



FUNDAMENTOS DE LA PROGRAMACIÓN

- 1. Definición de programación:** La programación se refiere a la acción de crear programas o aplicaciones a través del desarrollo de un código fuente.

La programación es el proceso de crear instrucciones que una computadora debe ejecutar.



Programas y aplicaciones



Código fuente

```
/* Suma de n números */  
#include <stdio.h>  
int main() {  
    int num=0, suma=0;  
  
    do {  
        suma=suma+num;  
        printf("un número: ");  
        scanf("%d", &num);  
    } while(num>=0);  
    printf("suma es: %d", suma);  
    return 0;  
}
```

Conjunto de instrucciones escritas en un lenguaje de programación

- 2. Definición de algoritmo:** Un algoritmo es un conjunto de pasos o instrucciones lógicas, ordenadas y finitas que se utilizan para resolver un problema o realizar una tarea.

Por ejemplo podemos compararla con una receta de cocina, donde cada paso o instrucción te lleva a un resultado final. Los algoritmos son la base de la programación y se utilizan en una amplia variedad de aplicaciones, desde las más sencillas hasta las más complejas



Los algoritmos están en todas partes



Están presentes en numerosas actividades cotidianas

Algunos ejemplos de su aplicación en la actualidad:

- **En la vida cotidiana:**

Preparar una receta de cocina: La serie de pasos a seguir para cocinar un plato es un algoritmo.

Buscar una dirección en un mapa: La aplicación utiliza un algoritmo para encontrar la ruta más corta o rápida.

Comprar en línea: Los sitios web de comercio electrónico utilizan algoritmos para recomendar productos, procesar pagos y calcular costos de envío.

Usar redes sociales: Las plataformas utilizan algoritmos para mostrar publicaciones relevantes, recomendar amigos y personalizar la experiencia del usuario.

- **En áreas más complejas:**

Medicina: Los algoritmos se utilizan para analizar datos médicos, diagnosticar enfermedades y personalizar tratamientos.

Transporte: Los algoritmos se utilizan para optimizar rutas de transporte, controlar el tráfico y desarrollar vehículos autónomos.



Inteligencia artificial: Los algoritmos son la base de la inteligencia artificial y se utilizan para desarrollar sistemas que pueden aprender y tomar decisiones.

Estos son solo algunos ejemplos de cómo los algoritmos están presentes en nuestra vida diaria. A medida que la tecnología avanza, es probable que los algoritmos jueguen un papel aún más importante en el futuro.

2.1. Tipos de algoritmos: Gráficos (diagramas de flujo) y no grafico (pseudocódigo)

2.1.1. Diagrama de flujo: Un diagrama de flujo en programación es una representación gráfica de los pasos que se deben seguir para resolver un problema. Es decir, es la representación gráfica de un algoritmo.

Los diagramas de flujo son una herramienta muy útil para los técnicos y no técnicos en muchos campos, como la programación, la tecnología, las finanzas y la gestión de proyectos

2.1.2. Pseudocódigo: El pseudocódigo es una descripción de un algoritmo o programa que se escribe en un lenguaje natural es decir humano. Se utiliza para planificar el proceso de programación antes de escribir el código real.

El pseudocódigo es una herramienta útil para programadores, estudiantes de informática y comunicadores.

3. Metodología para la solución de problemas por medio de computadoras.

A fin de resolver un problema utilizando sistemas de cómputo, debe seguirse una serie de pasos que permiten avanzar por etapas bien definidas hacia la solución.

Estas etapas son las siguientes:

- 1) Análisis o definición del problema.
- 2) Análisis de los datos.
- 3) Diseño de la solución.
- 4) Codificación.
- 5) Prueba y depuración.
- 6) Documentación.
- 7) Mantenimiento.

1) Análisis o definición del problema: Es importante saber que se desea obtener al final del problema.

2) Análisis de los datos: Se debe especificar la entrada y salida de datos, esta etapa consiste en colocarnos en lugar de la computadora deduciendo los elementos que necesitaremos para alcanzar el resultado.



- 3) **Diseño de la solución:** Esta etapa consiste en la obtención de un algoritmo que resuelva adecuadamente el problema. El diseño de un algoritmo utilizando diagrama de flujo o pseudocódigo.
- 4) **Codificación:** Se refiere a la obtención de un programa definitivo que pueda ser comprensible para la máquina. Incluye una etapa que se reconoce como compilación.

Programa Fuente: Está escrito en un lenguaje de programación. (pascal, C++, Visual Basic, Python, JavaScript, etc.). Que es entendible por el programador.

- 5) **Prueba y depuración:** Una vez que se ha obtenido el programa ejecutable, este es sometido a prueba a fin de determinar si resuelve o no el problema planteado en forma satisfactoria.
El proceso de depuración consiste en localizar los errores y corregirlos en caso de que estos existan. Si no existen errores, puede entenderse la depuración como una etapa de refinamiento en la que se ajustan detalles para optimizar el desempeño del programa.
- 6) **Documentación:** El programa resultante no podrá ser supervisada en todas las ocasiones por el programador, debe crearse un manual o guía de operación que indique los pasos a seguir para utilizar el programa.
- 7) **Mantenimiento:** Se refiere a las actualizaciones que deban aplicarse al programa cuando las circunstancias así lo requieran. Este programa deberá ser susceptible de ser modificado para adecuarlo a nuevas condiciones de operación. Cualquier actualización o cambio en el programa deberá reflejarse en su documentación.

