



UNIVERSIDAD NACIONAL DE CHIMBORAZO

**FACULTAD CIENCIAS DE LA EDUCACIÓN HUMANAS Y
TECNOLOGÍAS**

**CARRERA DE PEDAGOGÍA DE LAS CIENCIAS
EXPERIMENTALES INFORMÁTICA**

**DESARROLLO DE UN JUEGO EDUCATIVO SOBRE FUNDAMENTOS DE
PROGRAMACIÓN PARA ESTUDIANTES DE LA CARRERA DE PEDAGOGÍA
DE LAS CIENCIAS EXPERIMENTALES: INFORMÁTICA.**

**TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE
LICENCIADA EN PEDAGOGÍA DE LA INFORMÁTICA**

AUTORA:

Sefla Urquizo Maribel Alexandra

TUTOR:

Mgs. Jorge Eduardo Fernández Acevedo

Riobamba, Ecuador. 2025

DECLARATORIA DE AUTORÍA

Yo, Maribel Alexandra Sefla Urquizo, con cédula de ciudadanía 060540352-6, autor (a) (s) del trabajo de investigación titulado: Desarrollo de un juego educativo sobre fundamentos de programación para estudiantes de la Carrera de Pedagogía de las Ciencias Experimentales: Informática, certifico que la producción, ideas, opiniones, criterios, contenidos y conclusiones expuestas son de mí exclusiva responsabilidad.

Asimismo, cedo a la Universidad Nacional de Chimborazo, en forma no exclusiva, los derechos para su uso, comunicación pública, distribución, divulgación y/o reproducción total o parcial, por medio físico o digital; en esta cesión se entiende que el cesionario no podrá obtener beneficios económicos. La posible reclamación de terceros respecto de los derechos de autor (a) de la obra referida, será de mi entera responsabilidad; librando a la Universidad Nacional de Chimborazo de posibles obligaciones.

En Riobamba, 23 de enero del 2025.



Maribel Alexandra Sefla Urquizo

C.I: 0605403526

DICTAMEN FAVORABLE DEL PROFESOR TUTOR

Quien suscribe, Mgs. Jorge Eduardo Fernández Acevedo catedrático adscrito a la Facultad de Ciencias de la Educación Humanas y Tecnologías por medio del presente documento certifico haber asesorado y revisado el desarrollo del trabajo de investigación DESARROLLO DE UN JUEGO EDUCATIVO SOBRE FUNDAMENTOS DE PROGRAMACIÓN PARA ESTUDIANTES DE LA CARRERA DE PEDAGOGÍA DE LAS CIENCIAS EXPERIMENTALES: INFORMÁTICA, bajo la autoría de Maribel Alexandra Sefla Urquizo; por lo que se autoriza ejecutar los trámites legales para su sustentación.

Es todo cuanto informar en honor a la verdad; en Riobamba, a los 01 días del mes de Julio de 2024



Mgs. Jorge Eduardo Fernández Acevedo

C.I: 0602887630



CERTIFICADO DE LOS MIEMBROS DEL TRIBUNAL

Quienes suscribimos, catedráticos designados Miembros del Tribunal de Grado para la evaluación del trabajo de investigación **DESARROLLO DE UN JUEGO EDUCATIVO SOBRE FUNDAMENTOS DE PROGRAMACIÓN PARA ESTUDIANTES DE LA CARRERA DE PEDAGOGÍA DE LAS CIENCIAS EXPERIMENTALES: INFORMÁTICA**, presentado por **Maribel Alexandra Sella Urquiza**, con cédula de identidad número **060540352-6**, bajo la tutoría de **Mgs. Jorge Eduardo Fernández Acevedo**; certificamos que recomendamos la **APROBACIÓN** de este con fines de titulación. Previamente se ha evaluado el trabajo de investigación y escuchada la sustentación por parte de su autor; no teniendo más nada que observar.

De conformidad a la normativa aplicable firmamos, en **Riobamba** 23 de enero del 2025.

Mgs. Christiam Xavier Núñez Zavala
PRESIDENTE DEL TRIBUNAL DE GRADO

Mgs. José Luis Erazo Parra
MIEMBRO DEL TRIBUNAL DE GRADO

Mgs. Hernán Ramiro Pailiacho Yucta
MIEMBRO DEL TRIBUNAL DE GRADO



Dirección
Académica
VICERRECTORADO ACADÉMICO



UNACH-RGF-01-04-08-15
VERSIÓN 01-06-09-2021

CERTIFICACIÓN

Que, **SEFLA URQUIZO MARIBEL ALEXANDRA** con CC: **060540352-6**, estudiante de la Carrera **PEDAGOGÍA DE LAS CIENCIAS EXPERIMENTALES INFORMÁTICA**, Facultad de **CIENCIAS DE LA EDUCACIÓN, HUMANAS Y TECNOLOGÍAS**; ha trabajado bajo mi tutoría el trabajo de investigación titulado **"Desarrollo de un juego educativo sobre fundamentos de programación para estudiantes de la Carrera de Pedagogía de las Ciencias Experimentales: Informática"**, cumple con el **5%**, de acuerdo al reporte del sistema anti plagio **Compilatio Magister+**, porcentaje aceptado de acuerdo a la reglamentación institucional, por consiguiente autorizo continuar con el proceso.

Riobamba, 10 de enero de 2025



Mgs. Jorge Eduardo Fernández Acevedo
TUTOR

DEDICATORIA

El presente trabajo de investigación va dedicado a mis padres Carlos Sefla y Elsa Urquizo, quienes con su apoyo incondicional trabajo arduo, hoy estoy cumpliendo una meta más en vida académica y profesional.

Dedico esta tesis a mi esposo Luis Ovando quien ha sido un pilar fundamental en mi vida académica gracias a su apoyo diario, motivación y esfuerzo se llegó a la meta, y mi hija Noemi Ovando que me han brindado su amor, comprensión, apoyo en los años que no pudimos estar juntas por mis estudios, y vea en mi un ejemplo y logre cumplir sus sueños y objetivos.

Maribel Sefla

AGRADECIMIENTO

Primeramente, doy gracias, Dios darme esta oportunidad y tener una gran experiencia en la Universidad Nacional de Chimborazo, gracias a la Dra. Cristy Jiménez que ha sido una gran directora de carrera sobre todo que nos ha impulsado a seguir cada semestre dándonos ánimos para llegar a ser unos grandes profesionales en el área de la Informática.

Agradezco a mi tutor de tesis el Mgs. Jorge Fernández Acevedo por brindarme aquellos conocimientos y sabidurías a lo largo del proceso, gracias por la paciencia, dedicación y su capacidad de poder despertar en mí la curiosidad y la motivación para la creación de un juego educativo.

Agradezco a mi tribunal los docentes Mgs. José Luis Erazo, Mgs. Hernán Ramiro Pailiacho Yucta, Mgs. Christiam Xavier Núñez Zavala, por su tiempo dedicación y su gran aporte respectivo en la corrección de mi tesis, gracias por el ejemplo que nos dan a través de su experiencia.

Agradezco a mis hermanos por aquellos consejos y por su apoyo constante en mis estudios, ya que sus consejos han sido motivo de inspiración y superación.

Agradezco a las personas cercanas que estuvieron junto a mí en este proceso Sra. Jimena Chimbo gracias por sus consejos que me han servido y por ser una gran persona tener paciencia en el trabajo y ayudarme en mis estudios.

ÍNDICE GENERAL

DECLARATORIA DE AUTORÍA

DICTAMEN FAVORABLE DEL PROFESOR TUTOR

CERTIFICADO DE LOS MIEMBROS DEL TRIBUNAL

CERTIFICADO ANTIPLAGIO

DEDICATORIA

AGRADECIMIENTO

ÍNDICE GENERAL

ÍNDICE DE TABLAS

ÍNDICE DE GRÁFICOS

TABLA DE ILUSTRACIONES

RESUMEN

ABSTRACT

CAPÍTULO I

1.	INTRODUCCION.....	16
1.1.	Antecedentes.....	17
1.2.	Planteamiento del Problema.....	19
1.3.	Justificación.....	19
1.4.	Objetivos.....	21
1.4.1.	General.....	21
1.4.2.	Específicos.....	21
	CAPÍTULO II.....	22
2.	MARCO TEÓRICO.....	22
2.1.	Informática educativa.....	22
2.1.1.	Historia y evolución.....	22
2.1.2.	Tendencias actuales en la enseñanza de la Informática.....	23
2.2.	Fundamentos de programación.....	23
2.2.1.	Resumen de conceptos claves.....	23
2.2.2.	Retos y desafíos comunes en la enseñanza de la Programación.....	24
2.3.	Juegos educativos.....	25
2.3.1.	Concepto.....	25
2.3.2.	Características.....	25
2.3.3.	Beneficios.....	26

2.4.	Aprendizaje de la Programación.....	26
2.4.1.	¿Qué es la programación?.....	26
2.4.2.	Tipos de programación.....	27
2.5.	EdiLim	28
2.5.1.	Características.....	28
2.5.2.	Usos de EdiLim en la educación.....	28
2.6.	Gamificación.....	29
2.6.1.	Características principales de la gamificación en la educación.....	29
2.6.2.	Importancia de la gamificación dentro del aula de clases.....	29
2.7.	Metodología Dicrevoa	30
2.7.1.	Fases	30
CAPÍTULO III		32
3.	METODOLOGIA.....	32
3.1.	Tipo de investigación.....	32
3.2.	Enfoque de la investigación.....	32
3.3.	Alcance de la investigación	33
3.4.	Diseño de la investigación	33
3.5.	Técnicas de recolección de Datos	33
3.6.	Población de estudio y tamaño de muestra.....	34
3.7.	Análisis de recolección de datos	34
CAPÍTULO IV		35
4.	RESULTADOS	35
4.1.	Datos demográficos	35
4.2.	Contenidos básicos de Programación	38
4.3.	Análisis del dominio de comprensión y aplicación de conceptos básicos de Programación	49
5.	PROPUESTA	51
5.1.	Introducción	51
5.2.	Objetivo General de la Propuesta	51
5.3.	Objetivos Específicos	52
5.4.	Desarrollo de la Propuesta	52
CAPITULO IV		59
6.	CONCLUSIONES DE RECOMENDACIONES.....	59
6.1.	Conclusiones.....	59

6.2. Recomendaciones	59
7. BIBLIOGRAFÍA	60
ANEXOS	64

ÍNDICE DE TABLAS

Tabla 1: Características EdiLim	28
Tabla 2. Género de los estudiantes encuestados.....	35
Tabla 3. Distribución de estudiantes encuestados por semestre.....	36
Tabla 4. Experiencia y conocimientos previos en el campo de la programación de computadores.....	37
Tabla 5. Nivel de comprensión sobre los tipos de datos y su uso en la programación.	38
Tabla 6. Nivel de seguridad en el uso de operadores y expresiones	39
Tabla 7. Habilidad para escribir programas que sigan una secuencia lógica	40
Tabla 8. Eficiencia al planificar y organizar un código.....	41
Tabla 9. Nivel de confianza al utilizar estructuras condicionales	42
Tabla 10. Habilidad para diseñar y aplicar estructuras de control condicionales.....	43
Tabla 11. Capacidad para implementar bucles en programas para repetir acciones de manera controlada	44
Tabla 12. Habilidad para diseñar bucles para optimizar el rendimiento y reducir la complejidad de soluciones.....	45
Tabla 13. Comprensión y aplicación de funciones.....	46
Tabla 14. Habilidad para crear funciones que mejoren la legibilidad y reutilización del código	47
Tabla 15. Capacidad para depurar código	48
Tabla 16. Dominio de conceptos de Programación.....	49
Tabla 17 Matriz de necesidades.....	52
Tabla 18 Plantilla para el Diseño del Objeto de Aprendizaje.....	54
Tabla 19 Herramientas utilizadas	57

ÍNDICE DE GRÁFICOS

Gráfico 1. Género de los estudiantes encuestados.....	36
Gráfico 2. Distribución porcentual de los estudiantes encuestados.....	37
Gráfico 3. Experiencia y conocimientos previos en el campo de la programación de computadores.....	38
Gráfico 4. Nivel de comprensión sobre los tipos de datos y su uso en la programación. ...	39
Gráfico 5. Nivel de seguridad en el uso de operadores y expresiones	40
Gráfico 6. Habilidad para escribir programas que sigan una secuencia lógica	41
Gráfico 7. Eficiencia al planificar y organizar un código.....	42
Gráfico 8. Nivel de confianza al utilizar estructuras condicionales	43
Gráfico 9. Habilidad para diseñar y aplicar estructuras de control condicionales	44
Gráfico 10. Capacidad para implementar bucles en programas para repetir acciones de manera controlada.....	45
Gráfico 11. Habilidad para diseñar bucles para optimizar el rendimiento y reducir la complejidad de soluciones.....	46
Gráfico 12. Comprensión y aplicación de funciones.....	47
Gráfico 13. Habilidad para crear funciones que mejoren la legibilidad y reutilización del código	48
Gráfico 14. Capacidad para depurar código	49
Gráfico 15. Dominio de conceptos de Programación.....	50

TABLA DE ILUSTRACIONES

Ilustración 1: Evolución Informática Educativa.....	22
-----------------------------------------------------	----

RESUMEN

La programación de computadoras se ha convertido en una habilidad esencial en la era digital actual, impulsando la innovación y el desarrollo tecnológico en prácticamente todas las áreas del conocimiento. Sin embargo, a pesar de su importancia, aprender a programar presenta grandes desafíos para los estudiantes. Muchos principiantes enfrentan dificultades relacionadas con el pensamiento abstracto, la resolución lógica de problemas o la compleja sintaxis de los lenguajes de programación. Estas barreras suelen generar frustración, desmotivación, dificultando el desarrollo de habilidades fundamentales para el pensamiento computacional. En este contexto, el presente trabajo de investigación tuvo como objetivo el desarrollo de un juego educativo como una estrategia para apoyar el aprendizaje y la nivelación de conocimientos de los fundamentos de programación en los estudiantes de nuevo ingreso a la Carrera de Pedagogía de las Ciencias Experimentales - Informática. Para el desarrollo de la aplicación educativa se empleó la metodología DICREVOA, la cual comprende cinco fases secuenciales: análisis, diseño, implementación, evaluación y publicación. Durante la fase de Análisis, se aplicó una encuesta a los estudiantes de primero, segundo y tercer semestre matriculados en el período académico 2023-2S, con el propósito de identificar sus necesidades de aprendizaje y las principales brechas de conocimiento. Finalmente, la información obtenida permitió identificar los temas más idóneos, que fueron incorporados mediante actividades interactivas, en el juego educativo que fue creado empleando la herramienta EdiLim. La revisión bibliográfica evidencia que los juegos educativos son herramientas efectivas para reforzar la adquisición de conocimientos en diversas disciplinas, especialmente en el ámbito de la programación, fomentando un mejor desempeño académico y reduciendo las dificultades asociadas a este campo.

Palabras claves: Programación de Computadores, Pensamiento Computacional, Metodología DICREVOA, Juego Educativo, Tecnologías de Aprendizaje y Conocimiento.

