



**UNIVERSIDAD BOLIVARIANA DE ECUADOR**

**MAESTRÍA EN EDUCACIÓN**

**TRABAJO DE TITULACIÓN**

**PREVIO A LA OBTENCIÓN DEL TÍTULO DE  
MAGÍSTER EN EDUCACIÓN, MENCIÓN PEDAGOGÍA DE LOS ENTORNOS  
DIGITALES**

**TEMA**

**“Estrategia didáctica para la integración de las tecnologías digitales en el proceso de enseñanza-aprendizaje de la asignatura Programación en la especialidad de Informática”**

**Autor/es:**

**Gavilánez Soria Pablo Xavier  
Oña Chanaluisa Juan Carlos**

**Tutor/a:**

**PhD Janette Santos Baranda**

**ECUADOR**

**2024**



UNIVERSIDAD  
BOLIVARIANA  
DEL ECUADOR

## TRABAJO DE TITULACION

### DEDICATORIA

Dedicado para mi querido y amado padre que en paz descansa Viche como le decían cariñosamente, por siempre animarme a superarme incluso en los momentos más difíciles de mi vida te amo papito bello algún día nos reuniremos.

**Pablo Xavier Gavilánez Soria**

A mi madrecita dedico esta tesis por siempre guiarme primero por el camino de Dios y luego en alcanzar mis metas de estudio, gracias madre bella.

**Juan Carlos Oña Chanaluisa**



UNIVERSIDAD  
BOLIVARIANA  
DEL ECUADOR

## TRABAJO DE TITULACION

### AGRADECIMIENTO

Un agradecimiento especial para la Universidad que me dio la oportunidad de alcanzar esta meta, a todos los catedráticos y tutor de tesis por guiarme en tan arduo camino del conocimiento.

**Pablo Xavier Gavilánez Soria**

Doy un agradecimiento especial a todos los compañeros y amigos que pusieron un granito para ayudarme en alcanzar esta meta, y en especial a la UBE por permitirme realizar este sueño de alcanzar el Posgrado.

**Juan Carlos Oña Chanaluisa**



## RESUMEN

Las tecnologías digitales están influyendo significativamente en la educación por cuanto su uso cada día es mayor en los planteles educativos. Estas tecnologías favorecen la transformación del proceso de enseñanza-aprendizaje en la especialidad de Informática al posibilitar un enfoque dinámico, participativo y colaborativo en el aula. El empleo de herramientas y recursos informáticos que facilitan la comprensión conceptual, suscitan la colaboración y la creatividad de los estudiantes para afrontar con eficiencia los desafíos de la nueva era digital. El objetivo es plantear estrategia didáctica para integración de las tecnologías digitales en el proceso de enseñanza- aprendizaje de la asignatura Programación en la especialidad de Informática de segundo año de Bachillerato Técnico en la Unidad Educativa Shushufindi. El problema científico es: ¿cómo concebir la integración de las tecnologías digitales en el proceso de enseñanza-aprendizaje de la asignatura Programación en la especialidad Informática? Metodología: El enfoque de la investigación es cuantitativo, el tipo de investigación empleado es descriptiva por cuanto describe: las características, fenómenos o situaciones actuales y es No Experimental de corte Transversal, por cuanto se observan a diferentes sujetos en un único estudio en el tiempo para examinar la prevalencia de ciertas características, actitudes y comportamientos con respecto al objeto de estudio. Los resultados posibilitan confirmar la necesidad de la integración de estas tecnologías digitales en el proceso de enseñanza-aprendizaje de la Programación, para el desarrollo de las habilidades y conocimientos, mientras que el Aprendizaje Experiencial ofrece oportunidades prácticas para aplicar y contextualizar los conceptos teóricos, promoviendo un aprendizaje significativo y relevante.

**Palabras clave:** Informática, Programación Informática, Proceso de aprendizaje, Informática Educativa



## ABSTRACT

Digital technologies are significantly influencing education because their use is increasing every day in educational establishments. These technologies favor the transformation of the teaching-learning process in the Computer Science specialty by enabling a dynamic, participatory and collaborative approach in the classroom. The use of computer tools and resources that facilitate conceptual understanding, encourage collaboration and creativity among students to efficiently face the challenges of the new digital age. The objective is to propose a didactic strategy for the integration of digital technologies in the teaching-learning process of the subject Programming in the Computer Science specialty of the second year of Technical Baccalaureate at the Shushufindi Educational Unit. The scientific problem is: how to conceive the integration of digital technologies in the teaching-learning process of the subject Programming in the Computer Science specialty? Methodology: The research approach is quantitative, the type of research used is descriptive in that it describes: the characteristics, phenomena or current situations and is Non-Experimental Cross-sectional, in that different subjects are observed in a single study over time. to examine the prevalence of certain characteristics, attitudes and behaviors with respect to the object of study. The results make it possible to confirm the need to integrate these digital technologies in the teaching-learning process of Programming, for the development of skills and knowledge, while Experiential Learning offers practical opportunities to apply and contextualize theoretical concepts, promoting meaningful and relevant learning.

**Keywords:** Computer Science, Computer Programming, Learning Process, Educational Informatics



## ÍNDICE GENERAL

PORTADA.....	i
FICHA SENESCYT PARA EL REPOSITORIO.....	ii
COPIA INFORME DE SIMILITUD (ANTIPLAGIO). ....	iv
CERTIFICACIÓN DE AUTORÍA Y CESIÓN DE DERECHOS DEL AUTOR (ES).....	v
AVAL DEL TUTOR DE LA TESIS .....	vi
DEDICATORIA.....	vii
AGRADECIMIENTO .....	viii
RESUMEN.....	ix
ABSTRACT.....	x
ÍNDICE GENERAL .....	xi
ÍNDICE DE TABLAS.....	xiv
ÍNDICE DE FIGURAS.....	xvi
INTRODUCCIÓN .....	1
CAPÍTULO 1: MARCO TEÓRICO.....	9
1.1.    Antecedentes investigativos .....	9
1.2.    Fundamentación teórica o bases teóricas.....	11
1.2.1. Teoría del Constructivismo .....	13
1.2.2. El aprendizaje basado en problemas.....	14
1.2.3. Teoría del aprendizaje experiencial .....	15
1.2.4. Teoría del Procesamiento de la Información.....	17
1.2.5. Las tecnologías de la información y las comunicaciones .....	18
1.2.6. Teoría de la Computación .....	20
1.2.7. La Teoría del Pensamiento Complejo.....	20



1.2.8. Teoría de Sistemas.....	21
1.2.9. Bases normativas y legales.....	23
<b>CAPÍTULO 2: METODOLOGÍA PARA EL DESARROLLO DE LA INVESTIGACIÓN Y ESTUDIO DIAGNÓSTICO .....</b>	<b>25</b>
2.1. Enfoque de investigación .....	25
2.2. Alcance de la investigación .....	25
2.3. Declaración y justificación del tipo de investigación.....	26
2.4. Delimitación de la población y la muestra .....	27
2.5. Proceder metodológico general seguido durante el proceso de investigación.....	28
2.6. Análisis de resultados.....	30
2.6.1. Análisis de la encuesta aplicada a los estudiantes .....	30
2.6.2. Análisis de la encuesta aplicada a los docentes.....	35
2.6.3. Caracterización del proceso de integración de las tecnologías digitales en el proceso de enseñanza-aprendizaje de la asignatura Programación en la especialidad Informática .....	38
<b>CAPÍTULO 3: PRESENTACIÓN Y VALIDACIÓN DE LA PROPUESTA.....</b>	<b>42</b>
3.1. Presentación.....	42
3.2. Objetivos generales y específicos .....	42
3.3. Fundamentación (marco teórico).....	43
3.4. Características (Caracterización de la propuesta).....	44
3.4.1. Tecnologías digitales .....	44
3.4.2. Aplicaciones online y offline .....	45
3.5. Estructura y dinámica de sus componentes (según el tipo de propuesta).....	47



<b>3.6. Demostración de las prácticas de actividades de programación en el laboratorio de Informática .....</b>	<b>52</b>
<b>3.7. Validación de la propuesta .....</b>	<b>67</b>
<b>CONCLUSIONES.....</b>	<b>69</b>
<b>RECOMENDACIONES .....</b>	<b>71</b>
<b>REFERENCIAS BIBLIOGRÁFICAS.....</b>	<b>72</b>
<b>ANEXOS.....</b>	<b>76</b>
<b>Anexo 1: Encuesta Docentes.....</b>	<b>76</b>
<b>Anexo 2: Encuesta Estudiantes.....</b>	<b>81</b>
<b>Anexo 3: Tabulaciones encuestas estudiantes.....</b>	<b>85</b>
<b>Anexo 4: Tabulaciones encuestas docentes .....</b>	<b>96</b>
<b>Anexo 5: Matriz rúbrica analíticas evaluación de expertos .....</b>	<b>106</b>
<b>Anexo 6: Rúbricas analíticas de evaluaciones de expertos.....</b>	<b>107</b>



## ÍNDICE DE TABLAS

<b>Tabla 1.</b> Operacionalización de la variable objeto de investigación .....	29
<b>Tabla 2.</b> Unidades de trabajo.....	46
<b>Tabla 3.</b> Etapa 1. Diagnóstico del nivel de conocimientos y habilidades de los estudiantes .....	49
<b>Tabla 4.</b> Etapa 2. Planificación del sistema de clases con sus respectivos componentes didácticos: objetivos, contenidos, métodos, medios digitales, formas de organización y evaluación .....	49
<b>Tabla 5.</b> Etapa 3.- Ejecución de las actividades planificadas .....	50
<b>Tabla 6.</b> Etapa 4.- Evaluación del nivel de conocimientos y habilidades .....	50
<b>Tabla 7.</b> Etapa 5.- Implementación en proyectos prácticos.....	51
<b>Tabla 8.</b> Matriz de evaluaciones de los expertos.....	68
<b>Tabla 9.</b> Pregunta 1.-¿En qué medida la institución te facilita el acceso a recursos tecnológicos para el aprendizaje de la programación?.....	85
<b>Tabla 10.</b> Pregunta 2.-¿En qué medida la institución te informa sobre las políticas relacionadas con el uso de tecnologías digitales en la asignatura de programación? .....	86
<b>Tabla 11.</b> Pregunta 3.-¿En qué medida los docentes de la asignatura de programación te están capacitando para utilizar tecnologías digitales en el proceso de enseñanza-aprendizaje?.....	86
<b>Tabla 12.</b> Pregunta 4.- ¿En qué medida la institución evalúa y realiza seguimiento al uso de las tecnologías digitales que aplicas en la asignatura de programación? .....	87
<b>Tabla 13.</b> Pregunta 5.-¿En qué medida las actividades que realizas con tecnologías digitales en la clase de programación te ayudan a alcanzar los objetivos de aprendizaje? .....	88
<b>Tabla 14.</b> Pregunta 6.- ¿En qué medida las tecnologías digitales que se utilizan en la clase de programación te permiten interactuar y colaborar con tus compañeros y docente? .....	89
<b>Tabla 15.</b> Pregunta 7.- ¿En qué medida las tecnologías digitales que se utilizan en la clase de programación te permiten personalizar tu aprendizaje?.....	90
<b>Tabla 16.</b> Pregunta 8.-¿En qué medida las tecnologías digitales que se utilizan en la clase de programación te ayudan a desarrollar competencias transversales?.....	91



<b>Tabla 17.</b> Pregunta 9.-¿En qué medida tienes acceso a las tecnologías digitales que se utilizan en la clase de programación?.....	92
<b>Tabla 18.</b> Pregunta 10.-¿En qué medida las tecnologías digitales se integran de forma efectiva en el proceso de enseñanza-aprendizaje de la programación?.....	93
<b>Tabla 19.</b> Pregunta 11.-¿En qué medida se le garantiza la seguridad y privacidad al utilizar las tecnologías digitales en la clase de Programación? .....	94
<b>Tabla 20.</b> Pregunta 1.- ¿Cómo evaluaría el nivel de aprendizaje de programación de sus estudiantes en general? .....	96
<b>Tabla 21.</b> Pregunta 2.-¿En qué medida cree que la mayoría de sus estudiantes demuestra habilidades sólidas en la resolución de problemas a través de la programación?.....	97
<b>Tabla 22.</b> Pregunta 3.- ¿Considera que la aplicación de proyectos prácticos en sus clases ha influido en el nivel de aprendizaje de programación de sus estudiantes?.....	98
<b>Tabla 23.</b> Pregunta 4.-En su experiencia, ¿qué tan efectivas son las estrategias de enseñanza que utilizan para mejorar el nivel de programación de sus estudiantes?.....	99
<b>Tabla 24.</b> Pregunta 5.- ¿Cómo calificaría la participación activa de sus estudiantes durante las clases de Programación?.....	100
<b>Tabla 25.</b> Pregunta 6.- ¿Qué tanto cree que el uso de tecnologías digitales ha mejorado la experiencia de aprendizaje de programación para sus estudiantes? .....	101
<b>Tabla 26.</b> Pregunta 7.-¿En qué medida emplea tecnologías digitales para facilitar la práctica y aplicación de conceptos de programación en sus clases? .....	102
<b>Tabla 27.</b> Pregunta 8.-¿Considera que la carencia de recursos tecnológicos afecta el rendimiento y la participación de los estudiantes en la programación? .....	103
<b>Tabla 28.</b> Pregunta 9.-¿En su opinión, cómo afecta el uso de tecnologías digitales al interés y la motivación de los estudiantes por la programación? .....	104
<b>Tabla 29.</b> Pregunta 10.-¿Qué tan preparado/a se siente para integrar de manera efectiva las tecnologías digitales en su enseñanza de programación? .....	105



## ÍNDICE DE FIGURAS

<b>Figura 1.</b> Árbol de problemas .....	3
<b>Figura 2.</b> Estructura de las relaciones entre las etapas de la estrategia didáctica.....	48
<b>Figura 3.</b> Aplicación utilizada dev-c++.....	53
<b>Figura 4.</b> Aplicación utilizada PSEINT .....	56
<b>Figura 5.</b> Aplicación PSEINT .....	57
<b>Figura 6.</b> Aplicación utilizada SLE2.....	60
<b>Figura 7.</b> Aplicación utilizada ONLINE C .....	63
<b>Figura 8.</b> Aplicación utilizada SCARCH.....	66
<b>Figura 9.</b> Aplicación utilizada Quizziz .....	66
<b>Figura 10.</b> <i>Calificaciones docentes expertos</i> .....	68
<b>Figura 11.</b> Pregunta 1.-¿En qué medida la institución te facilita el acceso a recursos tecnológicos para el aprendizaje de la programación?.....	85
<b>Figura 12.</b> Pregunta 2.-¿En qué medida la institución te informa sobre las políticas relacionadas con el uso de tecnologías digitales en la asignatura de programación? .....	86
<b>Figura 13.</b> Pregunta 3.- ¿En qué medida los docentes de la asignatura de programación te están capacitando para utilizar tecnologías digitales en el proceso de enseñanza-aprendizaje?.....	87
<b>Figura 14.</b> Pregunta 4.-¿En qué medida la institución evalúa y realiza seguimiento al uso de las tecnologías digitales que aplicas en la asignatura de programación? .....	88
<b>Figura 15.</b> <i>Pregunta 5.-¿En qué medida las actividades que realizas con tecnologías digitales en la clase de programación te ayudan a alcanzar los objetivos de aprendizaje?.....</i>	89
<b>Figura 16.</b> Pregunta 6.- ¿En qué medida las tecnologías digitales que se utilizan en la clase de programación te permiten interactuar y colaborar con tus compañeros y docente? .....	90
<b>Figura 17.</b> Pregunta 7.-¿En qué medida las tecnologías digitales que se utilizan en la clase de programación te permiten personalizar tu aprendizaje?.....	91
<b>Figura 18.</b> Pregunta 8.- ¿En qué medida las tecnologías digitales que se utilizan en la clase de programación te ayudan a desarrollar competencias transversales?.....	92

<b>Figura 19.</b> Pregunta 9.- ¿En qué medida tienes acceso a las tecnologías digitales que se utilizan en la clase de programación? .....	93
<b>Figura 20.</b> Pregunta 10.-¿En qué medida las tecnologías digitales se integran de forma efectiva en el proceso de enseñanza-aprendizaje de la programación? .....	94
<b>Figura 21.</b> Pregunta 11.-¿En qué medida se le garantiza la seguridad y privacidad al utilizar las tecnologías digitales en la clase de Programación? .....	95
<b>Figura 22.</b> Pregunta 1.- ¿Cómo evaluaría el nivel de aprendizaje de programación de sus estudiantes en general? .....	96
<b>Figura 23.</b> Pregunta 2.-¿En qué medida cree que la mayoría de sus estudiantes demuestra habilidades sólidas en la resolución de problemas a través de la programación?.....	97
<b>Figura 24.</b> Pregunta 3.-¿Considera que la aplicación de proyectos prácticos en sus clases ha influido en el nivel de aprendizaje de programación de sus estudiantes?.....	98
<b>Figura 25.</b> Pregunta 4.-En su experiencia, ¿qué tan efectivas son las estrategias de enseñanza que utilizan para mejorar el nivel de programación de sus estudiantes? .....	99
<b>Figura 26.</b> Pregunta 5.- ¿Cómo calificaría la participación activa de sus estudiantes durante las clases de Programación?.....	100
<b>Figura 27.</b> Pregunta 6.-¿Qué tanto cree que el uso de tecnologías digitales ha mejorado la experiencia de aprendizaje de programación para sus estudiantes? .....	101
<b>Figura 28.</b> Pregunta 7.-¿En qué medida emplea tecnologías digitales para facilitar la práctica y aplicación de conceptos de programación en sus clases? .....	102
<b>Figura 29.</b> Pregunta 8.-¿Considera que la disponibilidad de recursos tecnológicos afecta el rendimiento y la participación de los estudiantes en la programación?.....	103
<b>Figura 30.</b> Pregunta 9.-¿En su opinión, cómo afecta el uso de tecnologías digitales al interés y la motivación de los estudiantes por la programación? .....	104
<b>Figura 31.</b> Pregunta 10.-¿Qué tan preparado/a se siente para integrar de manera efectiva las tecnologías digitales en su enseñanza de programación? .....	105



## INTRODUCCIÓN

El avance tecnológico está influyendo significativamente en la educación, ya que cada vez es mayor el empleo de las tecnologías digitales en el proceso de enseñanza-aprendizaje. Estas tecnologías en la actualidad permiten desarrollar nuevos procedimientos para aprender. Asimismo, durante su evolución ellas han cambiado el mundo de forma irreversible, dando lugar a nuevas prácticas culturales y diferentes formas para conectarse, comunicarse y trabajar de manera colaborativa. Sin embargo, también han generado nuevos problemas, como la desigualdad social, las brechas culturales y tecnológicas, así como las desigualdades económicas.

Al respecto Dorado (2021) describe como aún son insuficientes los estudios e investigaciones acerca de la influencia y las metodologías a utilizar con vistas a incorporar de manera armónica estas tecnologías en las clases y cómo a pesar de los grandes avances tecnológicos, los maestros en muchas ocasiones continúan sus prácticas docentes de manera tradicional utilizando recursos básicos como son la pizarra, los libros, los afiches, mapamundis, entre otras.

En el caso de la asignatura Programación estas tecnologías digitales favorecen la transformación del proceso de enseñanza-aprendizaje en la especialidad de Informática al posibilitar un enfoque dinámico, participativo y colaborativo en el aula mediante la utilización de herramientas interactivas y recursos multimedia que facilitan la comprensión de los conceptos y la formación de habilidades prácticas. El uso de Internet y las distintas plataformas educativas en línea ha incrementado la mejora en la materia programación.

Aunque la programación sigue siendo un componente fundamental para la formación de los profesionales de la informática, su enfoque en la educación está evolucionando hacia un entendimiento más amplio de cómo las tecnologías pueden transformar la comprensión de los estudiantes acerca del mundo.

El uso de software especializado y entornos de desarrollo integrados (IDE, por sus siglas en inglés) proporciona a los estudiantes herramientas poderosas para escribir, depurar y ejecutar códigos de manera eficiente. Estos programas suelen incluir características como resaltado de sintaxis,



autocompletado de código y herramientas de depuración que facilitan el proceso de aprendizaje y mejora la productividad de los estudiantes.

En consonancia con las premisas antes descritas, la utilización de plataformas de colaboración en línea y herramientas de control de versiones, como Git, que es un programa para controlar las versiones, permitiéndoles compartir y revisar los códigos de manera eficiente, así como gestionar proyectos de desarrollo de software de forma colaborativa. El empleo de las NTICs en la enseñanza de la programación ofrece nuevas oportunidades para la experimentación y el aprendizaje práctico, los estudiantes pueden utilizar entornos de programación basados en inteligencia artificial para crear modelos predictivos o sistemas de recomendación, o pueden explorar conceptos de programación en entornos de realidad virtual que simulan situaciones del mundo real.

Por tanto, las tecnologías digitales han revolucionado el proceso de enseñanza-aprendizaje de la Programación en la especialidad de Informática, al propiciar el empleo de herramientas y recursos informáticos que facilitan la comprensión conceptual, suscitan la colaboración y la creatividad de los estudiantes para afrontar con eficiencia los desafíos de la nueva era digital.

