



Programación

Unidad 1: Introducción a Java

Unidad 1

- Fundamentos del lenguaje Java
- IDEs
- Datos y tipos de datos
- Constantes y variables. Expresiones
- Entrada y salida de información
- Estructuras de control

Fundamentos de Java



Fundamentos de Java

Java se creó como parte de un proyecto de investigación para el desarrollo de software avanzado para una amplia variedad de dispositivos de red y sistemas embebidos. La meta era diseñar una plataforma operativa sencilla, segura, portable, distribuida y de tiempo real.



Fundamentos de Java

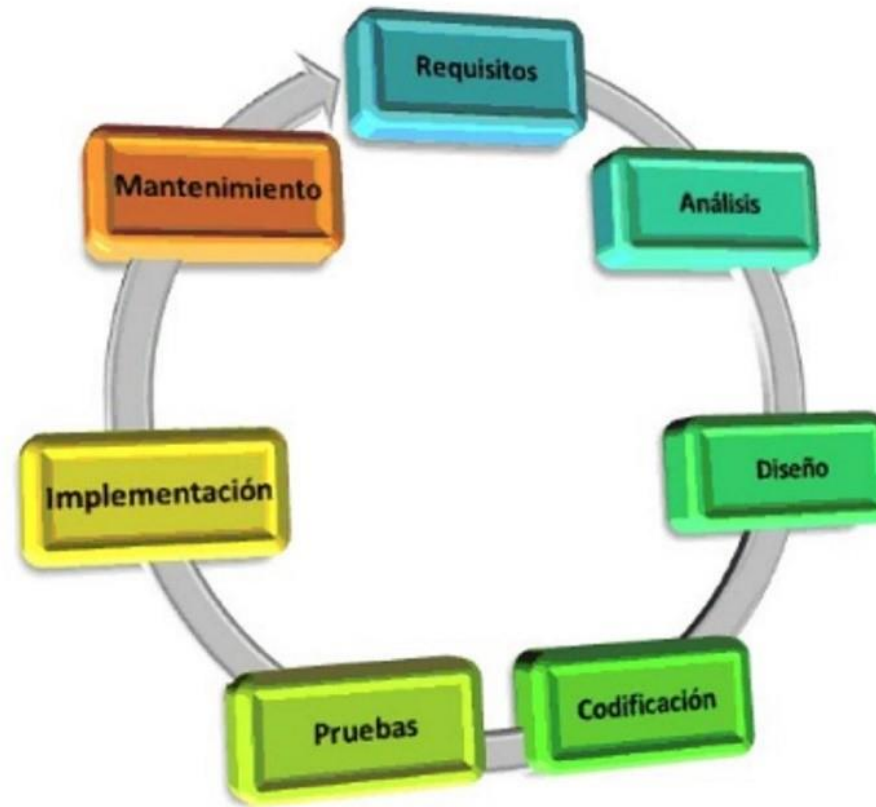
Java es un lenguaje desarrollado por Sun, apareciendo la primera versión en 1991.

¡Java nace como un lenguaje ideado en sus comienzos para programar electrodomésticos!

En sus primeras versiones, se llamó OAK (roble en inglés)

Es un lenguaje orientado a objetos, siendo un OBJETO un conjunto de variables y de métodos relacionados con esas variables.

Unidad 1: Introducción a Java

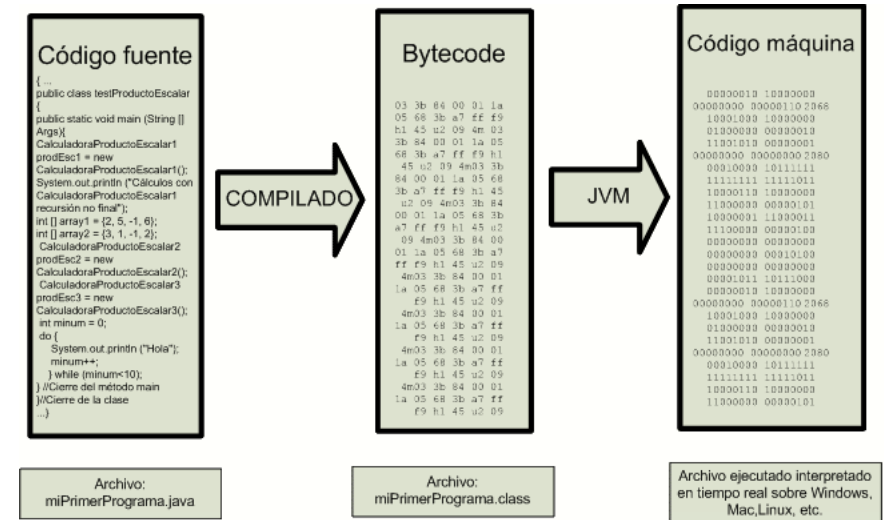


Fases o Etapas del Ciclo de Vida del Software

Compilación

Compilación. “Compilar” significa traducir el código escrito en “Lenguaje entendible por humanos” (por ejemplo: Java, C, Pascal, Fortran), a un código en “Lenguaje Máquina”, que entienden las máquinas, pero no entendible por nosotros.