ABSTRACT

Computer programming has become an essential skill in the current digital era, driving innovation and technological development across virtually all areas of knowledge. However, despite its importance, learning to program presents significant challenges for students. Many beginners face difficulties related to abstract thinking, logical problem-solving, or the complex syntax of programming languages. These barriers often lead to frustration and demotivation, hindering the development of fundamental skills required for computational thinking. In this context, the objective of this research work was to develop an educational game as a strategy to support the learning and leveling of knowledge of programming fundamentals in students new to the Experimental Sciences Pedagogy Career - Computer Science. The development of the educational application followed the DICREVOA methodology, which consists of five sequential phases: analysis, design, implementation, evaluation, and publication. During the analysis phase, a survey was conducted with first, second, and third-semester students enrolled in the 2023-2S academic term to identify their learning needs and the primary knowledge gaps. Ultimately, the information obtained made it possible to determine the most relevant topics, which were incorporated into the educational game through interactive activities. The game was developed using the EdiLim tool. The bibliographic review highlights that educational games are effective tools for strengthening knowledge acquisition in various disciplines, particularly in programming, fostering improved academic performance, and reducing difficulties associated with this field.

Keywords: Computer Programming, Computational Thinking, DICREVOA Methodology, Educational Game, Learning and Knowledge Technologies.



Reviewed by:
Mg. Dario Javier Cutiopala Leon
ENGLISH PROFESSOR
c.c. 0604581066

CAPÍTULO I

1. INTRODUCCION

En la era digital actual, los lenguajes de programación juegan un papel fundamental en el desarrollo de software, sistemas informáticos y aplicaciones que impulsan gran parte de la infraestructura tecnológica de nuestra sociedad. Desde los sistemas operativos y las aplicaciones móviles hasta la inteligencia artificial y el análisis de grandes volúmenes de datos, los lenguajes de programación son herramientas esenciales que permiten a los desarrolladores traducir ideas abstractas en soluciones funcionales y eficientes (Chakray, 2018).

La presente investigación ofrece un análisis de los diferentes temas relacionados con el juego aplicado en el aprendizaje, permitiendo hacerse una idea del papel que pueden tener cuando son más las instituciones que hacen uso de ellos como una forma de enseñanza (Montero, 2017). De manera especial se plantea el desarrollo de un juego educativo para apoyar el dominio de los fundamentos de programación en los estudiantes del primer semestre de la Carrera de Pedagogía de las Ciencias Experimentales Informática.

Este trabajo pretende aportar con una herramienta lúdica para apoyar el desarrollo de competencias cognitivas y habilidades que le permitan al estudiante del primer semestre enfrentar nuevos escenarios y paradigmas de programación en el resto de la Carrera. Además, de la indagación previa no se ha encontrado trabajos de investigación que traten el tema de los juegos como recurso de aprendizaje en el área de programación.

La sociedad ha evolucionado y esto también incluye a la educación, las metodologías tradicionales (conductistas) ya no generan el deseo de aprender sino más bien aburrimiento y desmotivación, por lo que son necesarias nuevas y mejores estrategias para llevar a cabo el proceso de enseñanza-aprendizaje, uno de esos métodos es por medio de la aplicación de juegos educativos en el aula (Montero, 2017).

En informática, se conoce como lenguaje de programación a un programa destinado a la construcción de otros programas informáticos. Su nombre se debe a que comprende un lenguaje formal que está diseñado para organizar algoritmos y procesos lógicos que serán

luego llevados a cabo por un ordenador o sistema informático, permitiendo controlar así su comportamiento físico, lógico y su comunicación con el usuario humano (Chakray, 2018).

1.1. Antecedentes

Para el presente trabajo de investigación se realizó una revisión bibliográfica, realizada en repositorios digitales académicos como: Revistas de la Universidad Nacional de Córdoba, Repositorio Institucional UNAB, Revista Electrónica de Investigación Educativa, en relación con el objeto de estudio: aprendizaje de los fundamentos de programación mediante la aplicación de juegos.

A continuación, se detalla las publicaciones relacionadas con el objeto de estudio:

La investigación realizada por Durán (2023), titulada “Desarrollo de un prototipo de juego serio bajo el paradigma de aprendizaje basado en juegos para la enseñanza de los fundamentos de programación”, menciona que los videojuegos concebidos como herramientas didácticas cuentan con una ventaja muy particular a la hora de ser comparados con metodologías tradicionales como los dictados y las dinámicas de clase; y es la incorporación de distintos medios (visuales, auditivos, kinestésicos, lógicos, etc.) a la naturaleza de su definición; permitiendo así un nivel de inclusión en la educación que permite a los estudiantes a través de una sola herramienta sentirse incluidos, motivados y enganchados con los procesos de aprendizaje, mientras se complementan las metodologías de aprendizaje en los salones de clase, sin discriminar las edades o el nivel educativo.

Esta investigación fue realizada en Colombia en la Universidad Autónoma de Bucaramanga en el año 2023; con un enfoque de carácter mixto, se realizó una encuesta a los docentes y estudiantes, se obtiene como resultado que un videojuego orientado hacia la enseñanza del contenido temático puede afectar positivamente la motivación del estudiante para aprender en la asignatura “Fundamentos de Programación”.

Por otra parte Astudillo et al. (2016), realizaron una investigación denominada “Enfoque basado en gamificación para el aprendizaje de un lenguaje de programación”, que tuvo como objetivo diseñar actividades didácticas, específicamente para el aprendizaje de algunos conceptos de la programación de computadoras, como son: algoritmos, variables y estructuras de control. En el contexto de la investigación sobre juegos y gamificación que llevó adelante el grupo de investigación se logró localizar y evaluar un conjunto de juegos

serios y con ellos definir una secuencia de aprendizaje basada en la estrategia de gamificación para el diseño de un taller de introducción a la programación.

Tras la aplicación del “Enfoque basado en gamificación para el aprendizaje de un lenguaje de programación”, se pudo observar un mejoramiento en la participación y en la motivación de los participantes resolviendo las actividades propuestas. También, se pudo corroborar buenos resultados en la evaluación conceptual al final del Taller de Introducción a la Programación (TIP), la mayoría de los participantes no contaban con experiencia en programación. Asimismo, se apreció un impacto positivo en el rendimiento de los estudiantes que completaron el TIP. Cabe recalcar que durante el 2015 se observaron buenos resultados en la evaluación final. La misma, fue realizada por 25 de los 57 participantes y el 70% tuvieron un promedio por encima de 7 en la calificación final. Esto muestra un dominio aceptable de los conceptos. Debe tomarse en cuenta que: de los 57 inscriptos, 44 contestaron la encuesta inicial, y de ellos el 80% no habían utilizado nunca un lenguaje de programación.

Del mismo modo, Zatarain, (2018), en su trabajo de investigación titulado “Reconocimiento afectivo y gamificación aplicados al aprendizaje de Lógica algorítmica y programación”, cuyos objetivos fueron: desarrollar un sitio web que cumpla con el rol de tutor para la enseñanza de la Lógica algorítmica y programación, reconocer las emociones que los estudiantes presentan en tiempo real, integrar técnicas de gamificación para motivar a los estudiantes y validar si el sistema propuesto contribuye a mejorar el aprendizaje en los estudiantes. Para evaluar el logro de los objetivos propuestos en esta investigación, el sistema fue aplicado con estudiantes del Instituto Tecnológico de Culiacán, Sinaloa (México). En la evaluación participaron estudiantes de la asignatura: Algoritmos y Lenguajes de Programación, quienes cursaban en ese momento la carrera de Ingeniería Industrial.

Como resultado de este trabajo investigativo se presentó la aplicación EasyLogic, un ambiente afectivo para el aprendizaje de Lógica algorítmica y programación. La herramienta apoyó a los estudiantes en el proceso de aprendizaje utilizando técnicas de gamificación – además de monitorear el estado emocional del estudiante, con el fin de intervenir en caso de que el estudiante lo precise.

Los resultados demostraron que el aprendizaje del estudiante es estadísticamente mejor si se toma en cuenta el estado afectivo del estudiante y si éste es motivado por medio de la gamificación.

1.2. Planteamiento del Problema

El enseñar programación desde la escuela, pretende generar nuevas formas de razonamiento, en la búsqueda de soluciones a los problemas. El conocimiento adquirido, en esta área, permite desarrollar múltiples habilidades y competencias, las cuales se pretende perduren para toda la vida, puesto que a los estudiantes les permitirán aplicarlos en diferentes áreas del conocimiento, a la vez que potencian su creatividad y pensamiento lógico (Pinchao, 2021).

En la Universidad Nacional de Chimborazo, en la carrera de Pedagogía de las Ciencias Experimentales Informática, no existe una materia en nivelación que ayude a los estudiantes que ingresan a primer semestre a nivelar los conocimientos relacionados con la programación. La falta de bases en este campo ocasiona dificultad en el abordaje de contenidos más complejos estipulados en otras asignaturas como Lenguajes de Programación I y II, o Desarrollo de Aplicaciones Educativas I y II. Esta situación ha provocado que contados estudiantes opten por el cambio de Carrera.

Por otro lado, los estudiantes que ingresan al primer semestre de la Carrera de Pedagogía de las Ciencias Experimentales Informática, presentan en muchos casos vacíos teóricos o ausencia de destrezas que les permitan abordar cómodamente el estudio y aplicación de contenidos inmersos con la programación de computadores.

Ante esta situación se consideró necesario la creación de un juego educativo sobre fundamentos de programación para estudiantes de la Carrera de Pedagogía de las Ciencias Experimentales: Informática, contribuyendo al desarrollo de sus habilidades cognitivas conjuntamente con el dictado regular de las asignaturas afines.

1.3. Justificación

La implementación de juegos educativos en la asignatura de Desarrollo del Pensamiento Computacional para estudiantes del primer semestre de la Carrera de Pedagogía de las Ciencias Experimentales Informática, se presenta como una estrategia eficaz para nivelar los conocimientos previos y mejorar la comprensión de conceptos fundamentales, tales como variables, estructuras condicionales y bucles. Los estudiantes que ingresan al primer semestre de esta carrera, a menudo enfrentan dificultades debido a la carencia de formación previa en programación, dado que, en muchas instituciones educativas, esta materia no forma

parte del currículo. Esta deficiencia en los conocimientos básicos dificulta el aprendizaje de los conceptos esenciales de programación, generando frustración y afectando tanto la motivación como el desempeño académico. En este sentido, el uso de juegos educativos ofrece una solución innovadora al transformar el proceso de aprendizaje en una experiencia interactiva y lúdica. Dichos juegos proporcionan una alternativa atractiva y menos intimidante para abordar los fundamentos de la programación, permitiendo que los estudiantes se familiaricen con estos conceptos de manera más dinámica y amena.

La programación es un campo extenso que requiere de práctica constante para alcanzar un dominio efectivo. En este sentido, los juegos educativos proporcionan un entorno seguro y lúdico, en el cual los estudiantes pueden experimentar y aplicar libremente los conceptos aprendidos. Este enfoque facilita el desarrollo progresivo de habilidades clave, como la resolución de problemas y el razonamiento lógico, fundamentales en la programación. A través de estos juegos, los estudiantes no solo reforzarán su comprensión de los fundamentos de la programación, sino que también incrementarán su confianza y motivación para continuar aprendiendo y profundizando en este campo (Garrido, 2023).

Por otra parte, la gamificación en la educación tiene el potencial de incrementar significativamente la motivación de los estudiantes para aprender. Un juego bien diseñado puede generar un sentido de logro y satisfacción a medida que los estudiantes avanzan en los niveles y superan los desafíos, lo que fomenta un mayor compromiso con el proceso de aprendizaje (Garrido, 2023).

En este contexto, el uso de juegos educativos en la enseñanza de los Fundamentos de Programación ofrece una forma atractiva y divertida de aprender, apoyando la comprensión de conceptos clave. Este enfoque no solo complementa los conocimientos adquiridos en clase, sino que también enriquece la experiencia de aprendizaje, facilitando la asimilación de los contenidos. De esta manera, se contribuye al logro de mejores resultados académicos y al desarrollo integral de los estudiantes, promoviendo habilidades cognitivas y emocionales, y motivando una participación constante en su proceso educativo.

1.4. Objetivos

1.4.1. General

- Desarrollar un juego educativo sobre Fundamentos de Programación para estudiantes de la Carrera de Pedagogía de las Ciencias Experimentales Informática.

1.4.2. Específicos

- Identificar las principales dificultades cognitivas que tienen los estudiantes del primer semestre de la Carrera de Pedagogía de las Ciencias Experimentales: Informática en el área de programación.
- Elaborar el estado del arte sobre el uso de juegos educativos como apoyo en el proceso de aprendizaje de los fundamentos de programación.
- Diseñar un juego educativo mediante la herramienta EdiLim para apoyar el aprendizaje de los fundamentos de programación.

CAPÍTULO II

2. MARCO TEÓRICO

2.1. Informática educativa

La informática educativa está transformando los métodos de enseñanza y aprendizaje, generando un impacto positivo en las personas. Actualmente, la manera en que se adquieren y comparten los conocimientos ha evolucionado significativamente, gracias a la amplia disponibilidad de recursos enfocados en la educación informática. Este campo incluye una diversidad de actividades y enfoques, tales como el desarrollo de software educativo, la incorporación de herramientas digitales en los entornos escolares y la investigación orientada a comprender cómo las tecnologías pueden potenciar y enriquecer los procesos de aprendizaje (Rodríguez, 2006).

2.1.1. Historia y evolución

La informática educativa se refiere a la aplicación de las tecnologías de la información y la comunicación (TIC) con el propósito de optimizar los procesos de enseñanza y aprendizaje. Su evolución ha estado marcado por importantes avances tecnológicos y transformaciones en el ámbito educativo. A continuación, se expone un resumen sobre su historia y desarrollo:

Ilustración 1: Evolución de la Informática Educativa



Nota: Línea de tiempo acerca de la Evolución de la Informática.

2.1.2. Tendencias actuales en la enseñanza de la Informática

La enseñanza de la informática busca promover y divulgar las tecnologías de la información mediante métodos pedagógicos efectivos. Para lograrlo, es importante que los educadores posean un conocimiento profundo en diversas áreas de la tecnología y la informática.

El profesional en este ámbito no solo domina el contenido técnico, sino que también cuenta con estrategias didácticas para motivar a los estudiantes y gestionar grupos de manera efectiva. Además, está en constante renovación de sus métodos de enseñanza, adaptándose a las nuevas tendencias y herramientas tecnológicas para garantizar que su conocimiento sea transmitido de manera clara y eficaz, fomentando un aprendizaje significativo en el uso de las tecnologías de la información (Pearson, 2022).

2.2. Fundamentos de programación

Los fundamentos de la programación constituyen los principios esenciales para el desarrollo de programas informáticos. Estos fundamentos abarcan conceptos clave como variables, estructuras de control, bucles, funciones y estructuras de datos, que forman la base de cualquier lenguaje de programación.

Es indispensable, para los programadores, tener un conocimiento sólido de estos principios, ya que les permite escribir código eficiente y estructurado, crear nuevos programas desde cero, y realizar tareas críticas como la depuración y el mantenimiento de software existente. Además, dominar estos fundamentos facilita la adaptación a nuevos lenguajes y tecnologías, convirtiéndolos en habilidades esenciales en el campo de la informática (Rasig, 2024).

2.2.1. Resumen de conceptos claves

- **Algoritmo.** – Es la secuencia de instrucciones ordenadas de manera lógica que resuelven un problema, son la base de la programación. Dominar los algoritmos ayuda a pensar como programador y a implementar soluciones en cualquier lenguaje (Ramírez, 2022).
- **Variable.** - Es una unidad de almacenamiento y recuperación de datos, los mismos que se utilizan con mucha frecuencia al programar. Se identifican con un nombre único y sus valores pueden cambiar (Ramírez, 2022).

