

# Centro Formación Profesional Nro 35

## *Técnicas de programación, Lenguaje JAVA*

### Introducción

Java es un lenguaje versátil que nos permite aplicar múltiples paradigmas de programación estructurada, de objetos y funcional.

Desde el comienzo del curso fuimos viendo distintas estructuras de control como el if, while, for, etc. También hemos empezado a notar que nuestros programas empiezan a ser cada vez más largos e incluso a tener código duplicado.

Para atacar estas complejidades veremos cómo utilizar **funciones y procedimientos**.



# Centro Formación Profesional Nro 35

## *Técnicas de programación, Lenguaje JAVA*

### Procedimientos

Un **procedimiento** es un **bloque de código** que **contiene una o más instrucciones, al cual podemos invocar para que sean ejecutadas**. Los procedimientos nos van a ayudar a hacer nuestro código más **legible** y evitar **código duplicado**.

#### Características

- No retornan **ningún valor (void)**.
- **Pueden recibir parámetros**, aunque no es obligatorio.

# Centro Formación Profesional Nro 35

## *Técnicas de programación, Lenguaje JAVA*

### Definición

The diagram illustrates the components of a Java method signature using the example `public static void ejecutar() {`. Annotations with arrows point to specific parts of the code: 'Retorno' points to `void`, 'Nombre del procedimiento' points to `ejecutar()`, and 'Parámetros' points to the empty parentheses `()`. The method body is enclosed in curly braces `{ ... }`. Inside the body, a `for` loop is shown: `for (int i = 0; i < 10; i++) {`. The loop's body contains the instruction `System.out.println(i);`. An arrow labeled 'Instrucciones' points to this line of code. Vertical lines are used to group the opening and closing braces of the method and the loop.

```
public static void ejecutar() {  
    for (int i = 0; i < 10; i++) {  
        System.out.println(i);  
    }  
}
```

Retorno

Nombre del procedimiento

Parámetros

Instrucciones