

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: 1 Septiembre al 14 de Octubre de 2025
MTRA. SUSANA DURAND CRUZ				
MISIÓN: 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	VISIÓN: 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.		VALORES: Respeto, honestidad, responsabilidad y solidaridad para generar una cultura de paz, convivencia armónica y cuidado del ambiente.	
Descripción del Diagnóstico				
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.				
Meta de Aprendizaje				
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.				
Propósito Formativo del Diagnóstico: 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. Contenido Formativo del Diagnóstico: Conceptos básicos de programación: algoritmos, pseudocódigo, diagramas de flujo y estructuras de control (secuencias, decisiones y ciclos). Actividad(es) de Aprendizaje para el logro del Contenido Formativo del Diagnóstico Aplicado: "Desbloqueando mi lógica de programación" con la consigna la tecnología nos une o nos separa de las personas.				
Propósito Formativo: 1		Contenidos Formativos:		
Plantea el uso de diagramas de flujo y algoritmos, fomentando su desarrollo creativo, para solucionar problemas cotidianos de su contexto.		1A Lógica de programación. 1B Aplica estructuras de control con un lenguaje de programación 1C Reconoce los algoritmos.		
Actividades de aprendizaje:		Actividad de aprendizaje 1A-1 . Describiendo la estructura general de un programa Actividad de aprendizaje 1A-2 . Diagramas de flujo Actividad de aprendizaje 1B-1 . Algoritmos de la vida diaria		



Objetivo de las actividades de aprendizaje:	<p>Objetivo de la Actividad Didáctica 1A. El estudiante identificará y aplicará correctamente las partes fundamentales de un programa estructurado (inicio, declaración de variables, entrada, proceso, salida y fin)</p> <p>Objetivo de la Actividad Didáctica 1B.</p> <p>Objetivo de la Actividad Didáctica 1C. El estudiante identificará las características de un algoritmo, sus componentes (inicio, proceso, fin) y su utilidad en la solución de problemas cotidianos, mediante el análisis y elaboración de ejemplos simples.</p>
---	---

Desarrollo de las Actividades Didácticas (aprendizaje, enseñanza y evaluación)					
Actividades de Enseñanza y Aprendizaje	Instrumento(s) de evaluación	Recursos didácticos	Responsable	Escenario	Duración
<p>Actividad 1A-1. El estudiante contestará las siguientes preguntas en su libreta a. ¿Qué un programa? b. ¿Qué es la programación? c. ¿Qué es un Pseudocódigo? d. ¿Qué entiendes por estructura?</p> <p>Los estudiantes se apoyan en la siguiente información: Inicialmente el término Programa sirve para denotar aquella agrupación de actividades que tanto en secuencia o simultáneas son ejecutadas por un equipo de individuos a fin de que se cumpla un objetivo. Un programa informático es una serie de comandos ejecutados por el equipo. Sin embargo, el equipo solo es capaz de procesar elementos binarios, es decir, una serie de 0 y 1. Por lo tanto, necesitamos un lenguaje de programación para escribir de manera legible, es decir, con comandos que el ser humano pueda comprender (por ser similares a su propio lenguaje) los comandos que el equipo deberá ejecutar. Estos programas se traducen después a un lenguaje máquina (en binario) a través de un compilador. La programación informática es el arte del proceso por el cual se limpia, codifica, traza y protege el código fuente de programas computacionales, en otras palabras, es indicarle a la computadora lo que tiene que hacer. La programación se guía por una serie de normas y un conjunto de órdenes, instrucciones y expresiones que tienden a ser semejantes a una lengua natural acotada. Por lo cual recibe el nombre de lenguaje de programación.</p>	Lista de cotejo para el cuestionario.	Lecturas	Docentes. Estudiantes	Aula Aula de cómputo	50 min.



