





Programa de Estudios de la Unidad de Aprendizaje:						LENGUAJE DE PROGRAMACIÓN																						
Clave:	4FP-FM785					Créditos:	4.5					Programa Académico: TÉCNICO EN MECATRÓNICA																
Ramas de Conocimiento											Unidades Académicas donde se imparte:																	
Ingeniería y Ciencias Físico Matemáticas	<input checked="" type="checkbox"/>	Ciencias Sociales Administrativas	<input type="checkbox"/>	Ciencias Médico-Biológicas	<input type="checkbox"/>	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	<input type="checkbox"/>	Científica, Humanística y Tecnológica Básica	<input type="checkbox"/>	Profesional	<input checked="" type="checkbox"/>	Global: <u>72</u> Hrs/18 semanas/Semestre Aula: <u>1</u> Hrs/Semana Total: <u>18</u> Hrs/Semestre Taller: <u>--</u> Hrs/Semana Total: <u>--</u> Hrs/Semestre Laboratorio: <u>3</u> Hrs/Semana Total: <u>54</u> Hrs/Semestre Otros ambientes de aprendizaje: <u>--</u> Hrs/Semana Total: <u>--</u> Hrs/Semestre																						
Tipo de Espacio																												
Aula	<input checked="" type="checkbox"/>	Taller	<input type="checkbox"/>	Laboratorio	<input checked="" type="checkbox"/>	Otros ambientes de Aprendizaje																						
Modalidad																												
Escolarizada	<input checked="" type="checkbox"/>	No Escolarizada	<input type="checkbox"/>	Mixta	<input type="checkbox"/>																							
Vigencia:	AGOSTO 2021																											
Proceso de Diseño y Autorización:						Día	Mes	Año	Organización																			
									Por Unidad de Aprendizaje:	<input checked="" type="checkbox"/>	Por Área:	<input type="checkbox"/>	Por Módulo:	<input type="checkbox"/>	Firma y Sello de Autorización:													
Elaborado por:	REP. ACAD. NMS IPN	Fecha de Elaboración:	05	03	2021	  INSTITUTO POLITÉCNICO NACIONAL Dirección de Educación Media Superior ING. JUAN SAN GERMAN TISCAREÑO Director de Educación Media Superior																						
Revisado por:	DEMS	Fecha de Revisión:	21	05	2021																							
Aprobado por:	CTCE	Fecha de Aprobación:	10	06	2021																							
Autorizado por:	CPA	Fecha de Autorización:	09	07	2021																							



Programa Académico: Técnico en Mecatrónica

Unidad de Aprendizaje: Lenguaje de Programación

FUNDAMENTACIÓN

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

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 Mecatrónica** y se imparte en la modalidad escolarizada, de manera **obligatoria** en la rama del conocimiento de Ingeniería y Ciencias Físico-Matemáticas.

Esta unidad de aprendizaje coadyuva a comprender la lógica de la programación como una dimensión 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 imaginación, la iniciativa y diversas habilidades cognitivas. Introduce al estudiante al campo conceptual, procedimental y actitudinal para definir problemas computables de mecatrónica, diseñar y desarrollar los respectivos sistemas computacionales básicos para solucionarlos, considerando tanto los principios y dimensiones del desarrollo humano sustentable como la perspectiva de género. La adquisición de estas destrezas y habilidades relacionadas con el pensamiento eficaz favorecerán en el estudiante el desarrollo de una visión crítica y holística, cuya puesta en práctica, en forma autónoma, en el futuro le coadyuvará a responder en forma eficiente y eficaz a los retos que se le presenten cuando se incorpore a estudios superiores o al campo laboral.

La unidad de aprendizaje Lenguaje de Programación contribuye al desarrollo del Talento 4.0 requerido por la Industria 4.0 y para la transformación del país, orientada al logro del desarrollo humano sustentable de México. Esto debido a que discente adquirirá, desarrollará y aplicará conceptos, técnicas y métodos que favorecen el desarrollo y fortalecimiento del pensamiento computacional, que es un instrumento estructural para organizar y analizar datos en forma lógica, representar datos mediante abstracciones, como los diagramas de flujo, diseñar y desarrollar soluciones a ser ejecutadas por un humano, una computadora o ambos; así como automatizar soluciones mediante el pensamiento algorítmico, optimizar e innovar soluciones y extrapolar este proceso de solución de problemas computables de mecatrónica a otras actividades y problemas.

Lenguaje de Programación es una unidad de aprendizaje enfocada al desarrollo de habilidades técnicas, cognitivas y socioemocionales inherentes al estudio, análisis y aplicación de tanto el paradigma de la programación estructurada como el de la programación orientada a objetos, así como los elementos de un lenguaje de programación, con la finalidad de desarrollar sistemas computacionales básicos, orientados a solucionar problemas computables de mecatrónica.

La unidad de aprendizaje Lenguaje de Programación estará fundamentada en el Modelo Educativo Institucional vigente y en la Educación para la industria 4.0, por esto, se emplearán metodologías didácticas activas como el Aprendizaje Basado en el Juego, 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.



Programa Académico: Técnico en Mecatrónica

Unidad de Aprendizaje: Lenguaje de Programación

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 discente 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 del estudiante. Este momento se enriquecerá con diversos tipos de evaluación, como la autoevaluación y la coevaluación, puesto que coadyuvarán a dar seguimiento al desarrollo de los saberes y habilidades en contexto. Cabe señalar que estas clases de evaluación serán reforzadas a través de la retroalimentación efectiva y oportuna.

En el tercer momento de la evaluación, con fines de acreditación, se diseñarán situaciones integradoras que permitan recuperar el nivel de logro y conducir al estudiante a la metacognición en la unidad de aprendizaje Lenguaje de Programación, esto mediante evidencias de conocimiento, producto y desempeño, como diseño de algoritmos, elaboración de diagramas de flujo, codificaciones, 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 Lenguaje de Programación, previo a su inicio. De esa forma, el programa de estudios de esta unidad de aprendizaje tiene una naturaleza normativa, puesto que establece los estándares para el desarrollo de conocimientos, habilidades prácticas del área de formación, habilidades socioemocionales, actitudes y valores.





Programa Académico: Técnico en Mecatrónica

Unidad de Aprendizaje: Lenguaje de Programación

DESCRIPCIÓN DE LA UNIDAD DE APRENDIZAJE

Unidad de Aprendizaje: Lenguaje de programación

Desarrolla sistemas computacionales básicos, aplicando la lógica de programación, con la finalidad de procesar datos de problemas computables de mecatrónica, en forma creativa, innovadora y centrada en las necesidades de los usuarios, bajo el enfoque del desarrollo sustentable.

Unidad 1. Introducción a la programación estructurada		
Unidad de competencia	Aprendizajes esperados	Contenidos
Utiliza la estructura secuencial de la programación estructurada y los elementos de un lenguaje de programación para solucionar problemas computables, aplicando razonamiento abstracto, creatividad y pensamiento lógico.	Formula algoritmos o diagramas de flujo sistematizados, normalizados y sin brechas en la lógica, con el propósito de diseñar la solución de problemas computables de ICFM, aplicando con creatividad las fases de solución de un problema computable y razonamiento abstracto.	<p>Conceptuales</p> <ol style="list-style-type: none"> 1) Fases de solución de un problema computable <ol style="list-style-type: none"> 1.1) Noción de problema 1.2) Problemas no computables y problemas computables 1.3) Fases de solución de un problema computable 2) Diseño de algoritmos <ol style="list-style-type: none"> 2.1) Definición, elementos y características de un algoritmo 2.2) Diseño de algoritmos de problemas computables 3) Diseño de diagramas de flujo <ol style="list-style-type: none"> 3.1) Definición, elementos y características de un diagrama de flujo 3.2) Simbología normalizada para elaborar diagramas de flujo 3.3) Diseño de diagramas de flujo de problemas computables 3.4) Tipos de datos 3.5) Prueba de escritorio <p>Procedimentales</p> <p>Formula algoritmos o diagramas de flujo sistematizados, normalizados y sin brechas en la lógica para diseñar la solución de problemas computables de ICFM (Ingeniería y Ciencias Fisicomatemáticas).</p> <p>Actitudinales</p> <p>Emplea creatividad y razonamiento abstracto en el diseño de la solución de problemas computables de ICFM.</p>





Programa Académico: Técnico en Mecatrónica

Unidad de Aprendizaje: Lenguaje de Programación

	<p>Codifica diagramas de flujo para desarrollar la solución de problemas computables de ICFM, empleando estructuras secuenciales, elementos de un lenguaje de programación y pensamiento lógico.</p>	<p>Conceptuales 1) Estructura secuencial 2) Codificación de diagramas de flujo 3) Problemática ambiental en mecatrónica</p> <p>Procedimentales Utiliza estructuras secuenciales y simbología normalizada para diseñar diagramas de flujo estructurados, con el propósito de solucionar problemas computables de ICFM.</p> <p>Traduce las especificaciones de los diagramas de flujo diseñados a instrucciones equivalentes de un lenguaje de programación para desarrollar la solución de problemas computables de ICFM.</p> <p>Actitudinales Aplica pensamiento lógico para codificar diagramas de flujo estructurados.</p>
--	--	--

Unidad 2. Estructuras de control de flujo		
Unidad de competencia	Aprendizajes esperados	Contenidos
<p>Realiza programas computacionales, aplicando estructuras de control de flujo y elementos de un lenguaje de programación, para solucionar problemas computables de electricidad, electrónica o tecnología de materiales, con creatividad y pensamiento analítico.</p>	<p>Utiliza las estructuras de bifurcación simple o múltiple de la programación estructurada y un lenguaje de programación para desarrollar la solución de problemas computables de electricidad, electrónica o tecnología de materiales, aplicando pensamiento analítico y creativo.</p>	<p>Conceptuales 1. Estructuras de bifurcación 1.1) Bifurcación simple 1.1.1) <i>Top down design</i> 1.1.2) Anidamiento 1.1.3) Operadores lógicos y relacionales 1.2) Bifurcación múltiple</p> <p>Procedimentales Aplica en forma diferenciada las estructuras de decisión simple o múltiple y un lenguaje de programación en el desarrollo de la solución de problemas computables de electricidad, electrónica o tecnología de materiales.</p> <p>Actitudinales Desarrolla la solución de problemas computables de electricidad, electrónica o tecnología de materiales con pensamiento analítico y creativo.</p>





Programa Académico: Técnico en Mecatrónica

Unidad de Aprendizaje: Lenguaje de Programación

	<p>Emplea las estructuras iterativas de la programación estructurada y un lenguaje de programación para desarrollar la solución de problemas computables de electricidad, electrónica o tecnología de materiales, aplicando creatividad y pensamiento analítico.</p>	<p>Conceptuales 1) Estructuras cíclicas 1.1) Ciclo for 1.2) Ciclo while 1.3) Ciclo do-while</p> <p>Procedimentales Aplica en forma diferenciada las estructuras iterativas y un lenguaje de programación en el desarrollo de la solución de problemas computables de electricidad, electrónica o tecnología de materiales.</p> <p>Actitudinales Desarrolla la solución de problemas computables de electricidad, electrónica o tecnología de materiales, con creatividad y pensamiento analítico.</p>
--	--	--

Unidad 3. Estructuras de datos		
Unidad de competencia	Aprendizajes esperados	Contenidos
<p>Desarrolla sistemas computacionales básicos, creativos, innovadores y centrados en las necesidades de los usuarios, empleando módulos computacionales, estructuras de datos, un lenguaje de programación y pensamiento computacional para resolver problemas computables de mecatrónica, de acuerdo con los preceptos de sustentabilidad.</p>	<p>Realiza módulos computacionales para solucionar problemas computables de mecatrónica, empleando funciones, arreglos y un lenguaje de programación, con pensamiento creativo e innovador.</p>	<p>Conceptuales 1) Funciones 1.1) Prototipado, definición e invocación de funciones 1.2) Funciones definidas por el usuario 1.3) Tipos de variables 1.3.1) Locales 1.3.2) Globales 1.4) Paso por valor 2) Arreglos 2.1) Arreglos unidimensionales 2.2) Arreglos bidimensionales 2.3) Hipermatrices 2.4) Arreglos dinámicos</p> <p>Procedimentales Utiliza funciones, arreglos y un lenguaje de programación para realizar módulos computacionales orientados a solucionar problemas computables de mecatrónica.</p> <p>Actitudinales</p>





Programa Académico: Técnico en Mecatrónica

Unidad de Aprendizaje: Lenguaje de Programación

	<p>Elabora sistemas computacionales básicos, centrados en las necesidades de los usuarios, para resolver problemas computables de mecatrónica, utilizando estructuras de datos, un lenguaje de programación y pensamiento computacional, bajo el enfoque del desarrollo sustentable.</p>	<p>Aplica pensamiento creativo e innovador en la realización de módulos computacionales para solucionar problemas computables de mecatrónica.</p> <p>Conceptuales</p> <ol style="list-style-type: none"> 1) Estructuras <ol style="list-style-type: none"> 1.1) Definición y declaración de una estructura 1.2) Creación de una estructura 1.3) Uso de estructuras 2) Implementación de bibliotecas o módulos del usuario del lenguaje de programación 3) Ecodiseño y modelo de desarrollo de <i>software</i> verde 4) Programación orientada a objetos <ol style="list-style-type: none"> 4.1) Definiciones básicas <ol style="list-style-type: none"> 4.1.1) Clase y objeto 4.1.2) Encapsulación 4.1.3) Herencia 4.1.4) Polimorfismo 4.2) Programación de una clase <p>Procedimentales</p> <p>Emplea estructuras de datos y un lenguaje de programación en la elaboración de sistemas computacionales básicos, enfocados a la resolución de problemas computables de mecatrónica.</p> <p>Actitudinales</p> <p>Efectúa sistemas computacionales básicos, centrados en las necesidades de los usuarios, aplicando pensamiento computacional, bajo el enfoque del desarrollo sustentable, para resolver problemas computables de mecatrónica.</p>
--	--	---





Programa Académico: Técnico en Mecatrónica

Unidad de Aprendizaje: Lenguaje de Programación

MATRIZ DE VINCULACIÓN

HABILIDADES BLANDAS Y SOCIOEMOCIONALES	Unidad de Competencia 1		Unidad de Competencia 2		Unidad de Competencia 3	
	AE1	AE2	AE1	AE2	AE1	AE2
1. Creatividad	X		X	X	X	
2. Innovación					X	
3. Pensamiento analítico			X	X		
4. Pensamiento lógico		X				
5. Pensamiento computacional						X
6. Razonamiento abstracto	X					
7. Sustentabilidad	X					X





Programa Académico: Técnico en Mecatrónica

Unidad de Aprendizaje: Lenguaje de Programación

COMPETENCIAS EDUCACIÓN 4.0						
Gestión, análisis e interpretación de la información a partir de las grandes cantidades de datos disponibles.					X	X
Descubrimiento de conocimiento, a partir de su interacción con pares y no pares, así como de colaboración profesional y organización que permita nuevas fases de desarrollo del talento.	X	X	X	X	X	X
Comunicación, socialización, colaboración, empatía, liderazgo y toma de decisiones, en el marco de un pensamiento estratégico.	X	X	X	X	X	X
Integra habilidades digitales para el aprovechamiento de la tecnología como factor de maximización en la producción del conocimiento y eficiencia de los procesos organizacionales.	X	X	X	X	X	X
Adaptación al cambio, por medio de procesos autogestivos de aprendizaje, movilización de saberes y pensamiento crítico.	X	X	X	X	X	X





Programa Académico: Técnico en Mecatrónica

Unidad de Aprendizaje: Lenguaje de Programación

PERFIL DOCENTE

Para impartir la unidad de aprendizaje **Lenguaje de Programación**, se considera necesario contar con **un docente titular y un docente adjunto o auxiliare** que se responsabilice junto con el titular, del trabajo dentro del laboratorio, con la finalidad de garantizar la atención puntual al proceso de aprendizaje, la seguridad e integridad física de los estudiantes; así como el cuidado y uso del laboratorio y los equipos.

El docente y el auxiliar que impartan la unidad de aprendizaje **Lenguaje de Programación** deberán contar con las competencias relativas al manejo de los saberes disciplinares y profesionales, así como disposición, autoridad y tolerancia en el manejo de grupos de aprendizaje. Por lo tanto, debe poseer las competencias que favorezcan el desarrollo del Talento 4.0.

