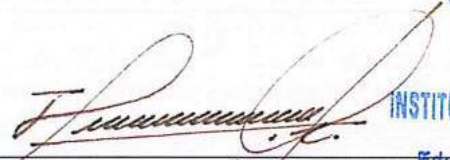





Programa de Estudios de la Unidad de Aprendizaje: ALGORITMIA																																
Clave:	3FP-FM976				Créditos:	3.37				Programa Académico: TÉCNICO EN PROGRAMACIÓN																						
										Nivel:																						
										1°		2°		3°		4°		5°		6°												
Ramas de Conocimiento										Unidades Académicas donde se Imparte:																						
Ingeniería y Ciencias Físico Matemáticas		X		Ciencias Sociales Administrativas				Ciencias Médico Biológicas				TODAS LAS U.A.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	CET1
Área de Formación Curricular										Tiempos Asignados:																						
Institucional				Científica, Humanística y Tecnológica Básica				Profesional		X		Global: <u>54</u> Hrs/18 semanas/Semestre Aula: <u>1</u> Hrs/Semana Total: <u>18</u> Hrs/Semestre Taller: <u>0</u> Hrs/Semana Total: <u>0</u> Hrs/Semestre Laboratorio: <u>2</u> Hrs/Semana Total: <u>36</u> Hrs/Semestre Otros ambientes de aprendizaje: <u>0</u> Hrs/Semana Total: <u>0</u> Hrs/Semestre																				
Tipo de Espacio																																
Aula	X	Taller		Laboratorio	X	Otros ambientes de Aprendizaje																										
Modalidad																																
Escolarizada	X	No Escolarizada		Mixta																												
Vigencia a Partir:		AGOSTO 2023																														
Proceso de Diseño y Autorización:										Organización																						
										Por Unidad de Aprendizaje:		X		Por Área:				Por Módulo:														
										Firma y Sello de Autorización:																						
Elaborado por:	REP. ACAD. NMS		Fecha de Elaboración:	04	02	2022																										
Revisado por:	DEMS		Fecha de Revisión:	10	01	2023																										
Aprobado por:	CTCE-NMS		Fecha de Aprobación:	12	01	2023																										
Autorizado por:	CPA-CGC		Fecha de Autorización:	16	02	2023																										
										  <p>INSTITUTO POLITÉCNICO NACIONAL Dirección de Educación Media Superior</p>																						
										M. EN E.N.A. MARÍA ISABEL ROJAS RUIZ Directora de Educación Media Superior																						



Programa Académico: Técnico en Programación

Unidad de Aprendizaje: Algoritmia

FUNDAMENTACIÓN

Enseguida se presentan los argumentos que exponen las bases epistémicas y didácticas, así como la relevancia del Programa de Estudios de la unidad de aprendizaje **Algoritmia** para la formación del estudiante.

La unidad de aprendizaje Algoritmia pertenece al área de formación profesional del Bachillerato Tecnológico Bivalente del Nivel Medio Superior del Instituto Politécnico Nacional, se ubica en el tercer nivel del Plan de Estudios del Programa Académico **Técnico en Programación** y se imparte en la modalidad escolarizada, de manera **obligatoria** en la rama del conocimiento de Ingeniería y Ciencias Físico-Matemáticas.

Esta unidad de aprendizaje coadyuva a crear programas computacionales utilizando como una dimensión científica, técnica, tecnológica, social, responsable, metodológica y sustentable, que incentiva la adquisición, desarrollo y la lógica de la programación aplicación del pensamiento lógico, el razonamiento abstracto, el pensamiento analítico, la creatividad, la imaginación, la iniciativa y diversas habilidades cognitivas. Introduce al estudiante al campo conceptual, procedimental y actitudinal para dar solución a problemas de programación en diversos contextos, considerando tanto los principios y dimensiones del desarrollo humano sustentable como la perspectiva de género. La adquisición de estas destrezas y habilidades relacionadas con el pensamiento eficaz favorecerán en el estudiante el desarrollo de una visión crítica y holística, cuya puesta en práctica, en forma autónoma, en el futuro le coadyuvará a responder en forma eficiente y eficaz a los retos que se le presenten cuando se incorpore a estudios superiores o al campo laboral.

La unidad de aprendizaje Algoritmia contribuye al desarrollo del Talento 4.0 requerido por la Industria 4.0 y para la transformación del país, orientada al logro del desarrollo humano sustentable de México. Esto debido a que docente adquirirá, desarrollará y aplicará conceptos, técnicas y métodos que favorecen el desarrollo y fortalecimiento del pensamiento lógico y algorítmico, que es un instrumento estructural para organizar y analizar datos y procesos en forma lógica, representar algoritmos mediante abstracciones, como los diagramas de flujo, pseudocódigos y códigos para dar soluciones al ser ejecutadas en una computadora; así como automatizar procesos de cálculo, búsqueda y ordenamiento, y que son las bases para el desarrollo de aplicaciones de software que usa la inteligencia artificial.

Algoritmia es una unidad de aprendizaje enfocada al desarrollo de habilidades técnicas, cognitivas y socioemocionales inherentes al estudio, análisis y aplicación de algoritmos, así como los elementos de un lenguaje de programación, con la finalidad de desarrollar aplicaciones de software, orientadas a solucionar problemas en diferentes contextos.

La unidad de aprendizaje Algoritmia estará fundamentada en el Modelo Educativo Institucional vigente y en la Educación para la industria 4.0, por esto, se emplearán metodologías didácticas activas como el Aprendizaje Basado en Problemas, casos, escenarios, ejercicios, Aula Invertida, Aprendizaje Basado en la Colaboración y el Diálogo, entre otras; esto con el propósito de que el estudiante desarrolle competencias del siglo XXI, como el trabajo colaborativo, trabajo en equipo, reto al cambio, autodirección, resolución de problemas cercanos a la realidad, autogestión del aprendizaje y resiliencia. Además, se emplearán herramientas tecnológicas especializadas para compilar y ejecutar códigos de programación, diseño de diagramas de flujo y plataformas digitales que fomentarán la colaboración e interacción presenciales y virtuales, en forma sincrónica o asincrónica, que corresponden a la Educación 4.0.





Programa Académico: Técnico en Programación

Unidad de Aprendizaje: Algoritmia

El rol del docente será de mediador entre el estudiante y los contenidos didácticos a abordar, puesto que se centrará en la planeación, diseño, creación y organización de estrategias y contenidos didácticos, mientras que el docente auxiliar apoyará en la supervisión y mediación de los espacios de trabajo, incluidos los ciberespacios en plataformas digitales, atendiendo las necesidades técnicas, de conocimientos, apoyo logístico y metodológico en los procesos de aprendizaje individual y grupal, con el objetivo de generar ambientes que favorezcan la educación inclusiva, flexible, sustentable y con perspectiva de género.

El estudiante desarrollará un trabajo autónomo en diferentes ambientes de aprendizaje, organizará su trabajo de manera independiente y articulará saberes de diversos campos del conocimiento, que le posibilitarán construir y expresar su propio conocimiento en beneficio de la sociedad; también adquirirá habilidades tanto tecnológicas como personales que promoverán la comunicación asertiva, la creatividad, la negociación, la gestión del tiempo, la motivación, el liderazgo y la responsabilidad social vinculada a la protección del medio ambiente, la erradicación de toda manifestación de violencia de género, la inclusión y la accesibilidad.

La evaluación se efectuará en el marco de la evaluación auténtica, por esto, comprenderá tres momentos: diagnóstica, formativa y sumativa. La evaluación diagnóstica se llevará a cabo mediante un cuestionario informatizado con evaluación y retroalimentación automatizadas, la finalidad de que el docente efectúe los ajustes didácticos pertinentes y si es necesario, el estudiante nivele los conocimientos previos adquiridos en otras unidades de aprendizaje para que establezca conexiones significativas con la propuesta didáctica de la unidad de aprendizaje Algoritmia. Un segundo momento de la evaluación hace referencia a la evaluación formativa, que se desarrollará a lo largo del proceso de enseñanza-aprendizaje mediante las secuencias didácticas y actividades de aprendizaje formativas que estimulen el aprendizaje activo y significativo del estudiante. Este momento se enriquecerá con diversos tipos de evaluación, como la autoevaluación y la coevaluación, puesto que coadyuvarán a dar seguimiento al desarrollo de los saberes y habilidades en contexto. Cabe señalar que estas clases de evaluación serán reforzadas a través de la retroalimentación efectiva y oportuna.

En el tercer momento de la evaluación, con fines de acreditación, se diseñarán situaciones integradoras que permitan recuperar el nivel de logro y conducir al estudiante a la metacognición en la unidad de aprendizaje Algoritmia, esto mediante evidencias de conocimiento, producto y desempeño, como diseño de algoritmos, elaboración de diagramas de flujo, pseudocódigos, códigos, entre otras evidencias de aprendizaje, cuyos criterios, aspectos e indicadores serán conocidos por los estudiantes en forma previa. Las evidencias de evaluación formativa e integradora mostrarán el saber hacer de manera reflexiva de los estudiantes, utilizando el conocimiento que van adquiriendo durante el proceso didáctico para luego transferir ese aprendizaje a situaciones similares y diferentes, en contextos escolares, sociales y laborales.