<p>Pseudocódigo (o falso Lenguaje). Es comúnmente utilizado por los programadores para omitir secciones de Código o para dar una explicación del paradigma que tomó el mismo programador para hacer sus códigos, esto quiere decir que el pseudocódigo no es programable sino facilita la programación. El principal objetivo del pseudocódigo es el de representar la solución a un algoritmo de la forma más detallada posible, y a su vez lo más parecida posible al lenguaje que posteriormente se utilizará para la codificación del mismo. La programación estructurada es un paradigma de programación basado en utilizar funciones o subrutinas, y únicamente tres estructuras de control: Secuencia: ejecución de una sentencia tras otra. Selección o condicional: ejecución de una sentencia o conjunto de sentencias, según el valor de una variable booleana. Iteración (ciclo o bucle): ejecución de una sentencia o conjunto de sentencias, mientras una variable booleana sea verdadera. Este paradigma se fundamenta en el teorema correspondiente, que establece que toda función computable puede ser implementada en un lenguaje de programación que combine sólo estas tres estructuras lógicas o de control.</p> <p>Actividad 1B. Los estudiantes contestan la siguiente pregunta ¿Cuál es la estructura general de un programa? con los datos obtenidos, realizar una pirámide de información en el cuaderno. Puedes apoyarte en la siguiente información: Un programa puede considerarse como una secuencia de acciones (instrucciones) que manipulan un conjunto de objetos (datos). Bloques de un programa: • Bloque de declaraciones: en él se especifican todos los objetos que utiliza el programa (constantes, variables, tablas, registros, archivos, etc.). • Bloque de instrucciones: constituido por el conjunto de operaciones que se han de realizar para la obtención de los resultados deseados. Partes principales de un programa:</p>	<p>Pirámide de información.</p>	<p>Investigación documental.</p>	<p>Docentes. Estudiantes</p>	<p>Aula Aula de cómputo</p>	<p>50 min.</p>
---	---------------------------------	----------------------------------	----------------------------------	---------------------------------	----------------



<p>Dentro del bloque de instrucciones de un programa se pueden diferenciar tres partes fundamentales. En algunos casos, estas tres partes están perfectamente delimitadas, pero en la mayoría sus instrucciones quedan entremezcladas a lo largo del programa, si bien mantienen una cierta localización geométrica impuesta por la propia naturaleza de las mismas.</p> <ul style="list-style-type: none"> • Entrada de datos: la constituyen todas aquellas instrucciones que toman datos de un dispositivo externo, almacenándolos en la memoria central para que puedan ser procesados. • Proceso o algoritmo: está formado por las instrucciones que modifican los objetos a partir de su estado inicial hasta el estado final, dejando éstos disponibles en la memoria central. • Salida de resultados: conjunto de instrucciones que toman los datos finales de la memoria central y los envían a los dispositivos externos. <p>Actividad 1C. Investigar ¿Qué es el lenguaje C? ¿Quién fue su desarrollador? Identificar 3 ventajas y 3 desventajas del mismo. Los estudiantes realizan un mapa conceptual con lo obtenido. Puedes apoyarte en la siguiente información: Lenguaje de programación C. También conocido como "Lenguaje de programación de sistemas" desarrollado en el año 1972 por Dennis Ritchie para UNIX un sistema operativo multiplataforma. El lenguaje C es del tipo lenguaje estructurado como son Pascal, Fortran, Basic. Sus instrucciones son muy parecidas a otros lenguajes incluyendo sentencias como if, else, for, do y while... . Aunque C es un lenguaje de alto nivel (puesto que es estructurado y posee sentencias y funciones que simplifican su funcionamiento) tenemos la posibilidad de programar a bajo nivel (como en el Assembler tocando los registros, memoria etc.). Para simplificar el funcionamiento del lenguaje C tiene incluidas librerías de funciones que pueden ser incluidas haciendo referencia la librería que las incluye, es decir que si queremos usar una función</p>	<p>Mapa conceptual</p>	<p>Investigación documental.</p>	<p>Docentes. Estudiantes</p>	<p>Aula Aula de cómputo</p>	<p>50 min.</p>
--	------------------------	----------------------------------	----------------------------------	--------------------------------------	----------------