Habilidades docentes en el desarrollo del Talento 4.0:

- Guía y facilita los aprendizajes de los estudiantes.
- Aprovecha los productos de la investigación.
- Se apoya de los recursos de la Educación 4.0.
- Basa su enseñanza en proyectos reales y necesidades sociales.
- Innova y es arquitecto del aprendizaje.
- Mantiene la enseñanza para los perfiles laborales del presente y del futuro.
- Guía a los discentes en las estrategias de búsqueda, selección, organización y uso de la información.
- Coadyuva al estudiante a descubrir nuevo conocimiento por sí mismo.
- Emplea estrategias de extrapolación de los aprendizajes para que puedan ser puestos en práctica por los estudiantes, en el futuro, de manera autónoma en su vida académica, personal, profesional, social o laboral.
- Se comunica en forma constante con los estudiantes.
- Utiliza herramientas tecnológicas para la comunicación y la colaboración, tanto síncrona como asíncrona.

En el campo de su especialización:

- Adquiere, desarrolla, aplica y transfiere habilidades digitales actualizadas.
- Desarrolla procesos de enseñanza-aprendizaje, utilizando métodos basados en administración de proyectos reales, aprovechando espacios educativos distintos a las aulas, para mejorar la calidad, pertinencia y relevancia de la enseñanza.
- Adquiere, desarrolla, aplica y transfiere competencias STEAM.
- Es creativo, innovador y arquitecto del aprendizaje.
- Favorece la realización de actividades y proyectos inter, multi y transdisciplinarios.
- Cuenta con las competencias específicas de su campo disciplinar.
- Participa en procesos de mejora continua en su práctica profesional.

En el campo pedagógico:





Programa Académico: Técnico en Mecatrónica

Unidad de Aprendizaje: Lenguaje de Programación

- Implementa metodologías activas para incentivar en los estudiantes el pensamiento eficaz y el aprendizaje profundo.
- Promueve el trabajo colaborativo y la construcción conjunta de conocimientos.
- Propicia que el estudiante se responsabilice de su proceso educativo.
- Fomenta procesos de enseñanza que le permitan interpretar y resolver las necesidades de aprendizaje de los estudiantes, tomando en cuenta sus capacidades, habilidades, vocación e intereses.
- Aprovecha los conocimientos de los nativos digitales.
- Promueve la aptitud o competencia poliédrica que no se reduce exclusivamente a una alta capacidad o competencia intelectual en áreas STEAM, sino que también incluye actitud digital, pasión por el cambio, aprendizaje autónomo y resiliencia.
- Cuenta y pone en práctica el soporte psicopedagógico pertinente.

En el campo de la investigación:

- Fortalece el trabajo académico a partir del aprovechamiento de los resultados y productos de los proyectos de investigación.
- Da resultados de un proceso de desarrollo que consiste en transformar aptitudes naturales (intelectuales, creativas y sociales) en competencias o talentos específicos, fruto de la práctica deliberada y de la existencia de una serie de catalizadores o facilitadores tanto en forma de programas formativos y oportunidades educativas como de actitudes intra e interpersonales (motivación, emprendimiento, búsqueda constante, capacidad para aprender por uno mismo, resiliencia, colaboración, generación de redes, trabajo en equipo, liderazgo, entre otras).

Perfil profesional del docente titular y del docente auxiliar

El papel del docente tendrá una intervención mediadora entre los contenidos disciplinares, las características del contexto y los instrumentos o herramientas que proveerán al estudiante para facilitar un aprendizaje activo, significativo, estratégico, autónomo, colaborativo, reflexivo, crítico y creativo. Por esto, debe:

- Contar con título en Ingeniería en Computación, Sistemas Computacionales, Informática, Mecatrónica, Mecánica, Eléctrica, Electromecánica, Electrónica, Robótica Industrial, Biónica, ser Licenciado en Informática o poseer un título en una licenciatura afín.
- Contar, de preferencia, con Maestría en alguna área de Ingeniería.
- Poseer experiencia mínima de tres años en el campo laboral público o privado, en el área de desarrollo de sistemas computacionales.
- Tener experiencia en manejo de grupo, empleo de las TIC, capacidad de análisis, síntesis e integración de información, empleo de técnicas de solución de conflictos, elaboración de instrumentos de evaluación e implantación de metodologías didácticas activas.
- Poseer actitud de cambio, pasión por el cambio, aprendizaje autónomo y resiliencia.
- Practicar actitudes positivas y valores, como responsabilidad, puntualidad, tolerancia, respeto, asertividad, liderazgo y trabajo en equipo.





Programa Académico: Técnico en Mecatrónica

Unidad de Aprendizaje: Lenguaje de Programación

ESTRUCTURA DIDÁCTICA

Unidad didáctica:	Unidad 1. Introducción a la programación estructurada	Nivel:	Cuarto
Propósito:	Desarrolla sistemas computacionales básicos, aplicando la lógica de programación, con la finalidad de procesar datos de problemas computables de mecatrónica, en forma creativa, innovadora y centrada en las necesidades de los usuarios, bajo el enfoque del desarrollo sustentable.		
Unidad de competencia N°1:	Utiliza la estructura secuencial de la programación estructurada y los elementos de un lenguaje de programación para solucionar problemas computables, aplicando razonamiento abstracto, creatividad y pensamiento lógico.		
Aprendizaje Esperado No 1:	Formula algoritmos o diagramas de flujo sistematizados, normalizados y sin brechas en la lógica, con el propósito de diseñar la solución de problemas computables de ICFM, aplicando con creatividad las fases de solución de un problema computable y razonamiento abstracto.	Tiempo estimado para obtener el Aprendizaje Esperado:	8 horas
Contenidos de Aprendizaje			
Conceptuales:	Procedimentales:	Actitudinales:	
<ol style="list-style-type: none"> 1) Fases de solución de un problema computable <ol style="list-style-type: none"> 1.1) Noción de problema 1.2) Problemas no computables y problemas computables 1.3) Fases de solución de un problema computable 2) Diseño de algoritmos <ol style="list-style-type: none"> 2.1) Definición, elementos y características de un algoritmo 2.2) Diseño de algoritmos de problemas computables 3) Diseño de diagramas de flujo <ol style="list-style-type: none"> 3.1) Definición, elementos y características de un diagrama de flujo 3.2) Simbología normalizada para elaborar diagramas de flujo 3.3) Diseño de diagramas de flujo de problemas computables 3.4) Tipos de datos 3.5) Prueba de escritorio 	<p>Formula algoritmos o diagramas de flujo sistematizados, normalizados y sin brechas en la lógica para diseñar la solución de problemas computables de ICFM (Ingeniería y Ciencias Fisicomatemáticas).</p> <p>Práctica 1. Punto medio entre dos puntos.</p>	<p>Emplea creatividad y razonamiento abstracto en el diseño de la solución de problemas computables de ICFM.</p>	





Programa Académico: Técnico en Mecatrónica

Unidad de Aprendizaje: Lenguaje de Programación

Estrategias Didácticas y Ambientes de Aprendizaje

Estrategia didáctica: Ejemplificación y Aprendizaje Basado en el Juego

El docente explica la importancia de desarrollar sistemas computacionales en los diferentes ámbitos de la vida de los seres humanos, exhorta a los estudiantes a buscar diversas definiciones tanto de problemas no computables como de problemas computables.

Los estudiantes indagan en diversas fuentes de información las nociones solicitadas, las socializan, construyen su concepto y listan las características de un problema computable y de un problema no computable. El docente retroalimenta las aportaciones de los estudiantes, ejemplifica la aplicación de las fases de solución de un problema computable e insta a los discentes a consultar en la Web, la definición, elementos y características de un algoritmo. Los estudiantes buscan, discriminan y sintetizan información para construir su conceptualización de algoritmo, sus elementos y características; dan a conocer al grupo, sus conceptualizaciones, los elementos y atributos de un algoritmo.

El docente retroalimenta las aportaciones de los discentes, ejemplifica el diseño de un algoritmo de un problema computable de ICFM, invita a los estudiantes a conformar equipos, plantea tres problemas computables de ICFM y les solicita que diseñen el algoritmo de cada uno de los tres problemas computables de ICFM planteados y los presenten en plenaria mediante una presentación electrónica.

Los estudiantes se organizan en equipos, aplican tanto su creatividad como razonamiento abstracto para diseñar los tres algoritmos, elaboran la presentación y, en plenaria, cada equipo presenta sus algoritmos diseñados, resuelven sus dudas y se retroalimentan. El docente retroalimenta las aportaciones de los estudiantes y exhorta a los estudiantes a elaborar un resumen sobre la definición, los elementos, las características y la simbología normalizada para elaborar diagramas de flujo. Los estudiantes consultan fuentes de información físicas o digitales, discriminan información sobre el tópico establecido, elaboran su resumen, se retroalimentan y solventan sus dudas.

El docente retroalimenta las aportaciones de los discentes y ejemplifica el diseño de un diagrama de flujo sistematizado, normalizado y sin brechas en la lógica, los tipos de datos empleados y la realización de la prueba de escritorio de un diagrama de flujo. Plantea un problema computable de ICFM e insta a los estudiantes a realizar el diagrama de flujo, determinar los tipos de datos a utilizar y efectuar la prueba de escritorio correspondiente.

Los estudiantes aplican tanto su creatividad como razonamiento abstracto para diseñar el diagrama de flujo sistematizado, normalizado y sin brechas en la lógica; especifican los tipos de datos utilizados y llevan a cabo la prueba de escritorio del diagrama de flujo diseñado. El docente selecciona al azar a tres estudiantes para que expliquen su diagrama, los tipos de datos empleados y la prueba de escritorio. Los estudiantes elegidos presentan su diagrama y prueba de escritorio con los casos y valores correspondientes al tipo de datos empleados, el grupo retroalimenta y resuelve sus dudas.

El docente retroalimenta las aportaciones de los estudiantes y exhorta a los discentes a participar en el juego (físico o digital) propuesto por él para repasar y profundizar sus aprendizajes. Los estudiantes juegan y comentan en plenaria la utilidad del juego en la consolidación de sus aprendizajes relativos a la aplicación de las fases de solución de un problema computable, el diseño de algoritmos y diseño de diagramas de flujo normalizados, así como la realización de la prueba de escritorio.

El docente resume los comentarios de los estudiantes y les solicita que realicen por equipo, la evidencia de aprendizaje formativa.





Programa Académico: Técnico en Mecatrónica

Unidad de Aprendizaje: Lenguaje de Programación

Herramientas Tecnológicas y Recursos Didácticos	Evidencia de Aprendizaje para la Evaluación Formativa	Criterios e Instrumento de Evaluación
<ul style="list-style-type: none"> • Videos sobre: <ul style="list-style-type: none"> ○ Definición y características de un algoritmo. ○ Elaboración de diagramas de flujo. ○ Realización de la prueba de escritorio de un algoritmo. • Infografías sobre la simbología para elaborar diagramas de flujo. • Presentaciones multimedia sobre ejemplos de algoritmos y diagramas de flujo. • Sitios web sobre problemas computables. • Libros impresos o digitales sobre: <ul style="list-style-type: none"> ○ Diseño de algoritmos. ○ Simbología para elaborar diagramas de flujo. ○ Diseño de diagramas de flujo. • Formulario. • Apuntes digitales o impresos sobre problemas computables, algoritmos y diagramas de flujo. • <i>Software</i> gratuito para elaborar diagramas de flujo. 	<p>Reporte escrito “Volumen, área y capacidad de un cuerpo tridimensional”.</p>	<p>El reporte incluye:</p> <ul style="list-style-type: none"> ➤ Carátula ➤ Desarrollo: <ul style="list-style-type: none"> ○ Describe el problema computable a resolver. ○ Reseña las variables de entrada, proceso y salida a emplear. ○ Explica los tipos de datos de las variables. ○ Incluye el diagrama de flujo sistematizado, normalizado y sin brechas en la lógica para calcular el volumen, área y capacidad de un cuerpo tridimensional. ○ El diseño del diagrama de flujo evidencia creatividad y la aplicación de razonamiento abstracto. ○ Incorpora prueba de escritorio con los casos y valores implicados en el problema computable. ➤ Conclusiones: <ul style="list-style-type: none"> ○ Argumenta, con base en una o dos referencias, la importancia de los diagramas de flujo como herramienta para diseñar la solución de problemas computables. ○ Describe los obstáculos enfrentados durante el diseño del diagrama de flujo. ➤ Referencias: <ul style="list-style-type: none"> ○ Incorpora una o dos referencias en estilo APA. <p>Instrumento de evaluación Lista de cotejo</p>





Programa Académico: Técnico en Mecatrónica

Unidad de Aprendizaje: Lenguaje de Programación

Unidad didáctica:	Unidad 1. Introducción a la programación estructurada	Nivel:	Cuarto
Propósito:	Desarrolla sistemas computacionales básicos, aplicando la lógica de programación, con la finalidad de procesar datos de problemas computables de mecatrónica, en forma creativa, innovadora y centrada en las necesidades de los usuarios, bajo el enfoque del desarrollo sustentable.		
Unidad de competencia N°1:	Utiliza la estructura secuencial de la programación estructurada y los elementos de un lenguaje de programación para solucionar problemas computables, aplicando razonamiento abstracto, creatividad y pensamiento lógico.		
Aprendizaje Esperado No 2:	Codifica diagramas de flujo para desarrollar la solución de problemas computables de ICFM, empleando estructuras secuenciales, elementos de un lenguaje de programación y pensamiento lógico.	Tiempo estimado para obtener el Aprendizaje Esperado:	8 horas

Contenidos de Aprendizaje

Conceptuales:	Procedimentales:	Actitudinales:
1) Estructura secuencial 2) Codificación de diagramas de flujo 3) Problemática ambiental en mecatrónica	Utiliza estructuras secuenciales y simbología normalizada para diseñar diagramas de flujo estructurados, con el propósito de solucionar problemas computables de ICFM. Traduce las especificaciones de los diagramas de flujo diseñados a instrucciones equivalentes de un lenguaje de programación para desarrollar la solución de problemas computables de ICFM. Práctica 2. Conversión de grados Celsius a grados Fahrenheit.	Aplica pensamiento lógico para codificar diagramas de flujo estructurados.

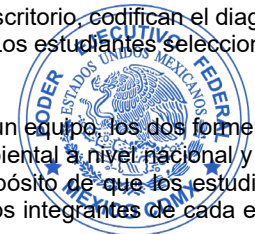
Estrategias Didácticas y Ambientes de Aprendizaje

Estrategia didáctica: Aprendizaje Basado en la Colaboración y el Diálogo (ABCD)

El docente describe tanto las características como los beneficios de la programación estructurada, las propiedades y representaciones de asignación, entrada y salida de datos de la estructura secuencial, así como los elementos del lenguaje de programación empleados para codificar estructuras secuenciales. Ejemplifica la traducción de cada símbolo de un diagrama de flujo estructurado a las instrucciones correspondientes del lenguaje de programación y la respectiva validación del código generado. Propone un problema computable de ICFM, solicita a los discentes que lleven a cabo el diagrama de flujo estructurado, la codificación y la validación tanto del diagrama de flujo como del código.

Los estudiantes dialogan, colaboran y comparten conocimientos, ideas y experiencias para diseñar el diagrama de flujo estructurado solicitado, realizan la prueba de escritorio, codifican el diagrama de flujo y validan el código. El docente selecciona al azar a tres estudiantes para que expongan su diagrama de flujo, la prueba de escritorio, el código y su validación. Los estudiantes seleccionados presentan y explican su diagrama de flujo, los tipos de datos empleados y la prueba de escritorio con los casos y valores pertinentes.

El docente retroalimenta y resuelve sus dudas. Numera a los estudiantes del uno al ocho, solicita que los estudiantes que tienen asignado el número uno, conformen un equipo, los dos formen otro equipo, los tres integren otro equipo y así sucesivamente; esto con la finalidad de obtener ocho equipos heterogéneos. Suministra documentos sobre legislación ambiental a nivel nacional y local, inventarios de emisiones, legislación de residuos tecnológicos, contaminantes generados por desarrollar un programa computacional y *software* verde, con el propósito de que los estudiantes reconozcan la problemática ambiental en mecatrónica. Distribuye los temas entre los equipos, precisa los puntos a desarrollar en cada tópico, establece que todos los integrantes de cada equipo deben saber en forma completa el tema asignado y presentar su tema al grupo.





Programa Académico: Técnico en Mecatrónica

Unidad de Aprendizaje: Lenguaje de Programación

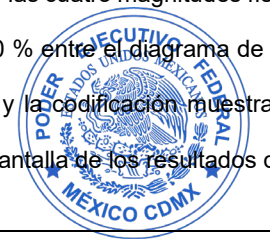
Los estudiantes, en equipo, se organizan, distribuyen tareas, analizan y sintetizan la información de los documentos proporcionados; controlan sus tiempos, colaboran, dialogan y comparten tanto conocimientos como experiencias para desarrollar el tópico, se formulan preguntas entre ellos para verificar su dominio sobre el tema, se retroalimentan, solventan sus dudas, homogeneizan sus conocimientos, elaboran una presentación electrónica y exponen su tópico al grupo.