Con base en la flexibilidad curricular y en el reconocimiento de aprendizajes múltiples, también podrá aplicarse una evaluación para verificar que el estudiante domina los saberes y propósitos de Algoritmia, previo a su inicio. De esa forma, el programa de estudios de esta unidad de aprendizaje tiene una naturaleza normativa, puesto que establece los estándares para el desarrollo de conocimientos, habilidades prácticas del área de formación, habilidades socioemocionales, actitudes y valores.



Programa Académico: Técnico en Programación

Unidad de Aprendizaje: Algoritmia

► DESCRIPCIÓN DE LA UNIDAD DE APRENDIZAJE ◀

Unidad de Aprendizaje: Algoritmia		
Construye algoritmos, diagramas de flujo, pseudocódigos y códigos para dar solución a diversos problemas de programación mediante el pensamiento lógico y algorítmico de forma eficaz, eficiente, creativa y con un enfoque sustentable.		
Unidad 1: Pensamiento lógico y algorítmico		
Unidad de competencia	Aprendizajes esperados	Contenidos de aprendizaje
Representa algoritmos para expresar situaciones cotidianas y dar solución a problemas de diversos contextos con base en el pensamiento lógico y algorítmico de forma creativa y responsable.	1.1 Contextualiza el pensamiento lógico aplicado a computadoras, para conocer su uso en diversos contextos de acuerdo con experiencias vividas, de forma crítica y responsable.	<p>Conceptual:</p> <ol style="list-style-type: none"> Tipos de pensamiento <ol style="list-style-type: none"> Pensamiento lógico, relacional, deductivo, secuencial Procesos de pensamiento lógico, sistémico, algorítmico Pensamiento computacional <p>Procedimental: Procesos y ejecución de tareas. Desarrolla presentaciones.</p> <p>Actitudinal: Toma conciencia de la aplicación de algoritmos. Aprecia el trabajo de sus compañeros. Disposición para la interacción y el intercambio de ideas.</p>
	1.2 Analiza las partes de un problema, para brindar propuestas de solución a problemáticas en diversas situaciones, con base en la definición, análisis, pensamiento lógico y algorítmico, de manera eficiente y responsable.	<p>Conceptual:</p> <ol style="list-style-type: none"> El problema y sus partes <ol style="list-style-type: none"> Definición de Problema Partes de un problema Variables Interacción de datos de entrada Procesamiento de datos en la solución del problema Interacción de datos de salida





Programa Académico: Técnico en Programación

Unidad de Aprendizaje: Algoritmia

		<p>Procedimental: Expresa de forma lógica la secuencia de pasos a seguir para resolver problemas. Aplica de manera creativa alternativas de solución por medio del pensamiento lógico.</p> <p>Actitudinal: Valora situaciones para aplicar algoritmos. Comparte experiencias de situaciones cotidianas. Aplica creatividad en la determinación de alternativas. Desarrolla Interés cognitivo en la solución de problemas</p>
Unidad 2: Construcción de algoritmos		
Unidad de competencia	Aprendizajes esperados	Contenidos de aprendizaje
Elabora algoritmos, diagramas de flujo, pseudocódigos y códigos para representar paso a paso el proceso de ejecución de un algoritmo con base en variables, operadores, estructuras de control, lenguaje de programación y elementos de almacenamiento en memoria de forma analítica y creativa.	2.1 Realiza operaciones para resolver problemas, en función de un lenguaje de programación, datos de entrada, tipos de datos, variables, operadores aritméticos, de incremento, decremento, lógicos y de asignación de manera eficiente y creativa.	<p>Conceptual:</p> <ol style="list-style-type: none"> 1. Lenguaje de programación <ol style="list-style-type: none"> 1.1 Elementos de un lenguaje de programación 1.2 Identificador 1.3 Datos numéricos 1.4 Datos lógicos o booleanos 1.5 Caracteres 1.6 Variable 1.7 Constante 1.8 Operadores <ol style="list-style-type: none"> 1.8.1 aritméticos 1.8.2 incremento 1.8.3 decremento 1.8.4 relacional 1.8.5 lógico 1.8.6 asignación <p>Procedimental: Experimenta operaciones con distintos tipos de datos y operadores. Determina la aplicación de los tipos de datos de acuerdo a los requerimientos del problema a resolver</p> <p>Actitudinal: Comunicación asertiva</p>





Programa Académico: Técnico en Programación

Unidad de Aprendizaje: Algoritmia

		<p>Disposición para la interacción y el intercambio de ideas Tolerancia a la frustración Compromiso Disposición</p>
	<p>2.2 Construye diagramas de flujo, pseudocódigos y códigos para representar el flujo de ejecución de un algoritmo, con base en algoritmos, estructuras de control, elementos de almacenamiento en memoria, funciones, procedimientos y un lenguaje de programación de manera analítica, eficiente y creativa.</p>	<p>Conceptual: 1. Diagramas de flujo 1.1 Símbolos para representar inicio, fin, flujo, procesos, toma de decisiones 1.2 Algoritmos considerando bucles</p> <p>2. Estructuras de control 2.1 If-Else 2.2 While 2.3 Do 2.4 Do-while 2.5 For 2.6 Switch</p> <p>Procedimental: Resuelve problemas creando diagramas de flujo, pseudocódigos y códigos con base en algoritmos, empleado estructuras de control.</p> <p>Actitudinal: Comunicación asertiva Disposición para la interacción y el intercambio de ideas Tolerancia a la frustración Compromiso Disposición</p>
Unidad 3: Búsqueda y ordenamiento		
Unidad de competencia	Aprendizajes esperados	Contenidos de aprendizaje
<p>Construye códigos computacionales para agilizar procesos, búsqueda y ordenamiento de información, con base en técnicas y</p>	<p>3.1 Codifica algoritmos para agilizar procesos y resolver problemas mediante las técnicas de recursividad de forma eficaz y eficiente.</p>	<p>Conceptual: 1. Códigos computacionales para agilizar procesos 1.1 Recursividad 1.2 Iteratividad 1.3 Recursividad directa 1.4 Recursividad indirecta 1.5 Árboles</p>





Programa Académico: Técnico en Programación

Unidad de Aprendizaje: Algoritmia

<p>estrategias de manera eficaz y eficiente.</p>		<p>1.6 Árboles binarios</p> <p>Procedimental: Diseñar métodos recursivos para agilizar procesos</p> <p>Actitudinal: Se dirige con iniciativa y entusiasmo Tolerancia a enfrentarse a situaciones ambiguas</p>
	<p>3.2 Construye programas para ejecutar procesos de ordenamiento y búsqueda con base en algoritmos de selección, ordenación por inserción, burbuja, shell, búsqueda secuencial y binaria de forma eficaz y eficiente.</p>	<p>Conceptual: 1. Técnicas de búsqueda y ordenamiento 1.1 Algoritmo de búsqueda 1.2 Algoritmo de ordenamiento 1.3 Ordenación por inserción 1.4 Método Burbuja, Shell 1.5 Búsqueda secuencial 1.6 Búsqueda binaria</p> <p>Procedimental: Programar algoritmos de búsqueda y ordenamiento</p> <p>Actitudinal: Tolerancia a enfrentarse a situaciones ambiguas Habilidad para resolver problemas de forma ética y creativa</p>



Programa Académico: Técnico en Programación

Unidad de Aprendizaje: Algoritmia

➔ **MATRIZ DE VINCULACIÓN** ◀

COMPETENCIAS PARA EL SIGLO XXI HABILIDADES BLANDAS Y SOCIOEMOCIONALES	Unidad de Competencia 1		Unidad de Competencia 2		Unidad de Competencia 3	
	AE 1	AE 2	AE 1	AE 2	AE 1	AE 2
Trabajo en equipo					X	X
Creatividad	X	X	X	X	X	X
Pasión	X	X	X	X	X	X
Adaptabilidad			X	X	X	X
Empatía	X	X	X	X	X	X
Pensamiento crítico			X	X	X	X
Liderazgo					X	X
Innovación			X	X	X	X
Proactividad	X	X	X	X	X	X
Inteligencia emocional					X	X
Manejo de la frustración			X	X	X	X



Programa Académico: Técnico en Programación

Unidad de Aprendizaje: Algoritmia

PERFIL DOCENTE

El profesor que imparta la Unidad de Aprendizaje Algoritmia contará con las habilidades en el manejo de los saberes disciplinares y/o profesionales, así como su disposición, autoridad y tolerancia en el manejo de grupos de aprendizaje. Por lo tanto, debe poseer las habilidades que favorezcan el desarrollo del talento 4.0.

Habilidades docentes en el desarrollo del Talento

En el campo de su especialización:

- Habilidades y conocimientos profesionales que se requiere para la impartición de la Unidad de Aprendizaje.
- Adquirir habilidades digitales, desarrollarlas y actualizarlas.

En el campo pedagógico:

1. Fomentar procesos de enseñanza que le permitan interpretar y resolver las necesidades de aprendizaje de los alumnos, tomando en cuenta sus capacidades, habilidades, vocación e intereses.
2. Desarrollar procesos de enseñanza aprendizaje, utilizando métodos basados en administración de proyectos reales, aprovechando espacios educativos distintos a las aulas, para mejorar la calidad y pertinencia de la enseñanza.

