para lograrlo, se aplican varios principios clave de diseño:

- **Simplicidad:** La interfaz fue construida con un enfoque minimalista, eliminando elementos innecesarios para reducir la carga cognitiva. Las vistas se diseñaron con una distribución lógica y organizada, permitiendo que los usuarios comprendan y naveguen de manera natural por las distintas secciones como cursos, evaluaciones o resultados.
- **Jerarquía visual:** Las funcionalidades principales se destacan a través del uso de botones con íconos representativos y colores contrastantes, lo cual facilita la identificación rápida de acciones como crear evaluaciones, ver resultados o añadir estudiantes. Esta jerarquización mejora la eficiencia del uso y reduce los errores por parte del usuario.

- **Accesibilidad:** Se utilizaron **fuentes legibles**, tamaños de texto adecuados y una estructura visual clara, pensada para ser accesible incluso para usuarios con poca experiencia digital. El diseño también contempla buenas prácticas de contraste de color para garantizar la lectura en distintas condiciones visuales.
- **Enfoque en la usabilidad:** El prototipo permite que los usuarios interactúen con la aplicación sin necesidad de instrucciones adicionales. Las vistas responden a una lógica visual coherente y familiar, lo que favorece una curva de aprendizaje baja y una experiencia positiva desde el primer uso.

En cuanto a la **paleta de colores**, se utilizó una combinación de **azul, blanco y gris**, seleccionados cuidadosamente para transmitir profesionalismo, orden y confianza:

- El **azul**, color predominante en la barra lateral y elementos destacados, simboliza responsabilidad, estabilidad y confianza, atributos esenciales en contextos educativos. Su uso también contribuye a una navegación clara y guía visual constante.
- El **blanco**, presente en los fondos principales, ofrece un entorno limpio y sin distracciones, lo cual potencia la legibilidad de los textos y facilita la concentración en las tareas.
- El **gris**, aplicado en íconos y textos secundarios, genera contraste equilibrado sin generar ruido visual, reforzando la estructura jerárquica de la información.

La estructura de navegación, representada visualmente en la figura correspondiente, incluye un menú lateral persistente que brinda acceso inmediato a las secciones esenciales del sistema (Inicio, Cursos, Evaluaciones, Resultados, Configuraciones), promoviendo una experiencia fluida y sin interrupciones. Todo este conjunto de decisiones de diseño refleja una apuesta por la inclusión digital, la eficacia funcional y la adopción natural por parte de los docentes.

7.8 Descripción técnica del prototipo

Registro

Esta funcionalidad permite a los usuarios crear una cuenta en el sistema mediante un formulario que solicita información básica como nombre completo, nombre de usuario, correo electrónico y contraseña. Una vez diligenciado el formulario, el sistema verifica si el correo ya ha sido utilizado. Si no existe, los datos se almacenan en la base de datos y se habilita el acceso al sistema. En caso de detectar un correo duplicado o un campo incompleto, se lanza un mensaje de advertencia informando al usuario sobre el error.

Si el usuario ya tiene una cuenta, puede acceder desde el formulario de inicio de sesión. En este formulario se validan las credenciales contra los registros existentes en la base de datos. Si la información coincide, se establece una sesión y se redirige al menú principal; si no coincide, se muestra un mensaje de error y no se permite el ingreso.

Menú principal.

Después del inicio de sesión exitoso, el usuario accede al menú principal. Esta vista permite navegar entre las funcionalidades del sistema como creación de cursos, gestión de estudiantes, aplicación de evaluaciones, visualización de resultados y configuración del perfil. El menú se presenta en forma de tarjetas interactivas que permiten seleccionar la acción deseada sin necesidad de recargar toda la página.

Creación de cursos.

Desde el menú principal, el usuario puede acceder a un formulario para registrar un nuevo curso. En este formulario se solicita el nombre del curso y el nombre de la institución. Después de guardar el curso, se habilita la opción para agregar estudiantes. Los estudiantes pueden añadirse uno por uno o mediante un archivo CSV estructurado. El sistema valida que no existan duplicados y que el archivo cargado cumpla con el formato requerido.

Creación de evaluaciones.

Esta funcionalidad permite al usuario diseñar evaluaciones nuevas o seleccionar evaluaciones predeterminadas. Para una evaluación personalizada, el usuario define los resultados de aprendizaje, criterios de evaluación, indicadores de logro y preguntas correspondientes. A cada pregunta se le asigna un puntaje máximo, y se establecen los niveles de logro esperados según umbrales predefinidos. La información se estructura jerárquicamente y se almacena en la base de datos para su posterior aplicación.

Aplicación de evaluaciones.

Una vez creada la evaluación, el usuario puede seleccionarla para aplicarla a un curso registrado. Para cada estudiante, se ingresan los puntajes obtenidos por pregunta. El sistema valida automáticamente que el puntaje no exceda el valor máximo definido. Los resultados se almacenan y se utilizan para calcular los niveles por resultado de aprendizaje según la configuración del clasificador.

Visualización de resultados.

El usuario puede consultar los resultados por estudiante y por resultado de aprendizaje. La interfaz muestra los niveles alcanzados en cada RA usando colores diferenciados. Se incluye un botón para expandir la vista y mostrar el detalle de criterios, indicadores y puntajes. Adicionalmente, el sistema genera representaciones gráficas como bigotes(boxplot), radar, barras comparativas y mapas de calor, los cuales permiten realizar análisis visuales del desempeño.

- **Boxplot (Gráfico de caja y bigotes):** Implementado mediante Plotly en la función `renderBoxplotRA`, este gráfico muestra la dispersión de los puntajes obtenidos por los estudiantes en un Resultado de Aprendizaje (RA) específico.