El docente y el grupo retroalimentan las presentaciones. El docente solicita la integración de ocho equipos de expertos, considerando que el nuevo equipo uno debe contar con un integrante del equipo uno anterior, uno del equipo dos, uno del equipo tres, uno del equipo cuatro y uno del equipo cinco, el resto de los equipos se conformará de la misma forma, con el objetivo de que los nuevos equipos incluyan un experto de cada uno de los temas desarrollados con antelación. Insta a los estudiantes especificar los contaminantes generados por un sistema mecatrónico y a codificar el diagrama de flujo estructurado que diseñarán, utilizando estructuras secuenciales, para calcular las emisiones contaminantes por realizar un programa computacional.

Los estudiantes se integran a su nuevo equipo, discuten, en función de sus aprendizajes adquiridos, para identificar los contaminantes producidos por un sistema mecatrónico y un programa computacional, aplican pensamiento lógico para diseñar el diagrama de flujo utilizando simbología normalizada y estructuras secuenciales, realizar la prueba de escritorio, codificar el diagrama de flujo y validar el código. Eligen un orador para que presente en plenaria su código y responda las preguntas que formule el grupo. El grupo realiza comentarios, recomendaciones a cada orador y al equipo, se retroalimentan y manifiestan sus conclusiones.

El docente retroalimenta las conclusiones y solicita a los estudiantes realicen la evidencia de aprendizaje formativa.

Herramientas Tecnológicas y Recursos Didácticos	Evidencia de Aprendizaje para la Evaluación Formativa	Criterios e Instrumento de Evaluación
<ul style="list-style-type: none"> • Juego físico o digital, propuesto por el docente para recuperar los conocimientos previos. • Videos sobre: <ul style="list-style-type: none"> ○ Definición y características de la programación estructurada. ○ Ejemplos de codificación de estructuras secuenciales. ○ Ejemplos de códigos de soluciones a problemas de ICFM. ○ <i>Software</i> verde. • Lenguaje de programación o IDE (Integrated Development Environment). • Tutorial sobre instalación del lenguaje de programación empleado: DEV C++, Cxx, entre otros. • Presentaciones multimedia sobre: <ul style="list-style-type: none"> ○ Elementos generales del lenguaje de programación empleado. ○ Programación estructurada y estructura secuencial. 	<p>Presentación “Cálculo de magnitudes físicas”.</p>	<p>La presentación incluye:</p> <ul style="list-style-type: none"> ➤ Portada: ➤ Desarrollo: <ul style="list-style-type: none"> ○ Reseña las variables de entrada, proceso y salida a emplear. ○ Puntualiza los tipos de las variables. ○ Contiene el diagrama de flujo estructurado para calcular las cuatro magnitudes físicas indicadas. ○ Incluye la prueba de escritorio con los casos y valores pertinentes. ○ Integra la codificación para calcular las cuatro magnitudes físicas solicitadas. ○ Existe concordancia de 90 % - 100 % entre el diagrama de flujo estructurado y el código. ○ El diagrama de flujo estructurado y la codificación muestran la aplicación de pensamiento lógico. ○ Incluye fotografías o capturas de pantalla de los resultados de la ejecución del código. ➤ Conclusiones:





Programa Académico: Técnico en Mecatrónica

Unidad de Aprendizaje: Lenguaje de Programación

<ul style="list-style-type: none"> ○ Léxico, sintaxis y semántica para codificar estructuras secuenciales en el lenguaje de programación empleado. ○ Ejemplos de codificación de estructuras secuenciales. ● Manuales de programación en el lenguaje de programación empleado (C++, Dev C++, Cxxdroid, entre otros) ● Libros impresos o digitales sobre: <ul style="list-style-type: none"> ○ Legislación ambiental nacional y local. ○ Legislación de residuos tecnológicos. ○ Inventario de emisiones. ● Sitios WEB sobre: <ul style="list-style-type: none"> ○ Contaminantes producidos por sistemas mecatrónicos, equipos eléctricos y electrónicos. ○ <i>Software</i> verde. ● Informes sobre contaminantes generados por <i>software</i>, aplicaciones informáticas y equipos utilizados en mecatrónica. 		<ul style="list-style-type: none"> ○ Explica, con base en una o dos fuentes de información, la importancia de las estructuras secuenciales en el desarrollo de la solución de problemas computables de ICFM. ○ Comenta las dificultades sorteadas durante el diseño del diagrama de flujo y su codificación. ➤ Referencias: <ul style="list-style-type: none"> ○ Incorpora una o dos referencias en estilo APA. ➤ Incorpora videos o audios (con duración de dos a cuatro minutos), imágenes estáticas o gifs alusivos al tema, texto simple y tipografía legible. ➤ Tiene una estructura ordenada, clara y legible. ➤ No está saturada de texto. ➤ No incurre en plagio. ➤ Está elaborada con un <i>software</i> y sitio web de herramientas de diseño gráfico <p>Instrumento de evaluación Lista de cotejo</p>
---	--	---





Programa Académico: Técnico en Mecatrónica

Unidad de Aprendizaje: Lenguaje de Programación

Unidad didáctica:	Unidad 2. Estructuras de control de flujo	Nivel:	Cuarto
Propósito:	Desarrolla sistemas computacionales básicos, aplicando la lógica de programación, con la finalidad de procesar datos de problemas computables de mecatrónica, en forma creativa, innovadora y centrada en las necesidades de los usuarios, bajo el enfoque del desarrollo sustentable.		
Unidad de competencia N°2:	Realiza programas computacionales, aplicando estructuras de control de flujo y elementos de un lenguaje de programación, para solucionar problemas computables de electricidad, electrónica o tecnología de materiales, con creatividad y pensamiento analítico.		
Aprendizaje Esperado No 1:	Utiliza las estructuras de bifurcación simple o múltiple de la programación estructurada y un lenguaje de programación para desarrollar la solución de problemas computables de electricidad, electrónica o tecnología de materiales, aplicando pensamiento analítico y creativo.	Tiempo estimado para obtener el Aprendizaje Esperado:	8 horas

Contenidos de Aprendizaje

Conceptuales:	Procedimentales:	Actitudinales:
1. Estructuras de bifurcación 1.1) Bifurcación simple 1.1.1) <i>Top down design</i> 1.1.2) Anidamiento 1.1.3) Operadores lógicos y relacionales 1.2) Bifurcación múltiple	Aplica en forma diferenciada las estructuras de decisión simple o múltiple y un lenguaje de programación en el desarrollo de la solución de problemas computables de electricidad, electrónica o tecnología de materiales. Práctica 3. Usos de aleaciones de acero.	Desarrolla la solución de problemas computables de electricidad, electrónica o tecnología de materiales, con pensamiento analítico y creativo.

Estrategias Didácticas y Ambientes de Aprendizaje

Estrategia didáctica: STEAM

El docente insta a los estudiantes a efectuar el programa para dividir dos números dados vía teclado por el usuario. Los estudiantes, en forma individual, diseñan el diagrama de flujo estructurado, realizan la prueba de escritorio y codifican el diagrama de flujo estructurado en el lenguaje de programación empleado. El docente solicita la validación del programa computacional con dividiendo diferente de cero y divisor igual a cero para que resulte una indeterminación y, por lo tanto, el programa computacional presente un error en la ejecución. Describe las razones del error y la forma de resolverlo mediante estructuras de bifurcación de la programación estructurada,

El docente ejemplifica, utilizando diagramas de flujo, el empleo de las estructuras de bifurcación simple *if-then* e *if-then-else* para controlar el flujo de las acciones a llevar a cabo en la resolución del ejercicio anterior, realiza las pruebas de escritorio y codificaciones correspondientes, aplicando *top down design*.

Los estudiantes codifican los diagramas de flujo en el lenguaje de programación empleado y validan, con los casos y valores de las pruebas de escritorio, los programas computacionales generados. El docente solicita a los estudiantes realizar el programa computacional para calcular la resistencia total en un circuito serie, compuesto por cinco resistores. Indica a los estudiantes que en el diagrama de flujo y en el programa computacional se deben tomar decisiones sobre el curso a seguir ante valores de entrada válidos y no válidos.

Los estudiantes se organizan en equipos para diseñar el diagrama de flujo, utilizando estructuras de bifurcación simple, anidamiento u operadores lógicos y relacionales, realizan la prueba de escritorio, efectúan el programa computacional y lo validan con tanto los casos como los valores de la prueba de escritorio. El docente elige un equipo al azar para exponer su diagrama de flujo y el programa computacional. El equipo seleccionado presenta su trabajo, responde a las preguntas formuladas por sus pares y emiten las conclusiones.

El docente resume y retroalimenta las conclusiones de los discentes. Ejemplifica el empleo de las estructuras de bifurcación múltiple *case-of* y *case-of-otherwise* por medio del diagrama de flujo estructurado y el programa computacional que despliega un menú, cuyas opciones corresponden a los sistemas de medición angular: 1) Sistema sexagesimal o inglés, 2) Sistema centesimal o francés,





Programa Académico: Técnico en Mecatrónica

Unidad de Aprendizaje: Lenguaje de Programación

3) Sistema radial o internacional y 4) Salir del menú; en el diagrama de flujo y en el programa computacional están incluidos los elementos para leer los datos de entrada, realizar los cálculos correspondientes, controlar los valores de entrada no válidos e imprimir el resultado obtenido, en función del tipo de conversión elegida, o enviar un mensaje al usuario si ingresa una opción válida o datos erróneos. Solicita a los discentes efectuar un tríptico para utilizar el programa computacional que desarrollen para calcular cuatro propiedades mecánicas de un material; el programa debe presentar el título, el menú con cinco alternativas: 1) Fuerza de resistencia de fricción estática, 2) Deformación unitaria, 3) Esfuerzo normal, 4) Módulo de elasticidad y 5) Salir; asimismo, debe imprimir los resultados o los mensajes correspondientes si el usuario introduce valores incorrectos o efectúa una operación no válida.

Los estudiantes se organizan en equipos, aplican en forma diferenciada las estructuras de bifurcación simple o múltiple, operadores lógicos y relacionales o anidamiento, *top down design* y pensamiento tanto analítico como creativo para diseñar el diagrama de flujo, realizan la prueba de escritorio con tanto los casos como los valores implicados en el problema computable planteado, efectúan el programa computacional, validan su funcionalidad y elaboran el tríptico; en éste describen los pasos a llevar a cabo para usar su programa computacional e incluyen imágenes que faciliten la comprensión del funcionamiento del programa. Efectúan una presentación electrónica para dar a conocer su tríptico al grupo, exponen y, al final de su presentación, responden a las preguntas que les formule el grupo. El grupo retroalimenta a cada equipo y expresa sus conclusiones.

El docente retroalimenta las participaciones de los estudiantes y les solicita que entreguen por equipo su tríptico complementado; asimismo, que elaboren la evidencia de aprendizaje formativa.

Herramientas Tecnológicas y Recursos Didácticos	Evidencia de Aprendizaje para la Evaluación Formativa	Criterios e Instrumento de Evaluación
<ul style="list-style-type: none"> • Videos sobre: <ul style="list-style-type: none"> ○ Empleo de las estructuras if-then.e if-then-else. ○ Utilización de las estructuras case-of y case-of-otherwise. ○ <i>Top down design</i>. ○ Operadores relacionales y lógicos. ○ Ejemplos de codificación de las estructuras de bifurcación simple y múltiple. ○ Ejemplos de codificación de operadores lógicos y relacionales. • Lenguaje de programación o IDE (Integrated Development Environment). • Presentaciones multimedia sobre: <ul style="list-style-type: none"> ○ Anidamiento ○ Programación estructurada, estructuras selectivas y estructura secuencial. ○ Léxico, sintaxis y semántica para codificar estructuras de bifurcación simple y bifurcación múltiple en el lenguaje de programación empleado. ○ Ejemplos de codificación de estructuras de bifurcación simple y bifurcación múltiple. • <i>Software</i> gratuito para elaborar diagramas de flujo. • Computadora o celular con conectividad a Internet. 	<p>Videotutorial “Cálculo de reactancias e impedancia”.</p>	<p>El videotutorial incluye:</p> <ul style="list-style-type: none"> ➤ Datos de identificación: ➤ Uso del lenguaje: sin errores gramaticales, dicción u ortográficos. ➤ Presencia del estudiante durante el video. ➤ Introducción: <ul style="list-style-type: none"> ○ Describe en forma breve el procedimiento para calcular las reactancias e impedancia. ○ Explica el propósito del videotutorial. ➤ Contenido: <ul style="list-style-type: none"> ○ Incluye el diagrama de flujo con los símbolos normalizados para presentar el menú: 1) Reactancia capacitiva, 2) Reactancia inductiva, 3) Impedancia y 4) Salir, solicitar la opción al usuario y, en función de la opción elegida, leer las variables, realizar los cálculos e imprimir los resultados, salir del programa o enviar una notificación al usuario si ingresa datos erróneos. ○ Integra la prueba de escritorio con casos y valores válidos e incorrectos para corroborar el control del flujo sobre la impresión de resultados o envío de avisos al usuario si efectúa una operación no válida o ingresa datos equivocados. ○ Incluye el programa computacional documentado, con sangrados y sin errores de léxico, sintaxis o de semántica e imprime los resultados relacionados con la opción elegida o mensajes para indicarle al usuario si ingresa valores no válidos



Programa Académico: Técnico en Mecatrónica

Unidad de Aprendizaje: Lenguaje de Programación

<ul style="list-style-type: none"> • Manuales de programación en el lenguaje de programación empleado (C++, Dev C++, Cxxdroid, entre otros). • Tutoriales sobre aplicaciones del lenguaje empleado. • Cursos en línea de programación en el lenguaje de programación empleado. 		<ul style="list-style-type: none"> ○ El diagrama de flujo y el programa computacional coinciden entre 90 y 100 %. ○ El diagrama de flujo diseñado y el programa computacional elaborado reflejan el uso de pensamiento analítico y creativo. ○ Incorpora fotografías o capturas de pantalla de los resultados de la ejecución del programa computacional. ➤ Conclusiones: <ul style="list-style-type: none"> ○ Describe la importancia, con base en dos o tres referencias, de utilizar estructuras de bifurcación, <i>top down</i> design, un lenguaje de programación pensamiento analítico y creativo en el desarrollo de la solución de problemas computables de electricidad, electrónica o tecnología de materiales. ○ Describe los obstáculos tenidos en el proceso de solución. ➤ Referencias <ul style="list-style-type: none"> ○ Incorpora dos o tres referencias en estilo APA. <p>Instrumento de evaluación Lista de cotejo</p>
---	--	---





Programa Académico: Técnico en Mecatrónica

Unidad de Aprendizaje: Lenguaje de Programación

Unidad didáctica:	Unidad 2. Estructuras de control de flujo	Nivel:	Cuarto
Propósito:	Desarrolla sistemas computacionales básicos, aplicando la lógica de programación, con la finalidad de procesar datos de problemas computables de mecatrónica, en forma creativa, innovadora y centrada en las necesidades de los usuarios, bajo el enfoque del desarrollo sustentable.		
Unidad de competencia N°2:	Realiza programas computacionales, aplicando estructuras de control de flujo y elementos de un lenguaje de programación, para solucionar problemas computables de electricidad, electrónica o tecnología de materiales, con creatividad y pensamiento analítico.		
Aprendizaje Esperado No 2:	Emplea las estructuras iterativas de la programación estructurada y un lenguaje de programación para desarrollar la solución de problemas computables de electricidad, electrónica o tecnología de materiales, aplicando creatividad y pensamiento analítico.	Tiempo estimado para obtener el Aprendizaje Esperado:	16 horas

Contenidos de Aprendizaje

Conceptuales:	Procedimentales:	Actitudinales:
1) Estructuras cíclicas 1.1) Ciclo for 1.2) Ciclo while 1.3) Ciclo do-while	Aplica en forma diferenciada las estructuras iterativas y un lenguaje de programación en el desarrollo de la solución de problemas computables de electricidad, electrónica o tecnología de materiales. Práctica 4. Resistencia total en un circuito paralelo.	Desarrolla la solución de problemas computables de electricidad, electrónica o tecnología de materiales, con creatividad y pensamiento analítico.

Estrategias Didácticas y Ambientes de Aprendizaje

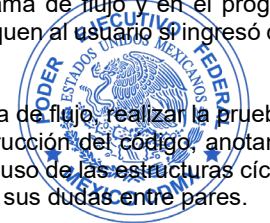
Estrategias didácticas: Juego de roles

El docente pide a los estudiantes que se organicen en equipos de cinco personas para indagar, estudiar y preparar lo relacionado al diagrama de flujo, sintaxis y ejemplos de las estructuras cíclicas (iterativas o de repetición) *for*, *while* y *do-while*, asimismo, realicen la debida presentación electrónica, utilizando un *software* y sitio web de herramientas de diseño gráfico.