4. Traductores (intérpretes y compiladores):

Los traductores son programas que convierten código escrito en un lenguaje de programación a otro lenguaje. Los traductores se dividen en dos categorías principales:

- 4.1. **Intérpretes:** Los intérpretes traducen el código línea por línea, ejecutando cada línea inmediatamente después de traducirla. Los intérpretes son más lentos que los compiladores, pero son más fáciles de usar para depurar programas.

Ejemplos de Intérpretes: Python, JavaScript, PHP, Ruby

- 4.2. **Compiladores:** Los compiladores traducen todo el código fuente a un archivo ejecutable que se puede ejecutar directamente por la computadora. Los compiladores son más rápidos que los intérpretes, pero requieren más tiempo para compilar el código fuente.

Ejemplos de compiladores: C, C++, Java

¿Cuál es la diferencia entre un intérprete y un compilador?

La principal diferencia entre un intérprete y un compilador es cómo traducen el código fuente. Los intérpretes traducen el código línea por línea, mientras que los compiladores traducen todo el código fuente a un archivo ejecutable.



5. Análisis del problema

Introducción. - En la vida diaria se va enfrentando diferentes tipos de problemas, desde los más simples hasta los más complicados. En el transcurso de la vida se ha tenido problemas: inicialmente aprender a escribir, identificar al sujeto y predicado en una oración, resolver sistemas de ecuaciones y muchos más.

En todos los problemas se busca una solución, se debe examinar con cuidado que tipo de problema es el que se enfrenta, con qué elementos se cuenta para hallar dicha solución y cuál es el procedimiento más apropiado para llegar a dicha solución.

Para resolver un problema se debe dar respuesta al que y al como:

1. **¿El Qué?** Se deberá analizar los elementos de entrada y salida
 - ❖ ¿Qué es lo que deseo obtener? Es la salida de datos.
 - ❖ ¿Qué es lo que necesito para obtener lo que deseo? Es la entrada de datos.
2. **¿El cómo?** El cómo exigirá la máxima imaginación y creatividad que se pueda tener para proponer un camino o secuencia de pasos para la resolución del problema dado.
 - ❖ ¿Cómo soluciono el problema, cual es el camino más apropiado?

Por ejemplo: se desea cocinar un picante de pollo

¿El Qué?

- ❖ ¿Qué es lo que tengo que preparar? Se refiere a la salida de datos, en este caso un picante de pollo.
- ❖ ¿Qué es lo que necesito para lograr esa preparación? En este caso identificare los ingredientes (entrada de datos).

¿El cómo?

- ❖ ¿Cómo realizare la preparación? Es el conjunto de acciones que se seguirán para obtener el resultado buscado, en este caso cocinar un picante de pollo a través de la preparación que serán los pasos que me permitirán obtener lo que quiero.

Ejemplo.

Algoritmo para hacer una tortilla de papas:



El algoritmo Inicia	1) Inicio
	2) Ingredientes:
	4 patatas medianas
	1 cebolla
	6 huevos
	Aceite de oliva
	Sal
¿Qué?	3) Preparación:
Entrada de datos:	Pelar y cortar las patatas en rodajas finas. Pelar y cortar la
Ingredientes	cebolla en juliana.
Salida de datos	En una sartén grande, calentar abundante aceite de
Picante de Pollo	oliva. Añadir las patatas y la cebolla y freír a fuego medio
	hasta que estén blandas. Mientras tanto, batir los huevos en
	un bol con sal, añadir los huevos batidos. Mezclar bien.
	En la misma sartén, calentar un poco de aceite. Verter la
	mezcla de huevo, patata y cebolla en la sartén.
	Cocinar a fuego medio-bajo durante unos minutos, hasta
	que la tortilla esté dorada, por un lado.
	Con la ayuda de un plato o una tapadera, dar la vuelta a la
	tortilla y cocinarla por el otro lado hasta que esté dorada.
	Retirar la tortilla de la sartén y dejarla enfriar un poco.
	4) Servir:
	Cortar la tortilla en porciones y servir.
El algoritmo termina	5) Fin

Cada paso del algoritmo es una instrucción clara y precisa que se debe realizar. Además, el orden de los pasos es importante, ya que algunos pasos dependen de otros anteriores.

Los algoritmos están presentes en muchas de nuestras actividades diarias, desde seguir una receta de cocina hasta utilizar un cajero automático.

ACTIVIDAD

1. ¿Qué es la programación?
2. ¿Qué es un algoritmo?
3. Escribe un ejemplo de algoritmo utilizado en la vida cotidiana, fuera de los que ya se mencionó en la clase.
4. ¿Qué es un diagrama de flujo?
5. ¿Qué es un pseudocódigo?
6. Explique de manera breve los 7 pasos de la metodología para la solución de problemas por medio de computadoras.
7. ¿Qué diferencia existe entre un compilador y un intérprete?
8. Para dar respuesta a un problema se debe responde a: