

Fundamentos de Programación y Algoritmos

Bloque 1: Introducción a la Programación y Algoritmos

Formulación de problemas.

Etapas en la resolución de problemas computacionales.

Algoritmo.

Definición.

Características.

> ¿QUÉ ES PROGRAMAR?

Programar es definir instrucciones para ser ejecutadas por un ordenador, el objetivo es resolver un problema.

Si deseamos aprender a programar debemos saber lo siguiente:

Los programadores son personas que resuelven problemas, por lo que para llegar a ser programador eficaz se necesita aprender a resolver problemas de un modo riguroso y sistemático.

La resolución de un problema exige el diseño de un algoritmo.

algoritmo

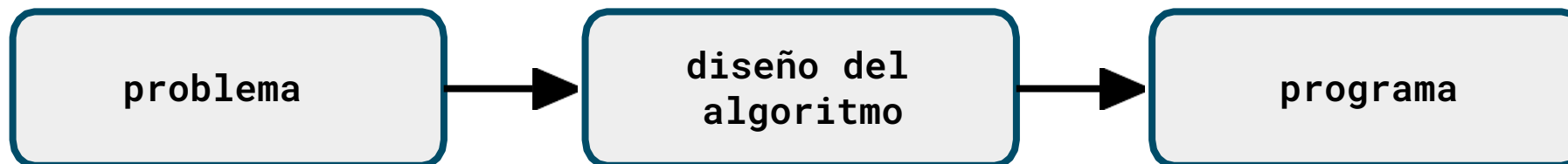
\neq

programa

Es un conjunto de pasos ordenados y finitos que permiten resolver un problema o tarea específica.

Es una lista de instrucciones “escritas” en un lenguaje de programación, que una computadora interpreta para resolver un problema.

> IMPORTANCIA DEL ALGORITMO



- Sin algoritmo no puede existir un programa.
- Un algoritmo es independiente del lenguaje de programación y de la computadora que lo ejecuta.
- Un mismo algoritmo puede expresarse en lenguajes diferentes de programación y ejecutarse en computadoras distintas.

> PARTES DE UN ALGORITMO INFORMÁTICO

Las tres partes de un algoritmo son:

1. **Input (entrada).** Información que damos al algoritmo con el que va a trabajar para ofrecer la solución esperada.
2. **Proceso.** Conjunto de pasos para que, a partir de los datos de entrada, llegue a la solución de la situación.
3. **Output (salida).** Resultados, a partir de la transformación de los valores de entrada durante el proceso.

De este modo, un algoritmo informático parte de un estado inicial y de unos valores de entrada, sigue una serie de pasos sucesivos y llega a un estado final en el que ha obtenido una solución.

> CARACTERÍSTICAS DE ALGORITMOS

Asimismo, los algoritmos presentan una serie de **características comunes**, son:

- **Precisos.** Objetivos, sin ambigüedad.
- **Ordenados.** Presentan una secuencia clara y precisa para poder llegar a la solución.
- **Finitos.** Contienen un número determinado de pasos.
- **Concretos.** Ofrecen una solución determinada para la situación o problema planteados.
- **Definidos.** El mismo algoritmo debe dar el mismo resultado al recibir la misma entrada.

> **ALGUNOS ALGORITMOS FAMOSOS (E INFLUYENTES)**

Aunque todo esto te suene formal, e incluso aburrido, el ingenio de programadorxs de todo el mundo ha conseguido que algunos algoritmos se hayan hecho famosos.

- **PageRank, de Google**

Uno de los más utilizados del mundo. Se trata del conjunto de algoritmos que utiliza Google para determinar la importancia de los documentos indexados por su motor de búsqueda.

Dicho de otro modo, cuando realizas una búsqueda en Google, es uno de los elementos que **decide el orden en el que se te muestran los resultados**.

> **ALGUNOS ALGORITMOS FAMOSOS (E INFLUYENTES)**

- **El algoritmo del Timeline de Facebook.**

Se trata de otro algoritmo que influye en nuestra vida mucho más de lo que creemos. El conjunto de algoritmos que alimentan el Timeline de Facebook determina los contenidos que se nos muestran en el espacio más frecuentado de la red social. Así, en base a una serie de parámetros (gustos personales, respuesta a contenidos anteriores, etc), los algoritmos deciden cuál es contenido que nos va a mostrar la red social y en qué orden lo hará.