En el campo de la investigación:

- Fortalecer el trabajo académico a partir del aprovechamiento de los resultados y productos de los proyectos de investigación

Perfil Profesional

1. Licenciado Titulado en Ing. En Sistemas Computacionales o a fin o Maestría en computación con experiencia de dos años en el área docente
- Experiencia comprobable de tres años en la iniciativa pública o privada aplicando los conocimientos de la unidad de aprendizaje

La unidad de aprendizaje Algoritmia es de las primeras bases que se sientan para poder comenzar a programar, debido al tamaño de los grupos y que se necesita de una supervisión más personalizada se requiere de tres titulares, ya que existen dos horas en laboratorios en esta unidad de aprendizaje de algoritmia donde aplicarán lo teórico a la práctica y cada docente podrá atender varios alumnos, cubriendo la totalidad del grupo, de lo contrario el estudiante se atrasa, no resuelve sus dudas, originando que su aprendizaje no sea el adecuado, con la repercusión en las otras unidades de aprendizaje del mismo nivel y las siguientes unidades de aprendizaje del cuarto nivel, siendo pertinente incluir los tres auxiliares para apoyar al docente titular.



Programa Académico: Técnico en Programación

Unidad de Aprendizaje: Algoritmia

ESTRUCTURA DIDÁCTICA

Unidad Didáctica 1:	Pensamiento lógico y algorítmico	Nivel:	Tercer
Propósito General:	Construye algoritmos, diagramas de flujo, pseudocódigos y códigos para dar solución a diversos problemas de programación mediante el pensamiento lógico y algorítmico de forma eficaz, eficiente, creativa y con un enfoque sustentable.		
Unidad de Competencia No 1:	Representa algoritmos para expresar situaciones cotidianas y dar solución a problemas de diversos contextos con base en el pensamiento lógico y algorítmico de forma creativa y responsable.		
Aprendizaje Esperado No 1:	Contextualiza el pensamiento lógico aplicado a computadoras, para conocer su uso en diversos contextos de acuerdo con experiencias vividas, de forma crítica y responsable.	Tiempo estimado para obtener el Aprendizaje Esperado:	9 horas

Contenidos de Aprendizaje

Conceptuales	Procedimentales	Actitudinales
1. Tipos de pensamiento. 1.1. Pensamiento lógico, relacional, deductivo, secuencial. 1.2. Procesos de pensamiento lógico, sistémico, algorítmico. 1.3. Pensamiento computacional	Procesos y ejecución de tareas Desarrolla presentaciones	Toma conciencia de la aplicación de algoritmos Aprecia el trabajo de sus compañeros Disposición para la interacción y el intercambio de ideas

Estrategia Didáctica y Ambiente de Aprendizaje

Para lograr el aprendizaje esperado de esta unidad de competencia se propone utilizar el Aprendizaje Basado en Problemas, además de integrar recursos digitales para guiar el aprendizaje del estudiante, hacer uso de la computadora y aplicaciones en Internet, así como la publicación de guías y manuales.

Apertura: los estudiantes en acompañamiento de los docentes contextualizan las características del pensamiento lógico, relacional, deductivo, secuencial, así como los procesos de pensamiento lógico, sistémico, algorítmico y la relación que tiene con la tecnología y la sociedad.

Desarrollo: el docente coordinará una serie de foros de discusión en el que los estudiantes podrán exponer, mediante presentaciones, sus puntos de vista en diversas situaciones y experiencias de vida, además el docente sugerirá una serie de casos o problemas para reforzar el tema.

Cierre: al terminar la solución de los casos o problemas, se debe promover la reflexión y el intercambio de ideas a través de foros presenciales o en línea y las evidencias deberán quedar respaldadas de forma individual en un portafolio de evidencias.





Programa Académico: Técnico en Programación

Unidad de Aprendizaje: Algoritmia

Herramientas Tecnológicas y Recursos Didácticos	Evidencia de Aprendizaje para la Evaluación Formativa	Instrumento y Criterios de Evaluación
<p>Equipo de cómputo Plataforma virtual Software orientado a la unidad de competencia Internet Proyector Videos Repositorios de contenidos y material de consulta Infografías Repositorio digital.</p>	<p>Portafolio de evidencias con las soluciones de los casos, problemas y reflexión personal.</p>	<p>Instrumento de evaluación:</p> <ul style="list-style-type: none"> • Lista de cotejo <p>Criterios de evaluación de forma:</p> <ul style="list-style-type: none"> • El portafolio de evidencias cuenta con una portada con los datos de identificación del estudiante. • Los archivos se encuentran ordenados y con la nomenclatura indicada. • Los textos están creados con fuente Arial 12, interlineado intermedio. <p>Criterios de evaluación de fondo:</p> <ul style="list-style-type: none"> • La reflexión está conformada por una introducción, desarrollo y conclusión. • La redacción a la reflexión mantiene un tono académico y profesional, crítica y responsable. • Aplica el pensamiento lógico para la solución a los problemas y casos.



Programa Académico: Técnico en Programación

Unidad de Aprendizaje: Algoritmia

Unidad Didáctica 1:	Pensamiento lógico y algorítmico	Nivel:	Tercer
Propósito General:	Construye algoritmos, diagramas de flujo, pseudocódigos y códigos para dar solución a diversos problemas de programación mediante el pensamiento lógico y algorítmico de forma eficaz, eficiente, creativa y con un enfoque sustentable.		
Unidad de Competencia No 1:	Representa algoritmos para expresar situaciones cotidianas y dar solución a problemas de diversos contextos con base en el pensamiento lógico y algorítmico de forma creativa y responsable.		
Aprendizaje Esperado No 2:	Analiza las partes de un problema, para brindar propuestas de solución a problemáticas en diversas situaciones, con base en la definición, análisis, pensamiento lógico y algorítmico, de manera eficiente y responsable	Tiempo estimado para obtener el Aprendizaje Esperado:	9 horas

Contenidos de Aprendizaje

Conceptuales	Procedimentales	Actitudinales
1. El problema y sus partes 1.1 Definición de Problema 1.2 Partes de un problema 1.3 Variables 1.4 Interacción de datos de entrada 1.5 Procesamiento de datos en la solución del problema 1.6 Interacción de datos de salida	Expresa de forma lógica la secuencia de pasos a seguir para resolver problemas. Ejercicio 1. Contextualización del pensamiento lógico Aplica de manera creativa alternativas de solución por medio del pensamiento lógico.	Valora situaciones para aplicar algoritmos. Comparte experiencias de situaciones cotidianas. Aplica creatividad en la determinación de alternativas. Desarrolla Interés cognitivo en la solución de problemas.

Estrategia Didáctica y Ambiente de Aprendizaje

Para lograr el aprendizaje esperado de esta unidad de competencia se propone utilizar el Aprendizaje Basado en Problemas, además de integrar recursos digitales para guiar el aprendizaje del estudiante, hacer uso de la computadora y aplicaciones en Internet, así como la publicación de guías y manuales haciendo uso de una plataforma virtual

Apertura: los estudiantes en acompañamiento de los docentes identifican las partes de un problema, variables, datos de entrada y de salida, comprenden la definición de análisis y la relación que tiene con el pensamiento lógico y algorítmico en la solución de problemas.

Desarrollo: el docente titular mostrará ejemplos de la solución a problemas empleando el pensamiento lógico y algorítmico para que posteriormente el estudiante por medio de ésta resuelva problemas aplicados a situaciones cotidianas, considerando datos de entrada, variables, procesamiento, interacciones y datos de salida las actividades podrán ser individual o colaborativas, ya sea presencial o por medio de una plataforma de aprendizaje.

Cierre: al terminar la solución de los casos o problemas, se debe promover la reflexión y el intercambio de ideas a través de foros presenciales o en línea y las evidencias deberán quedar respaldadas de forma individual en un portafolio de evidencias.





Programa Académico: Técnico en Programación

Unidad de Aprendizaje: Algoritmia

Herramientas Tecnológicas y Recursos Didácticos	Evidencia de Aprendizaje para la Evaluación Formativa	Instrumento y Criterios de Evaluación
<p>Equipo de cómputo Software orientado a la unidad de competencia Internet Proyector Videos Repositorios de contenidos y material de consulta Infografías Plataforma virtual Repositorio digital</p>	<p>Repositorio digital con la solución a problemas</p>	<p>Instrumento de evaluación:</p> <ul style="list-style-type: none"> • Lista de cotejo <p>Criterios de evaluación de forma:</p> <ul style="list-style-type: none"> • El repositorio digital contiene la solución a cada uno de los problemas propuestos. • Existe una portada con los datos de identificación del estudiante. • Los archivos se encuentran ordenados y con la nomenclatura indicada. • Los textos están creados con fuente Arial 12, interlineado intermedio. <p>Criterios de evaluación de fondo:</p> <ul style="list-style-type: none"> • Identifica variables • Aplica el pensamiento lógico para la solución a los problemas y casos. • Se observa claramente el proceso de solución a los problemas • La solución a problemas es eficiente y responsable.