Utilidad para el análisis:

- Permite visualizar rápidamente la mediana, cuartiles y valores atípicos del puntaje global de los estudiantes en un RA.
 - Facilita identificar la variabilidad del rendimiento en un grupo, detectando posibles brechas o desigualdades.
 - Es útil para evaluar la homogeneidad del aprendizaje: un boxplot estrecho indica consistencia entre estudiantes, mientras que uno amplio puede evidenciar diferencias significativas en el dominio de la competencia.
- **Gráfico Radar:** Implementado en `renderRadarRA`, este gráfico tiene dos modos:
- Grupo: Muestra el promedio por criterio dentro de un RA.
 - Estudiante: Muestra el nivel individual por criterio para un estudiante.

Utilidad para el análisis:

- En modo grupo, permite comparar visualmente el dominio colectivo por criterio, detectando fortalezas y debilidades comunes.
 - En modo estudiante, ofrece una vista integral y personalizada del desempeño, ayudando a identificar criterios específicos que requieren refuerzo.
 - Su forma permite evaluar si el rendimiento está equilibrado entre criterios o si hay descompensaciones significativas en la adquisición de habilidades.
- **Barras Comparativas por Nivel:** A través de `renderBarrasNivelesRA`, este gráfico permite comparar la cantidad de estudiantes por nivel de desempeño (Básico, Intermedio, Avanzado), agrupados:
- Por Resultado de Aprendizaje.
 - Por criterio específico.

Utilidad para el análisis:

- Muestra la distribución de niveles alcanzados en el grupo, permitiendo identificar qué proporción necesita apoyo (niveles bajos) o consolidación (niveles altos).
 - Al agruparse por criterio, permite analizar cuáles aspectos dentro de un RA son más desafiantes o dominados.
 - Es útil para tomar decisiones pedagógicas basadas en los porcentajes de estudiantes en cada nivel.
- **Barras por Estudiante:** Implementado en `renderBarrasPorEstudianteRA`, representa el nivel alcanzado por un único estudiante en todos los RA seleccionados.

Utilidad para el análisis:

- Proporciona una visión comparativa del rendimiento individual en los diferentes RAs.
 - Ayuda a identificar si un estudiante tiene un rendimiento consistente o si existen áreas en las que presenta dificultades específicas.
 - Facilita la generación de recomendaciones individualizadas y el seguimiento del progreso estudiantil.
- **Mapa de Calor (Heatmap):** Desarrollado en `renderHeatmapRA`, muestra una matriz con los estudiantes en el eje Y y los RAs en el eje X, codificando el nivel de logro con colores.

Utilidad para el análisis:

- Permite identificar patrones colectivos de desempeño.
- Facilita la detección visual de estudiantes con bajo rendimiento constante o destacada excelencia.
- Es particularmente útil para análisis masivos, ya que sintetiza en un solo gráfico el rendimiento general de toda la cohorte educativa.

Exportación de resultados.

La plataforma ofrece la posibilidad de exportar los resultados en formato PDF y Excel. Esta exportación puede realizarse desde el módulo de resultados mediante un botón específico que genera y descarga los archivos automáticamente.

Capítulo 8

Pruebas

En esta sección se muestra el proceso de pruebas del prototipo y un análisis de los resultados obtenidos.

8.1. Pruebas de Usabilidad

Para medir la usabilidad del prototipo, se aplicó un cuestionario estructurado a **25 docentes e instructores** que utilizaron el sistema. El objetivo fue recoger información cuantitativa y cualitativa sobre la experiencia de uso, evaluar la curva de aprendizaje, identificar puntos críticos de interacción y verificar si el prototipo cumple con las expectativas funcionales básicas. La encuesta se realizó durante el mes de **abril de 2025** y fue aplicada mediante un formulario de **Google Forms** compartido a través de enlaces enviados por correo y redes de comunicación institucional.

8.2. Objetivo y planteamiento de las pruebas

8.2.1. Objetivos

- **Evaluar la Experiencia del Usuario (UX):** Obtener información sobre la experiencia general del usuario al interactuar con el prototipo web, incluyendo la facilidad de navegación, la claridad de la interfaz y la satisfacción general.
- **Validar Decisiones de Diseño:** Validar la efectividad de las decisiones de diseño tomadas durante el desarrollo del prototipo web, asegurándose de que cumplan con las necesidades y expectativas de los usuarios.
- **Evaluar la Comprensión de Resultados:** Analizar si los usuarios comprenden correctamente los niveles de logro mostrados (básico, intermedio, avanzado), así como la trazabilidad presentada por criterio e indicador dentro de cada resultado de aprendizaje.

- **Evaluar la Eficiencia de las Interacciones:** Medir la rapidez y claridad con la que los usuarios pueden completar tareas fundamentales como crear cursos, registrar estudiantes, configurar evaluaciones, ingresar puntajes y consultar resultados.
- **Identificar Áreas de Mejora:** Identificar áreas específicas del prototipo web que pueden ser mejoradas en términos de funcionalidad, usabilidad o diseño, basándose en los comentarios y sugerencias de los usuarios.
- **Proporcionar Datos para Interacciones Futuras:** Utilizar los datos recopilados durante las pruebas para informar y guiar las iteraciones futuras del prototipo web, con el objetivo de mejorar continuamente la experiencia del usuario y la efectividad del sistema de recomendación.