- **Sentencia.** - Las sentencias son los componentes básicos del código que permiten a los programadores proporcionar instrucciones a las computadoras (García, 2023).
- **Tipos de datos.** - Son los diferentes tipos de variables en las que se clasifica la información. Por ejemplo, los más utilizados son Number, String, y Boolean (Dtdata, 2023).
- **Estructuras de Control.** - El código generalmente se lee de arriba hacia abajo. Sin embargo, las estructuras de control como los ciclos y las condicionales permiten que el flujo de lectura y ejecución del código cambie, ya que permiten que se lea y ejecute de diferentes maneras según las condiciones o repeticiones establecidas (Maldonado, 2024).
- **Estructuras Repetitivas.** - Un bucle es una parte de un algoritmo o programa en la que un conjunto de instrucciones se repite múltiples veces mientras se cumpla una condición determinada. Este proceso de repetición depende de una condición que se evalúa en cada iteración, y puede ser verdadera o falsa. En cada ciclo, la condición se verifica, y si es verdadera, las instrucciones se ejecutan nuevamente; si es falsa, el bucle se detiene. De esta forma, los bucles permiten automatizar tareas repetitivas dentro de un programa (Gálvez, 2018).
- **Bucle.** – Es un mecanismo mediante el cual se repite un conjunto de instrucciones tantas veces como sea necesario, la culminación del bucle depende de la condición establecida (Guevara, 2021).
- **Condicionales.** - Estructuras condicionales como la if permite que el código tome decisiones en base a ciertos parámetros. Por ejemplo, si la condición se cumple, el programa hará algo, si no se cumple, hará otra cosa (Guevara, 2021).
- **Función.** - Una función es un bloque de código reutilizable que realiza tareas específicas. Cada vez que se invoca una función, su código se ejecuta, lo que permite ahorrar tiempo y evitar la necesidad de escribir las mismas líneas de código repetidamente (Guevara, 2021).

2.2.2. Retos y desafíos comunes en la enseñanza de la Programación

La enseñanza de la programación puede ser muy gratificante, pero también presenta varios retos y desafíos comunes. Algunos de estos desafíos incluyen (Díaz et al., 2018):

- **Abstracción de conceptos:** es común que algunos estudiantes encuentren dificultades al tratar de comprender conceptos abstractos como variables, funciones

y algoritmos. Es importante encontrar formas de explicar estos conceptos de manera clara y práctica.

- **Diversidad de habilidades:** Los estudiantes pueden tener diferentes niveles de habilidad y experiencia en programación. El desafío es adaptar el contenido para satisfacer las necesidades de aquellos que son principiantes y, al mismo tiempo, desafiar a aquellos que ya tienen experiencia.

2.3. Juegos educativos

2.3.1. Concepto

Un juego educativo según Putton (2021), promueve un desarrollo integral y equilibrado en los estudiantes. A través de esta metodología, los estudiantes no solo adquieren conocimientos, sino que también desarrollan habilidades importantes para su crecimiento personal. El juego fomenta la independencia, al permitir que los estudiantes tomen decisiones y resuelvan problemas por sí mismos. Además, mejora su sensibilidad visual y auditiva, ayudándoles a procesar información de manera más eficiente.

2.3.2. Características

Los juegos educativos tienen una serie de características que los hacen efectivos en el proceso de enseñanza-aprendizaje:

- **Interactividad:** Permiten a los jugadores interactuar directamente con el contenido y el entorno del juego, lo que refuerza el aprendizaje activo.
- **Feedback inmediato:** Ofrecen retroalimentación instantánea sobre el desempeño del jugador, lo que les ayuda a identificar errores y mejorar sus habilidades.
- **Motivación intrínseca:** La naturaleza lúdica y entretenida de los juegos mantiene a los estudiantes motivados y comprometidos con el proceso de aprendizaje.
- **Diversión y desafío:** La combinación de diversión y desafío en los juegos mantiene el interés de los estudiantes, mientras les permite aprender y enfrentarse a nuevos retos (Equipo editorial, 2024).

2.3.3. Beneficios

El uso de juegos en el aula proporciona numerosos beneficios para los estudiantes y el proceso de enseñanza:

- **Fomento de la motivación:** Los juegos educativos aumentan la motivación intrínseca, haciendo que el aprendizaje sea más atractivo y divertido, lo que puede mejorar el rendimiento académico.
- **Desarrollo de habilidades cognitivas:** Mejoran habilidades como el pensamiento crítico, la resolución de problemas, la toma de decisiones y la creatividad, al enfrentar a los jugadores a situaciones nuevas y desafiantes.
- **Desarrollo de habilidades sociales:** Los juegos colaborativos permiten que los estudiantes trabajen en equipo, mejoren sus habilidades de comunicación, negociación y colaboración.
- **Reducción de la ansiedad:** El ambiente lúdico reduce la presión de los exámenes tradicionales, permitiendo que los estudiantes aprendan a su propio ritmo y sin temor a cometer errores (Equipo editorial, 2024).

2.4. Aprendizaje de la Programación

La enseñanza de la programación tiene como uno de sus principales objetivos el desarrollo de habilidades en los estudiantes que les permitan resolver problemas tanto en el ámbito académico como profesional, y en situaciones cotidianas, aprovechando las herramientas que ofrecen los distintos lenguajes de programación como Python o Java. En el contexto de la formación de profesores de Informática, este proceso tiene particularidades que lo hacen único. El docente en formación debe no solo adquirir y perfeccionar estas habilidades, sino también apropiarse de los procedimientos y enfoques pedagógicos adecuados para guiar y dirigir eficazmente este proceso de enseñanza en el entorno escolar (Díaz et al., 2018).

2.4.1. ¿Qué es la programación?

Se trata de un conjunto de reglas y símbolos que permiten a los programadores escribir instrucciones que un computador puede comprender y ejecutar. Estos lenguajes actúan como intermediarios entre el lenguaje humano, que utilizamos para comunicarnos, y el lenguaje

de la máquina, que es el conjunto de instrucciones binarias que una computadora entiende directamente (Sulbarán, 2023).

2.4.2. Tipos de programación.

Existen diversos tipos de programación los cuales se detalla a continuación:

- **Programación Imperativa.** - El programa se estructura como una serie de comandos o instrucciones que modifican el estado del programa a medida que se ejecutan. Las instrucciones imperativas indican a la computadora qué pasos debe seguir para lograr un resultado deseado.
- **Programación Orientada A Objetos (Poo).** - Es un paradigma de programación, que se basa en la idea de organizar y estructurar el código, de manera que los conceptos del mundo real se reflejen de manera natural en él. En la POO, los programas se diseñan alrededor de "objetos", que son unidades independientes que encapsulan datos y funciones relacionadas.
- **Programación Funcional.** – En este caso, los programas se construyen mediante la composición de funciones, lo que permite un enfoque más declarativo y menos centrado en la manipulación directa de datos.
- **Programación Lógica.** – Este paradigma es especialmente adecuado para resolver problemas que implican razonamiento y manipulación simbólica.
- **Programación Declarativa.** - La programación declarativa se proporciona una especificación de lo que se desea obtener, permitiendo que el sistema o el lenguaje de programación se encargue de determinar cómo se llevará a cabo la ejecución.
- **Programación Orientada a Eventos.** - En la programación declarativa, se especifica qué se desea obtener, dejando que el sistema o el lenguaje de programación determine cómo se llevará a cabo la ejecución. (Sulbarán, 2023).

En conclusión, todos estos tipos de programación poseen características únicas que los hacen idóneos para abordar diferentes contextos y desafíos. Estas particularidades ofrecen a los desarrolladores un repertorio diverso de herramientas y estrategias, permitiéndoles elegir el enfoque más adecuado para diseñar y construir soluciones eficientes y efectivas.

2.5. EdiLim

Según el autor Rodríguez (2020), EdiLim es una herramienta de software enfocada en la creación de actividades interactivas, como ejercicios de asociación con flechas, especialmente útiles para pizarras digitales interactivas (PDI) o para la exposición de conceptos teóricos que pueden complementarse con actividades prácticas. Su principal ventaja radica en la posibilidad de diseñar libros interactivos que abarcan una amplia gama de temas, aprovechando las diversas opciones de actividades disponibles.

Además, esta herramienta permite crear un sitio web que concentre todos los elementos necesarios para desarrollar un bloque completo de contenidos, sin limitaciones en el número de páginas y sin incluir publicidad, lo que mejora la experiencia de uso.

2.5.1. Características

Tabla 1: Características EdiLim

NOMBRE	CARACTERÍSTICAS	VENTAJAS
EdiLim	Empaquetamiento: Scorm	• Permite el uso de recursos multimedia
	Licencia: Gratuita	• Permite crear 51 tipos de actividad.
	Plataforma: Windows - Linux - MacOS	• Fácil de utilizar e intuitiva Flexible

Nota: Tabla elaborada con las características principales de Edilim, por el autor (Rodríguez, 2020).

2.5.2. Usos de EdiLim en la educación.

EdiLim es una herramienta versátil que ofrece múltiples usos en el ámbito educativo, facilitando el diseño de actividades interactivas y dinámicas que enriquecen el proceso de enseñanza-aprendizaje. Algunos de sus usos más destacados son:

- **Recordar** los conocimientos previos mostrando resultados de unidades pasadas.
- **Comprender** el siguiente nivel a través de las actividades destinadas a mostrar información.

- **Aplicar** los nuevos conceptos desarrollando actividades interactivas de diversos tipos.
- Y **evaluar** todo el proceso (López, 2020).

2.6. Gamificación

Como afirma (Torres, 2022), la gamificación es una estrategia de aprendizaje que incorpora elementos y dinámicas propias de los juegos en contextos educativos y profesionales. Su propósito es mejorar los resultados del aprendizaje, ya sea mediante la adquisición más efectiva de conocimientos, el desarrollo de habilidades específicas o el reconocimiento de acciones concretas, entre otros objetivos.

Este enfoque está ganando popularidad en las metodologías de formación debido a su componente lúdico, que no solo hace el proceso más atractivo y entretenido, sino que también facilita la asimilación de los contenidos, creando experiencias positivas y motivadoras para los participantes.

2.6.1. Características principales de la gamificación en la educación.

La gamificación en la educación se convierte en una herramienta poderosa cuando se aplica al desarrollo de juegos educativos, ya que transforma el aprendizaje en una experiencia interactiva y motivadora. Al integrar elementos clave como recompensas, puntos y premios, los juegos educativos fomentan la participación de los estudiantes al establecer objetivos claros que pueden alcanzar de manera divertida.

Además, la incorporación de sistemas de ranking y puntuaciones en los juegos educativos crea un ambiente de competencia amigable, incentivando la superación personal y el compromiso con las actividades. La retroalimentación inmediata que ofrecen estos juegos permite a los estudiantes ajustar su enfoque, mientras que los logros y niveles les ayudan a medir su progreso y a mantenerse motivados para avanzar (Digital School, 2024).

2.6.2. Importancia de la gamificación dentro del aula de clases.

El principal propósito de la gamificación en el ámbito educativo es diseñar actividades dinámicas y atractivas que funcionen como un juego, donde los estudiantes, al completar correctamente sus asignaciones, puedan avanzar de nivel y obtener mejores puntuaciones.

El enfoque central radica en demostrar que el aprendizaje puede ser divertido y motivador, convirtiéndolo en una experiencia interactiva que fomenta la participación. A través de esta metodología, ya sea en el aula o en el hogar, los estudiantes aprenden mientras superan desafíos, escalando niveles y disfrutando del proceso educativo de una manera significativa y entretenida (Digital School, 2024).

2.7. Metodología Dicrevoa

La metodología DICREVOA contempla cinco fases que establecen el camino a seguir para desarrollar Objetos de Aprendizaje (OA). Este enfoque, ampliamente utilizado en el diseño y creación de materiales educativos digitales, ofrece una serie de directrices que permiten al docente tomar decisiones clave relacionadas con la planificación del proceso de aprendizaje mediante el OA que se desea crear (Maldonado et al., 2017). La metodología se organiza en 5 fases con una secuencia de 16 tareas consecutivas, orientadas a guiar al docente en el desarrollo del OA. Además, integra competencias didácticas y digitales, lo que contribuye a un diseño más efectivo y asegura una funcionalidad positiva de los OA.

2.7.1. Fases

La metodología Dicrevoa cuenta de 5 fases que se detalla a continuación:

- **Análisis**

Según la investigación de Maldonado et al., (2017), esta fase implica la recopilación de información relevante sobre la necesidad específica del Objeto de Aprendizaje y los destinatarios a quienes está dirigido. Este proceso es crucial, ya que permite entender el contexto, los objetivos educativos y las características del público objetivo. Por este motivo, el autor del Objeto de Aprendizaje desempeña un papel central, participando activamente en la identificación de los requerimientos y asegurando que el diseño responda de manera adecuada a las necesidades detectadas.

- **Diseño**

De acuerdo con Maldonado et al. (2017), en esta etapa se lleva a cabo la planificación del diseño del Objeto de Aprendizaje, considerando tanto los aspectos pedagógicos como los tecnológicos. Se aborda el diseño instruccional, que proporciona una estructura

educativa sólida, y el diseño multimedial, que se enfoca en crear una presentación visual y técnica que favorezca el proceso de aprendizaje.

- **Implementación**

Según Maldonado et al. (2017), utilizando herramientas informáticas, se construye la estructura del esquema general del Objeto de Aprendizaje desarrollado en la etapa de diseño.

- **Evaluación**

Como afirma Maldonado et al. (2017), la fase de evaluación debe garantizar que el Objeto de Aprendizaje promueva resultados académicos positivos y cumpla con características esenciales como interoperabilidad, valor educativo, capacidad generativa, accesibilidad y reutilización.

- **Publicación**

Como señalan Maldonado et al. (2017), la fase de publicación requiere un conocimiento adecuado de las licencias aplicables a los contenidos digitales, ya que el Objeto de Aprendizaje no se limita a incluir textos, sino que también integra elementos como fotografías, videos, animaciones, música y software, entre otros.

CAPÍTULO III

3. METODOLOGIA

El presente trabajo de investigación describe el proceso empleado para el desarrollo de un juego educativo sobre fundamentos de programación para estudiantes del primer semestre de la Carrera de Pedagogía de las Ciencias Experimentales Informática, el cual se basa en actividades que pretenden mejorar la comprensión de los contenidos, fomentando a su vez el desarrollo de habilidades cognitivas en los estudiantes.

Las actividades del juego educativo fueron desarrolladas partiendo de una encuesta aplicada a estudiantes de primero, segundo y tercer semestre de la Carrera al final del período académico 2023-2S (Anexo 2). La interfaz del juego de aprendizaje se desarrolló de forma dinámica e interactiva, puesto a que contiene un menú con sus respectivos temas y cada uno de ellos contiene actividades con identificación de imágenes, clasificación de imágenes, arrastrar textos, entre otras más las cuales vienen implementadas dentro de la herramienta EdiLim.

3.1. Tipo de investigación

La presente investigación es de tipo Aplicada, ya que según el autor (Duoc, 2024), nos indica que la Investigación Aplicada tiene por objetivo resolver un determinado problema o planteamiento específico, enfocándose en la búsqueda y consolidación del conocimiento para su aplicación. En este caso, se desarrolló un juego para nivelar los conocimientos de los estudiantes de la Carrera de Pedagogía de las Ciencias Experimentales Informática, sin embargo, se dejó como base para futuras investigaciones, en las cuales se pueda comprobar su eficacia.

3.2. Enfoque de la investigación

El presente trabajo de investigación tiene un enfoque cualitativo, ya que se hizo una revisión de bibliográfica para una mejor comprensión del objeto de estudio. Además, se ha indagado en diferentes repositorios bibliográficos institucionales (como el DSpace), revistas indexadas, artículos científicos y Google académico, lo que permitió recabar información a

partir de una variedad de documentos digitales sobre los juegos educativos en el área de la programación.