Los estudiantes se organizan en equipos de cinco integrantes, investigan, en fuentes impresas o digitales, los tópicos señalados por el docente, analizan y sintetizan la información, y efectúan la presentación. El docente elige en forma aleatoria un equipo para presentar el tema ante el grupo. El equipo seleccionado expone en plenaria el tema, en tanto, el resto de los equipos complementan la información, hacen comentarios o recomendaciones y retroalimentan. El docente sintetiza las aportaciones de los estudiantes, lista tanto las ventajas como las diferencias y los casos de aplicación de las estructuras cíclicas, solventa dudas e invita a los estudiantes a integrarse en equipos de ocho estudiantes e indica que cada integrante, de cada equipo, debe elegir un rol dentro de los siguientes aspectos de *software*: bifurcación simple, bifurcación múltiple, estructura cíclica, estructura secuencial u operadores, así como uno de los siguientes elementos de hardware: memoria RAM, ULA (Unidad Lógico- Aritmética), pantalla, entre otros; puntualiza que él jugará el rol de usuario y que todos los discentes deben desempeñar su respectivo rol para diseñar el diagrama de flujo, efectuar la prueba de escritorio y realizar el programa computacional para calcular la capacitancia total en un circuito paralelo compuesto por cinco capacitores. En el diagrama de flujo y en el programa computacional se deben emplear estructuras iterativas e integrar acciones para leer, calcular y proporcionar el resultado correspondiente o enviar mensajes que le notifiquen al usuario si ingresó datos incorrectos.

Los estudiantes se organizan en equipos, eligen un rol de hardware o *software*, se organizan entre ellos, juegan su rol de aspectos de *software* para diseñar el diagrama de flujo, realizar la prueba de escritorio y elaborar el programa computacional; una vez realizado el programa, los integrantes con rol de hardware, desempeñan su papel para ejecutar cada instrucción del código, anotan sus resultados parciales y finales en su pizarra. En ambas etapas los estudiantes emplean sus creatividad y pensamiento analítico para identificar y comprobar los casos de uso de las estructuras cíclicas, llegar a acuerdos para presentar su programa computacional desarrollado y resolver conflictos. En plenaria, presentan sus conclusiones, se retroalimentan y resuelven sus dudas entre pares.

El docente sintetiza las aportaciones de los estudiantes, retroalimenta y les solicita que elaboren la evidencia de aprendizaje formativa.





Programa Académico: Técnico en Mecatrónica

Unidad de Aprendizaje: Lenguaje de Programación

Herramientas Tecnológicas y Recursos Didácticos	Evidencia de Aprendizaje para la Evaluación Formativa	Criterios e Instrumento de Evaluación
<ul style="list-style-type: none"> • Videos acerca de: <ul style="list-style-type: none"> ○ Empleo de las estructuras repetitivas: for, while, do-while. ○ Ejemplos de codificación de las estructuras repetitivas for, while, do while. ○ Ejemplos de reducción de expresiones. • Lenguaje de programación o IDE (Integrated Development Environment). • Presentaciones multimedia sobre: <ul style="list-style-type: none"> ○ Estructuras repetitivas for, while, do-while. ○ Léxico, sintaxis y semántica para codificar estructuras repetitivas for, while, do-while, en el lenguaje de programación empleado. • <i>Software</i> gratuito para elaborar diagramas de flujo. • Computadora o celular con conectividad a Internet. • Manuales de programación en el lenguaje de programación empleado (C++, Dev C++, Cxxdroid, entre otros). • Tutoriales sobre aplicaciones del lenguaje empleado. • Cuadros de doble entrada de las estructuras repetitivas. • Cursos en línea de programación en el lenguaje de programación utilizado. 	<p>Manual técnico “Cálculo de la media aritmética de la dureza Vickers en un material”</p>	<p>El manual técnico contiene:</p> <ul style="list-style-type: none"> ➤ Portada: ➤ Datos de identificación: ➤ Índice <ul style="list-style-type: none"> ○ Introducción: Presenta el propósito, síntesis del contenido y las conclusiones más importantes. ➤ Contenido: <ul style="list-style-type: none"> ○ Describe el procedimiento para calcular la media aritmética de la dureza Vickers de un material medida en n ensayos. ○ Integra el diagrama de flujo, con una estructura cíclica para leer n medidas de la dureza Vickers de un material, los símbolos para validar que las medidas son positivas, calcular la media aritmética e imprimir los resultados o un aviso si ingresan datos incorrectos. ○ Contiene la prueba de escritorio con casos y valores tanto válidos como incorrectos para comprobar el control del flujo. ○ Incorpora el programa computacional con comentarios y sangrados, no tiene errores de léxico, sintaxis ni de semántica y coincide 90 % - 100 % con el diagrama de flujo. ○ El diseño del diagrama de flujo y el programa computacional reflejan el despliegue de creatividad y pensamiento analítico. ➤ Instrucciones para el usuario: <ul style="list-style-type: none"> ○ Detalla el funcionamiento del programa. ○ Incluye fotografías o capturas de pantalla de los elementos mencionados en las instrucciones. ➤ Conclusiones: <ul style="list-style-type: none"> ○ Describe, con base en dos o tres fuentes de información, la trascendencia de emplear estructuras cíclicas, un lenguaje de programación, pensamiento analítico y creatividad para desarrollar la solución de problemas computables de electricidad, electrónica o tecnología de materiales. ○ Pormenoriza los obstáculos encarados durante la realización de la actividad. ➤ Referencias <ul style="list-style-type: none"> ○ Incorpora dos o tres referencias en estilo APA. ➤ La redacción es sencilla, clara y ágil. <p>Instrumento de evaluación Lista de cotejo</p>





Programa Académico: Técnico en Mecatrónica

Unidad de Aprendizaje: Lenguaje de Programación

Unidad didáctica:	Unidad 3. Estructuras de datos	Nivel:	Cuarto
Propósito:	Desarrolla sistemas computacionales básicos, aplicando la lógica de programación, con la finalidad de procesar datos de problemas computables de mecatrónica, en forma creativa, innovadora y centrada en las necesidades de los usuarios, bajo el enfoque del desarrollo sustentable.		
Unidad de competencia N°3:	Desarrolla sistemas computacionales básicos, creativos, innovadores y centrados en las necesidades de los usuarios, empleando módulos computacionales, estructuras de datos, un lenguaje de programación y pensamiento computacional para resolver problemas computables de mecatrónica, de acuerdo con los preceptos de sustentabilidad.		
Aprendizaje Esperado No 1:	Realiza módulos computacionales para solucionar problemas computables de mecatrónica, empleando funciones, arreglos y un lenguaje de programación, con pensamiento creativo e innovador.	Tiempo estimado para obtener el Aprendizaje Esperado:	16 horas

Contenidos de Aprendizaje

Conceptuales:	Procedimentales:	Actitudinales:
1) Funciones 1.1) Prototipado, definición e invocación de funciones 1.2) Funciones definidas por el usuario 1.3) Tipos de variables 1.3.1) Locales 1.3.2) Globales 1.4) Paso por valor 2) Arreglos 2.1) Arreglos unidimensionales 2.2) Arreglos bidimensionales 2.3) Hipermatrices 2.4) Arreglos dinámicos	Utiliza funciones, arreglos y un lenguaje de programación para realizar módulos computacionales orientados a solucionar problemas computables de mecatrónica. Práctica 5. Magnitud del torque de un centro de lavado.	Aplica pensamiento creativo e innovador en la realización de módulos computacionales para solucionar problemas computables de mecatrónica.

Estrategias Didácticas y Ambientes de Aprendizaje

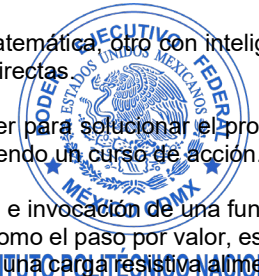
Estrategia didáctica: Aprendizaje Basado en Problemas (ABP)

El docente sugiere a los estudiantes responder un cuestionario automatizado para que identifiquen sus inteligencias múltiples y guardar sus resultados en formato electrónico. Los estudiantes realizan el cuestionario propuesto para reconocer sus inteligencias múltiples.

El docente conforma equipos heterogéneos, procurando que en cada equipo esté presente un estudiante con inteligencia intrapersonal, otro con inteligencia lógico-matemática, otro con inteligencia interpersonal, otro con inteligencia lingüística, así sucesivamente. Plantea el problema computable relativo al cálculo de las incertidumbres resultantes de mediciones directas.

En cada equipo, cada uno de sus integrantes aporta su punto de vista, plantea qué sabe y qué necesita conocer del problema computable, define qué requiere saber para solucionar el problema computable y considera en forma reflexiva los puntos de vista de los otros para solucionar el problema computable. Cada equipo de trabajo propone su solución, definiendo un curso de acción.

El docente especifica qué es una función y su utilidad mediante la descripción del proceso de diseño y desarrollo de un videojuego, ejemplifica el prototipado, definición e invocación de una función a través de una función del lenguaje de programación empleado e ilustra la creación de funciones definidas por el usuario, el empleo de variables locales y globales, así como el paso por valor, esto por medio de un diagrama de flujo y un programa computacional que presenten el título “Electrotecnia”, el menú con las siguientes opciones: 1) Cálculo de corriente media en una carga resistiva alimentada





Programa Académico: Técnico en Mecatrónica

Unidad de Aprendizaje: Lenguaje de Programación

por C.A., 2) Cálculo de la potencia consumida RMS en la carga, 3) Cálculo de voltaje medio a voltaje RMS y 4) Salir; en el diagrama de flujo y en el programa computacional se deben imprimir los resultados correspondientes o enviar mensajes para avisarle al usuario si introdujo valores no válidos. Cada opción del menú corresponde a una función definida por el usuario. Solicita a los discentes que se organicen en sus equipos para que precisen las funciones necesarias para resolver el problema computable planteado.

Los estudiantes, en equipos, practican la escucha activa y aplican pensamiento creativo e innovador para determinar el número, prototipo y estructura de las funciones, así como las variables tanto globales como locales a utilizar y determinar si usarán el paso por valor entre funciones para resolver el problema computable relativo al cálculo de las incertidumbres derivadas de mediciones directas. El docente elige al azar un equipo para que presente su propuesta de solución. El equipo seleccionado expone su propuesta, responde los cuestionamientos de los demás equipos y complementa su propuesta, considerando los comentarios y recomendaciones de otros equipos.

El docente retroalimenta las aportaciones. Ilustra la declaración, definición y uso de arreglos unidimensionales para solucionar el problema computable sobre la conversión de un número decimal a binario, pormenoriza tanto la sintaxis como los beneficios de utilizar arreglos bidimensionales, hipermatrices y arreglos dinámicos. Exhorta a los discentes a efectuar el diagrama de flujo estructurado y el programa computacional para llenar e imprimir una tabla de ajustes y tolerancias, usando arreglos y los elementos del lenguaje de programación empleado.

Los estudiantes, en su diagrama de flujo estructurado y en el programa computacional, emplean arreglos para completar y desplegar en pantalla la tabla indicada. El docente retroalimenta tanto el diagrama de flujo como el programa computacional, ejemplifica el empleo de funciones, arreglos y un lenguaje de programación en la realización de módulos computacionales para solucionar problemas computables de mecatrónica y solicita a los discentes que apliquen para resolver el problema computable planteado: Cálculo de las incertidumbres resultantes de mediciones directas.

Los estudiantes, en su equipo, diseñan los diagramas de flujo, efectúan las pruebas de escritorio, utilizan funciones, arreglos y un lenguaje de programación para elaborar los módulos computacionales encaminados a resolver el problema computable planteado, validan los módulos computacionales con tanto los casos como los valores de las pruebas de escritorio y efectúan una presentación multimedia para exponer en plenaria su solución al problema computable; durante la resolución del problema computable, dirimen conflictos, trabajan en forma colaborativa y aplican pensamiento creativo e innovador. Eligen un orador para presentar en plenaria su resolución del problema computable, el orador expone su solución, responde las preguntas del grupo y anota los comentarios y recomendaciones del grupo. Al finalizar las exposiciones, los discentes se retroalimentan, solventan sus dudas y expresan las ventajas e importancia de activar sus inteligencias múltiples para incentivar el pensamiento creativo e innovador y aplicarlos en la realización de módulos computacionales.

El docente retroalimenta las soluciones presentadas por los estudiantes y les invita a elaborar la evidencia de aprendizaje formativa.

Herramientas Tecnológicas y Recursos Didácticos	Evidencia de Aprendizaje para la Evaluación Formativa	Criterios e Instrumento de Evaluación
<ul style="list-style-type: none"> • Videos sobre: <ul style="list-style-type: none"> ○ Uso de funciones, bibliotecas o módulos. ○ Creación y utilización de funciones, bibliotecas y módulos definidos por el usuario. ○ Ejemplos de funciones definidas por el usuario. ○ Ejemplos de empleo de arreglos. ○ Arreglos unidimensionales, bidimensionales, hipermatrices y arreglos dinámicos. • Lenguaje de programación o IDE (Integrated Development Environment). • Blog sobre: <ul style="list-style-type: none"> ○ Variables globales y locales. ○ Paso por valor en funciones. ○ Léxico, sintaxis y semántica para prototipar, definir e invocar funciones. 	<ul style="list-style-type: none"> • Historieta “Dispensador de efectivo”. 	<p>La historieta incluye:</p> <ul style="list-style-type: none"> ➢ Portada ➢ Datos de identificación ➢ Personaje central alrededor del cual gira el argumento. ➢ Diálogos, movimiento y expresión de los personajes dibujados para explicar el argumento. ➢ Contenido <ul style="list-style-type: none"> ○ Tiene secuencia lógica y estructura de cuento o novela ○ Combina elementos verbales con imágenes. ○ Describe las funciones del dispensador, validación de la clave de acceso, lectura y entrega de la cantidad a retirar. ○ Explica el procedimiento para entregar el importe solicitado, considerando el saldo del usuario, el importe disponible en el dispensador y cantidad mínima de billetes a entregar.





Programa Académico: Técnico en Mecatrónica

Unidad de Aprendizaje: Lenguaje de Programación

<ul style="list-style-type: none"> ○ Declaración, definición y uso de arreglos. ● <i>Software</i> gratuito para elaborar diagramas de flujo. ● Computadora o celular con conectividad a internet. ● Manuales de programación en el lenguaje de programación empleado (C++, Dev C++, Cxxdroid, entre otros). ● Tutoriales sobre aplicaciones del lenguaje empleado. ● Libros impresos o digitales sobre el lenguaje de programación empleado ● Cursos en línea de programación en el lenguaje de programación empleado. 		<ul style="list-style-type: none"> ○ Integra los diagramas de flujo de las funciones para validar la clave de acceso, presentar el menú de cantidades a retirar y proporcionar la cantidad solicitada. La función definida por el usuario para validar la clave permite un máximo de tres intentos para ingresar la clave correcta. La función definida por el usuario para entregar la cantidad solicitada usa arreglos para guardar la cantidad de billetes de las denominaciones disponibles en el dispensador, incorpora fórmulas, estructuras cíclicas y de bifurcación para calcular el número mínimo de billetes a entregar o enviar mensajes pertinentes si usuario ingresa opciones o datos de entrada incorrectos. ○ Incluye las pruebas de escritorio de las funciones, con casos y valores válidos e incorrectos para verificar el control del flujo. ○ Integra los módulos computacionales, codificados a partir de los diagramas de flujo diseñados. ○ Existe coincidencia de 95 - 100 % entre los diagramas de flujo y los módulos computacionales. ○ Los diagramas de flujo diseñados y los módulos computacionales elaborados evidencian la aplicación de pensamiento creativo e innovador. ○ Adjunta fotografías o capturas de pantalla de los resultados de la ejecución de cada módulo computacional. ➤ Conclusiones: <ul style="list-style-type: none"> ○ Explica la importancia de emplear funciones, arreglos, un lenguaje de programación y pensamiento creativo e innovador en la realización de módulos computacionales para solucionar problemas computables de mecatrónica. ○ Describe las dificultades enfrentadas durante la realización de los módulos computacionales. ○ Utiliza tres o más fuentes de información para argumentar su punto de vista. ➤ Referencias: <ul style="list-style-type: none"> ○ Integra tres o más referencias en estilo APA. ➤ Su enfoque es serio. <p>Instrumento de evaluación Lista de cotejo</p>
---	--	--





Programa Académico: Técnico en Mecatrónica

Unidad de Aprendizaje: Lenguaje de Programación

Unidad didáctica:	Unidad 3. Estructuras de datos	Nivel:	Cuarto
Propósito:	Desarrolla sistemas computacionales básicos, aplicando la lógica de programación, con la finalidad de procesar datos de problemas computables de mecatrónica, en forma creativa, innovadora y centrada en las necesidades de los usuarios, bajo el enfoque del desarrollo sustentable.		
Unidad de competencia N°3:	Desarrolla sistemas computacionales básicos, creativos, innovadores y centrados en las necesidades de los usuarios, empleando módulos computacionales, estructuras de datos, un lenguaje de programación y pensamiento computacional para resolver problemas computables de mecatrónica, de acuerdo con los preceptos de sustentabilidad.		
Aprendizaje Esperado No 2:	Elabora sistemas computacionales básicos, centrados en las necesidades de los usuarios, para resolver problemas computables de mecatrónica, utilizando estructuras de datos, un lenguaje de programación y pensamiento computacional, bajo el enfoque del desarrollo sustentable.	Tiempo estimado para obtener el Aprendizaje Esperado:	16 horas

Contenidos de Aprendizaje

Conceptuales:	Procedimentales:	Actitudinales:
1) Estructuras <ul style="list-style-type: none"> 1.1) Definición y declaración de una estructura 1.2) Creación de una estructura 1.3) Uso de estructuras 2) Implementación de bibliotecas o módulos del usuario del lenguaje de programación 3) Ecodiseño y modelo de desarrollo de <i>software</i> verde 4) Programación orientada a objetos <ul style="list-style-type: none"> 4.1) Definiciones básicas <ul style="list-style-type: none"> 4.1.1) Clase y objeto 4.1.2) Encapsulación 4.1.3) Herencia 4.1.4) Polimorfismo 4.2) Programación de una clase 	Emplea estructuras de datos y un lenguaje de programación en la elaboración de sistemas computacionales básicos, enfocados a la resolución de problemas computables de mecatrónica. Práctica 6. Bitácora del taller de mecatrónica.	Efectúa sistemas computacionales básicos, centrados en las necesidades de los usuarios, aplicando pensamiento computacional, bajo el enfoque del desarrollo sustentable, para resolver problemas computables de mecatrónica.