La creación de programas en muchos lenguajes se basa en el proceso: escribir código fuente --> compilar y obtener programa ejecutable. El compilador se encarga de evitar que se pueda traducir un programa con código fuente mal escrito y de hacer otras verificaciones previas, de modo que el código máquina tiene ciertas garantías de que cumple cuando menos con los estándares de sintaxis obligatorios de un lenguaje.

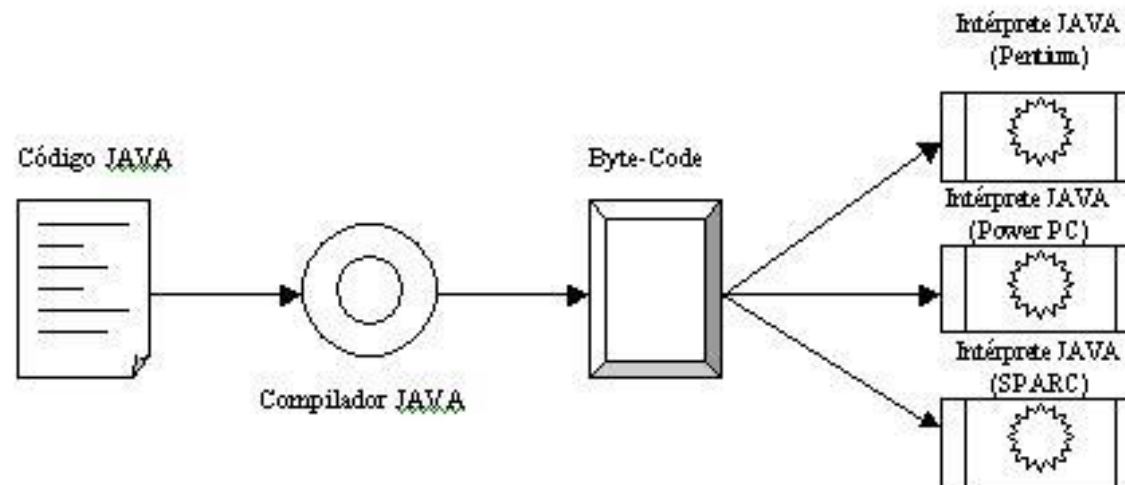


Unidad 1: Introducción a Java

Java incluye dos elementos:

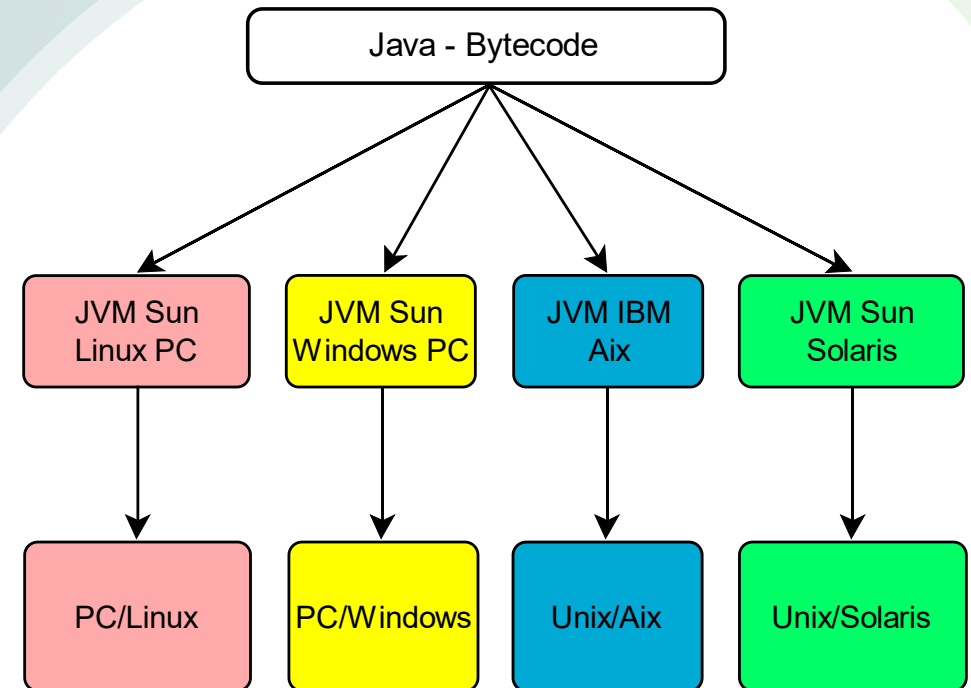
Compilador: traduce a un código intermedio denominado **bytecode** que se almacena en un fichero (Lenguaje Precompilado).

