



Instituto Politécnico Nacional
Secretaría Académica
Dirección de Educación Media Superior



"La Técnica al Servicio de la Patria"

Programa de Estudios de la Unidad de Aprendizaje:										LENGUAJE DE PROGRAMACIÓN																				
Clave:		4FP-FM1048				Créditos:		4.50				Programa Académico:					TÉCNICO EN SISTEMAS DIGITALES													
										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>72</u> Hrs/18 semanas/Semestre																				
Tipo de Espacio										Aula: <u>1</u> Hrs/Semana Total: <u>18</u> Hrs/Semestre																				
Aula	X	Taller		Laboratorio	X	Otros ambientes de Aprendizaje				Taller: <u>0</u> Hrs/Semana Total: <u>0</u> Hrs/Semestre																				
Modalidad										Laboratorio: <u>3</u> Hrs/Semana Total: <u>54</u> Hrs/Semestre																				
Escolarizada	X	No Escolarizada		Mixta						Otros ambientes de aprendizaje: <u>0</u> Hrs/Semana Total: <u>0</u> Hrs/Semestre																				
Vigencia a Partir:		ENERO 2024								Organización																				
Proceso de Diseño y Autorización:										Día	Mes	Año	Por Unidad de Aprendizaje:	X	Por Área:		Por Módulo:													
Elaborado por:										REP. ACAD. NMS	Fecha de Elaboración:	13	09	2021	Firma y Sello de Autorización:															
Revisado por:										DEMS	Fecha de Revisión:	04	09	2023																
Aprobado por:										CTCE-NMS	Fecha de Aprobación:	08	09	2023																
Autorizado por:										CPA-CGC	Fecha de Autorización:	21	09	2023																
										M. EN E.N.A. MARÍA ISABEL ROJAS RUÍZ																				
										Directora de Educación Media Superior																				

Programa Académico: Técnico en Sistemas Digitales

Unidad de Aprendizaje: Lenguaje de Programación

FUNDAMENTACIÓN

La unidad de aprendizaje Lenguaje de Programación pertenece al área de formación profesional del Bachillerato Tecnológico Bivalente del Nivel Medio Superior del Instituto Politécnico Nacional, se ubica en el cuarto nivel del plan de estudios del programa académico Técnico en sistemas digitales y se imparte en la modalidad escolarizada, de manera obligatoria en la rama del conocimiento de Ingeniería y Ciencias Físico - Matemáticas.

La unidad de aprendizaje de Lenguaje de Programación promueve el desarrollo de soluciones a través de lenguajes de alto nivel, en las dimensiones científica, técnica, tecnológica, social, responsable, metodológica y sustentable, que incentiva la adquisición, desarrollo y aplicación del pensamiento lógico, el razonamiento abstracto, el pensamiento analítico, la creatividad, la innovación, el emprendimiento y diversas habilidades cognitivas. Introduce al estudiante al campo conceptual, procedimental y actitudinal para analizar problemas, diseñar y desarrollar en los respectivos códigos aplicables a sistemas digitales, considerando tanto los principios y ejes 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.

Lenguaje de Programación es una Unidad de Aprendizaje enfocada al desarrollo de habilidades técnicas, cognitivas y socioemocionales inherentes al estudio, análisis, diseño y desarrollo de programas aplicables a soluciones en sistemas digitales, mediante la codificación y aplicación de soluciones con metodología estructurada con el apoyo de entornos de desarrollo integrado (IDE), aplicando enfoques didácticos al trabajo colaborativo, la autonomía y ubicuidad a través de diversas herramientas enfocadas a la Educación para la industria 4.0, orientados a solucionar problemas elementales de la cotidianeidad.

La unidad de aprendizaje Lenguaje de Programación estará fundamentada en el Modelo Educativo Institucional vigente y en la Educación para la industria 4.0.

Se emplearán metodologías didácticas activas como el Aprendizaje, Gamificación, Design Thinking, Aprendizaje Por Proyectos, Science, Technology, Engineering Arts and Mathematics (STEAM), 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 que fomentarán la colaboración e interacción presenciales y virtuales, en forma síncrona o asíncrona, que corresponden a la Educación 4.0. También se emplearán aplicaciones informáticas en el campo del diseño de diagramas de flujo, diagramas de clase, entre otros.

El rol del profesor será de mediador entre el estudiante y los contenidos didácticos a abordar, puesto que se centrará en la creación, organización, supervisión y mediación de los espacios de trabajo, incluidos los ciberespacios, 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 que el estudiante conozca y, si es necesario, nivele sus conocimientos previos adquiridos en otras unidades de aprendizaje para que establezca conexiones significativas con la propuesta didáctica de la unidad de aprendizaje Lenguaje de Programación. 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.





Programa Académico: Técnico en Sistemas Digitales

del estudiante. Este momento se enriquecerá con diversos tipos de evaluación, como la heteroevaluación, 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, la evaluación sumativa, 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 Lenguaje de Programación, esto mediante evidencias de conocimiento, producto y desempeño, como diseño de algoritmos, elaboración de diagramas de flujo, codificaciones plasmadas en programas computacionales, 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 unidad de aprendizaje de Lenguaje de Programación, previo a su inicio. De esa forma, este programa de estudios 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.

Unidad de Aprendizaje: Lenguaje de Programación





Programa Académico: Técnico en Sistemas Digitales

Unidad de Aprendizaje: Lenguaje de Programación

DESCRIPCIÓN DE LA UNIDAD DE APREDIZAJE

Unidad de Aprendizaje: LENGUAJE DE PROGRAMACIÓN		
Propósito de la Unidad de Aprendizaje		
Crea soluciones a problemas de Sistemas Digitales mediante programas elaborados en un lenguaje de programación de alto nivel, aplicando estructuras de control y funciones de librería.		
Unidad 1: Análisis y diseño de Algoritmos		
Unidad de competencia	Aprendizajes esperados	Contenidos
1 Diseña algoritmos que den solución a problemas computables, realizando su análisis y representación a través de Diagramas de Flujo de Datos	1. Emplea la metodología para solución de problemas aplicables a los Sistemas digitales.	<p>Conceptuales</p> <ul style="list-style-type: none"> Análisis de problemas computables Concepto y características de algoritmos Tipos de algoritmos (lineales y no lineales) <p>Procedimentales</p> <ul style="list-style-type: none"> Práctica 1. Análisis de problemas. Analiza problemas computables en diferentes contextos (Definición del problema, especificaciones de entrada y de salida), para la construcción de soluciones en forma de algoritmos. <p>Actitudinales</p> <ul style="list-style-type: none"> Emplea el pensamiento lógico y matemático. Reflexiona sobre los aprendizajes desarrollados. Comunicación asertiva Creatividad Pensamiento crítico Trabajo colaborativo
	2. Utiliza los elementos y metodología para la representación de algoritmos	<p>Conceptuales</p> <ul style="list-style-type: none"> Diseño de algoritmos Diagramas de flujo de datos (ANSI – ISO) <p>Procedimentales</p> <ul style="list-style-type: none"> Práctica 2. Diseño de algoritmos en Diagramas de Flujo de Datos. Diseña algoritmos representados en Diagramas de Flujo de Datos para dar solución a problemas computables. <p>Actitudinales</p> <ul style="list-style-type: none"> Emplea el pensamiento lógico y matemático. Reflexiona sobre los aprendizajes desarrollados. Comunicación asertiva Creatividad





Programa Académico: Técnico en Sistemas Digitales

Unidad de Aprendizaje: Lenguaje de Programación

		<ul style="list-style-type: none"> • Pensamiento crítico • Trabajo colaborativo
Unidad 2: Codificación en Lenguaje C		
Unidad de competencia	Aprendizajes esperados	Contenidos
<ul style="list-style-type: none"> • Construye programas codificados en lenguaje de alto nivel, a partir de fundamentos de programación, y el uso de estructuras de control. 	<p>1. Aplica los fundamentos de programación en la codificación de soluciones de problemas de Sistemas Digitales</p>	<p>Conceptuales</p> <ul style="list-style-type: none"> • Clasificación de los tipos de lenguaje de programación • Estructura de un programa en C/C++ • Funciones de entrada y salida • Variables y constantes • Tipos de datos • Detección de errores y desbordamiento de memoria • Operadores aritméticos, relacionales y lógicos • Expresiones <p>Procedimentales</p> <ul style="list-style-type: none"> • Práctica 3. Prueba de compilador. Realiza ejercicios en un compilador de lenguaje C. • Práctica 4. Programas con entrada y salida de datos. Emplea las funciones de entrada y salida de datos en lenguaje C <p>Actitudinales</p> <ul style="list-style-type: none"> • Emplea el pensamiento lógico y matemático. • Reflexiona sobre los aprendizajes desarrollados. • Comunicación asertiva • Creatividad • Pensamiento crítico • Trabajo colaborativo
	<p>2. Emplea las estructuras de control en la codificación de soluciones de problemas.</p>	<p>Conceptuales</p> <ul style="list-style-type: none"> • Definición de estructuras de control secuenciales • Definición de estructuras de control selectivas <ul style="list-style-type: none"> ○ Simple ○ Doble ○ Múltiple • Definición de estructuras de control repetitivas. <ul style="list-style-type: none"> ○ while ○ do-while ○ for





Programa Académico: Técnico en Sistemas Digitales

Unidad de Aprendizaje: Lenguaje de Programación

		<p>Procedimentales</p> <ul style="list-style-type: none"> • Práctica 5. Estructuras de control secuenciales. Emplea las estructuras de control secuencial en un programa en lenguaje C. • Práctica 6. Estructuras de control if/if-else. Emplea las estructuras de control de toma de decisiones en un programa en lenguaje C. • Práctica 7. Estructuras de control con if-esle-if / switch. Emplea las estructuras de control selectivas anidadas y múltiples en un programa en lenguaje C. • Práctica 8. Estructuras de control for. Emplea las estructuras de control desde/para en un programa en lenguaje C. • Práctica 9. Estructuras de control while / do-while. Emplea las estructuras de control mientras/haz mientras en un programa en lenguaje C. <p>Actitudinales</p> <ul style="list-style-type: none"> • Emplea el pensamiento lógico y matemático. • Reflexiona sobre los aprendizajes desarrollados. • Comunicación asertiva • Creatividad • Pensamiento crítico • Trabajo colaborativo
Unidad 3: Arreglos de Datos		
Unidad de competencia	Aprendizajes esperados	Contenidos
3.- Desarrolla arreglos de datos y manipulación de cadenas de caracteres para la solución de problemáticas planteadas.	1. Aplica Arreglos en la solución de problemáticas	<p>Conceptuales</p> <ul style="list-style-type: none"> • Concepto de Arreglo (Fila, columna, índice) • Tipos de arreglos (vectores y matrices) • Manejo y uso del arreglo (definición, inicialización, llenado, lectura, recorrido, impresión). • Operaciones básicas con arreglos. • Arreglos unidimensionales y bidimensionales. <p>Procedimentales</p> <ul style="list-style-type: none"> • Práctica 10. Uso de vectores. Emplea arreglos en la solución de problemáticas • Práctica 11. Uso de matrices. Emplea arreglos en la solución de problemáticas <p>Actitudinales</p> <ul style="list-style-type: none"> • Emplea el pensamiento lógico y matemático.





Programa Académico: Técnico en Sistemas Digitales

Unidad de Aprendizaje: Lenguaje de Programación

		<ul style="list-style-type: none"> • Reflexiona sobre los aprendizajes desarrollados. • Comunicación asertiva • Creatividad • Pensamiento crítico • Trabajo colaborativo
	2. Emplea cadenas de texto mediante lenguaje de programación	<p>Conceptuales</p> <ul style="list-style-type: none"> • Cadenas • Concepto de Cadena de caracteres (arreglo de tipo texto) • Importancia del fin de cadena '\0' • Manejo y uso de la cadena de caracteres (definición, inicialización, llenado, lectura, recorrido, impresión). • Operaciones básicas con cadenas de caracteres <p>Procedimentales</p> <ul style="list-style-type: none"> • Práctica 12. Uso de cadenas • Práctica 13. Operaciones con cadenas <p>Actitudinales</p> <ul style="list-style-type: none"> • Emplea el pensamiento lógico y matemático. • Reflexiona sobre los aprendizajes desarrollados. • Comunicación asertiva • Creatividad • Pensamiento crítico • Trabajo colaborativo
Unidad 4: Funciones		
Unidad de competencia	Aprendizajes esperados	Contenidos
4.-Construye soluciones a problemáticas planteadas en un lenguaje de alto nivel, implementando funciones básicas del compilador y de usuario.	1. Utiliza Funciones de librería en la resolución de problemas	<p>Conceptuales</p> <ul style="list-style-type: none"> • Prototipos generales de funciones • Catálogo de Funciones <p>Procedimentales</p> <ul style="list-style-type: none"> • Práctica 14. Funciones matemáticas. Conoce la sintaxis para la declaración de funciones e Identifica las principales funciones matemáticas propias del compilador • Práctica 15. Funciones de sistema. Identifica otras funciones propias del compilador de uso frecuente (de entrada y salida, limpiar ventana, cambiar color de fondo y texto, etc.)





