
Java™

Contents

Raíces del Código

Java es un lenguaje de programación **orientado a objetos (POO)** que se usa para crear desde aplicaciones de escritorio, móviles (como aplicaciones Android) hasta servidores y sistemas empresariales. Su filosofía principal es "**escribe una vez, corre en cualquier lugar**" porque funciona sobre la **Java Virtual Machine (JVM)**, que traduce el código Java a un lenguaje que cualquier sistema operativo puede ejecutar.

Fundamentos de Java a través de la gestión de viveros forestales

Descubre la sintaxis y la semántica fundamentales de Java analizando datos de un vivero forestal. Aprende métodos, variables y flujo de control mientras contribuyes a la reforestación tecnológica.

Instrucciones Generales

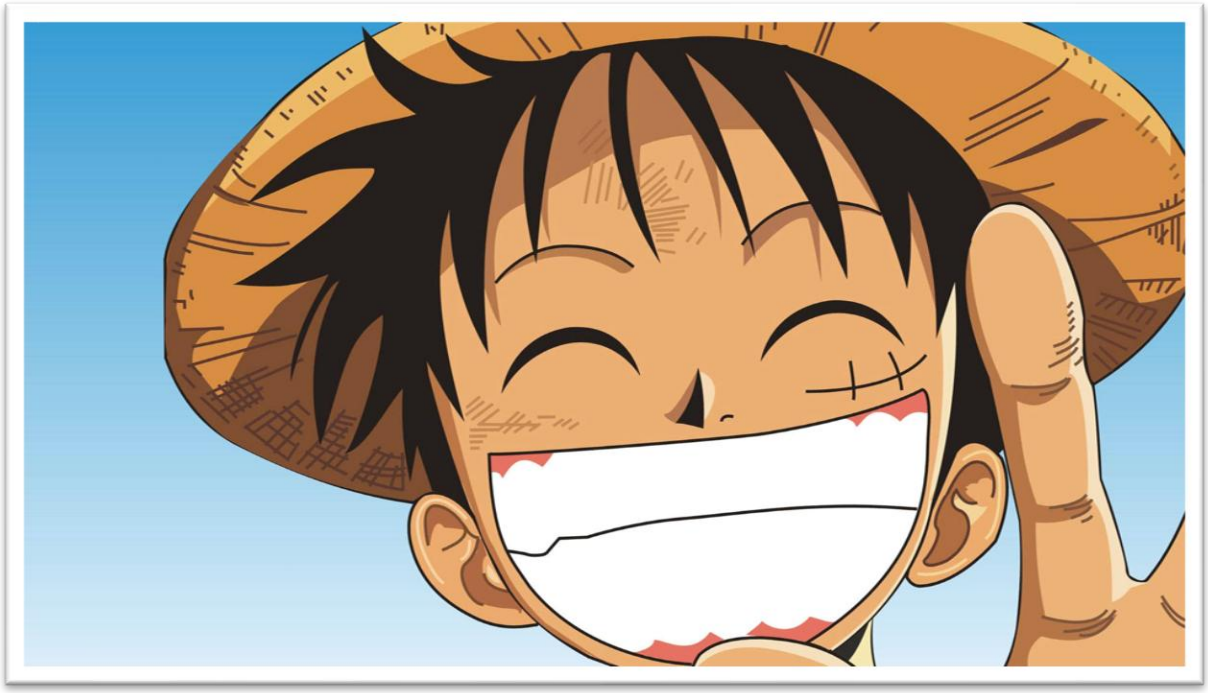
- ✓ ***Versión:*** Java 17+ (OpenJDK).
- ✓ ***Estructura:*** Una clase por archivo (sin package). ***Método public static void*.**
- ✓ ***Linter Obligatorio:*** Checkstyle (Google Java Style). Errores de estilo = KO.

- ✓ ***Comentarios:*** Obligatorios, solo en inglés y siguiendo formato Javadoc.
- ✓ ***IDE:*** IntelliJ IDEA (recomendado), VS Code, Eclipse, etc.
- ✓ ***Compilación:*** Vía terminal: javac Clase.java y java Clase.

I. Prólogo

En los viveros forestales, cada semilla representa el futuro de un ecosistema. El seguimiento preciso de la germinación y el crecimiento es vital para asegurar que los bosques del mañana sean fuertes.

Java, al igual que un bosque de secuoyas, es conocido por su estructura, su altura y su capacidad para perdurar en el tiempo. A través de este proyecto, aprenderás que programar, al igual que la silvicultura, requiere paciencia, estructura y una atención meticulosa a los detalles.



II. Introducción

*¡Te damos la bienvenida a **Raíces del Código**!*

Trabajarás con ejercicios prácticos que introducen conceptos de Java en un contexto de gestión forestal.

IMPORTANTE: En cada ejercicio, debes entregar una **Clase** que contenga el método estático solicitado.

- ✓ No incluyas el método ***main*** en tus archivos de entrega.
- ✓ La gestión de entrada y salida (***System.out y Scanner***) debe ocurrir dentro del método.