En tal sentido, la educación ecuatoriana no es ajena a estos cambios. En este contexto, se han logrado avances y logros importantes, pero también se han evidenciado dificultades y falencias. (Mendoza Bozada, 2020). Según el (Ministerio de Educación del Ecuador, 2022) la especialidad de Informática ha experimentado un gran avance en las escuelas del Ecuador, lo que ha permitido destacar la importancia de estas en el desarrollo de la alfabetización de digital.

Al respecto la (Universidad Central del Ecuador, 2021) referencia desde los estudios realizados que los estudiantes que reciben formación en informática desde temprana edad desarrollan habilidades de resolver problemas y la creatividad, lo que facilita su preparación para enfrentar nuevos desafíos.

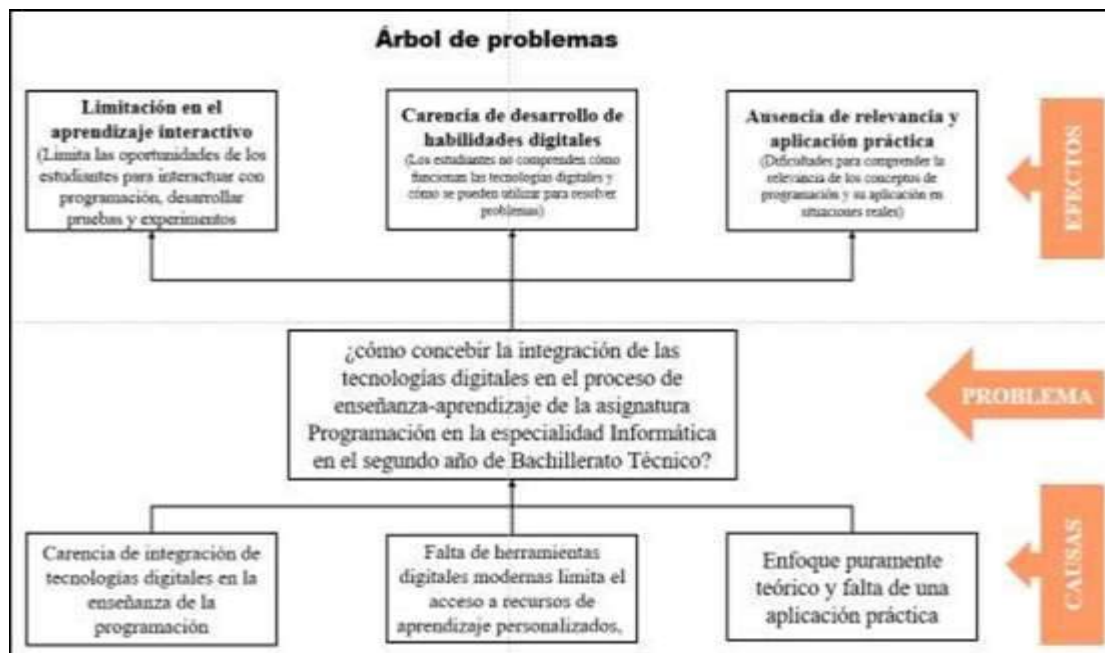
Para determinar la problemática de esta investigación, se empleó la técnica del árbol de problemas, según Hernández y Garnica (2015) que “consiste en desarrollar ideas creativas para identificar las posibles causas del conflicto, generando de forma organizada un modelo que explique las razones y consecuencias del problema” (p. 40). Además, permite identificar las causas, el problema principal y los efectos (figura 1 árbol de problemas). Asimismo, se tomó en cuenta el estudio exploratorio inicial y el trabajo por más de diez años de los autores vinculados a la enseñanza de la asignatura de Programación, lo que permitió establecer las siguientes manifestaciones en cuanto



a la integración de tecnologías digitales en esta asignatura en la especialidad Informática para el segundo año de Bachillerato Técnico en la Unidad Educativa Shushufindi:

- La falta de integración de tecnologías digitales en la enseñanza de la Programación provoca menor motivación y compromiso entre los estudiantes, ya que los métodos tradicionales de enseñanza provocan desinterés.
- La falta de herramientas digitales modernas limita el acceso a recursos de aprendizaje personalizados, lo que dificulta que los estudiantes progresen a su propio ritmo y se concentren en contenidos específicos de la asignatura que les resultan difíciles.
- El enfoque puramente teórico y falta de una aplicación práctica, en función a la falta de integración de las tecnologías digitales en la asignatura de Programación conduce a un enfoque teórico centrado en la transferencia de conocimientos y conceptos abstractos, sin aplicación práctica, limitando la capacidad de los estudiantes para desarrollar habilidades prácticas y aplicar sus conocimientos a proyectos reales.
- Existe resistencia al cambio, así como carencias en capacitación a docentes en las herramientas digitales utilizadas en la asignatura.

**Figura 1.**  
*Árbol de problemas*



Elaborado por: los autores



De los planteamientos antes descritos se establece el problema científico ¿cómo concebir la integración de las tecnologías digitales en el proceso de enseñanza-aprendizaje de la asignatura Programación en la especialidad Informática en el segundo año de Bachillerato Técnico?

La investigación del tema permite demostrar la importancia de proponer un enfoque educativo personalizado para incorporar tecnologías digitales en la enseñanza de la asignatura de Programación dentro del campo de la Informática. El estudio se orienta a la búsqueda de soluciones específicas y prácticas para mejorar la manera en que se enseña y se aprende programación, aprovechando las herramientas digitales y la innovación en la era actual de lo digital.

El objeto de esta investigación se establece como: la integración de las tecnologías digitales en el proceso de enseñanza-aprendizaje de la asignatura Programación en la especialidad de Informática. El objetivo general es: plantear una estrategia didáctica para la integración de las tecnologías digitales en el proceso de enseñanza aprendizaje de la asignatura Programación en la especialidad de Informática de segundo año de Bachillerato Técnico en la Unidad Educativa Shushufindi.

Los planteamientos hipotéticos para esta investigación se determinan mediante el planteamiento de las siguientes preguntas científicas.

1. ¿Cuáles son los antecedentes investigativos de la integración de tecnologías digitales en el proceso de enseñanza aprendizaje de la asignatura Programación en la especialidad Informática?
2. ¿Qué fundamentos teóricos sustentan la integración de las tecnologías digitales en el proceso de enseñanza-aprendizaje de la asignatura Programación en la especialidad Informática?
3. ¿Cuál es el estado actual de la integración de las tecnologías digitales en el proceso de enseñanza-aprendizaje de la asignatura Programación en la especialidad Informática de segundo año de Bachillerato Técnico en la Unidad Educativa Shushufindi?
4. ¿Cuáles son los componentes de la estrategia didáctica que posibilitan la integración de tecnologías digitales en el proceso de enseñanza-aprendizaje de la asignatura Programación en la especialidad Informática de segundo año de Bachillerato Técnico?



5. ¿Cuál es el resultado de la valoración de la estrategia didáctica para la integración de las tecnologías digitales en el proceso de enseñanza-aprendizaje de la asignatura Programación en la especialidad Informática de segundo año de Bachillerato Técnico en la Unidad Educativa Shushufindi?

Como objetivos específicos de la investigación se plantearon:

1. Determinar los antecedentes investigativos de la integración de las tecnologías digitales en el proceso de enseñanza-aprendizaje en la asignatura Programación en la especialidad Informática.
2. Establecer los fundamentos teóricos que sustentan la integración de tecnologías digitales en el proceso enseñanza-aprendizaje de la asignatura Programación en la especialidad Informática.
3. Diagnosticar el estado actual de la integración de las tecnologías digitales en el proceso de enseñanza-aprendizaje de la asignatura Programación en la especialidad Informática de segundo año de Bachillerato Técnico en la Unidad Educativa Shushufindi.
4. Determinar los componentes de la estrategia didáctica para la integración de tecnologías digitales en el proceso de enseñanza-aprendizaje de la asignatura programación en la especialidad Informática de segundo año de Bachillerato Técnico
5. Valorar la estrategia didáctica para la integración de las tecnologías digitales en el proceso de enseñanza-aprendizaje de la asignatura Programación en la especialidad Informática de segundo año de Bachillerato Técnico en la Unidad Educativa Shushufindi.

Para llevar a cabo la investigación, se emplean diversos métodos que facilitan la recopilación, análisis, interpretación y sistematización de los datos relacionados con el estudio del tema de investigación. Esto tiene como objetivo dar respuesta a la formulación del problema científico, las preguntas de investigación y los objetivos planteados, así como establecer el marco teórico que respalda el trabajo investigativo y proponer posibles soluciones al inconveniente identificado. Los métodos utilizados son de nivel teórico, empírico y estadístico.

Los métodos de nivel teórico facilitan la exploración y la explicación de las condiciones que rodean los fenómenos sociales. Estos se basan en teorías y conceptos relacionados con el objeto



de estudio. Los procedimientos empleados en la investigación incluyen el método deductivo-inductivo, el método analítico-sintético y la modelación.

Por otro lado, los métodos empíricos son técnicas prácticas utilizadas en la investigación científica para revelar los aspectos fundamentales del objeto de estudio. Estas técnicas incluyen realizar encuestas para recopilar las opiniones de estudiantes y profesores sobre cómo se ha llevado a cabo la integración de tecnologías digitales en el curso de Programación dentro del campo de Informática, así como recurrir al criterio de expertos para obtener sus opiniones sobre la estrategia didáctica propuesta.

Como métodos estadísticos: se emplea el análisis de distribución de frecuencias absolutas y relativas para el análisis del comportamiento de cada uno de los indicadores en los instrumentos aplicados.

La población de estudio son los docentes y estudiantes de la Unidad Educativa Shushufindi, la cual está conformada por 8 docentes y 120 estudiantes que cursan la asignatura Programación pertenecientes al segundo año de Bachillerato Técnico. Para determinar el tamaño de la muestra de los estudiantes se aplica la fórmula estadística de poblaciones finitas por cuanto se conoce la población. En consecuencia, el tamaño muestral para el grupo de estudio de los estudiantes es de 30 estudiantes. En cambio, para calcular la muestra de los docentes se aplica el muestreo No Probabilístico Intencional y se establece en 8 docentes que corresponde a la especialidad de Informática, para la muestra de los estudiantes se consideró el 25% de la población que corresponde a 30 estudiantes.

Para llevar a cabo este estudio, se considera el enfoque de investigación es cuantitativo. La investigación se clasifica como descriptiva, ya que se centra en describir características, fenómenos o situaciones actuales, y es no experimental de tipo transversal, observando a diferentes sujetos en un solo estudio a lo largo del tiempo para examinar la prevalencia de ciertas características, actitudes o comportamientos dentro de una población. Al no manipular variables ni asignar sujetos a grupos de manera controlada, la investigación transversal permite recopilar



datos sobre cómo se distribuyen ciertos fenómenos o características en una población en un momento específico.

Como contribución a la investigación se desarrolla una estrategia didáctica para integrar tecnologías digitales en el proceso de enseñanza-aprendizaje de la asignatura Programación en el campo de la Informática mediante un enfoque planificado, sistemático y personalizado. Esta estrategia propone también un conjunto de acciones que facilitan su implementación y ofrece recursos educativos en línea, como plataformas educativas virtuales, cursos online, tutoriales, libros electrónicos y vídeos educativos; estos recursos complementan y mejoran la experiencia educativa tradicional mediante el aprendizaje experiencial. Estos recursos están disponibles todo el tiempo lo cual proporciona facilidades a los estudiantes, así como también la libertad de desarrollar su aprendizaje a su propio ritmo y según sus necesidades, lo que les permite construir su propio entendimiento.

La innovación en el ámbito científico reside en la estrategia educativa propuesta que se centra en el uso de actividades que integran las TIC, con un enfoque sistemático, consciente y la posibilidad de acceso equitativo a recursos tecnológicos. Esto permite a los estudiantes interactuar y colaborar durante el proceso de enseñanza-aprendizaje del curso de Programación.

El estudio actual está organizado en capítulos que describen la línea principal de investigación, los cuales están estructurados de la siguiente manera:

En el primer capítulo se ofrece una explicación detallada sobre las investigaciones previas relacionadas con la integración de tecnologías digitales en los procesos educativos. También se mencionan las bases teóricas que respaldan la investigación, el marco conceptual relacionado con las TIC y su incorporación en el proceso de enseñanza-aprendizaje.

En el segundo capítulo se detallan los métodos para llevar a cabo la variable estudiada, el enfoque de la investigación y los procedimientos utilizados. Se define la población y muestra, se describe



cada etapa del proceso investigativo y se caracteriza la variable a través de los instrumentos empleados.

Por otro lado, en el tercer capítulo se expone una estrategia didáctica para integrar tecnologías digitales en la enseñanza de Informática. Se presentan los objetivos generales que indican lo que se busca lograr con la propuesta, así como los objetivos específicos que delimitan las acciones a seguir. Además, se detallan sus componentes estructurales, su funcionamiento dinámico y las condiciones necesarias para su ejecución. Por último, se describen los resultados obtenidos en la evaluación de la propuesta mediante consultas a expertos.

El informe finaliza con las conclusiones, sugerencias, referencias bibliográficas y una serie de anexos que enriquecen el estudio realizado.



## CAPÍTULO 1: MARCO TEÓRICO

En este capítulo se exploran los principios fundamentales que respaldan la incorporación de tecnologías digitales en el proceso de enseñanza-aprendizaje de la asignatura de Programación. Se examina la definición de integración de las TIC, así como las teorías educativas que facilitan su incorporación en los contextos formativos e institucionales. También se analizan las características clave de la asignatura de Programación en el campo de la Informática, con el fin de destacar las oportunidades que ofrece para alcanzar los objetivos establecidos.

### 1.1. Antecedentes investigativos

La investigación de antecedentes es un paso crucial en la investigación científica, ya que proporciona orientación para el desarrollo del marco teórico. Consiste en revisar la literatura relacionada con el tema para identificar aspectos clave y conceptos esenciales vinculados a la investigación.

Para encontrar trabajos académicos sobre la integración de tecnologías digitales en la enseñanza de Programación en Informática, se realizaron búsquedas en bases de datos como Dialnet, Scopus, Web of Science, Google Scholar y SciELO, donde se encontraron varios documentos relevantes.

Una definición que se acerca a las características de este concepto lo refiere Cabrera, (2013) cuando lo define como un proceso planificado, contextualizado, sistémico, continuo y reflexivo que va orientado a la transformación de la práctica educativa y a obtener las metas de aprendizaje en el proceso de enseñanza-aprendizaje.

Por otra parte, Álvarez et. al. (2013) en estudios posteriores refiere la necesidad de establecer indicadores que posibiliten caracterizarlo y para esto establecen tres dimensiones: una pedagógica, una tecnológica y una referida a la política institucional.

Al tratar los aspectos en estas tres dimensiones, los autores hacen hincapié en la importancia de definir las circunstancias relacionadas con el compromiso institucional y las medidas que abarcan desde las directrices del plan de estudios hasta su implementación por parte de los responsables de las instituciones educativas y su aplicación en el aula. Además, subrayan la relevancia de evaluar el nivel de preparación de los involucrados y la infraestructura disponible en cada escuela.



Al respecto Santos y Armas (2020) reconocen que “La integración no es una suma, una adición, o una interrelación, es una fusión de la que emergen nuevas cualidades en los objetivos, contenidos, métodos, medios, actividades y recursos, formas de organización y la evaluación; y su implementación depende de las características de los estudiantes, del grupo y el profesor”. (p. 7).

Es fundamental identificar las condiciones tecnológicas necesarias para que una institución educativa cuente con el equipamiento adecuado, la infraestructura necesaria, un nivel de conexión óptimo, entre otros aspectos relevantes. Estas condiciones, en conjunto con el estado de los aspectos pedagógicos y las políticas educativas, facilitarán la implementación de iniciativas para incorporar tecnologías digitales en los procesos de enseñanza. (Álvarez et. al, 2013), (Santos y Armas, 2020).

Investigaciones previas resaltan la importancia de abordar diversos desafíos, como el diseño, mantenimiento y gestión de la infraestructura tecnológica, así como las habilidades necesarias por parte de los docentes para incorporar las tecnologías en el plan de estudios del centro educativo. Además, se plantea el desafío de garantizar recursos y contenidos digitales que promuevan el uso e integración pedagógica de las capacidades disponibles en la institución educativa. Martínez (2021).

Otros autores (Tafur et al. (2022); Bravo y Suástegue (2022) y Rodríguez et al. (2023)) Subrayan la importancia de integrar las herramientas digitales en la educación, destacando su papel en la facilitación de un aprendizaje significativo y duradero para los estudiantes, teniendo en cuenta las posibilidades que brindan en cuanto a trabajo colaborativo, interacción social y comunicación.

A nivel regional, en el trabajo de investigación doctoral de Solano (2023) desarrollado en la Institución Unidades Tecnológicas de Santander (UTS), Bucaramanga – Colombia, se aborda una estrategia metodológica que permite a los docentes de la Educación Superior colombiana la integración de las TIC en los procesos de enseñanza-aprendizaje para propiciar el desarrollo de la competencia digital. En esta investigación se reconoce la importancia del desarrollo de la competencia digital docente por el rol que desempeña en los procesos formativos y en la innovación educativa, así como en el mejoramiento de los procesos de enseñanza-aprendizaje en



las instituciones de la Educación Superior. El autor propone una estrategia metodológica sustentado en el modelo ADDIE que contribuye al logro del objetivo propuesto.

A nivel local se identificó la tesis de maestría de Gualsaqui (2016) cuyo estudio fue aplicado en el Liceo Naval de Quito. El objetivo de la investigación consistió en determinar cómo incide el empleo de las estrategias didácticas en su relación con el uso de las TIC, para fortalecer el proceso de enseñanza-aprendizaje de la asignatura de Informática en los Primeros Cursos del Bachillerato Técnico.

El autor concluye que durante la investigación se comprobó que el ambiente de aprendizaje en el aula no propicia ejecutar estrategias didácticas constructivistas que apoyen el proceso de enseñanza-aprendizaje, ya que se emplea la pizarra, el proyector y el computador, como material didáctico exclusivo; los objetivos de aprendizaje y las actividades a realizar en clases, son conocidos por una mayoría no significativa de los estudiantes; y, las tecnológicas disponibles en la institución sirven para consultas y búsquedas en Internet. Pero, si estas tecnologías digitales se integran al proceso de enseñanza-aprendizaje, a través de modelos, métodos y procedimientos constructivistas, como estrategias didácticas apoyadas por las TIC, serán un apoyo efectivo para la construcción y reconstrucción del conocimiento, de la asignatura de Informática.

Estos resultados investigativos posibilitan tener un acercamiento teórico al concepto objeto de investigación, no obstante, es necesario profundizar en algunas teorías que desde las concepciones de aprendizaje y la Informática posibilitan demostrar la importancia y novedad del tema que se investiga.

## **1.2. Fundamentación teórica o bases teóricas**

La revolución tecnológica hace que se produzcan cambios en la educación de manera permanente con el empleo de las Tecnologías de la Información y la Comunicación (TIC), las cuales permiten el acceso a la información y la apropiación del conocimiento. Sin embargo, su desarrollo ha llevado a considerar que en su evolución ellas han rebasado esos roles y que actualmente se consideran Tecnologías para el Aprendizaje y el Conocimiento (TAC), incluso también Tecnologías para el Empoderamiento y la Participación (TEP), de ahí que en la investigación se asumen como tecnologías digitales en la educación para abarcar la totalidad de sus roles.



En el campo de la educación, la integración de tecnologías digitales en la enseñanza de la asignatura Programación en la especialidad de Informática requiere de un enfoque teórico y multidisciplinario. Este enfoque teórico no solo fundamenta el uso de herramientas tecnológicas en el aula, sino que también guía el desarrollo de estrategias didácticas innovadoras y efectivas. En el núcleo de esta integración se encuentran varias teorías pedagógicas y conceptos fundamentales que juntos forman la base para un aprendizaje significativo y relevante.

Desde una perspectiva educativa, enfoques como el Constructivismo, el Aprendizaje Basado en Problemas (ABP) y la teoría del aprendizaje experiencial destacan la importancia de un aprendizaje activo, colaborativo y centrado en el estudiante. Estas teorías ponen énfasis en la construcción del conocimiento a través de experiencias prácticas y la interacción social, proporcionando un marco para diseñar actividades educativas que integren de manera efectiva las tecnologías digitales.

En lo que respecta a las tecnologías digitales, algunas como la Teoría del Procesamiento de la Información, la Sociedad del Conocimiento y Tecnología Informatizada, Alfabetización Tecnológica Computacional y las Tecnologías de Información y Comunicación ofrecen ideas clave sobre cómo los estudiantes interactúan con la información digital y cómo esta debe ser estructurada y presentada. Estas teorías contribuyen a mejorar el uso de recursos tecnológicos en educación, garantizando que optimicen eficazmente los procesos de aprendizaje.

El estudio de Programación en Informática se beneficia de una base teórica que abarca aspectos como Teoría de la Computación, Pragmatismo en Programación, teoría de Lenguajes de Programación, Teoría de Sistemas y Aprendizaje Autodirigido.

Estas teorías no solo establecen los principios básicos en computación y algoritmos para la programación, sino que también resaltan la importancia de aplicarlos en la práctica y desarrollar soluciones a problemas complejos. La combinación de estas teorías sienta una base sólida para crear estrategias educativas que incorporen tecnologías digitales en la enseñanza de Programación, promoviendo un ambiente de aprendizaje colaborativo.



### 1.2.1. Teoría del Constructivismo

El constructivismo es una teoría que se emplea en la educación para realizar la transferencia de conocimiento para el proceso de aprendizaje de las asignaturas en las instituciones educativas, por cuanto permite la construcción de conocimientos mediante los conocimientos previos adquiridos desde la experiencia.

Esta teoría según Tejada et al. (2022) es un paradigma que “considera a los estudiantes como sujetos activos, con capacidades para tomar decisiones y emitir juicios, esto implica una participación tanto del profesor como de los estudiantes, quienes intercambian durante el desarrollo del proceso de enseñanza aprendizaje para construir conocimientos” (p. 212).

La teoría constructivista fomenta que los estudiantes sean guiados a través de estrategias conceptuales y mentales que les permitan asumir un rol activo en la comprensión y adquisición del conocimiento. En este enfoque, se destaca que el conocimiento no es simplemente transmitido por el docente, sino que el estudiante construye su propio saber a partir de sus experiencias. Por lo tanto, el rol del docente radica en promover y crear las condiciones para que el alumno pueda desarrollar su conocimiento de manera progresiva, siendo responsabilidad del estudiante la adquisición de su propio saber.

El estudiante dirige el desarrollo del conocimiento previamente adquirido para transformarlo en una representación de la realidad. El constructivismo sostiene que los estudiantes obtienen conocimientos y comprenden los eventos a partir de las experiencias previas vividas desde su propia perspectiva.

En las instituciones educativas, el constructivismo ha sido adoptado como un modelo pedagógico que indica que los estudiantes pasan de ser pasivos a ser activos al comenzar a establecer comparaciones entre sus conocimientos previos y los nuevos aprendizajes actuales. Este proceso se activa cuando los estudiantes mismos promueven la investigación y desarrollo académico.

Según Ordoñez, Ochoa y Espinoza (2020) “el constructivismo viabiliza la aprehensión de los saberes por parte del aprendiz sin necesariamente la presencia de un docente, aunque si es



acompañado por alguien que sea capaz de mediar en el proceso lo beneficiaría al sujeto que aprende” (p. 26).

### **1.2.2. El aprendizaje basado en problemas**

Esta teoría es un método de aprendizaje que permite resolver problemas trabajando de forma activa con proyectos reales. Según Guamán y Espinoza (2022) el ABP es considerado parte de las metodologías activas:

Al estar direccionado en las actividades del estudiante y le permite construir su propio aprendizaje en interacción con el entorno en el que se encuentra, por lo tanto, el ABP es un proceso intencional, organizado, planificado y sistemático que se establece como: una técnica de enseñanza, caracterizada por promover el aprendizaje autónomo y el pensamiento crítico orientado a la resolución de problemas generando la clave del éxito.

La teoría ABP: un procedimiento de aprendizaje fundamentado en el empleo de problemas (pp. 124-127).

Este enfoque fomenta la participación activa de los estudiantes, ya que estos contribuyen a la construcción de su propio conocimiento para abordar problemas reales. El estudiante impulsa el proceso de aprendizaje, mientras que el profesor actúa como facilitador. En resumen, se trata de un método que fomenta la colaboración, especialmente en entornos con grupos pequeños.

El Aprendizaje Basado en Problemas (ABP) es un método educativo que se ajusta en la resolución de problemas como la principal técnica del aprendizaje. Su contribución a la educación se resume en los siguientes factores:

Enfoque centrado en el estudiante: El Aprendizaje Basado en Problemas (ABP) coloca al estudiante como eje central del proceso educativo, ya que los problemas planteados son desafiantes y relevantes para su aprendizaje. Desarrollo de habilidades cognitivas: El ABP fomenta el desarrollo de habilidades cognitivas avanzadas, como el pensamiento crítico, la resolución de problemas y la toma de decisiones.

Los estudiantes aprenden a analizar, sintetizar y evaluar información de manera efectiva. Integración de conocimientos: Esta metodología promueve la integración de conocimientos



multidisciplinarios para abordar los problemas. Los estudiantes no solo adquieren conocimientos específicos sobre un tema, sino que también aprenden a aplicarlos de forma interdisciplinaria. Fomento del trabajo en equipo:

El ABP impulsa la colaboración y la comunicación efectiva entre los estudiantes. Al enfrentarse a problemas complejos, los alumnos deben trabajar juntos para identificar soluciones, intercambiar ideas y llegar a acuerdos, lo cual refleja la colaboración requerida en entornos profesionales.

Transferencia del aprendizaje:

Al enfrentarse a una variedad de problemas en distintos contextos, los alumnos adquieren la capacidad de aplicar sus conocimientos a nuevas situaciones. Esto les permite emplear sus saberes y habilidades en diferentes contextos y adaptarse a los desafíos cambiantes del mundo real.

### **1.2.3. Teoría del aprendizaje experiencial**

El aprendizaje experiencial, tal como lo describe Kolb (2014), destaca la importancia de adquirir conocimiento a través de la experiencia directa de aquellos que han aplicado el objeto de estudio en cuestión

Las tecnologías digitales contribuyen con recursos educativos que proveen simulaciones de roles que permiten a los estudiantes desarrollar el aprendizaje. Al respecto Espinar y Viguera (2020) describen que:

“El aprendizaje experiencial involucra “al alumno de manera directa y plasman los temas nuevos desde su mismo contexto. En este modelo, por ser significativo en el proceso de enseñanza, es necesario reconocer la significación de los aprendizajes previos del educando para lograr competencias que puedan desafiar las necesidades actuales. Esta teoría señala las ventajas para percibir y referir el proceso de aprendizaje, fundamentado en la práctica a partir de los sentidos, que son estímulos exteriores que proporcionan experiencias que encaminan el esfuerzo y la experticia de las habilidades” (p. 3).

Ella aporta resultados significativos a la enseñanza de la Programación en el campo de la Informática, ya que se basa en la idea de que el conocimiento se adquiere a través de la experiencia y la práctica. Esto permite a los estudiantes considerar las demandas de la programación, comprender la teoría, analizar funciones variables e algoritmos, comprender los datos y las



estructuras de control, y luego aplicar estos conocimientos para desarrollar productos informáticos que satisfagan las necesidades empresariales, educativas y sociales.

El uso del aprendizaje experiencial en la enseñanza de Programación en Informática involucra activamente a los estudiantes durante el proceso. Facilita el dominio de los lenguajes de programación para crear software y productos informáticos desde su concepción hasta su evaluación.

Este enfoque emergente en el desarrollo de productos informáticos ayuda a los alumnos a adquirir competencias digitales relacionadas con la programación. Se basa en habilidades técnicas fundamentales que fomentan el pensamiento crítico, resolución de problemas y trabajo colaborativo.

Estas destrezas son fundamentales en el ámbito de la programación y del conocimiento de la informática. En resumen, el aprendizaje experiencial ayuda a perfeccionar las habilidades técnicas de los estudiantes durante el proceso educativo; también les brinda una base sólida para prepararse con miras a ejercer su profesión basada en la resolución de problemas mediante la combinación del conocimiento teórico y su aplicación práctica, con el propósito de generar nuevas teorías y crear nuevos productos y servicios para beneficiar a la sociedad.

El Aprendizaje Experiencial se basa en una metodología educativa que se centra en aprender mediante la experiencia directa. Sus contribuciones al ámbito educativo se encuentran en los siguientes aspectos:

- Participación activa: esta estrategia implica que los alumnos participen activamente en su proceso de aprendizaje, permitiéndoles experimentar directamente situaciones educativas. Esto promueve un compromiso más profundo y duradero con el contenido académico.
- Aplicación práctica del conocimiento: este enfoque experiencial brinda oportunidades para aplicar el conocimiento adquirido en situaciones reales o simuladas, lo que ayuda a los estudiantes a comprender la importancia y utilidad del material académico.
- Reflexión crítica: el Aprendizaje Experiencial fomenta una reflexión crítica sobre las vivencias. Se motiva a los alumnos a analizar sus acciones, identificar patrones y extraer



lecciones de sus experiencias vividas. Esta reflexión profunda fomenta un aprendizaje que es más significativo y perdurable.

#### **1.2.4. Teoría del Procesamiento de la Información**

Esta teoría se basa en el procesamiento de la información desde una perspectiva computacional que implica el almacenamiento, la organización y la recuperación de datos. Según Morínigo y Fenner (2021), esta teoría también se utiliza para comprender cómo se abordan problemas a través de analogías y metáforas (p. 5).

Esta teoría permite analizar la información de una pregunta con el fin de obtener resultados similares a las respuestas generadas por seres humanos mediante tecnologías avanzadas como la inteligencia artificial. A pesar de esto, se encuentra en sus etapas iniciales, ya que solo ofrece soporte y no logra procesar la información de manera sostenible para resolver problemas. En otras palabras, la inteligencia artificial actúa como un apoyo para el procesamiento de datos, pero es el ser humano quien debe mejorar los resultados según las demandas sociales.

La Teoría del Procesamiento de la Información contribuye al campo informático con principios clave para el diseño de interfaces amigables, sistemas eficientes de almacenamiento y recuperación de datos, software educativo, modelos cognitivos y sistemas seguros. Todo esto colabora en el desarrollo de sistemas informáticos más efectivos, eficientes y centrados en las necesidades del usuario.

La Teoría del Procesamiento de la Información (TPI) es un marco conceptual que explora cómo las personas adquieren, almacenan, recuperan y utilizan la información. En términos informáticos, esta teoría aporta en múltiples aspectos:

**Diseño de interfaces de usuario:** La TPI proporciona principios fundamentales para diseñar interfaces efectivas y amigables para los usuarios. Al comprender cómo las personas procesan y asimilan la información, los diseñadores pueden crear interfaces que faciliten la interacción y una navegación intuitiva en los sistemas informáticos.

**Diseño de sistemas para almacenar y recuperar datos:** La Teoría del Procesamiento de la Información (TPI) proporciona conocimientos sobre cómo las personas almacenan y recuperan



información en la memoria. Este conocimiento es crucial para desarrollar sistemas de gestión de bases de datos, sistemas de archivos y algoritmos de búsqueda que mejoren la eficiencia y accesibilidad a la información.

Desarrollo de software educativo: En el ámbito educativo, la TPI orienta el desarrollo de software y aplicaciones que fomentan el aprendizaje y retención del conocimiento. Los desarrolladores pueden incorporar estrategias basadas en la TPI, como presentaciones secuenciales, retroalimentación inmediata y práctica espaciada, para mejorar la efectividad del software educativo.

Modelado cognitivo: La TPI se utiliza para modelar procesos cognitivos humanos como la resolución de problemas, toma de decisiones y razonamiento. Estos modelos mentales son muy útiles en informática para crear sistemas inteligentes y programas de inteligencia artificial que puedan simular o potenciar las capacidades cognitivas humanas.

Creación de sistemas de seguridad y privacidad: Comprender cómo las personas procesan y utilizan la información también es crucial en el desarrollo de sistemas informáticos seguros. Los programadores pueden aplicar los principios derivados del estudio sobre TPI para diseñar sistemas que protejan datos sensibles y reduzcan los errores humanos relacionados con la seguridad.

### **1.2.5. Las tecnologías de la información y las comunicaciones**

En la actualidad, todas las instituciones educativas utilizan las Tecnologías de la Información y Comunicación (TIC) para facilitar la transmisión del conocimiento durante el proceso de enseñanza. Por eso, se dedican a revisar sus planes de estudios con el objetivo de incorporar métodos didácticos que integren las TIC en el aprendizaje de todas las asignaturas.

Cedeño y Zambrano (2023) señalan que el uso de las TIC en el aula es fundamental para mejorar el proceso de enseñanza-aprendizaje, ya que permite dinamizar, reforzar, ampliar y enriquecer la labor educativa en concordancia con el cuarto Objetivo de Desarrollo Sostenible (ODS), que promueve una educación inclusiva.

Las Tecnologías de la Información y Comunicación han revolucionado la enseñanza de la Programación en informática al ofrecer herramientas dinámicas y accesibles que simplifican la



comprensión de ideas complejas y estimulan el interés de los alumnos. A través de plataformas interactivas, entornos virtuales y tutoriales en vídeo, las TIC promueven un aprendizaje más adaptable y autodidacta.

Además, fomentan la colaboración remota y el intercambio instantáneo de ideas, aspectos fundamentales para fortalecer las habilidades en programación. Las tecnologías digitales comprenden un conjunto de recursos, herramientas y técnicas utilizadas en el procesamiento, almacenamiento y transmisión de información, abarcando desde internet y telecomunicaciones hasta inteligencia artificial y big data. Estas tecnologías se basan en la digitalización e integración de datos, lo que facilita su rápida difusión y disponibilidad, transformando por completo la manera en que las sociedades se comunican, aprenden e interactúan comercialmente.

En cuanto al avance informático, han propiciado mejoras significativas al impulsar el desarrollo de software y hardware más sofisticados, sistemas de almacenamiento en la nube y plataformas informáticas distribuidas. Este progreso ha permitido gestionar eficazmente grandes volúmenes de datos, mejorando el funcionamiento e impulsando la innovación particularmente en áreas como inteligencia artificial donde el análisis masivo de datos desempeña un papel esencial.

Además, la interconexión global ha sido esencial para el surgimiento de Internet, lo que ha dado lugar a una economía digital, al comercio electrónico y a las redes sociales; redefiniendo así las prácticas comerciales y sociales a nivel mundial. En el ámbito educativo, las Tecnologías de la Información y la Comunicación (TIC) han cambiado la forma en que se lleva a cabo el proceso de enseñanza-aprendizaje al facilitar el acceso a recursos educativos digitales y permitir la creación de entornos virtuales de aprendizaje que van más allá de las limitaciones físicas del salón de clases tradicional. Las TIC desempeñan un papel vital en dar forma al futuro educativo en los siguientes aspectos:

Bibliotecas y marcos: las TIC proporcionan numerosas bibliotecas y marcos que permiten a los programadores aprovechar soluciones predefinidas para tareas comunes de programación. Estos recursos, que incluyen bibliotecas de funciones, marcos de desarrollo web y kits de desarrollo de software (SDK), aceleran el proceso de desarrollo y fomentan buenas prácticas programáticas.



Plataformas colaborativas: las TIC facilitan la colaboración entre programadores al ofrecer plataformas colaborativas en línea. Estas plataformas permiten a los desarrolladores trabajar en proyectos remotamente, compartir código, revisar cambios y colaborar en tiempo real, promoviendo así un desarrollo más ágil y colaborativo.

Recursos educativos: Las tecnologías de la información y la comunicación brindan una amplia gama de herramientas educativas en línea para programadores, que abarcan desde guías prácticas, cursos virtuales, manuales oficiales hasta grupos de trabajo colaborativo.

### **1.2.6. Teoría de la Computación**

La Teoría de la Computación es un conjunto de conceptos formales matemáticos y sus interacciones, que permiten analizar el funcionamiento de los algoritmos o programas informáticos sin considerar su codificación ni ejecución en un dispositivo físico. Según Ramos (2019), la informática es la disciplina que proporciona una base científica para estudiar el análisis de las computadoras, la programación, el procesamiento de información y el desarrollo de software.

El aporte de John Von Neumann (1995) en la Teoría de la Computación ha sido fundamental para el avance en informática, desde la arquitectura computacional hasta el procesamiento de datos. La premisa fundamental detrás de la teoría computacional es que una computadora almacena instrucciones en su memoria, las ejecuta y procesa datos según dichas instrucciones.

Esta noción rompió con el diseño previo de las máquinas, que eran programadas a través de la modificación física de sus componentes. La arquitectura de Neumann propone una estructura compuesta por una unidad de control, una unidad de aritmética/lógica, memoria y dispositivos de entrada/salida, todo interconectado. Este diseño sigue siendo la base de casi todas las computadoras actuales. En cuanto al aporte a la informática, la teoría de Neumann ha sido esencial para el avance de la computación digital.

### **1.2.7. La Teoría del Pensamiento Complejo**

El Pensamiento Complejo enseña a integrar de una forma posible los modos simplificadores de pensar y rechazar las consecuencias reduccionistas cegadoras de una simplificación de la realidad.



Según Morin (1995) “El pensamiento complejo aspira al conocimiento multidimensional. El pensamiento complejo está animado por una tensión permanente entre la aspiración a un saber no parcelado, no dividido, no reduccionista, y el reconocimiento de lo inacabado e incompleto de todo conocimiento” (p. 45). Aborda la forma como comprender los sistemas complejos de manera no lineal.

### **1.2.8. Teoría de Sistemas**

La Teoría General de Sistemas (TGS), propuesta por Bertalanffy en 1976, se presenta como un marco teórico que busca comprender y modelar la complejidad presente en sistemas interconectados. Esta teoría parte de la premisa de que ciertos principios y leyes pueden ser aplicados universalmente a sistemas, sin importar su naturaleza específica, ya sea biológica, social o tecnológica. Se fundamenta en la observación de que muchas propiedades de los sistemas no pueden ser totalmente entendidas mediante el análisis individual de sus partes constituyentes.

El surgimiento de la TGS fue motivado por la necesidad de abordar problemas complejos en diversas disciplinas científicas. Von Bertalanffy planteó que una visión sistémica podría superar las limitaciones del reduccionismo, el cual se enfoca únicamente en los componentes individuales. Con el tiempo, la TGS ha evolucionado para incorporar conceptos como la homeostasis, la retroalimentación y los subsistemas, proporcionando un lenguaje común y un conjunto de principios aplicables a todos los tipos de sistemas.

Las aplicaciones derivadas de la TGS han tenido un impacto significativo en áreas como Psicología, Ecología, Economía y Gestión Empresarial. En el ámbito informático, ha dejado una profunda huella especialmente en el desarrollo de sistemas informáticos, arquitectura del software y ingeniería de sistemas.

Proporciona una base para comprender la complejidad de los sistemas informáticos y para el desarrollo de software y hardware más eficaces, adaptables y sostenibles. La teoría resalta la importancia de la conexión y la dependencia mutua entre los componentes del sistema, principios esenciales en la creación de sistemas distribuidos, bases de datos y redes.



Además, su contribución se extiende a la integración de tecnologías digitales, inteligencia artificial y aprendizaje automático, donde el enfoque sistémico ayuda a modelar y simular redes neuronales complejas y algoritmos que imitan la capacidad de aprendizaje y adaptación presentes en sistemas biológicos. Asimismo, ha inspirado el desarrollo de metodologías para gestionar proyectos de software, como los métodos ágiles, que hacen hincapié en la adaptabilidad y la capacidad de respuesta rápida ante cambios, alineándose con una visión sistémica que considera interacciones dinámicas y evolución continua.

El análisis de estas teorías permite comprender la importancia de proponer diversas alternativas que faciliten el diseño de propuestas didácticas variadas para incorporar tecnologías digitales en el proceso educativo del curso Programación dentro del campo informático. Esto debe hacerse teniendo en cuenta las características esenciales del curso y el contexto específico en el que se aplique.

La Teoría de Sistemas aporta a la programación principios y herramientas que favorecen el diseño modular, la abstracción, la jerarquía y organización, la integración de sistemas, así como el análisis y optimización del software. Esto conlleva a una mayor estructuración, comprensión y eficiencia en los sistemas de software.

Este enfoque conceptual examina cómo interactúan los componentes relacionados entre sí para formar un conjunto unificado y funcional. En el ámbito de la programación, este marco teórico ofrece diversas perspectivas:

Diseño modular: fomenta que los sistemas de software se conciban como conjuntos de módulos interconectados, donde cada módulo desempeña una función específica y se comunica con otros módulos de manera predecible. Esta característica facilita tanto la comprensión como el mantenimiento del software, además de su escalabilidad.

Abstracción y encapsulamiento: la programación orientada a objetos, fuertemente influenciada por la Teoría de Sistemas, emplea conceptos como abstracción y encapsulamiento para representar entidades del mundo real como objetos con propiedades y comportamientos asociados. Esto posibilita un diseño más claro y una clara separación de responsabilidades en el código.



Jerarquía y organización: ofrece estructuras jerárquicas y organizativas para el desarrollo del software. Los desarrolladores pueden concebir sistemas con múltiples niveles de abstracción y complejidad, lo que facilita dividir problemas en subproblemas más manejables e implementar soluciones escalables.

Integración de sistemas: se centra en la interacción entre los distintos sistemas y subsistemas, lo cual es crucial para el desarrollo de software que requiere integrarse con otros sistemas o componentes externos. Los desarrolladores pueden emplear los principios de integración de sistemas para garantizar la interoperabilidad y la compatibilidad entre las distintas partes de un sistema de software.

