

Planeación didáctica por propósito formativo					
Nombre de la escuela:	PREPARATORIA OFICIAL NÚMERO 028		Nombre de la Asignatura:	Programación	
Semestre:	QUINTO	TURNO: VESPERTINO	Grupo(s):	3	PERIODO: 15 de octubre al 27 de noviembre de 2025
MTRA. SUSANA DURAND CRUZ					
<b>MISIÓN:</b> Formar estudiantes responsables y fortalecidos en su desarrollo integral, así como en valores para enfrentar las exigencias de la vida cotidiana y académica con una perspectiva ambientalmente responsable		<b>VISIÓN:</b> Ser reconocida como una institución de vanguardia, formadora de bachilleres académicamente íntegros acordes a las necesidades de la sociedad actual; mejorando cada ciclo escolar infraestructura, eficiente, suficiente, digna y segura con una gestión de inclusión, equidad, cooperación y colaboración.		<b>VALORES:</b> Respeto, honestidad, responsabilidad y solidaridad para generar una cultura de paz, convivencia armónica y cuidado del ambiente.	
<b>Descripción del Diagnóstico</b>					
Se realizó un diagnóstico para conocer los conocimientos previos de los estudiantes, mediante un formulario que se le realizó a todos los estudiantes de la academia de tercero. Asimismo, durante la semana de diagnóstico, se logró llevar a cabo diversas dinámicas de integración, los estudiantes identificaron los contenidos de su primer parcial.					
<b>Meta de Aprendizaje</b>					
Plantea soluciones críticas y responsables mediante la metodología de desarrollo de software para demostrar eficiencia en el manejo de base de datos y software de programación de alto nivel, que sean aplicables a necesidades de una empresa o institución para el tratamiento de información.					
<b>Propósito Formativo del Diagnóstico:</b> Identificar los conocimientos previos, habilidades y actitudes de los estudiantes en relación con conceptos básicos de programación estructurada, para establecer un punto de partida que oriente la planeación didáctica y el acompañamiento personalizado durante el curso. <b>Contenido Formativo del Diagnóstico:</b> Conceptos básicos de programación: algoritmos, pseudocódigo, diagramas de flujo y estructuras de control (secuencias, decisiones y ciclos). <b>Actividad(es) de Aprendizaje para el logro del Contenido Formativo del Diagnóstico Aplicado:</b> "Desbloqueando mi lógica de programación" con la consigna la tecnología nos une o nos separa de las personas.					
<b>Propósito Formativo: 1</b>			<b>Contenidos Formativos:</b>		
Que el estudiante comprenda la utilidad del pseudocódigo como herramienta de representación algorítmica y sea capaz de identificar y aplicar su estructura básica, para resolver problemas sencillos de forma lógica antes de traducirlos a un lenguaje de programación formal.			<b>1A</b> El pseudocódigo es una herramienta que permite representar un algoritmo de forma clara y ordenada, utilizando un lenguaje cercano al natural pero con estructura lógica. Aunque no puede ser ejecutado por la computadora, sirve como base para luego escribir el código en un lenguaje de programación. <b>2B</b> Un pseudocódigo debe contener las siguientes partes: <ul style="list-style-type: none"> <li>• Inicio y Fin</li> <li>• Declaración de variables</li> </ul>		



	<ul style="list-style-type: none"> <li>Entrada de datos (Leer)</li> <li>Procesamiento (operaciones)</li> </ul> Salida de datos (Escribir) También puede contener estructuras de control como condicionales (Si...Entonces) y ciclos (Mientras...Hacer).
Actividades de aprendizaje:	Actividad de aprendizaje <b>1A-1</b> . ¿Pseudocódigo o código real? Actividad de aprendizaje <b>1B-1</b> . Escribe tu primer pseudocódigo
Objetivo de las actividades de aprendizaje:	Objetivo de la Actividad Didáctica <b>1A</b> . El estudiante identificará las características del pseudocódigo y su utilidad en la planeación de soluciones algorítmicas, mediante la lectura, análisis y comparación de ejemplos. Objetivo de la Actividad Didáctica <b>1B</b> . El estudiante aplicará la estructura del pseudocódigo en la solución de problemas básicos, representando claramente el flujo de entrada, proceso y salida.