Programa Académico: Técnico en Sistemas Digitales

Unidad de Aprendizaje: Lenguaje de Programación

		<ul style="list-style-type: none"> ● Practica 16. Otras funciones. Utiliza otras funciones propias del compilador que ayuden a mejorar la codificación como las funciones para manejo de cadenas. <p>Actitudinales</p> <ul style="list-style-type: none"> ● Emplea el pensamiento lógico y matemático. ● Reflexiona sobre los aprendizajes desarrollados. ● Comunicación asertiva ● Creatividad ● Pensamiento crítico <p style="text-align: right;">Trabajo colaborativo</p>
	<p>2. Crea funciones de usuario aplicables a una problemática específica.</p>	<p>Conceptuales</p> <ul style="list-style-type: none"> ● Funciones de retorno por valor. ● Funciones de paso por valor. ● Funciones de paso por referencia. <p>Procedimentales</p> <ul style="list-style-type: none"> ● Practica 17: Funciones de retorno por valor. Identifica la forma de declarar funciones que regresan valores con y sin parámetros. ● Practica 18: Funciones de paso por valor. Crea una copia de variables para pasar valores al hacer una llamada de función con argumentos. ● Practica 19: Funciones de paso por referencia. Conoce los apuntadores, para pasar valores por la dirección de una variable al hacer una llamada de función. <p>Actitudinales</p> <ul style="list-style-type: none"> ● Emplea el pensamiento lógico y matemático. ● Reflexiona sobre los aprendizajes desarrollados. ● Comunicación asertiva ● Creatividad e Innovación ● Pensamiento crítico y reflexivo ● Trabajo colaborativo





Programa Académico: Técnico en Sistemas Digitales

Unidad de Aprendizaje: Lenguaje de Programación

➔ MATRIZ DE VINCULACIÓN ◀

	Unidad de Competencia 1		Unidad de Competencia 2		Unidad de competencia 3		Unidad de Competencia 4	
	AE 1	AE 2						
COMPETENCIAS PARA EL SIGLO XXI HABILIDADES BLANDAS Y SOCIOEMOCIONALES								
Trabajo colaborativo	X	X	X	X	X	X	X	X
Pensamiento crítico y reflexivo	X	X	X	X	X	X	X	X
Creatividad e Innovación	X	X	X	X	X	X	X	X
Responsabilidad	X	X	X	X	X	X	X	X
Sustentabilidad	X	X	X	X	X	X	X	X
Comunicación asertiva	X	X	X	X	X	X	X	X



Programa Académico: Técnico en Sistemas Digitales

Unidad de Aprendizaje: Lenguaje de Programación

PERFIL DOCENTE

- El profesor que imparta la Unidad de Aprendizaje Lenguaje de Programación contará con las habilidades en el manejo de los saberes disciplinares y profesionales, tales como: dominio de los temas en el área de algoritmos, metodologías de solución de problemas, uso de lenguajes de programación de alto nivel, conocimientos de electrónica, ingeniería de software, dominio de software aplicado a la programación, uso de las tecnologías de la información y la comunicación, dominio de las Tecnologías del Aprendizaje y Conocimiento, control de grupo, habilidad de ser adaptativo, reflexivo y crítico, deberá mostrar una conducta basada en los siguientes valores: empatía, honestidad, tolerancia, equidad y tendrá la disposición de fomentar una comunicación asertiva que favorezca el desarrollo del talento 4.0, debiendo conocer la normatividad del Instituto Politécnico Nacional y el Modelo Educativo vigente.
- Colaborará de forma colegiada, en la construcción de un proyecto de formación integral dirigido a los estudiantes, el cual deberá considerar su contexto social y brindará de forma constante el apoyo técnico pedagógico a alumnos y personal del plantel que lo requiera.
- De acuerdo con la normativa institucional, los profesores interesados en impartir la Unidad de aprendizaje, deberá evidenciar los conocimientos sobre el área y comprobar las habilidades que tienen en el manejo de las herramientas digitales y/o pedagógicas.
- Habilidades docentes en el desarrollo del Talento
- En el campo de los Sistemas Digitales:
- Organiza su formación continua a lo largo de su trayectoria profesional.
- Domina y estructura los saberes para facilitar experiencias de aprendizajes significativos.
- Lleva a la práctica procesos de enseñanza y aprendizaje de manera efectiva, creativa e innovadora a su contexto.
- Emplea diversas herramientas tecnológicas para generar espacios de aprendizaje sincrónicos y asíncronos.
- En el campo pedagógico:
- Planea considerando las características diversas de los estudiantes y el contexto institucional.
- Diseña planeaciones didácticas incorporando el uso de herramientas tecnológicas y recursos digitales
- Lleva a la práctica el proceso de enseñanza - aprendizaje, de forma efectiva, creativa e innovadora, en el contexto institucional.
- Evalúa los aprendizajes, tomando en cuenta los propósitos curriculares y particularidades de los estudiantes.
- Fomenta la participación constante de los estudiantes, sin distinguir condición social, física, de género o cultural.
- Utiliza estrategias didácticas que generen el aprendizaje activo en los estudiantes.
- Propone actividades o retos de acuerdo con propósitos o competencias específicas.
- En el campo de la investigación:
- Está atento a los avances científicos y tecnológicos dentro de la disciplina.
- Propone y fomenta la investigación como estímulo para la educación intelectual creadora y desarrollo tecnológico e innovador.
- Desarrollo de tecnologías orientadas a la industria 4.0.
- Perfil Profesional:
- Licenciatura en: Ing. en Comunicaciones y Electrónica, Ing. Computación, Ing. en Sistemas Computacionales, Ing. Electrónica, Ing. en Robótica, Ing. en Telemática, Ing. en Control y Automatización, Lic. o Ing. en Informática, o áreas afines al campo de la programación y la electrónica programable; con experiencia de al menos 6 años en el área docente.
- Interesado en formar a futuros profesionistas técnicos en el área de Sistemas Digitales promoviendo el trabajo en equipo, proyectando una actitud comprometida, dinámica creativa e innovadora, así como una empatía con sus alumnos.
- Experiencia comprobable de 1 año en la iniciativa pública o privada, además de desarrollar el gusto por la investigación y desarrollo tecnológico.
- Es importante hacer énfasis del requerimiento de un profesor titular para la unidad de aprendizaje de Lenguaje de Programación que defina los lineamientos y 2 profesores auxiliares que ayudaran a monitorear y reforzar el proceso académico de los estudiantes, de tal manera que la asesoría pueda ser más especializada.





Programa Académico: Técnico en Sistemas Digitales

Unidad de Aprendizaje: Lenguaje de Programación

ESTRUCTURA DIDÁCTICA

Unidad didáctica:	Análisis y Diseño de Algoritmos	Nivel:	Cuarto
Propósito:	Crea soluciones a problemas de Sistemas Digitales mediante programas elaborados en un lenguaje de programación de alto nivel, aplicando estructuras de control y funciones de librería.		
Unidad de competencia N°1:	Diseña algoritmos que den solución a problemas computables, realizando su análisis y representación a través de Diagramas de Flujo de Datos.		
Aprendizaje Esperado No 1:	Emplea la metodología para solución de problemas aplicables a los Sistemas digitales.	Tiempo esperado para obtener el Aprendizaje Esperado:	4 horas.
Contenidos de Aprendizaje			
Conceptuales:	Procedimentales:	Actitudinales:	
<ul style="list-style-type: none"> Análisis de problemas computables. Concepto y características de algoritmos. Tipos de algoritmos (lineales y no lineales). 	<ul style="list-style-type: none"> Práctica 01. Análisis de problemas Describe el análisis de problemas computables en diferentes contextos (Definición del problema, especificaciones de entrada y de salida), para construir soluciones en forma de algoritmos. 	<ul style="list-style-type: none"> Emplea el pensamiento lógico y matemático. Reflexiona sobre los aprendizajes desarrollados. Se Comunica asertivamente. Emplea la creatividad. Piensa de manera crítica. Realiza trabajo colaborativo. 	
Estrategia Didáctica y Ambiente de Aprendizaje			
<p>ESTRATEGIA DIDÁCTICA: Aprendizaje basado en problemas</p> <p>El docente: Expone las etapas para el análisis de problemas computables y el concepto de algoritmo. Expone el concepto de algoritmo y las técnicas para su diseño (Lineal y modular) a través de ejemplos contextualizados.</p> <p>El estudiante: Participa en dinámica de juego a para elegir a los participantes de la siguiente actividad aleatoriamente. Seleccionado de la actividad anterior aporta ideas sobre sus conocimientos previos relacionados con los algoritmos, su análisis y su diseño. Participa en lluvia de ideas para que de manera grupal se describa el concepto de problema computable. Describe el análisis de problemas computables en diferentes contextos a través de la definición, las especificaciones de entrada y las especificaciones de salida. Comparte el análisis de problemas computables de manera grupal, para recibir retroalimentación del docente y otros estudiantes. Describe soluciones a problemas computables a modo de algoritmos en prosa. Comparte de manera grupal los algoritmos creados para recibir retroalimentación.</p> <p>AMBIENTE DE APRENDIZAJE</p> <ul style="list-style-type: none"> Aula. Laboratorio de computación 			





Programa Académico: Técnico en Sistemas Digitales

Unidad de Aprendizaje: Lenguaje de Programación

Herramientas Tecnológicas y Recursos Didácticos	Evidencia de Aprendizaje para la Evaluación Formativa	Criterios e Instrumentos de Evaluación
<p>Herramientas tecnológicas</p> <ul style="list-style-type: none"> • Plataforma educativa virtual • Proyector • Equipo de cómputo con acceso a Internet <p>Recursos didácticos</p> <ul style="list-style-type: none"> • Diagrama representativo de las etapas para el análisis de problemas. • Diagramas sobre las técnicas lineal y modular para diseñar algoritmos. • Problemas computables contextualizados. 	<ul style="list-style-type: none"> • Informe con el análisis de problemas computables con la definición de problema, especificaciones de entrada y de salida plasmados. 	<p>Criterios de evaluación:</p> <p>Fondo:</p> <ul style="list-style-type: none"> • Incluye la definición de cada problema computable en el análisis. • Describe las especificaciones de entrada y de salida. • Presenta algoritmos en modo de prosa con la posible solución a cada problema. • Numera cada paso o instrucción dentro de los algoritmos. • Incluye una instrucción de inicio y fin dentro de los algoritmos <p>Forma:</p> <ul style="list-style-type: none"> • Entrega los algoritmos en documento de texto o PDF. • Incluye coetilla con información como nombre, nivel, grupo, Unidad de Aprendizaje y fecha. • Contiene buena ortografía. • Participa de manera activa durante la sesión, aportando ideas u opiniones a la clase. <p>Instrumento de evaluación.</p> <ul style="list-style-type: none"> • Lista de cotejo





Programa Académico: Técnico en Sistemas Digitales

Unidad de Aprendizaje: Lenguaje de Programación

Unidad didáctica:	Análisis y Diseño de Algoritmos	Nivel:	Cuarto
Propósito:	Crea soluciones a problemas de Sistemas Digitales mediante programas elaborados en un lenguaje de programación de alto nivel, aplicando estructuras de control y funciones de librería.		
Unidad de competencia N°1:	Diseña algoritmos que den solución a problemas computables, realizando su análisis y representación a través de Diagramas de Flujo de Datos.		
Aprendizaje Esperado No 2:	Utiliza los elementos y metodología para la representación de algoritmos.	Tiempo esperado para obtener el Aprendizaje Esperado:	4 horas.

Contenidos de Aprendizaje

Conceptuales:	Procedimentales:	Actitudinales:
<ul style="list-style-type: none"> Diseño de algoritmos Diagramas de flujo de datos (ANSI – ISO) 	<ul style="list-style-type: none"> Práctica 02. Diseña algoritmos representados en Diagramas de Flujo de Datos para dar solución a problemas computables. 	<ul style="list-style-type: none"> Emplea el pensamiento lógico y matemático. Reflexiona sobre los aprendizajes desarrollados. Se Comunica asertivamente Emplea la creatividad Piensa de manera crítica Realiza trabajo colaborativo

Estrategia Didáctica y Ambiente de Aprendizaje

ESTRATEGIA DIDÁCTICA: Aprendizaje basado en problemas

El docente:

Introduce el tema de Diagrama de Flujo de Datos, su creación y aplicación dentro de los sistemas digitales.
Ejemplifica mediante la exposición, la construcción de un algoritmo lineal en forma de Diagrama de Flujo de Datos para dar solución a un problema computable.

El estudiante:

Participa en una lluvia de ideas para identificar los beneficios para trabajar con elementos visuales o con elementos gráficos.
Investiga en medios electrónicos el significado de ANSI e ISO en el contexto de los sistemas digitales.
Comparte de manera grupal los resultados de la investigación.
Participa en una lluvia de ideas para concluir el significado de las siglas y el trabajo de ambas organizaciones.
Investiga las figuras estándar para la creación de Diagramas de Flujo de Datos y las reglas para su creación.
Manipula el Software para la creación de DFD de manera electrónica.
Construye Diagramas de Flujo de Datos de manera electrónica, para representar algoritmos que den solución a problemas computables en diferentes contextos.

AMBIENTE DE APRENDIZAJE

- Aula
- Laboratorio de computación





Programa Académico: Técnico en Sistemas Digitales

Unidad de Aprendizaje: Lenguaje de Programación

Herramientas Tecnológicas y Recursos Didácticos	Evidencias de Aprendizaje para la Evaluación Formativa	Criterios e Instrumentos de Evaluación
<p>Herramientas Tecnológicas</p> <ul style="list-style-type: none"> • Plataforma educativa virtual • Proyector • Equipo de cómputo con acceso a Internet <p>Recursos didácticos</p> <ul style="list-style-type: none"> • Diagramas con las figuras estándar para la creación de DFD. • Problemas computables contextualizados. 	<ul style="list-style-type: none"> • Diagramas de Flujo de Datos que represente algoritmos para dar solución a problemas computables. 	<p>Criterios de evaluación:</p> <p>Fondo:</p> <ul style="list-style-type: none"> • Investiga y analiza el significado de ANSI e ISO y su intervención en los sistemas digitales y el diseño de algoritmos. • Representa algoritmos mediante DFD. • Emplea las figuras estándar de manera correcta para la creación de DFD. • Cumple con las reglas para la creación de DFD. • Incluye una instrucción inicio y fin dentro de los algoritmos. • Participa de manera activa durante la sesión, aportando ideas u opiniones a la clase. <p>Forma:</p> <ul style="list-style-type: none"> • Entrega los algoritmos en forma de imagen con formato PNG o JPG. • Incluye coetilla con información como nombre, nivel, grupo, Unidad de Aprendizaje y fecha. • Contiene buena ortografía. <p>Instrumento de evaluación.</p> <ul style="list-style-type: none"> • Lista de cotejo





Programa Académico: Técnico en Sistemas Digitales

Unidad de Aprendizaje: Lenguaje de Programación

Unidad didáctica:	Codificación en lenguaje C	Nivel:	Cuarto
Propósito:	Crea soluciones a problemas de Sistemas Digitales mediante programas elaborados en un lenguaje de programación de alto nivel, aplicando estructuras de control y funciones de librería		
Unidad de competencia N°2:	Construye programas codificados en lenguaje de alto nivel, a partir de fundamentos de programación, y el uso de estructuras de control.		
Aprendizaje Esperado No 1:	Aplika los fundamentos de programación en la codificación de soluciones de problemas de Sistemas Digitales	Tiempo esperado para obtener el Aprendizaje Esperado:	8 hrs

Contenidos de Aprendizaje

Conceptuales:	Procedimentales:	Actitudinales:
<ul style="list-style-type: none"> Clasificación de los tipos de lenguaje de programación Estructura de un programa en C/C++ Funciones de entrada y salida Variables y constantes Tipos de datos Operadores aritméticos, relacionales y lógicos 	<ul style="list-style-type: none"> Realiza ejercicios en un compilador de lenguaje C. Realiza ejercicios donde emplea las funciones de entrada y salida de datos en lenguaje C Práctica 3. Prueba de compilador Práctica 4. Programas con entrada y salida de datos. 	<ul style="list-style-type: none"> Emplea el pensamiento lógico y matemático. Reflexiona sobre los aprendizajes desarrollados. Se Comunica asertivamente Emplea la creatividad Piensa de manera crítica Realiza trabajo colaborativo

Estrategia Didáctica y Ambiente de Aprendizaje

ESTRATEGIA DIDÁCTICA: Design thinking

El Docente

- Ilustra y explica en código de lenguaje C/C++, las partes que constituyen la forma básica de un programa.
- Explica el uso de las funciones de entrada y salida en lenguaje C/C++.
- Explica la diferencia entre el concepto de variable y constante.
- Explica cómo funcionan las estructuras de control de repetitivas en un programa en lenguaje C.
- Explica cómo funciona la estructura de control While y su aplicación en un programa en lenguaje C.
- Explica cómo funciona la estructura de control Do-While y su aplicación en un programa en lenguaje C.
- Explica y hace uso de los tipos de datos del lenguaje C/C++ para definir el uso de cada tipo de dato y su aplicación en un programa.
- Modela el orden jerárquico de los operadores aritmético, relacionales y lógicos.