Intérprete: utiliza las clases compiladas y las ejecuta. Es lo que se conoce como **Máquina Virtual** (Lenguaje interpretado).



Máquina virtual de Java

Una **máquina virtual Java** (en inglés Java Virtual Machine, **JVM**) es una máquina virtual de proceso nativo, es decir, ejecutable en una plataforma específica, capaz de interpretar y ejecutar instrucciones expresadas en un código binario especial (el bytecode Java), el cual es generado por el compilador del lenguaje Java.



Unidad 1: Introducción a Java

Características de Java

Lenguaje simple: resulta sencillo y rápido el aprendizaje desde el principio.

Orientado a objetos: agrupa información de forma estructurada. Además, incorpora mecanismos que facilitan la extensibilidad y reutilización del código: herencia, polimorfismo, ...

Robusto: proporciona numerosas comprobaciones en compilación y en tiempo de ejecución. Además, la recolección de basura elimina la necesidad de liberación explícita de memoria.

Seguro: tanto el lenguaje como el sistema de ejecución son en tiempo real, no afectan al terminal sobre el que se ejecuta el código.

Unidad 1: Introducción a Java

Características de Java

Distribuido: proporciona una colección de clases para su uso en aplicaciones de red, por ejemplo: RMI, CORBA.

Interpretado: su código fuente se transforma en un código llamado bytecodes.

Indiferente a la arquitectura: el compilador de Java genera bytecodes: un formato intermedio indiferente a la arquitectura, diseñado para transportar el código eficientemente a múltiples plataformas hardware y software.

Portable: la indiferencia a la arquitectura representa sólo una parte de su portabilidad. Además, Java especifica los tamaños de sus tipos de datos básicos y el comportamiento de sus operadores aritméticos, de manera que los programas son iguales en todas las plataformas.

** Estas dos últimas características se deben a la Máquina Virtual Java (JVM).*

Unidad 1: Introducción a Java

Características de Java

Multitarea: soporta sincronización de múltiples hilos de ejecución a nivel de lenguaje, especialmente útiles en la creación de aplicaciones de red distribuidas.

Dinámico: el lenguaje Java y su sistema de ejecución en tiempo real son dinámicos en la fase de enlazado, ya que las clases son creadas a medida que son necesitadas.

Unidad 1: Introducción a Java

¿Qué es una aplicación Java?

Es un **conjunto de clases** que desempeñan una función específica.

Los **pasos** para crear una aplicación Java son:

Editar: se generan archivos .java

Compilar: se generan archivos .class

Ejecutar: se obtiene la salida de la aplicación.

Entornos de Desarrollo Integrado (IDE)

Unidad 1: Introducción a Java

Eclipse: Última versión Eclipse 2024-06 (posiblemente 2024-09)



The screenshot shows the Eclipse IDE interface with the following components:

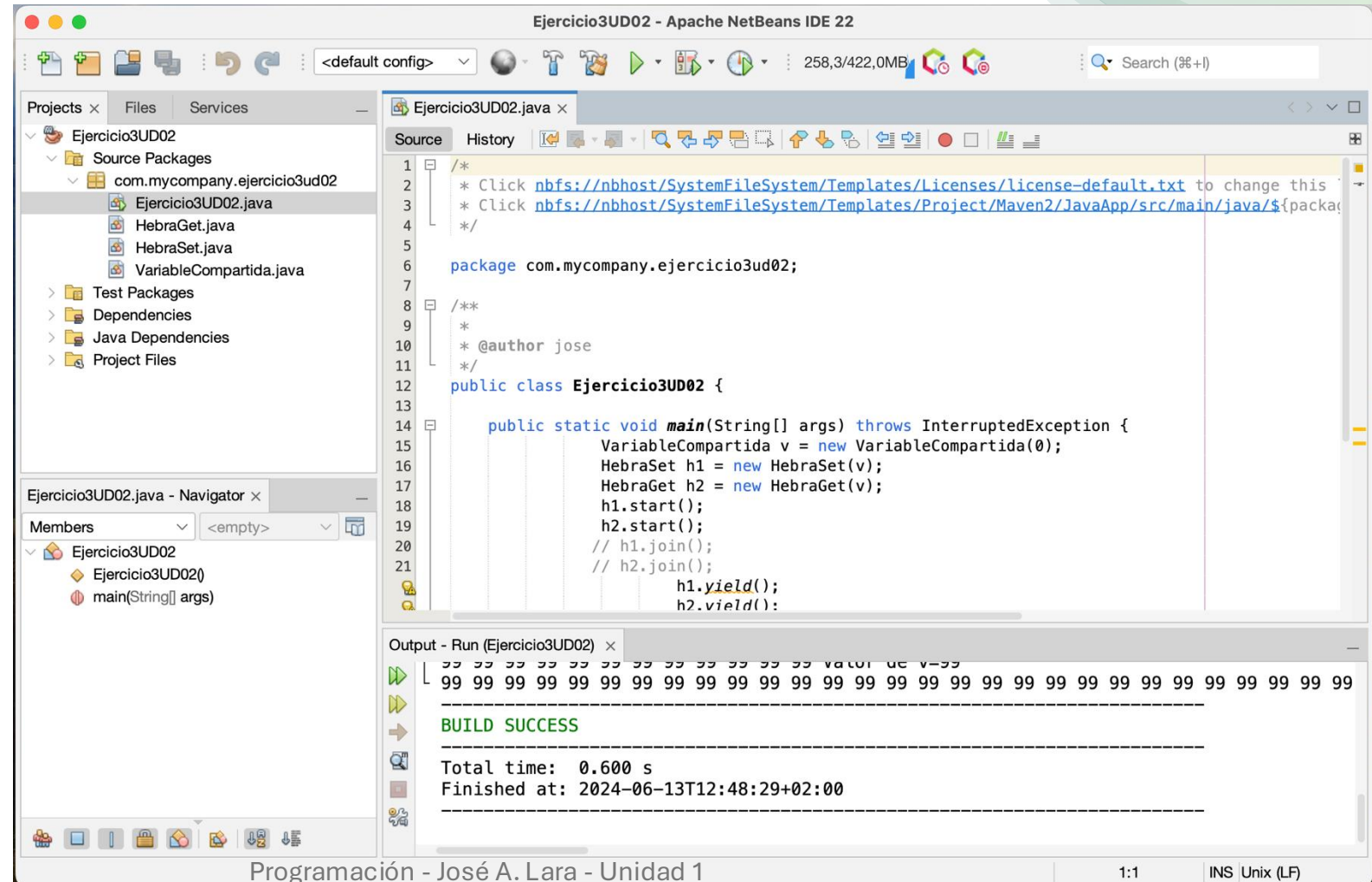
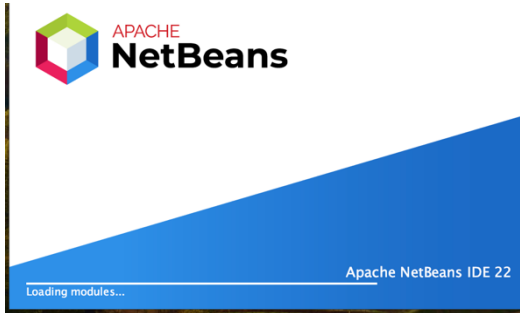
- Project Explorer:** Lists a project named "EjemplosString" containing a "src" folder with a "default package" containing files Ejemplo01.java through Ejemplo11.java, EjemploReto.java, EjemploString_Strir.java, Ejercicio4.java, and Reto.java. Other projects like EjemplosConjuntos, EjemplosListas, EjemplosMaps, EjemplosPOOJava, EjemplosRelacionesObjeto, EjemploSwing, and Ejercicio1 through Ejercicio3 are also visible.
- Editor:** Displays the code for Ejemplo11.java. The code implements a Caesar cipher with methods `convierteLetra` and `conviertePalabra`, and a `main` method that processes the input string "ABCDEFGHIJKLMNOPQRSTUVWXYZ".
- Console:** Shows the output of the program: "BCDEFGHIJKLMNOPQRSTUVWXYZA".

```
1 public class Ejemplo11 {
2
3     private char convierteLetra(int letra) {
4         int rango = 'Z' - 'A' + 1;
5         int siguiente = (letra - 'A') + 1;
6         int siguienteCircular = siguiente % rango;
7         char res = (char) (siguienteCircular + 'A');
8         return res;
9     }
10
11     private String conviertePalabra(String palabra) {
12         String res = "";
13         for (int i = 0; i < palabra.length(); i++)
14             res += convierteLetra(palabra.charAt(i));
15         return res;
16     }
17
18     public static void main(String[] args) {
19         // Entrada e inicialización
20         String frase = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";
21         String fraseCodificada = "";
22         Ejemplo11 programa = new Ejemplo11();
23
24         // Proceso
25         String[] palabras = frase.split(" ");
26         for (int i = 0; i < palabras.length; i++) {
27             fraseCodificada += programa.conviertePalabra(palabras[i]);
28             if (i < palabras.length - 1)
29                 fraseCodificada += " ";
30         }
31
32         // Salida
33         System.out.println(fraseCodificada);
34     }
35 }
36
```

Programación - José A. Lara - Unidad 1

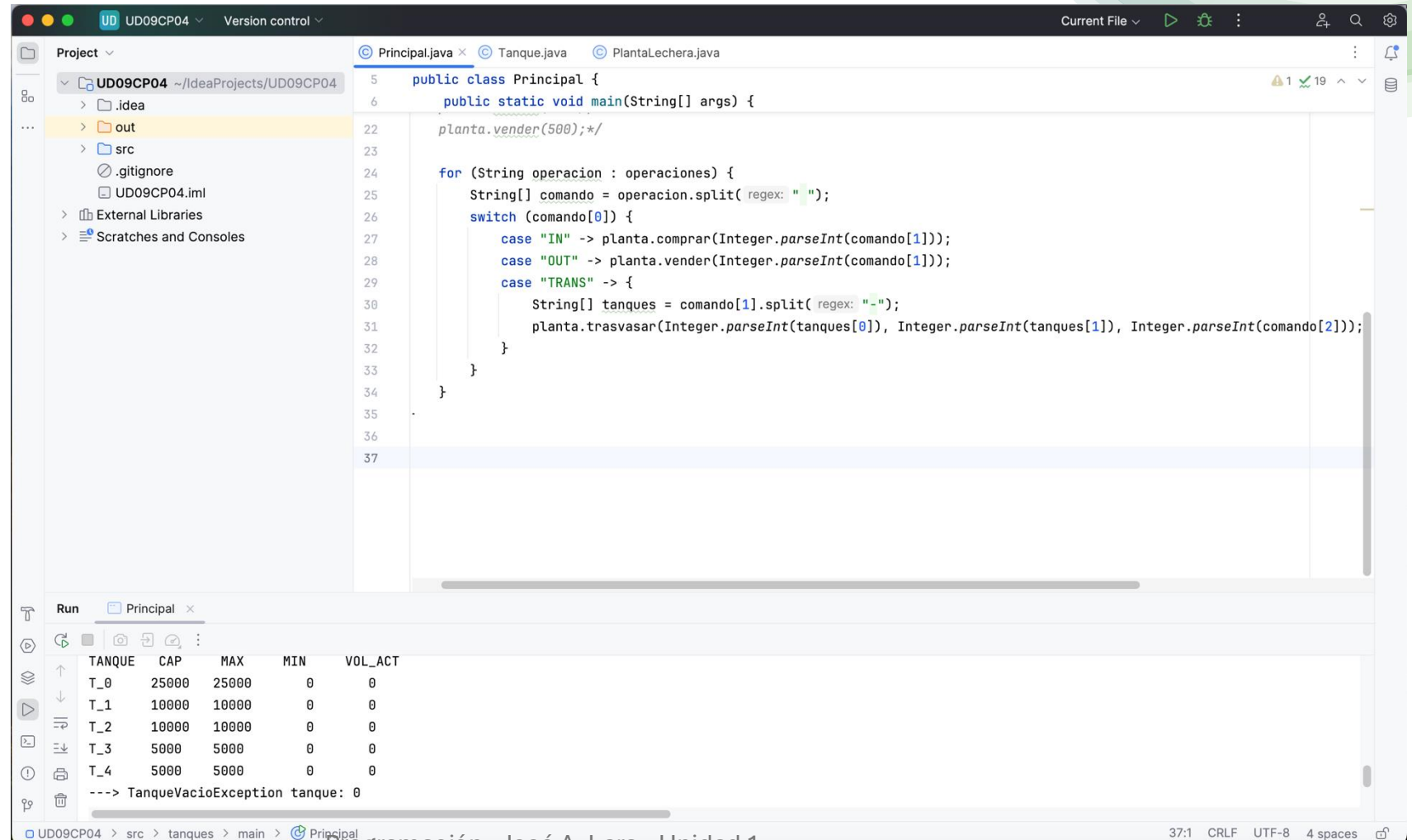
Unidad 1: Introducción a Java

NetBeans: última versión Apache NetBeans 22



Unidad 1: Introducción a Java

IntelliJ IDEA: última versión 2024.1.3

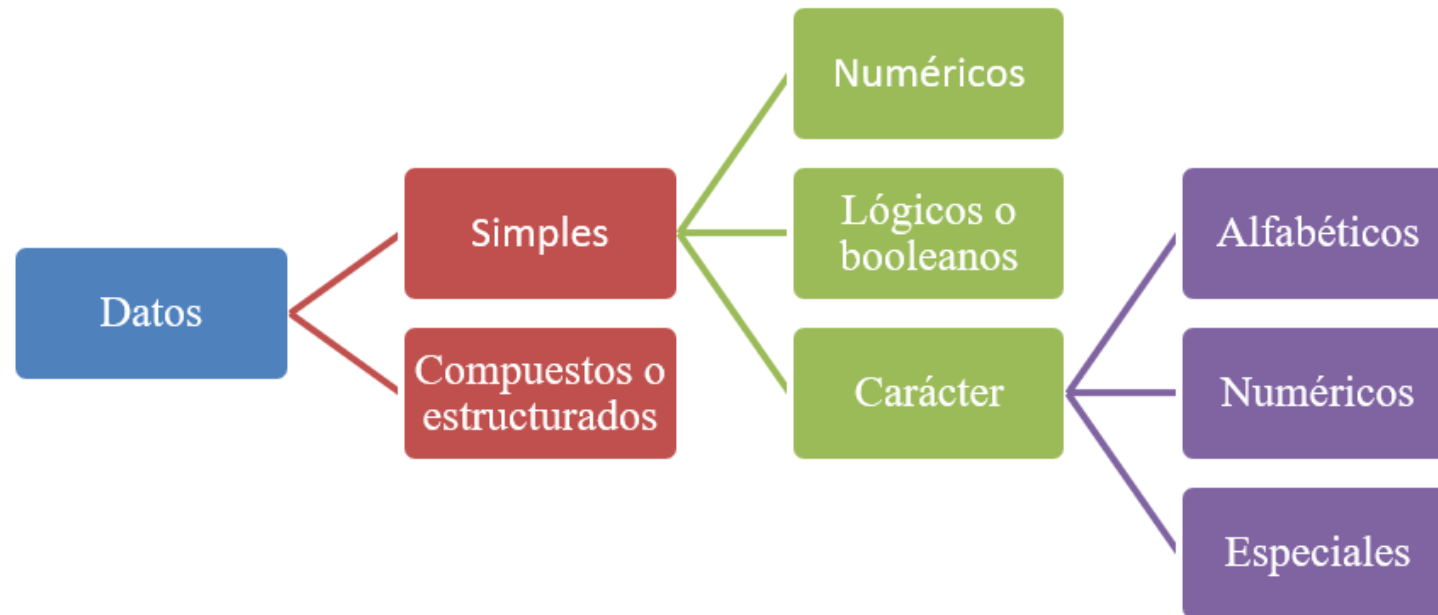


Datos y tipos de datos

Unidad 1: Introducción a Java

Datos y tipos de datos

El primer objetivo de todo ordenador es el manejo de datos. Un **dato** es la expresión general que describe los objetos con los cuales opera un ordenador. La mayoría de los ordenadores pueden trabajar con varios tipos de datos. Estos datos se suministrarán al programa, el cual los procesará y transformará en datos de salida o información.



Constantes y variables. Expresiones

Unidad 1: Introducción a Java

Constantes y variables. Expresiones

Los programas de ordenador contienen ciertos valores que **no deben cambiar** durante la ejecución del programa, estos valores se denominan **constantes**.

Hay otros valores que **cambian** durante la ejecución del programa, para almacenar dichos valores se utilizan **variables**.

En ambos casos lo que estamos haciendo es reservar un espacio en memoria en el cual se almacenarán el valor que tome la variable o el valor de la constante para cada uno de los identificadores definidos.

Unidad 1: Introducción a Java

Constantes y variables. Expresiones

Una expresión es una combinación de constantes, variables, símbolos de operaciones, paréntesis y nombres de funciones especiales, de cuya evaluación se obtiene un único resultado. Dependiendo del resultado obtenido en la evaluación de una expresión se puede hablar de **expresiones aritméticas** y **expresiones lógicas**.

$$z = 2x^2 \left(\frac{4x}{y} + \frac{9-x}{8x^2y} \right)$$

$\sqrt{\frac{3x}{2x}}$

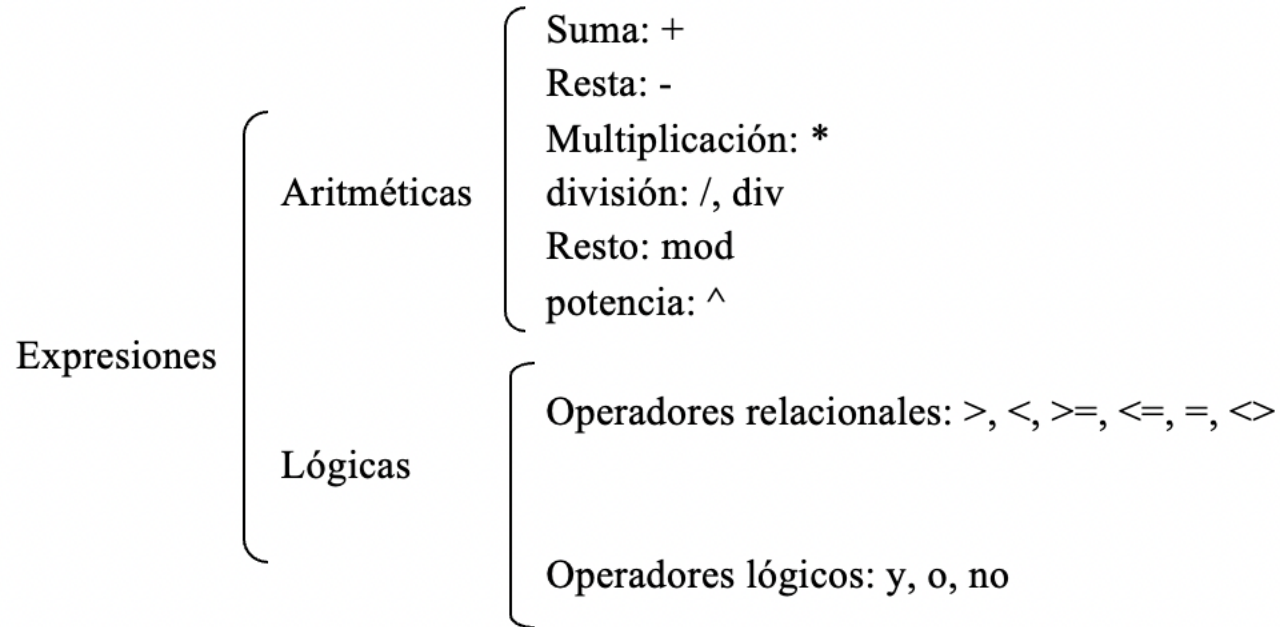
$$w = \frac{x+y}{x} + \frac{2x}{x-2}$$

$$y = (x-2)(x^2+2)$$

Unidad 1: Introducción a Java

Constantes y variables. Expresiones

- **Expresión aritmética:** es aquella compuesta por operaciones matemáticas



Unidad 1: Introducción a Java

Constantes y variables. Expresiones

- **Expresión lógica:** el resultado de su evaluación es un valor lógico o booleano. Pueden usar operadores relacionales u operadores lógicos.
 - Un operador **Y (AND)** devuelve V (true) sólo cuando los 2 operandos sean verdaderos.

Operando 1	Operando 2	Resultado
V	V	V
V	F	F
F	V	F
F	F	F

Unidad 1: Introducción a Java

Constantes y variables. Expresiones

- **Expresión lógica**

- Un operador **O** (OR) devuelve V (true) siempre que algún operando sea verdadero.

Operando 1	Operando 2	Resultado