IV. Instrucciones Generales

- ✓ Usa Java 17+.
- ✓ No utilizes packages.
- ✓ Cada archivo debe llamarse exactamente como la clase (ej: FtHelloForest.java).
- ✓ El método debe ser ***public static void.***
- ✓ Para leer datos, usa ***import java.util.Scanner;.***

V. Ejercicio 0: Hola Bosque

- **Directorio:** ex00/
- **Archivo:** FtHelloForest.java
- **Método:** ftHelloForest()
- **Funciones Autorizadas:** System.out.println

Escribe un método que muestre un mensaje de bienvenida al sistema de gestión forestal. **Crear un main** que lo ejecute y demuestre que funciona.

```
Welcome to the Forest Management System!
```

VI. Ejercicio 1: Área de Plantación

- **Directorio:** ex01/
- **Archivo:** FtPlantingArea.java
- **Método:** ftPlantingArea()
- **Funciones Autorizadas:** Scanner, System.out.print/ln, Integer.parseInt

Solicita la longitud y el ancho (en metros) de una sección de reforestación y muestra el área total. **Crear un main** que lo ejecute y demuestre que funciona.

```
Enter length: 10
Enter width: 20
Total planting area: 200 m2
```

VII. Ejercicio 2: Conteo de Brotes

- **Directorio:** ex02/
- **Archivo:** FtSproutCount.java
- **Método:** ftSproutCount()
- **Funciones Autorizadas:** Scanner, System.out.print/ln

Un guardabosques cuenta los brotes nuevos en 3 parcelas distintas. Solicita los 3 valores y muestra la suma total. **Crear un main** que lo ejecute y demuestre que funciona.

```
Plot 1 sprouts: 50
Plot 2 sprouts: 30
Plot 3 sprouts: 20
Total sprouts counted: 100
```

VIII. Ejercicio 3: Control de Altura

- **Directorio:** ex03/
- **Archivo:** FtTreeHeight.java
- **Método:** ftTreeHeight()
- **Funciones Autorizadas:** Scanner, System.out.print/ln

Solicita la altura de un ejemplar en centímetros. Si mide más de 100 cm, indica que está listo para ser trasplantado al bosque y si no muestra "Not ready for transplanting!". **Crear un main** que lo ejecute y demuestre que funciona.

```
Enter tree height (cm): 120
Ready for transplanting!
```

IX. Ejercicio 4: Alerta de Sequía

- **Directorio:** ex04/
- **Archivo:** FtDroughtAlert.java
- **Método:** ftDroughtAlert()
- **Funciones Autorizadas:** Scanner, System.out.print/ln

Solicita los litros de lluvia caídos esta semana. Si es menor a 15 litros, ¡muestra “Activate irrigation system!", de lo contrario "Moisture levels optimal". **Crear un main** que lo ejecute y demuestre que funciona.

```
Rainfall (liters): 10
Activate irrigation system!
```

X. Ejercicio 5: Registro de Crecimiento

- **Directorio:** ex05/
- **Archivo:** FtGrowthLog.java
- **Métodos:** ftGrowthLogIterative(), ftGrowthLogRecursive(int n) y ftGrowthLogRecursiveHelper(int n, int day)
- **Funciones Autorizadas:** Scanner, System.out.print/ln

Ambos métodos deben imprimir el progreso día a día hasta llegar al objetivo dado por el usuario. **Crear un main** que lo ejecute y demuestre que funciona.

```
Target day: 3
Day 1: Growing...
Day 2: Growing...
Day 3: Growing...
Target reached!
```

XI. Ejercicio 6: Ficha del Vivero

- **Directorio:** ex06/
- **Archivo:** FtNurserySummary.java
- **Método:** ftNurserySummary()
- **Funciones Autorizadas:** Scanner, System.out.print/ln, nextLine().

Solicita el nombre del vivero y la especie principal de árbol, mostrando un resumen formateado. **Crear un main** que lo ejecute y demuestre que funciona.

```
>>> FtNurserySummary.ft_nursery_summary()
Enter nursery name: Valle Verde
Enter main species: Roble
Nursery: Valle Verde | Species: Roble | Status: Active
```

XII. Ejercicio 7: Inventario de Semillas

- **Directorio:** ex07/
- **Archivo:** FtForestSeeds.java
- **Método:** ftForestSeeds(String type, int qty, String unit)
- **Funciones Autorizadas:** System.out.print/ln, String.substring, String.toUpperCase/LowerCase

El método recibe parámetros y debe validar la unidad ("kilograms", "seeds", "bags").

- Si la unidad no es válida, mostrar "Unknown unit".
- Formatear el tipo de semilla para que empiece en mayúscula.
- Usar switch.
- **Crear un main** que lo ejecute y demuestre que funciona.

```
>>> FtForestSeeds.ft_forest_seeds("pine", 50, "kilograms")
Pine: 50 kilograms in stock.

>>> FtForestSeeds.ft_forest_seeds("OAK", 200, "seeds")
Oak: 200 seeds ready for planting.

>>> FtForestSeeds.ft_forest_seeds("fir", 10, "bags")
Fir: 10 bags of seeds available.

>>> FtForestSeeds.ft_forest_seeds("cedar", 5, "boxes")
Unknown unit.
```

XIII. Ejecución

El flujo real en tu terminal

Entra en la carpeta:

Primero tienes que estar donde está el archivo. Si no, el comando javac te dirá "***archivo no encontrado***" y te entrarán ganas de tirar el PC por la ventana.

- ✓ cd Desktop/proyectos_java/ex00
- ✓ ***javac*** FtHelloForest.java **-> La Compilación (El traductor)**
- ✓ ***java*** FtHelloForest.java **-> La Ejecución (El lanzamiento)**