El estudiante:

- Realizan un cuadro comparativo de las diferencias entre diversos lenguajes de programación.
- Programan (compilar) ejercicios de funciones de salida y entrada.
- Investigan las ventajas y desventajas de usar variables y/o constantes para la programación en C/C++.
- Investigan los diferentes tipos de datos admisibles en lenguaje C/C++, realizando su cuadro comparativo resaltando las diferencias de estos.





Programa Académico: Técnico en Sistemas Digitales

Unidad de Aprendizaje: Lenguaje de Programación

- Clasifica los diferentes tipos de operadores en lenguaje C/C++.
- Programa (compilar) ejercicios de entrada y salida de datos combinando operadores aritméticos, relacionales y lógicos.

Ambiente de aprendizaje:

- Aula
- Laboratorio de computación

Herramientas Tecnológicas y Recursos Didácticos	Evidencias de Aprendizaje para la Evaluación Formativa	Criterios e Instrumentos de Evaluación
<p>Herramientas tecnológicas Uso de computadora con S.O. Windows Conectividad a internet</p> <p>Recursos didácticos Compilador de C (propuesto por el profesor) Software compilador C Computadora con S.O. Windows</p>	<ul style="list-style-type: none"> • Portafolio de evidencias en el que desarrolla programas que muestren en pantalla, textos preestablecidos en código de lenguaje C/C++, además programas que ejecuten operaciones matemáticas básicas en conjunto con los operadores lógicos y relacionales. 	<p>Criterios de evaluación:</p> <p>Fondo:</p> <ul style="list-style-type: none"> • Emplea la estructura básica de codificación en lenguaje C/C++. • Utiliza correctamente la estructura de las funciones de entrada y salida. • Respeta los tipos de datos necesarios para la ejecución de su programa. • Incluye operadores aritméticos, lógicos y relacionales en la solución de los problemas propuestos. • Utiliza la jerarquía de operaciones, respetando la prioridad de los operadores <p>Forma:</p> <ul style="list-style-type: none"> • Se entrega en formato digital, en la plataforma indicada por el docente. • Incluye archivos, código fuente de los programas desarrollados. • Se entrega en las fechas establecidas. <p>Instrumento de evaluación: Lista de cotejo</p>





Programa Académico: Técnico en Sistemas Digitales

Unidad de Aprendizaje: Lenguaje de Programación

Unidad didáctica:	Codificación en lenguaje C	Nivel:	Cuarto
Propósito:	Crea soluciones a problemas de Sistemas Digitales mediante programas elaborados en un lenguaje de programación de alto nivel, aplicando estructuras de control y funciones de librería		
Unidad de competencia N°2:	Construye programas codificados en lenguaje de alto nivel, a partir de fundamentos de programación, y el uso de estructuras de control.		
Aprendizaje Esperado No 2:	Emplea las estructuras de control en la codificación de soluciones de problemas.	Tiempo esperado para obtener el Aprendizaje Esperado:	20 hrs

Contenidos de Aprendizaje

Conceptuales:	Procedimentales:	Actitudinales:
<ul style="list-style-type: none"> Definición de estructuras de control secuenciales Definición de estructuras de control selectivas <ol style="list-style-type: none"> Simple Doble Múltiple Definición de estructuras de control repetitivas. <ol style="list-style-type: none"> while do-while for 	<ul style="list-style-type: none"> ✓ Práctica 5. Estructuras de control secuenciales ✓ Práctica 6. Estructuras de control if/if-else ✓ Práctica 7. Estructuras de control con if-esle-if / switch ✓ Práctica 8. Estructuras de control for. ✓ Práctica 9. Estructuras de control while / do-while. 	<ul style="list-style-type: none"> Emplea el pensamiento lógico y matemático. Reflexiona sobre los aprendizajes desarrollados. Comunicación asertiva Creatividad e Innovación Pensamiento crítico y reflexivo Trabajo colaborativo

Estrategia Didáctica y Ambiente de Aprendizaje

ESTRATEGIA DIDÁCTICA: Aprendizaje por problemas

El docente:

- Ilustra y explica la definición de Estructura de control y la clasificación de estas.
- Ilustra y explica el tema de estructuras de control secuencial.
- Ejemplifica la resolución y programación de problemas empleando la estructura secuencial.
- Ilustra y explica el tema de estructuras de control selectivas.
- Ejemplifica la resolución y programación de problemas empleando estructuras de control selectivas.
- Ilustra y explica el tema de estructuras de control repetitivas.
- Ejemplifica la resolución y programación de problemas empleando estructuras de control repetitivas.

El estudiante:

- Realizan un cuadro comparativo de las diferentes estructuras de control.
- Desarrolla la Practica 5 “Estructuras de control secuenciales”.
- Resuelve programas en lenguaje C empleando estructura de control secuencia.
- Desarrolla la Practica 6 “Estructuras de control “if/if-else””.
- Desarrolla la Practica 7 “Estructuras de control “if-else-if/switch”.
- Resuelve programas en lenguaje C empleando estructuras de control selectivas.
- Desarrolla la Practica 8 “Estructuras de control “for””.
- Desarrolla la Practica 9 “Estructuras de control “while / do-while”.





Programa Académico: Técnico en Sistemas Digitales

Unidad de Aprendizaje: Lenguaje de Programación

- Resuelve programas en lenguaje C empleando estructuras de control repetitivas.
- Desarrolla un proyecto que resuelva un problema que pueda trasladarse a un contexto fuera del educativo.

Ambiente de aprendizaje:

- Aula
- Laboratorio de computación

Herramientas Tecnológicas y Recursos Didácticos	Evidencias de Aprendizaje para la Evaluación Formativa	Criterios e Instrumentos de Evaluación
Uso de computadora Conectividad a internet Compilador de C (propuesto por el profesor) Computadora con S.O. Windows Software compilador C Proyector	Portafolio de evidencias en el que desarrolla programas que muestren el uso de las estructuras de control selectivas, selectivas y/ o repetitivas en código de lenguaje C/C++, además demuestren la integración de estas estructuras de control en su implementación en un proyecto.	<p>Criterios de evaluación:</p> <p>Fondo:</p> <ul style="list-style-type: none"> • Emplea estructuras de control secuencial en lenguaje C. • Emplea estructuras de control selectivas en lenguaje C. • Emplea estructuras de control repetitivas en lenguaje C. • Implementa las estructuras de control en un proyecto que resuelve una problemática. <p>Forma:</p> <ul style="list-style-type: none"> • Se entrega en formato digital, en la plataforma indicada por el docente. • Incluye archivos, código fuente de los programas desarrollados. • Se entrega en las fechas establecidas. <p>Instrumento de evaluación: Lista de cotejo</p>





Programa Académico: Técnico en Sistemas Digitales

Unidad de Aprendizaje: Lenguaje de Programación

Unidad didáctica:	Arreglo de datos	Nivel:	Cuarto
Propósito:	Crea soluciones a problemas de Sistemas Digitales mediante programas elaborados en un lenguaje de programación de alto nivel, aplicando estructuras de control y funciones de librería		
Unidad de competencia N°3:	Desarrolla arreglos de datos y manipulación de cadenas de caracteres para la solución de problemáticas planteadas.		
Aprendizaje Esperado No 1:	Aplica Arreglos en la solución de problemáticas	Tiempo esperado para obtener el Aprendizaje Esperado:	8 hrs

Contenidos de Aprendizaje

Conceptuales:	Procedimentales:	Actitudinales:
Arreglos <ul style="list-style-type: none"> • Concepto de Arreglo (Fila, columna, índice) • Tipos de arreglos (vectores y matrices) • Manejo y uso del arreglo (definición, inicialización, llenado, lectura, recorrido, impresión). • Operaciones básicas con arreglos • Arreglos Unidimensionales • Arreglos Bidimensionales 	<ul style="list-style-type: none"> ✓ Práctica 10. Uso de vectores ✓ Práctica 11. Uso de matrices 	<ul style="list-style-type: none"> • Emplea el pensamiento lógico y matemático. • Reflexiona sobre los aprendizajes desarrollados. • Se Comunica asertivamente. • Emplea la creatividad. • Piensa de manera crítica. • Realiza trabajo colaborativo.

Estrategia Didáctica y Ambiente de Aprendizaje

ESTRATEGIA DIDÁCTICA: Aprendizaje basado en problemas

El docente:

Expone la definición y características de un arreglo.
Muestra los tipos de arreglos y sus reglas de uso.
Proporciona ejemplos de uso de arreglos
Expone una problemática real y solicita al estudiante proponga una solución utilizando arreglos.

El estudiante:

Conceptualiza qué es un arreglo
Diferencia tipos de arreglos (vectores y matrices)
Ejecuta en la computadora los ejemplos de arreglos

Recuperación de conocimientos previos,
Lluvia de ideas,
Aprendizaje significativo,
Aprendizaje basado en problemas

AMBIENTE DE APRENDIZAJE

- Aula





Programa Académico: Técnico en Sistemas Digitales

Unidad de Aprendizaje: Lenguaje de Programación

- Laboratorio de computación

	Evidencias de Aprendizaje para la Evaluación Formativa	Criterios e Instrumentos de Evaluación
<p>Herramientas Tecnológicas:</p> <ul style="list-style-type: none"> • Computadora con S.O. Windows • Conectividad a internet • Software compilador C • Proyector <p>Recursos Didácticos:</p> <ul style="list-style-type: none"> • Presentaciones y/o Apuntes elaborados por el docente • Programas de ejemplo • Manual de ejercicios propuestos por el docente 	<p>Problemario resuelto, con soluciones a problemas que requieren el uso y aplicación de arreglos de datos unidimensionales y bidimensionales.</p>	<p>Criterios de evaluación:</p> <p>Fondo:</p> <ul style="list-style-type: none"> • Emplea arreglos unidimensionales en lenguaje C para la solución de problemas planteados. • Emplea arreglos bidimensionales en lenguaje C para la solución de problemas planteados • Respetar los tipos de datos para los arreglos correspondientes con el tipo de problema. • Utiliza operadores entre elementos de diferentes arreglos en la solución de los problemas propuestos. <p>Forma:</p> <ul style="list-style-type: none"> • Se entrega en formato digital, en la plataforma indicada por el docente. • Incluye archivos, código fuente de los programas desarrollados. • Se entrega en las fechas establecidas. <p>Instrumento de evaluación: Lista de cotejo</p>





Programa Académico: Técnico en Sistemas Digitales

Unidad de Aprendizaje: Lenguaje de Programación

Unidad didáctica:	Arreglos de Datos	Nivel:	Cuarto
Propósito:	Crea soluciones a problemas de Sistemas Digitales mediante programas elaborados en un lenguaje de programación de alto nivel, aplicando estructuras de control y funciones de librería		
Unidad de competencia N°3:	Desarrolla arreglos de datos y manipulación de cadenas de caracteres para la solución de problemáticas planteadas		
Aprendizaje Esperado No 2:	Emplea cadenas de caracteres mediante lenguaje de programación	Tiempo esperado para obtener el Aprendizaje Esperado:	8 hrs

Contenidos de Aprendizaje

Conceptuales:	Procedimentales:	Actitudinales:
<ul style="list-style-type: none"> • Concepto de Cadena de caracteres (arreglo de tipo texto) • Importancia del fin de cadena '\0' • Manejo y uso de la cadena de caracteres (definición, inicialización, llenado, lectura, recorrido, impresión). • Operaciones básicas con cadenas de caracteres 	<p>Practica 12. Uso de cadenas de caracteres. Emplea cadenas de caracteres para la representación de información como palabras y/o enunciados en soluciones a problemas con lenguaje C.</p> <p>Practica 13. Operaciones con cadenas de caracteres. Emplea operaciones con cadenas de caracteres como la ordenación, concatenación entre otras, empleadas en soluciones a problemas con lenguaje C.</p>	<ul style="list-style-type: none"> • Emplea el pensamiento lógico y matemático. • Reflexiona sobre los aprendizajes desarrollados. • Se Comunica asertivamente • Emplea la creatividad • Piensa de manera crítica • Realiza trabajo colaborativo

Estrategia Didáctica y Ambiente de Aprendizaje

ESTRATEGIA DIDÁCTICA: Aprendizaje basado en problemas

El docente:

El docente expone la definición y características de una cadena de caracteres
 El docente muestra el uso de las cadenas de caracteres y sus reglas de uso (fin de cadena).
 El docente proporciona ejemplos de uso de cadenas de caracteres
 El docente expone una problemática real y solicita al estudiante proponga una solución utilizando cadenas de caracteres.