<p>para borrar la pantalla tendremos que incluir en nuestro programa la librería que tiene la función para borrar la pantalla. La programación en C tiene una gran facilidad para escribir código compacto y sencillo a su misma vez. En el lenguaje C no tenemos procedimientos como en otros lenguajes solamente tenemos funciones los procedimientos los simula y esta terminante mente prohibido escribir funciones, procedimientos y los comandos en mayúscula todo se escribe en minúsculas Desventajas del lenguaje C: No es un lenguaje visual, no puede ser deducido de forma intuitiva, como por ejemplo el Visual Basic. Para el uso de funciones anidadas necesita de extensiones. No tiene instrucciones de entrada y salida, ni para el manejo de cadenas de caracteres.</p>					
<p>Propósito Formativo: 2</p>	<p>Contenidos Formativos:</p>				
<p>Enuncia de manera secuencial los elementos para la resolución de problemas mediante algoritmo, y diagramas de flujo.</p>	<p>2A Algoritmo, Datos, Variable, Constante, Sentencia, Compilar, Ejecutar, Iteración, Ciclo y Condición 2B Redactar Tipos de lenguaje.</p>				
<p>Actividades de aprendizaje:</p>	<p>Actividad de aprendizaje 2A-1. Elabora un glosario con el lenguaje de programación. Actividad de aprendizaje 2B-1. Elabora una tabla comparativa de los dos lenguajes de programación.</p>				
<p>Objetivo de las actividades de aprendizaje:</p>	<p>Objetivo de la Actividad Didáctica 2A. Identifica los elementos que integran la programación en un lenguaje de alto nivel. Objetivo de la Actividad Didáctica 2B. Emplea y estructura propuesta de solución empleando lenguaje de programación de alto nivel.</p>				

Desarrollo de las Actividades Didácticas (aprendizaje, enseñanza y evaluación)					
Actividades de Enseñanza y Aprendizaje	Instrumento(s) de evaluación	Recursos didácticos	Responsable	Escenario	Duración



<p>Actividad 2A-1. Los estudiantes realizan una búsqueda del significado de las siguientes palabras: Algoritmo, Datos, Variable, Constante, Sentencia, Compilar, Ejecutar, Iteración, Ciclo y Condición finalmente realizar un glosario.</p> <p>El estudiante se apoya en la siguiente información: En informática, un algoritmo es una secuencia de instrucciones secuenciales, gracias al cual pueden llevarse a cabo ciertos procesos y darse respuesta a determinadas necesidades o decisiones. Se trata de conjuntos ordenados y finitos de pasos, que nos permiten resolver un problema o tomar una decisión. Una instrucción condicional nos permite plantear la solución a un problema considerando los distintos casos que se pueden presentar. De esta manera, podemos utilizar un algoritmo distinto para enfrentar cada caso que pueda existir en el mundo. En programación se denomina bucle a la ejecución repetidas veces de un mismo conjunto de sentencias. Normalmente en cada nueva ejecución varía algún elemento. En programación, un tipo de dato informático o simplemente tipo es un atributo de los datos que indica al ordenador (y/o al programador) sobre la clase de datos que se va a trabajar. Esto incluye imponer restricciones en los datos, como qué valores pueden tomar y qué operaciones se pueden realizar. Los tipos de datos comunes son: números enteros, números con signo (negativos), números de coma flotante (decimales), cadenas alfanuméricas, estados (booleano), etc. En programación, las variables son espacios reservados en la memoria que, como su nombre indica, pueden cambiar de contenido a lo largo de la ejecución de un programa. Una variable corresponde a un área reservada en la memoria principal del ordenador. En programación, una constante es un valor que no puede ser alterado durante la ejecución de un programa. Una constante corresponde a una longitud fija de un área reservada en la memoria principal del ordenador, donde</p>	<p>Glosario</p>	<p>Investigación documentada.</p>	<p>Docente</p> <p>Estudiantes</p>	<p>Aula</p> <p>Aula de cómputo</p>	<p>50 min.</p>
--	-----------------	-----------------------------------	-----------------------------------	------------------------------------	----------------