8.2.2. Planteamiento de las pruebas

La primera sección de preguntas se basó en la escala de Likert, un tipo de instrumento de medición o recolección de datos comúnmente utilizado en investigación social para evaluar actitudes y opiniones. Esta escala consta de un conjunto de ítems en forma de afirmaciones o juicios, frente a los cuales se solicita a los participantes que expresen su reacción en una escala que va desde:

- Totalmente en desacuerdo (1)
- En desacuerdo (2)
- Neutral (3)
- De acuerdo (4)
- Totalmente de acuerdo (5)

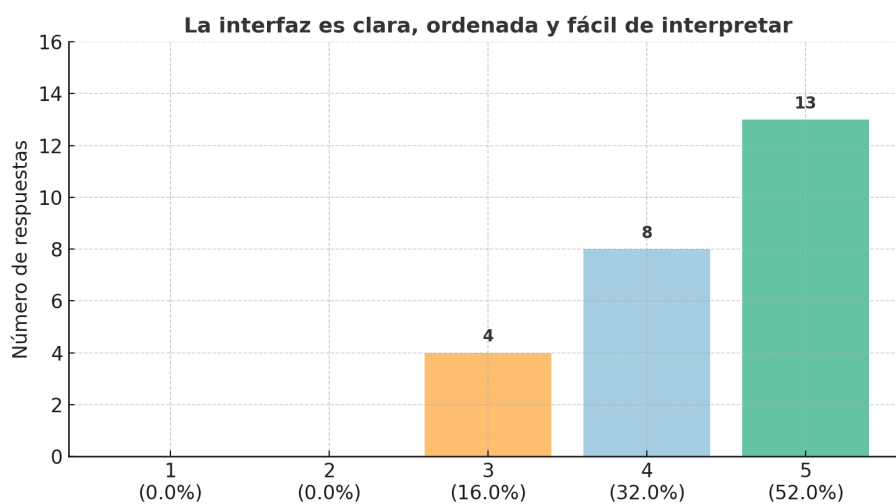
Este enfoque permite capturar tanto la dirección como la intensidad de las percepciones de los participantes frente a las funcionalidades y la experiencia de uso del sistema.

El cuestionario fue respondido por un total de **25 docentes e instructores**, todos vinculados a instituciones educativas. Los participantes cuentan con experiencia en procesos de evaluación y manejo básico de herramientas digitales. Varios de ellos hacen parte del cuerpo docente del Servicio Nacional de Aprendizaje (SENA), donde están habituados a registrar evidencias, aplicar instrumentos de evaluación y emitir resultados en plataformas institucionales. Esta experiencia previa permitió obtener valoraciones fundamentadas sobre la funcionalidad, facilidad de uso e impacto del prototipo en contextos reales de enseñanza.

A continuación, se presentan dos de los principales resultados obtenidos en la encuesta aplicada a los usuarios. Los demás gráficos y resultados detallados pueden consultarse en el anexo B.

Figura 9

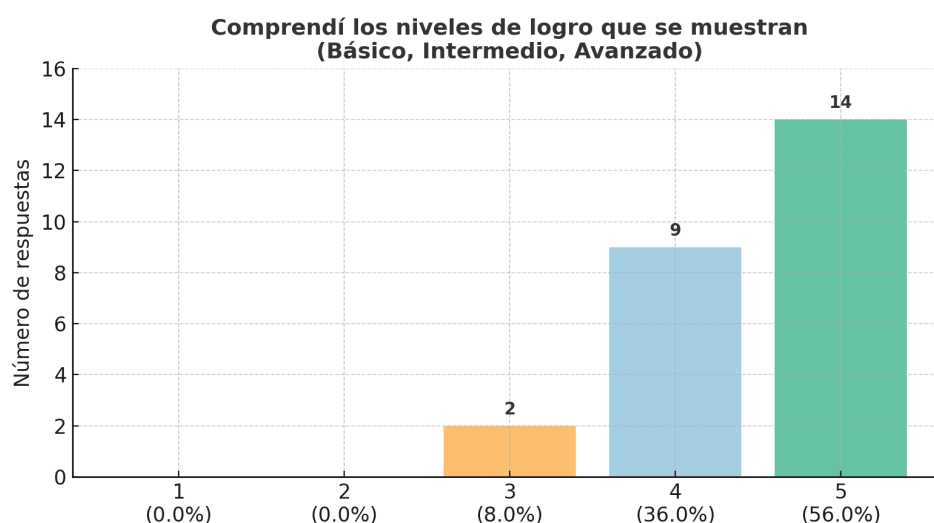
Pregunta 1



Nota. La figura muestra los resultados obtenidos. Fuente: Elaboración propia (González & Estrada, 2025).

Figura 10

Pregunta 2



Nota. La figura muestra los resultados obtenidos. Fuente: Elaboración propia (González & Estrada, 2025).

8.3. Análisis de resultados.

8.3.1 Atractivo de la presentación.

El diseño gráfico y la organización visual de la interfaz fueron evaluados como un componente fundamental de la experiencia del usuario, los resultados obtenidos reflejan un alto nivel de aceptación, lo que sugiere que la estructura visual del sistema cumple con las expectativas de los veinticinco docentes y profesionales que participaron en la prueba.

Según la figura 9, el 52 % de los encuestados indicó estar “*muy satisfecho*” con la claridad y orden de la interfaz, seleccionando el nivel 5 en la escala de Likert, este porcentaje representa una mayoría significativa y resalta la efectividad de los elementos visuales empleados, tales como la disposición jerárquica de secciones, la consistencia en el uso de tipografías, y la diferenciación por colores en los módulos clave del sistema.

