

L Ó G I C A D E P R O G R A M A C I Ó N

j i m c o s t d e v . c o m





C
O
N
T
E
N
I
D
O

CONTENIDO

1

Metodología para la creación de algoritmos.

2

Tipos de datos.

3

Tipos de operadores.





01

CREACIÓN DE ALGORITMOS





Definición

Cuando se enfrenta a un problema específico, el programador tiene la tarea de hallar una solución y expresarla mediante un algoritmo. Posteriormente, debe codificar dicho algoritmo en un lenguaje de programación específico y, finalmente, ejecutar el programa en una computadora para obtener una solución al problema planteado.



ETAPAS DE DESARROLLO

01

ANÁLISIS

Identificar mis **entradas**, saber que voy a realizar con ello (**proceso**). Y finalmente obtener un resultado (**salida**).

02

DISEÑO

Es la utilización de **pseudocódigo** o diagramas de flujo para plasmar lo que anteriormente analice.

03

IMPLEMENTACIÓN

Es utilizar un lenguaje de programación y un entorno de desarrollo para crear un programa o software.



Reto:

Tres amigos van a tomar café. Piden la cuenta y el camarero les dice que son **25 dólares** por los tres cafés. Cada uno pone **10 dólares**, en total **30**. Con los **5** que sobran, se queda cada uno **1** dólar, y los otros **2** para la propina del bar. Es decir, cada uno paga **9** dólares, que por los tres serían **27**, más las 2 de la propina, 29. **¿Dónde está el dólar que falta?**



SOLUCIÓN

El problema está en que el lenguaje comete un fallo.

Cada uno paga 9 dólares, en total 27, y dentro de esas, ya están las dos de propina. El razonamiento correcto es: 25 de los cafés, más 2 de propina, serían los 27 que en realidad han pagado.





02

TIPOS DE DATOS





Variable= espacio en memoria
que almacena un dato y este
puede cambiar a lo largo del
algoritmo.





01

Texto

02

Numéricos

03

Lógicos o Booleanos

04

Colecciones de Datos



Representación de Texto

1. **Caracteres (char):** Representan un único carácter, como una letra, número o símbolo.
2. **Cadenas de caracteres (string):** Representan una secuencia de caracteres, como palabras o frases.



Númericos

1. **Enteros (int)**: Representan números enteros sin decimales, tanto positivos como negativos.
2. **Reales (float, double)**: Representan números con decimales. Los tipos float y double se diferencian en la precisión que pueden manejar.



Lógicos o Booleanos

Booleanos (bool): Representan un valor lógico, que puede ser verdadero (true) o falso (false).



Colecciones de Datos

1. **Arreglos (arrays)**: Representan una colección de elementos del mismo tipo de datos, que se almacenan en posiciones consecutivas de memoria. - **Colecciones**
2. **Listas (lists)**: Representan una colección ordenada de elementos, que pueden ser de diferentes tipos de datos. - **Secuencias**
3. **Estructuras (structs)**: Permiten crear tipos de datos personalizados que pueden contener múltiples campos con diferentes tipos de datos. - **Objetos**





03

TIPOS DE OPERADORES



TIPOS DE OPERADORES

01

ARITMÉTICOS

Los operadores aritméticos se utilizan para realizar operaciones matemáticas en expresiones numéricas.