Programa Académico: Técnico en Programación

Unidad de Aprendizaje: Algoritmia

Unidad Didáctica 2:	Construcción de algoritmos	Nivel:	Tercer
Propósito General:	Construye algoritmos, diagramas de flujo, pseudocódigos y códigos para dar solución a diversos problemas de programación mediante el pensamiento lógico y algorítmico de forma eficaz, eficiente, creativa y con un enfoque sustentable.		
Unidad de Competencia No 2:	Elabora algoritmos, diagramas de flujo, pseudocódigos y códigos para representar paso a paso el proceso de ejecución de un algoritmo con base en variables, operadores, estructuras de control, lenguaje de programación y elementos de almacenamiento en memoria de forma analítica y creativa		
Aprendizaje Esperado No 1:	Realiza operaciones para resolver problemas, en función de un lenguaje de programación, datos de entrada, tipos de datos, variables, operadores aritméticos, de incremento, decremento, lógicos y de asignación de manera eficiente y creativa.	Tiempo estimado para obtener el Aprendizaje Esperado:	9 horas

Contenidos de Aprendizaje

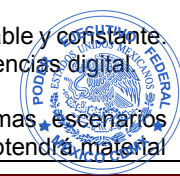
Conceptuales	Procedimentales	Actitudinales
1. Lenguaje de programación 1.1 Elementos de un lenguaje de programación 1.2 Identificador 1.3 Datos numéricos 1.4 Datos lógicos o booleanos 1.5 Caracteres 1.6 Variable 1.7 Constante 1.8 Operadores 1.8.1 aritméticos 1.8.2 incremento 1.8.3 decremento 1.8.4 relacional 1.8.5 lógico 1.8.6 asignación	Experimenta operaciones con distintos tipos de datos y operadores. Determina la aplicación de los tipos de datos de acuerdo a los requerimientos del problema a resolver. Ejercicio 2. Elaboración de operaciones para resolver problemas.	Comunicación asertiva Disposición para la interacción y el intercambio de ideas Tolerancia a la frustración Compromiso Disposición

Estrategia Didáctica y Ambiente de Aprendizaje

Se propone una metodología de Aprendizaje Basado en Problemas, solución a casos centrada en el estudiante así como ejercicios enriquecidos con recursos digitales utilizados en el desarrollo de software para acompañar el proceso de aprendizaje del estudiante, donde el estudiante podrá usar una computadora y apoyarse en manuales impresos y en línea.

Apertura: los estudiantes en compañía de los docentes abordan las características de un Lenguaje de programación, datos de entrada, tipos de datos, tipos de operadores, variable y constante. El acompañamiento presencial se dará en el aula de clases y laboratorios, mientras que el estudiante podrá usar una plataforma digital para conformar un repositorio de evidencias digital.

Desarrollo: el docente titular muestra ejemplos de la aplicación de operaciones con distintos tipos de datos y operadores. De manera individual el estudiante resuelve problemas, escenarios y ejercicios aplicando los tipos de datos y operadores de acuerdo a los requerimientos a resolver. Por medio del acceso a repositorios de contenidos orientados al tema, obtendrá material





Programa Académico: Técnico en Programación

Unidad de Aprendizaje: Algoritmia

complementario, casos adicionales de resolución de los problemas aplicados a situaciones cotidianas, con un enfoque en el uso de diversos datos en diversos requerimientos de aplicación, uso y desempeño de operadores, para proponer una solución, aplicando el pensamiento lógico y algorítmico, e incrementar la experiencia en la solución, además de ser un espacio para generar su portafolio online de tareas y actividades que se realicen durante la unidad didáctica, adicional a que podrá proponer nuevos casos en el espacio de foros y colaboración, considerando las diversas alternativas empleadas para solución alcanzada.

Cierre: el docente evalúa las soluciones entregadas por los estudiantes y los invita a hacer una demostración en la que explican al resto del grupo la manera en lo que lo hicieron.

Herramientas Tecnológicas y Recursos Didácticos	Evidencia de Aprendizaje para la Evaluación Formativa	Instrumento y Criterios de Evaluación
<p>Equipo de cómputo Software Internet Plataformas virtual Videos Repositorios de contenidos Repositorio de casos Repositorio problemas Material de consulta Guía de contenidos E-books Infografías Podcast Pintarón y Marcadores.</p>	<p>Repositorio digital con solución a problemas, casos y ejercicios.</p>	<p>Instrumento de evaluación:</p> <ul style="list-style-type: none"> • Lista de cotejo <p>Criterios de evaluación de forma:</p> <ul style="list-style-type: none"> • El repositorio digital contiene la solución a cada uno de los problemas, casos y ejercicios. • Existe una portada con los datos de identificación del estudiante. • Los archivos se encuentran ordenados y con la nomenclatura indicada. <p>Criterios de evaluación de fondo:</p> <ul style="list-style-type: none"> • Hace uso de variables y constantes. • Utiliza diversos operadores. • Aplica el pensamiento lógico para la solución a los problemas y casos. • Se observa claramente el proceso de solución a los problemas y casos. • La solución a problemas y casos es eficiente y creativa.





Programa Académico: Técnico en Programación

Unidad de Aprendizaje: Algoritmia

Unidad Didáctica 2:	Construcción de algoritmos	Nivel:	Tercer
Propósito General:	Construye algoritmos, diagramas de flujo, pseudocódigos y códigos para dar solución a diversos problemas de programación mediante el pensamiento lógico y algorítmico de forma eficaz, eficiente, creativa y con un enfoque sustentable.		
Unidad de Competencia No 2:	Elabora algoritmos, diagramas de flujo, pseudocódigos y códigos para representar paso a paso el proceso de ejecución de un algoritmo con base en variables, operadores, estructuras de control, lenguaje de programación y elementos de almacenamiento en memoria de forma analítica y creativa.		
Aprendizaje Esperado No 2:	Construye diagramas de flujo, pseudocódigos y códigos para representar el flujo de ejecución de un algoritmo, con base en algoritmos, estructuras de control, elementos de almacenamiento en memoria y un lenguaje de programación de manera analítica, eficiente y creativa.	Tiempo estimado para obtener el Aprendizaje Esperado:	9 horas

Contenidos de Aprendizaje

Conceptuales	Procedimentales	Actitudinales
1. Diagramas de flujo 1.1 Símbolos para representar inicio, fin, flujo, procesos, toma de decisiones 1.2 Algoritmos considerando bucles 2. Estructuras de control 2.1 If-Else 2.2 While 2.3 Do 2.4 Do-while 2.5 For 2.6 Switch	Resuelve problemas creando diagramas de flujo, pseudocódigos y códigos con base en algoritmos, empleando estructuras de control. Ejercicio 3. Realización de diagramas de flujo y pseudocódigos para resolver problemas.	Comunicación asertiva Disposición para la interacción y el intercambio de ideas Tolerancia a la frustración Compromiso Disposición

Estrategia Didáctica y Ambiente de Aprendizaje

Se propone una metodología de Aprendizaje Basado en Problemas, solución de casos centrada en el estudiante, así como ejercicios enriquecidos con recursos digitales y software especializado para acompañar el proceso de aprendizaje del estudiante, donde el estudiante podrá usar una computadora, software especializado y apoyarse en manuales impresos y en línea

Apertura: los estudiantes en compañía de los docentes abordan las características para crear diagramas de flujo, así como sus características y símbolos para representar el inicio, fin, flujo, procesos y toma de decisiones. El docente muestra ejemplos de la aplicación de las estructuras de control If-else, while, do, do-while, for y switch, muestra cómo crear un pseudocódigo y código para compilar un programa.





Programa Académico: Técnico en Programación

Unidad de Aprendizaje: Algoritmia

Desarrollo: El estudiante da solución de los problemas, casos y ejercicios aplicados a situaciones cotidianas que implique un uso efectivo de símbolos en diagramas de flujo, estructuras de control haciendo uso del pensamiento lógico y algorítmico para crear pseudocódigos y códigos. El estudiante obtendrá material complementario de casos adicionales de que se realicen durante las sesiones de clase, podrá proponer nuevos casos en el espacio de foros y colaboración, con un enfoque de análisis y aplicación de pensamiento lógico y algorítmico en foros de colaboración, considerando las diversas alternativas de solución.

Cierre: El estudiante conformará un portafolio de evidencias y un repositorio digital para almacenar sus evidencias, el docente evalúa las soluciones entregadas por los estudiantes y los invita a hacer una demostración en la que explican al resto del grupo la forma en que lo hicieron.

Herramientas Tecnológicas y Recursos Didácticos	Evidencia de Aprendizaje para la Evaluación Formativa	Instrumento y Criterios de Evaluación
<p>Equipo de cómputo Software especializado Plataforma virtual Internet Proyector Videos Repositorios de contenidos y material de consulta Infografías Repositorio digital.</p>	<p>Portafolio de evidencias con los algoritmos, diagramas de flujo y pseudocódigos Repositorio digital con las códigos de los programas</p>	<p>Instrumento de evaluación:</p> <ul style="list-style-type: none"> • Lista de cotejo <p>Criterios de evaluación de forma:</p> <ul style="list-style-type: none"> • El portafolio de evidencias, por cada problema o escenario contiene algoritmo, diagrama de flujo y pseudocódigo • El repositorio digital contiene el código correspondiente a cada problema o caso. • Existe una portada con los datos de identificación del estudiante. • Los archivos se encuentran ordenados y con la nomenclatura indicada. • Los textos están creados con fuente Arial 12, interlineado intermedio. <p>Criterios de evaluación de fondo:</p> <ul style="list-style-type: none"> • Los algoritmos mantienen un orden lógico • Los diagramas de flujo son representados con la simbología adecuada. • Los pseudocódigos corresponden al algoritmo y diagrama de flujo. • Los códigos tienen un orden y pueden ser compilados sin errores. • La solución a problemas y casos es analítica, eficiente y creativa.





Programa Académico: Técnico en Programación

Unidad de Aprendizaje: Algoritmia

Unidad Didáctica 3:	Búsqueda y ordenamiento	Nivel:	3
Propósito General:	Construye algoritmos, diagramas de flujo, pseudocódigos y códigos para dar solución a diversos problemas de programación mediante el pensamiento lógico y algorítmico de forma eficaz, eficiente, creativa y con un enfoque sustentable.		
Unidad de Competencia No 3:	Construye códigos computacionales para agilizar procesos, búsqueda y ordenamiento de información, con base en técnicas y estrategias de manera eficaz y eficiente.		
Aprendizaje Esperado No 1:	Codifica algoritmos para agilizar procesos y resolver problemas mediante las técnicas de recursividad de forma eficaz y eficiente.	Tiempo estimado para obtener el Aprendizaje Esperado:	9 horas

Contenidos de Aprendizaje

Conceptuales	Procedimentales	Actitudinales
1. Códigos computacionales para agilizar procesos 1.1 Recursividad 1.2 Iteratividad 1.3 Recursividad directa 1.4 Recursividad indirecta 1.5 Arboles 1.6 Arboles binarios	Diseñar métodos recursivos para agilizar procesos. Ejercicio 4. Usando recursividad para resolver problemas	Se dirige con iniciativa y entusiasmo Tolerancia a enfrentarse a situaciones ambiguas

Estrategia Didáctica y Ambiente de Aprendizaje

Se propone una metodología centrada en el estudiante, y enriquecida con recursos digitales utilizados en el desarrollo de software para acompañar el proceso de aprendizaje del estudiante, donde el estudiante podrá usar una computadora y apoyarse en manuales impresos y en línea.

Apertura: los estudiantes en compañía de los docentes abordan los conceptos de recursividad, iteratividad, recursividad directa, recursividad indirecta, arboles y arboles binarios. El acompañamiento presencial se dará en el aula de clases y laboratorios, mientras que el estudiante podrá usar una plataforma digital para conformar un portafolio de evidencias digital.

Desarrollo: el docente titular explica utilizando ejemplos de la aplicación de los métodos recursivos para optimizar las búsquedas de información, tal cual la hace un algoritmo de un navegador en internet donde existen grandes cantidades de datos.

De manera individual el estudiante resuelve prácticas, problemas y escenarios, utilizando los métodos de recursividad y empleando la menor cantidad de código posible para brindar soluciones de forma crítica y responsable.





Programa Académico: Técnico en Programación

Unidad de Aprendizaje: Algoritmia

Cierre: el docente evalúa las mejores soluciones entregadas por los estudiantes y los invita a hacer una demostración en la que explican al resto del grupo la manera en lo que lo hicieron.

Herramientas Tecnológicas y Recursos Didácticos	Evidencia de Aprendizaje para la Evaluación Formativa	Instrumento y Criterios de Evaluación
<p>Plataforma virtual Presentaciones electrónicas Software para el desarrollo de software Editores de texto Documentos digitales e impresos</p>	<p>Portafolio de evidencias de la resolución de problemas y escenarios propuestos</p>	<p>Instrumento de evaluación: Lista de cotejo</p> <p>Criterios de evaluación de forma:</p> <ul style="list-style-type: none"> • El portafolio de evidencias, por cada problema o escenario contiene algoritmo, diagrama de flujo y pseudocódigo • Existe una portada con los datos de identificación del estudiante. • Los archivos se encuentran ordenados y con la nomenclatura indicada. • Los textos están creados con fuente Arial 12, interlineado intermedio. <p>Criterios de evaluación de fondo:</p> <ul style="list-style-type: none"> • Los algoritmos mantienen un orden lógico • Los diagramas de flujo son representados con la simbología adecuada • Los pseudocódigos corresponden al algoritmo y diagrama de flujo. • Los códigos tienen un orden y pueden ser compilados sin errores • El algoritmo presentado agiliza el proceso utilizando recursividad. • Analiza problemas con árboles binarios • La solución a problemas y casos es eficaz y eficiente.





Programa Académico: Técnico en Programación

Unidad de Aprendizaje: Algoritmia

Unidad Didáctica 3:	Búsqueda y ordenamiento	Nivel:	
Propósito General:	Construye algoritmos, diagramas de flujo, pseudocódigos y códigos para dar solución a diversos problemas de programación mediante el pensamiento lógico y algorítmico de forma eficaz, eficiente, creativa y con un enfoque sustentable.		
Unidad de Competencia No 3:	Construye códigos computacionales para agilizar procesos, búsqueda y ordenamiento de información, con base en técnicas y estrategias de manera eficaz y eficiente.		
Aprendizaje Esperado No 2:	Construye programas para ejecutar procesos de ordenamiento y búsqueda con base en algoritmos de selección, ordenación por inserción, burbuja, shell, búsqueda secuencial y binaria de forma eficaz y eficiente.	Tiempo estimado para obtener el Aprendizaje Esperado:	9 horas

Contenidos de Aprendizaje

Conceptuales	Procedimentales	Actitudinales
1. Ordenamiento y búsqueda 1.1 Algoritmo de búsqueda 1.2 Algoritmo de ordenamiento 1.3 Ordenación por inserción 1.4 Método Burbuja, Shell 1.5 Búsqueda secuencial 1.6 Búsqueda binaria	Programar algoritmos de búsqueda y ordenamiento Ejercicio 5. Ordenamientos y búsquedas	Tolerancia a enfrentarse a situaciones ambiguas Habilidad para resolver problemas de forma ética y creativa

Estrategia Didáctica y Ambiente de Aprendizaje

Se propone una metodología de aprendizaje Basado en Problemas, solución a casos, escenarios y prácticas centradas en el estudiante, y enriquecida con recursos digitales utilizados en el desarrollo de software para acompañar el proceso de aprendizaje del estudiante.

Apertura: los estudiantes en compañía de los docentes abordan los conceptos, algoritmo de búsqueda, algoritmo de ordenamiento, ordenación por inserción, método burbuja, Shell, búsqueda secuencial y búsqueda binaria.

Desarrollo: el docente titular demuestra ejemplos de la aplicación de las técnicas de búsqueda los cuales son empleados en los principales motores de bases de datos que existen. De manera individual el estudiante desarrolla prácticas, resuelve problemas y escenarios, construyendo programas utilizando algunas de las técnicas ordenamiento y búsqueda, utiliza un repositorio digital para guardar sus programas.

Cierre: al terminar la solución de los casos o problemas, se debe promover la reflexión y el intercambio de ideas a través de foros presenciales o en línea y las evidencias deberán quedar respaldadas de forma individual en un portafolio de evidencias.





Programa Académico: Técnico en Programación

Unidad de Aprendizaje: Algoritmia

Herramientas Tecnológicas y Recursos Didácticos	Evidencia de Aprendizaje para la Evaluación Formativa	Instrumento y Criterios de Evaluación
<p>Plataforma virtual Presentaciones electrónicas Software para el desarrollo de software Editores de texto</p>	<p>Repositorio digital con el portafolio de evidencias de la resolución de problemas y escenarios propuestos</p>	<p>Instrumento de evaluación:</p> <ul style="list-style-type: none"> • Lista de cotejo <p>Criterios de Evaluación de forma:</p> <ul style="list-style-type: none"> • El repositorio digital se encuentra disponible • Existe una portada con los datos de identificación del estudiante. • Los archivos se encuentran ordenados y con la nomenclatura indicada. <p>Criterios de evaluación de fondo:</p> <ul style="list-style-type: none"> • Los programas mantienen un orden lógico. • Emplea tres de las siguientes técnicas ordenación por inserción Método Burbuja, Shell, Búsqueda secuencial, Búsqueda binaria en los programas que desarrolla. • Soluciona los problemas, casos y escenarios de forma eficaz y eficiente.



Programa Académico: Técnico en Programación

Unidad de Aprendizaje: Algoritmia

PRACTICAS

Nombre del ejercicio:	Contextualización del pensamiento lógico	N° del ejercicio:	1	Tiempo:	6 horas
Unidad de Competencia:	Representa algoritmos para expresar situaciones cotidianas y dar solución a problemas de diversos contextos con base en el pensamiento lógico y algorítmico de forma creativa y responsable				
Aprendizajes Esperados Relacionados con el ejercicio:	Analiza las partes de un problema, para brindar propuestas de solución a problemáticas en diversos situaciones, con base en la definición, análisis, pensamiento lógico y algorítmico, de manera eficiente y responsable.				

Contenidos de Aprendizaje Relacionados con el ejercicio

Conceptuales	Procedimentales	Actitudinales
1. El problema y sus partes 1.1 Definición de Problema 1.2 Partes de un problema 1.3 Variables 1.4 Interacción de datos de entrada 1.5 Procesamiento de datos en la solución del problema 1.6 Interacción de datos de salida	Expresa de forma lógica la secuencia de pasos a seguir para resolver problemas.	Valora situaciones para aplicar algoritmos.