El estudiante:

El estudiante conceptualiza qué es una cadena de caracteres.
 El estudiante explica las cadenas de caracteres y sus reglas de uso.
 El estudiante ejecuta en la computadora los ejemplos de cadenas de caracteres

AMBIENTE DE APRENDIZAJE

- Aula
- Laboratorio de computación





Programa Académico: Técnico en Sistemas Digitales

Unidad de Aprendizaje: Lenguaje de Programación

Herramientas Tecnológicas y Recursos Didácticos	Evidencias de Aprendizaje para la Evaluación Formativa	Criterios e Instrumentos de Evaluación
<p>Herramientas Tecnológicas:</p> <ul style="list-style-type: none"> • Computadora con S.O. Windows • Conectividad a internet • Software compilador C <p>Recursos Didácticos:</p> <ul style="list-style-type: none"> • Presentaciones y/o Apuntes elaborados por el docente • Programas de ejemplo • Manual de ejercicios propuestos por el docente 	<p>Problemario resuelto, con soluciones a problemas que requieren el uso y aplicación de cadenas de texto de forma independiente y en conjunto con operaciones que trabajen con las cadenas de texto completas, como la concatenación y algoritmos de ordenación y problemas caso, en soluciones con lenguaje C.</p>	<p>Criterios de evaluación:</p> <p>Fondo:</p> <ul style="list-style-type: none"> • Emplea cadenas de texto en lenguaje C para la solución de problemas planteados. • Emplea operaciones con cadenas de texto en lenguaje C para la solución de problemas planteados • Resuelve problemas con el uso de cadenas de texto en lenguaje C. • Emplea operadores entre elementos de diferentes cadenas de texto en la solución de los problemas propuestos. <p>Forma:</p> <ul style="list-style-type: none"> • Se entrega en formato digital, en la plataforma indicada por el docente. • Incluye archivos, código fuente de los programas desarrollados. • Se entrega en las fechas establecidas. <p>Instrumento de evaluación: Lista de cotejo.</p>





Programa Académico: Técnico en Sistemas Digitales

Unidad de Aprendizaje: Lenguaje de Programación

Unidad didáctica:	Funciones	Nivel:	Cuarto
Propósito:	Crea soluciones a problemas de Sistemas Digitales mediante programas elaborados en un lenguaje de programación de alto nivel, aplicando estructuras de control y funciones de librería.		
Unidad de competencia N°4:	Construye soluciones a problemáticas planteadas en un lenguaje de alto nivel, implementando funciones básicas del compilador y de usuario		
Aprendizaje Esperado No 1:	Utiliza funciones de librería en la resolución de problemas	Tiempo esperado para obtener el Aprendizaje Esperado:	12 hrs

Contenidos de Aprendizaje

Conceptuales:	Procedimentales:	Actitudinales:
<p>Prototipo general de Funciones Catálogo de Funciones</p>	<p>Práctica 14. Funciones matemáticas. Conoce la sintaxis para la declaración de funciones. Identifica las principales funciones matemáticas propias del compilador.</p> <p>Práctica 15. Funciones de sistema. Identifica otras funciones propias del compilador de uso frecuente (de entrada y salida, limpiar ventana, cambiar color de fondo y texto, etc.)</p> <p>Practica 16. Otras funciones. Utilizar otras funciones propias del compilador que ayuden a mejorar la codificación como las funciones para manejo de cadenas.</p>	<ul style="list-style-type: none"> • Emplea el pensamiento lógico y matemático. • Reflexiona sobre los aprendizajes desarrollados. • Se Comunica asertivamente • Emplea la creatividad • Piensa de manera crítica • Realiza trabajo colaborativo

Estrategia Didáctica y Ambiente de Aprendizaje

Estrategias didácticas: Aprendizaje basado en problemas

Docente

El docente expone la librería matemática y algunas de sus funciones principales: potencia, raíz cuadrada, funciones trigonométricas.

El docente presenta programas que muestren el uso de funciones matemáticas.

El docente muestra el uso de funciones de uso frecuente en diversos programas utilizadas durante el curso: entrada y salida, limpiar ventana, cambiar color de fondo y del texto, esperar carácter.

El docente muestra funciones básicas de cadena, para la manipulación de datos.

Estudiante

El estudiante resuelve ejercicios propuestos por el docente donde aplique las funciones matemáticas.

El estudiante resuelve diversos ejercicios propuestos por el docente donde aplique diversas funciones propias del compilador.

Ambiente de Aprendizaje:

- Aula
- Laboratorio de computación





Programa Académico: Técnico en Sistemas Digitales

Unidad de Aprendizaje: Lenguaje de Programación

Herramientas Tecnológicas y Recursos Didácticos	Evidencias de Aprendizaje para la Evaluación Formativa	Criterios e Instrumentos de Evaluación
<p>Herramientas Tecnológicas: Computadora con S.O. Windows Software compilador C Proyector</p> <p>Recursos Didácticos: Presentaciones y/o Apuntes elaborados por el docente Programas de ejemplo Manual de ejercicios propuestos por el docente.</p>	<p>Diseña programas con soluciones que integran funciones matemáticas del sistema propio y algunas especiales para dar estilo a la salida de los programas</p>	<p>Criterios de evaluación:</p> <p>Fondo:</p> <ul style="list-style-type: none"> • Código funcionando • El código resuelve la problemática planteada • Resultado esperado <p>Forma:</p> <ul style="list-style-type: none"> • Comentarios dentro del código con los datos personales del estudiante y grupo • Alineación y espaciado adecuado en la codificación • Código fácil de lectura • Secuencia • Lógica <p>Instrumento de evaluación:</p> <ul style="list-style-type: none"> • Lista de Cotejo.





Programa Académico: Técnico en Sistemas Digitales

Unidad de Aprendizaje: Lenguaje de Programación

Unidad didáctica:	Funciones	Nivel:	Cuarto
Propósito:	Crea soluciones a problemas de Sistemas Digitales mediante programas elaborados en un lenguaje de programación de alto nivel, aplicando estructuras de control y funciones de librería.		
Unidad de competencia N°4:	-Construye soluciones a problemáticas planteadas en un lenguaje de alto nivel, implementando funciones básicas del compilador y de usuario		
Aprendizaje Esperado No 2:	Crea Funciones de usuario aplicables a una problemática específica	Tiempo esperado para obtener el Aprendizaje Esperado:	8 hrs
Contenidos de Aprendizaje			
Conceptuales:	Procedimentales:	Actitudinales:	
Funciones de retorno por valor Funciones de paso por valor Funciones de paso por referencia	<ul style="list-style-type: none"> ● Practica 17: Funciones de retorno por valor. ● Identifica la forma de declarar funciones que regresan valores con y sin parámetros. ● Practica 18: Funciones de paso por valor. ● Crea una copia de variables para pasar valores al hacer una llamada de función con argumentos. 	<ul style="list-style-type: none"> ● Emplea el pensamiento lógico y matemático. ● Reflexiona sobre los aprendizajes desarrollados. ● Se Comunica asertivamente. ● Emplea la creatividad. ● Piensa de manera crítica. ● Realiza trabajo colaborativo. 	
Estrategia Didáctica y Ambiente de Aprendizaje			
<p>Estrategias didácticas: Aprendizaje basado en problemas</p> <p>El Docente El docente expone las variables globales y las variables locales. El docente expone las funciones de retorno por valor, con y sin parámetros. El docente presenta programas que muestren el uso de funciones de retorno por valor, con y sin parámetros. El docente expone las funciones de paso por valor con argumentos. El docente expone los apuntadores el lenguaje C El docente expone las funciones de paso por referencia El docente presenta programas que muestren el uso de funciones de paso por referencia.</p> <p>El estudiante El estudiante resuelve ejercicios propuestos por el docente donde aplique el uso de funciones que utilicen el paso por valor con argumentos. El estudiante resuelve ejercicios propuestos por el docente donde aplique las funciones de paso por referencia. El estudiante incorpora en otros ejercicios ya resueltos previamente, el manejo de funciones para mejorar la codificación y el desarrollo de sus programas. El estudiante resuelve ejercicios propuestos por el docente donde aplique el uso de funciones que retornen valores con y sin parámetros.</p> <p>Ambiente de Aprendizaje: En el Aula y Laboratorio</p>			
Herramientas Tecnológicas y Recursos Didácticos	Evidencias de Aprendizaje para la Evaluación Formativa	Criterios e Instrumentos de Evaluación	
<p>Herramientas Tecnológicas: Computadora con S.O. Windows Software compilador C Proyector</p> <p>Recursos Didácticos: Presentaciones y/o Apuntes elaborados por el docente Programas de ejemplo Manual de ejercicios propuestos por el docente.</p>	Diseña programas que resuelvan problemáticas, aplicando las funciones con parámetros y haciéndolos más eficientes con el uso de los apuntadores.	<p>Criterios de forma:</p> <ul style="list-style-type: none"> ● Comentarios dentro del código con los datos personales del estudiante y grupo ● Alineación y espaciado adecuado en la codificación ● Código fácil de lectura ● Secuencia ● Lógica <p>Criterios de fondo:</p> <ul style="list-style-type: none"> ● Código funcionando con funciones ● El código resuelve la problemática planteada ● Resultado esperado <p>Instrumento de evaluación:</p> <ul style="list-style-type: none"> ● Lista de Cotejo. 	





Programa Académico: Técnico en Sistemas Digitales

Unidad de Aprendizaje: Lenguaje de Programación

PRÁCTICAS

Nombre de la Práctica:	Análisis de Problemas	N° de la Práctica:	1	Tiempo:	3 horas
Unidades del Programa de Estudio:	1. Análisis y Diseño de Algoritmos				
Aprendizajes Esperados Relacionados con la Práctica:	Emplea la metodología para solución de problemas aplicables a los Sistemas digitales.				
Contenidos de Aprendizaje					
Conceptuales:	Procedimentales:	Actitudinales:			
<ul style="list-style-type: none"> Análisis de problemas computables. 	Describe el análisis de problemas computables en diferentes contextos (Definición del problema, especificaciones de entrada y de salida), para construir soluciones en forma de algoritmos.	<ul style="list-style-type: none"> Emplea el pensamiento lógico y matemático. Reflexiona sobre los aprendizajes desarrollados. Se Comunica asertivamente. Emplea la creatividad. Piensa de manera crítica. Realiza trabajo colaborativo. 			
Estrategia Didáctica y Ambiente de Aprendizaje					
ESTRATEGIA DIDÁCTICA: Aprendizaje basado en problemas					
El docente: Retroalimenta el análisis de los problemas que ha realizado el estudiante.					
El estudiante: Realiza el análisis de problemas computables en diferentes contextos, plasmando dicho análisis en un documento de texto. Describe la definición de varios problemas, asegurándose que se encuentren atomizados. Describe las especificaciones de entrada que permitan desarrollar una solución para los problemas planteados. Describe las especificaciones de salida que permitan desarrollar una solución para los problemas planteados. Comparte el análisis de problemas computables de manera grupal.					
AMBIENTE DE APRENDIZAJE Laboratorio de cómputo					
Herramientas Tecnológicas y Recursos Didácticos	Evidencias de Aprendizaje para la Evaluación Formativa	Criterios e Instrumentos de Evaluación			
Herramientas Tecnológicas <ul style="list-style-type: none"> Equipo de cómputo. Recursos didácticos <ul style="list-style-type: none"> Problemas computables contextualizados. 	<ul style="list-style-type: none"> Informe con el análisis de problemas computables con la definición de problema, especificaciones de entrada y de salida plasmados en un documento de texto. 	Criterios de fondo <ul style="list-style-type: none"> Incluye la definición de cada problema computable en el análisis. Describe las especificaciones de entrada. Describe las especificaciones de salida. Criterios de forma <ul style="list-style-type: none"> Entrega los algoritmos en documento de texto o PDF. Incluye coetilla con información como nombre, nivel, grupo, Unidad de Aprendizaje y fecha. Contiene buena ortografía. Instrumento <ul style="list-style-type: none"> Rúbrica de evaluación. 			





Programa Académico: Técnico en Sistemas Digitales

Unidad de Aprendizaje: Lenguaje de Programación

N° y Nombre de la Práctica:	Diseño de Algoritmos en Diagrama de Flujo de Datos	N° de la Práctica:	2	Tiempo:	3 horas
Unidades del Programa de Estudio:	1. Análisis y Diseño de Algoritmos				
Aprendizajes Esperados Relacionados con la Práctica:	Utiliza los elementos y metodología para la representación de algoritmos				

Contenidos de Aprendizaje		
Conceptuales:	Procedimentales:	Actitudinales:
<ul style="list-style-type: none"> Diagramas de flujo de datos (ANSI – ISO) 	<ul style="list-style-type: none"> Diseña algoritmos representados en Diagramas de Flujo de Datos para dar solución a problemas computables. 	<ul style="list-style-type: none"> Emplea el pensamiento lógico y matemático. Piensa de manera crítica Reflexiona sobre los aprendizajes desarrollados.

Estrategia Didáctica y Ambiente de Aprendizaje

ESTRATEGIA DIDÁCTICA: Aprendizaje basado en problemas

El docente:

Introduce el tema de Diagrama de Flujo de Datos, su creación y aplicación dentro de los sistemas digitales.
Realiza práctica guiada sobre la construcción de un DFD con el uso de las figuras estándar en un Software especializado.

El estudiante:

Investiga las figuras estándar para la creación de Diagramas de Flujo de Datos.
Investiga y analiza las reglas para la creación de DFD.
Manipula el Software para la creación de DFD de manera electrónica.
Construye Diagramas de Flujo de Datos de manera electrónica, para representar algoritmos que den solución a problemas computables en diferentes contextos.

AMBIENTE DE APRENDIZAJE

Laboratorio de cómputo

Herramientas Tecnológicas y Recursos Didácticos	Evidencias de Aprendizaje para la Evaluación Formativa	Criterios e Instrumentos de Evaluación
<p>Herramientas Tecnológicas</p> <ul style="list-style-type: none"> Proyector Equipo de cómputo con acceso a Internet <p>Recursos didácticos</p> <ul style="list-style-type: none"> Problemas computables contextualizados 	<ul style="list-style-type: none"> Diagramas de Flujo de Datos que representen algoritmos para dar solución a problemas computables planteados. 	<p>Criterios de fondo</p> <ul style="list-style-type: none"> Representa algoritmos mediante DFD. Emplea las figuras estándar de manera correcta para la creación de DFD. Cumple con las reglas para la creación de DFD. Incluye una instrucción inicio y fin dentro de los algoritmos. <p>Criterios de forma</p> <ul style="list-style-type: none"> Entrega los algoritmos en forma de imagen con formato PNG o JPG. Incluye coetilla con información como nombre, nivel, grupo, Unidad de Aprendizaje y fecha. Contiene buena ortografía. <p>Instrumento</p> <ul style="list-style-type: none"> Rúbrica de evaluación.

