

Programación I

# Modularización

# Modularización

Consiste en dividir un programa grande y complejo en partes más pequeñas y manejables llamadas módulos o subprogramas. Estos módulos pueden ser funciones o procedimientos, que son unidades independientes de código que realizan tareas específicas.

A vibrant, stylized illustration of a futuristic city skyline at night. The scene is dominated by tall, dark skyscrapers with glowing windows and neon lights in shades of blue, purple, and pink. A flying car is visible in the upper right, leaving a red light trail. The overall atmosphere is high-tech and digital.

**Beneficios**

# **Facilita el mantenimiento del código**

Al dividir el código en módulos más pequeños y manejables, es más fácil realizar cambios y correcciones.



# Beneficios Reutilización de código

Los subprogramas pueden ser reutilizados en diferentes partes del programa o incluso en otros programas.

# Beneficios

# Facilita la

# depuración

Al tener partes más pequeñas y específicas del código, es más fácil identificar y corregir errores.

# Subprogramas

Son unidades independientes de código que realizan tareas específicas y pueden ser invocadas desde otras partes del programa principal.

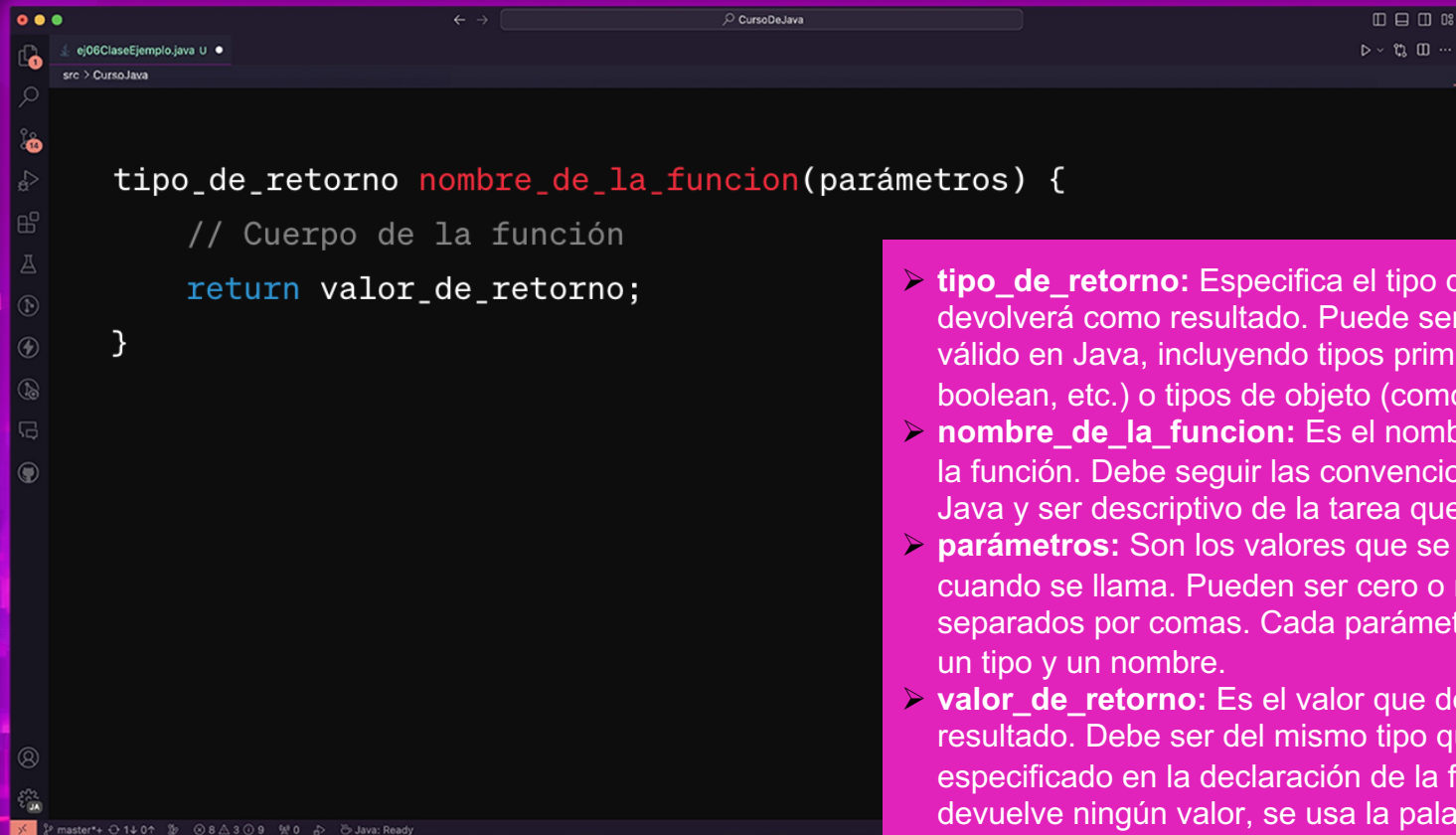
Pueden ser de dos tipos: funciones y procedimientos.