Adicionalmente, un 32 % manifestó estar “*satisfecho*” seleccionando el nivel 4, lo que refuerza la percepción positiva generalizada. Esta evaluación indica que, aunque algunos

usuarios no consideraron la experiencia completamente sobresaliente, sí la encontraron adecuada, comprensible y funcional para el desarrollo de sus tareas pedagógicas.

El 16 % restante seleccionó el nivel 3, correspondiente a una opinión *neutral*. Aunque este grupo no expresó insatisfacción directa, representa un segmento relevante para la mejora continua.

Es importante destacar que no se registraron respuestas en los niveles 1 ni 2, lo cual confirma que ningún usuario consideró la interfaz como confusa, desorganizada o difícil de usar.

8.3.2. Comprensión de los niveles de logro.

La comprensión de los niveles de logro utilizados por el sistema Básico, Intermedio y Avanzado fue evaluada positivamente por la gran mayoría de los usuarios. Esta dimensión fue considerada clave, ya que representa la manera en que se traduce la puntuación cuantitativa en información cualitativa relevante para la evaluación del aprendizaje.

Según la figura 10, el 56 % de los encuestados seleccionó el nivel 5, indicando una comprensión plena del significado de cada categoría, este resultado sugiere que los elementos gráficos y textuales utilizados para representar los niveles (etiquetas, colores y disposición) resultaron adecuados para facilitar su interpretación por parte de los docentes.

Por otra parte, un 36 % de los participantes optó por el nivel 4, lo que indica una comprensión sólida, aunque posiblemente con dudas menores o necesidad de refuerzo en algunos momentos, entre ambos niveles, se acumula un 92 % de respuestas claramente positivas.

El 8% restante seleccionó el nivel 3, correspondiente a una posición neutral. Este grupo podría estar compuesto por usuarios que comprendieron los niveles de forma parcial o que necesitan mayor orientación durante su primer contacto con el sistema, Para atender esta posible dificultad, el prototipo incorpora una funcionalidad que permite visualizar

descripciones detalladas de los niveles (Básico, Intermedio, Avanzado) dentro de los resultados de evaluación, tanto en las tablas como en el modal de detalles por Resultado de Aprendizaje, donde se especifica el nivel alcanzado, el puntaje obtenido y una recomendación asociada. Cabe resaltar que no se registraron respuestas en los niveles 1 ni 2, lo que indica que no se presentó confusión grave respecto a este aspecto.

En general, los resultados validan que la metodología de representación de niveles en el prototipo es comprensible para su público objetivo.

8.3.3 Facilidad en la creación de evaluaciones.

La facilidad para configurar evaluaciones personalizadas fue uno de los aspectos evaluados por los usuarios del prototipo, este proceso incluye la definición de criterios, la asignación de indicadores, el diseño de preguntas y la posterior inserción de puntajes por estudiante.

Según la imagen 1 del anexo B, el 48% de los encuestados seleccionó el nivel 5, indicando que el proceso de creación y aplicación de evaluaciones les resultó completamente claro y manejable, con esta percepción se sugiere que el diseño secuencial y la segmentación de pasos implementados en la interfaz fueron efectivos para guiar al usuario durante el flujo de trabajo.

El 40 % restante optó por el nivel 4, lo cual refleja una experiencia funcional con pequeñas oportunidades de mejora, como la inclusión de ayudas contextuales, mensajes explicativos o validaciones más detalladas en la configuración.

Un 12 % de los participantes eligió el nivel 3, lo que indica una experiencia neutra. Aunque no se manifestaron dificultades críticas, este grupo puede representar usuarios que tuvieron una curva de aprendizaje inicial o que necesitaron mayor orientación al crear su primera evaluación.

No se registraron respuestas en los niveles 1 ni 2, lo que indica que todos los usuarios pudieron completar esta funcionalidad sin bloqueos o frustraciones.

En conjunto, los resultados indican que el sistema ofrece una experiencia técnica adecuada para la creación de evaluaciones educativas, aunque se recomienda evaluar formas de mejorar la autoexplicación del flujo, especialmente en los primeros usos.

8.3.4 Utilidad de las gráficas en la interpretación del rendimiento.

Las visualizaciones gráficas fueron evaluadas como un componente clave para la comprensión de los resultados académicos en el prototipo. Esta funcionalidad incluía representaciones como boxplots, gráficas de radar, barras comparativas y mapas de calor, cada una enfocada en diferentes dimensiones del rendimiento por estudiante y por resultado de aprendizaje.

De acuerdo con la imagen 2 del anexo B, un 44 % de los encuestados seleccionó el nivel 5, indicando que las gráficas estadísticas proporcionadas fueron completamente útiles para interpretar el desempeño de los estudiantes. Este grupo consideró que las visualizaciones facilitaban la lectura de los niveles de logro, la comparación entre estudiantes y la detección de patrones relevantes.

Otro 44 % indicó nivel 4, lo que refleja una experiencia positiva con las gráficas estadísticas, aunque con posibles oportunidades de mejora en aspectos como interacción, personalización o presentación en distintos dispositivos.

El 12 % restante seleccionó el nivel 3, representando a usuarios con una percepción neutra. Este segmento podría beneficiarse de mejoras en la explicación inicial de las gráficas o en la forma en que se conectan los datos con la evaluación pedagógica. No se registraron respuestas en los niveles 1 ni 2.

En términos generales, los resultados sugieren que las gráficas están cumpliendo su función informativa, aunque incorporar elementos como leyendas dinámicas, filtros por estudiante o una guía visual podría fortalecer aún más su impacto en el análisis pedagógico.