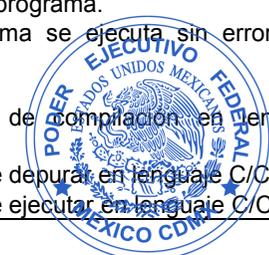




Programa Académico: Técnico en Sistemas Digitales

Unidad de Aprendizaje: Lenguaje de Programación

N° y Nombre de la Práctica:	Prueba de compilador	N° de la Práctica:	3	Tiempo:	3hrs
Unidades del Programa de Estudio:	2. Codificación en Lenguaje C				
Aprendizajes Esperados Relacionados con la Práctica:	Realiza ejercicios en un compilador de lenguaje C				
Contenidos de Aprendizaje					
Conceptuales:	Procedimentales:	Actitudinales:			
<ul style="list-style-type: none"> • Concepto y uso de compilador. • Compilar • Depurar • Ejecutar • Formatos de archivo. 	Realiza ejercicios básicos en un compilador de lenguaje C.	<ul style="list-style-type: none"> • Realiza trabajo colaborativo • Aplica el Pensamiento reflexivo y critico • Emplea el pensamiento y matemático 			
Estrategia Didáctica y Ambiente de Aprendizaje					
<p>Estrategia didáctica. Design thinking</p> <p>El docente: Explica al alumno los conceptos de compilador, depurar, ejecutar aplicados a la programación. Describe el entorno de trabajo del software del lenguaje C/C++. Induce al alumno a utilizar el lenguaje de C/C++ para el desarrollo de programas básicos que permitan demostrar los conceptos de compilador, depurar, ejecutar y tipos de archivo.</p> <p>El estudiante: Realiza un mapa mental sobre que es un lenguaje de programación, su clasificación y características. En base a la explicación e inducción del profesor realiza programas básicos en lenguaje C/C++ que le permitan poner en comprobar la diferencia que hay entre compilar, depurar y ejecutar.</p> <p>AMBIENTE DE APRENDIZAJE Laboratorio de cómputo</p>					
Herramientas Tecnológicas y Recursos Didácticos	Evidencias de Aprendizaje para la Evaluación Formativa	Criterios e Instrumentos de Evaluación			
<p>Herramientas Tecnológicas</p> <ul style="list-style-type: none"> • Proyector • Equipo de cómputo con acceso a Internet <p>Recursos didácticos</p> <ul style="list-style-type: none"> • Problemas computables contextualizados 	Mapa mental donde se explique la función de las herramientas: compilación, depurar, ejecución de un programa en lenguaje C/C++.	<p>Criterios de Forma:</p> <ul style="list-style-type: none"> • Incluye archivos, código fuente de los programas desarrollados. • Entrega en tiempo y forma la evidencia de la práctica. • Entrega el código del programa. • El código del programa se ejecuta sin errores ni warnings. <p>Criterios de Fondo:</p> <ul style="list-style-type: none"> • Explica el concepto de compilación en lenguaje C/C++. • Explica el concepto de depurar en lenguaje C/C++. • Explica el concepto de ejecutar en lenguaje C/C++. 			



Programa Académico: Técnico en Sistemas Digitales

Unidad de Aprendizaje: Lenguaje de Programación

N° y Nombre de la Práctica:	Programas con entrada y salida de datos.	N° de la Práctica:	4	Tiempo:	3hrs
Unidades del Programa de Estudio:	2. Codificación en Lenguaje C				
Aprendizajes Esperados Relacionados con la Práctica:	Emplea las funciones de entrada y salida de datos en lenguaje C.				
Contenidos de Aprendizaje					
Conceptuales:	Procedimentales:	Actitudinales:			
<ul style="list-style-type: none"> Estructura de un programa en C/C++ Funciones de entrada y salida Variables y constantes Tipos de datos Operadores aritméticos, lógicos y relacionales. Jerarquía de operadores. 	<p>Realiza programas en los cuales utilice funciones de salida y entrada con variables y constantes.</p> <p>Realiza programas en los cuales utilice operadores aritméticos, lógicos y relacionales.</p>	<ul style="list-style-type: none"> Realiza trabajo colaborativo Aplica el Pensamiento reflexivo y critico Emplea el pensamiento y matemático 			
Estrategia Didáctica y Ambiente de Aprendizaje					
<p>Estrategia didáctica: Design thinking</p> <ul style="list-style-type: none"> El docente ilustra y explica en código de lenguaje C/C++, las partes que constituyen la forma básica de un programa. El docente explica el uso de las funciones de entrada y salida en lenguaje C/C++. El docente explica la diferencia entre el concepto de variable y constante. El docente explica y hace uso de los tipos de datos del lenguaje C/C++ para definir el uso de cada tipo de dato y su aplicación en un programa. El docente explica la función y uso de los operadores aritméticos, lógicos y relacionales y su jerarquía al momento de aplicarlos en un programa. El estudiante programa (compilar) ejercicios de funciones de salida y entrada. El estudiante investiga las ventajas y desventajas de usar variables y/o constantes para la programación en C/C++. El estudiante investiga los diferentes tipos de datos admisibles en lenguaje C/C++, realizando su cuadro comparativo resaltando las diferencias de estos. <p>AMBIENTE DE APRENDIZAJE</p> <ul style="list-style-type: none"> Laboratorio de computo 					
Herramientas Tecnológicas y Recursos Didácticos	Evidencias de Aprendizaje para la Evaluación Formativa	Criterios e Instrumentos de Evaluación			
<p>HERRAMIENTAS TECNOLÓGICAS</p> <ul style="list-style-type: none"> Computadora con SO Windows Conectividad a internet Software compilador C Proyector <p>RECURSOS DIDÁCTICOS</p> <ul style="list-style-type: none"> Presentaciones y/o apuntes elaborados por el docente Programas de ejemplo 	<ul style="list-style-type: none"> Programas que muestren en pantalla, textos preestablecidos en código de lenguaje C/C++. Programas que ejecuten operaciones matemáticas básicas en conjunto con los operadores lógicos y relacionales. 	<p>De forma:</p> <ul style="list-style-type: none"> Se entrega en formato digital, en la plataforma indicada por el docente. Incluye archivos, código fuente de los programas desarrollados. Entrega en tiempo y forma la evidencia de la práctica. Entrega el código del programa. El código del programa se ejecuta sin errores ni warnings. <p>De fondo:</p> <ul style="list-style-type: none"> Emplea la estructura básica de codificación en lenguaje C/C++. Utiliza correctamente la estructura de las funciones de entrada y salida. Respeto los tipos de datos necesarios para la ejecución de su programa. Utiliza correctamente los operadores aritméticos, lógicos y relacionales. Jerarquía de operadores. <p>Instrumento de evaluación Lista de cotejo</p>			





Programa Académico: Técnico en Sistemas Digitales

Unidad de Aprendizaje: Lenguaje de Programación

N° y Nombre de la Práctica:	Estructuras de control secuenciales	N° de la Práctica:	5	Tiempo:	3hrs
Unidades del Programa de Estudio:	2. Codificación en Lenguaje C				
Aprendizajes Esperados Relacionados con la Práctica:	Emplea las estructuras de control secuencial en un programa en lenguaje C.				

Contenidos de Aprendizaje

Conceptuales:	Procedimentales:	Actitudinales:
Definición de estructuras de control secuenciales.	Realiza programas que resuelvan problemas relacionados a su entorno, mediante estructuras de control secuenciales.	<ul style="list-style-type: none"> Realiza trabajo colaborativo Aplica el Pensamiento reflexivo y critico Emplea el pensamiento y matemático

Estrategia Didáctica y Ambiente de Aprendizaje

Estrategia didáctica: Aprendizaje basado en problemas

- El docente explica mediante diagramas de flujo la forma en la que trabajan las estructuras de control secuencial.
- El docente demuestra cómo funcionan las estructuras de control secuenciales y su aplicación para la solución de problemas en lenguaje C.
- El estudiante realiza programas en lenguaje C/C++ donde resuelva problemas por medio la aplicación de estructuras de control secuenciales.

AMBIENTE DE APRENDIZAJE

- Laboratorio de computo

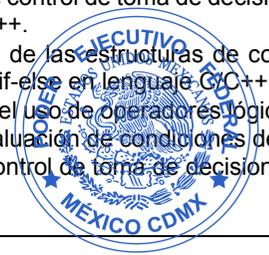
Herramientas Tecnológicas y Recursos Didácticos	Evidencias de Aprendizaje para la Evaluación Formativa	Criterios e Instrumentos de Evaluación
<p>HERRAMIENTAS TECNOLÓGICAS</p> <ul style="list-style-type: none"> Computadora con SO Windows Conectividad a internet Software compilador C Proyector <p>RECURSOS DIDÁCTICOS</p> <ul style="list-style-type: none"> Presentaciones y/o apuntes elaborados por el docente Programas de ejemplo 	<ul style="list-style-type: none"> Códigos de programa en los que, por medio de estructuras de control secuenciales, el uso de operadores y variables resuelvan problemas relacionados al entorno del alumno. 	<p>De forma:</p> <ul style="list-style-type: none"> Incluye archivos, código fuente de los programas desarrollados. Entrega en tiempo y forma la evidencia de la práctica. Entrega el código del programa. El código del programa se ejecuta sin errores ni warnings. <p>De fondo:</p> <ul style="list-style-type: none"> Emplea la estructura de control secuencial en lenguaje C/C++. Utiliza correctamente la estructura de las funciones de entrada y salida. Respeto los tipos de datos necesarios para la ejecución de su programa. Utiliza la jerarquía de operaciones, respetando la prioridad de los operadores. <p>Instrumento de evaluación Lista de cotejo</p>



Programa Académico: Técnico en Sistemas Digitales

Unidad de Aprendizaje: Lenguaje de Programación

N° y Nombre de la Práctica:	Estructuras de control if / if-else.	N° de la Práctica:	6	Tiempo:	3hrs
Unidades del Programa de Estudio:	2. Codificación en Lenguaje C.				
Aprendizajes Esperados Relacionados con la Práctica:	Emplea las estructuras de control de toma de decisiones en un programa en lenguaje C.				
Contenidos de Aprendizaje					
Conceptuales:	Procedimentales:	Actitudinales:			
Estructuras de control de toma de decisiones. <ul style="list-style-type: none"> Ciclo if Ciclo if-else Anidación de sentencias de control de toma de decisiones.	Realiza programas que resuelvan problemas relacionados a su entorno, mediante estructuras de control de toma de decisiones.	<ul style="list-style-type: none"> Realiza trabajo colaborativo Aplica el Pensamiento reflexivo y critico Emplea el pensamiento y matemático 			
Estrategia Didáctica y Ambiente de Aprendizaje					
Estrategia didáctica: Aprendizaje basado en problemas <ul style="list-style-type: none"> El docente explica cómo funcionan las estructuras de control de toma de decisiones y su aplicación en un programa en lenguaje C. El docente explica cómo funciona el anidamiento de las estructuras de control de toma de decisiones y su aplicación en un programa en lenguaje C. El docente demuestra cómo funcionan las estructuras de control de toma de decisiones su aplicación para la solución de problemas en lenguaje C. El estudiante realiza programas en lenguaje C/C++ donde aplique estructuras de control de toma de decisiones para la resolución de problemas. El estudiante realiza programas en lenguaje C/C++ donde aplique el anidamiento de estructuras de control de toma de decisiones para la resolución de problemas. 					
AMBIENTE DE APRENDIZAJE					
Laboratorio de cómputo					
Herramientas Tecnológicas y Recursos Didácticos	Evidencias de Aprendizaje para la Evaluación Formativa	Criterios e Instrumentos de Evaluación			
HERRAMIENTAS TECNOLÓGICAS <ul style="list-style-type: none"> Computadora con SO Windows Conectividad a internet Software compilador C Proyector RECURSOS DIDÁCTICOS <ul style="list-style-type: none"> Presentaciones y/o apuntes elaborados por el docente Programas de ejemplo 	<ul style="list-style-type: none"> Códigos de programa en los que por medio de estructuras de control de toma de decisiones realice la evaluación de condiciones para la solución de problemas relacionados al entorno del alumno. 	De forma: <ul style="list-style-type: none"> Incluye archivos, código fuente de los programas desarrollados. Entrega en tiempo y forma la evidencia de la práctica. Entrega el código del programa. El código del programa se ejecuta sin errores ni warnings. De fondo: <ul style="list-style-type: none"> Emplea la estructura de control de toma de decisiones if en lenguaje C/C++. Emplea la estructura de control de toma de decisiones if-else en lenguaje C/C++. Emplea el anidamiento de las estructuras de control de toma de decisiones if-else en lenguaje C/C++. Emplea correctamente el uso de operadores lógicos y relacionales para la evaluación de condiciones dentro de las estructuras de control de toma de decisiones. Instrumento de evaluación Lista de cotejo			





Programa Académico: Técnico en Sistemas Digitales

Unidad de Aprendizaje: Lenguaje de Programación

N° y Nombre de la Práctica:	Estructuras de control con if-else-if / switch	N° de la Práctica:	7	Tiempo:	3hrs
Unidades del Programa de Estudio:	2. Codificación en Lenguaje C.				
Aprendizajes Esperados Relacionados con la Práctica:	Emplea las estructuras de control selectivas anidadas y múltiples en un programa en lenguaje C.				

Contenidos de Aprendizaje		
Conceptuales:	Procedimentales:	Actitudinales:
Estructuras de control selectivas: <ul style="list-style-type: none"> • Ciclo de control if-else-if • Ciclo de control switch 	Realiza programas que resuelvan problemas relacionados a su entorno, mediante estructuras de control selectivas.	<ul style="list-style-type: none"> • Realiza trabajo colaborativo • Aplica el Pensamiento reflexivo y critico • Emplea el pensamiento y matemático

Estrategia Didáctica y Ambiente de Aprendizaje

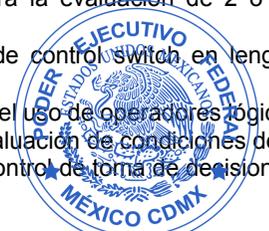
Estrategia didáctica: Aprendizaje basado en problemas

- El docente explica mediante diagramas de flujo cómo funcionan las estructuras de control de selectivas en un programa en lenguaje C.
- El docente explica mediante diagramas de flujo cómo funciona la estructura de control if-else-if y su aplicación en un programa en lenguaje C.
- El docente explica mediante diagramas de flujo cómo funciona la estructura de control switch y su aplicación en un programa en lenguaje C
- El estudiante realiza programas en lenguaje C/C++ donde aplique la estructura de control if-else-if para la resolución de problemas.
- El estudiante realiza programas donde aplique la estructura de control switch para la resolución de problemas.

AMBIENTE DE APRENDIZAJE

Laboratorio de cómputo

Herramientas Tecnológicas y Recursos Didácticos	Evidencias de Aprendizaje para la Evaluación Formativa	Criterios e Instrumentos de Evaluación
HERRAMIENTAS TECNOLÓGICAS <ul style="list-style-type: none"> • Computadora con SO Windows • Conectividad a internet • Software compilador C • Proyector RECURSOS DIDÁCTICOS <ul style="list-style-type: none"> • Presentaciones y/o apuntes elaborados por el docente • Programas de ejemplo • 	<ul style="list-style-type: none"> • Códigos de programa en los que por medio de estructuras de control selectivas resuelva problemas relacionados al entorno del alumno. 	De forma: <ul style="list-style-type: none"> • Incluye archivos, código fuente de los programas desarrollados. • Entrega en tiempo y forma la evidencia de la práctica. • Entrega el código del programa. • El código del programa se ejecuta sin errores ni warnings. De fondo: <ul style="list-style-type: none"> • Emplea la estructura de control if-else-if en lenguaje C/C++. • Realiza correctamente el anidamiento de la estructura de control if-else-if para la evaluación de 2 o más condiciones. • Emplea la estructura de control switch en lenguaje C/C++. • Emplea correctamente el uso de operadores lógicos y relacionales para la evaluación de condiciones dentro de las estructuras de control de toma de decisiones. Instrumento de evaluación Lista de cotejo





Programa Académico: Técnico en Sistemas Digitales

Unidad de Aprendizaje: Lenguaje de Programación

N° y Nombre de la Práctica:	Estructuras de control for.	N° de la Práctica:	8	Tiempo:	3hrs
Unidades del Programa de Estudio:	2. Codificación en Lenguaje C.				
Aprendizajes Esperados Relacionados con la Práctica:	Emplea las estructuras de control desde/para en un programa en lenguaje C.				

