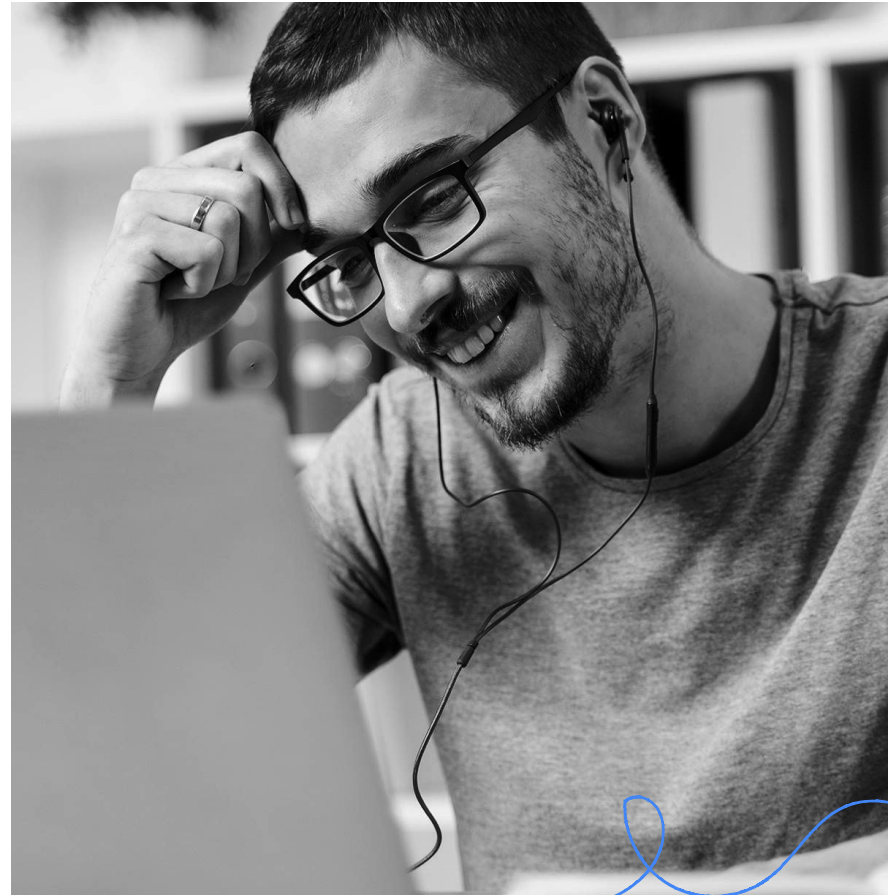


MÓDULOS

Tipos, definición y uso, comunicación



TIPOS DE MÓDULOS

FUNCIONES

Son módulos que realizan alguna **operación sobre sus datos de entrada**, y devuelven un **único resultado**.

PROCEDIMIENTOS

Son módulos que realizan **alguna operación** sobre sus **datos de entrada** pero a diferencia de las funciones, puede **devolver uno, muchos o incluso ningún**



Elementos de una función

tipo de dato del valor que la función retorna

La lista de parámetros se refiere al tipo, orden y número de parámetros de una función. Los parámetros son opcionales; es decir una función puede no tener parámetros.

```
tipo_de_retorno nombre_funcion ( lista_parametros ) {  
    cuerpo de la función;  
}
```

contiene el conjunto de sentencias que define lo qué se ejecuta al invocar la función.

Elementos de una función - Ejemplo

Valor de retorno

Nombre y lista de parámetros

```
int max(int num1, int num2) {  
    // declaracion de variable local  
    int result;  
  
    if (num1 > num2)  
        result = num1;  
    else  
        result = num2;  
  
    return result;  
}
```

Cuerpo de la
función

Elementos de un PROCEDIMIENTO - Ejemplo

Valor de retorno

Nombre y lista de parámetros

```
void obtenerNotas(int &val1, int &val2) {  
    cout << "Ingreso primer nota:";  
    cin >> val1;  
  
    cout << "Ingreso segunda nota:";  
    cin >> val2;  
  
    return;  
}
```

Cuerpo del
procedimiento

Funciones y procedimientos

- Son un conjunto de declaraciones (statements) cerradas entre { }
- A esto se le llama el cuerpo (body) de la función/procedimiento.
- Estas declaraciones definen qué hará la función.
- Cada módulo posee un nombre único
- El nombre de módulo es utilizado para invocar la función/procedimiento.
- Las funciones son piezas fundamentales en la construcción de programas en C y C++.

¿Por qué necesitamos funciones?

- Para evitar escribir código o lógicas repetidas en nuestra aplicación.
- El mismo código puede ser reutilizado en diferentes partes del programa.
- No hay límite a la cantidad de veces que puedo utilizar una función.
- Hace nuestro código modular, dividiendo el programa en módulos cohesivos.
- Reduce la complejidad de nuestro código.

Estructura.

- Una función recibe parámetros.
- La función puede utilizar los parámetros para procesar.
- Luego de procesar, la función puede devolver (return) un valor.
- Las funciones solo pueden devolver 1 valor.
- Tanto los parámetros como el return son opcionales.

<return type> <nombre> (<parametros>) { <cuerpo / body> }

- `int Add(int, int){ }`
- `void Print(char ch) { }`

a practicar...