V	V	V
V	F	V
F	V	V
F	F	F

- Operador **NO** (NOT) es un operador unario (sólo 1 operando). Devuelve el valor opuesto al operador

Operando	Resultado
V	F
F	V

Entrada y salida de información

Unidad 1: Introducción a Java

Entrada y salida de información

Los cálculos que realizan los ordenadores requieren, para ser útiles, la **entrada de los datos** necesarios para ejecutar las operaciones que posteriormente se convertirán en resultados o salidas.

Las **operaciones de entrada** permiten leer determinados valores introducidos por el usuario y asignarlos a determinadas variables.

Esta **entrada** se conoce como **operación de lectura o entrada**.

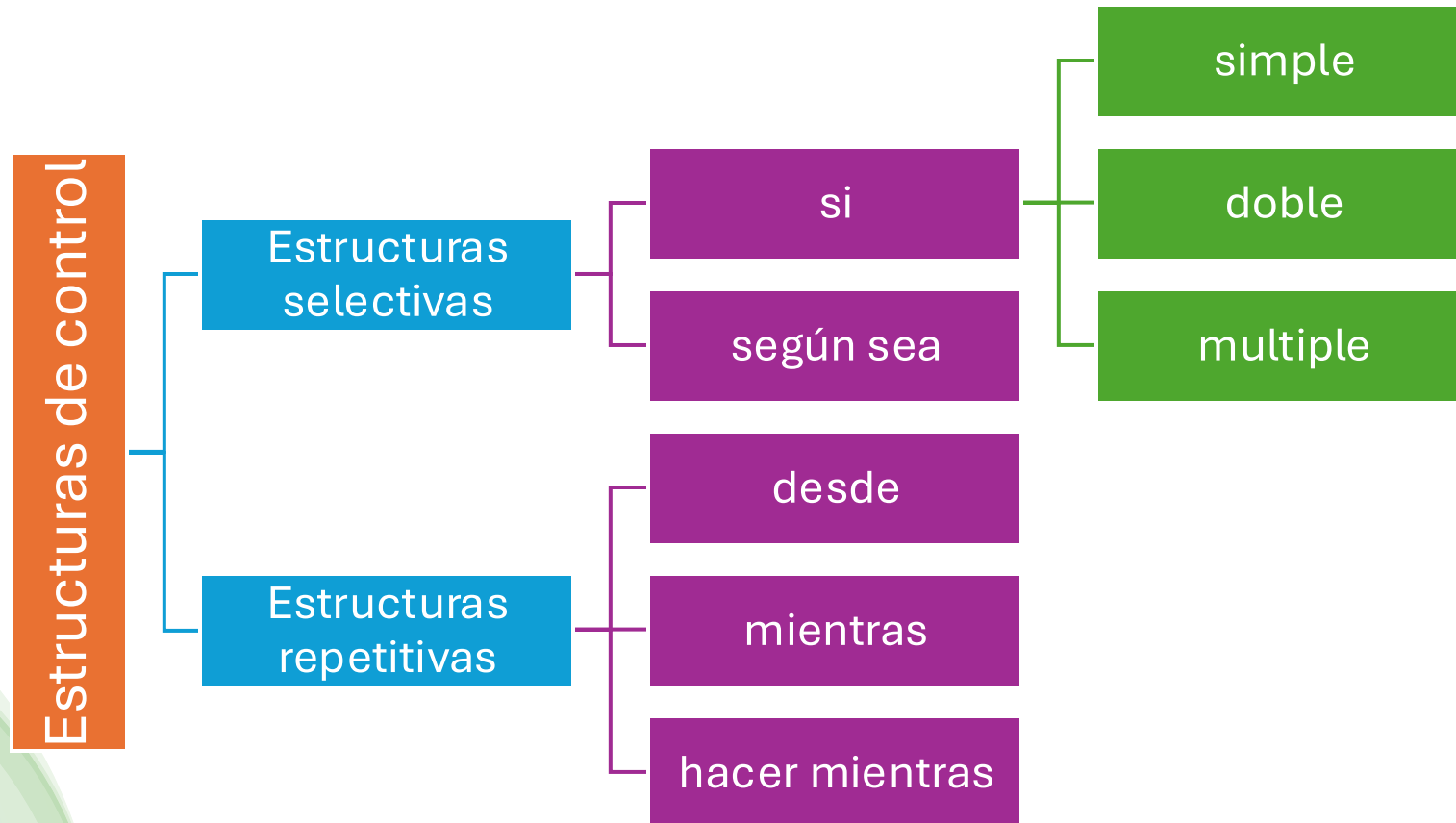
Con los datos de entrada, el ordenador realiza las operaciones oportunas en cada momento y el resultado obtenido aparece en un **dispositivo de salida** mediante una **operación de salida o escritura**.