**Desarrollo de las Actividades Didácticas (aprendizaje, enseñanza y evaluación)**

Actividades de Enseñanza y Aprendizaje	Instrumento(s) de evaluación	Recursos didácticos	Responsable	Escenario	Duración
<b>Actividad 1A-1.</b> El docente presenta dos ejemplos: uno de pseudocódigo y otro en lenguaje Python. Los estudiantes deben identificar cuál es pseudocódigo, explicar sus características y escribir su propia versión de pseudocódigo para un problema sencillo, como calcular el área de un triángulo.	Identificación de pseudocódigo.	Investigación documentada.	Docente Estudiantes.	Aula Aula de cómputo.	50 min.
<b>Actividad 1B-1.</b> Los estudiantes escriben el pseudocódigo de un programa que reciba dos números y diga cuál es mayor. Deberán utilizar: <ul style="list-style-type: none"> <li>Instrucciones de entrada (Leer)</li> <li>Condicional (Si...Entonces...Sino)</li> </ul> Instrucción de salida (Escribir) Al finalizar, compartirán su pseudocódigo con un compañero para revisarlo en pares.	Identificación de pseudocódigo.	Investigación documentada.	Docente Estudiantes.	Aula Aula de cómputo	50 min.

<b>Propósito Formativo: 2</b>	<b>Contenidos Formativos:</b>
-------------------------------	-------------------------------

Que el estudiante identifique y aplique estructuras de decisión en un lenguaje de programación, comprendiendo su importancia para controlar el flujo lógico de un programa en la resolución de problemas cotidianos.	<b>2A</b> Estructuras de decisión en programación: Condicionales simples (if), dobles (if-else), múltiples (if-elif-else en Python o switch en C++), y su aplicación para ejecutar instrucciones dependiendo del valor de una condición lógica. <b>2B</b> Estructuras repetitivas o ciclos en programación: Uso de estructuras como for, while y do-while para ejecutar instrucciones de forma
--	---



	repetitiva hasta que se cumpla una condición lógica.
Actividades de aprendizaje:	Actividad de aprendizaje 2A-1. ¿Mayor o menor?" Actividad de aprendizaje 2B-1. Entendiendo el ciclo
Objetivo de las actividades de aprendizaje:	Objetivo de la Actividad Didáctica 2A. El estudiante aplicará estructuras condicionales para resolver un problema lógico, identificando el uso correcto de la sintaxis y el flujo de decisiones en un programa. Objetivo de la Actividad Didáctica 2B. El estudiante identificará la estructura y funcionamiento de un ciclo por al desarrollar un programa que automatice una secuencia numérica, aplicando lógica de programación para resolver un problema sencillo.

Desarrollo de las Actividades Didácticas (aprendizaje, enseñanza y evaluación)					
Actividades de Enseñanza y Aprendizaje	Instrumento(s) de evaluación	Recursos didácticos	Responsable	Escenario	Duración
<p><b>Actividad 2A-1. Lluvia de ideas</b></p> <ul style="list-style-type: none"> <li>El docente pregunta: "¿Qué hacemos cuando tomamos decisiones en la vida diaria?" "¿Qué tipo de preguntas nos llevan a tomar una u otra acción?"</li> <li>Se escriben en el pizarrón ejemplos como:                             <ul style="list-style-type: none"> <li>Si está lloviendo, llevo paraguas.</li> <li>Si saqué más de 6, apruebo.</li> </ul> </li> <li>Luego, el docente introduce la idea: "En programación también tomamos decisiones usando estructuras condicionales."</li> <li>explica con apoyo visual:                             <ul style="list-style-type: none"> <li>¿Qué es una estructura condicional (if, else)?</li> <li>¿Cómo funciona la evaluación lógica de condiciones?</li> <li>Ejemplo en pseudocódigo y lenguaje de programación (Python o C++):</li> </ul> </li> </ul> <p>Inicio Leer num1, num2 Si num1 &gt; num2 Entonces</p>	<p>Captura de pantalla del código y el resultado. Una explicación corta en su libreta:</p>	<p>Computadora con acceso a editor de código (como Replit, Thonny o similar)</p>	<p>Docente Estudiantes</p>	<p>Aula Aula de cómputo.</p>	<p>50 min.</p>