# Funciones

Una función es un subprograma que devuelve un valor después de realizar una operación. En Java, las funciones se definen usando el tipo de dato que devuelven.



# Declaración de funciones

A screenshot of an IDE window titled 'CursoDeJava'. The editor shows a Java function declaration. The code is: 

```
tipo_de_retorno nombre_de_la_funcion(parámetros) {  
    // Cuerpo de la función  
    return valor_de_retorno;  
}
```

 The IDE has a dark theme and a sidebar on the left with various icons. The status bar at the bottom shows 'master\*+ 14 0' and 'Java: Ready'.

- **tipo\_de\_retorno:** Especifica el tipo de dato que la función devolverá como resultado. Puede ser cualquier tipo de dato válido en Java, incluyendo tipos primitivos (como int, double, boolean, etc.) o tipos de objeto (como String, ArrayList, etc.).
- **nombre\_de\_la\_funcion:** Es el nombre único que identifica a la función. Debe seguir las convenciones de nombrado de Java y ser descriptivo de la tarea que realiza la función.
- **parámetros:** Son los valores que se pasan a la función cuando se llama. Pueden ser cero o más parámetros, separados por comas. Cada parámetro está compuesto por un tipo y un nombre.
- **valor\_de\_retorno:** Es el valor que devuelve la función como resultado. Debe ser del mismo tipo que el tipo de retorno especificado en la declaración de la función. Si la función no devuelve ningún valor, se usa la palabra clave void.