<p>el programa almacena valores fijos. Las sentencias son los elementos básicos en los que se divide el código en un lenguaje de programación. Al fin y al cabo, un programa no es más que un conjunto de sentencias que se ejecutan para realizar una cierta tarea.</p> <p>El programa escrito en un lenguaje de programación (fácilmente comprensible por el programador) es llamado programa fuente y no se puede ejecutar directamente en una computadora. La opción más común es compilar el programa obteniendo un módulo objeto, aunque también puede ejecutarse en forma más directa a través de un intérprete informático. El código fuente del programa se debe someter a un proceso de traducción para convertirlo a lenguaje máquina o bien a un código intermedio, generando así un módulo denominado "objeto". A este proceso se le llama compilación. En informática, ejecutar es la acción de iniciar la carga de un programa o de cualquier archivo ejecutable. En otras palabras, la ejecución es el proceso mediante el cual una computadora lleva a cabo las instrucciones de un programa informático. Para la programación, por su parte, la iteración consiste en reiterar un conjunto de instrucciones o acciones con uno o varios objetivos. Para citar un ejemplo, muchas páginas web están preparadas para adaptarse a cambios en su estructura, tales como alteraciones estéticas o del número de secciones accesibles, cuyos enlaces se muestran en forma de pestañas</p> <p>Actividad 2B-1. Los estudiantes realizan la siguiente investigación: ¿cuál es la función principal del lenguaje C y C++? ¿cuáles son sus costos? ¿en qué sistema operativo se pueden instalar? ¿en qué año salieron al mercado? ¿Quién fue su creador? para finalizar elaborar una tabla comparativa de los 2 lenguajes.</p>	<p>Tabla comparativa de los dos lenguajes.</p>	<p>Investigación documentada.</p>	<p>Docente Estudiantes</p>	<p>Aula Aula de cómputo</p>	<p>50 min.</p>
---	--	-----------------------------------	-------------------------------------	--------------------------------------	----------------





<p>El estudiante se apoya en la siguiente información: El lenguaje C, está orientado a la programación estructurada. ¿En qué consiste la programación estructurada? Pues, básicamente, en trabajar con código secuencial, con un conjunto de sentencias o instrucciones que se ejecutan una por una. Las podemos clasificar en:</p> <ul style="list-style-type: none">• Instrucciones condicionales.• Instrucciones de iteración (bucle de instrucciones). <p>El concepto de estructurada viene de trabajar con funciones. En cambio, C++ también está orientado a la Programación POO (Programación orientada a Objetos). Esta es la diferencia más grande entre los dos idiomas. El lenguaje C es de código Abierto u Open Source se refiere al código fuente del software que es abiertamente accesible y que puede ser cambiado y distribuido por cualquier persona. No se requiere ninguna licencia adicional en ningún momento. C ha tenido distintos usos a lo largo de la historia, con aplicaciones en sistemas operativos, compiladores y desarrollo de software. El lenguaje C puede ser utilizado en diferentes sistemas entre los principales son los siguientes: Windows, MacOS, Linux, Unix. El lenguaje C tiene otros lenguajes que se consideran sus antecesores (BCPL, B) y comenzó a utilizarse en los años 70. Su fecha de "nacimiento" como lenguaje de uso extendido suele decirse que es 1978 cuando Brian Kernighan y Dennis Ritchie publicaron el libro The C Programming Language, popularmente denominado "La Biblia de C". En este libro se definía de forma clara y precisa este lenguaje de programación. El 14 de octubre del año 1985 salió publicada la primera guía de referencia de C++, por lo que es considerada como la fecha de "nacimiento" de este lenguaje de programación. El C++ fue diseñado por Bjarne Stroustrup en el año 1980 (en los míticos Laboratorios Bell) como una extensión del</p>					
---	--	--	--	--	--