Además de ello, se realizó una encuesta dirigida a la población de estudio para tener una orientación objetiva de los contenidos sobre los cuales se crearían las actividades del juego educativo.

3.3. Alcance de la investigación

La presente investigación se realizó a través de un estudio descriptivo, porque se desarrolla bajo un enfoque cualitativo dado que permite describir una situación educativa determinada en un contexto puntual señalando que dentro de este estudio se analiza cuáles son las principales dificultades que tienen los estudiantes en el área de la programación.

3.4. Diseño de la investigación

Esta investigación tiene un diseño tecnológico ya que estará enfocada en la exploración de conocimientos y busca la resolución de problemas, mediante la aplicación de las tecnologías informáticas apropiadas para resolver un problema o satisfacer una necesidad.

3.5. Técnicas de recolección de Datos

Se utilizó la técnica de la encuesta para recopilar los datos, ya que permitió obtener información importante sobre el objeto de estudio. Esto facilitó la identificación de aspectos clave para el desarrollo del juego educativo, tomando como base los resultados obtenidos tras el proceso de tabulación y análisis de los datos recolectados.

Según Casas et al., (2003), menciona que la encuesta es utilizada como procedimiento de investigación, ya que permite obtener y elaborar datos de modo rápido y eficaz.

En este caso se aplicó una encuesta estructurada entre los estudiantes de primer, segundo y tercer semestre de la Carrera de Pedagogía de las Ciencias Experimentales Informática, matriculados en el período académico 2023-2S (Ver Anexo 2)

3.6. Población de estudio y tamaño de muestra

La población estuvo compuesta por estudiantes de primer, segundo y tercer semestre de la Carrera de Pedagogía de las Ciencias Experimentales en Informática, correspondientes al periodo académico 2023-2S. La muestra fue de tipo no probabilística intencional y estuvo conformada por 67 estudiantes de los grupos mencionados anteriormente.

3.7. Análisis de recolección de datos

Los datos fueron recopilados por medio de un formulario de Office 365 (Microsoft Forms Anexo 2), al final del período académico 2023-2S (esto fue en febrero del 2024). Los datos recopilados se descargaron en un formato compatible con Microsoft Excel, a partir del cual se elaboraron tablas dinámicas y gráficos para facilitar su análisis e interpretación.

CAPÍTULO IV

4. RESULTADOS

El presente capítulo muestra los resultados de la tabulación y análisis de los datos obtenidos mediante la aplicación de una encuesta (anexo 2) a 67 estudiantes de la Carrera de Pedagogía de las Ciencias Experimentales Informática pertenecientes a primero, segundo y tercer semestre. Lo cual permitió alcanzar el segundo objetivo específico de la investigación relacionada con las dificultades cognitivas que tienen los estudiantes en el área de la programación.

La encuesta aplicada consta de 16 preguntas cerradas y 3 preguntas abiertas, las mismas que se encuentran divididas en cuatro secciones (datos demográficos, aprendizaje de los contenidos, principales desafíos de aprender a programar, sugerencias y resultados).

4.1. Datos demográficos

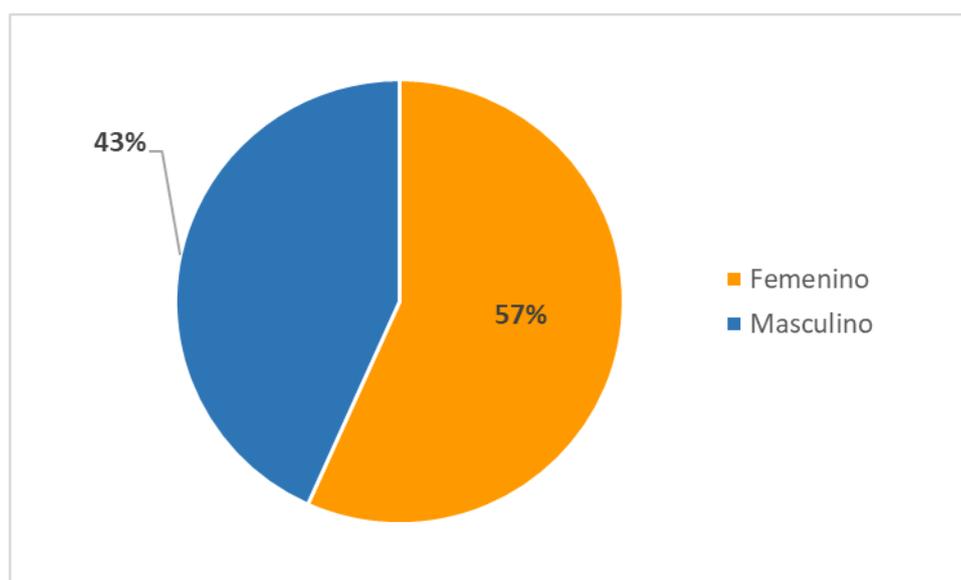
Pregunta 1. Género

Tabla 2. Género de los estudiantes encuestados

ALTERNATIVA	FRECUENCIA	PORCENTAJE
Masculino	29	43%
Femenino	38	57%
TOTAL	67	100 %

Nota: Encuesta diagnóstica de las dificultades cognitivas en el área de la programación.

Gráfico 1. Género de los estudiantes encuestados



Nota: Elaboración propia a partir de la Tabla 1.

De acuerdo con la Tabla 2, existe un 14% de diferencia entre el género Masculino y Femenino de los estudiantes encuestados de la carrera de Pedagogía de las Ciencias Experimentales Informática. Por medio de esta encuesta se puede observar que en la carrera existen menos hombres que mujeres a la fecha de la redacción del presente informe.

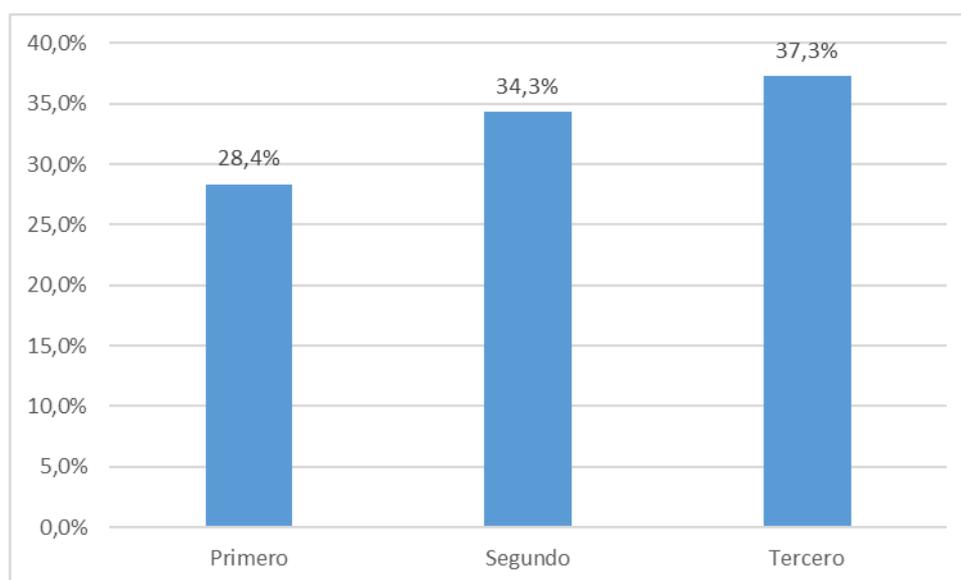
Pregunta 2. Semestre

Tabla 3. Distribución de estudiantes encuestados por semestre

ALTERNATIVA	FRECUENCIA	PORCENTAJE
Primero	19	28,4%
Segundo	23	34,3%
Tercero	25	37,3%
TOTAL	67	100 %

Nota: Cuestionario aplicado a los estudiantes de primero, segundo y tercer semestre de la Carrera de Pedagogía de las Ciencias Experimentales Informática.

Gráfico 2. Distribución porcentual de los estudiantes encuestados



Nota: Elaboración propia a partir de la Tabla 3.

Los estudiantes que conformaron la población fueron los estudiantes matriculados en los tres primeros semestres de la carrera de Pedagogía de las Ciencias Experimentales: Informática, durante el período académico 2023-2S. En el gráfico 2 se puede observar la distribución de la población.

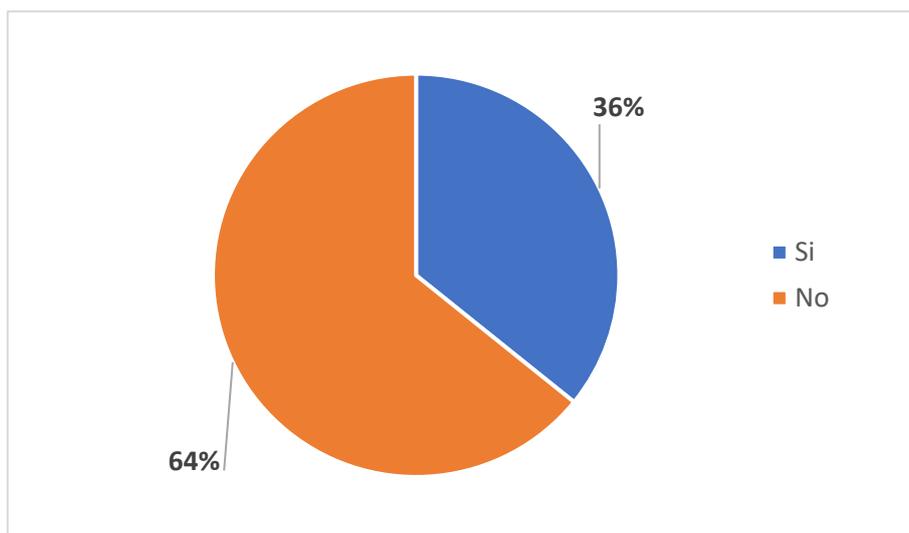
Pregunta 3. ¿Tenías una experiencia o conocimiento previo en cuanto a Programación, antes de ingresar a la Carrera de Pedagogía de la Informática?

Tabla 4. Experiencia y conocimientos previos en el campo de la programación de computadores

ALTERNATIVA	FRECUENCIA	PORCENTAJE
No	43	64%
Si	24	36%
TOTAL	67	100 %

Nota: Cuestionario aplicado a los estudiantes de primero, segundo y tercer semestre de la Carrera de Pedagogía de las Ciencias Experimentales Informática.

Gráfico 3. Experiencia y conocimientos previos en el campo de la programación de computadores



Nota: Elaboración propia a partir de la Tabla 4.

Se puede observar en el gráfico 3 que más de la mitad de los estudiantes no tenían conocimiento alguno en cuanto a programación al momento de ingresar a la Carrera, mientras que un 36% de los encuestados afirmaron haber tenido conocimientos básicos al momento de su primer ingreso.

4.2. Contenidos básicos de Programación

A. DATOS, OPERADORES Y EXPRESIONES

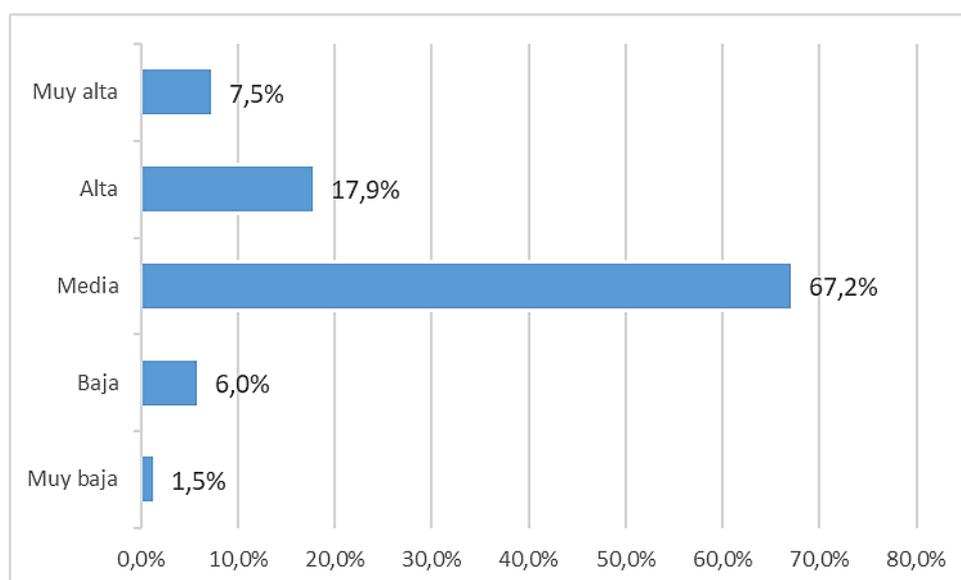
Pregunta 4. ¿Cómo calificarías tu comprensión de los tipos de datos y su uso en la programación?

Tabla 5. Nivel de comprensión sobre los tipos de datos y su uso en la programación.

ALTERNATIVA	FRECUENCIA	PORCENTAJE
Muy baja	1	1,5%
Baja	4	6,0%
Media	45	67,2%
Alta	12	17,9%
Muy alta	5	7,5%
TOTAL	67	100 %

Nota: Cuestionario aplicado a los estudiantes de primero, segundo y tercer semestre de la Carrera de Pedagogía de las Ciencias Experimentales Informática.

Gráfico 4. Nivel de comprensión sobre los tipos de datos y su uso en la programación.



Nota: Elaboración propia a partir de la Tabla 5.

Se puede observar en el gráfico 4 que un 67,2% de los estudiantes encuestados tienen un nivel medio de comprensión en cuanto a los tipos de datos y su uso en la programación. Apenas una 25% de ellos afirman haber logrado una comprensión adecuada de este contenido y su aplicabilidad.

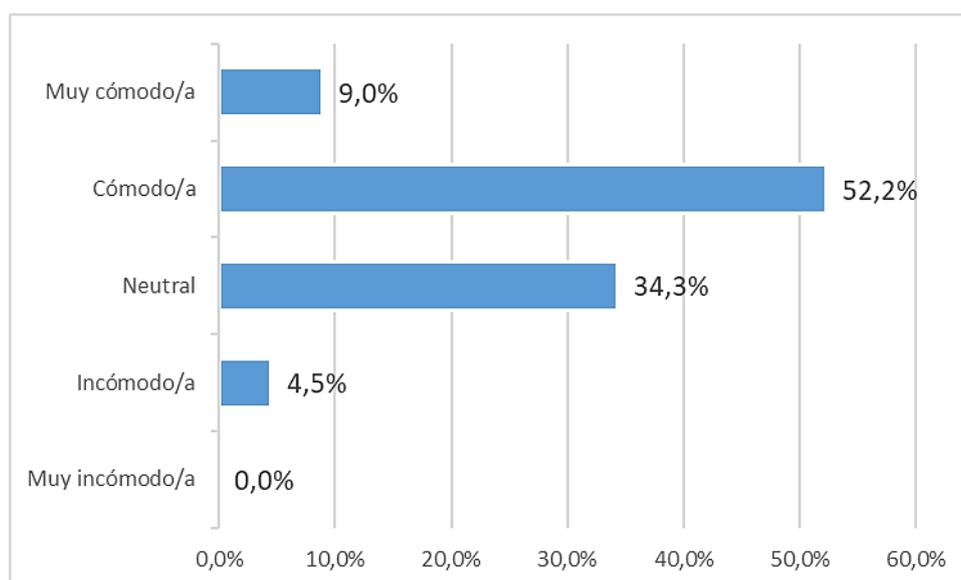
Pregunta 5. Al trabajar con operadores y expresiones, ¿qué tan seguro(a) te sientes realizando operaciones complejas y combinando diferentes tipos de datos?