# Inicialización con valores explícitos

```
ej06ClaseEjemplo.java U
src > CursoJava

public class Suma {
    // Función para sumar dos números y devolver el resultado
    public static int sumar(int a, int b) {
        return a + b;
    }

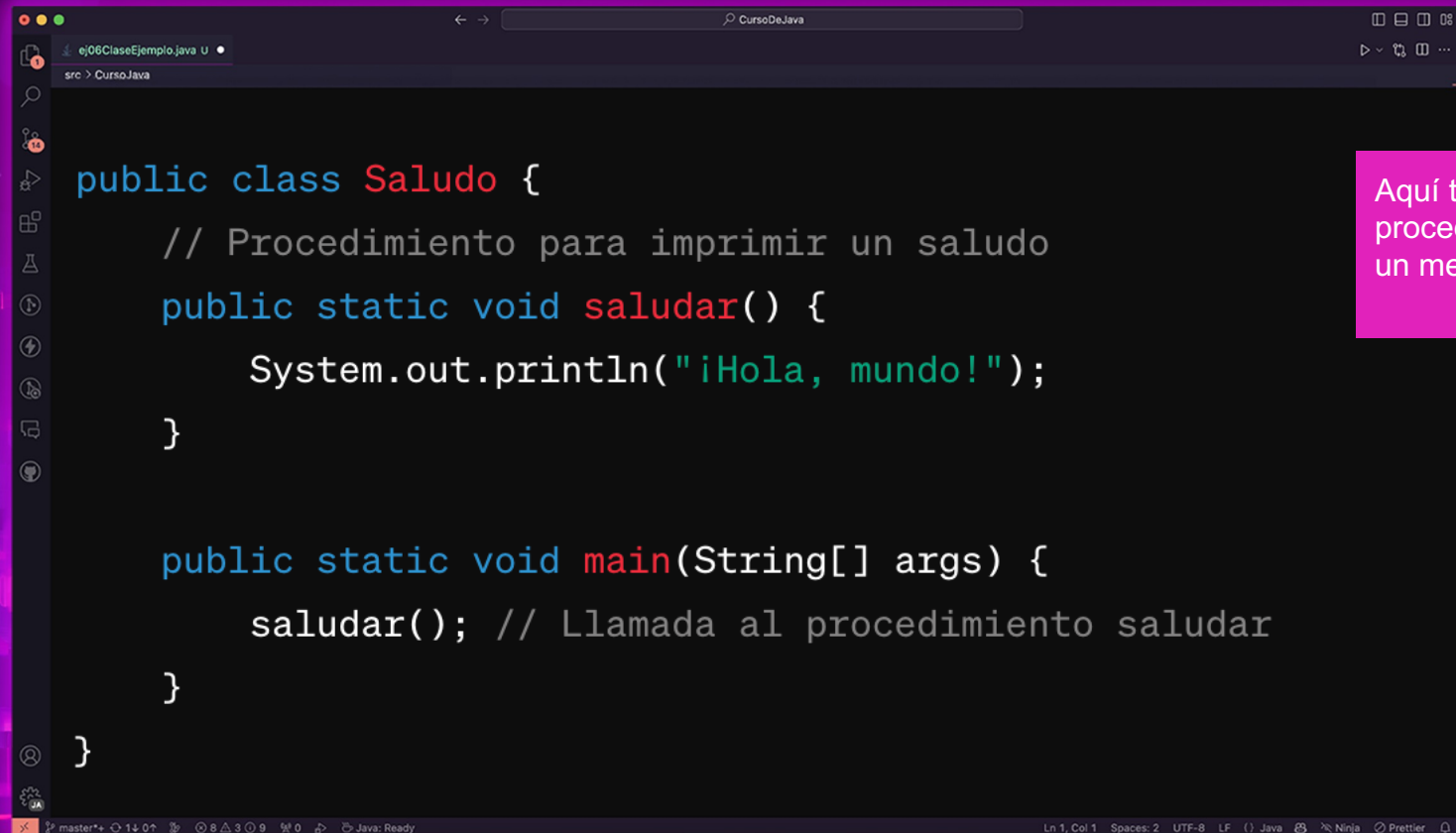
    public static void main(String[] args) {
        int num1 = 5;
        int num2 = 3;
        int resultado = sumar(num1, num2); // Llamada a la función sumar
        System.out.println("La suma es: " + resultado);
    }
}
```

Aquí tienes un ejemplo de una función que calcula la suma de dos números

# Procedimientos

Un procedimiento es un subprograma que realiza una tarea sin devolver un valor. En Java, los procedimientos se definen con el tipo de retorno void.

# Inicialización con valores explícitos



```
public class Saludo {  
    // Procedimiento para imprimir un saludo  
    public static void saludar() {  
        System.out.println("¡Hola, mundo!");  
    }  
  
    public static void main(String[] args) {  
        saludar(); // Llamada al procedimiento saludar  
    }  
}
```

The screenshot shows a code editor window titled 'CursoDeJava' with a file named 'ej06ClaseEjemplo.java'. The code defines a class 'Saludo' with a static method 'saludar' that prints '¡Hola, mundo!' and a static 'main' method that calls 'saludar'. The editor has a dark theme and a sidebar with icons for file explorer, search, and other IDE features.

Aquí tienes un ejemplo de un procedimiento que imprime un mensaje por pantalla