

# Guía de Estudio para Laboratorio 1:

## Introducción a Java

Instrucciones Generales:

Desarrollar los siguientes temas en hojas tamaño carta con líneas, escritos a mano, Se puede agregar ilustraciones de tamaño no más de ¼ de carta. Incluir carátula informativa, índice, separadores de temas creativos. Letra clara de molde, cuidar ortografía. Cuando se escriba en lenguaje de programación utilizar un lapicero de diferente color al del resto del documento. Se evaluarán: 1. Contenido y Estructura, 2. Claridad y Calidad, 3. Formato. Debe entregarse en un folder tamaño carta color rojo con gancho. En la actividad complementaria deberá venir el programa en una USB listo para ejecutar. Se tomará en cuenta: 1. Si el programa compila y se ejecuta correctamente. 2. Las notas adicionales explicando cada parte del programa. La lista de cotejo deberá ser colocada en el folder en la parte interna.

---

### 1. Fundamentos y Características de Java

- Historia, creador y lema de Java ("Write Once, Run Anywhere" - WORA).
- Principales características: **orientado a objetos**, portable, robusto, seguro y de alto rendimiento.
- Diferencia entre **JDK (Java Development Kit)**, **JRE (Java Runtime Environment)** y **JVM (Java Virtual Machine)**.

### 2. Configuración del Entorno de Desarrollo

- Pasos para la instalación del JDK.
- Configuración de la variable de entorno `PATH`.
- Introducción a los **Entornos de Desarrollo Integrados (IDE)** populares como **NetBeans** o **IntelliJ IDEA**.
- Creación y ejecución del primer programa simple ("Hola Mundo").

### 3. Estructura Básica de un Programa Java

- Concepto de **clase** (`class`) como unidad fundamental.
- Función del método principal (`public static void main(String[] args)`).
- Uso de **comentarios** (línea y bloque) y **sentencias** (finalizadas con punto y coma).
- Manejo de la salida por consola (`System.out.println()`).

### 4. Variables, Tipos de Datos y Operadores

- Definición y declaración de **variables**.
- Tipos de datos **primitivos** (ej: `int`, `double`, `boolean`, `char`) y sus rangos.
- Tipos de datos de **referencia** (ej: `String`).
- **Operadores** aritméticos (+, -, \*, /, %), de asignación (=, +=), relacionales (==, !=, <, >) y lógicos (&&, ||, !).

### 5. Estructuras de Control de Flujo

- **Sentencias condicionales:**
  - `if-else` simple y anidado.
  - `switch` para múltiples opciones.
- **Sentencias repetitivas (Ciclos):**
  - `for` (estructura básica y *for-each*).
  - `while` y `do-while`.
- Uso de `break` y `continue` en ciclos.

## 6. Introducción a la Programación Orientada a Objetos (POO)

- Conceptos clave de la POO: **Objeto, Clase, Atributo y Método.**
- Los 4 pilares de la POO: **Encapsulamiento, Herencia, Polimorfismo y Abstracción** (definiciones básicas).
- Declaración de una clase sencilla con atributos y métodos.
- Creación de **objetos** (instanciación) con la palabra clave `new`.

### **Actividad Complementaria: "Mi Primer Proyecto Java"**

Esta actividad tiene como objetivo reforzar los 6 subtemas de la guía de estudio mediante una aplicación práctica, consolidando especialmente la configuración del entorno y los conceptos iniciales de la **Programación Orientada a Objetos (POO)**.

### **Programa Java: "Calculadora de Edad y Mayoría de Edad"**

Este programa pedirá al usuario su año de nacimiento y calculará su edad aproximada, indicando si ya alcanzó la mayoría de edad (considerando 18 años).

```
import java.util.Scanner; // 1. Importar la clase Scanner para entrada de datos.

/**
 * Clase principal del programa: CalculadoraEdad.
 * Demuestra la entrada de datos, variables y la sentencia if-else.
 */
public class CalculadoraEdad {

    public static void main(String[] args) {

        // 2. Definición de constantes y variables
        final int ANIO_ACTUAL = 2025; // 'final' indica que es una constante (no cambia)
        final int MAYORIA_EDAD = 18;
        int anioNacimiento;
        int edad;

        // 3. Crear el objeto Scanner para leer la entrada del usuario
        Scanner entrada = new Scanner(System.in);

        // 4. Salida de datos (pedir información al usuario)
        System.out.println("--- Calculadora de Edad y Mayoría de Edad ---");
        System.out.print("Por favor, introduce tu año de nacimiento (Ej: 2005): ");

        // 5. Entrada de datos (leer el valor ingresado por el usuario)
        anioNacimiento = entrada.nextInt();

        // 6. Procesamiento (cálculo de la edad)
        edad = ANIO_ACTUAL - anioNacimiento;

        // 7. Salida de datos (mostrar el resultado)
        System.out.println("\nTu edad aproximada es: " + edad + " años.");

        // 8. Sentencia Condicional (verificar la mayoría de edad)
        if (edad >= MAYORIA_EDAD) {
            System.out.println("¡Felicitaciones! Ya eres mayor de edad (>= " + MAYORIA_EDAD + " años).");
        }
    }
}
```

```
} else {
    // El 'else' se ejecuta si la condición anterior es falsa
    System.out.println("Aún eres menor de edad (menor de " + MAYORIA_EDAD + " años.");
}

// 9. Cerrar el objeto Scanner
entrada.close();
}
}
```