<p>lenguaje de programación C, diseñado para ser un "lenguaje de uso general": puede correr sobre cualquier plataforma, y está en todos lados, sobre todo en videojuegos y sistemas integrados. Este éxito llevó a que en el año 1990 se reunieran las organizaciones ANSI e ISO con el objeto de definir un estándar que formalizara al lenguaje, proceso que culminó en el año 1998 cuando salió aprobado ANSI C++</p> <p>ACTIVIDAD 3-1. El estudiante elabora un concepto propio con base a la información proporcionada. Se apoya en la siguiente información:</p> <p>Las palabras reservadas en programación, o palabras clave, tienen un significado especial para el compilador de cualquier lenguaje de programación, estas palabras pueden identificar los tipos de datos que se pueden usar, además de las diferentes rutinas de programación que permite cada lenguaje.</p>	<p>Concepto personal</p>	<p>Investigación documentada.</p>	<p>Docente Estudiantes</p>	<p>Aula Aula de cómputo</p>	<p>50 min.</p>
<p>Propósito Formativo: 3</p>	<p>Contenidos Formativos:</p>				
<p>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.</p>	<p>3A 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.</p> <p>3B Un pseudocódigo debe contener las siguientes partes:</p> <ul style="list-style-type: none"> • Inicio y Fin • Declaración de variables • Entrada de datos (Leer) • Procesamiento (operaciones) • Salida de datos (Escribir) <p>También puede contener estructuras de control como condicionales (Si...Entonces) y ciclos (Mientras...Hacer).</p>				
<p>Actividades de aprendizaje:</p>	<p>Actividad de aprendizaje 3A-1. ¿Pseudocódigo o código real?</p> <p>Actividad de aprendizaje 3B-1. Escribe tu primer pseudocódigo</p>				
<p>Objetivo de las actividades de aprendizaje:</p>	<p>Objetivo de la Actividad Didáctica 3A. El estudiante identificará las características del pseudocódigo y su utilidad en la planeación de soluciones algorítmicas, mediante la lectura, análisis y comparación de ejemplos.</p> <p>Objetivo de la Actividad Didáctica 3B. El estudiante aplicará la estructura del pseudocódigo</p>				



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
<p>Actividad 3A-1. 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.</p>	Identificación de pseudocódigo.	Investigación documentada.	Docente Estudiantes.	Aula Aula de cómputo.	50 min.
<p>Actividad 3B-1. Los estudiantes escriben el pseudocódigo de un programa que reciba dos números y diga cuál es mayor. Deberán utilizar:</p> <ul style="list-style-type: none"> Instrucciones de entrada (Leer) Condicionales (Si...Entonces...Sino) Instrucción de salida (Escribir) <p>Al finalizar, compartirán su pseudocódigo con un compañero para revisarlo en pares.</p>	Identificación de pseudocódigo.	Investigación documentada.	Docente Estudiantes.	Aula Aula de cómputo	50 min.
Propósito Formativo: 4	Contenidos Formativos:				
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.	<p>4A 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.</p> <p>4B 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.</p>				
Actividades de aprendizaje:	<p>Actividad de aprendizaje 4A-1. ¿Mayor o menor?"</p> <p>Actividad de aprendizaje 4B-1. Entendiendo el ciclo</p>				
Objetivo de las actividades de aprendizaje:	<p>Objetivo de la Actividad Didáctica 4A. 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.</p> <p>Objetivo de la Actividad Didáctica 4B. El estudiante identificará la estructura y funcionamiento de un ciclo for al desarrollar un programa que automatice una secuencia numérica, aplicando lógica de programación para resolver un problema sencillo.</p>				



<p>Actividad 4B-1. El docente explica que los ciclos sirven para repetir acciones sin escribir muchas veces la misma instrucción. Se presenta un ejemplo de código sencillo:</p> <pre>for i in range(1, 11): print(i)</pre> <p>Los estudiantes deben:</p> <ul style="list-style-type: none"> ○ Escribir este mismo código en su editor (puede ser en línea como Replit). ○ Ejecutarlo y observar cómo el ciclo imprime los números del 1 al 10 automáticamente. ○ Cambiar el rango para imprimir del 1 al 5, y luego del 1 al 20 como práctica. 					
---	--	--	--	--	--

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:

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

Revisó
Subdirector
Mtro. Adrián Andrade Almanza