Análisis y optimización: al visualizar el software como un sistema complejo compuesto por componentes interrelacionados, la Teoría de Sistemas ofrece herramientas y técnicas para analizar y mejorar el rendimiento del sistema en su totalidad. Esto implica detectar posibles cuellos de botella, optimizar los recursos disponibles y elevar la eficiencia general del sistema.

### **1.2.9. Bases normativas y legales**

Según la Constitución de la República del Ecuador (2008) promueve el acceso a la tecnología a través de su marco normativo, establecido en los siguientes artículos: 16, 27, 347 y 349.

En dichos artículos se reconoce los derechos que poseen todas las personas al acceso a las tecnologías digitales. Destaca el respeto a la inclusión, la diversidad, la justicia, la paz, pero también se menciona el respeto

Asimismo, resalta la necesidad de promover las Tecnologías de la información y las comunicaciones en la educación con la finalidad de coadyuvar la vinculación de la enseñanza con los procesos de producción económicos y sociales y destaca que esto es responsabilidad del Estado. De igual forma los preceptos constitucionales garantizan a los docentes la estabilidad laboral, formación, capacitación, salarios justos en relación con su formación académica

Otro documento que norma la posibilidad de integración de las tecnologías en los procesos formativos es la Ley Orgánica de Educación Intercultural (LOEI) (2011). En sus arts. 2, 3, 6



UNIVERSIDAD  
BOLIVARIANA  
DEL ECUADOR

## TRABAJO DE TITULACION

Reconoce el desarrollo integral de la personalidad estudiantes, el desarrollo de las capacidades con enfoque de pensamiento crítico y vocación transformadora para que se edifique una sociedad justa.

Por otro lado, expone que el Estado tiene las obligaciones de garantizar a que todos los ciudadanos tengan acceso a la educación y mejorar la educación y contribuir la alfabetización digital y el uso de las TIC en el proceso educativo.



## **CAPÍTULO 2: METODOLOGÍA PARA EL DESARROLLO DE LA INVESTIGACIÓN Y ESTUDIO DIAGNÓSTICO**

En el presente capítulo se describe la metodología empleada, el enfoque de investigación, su alcance, así como los métodos que orientan el estudio. Se describen los instrumentos aplicados y se delimita la población y muestra. Se determina la operacionalización de las variables en dimensiones e indicadores a partir de la posición que se asume en el capítulo 1 y se recogen los resultados y las valoraciones realizadas, a partir de la aplicación de los métodos utilizados.

### **2.1. Enfoque de investigación**

El presente proyecto, se desarrolla bajo un enfoque de investigación cuantitativo. Hernández et. al. (2017), considera al enfoque cuantitativo se utiliza la recolección y el análisis de datos para contestar preguntas de investigación, se sustenta en la medición numérica, el conteo y frecuentemente en el uso de estadística para establecer con exactitud patrones de comportamiento en una población.

### **2.2. Alcance de la investigación**

La finalidad del presente estudio es analizar cómo se integran las tecnologías digitales en el proceso de enseñanza aprendizaje de la asignatura Programación en la especialidad de Informática, particularmente en el segundo año de Bachillerato Técnico en la Unidad Educativa Shushufindi.

La investigación se desarrolla tomando como muestra a los estudiantes de segundo año de Bachillerato Técnico en Informática y a los docentes que imparten la asignatura de Programación. El estudio se aplicará en un tiempo determinado y se parte de la experiencia de estudiantes y docentes que participan activamente durante la investigación.

La propuesta se particulariza a partir de las condiciones y el estado en que se encuentran los indicadores analizados. Tomando en consideración estas características, el tipo de investigación por alcance empleado es Descriptivo, porque nos permite: buscar y determinar las características del objeto de estudio sometido al análisis. Según Hernández et. al. (2014) “busca especificar las características significativas del fenómeno que se analiza” (p. 92). La investigación descriptiva permite estudiar y recolectar información sobre los conceptos y variables del objeto de estudio, para proporcionar un análisis detallado y sistemático de las características y de los comportamientos de fenómenos estudiados.



### 2.3. Declaración y justificación del tipo de investigación

El tipo de investigación por las fuentes de información que se empleo es Documental, según (Arias, 2012) la describe como “un proceso basado en la búsqueda a recuperación análisis crítica e interpretación de datos secundarios es decir los obtenidos y registrados por otros investigadores en fuentes documentales, este tipo de investigación tiene su propósito en el aporte de nuevos conocimientos (p. 27).

Para operacionalizar y recopilar la información se empleó la investigación de campo, como la describe (Arias, 2012) “es aquella que se realiza en el mismo lugar y en el tiempo donde ocurre los hechos (p. 67).

El diseño de investigación es el modelo que adopta el investigador para estudiar las variables que se relacionan con el tema, para lo cual se empleó el diseño de investigación No Experimental, según Hernández et. al. (2014) son “estudios que se realizan sin la manipulación deliberada de variables y en los que sólo se observan los fenómenos en su ambiente natural para analizarlos” (p. 164), debido a que en el estudio no existen condiciones experimentales a las que se sometan las variables de estudio, los sujetos del estudio son evaluados en su contexto natural sin alterar ni manipular ninguna situación de las variables.

Esta investigación el tiempo es de corte Transversal según Hernández et. al. (2014) “recolectan datos en un solo momento, en un tiempo único, Su propósito es describir variables y analizar su incidencia e interrelación en un momento dado” (p. 154).

Los métodos utilizados durante el estudio diagnóstico y la caracterización del objeto de investigación son: métodos teóricos, empíricos y estadísticos la técnica empleada para recopilar la información es la encuesta a docentes: es un instrumento de carácter cualitativo, Díaz y Hernández (2016) que se usa como un instrumento para recopilar información primaria. Es así que en la presente investigación se aplica una encuesta a docentes para conocer cómo se utilizan las



tecnologías digitales en el proceso enseñanza aprendizaje de la asignatura de Programación. Anexo 1.

Encuesta a estudiantes: para comprobar y conocer las opiniones de los estudiantes a acerca de cómo se integran o no las tecnologías digitales en el proceso de enseñanza-aprendizaje de la asignatura Programación en la especialidad de Informática. Anexo 2.

En la elaboración de los instrumentos se utilizó una escala con cinco niveles que permitieron caracterizar la frecuencia con la que se realizan las acciones contenidas en los indicadores. El análisis se realiza a través de la distribución de frecuencias y los gráficos que permiten mostrar los resultados que se obtuvieron.

#### **2.4. Delimitación de la población y la muestra**

La población en esta investigación está conformada por los estudiantes de segundo año de Bachillerato Técnico en Informática correspondiente a 120 estudiantes y de 8 docentes pertenecientes a la asignatura de Programación de la Unidad Educativa Shushufindi, se ha determinado esta población debido a que son los participantes directos en el proceso de enseñanza-aprendizaje de esta asignatura.

Dada la naturaleza de esta investigación, se utilizó el Muestreo No Probabilístico Intencional, por cuanto es una técnica que permite seleccionar a la muestra mediante el criterio del investigador y depende principalmente de su experiencia; para lo cual se fundamenta en el conocimiento y su experticia que da confiabilidad, en consecuencia, los investigadores sólo seleccionan a las personas que creen que son adecuadas para participar en el estudio.

Con el tipo de muestreo especificado se ha tomado en cuenta que, partiendo de la naturaleza de la investigación, en relación a los estudiantes se consideró el 25% de la población en general que corresponde a 30 estudiantes, mientras que con los docentes al ser un número reducido se toma en cuenta el 100% de la población.

Fórmula para el cálculo del tamaño de la muestra:  $n = (Z^2 * p * q) / E^2$ , donde **n** es el tamaño de la muestra, **Z** es el valor representa al nivel de confianza, **p** representa a la población, **q** es la proporción de la población que no tiene la característica de interés y E es el margen de error deseado. El tamaño muestral se determina de acuerdo al siguiente procedimiento:

$$n = \frac{120 * 1.96^2 * 0.95 * 0.05}{0.0680^2 * (120 - 1) + 1.96^2 * 0.95 * 0.05}$$



$$n = \frac{120 * 3.8416 * 0.0475}{0.0046 * (119) + 3.8416 * 0.0475}$$

$$n = \frac{21.90}{0.73} = 29.88$$

El tamaño muestral determinado de los estudiantes es de 30 estudiantes y 8 docentes pertenecientes a la Unidad Educativa Shushufindi.

## **2.5. Proceder metodológico general seguido durante el proceso de investigación**

La investigación parte del análisis teórico de la definición de integración de las tecnologías digitales en el proceso de enseñanza-aprendizaje. La delimitación de sus características esenciales al asumirla como un proceso planificado, sistemático, organizado y reflexivo para lograr los objetivos de aprendizaje mediante las nuevas cualidades de los componentes didácticos del proceso de enseñanza-aprendizaje posibilita a los autores asumir las dimensiones e indicadores mencionados en el capítulo 1 y adecuarla al contexto estudiado.

Posteriormente la concreción de la operacionalización en los instrumentos propuestos posibilitó su aplicación para luego recopilar y analizar los datos obtenidos, de forma tal que permita caracterizar el estado de cada uno de ellos. Luego mediante el proceso de la triangulación de datos y la comparación determinar el estado en que se encuentra la variable estudiada.

Los resultados que se obtengan facilitarán tomar decisiones en cuanto al tipo de resultado a presentar, sus características y componentes estructurales.

En este sentido Marulanda, Giraldo y Serna (2017) considera que para el desarrollo y ejecución de un software se toma en cuenta las siguientes fases: planificación, análisis, diseño e implementación.

- **Planificación:** previo a la implementación de la estrategia didáctica es necesario planificar y plasmar las actividades que se desarrollarán durante su ejecución, lo que a su vez contribuirá a evitar improvisaciones y a trabajar con un proceso organizado y sistemático.
- **Análisis:** en el proceso del análisis se pretende descubrir las necesidades, prioridades y requerimientos de la estrategia.
- **Diseño:** el diseño se considera una de las etapas más complejas, debido a que en esta se detallan las posibles opciones de implementación y su estructura general.



- Implementación: en esta etapa, es importante tomar en consideración el ambiente en donde se desarrolla, es decir que basado en el diseño la implementación debe satisfacer las necesidades del entorno.

A continuación, se presenta en la tabla 1 la operacionalización del objeto que se investiga.

**Tabla 1.**

*Operacionalización de la variable objeto de investigación*

Dimensiones	Indicadores	Ítems
1. Política institucional	1. Disponibilidad de recursos.	¿En qué medida la institución les facilita el acceso a recursos tecnológicos para el aprendizaje de la asignatura Programación?
	2. Orientaciones institucionales:	¿En qué medida la institución le informa sobre las políticas relacionadas con el uso de tecnologías digitales en la asignatura Programación?
	3. Capacitación docente:	¿En qué medida los docentes de la asignatura Programación están capacitados para utilizar tecnologías digitales en el proceso de enseñanza-aprendizaje?
	4. Evaluación y seguimiento	¿En qué medida la institución evalúa y realiza seguimiento al uso de las tecnologías digitales en la asignatura de Programación?
2. Pedagógica	1. Alineación con los objetivos de aprendizaje	¿En qué medida las actividades que realiza con tecnologías digitales en la clase de Programación le ayudan a alcanzar los objetivos de aprendizaje?
	2. Interactividad y colaboración	¿En qué medida las tecnologías digitales que se utilizan en la clase de Programación le permiten interactuar y colaborar con sus compañeros y el docente?
	3. Personalización del aprendizaje	¿En qué medida las tecnologías digitales que se utilizan en la clase de Programación le permiten personalizar su aprendizaje?
	4. Desarrollo de competencias transversales	¿En qué medida las tecnologías digitales que se utilizan en la clase de Programación



		le ayudan a desarrollar competencias transversales?
3. Tecnológica	1. Accesibilidad y disponibilidad de tecnologías	¿En qué medida tiene acceso a las tecnologías digitales que se utilizan en la clase de Programación?
	2. Utilización de tecnologías digitales	¿En qué medida las tecnologías digitales se utilizan de forma efectiva en el proceso de enseñanza-aprendizaje de la asignatura Programación?
	3 Innovación y actualización tecnológica	¿En qué medida las tecnologías digitales que se utilizan en la clase de Programación son innovadoras y se encuentran actualizadas?
	4.Seguridad y privacidad	¿En qué medida se le garantiza la seguridad y privacidad al utilizar las tecnologías digitales en la clase de Programación?

**Elaborado por:** los autores

## 2.6. Análisis de resultados

A continuación, se presenta el análisis de los resultados de la encuesta a estudiantes y a docentes. Posteriormente se triangulan los datos provenientes de los resultados y se caracteriza el objeto investigado.

### 2.6.1. Análisis de la encuesta aplicada a los estudiantes

En cuanto a la pregunta uno referida a cómo la institución les facilita el acceso a recursos tecnológicos para el aprendizaje de la asignatura Programación, los resultados fueron los siguientes:

*1.- ¿En qué medida la institución te facilita el acceso a recursos tecnológicos para el aprendizaje de la Programación?*

La pregunta realizada sobre el acceso a recursos tecnológicos para el aprendizaje de la Programación muestra que la mayoría de los encuestados (27 de 30) equivalente 90% respectivamente, no sienten que la institución les facilita el acceso a estos recursos. Solo el 3% de los encuestados respondió que a veces se les facilita el acceso, mientras que dos correspondiente al 7% de ellos afirmaron que siempre se les facilita. Estos resultados son preocupantes, ya que la



Programación es una asignatura esencial en el mundo actual y el acceso a recursos tecnológicos es fundamental para su aprendizaje. Véase anexo 3, tabla 9 y figura 11.

2.- *¿En qué medida la institución te informa sobre las políticas relacionadas con el uso de tecnologías digitales en la asignatura de programación?*

La pregunta dos referida a valorar en qué medida la institución informa sobre las políticas relacionadas con el uso de tecnologías digitales en la asignatura Programación.

Según los resultados obtenidos, la información proporcionada por la institución sobre las políticas relacionadas con el uso de tecnologías digitales en la asignatura Programación varía entre los encuestados. Los datos muestran el 13% de los estudiantes indicaron que nunca se les informa sobre estas políticas, el 67% reportaron que rara vez se les proporciona esta información, 10% mencionaron que a veces se les informa sobre estas políticas y 10% de los estudiantes afirmaron que siempre se les proporciona esta información. Estos resultados evidencian una falta de información constante o frecuente sobre las políticas relacionadas con el uso de tecnologías digitales en la asignatura de Programación. Véase anexo 3 tabla 10 y figura 12.

3.- *¿En qué medida los docentes de la asignatura de programación te están capacitando para utilizar tecnologías digitales en el proceso de enseñanza-aprendizaje?*

La pregunta tres referida a en qué medida los docentes de la asignatura Programación los preparan para utilizar tecnologías digitales en el proceso de enseñanza-aprendizaje.

Según los resultados de la encuesta, la mayoría de los alumnos considera que sus profesores de Programación no los preparan adecuadamente para utilizar las tecnologías digitales en el proceso de enseñanza-aprendizaje. Sólo 13% estudiantes afirman que siempre los preparan para utilizar estas tecnologías, lo que sugiere que hay posibilidades de mejora en la enseñanza que se imparte. El 80% de estudiantes indican que rara vez o nunca reciben formación en este ámbito, el 7% manifiestan a veces haber recibido capacitación, esto podría ser indicativo de una falta de énfasis en la importancia de la alfabetización digital en el plan de estudios. Véase anexo 3 tabla 11 y figura 13.

4.- *¿En qué medida la institución evalúa y realiza seguimiento al uso de las tecnologías digitales que aplicas en la asignatura de programación?*



La pregunta cuatro que indaga acerca de en qué medida la institución evalúa y realiza seguimiento al uso de las tecnologías digitales que se aplica en la asignatura Programación.

Según los resultados de la encuesta, es preocupante por cuanto el 93% de los estudiantes afirman que la institución rara vez o nunca evalúa o realiza un seguimiento del uso de las tecnologías digitales en las clases de Programación. Sólo 3% de los estudiantes informa de que la evaluación se produce a veces, y sólo 3% informa de que se produce de forma sistemática. Esta falta de supervisión podría llevar a una desconexión entre los resultados de la enseñanza y el aprendizaje, así como a una falta de responsabilidad sobre la eficacia de los métodos de enseñanza, así como de los recursos y tecnologías digitales utilizados. Véase anexo 3 tabla 12 y figura 14.

*5.- ¿En qué medida las actividades que realizas con tecnologías digitales en la clase de programación te ayudan a alcanzar los objetivos de aprendizaje?*

La pregunta cinco se refiere a en qué medida las actividades que realiza con tecnologías digitales en la clase Programación le ayudan a alcanzar los objetivos de aprendizaje.

Según los resultados de la encuesta, las tecnologías digitales utilizadas en las clases de Programación son bastante eficaces para alcanzar los objetivos. El 13% de los estudiantes manifiestan rara vez o nunca ayudan alcanzar el aprendizaje; en cambio el 77% revelan que siempre y a veces alcanzan los objetivos de aprendizaje. Este resultado contrasta con los alcanzados en la pregunta 3, por lo que es necesario seguir profundizando al respecto. Véase anexo 3 tabla 13 y figura 15.

*6.- ¿En qué medida las tecnologías digitales que se utilizan en la clase de programación te permiten interactuar y colaborar con tus compañeros y docente?*

En la pregunta seis se interroga acerca de cómo las tecnologías digitales que se utilizan en la clase de Programación le permiten interactuar y colaborar con tus compañeros y docente.

Según los resultados de la encuesta, las tecnologías digitales utilizadas en las clases de Programación son moderadamente eficaces para facilitar la interacción y la colaboración entre estudiantes e instructores. Mientras que el 70% de los estudiantes respondieron positivamente, indicando que siempre se sienten capaces de interactuar y colaborar a través de las tecnologías digitales, todavía existe un 30% de estudiantes que informaron que rara vez o nunca se sienten capaces de hacerlo. Esto sugiere que puede haber margen de mejora en el uso y utilización de estas



tecnologías para crear un entorno que promueva la colaboración y la interacción. Véase anexo 3 tabla 14 y figura 16.

*7.- ¿En qué medida las tecnologías digitales que se utilizan en la clase de programación te permiten personalizar tu aprendizaje?*

En la pregunta siete se indaga acerca de en qué medida las tecnologías digitales que se utilizan en la clase de Programación le permiten personalizar su aprendizaje.

Según los resultados de la encuesta, las tecnologías digitales utilizadas en las clases de Programación son bastante eficaces a la hora de permitir a los estudiantes personalizar su experiencia de aprendizaje. El 84% de los estudiantes afirmaron que siempre se sienten capaces de personalizar su experiencia de aprendizaje a través de las tecnologías digitales, y sólo 16% indicaron lo contrario. Véase anexo 3 tabla 15 y figura 17.

*8.-¿En qué medida las tecnologías digitales que se utilizan en la clase de programación te ayudan a desarrollar competencias transversales?*

En la pregunta ocho se pregunta en qué medida las tecnologías digitales que se utilizan en la clase de Programación te ayudan a desarrollar competencias transversales.

Según los resultados de la encuesta, las tecnologías digitales utilizadas en las clases de Programación son moderadamente eficaces para ayudar a los estudiantes a desarrollar las competencias transversales. Aunque hubo 7% y 20% respectivamente de estudiantes afirmaron sentirse raramente o nunca ayudados en el desarrollo de estas competencias, el 74% de los estudiantes respondieron positivamente, indicando que a veces o siempre se sienten ayudados a través de las tecnologías digitales. Esto sugiere que puede haber oportunidades para que las instituciones y los instructores mejoren aún más el uso y la aplicación de las tecnologías digitales para desarrollar estas competencias. Véase anexo 3 tabla 16 y figura 18.

*9.-¿En qué medida tienes acceso a las tecnologías digitales que se utilizan en la clase de programación?*

La pregunta nueve se refiere a en qué medida tiene acceso a las tecnologías digitales que se utilizan en la clase de Programación.



Según los resultados de la encuesta, el 7% de los estudiantes no tiene acceso a las tecnologías digitales utilizadas en las clases de Programación. El 67% que equivale a la mayoría de los estudiantes informaron que rara vez o nunca tienen acceso, mientras que sólo el 26% de los estudiantes indicaron que a veces o siempre tienen acceso. El acceso limitado a las tecnologías digitales puede crear barreras para los estudiantes en su aprendizaje y desarrollo de habilidades en la práctica. Véase anexo 3 tabla 17 y figura 19.

*10.- ¿En qué medida las tecnologías digitales se integran de forma efectiva en el proceso de enseñanza-aprendizaje de la programación?*

La pregunta diez hace referencia a en qué medida las tecnologías digitales en su clase se utilizan de forma efectiva en el proceso de enseñanza-aprendizaje de la asignatura.

Según los resultados de la encuesta, las tecnologías digitales no se integran eficazmente en el proceso de enseñanza-aprendizaje de la asignatura Programación en las clases. El 70% de los estudiantes declararon que rara vez o nunca habían presenciado una integración eficaz, mientras que sólo el 30% indicaron que a veces o siempre observaban una integración eficaz. Esto sugiere que puede haber oportunidades para que las instituciones y los docentes exploren formas de mejorar la eficacia de las tecnologías digitales en la enseñanza de la Programación. Véase anexo 3 tabla 18 y figura 20.