Tabla 6. Nivel de seguridad en el uso de operadores y expresiones

ALTERNATIVA	FRECUENCIA	PORCENTAJE
Muy incómodo/a	0	0,0%
Incómodo/a	3	4,5%
Neutral	23	34,3%
Cómodo/a	35	52,2%
Muy cómodo/a	6	9,0%
TOTAL	67	100,0 %

Nota: Cuestionario aplicado a los estudiantes de primero, segundo y tercer semestre de la Carrera de Pedagogía de las Ciencias Experimentales Informática.

Gráfico 5. Nivel de seguridad en el uso de operadores y expresiones



Nota: Elaboración propia a partir de la Tabla 6.

En el gráfico 5 se puede observar que más del 50% de los estudiantes se sienten cómodos al momento de utilizar operadores y resolver expresiones. Sin embargo, un 4,5% de ellos afirman tener inseguridad en el correcto uso de estos contenidos.

B. ESTRUCTURA DE CONTROL SECUENCIAL

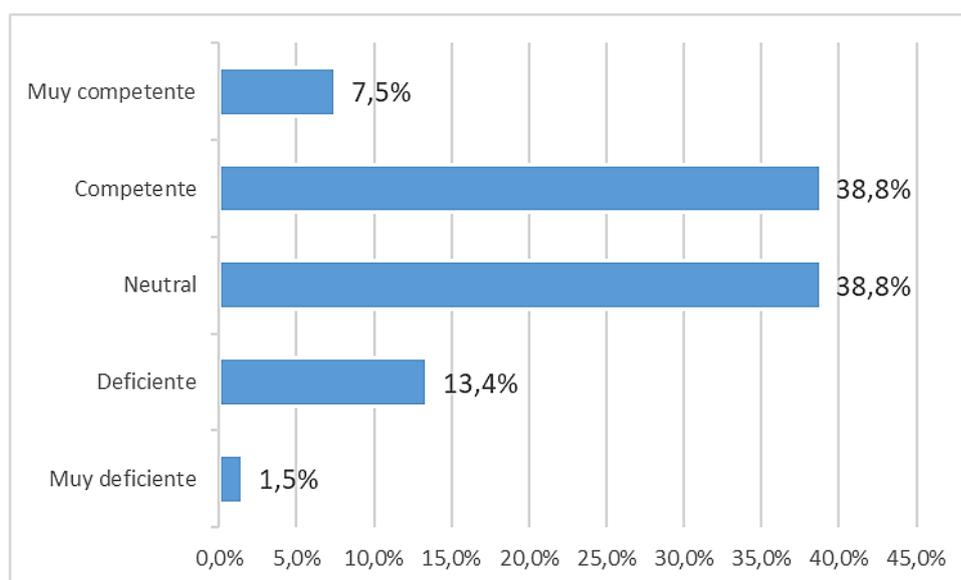
Pregunta 6. ¿Cómo evaluarías tu habilidad para escribir programas que sigan una secuencia lógica de instrucciones?

Tabla 7. Habilidad para escribir programas que sigan una secuencia lógica

ALTERNATIVA	FRECUENCIA	PORCENTAJE
Muy deficiente	1	1,5%
Deficiente	9	13,4%
Neutral	26	38,8%
Competente	26	38,8%
Muy competente	5	7,5%
TOTAL	67	100,0 %

Nota: Cuestionario aplicado a los estudiantes de primero, segundo y tercer semestre de la Carrera de Pedagogía de las Ciencias Experimentales Informática.

Gráfico 6. Habilidad para escribir programas que sigan una secuencia lógica



Nota: Elaboración propia a partir de la Tabla 7.

En el gráfico 6, se puede observar que más del 40% de los estudiantes encuestados se consideran competentes en la escritura secuencial y lógica de instrucciones. Por otra parte, menos del 15% de los estudiantes manifiestan tener un nivel deficiente en cuanto a sus habilidades para escribir y resolver problemas empleando correctamente la escritura de secuencias lógicas de instrucciones.

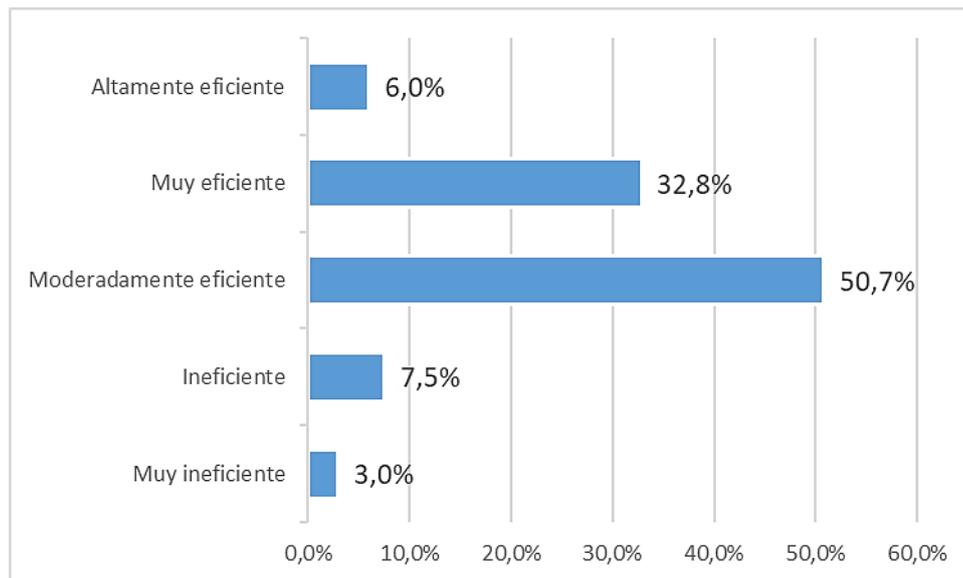
Pregunta 7. Al enfrentarte a problemas que requieren el uso de procesos secuenciales, ¿qué tan eficiente puedes ser al planificar y organizar tu código?

Tabla 8. Eficiencia al planificar y organizar un código

ALTERNATIVA	FRECUENCIA	PORCENTAJE
Muy ineficiente	2	3,0%
Ineficiente	5	7,5%
Moderadamente eficiente	34	50,7%
Muy eficiente	22	32,8%
Altamente eficiente	4	6,0%
TOTAL	67	100,0 %

Nota: Cuestionario aplicado a los estudiantes de primero, segundo y tercer semestre de la Carrera de Pedagogía de las Ciencias Experimentales Informática.

Gráfico 7. Eficiencia al planificar y organizar un código



Nota: Elaboración propia a partir de la Tabla 8.

En el gráfico 7 se puede apreciar que más del 80% de los estudiantes son eficientes al momento de planificar y organizar un código. Por otro lado, apenas un 10,5% de los encuestados afirman ser ineficientes en el logro de esta habilidad.

C. ESTRUCTURAS DE CONTROL CONDICIONAL

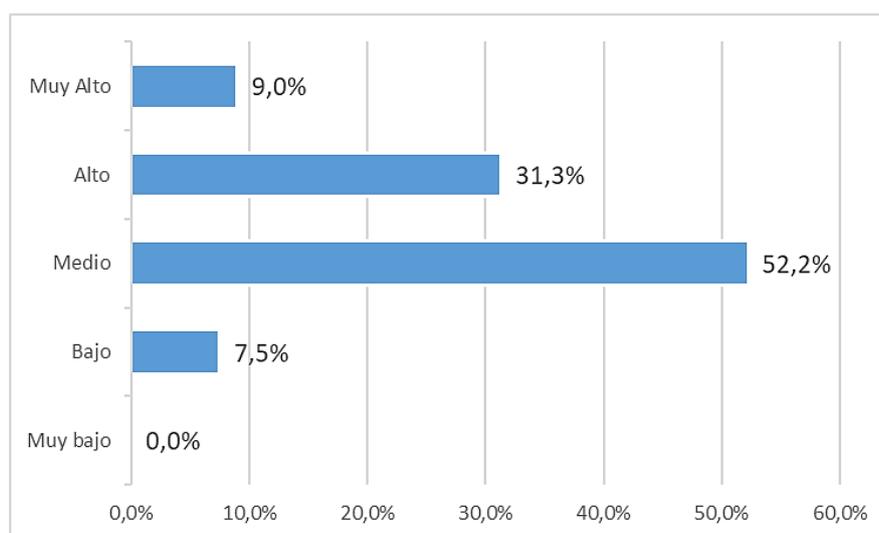
Pregunta 8. ¿Cuál es tu nivel de confianza al utilizar estructuras condicionales (if ... else) para controlar el flujo de ejecución en un programa?

Tabla 9. Nivel de confianza al utilizar estructuras condicionales

ALTERNATIVA	FRECUENCIA	PORCENTAJE
Muy bajo	0	0,0%
Bajo	5	7,5%
Medio	35	52,2%
Alto	21	31,3%
Muy Alto	6	9,0%
TOTAL	67	100,0 %

Nota: Cuestionario aplicado a los estudiantes de primero, segundo y tercer semestre de la Carrera de Pedagogía de las Ciencias Experimentales Informática.

Gráfico 8. Nivel de confianza al utilizar estructuras condicionales



Nota: Elaboración propia a partir de la Tabla 9.

Los resultados del gráfico 8 muestran que la gran mayoría de los estudiantes encuestados, esto es el 92,5%, tienen un nivel aceptable de confianza al utilizar estructuras condicionales, lo cual es una señal positiva de su competencia para controlar el flujo de ejecución de la programación. Mientras que, un pequeño grupo de ellos (7,5%) todavía no se sienten seguros de poder utilizar eficientemente la estructura condicional. Esto sugiere la necesidad de trabajar y reforzar la comprensión y práctica en el uso de estructuras condicionales con esta minoría de estudiantes para lograr una correcta nivelación de conocimientos y habilidades.

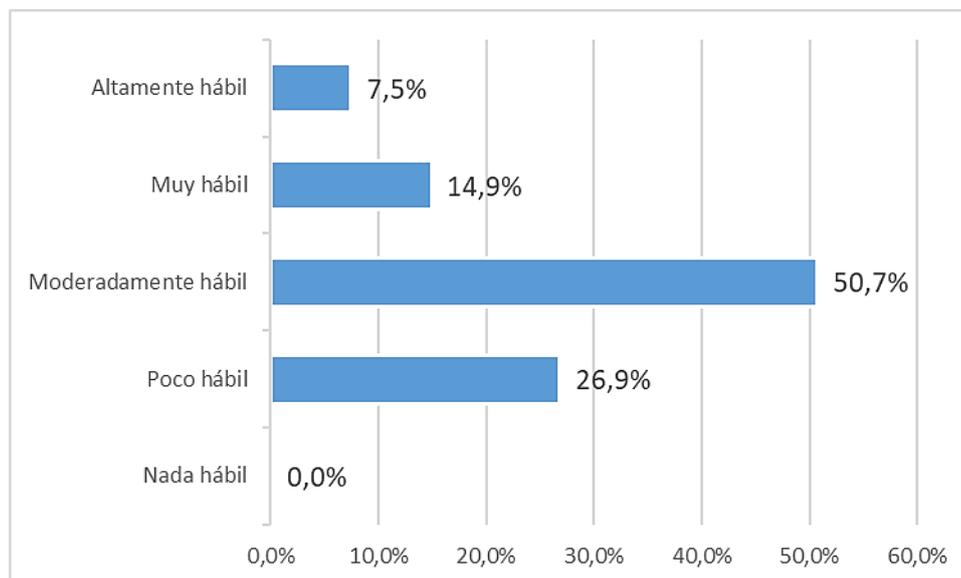
Pregunta 9. Al enfrentarte a problemas que involucran decisiones condicionales, ¿qué tan hábil eres para diseñar y aplicar estructuras de control condicional de manera efectiva?

Tabla 10. Habilidad para diseñar y aplicar estructuras de control condicionales

ALTERNATIVA	FRECUENCIA	PORCENTAJE
Nada hábil	0	0,0%
Poco hábil	18	26,9%
Moderadamente hábil	34	50,7%
Muy hábil	10	14,9%
Altamente hábil	5	7,5%
TOTAL	67	100,0 %

Nota: Cuestionario aplicado a los estudiantes de primero, segundo y tercer semestre de la Carrera de Pedagogía de las Ciencias Experimentales Informática.

Gráfico 9. Habilidad para diseñar y aplicar estructuras de control condicionales



Nota: Elaboración propia a partir de la Tabla 10.

En la figura 9 se puede evidenciar que un 73% de los estudiantes encuestados han desarrollado habilidades para diseñar y aplicar eficientemente estructuras de control condicionales en la resolución de problemas. Mientras que un 27% de ellos afirman ser poco hábiles en el uso de las estructuras de control condicionales.

D. ESTRUCTURA DE CONTROL CÍCLICA

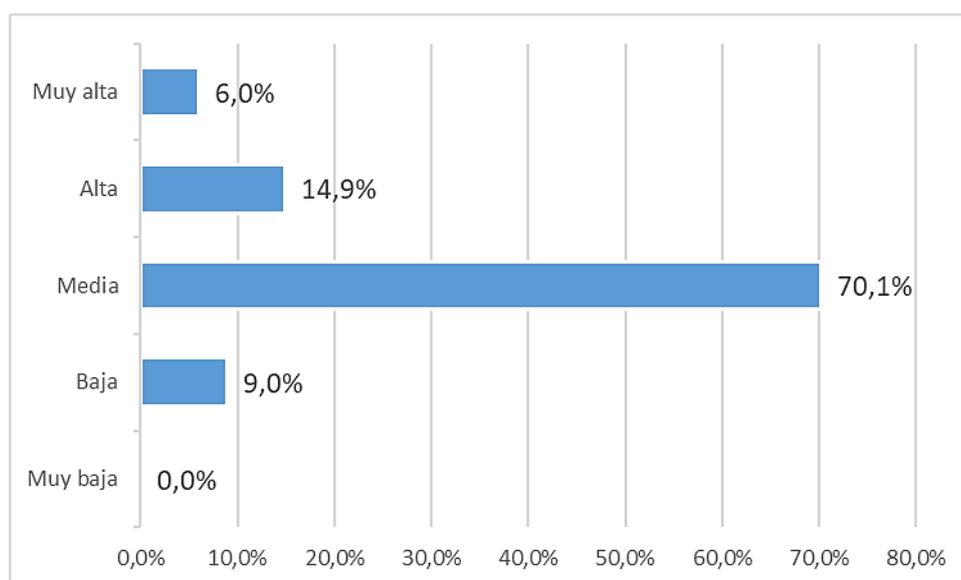
Pregunta 10. ¿Cómo evaluarías tu capacidad para implementar bucles (for, while) en programas (o soluciones) para repetir acciones de manera controlada?

Tabla 11. Capacidad para implementar bucles en programas para repetir acciones de manera controlada

ALTERNATIVA	FRECUENCIA	PORCENTAJE
Muy baja	0	0,0%
Baja	6	9,0%
Media	47	70,1%
Alta	10	14,9%
Muy alta	4	6,0%
TOTAL	67	100,0 %

Nota: Cuestionario aplicado a los estudiantes de primero, segundo y tercer semestre de la Carrera de Pedagogía de las Ciencias Experimentales Informática.

Gráfico 10. Capacidad para implementar bucles en programas para repetir acciones de manera controlada



Nota: Elaboración propia a partir de la Tabla 11.