<p><b>Escribir "El primer número es mayor"</b>  <b>Sino Si num1 &lt; num2 Entonces</b>  <b>Escribir "El segundo número es mayor"</b>  <b>Sino</b>  <b>Escribir "Ambos números son iguales"</b>  <b>Fin</b></p> <ul style="list-style-type: none"> <li>El docente muestra un ejemplo en el editor de código y ejecuta un programa que compara edades o calificaciones.</li> </ul> <p><b>Actividad 2B-1.</b> El docente explica que los ciclos sirven para repetir acciones sin escribir muchas veces la misma instrucción. Se presenta un ejemplo de código sencillo:  <b>for i in range(1, 11):</b>  <b>print(i)</b>  <b>Los estudiantes deben:</b></p> <ul style="list-style-type: none"> <li>Escribir este mismo código en su editor (puede ser en línea como Replit).</li> <li>Ejecutarlo y observar cómo el ciclo imprime los números del 1 al 10 automáticamente.</li> <li>Cambiar el rango para imprimir del 1 al 5, y luego del 1 al 20 como práctica.</li> </ul>	<p>Captura de pantalla del código y el resultado.                  Una explicación corta en su libreta:</p>	<p>Computadora con acceso a editor de código (como Replit, Thonny o similar)</p>	<p>Docente                  Estudiantes</p>	<p>Aula                  Aula de cómputo</p>	<p>50 min.</p>
<p><b>Propósito Formativo: 3</b></p>	<p><b>Contenidos Formativos:</b></p>				
<p>Que el estudiante <b>comprenda los tipos y metodologías de los lenguajes de programación</b>, identificando sus características, ventajas y aplicaciones, para que sea capaz de <b>distinguir entre programación estructurada y orientada a objetos</b>, aplicando principios básicos en ejemplos sencillos de código.</p>	<p><b>Tipos de lenguajes de programación:</b> Lenguajes de bajo y alto nivel, compilados e interpretados, ejemplos comunes.  <b>Metodología de programación:</b> Enfoque estructurado vs. orientado a objetos, análisis del problema, diseño, codificación y prueba.</p>				
<p><b>Actividades de aprendizaje:</b></p>	<p>Actividad de aprendizaje <b>3A-1.</b> Mapa conceptual comparativo: Los alumnos elaborarán un mapa conceptual que relacione los tipos de lenguajes y sus principales características (nivel, paradigma, ejemplos, ventajas).                  Actividad de aprendizaje <b>3B-1</b> Mini proyecto de análisis y diseño: En equipos, los estudiantes desarrollarán un pequeño caso práctico (por ejemplo, un sistema de registro de alumnos) y deberán comparar cómo se resolvería con programación estructurada y con programación orientada a objetos.</p>				



Objetivo de las actividades de aprendizaje:	Que el estudiante clasifique y analice los distintos tipos de lenguajes de programación según su nivel y propósito. Que el estudiante aplique la metodología de programación estructurada y orientada a objetos para resolver un problema simple.
---	--