*11.- ¿En qué medida se le garantiza la seguridad y privacidad al utilizar las tecnologías digitales en la clase de Programación?*

El 93% de los estudiantes declaran que se evidencia una fuerte confianza en la seguridad y privacidad de las tecnologías digitales utilizadas en la clase de Programación. Esta confianza se debe a los siguientes factores: la implementación de medidas de seguridad robustas, la capacitación adecuada del personal docente y la conciencia sobre la importancia de la seguridad digital en el entorno educativo. Sin embargo, es importante tomar en cuenta la preocupación expresada por el (7%) de los encuestados que manifiesta que la seguridad y privacidad a veces no están garantizadas. Esto podría sugerir que aún existen áreas de mejora en términos de seguridad y privacidad en el entorno a los recursos digitales educativos en la programación digital. Estos antecedentes de inseguridad se deben verificar y tomarlas acciones correctivas para fortalecer la protección de datos de los estudiantes. Véase anexo 3 tabla 19 y figura 21.



### 2.6.2. Análisis de la encuesta aplicada a los docentes

En el caso de este instrumento se ajustaron los indicadores con vistas a profundizar en algunos resultados teniendo en cuenta las respuestas ofrecidas por los estudiantes en la encuesta, con vista a comparar los datos obtenidos e indagar en posibles causas y efectos del proceso analizado.

#### 1.- *¿Cómo evaluaría el nivel de aprendizaje de programación de sus estudiantes en general?*

El 25% de los docentes correspondiente a (2) manifiestan que el nivel de aprendizaje de los estudiantes en general se puede evaluar como muy bajo, el 63% de docentes (5) afirman que tienen un nivel bajo, lo que indica la necesidad de implementar estrategias de enseñanza más efectivas o de revisar el plan de estudios para garantizar un mejor aprendizaje en esta área del conocimiento; sin embargo, no todo es negativo ya que el 13% de docentes (1) declara como nivel moderado, Véase anexo 4 tabla 20 y figura 22.

#### 2.- *¿En qué medida cree que la mayoría de sus estudiantes demuestran habilidades sólidas en la resolución de problemas a través de la Programación?*

La resolución de problemas mediante la Programación, se evidencia que el 75% evidencia un nivel bajo, el 13% presenta un nivel muy bajo. Estos resultados nos evidencian que los estudiantes tienen pocas habilidades para resolver problemas mediante la programación; estas causas requieren mejorar el desarrollo de habilidades de la Programación fortaleciendo mediante refuerzos en el aprendizaje. Sin embargo, el 13% presenta un nivel moderado de habilidades de programación. Véase anexo 4 tabla 21 y figura 23.

#### 3.- *¿Considera que la aplicación de proyectos prácticos en sus clases ha influido en el nivel de aprendizaje de sus estudiantes en la asignatura Programación?*

Los resultados muestran que no aplicar proyectos prácticos en las clases influye en el nivel de aprendizaje de los estudiantes. Los bajos resultados indican que, si bien la aplicación de proyectos prácticos ha tenido cierto impacto en el aprendizaje de la programación, es así que el 63% evidencia un nivel bajo y el 25% nivel muy bajo; Esto quiere decir que los estudiantes aún no alcanzan un nivel de aprendizaje adecuado en función de los objetivos y metas de aprendizaje. Se podrían explorar otros enfoques para mejorar la efectividad de los proyectos prácticos en el proceso de enseñanza-aprendizaje de la Programación. Únicamente el 13% de los estudiantes consideran



que el desarrollo de los proyectos influye positivamente en el aprendizaje. Véase anexo 4 tabla 22 y figura 24.

*4.-En su experiencia, ¿qué tan efectivas son las estrategias de enseñanza que utiliza para mejorar el aprendizaje de sus estudiantes?*

En cuanto a la efectividad de las estrategias de enseñanza utilizadas para mejorar el aprendizaje de los estudiantes, el 57% de los estudiantes tienen un nivel bajo, y el 14% tienen un nivel muy bajo; estos resultados indican que la mayoría de los educadores emplean estrategias de baja efectividad, estas realidades sugieren que hay áreas para mejorar en la enseñanza de Programación, ya que la mayoría de las estrategias utilizadas no están generando un impacto significativo en el aprendizaje de los estudiantes; sin embargo el 29% evidencian un nivel moderado de que las estrategias utilizadas en programación son efectivas para aprender programación. Véase anexo 4 tabla 23 y figura 25.

*5.- ¿Cómo calificaría la participación activa de sus estudiantes durante las clases de Programación?*

La participación activa de los estudiantes durante las clases de Programación se califica mayoritariamente como baja, estos resultados indican que el 13% tiene un nivel muy bajo y el 75% nivel bajo, lo que quiere decir que no participan activamente en las clases, lo que puede impactar negativamente en su aprendizaje. Únicamente el 13% de los estudiantes participan en forma activa. Por lo cual se sugiere que se podrían explorar estrategias para fomentar una mayor participación, en el uso de tecnologías digitales, la implementación de proyectos prácticos y el fomento de un ambiente colaborativo en el aula. Véase anexo 4 tabla 24 y figura 26.

*6.- ¿Qué tanto cree que el uso de tecnologías digitales ha mejorado la experiencia de aprendizaje de Programación para sus alumnos?*

El empleo de tecnologías digitales en el ámbito educativo no ha tenido un impacto significativo en la experiencia de los estudiantes en varios aspectos, debido a que 13% tienen un nivel muy bajo y el 63% un nivel bajo; estos resultados demuestran que no ha sido efectivo el uso de las tecnologías digitales para aprender programación. Estos hallazgos sugieren que todavía existe un nivel bajo de utilización de tecnologías digitales, mientras que el 25% tiene un nivel moderado, lo cual se



entiende que ha mejorado el aprendizaje de programación por el uso de tecnologías digitales. Lo cual señala áreas donde se puede mejorar la experiencia educativa. Consulte el anexo 4 tabla 25 y figura 27.

*7.- ¿En qué medida emplea tecnologías digitales para facilitar la práctica y aplicación de conceptos de programación en sus clases?*

El uso de herramientas digitales para ayudar con la práctica y aplicación de conceptos programáticos se aplica hasta cierto punto, según los resultados obtenidos. El 38% de los estudiantes opinan que su uso es moderado; El 50% lo califican como bajo y el 13% como muy bajo. Estos datos muestran, aunque se están implementando tecnologías digitales para facilitar la práctica y aplicación de conceptos programáticos, aún existen oportunidades para mejorar. Se podrían explorar otras estrategias para una mejora de la integración exitosa de estas herramientas en el proceso educativo de la programación. Consulte el anexo 4 tabla 26 y figura 28.

*8.-¿Considera que la carencia de recursos tecnológicos afecta el rendimiento y la participación de los estudiantes en la programación?*

Los resultados muestran que existe un impacto negativo, el 38% de los estudiantes lo califican como muy bajo, el 63% lo consideran bajo, estos resultados indican una afectación en el rendimiento de los estudiantes. Estos hallazgos subrayan la importancia de asegurar una adecuada provisión de recursos tecnológicos para aumentar el rendimiento y la participación estudiantil. La falta de acceso a tales recursos limita el aprendizaje y la práctica efectiva de la programación. Véase anexo 4 tabla 27 y figura 29.

*9.-¿En su opinión, cómo afecta el uso de tecnologías digitales al interés y la motivación de los estudiantes por la programación?*

Los resultados presentan un impacto moderado que ha tenido el uso de tecnologías digitales en el interés y motivación de los estudiantes hacia la programación. De acuerdo con las respuestas obtenidas, 88% de los estudiantes señalan un impacto moderado, mientras que 13% lo consideran bajo. Estos hallazgos indican que, en su mayoría, el uso de tecnologías digitales ha tenido un impacto ligero en despertar mayor interés y motivación entre los estudiantes hacia la programación. La disponibilidad de herramientas interactivas, la oportunidad de experimentar con



aplicaciones prácticas y la visualización de conceptos complejos a través de herramientas digitales son algunos de los factores que han influido en esta perspectiva. Consulte el anexo 4 tabla 28 y figura 30.

*10.- ¿Qué tan confiado/a se siente en integrar efectivamente las tecnologías digitales en su enseñanza de programación?*

Según los resultados, la mayoría de los encuestados consideran que su preparación para integrar las tecnologías digitales en la enseñanza de programación es moderada. El 25% se sitúan en una categoría moderada y 75% en una categoría baja. Estos resultados subrayan la necesidad de mejorar la formación docente en el uso efectivo de las tecnologías digitales para enseñar programación. La falta de capacitación adecuada y el acceso limitado a recursos tecnológicos son desafíos que deben abordarse para promover una mejor integración de las tecnologías digitales en el ámbito educativo. Consulte el anexo 4 tabla 29 y figura 31.

### **2.6.3. Caracterización del proceso de integración de las tecnologías digitales en el proceso de enseñanza-aprendizaje de la asignatura Programación en la especialidad Informática**

Analizados los resultados de las encuestas realizadas a los estudiantes, se pueden extraer las siguientes conclusiones:

- La dificultad para acceder a recursos tecnológicos afecta significativamente el aprendizaje de Programación en la especialidad de Informática. Se recomienda mejorar la infraestructura tecnológica y establecer políticas que garanticen un acceso equitativo para todos los estudiantes.
- La falta de información sobre las políticas relacionadas con el uso de tecnologías digitales refleja una comunicación deficiente entre la institución y los alumnos. Es fundamental establecer canales claros y efectivos de comunicación para informar a los estudiantes sobre estas políticas institucionales.
- - La percepción de que los docentes no brindan una capacitación adecuada a los estudiantes en el uso de tecnologías digitales indica la necesidad de implementar programas de formación para docentes en el ámbito de integración de competencias digitales.



- - La falta de evaluación y seguimiento del uso de tecnologías digitales revela una brecha en la supervisión de la calidad educativa. Se requiere implementar sistemas sólidos para evaluar la calidad y garantizar la efectividad del uso de herramientas digitales en el proceso educativo.
- Las diferencias en la percepción sobre la eficacia real de las tecnologías digitales resaltan la importancia de investigar a fondo los métodos pedagógicos y su impacto en el logro de objetivos educativos definidos en el Plan curricular Institucional.
- Hay posibles incumplimientos en el uso de las tecnologías digitales para potenciar la colaboración y la interacción en entornos virtuales de aprendizaje, promoviendo la participación activa y la comunicación entre estudiantes y profesores.
- La presencia de obstáculos para acceder a tecnologías digitales resalta la importancia de mejorar las políticas relacionadas con las TIC para garantizar igualdad de oportunidades educativas para todos los estudiantes, sin importar su situación socioeconómica.
- Lograr una integración sólida, confiable, coherente y estructurada de las tecnologías digitales es fundamental para desarrollar competencias transversales de manera eficiente.
- Según los resultados y percepciones de los profesores, es urgente investigar las limitaciones en el acceso a tecnologías digitales dentro de programas universales y políticas de préstamo de dispositivos destinados a estudiantes vulnerables.
- Mejorar la incorporación de tecnologías digitales en la enseñanza de Programación implica capacitar a los docentes en metodologías pedagógicas innovadoras y diseñar actividades educativas centradas en el uso efectivo de herramientas digitales.
- La preocupación por la seguridad y privacidad de los datos de los estudiantes destaca la importancia de establecer políticas sólidas y transparentes para proteger los datos, así como de ofrecer programas educativos digitales que fomenten una cultura de ciberseguridad en la comunidad educativa.



*Según las conclusiones obtenidas de las encuestas aplicadas a los docentes:*

- Se observa un bajo nivel en la evaluación del aprendizaje, lo que indica la necesidad de revisar detenidamente el plan de estudios y actualizar las estrategias docentes para mejorar el rendimiento estudiantil y garantizar una formación eficaz y de calidad.
- La falta de motivación para desarrollar habilidades a través del método de resolución de problemas subraya la urgente necesidad de una formación más sólida y específica en este ámbito, con el fin de aumentar el nivel competencial y preparar a los estudiantes adecuadamente para enfrentar desafíos prácticos en informática.
- La ausencia de proyectos prácticos en el aula tiene un impacto negativo en el proceso educativo, lo que señala la importancia de explorar métodos alternativos pedagógicos y fomentar una mayor aplicación práctica del conocimiento para mejorar la comprensión y habilidades programáticas entre los estudiantes.
- Las estrategias utilizadas por los profesores no son muy efectivas, lo que indica la necesidad urgente de revisar y actualizar los métodos y enfoques pedagógicos empleados.
- La escasa participación de los estudiantes en clase requiere identificar posibles áreas de mejora para fomentar el compromiso de los alumnos en el proceso de aprendizaje de programación, con el objetivo de lograr una participación más activa.
- La escasa utilización de tecnologías digitales en el aula resalta la importancia de implementar estrategias para integrar estas tecnologías y mejorar así el proceso educativo, brindando a los estudiantes oportunidades para desarrollar habilidades prácticas y armoniosas en programación.
- La falta de integración adecuada de tecnologías digitales en clase señala la necesidad de mejorar su uso para enseñar programación, promoviendo la aplicación práctica junto con la teoría.
- La limitada disponibilidad de recursos tecnológicos impacta negativamente en el desempeño y participación estudiantil en clases sobre programación.
- El efecto positivo que las tecnologías digitales tienen en el interés y motivación estudiantil impulsa a los alumnos a mejorar su experiencia educativa, fomentando un mayor compromiso y participación en clase.



- La formación de los profesores en la integración de tecnologías educativas digitales para enseñar programación ayuda a fortalecer las habilidades y competencias, además de fomentar el uso adecuado y técnico de las herramientas tecnológicas en el aula, lo que mejora el proceso de enseñanza-aprendizaje y promueve oportunidades laborales.



### **CAPÍTULO 3: PRESENTACIÓN Y VALIDACIÓN DE LA PROPUESTA**

En este capítulo se presentan los componentes de la “estrategia didáctica para la integración de las tecnologías digitales en el proceso de enseñanza-aprendizaje de la asignatura Programación en la especialidad de Informática”. Se analizan los resultados obtenidos mediante la valoración de la consulta a especialistas teniendo en cuenta el contexto y las características de la institución en la que se realiza la investigación, lo que permite arribar a conclusiones respecto a los logros y las mejoras a realizar.

#### **3.1. Presentación**

La asignatura Programación, por su naturaleza es de carácter práctico requiere de un gran empleo de tecnologías digitales, ya que permite a los estudiantes una mejor comprensión de los conocimientos abstractos y también promueve el desarrollo de habilidades relacionadas con la programación, tales como: el desarrollo del lenguaje de programación, uso de plataformas de programación cruzadas, desarrollar estructuras y algoritmos de datos, dominar funciones de pruebas, comprender las necesidades de los usuarios y la capacidad de adaptarse a nuevas herramientas digitales, entre otras.

Según Flores (2021) en la enseñanza de las asignaturas que se relacionan con la Informática, la utilización de metodologías activas permite obtener buenos resultados en el aprendizaje. De ahí la importancia de proponer estrategias didácticas integrando tecnologías digitales en el proceso de enseñanza-aprendizaje de la asignatura Programación para el segundo año de Bachillerato Técnico

#### **3.2. Objetivos generales y específicos**

##### **Objetivo general:**

Mejorar la integración de las tecnologías digitales en el proceso de enseñanza-aprendizaje de la asignatura Programación en la especialidad Informática, con el fin de lograr un aprendizaje significativo en los estudiantes.

##### **Objetivos específicos**

- a) Desarrollar estrategias para la implementación de las tecnologías digitales seleccionadas, considerando las necesidades y características específicas de los estudiantes.
- b) Proponer recomendaciones y ajustes al proceso de enseñanza-aprendizaje de la asignatura Programación, con el objetivo de optimizar su eficacia y adaptabilidad a diferentes contextos educativos.



### 3.3. Fundamentación (marco teórico)

Teniendo en cuenta las ideas anteriores, se presentan algunas explicaciones sobre lo que implica una estrategia didáctica. Una estrategia se concibe como un plan de acción diseñado para alcanzar un objetivo específico o una serie de metas en un determinado contexto. Según Díaz Barriga (2006), la estrategia consiste en un conjunto de acciones y actividades planificadas por el docente con el propósito de facilitar el aprendizaje de los alumnos y lograr objetivos educativos específicos. Es relevante señalar que estas acciones involucran métodos, técnicas y recursos didácticos que se seleccionan según las características y necesidades individuales de los estudiantes. Por otro lado, Tobón (2010) plantea que las estrategias didácticas están relacionadas con un conjunto coordinado de acciones diseñadas para alcanzar una meta u objetivo particular. Asimismo, Mansilla y Beltrán (2013) sostienen que una estrategia didáctica no se reduce solo a la aplicación de técnicas o al desarrollo de actividades grupales. Estos autores enfatizan la importancia de tener en cuenta los objetivos y contenidos al emplear estrategias educativas.

En este sentido, la estrategia didáctica implementada en la Unidad educativa Shushufindi para la asignatura de Programación en el área de Informática se convierte en el escenario principal del proceso de enseñanza-aprendizaje. No solo se considera qué enseñar, sino también cómo hacerlo. Esto implica una planificación detallada que abarca la selección de métodos, recursos y actividades para lograr los objetivos educativos de manera efectiva.

Las características clave de esta estrategia didáctica son las siguientes:

- **Enfoque sistemático y deliberado:** se destacan actividades específicas que coinciden con los contenidos tratados en la asignatura para lograr los objetivos específicos en cada tema. Esta planificación meticulosa es esencial para garantizar un abordaje efectivo de los objetivos de aprendizaje para los estudiantes del segundo año del Bachillerato Técnico, especializado en Informática.
- **Facilita el aprendizaje:** el objetivo principal es facilitar el proceso de aprendizaje de los estudiantes en Programación.
- **Fomenta una comprensión profunda:** se centra en enseñar Programación y promover una comprensión sólida del contenido.
- **Utiliza una variedad de métodos:** la estrategia implementada en la Unidad Educativa Shushufindi se centra en la importancia de emplear diversos métodos, técnicas, herramientas



y un enfoque didáctico que permita involucrar a los estudiantes. Esto garantiza que se atiendan las distintas necesidades educativas especiales y los estilos de aprendizaje de los alumnos de manera efectiva.

- Fomenta la transferencia de conocimientos y habilidades: fomenta una comprensión profunda y facilita la transferencia de conocimientos y habilidades. Esto implica ayudar a los alumnos a aplicar lo aprendido en diferentes contextos y situaciones reales o relacionadas con su futura profesión.

### **3.4. Características (Caracterización de la propuesta)**

#### **3.4.1. Tecnologías digitales**

A continuación, se presentan los requisitos fundamentales que se deben tener en cuenta en esta estrategia.

1. Realización del diagnóstico de las condiciones acerca de la integración de las tecnologías de digitales en el proceso de enseñanza-aprendizaje:

Se diagnostica el estado del proceso investigado a partir de las dimensiones, indicadores e instrumentos que permiten constatar el estado actual, con vistas a determinar el resto de las acciones y los planes de mejora.

2. Diseño de actividades y determinación de las condiciones, infraestructura y recursos a utilizar: se deben diseñar y planificar los procedimientos, actividades y recursos en dependencia de las condiciones, contenidos y necesidades de los estudiantes.
3. Acceso equitativo a recursos tecnológicos: la estrategia garantiza que todos los estudiantes tengan acceso equitativo a las computadoras y a una conexión a internet, de manera que puedan participar activamente en las actividades de programación. La institución educativa cuenta con tres laboratorios de informática que poseen 20 computadoras cada una y acceso a internet banda ancha.

4. Utilización de Plataformas de Aprendizaje en online y offline:

La estrategia incluye el uso de plataformas de aprendizaje en programación online y offline que son gratuitas y que permiten a los estudiantes acceder a recursos, realizar actividades prácticas y colaborar en proyectos de programación desde cualquier computadora en el laboratorio o incluso desde fuera de la institución educativa. Los programas online y offline de programación que se proponen utilizar para esta estrategia didáctica son:



### 3.4.2. Aplicaciones online y offline

**Scratch:** es un programa de desarrollo de software diseñado para introducir a los niños y jóvenes en el mundo de la programación de una manera divertida y accesible. Sus características incluyen una interfaz gráfica de bloques para construir programas, lo que facilita la comprensión de conceptos de programación como bucles, condicionales y variables además se puede utilizar de forma online y offline. Scratch (Scratch, 2024).

**Online C++ Compiler - online editor:** es una herramienta en línea que permite a los usuarios escribir, compilar y ejecutar programas en C++ directamente desde su navegador web. Con una interfaz sencilla y fácil de usar, este editor en línea brinda a los programadores un entorno conveniente para probar y depurar su código sin necesidad de instalar un software adicional GDB online (2024). **Un IDE de compilador C en línea:** es una herramienta que proporciona un entorno integrado para desarrollar, compilar y ejecutar programas en lenguaje C directamente desde un navegador web. Estos entornos de desarrollo en línea suelen ofrecer funciones como resaltado de sintaxis, autocompletado, integración con herramientas de depuración y la posibilidad de compartir y colaborar en proyectos con otros usuarios a través de Internet. Por ejemplo, Jdoodle (2024) ofrece estas características.