Estrategias Didácticas y Ambientes de Aprendizaje

Estrategia didáctica: Aprendizaje por Proyectos (APP) y Design Thinking (DT)

El docente plantea el proyecto “Control del almacén de un taller de mecatrónica”, mostrando un escenario por medio de una presentación, y exhorta a los estudiantes a organizarse en equipos para realizar el proyecto.

Los estudiantes se organizan en equipo, cada uno de sus integrantes aporta su punto de vista, plantea qué sabe y qué necesita saber del proyecto, considera reflexivamente los puntos de vista de los otros para determinar la forma de llevar a cabo el proyecto. Cada equipo de trabajo propone su solución a la situación-problema, definiendo un curso de acción.

El docente describe las fases del *Design Thinking* (empatizar, definir, idear, prototipar y testear), así como diversas técnicas utilizadas en cada etapa (mapa de empatía, lluvia de ideas, *story telling*, entre otras), asimismo, detalla la definición, declaración, uso y codificación de estructuras, el ecodiseño, el modelo de desarrollo de *software* verde y la implementación de bibliotecas o módulos del usuario del lenguaje de programación, con la finalidad de emplearlos en la elaboración de sistemas computacionales básicos, centrados en las necesidades de los usuarios, bajo el enfoque del desarrollo sustentable. Solicita a los estudiantes que desarrollen el proyecto y reflexionen sobre cómo vincularían su solución con el Proyecto Aula.





Programa Académico: Técnico en Mecatrónica

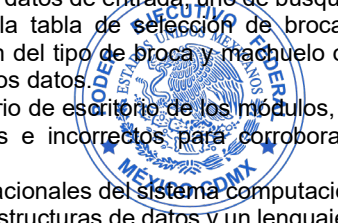
Unidad de Aprendizaje: Lenguaje de Programación

Los estudiantes realizan el mapa de empatía para acotar el proyecto, en función de las necesidades del usuario y las dimensiones del desarrollo sustentable, llevan a cabo una lluvia de ideas para proponer soluciones y elegir la mejor, es decir, aplican las fases de empatizar, definir e idear del DT. Diseñan los diagramas de flujo, realizan las pruebas de escritorio y efectúan las bibliotecas o los módulos del usuario del lenguaje de programación; en otras palabras, aplican la fase de prototipar del DT. Cabe señalar que la estructura a emplear para guardar los datos de los componentes presentes en el almacén de un taller de mecatrónica debe contener al menos las variables nombre, unidad de medida, número de serie y clave de localización.

Los discentes realizan la fase de testear los módulos computacionales; una vez que verifiquen que los módulos realizan la función para la que fueron desarrollados, los integran para obtener el sistema computacional básico y someterlo a la fase de testear, a fin de corroborar que es estructurado, funcional y está acorde con tanto las necesidades de los usuarios como el enfoque del desarrollo sustentable; en cada fase del desarrollo del sistema computacional básico, los estudiantes aplican pensamiento computacional. Redactan el informe ejecutivo, proponen alternativas para enlazar su solución con el Proyecto Aula, presentan en asamblea su sistema computacional, evalúan lo aprendido, emiten sus conclusiones sobre las dificultades que se les presentaron, así como sus reflexiones en torno a que saben ahora y qué necesitan saber del proyecto abordado.

El docente sintetiza y retroalimenta las conclusiones de los estudiantes. Describe qué es la programación orientada a objetos, las definiciones básicas y cómo programar una clase, asimismo, exhorta a los estudiantes a comparar una estructura y una clase para identificar tanto sus similitudes como sus diferencias. Los estudiantes contrastan las características de una estructura contra una clase, determinan sus casos de uso y programan una clase. El docente solicita a los estudiantes que realicen la evidencia de aprendizaje formativa.

Herramientas Tecnológicas y Recursos Didácticos	Evidencia de Aprendizaje para la Evaluación Formativa	Criterios e Instrumento de Evaluación
<ul style="list-style-type: none"> • Videos acerca de: <ul style="list-style-type: none"> ○ Declaración, definición y uso de estructuras en el lenguaje de programación utilizado. ○ Ejemplos de uso de estructuras, bibliotecas y módulos en el lenguaje de programación empleado. • Lenguaje de programación o IDE (Integrated Development Environment). • Blog sobre: <ul style="list-style-type: none"> ○ Estructuras. ○ Bibliotecas o módulos creados por el usuario. ○ Léxico, sintaxis y semántica para estructuras, bibliotecas y módulos. ○ <i>Software</i> gratuito para elaborar diagramas de flujo. • Sitios web sobre: <ul style="list-style-type: none"> ○ Beneficios y principios del ecodiseño y del modelo de desarrollo de <i>software</i> verde. ○ Ejemplos de ecodiseño en ingeniería. ○ Ejemplos de ecodiseño en desarrollo de <i>software</i>. ○ Ejemplos de programación de una clase. • Mapas conceptuales acerca de: <ul style="list-style-type: none"> ○ Definiciones y características de la programación orientada a objetos. 	<p>Ensayo analítico “Selección de brocas y machuelos para cuerdas interiores”.</p>	<p>El ensayo analítico incluye:</p> <ul style="list-style-type: none"> ➤ Carátula: ➤ Título: Define los contenidos del ensayo analítico. <ul style="list-style-type: none"> ○ Introducción: Establece el objetivo del ensayo, situación-problema o pregunta detonadora. ➤ Desarrollo: <ul style="list-style-type: none"> ○ Describe el proceso para machuelear una superficie (hacer las cuerdas interiores) para alojar un tornillo. ○ Caracteriza las cuerdas fina NF, estándar NC y especial NS. ○ Explica el uso de tablas de selección de brocas y machuelos en el proceso de hacer cuerdas interiores. ○ Reseña los módulos que integran el sistema computacional. ○ Presenta los diagramas de flujo estructurados de los módulos del sistema computacional básico; éste incluye, al menos, un módulo principal, uno de lectura de los datos de entrada, uno de búsqueda de los datos de entrada en la tabla de selección de brocas y machuelos y uno de impresión del tipo de broca y machuelo o un mensaje si no se encuentran los datos. ○ Incluye las pruebas de escritorio de escritorio de los módulos, con los casos y valores correctos e incorrectos para corroborar el control del flujo., ○ Contiene los módulos computacionales del sistema computacional básico; en éstos se emplean estructuras de datos y un lenguaje de





Programa Académico: Técnico en Mecatrónica

Unidad de Aprendizaje: Lenguaje de Programación

<ul style="list-style-type: none"> ○ Características y representaciones de clases y objetos. ○ Propiedades de los objetos. ○ Lenguajes orientados a objetos. ● Infografías sobre: <ul style="list-style-type: none"> ○ Programación orientada a objetos. ○ Componentes de la programación orientada a objetos. ● Computadora o celular con conectividad a Internet. ● Manuales de programación en el lenguaje de programación empleado (C++, Dev C++, Cxxdroid, entre otros). ● Tablas de selección de brocas y machuelos. ● Tutoriales sobre aplicaciones del lenguaje empleado. ● Libros impresos o digitales sobre el lenguaje de programación empleado. ● Cuadros de doble entrada de las estructuras y clases. ● Cuadros de doble entrada de la programación estructurada y la programación orientada a objetos. ● Cursos en línea de programación en el lenguaje de programación empleado. 		<p>programación para resolver el problema computable de mecatrónica planteado.</p> <ul style="list-style-type: none"> ○ Los módulos contienen estructuras de control para realizar la tarea asignada: lectura, búsqueda, cálculos e impresión de resultados o mensaje pertinente si el usuario realiza una operación no válida o introduce datos erróneos. ○ Existe concordancia de 95 % a 100 % del módulo con su diagrama de flujo. ○ Los módulos cuentan con sangrías, están documentados y carecen de errores de sintaxis, léxico y semántica . ○ Aplica <i>design thinking</i>, pensamiento computacional y el modelo de desarrollo de <i>software</i> verde en la elaboración del sistema computacional básico. ○ Incluye fotografías o capturas de pantalla de la funcionalidad del sistema computacional básico. <p>➤ Conclusiones:</p> <ul style="list-style-type: none"> ○ Responde a la situación-problema, el objetivo o la pregunta detonadora, describiendo la importancia del empleo de módulos computacionales, estructuras de datos, un lenguaje de programación, pensamiento computacional y el enfoque del desarrollo sustentable en la elaboración del sistema computacional básico. ○ Describe los obstáculos encarados durante la realización del sistema computacional básico. ○ Sustenta sus argumentos en tres o más fuentes de información. <p>➤ Referencias</p> <ul style="list-style-type: none"> ○ Incorpora tres o más referencias en estilo APA. <p>Instrumento de evaluación Lista de cotejo</p>
---	--	---





Programa Académico: Técnico en Mecatrónica

Unidad de Aprendizaje: Lenguaje de Programación

PRÁCTICA

N° y Nombre de la Práctica:	Punto medio entre dos puntos	N° de la Práctica:	1	Tiempo:	6 horas
Unidades del Programa de Estudio: 1	Utiliza la estructura secuencial de la programación estructurada y los elementos de un lenguaje de programación para solucionar problemas computables, aplicando razonamiento abstracto, creatividad y pensamiento lógico.				
Aprendizajes Esperados Relacionados con la Práctica: 1	Formula algoritmos o diagramas de flujo sistematizados, normalizados y sin brechas en la lógica, con el propósito de diseñar la solución de problemas computables de ICFM, aplicando con creatividad las fases de solución de un problema computable y razonamiento abstracto.				

Contenidos de Aprendizaje

Conceptuales:	Procedimentales:	Actitudinales:
1) Fases de solución de un problema computable 1.1) Noción de problema 1.2) Problemas no computables y problemas computables 1.3) Fases de solución de un problema computable 2) Diseño de algoritmos 2.1) Definición, elementos y características de un algoritmo 2.2) Diseño de algoritmos de problemas computables 3) Diseño de diagramas de flujo 3.1) Definición, elementos y características de un diagrama de flujo 3.2) Simbología normalizada para elaborar diagramas de flujo 3.3) Diseño de diagramas de flujo de problemas computables 3.4) Tipos de datos 3.5) Prueba de escritorio	Formula algoritmos o diagramas de flujo sistematizados, normalizados y sin brechas en la lógica para diseñar la solución de problemas computables de ICFM (Ingeniería y Ciencias Físico-Matemáticas).	Emplea creatividad y razonamiento abstracto en el diseño de la solución de problemas computables de ICFM.

Estrategias Didácticas y Ambientes de Aprendizaje

Estrategia didáctica: Ejercitación

El docente, mediante una presentación, describe el procedimiento para calcular el punto medio entre dos puntos, dadas sus coordenadas cartesianas.

Los estudiantes aplican las fases de solución de un problema computable para definir el problema computable a resolver, determinan los tipos de datos a emplear, diseñan el algoritmo, diseñan el diagrama de flujo y realizan la prueba de escritorio. Elaboran el reporte y lo entregan al docente.





Programa Académico: Técnico en Mecatrónica

Unidad de Aprendizaje: Lenguaje de Programación

Herramientas Tecnológicas y Recursos Didácticos	Evidencias de Aprendizaje para la Evaluación Formativa	Criterios e Instrumentos de Evaluación
<ul style="list-style-type: none"> • Presentación del escenario del problema computable a resolver. • <i>Software</i> gratuito para elaborar diagramas de flujo. • Computadora o celular con conectividad a Internet. 	<p>Reporte de la practica 1.</p>	<p>El reporte de la práctica incluye:</p> <ul style="list-style-type: none"> ➤ Datos de identificación: ➤ Introducción: Contiene objetivo y descripción del contenido. ➤ Desarrollo: <ul style="list-style-type: none"> ○ Especifica el tipo de las variables de entrada, proceso y salida. ○ Incluye algoritmo finito, preciso y definido; éste lee las coordenadas cartesianas de los dos puntos, calcula e imprime las coordenadas del punto medio entre los dos puntos. ○ Contiene diagrama de flujo con los símbolos normalizados para representar cada acción del algoritmo. ○ Integra la prueba de escritorio con los casos y valores del problema computable. ○ El diseño del algoritmo y la elaboración del diagrama de flujo evidencian el empleo de creatividad y razonamiento abstracto. ➤ Conclusiones: <ul style="list-style-type: none"> ○ Arguye, con base en una o dos fuentes de información, las oportunidades de mejora de la solución planteada. ➤ Referencias: <ul style="list-style-type: none"> ○ Incorpora una o dos referencias en estilo APA. <p>Instrumento de evaluación Lista de cotejo</p>



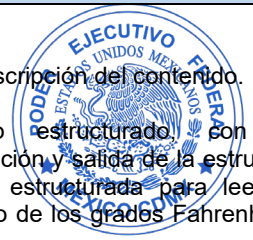


Programa Académico: Técnico en Mecatrónica

Unidad de Aprendizaje: Lenguaje de Programación

PRÁCTICA

N° y Nombre de la Práctica:	Conversión de grados Celsius a grados Fahrenheit	N° de la Práctica:	2	Tiempo:	6 horas
Unidades del Programa de Estudio: 1	Utiliza la estructura secuencial de la programación estructurada y los elementos de un lenguaje de programación para solucionar problemas computables, aplicando razonamiento abstracto, creatividad y pensamiento lógico.				
Aprendizajes Esperados Relacionados con la Práctica: 2	Codifica diagramas de flujo para desarrollar la solución de problemas computables de ICFM, empleando estructuras secuenciales, elementos de un lenguaje de programación y pensamiento lógico.				
Contenidos de Aprendizaje					
Conceptuales:		Procedimentales:		Actitudinales:	
1) Estructura secuencial 2) Codificación de diagramas de flujo 3) Problemática ambiental en mecatrónica		Utiliza estructuras secuenciales y simbología normalizada para diseñar diagramas de flujo estructurados, con el propósito de solucionar problemas computables de ICFM. Traduce las especificaciones de los diagramas de flujo diseñados a instrucciones equivalentes de un lenguaje de programación para desarrollar la solución de problemas computables de ICFM.		Aplica pensamiento lógico para codificar diagramas de flujo estructurados.	
Estrategias Didácticas y Ambientes de Aprendizaje					
Estrategia Didáctica: Ejercitación El docente pregunta a los estudiantes ¿cómo se convierten grados Celsius a grados Fahrenheit? Los estudiantes expresan la fórmula para la conversión. El docente retroalimenta a los discentes y les solicita que efectúen práctica. Los estudiantes elaboran el diagrama de flujo, la prueba de escritorio y la codificación para solucionar el problema computable planteado. Elaboran el reporte y lo entregan al docente.					
Herramientas Tecnológicas y Recursos Didácticos		Evidencias de Aprendizaje para la Evaluación Formativa		Criterios e Instrumentos de Evaluación	
<ul style="list-style-type: none"> • <i>Software</i> gratuito para elaborar diagramas de flujo . • Computadora o celular con conectividad a Internet. • Lenguaje de programación o IDE (Integrated Development Environment). • Manuales de programación en el lenguaje de programación empleado (C++, Dev C++, Cxxdroid, entre otros). 		Reporte de la práctica 2.		El reporte de la practica incluye: ➤ Datos de identificación ➤ Introducción: Incluye objetivo y descripción del contenido. ➤ Desarrollo: <ul style="list-style-type: none"> ○ Contiene diagrama de flujo estructurado con las representaciones entrada, asignación y salida de la estructura secuencial de la programación estructurada para leer los grados Celsius, realizar el cálculo de los grados Fahrenheit e imprimir el resultado. 	