**Desarrollo de las Actividades Didácticas (aprendizaje, enseñanza y evaluación)**

Actividades de Enseñanza y Aprendizaje	Instrumento(s) de evaluación	Recursos didácticos	Responsable	Escenario	Duración
<p><b>Actividad 3A-1. Elaboración de mapa conceptual sobre los tipos de lenguajes de programación. Los alumnos crearán un mapa conceptual que muestre la clasificación de los lenguajes, sus características principales, ejemplos representativos y ventajas de uso en distintos contextos.</b></p>	Lista de cotejo del mapa conceptual.	Investigación documentada.	Docente Estudiantes.	Aula Aula de cómputo.	50 min.
<p><b>Actividad 3B-1. Mini proyecto de análisis y diseño comparativo.</b></p> <p>En equipos, los estudiantes desarrollarán un caso práctico (por ejemplo, un sistema de registro de alumnos o una calculadora básica) y compararán cómo se resolvería mediante programación estructurada y programación orientada a objetos. Presentarán sus conclusiones en forma de pseudocódigo o diagrama de flujo.</p> <p>El docente inicia la clase con una breve exposición sobre la evolución de los lenguajes de programación, apoyándose en una línea del tiempo proyectada o infografía. Explica la diferencia entre lenguajes de bajo y alto nivel, mostrando ejemplos en código (ensamblador, C, Python, Java).</p> <p>Guía conceptual: Explica los conceptos de paradigma estructurado y orientado a objetos, usando ejemplos visuales de diagramas y fragmentos de código en ambos estilos. Resalta cómo cambian la organización y la lógica entre uno y otro enfoque.</p> <p>Demostración práctica: Muestra en vivo (en</p>		Investigación documentada.	Docente Estudiantes.	Aula Aula de cómputo	50 min.





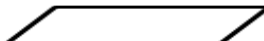
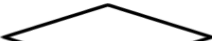

<p>computadora o proyector) cómo un mismo problema sencillo se aborda de forma distinta con cada paradigma. Comenta ventajas y limitaciones de cada metodología.</p> <p>Acompañamiento: Durante la elaboración del mapa conceptual y el proyecto comparativo, resuelve dudas, orienta el trabajo en equipo y fomenta la reflexión crítica sobre cuál paradigma se adapta mejor a distintos contextos.</p> <p>Cierre reflexivo: Conduce una puesta en común donde los equipos explican sus resultados y el grupo analiza qué metodología resulta más eficiente según el problema.</p>					
--	--	--	--	--	--

**ESCALA DE EVALUACIÓN DEL PARCIAL:**

- >Examen Escrito: 30%
- >Entrega de actividades: 20%
- >Investigación Documental u otros: 30%
- >Proyecto de academia: 20%
- Total: 100%

**REFERENCIAS BIBLIOGRÁFICAS Y ELECTRÓNICAS**

**ANEXO: Links o Antología de Ejercicios, lecturas, diagramas, resúmenes, etc. para el trabajo complementario del alumno o docente:**

<b>Símbolo</b>	<b>Significado/Función</b>
	<b>Inicio/Fin/Conector</b>
	<b>Proceso</b>
	<b>Entrada/Salida</b>
	<b>Decisión</b>
	<b>Flujo de ejecución</b>





"2025. Bicentenario de la vida municipal en el Estado de México".

### Instrumentos de evaluación

Criterio	Indicador	Instrumento
Comprensión de los tipos de lenguajes	Identifica correctamente niveles, paradigmas y ejemplos de lenguajes.	Lista de cotejo del mapa conceptual.
Aplicación de metodologías	Compara adecuadamente las metodologías estructurada y orientada a objetos en su caso práctico.	Rúbrica del proyecto comparativo.
Trabajo colaborativo	Participa activamente, cumple tareas asignadas y respeta opiniones.	Observación directa / autoevaluación.

### Rúbrica breve de evaluación del proyecto final

Criterio	Excelente (3)	Satisfactorio (2)	En proceso (1)
Comprensión del tema	Explica claramente las diferencias entre los tipos de programación.	Muestra comprensión parcial.	Confunde los conceptos.
Aplicación práctica	Propone soluciones coherentes y bien justificadas.	Soluciones parcialmente coherentes.	No logra aplicar los conceptos.
Presentación y trabajo en equipo	Trabajo claro, ordenado y colaborativo.	Trabajo con pequeños errores.	Trabajo desorganizado o sin colaboración.

Elaboró
Susana Durand Cruz
Nombre del (a) docente que elabora la planeación (Redactar)

Revisó
Subdirector Mtro. Adrián Andrade Almanza