- **Algoritmos de Trading de Alta Frecuencia.**

Mueven miles de millones de dólares en los mercados cada día. Se trata de algoritmos utilizados por muchas de las más importantes entidades financieras del mundo, que lanzan órdenes al mercado en función del beneficio que éstos prevén obtener, según las condiciones de mercado dadas en cada momento.



>> Recomendación

El código de la discordia

Un grupo de jóvenes sostiene una batalla judicial larga y cuesta arriba contra Google. Ellos afirman ser los creadores del algoritmo que dio origen a Google Earth.

>> Ejemplo:

Un procedimiento que realizamos varias veces al día consiste en lavarnos los dientes. Veamos la forma de expresar este procedimiento como un Algoritmo:

> ALGORITMO

1. Tomar la crema dental.
2. Destapar la crema dental.
3. Tomar el cepillo de dientes.
4. Aplicar crema dental al cepillo.
5. Tapar la crema dental.
6. Abrir la llave del lavamanos.
7. Remojar el cepillo con la crema dental.
8. Cerrar la llave del lavamanos.
9. Frotar los dientes con el cepillo.

- 10.** Abrir la llave del lavamanos.
- 11.** Enjuagarse la boca.
- 12.** Enjuagar el cepillo.
- 13.** Cerrar la llave del lavamanos.
- 14.** Cerrar la llave del lavamanos.

FIN DEL ALGORITMO. <<

Tengo que crear un
algoritmo en el
cual pueda sumar
dos números y
mostrar el
resultado de esa
suma.





>> Rescribimos el problema:

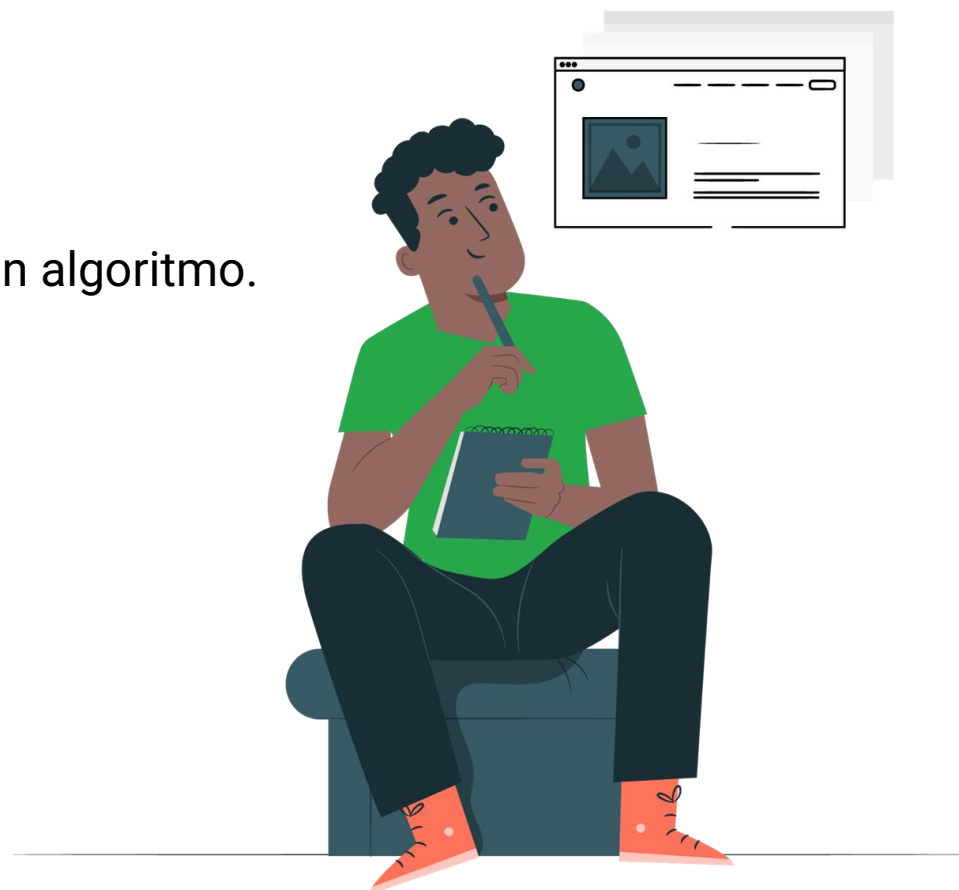
Tengo que crear un algoritmo en el cual pueda sumar dos números y mostrar el resultado de esa suma



Dados 2 números reales, calcular y mostrar el resultado de la suma entre ellos.