8.3.5 Satisfacción general con la experiencia de uso.

La percepción global del usuario sobre la experiencia ofrecida por el prototipo fue evaluada mediante esta afirmación, que resume la interacción integral con todas las funcionalidades de la plataforma: navegación, diseño visual, tiempos de respuesta, estructura de módulos y calidad de resultados.

Según la imagen 3 del anexo B, el 56 % de los encuestados expresó un alto nivel de satisfacción, seleccionando el nivel 5 en la escala de Likert, esto indica que más de la mitad de los usuarios consideraron que el sistema cumple satisfactoriamente con los requisitos funcionales y ofrece una experiencia sólida desde el punto de vista técnico y visual.

El 36 % seleccionó el nivel 4, lo que refuerza la evaluación positiva general, este grupo percibió la plataforma como funcional, aunque es posible que hayan identificado detalles menores susceptibles de mejora, como la disposición de botones o tiempos de carga en ciertas operaciones.

El 8 % restante marcó el nivel 3, lo cual representa una valoración neutral. Aunque este grupo no expresó insatisfacción, podría beneficiarse de mejoras orientadas a personalización, ayuda contextual o simplificación de procesos iniciales. No se reportaron respuestas en los niveles 1 ni 2, lo que indica que ningún usuario se sintió insatisfecho con la experiencia general.

En conjunto, los resultados validan que el sistema ofrece una experiencia globalmente satisfactoria, aunque se identifican oportunidades puntuales de mejora que podrían optimizar la interacción y percepción en futuras versiones del prototipo.

8.3.6. Evaluación efectiva de los Resultados de Aprendizaje según el método de competencias.

Además de la percepción del usuario, se evaluó si el sistema implementa correctamente la lógica del método de evaluación por competencias de Pensamiento Computacional (PC). Este método implica la traducción de puntajes asignados a indicadores de logro y criterios en niveles cualitativos (Básico, Intermedio, Avanzado) para cada Resultado de Aprendizaje (RA).

El sistema realiza este proceso automáticamente a través de un algoritmo que sigue la siguiente secuencia:

1. Evaluación de Indicadores de Logro (IL):

Suma los puntajes obtenidos en las actividades asociadas a cada IL y los compara con rangos porcentuales predefinidos. Se asigna nivel Básico si el puntaje es igual o inferior al 40 % de la ponderación del IL, nivel Intermedio si se encuentra entre el 41 % y el 80 %, y nivel Avanzado si supera el 81 % hasta el 100 %.

2. Determinación del nivel de los Criterios de Evaluación (CE):

Agrupar los puntajes de los IL que conforman cada CE y aplica una lógica de umbral. El sistema calcula un **Umbral de Básico** sumando el valor correspondiente al nivel Básico del IL con mayor ponderación, más los valores de nivel Intermedio de los demás IL. Si la suma de los puntajes del CE es menor o igual al umbral, se asigna nivel Básico; si es igual al valor máximo del CE, se asigna nivel Avanzado; en cualquier otro caso, se otorga nivel Intermedio.

3. Clasificación de Resultados de Aprendizaje (RA):

Acumula los valores cuantitativos de los CE asociados a cada RA. Luego, compara esta suma con un **Umbral de Básico** específico para el RA y con el puntaje máximo

(100 puntos). Si la suma es menor o igual al umbral, se asigna nivel Básico; si equivale a 81 o más, se asigna nivel Avanzado; y si está entre ambos, se otorga nivel Intermedio.

Para verificar la efectividad de esta funcionalidad, se analizaron los resultados generados automáticamente por el sistema y se contrastaron con los niveles esperados según la rúbrica predefinida. Los resultados fueron validados por los usuarios mediante la interfaz de visualización detallada y gráfica (como se muestra en las Figuras 11 y 12). No se reportaron inconsistencias en los niveles asignados ni errores de cálculo en las pruebas realizadas.

Figura 11

Resultado manual

Calificación/100	P, 1 /10,00	P, 2 /5,00	P, 3 /15,00	P, 4 /7,50	P, 5 /7,50	P, 6 /7,50	P, 7 /7,50	P, 8 /15,00	P, 9 /25,00	C1-RA1 Cualitativo	C1-1-RA1 Cualitativo	C1-2-RA1 Cualitativo	C2-RA1 Cualitativo	C2-1-RA1 Cualitativo	C2-2-RA1 Cualitativo	C2-3-RA1 Cualitativo	Nivel RA1 Mayor peso a C2-RA1 Propuesta 2
66,61	5	0	0	6,25	5,36	5	5	15	25	B	I	B	I	I	I	A	I
68,83	1,33	5	5	3,75	7,5	3,75	2,5	15	25	I	B	I	I	I	I	A	I
22,74	1,67	0	1,25	1,25	1,07	2,5	2,5	0	12,5	B	B	B	B	B	B	B	B
66,31	1,67	5	5	0	2,14	5	7,5	15	25	I	B	I	I	B	I	A	I
45,89	1,67	5	5	3,75	2,14	2,5	2,5	15	8,33	I	B	I	I	I	B	I	I
69,29	1,67	5	5	3,75	4,29	6,25	7,5	15	20,83	I	B	I	I	I	I	I	I
39,05	1,67	0	5	3,75	3,21	3,75	2,5	15	4,17	B	B	B	I	I	I	I	I
42,92	1,67	0	5	0	0	3,75	5	15	12,5	B	B	B	I	B	I	I	I
72,92	1,67	5	10	0	7,5	6,25	2,5	15	25	I	B	I	I	I	I	A	I
32,08	3,33	0	1,25	1,25	0	3,75	0	10	12,5	B	I	B	I	B	B	I	I

Nota. La figura muestra los resultados de una evaluación usando excel para su cálculo.