Continuando con el análisis de la información, en el gráfico 10 se aprecia que 70% de los estudiantes encuestados poseen un nivel medio de destrezas para implementar bucles en soluciones informáticas. Un 21% de los estudiantes han logrado desarrollar la capacidad de integrar bucles en las soluciones y solo un 9% de ellos considera que su nivel es bajo al tratar de utilizar bucles en la programación de soluciones.

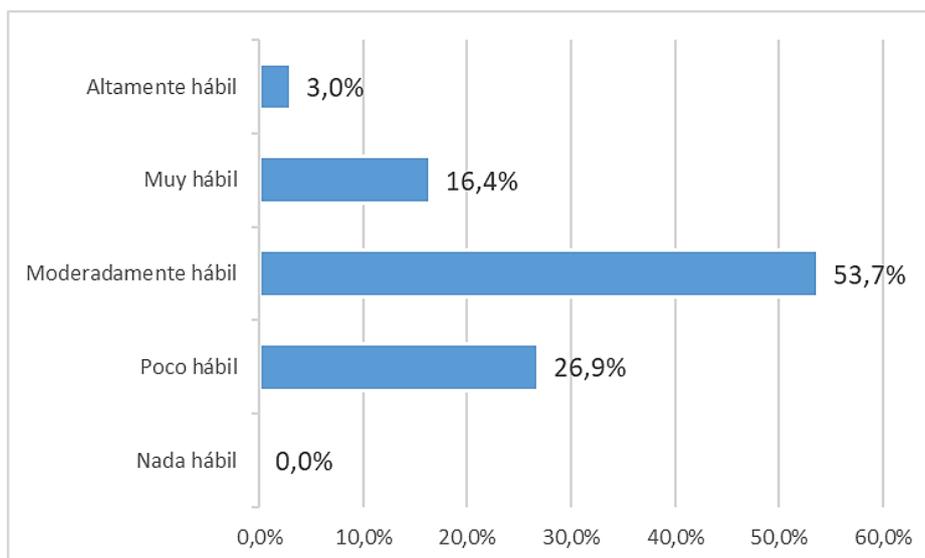
Pregunta 11. En proyectos que requieren iteraciones, ¿qué tan hábil puedes ser para diseñar bucles que optimicen el rendimiento y reduzcan la complejidad de la solución?

Tabla 12. Habilidad para diseñar bucles para optimizar el rendimiento y reducir la complejidad de soluciones

ALTERNATIVA	FRECUENCIA	PORCENTAJE
Nada hábil	0	0,0%
Poco hábil	18	26,9%
Moderadamente hábil	36	53,7%
Muy hábil	11	16,4%
Altamente hábil	2	3,0%
TOTAL	67	100,0 %

Nota: Cuestionario aplicado a los estudiantes de primero, segundo y tercer semestre de la Carrera de Pedagogía de las Ciencias Experimentales Informática.

Gráfico 11. Habilidad para diseñar bucles para optimizar el rendimiento y reducir la complejidad de soluciones



Nota: Elaboración propia a partir de la Tabla 12.

En el gráfico 11, la mayoría de los estudiantes (73,1%) se consideran hábiles para diseñar bucles que optimicen el rendimiento y la complejidad de una solución informática. Sin embargo, un 26,9% de los estudiantes se consideran poco hábiles, lo que conlleva a entender que estos estudiantes requieren de refuerzo para mejorar su habilidad y así poder lograr un mejor nivel de destreza para diseñar bucles efectivos en los proyectos.

E. MODULARIDAD

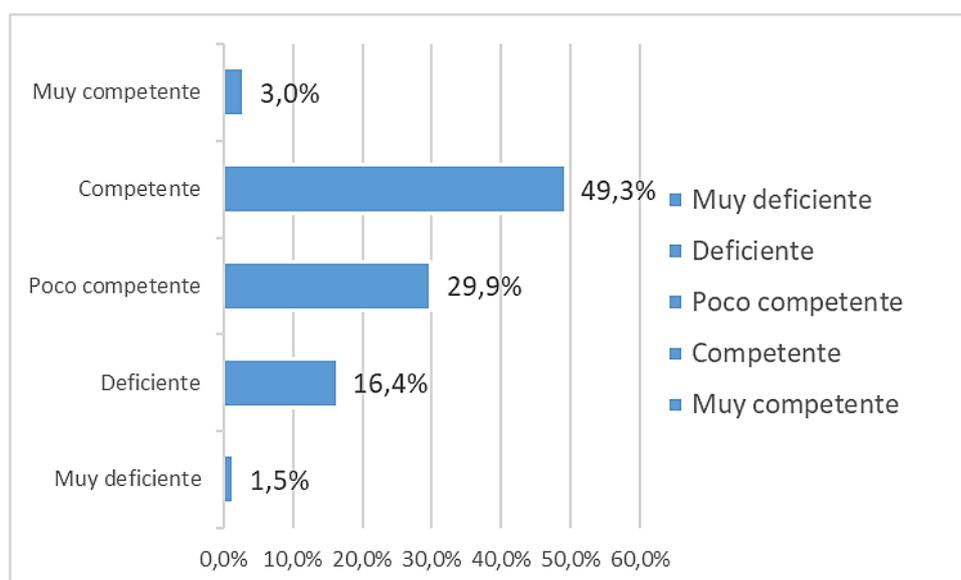
Pregunta 12. ¿Cómo evaluarías tu comprensión y la aplicación de funciones en la programación de soluciones informáticas?

Tabla 13. Comprensión y aplicación de funciones

ALTERNATIVA	FRECUENCIA	PORCENTAJE
Muy deficiente	1	1,5%
Deficiente	11	16,4%
Poco competente	20	29,9%
Competente	33	49,3%
Muy competente	2	3,0%
TOTAL	67	100,0 %

Nota: Cuestionario aplicado a los estudiantes de primero, segundo y tercer semestre de la Carrera de Pedagogía de las Ciencias Experimentales Informática.

Gráfico 12. Comprensión y aplicación de funciones



Nota: Elaboración propia a partir de la Tabla 13.

En el gráfico 12, se puede observar que poco más del 50% de los estudiantes encuestados afirman ser competentes en cuanto a la comprensión y aplicación de funciones en la programación de soluciones informáticas. Por otra parte, cerca del 30% de ellos manifiestan ser poco competentes en este aspecto, mientras que poco más del 16% señalan tener casi nada de competencia en el uso de funciones. Este último resultado resaltaría la necesidad de implementar un refuerzo educativo en el uso de funciones para asegurar que todos los estudiantes alcancen un nivel de competencia aceptable.

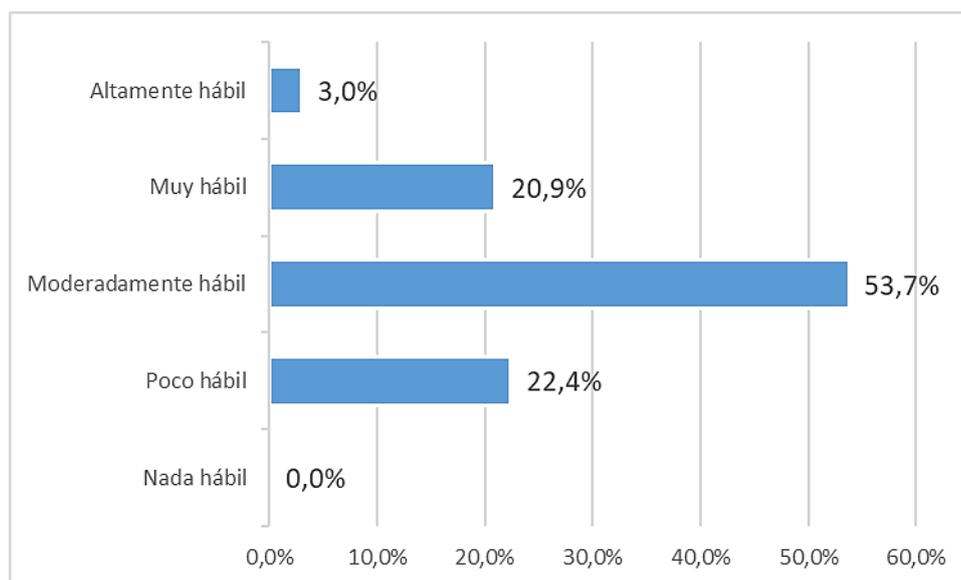
Pregunta 13. Al diseñar programas, ¿qué tan hábil eres para modularizar el código creando funciones que permitan mejorar la legibilidad y la reutilización del código?

Tabla 14. Habilidad para crear funciones que mejoren la legibilidad y reutilización del código

ALTERNATIVA	FRECUENCIA	PORCENTAJE
Nada hábil	0	0,0%
Poco hábil	15	22,4%
Moderadamente hábil	36	53,7%
Muy hábil	14	20,9%
Altamente hábil	2	3,0%
TOTAL	67	100,0 %

Nota: Cuestionario aplicado a los estudiantes de primero, segundo y tercer semestre de la Carrera de Pedagogía de las Ciencias Experimentales Informática.

Gráfico 13. Habilidad para crear funciones que mejoren la legibilidad y reutilización del código



Nota: Elaboración propia a partir de la Tabla 14.

La gran mayoría de los estudiantes (77,6%) afirman ser hábiles para crear funciones que permitan mejorar la legibilidad y reutilización del código, tal como se aprecia en el gráfico 13. Sin embargo, el 22,4% de los encuestados se reconocen como poco hábiles para la creación de funciones, lo que hace referencia a la poca práctica y a la necesidad de mejorar su competencia en la modularización del código a través de mecanismos o recursos didácticos efectivos.

F. RESOLUCIÓN DE PROBLEMAS

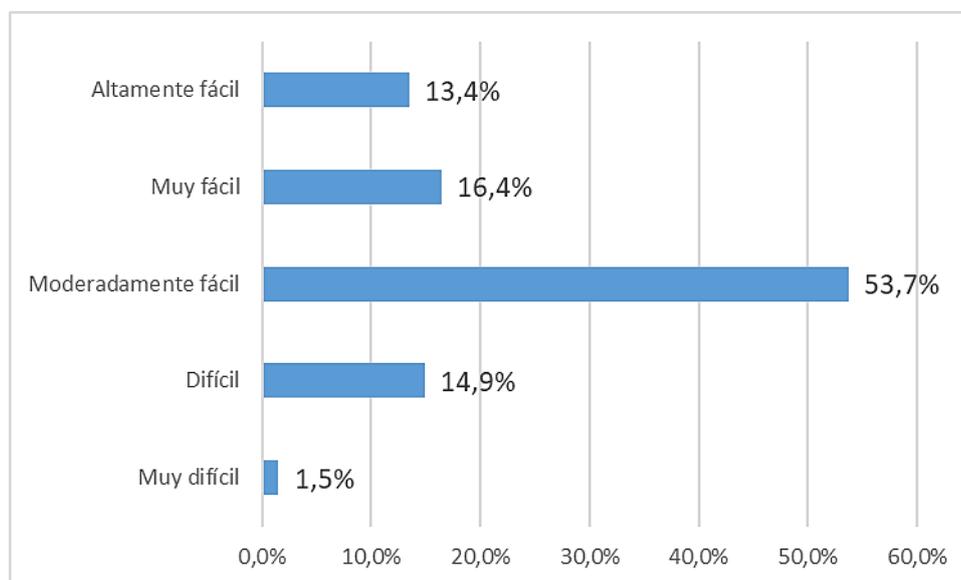
Pregunta 14. Detectar errores en el código y corregirlos

Tabla 15. Capacidad para depurar código

ALTERNATIVA	FRECUENCIA	PORCENTAJE
Muy difícil	1	1,5%
Difícil	10	14,9%
Moderadamente fácil	36	53,7%
Muy fácil	11	16,4%
Altamente fácil	9	13,4%
TOTAL	67	100,0 %

Nota: Cuestionario aplicado a los estudiantes de primero, segundo y tercer semestre de la Carrera de Pedagogía de las Ciencias Experimentales Informática.

Gráfico 14. Capacidad para depurar código



Nota: Elaboración propia a partir de la Tabla 15.

En el gráfico 15, se puede ver como el 83,6% de los estudiantes encuestados consideran la depuración de código como una acción que fácilmente lo pueden lograr. Lo que sugiere que este grupo posee una buena capacidad para identificar y corregir errores en el código de sus programas. Todo lo contrario, pasa con un grupo de estudiantes que conforman el 16,4% de los encuestados, los cuales consideran difícil la depuración del código, lo cual explicaría su poca capacidad para detectar errores en el código y lograr obtener programas funcionales.

4.3. Análisis del dominio de comprensión y aplicación de conceptos básicos de Programación

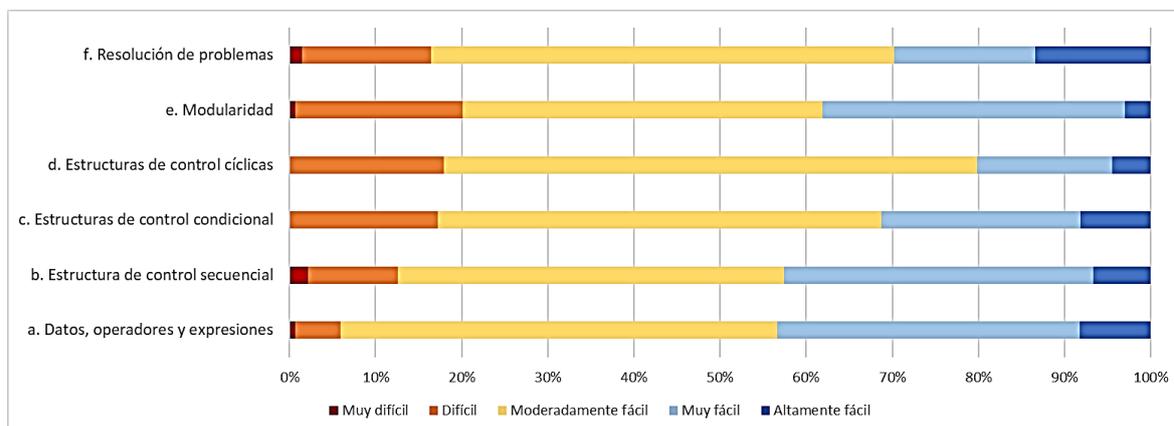
Tabla 16. Dominio de conceptos de Programación

Categoría	Muy difícil		Difícil		Moderadamente fácil		Muy fácil		Altamente fácil		Sum
	Fr	%	Fr	%	Fr	%	Fr	%	Fr	%	
a. Datos, operadores y expresiones	1	0,7%	7	5,2%	68	50,7%	47	35,1%	11	8,2%	134
b. Estructura de control secuencial	3	2,2%	14	10,4%	60	44,8%	48	35,8%	9	6,7%	134
c. Estructuras de control condicional	0	0,0%	23	17,2%	69	51,5%	31	23,1%	11	8,2%	134
d. Estructuras de control cíclicas	0	0,0%	24	17,9%	83	61,9%	21	15,7%	6	4,5%	134

e. Modularidad	1	0,7%	26	19,4%	56	41,8%	47	35,1%	4	3,0%	134
f. Resolución de problemas	1	1,5%	10	14,9%	36	53,7%	11	16,4%	9	13,4%	67

Nota: Elaboración propia a partir de los datos de las tablas 5 a la tabla 15

Gráfico 15. Dominio de conceptos de Programación



Nota: Elaboración propia a partir de la Tabla 15.