> ¿Cómo sería el proceso?

1. Definición del problema.
2. Análisis del problema.
3. Diseño y técnicas para la formulación de un algoritmo.
4. Codificación.
5. Prueba y Depuración.
6. Documentación.
7. Mantenimiento.





Definición del problema: Dados 2 números reales, calcular y mostrar el resultado de la suma entre ellos.



Análisis del problema: Necesitamos dos números reales (entrada), los cuales vamos a sumar (proceso), y mostrar el resultado (salida).



Diseño del algoritmo: Ingresar un número (llamado num1), luego el segundo número (llamado num2), después vamos a calcular la suma ($\text{num1} + \text{num2}$), y finalmente vamos a mostrar el resultado.



Codificación

```
num1=int(input("ingrese primer numero:"))  
num2=int(input("ingrese segundo numero:"))  
suma=num1+num2  
print("el resultado es", suma)
```



Prueba y Depuración

num1	num2	suma	salida
3	5	8	8



Documentación

Vamos a crear una anotación sobre qué es lo que realiza nuestro programa, que en este caso, simplemente suma dos números

```
1 import java.util.Scanner;
2
3 class Suma {
4
5     public static void main(String[] args) {
6         int num1, num2, suma;
7         Scanner teclado = new Scanner(System.in);
8         System.out.println("Introduce dos números: ");
9         num1 = teclado.nextInt();
10        num2 = teclado.nextInt();
11        suma = num1 + num2;
12        System.out.println("La suma de num1 y num2 es: " + suma);
13    }
14 }
```

este programa realiza la suma de dos números, donde recibe num1 y num2, luego los suma y finalmente muestra el resultado de esa suma



Mantenimiento

Si a futuro queremos mejorar nuestro programa, añadirle más funcionalidades o que realice otras operaciones, podemos hacerlo, siempre dependerá del contexto donde se lo utilice y el problema a resolver.



**Quédate a escuchar
otros ejemplos y
aprende a diseñar
algoritmos un poco
más complejos.**

>> Ejemplo 1:

Calcular el área y el perímetro de un rectángulo, para lo cual se deben ingresar el valor del lado A y el valor del lado B, ambos números reales.

1. Definición del problema

- ¿El enunciado es preciso?
- ¿Es posible calcular el área y perímetro de un rectángulo?

2. Análisis del problema

- ¿Cuáles son los datos de entrada y salida?
- ¿Son suficientes los datos de entrada para resolver el problema?
- ¿Conocemos las fórmulas para obtener el área y el perímetro?
- ¿Somos capaces de ejemplificar con algunos valores de entrada cuáles serían los valores de salida?
- ¿Qué tipo de instrucciones necesitamos para diseñar el algoritmo?

Tomemos nota de lo que tenemos y lo que sabemos.



Datos de Entrada: Dos números reales llamados LADO_A y LADO_B.

Datos de Salida: Valor de AREA y PERIMETRO, que serán reales.

Caso de Prueba 1:

Datos de Entrada: LADO_A = 4; LADO_B = 8

Datos de Salida: AREA = 32

Datos de Salida: PERIMETRO = 24

Caso de prueba 2:

Datos de Entrada: LADO_A = 3; LADO_B = 7

Datos de Salida: AREA = 21

Datos de Salida: PERIMETRO = 20

Necesitamos usar:

- *Instrucciones para el ingreso de datos*
- *Instrucciones para la asignación de datos*
- *Instrucciones para la salida de datos*

3. Diseñar el algoritmo

1. **Ingresar** un valor para LADO_A
2. **Ingresar** un valor para LADO_B
3. **Asignar** a ÁREA el resultado de: $LADO_A * LADO_B$.
4. **Asignar** a PERÍMETRO el resultado de: $2 * LADO_A + 2 * LADO_B$.
5. **Mostrar** el valor de ÁREA.
6. **Mostrar** el valor de PERÍMETRO.