Contenidos de Aprendizaje		
Conceptuales:	Procedimentales:	Actitudinales:
Estructura de control repetitiva: <ul style="list-style-type: none"> • Ciclo For • Ciclos For anidados. 	Realiza programas que resuelvan problemas relacionados a su entorno, mediante estructuras de control repetitivas.	<ul style="list-style-type: none"> • Realiza trabajo colaborativo • Aplica el Pensamiento reflexivo y critico • Emplea el pensamiento y matemático

Estrategia Didáctica y Ambiente de Aprendizaje

Estrategia didáctica: Aprendizaje basado en problemas

- El docente explica cómo funcionan las estructuras de control de repetitivas en un programa en lenguaje C.
- El docente explica cómo funciona la estructura de control for y su aplicación en un programa en lenguaje C.
- El docente explica cómo funciona el anidamiento del ciclo de control for y su aplicación en un programa en lenguaje C.
- El estudiante realiza programas donde aplique la estructura de control for para la resolución de problemas.
- El estudiante realiza programas donde aplique el anidamiento de ciclos de control for para la resolución de problemas.

AMBIENTE DE APRENDIZAJE

Laboratorio de cómputo

Herramientas Tecnológicas y Recursos Didácticos	Evidencias de Aprendizaje para la Evaluación Formativa	Criterios e Instrumentos de Evaluación
HERRAMIENTAS TECNOLÓGICAS <ul style="list-style-type: none"> • Computadora con SO Windows • Conectividad a internet • Software compilador C • Proyector RECURSOS DIDÁCTICOS <ul style="list-style-type: none"> • Presentaciones y/o apuntes elaborados por el docente • Programas de ejemplo • 	<ul style="list-style-type: none"> • Códigos de programa en los que por medio de estructuras de control repetitivas resuelva problemas relacionados al entorno del alumno. 	De forma: <ul style="list-style-type: none"> • Incluye archivos, código fuente de los programas desarrollados. • Entrega en tiempo y forma la evidencia de la práctica. • Entrega el código del programa. • El código del programa se ejecuta sin errores ni warnings. De fondo: <ul style="list-style-type: none"> • Emplea la estructura de control for en lenguaje C/C++. • Emplea correctamente el uso de operadores lógicos y relacionales para la evaluación de condiciones dentro de las estructuras de control repetitivas. • Emplea correctamente los operadores aritméticos para el incremento o decremento de variables para el control del número de repeticiones de la sentencia de control. • Aplica correctamente el anidamiento del ciclo de control for para la solución de problemas. Instrumento de evaluación Lista de cotejo





Programa Académico: Técnico en Sistemas Digitales

Unidad de Aprendizaje: Lenguaje de Programación

N° y Nombre de la Práctica:	Estructuras de control while / do-while.	N° de la Práctica:	9	Tiempo:	3hrs
Unidades del Programa de Estudio:	2. Codificación en Lenguaje C.				
Aprendizajes Esperados Relacionados con la Práctica:	Emplea las estructuras de control mientras/haz mientras en un programa en lenguaje C.				
Contenidos de Aprendizaje					
Conceptuales:	Procedimentales:	Actitudinales:			
Estructura de control repetitiva: <ul style="list-style-type: none"> Ciclo While Ciclos Do-While 	Realiza programas que resuelvan problemas relacionados a su entorno, mediante estructuras de control repetitivas.	<ul style="list-style-type: none"> Realiza trabajo colaborativo Aplica el Pensamiento reflexivo y critico Emplea el pensamiento y matemático 			
Estrategia Didáctica y Ambiente de Aprendizaje					
Estrategia didáctica: Aprendizaje basado en problemas <ul style="list-style-type: none"> El docente explica mediante diagramas de flujo la forma en la que trabajan las estructuras de control While. El docente demuestra cómo funcionan la estructura de control While y su aplicación para la solución de problemas en lenguaje C. El docente explica mediante diagramas de flujo la forma en la que trabajan las estructuras de control do-While. El docente demuestra cómo funcionan la estructura de control do-While y su aplicación para la solución de problemas en lenguaje C. El estudiante realiza programas en lenguaje C/C++ donde resuelva problemas por medio la aplicación de la estructura de control While. El estudiante realiza programas en lenguaje C/C++ donde resuelva problemas por medio la aplicación de la estructura de control do-While 					
AMBIENTE DE APRENDIZAJE					
Laboratorio de cómputo					
Herramientas Tecnológicas y Recursos Didácticos	Evidencias de Aprendizaje para la Evaluación Formativa	Criterios e Instrumentos de Evaluación			
HERRAMIENTAS TECNOLÓGICAS <ul style="list-style-type: none"> Computadora con SO Windows Conectividad a internet Software compilador C Proyector RECURSOS DIDÁCTICOS <ul style="list-style-type: none"> Presentaciones y/o apuntes elaborados por el docente Programas de ejemplo 	<ul style="list-style-type: none"> Códigos de programa en los que por medio de estructuras de control repetitivas resuelva problemas relacionados al entorno del alumno. 	De forma: <ul style="list-style-type: none"> Incluye archivos, código fuente de los programas desarrollados. Entrega en tiempo y forma la evidencia de la práctica. Entrega el código del programa. El código del programa se ejecuta sin errores ni warnings De fondo: <ul style="list-style-type: none"> Emplea la estructura de control While en lenguaje C/C++. Emplea la estructura de control Do-While en lenguaje C/C++. Emplea correctamente el uso de operadores lógicos y relacionales para la evaluación de condiciones dentro de las estructuras de control repetitivas. Emplea correctamente los operadores aritméticos para el incremento o decremento de variables para el control del número de repeticiones de la sentencia de control. Instrumento de evaluación Lista de cotejo			





Programa Académico: Técnico en Sistemas Digitales

Unidad de Aprendizaje: Lenguaje de Programación

Nombre de la Práctica: Uso de vectores		N° de la Práctica: 10	Tiempo: 3hrs
Unidades del Programa de Estudio: Arreglo de datos			
Aprendizajes Esperados Relacionados con la Práctica: Aplica arreglos en la solución de problemáticas			
Contenidos de Aprendizaje			
Conceptuales:	Procedimentales:	Actitudinales:	
Arreglos bidimensionales <ul style="list-style-type: none"> • Concepto de Arreglo unidimensional. • Estructura de un arreglo unidimensional (vector). • Manejo y uso del arreglo unidimensional (definición, inicialización, llenado, lectura, recorrido, impresión). Operaciones básicas con arreglos unidimensional.	Aplica los arreglos unidimensionales para la resolución de problemas.	<ul style="list-style-type: none"> • Realiza trabajo colaborativo • Aplica el Pensamiento reflexivo y critico • Emplea el pensamiento y matemático 	
Estrategia Didáctica y Ambiente de Aprendizaje			
ESTRATEGIA DIDÁCTICA. Aprendizaje basado en problemas. <ul style="list-style-type: none"> • El docente explica le principio de funcionamiento de los arreglos en lenguaje C. • El docente demuestra cómo funcionan los arreglos unidimensionales para la solución de problemas en lenguaje C. • El estudiante realiza programas en lenguaje C/C++ donde aplique los arreglos unidimensionales para la resolución de problemas. AMBIENTE DE APRENDIZAJE Laboratorio de cómputo			
Herramientas Tecnológicas y Recursos Didácticos	Evidencias de Aprendizaje para la Evaluación Formativa	Criterios e Instrumentos de Evaluación	
HERRAMIENTAS TECNOLÓGICAS <ul style="list-style-type: none"> • Computadora con SO Windows • Conectividad a internet • Software compilador C • Proyector RECURSOS DIDÁCTICOS <ul style="list-style-type: none"> • Presentaciones y/o apuntes elaborados por el docente • Programas de ejemplo 	<ul style="list-style-type: none"> • Códigos de programa en los que por medio de arreglos unidimensionales de solución a problemas relacionados al entorno del alumno. 	De forma: <ul style="list-style-type: none"> • Incluye archivos, código fuente de los programas desarrollados. • Entrega en tiempo y forma la evidencia de la práctica. • Entrega el código del programa. • El código del programa se ejecuta sin errores ni warnings. De fondo: <ul style="list-style-type: none"> • Declara correctamente los arreglos unidimensionales. • Utiliza ciclos de control repetitivo para el manejo de los arreglos. • Asigna datos a las diferentes posiciones del arreglo para realizar operaciones. • Realiza operaciones con arreglos. • Asigna los resultados de las operaciones realizadas entre arreglos a un arreglo final. • Despliega en pantalla los contenidos de los arreglos utilizados para la realización de las operaciones. Instrumento de Evaluación <ul style="list-style-type: none"> • Lista de cotejo 	





Programa Académico: Técnico en Sistemas Digitales

Unidad de Aprendizaje: Lenguaje de Programación

N° y Nombre de la Práctica:	Uso de matrices.	N° de la Práctica:	11	Tiempo:	2hrs
Unidades del Programa de Estudio:	Arreglo de Datos				
Aprendizajes Esperados Relacionados con la Práctica:	Emplea arreglos bidimensionales en la solución de problemáticas				
Contenidos de Aprendizaje					
Conceptuales:	Procedimentales:	Actitudinales:			
Arreglos bidimensionales <ul style="list-style-type: none"> • Concepto de Arreglo bidimensional. • Estructura de un arreglo bidimensional(matriz). • Manejo y uso del arreglo bidimensional (definición, inicialización, llenado, lectura, recorrido, impresión). • Operaciones básicas con arreglos bidimensionales. 	Aplica los arreglos bidimensionales para la resolución de problemas.	<ul style="list-style-type: none"> • Realiza trabajo colaborativo • Aplica el Pensamiento reflexivo y critico • Emplea el pensamiento y matemático 			
Estrategia Didáctica y Ambiente de Aprendizaje					
ESTRATEGIA DIDÁCTICA. Aprendizaje basado en problemas. <ul style="list-style-type: none"> • El docente explica le principio de funcionamiento de los arreglos bidimensional en lenguaje C. • El docente demuestra cómo funcionan los arreglos bidimensionales para la solución de problemas en lenguaje C. • El estudiante realiza programas en lenguaje C/C++ donde aplique los arreglos bidimensionales para la resolución de problemas. 					
AMBIENTE DE APRENDIZAJE <ul style="list-style-type: none"> • Laboratorio de cómputo 					
Herramientas Tecnológicas y Recursos Didácticos	Evidencias de Aprendizaje para la Evaluación Formativa	Criterios e Instrumentos de Evaluación			
HERRAMIENTAS TECNOLÓGICAS <ul style="list-style-type: none"> • Computadora con SO Windows • Conectividad a internet • Software compilador C • Proyector RECURSOS DIDÁCTICOS <ul style="list-style-type: none"> • Presentaciones y/o apuntes elaborados por el docente • Programas de ejemplo • 	<ul style="list-style-type: none"> • Códigos de programa en los que por medio de arreglos bidimensionales de solución a problemas relacionados al entorno del alumno. 	De forma: <ul style="list-style-type: none"> • Incluye archivos, código fuente de los programas desarrollados. • Entrega en tiempo y forma la evidencia de la práctica. • Entrega el código del programa. • El código del programa se ejecuta sin errores ni warnings De fondo: <ul style="list-style-type: none"> • Declara correctamente los arreglos bidimensionales. • Utiliza ciclos de control repetitivo anidados para el manejo de los arreglos. • Asigna datos a las diferentes posiciones de los arreglos bidimensionales para realizar operaciones. • Realiza operaciones con arreglos bidimensionales. • Asigna los resultados de las operaciones realizadas entre arreglos bidimensionales a un arreglo bidimensional final. • Despliega en pantalla los contenidos de los arreglos bidimensionales utilizados para la realización de las operaciones. Instrumento de Evaluación Lista de cotejo			





Programa Académico: Técnico en Sistemas Digitales

Unidad de Aprendizaje: Lenguaje de Programación

N° y Nombre de la Práctica:	Uso de cadenas de caracteres.	N° de la Práctica:	12	Tiempo:	2hrs
Unidades del Programa de Estudio:	Arreglos de Datos				
Aprendizajes Esperados Relacionados con la Práctica:	Emplea cadenas de texto mediante lenguaje de programación				
Contenidos de Aprendizaje					
Conceptuales:	Procedimentales:	Actitudinales:			
<ul style="list-style-type: none"> Concepto de Cadena de caracteres (arreglo de tipo texto) Importancia del fin de cadena '\0' Manejo y uso de la cadena de caracteres (definición, inicialización, llenado, lectura, recorrido, impresión). 	<p>Realiza programas donde por medio del uso de cadenas de caracteres resuelvan problemas relacionados a su entorno.</p>	<ul style="list-style-type: none"> Realiza trabajo colaborativo Aplica el Pensamiento reflexivo y critico Emplea el pensamiento y matemático 			
Estrategia Didáctica y Ambiente de Aprendizaje					
ESTRATEGIA DIDÁCTICA: Aprendizaje basado en problemas					
<ul style="list-style-type: none"> El docente expone la definición y características de una cadena de caracteres El docente muestra el uso de las cadenas de caracteres y sus reglas de uso (fin de cadena). El estudiante realiza programas donde aplique cadenas de caracteres para la solución de problemas. 					
AMBIENTE DE APRENDIZAJE					
Laboratorio de cómputo					
Herramientas Tecnológicas y Recursos Didácticos	Evidencias de Aprendizaje para la Evaluación Formativa	Criterios e Instrumentos de Evaluación			
<p>HERRAMIENTAS TECNOLÓGICAS</p> <ul style="list-style-type: none"> Computadora con SO Windows Conectividad a internet Software compilador C Proyector <p>RECURSOS DIDÁCTICOS</p> <ul style="list-style-type: none"> Presentaciones y/o apuntes elaborados por el docente Programas de ejemplo 	<ul style="list-style-type: none"> Códigos de programa en los que por medio de cadenas de caracteres de solución a problemas relacionados al entorno del alumno. 	<p>De forma:</p> <ul style="list-style-type: none"> Incluye archivos, código fuente de los programas desarrollados. Entrega en tiempo y forma la evidencia de la práctica. Entrega el código del programa. El código del programa se ejecuta sin errores ni warnings <p>De fondo:</p> <ul style="list-style-type: none"> Declara correctamente las cadenas de caracteres. Utiliza ciclos de control repetitivo anidados para el manejo de las cadenas de caracteres. Emplea cadenas de caracteres para la representación de información como palabras y/o enunciados en soluciones a problemas con lenguaje C Despliega en pantalla los contenidos de las cadenas de caracteres. <p>Instrumento de Evaluación Lista de cotejo</p>			





Programa Académico: Técnico en Sistemas Digitales

Unidad de Aprendizaje: Lenguaje de Programación

N° y Nombre de la Práctica:	Operaciones con cadenas de caracteres.	N° de la Práctica:	13	Tiempo:	2hrs
Unidades del Programa de Estudio:	Arreglos de Datos				
Aprendizajes Esperados Relacionados con la Práctica:	Emplea operaciones con cadenas de caracteres mediante lenguaje de programación				
Contenidos de Aprendizaje					
Conceptuales:	Procedimentales:	Actitudinales:			
Manejo y uso de la cadena de caracteres (definición, inicialización, llenado, lectura, recorrido, impresión). Operaciones con cadenas de caracteres.	Códigos de programa en los que por medio de operación de cadenas de caracteres de solución a problemas relacionados al entorno del alumno.	<ul style="list-style-type: none"> Realiza trabajo colaborativo Aplica el Pensamiento reflexivo y critico Emplea el pensamiento y matemático 			
Estrategia Didáctica y Ambiente de Aprendizaje					
ESTRATEGIA DIDÁCTICA: Aprendizaje basado en problemas					
<ul style="list-style-type: none"> El docente expone como realizar operaciones con cadenas de caracteres El estudiante realiza programas donde realice operaciones con cadenas de caracteres para la solución de problemas. 					
AMBIENTE DE APRENDIZAJE					
Laboratorio de cómputo					
Herramientas Tecnológicas y Recursos Didácticos	Evidencias de Aprendizaje para la Evaluación Formativa	Criterios e Instrumentos de Evaluación			
HERRAMIENTAS TECNOLÓGICAS <ul style="list-style-type: none"> Computadora con SO Windows Conectividad a internet Software compilador C Proyector RECURSOS DIDÁCTICOS <ul style="list-style-type: none"> Presentaciones y/o apuntes elaborados por el docente Programas de ejemplo 	Códigos de programa en los que por medio de operaciones con cadenas de caracteres de solución a problemas relacionados al entorno del alumno.	De forma: <ul style="list-style-type: none"> Incluye archivos, código fuente de los programas desarrollados. Entrega en tiempo y forma la evidencia de la práctica. Entrega el código del programa. El código del programa se ejecuta sin errores ni warnings De fondo: <ul style="list-style-type: none"> Declara correctamente las cadenas de caracteres. Utiliza ciclos de control repetitivo anidados para el manejo de las cadenas de caracteres. Realiza operaciones con cadenas de caracteres para dar solución a problemas en lenguaje C. Despliega en pantalla los resultados de operaciones con cadenas de caracteres. Instrumento de Evaluación Lista de cotejo			





Programa Académico: Técnico en Sistemas Digitales

Unidad de Aprendizaje: Lenguaje de Programación

Nombre de la Práctica:	Funciones matemáticas	N° de la Práctica:	14	Tiempo:	3hrs
Unidades del Programa de Estudio:	Funciones				
Aprendizajes Esperados Relacionados con la Práctica:	Utiliza Funciones de librería en la resolución de problemas				
Contenidos de Aprendizaje					
Conceptuales:	Procedimentales:	Actitudinales:			
Prototipos generales de funciones Catálogo de Funciones	Conoce la sintaxis para la declaración de funciones, Identifica las principales funciones matemáticas propias del compilador	<ul style="list-style-type: none"> • Trabajo colaborativo • Perseverancia • Pensamiento reflexivo y crítico • Creatividad e Innovación • Comunicación 			
Estrategia Didáctica y Ambiente de Aprendizaje					
<p>Estrategias didácticas: Aprendizaje basado en problemas El docente expone la librería matemática y algunas de sus funciones principales: potencia, raíz cuadrada, funciones trigonométricas. El docente presenta programas que muestren el uso de funciones matemáticas. El estudiante resuelve ejercicios propuestos por el docente donde aplique las funciones matemáticas.</p>					
AMBIENTE DE APRENDIZAJE Laboratorio de cómputo					
Herramientas Tecnológicas y Recursos Didácticos	Evidencias de Aprendizaje para la Evaluación Formativa	Criterios e Instrumentos de Evaluación			
<p>Herramientas Tecnológicas: Computadora con S.O. Windows Software compilador C Proyector</p> <p>Recursos Didácticos: Presentaciones y/o Apuntes elaborados por el docente Programas de ejemplo Manual de ejercicios propuestos por el docente.</p>	<p>Práctica 14. Funciones matemáticas.</p> <ul style="list-style-type: none"> - Codificado en el lenguaje establecido - Programa ejecutable. - Resultados esperados - Prueba de escritorio 	<p>Criterios de forma:</p> <ul style="list-style-type: none"> • Comentarios dentro del código con los datos personales del estudiante y grupo • Alineación y espaciado adecuado en la codificación • Código fácil de lectura • Secuencia y Lógica <p>Criterios de fondo:</p> <ul style="list-style-type: none"> • Código funcionando • El código resuelve la problemática planteada <p>Instrumento de evaluación:</p> <ul style="list-style-type: none"> • Lista de Cotejo. 			





Programa Académico: Técnico en Sistemas Digitales

Unidad de Aprendizaje: Lenguaje de Programación

Nombre de la Práctica:	Funciones de sistema.	N° de la Práctica:	15	Tiempo:	3hrs
Unidades del Programa de Estudio:	Funciones				
Aprendizajes Esperados Relacionados con la Práctica:	Utiliza Funciones de librería en la resolución de problemas				
Contenidos de Aprendizaje					
Conceptuales:	Procedimentales:	Actitudinales:			
Prototipos generales de funciones Catálogo de Funciones	Identifica otras funciones propias del compilador de uso frecuente (de entrada y salida, limpiar ventana, cambiar color de fondo y texto, etc.)	<ul style="list-style-type: none"> Realiza trabajo colaborativo Aplica la perseverancia Usa el Pensamiento reflexivo y critico Aplica la Creatividad e Innovación Emplea la Comunicación 			
Estrategia Didáctica y Ambiente de Aprendizaje					
<p>Estrategias didácticas: Aprendizaje basado en problemas El docente muestra el uso de funciones de uso frecuente en diversos programas utilizadas durante el curso: entrada y salida, limpiar ventana, cambiar color de fondo y del texto, esperar carácter. El estudiante resuelve diversos ejercicios propuestos por el docente donde aplique diversas funciones propias del compilador.</p>					
<p>AMBIENTE DE APRENDIZAJE Laboratorio de cómputo</p>					
Herramientas Tecnológicas y Recursos Didácticos	Evidencias de Aprendizaje para la Evaluación Formativa	Criterios e Instrumentos de Evaluación			
<p>Herramientas Tecnológicas: Computadora con S.O. Windows Software compilador C Proyector</p> <p>Recursos Didácticos: Presentaciones y/o Apuntes elaborados por el docente Programas de ejemplo Manual de ejercicios propuestos por el docente.</p>	<p>Reporte de la Práctica 15. Funciones de sistema:</p> <ul style="list-style-type: none"> - Codificado en el lenguaje establecido - Programa ejecutable. - Resultados esperados - Prueba de escritorio 	<p>Criterios de forma:</p> <ul style="list-style-type: none"> Comentarios dentro del código con los datos personales del estudiante y grupo Alineación y espaciado adecuado en la codificación Código fácil de lectura Secuencia lógica <p>Criterios de fondo:</p> <ul style="list-style-type: none"> Código funcionando El código resuelve la problemática planteada <p>Instrumento de evaluación:</p> <ul style="list-style-type: none"> Lista de Cotejo. 			





Programa Académico: Técnico en Sistemas Digitales

Unidad de Aprendizaje: Lenguaje de Programación

Nombre de la Práctica:	Otras funciones	N° de la Práctica:	16	Tiempo:	3hrs
Unidades del Programa de Estudio:	Funciones				
Aprendizajes Esperados Relacionados con la Práctica:	Utiliza Funciones de librería en la resolución de problemas				
Contenidos de Aprendizaje					
Conceptuales:	Procedimentales:	Actitudinales:			
Prototipos generales de funciones Catálogo de Funciones	Utiliza otras funciones propias del compilador que ayuden a mejorar la codificación como las funciones para manejo de cadenas.	<ul style="list-style-type: none"> Realiza trabajo colaborativo Aplica la perseverancia Usa el Pensamiento reflexivo y critico Aplica la Creatividad e Innovación Emplea la Comunicación 			
Estrategia Didáctica y Ambiente de Aprendizaje					
<p>Estrategia didáctica: Aprendizaje basado en problemas. El docente muestra funciones básicas de cadena, para la manipulación de datos. El estudiante resuelve diversos ejercicios propuestos por el docente donde aplique diversas funciones propias del compilador.</p> <p>AMBIENTE DE APRENDIZAJE Laboratorio de cómputo</p>					
Herramientas Tecnológicas y Recursos Didácticos	Evidencias de Aprendizaje para la Evaluación Formativa	Criterios e Instrumentos de Evaluación			
<p>Herramientas Tecnológicas: Computadora con S.O. Windows Software compilador C Proyector</p> <p>Recursos Didácticos: Presentaciones y/o Apuntes elaborados por el docente Programas de ejemplo Manual de ejercicios propuestos por el docente.</p>	<p>Practica 16. Otras funciones.</p> <ul style="list-style-type: none"> Codificado en el lenguaje establecido Programa ejecutable. Resultados esperados Prueba de escritorio 	<p>Criterios de forma:</p> <ul style="list-style-type: none"> Comentarios dentro del código con los datos personales del estudiante y grupo Alineación y espaciado adecuado en la codificación Código fácil de lectura Secuencia lógica <p>Criterios de fondo:</p> <ul style="list-style-type: none"> Código funcionando El código resuelve la problemática planteada <p>Instrumento de evaluación:</p> <ul style="list-style-type: none"> Lista de Cotejo. 			





Programa Académico: Técnico en Sistemas Digitales

Unidad de Aprendizaje: Lenguaje de Programación

Nombre de la Práctica:	Funciones de retorno por valor.	N° de la Práctica:	17	Tiempo:	3hrs
Unidades del Programa de Estudio:	Funciones				
Aprendizajes Esperados Relacionados con la Práctica:	Crea funciones de usuario aplicables a una problemática específica.				
Contenidos de Aprendizaje					
Conceptuales:	Procedimentales:	Actitudinales:			
Funciones de retorno por valor.	Identifica la forma de declarar funciones que regresan valores con y sin parámetros.	<ul style="list-style-type: none"> Realiza trabajo colaborativo Aplica la perseverancia Usa el Pensamiento reflexivo y critico Aplica la Creatividad e Innovación Emplea la Comunicación 			
Estrategia Didáctica y Ambiente de Aprendizaje					
<p>Estrategias didácticas: Aprendizaje basado en problemas El docente expone las variables globales y las variables locales. El docente expone las funciones de retorno por valor, con y sin parámetros. El docente presenta programas que muestren el uso de funciones de retorno por valor, con y sin parámetros. El estudiante resuelve ejercicios propuestos por el docente donde aplique el uso de funciones que retornen valores con y sin parámetros.</p> <p>AMBIENTE DE APRENDIZAJE Laboratorio de cómputo</p>					
Herramientas Tecnológicas y Recursos Didácticos	Evidencias de Aprendizaje para la Evaluación Formativa	Criterios e Instrumentos de Evaluación			
<p>Herramientas Tecnológicas: Computadora con S.O. Windows Software compilador C Proyector</p> <p>Recursos Didácticos: Presentaciones y/o Apuntes elaborados por el docente Programas de ejemplo Manual de ejercicios propuestos por el docente.</p>	<p>Practica 17: Funciones de retorno por valor.</p> <ul style="list-style-type: none"> Codificado en el lenguaje establecido Programa ejecutable. Resultados esperados Prueba de escritorio 	<p>Criterios de forma:</p> <ul style="list-style-type: none"> Comentarios dentro del código con los datos personales del estudiante y grupo Alineación y espaciado adecuado en la codificación Código fácil de lectura Secuencia lógica <p>Criterios de fondo:</p> <ul style="list-style-type: none"> Código funcionando El código resuelve la problemática planteada <p>Instrumento de evaluación:</p> <ul style="list-style-type: none"> Lista de Cotejo. 			





Programa Académico: Técnico en Sistemas Digitales

Unidad de Aprendizaje: Lenguaje de Programación

Nombre de la Práctica:	Funciones de paso por valor.	N° de la Práctica:	18	Tiempo:	3hrs
Unidades del Programa de Estudio:	Funciones				
Aprendizajes Esperados Relacionados con la Práctica:	Crea funciones de usuario aplicables a una problemática específica.				
Contenidos de Aprendizaje					
Conceptuales:	Procedimentales:	Actitudinales:			
Funciones de paso por valor. Funciones de paso por referencia.	Crea una copia de variables para pasar valores al hacer una llamada de función con argumentos. Conoce los apuntadores, para pasar valores por la dirección de una variable al hacer una llamada de función	<ul style="list-style-type: none"> Realiza trabajo colaborativo Aplica la perseverancia Usa el Pensamiento reflexivo y critico Aplica la Creatividad e Innovación Emplea la Comunicación 			
Estrategia Didáctica y Ambiente de Aprendizaje					
<p>Estrategias didácticas: Aprendizaje basado en problemas El docente expone las funciones de paso por valor con argumentos. El docente expone los apuntadores el lenguaje C El docente expone las funciones de paso por referencia. El docente presenta programas que muestren el uso de funciones de paso por referencia.</p> <p>El estudiante resuelve ejercicios propuestos por el docente donde aplique el uso de funciones que utilicen el paso por valor con argumentos. El estudiante resuelve ejercicios propuestos por el docente donde aplique las funciones de paso por referencia. El estudiante incorpora en otros ejercicios ya resueltos previamente, el manejo de funciones para mejorar la codificación y el desarrollo de sus programas</p>					
AMBIENTE DE APRENDIZAJE					
Laboratorio de cómputo					
Herramientas Tecnológicas y Recursos Didácticos	Evidencias de Aprendizaje para la Evaluación Formativa	Criterios e Instrumentos de Evaluación			
<p>Herramientas Tecnológicas: Computadora con S.O. Windows Software compilador C Proyector</p> <p>Recursos Didácticos: Presentaciones y/o Apuntes elaborados por el docente Programas de ejemplo Manual de ejercicios propuestos por el docente.</p>	<p>Practica 18: Funciones de paso por valor y por referencia.</p> <ul style="list-style-type: none"> Codificado en el lenguaje establecido Programa ejecutable. Resultados esperados Prueba de escritorio 	<p>Criterios de forma:</p> <ul style="list-style-type: none"> Comentarios dentro del código con los datos personales del estudiante y grupo Alineación y espaciado adecuado en la codificación Código fácil de lectura Secuencia lógica <p>Criterios de fondo:</p> <ul style="list-style-type: none"> Código funcionando El código resuelve la problemática planteada <p>Instrumento de evaluación:</p> <ul style="list-style-type: none"> Lista de Cotejo. 			