Como se puede apreciar en el gráfico 15, aproximadamente el 50% de los estudiantes encuestados consideran moderadamente fácil aplicar cada uno de los conceptos de Programación. Sin embargo, más allá del 17% manifiestan que fue difícil comprender y utilizar las estructuras de control condicionales. Una situación similar ocurre al momento de resolver los problemas de codificación, puesto que 15% de los encuestados afirman que les es complicado alcanzar la habilidad de detectar y corregir errores de codificación. Por otra parte, el 20% de los estudiantes afirman tener dificultades al diseñar o utilizar funciones en sus programas.

A través del análisis de estos resultados (gráfico 15) se puede determinar que, según el criterio de los encuestados, un 15% de ellos afirman tener dificultad en entender y aplicar los conceptos básicos de programación. Sin embargo, el 85% de ellos superó dificultades cognitivas y logró alcanzar habilidades básicas de programación, lo cual hace deducir que el proceso de enseñanza-aprendizaje mejoró el estado cognitivo (tabla 4) del grupo de estudiantes de los tres semestres luego de ingresar a la Carrera de Pedagogía de las Ciencias Experimentales Informática.

Se concluye que el juego educativo debe incorporar actividades específicas para cada temática, poniendo un mayor énfasis en aquellas que presentan mayores dificultades de aprendizaje.

CAPÍTULO V

5. PROPUESTA

5.1. Introducción

En la actualidad, el aprendizaje debe adaptarse a las nuevas realidades tecnológicas y a las necesidades de una generación que interactúa de manera natural con entornos digitales. Los métodos tradicionales de enseñanza, aunque efectivos, muchas veces no logran captar la atención ni el interés de los estudiantes. En este caso, los juegos educativos se presentan como una herramienta poderosa para fomentar el autoaprendizaje.

Esta propuesta tiene como objetivo desarrollar un juego educativo, diseñado para mejorar el rendimiento académico de los estudiantes de la Carrera de Pedagogía de las Ciencias Experimentales Informática, de una manera divertida, interactiva y atractiva para el estudiante de nuevo ingreso. El juego está dirigido a los estudiantes de primer semestre de la Carrera y se desarrolló bajo una metodología que combina el aprendizaje basado en retos, y la gamificación.

A través de esta propuesta, se busca ofrecer una oportunidad a la nivelación de los conocimientos en los estudiantes y mejorar los métodos de enseñanza, proporcionando a los estudiantes un entorno seguro e inspirador donde puedan experimentar, descubrir y aprender mientras juegan. El juego educativo no solo busca mejorar las habilidades académicas de los estudiantes, sino también fomentar el trabajo en equipo, la resolución de problemas y el crecimiento de los estudiantes en la carrera para reducir la deserción estudiantil o en el peor de los casos la reprobación de las materias del área de programación.

5.2. Objetivo General de la Propuesta

- Mejorar el rendimiento académico de los estudiantes de nuevo ingreso de la Carrera de Pedagogía de las Ciencias Experimentales: Informática en el área de Programación.

5.3. Objetivos Específicos

- Diseñar un juego básico de programación dirigido a estudiantes de primer semestre de Informática, con el objetivo de reforzar los conocimientos adquiridos en clase.
- Fomentar el uso del juego educativo entre los estudiantes para potenciar su aprendizaje y participación.

5.4. Desarrollo de la Propuesta

En este trabajo de investigación se aplicó la metodología DICREVOA 2.0, que consta de cinco fases: Análisis, Diseño, Implementación, Evaluación y Publicación. Esta metodología fue utilizada para desarrollar el juego educativo.

El juego educativo se creó utilizando la herramienta de autor EdiLim, elegida por su facilidad de uso y la variedad de actividades que permite diseñar. Esto facilita que los estudiantes aprendan de manera interactiva, potenciando su participación y comprensión.

- **FASE DE ANALISIS**

Como se puede observar en la tabla 17, se presenta información sobre la necesidad del OA, utilizando la matriz de necesidades que plantea (*J. Maldonado et al., 2017*).

Tabla 17 Matriz de necesidades

MATRIZ DE NECESIDADES	
Tema del OA	Fundamentos de programación
Descripción del Objeto de Aprendizaje	El OA presenta información detallada de los temas: modularidad, estructuras de control cíclicas, estructuras de control condicionales, resolución de problemas, estructuras de control secuenciales y datos operadores y expresiones, en los cuales se utilizaron las actividades arrastrar imágenes, completar, escoger, texto.
Nivel	Universidad
Perfil del estudiante	El OA está dirigida a los estudiantes de Primer Semestre de la Universidad

	Nacional de Chimborazo para la Carrera de Pedagogía de las Ciencias Experimentales Informática, para ayudar a una mejor comprensión en el área de programación.
Tiempo estimado para recorrer el Objeto de Aprendizaje	El tiempo requerido para recorrer e interactuar con el OA es de 1 hora.
Contexto educativo	Las asignaturas técnicas del área de programación en muchos casos suelen ser difíciles de entender, puesto que varios docentes utilizan las metodologías de años anteriores que no brindan un conocimiento adecuado, también existen estudiantes que no tienen bases de la programación. Por lo tanto, la creación de los Objetos de Aprendizaje pretende brindar refuerzo académico mediante contenidos y ejercicios básicos.
Tipo de licencia	Software libre
Requerimientos funcionales del Objeto de Aprendizaje	Requerimientos técnicos: Sistema operativo Windows, Navegador Web, opcional conexión a Internet.

- **FASE DE DISEÑO**

En la tabla 18, se detalla los aspectos referentes al diseño instruccional del juego, como el diseño multimedia.

Tabla 18 Plantilla para el Diseño del Objeto de Aprendizaje

PLANTILLA PARA EL DISEÑO DEL OBJETO DE APRENDIZAJE	
Diseño Instruccional	
1	Descripción textual del contenido
	<p>El OA presenta la siguiente información:</p> <ul style="list-style-type: none"> • Se inicia desde teoría básica acerca de cada tema a abordar sobre programación. • Se plantea actividades básicas sobre tipos de datos. • Se plantea actividades basadas en seleccionar, arrastrar y completar entre otros.
2	Objeto de Aprendizaje
	Nivelar los conocimientos académicos en el área de lenguajes de programación en los estudiantes de primer semestre de la Carrera de Pedagogía de la Informática de la Universidad Nacional de Chimborazo.
3	Contenidos
	<p>Inicio</p> <p>Contenidos</p> <ul style="list-style-type: none"> ✚ Datos Operadores y Expresiones. ✚ Estructuras de Control Secuencial. ✚ Estructuras de Control Condicional. ✚ Estructuras de Control Cíclicas. ✚ Modularidad.
4	Actividades
	El OA presenta actividades unir, arrastrar, completar, entre otras que proviene dentro de las herramientas de autor.
Diseño Multimedial	
1	Diseño de la Interfaz
	El OA emplea una interfaz sencilla, amigable e interactiva, tiene un aspecto agradable, en tonalidad de colores, ya que se utilizó colores cálidos para evitar distracción del estudiante, además el color del área de trabajo, la barra de navegación y el título colores similares a los logos de la Carrera de Pedagogía de la Informática.



Ilustración 2: Portada Juego

<p>2</p>	<p>Estructura de las Pantallas</p>
	<p>La estructura de las pantallas del juego se basa en un diseño de bloque de navegación en la parte posterior de lado izquierdo.</p> <div data-bbox="715 1003 975 1391" data-label="Image"> </div>
<p>3</p>	<p>Navegación</p>
	<p>Los contenidos del OA sigue una secuencia de navegación estructurada, que va de acuerdo con los temas.</p>

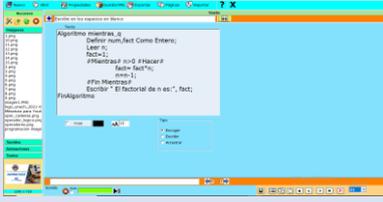
Panel	Descripción
1	
2	Tipos de Datos
3	Tipos de Datos
4	Operadores Aritméticos
5	Operadores lógicos
6	Operaciones de cadenas
7	Seleccione la respuesta correcta
8	Arrastre según corresponda, para cada tipo de datos de programación.
9	Complete el algoritmo que permita ingresar dos números y devuelva el resultado de la suma entre ambos.
10	Complete los espacios en blanco del algoritmo que permitira realizar las cuatro operaciones básicas de dos numeros
11	Estructuras de control secuencial
12	
13	Complete la siguiente condicional mediante la cual me permita mediante calcular si el estudiante pasa o se queda en suspenso
14	
15	Arrastra las partes de la selección anidada
16	Escribe en los espacios en blanco los faltantes para que el algoritmo
17	calcular el promedio de un estudiante que si tiene una nota mayor o igual a 7 imprima el estudiante esta aprobado y si la nota es menor o igual a 5 imprima el estudiante esta reprobado sino el estudiante esta en suspenso
18	Estructuras de control ciclica
19	
20	
21	Escribe en los espacios en blanco
22	
23	Arrastra las palabras reservadas para completar el algoritmo
24	
25	
26	
27	Completa los espacios en blanco para completar el algoritmo de la tabla de multiplicar de un número hasta 12
28	Completa los espacios en blanco para completar el algoritmo que permite obtener

- **FASE DE IMPLEMENTACIÓN**

En esta fase, se detalla el uso de herramientas tecnológicas, junto a la herramienta de autor, los cuales ayudaron a integrar cada componente.

A continuación, en la Tabla 19 se detalla las herramientas que se utilizaron para la creación elaboración del juego.

Tabla 19 Herramientas utilizadas

DESCRIPCION DE LAS HERRAMIENTAS UTILIZADAS			
	Nombre	Descripción	Captura
1	Canva	Canva es una aplicación de edición gratuita que se puede usar computadora (o directamente desde el teléfono móvil) y te permite crear diferentes tipos de arte. (Orozco, 2020).	
2	EdiLim	Edilim es un editor de Libros Interactivos Multimedia (LIM) gratuito que permite crear materiales educativos para la Web.(Equipo TIC del Instituto Nacional de Formación Docente, 2012).	
3	PSeInt	PSeInt es un software libre educativo multiplataforma dirigido a personas que se inician en la programación. (Monroy, 2024).	<pre>1 Algoritmo NumeroMayorMenor 2 Si num>0 Entonces 3 Escribir "Número positivo" 4 SiNo 5 Si num=0 Entonces 6 Escribir "Número es cero" 7 SiNo 8 Escribir "Número negativo" 9 FinSi 10 FinSi 11 FinAlgoritmo</pre>
4	Power Point	Power Point es una hoja que forma una presentación que se puede agregar texto, imágenes, fotografías, gif, entre otros. (Santos, 2022).	
5	Google Sites	Google Sites es una aplicación en línea gratuita, incluida en GSuite para la creación de páginas web. Permite crear un sitio web de una forma tan sencilla como editar un documento (Martínez, 2023)	

- **FASE DE EVALUACIÓN**

A continuación, se presenta la plantilla de evaluación en la que cada criterio se puntúa de 1 a 5, siendo 1 el mínimo y 5 el valor máximo, y N/A si el subcriterio o criterio no es aplicable.

Esta es una evaluación aproximada y fue realizada en consenso con el tutor, más no fue aplicada a los estudiantes del primer semestre debido al hecho de que el juego no fue aplicado en el grupo por la culminación del semestre.

Evaluación de Calidad del Objeto de Aprendizaje							
	Factores Para Evaluar	1	2	3	4	5	N/A
1	Objetivos y coherencia didáctica del objeto de aprendizaje					X	
2	Calidad de los contenidos del objeto de aprendizaje				X		
3	Capacidad de generar reflexión, crítica e innovación				X		
4	Interactividad y adaptabilidad					X	
5	Motivación					X	
6	Formato y diseño					X	
7	Usabilidad					X	
8	Accesibilidad					X	
9	Reusabilidad					X	
10	Interoperabilidad					X	

- **FASE DE PUBLICACIÓN**

En esta fase se realiza la publicación del juego educativo en un Sitio Web, siendo accesible para los creadores del juego.

CAPITULO IV

6. CONCLUSIONES DE RECOMENDACIONES

6.1. Conclusiones

- En conclusión, al analizar los resultados de la encuesta, se identificaron las principales dificultades de los estudiantes de nuevo ingreso en la Carrera. Esto permitió determinar de manera clara los temas clave que fueron abordados en el desarrollo del juego educativo.
- La revisión bibliográfica desarrollada respecto al tema, permitió conocer que un juego educativo puede ayudar a los estudiantes a reforzar sus conocimientos en programación de una manera efectiva a través de la interacción.
- El juego educativo fue desarrollado utilizando la herramienta de autor EdiLim, destacándose por su interactividad y facilidad de uso. Estas características proporcionan una experiencia accesible y dinámica para los usuarios, favoreciendo el desarrollo de la inteligencia y fomentando la curiosidad de los estudiantes para aprender sobre programación.

6.2. Recomendaciones

Después de concluir con este trabajo de investigación se llegaron a las siguientes recomendaciones.

- Se recomienda que la Dirección de la Carrera de Pedagogía de las Ciencias Experimentales Informática sugiera la incorporación y aplicación de talleres de introducción a la programación de computadores en los estudiantes de nivelación, de tal manera que una vez promocionados al primer semestre tengan mejores oportunidades y menos dificultades para abordar temas complejos o problemas inherentes al campo de la programación.
- Se recomienda a los docentes del área de programación considerar la aplicación del juego educativo resultado de este trabajo de investigación, para apoyar procesos de aprendizaje y diagnosticar su eficacia en la adquisición de habilidades cognitivas asociadas con el campo de la programación de computadores.

7. BIBLIOGRAFÍA

- Astudillo, G., Bast, S., & Willging, P. (2016). Enfoque basado en gamificación para el aprendizaje de un lenguaje de programación. *Virtualidad, Educación y Ciencia*, 7(12), Article 12. <https://doi.org/10.60020/1853-6530.v7.n12.14739>
- Casas, J., Repullo, J. R., & Donado, J. (2003). La encuesta como técnica de investigación. Elaboración de cuestionarios y tratamiento estadístico de los datos (I). *Atención Primaria*, 31(8), 527-538.
- Chakray. (2018, diciembre 4). Lenguajes de programación: Tipos, características y diferencias. *Chakray*. <https://www.chakray.com/es/lenguajes-programacion-tipos-caracteristicas/>
- Díaz, K., Fierro, E., & Muñoz, M. (2018). La enseñanza de la programación: Una experiencia en la formación de profesores de informática. *Educación*, 27(53), 73-91. <https://doi.org/10.18800/educacion.201802.005>
- Digital School. (2024, septiembre 23). *Qué es la Gamificación: Características, Beneficios, Ventajas, Desventajas y Herramientas* /. <https://www.itmadrid.com/que-es-la-gamificacion-caracteristicas-beneficios-ventajas-desventajas-y-herramientas/>
- Dtdata. (2023, marzo 24). *Tipos de datos*. <https://www.datdata.com/blog/tipos-de-datos>
- Duoc, B. (2024). *Biblioteca: Investigación Aplicada, Innovación y Transferencia: Definición y propósito de la Investigación Aplicada*. <https://bibliotecas.duoc.cl/investigacion-aplicada/definicion-proposito-investigacion-aplicada>
- Durán, H. (2023). *Desarrollo de un prototipo de juego serio bajo el paradigma de aprendizaje basado en juegos para la enseñanza de los fundamentos de programación*. <https://repository.unab.edu.co/handle/20.500.12749/21264>
- Equipo editorial, E. (2024, mayo 17). *Juego: Características, tipos y beneficios*. <https://humanidades.com/>. <https://humanidades.com/juego/>

- Equipo TIC del Instituto Nacional de Formación Docente. (2012). *EdiLim es un editor de Libros Interactivos Multimedia (LIM) gratuito que permite crear materiales educativos*.
- Gálvez, J. (2018, mayo 8). *Estructuras de Repetición*. Unidades de Apoyo para el Aprendizaje - CUAIEED - UNAM. <https://uapa.cuaieed.unam.mx/node/1100>
- García, F. (2023, septiembre 17). *▷ Que es una SENTENCIA en Programación? Todo lo que Debes Saber*. <https://programacion.top/conceptos/sentencia/>
- Garrido, M. (2023). *Formación basada en las Tecnologías de la Información y Comunicación: Análisis didáctico del proceso de enseñanza-aprendizaje*. Universitat Rovira I Virgili.
- Guevara, S. (2021, mayo 5). *5 Conceptos básicos de la programación*. <https://ed.team/blog/7-conceptos-basicos-de-la-programacion>
- López, M. (2020). *EdiLim: Materiales de aprendizaje con libros interactivos*. INTEF. https://intef.es/observatorio_tecno/edilim-materiales-de-aprendizaje-con-libros-interactivos/
- Maldonado, J., Bermeo, J., & Vélez, F. (2017). *Diseño, creación y evaluación de objetos de aprendizaje: Metodología dicrevoa 2.0* [Universidad de Cuenca,, Cuenca: 2016.]. <http://repositoriointerculturalidad.ec/jspui/handle/123456789/36575>
- Maldonado, R. (2024, septiembre 28). *Estructuras de control: ¿qué son y qué tipos existen?* <https://keepcoding.io/blog/que-son-estructuras-de-control-en-programacion/>
- Martínez, M. (2023). *Google Sites como herramienta de portfolio educativo*. INTEF. https://intef.es/observatorio_tecno/google-sites-como-herramienta-de-portfolio-educativo/
- Monroy, F. (2024). *¿Qué es PSeInt? – Fernando Monroy*. <https://fernandomonroytenorio.com/tema/que-es-pseint/>
- Montero, B. (2017). Aplicación de juegos didácticos como metodología de enseñanza: Una Revisión de la Literatura. *Pensamiento Matemático*, 7(1), 75-92.