En el caso de **DEV-C++**, este software proporciona opciones como resaltado de sintaxis, autocompletado, depuración del código y la capacidad de compilar y ejecutar programas directamente desde la interfaz del entorno. Este IDE se centra en el desarrollo (2024).

**Cxxdroid** se presenta como un ambiente educativo para C y C++ en dispositivos Android que incluye un compilador sin necesidad de conexión a Internet, permitiendo ejecutar programas sin estar conectado. La aplicación Cxxdroid (2024) ofrece esta funcionalidad.

Por otra parte, **PseInt** es una herramienta para aprender programación utilizando pseudocódigo. Diseñada para estudiantes y principiantes en programación, este entorno amigable posibilita escribir algoritmos en pseudocódigo fácilmente. Pseint (2024) es reconocido por esta característica.

**SLE2** es un programa destinado a diseñar algoritmos para programar en cualquier aplicación o IDE sin requerir conexión a Internet. Funciona tanto en computadoras personales como en teléfonos inteligentes. Sledos (2024) es el nombre que lo identifica.

La promoción de un método colaborativo y activo para aprender fomenta la interacción entre estudiantes mediante herramientas digitales que estimulan el aprendizaje activo y la solución



conjunta de problemas relacionados con la programación. Esta colaboración se lleva a cabo en el aula y en el laboratorio, donde se organizan a los estudiantes en parejas, asignando a cada pareja un ordenador para resolver problemas utilizando programas de código abierto como los que se utilizan en Dev-C++.

**Tabla 2.**

*Unidades de trabajo*

<b>1. DATOS INFORMATIVOS</b>			
<b>Nivel Educativo:</b>	Bachillerato Técnico	<b>Figura Profesional:</b>	Informática
<b>Docente(s):</b>	Lcda. Verónica Peñafiel	<b>Área:</b>	Técnica. Informático
<b>Curso:</b>	Segundo Bachillerato	<b>Módulo:</b>	Programación y Base de Datos
<b>Sección:</b>	Diurna	<b>Duración:</b>	136 periodos
<b>2. OBJETIVOS GENERALES</b>			
<b>Competencia General:</b>		<b>Unidad de Competencia:</b>	
<b>OG:</b> Optimizar el tratamiento de la información mediante el procesamiento automático, utilizando lenguajes de programación, bases de datos, herramientas ofimáticas, redes informáticas, herramientas web, sistemas operativos y soporte técnico; proponiendo soluciones creativas e innovadoras que respondan a los requerimientos de los usuarios, aplicando procedimientos y metodologías informáticas vigentes. <b>(FIP)</b>		<b>UC3:</b> Desarrollar sistemas informáticos con lenguajes de programación y bases de datos, aplicando diferentes metodologías según los requerimientos de funcionalidad de la organización o usuario final. <b>(FIP)</b>	
<b>3. ELEMENTOS DE COMPETENCIA</b>			
<b>EC 3.1.</b> Analizar las necesidades de automatización de la información, de conformidad con los requerimientos establecidos por la organización o usuario final. <b>(FIP)</b>			
<b>EC 3.2.</b> Diseñar programas o bases de datos para atender los requerimientos identificados, empleando metodologías de programación y/o diseño de base de datos. <b>(FIP)</b>			
<b>EC 3.3.</b> Construir aplicaciones para el procesamiento de la información, empleando lenguajes de programación y gestores de base de datos. <b>(FIP)</b>			
<b>EC 3.4.</b> Validar el funcionamiento de los programas y bases de datos elaborados, verificando que satisfacen los requerimientos de la organización o usuario final. <b>(FIP)</b>			
<b>EC 3.5.</b> Elaborar el manual de usuario y de programador, detallándolos procedimientos de instalación y manipulación de la aplicación. <b>(FIP)</b>			
<b>Objetivo del Módulo:</b>			
Desarrollar sistemas informáticos con lenguajes de programación y base de datos, aplicando diferentes metodologías según los requerimientos de funcionalidad. <b>(EGC)</b>			



Segundo Año: 136 Periodos 34 Semanas			
1	Propiedades y ventajas de la POO	Describir las propiedades que debe tener la programación orientada a objetos, utilizando sus funciones para dar solución a los problemas.	16 periodos 4 S
2	Tipos de lenguaje de POO	Utilizar lenguaje C y Visual Basic como herramienta para creación interfaces de diseño de usuario.	20 periodos 5 S
3	Clases y Objetos POO	Desarrollar problemas para comprender el uso y funcionamiento de las clases y objetos de una Programación orientada a objetos (POO).	20 periodos 5 S
4	Estructuras de control:	Aplicar estructuras de control en la solución de problemas, empleando software de programación privativo o libre.	50 periodos 13 S
5	Documentación de programas (POO).	Utilizar un ambiente visual, para el desarrollo de los programas orientados objetos en las diferentes fases de su elaboración: Análisis, diseño, codificación y prueba.	20 periodos 5 S
6	Análisis estructurado	Conocer los componentes de un análisis estructurado en las primeras fases del ciclo de vida de un software.	10 periodos 2 S

**Elaborado por:** los autores

### 3.5. Estructura y dinámica de sus componentes (según el tipo de propuesta)

Estructura y dinámica de sus componentes: Para implementar la estrategia didáctica que integre tecnologías digitales en la enseñanza de programación, la institución educativa cumple con las siguientes condiciones:

#### a) Infraestructura tecnológica

- Acceso a internet: la institución cuenta con una conexión a internet estable y de banda ancha para que los estudiantes y profesores puedan acceder a los recursos digitales necesarios.
- Dispositivos electrónicos: la institución cuenta con suficientes dispositivos electrónicos (computadoras, laptops) para que los estudiantes puedan trabajar en ellos.
- Software: la institución tiene instalado el software necesario para la enseñanza de programación, como lenguajes de programación, IDEs (Entornos de desarrollo integrados).
- Mantenimiento: la institución cuenta con un plan de mantenimiento para los equipos y el software para asegurar su correcto funcionamiento con los docentes encargados de los laboratorios.

**b) Formación docente:**

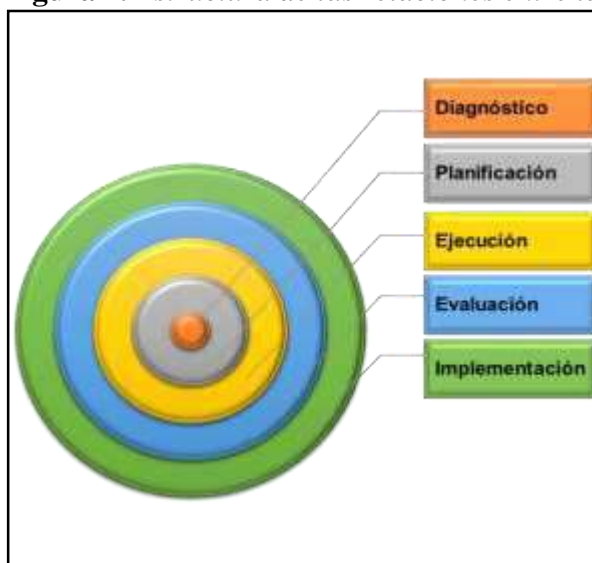
- Capacitación: los docentes de programación reciben capacitación de tecnologías digitales que se implementarán en la estrategia didáctica por medio del Ministerio de Educación en la plataforma.
- Soporte: la institución ofrece soporte técnico y pedagógico a los profesores para que puedan usar las tecnologías digitales de manera efectiva.

**c) Política institucional:**

- Apoyo: la dirección de la institución apoya la implementación de la estrategia didáctica y proporcionar los recursos necesarios por medio de un documento del departamento de rectorado
- Participación: el área de informática (profesores, estudiantes) participan en la planificación e implementación de la estrategia didáctica.
- Evaluación: la institución realiza una evaluación periódica en cada trimestre de la estrategia didáctica para identificar sus fortalezas y debilidades y realizar las mejoras necesarias.

La implementación de la estrategia didáctica para la integración de las tecnologías digitales en el proceso de enseñanza-aprendizaje de la asignatura Programación, permite desarrollar habilidades digitales y pensamiento computacional. Está estructurada en las siguientes etapas: diagnóstico, planificación, ejecución, evaluación y la implementación como lo muestra la figura 2:

**Figura 2.** Estructura de las relaciones entre las etapas de la estrategia didáctica



**Elaborado por:** los autores



**Tabla 3.**

*Etapa 1. Diagnóstico del nivel de conocimientos y habilidades de los estudiantes*

Objetivo del diagnóstico:	Evaluar el nivel de conocimiento previo de los estudiantes en programación y sus habilidades técnicas.
Actividad:	Se les proporciona a los estudiantes una serie de problemas simples relacionados con conceptos básicos de programación, como variables, condicionales y bucles. Practica: Escribe un pequeño programa que muestre los números pares del 1 al 10. Esta actividad lo realiza Utilizando la aplicación PsenInt, en el laboratorio de Informática.
Metodología:	Los estudiantes resuelven el problema individualmente El docente observará el desempeño de los estudiantes y toma nota de las partes en las que puedan necesitar refuerzo.
Resultados esperados:	Se Identifica fortalezas y debilidades en el conocimiento inicial de programación de los estudiantes. Esta información es relevante para adaptar la planificación del curso según las necesidades detectadas durante el diagnóstico.
Esta actividad práctica permite al docente obtener información valiosa sobre el punto de partida de los estudiantes en cuanto a sus habilidades y conocimientos en programación, lo que facilitará la adaptación de las actividades para satisfacer las necesidades específicas del grupo.	

Elaborado por: los autores

**Tabla 4.**

*Etapa 2. Planificación del sistema de clases con sus respectivos componentes didácticos: objetivos, contenidos, métodos, medios digitales, formas de organización y evaluación*

Objetivo de la Planificación:	Establecer la planeación docente con el sistema de clases que abarque todos los temas y habilidades necesarias para que los estudiantes alcancen los objetivos de aprendizaje.
Actividad propuesta:	El docente elaborará un plan de enseñanza que incluirá los temas a abordar, las actividades prácticas y las evaluaciones a realizar. Ejemplo de clase Temas a abordar: Variables, condicionales, bucles, funciones, objetos, herencia, etc. Actividades prácticas: Programación de juegos, desarrollo de aplicaciones, resolución de problemas, etc. Evaluaciones: Pruebas, proyectos, presentaciones, etc.
Metodología:	El docente compartirá el plan de enseñanza con los estudiantes y solicitará sugerencias y comentarios. El plan de enseñanza se adaptará según las necesidades y sugerencias de los estudiantes.



Resultados esperados:	Un sistema de clases adaptado a las necesidades de los estudiantes. Una estructura clara y organizada para la ejecución del curso.
Este ejemplo práctico permitirá al docente establecer un sistema de clases que abarque todos los temas y habilidades necesarias para que los estudiantes alcancen los objetivos de aprendizaje, y también permitirá a los estudiantes participar en la planificación del curso, lo que aumentará su compromiso y motivación.	

Elaborado por: los autores

### Tabla 5.

#### *Etapa 3.- Ejecución de las actividades planificadas*

Objetivo de la Ejecución:	Llevar a cabo las actividades planificadas en el plan de enseñanza para que los estudiantes adquieran habilidades prácticas en programación.
Actividad propuesta:	Los estudiantes participarán en sesiones prácticas donde aplicarán los conceptos teóricos aprendidos en situaciones reales de programación. Ejemplos de actividades: Desarrollo de pequeños proyectos de programación. Resolución de problemas prácticos utilizando diferentes lenguajes de programación. Trabajo en equipo para desarrollar aplicaciones simples.
Metodología:	El docente guiará a los estudiantes durante las sesiones prácticas, brindando apoyo y resolviendo dudas.
Resultados esperados:	Aplicación efectiva de los conceptos teóricos en situaciones prácticas. Desarrollo de habilidades de resolución de problemas y trabajo en equipo. Retroalimentación constante para mejorar el aprendizaje durante la ejecución del curso.
Este ejemplo práctico permitirá a los estudiantes poner en práctica los conocimientos adquiridos, desarrollar habilidades prácticas en programación y recibir retroalimentación constante para mejorar su aprendizaje. La ejecución efectiva del plan de enseñanza garantizará que los estudiantes adquieran experiencia real en el campo de la programación.	

Elaborado por: los autores

### Tabla 6.

#### *Etapa 4.- Evaluación del nivel de conocimientos y habilidades*

Objetivo de la Evaluación:	Evaluar el nivel de conocimiento y habilidades de los estudiantes en programación y determinar si han alcanzado los objetivos de aprendizaje.
Actividad propuesta:	Se realizarán evaluaciones periódicas que incluirán pruebas, proyectos y presentaciones. Ejemplo de Evaluaciones: Pruebas teóricas sobre conceptos básicos de programación.



	<p>Pruebas prácticas que evalúen la capacidad de los estudiantes para aplicar los conceptos teóricos en situaciones reales de programación. Presentaciones sobre proyectos de programación desarrollados por los estudiantes.</p>
Metodología:	<p>Se realizarán evaluaciones periódicas durante el curso para evaluar el avance del aprendizaje. Se proporcionará retroalimentación constante para ayudar a los estudiantes a mejorar su aprendizaje.</p>
Resultados esperados:	<p>Evaluación objetiva del nivel de conocimiento y habilidades de los estudiantes en programación. Identificación de áreas en las que los estudiantes necesitan más apoyo y refuerzo. Información relevante para adaptar el plan de enseñanza y mejorar el aprendizaje durante el curso.</p>
<p>Este ejemplo práctico permitirá a los docentes evaluar el progreso de los estudiantes en programación y adaptar el plan de enseñanza según sus necesidades específicas. La evaluación periódica y la retroalimentación constante garantizarán que los estudiantes reciban el apoyo necesario para mejorar su aprendizaje y alcanzar los objetivos de aprendizaje.</p>	

Elaborado por: los autores

### Tabla 7.

#### *Etapa 5.- Implementación en proyectos prácticos*

<b>Objetivo de la Implementación:</b>	<p>Poner en práctica los conocimientos adquiridos durante el curso y aplicarlos en proyectos reales o simulaciones.</p>
Actividad propuesta:	<p>Los estudiantes trabajarán en proyectos prácticos que les permitirán utilizar sus habilidades de programación en situaciones reales. Ejemplo de implementación: Desarrollo de una aplicación móvil sencilla. Creación de un juego interactivo utilizando un lenguaje de programación específico. Participación en competencias de programación para resolver problemas desafiantes.</p>
Metodología:	<p>Los estudiantes trabajarán en equipos o de forma individual, dependiendo del tipo de proyecto. Se fomentará la creatividad, la resolución de problemas y la colaboración entre los estudiantes durante la implementación.</p>
Resultados esperados:	<p>Aplicación efectiva de los conocimientos adquiridos en proyectos prácticos. Desarrollo de habilidades avanzadas en programación y resolución de problemas.</p>



	Experiencia práctica que prepara a los estudiantes para enfrentar desafíos del mundo real en el campo de la programación.
Este ejemplo práctico permitirá a los estudiantes aplicar sus habilidades de programación en proyectos concretos, lo que les brindará una experiencia práctica invaluable y les preparará para enfrentar desafíos del mundo real en el campo de la programación. La implementación exitosa garantizará que los estudiantes consoliden sus conocimientos y habilidades adquiridas durante el curso.	

Elaborado por: los autores

### **3.6. Demostración de las prácticas de actividades de programación en el laboratorio de Informática**

#### **a) Propiedades y ventajas de la Programación Estructurada.**

**Título:** Clase estructurada de programación

**Objetivo:** Introducir al estudiante en los fundamentos de la programación estructurada, enfatizando sus propiedades y ventajas.

Duración de la clase: 60 minutos

#### **Esquema de clase práctica:**

Introducción (5 minutos)

Analice brevemente la importancia de la programación en el mundo actual.

Introducir el concepto de estructuras de programación: procedimental.

Explique el enfoque de esta clase: programación estructurada.

Propiedades de la Programación Estructurada (20 minutos)

#### **Analice las propiedades clave de la programación estructurada:**

Secuencia, Selección (declaraciones if-else), Iteración (bucles)

Explique la importancia de cada propiedad y muestre ejemplos utilizando un lenguaje de programación visual o un lenguaje simple basado en texto como Python.

#### **Analice los beneficios de la programación estructurada:**

Legibilidad de código mejorada

Mantenimiento de código más sencillo

Mejor depuración y pruebas

Proporcione ejemplos del mundo real y estudios de casos en los que la programación estructurada haya sido beneficiosa.



### Ejercicio práctico de codificación (15 minutos)

Asigne un ejercicio de codificación simple que requiera que los estudiantes apliquen los conceptos de programación estructurada.

Anime a los estudiantes a trabajar en parejas o grupos pequeños para promover la colaboración y la discusión.

### Resumen y sesión de preguntas y respuestas (5 minutos)

Dé la oportunidad para que se planteen preguntas y se den respuestas.

Ofrezca recursos adicionales a los alumnos interesados en profundizar en la programación estructurada.

### Evaluación:

Observe a los alumnos mientras realizan ejercicios prácticos de codificación para evaluar su comprensión de los conceptos de programación estructurada.

Recopile y revise los ejercicios de codificación para proporcionar retroalimentación.

### Materiales:

Lenguaje visual o basado en texto como Dev C++ para programar.

Folletos o presentaciones con conceptos clave y ejemplos.

Hojas de ejercicios de codificación.

### Aplicación utilizada dev-C++

Figura 3. Aplicación utilizada dev-c++

```
1 #include <iostream>
2 #include <vector>
3
4 using namespace std;
5
6 int main() {
7
8     // Declaración de variables
9     int opcionesPrincipales, opcionesSubPrincipales;
10    vector<int> vectorNumeros;
11
12    // ...
13
14    return 0;
15 }
16
17 while (true) {
18     // Mostrar opciones del menú principal
19     cout << "OPCIONES PRINCIPAL" << endl;
20     cout << "1. Operaciones básicas con vectores" << endl;
21     cout << "2. Operaciones avanzadas con vectores" << endl;
22     cout << "3. Salir del programa" << endl;
23     cout << "Ingrese la opción deseada: ";
24     cin >> opcionesPrincipales;
25
26     // Validar la opción ingresada
27     if (opcionesPrincipales < 1 || opcionesPrincipales > 3) {
```

Elaborado por: los autores



**b) Tipos de lenguaje de programación estructurada.**

**Clase Práctica:** Tipos de Lenguajes de Programación Estructurada

**Objetivo:** Introducir a los estudiantes a los diferentes tipos de lenguajes de programación estructurada, sus características y aplicaciones.

Duración: 2 horas

**Materiales:**

Computadoras con acceso a internet

Proyector (opcional)

Pizarrón o rotafolio

Marcadores

Apuntes de la clase (opcional)

Metodología:

Introducción (15 minutos)

Iniciar la clase con una breve introducción a la programación estructurada, sus ventajas y beneficios.

Explicar la diferencia entre lenguajes de bajo y alto nivel, y cómo los lenguajes de programación estructurada se ubican en esta clasificación.

**Presentar los diferentes tipos de lenguajes de programación estructurada, incluyendo:**

Lenguajes imperativos: Pascal, C, C++, Java, Python

Lenguajes funcionales: Lisp, Haskell, Erlang

Lenguajes lógicos: Prolog

Exposición de casos prácticos (30 minutos)

Mostrar ejemplos de código en diferentes lenguajes de programación estructurada para ilustrar sus características y sintaxis.

Explicar las aplicaciones y usos comunes de cada tipo de lenguaje.

Brindar ejemplos de proyectos o programas que se pueden desarrollar con cada lenguaje.

**Actividad práctica (45 minutos)**

Dividir a los estudiantes en grupos y asignar un lenguaje de programación estructurada.

Entregar a cada grupo una guía con instrucciones para realizar una actividad práctica sencilla en el lenguaje asignado.



Los estudiantes deben trabajar en equipo para completar la actividad y luego presentar sus resultados al resto de la clase.

Conclusiones y preguntas (15 minutos)

Resumir los puntos clave de la clase.

Responder a las preguntas de los estudiantes.

Brindar recomendaciones para que los estudiantes continúen aprendiendo sobre lenguajes de programación estructurada.

**Recursos adicionales:**

Wikipedia - Programación estructurada:

[https://es.wikipedia.org/wiki/Programaci%C3%B3n\\_estructurada](https://es.wikipedia.org/wiki/Programaci%C3%B3n_estructurada)

Introducción a la programación: <https://www.khanacademy.org/computing/computer-programming>

Tutoriales de lenguajes de programación: <https://www.tutorialspoint.com/>

**Evaluación:**

Se puede evaluar la participación de los estudiantes en la clase, su comprensión de los conceptos y su desempeño en la actividad práctica.

**Recomendaciones:**

Es importante adaptar la clase al nivel de conocimiento de los estudiantes.