Fin.

4. Prueba de Escritorio

LADO_A	LADO_B	ÁREA	PERÍMETRO	SALIDA
4	8	32	24	32 24
3	7	21	20	21 20
8,5	2	17	21	17 21

El sueldo de un vendedor es la suma de un monto fijo pagado por el gerente más el 20% de sus ventas mensuales. Teniendo como dato el sueldo fijo y el monto total de la venta del mes del vendedor, calcule y muestre el salario final que recibirá.

- ¿Sabemos obtener un porcentaje?
- ¿Podemos escribir una expresión algebraica que calcule el salario final?

Datos de Entrada: Dos valores reales llamados sueldo_inicial y monto_ventas

Datos de Salida: Valor del sueldo_final que será un número real.

Caso de Prueba 1:

Datos de Entrada:

sueldo_inicial = 10000

monto_ventas = 56000

Datos de Salida:

sueldo_final = 21200

Caso de Prueba 2:

Datos de Entrada:

sueldo_inicial = 12000

monto_ventas = 90000

Datos de Salida:

sueldo_final = 30000

Necesitamos usar:

- *Instrucciones para el ingreso de datos*
- *Instrucciones para la asignación de datos*
- *Instrucciones para la salida de datos*

3. Diseñar el algoritmo

1. **Ingresar** un valor para sueldo_inicial
 2. **Ingresar** un valor para monto_ventas
 3. **Asignar** a sueldo_final el resultado de: $\text{sueldo_inicial} + (\text{monto_ventas} * 20 / 100)$.
 4. **Mostrar** el valor de sueldo_final.
- Fin.

4. Prueba de Escritorio

sueldo_inicial	monto_ventas	sueldo_final	SALIDA
10000	56000	21200	21200
12000	90000	30000	30000
9500	25800	14660	14660

El sueldo de un vendedor es la suma de un monto fijo pagado por el gerente más un porcentaje de sus ventas mensuales en caso que superen un monto determinado.

Si el monto de sus ventas fue menor a \$20000, no recibe porcentaje, en caso contrario recibe el 20% de ese monto.

Teniendo como dato el sueldo fijo y el monto de la venta mensual del vendedor, calcule y muestre el salario final que recibirá.

- ¿Reconocemos las diferencias entre este ejemplo y el anterior?
- La condición para recibir el porcentaje de ventas es simple o compuesta?

Datos de Entrada: 2 valores reales llamados sueldo_inicial y monto_ventas
Datos de Salida: Valor del sueldo_final que será un número real.

Caso de Prueba 1:

Datos de Entrada:
sueldo_inicial = 10000
monto_ventas = 56000

Datos de Salida:
sueldo_final = 21200



sí recibe porcentaje

Caso de Prueba 2:

Datos de Entrada:
sueldo_inicial = 10000
monto_ventas = 15000

Datos de Salida:
sueldo_final = 10000



no recibe porcentaje

Caso de Prueba 3:

Datos de Entrada:
sueldo_inicial = 10000

monto_ventas = 20000

Datos de Salida:
sueldo_final =



¿?

Necesitamos usar:

- *Instrucciones para el ingreso de datos, de alternativa/condición*
- *Instrucciones para la asignación de datos, para la salida de datos*

3. Diseñar el algoritmo

1. Ingresar un valor para sueldo_inicial
 2. Ingresar un valor para monto_ventas
 3. si (monto_ventas < 20000) entonces
 - a) Asignar a sueldo_final el resultado de: sueldo_inicialsino
 - a) Asignar a sueldo_final el resultado de: $\text{sueldo_inicial} + (\text{monto_ventas} * 20 / 100)$
 1. Mostrar el valor de sueldo_final
- Fin

4. Prueba de Escritorio

1. Ingresar un valor para sueldo_inicial
 2. Ingresar un valor para monto_ventas
 3. si (monto_ventas < 20000) entonces
 - a) Asignar a sueldo_final el resultado de: sueldo_inicial
 sino
 - a) Asignar a sueldo_final el resultado de: sueldo_inicial + (monto_ventas*20/100)
 1. Mostrar el valor de sueldo_final
- Fin

sueldo_inicial	monto_ventas	sueldo_final	SALIDA
20000	25000	25000	25000
15000	20000	19000	19000
15000	10000	15000	15000