- Orozco, L. M. (2020, septiembre 16). ¿Qué es Canva y cómo nos puede ayudar a crear contenido para redes sociales? *BuzzMonitor*. <https://getbuzzmonitor.com/es/que-es-canva-y-como-nos-puede-ayudar-a-crear-contenido-para-redes-sociales/>
- Pearson, I. (2022, marzo 2). *Proceso de enseñanza-aprendizaje: Cambios y tendencias actuales*. Defensa.com. <https://blog.pearsonlatam.com/educacion-del-futuro/tendencias-actuales-del-proceso-de-ensenanza-aprendizaje>
- Pinchao, J. (2021, septiembre 10). *La escuela, ¿debe enseñar a programar?* El Comercio. <https://www.elcomercio.com/cartas/escuela-formacion-conocimientos-informatica-computacion.html>
- Putton, G. (2021). La importancia del juego en el proceso de aprendizaje de la enseñanza en la educación infantil. *Revista Científica Multidisciplinar Núcleo do Conhecimento*, 11(05), 114-125.
- Ramírez, A. (2022, julio 28). Algoritmo en Informática: Qué es y qué tipos hay - Epitech España. *Epitech Spain*. <https://www.epitech-it.es/algoritmo-diferentes-tipos/>
- Rasig, E. (2024). *Curso de Fundamentos de la Programación—UTN BA*. <https://julia.redtecnologica.orghttps://sceu.frba.utn.edu.ar/e-learning>
- Rodríguez, M. (2020, mayo 5). *EdiLim: Materiales de aprendizaje con libros interactivos*. INTEF. https://intef.es/observatorio_tecno/edilim-materiales-de-aprendizaje-con-libros-interactivos/
- Rodríguez, R. (2006). La informática educativa en el contexto actual. *EduTec. Revista Electrónica de Tecnología Educativa*, 13. <https://doi.org/10.21556/edutec.2000.13.553>
- Santos, D. (2022). *Guía sobre PowerPoint: Qué es, características y preguntas frecuentes*. <https://blog.hubspot.es/marketing/que-es-powerpoint>
- Sulbarán, I. (2023, agosto 28). *¿Qué es la programación, sus lenguajes y tipos de enfoque?* Tiffin University. <https://global.tiffin.edu/blog/que-es-programacion>
- Torres, M. (2022, junio 9). *¿Qué es la gamificación? 10 formas para llevar esta técnica a tu clase*. <https://conecta.tec.mx/es/noticias/nacional/educacion/que-es-gamificacion>

Zatarain, R. (2018). Reconocimiento afectivo y gamificación aplicados al aprendizaje de Lógica algorítmica y programación. *REDIE: Revista Electrónica de Investigación Educativa*, 20(3), 115-125.

ANEXOS

Anexo 1. Captura de los contenidos

UNIVERSIDAD NACIONAL DE CHIMBORAZO
Lo difícil hoy será fácil mañana.

Programación

Portada
Tipos de Datos
Estructuras de Control Secuencial
Estructuras de Control Cíclicas
Modularización

UNIVERSIDAD NACIONAL DE CHIMBORAZO
DE PRINCIPIANTE A PROGRAMADOR

Autor: Maribel Sefla

Activar Windows
Vé a Configuración para activar Windows.

TIPOS DE DATOS

LOS TIPOS DE DATOS BASICOS SON:

NUMERICOS <ul style="list-style-type: none">• Integer: Número sin decimales.• Float: Número con punto flotante o punto decimal.• Double: Decimales con valores más precisos.	BOOLEANOS <ul style="list-style-type: none">• Bool: Verdadero o falso.
CADENAS DE CARACTERES <ul style="list-style-type: none">• Chart: Un solo una letra o un símbolo.• String: Una cadena de caracteres.	SIN VALOR <ul style="list-style-type: none">• Null: No hay valor.• Undefined: Hay una variable que no tiene ningún valor.

(Mathieu, 2014)

Activar Windows
Vé a Configuración para activar Windows.

UNIVERSIDAD NACIONAL DE CHIMBORAZO **Lo difícil hoy será fácil mañana.**

Programación

Seleccione la respuesta correcta

Define el tipo de dato para la siguiente información:

¿Qué operador me permite obtener el residuo de una división?

1	AND	2	MOD
3	OR	4	NOT

Activar Windows
Ve a Configuración para activar Windows.

UNIVERSIDAD NACIONAL DE CHIMBORAZO **Lo difícil hoy será fácil mañana.**

Programación

Arrastre y asocie cada elemento con el tipo de dato de programación correspondiente.

Números enteros	int
Representan texto	double
Números con decimal	char
Caracteres	string

Activar Windows
Ve a Configuración para activar Windows.

UNIVERSIDAD NACIONAL DE CHIMBORAZO **Lo difícil hoy será fácil mañana.**

Programación

Completa la condicional para saber si el estudiante aprueba o queda en suspenso según su calificación.

Imprimir

Leer

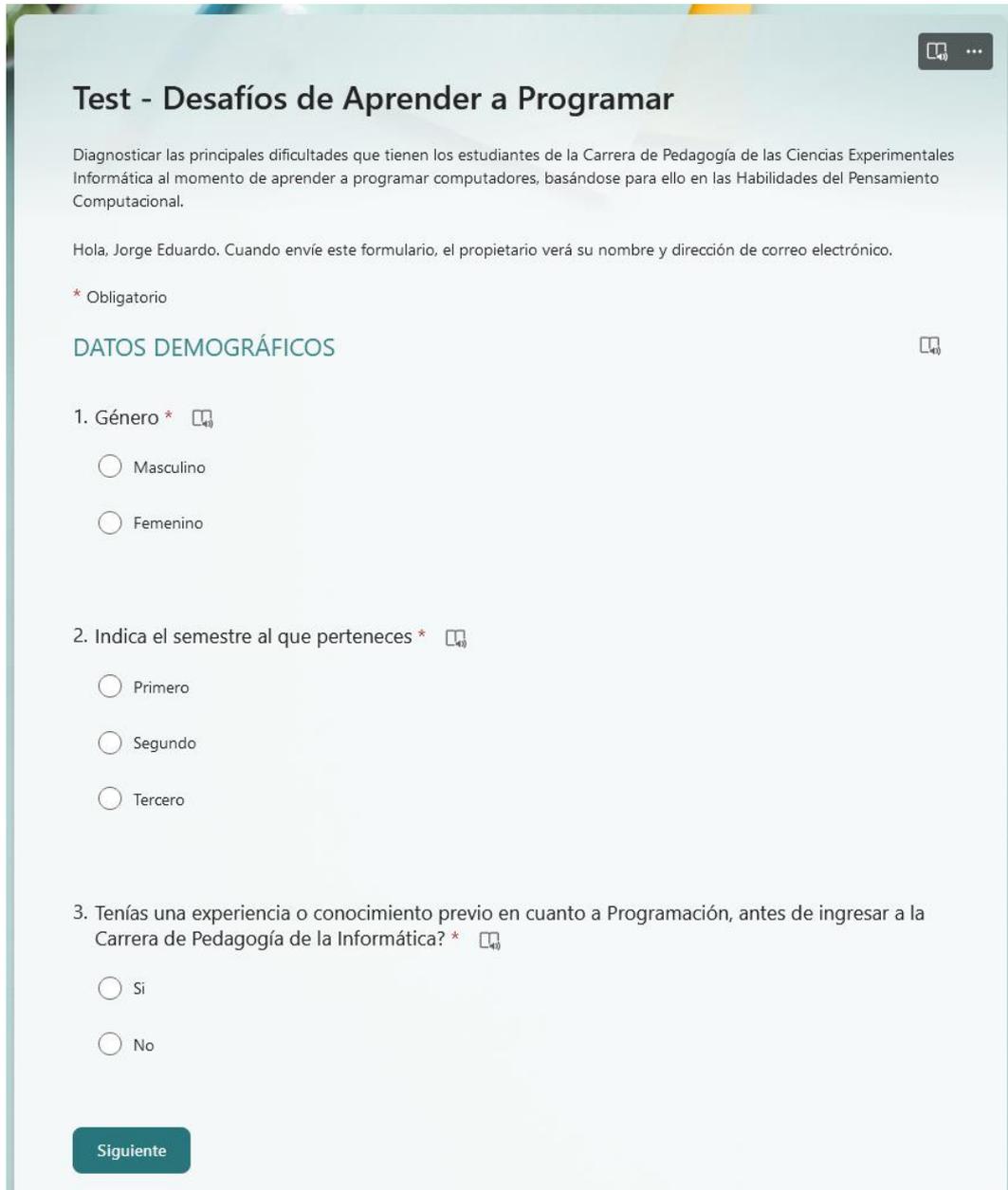
Si Entonces

Imprimir

Sino

Activar Windows
Ve a Configuración para activar Windows.

Anexo 2. Captura de pantalla de la encuesta aplicada a los estudiantes de primer, segundo y tercer semestre de la Carrera



The screenshot shows a survey interface with a title, a description, a greeting, and three questions. The questions are about gender, semester, and prior programming experience. A 'Siguiete' button is at the bottom.

Test - Desafíos de Aprender a Programar

Diagnosticar las principales dificultades que tienen los estudiantes de la Carrera de Pedagogía de las Ciencias Experimentales Informática al momento de aprender a programar computadores, basándose para ello en las Habilidades del Pensamiento Computacional.

Hola, Jorge Eduardo. Cuando envíe este formulario, el propietario verá su nombre y dirección de correo electrónico.

* Obligatorio

DATOS DEMOGRÁFICOS

1. Género *

Masculino

Femenino

2. Indica el semestre al que perteneces *

Primero

Segundo

Tercero

3. Tenías una experiencia o conocimiento previo en cuanto a Programación, antes de ingresar a la Carrera de Pedagogía de la Informática? *

Si

No

Siguiete

Test - Desafíos de Aprender a Programar

* Obligatorio

APRENDIZAJE DE LOS CONTENIDOS

4. ¿Cómo calificarías tu comprensión de los tipos de datos y su uso en la programación? *

* 

- Muy baja
- Baja
- Media
- Alta
- Muy alta

5. Al trabajar con operadores y expresiones, ¿qué tan cómodo/a te sientes realizando operaciones complejas y combinando diferentes tipos de datos? *

* 

- Muy incómodo/a
- Incómodo/a
- Neutral
- Cómodo/a
- Muy cómodo/a

6. ¿Cómo evaluarías tu habilidad para escribir programas que sigan una secuencia lógica de instrucciones? *

* 

- Muy deficiente
- Deficiente
- Neutral
- Competente
- Muy competente

7. Al enfrentarte a problemas que requieren el uso de procesos secuenciales, ¿qué tan eficiente puedes ser al planificar y organizar tu código? *

* 

- Muy ineficiente
- Ineficiente
- Moderadamente eficiente
- Muy eficiente
- Altamente eficiente

8. ¿Cuál es tu nivel de confianza al utilizar estructuras condicionales (if ... else) para controlar el flujo de ejecución en un programa? *

- Muy bajo
- Bajo
- Medio
- Alto
- Muy Alto

9. Al enfrentarte a problemas que involucran decisiones condicionales, ¿qué tan hábil eres para diseñar y aplicar estructuras de control condicional de manera efectiva? *

- Nada hábil
- Poco hábil
- Moderadamente hábil
- Muy hábil
- Altamente hábil

10. ¿Cómo evaluarías tu capacidad para implementar bucles (for, while) en programas (o soluciones) para repetir acciones de manera controlada? *

- Muy baja
- Baja
- Media
- Alta
- Muy alta

11. En proyectos que requieren iteraciones, ¿qué tan hábil puedes ser para diseñar bucles que optimicen el rendimiento y reduzcan la complejidad de la solución? *

- Nada hábil
- Poco hábil
- Moderadamente hábil
- Muy hábil
- Altamente hábil

12. ¿Cómo evaluarías tu comprensión y la aplicación de funciones en la programación de soluciones informáticas? * 

- Muy deficiente
- Deficiente
- Poco competente
- Competente
- Muy competente

13. Al diseñar programas, ¿qué tan hábil eres para modularizar el código creando funciones que permitan mejorar la legibilidad y la reutilización del código? * 

- Nada hábil
- Poco hábil
- Moderadamente hábil
- Muy hábil
- Altamente hábil

Atrás

Siguiente

Test - Desafíos de Aprender a Programar

* Obligatorio

PRINCIPALES DESAFÍOS DE APRENDER A PROGRAMAR

14. Por cada uno de los desafíos selecciona el nivel de dificultad al momento de superarlo. *

Todos los indicadores que abajo se lista tienen relación directa con la programación.

	Altamente fácil	Muy fácil	Moderadamente fácil	Difícil	Muy difícil
Comprender el uso de variables, tipos de datos, operadores y estructuras de control	<input type="radio"/>				
Comprender la sintaxis de un Lenguaje de Programación	<input type="radio"/>				
Diseñar un algoritmo aplicando el pensamiento lógico	<input type="radio"/>				
Detectar errores en el código y corregirlos	<input type="radio"/>				
Abstraer un problema y modularizar la solución	<input type="radio"/>				
Gestionar el tiempo de las tareas y proyectos de todas las asignaturas	<input type="radio"/>				
Ser autodidacta y mantenerse actualizado entorno al uso de las TIC	<input type="radio"/>				
Ser constante y paciente para superar problemas al programar	<input type="radio"/>				
Colaborar y trabajar en equipo	<input type="radio"/>				
Enfrentar el desarrollo de nuevos proyectos	<input type="radio"/>				
Acceder a recursos TIC (Computador, internet, etc.)	<input type="radio"/>				