Programa Académico: Técnico en Mecatrónica

Unidad de Aprendizaje: Lenguaje de Programación

		<ul style="list-style-type: none"> ○ Integra la prueba de escritorio con los casos y valores del problema computable. ○ Incluye la codificación de las acciones expresadas en el diagrama de flujo, está documentada, cuenta con los sangrías pertinentes y carece de errores de sintaxis, léxico ni semántica. ○ El código es validado con la prueba de escritorio efectuada. ○ El diagrama de flujo y la codificación coinciden 95 - 100 % ○ El diagrama de flujo diseñado y la codificación evidencian la aplicación de pensamiento lógico. <p>➤ Conclusiones:</p> <ul style="list-style-type: none"> ○ Argumenta, de acuerdo con una o dos fuentes de información, la importancia de los diagramas de flujo, las estructuras secuenciales, la codificación de diagramas de flujo y el pensamiento lógico en el desarrollo de la solución de problemas computables de ICFM. <p>➤ Referencias:</p> <ul style="list-style-type: none"> ○ Incorpora una o dos referencias en estilo APA. <p>Instrumento de evaluación Lista de cotejo</p>
--	--	--



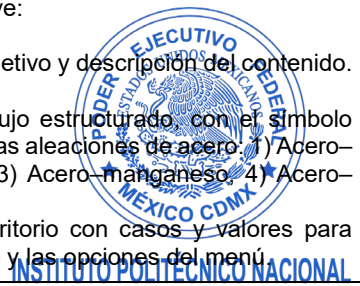


Programa Académico: Técnico en Mecatrónica

Unidad de Aprendizaje: Lenguaje de Programación

PRÁCTICA

N° y Nombre de la Práctica:	Usos de aleaciones de acero	N° de la Práctica:	3	Tiempo:	6 horas
Unidades del Programa de Estudio: 2	Realiza programas computacionales, aplicando estructuras de control de flujo y elementos de un lenguaje de programación, para solucionar problemas computables de electricidad, electrónica o tecnología de materiales, con creatividad y pensamiento analítico.				
Aprendizajes Esperados Relacionados con la Práctica: 1	Utiliza las estructuras de bifurcación simple o múltiple de la programación estructurada y un lenguaje de programación para desarrollar la solución de problemas computables de electricidad, electrónica o tecnología de materiales, aplicando pensamiento analítico y creativo.				
Contenidos de Aprendizaje					
Conceptuales:	Procedimentales:	Actitudinales:			
1. Estructuras de bifurcación 1.1) Bifurcación simple 1.1.1) <i>Top down design</i> 1.1.2) Anidamiento 1.1.3) Operadores lógicos y relacionales 1.2) Bifurcación múltiple	Aplica en forma diferenciada las estructuras de decisión simple o múltiple y un lenguaje de programación en el desarrollo de la solución de problemas computables de electricidad, electrónica o tecnología de materiales.	Desarrolla la solución de problemas computables de electricidad, electrónica o tecnología de materiales, con pensamiento analítico y creativo.			
Estrategias Didácticas y Ambientes de Aprendizaje					
Estrategia didáctica: Ejercitación El docente suministra un cuadro de doble entrada sobre las aleaciones del acero con cromo, níquel, manganeso y silicio, con sus respectivos usos en diversos procesos de fabricación; exhorta a los estudiantes a efectuar la práctica. Los discentes definen el problema computable a resolver, los tipos de datos a emplear, diseñan el diagrama de flujo, efectúan la prueba de escritorio y realizan el programa computacional empleando estructuras de bifurcación simple o múltiple. Elaboran su reporte y lo entregan al docente.					
Herramientas Tecnológicas y Recursos Didácticos	Evidencias de Aprendizaje para la Evaluación Formativa	Criterios e Instrumentos de Evaluación			
<ul style="list-style-type: none"> Cuadro de doble entrada sobre las aleaciones del acero con cromo, níquel, manganeso y silicio, con sus respectivos usos en diversos procesos de fabricación Lenguaje de programación o IDE (Integrated Development Environment). Software gratuito para elaborar diagramas de flujo. Computadora o celular con conectividad a Internet. Manuales de programación en el lenguaje de programación empleado (C++, Dev C++, Cxxdroid, entre otros). Tutoriales sobre aplicaciones del lenguaje empleado. 	Reporte de la práctica 3.	El reporte de la práctica incluye: <ul style="list-style-type: none"> ➤ Datos de identificación <ul style="list-style-type: none"> ○ Introducción: Contiene objetivo y descripción del contenido. ➤ Desarrollo: <ul style="list-style-type: none"> ○ Incluye el diagrama de flujo estructurado, con el símbolo para imprimir el menú de las aleaciones de acero: 1) Acero-cromo, 2) Acero-níquel, 3) Acero-manganeso, 4) Acero-silicio y 5) Salir. ○ Integra la prueba de escritorio con casos y valores para verificar el control del flujo y las opciones del menú 			





Programa Académico: Técnico en Mecatrónica

Unidad de Aprendizaje: Lenguaje de Programación

		<ul style="list-style-type: none"> ○ Incluye la codificación para indicar el uso de la aleación de acero elegida, tiene comentarios y sangrados, y no tiene errores de sintaxis, léxico y semántica. ○ El programa es validado con los casos y valores de la prueba de escritorio. ○ El diagrama de flujo y la codificación coinciden entre 90 y 100 %. ○ Imprime el uso de la aleación seleccionada, su inexistencia o un aviso si los datos no son válidos. ○ El diagrama de flujo diseñado y el programa computacional elaborado patentizan la aplicación de pensamiento analítico y creativo. ➤ Conclusiones: <ul style="list-style-type: none"> ○ Argumenta, con base en dos o más fuentes de información, la importancia de usar estructuras de bifurcación, <i>top down design</i>, elementos de un lenguaje de programación, pensamiento analítico y creativo en el desarrollo de la solución de problemas computables de electricidad, electrónica o tecnología de materiales. ➤ Referencias <ul style="list-style-type: none"> ○ Incorpora dos o más referencias en estilo APA. <p>Instrumento de evaluación Lista de cotejo</p>
--	--	---





Programa Académico: Técnico en Mecatrónica

Unidad de Aprendizaje: Lenguaje de Programación

PRÁCTICA

N° y Nombre de la Práctica:	Resistencia total en un circuito paralelo	N° de la Práctica:	4	Tiempo:	12 horas
Unidades del Programa de Estudio: 2	Realiza programas computacionales, aplicando estructuras de control de flujo y elementos de un lenguaje de programación, para solucionar problemas computables de electricidad, electrónica o tecnología de materiales, con creatividad y pensamiento analítico.				
Aprendizajes Esperados Relacionados con la Práctica: 2	Emplea las estructuras iterativas de la programación estructurada y un lenguaje de programación para desarrollar la solución de problemas computables de electricidad, electrónica o tecnología de materiales, aplicando creatividad y pensamiento analítico.				
Contenidos de Aprendizaje					
Conceptuales:		Procedimentales:		Actitudinales:	
1) Estructuras cíclicas 1.1) Ciclo for 1.2) Ciclo while. 1.3) Ciclo do-while		Aplica en forma diferenciada las estructuras iterativas y un lenguaje de programación en el desarrollo de la solución de problemas computables de electricidad, electrónica o tecnología de materiales.		Desarrolla la solución de problemas computables de electricidad, electrónica o tecnología de materiales, con creatividad y pensamiento analítico.	
Estrategias Didácticas y Ambientes de Aprendizaje					
Estrategia didáctica: Ejercitación El docente muestra el diagrama eléctrico de un circuito con cinco resistores en paralelo. Insta a los estudiantes a efectuar el programa computacional para obtener la resistencia total del circuito presentado. Los estudiantes utilizan estructuras iterativas en el diseño del diagrama de flujo y la elaboración del programa computacional. Verifican la funcionalidad del programa mediante los casos y valores de la prueba de escritorio. Elaboran su reporte y lo entregan al docente.					
Herramientas Tecnológicas y Recursos Didácticos		Evidencias de Aprendizaje para la Evaluación Formativa		Criterios e Instrumentos de Evaluación	
<ul style="list-style-type: none"> Diagrama eléctrico de un circuito paralelo con cinco resistores. Lenguaje de programación o IDE (Integrated Development Environment). Software gratuito para elaborar diagramas de flujo. Computadora o celular con conectividad a Internet. Manuales de programación en el lenguaje de programación empleado (C++, Dev C++, Cxxdroid, entre otros). Tutoriales sobre aplicaciones del lenguaje empleado. 		Reporte de la práctica 4.		El reporte de la práctica incluye: <ul style="list-style-type: none"> ➤ Datos de identificación: Logotipo de la unidad académica, programa académico, grupo, título, nombre y fecha. ➤ Introducción: Contiene objetivo y descripción del contenido. ➤ Desarrollo: <ul style="list-style-type: none"> ○ Describe el procedimiento para calcular la resistencia total en un circuito paralelo. ○ Contiene diagrama de flujo con una estructura cíclica para leer los valores de los resistores, una estructura de bifurcación para validar que sean positivos, calcular la resistencia total e imprimir el resultado o un aviso si se ingresa datos no válidos. 	



Programa Académico: Técnico en Mecatrónica

Unidad de Aprendizaje: Lenguaje de Programación

		<ul style="list-style-type: none"> ○ Incluye prueba de escritorio con los casos y valores para comprobar el control de flujo. ○ Incorpora programa computacional documentado, con sangrados, sin errores de léxico, sintaxis ni de semántica. Coincide 90-100 % con el diagrama de flujo. ○ EL programa lee n medidas, verifica que sean positivas, calcula la media e imprime el resultado o los mensajes de error correspondientes. ○ El diseño del diagrama de flujo estructurado y programa computacional reflejan creatividad y pensamiento analítico ➤ Conclusiones: <ul style="list-style-type: none"> ○ Explica, apoyándose en dos o más fuentes de información, la utilidad de las estructuras iterativas, los elementos de un lenguaje de programación, el pensamiento analítico y creativo para solucionar problemas computables de electricidad, electrónica y tecnología de materiales. ➤ Referencias <ul style="list-style-type: none"> ○ Incorpora dos o más referencias en estilo APA. <p>Instrumento de evaluación: Lista de cotejo</p>
--	--	---





Programa Académico: Técnico en Mecatrónica

Unidad de Aprendizaje: Lenguaje de Programación

PRÁCTICA

N° y Nombre de la Práctica:	Magnitud del torque de un centro de lavado	N° de la Práctica:	5	Tiempo:	12 horas
Unidades del Programa de Estudio: 3	Desarrolla sistemas computacionales básicos, creativos, innovadores y centrados en las necesidades de los usuarios, empleando módulos computacionales, estructuras de datos, un lenguaje de programación y pensamiento computacional para resolver problemas computables de mecatrónica, de acuerdo con los preceptos de sustentabilidad.				
Aprendizajes Esperados Relacionados con la Práctica: 1	Realiza módulos computacionales para solucionar problemas computables de mecatrónica, empleando funciones, arreglos y un lenguaje de programación, con pensamiento creativo e innovador.				

Contenidos de Aprendizaje

Conceptuales:	Procedimentales:	Actitudinales:
1) Funciones 1.1) Prototipado, definición e invocación de funciones 1.2) Funciones definidas por el usuario 1.3) Tipos de variables 1.3.1) Locales 1.3.2) Globales 1.4) Paso por valor 2) Arreglos 2.1) Arreglos unidimensionales 2.2) Arreglos bidimensionales 2.3) Hipermatrices 2.4) Arreglos dinámicos	Utiliza funciones, arreglos y un lenguaje de programación para realizar módulos computacionales orientados a solucionar problemas computables de mecatrónica.	Aplica pensamiento creativo e innovador en la realización de módulos computacionales para solucionar problemas computables de mecatrónica.

Estrategias Didácticas y Ambientes de Aprendizaje

Estrategia didáctica: Ejercitación

El docente proporciona la tabla (real o hipotética) de torques a aplicar por un centro de lavado, en función del tipo de tejido de las prendas de vestir, y exhorta a los estudiantes a llevar a cabo los módulos computacionales para seleccionar la magnitud del torque que debe aplicar el sistema mecatrónico durante el ciclo de lavado.

Los estudiantes diseñan los diagramas de flujo, realizan las pruebas de escritorio y realizan los módulos computacionales, utilizando funciones, arreglos, elementos del lenguaje de programación utilizado y pensamiento creativo e innovador; verifican la funcionalidad de los módulos mediante los casos y valores de la prueba de escritorio. Elaboran su reporte y lo entregan al docente.

Herramientas Tecnológicas y Recursos Didácticos	Evidencias de Aprendizaje para la Evaluación Formativa	Criterios e Instrumentos de Evaluación
<ul style="list-style-type: none"> Tabla (real o hipotética) de torques a aplicar por un centro de lavado, en relación al tipo de tejido de las prendas de vestir. Lenguaje de programación o IDE (Integrated Development Environment). 	Reporte escrito de la práctica 5.	El reporte incluye: <ul style="list-style-type: none"> ➤ Datos de identificación ○ Introducción: Contiene objetivo y descripción del contenido. ➤ Desarrollo ○ Especifica cómo usar la tabla de torques.



Programa Académico: Técnico en Mecatrónica

Unidad de Aprendizaje: Lenguaje de Programación

<ul style="list-style-type: none"> • <i>Software</i> gratuito para elaborar diagramas de flujo. • Computadora o celular con conectividad a Internet. • Manuales de programación en el lenguaje de programación empleado (C++, Dev C++, Cxxdroid). • Tutoriales sobre aplicaciones del lenguaje empleado. 		<ul style="list-style-type: none"> ○ Incluye el diagrama de flujo de cada módulo computacional. El módulo principal imprime el título, presenta el menú de ciclos de lavado, solicita la elección de un ciclo, lee la opción e invoca al módulo para seleccionar la magnitud del torque o imprimir un aviso si se introduce una opción errónea. El módulo computacional para seleccionar la magnitud del torque contiene la tabla correspondiente, lee la opción de lavado enviada por el módulo principal, busca en la tabla e imprime la magnitud del torque. Cada módulo coincide entre 95-100% con su diagrama de flujo, emplean funciones o arreglos y los elementos del lenguaje de programación, carece de errores de sintaxis, léxico ni de semántica, incluye comentarios y sangrados e imprime los resultados o avisos si el usuario final ingresa opciones o datos no válidos ○ Incorpora la prueba de escritorio de cada módulo. ○ Cada diagrama de flujo diseñado y cada módulo elaborado patentizan la aplicación de pensamiento creativo e innovador. ○ Sintetiza cómo se resolvió el problema computable de mecatrónica planteado. ○ Integra fotografías o capturas de pantalla de la ejecución de cada módulo computacional. ➤ Conclusiones: <ul style="list-style-type: none"> ○ Explica, con base en dos fuentes de información, la importancia de diseñar los diagramas de flujo y emplear funciones, arreglos, un lenguaje de programación y pensamiento creativo e innovador en la solución de problemas computables de mecatrónica. ➤ Referencias <ul style="list-style-type: none"> ○ Incorpora tres o más referencias en estilo APA. <p>Instrumento de evaluación Lista de cotejo</p>
--	--	---





Programa Académico: Técnico en Mecatrónica

Unidad de Aprendizaje: Lenguaje de Programación

PRÁCTICA

N° y Nombre de la Práctica:	Bitácora del taller de mecatrónica	N° de la Práctica:	6	Tiempo:	12 horas
Unidades del Programa de Estudio: 3	Desarrolla sistemas computacionales básicos, creativos, innovadores y centrados en las necesidades de los usuarios, empleando módulos computacionales, estructuras de datos, un lenguaje de programación y pensamiento computacional para resolver problemas computables de mecatrónica, de acuerdo con los preceptos de sustentabilidad.				
Aprendizajes Esperados Relacionados con la Práctica: 2	Elabora sistemas computacionales básicos, centrados en las necesidades de los usuarios, para resolver problemas computables de mecatrónica, utilizando estructuras de datos, un lenguaje de programación y pensamiento computacional, bajo el enfoque del desarrollo sustentable.				