Fuente: Elaboración propia (González & Estrada, 2025)

Figura 12

Resultado calculado en la página

Nombre	Código	Interiorización	Puntaje Total	CE1	CE2	IL1-1	IL1-2	IL2-1	IL2-2	IL2-3
Laura Gómez	301	Intermedio	66.61/100 Intermedio	Básico (5/30)	Intermedio (61.61/70)	Intermedio (5/10)	Básico (0/20)	Intermedio (11.61/15)	Intermedio (10/15)	Avanzado (40/40)
Mateo Rodríguez	302	Intermedio	68.83/100 Intermedio	Intermedio (11.33/30)	Intermedio (57.5/70)	Básico (1.33/10)	Intermedio (10/20)	Intermedio (11.25/15)	Intermedio (6.25/15)	Avanzado (40/40)
Isabella Torres	303	Básico	22.74/100 Básico	Básico (2.92/30)	Básico (19.82/70)	Básico (1.67/10)	Básico (1.25/20)	Básico (2.32/15)	Básico (5/15)	Básico (12.5/40)
Juan Pérez	304	Intermedio	66.31/100 Intermedio	Intermedio (11.67/30)	Intermedio (54.64/70)	Básico (1.67/10)	Intermedio (10/20)	Básico (2.14/15)	Intermedio (12.5/15)	Avanzado (40/40)
Sofía Martínez	305	Intermedio	45.89/100 Intermedio	Intermedio (11.67/30)	Intermedio (34.22/70)	Básico (1.67/10)	Intermedio (10/20)	Intermedio (5.89/15)	Básico (5/15)	Intermedio (23.33/40)
Andrés López	306	Intermedio	69.29/100 Intermedio	Intermedio (11.67/30)	Intermedio (57.62/70)	Básico (1.67/10)	Intermedio (10/20)	Intermedio (8.04/15)	Intermedio (13.75/15)	Intermedio (35.83/40)
Daniela Sánchez	307	Intermedio	39.05/100 Intermedio	Básico (6.67/30)	Intermedio (32.38/70)	Básico (1.67/10)	Básico (5/20)	Intermedio (6.96/15)	Intermedio (6.25/15)	Intermedio (19.17/40)
Lucas Ramírez	308	Intermedio	42.92/100 Intermedio	Básico (6.67/30)	Intermedio (36.25/70)	Básico (1.67/10)	Básico (5/20)	Básico (0/15)	Intermedio (8.75/15)	Intermedio (27.5/40)
Valeria Moreno	309	Intermedio	72.92/100 Intermedio	Intermedio (16.67/30)	Intermedio (56.25/70)	Básico (1.67/10)	Intermedio (15/20)	Intermedio (7.5/15)	Intermedio (8.75/15)	Avanzado (40/40)
Samuel Herrera	310	Intermedio	32.08/100 Intermedio	Básico (4.58/30)	Intermedio (27.5/70)	Intermedio (3.33/10)	Básico (1.25/20)	Básico (1.25/15)	Básico (3.75/15)	Intermedio (22.5/40)

Nota. La figura muestra los resultados de una evaluación usando la aplicación para su cálculo. Fuente: Elaboración propia (González & Estrada, 2025)

El prototipo web para la evaluación de competencias ha sido valorado positivamente por los usuarios, especialmente por docentes que participaron en su validación. Las pruebas realizadas demostraron que el sistema facilita la gestión de evaluaciones por criterios e indicadores, y que los niveles de logro generados permiten una interpretación clara del rendimiento de los estudiantes.

Los usuarios destacaron la utilidad de las visualizaciones gráficas y la posibilidad de exportar los resultados en diferentes formatos, aspectos que refuerzan el valor práctico de la herramienta en entornos educativos. La estructura jerárquica de resultados de aprendizaje, criterios e indicadores fue comprendida y aplicada con éxito en los escenarios de prueba.

Aunque se identificaron oportunidades de mejora en términos de guía visual y orden de navegación durante las primeras interacciones, los ajustes realizados permitieron obtener una experiencia más fluida y coherente en versiones posteriores.

Consideramos que este proyecto ha sido significativo, ya que aborda una necesidad frecuente en el ámbito académico: disponer de una herramienta funcional para aplicar, sistematizar y analizar evaluaciones cualitativas, este desarrollo nos permitió aplicar conocimientos técnicos en un contexto real, así como mejorar habilidades en diseño de interfaces, modelado de bases de datos y retroalimentación basada en usuarios y esto nos motiva para continuar perfeccionando la plataforma y explorar nuevas funcionalidades que aporten al proceso formativo de docentes y estudiantes.

Capítulo 9

Conclusiones del proyecto

El desarrollo del *Prototipo de aplicación web como herramienta de apoyo para la evaluación de Competencias en Pensamiento Computacional* permitió abordar una problemática clave en el contexto educativo: la necesidad de contar con instrumentos sistemáticos, comprensibles y escalables para la evaluación de estas competencias. A través de la implementación de un método estructurado basado en rúbricas de criterios e indicadores, el sistema ofrece una forma clara y trazable de calificar el desempeño de los estudiantes, permitiendo asignar niveles de logro cualitativos a partir de puntajes ingresados en la plataforma.