03

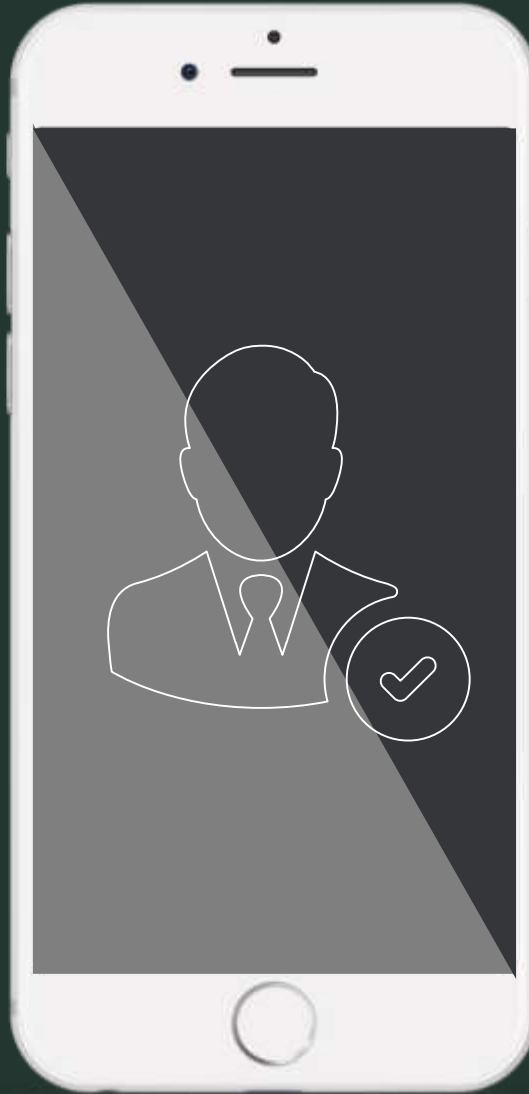
COMPARACIÓN

Los operadores de comparación, se utilizan para comparar dos valores y devuelven un resultado booleano (verdadero o falso) que indica si la comparación es cierta o no.

02

LÓGICOS

Los operadores lógicos se utilizan para evaluar condiciones lógicas y devuelven un valor de verdadero (true) o falso (false). Estos operadores son fundamentales para la toma de decisiones en programación.



ARITMÉTICOS

1. **Suma (+)**: se utiliza para sumar dos valores.
2. **Resta (-)**: se utiliza para restar un valor de otro.
3. **Multiplicación (*)**: se utiliza para multiplicar dos valores.
4. **División (/)**: se utiliza para dividir un valor por otro.
5. **Módulo (%)**: se utiliza para obtener el resto de una división.
6. **Incremento (++) y decremento (--)**: se utilizan para aumentar o disminuir en una unidad el valor de una variable.



LÓGICOS

1. **Y lógico (&&):** devuelve verdadero si ambas condiciones son verdaderas.
2. **O lógico (||):** devuelve verdadero si al menos una de las condiciones es verdadera.
3. **Negación lógica (!):** invierte el valor de verdad de una condición.



COMPARACIÓN

1. **Igualdad (==):** Compara si dos valores son iguales y devuelve verdadero si lo son. Por ejemplo: $3 == 3$ devuelve verdadero, "Hola" == "Hola" devuelve verdadero.
2. **Desigualdad (!=):** Compara si dos valores son diferentes y devuelve verdadero si lo son. Por ejemplo: $3 != 5$ devuelve verdadero, "Hola" != "Mundo" devuelve verdadero.
3. **Mayor que (>):** Compara si el valor de la izquierda es mayor que el valor de la derecha y devuelve verdadero si lo es. Por ejemplo: $5 > 3$ devuelve verdadero.



COMPARACIÓN

1. **Menor que ($<$):** Compara si el valor de la izquierda es menor que el valor de la derecha y devuelve verdadero si lo es. Por ejemplo: $2 < 4$ devuelve verdadero.
2. **Mayor o igual que (\geq):** Compara si el valor de la izquierda es mayor o igual que el valor de la derecha y devuelve verdadero si lo es. Por ejemplo: $5 \geq 5$ devuelve verdadero.
3. **Menor o igual que (\leq):** Compara si el valor de la izquierda es menor o igual que el valor de la derecha y devuelve verdadero si lo es. Por ejemplo: $3 \leq 4$ devuelve verdadero.



Reto



Investigar sobre:

1. ¿Qué es una variable y que es una constante? ¿En qué se diferencian?
2. ¿Que son las estructuras de control y para qué se usan?



GRACIAS