Estrategia Didáctica y Ambiente de Aprendizaje

Estrategia didáctica (ABP): Se plantea una estrategia a partir de un problema, buscando darle solución, a través de actividades, prácticas y ejercicios logrando un aprendizaje significativo en los estudiantes.

Desarrollo del ejercicio:

El docente plantea el problema, escenario o caso.
Los estudiantes resuelven en equipo o parejas.

Primera parte del ejercicio:

- Analizan las partes del problema: entradas, proceso, salidas.
- Estructuran el algoritmo.
- Le proporciona orden lógico.
- Describen el algoritmo en lenguaje natural.

Segunda parte del ejercicio:

- Realizan el paso a paso de inicio a fin de la resolución del caso o escenario dado con pseudocódigo, identificando variables.
- Diagraman el algoritmo

Ambiente de Aprendizaje: Laboratorio de computación





Programa Académico: Técnico en Programación

Unidad de Aprendizaje: Algoritmia

Herramientas Tecnológicas y Recursos Didácticos	Evidencia de Aprendizaje para la Evaluación Formativa	Criterios e Instrumentos de Evaluación
<p>Herramientas Tecnológicas: Plataforma virtual, software para diagramar y texto.</p> <p>Recursos Didácticos: Problemas, casos de estudio, escenarios.</p>	<p>Resolución del ejercicio.</p>	<p>Instrumento de evaluación:</p> <ul style="list-style-type: none"> • Escala de valor <p>Criterios de evaluación de forma:</p> <ul style="list-style-type: none"> • El ejercicio cuenta con una portada con los datos de identificación del estudiante. • Los textos están creados con fuente Arial 12, interlineado intermedio. • El alumno entrega en tiempo. <p>Criterios de evaluación de fondo:</p> <ul style="list-style-type: none"> • El alumno identifica las partes del problema • El estudiante expresa la resolución del caso de forma lógica • La resolución esta por pasos de inicio a fin en secuencia lógica. • La resolución es una alternativa de solución cotidiana. • El alumno entrega la representación gráfica de forma digital correctamente.





Programa Académico: Técnico en Programación

Unidad de Aprendizaje: Algoritmia

Nombre del ejercicio:	Elaboración de operaciones para resolver problemas	N° del ejercicio:	2	Tiempo:	7 horas
Unidad de Competencia:	Elabora algoritmos, diagramas de flujo, pseudocódigos y códigos para representar paso a paso el proceso de ejecución de un algoritmo con base en variables, operadores, estructuras de control, lenguaje de programación y elementos de almacenamiento en memoria de forma analítica y creativa.				
Aprendizajes Esperados Relacionados con el ejercicio:	Realiza operaciones para resolver problemas, en función de un lenguaje de programación, datos de entrada, tipos de datos, variables, operadores aritméticos, de incremento, decremento, lógicos y de asignación de manera eficiente y creativa.				

Contenidos de Aprendizaje Relacionados con el ejercicio		
Conceptuales	Procedimentales	Actitudinales
1. Lenguaje de programación 1.3 Datos numéricos 1.4 Datos lógicos o booleanos 1.8 Operadores 1.8.1 aritméticos 1.8.2 incremento 1.8.3 decremento 1.8.4 relacional 1.8.5 lógico 1.8.6 asignación	Resuelve problemas con diversos operadores.	Tolerancia a la frustración.

Estrategia Didáctica y Ambiente de Aprendizaje

Estrategia didáctica (ABP): Se plantea una estrategia a partir de un problema, buscando darle solución, a través de actividades, ejercicios logrando un aprendizaje significativo en los estudiantes.

Desarrollo del ejercicio:
 El docente muestra los problemas, escenarios o casos de estudio con distintos tipos de datos y operadores.
 El estudiante resuelve en equipo o pares los escenarios o casos de estudio con distintos tipos de datos, operadores banderas y contadores.

- Identifica datos de entrada de acuerdo con el escenario o caso de estudio dado.
- Identifica tipos de datos a usar de acuerdo con el escenario o caso de estudio proporcionado.
- Determina las operaciones a realizar para resolver el escenario o caso de estudio mostrado.
- Incluye banderas y contadores para resolver el escenario o caso de estudio expuesto.

Ambiente de Aprendizaje: Laboratorio de computación

Herramientas Tecnológicas y Recursos Didácticos	Evidencia de Aprendizaje para la Evaluación Formativa	Criterios e Instrumentos de Evaluación
Herramientas Tecnológicas: Plataforma virtual, software para diagramar y texto. Recursos Didácticos: Casos de estudio, escenarios.	Resolución del caso	Instrumento de evaluación: <ul style="list-style-type: none"> • Escala de valor Criterios de evaluación de forma: <ul style="list-style-type: none"> • El ejercicio cuenta con una portada con los datos de identificación del estudiante.





Programa Académico: Técnico en Programación

Unidad de Aprendizaje: Algoritmia

		<ul style="list-style-type: none"> • Los textos están creados con fuente Arial 12, interlineado intermedio. • El alumno entrega en tiempo. <p>Criterios de evaluación de fondo:</p> <ul style="list-style-type: none"> • El alumno determina las variables a usar como entradas de acuerdo con el escenario o caso de estudio. • El estudiante fija los tipos de datos de acuerdo a las variables seleccionadas para la resolución. • La resolución incluye banderas y contadores. • La resolución abarca la totalidad del escenario o caso de estudio propuesto. • El alumno entrega la representación gráfica de forma digital correctamente. • El alumno hace el algoritmo usando la simbología indicada.
--	--	--





Programa Académico: Técnico en Programación

Unidad de Aprendizaje: Algoritmia

Nombre del ejercicio:	Realización de diagramas de flujo y pseudocódigos para resolver problemas.	N° del ejercicio:	3	Tiempo:	8 horas
Unidad de Competencia:	Elabora algoritmos, diagramas de flujo, pseudocódigos y códigos para representar paso a paso el proceso de ejecución de un algoritmo con base en variables, operadores, estructuras de control, lenguaje de programación y elementos de almacenamiento en memoria de forma analítica y creativa.				
Aprendizajes Esperados Relacionados con el ejercicio:	Construye diagramas de flujo, pseudocódigos y códigos para representar el flujo de ejecución de un algoritmo, con base en algoritmos, estructuras de control, elementos de almacenamiento en memoria y un lenguaje de programación de manera analítica, eficiente y creativa				

Contenidos de Aprendizaje Relacionados con el ejercicio

Conceptuales	Procedimentales	Actitudinales
1. Diagramas de flujo 1.1 Símbolos para representar inicio, fin, flujo, procesos, toma de decisiones 1.2 Algoritmos considerando bucles 2. Estructuras de control 2.1 If-Else 2.2 While 2.3 Do 2.4 Do-while 2.5 For 2.6 Switch	Resuelve problemas creando diagramas de flujo, pseudocódigos y códigos con base en algoritmos, empleando estructuras de control.	Tolerancia a la frustración.

Estrategia Didáctica y Ambiente de Aprendizaje

Estrategia didáctica (ABP): Se plantea una estrategia a partir de un problema, buscando darle solución, a través de actividades, prácticas y ejercicios logrando un aprendizaje significativo en los estudiantes.

Desarrollo del ejercicio:

El docente muestra los problemas, escenarios o casos de estudio.

El estudiante en equipos o pares resuelve el escenario o caso usando los elementos de diagramas de flujo y estructuras de control.

- Identifica datos de entrada de acuerdo con el escenario o caso de estudio.
- Identifica tipos de datos a usar de acuerdo con el escenario o caso de estudio.
- Determina las operaciones a realizar para resolver el escenario o caso de estudio presentado.
- Elabora el diagrama de flujo de la solución al caso de estudio o escenario mostrado.
- Realiza el pseudocódigo de la solución al caso de estudio o escenario proporcionado.
- Programa la solución del caso de estudio o escenario dado.





Programa Académico: Técnico en Programación

Unidad de Aprendizaje: Algoritmia

Ambiente de Aprendizaje: Laboratorio de computación		
Herramientas Tecnológicas y Recursos Didácticos	Evidencia de Aprendizaje para la Evaluación Formativa	Criterios e Instrumentos de Evaluación
<p>Herramientas Tecnológicas: Plataforma virtual, software para diagramar y texto.</p> <p>Recursos Didácticos: Casos de estudio, escenarios.</p>	Resolución del caso	<p>Instrumento de evaluación:</p> <ul style="list-style-type: none"> • Escala de valor <p>Criterios de evaluación de forma:</p> <ul style="list-style-type: none"> • El ejercicio cuenta con una portada con los datos de identificación del estudiante. • Los textos están creados con fuente Arial 12, interlineado intermedio. • El alumno entrega en tiempo. <p>Criterios de evaluación de fondo:</p> <ul style="list-style-type: none"> • El alumno determina las variables a usar como entradas de acuerdo con el escenario o caso de estudio. • El estudiante fija los tipos de datos de acuerdo con las variables seleccionadas para la resolución. • La resolución incluye estructuras de control selectivas e interactivas. • La programación de la resolución abarca la totalidad del escenario o caso de estudio propuesto.