Se pueden utilizar recursos adicionales como videos, presentaciones o ejercicios interactivos para complementar la clase.

Es recomendable que los estudiantes practiquen por su cuenta para reforzar los conceptos aprendidos.

**Ejemplo de actividad práctica:**

Lenguaje: c++

Actividad: Escribir un programa que imprima los números del 1 al 10.

**Guía de la actividad:**

Importar la biblioteca math.

Definir una variable número e inicializarla con el valor 1.

Utilizar un bucle while para imprimir los números del 1 al 10.

Incrementar el valor de la variable número en 1 en cada iteración del bucle.



### Ejemplo de código:

C++

```
import math
numero = 1
while numero <= 10:
    print(numero)
    numero += 1
```

Use code with caution.

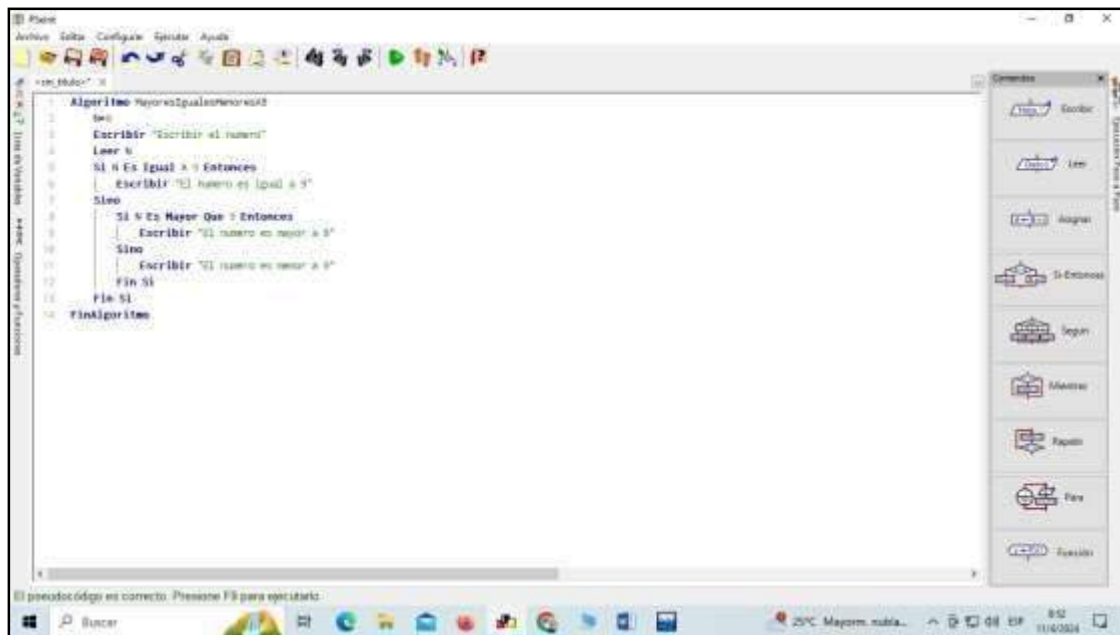
### Conclusion:

Esta clase práctica proporcionará a los estudiantes una base sólida para comprender los diferentes tipos de lenguajes de programación estructurada, sus características y aplicaciones.

### APLICACIÓN UTILIZADA PSEINT

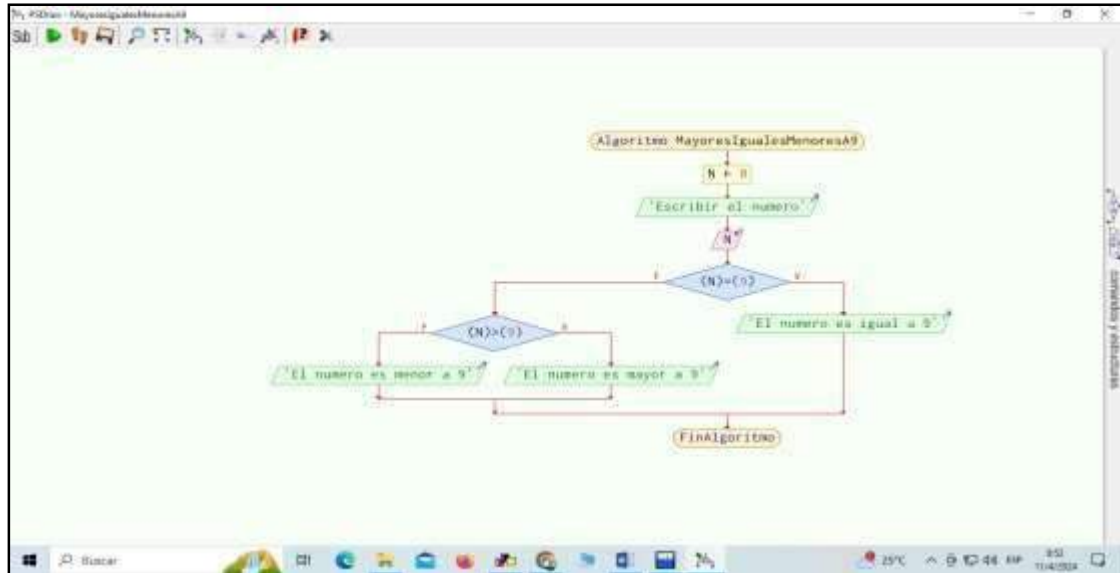
#### Figura 4.

*Aplicación utilizada PSEINT*



**Elaborado por:** los autores

**Figura 5.**  
*Aplicación PSEINT*



**Elaborado por:** los autores

c) **Desarrollar programas para comprender el funcionamiento de la estructura de la programación en DEV-C++**

**Clase Práctica:** Desarrollo de Programas en DEV-C++ para comprender el Funcionamiento de la Estructura de la Programación

**Objetivo:** Introducir a los estudiantes al desarrollo de programas en DEV-C++ para comprender el funcionamiento de la estructura de la programación.

Duración: 2 horas

Materiales:

Computadoras con DEV-C++ instalado

Proyector (opcional)

Pizarrón o rotafolio

Marcadores

Apuntes de la clase (opcional)

Metodología:

1. Introducción (15 minutos)



Iniciar la clase con una breve introducción a DEV-C++, un entorno de desarrollo integrado para C/C++.

Explicar la importancia de la estructura de la programación y sus componentes básicos:

Secuencias

Selecciones

Iteraciones

Introducir las herramientas y funcionalidades básicas de DEV-C++:

Editor de código

Compilador

Depurador

## **2. Demostración práctica (30 minutos)**

Mostrar cómo crear un nuevo proyecto en DEV-C++.

Escribir un programa sencillo en C++ que ilustre las estructuras de control básicas:

if-else para las selecciones

for y while para las iteraciones

Explicar la sintaxis y el funcionamiento de cada estructura de control.

Ejecutar el programa y observar el resultado.

## **3. Actividad práctica (45 minutos)**

Dividir a los estudiantes en grupos pequeños y asignar a cada grupo un problema sencillo de resolver utilizando estructuras de control.

Los estudiantes deben trabajar en equipo para desarrollar el programa en DEV-C++, probarlo y corregir errores.

Cada grupo debe presentar su solución al resto de la clase.

## **4. Conclusiones y preguntas (15 minutos)**

Resumir los puntos clave de la clase.

Responder a las preguntas de los estudiantes.

Brindar recomendaciones para que los estudiantes continúen aprendiendo sobre la programación en C++ y DEV-C++.

### **Recursos adicionales:**

Sitio web de DEV-C++: [se quitó una URL no válida]



Tutoriales de C++: [se quitó una URL no válida]

Documentación de DEV-C++: [se quitó una URL no válida]

### **Evaluación:**

Se puede evaluar la participación de los estudiantes en la clase, su comprensión de los conceptos, su capacidad para desarrollar programas en DEV-C++ y su trabajo en equipo.

### **Recomendaciones:**

Es importante adaptar la clase al nivel de conocimiento de los estudiantes.

Se pueden utilizar recursos adicionales como videos, presentaciones o ejercicios interactivos para complementar la clase.

Es recomendable que los estudiantes practiquen por su cuenta para reforzar los conceptos aprendidos.

Ejemplo de problema para la actividad práctica:

Escribir un programa que solicite al usuario un número entero y luego imprima si es par o impar.

Ejemplo de código:

C++

```
#include <iostream>
```

```
int main() {
```

```
    int numero;
```

```
    std::cout << "Introduzca un número entero: ";
```

```
    std::cin >> numero;
```

```
    if (numero % 2 == 0) {
```

```
        std::cout << "El número " << numero << " es par" << std::endl;
```

```
    } else {
```

```
        std::cout << "El número " << numero << " es impar" << std::endl;
```

```
    }
```

```
    return 0;
```

```
}
```

Use code with caution.

### **Conclusión:**



Esta clase práctica proporcionará a los estudiantes una base sólida para comprender el funcionamiento de la estructura de la programación y desarrollar programas sencillos en DEV-C++.

## APLICACIÓN UTILIZADA SLE2

### Figura 6.

Aplicación utilizada SLE2

```
1 // * Calculo recursivo del factorial de n.
2
3
4 var
5   i, n : numerico
6 inicio
7
8   * Estructura sobre fondo negro
9
10  set_color (8, 1)
11  clr()
12
13  imprimir ("CALCULO DE FACTORIAL",
14           "n",
15           "Ingrese un numero (0-n) ")
16  leer (n)
17  si ( n >= 0 and n == int (n) ) {
18    f=fact (n)
19
20    * Como el factorial tienen a ser un numero grande, por defecto se
21    * mostrará en notación científica desde n=10. Usando str()
22    * podemos hacer que se muestre con todos sus digitos, y no
23    * en notación científicas.
24
25  }
```

Elaborado por: los autores

### d) Estructuras de control en programación

**Clase Práctica:** Estructuras de Control en Programación Estructurada

**Objetivo:** Introducir a los estudiantes a las estructuras de control básicas en la programación estructurada y su aplicación en la resolución de problemas.

Duración: 2 horas

Materiales:

Computadoras con acceso a internet y un IDE de su preferencia (Python, Java, C++, etc.)

Proyector (opcional)

Pizarrón o rotafolio

Marcadores

Apuntes de la clase (opcional)



Metodología:

### **1. Introducción (15 minutos)**

Iniciar la clase con una breve introducción a la programación estructurada y sus beneficios.

Definir las estructuras de control básicas:

Secuencias

Selecciones (condicionales)

Iteraciones (bucles)

### **2. Demostración práctica (30 minutos)**

Mostrar ejemplos de código en diferentes lenguajes de programación que ilustren las estructuras de control básicas:

Secuencias: imprimir una serie de números

Selecciones: determinar si un número es par o impar

Iteraciones: calcular la suma de los números del 1 al 10

Explicar la sintaxis y el funcionamiento de cada estructura de control en el lenguaje elegido.

Ejecutar los programas y observar el resultado.

### **3. Actividad práctica (45 minutos)**

Dividir a los estudiantes en grupos pequeños y asignar a cada grupo un problema sencillo de resolver utilizando estructuras de control.

Los estudiantes deben trabajar en equipo para:

Diseñar el algoritmo utilizando las estructuras de control adecuadas.

Implementar el algoritmo en el lenguaje de su preferencia.

Probar y corregir errores en su programa.

Presentar su solución al resto de la clase, incluyendo el código y la explicación del algoritmo.

### **4. Conclusiones y preguntas (15 minutos)**

Resumir los puntos clave de la clase.

Responder a las preguntas de los estudiantes.

Brindar recomendaciones para que los estudiantes continúen aprendiendo sobre las estructuras de control y la programación en general.

Recursos adicionales:

Tutoriales de programación:



Python: [se quitó una URL no válida]

Java: [se quitó una URL no válida]

C++: [se quitó una URL no válida]

Documentación de los IDEs:

Python: [se quitó una URL no válida]

Java: [se quitó una URL no válida]

C++: [se quitó una URL no válida]

### **Evaluación:**

Se puede evaluar la participación de los estudiantes en la clase, su comprensión de los conceptos, su capacidad para resolver problemas utilizando estructuras de control, su trabajo en equipo y la calidad de su código.

### **Recomendaciones:**

Adaptar la clase al nivel de conocimiento de los estudiantes.

Utilizar recursos adicionales como videos, presentaciones o ejercicios interactivos para complementar la clase.

Recomendar a los estudiantes que practiquen por su cuenta para reforzar los conceptos aprendidos.

Aquí tienes una tarea práctica para practicar:

Desarrolla un programa que pida al usuario dos números enteros y luego muestre en pantalla cuál es el mayor de los dos.

Te dejo un ejemplo en Python:

Python

```
num1 = int(input("Ingresa el primer número: "))
```

```
num2 = int(input("Ingresa el segundo número: "))
```

```
if num1 > num2:
```

```
    print("El número más grande es:", num1)
```

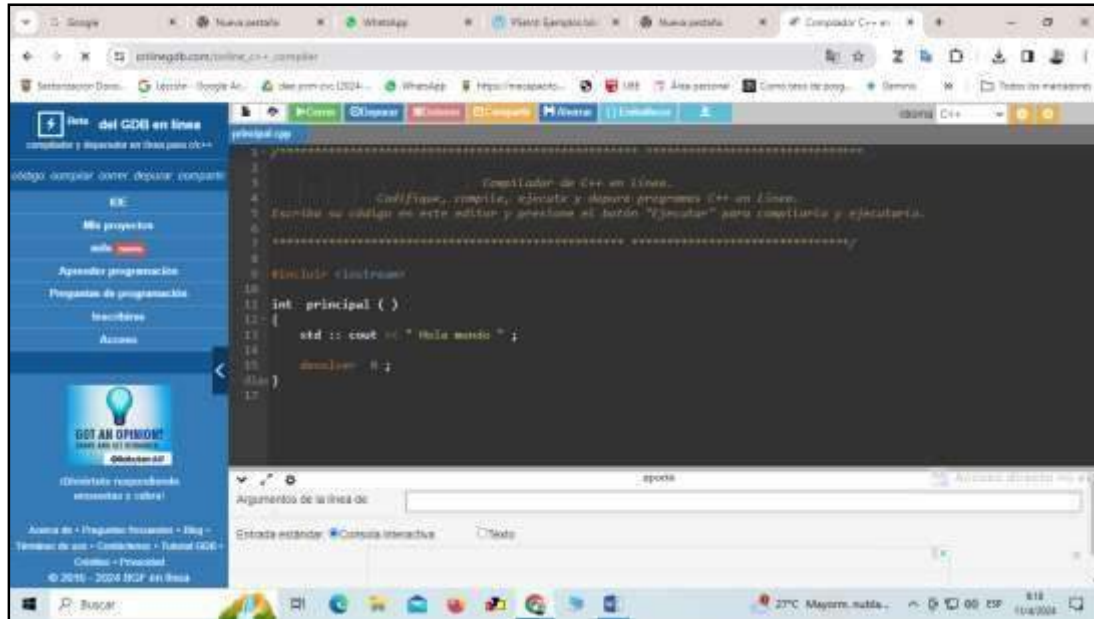
```
else:
```

```
    print("El número más grande es:", num2)
```

Recuerda manejar el código con precaución.

**APLICACIÓN UTILIZADA ONLINE C**

**Figura 7.**  
*Aplicación utilizada ONLINE C*



**Elaborado por:** los autores

**Documentación de la programación estructurada.**

Clase Práctica: Salvaguarda del Código de Programación

Propósito: Iniciar a los alumnos en la importancia de respaldar el código de programación y enseñarles diversas formas de hacerlo.

Duración: 1 hora

Recursos:

- Computadoras con acceso a internet
- Proyector (opcional)
- Pizarra o papelógrafo
- Marcadores
- Notas de la clase (opcional)

Método:

1. Introducción (15 minutos)

Comenzar la clase con una breve presentación sobre la relevancia de realizar respaldos del código de programación.



Explicar las consecuencias de perder el código fuente, como:

- Pérdida del tiempo y esfuerzo dedicados al proyecto.
- Dificultad o imposibilidad para recuperar el código original.
- Posibles implicaciones económicas o legales.

## 2. Formas de Respaldo (30 minutos)

Almacenamiento local:

Guardar el código en distintas carpetas del disco duro.

Utilizar un software para control de versiones como Git o Mercurial.

Almacenamiento en la nube:

Subir el código a servicios como GitHub, GitLab o Bitbucket.

Respaldo automático:

Configurar un sistema automático de respaldo que se active periódicamente.

## 3. Actividad práctica (15 minutos)

Dividir a los estudiantes en grupos pequeños y asignar a cada grupo una forma específica de respaldo.

Los estudiantes deben:

Investigar en profundidad el método de respaldo asignado.

Elaborar una breve presentación sobre los pros y contras de dicho método.

Exponer la información al resto de la clase.

## 4. Conclusiones y preguntas (10 minutos) Resumir

los puntos esenciales abordados en clase.

Responder las preguntas planteadas por los estudiantes.

Sugerir a los alumnos que consideren implementar un sistema de respaldo para su código de programación.

Recursos adicionales:

Guías sobre control de versiones:

Git: [una URL válida se ha eliminado]

Mercurial: [una URL válida se ha eliminado]

Información sobre almacenamiento en la nube:

Google Drive: [una URL válida se ha eliminado]



Dropbox: [una URL válida se ha eliminado]

OneDrive: [una URL válida se ha eliminado]

Evaluación:

Se evaluará la participación de los estudiantes, su comprensión acerca de las técnicas de respaldo y la calidad de sus exposiciones.

Recomendaciones:

Adaptar el contenido al nivel académico del alumnado.

Emplear recursos audiovisuales o presentaciones para complementar las clases.

Proporcionar a los estudiantes ejemplos prácticos sobre cómo instaurar un sistema adecuado para asegurar su código fuente.

Presentación de ejemplo:

Tema: Respaldo de código de programación utilizando Git

Beneficios:

Gestión de versiones: permite rastrear el historial de cambios del código.

Colaboración: facilita la colaboración en equipo en proyectos de programación.

Seguridad: protege el código contra pérdidas accidentales o corrupción.

Inconvenientes:

Requiere tiempo para comprender los comandos.

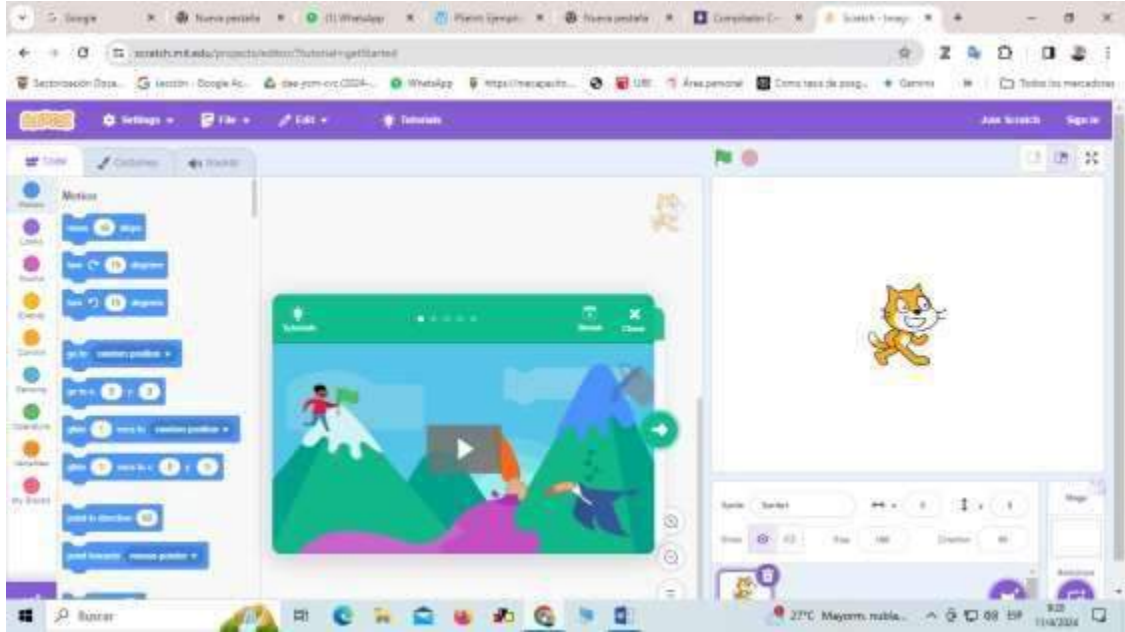
Puede resultar complicado en proyectos extensos o con múltiples colaboradores.

Conclusión:

Realizar respaldos del código de programación es una práctica fundamental para salvaguardar el trabajo de los programadores. Existen diversas formas de respaldo disponibles, cada una con sus pros y contras. Los estudiantes deben seleccionar la alternativa que mejor se ajuste a sus necesidades y preferencias.

