



PROGRAMACIÓN I

TUP - 2025 - 1C - Turno noche

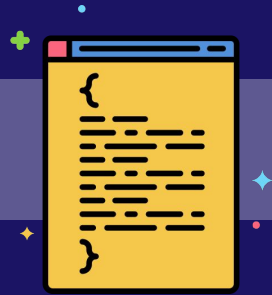


01

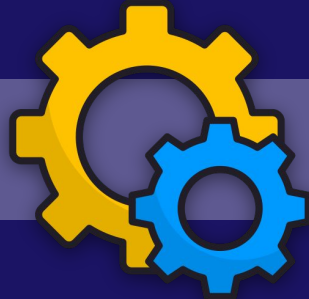
SECUENCIALES

DEL CÓDIGO AL PROGRAMA

- El proceso de transformar un **código fuente** en un **programa** que contiene las instrucciones que sean comprensibles por la computadora se llama **compilación**.



Código fuente



Compilación



Programa

CODIFICACIÓN

Cada elemento que utilicemos en un lenguaje de programación debe estar sujeto a una estricta sintaxis. Los elementos que el lenguaje admite son:

- **Variables y constantes**
- **Operadores**
- **Expresiones**
- **Palabras reservadas del lenguaje**

Como muchos lenguajes, C y C++ son case-sensitive. Esto significa que hace diferencias entre mayúsculas y minúsculas.

PALABRAS RESERVADAS DEL LENGUAJE

Palabras que el lenguaje utiliza para identificar tipos de datos, estructuras de programación, etc. Tienen un significado especial para el lenguaje y no pueden ser utilizados como identificadores de nombre.

| Propósito | Palabras reservadas |
|---------------------------|------------------------------------------------------------------------------------------------|
| Tipos de datos | bool, int, float, char, short, void, long, double |
| Elementos de programación | if, else, switch, default, break, for, while, do, return, auto, struct, class, static, virtual |
| Operadores | new, delete, sizeof |

VARIABLES Y CONSTANTES

Representación simbólica de espacios de memoria. Es donde se almacenan los datos en procesamiento.

Una variable se identifica con un **tipo de dato** y un **identificador de nombre** (los elige el programador), y permite escribir un dato en la memoria o leer un dato de la memoria. Se puede modificar su valor las veces que sea necesario.

Una constante se identifica con la palabra reservada ***const***, un tipo, un nombre (lo elige el programador), y un valor que no puede ser modificado durante el transcurso del programa en ejecución.

VARIABLES Y CONSTANTES

Ejemplos de declaraciones de variables:

- `int` legajo;
- `float` sueldo;
- `string` nombre = "Programación I";
- `string` apellido;
- `char` letra;

Ejemplos de declaraciones de constantes:

- `const int` EDAD_MIN = 18;
- `const float` IMPUESTO = 10.5;
- `const char` PAIS = 'A';

OPERADORES MATEMÁTICOS

Necesarios para realizar cálculos matemáticos. Los paréntesis tienen el mismo efecto que en la matemática en las expresiones algebraicas. Sin embargo, en programación no se utilizan corchetes ni llaves para la separación de términos.

| Operador | Operación |
|----------|-----------------------------|
| + | Suma |
| - | Resta |
| * | Producto |
| / | División real |
| % | Resto de la división entera |

Ejemplos:

$$((2+3)*5)+10 \quad \square \quad 35$$

$$2+3*5+10 \quad \square \quad 27$$

$$5/2 \quad \square \quad 2$$

$$5.0/2 \quad \square \quad 2.5$$

$$5\%2 \quad \square \quad 1$$

OPERADORES DE ASIGNACIÓN

Nos permite asignar la expresión que se encuentra a la derecha del operador en la variable que se encuentra a la izquierda.

Formas correctas:

- edad = 50;
- a = b;
- nombre = "Jose";
- nombreApellido = "Sol" + "Rios"
- caracter = 'Y';
- precio = cant * pu;
- cont = 0;
- cont = cont + 1;
- cont = 3 + 1
- cont++;

Errores de sintaxis

- 50 = 50;
- 50 = edad;

Sintácticamente correcto pero sin sentido.

- edad = edad;

¿COMO EMPEZAR CON UN EJERCICIO?

Lo primero, antes de comenzar a resolver un ejercicio o problema, primeramente en diagrama y luego en código, es hacer un análisis del mismo en el que deberemos identificar:

1 Datos de entrada

Determinar cuántos y cuáles son los datos de entrada de nuestro programa.

Ponerles un nombre y determinar su tipo de datos.

2 Proceso

El algoritmo ideado debe poder transformar los datos de entrada en la información de salida.

3 Datos de salida

La información de salida debe ser clara, prolija e informar estrictamente lo necesario.

EJEMPLO

1. Hacer un programa para ingresar por teclado la cantidad de horas trabajadas por un operario y el valor que se le paga por hora trabajada y listar por pantalla el sueldo que le corresponda.

1 Datos de entrada

- Horas trabajadas
- Valor por hora

2 Proceso

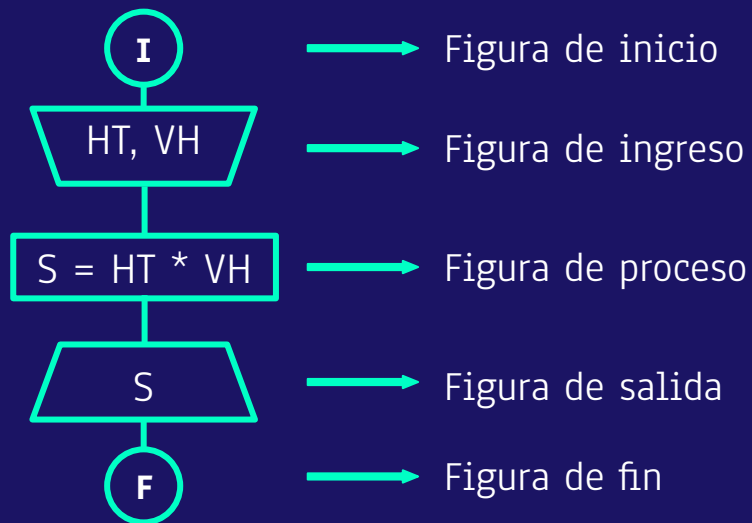
- Calcular el sueldo
 $\text{HorasTrabajadas} * \text{ValorHora}$

3 Datos de salida

- Sueldo correspondiente

DIAGRAMACIÓN

1. Hacer un programa para ingresar por teclado la cantidad de horas trabajadas por un operario y el valor que se le paga por hora trabajada y listar por pantalla el sueldo que le corresponda.



CODIFICACIÓN

1. Hacer un programa para ingresar por teclado la cantidad de horas trabajadas por un operario y el valor que se le paga por hora trabajada y listar por pantalla el sueldo que le corresponda.