- El prototipo cumplió con los objetivos planteados inicialmente. A partir del método de evaluación ya existente, se realizó un proceso de análisis, comprensión y estructuración técnica que permitió organizar los resultados de aprendizaje, criterios, indicadores y preguntas en función de la lógica de la aplicación. Posteriormente, se desarrolló un sistema funcional que implementa dicha estructuración técnica a través de formularios interactivos, validación de puntajes, cálculo automatizado de niveles, generación de trazabilidad y visualización gráfica. Finalmente, se aplicaron pruebas de usabilidad que evidenciaron una recepción positiva por parte de los usuarios, quienes valoraron la claridad de la interfaz, la utilidad de los resultados generados y la facilidad para aplicar evaluaciones completas desde la plataforma.
- Los resultados obtenidos durante la validación confirman que la aplicación responde adecuadamente a las necesidades de los docentes que evalúan competencias en pensamiento computacional, ofreciendo una experiencia técnica estable y alineada

con los procesos pedagógicos actuales. Las gráficas incorporadas, junto con las funciones de exportación, fortalecen el análisis individual y grupal del aprendizaje.

- No obstante, se identificaron aspectos que pueden ser optimizados en futuras versiones, como el fortalecimiento de las ayudas visuales para usuarios principiantes, la personalización de reportes exportables y la incorporación de filtros avanzados en la sección de resultados. Estas oportunidades de mejora no afectan la funcionalidad central del sistema, pero representan posibles líneas de evolución del prototipo.
- El desarrollo de este proyecto representó una experiencia integral que permitió aplicar conocimientos profesionales en programación web, diseño de bases de datos, modelado de procesos educativos y pruebas con usuarios. Este proceso no sólo consolidó competencias profesionales, sino que también nos preparó para abordar desafíos complejos en futuros proyectos tecnológicos, reforzando nuestra capacidad de adaptarnos a contextos dinámicos y exigentes.

9.1 Trabajos futuros

De acuerdo con el alcance definido en este trabajo de grado, se lograron cumplir los objetivos propuestos. No obstante, existen funcionalidades y componentes que podrían ser ampliados o mejorados en desarrollos posteriores. A continuación, se presentan algunas recomendaciones y posibles líneas de trabajo futuro:

- **Desarrollo de una versión móvil:** Considerar la creación de una aplicación móvil complementaria al prototipo web, que permita a los docentes aplicar evaluaciones o consultar resultados desde dispositivos móviles o tabletas, aumentando así la accesibilidad del sistema en entornos con movilidad o recursos limitados.
- **Incorporación de un sistema de tutoriales interactivos:** Diseñar ayudas visuales o recorridos guiados que orienten a los nuevos usuarios en el uso del sistema,

especialmente en los procesos de creación de evaluaciones, asignación de puntajes y análisis de resultados. Esto facilitaría el aprendizaje autónomo y reduciría la necesidad de soporte técnico externo.

- **Personalización avanzada de evaluaciones:** Ampliar las opciones de configuración del módulo de evaluación, permitiendo al usuario modificar visualmente escalas, tipos de preguntas, colores de niveles y estructuras de rúbricas, adaptando el sistema a diferentes enfoques pedagógicos.
- **Análisis comparativo de resultados:** Desarrollar un módulo que permita comparar el rendimiento de estudiantes entre diferentes cursos, evaluaciones o momentos del tiempo, facilitando así el seguimiento del progreso académico y la toma de decisiones pedagógicas basadas en datos.
- **Explorar otras competencias de pensamiento computacional a evaluar:** Ampliar la cobertura del sistema incorporando nuevas dimensiones del pensamiento computacional más allá de la actualmente implementada, esto incluye competencias como descomposición, pensamiento algorítmico, depuración y análisis de datos. La incorporación de estas categorías permitiría diversificar los instrumentos de evaluación y ofrecer una visión más integral del desarrollo cognitivo de los estudiantes en entornos educativos orientados a la programación, la resolución de problemas y la lógica computacional.

Referencias

- Arévalo-Sánchez, J., Suárez-Rodríguez, C., & Suárez-Rodríguez, J. (2023). *Diseño y evaluación de un programa de tutoría de pensamiento computacional para estudiantes de primaria en Colombia*.
- Barker, L., & Garvin-Doxas, K. (2016). The computer science concept of loops in K–12 mathematics. *Computer Science Education*, 26(3), 215–226.
- Bermúdez Mazuera, Y. M. (s.f). *Método clasificador de competencias en pensamiento computacional* [Tesis de maestría, Universidad del Valle].
- Black, P., & Wiliam, D. (1998). Inside the black box: Raising standards through classroom assessment. *Phi Delta Kappan*, 80(2), 139–148.
- Concepto.de. (s. f.). *Programación – concepto*.
<https://concepto.de/programacion/>
- Crenshaw, J. (2022). How to hash passwords in Node.js. Honeybadger.
<https://www.honeybadger.io/blog/node-password-hashing/>
- Cuny, J. (2010). *The ACM 2010 task force curricular guidelines for K–12 computer science*. ACM.
<https://www.acm.org/education/CS2013-final-report.pdf>
- EJS. (2024). *Embedded JavaScript templates*.
<https://ejs.co/>
- Express.js. (2024). cookie-parser middleware. Express.js Documentation.
<https://expressjs.com/en/resources/middleware/cookie-parser.html>
- Express.js. (2024). *express-session middleware*. *Express.js Documentation*.
<https://expressjs.com/en/resources/middleware/session.html>