Programa Académico: Técnico en Programación

Unidad de Aprendizaje: Algoritmia

Nombre del ejercicio:	Usando recursividad para resolver problemas	N° del ejercicio:	4	Tiempo:	7 horas
Unidad de Competencia:	Construye códigos computacionales para agilizar procesos, búsqueda y ordenamiento de información, con base en técnicas y estrategias de manera eficaz y eficiente.				
Aprendizajes Esperados Relacionados con el ejercicio:	Codifica algoritmos para agilizar procesos y resolver problemas mediante las técnicas de recursividad de forma eficaz y eficiente.				
Contenidos de Aprendizaje Relacionados con el ejercicio					
Conceptuales	Procedimentales	Actitudinales			
1. Códigos computacionales para agilizar procesos 1.1 Recursividad 1.2 Iteratividad 1.3 Recursividad directa 1.4 Recursividad indirecta 1.5 Árboles 1.6 Árboles binarios	Diseñar métodos recursivos para agilizar procesos	Tolerancia a enfrentarse a situaciones ambiguas			
Estrategia Didáctica y Ambiente de Aprendizaje					
<p>Estrategia didáctica (ABP): Se plantea una estrategia a partir de un problema, buscando darle solución, a través de actividades, prácticas y ejercicios logrando un aprendizaje significativo en los estudiantes.</p> <p>Desarrollo del ejercicio: El docente muestra los problemas, escenarios o casos de estudio. El estudiante en equipo o pares resuelve los escenarios o casos de estudio.</p> <ol style="list-style-type: none"> 1. Realiza la solución de los escenarios o casos de estudio usando recursividad, interactividad, recursividad directa, recursividad indirecta, árboles, árboles binarios. 2. Elabora el diagrama de flujo de las soluciones a los casos de estudio o escenarios. 3. Realiza el pseudocódigo de las soluciones a los casos de estudio o escenarios. 4. Programa las soluciones de los casos de estudio o escenarios. <p>Ambiente de Aprendizaje: Laboratorio de computación</p>					
Herramientas Tecnológicas y Recursos Didácticos	Evidencia de Aprendizaje para la Evaluación Formativa	Criterios e Instrumentos de Evaluación			
<p>Herramientas Tecnológicas: Plataforma virtual, software para diagramar y texto.</p> <p>Recursos Didácticos: Casos de estudio, escenarios.</p>	Resolución del caso	<p>Instrumento de evaluación:</p> <ul style="list-style-type: none"> • Escala de valor <p>Criterios de evaluación de forma:</p>			





Programa Académico: Técnico en Programación


Unidad de Aprendizaje: Algoritmia

		<ul style="list-style-type: none">• El ejercicio cuenta con una portada con los datos de identificación del estudiante.• Los textos están creados con fuente Arial 12, interlineado intermedio.• El alumno entrega en tiempo. <p>Criterios de evaluación de fondo:</p> <ul style="list-style-type: none">• La resolución incluye recursividad.• La programación de la resolución abarca la totalidad del escenario o caso de estudio propuesto.• Realiza el diagrama de flujo correctamente.• Crea en pseudocódigo la solución del caso de estudio o escenario de forma lógica.• El código corresponde con la sintaxis y reglas del lenguaje usado.
--	--	---



Programa Académico: Técnico en Programación

Unidad de Aprendizaje: Algoritmia

Nombre del ejercicio:	Ordenamientos y búsquedas	N° del ejercicio:	5	Tiempo:	8 horas
Unidad de Competencia:	Construye códigos computacionales para agilizar procesos, búsqueda y ordenamiento de información, con base en técnicas y estrategias de manera eficaz y eficiente				
Aprendizajes Esperados Relacionados con el ejercicio:	Construye programas para ejecutar procesos de ordenamiento y búsqueda con base en algoritmos de selección, ordenación por inserción, burbuja, shell, búsqueda secuencial y binaria de forma eficaz y eficiente.				
Contenidos de Aprendizaje Relacionados con el ejercicio					
Conceptuales	Procedimentales	Actitudinales			
1. Ordenamiento y búsqueda 1.1 Algoritmo de búsqueda 1.2 Algoritmo de ordenamiento 1.3 Ordenación por inserción 1.4 Método Burbuja, Shell 1.5 Búsqueda secuencial 1.6 Búsqueda binaria	Programar algoritmos de búsqueda y ordenamiento	Habilidad para resolver problemas de forma ética y creativa			
Estrategia Didáctica y Ambiente de Aprendizaje					
<p>Estrategia didáctica: Aula invertida, donde el estudiante abordara parte del ejercicio en su hogar y otra parte el laboratorio de computo.</p> <p>Desarrollo del ejercicio: El docente proporciona la teoría del ordenamiento y búsqueda a aplicar: ordenación por inserción, método burbuja, shell, búsqueda secuencial, búsqueda binaria. El estudiante en equipos o pares emplea el ordenamiento y búsqueda codificando en programas donde se use estos.</p> <ul style="list-style-type: none"> • Ocupa el ordenamiento acorde a la teoría proporcionada. • El programa permite la búsqueda de acuerdo a la teoría sumistrada. • Realiza el seudocódigo, diagrama de flujo acorde al tipo de ordenamiento y búsqueda. • Programa los algoritmos de ordenamiento y búsqueda. <p>El docente aclara dudas generales. El docente retroalimenta.</p> <p>Ambiente de Aprendizaje: Laboratorio de computación</p>					
Herramientas Tecnológicas y Recursos Didácticos	Evidencia de Aprendizaje para la Evaluación Formativa	Criterios e Instrumentos de Evaluación			
Herramientas Tecnológicas: Plataforma virtual, software para diagramar, codificar y texto.	Seudocódigo Diagrama de flujo. Código	Instrumento de evaluación: 			



Programa Académico: Técnico en Programación

Unidad de Aprendizaje: Algoritmia

<p>Recursos Didácticos: Teoría sobre ordenamiento, teoría sobre búsquedas.</p>		<ul style="list-style-type: none"> • Escala de valor <p>Criterios de evaluación de forma:</p> <ul style="list-style-type: none"> • El ejercicio cuenta con una portada con los datos de identificación del estudiante. • Los textos están creados con fuente Arial 12, interlineado intermedio. • El alumno entrega en tiempo. <p>Criterios de evaluación de fondo:</p> <ul style="list-style-type: none"> • La búsqueda se realiza de acuerdo a la teoría proporcionada. • El ordenamiento es el solicitado • Realiza seudocódigo del programa usando la búsqueda y ordenamiento planteado. • La solución empleada es lógica. • El ejercicio está completo. • El seudocódigo está representado de forma correcta. • Elabora el diagrama de flujo utilizando el ordenamiento y búsqueda solicitada. • El diagrama de flujo tiene la simbología adecuada. • El programa, seudocódigo y diagrama de flujo concuerdan. • El programa codificado está bien estructurado, es legible.
---	--	--





Programa Académico: Técnico en Programación

Unidad de Aprendizaje: Algoritmia

PLAN DE EVALUACIÓN SUMATIVA


N°	Unidad de Competencia	Evidencia Integradora	Criterios e Instrumento de Evaluación	Porcentaje de Acreditación
1	Representa algoritmos para expresar situaciones cotidianas y dar solución a problemas de diversos contextos con base en el pensamiento lógico y algorítmico de forma creativa y responsable.	Repositorio digital con la solución a problemas	Instrumento de evaluación: <ul style="list-style-type: none"> • Lista de cotejo Criterios de evaluación de forma: <ul style="list-style-type: none"> • El repositorio digital contiene la solución a cada uno de los problemas propuestos. • Existe una portada con los datos de identificación del estudiante. • Los archivos se encuentran ordenados y con la nomenclatura indicada. • Los textos están creados con fuente Arial 12, interlineado intermedio. Criterios de evaluación de fondo: <ul style="list-style-type: none"> • Identifica variables • Aplica el pensamiento lógico para la solución a los problemas y casos. • Se observa claramente el proceso de solución a los problemas • La solución a problemas es eficiente y responsable. 	30 %
2	Elabora algoritmos, diagramas de flujo, pseudocódigos y códigos para representar paso a paso el proceso de ejecución de un algoritmo con base en variables, operadores, estructuras de control, lenguaje de programación y elementos de almacenamiento en memoria de forma analítica y creativa.	Portafolio de evidencias con los algoritmos, diagramas de flujo y pseudocódigos. Repositorio digital con las códigos de los programa.	Instrumento de evaluación: <ul style="list-style-type: none"> • Lista de cotejo Criterios de evaluación de forma:	35%





Programa Académico: Técnico en Programación

Unidad de Aprendizaje: Algoritmia

			<ul style="list-style-type: none"> • El portafolio de evidencias, por cada problema o escenario contiene algoritmo, diagrama de flujo y pseudocódigo • El repositorio digital contiene el código correspondiente a cada problema o caso. • Existe una portada con los datos de identificación del estudiante. • Los archivos se encuentran ordenados y con la nomenclatura indicada. • Los textos están creados con fuente Arial 12, interlineado intermedio. <p>Criterios de evaluación de fondo:</p> <ul style="list-style-type: none"> • Los algoritmos mantienen un orden lógico • Los diagramas de flujo son representados con la simbología adecuada • Los pseudocódigos corresponden al algoritmo y diagrama de flujo • Los códigos tienen un orden y pueden ser compilados sin errores • La solución a problemas y casos es analítica, eficiente y creativa. 	
3	<p>Construye códigos computacionales para agilizar procesos, búsqueda y ordenamiento de información, con base en técnicas y estrategias de manera eficaz y eficiente.</p>	<p>Portafolio de evidencias de la resolución de problemas y escenarios propuestos</p>	<p>Instrumento de evaluación:</p> <ul style="list-style-type: none"> • Lista de cotejo <p>Criterios de evaluación de forma:</p> <ul style="list-style-type: none"> • El repositorio digital se encuentra disponible • Existe una portada con los datos de identificación del estudiante. • Los archivos se encuentran ordenados y con la nomenclatura indicada. 	<p>35 %</p> 