Contenidos de Aprendizaje

Conceptuales:	Procedimentales:	Actitudinales:
1) Estructuras <ul style="list-style-type: none"> 1.1) Definición y declaración de una estructura 1.2) Creación de una estructura 1.3) Uso de estructuras 2) Implementación de bibliotecas o módulos del usuario del lenguaje de programación 3) Ecodiseño y modelo de desarrollo de <i>software</i> verde 4) Programación orientada a objetos <ul style="list-style-type: none"> 4.1) Definiciones básicas <ul style="list-style-type: none"> 4.1.1) Clase y objeto 4.1.2) Encapsulación 4.1.3) Herencia 4.1.4) Polimorfismo 4.2) Programación de una clase 	Emplea estructuras de datos y un lenguaje de programación en la elaboración de sistema computacionales básicos, enfocados a la resolución de problemas computables de mecatrónica.	Efectúa sistemas computacionales básicos, centrados en las necesidades de los usuarios, aplicando pensamiento computacional, bajo el enfoque del desarrollo sustentable, para resolver problemas computables de mecatrónica.

Estrategias Didácticas y Ambientes de Aprendizaje

Estrategia didáctica: Ejercitación

El docente, con el apoyo de una presentación, plantea la importancia de controlar mediante una bitácora digital el acceso y permanencia de los estudiantes en el taller de mecatrónica, asimismo, describe los datos que ésta debe incluir: nombre del estudiante, número de boleta, hora de entrada, hora de salida y tiempo de estancia. Solicita a los estudiantes que elaboren el sistema computacional básico para llevar a cabo esta tarea.

Los estudiantes diseñan los diagramas de flujo, efectúan las pruebas de escritorio y realizan los módulos computacionales, empleando estructuras de datos, un lenguaje de programación, pensamiento computacional y el enfoque del desarrollo sustentable. Verifican la funcionalidad de los módulos mediante los casos y valores de su prueba de escritorio, integran los módulos para obtener el sistema computacional básico, comprueban su funcionalidad, elaboran el reporte de la práctica y lo entregan al docente.





Programa Académico: Técnico en Mecatrónica

Unidad de Aprendizaje: Lenguaje de Programación

Herramientas Tecnológicas y Recursos Didácticos	Evidencias de Aprendizaje para la Evaluación Formativa	Criterios e Instrumentos de Evaluación
<ul style="list-style-type: none"> • Presentación “Bitácora para controlar el acceso y permanencia en el taller de mecatrónica”. • Lenguaje de programación o IDE (Integrated Development Environment). • <i>Software</i> gratuito para elaborar diagramas de flujo. • Computadora o celular con conectividad a Internet. • Manuales de programación en el lenguaje de programación empleado (C++, Dev C++, Cxxdroid, entre otros). • Tutoriales sobre aplicaciones del lenguaje empleado. 	<p>Reporte de la práctica 6.</p>	<p>El reporte de la práctica incluye:</p> <ul style="list-style-type: none"> ➤ Carátula. <ul style="list-style-type: none"> ○ Introducción: Contiene objetivo, síntesis del contenido del reporte, y los resultados y las conclusiones más relevantes. ➤ Desarrollo: <ul style="list-style-type: none"> ○ Reseña el procedimiento para llevar el control de la entrada y estancia de los estudiantes en el taller de mecatrónica. ○ Reseña la estructura del sistema computacional básico y los módulos que lo integran. ○ Contiene los diagramas de flujo de los módulos computacionales, que resuelven la tarea para la que fueron diseñados. Su diseño patentiza la aplicación de pensamiento computacional y del modelo de desarrollo de <i>software</i> verde. El módulo principal usa una estructura para guardar los datos de los usuarios del taller de mecatrónica, imprime el título, presenta el menú con las opciones de la bitácora digital: 1) Ingreso, 2) Salida, 3) Impresión y 4) Salir. La opción 1 llama al módulo de lectura de entrada del usuario, la opción 2 llama al módulo de salida del usuario, la opción 3 llama al módulo de impresión y la opción 4 termina la ejecución del módulo principal. Utiliza una estructura iterativa para presentar el menú mientras que la opción no sea “Salir”. En caso de que el usuario ingrese una opción válida, se le envía un aviso. El módulo de lectura de entrada del usuario lee los datos y la hora de entrada del estudiante y los guarda en la estructura. El módulo de salida del usuario lee la hora de salida, la guarda en la estructura y llama al módulo de estancia. El módulo de estancia, a partir de la hora de entrada y la hora de salida del usuario, calcula el tiempo de permanencia del usuario y guarda esta variable en la estructura. El módulo de impresión utiliza una estructura de repetición para imprimir los datos y el tiempo de estancia de los usuarios del taller de mecatrónica guardados en la estructura. ○ Considera en la prueba de escritorio de cada módulo computacional, casos y valores válidos e incorrectos para comprobar el control del flujo. ○ Adjunta la codificación de los módulos computacionales; cada módulo contiene las bibliotecas para su ejecución, coincide



Programa Académico: Técnico en Mecatrónica

Unidad de Aprendizaje: Lenguaje de Programación

		<p>95-100 % con su diagrama de flujo, carece de errores de sintaxis, léxico ni de semántica, utiliza estructuras de control de la programación estructurada, está validado con los casos y valores de su prueba de escritorio e imprime los resultados correspondientes o los mensajes pertinentes si el usuario ingresa opciones o datos no válidos. Su elaboración evidencia la aplicación de pensamiento computacional y el modelo de desarrollo de <i>software</i> verde.</p> <ul style="list-style-type: none"> ○ Incorpora fotografías o capturas de pantalla de su funcionamiento integral del sistema computacional básico. Éste es creativo e innovador, está centrado en las necesidades de los usuarios y considera las dimensiones del desarrollo sustentable <p>➤ Conclusiones:</p> <ul style="list-style-type: none"> ○ Explica, con base en dos fuentes de información, la importancia de utilizar módulos computacionales, estructuras de datos, estructuras de control de la programación estructurada, un lenguaje de programación, pensamiento computacional y el enfoque del desarrollo sustentable en el desarrollo del sistema computacional básico, creativo, innovador y centrado en las necesidades de los usuarios. <p>➤ Referencias</p> <ul style="list-style-type: none"> ○ Incorpora tres o más referencias en estilo APA. <p>Instrumento de evaluación Lista de cotejo</p>
--	--	--





Programa Académico: Técnico en Mecatrónica

Unidad de Aprendizaje: Lenguaje de Programación

PLAN DE EVALUACIÓN SUMATIVA DEL CURSO

N°	Unidad de Competencia	Evidencia integradora	Criterios de Evaluación	Porcentaje de Acreditación
1	<p>Introducción a la programación estructurada Utiliza la estructura secuencial de la programación estructurada y los elementos de un lenguaje de programación para solucionar problemas computables, aplicando razonamiento abstracto, creatividad y pensamiento lógico.</p>	<p>Presentación ejecutiva “Problema computable de fisicomatemáticas”</p>	<p>La presentación ejecutiva incluye:</p> <ul style="list-style-type: none"> ➤ Portada ➤ Índice. ➤ Introducción: <ul style="list-style-type: none"> ○ Reseña el contenido de la presentación, el objetivo, los resultados y las conclusiones más relevantes. ➤ Desarrollo: <ul style="list-style-type: none"> ○ Incluye el diagrama de flujo estructurado, con las representaciones entrada, asignación y salida de la estructura secuencial de la programación estructurada para resolver el problema computable planteado. ○ Integra la prueba de escritorio con los casos y valores para corroborar el control del flujo. ○ Incluye la codificación del diagrama de flujo diseñado, está documentada, cuenta con las sangrías correspondientes y no tiene errores de sintaxis, léxico y semántica. ○ El programa es validado con los casos y valores de la prueba de escritorio. ○ El diagrama de flujo y la codificación coinciden entre 95 y 100 % ○ El diagrama de flujo diseñado y el programa computacional elaborado evidencian la aplicación de razonamiento abstracto, creatividad y pensamiento lógico. ➤ Conclusiones: <ul style="list-style-type: none"> ○ Argumenta, con base en una o dos fuentes de información, la importancia de diseñar algoritmos o diagramas de flujo, utilizar la estructura secuencial de la programación y aplicar razonamiento abstracto, creatividad y pensamiento lógico para solucionar problemas computables. ➤ Referencias <ul style="list-style-type: none"> ○ Incorpora referencias en estilo APA. 	<p>30%</p>

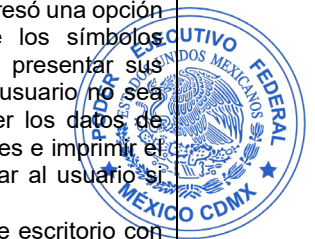




Programa Académico: Técnico en Mecatrónica

Unidad de Aprendizaje: Lenguaje de Programación

			<ul style="list-style-type: none"> ➤ Diseño <ul style="list-style-type: none"> ○ Incorpora videos o audios breves, imágenes estáticas, gifs o enlaces alusivos al tema. ○ Tiene una estructura coherente y sencilla de comprender. ○ Las diapositivas incluyen textos cortos y fáciles de leer. ○ Emplea una tipografía legible. ○ Utiliza fondos claros que posibilitan la lectura. ○ No tiene errores gramaticales ni ortográficos. ○ Está elaborada con un <i>software</i> y sitio web de herramientas de diseño gráfico. <p>Instrumento de evaluación Rúbrica de la presentación ejecutiva.</p>	
2	<p>Estructuras de control de flujo Realiza programas computacionales, aplicando estructuras de control de flujo y elementos de un lenguaje de programación, para solucionar problemas computables de electricidad, electrónica o tecnología de materiales, con creatividad y pensamiento analítico.</p>	Revista “Calculadora básica de electrónica y tecnología de materiales”.	<p>La revista incluye:</p> <ul style="list-style-type: none"> ➤ Portada: Contiene nombre de la revista, temas relevantes, sitio web y código QR. ○ Contraportada: Incorpora directorio y datos de identificación del equipo editor. ○ Índice: Incluye tema, página, síntesis de los temas y el número de página coincide con el tema. ➤ Contenido: Integra tres secciones: <ol style="list-style-type: none"> 1. Requisitos teóricos y prácticos para usar la calculadora básica de electrónica y tecnología de materiales, así como descripción de sus funciones. 2. Diagrama de flujo y programa computacional: <ul style="list-style-type: none"> ○ El diagrama de flujo incluye los símbolos normalizados para imprimir el título, presentar el menú principal, el submenú de electrónica, el submenú de tecnología de materiales, terminar el programa o imprimir un mensaje para avisarle al usuario si ingresó una opción no válida. Cada submenú contiene los símbolos normalizados para imprimir su título, presentar sus opciones mientras que la opción del usuario no sea “Salir”; leer la opción del usuario, leer los datos de entrada, realizar los cálculos pertinentes e imprimir el resultado, salir del submenú o notificar al usuario si ingresa una opción no permitida. ○ El diagrama de flujo incluye prueba de escritorio con casos y valores para comprobar el control de flujo en 	30%

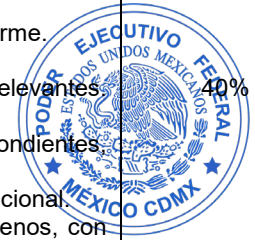




Programa Académico: Técnico en Mecatrónica

Unidad de Aprendizaje: Lenguaje de Programación


			<p>cada una de las opciones del menú principal y de los submenús.</p> <ul style="list-style-type: none"> ○ El programa computacional coincide 95 – 100 % con el diagrama de flujo, está documentado, cuenta con los sangrados pertinentes, carece de errores de sintaxis, léxico o semántica y está validado por los casos y valores de la prueba de escritorio. ○ El programa computacional es funcional. ○ El diagrama de flujo diseñado y el programa computacional elaborado patentizan el uso de estructuras de control de flujo, los elementos de un lenguaje de programación, creatividad razonamiento abstracto y pensamiento lógico para solucionar los problemas computables de electricidad, electrónica o tecnología de materiales. <p>3. Funcionamiento de la calculadora:</p> <ul style="list-style-type: none"> ○ Pormenoriza los pasos para utilizar cada opción del menú principal y de cada submenú de la calculadora. ○ Incluye imágenes que faciliten la comprensión del funcionamiento de cada sección y del menú principal. <p>➤ Referencias</p> <ul style="list-style-type: none"> ○ Incorpora de dos o tres referencias en estilo APA. <p>Instrumento de evaluación Rúbrica de la revista. Rúbrica de coevaluación.</p>	
3	<p>Estructuras de datos Desarrolla sistemas computacionales básicos, creativos, innovadores y centrados en las necesidades de los usuarios, empleando módulos computacionales, estructuras de datos, un lenguaje de programación y pensamiento computacional para resolver problemas computables de mecatrónica, de acuerdo con los preceptos de sustentabilidad.</p>	Informe ejecutivo “Parámetros de maquinado”	<p>El informe ejecutivo incluye:</p> <ul style="list-style-type: none"> ➤ Carátula ➤ Índice ➤ Introducción: <ul style="list-style-type: none"> ○ Describe de manera breve el contenido del informe. ○ Reseña el propósito del informe. ○ Reseña los resultados y las conclusiones más relevantes. ➤ Desarrollo <ul style="list-style-type: none"> ○ Explica la forma de utilizar las tablas correspondientes con base en el Sistema Métrico Decimal. ○ Reseña el funcionamiento del sistema computacional. ○ El sistema computacional básico cuenta, al menos, con un módulo principal y un módulo de apoyo para imprimir 	40%





Programa Académico: Técnico en Mecatrónica

Unidad de Aprendizaje: Lenguaje de Programación

			<p>los resultados o enviar mensajes pertinentes si el usuario cometió error. Emplea estructura de datos y un lenguaje de programación.</p> <ul style="list-style-type: none"> ○ El sistema computacional básico es creativo, innovador y centrado en las necesidades de los usuarios. ○ El sistema computacional básico evidencia que durante su desarrollo se aplicó pensamiento computacional y los preceptos de sustentabilidad. ○ Incorpora fotografías o capturas de pantalla para mostrar la funcionalidad del sistema computacional básico. ○ Justifica y defiende la forma de elaboración del sistema computacional básico. <p>➤ Conclusiones:</p> <ul style="list-style-type: none"> ○ Reseña la contribución de los módulos computacionales, las estructuras de datos, el lenguaje de programación, el pensamiento computacional y los preceptos del desarrollo sustentable para facilitar el desarrollo de sistemas computacionales básicos, creativos, innovadores y centrados en las necesidades de los usuarios para resolver problemas computables de mecatrónica, derivados de necesidades sociales. ○ Expresa sus reflexiones sobre cómo la puesta en práctica, en forma autónoma, en el futuro, de lo aprendido en la unidad de aprendizaje Lenguaje de Programación 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 <p>➤ Referencias</p> <ul style="list-style-type: none"> ○ Incorpora tres o más referencias en estilo APA. <p>Instrumento de evaluación Rúbrica del informe ejecutivo. Rubrica de coevaluación. Lista de cotejo de autoevaluación.</p> 
--	--	--	--



Programa Académico: Técnico en Mecatrónica

Unidad de Aprendizaje: Lenguaje de Programación

Propósito	Evidencia Integradora	Criterios de Evaluación	Porcentaje de Acreditación
<p>Desarrolla sistemas computacionales básicos, aplicando la lógica de programación, con la finalidad de procesar datos de problemas computables de mecatrónica, en forma creativa, innovadora y centrada en las necesidades de los usuarios, bajo el enfoque del desarrollo sustentable.</p>	<p>Trabajo monográfico “Funciones de un dispositivo mecatrónico”</p>	<p>El trabajo monográfico incluye:</p> <ul style="list-style-type: none"> ➤ Portada; Su diseño es atractivo. ➤ Datos de identificación ➤ Índice ➤ Introducción: <ul style="list-style-type: none"> ○ Incluye el objetivo. ○ Sintetiza el contenido del trabajo monográfico, los resultados y las conclusiones más importantes. ➤ Capítulo I: <ul style="list-style-type: none"> ○ Tiene un título asignado por el discente. ○ Explica la estructura del sistema computacional básico para procesar los datos inherentes a las funciones de un dispositivo mecatrónico. ➤ Capítulo II: <ul style="list-style-type: none"> ○ Tiene un título elegido por el estudiante. ○ Incorpora los diagramas de flujo y las pruebas de escritorio de los módulos correspondientes para presentar el menú de las funciones del dispositivo mecatrónico, realizar la operación solicitada e imprimir los resultados o avisos si el usuario efectúa una operación no válida o introduce datos erróneos. ➤ Capítulo III: <ul style="list-style-type: none"> ○ Tiene un título designado por el discente. ○ Incluye la codificación de los diagramas de flujo diseñados en el capítulo II. ○ Cada módulo coincide 100 % con su diagrama de flujo. ○ Cada módulo soluciona la tarea para la que fue elaborado. ➤ El diseño y desarrollo de los módulos computacionales evidencian el empleo de la lógica de programación, pensamiento computacional y el modelo de desarrollo de <i>software</i> verde. ➤ El sistema computacional básico es creativo, innovador, está centrado en las necesidades de los usuarios y considera el enfoque del desarrollo sustentable. ➤ Conclusiones: <ul style="list-style-type: none"> ○ Describe cómo la puesta en práctica, en forma autónoma, en el futuro, de las competencias adquiridas 	<p>100%</p>