- Fedesoft. (2023, diciembre 6). *Bebras: Una iniciativa internacional para promover la informática y el pensamiento computacional*.
<https://fedesoft.org/bebras-colombia-2023/>
- Ferrovial. (s. f.). *¿Qué son los algoritmos?* Ferrovial STEM.
<https://www.ferrovial.com/es/stem/algoritmos/>
- García, P., & Sánchez, P. (2024). El pensamiento crítico en el ámbito educativo. *IEEE Transactions on Education*, 67(6), 1–10.
- García, P., & Sánchez, P. (2024). La abstracción en el pensamiento crítico. *IEEE Transactions on Education*, 67(2), 1–10.
- Godoy, S. (2023). Modelo de investigación en ingeniería del software: Una propuesta de investigación tecnológica. *Academia.edu*.
https://www.academia.edu/2927485/Modelo_de_Investigaci%C3%B3n_en_Ingenier%C3%A1Da_del_Software_Una_propuesta_de_investigaci%C3%B3n_tecnol%C3%B3gica
- González-Calero, J. A. (2023). Evaluación del pensamiento computacional: Una revisión de la literatura. *Revista de Educación*, (391), 23–44.
- Grover, S., & Pea, R. (2013). Computational thinking in K–12: A review of the state of the field. *Educational Researcher*, 42(1), 38–43.
- Kinsta. (2024). *¿Qué es MySQL?*
<https://kinsta.com/es/base-de-conocimiento/que-es-mysql/>
- Kinsta. (2024). *¿Qué es Node.js?*
<https://kinsta.com/es/base-de-conocimiento/que-es-node-js>

- Martínez-Sánchez, M. Á., Muñoz-Merino, P. J., & Sancho-Vinuesa, M. T. (2017). Evaluación del pensamiento computacional en educación primaria. *Revista Interuniversitaria de Investigación en Tecnología Educativa*, (3), 1–13.
- Nkubito, N. (2023). How to use Multer in Node.js and Express for file uploads. LogRocket Blog. <https://blog.logrocket.com/multer-nodejs-express-upload-file/>
- López-Gómez, D. C., Sierra-Rodríguez, J. L., & Suárez-Castañón, M. del C. (2021). Método de evaluación del pensamiento computacional y alfabetización en código. *Praxis & Saber*, 12(30), 1–25.
- OpenWebinars. (2024). *¿Qué es JSON Web Token?* <https://openwebinars.net/blog/que-es-json-web-token-y-como-funciona/>
- Organización para la Cooperación y el Desarrollo Económicos (OCDE). (2018). *El futuro de la educación y las competencias: Educación para el mundo del trabajo cambiante*.
- Ortega-Ruipérez, B., Moreno-León, M. J., & Sancho-Vinuesa, F. (2021). Evaluar el pensamiento computacional mediante resolución de problemas: Validación de un instrumento de evaluación. *Revista Iberoamericana de Evaluación Educativa*, 14(1), 157–174.
- Pressman, R. S. (2019). *Software engineering: A practitioner's approach* (8.^a ed.). McGraw-Hill Education.
- Python Software Foundation. (2024). *csv — Lectura y escritura de archivos CSV. Documentación oficial de Python 3.13*. <https://docs.python.org/es/3.13/library/csv.html>.
- Pérez, J., & Rodríguez, M. (2024). La competencia de abstracción en el pensamiento computacional. *IEEE Transactions on Education*, 67(7), 1–10.
- Pérez, J., & Rodríguez, M. (2024). La descomposición en el pensamiento computacional. *IEEE Transactions on Education*, 67(3), 1–10.

- Teinmaurer, A., Pirker, J., & Gütl, C. (2019). sCool – Game-based learning in computer science class: A case study in secondary education. *International Journal of Engineering Pedagogy*, 9(2), 9942.
<https://doi.org/10.3991/ijep.v9i2.9942>.
- Tobón, S. (2005). *Formación basada en competencias: pensamiento complejo, diseño curricular y evaluación*. ECOE.
- Vlahović, I., & Biškupić, I. O. (2023). Fostering critical and computational thinking in the field of primary and secondary education in non-STEM subjects by using data sets and applications. En 2023 46th MIPRO ICT and Electronics Convention (MIPRO) (pp. 672–677).
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33–35.
- npm. (2024). *dotenv*.
<https://www.npmjs.com/package/dotenv>.

Anexos

Se incluye el código fuente de la aplicación junto con el link de uso de la misma, para facilitar su revisión, también se incluye los links a los mockups realizados previamente. la bitácora de reuniones, y la estructura del método de evaluación.

Anexo A:

https://docs.google.com/document/d/1QtbThT7jy9HtbUcZJmN-02rWy_G7x0PpfUVjGudWibI/edit?tab=t.0#heading=h.ck9wt01takeq

Anexo B:

https://docs.google.com/document/d/1oOksBvZhwa50T8JrT4KJmOa_RrH9Gu9zI_ufArcG28/edit?usp=sharing

Código de la aplicación: <https://github.com/Sergioestrada1601/proyecto-de-grado>

link de la aplicación: <http://45.5.167.44:3000/>

mockups realizados previamente:

<https://docs.google.com/document/d/14EnNDkTcd3V35Uk0rzXWEOw0qw1CCmcid2ktld6nVp4/edit?tab=t.0>

formulario de encuesta: <https://forms.gle/66gHMkuV9i52FA6c7>

hoja de cálculo encuesta:

https://docs.google.com/spreadsheets/d/1S1TYseDM_Y9ObSfWm7T2XxToe32-a_EAkr7tv67yoac/edit?usp=sharing

Product backlog:

<https://docs.google.com/spreadsheets/d/1LoFPJtopDwfsCfk3Lbo5ysIdEhBrHE9389HLGgNNf2U/edit?usp=sharing>

CSV de prueba:

https://drive.google.com/drive/folders/1FO12hfjHPlatKbVOYILaOTJ7mjdJnsph?usp=drive_link