Programa Académico: Técnico en Programación

Unidad de Aprendizaje: Algoritmia

			<p>Criterios de evaluación de fondo:</p> <ul style="list-style-type: none"> • Los programas mantienen un orden lógico. • Emplea tres de las siguientes técnicas Ordenación por inserción Método Burbuja, Shell, Búsqueda secuencial, Búsqueda binaria en los programas que desarrolla. • Soluciona los problemas, casos y escenarios de forma eficaz y eficiente. 	
Propósito de la Unidad de Aprendizaje	Evidencia Integradora	Criterios e Instrumento de Evaluación	Porcentaje de Acreditación	
<p>Construye algoritmos, diagramas de flujo, pseudocódigos y códigos para dar solución a diversos problemas de programación mediante el pensamiento lógico y algorítmico de forma eficaz, eficiente, creativa y con un enfoque sustentable.</p>	<p>Portafolio de evidencias de la resolución de problemas y escenarios propuestos</p>	<p>Solución a problemas de programación Crea algoritmos Representa diagramas de flujo Construye pseudocódigos Crea programas con lenguaje de programación Eficaz Eficiente Creativo Pensamiento lógico Pensamiento algorítmico</p>	<p>100 %</p>	



Programa Académico: Técnico en Programación

Unidad de Aprendizaje: Algoritmia

PROGRAMA SINTÉTICO

PROPÓSITO DE LA UNIDAD DE APRENDIZAJE

Construye algoritmos, diagramas de flujo, pseudocódigos y códigos para dar solución a diversos problemas de programación mediante el pensamiento lógico y algorítmico de forma eficaz, eficiente, creativa y con un enfoque sustentable.

N°	UNIDAD DE COMPETENCIA	APRENDIZAJES ESPERADOS	CONTENIDOS DE APRENDIZAJE/SABERES
1	Representa algoritmos para expresar situaciones cotidianas y dar solución a problemas de diversos contextos con base en el pensamiento lógico y algorítmico de forma creativa y responsable.	<p>1.1 Contextualiza el pensamiento lógico aplicado a computadoras, para conocer su uso en diversos contextos de acuerdo con experiencias vividas, de forma crítica y responsable.</p> <p>1.2 Analiza las partes de un problema, para brindar propuestas de solución a problemáticas en diversas situaciones, con base en la definición, análisis, pensamiento lógico y algorítmico, de manera eficiente y responsable.</p>	<p>Conceptual</p> <p>Pensamiento lógico, relacional, deductivo, secuencial Procesos de pensamiento lógico, sistémico, algorítmico Pensamiento computacional Problema Análisis Pensamiento algorítmico Variables Interacción de datos de entrada Procesamiento de datos en la solución del problema Interacción de datos de salida</p> <p>Procedimental</p> <p>Procesos y ejecución de tareas Desarrolla presentaciones Expresar de forma lógica la secuencia de pasos a seguir para resolver problemas. Aplica de manera creativa alternativas de solución por medio del pensamiento lógico</p> <p>Actitudinal</p> <p>Toma conciencia de la aplicación de algoritmos Aprecia el trabajo de sus compañeros Disposición para la interacción y el intercambio de ideas Valora situaciones para aplicar algoritmos. Comparte experiencias de situaciones cotidianas.</p>





Programa Académico: Técnico en Programación

Unidad de Aprendizaje: Algoritmia

			<p>Aplica creatividad en la determinación de alternativas. Desarrolla Interés cognitivo en la solución de problemas</p>
<p>2</p>	<p>Elabora algoritmos, diagramas de flujo, pseudocódigos y códigos para representar paso a paso el proceso de ejecución de un algoritmo con base en variables, operadores, estructuras de control, lenguaje de programación y elementos de almacenamiento en memoria de forma analítica y creativa.</p>	<p>2.1 Realiza operaciones para resolver problemas, en función de un lenguaje de programación, datos de entrada, tipos de datos, variables, operadores aritméticos, de incremento, decremento, lógicos y de asignación de manera eficiente y creativa.</p> <p>2.2 Construye diagramas de flujo, pseudocódigos y códigos para representar el flujo de ejecución de un algoritmo, con base en algoritmos, estructuras de control, elementos de almacenamiento en memoria, funciones, procedimientos y un lenguaje de programación.</p>	<p>Conceptual</p> <p>Lenguaje de programación Identificador Datos numéricos Datos lógicos o booleanos Caracteres Variable Constante Operador aritmético, incremento, decremento, relacional, lógico y asignación Símbolos para representar inicio, fin, flujo, procesos, toma de decisiones Crea algoritmos considerando bucles Estructuras de control If-Else While Do Do-while For switch</p> <p>Procedimental</p> <p>Experimenta operaciones con distintos tipos de datos y operadores. Determina la aplicación de los tipos de datos de acuerdo a los requerimientos del problema a resolver. Resuelve problemas creando diagramas de flujo, pseudocódigos y códigos con base en algoritmos, empleado estructuras de control.</p> <p>Actitudinal</p> <p>Comunicación asertiva Disposición para la interacción y el intercambio de ideas Tolerancia a la frustración</p>





Programa Académico: Técnico en Programación

Unidad de Aprendizaje: Algoritmia

			Compromiso Comunicación asertiva
3	Construye códigos computacionales para agilizar procesos, búsqueda y ordenamiento de información, con base en técnicas y estrategias de manera eficaz y eficiente.	<p>3.1 Codifica algoritmos para agilizar procesos y resolver problemas mediante las técnicas de recursividad de forma eficaz y eficiente.</p> <p>3.2 Construye programas para ejecutar procesos de ordenamiento y búsqueda con base en algoritmos de selección, ordenación por inserción, burbuja, shell, búsqueda secuencial y binaria de forma eficaz y eficiente.</p>	<p>Conceptual</p> <p>Recursividad Iteratividad Recursividad directa Recursividad indirecta Arboles Arboles binarios Algoritmo de búsqueda Algoritmo de ordenamiento Ordenación por inserción Método burbuja, Shell Búsqueda secuencial Búsqueda binaria</p> <p>Procedimental</p> <p>Diseñar métodos recursivos para agilizar procesos Programar algoritmos de búsqueda y ordenamiento</p> <p>Actitudinal</p> <p>Se dirige con iniciativa y entusiasmo Tolerancia a enfrentarse a situaciones ambiguas Habilidad para resolver problemas de forma ética y creativa</p>



Programa Académico: Técnico en Programación

Unidad de Aprendizaje: Algoritmia

BIBLIOGRAFÍA BÁSICA Y COMPLEMENTARIA

Número y Nombre de la Unidad Didáctica	FORMATO APA	CLASIFICACIÓN	
		Básico	Consulta
1. Pensamiento lógico y algorítmico 2. Construcción de algoritmos	Domínguez V. (2017) Algoritmos y Diagramas de Flujo con Raptor. Alfaomega.	x	
2. Construcción de algoritmos	Gutiérrez R. (2020) Algoritmos Genéticos con Python. Un enfoque práctico para resolver problemas de ingeniería Alfaomega.		x
1. Pensamiento lógico y algorítmico 2. Construcción de algoritmos 3. Búsqueda y ordenamiento	Jiménez J.(2014) Fundamentos de programación - Diagramas de Flujo, Diagramas N-S, Pseudocódigo y Java. Alfaomega.	x	
1. Pensamiento lógico y algorítmico 2. Construcción de algoritmos	Méndez G. (2013) Diseño de algoritmos y su programación en c. Alfaomega	x	
1. Pensamiento lógico y algorítmico 2. Construcción de algoritmos	Rámirez F. (2007) Introducción a la Programación - Algoritmos y su Implementación en VB .NET, C#, Java y C++ . Alfaomega	x	
1. Pensamiento lógico y algorítmico 2. Construcción de algoritmos 3. Búsqueda y ordenamiento	Sznajdleder, Pablo (2012) Algoritmos a Fondo con Implementaciones en C y Java. Alfaomega	x	
1. Pensamiento lógico y algorítmico 2. Construcción de algoritmos 3. Búsqueda y ordenamiento	Vega, Augusto (2020) Análisis y Diseño de Algoritmos - implementaciones en c y pascal. Alfaomega	x	
3. Búsqueda y ordenamiento	Vickler A. (12 junio 2022), Algoritmos: Guía práctica para aprender algoritmos para principiantes (Spanish Edition), Editorial Ladoo Publishing LLC	x	
2 Construcción de algoritmos	Hu Y. (junio 2020), Fácil Aprendizaje Estructuras De Datos y Algoritmos Java: Aprenda fácilmente estructuras de datos y algoritmos gráficamente. Amazon Digital Services LLC - KDP Print US	x	