Programa Académico: Técnico en Mecatrónica

Unidad de Aprendizaje: Lenguaje de Programación

		<p>desarrolladas y aplicadas en la unidad de aprendizaje Lenguaje de Programación le coadyuvaría a responder en forma eficiente y eficaz a los retos que se le presenten al incorporarse a estudios superiores o al campo laboral</p> <ul style="list-style-type: none"> ○ Discurre sobre los beneficios y retos a enfrentar al desarrollar, bajo el enfoque del desarrollo sustentable, sistemas computacionales básicos para resolver problemas computables de mecatrónica, derivados de necesidades sociales. ○ Sustenta sus reflexiones con base en cuatro o más referencias <p>➤ Referencias:</p> <ul style="list-style-type: none"> ○ Incorpora cuatro o más referencias en estilo APA. <p>➤ Anexo</p> <ul style="list-style-type: none"> ○ Tiene un título proporcionado por el estudiante. ○ Incluye fotografías, ilustraciones, o capturas de pantalla para evidenciar la funcionalidad de cada módulo y del sistema computacional básico en su conjunto. <p>Instrumento de evaluación Rúbrica del trabajo monográfico. Rúbrica de coevaluación Lista de cotejo de autoevaluación</p>	
--	--	---	--





Programa Académico: Técnico en Mecatrónica

Unidad de Aprendizaje: Lenguaje de Programación

PROGRAMA SINTÉTICO

PROGRAMA ACADÉMICO:		Técnico en Mecatrónica		UNIDAD DE APRENDIZAJE:		Lenguaje de Programación	
Horas Teóricas/Prácticas:		1 / 3		Área de Formación: Profesional		Créditos: 4.5	
		Rama de Conocimiento:		ICFM			
PROPÓSITO DE LA UNIDAD DE COMPETENCIA							
Desarrolla sistemas computacionales básicos, aplicando la lógica de programación, con la finalidad de procesar datos de problemas computables de mecatrónica, en forma creativa, innovadora y centrada en las necesidades de los usuarios, bajo el enfoque del desarrollo sustentable.							
N°	UNIDAD DE COMPETENCIA	APRENDIZAJES ESPERADOS		CONTENIDOS/SABERES			
1	Utiliza la estructura secuencial de la programación estructurada y los elementos de un lenguaje de programación para solucionar problemas computables, aplicando razonamiento abstracto, creatividad y pensamiento lógico.	1. Formula algoritmos o diagramas de flujo sistematizados, normalizados y sin brechas en la lógica, con el propósito de diseñar la solución de problemas computables de ICFM, aplicando con creatividad las fases de solución de un problema computable y razonamiento abstracto.		<p>Conceptuales</p> <p>1) Fases de solución de un problema computable</p> <p>1.1) Noción de problema</p> <p>1.2) Problemas no computables y problemas computables</p> <p>1.3) Fases de solución de un problema computable</p> <p>2) Diseño de algoritmos</p> <p>2.1) Definición, elementos y características de un algoritmo</p> <p>2.2) Diseño de algoritmos de problemas computables</p> <p>3) Diseño de diagramas de flujo</p> <p>3.1) Definición, elementos y características de un diagrama de flujo</p> <p>3.2) Simbología normalizada para elaborar diagramas de flujo</p> <p>3.3) Diseño de diagramas de flujo de problemas computables</p> <p>3.4) Tipos de datos</p> <p>3.5) Prueba de escritorio</p> <p>Procedimentales</p> <p>Formula algoritmos o diagramas de flujo sistematizados, normalizados y sin brechas en la lógica para diseñar la solución de problemas computables de ICFM (Ingeniería y Ciencias Fisicomatemáticas).</p> <p>Actitudinales</p> <p>Emplea creatividad y razonamiento abstracto en el diseño de la solución de problemas computables de ICFM</p>			
		2. Codifica diagramas de flujo para desarrollar la solución de problemas computables de ICFM, empleando estructuras secuenciales, elementos de un lenguaje de programación y pensamiento lógico.		<p>Conceptuales</p> <p>1) Estructura secuencial</p> <p>2) Codificación de diagramas de flujo</p> <p>3) Problemática ambiental en mecatrónica</p> <p>Procedimentales</p>			





Programa Académico: Técnico en Mecatrónica

Unidad de Aprendizaje: Lenguaje de Programación

			<p>Utiliza estructuras secuenciales y simbología normalizada para diseñar diagramas de flujo estructurados, con el propósito de solucionar problemas computables de ICFM.</p> <p>Traduce las especificaciones de los diagramas de flujo diseñados a instrucciones equivalentes de un lenguaje de programación para desarrollar la solución de problemas computables de ICFM.</p> <p>Actitudinales Aplica pensamiento lógico para codificar diagramas de flujo estructurados.</p>
2	<p>Realiza programas computacionales, aplicando estructuras de control de flujo y elementos de un lenguaje de programación, para solucionar problemas computables de electricidad, electrónica o tecnología de materiales, con creatividad y pensamiento analítico.</p>	<p>1. Utiliza las estructuras de bifurcación simple o múltiple de la programación estructurada y un lenguaje de programación para desarrollar la solución de problemas computables de electricidad, electrónica o tecnología de materiales, aplicando pensamiento analítico y creativo.</p> <p>2. Emplea las estructuras iterativas de la programación estructurada y un lenguaje de programación para desarrollar la solución de problemas computables de electricidad, electrónica o tecnología de materiales, aplicando creatividad y pensamiento analítico.</p>	<p>Conceptuales 1. Estructuras de bifurcación 1.1) Bifurcación simple 1.1.1) <i>Top down design</i> 1.1.2) Anidamiento 1.1.3) Operadores lógicos y relacionales 1.1.4) Bifurcación múltiple</p> <p>Procedimentales Aplica en forma diferenciada las estructuras de decisión simple o múltiple y un lenguaje de programación en el desarrollo de la solución de problemas computables de electricidad, electrónica o tecnología de materiales.</p> <p>Actitudinales Desarrolla la solución de problemas computables de electricidad, electrónica o tecnología de materiales, con pensamiento analítico y creativo.</p> <p>Conceptuales 1) Estructuras cíclicas 1.1) Ciclo for 1.2) Ciclo while 1.3) Ciclo do-while</p> <p>Procedimentales Aplica en forma diferenciada las estructuras iterativas y un lenguaje de programación en el desarrollo de la solución de problemas computables de electricidad, electrónica o tecnología de materiales.</p>





Programa Académico: Técnico en Mecatrónica

Unidad de Aprendizaje: Lenguaje de Programación

			<p>Actitudinales Desarrolla la solución de problemas computables de electricidad, electrónica o tecnología de materiales, con creatividad y pensamiento analítico.</p>
3	<p>Desarrolla sistemas computacionales básicos, creativos, innovadores y centrados en las necesidades de los usuarios, empleando módulos computacionales, estructuras de datos, un lenguaje de programación y pensamiento computacional para resolver problemas computables de mecatrónica, de acuerdo con los preceptos de sustentabilidad.</p>	<p>1. Realiza módulos computacionales para solucionar problemas computables de mecatrónica, empleando funciones, arreglos y un lenguaje de programación, con pensamiento creativo e innovador.</p>	<p>Conceptuales 1) Funciones 1.1) Prototipado, definición e invocación de funciones 1.2) Funciones definidas por el usuario 1.3) Tipos de variables 1.3.1) Locales 1.3.2) Globales 1.4) Paso por valor 2) Arreglos 2.1) Arreglos unidimensionales 2.2) Arreglos bidimensionales 2.3) Hipermatrices 2.4) Arreglos dinámicos</p> <p>Procedimentales Utiliza funciones, arreglos y un lenguaje de programación para realizar módulos computacionales orientados a solucionar problemas computables de mecatrónica.</p> <p>Actitudinales Aplica pensamiento creativo e innovador en la realización de módulos computacionales para solucionar problemas computables de mecatrónica.</p>
		<p>2. Elabora sistemas computacionales básicos, centrados en las necesidades de los usuarios, para resolver problemas computables de mecatrónica, utilizando estructuras de datos, un lenguaje de programación y pensamiento computacional, bajo el enfoque del desarrollo sustentable.</p>	<p>Conceptuales 1) Estructuras 1.1) Definición y declaración de una estructura 1.2) Creación de una estructura 1.3) Uso de estructuras 2) Implementación de bibliotecas o módulos del usuario del lenguaje de programación 3) Ecodiseño y modelo de desarrollo de <i>software</i> verde 4) Programación orientada a objetos 4.1) Definiciones básicas 4.1.1) Clase y objeto 4.1.2) Encapsulación</p>





Programa Académico: Técnico en Mecatrónica

Unidad de Aprendizaje: Lenguaje de Programación

				<p>4.1.3) Herencia 4.1.4) Polimorfismo 4.2) Programación de una clase</p> <p>Procedimentales Emplea estructuras de datos y un lenguaje de programación en la elaboración de sistemas computacionales básicos, enfocados a la resolución de problemas computables de mecatrónica.</p> <p>Actitudinales Efectúa sistemas computacionales básicos, centrados en las necesidades de los usuarios, aplicando pensamiento computacional, bajo el enfoque del desarrollo sustentable, para resolver problemas computables de mecatrónica.</p>	
BIBLIOGRAFÍA BÁSICA	AUTORES	AÑO	TÍTULO DEL DOCUMENTO	PAÍS	EDITORIAL/ISBN/DIRECCIÓN ELECTRÓNICA
Unidad de competencia 1	Joyanes, L.	2020	Fundamentos de Programación	España	Mc Graw Hill
	Trejos, O.	2018	Lógica de programación	Colombia	Ediciones de la U
Unidad de competencia 2	Ceballos, C.	2020	C/C++ Curso de programación	España	Grupo Editorial Ra-Ma
Unidad de competencia 3	Betancourt, J. y Polanco, I.	2021	115 ejercicios resueltos de programación C++	España	Grupo Editorial Ra-Ma





Programa Académico: Técnico en Mecatrónica

Unidad de Aprendizaje: Lenguaje de Programación

REFERENCIA BÁSICA Y COMPLEMENTARIA

Número y Nombre de la Unidad de Competencia	TIPO			CLASIFICACIÓN	
	LIBRO	REVISTA	OTRO	BÁSICO	CONSULTA
	Formato APA Autor, Año de Publicación, Título del Libro, Lugar de Publicación, Editorial.	Formato APA Autor, Fecha, Título del Artículo, Nombre de la Revista, Volumen, Páginas.	Especificar		
1. Utiliza la estructura secuencial de la programación estructurada y los elementos de un lenguaje de programación para solucionar problemas computables, aplicando razonamiento abstracto, creatividad y pensamiento lógico	Joyanes, L. (2020). <i>Fundamentos de programación</i> . España: Mc Graw Hill.			X	
	Ruiz, J. (2019). <i>Geometría analítica. Bachillerato</i> . México: Patria.				X
	Trejos, O.(2018). <i>Lógica de programación</i> . Colombia: Ediciones de la U.			X	
2. Realiza programas computacionales aplicando estructuras de control de flujo y elementos de un lenguaje de programación para solucionar problemas computables de electricidad, electrónica y tecnología de materiales, con creatividad y pensamiento analítico.	Cairó, O. (2018). <i>Fundamentos de programación: Piensa en C</i> . México. Pearson Educación				X
	Ceballos, C. (2020). <i>C/C++ Curso de programación</i> . España: Grupo Editorial Ra-Ma.			X	
	Hibbeler, C. (2017). <i>Mecánica de materiales</i> . México: Pearson Educación.				X
	Moro, M. (2018). <i>Fundamentos de Metrología Dimensional</i> . España: Alfaomega Grupo Editor.				
	Trashorras, J. (2018). <i>Electrotecnia</i> . España: Alfaomega Grupo Editor.				





Programa Académico: Técnico en Mecatrónica

Unidad de Aprendizaje: Lenguaje de Programación

3. Realiza programas computacionales aplicando estructuras de control de flujo y elementos de un lenguaje de programación para solucionar problemas computables de electricidad, electrónica y tecnología de materiales, con creatividad y pensamiento analítico.	Betancourt, J. y Polanco, I. (2021). <i>115 ejercicios resueltos de programación C++</i> . España: Grupo Editorial Ra-Ma.			X	
	Bolton, W. (2017). <i>Mecatrónica - Sistemas de control electrónico en la ingeniería mecánica y eléctrica</i> . México: Alfaomega Grupo Editor.				X
	Escamilla, A. (2018). <i>Metrología y sus aplicaciones</i> . México: Patria.				X
	Lospinoso, J. (2019). <i>C++ Crash Course: A fast-paced introduction</i> . EE.UU.: No Starch Press, Inc.				X
	Roldán, J. (2019). <i>Máquinas y Herramientas "Procesos de cálculo"</i> . Madrid, España: Paraninfo.				X





Programa Académico: Técnico en Mecatrónica

Unidad de Aprendizaje: Lenguaje de Programación

REFERENCIA BÁSICA Y COMPLEMENTARIA

Número y Nombre de la Unidad de Competencia	FORMATO APA	CLASIFICACIÓN	
	ARTÍCULO PÁGINA WEB: Autor, Año de publicación, Título del Artículo, Fecha de Recuperación del Documento, Asociación que publica el Artículo, URL. REVISTA, JOURNAL O PERIODICO EN BASE DE DATOS: Autor, Fecha, Título de la Revista o Periódico, Volumen, Páginas, Fecha que se obtuvo la Información de la Base de Datos, Nombre de la Base de Datos de la cual se obtuvo la Información, Fecha (año, mes, día) ARTICULO O CAPÍTULO DE LIBRO: Autor, Año de Publicación, Título del Libro, Lugar de Publicación, Editorial, Fecha en la que se obtuvo la Información, URL.	Básico	Consulta
1. Utiliza la estructura secuencial de la programación estructurada y los elementos de un lenguaje de programación para solucionar problemas computables, aplicando razonamiento abstracto, creatividad y pensamiento lógico.	Cogollo, J. (2020, septiembre 7). <i>¿Qué es un diagrama de flujo? ¿Cómo hacer un diagrama de flujo paso a paso?</i> [Archivo de video]. Recuperado de https://www.youtube.com/watch?v=Kucgc6NpGwc		X
	Khan Academy (2021). <i>Unidad: Geometría Analítica</i> . Recuperado de https://es.khanacademy.org/math/geometry/hs-geo-analytic-geometry		X
	Programación para todos (2020, julio 21). <i>Definición y Tipos. Introducción a los algoritmos</i> . [Archivo de video]. Recuperado de http://www.youtube.com/watch?v=akQtuSrr8jg		X
	W3SCOOL (2021). <i>C Tutorial - Learn C Programming</i> . Recuperado de https://www.w3schools.in/c-tutorial/	X	
2. Realiza programas computacionales aplicando estructuras de control de flujo y elementos de un lenguaje de programación para solucionar problemas computables de electricidad, electrónica y tecnología de materiales, con creatividad y pensamiento analítico.	Bejumea, V. y Roldán, M. (2017, octubre 23). <i>Fundamentos de programación con el lenguaje de programación C++</i> . España: Universidad de Málaga. Departamento de lenguajes y CC. Computación. E.T.S.I. Informática. Recuperado de http://www.lcc.uma.es/~vicente/docencia/cppdoc/programacion_cxx.pdf	X	
	Díaz del Castillo, F. (2010). Última edición. <i>Lecturas de Ingeniería. Metrología Dimensional</i> . Facultad de Estudios Superiores Cuautitlán Izcalli. Recuperado de http://olimpia.cuautitlan2.unam.mx/pagina_ingenieria/mecanica/mat/mat_mec/m2/METROLOGIA.pdf		X
	Monroy, D. (2020, mayo 13). <i>Clases de Ingeniería de Materiales. Clasificación de aceros</i> . [Archivo de video]. Recuperado de https://www.youtube.com/watch?v=DqJ7b4VXSik		X



Programa Académico: Técnico en Mecatrónica

Unidad de Aprendizaje: Lenguaje de Programación

3. Realiza programas computacionales aplicando estructuras de control de flujo y elementos de un lenguaje de programación para solucionar problemas computables de electricidad, electrónica y tecnología de materiales, con creatividad y pensamiento analítico.	C++ reference (2021, enero 29). <i>C++ language</i> . Recuperado de https://en.cppreference.com/w/cpp/language	X	
	Fismat, M. (2021, febrero 20). <i>Cómo calcular el error absoluto, relativo y porcentual.</i> . [Archivo de video]. Recuperado de https://www.youtube.com/watch?v=ml5D-Qm9BnQ		X
	MECANICOSESTECA (2020, mayo 17). <i>¿Que broca necesito para machuelar?</i> [Archivo de video]. Recuperado de https://www.youtube.com/watch?v=9IZDsWah6zM		X