Estructuras de control

Unidad 1: Introducción a Java

Estructuras de control

Se pueden encontrar dos tipos principales de estructuras de control

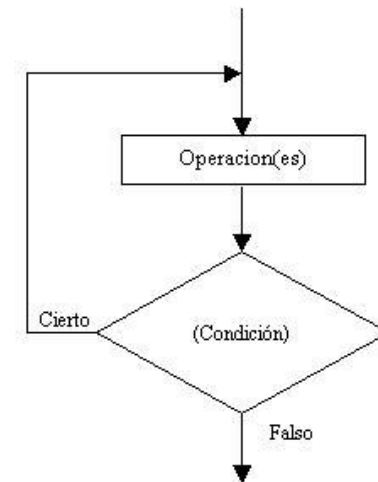
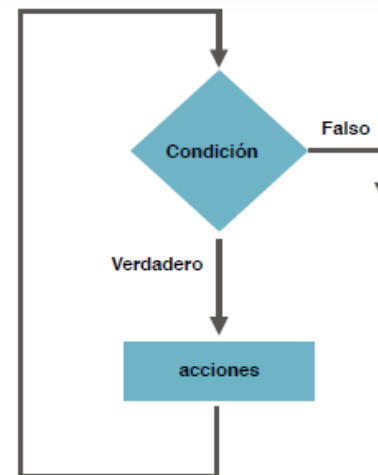
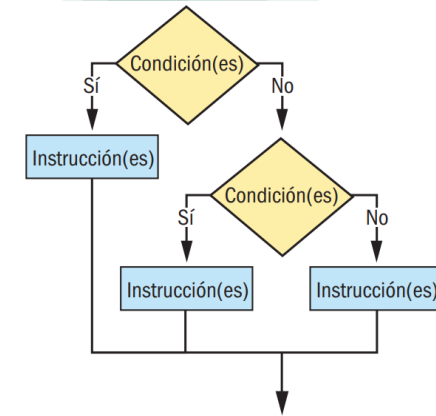
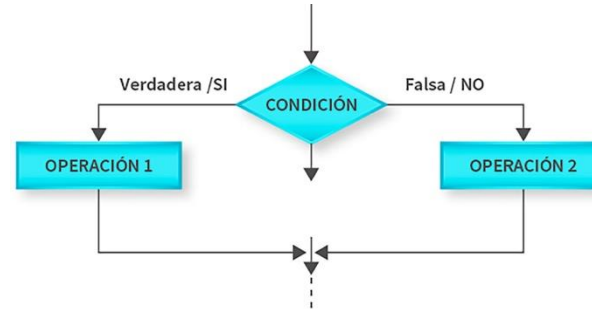


Unidad 1: Introducción a Java

Estructuras de control

Las **estructuras selectivas** son las que permiten ejecutar una serie de instrucciones u otra dependiendo del valor de una expresión booleana o del valor de una variable numérica, de carácter o de texto.

Las **estructuras repetitivas** son las que permiten ejecutar una serie de instrucciones un número de veces indeterminado que va a depender de una expresión booleana, o bien un número predeterminado de veces.



Unidad 1: Introducción a Java

Estructuras de control