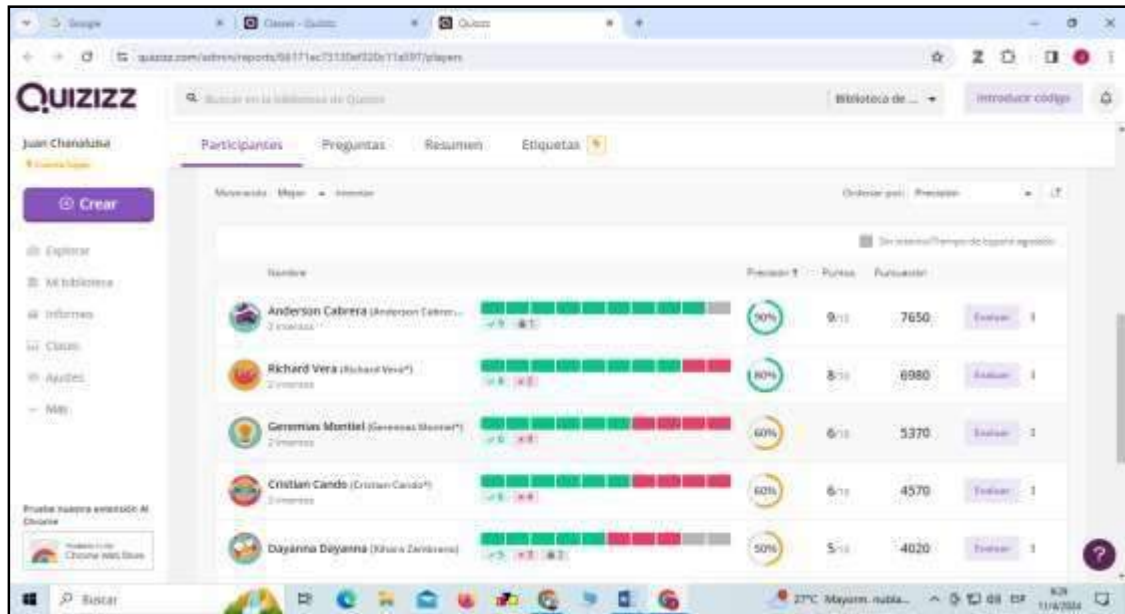
**Aplicación utilizada search y quizziz**

**Figura 8.**  
*Aplicación utilizada SCARCH*



**Elaborado por:** los autores

**Figura 9.**  
*Aplicación utilizada Quizziz*



**Elaborado por:** los autores



### 3.7. Validación de la propuesta

Para asegurar que la estrategia didáctica propuesta sea efectiva y a fin de mejorar el proceso de enseñanza-aprendizaje de la asignatura Programación y preparar a los estudiantes para los desarrollar sus habilidades en programación, se validó tomando en cuenta las siguientes características:

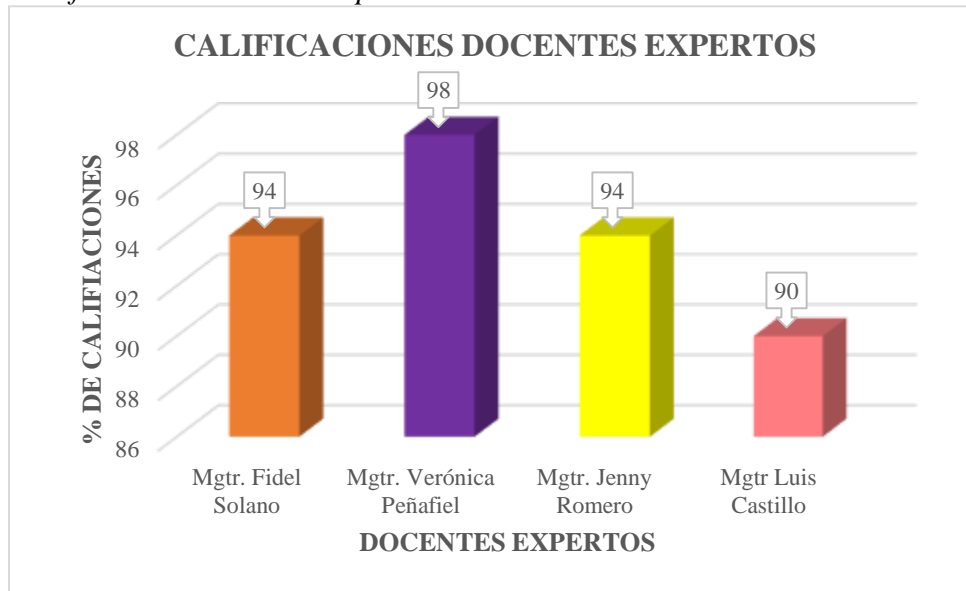
- a) La propuesta tiene una adecuación a la planificación curricular institucional, es decir, está organizada con la competencia general, unidad de competencia, elementos de competencia, objetivo del módulo y unidades de trabajo de la asignatura de Programación en Informática.
- b) Tiene una interrelación con los recursos para asegurar que los recursos digitales utilizados originen la interacción activa entre los estudiantes y promuevan la exploración y práctica de conceptos de programación.
- c) Dispone de una feedback continuo por cuanto la estrategia incluyen actividades para brindar a los estudiantes retroalimentación inmediata a fin de mejorar sus habilidades y comprensión.
- d) También tiene accesibilidad de las herramientas, lo cual conlleva a seleccionar herramientas y plataformas digitales accesibles para todos los estudiantes
- e) Se tomo en cuenta la capacitación docente para brindar una capacitación continua para que puedan integrar efectivamente las tecnologías digitales en la enseñanza de la programación informática.

La estrategia propuesta se llevó a cabo en cinco fases: diagnóstico, planificación, ejecución, evaluación e implementación en la Unidad Educativa Shushufindi durante los años 2023 y 2024. Actualmente, al encontrarse en proceso de implementación, se verificó la aceptación de la propuesta a través de una consulta con expertos en Informática. Se realizó un análisis de resultados mediante la evaluación de una rúbrica analítica durante el desarrollo del proceso educativo de la asignatura de Programación. Las evaluaciones se basaron en los siguientes aspectos:

- a) Planificación de la estrategia educativa.
- b) Utilización de recursos tecnológicos.
- c) Estímulo a la participación y colaboración con tecnologías digitales.
- d) Evaluación y retroalimentación proporcionada por las tecnologías digitales.
- e) Adaptabilidad y accesibilidad ofrecida por las tecnologías digitales.

**Figura 10.**

*Calificaciones docentes expertos*



**Elaborado por:** los autores

Los expertos valoraron cada componente de la estrategia, con base en las calificaciones de: deficiente (0 a 20%), regular (21 a 40%), buena (41 a 60%), muy buena (61 a 80%) y excelente (81 a 100%).

Se especifica a continuación los resultados: el Mgtr Fidel Solano, evalúa la propuesta con una calificación de 94 sobre 100 puntos, a Mgtr. Verónica Peñafiel evalúa con 98 puntos, la Mgtr. Jenny Romero califica con 94 puntos, el Mgtr Luis Castillo califica con 90 puntos; equivalente a un criterio de calificación de excelente.

En conclusión, la propuesta se valora como efectiva para el desarrollo del proceso de enseñanza-aprendizaje de la asignatura Programación para los estudiantes del segundo año de Bachillerato Técnico. Las evidencias de la evaluación se describen en el anexo de Rubrica de validación de expertos.

**Tabla 8.**

*Matriz de evaluaciones de los expertos*

Indicador	CRITERIOS DE CALIFICACIÓN		Sugerencia
	Excelente (81 a 100%)	Calificación: parcial y total	



<b>Planificación de la estrategia didáctica</b>	La planificación es excelente y se alinea con los objetivos. Además, existe integración eficiente de TICs	90	<b>Capacitación Continua para Docentes:</b> Proporcionar oportunidades de capacitación continua para los docentes en el uso efectivo de TICs para fortalecer la integración de estas tecnologías en el aula.
<b>Implementación de recursos TICs</b>	Implementación innovadora y creativa de TICs que agrega valor al proceso de aprendizaje y dispone de interacción colaborativa en el aula	95	<b>Promover la Colaboración:</b> Diseñar actividades que fomenten la interacción colaborativa entre los estudiantes y docentes utilizando las TICs.
<b>Fomento de la participación y colaboración de TICs.</b>	Efectiva integración de TICs y participación colaborativa	100	<b>Capacitar a los estudiantes en el Uso de TICs:</b> Brindar a los estudiantes la capacitación necesaria para utilizar las herramientas tecnológicas que permita desarrollar las habilidades de programación
<b>Evaluación y retroalimentación de las TICs</b>	Evaluación y retroalimentación de TICs excelentes	95	<b>Promover la Colaboración entre Pares:</b> Facilitar la retroalimentación entre pares académicos de la especialidad informática sobre el uso de las TICs, lo que favorecerá el aprendizaje y la mejora de las habilidades tecnológicas de manera colaborativa.
<b>Adaptabilidad y accesibilidad de TICs</b>	Excelente adaptabilidad y accesibilidad de las TICs que garanticen el proceso de aprendizaje	90	<b>Evaluar continuamente la efectividad de las TICs:</b> Realizar evaluaciones mensuales para determinar la eficiencia de del funcionamiento de las TICs
<b>TOTAL CALIFICACIÓN</b>		94	

**Elaborado por:** los autores

## CONCLUSIONES

El proyecto de investigación es importante para mejorar la experiencia educativa que debe aprovechar herramientas y recursos digitales para promover la comprensión y la aplicación práctica de los conceptos informáticos, desde este enfoque y en correlación con los objetivos se derivan las siguientes conclusiones:

1. Las conceptualizaciones acerca de la integración de las tecnologías digitales en el proceso de enseñanza-aprendizaje de la Programación en la especialidad Informática destacan su carácter planificado, contextualizado y reflexivo, así como la importancia de las dimensiones



pedagógica, tecnológica y política educativa para caracterizar este proceso. De igual manera se asume a la integración como fusión que posibilita destacar nuevas cualidades en los componentes del proceso de enseñanza-aprendizaje.

2. La integración de diversas teorías educativas y conceptos en el ámbito de la Informática y la Programación proporciona un enfoque holístico para el desarrollo de habilidades y conocimientos en esta disciplina. Desde la teoría del Constructivismo, que destaca el papel activo del estudiante en la construcción del conocimiento, hasta la Teoría del Aprendizaje Basado en Problemas, que fomenta la resolución de problemas como motor del aprendizaje, aportan elementos clave para el desarrollo de profesionales informáticos competentes. Asimismo, el Aprendizaje Experiencial, ofrece oportunidades prácticas para aplicar y contextualizar los conceptos teóricos, promoviendo un aprendizaje significativo y relevante. Complementando estas teorías, el Procesamiento de la Información proporciona una comprensión profunda de cómo los individuos adquieren, almacenan, recuperan y utilizan la información, lo que es fundamental en la programación para diseñar interfaces intuitivas y sistemas eficientes de almacenamiento y recuperación de datos. Así como también la Teoría del Pensamiento Complejo y la Teoría General de Sistemas, ofrecen marcos conceptuales para abordar la complejidad inherente en los sistemas informáticos, fomentando un enfoque holístico, adaptativo y creativo en el diseño y la resolución de problemas.
3. Según los resultados del diagnóstico en su mayoría los estudiantes consideran que sus profesores de Programación no los preparan adecuadamente para utilizar las tecnologías digitales en el proceso de enseñanza-aprendizaje. La gran mayoría de los estudiantes afirman que la institución rara vez o nunca evalúa o realiza un seguimiento del uso de las tecnologías digitales en las clases de Programación; las tecnologías digitales utilizadas en las clases de Programación son moderadamente eficaces para ayudar a los estudiantes a desarrollar las competencias transversales. En consideración con la efectividad de las estrategias de enseñanza utilizadas para mejorar el aprendizaje de los estudiantes, los resultados indican que la mayoría de los educadores emplean estrategias de baja efectividad, los resultados sugieren que hay áreas para mejorar en la enseñanza de Programación.
4. La propuesta se establece en base a los siguientes componentes: empleo de las tecnologías digitales estructuradas con los siguientes elementos: realización del diagnóstico de las



condiciones acerca de la integración de las tecnologías de digitales en el proceso de enseñanza-aprendizaje, diseño de actividades y determinación de las condiciones, infraestructura y recursos a utilizar, acceso equitativo a recursos tecnológicos y utilización de plataformas de aprendizaje en online y offline.

5. La valoración por cuatro expertos en la asignatura Programación en la especialidad Informática evidenciaron las posibilidades de continuar aplicando la estrategia que se propone teniendo en cuenta el contexto, ya que los expertos otorgaron calificaciones positivas en todos los casos.

### RECOMENDACIONES

Con base en las conclusiones derivadas de esta investigación realizamos las siguientes recomendaciones:

1. Recomendar que en la planeación curricular institucional los docentes se centren en planificar de manera reflexiva y constructivista la integración de tecnologías digitales en el aula, teniendo en cuenta los aspectos pedagógicos, tecnológicos y político-educativos para una implementación exitosa.
2. Proponer a la institución educativa que ofrezca formación continua a los docentes sobre el uso efectivo de las tecnologías digitales y a establecer mecanismos de evaluación y seguimiento del uso de la tecnología en el aula para mejorar la calidad del proceso de enseñanza-aprendizaje.
3. Continuar evaluando y mejorando el proceso de integración de tecnologías digitales en la enseñanza de la Programación, utilizando la sistematización de experiencias para ajustar y perfeccionar la estrategia didáctica propuesta.



## REFERENCIAS BIBLIOGRÁFICAS

- Arduino. (14 de enero de 2024). *One Compiler*. Recuperado el 02 de abril de 2024, de <https://onecompiler.com/cpp/3ygvuaj5c>
- Arias, F. (2012). *El proyecto de investigación: Introducción a la metodología científica* (Sexta ed.). Caracas, Venezuela: Episteme C.A.
- Asamblea Constituyente del Ecuador. (2008). Constitución de la República del Ecuador. *Registro Oficial, 449, 20-10*. Quito, Ecuador: Editora Nacional.
- Asamblea Nacional de la República del Ecuador. (2011). Ley Orgánica de Educación Intercultural (LOEI). *Segundo Suplemento - Registro Oficial N° 417*. Quito, Ecuador: Editora Nacional.
- Bernal, C. A. (2010). *Metodología de la investigación: administración, economía, humanidades y ciencias sociales*. México D.F: Pearson.
- Bertalanffy, L. V. (1976). *Teoría General de los Sistemas: fundamentos, desarrollos y aplicaciones*. México: Fondo de cultura Económica .
- Bravo, A., & Suástegue, S. (2022). Herramientas Digitales para el Desarrollo de la Motivación en el Aprendizaje de Matemática del Nivel Básico Superior. *Polo del Conocimiento* , 7(6), 372-397.
- Cabrera Ramos, J. (2013). *Las herramientas tecnológicas simples en el proceso de enseñanza-aprendizaje. Preparación Pedagógica para Profesores de la Nueva Universidad* . La Habana: Editorial Félix Varela.
- Cedeño Pincay, F., & Zambrano Sornoza, J. (2023). Integración de las Tecnologías de Información y Comunicación en el proceso de enseñanza-aprendizaje. *Revista cognosis, III(Especial 1)*, 73-96.



Cxxdroid. (11 de enero de 2024). *IDE del compilador*. Recuperado el 02 de abril de 2024, de [https://play.google.com/store/apps/details?id=ru.iiec.cxxdroid&hl=es\\_AR&pli=1](https://play.google.com/store/apps/details?id=ru.iiec.cxxdroid&hl=es_AR&pli=1)

Dorado Ceballos, C. (2021). La evaluación de la educación en tecnología e informática en instituciones educativas del sur de Colombia y norte del Ecuador. *Conrado*, 17(79), 163-168.

Espinar Álava, E., & Viguera Moreno, J. (2020). El aprendizaje experiencial y su impacto en la educación actual. *Revista Cubana de Educación Superior*, 39(3), 1-14.

Flores, V. (2021). Aprendizaje significativo con estrategia de enseñanza activa para un curso de proyecto software. Una experiencia en el norte de Chile. *Ingeniare. Revista chilena de ingeniería*, 29(1), 120-128.

GDB. (10 de enero de 2024). *Online GDB*. Recuperado el 02 de abril de 2024, de [https://www.onlinegdb.com/online\\_cplusplus\\_compiler](https://www.onlinegdb.com/online_cplusplus_compiler)

Gualsaqui Miranda., L. (2016). Las estrategias didácticas apoyadas en las tecnologías de la información y comunicación y el proceso de enseñanza-aprendizaje de la asignatura de informática en los primeros cursos de bachillerato general unificado del régimen costa del Liceo Naval de Qui. [*Tesis de maestría*]. Universidad Técnica de Ambato. Facultad de Ciencias Humanas y de la Educación. Carrera de Cultura Física, Ambato, Ecuador.

Guamán Gómez, V., & Espinoza Freire, E. (2022). Aprendizaje basado en problemas para el proceso de enseñanza-aprendizaje. *Revista Universidad y Sociedad*, 14(2), 124-131.

Hernández Hernández, N., & Garnica González, J. (2015). Árbol de Problemas del Análisis al Diseño y Desarrollo de Productos. *Conciencia tecnológica*, 50(1), 38-46.

Hernández Sampieri, R., Fernández Collado, C., & Baptista Lucio, P. (2014). *Metodología de la investigación*. México: Mc Graw Hill .

IDE. (09 de enero de 2024). *Entorno de desarrollo (IDE) para Windows*. Recuperado el 02 de abril de 2024, de <https://www.bloodshed.net/>



Jdoodle. (08 de enero de 2024). *CCompilador IDE*. Recuperado el 02 de abril de 2024, de <https://www.jdoodle.com/c-online-compiler>

Kolb, D. A. (2014). *Experiential Learning: Experience as the Source of Learning and Development*. Estados Unidos: Pearson Education.

Martínez Alvarado, H. (2021). *La integración de las TIC en instituciones. Los desafíos de las TIC para el cambio educativo. Organización de Estados Iberoamericanos para la Educación, la Ciencia y la Cultura (OEI). Colección Metas Educativas. .* Fundación Santillana.

Morin, E. (1995). *El pensamiento complejo*. Madrid: Gedisa.

Morinigo, C., & Fenner, I. (2021). Teorías del aprendizaje. *Minerva Magazine of Science*, 9(2), 1-36.

Ordoñez Ocampos, B. P., Ochoa Romero, M. E., & Espinoza Freire, E. E. (2020). El constructivismo y su prevalencia en el proceso de enseñanza-aprendizaje en la educación básica en Machala. Caso de estudio. *Revista Metropolitana de Ciencias Aplicadas*, 3(3), 24-31.

Otero Ortega, A. (2018). Enfoques de investigación. Métodos para el diseño urbano–Arquitectónico, 1, . *Diseño Urbano*, 9-10.

Pseint. (12 de enero de 2024). *PSeInt es una herramienta de programación*. Recuperado el 02 de abril de 2024, de <https://pseint.sourceforge.net/?page=descargas.php>

Ramos Lovón, W. (2019). *Teoría de la computación*.

Robles Garrote, P., & Rojas, M. (2015). La validación por juicio de expertos: dos investigaciones cualitativas en Lingüística aplicada. *Revista Nebrija de lingüística aplicada a la enseñanza de lenguas*, 18(2), 124-139.

Rodríguez, J., Huamani, R., Deneri, E., Ramos, D., & Rodríguez, M. (2023). Innovación educativa en acción: herramientas digitales y su impacto en la motivación de estudiantes



universitarios. *Horizontes. Revista de Investigación en Ciencias de la Educación*, 7(30), 1739 - 1751.

Santos Baranda, J., & Armasa Velasco, C. (2020). La integración de las Tecnologías de la Información y la Comunicación en los procesos formativos universitarios. *Trabajo presentado en el evento Curso Pre-Congreso Universidad (Vol. 2020), Placio de convenciones Curso 15, pre Congreso INternacional* . In La Habana, Cuba.

Scratch. (06 de enero de 2024). *Comunidad de programación*. Recuperado el 02 de abril de 2024, de <https://scratch.mit.edu/>

Sledos. (13 de enero de 2024). *SLE2 Lenguaje Para Apoyar La Formación Profesional De Estudiantes De Informática*. Recuperado el 02 de abril de 2024, de <https://sledos.blogspot.com/p/descarga.html>

Solano Hernández, E. (2023). Estrategia metodológica para la integración de las TIC en los procesos de enseñanza-aprendizaje por parte de docentes de la educación superior colombiana. [*Teis doctoral*]. Iniversitat de les Illes Balears, Palma, España.

Tafur Méndez, F., Zambrano Chamba, M., Malvacias Escalona, A., & Almao Malvacias, V. (2022). ¿Son necesarias las herramientas digitales para generar motivación en los procesos de aprendizaje? *593 Digital Publisher CEIT*, 7(2), 2(1), 56-63.

Tafur, F., Zambrano, M., Malvacias, A., & Almao, V. (2022). ¿Son necesarias las herramientas digitales para generar motivación en los procesos de aprendizaje? *593 Digital Publisher CEIT*, 2(1), 56-63.

Tejeda Marroquín, A., Macz Caal, I., Díaz Vásquez, R., & Villela Cervantes, C. (2022). El constructivismo en la era digital. *Revista Guatemalteca de Educación Superior*, 5(2), 210-220. doi:<https://doi.org/10.46954/revistages.v5i2.103>