Programa Académico: Técnico en Sistemas Digitales

Unidad de Aprendizaje: Lenguaje de Programación

PLAN DE EVALUACIÓN SUMATIVA

N°	Unidad de Competencia	Evidencia integradora	Criterios e Instrumentos de Evaluación	Porcentaje de Acreditación
1	Diseña algoritmos que den solución a problemas computables, realizando su análisis y representación a través de Diagramas de Flujo de Datos	Describe soluciones en un documento de texto, a problemas computables a través del análisis de problemas y su diseño, representados a través de Diagramas de Flujo de Datos.	<p>Criterios de evaluación</p> <ul style="list-style-type: none"> Analiza problemas describiendo su definición, especificaciones de entrada y de salida. Realiza el diseño de los problemas planteados a través de DFD. Emplea las reglas para la construcción de DFD. <p>Instrumento de evaluación</p> <ul style="list-style-type: none"> Lista de cotejo 	10%
2	Construye programas codificados en lenguaje de alto nivel, a partir de fundamentos de programación, y el uso de estructuras de control	Aplica soluciones a problemas de Sistemas Digitales a través programas elaborados en el lenguaje de alto nivel, utilizando estructuras de control y funciones de librería.	<p>Criterios de evaluación</p> <ul style="list-style-type: none"> Emplea estructuras de control secuencial, selectivas y repetitivas en lenguaje C. Implementa las estructuras de control en un proyecto que resuelve una problemática <p>Instrumento de evaluación</p> <ul style="list-style-type: none"> Lista de cotejo 	35%
3	Desarrolla arreglos de datos y manipulación de cadenas de caracteres para la solución de problemáticas planteadas	Realiza programas donde aplica los arreglos y/o cadenas de texto en lenguaje como medio de solución de las problemáticas planteadas	<p>Criterios de Fondo</p> <ul style="list-style-type: none"> Emplea arreglos unidimensionales y bidimensionales en lenguaje C para la solución de problemas planteados. Respetar los tipos de datos para los arreglos correspondientes con el tipo de problema. Utiliza operadores entre elementos de diferentes arreglos en la solución de los problemas propuestos. Emplea operaciones con cadenas de texto en lenguaje C para la solución de problemas planteados Utiliza operadores entre elementos de diferentes cadenas de texto en la solución de los problemas propuestos <p>Criterios de Forma:</p> <ul style="list-style-type: none"> Se entrega en formato digital, en la plataforma indicada por el docente. Incluye archivos, código fuente de los programas desarrollados. Se entrega en las fechas establecidas. 	25%





Programa Académico: Técnico en Sistemas Digitales

Unidad de Aprendizaje: Lenguaje de Programación

			Instrumento de evaluación	
4	Construye soluciones a problemáticas planteadas en un lenguaje de alto nivel, implementando funciones básicas del compilador y de usuario.	Desarrolla programas implementando funciones matemáticas, de propósito específico y diseñadas por el propio usuario para la solución de la problemática planteada	<p>Criterios de evaluación</p> <ul style="list-style-type: none"> • Codifica el programa utilizando las funciones respectivas. • Tiene orden y secuencia en su elaboración. • Presenta una prueba de escritorio como parámetro de comparación. • El programa proporciona resultados esperados y con una buena presentación. <p>Instrumento de evaluación</p> <ul style="list-style-type: none"> • Rúbrica 	30%
Propósito General de la unidad de aprendizaje		Evidencia Integradora del Propósito General de una Unidad de Aprendizaje	Criterios e Instrumentos de Evaluación	Porcentaje de Acreditación
Crea soluciones a problemas de Sistemas Digitales mediante programas elaborados en un lenguaje de programación de alto nivel, aplicando estructuras de control y funciones de librería.		Desarrolla un programa que integra estructuras de control, uso de datos como entrada y salida, además de los recursos que provee el compilador para resolver problemas de su ámbito de desarrollo	<p>Criterios de evaluación</p> <ul style="list-style-type: none"> • Implementa librerías y funciones del compilador • Utiliza variables y constantes • Emplea las sentencias de control (secuencial, selectivas y repetitivas) en el funcionamiento del programa. • Utiliza arreglos y estructuras de datos para la administración de entradas y salidas • Crea funciones propias. <p>Instrumento de evaluación</p> <ul style="list-style-type: none"> • Rúbrica de evaluación 	100%





Programa Académico: Técnico en Sistemas Digitales

Unidad de Aprendizaje: Lenguaje de Programación

PROGRAMA SINTÉTICO

PROPÓSITO GENERAL DE LA UNIDAD DE APRENDIZAJE

Crea soluciones a problemas de Sistemas Digitales mediante programas elaborados en un lenguaje de programación de alto nivel, aplicando estructuras de control y funciones de librería.

N°	UNIDAD DE COMPETENCIA	APRENDIZAJES ESPERADOS	CONTENIDOS/SABERES
1	Diseña algoritmos que den solución a problemas computables, realizando su análisis y representación a través de Diagramas de Flujo de Datos	<ol style="list-style-type: none"> 1. Emplea la metodología para solución de problemas aplicables a los Sistemas digitales. 2. Utiliza los elementos y metodología para la representación de algoritmos. 	<p>Conceptuales</p> <ul style="list-style-type: none"> • Análisis de problemas computables • Concepto y características de algoritmos • Tipos de algoritmos (lineales y no lineales) • Diseño de algoritmos • Diagramas de flujo de datos (ANSI – ISO) <p>Procedimentales</p> <ul style="list-style-type: none"> • Análisis de problemas (Definición del problema, especificaciones de entrada y de salida) • Construcción de soluciones en forma de algoritmos. • Diseño de algoritmos en Diagramas de Flujo de Datos. • Diseño algoritmos representados en Diagramas de Flujo de Datos. <p>Actitudinales</p> <ul style="list-style-type: none"> • Emplea el pensamiento lógico y matemático. • Reflexiona sobre los aprendizajes desarrollados. • Se comunica de manera asertiva. • Emplea la creatividad. • Piensa de manera crítica. • Trabajo de forma colaborativa
2	Construye programas codificados en lenguaje de alto nivel, a partir de fundamentos de programación, y el uso de estructuras de control.	<ol style="list-style-type: none"> 1. Aplica los fundamentos de programación en la codificación de soluciones de problemas de Sistemas Digitales. 2. Emplea las estructuras de control en la codificación de soluciones de problemas. 	<p>Conceptuales</p> <ul style="list-style-type: none"> • Clasificación de los tipos de lenguaje de programación • Estructura de un programa en C/C++ • Funciones de entrada y salida • Variables y constantes • Tipos de datos • Detección de errores y desbordamiento de memoria





Programa Académico: Técnico en Sistemas Digitales

Unidad de Aprendizaje: Lenguaje de Programación

PROPÓSITO GENERAL DE LA UNIDAD DE APRENDIZAJE

Crea soluciones a problemas de Sistemas Digitales mediante programas elaborados en un lenguaje de programación de alto nivel, aplicando estructuras de control y funciones de librería.

N°	UNIDAD DE COMPETENCIA	APRENDIZAJES ESPERADOS	CONTENIDOS/SABERES
			<ul style="list-style-type: none"> • Operadores aritméticos, relacionales y lógicos • Expresiones • Definición de estructuras de control secuenciales • Definición de estructuras de control selectivas <ul style="list-style-type: none"> ○ Simple ○ Doble ○ Múltiple • Definición de estructuras de control repetitivas. <ul style="list-style-type: none"> ○ while ○ do-while ○ for <p>Procedimentales</p> <ul style="list-style-type: none"> • Emplea las funciones de entrada y salida de datos en lenguaje C • Emplea las estructuras de control secuencial en un programa en lenguaje C. • Emplea las estructuras de control de toma de decisiones en un programa en lenguaje C. • Emplea las estructuras de control selectivas anidadas y múltiples en un programa en lenguaje C. • Emplea las estructuras de control desde/para en un programa en lenguaje C • Emplea las estructuras de control mientras/haz mientras en un programa en lenguaje C.. <p>Actitudinales</p> <ul style="list-style-type: none"> • Trabajo colaborativamente. • Piensa de manera reflexiva y crítica.
3	Desarrolla arreglos de datos y manipulación de cadenas de caracteres para la solución de problemáticas planteadas.	3. Aplica Arreglos en la solución de problemáticas. 4. Emplea cadenas de texto mediante lenguaje de programación.	<p>Conceptuales</p> <ul style="list-style-type: none"> • Arreglos unidimensionales • Arreglos bidimensionales • Cadenas <p>Procedimentales</p>





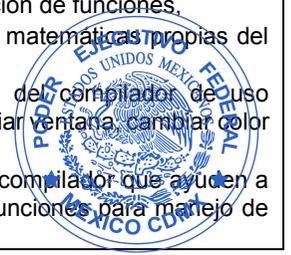
Programa Académico: Técnico en Sistemas Digitales

Unidad de Aprendizaje: Lenguaje de Programación

PROPÓSITO GENERAL DE LA UNIDAD DE APRENDIZAJE

Crea soluciones a problemas de Sistemas Digitales mediante programas elaborados en un lenguaje de programación de alto nivel, aplicando estructuras de control y funciones de librería.

N°	UNIDAD DE COMPETENCIA	APRENDIZAJES ESPERADOS	CONTENIDOS/SABERES
			<ul style="list-style-type: none"> • Emplea vectores / arreglos unidimensionales en soluciones de problemas. • Usa matrices / arreglos bidimensionales en soluciones de problemas. • Emplea cadenas de caracteres para la representación de información como palabras y/o enunciados en soluciones a problemas con lenguaje C • Emplea operaciones con cadenas de caracteres como la ordenación, concatenación entre otras, empleadas en soluciones a problemas con lenguaje C. <p>Actitudinales</p> <ul style="list-style-type: none"> • Trabaja colaborativamente • Piensa de manera reflexiva y crítica
4	<p>Construye soluciones a problemáticas planteadas en un lenguaje de alto nivel, implementando funciones básicas del compilador y de usuario.</p>	<ol style="list-style-type: none"> 1. Utiliza Funciones de librería en la resolución de problemas. 2. Crea funciones de usuario aplicables a una problemática específica. 	<p>Conceptuales</p> <ul style="list-style-type: none"> • Prototipos generales de funciones. • Catálogo de Funciones. • Funciones de retorno por valor. • Funciones de paso por valor. • Funciones de paso por referencia. <p>Procedimentales</p> <ul style="list-style-type: none"> • Conoce la sintaxis para la declaración de funciones. • Identifica las principales funciones matemáticas propias del compilador • Identifica otras funciones propias del compilador de uso frecuente (de entrada y salida, limpiar ventana, cambiar color de fondo y texto, etc.) • Utiliza otras funciones propias del compilador que ayuden a mejorar la codificación como las funciones para manejo de cadenas.





Programa Académico: Técnico en Sistemas Digitales

Unidad de Aprendizaje: Lenguaje de Programación

PROPÓSITO GENERAL DE LA UNIDAD DE APRENDIZAJE

Crea soluciones a problemas de Sistemas Digitales mediante programas elaborados en un lenguaje de programación de alto nivel, aplicando estructuras de control y funciones de librería.

N°	UNIDAD DE COMPETENCIA	APRENDIZAJES ESPERADOS	CONTENIDOS/SABERES
			<ul style="list-style-type: none"> ● Identifica la forma de declarar funciones que regresan valores con y sin parámetros. ● Crea una copia de variables para pasar valores al hacer una llamada de función con argumentos. ● Conoce los apuntadores, para pasar valores por la dirección de una variable al hacer una llamada de función. <p>Actitudinales</p> <ul style="list-style-type: none"> ● Trabaja colaborativamente ● Es perseverancia ● Piensa de manera reflexiva y crítica ● Emplea la creatividad e innovación ● Se comunicación asertivamente





Programa Académico: Técnico en Sistemas Digitales

Unidad de Aprendizaje: Lenguaje de Programación

BIBLIOGRAFÍA BÁSICA Y COMPLEMENTARIA

Número y Nombre de la Unidad Didáctica	FORMATO APA	CLASIFICACIÓN	
		Básico	Consulta
1. Análisis y Diseño de Algoritmos	Joyanes Aguilar, L. (2020). <i>Fundamentos de Programación. Algoritmos, Estructura de datos y Objetos</i> (Quinta ed.). McGraw-Hill.	X	
1. Análisis y Diseño de Algoritmos	Cormen, T., Leiserson, C., Rivest, R., & Stein, C. (2022). <i>Introduction to algorithms</i> . The MIT Press.		X
1. Análisis y Diseño de Algoritmos	Thomas, D., Hunt, A. (2019). <i>The Pragmatic Programmer: Your Journey to MAstery, 20th Aniversary Edition</i> (2nd Ed.). Addison-Wesley Professional		X
2. Codificación en Lenguaje C.	Gregoire, M. (2021). <i>Professional C++</i> . (fifth Edition). Wrox Press		
2. Codificación en Lenguaje C.	Joyanes. L. (2020). <i>Fundamentos de Programación</i> . (Quinta Edición). McGraw-Hill		
3. Programar en C	Vozmediano, A.M. (2021). <i>Aprender a programar en C: de 0 a 99 en un solo libro</i> . Independiente		X
3. Programar en C	Kernighan, B., Ritchie, D. (2021). <i>El lenguaje de programación C</i> . (Spanish Ed.). Independiente	X	
4. Funciones	Rodríguez-Lozada, D., Muñoz, J., García C. (2022). <i>Introducción a la programación en C</i> . (Primera Ed.) UPM Press		X
4. Funciones	Deitel, P., Deitel, H., (2022). <i>C How to Program</i> . (ninth Ed.). Pearson		X