## Explicación Paso a Paso del Proceso

El proceso se divide en tres etapas: **Preparación, Codificación y Ejecución.**

Etapa I: Preparación del Entorno (Subtema 2 y 3)

1. **Asegurar el JDK:** Verifica que tienes instalado el **JDK (Java Development Kit)**. Esto te permite compilar y ejecutar programas.
2. **Abrir el IDE:** Inicia tu **Entorno de Desarrollo Integrado (IDE)**, como **NetBeans** o **IntelliJ IDEA**.
3. **Crear el Proyecto:** Ve a **File -> New Project** y selecciona **Java Application** o similar. Nómbralo, por ejemplo, **IntroduccionJava**.
4. **Crear la Clase Principal:** El IDE automáticamente creará una clase con el método `main`. Si no lo hace, crea un archivo llamado `CalculadoraEdad.java` y asegúrate de que el nombre de la clase (`public class CalculadoraEdad`) coincida con el nombre del archivo.

Etapa II: Codificación (Subtemas 3, 4 y 5)

Abre el archivo `CalculadoraEdad.java` y sigue los pasos comentados en el código:

1. **Importar la Clase `Scanner`:**
  - o **Línea:** `import java.util.Scanner;`
  - o **Explicación:** Java organiza su funcionalidad en paquetes. Para leer datos que el usuario ingresa por el teclado, necesitamos la herramienta llamada `Scanner`, que está en el paquete `java.util`. La sentencia `import` la trae a nuestro código.
2. **Estructura y Variables:**
  - o **Líneas:** Declaración de `public class CalculadoraEdad` y el método `public static void main(String[] args)`.
  - o **Explicación:** Esta es la estructura fundamental de cualquier programa Java. Dentro de `main` declaramos las **variables** y **constantes**:
    - `final int ANIO_ACTUAL = 2025;` Usamos el tipo **primitivo** `int` para enteros y `final` para una constante que no cambiará.
    - `int anioNacimiento;` y `int edad;` Variables de tipo **entero** que almacenarán datos.
3. **Crear el Objeto `Scanner`:**
  - o **Línea:** `Scanner entrada = new Scanner(System.in);`
  - o **Explicación:** Aquí se aplica la **Programación Orientada a Objetos (POO)**. Usamos la palabra clave `new` para crear un **objeto** llamado `entrada` de la clase `Scanner`. Este objeto está listo para leer datos (`System.in` significa "entrada del sistema", el teclado).
4. **Entrada y Salida de Datos:**
  - o **Línea de Salida (pedir datos):** `System.out.print(...)`
  - o **Línea de Entrada (leer datos):** `anioNacimiento = entrada.nextInt();`
  - o **Explicación:** Imprimimos un mensaje en la consola (`System.out.print`). Luego, el método `nextInt()` del objeto `entrada` pausa el programa, espera que el usuario escriba un número y presione **Enter**, y guarda ese número en la variable `anioNacimiento`.
5. **Procesamiento y Sentencia Condicional:**
  - o **Línea de Cálculo:** `edad = ANIO_ACTUAL - anioNacimiento;`

- **Estructura Condicional:** `if (edad >= MAYORIA_EDAD) { ... } else { ... }`
- **Explicación:** Usamos el **operador aritmético** de resta (-). Luego, la estructura de **control de flujo** `if-else` verifica la condición: si la `edad` es **mayor o igual (>=)** a 18. Si es `true`, ejecuta el bloque `if`; si es `false`, ejecuta el bloque `else`.

### Etapa III: Compilación y Ejecución

#### 1. **Compilación:**

- El IDE automáticamente compila el archivo (`.java`) a **Bytecode** (`.class`).
- *Si lo hicieras manualmente en la terminal:* Usarías el comando `javac CalculadoraEdad.java`.

#### 2. **Ejecución:**

- En el IDE, simplemente presiona el botón "Run" (un triángulo verde).
- *Si lo hicieras manualmente en la terminal:* Usarías el comando `java CalculadoraEdad`.

#### 3. **Resultado:** La **JVM** lee el Bytecode y comienza la ejecución, pidiéndote el año de nacimiento y mostrando la edad y la conclusión sobre la mayoría de edad.

#### 4. **Cierre:** La línea `entrada.close()`; libera los recursos del `Scanner`.

# LISTA DE COTEJO

No.	Descripción del aspecto a calificar	Valor Máximo Asignado	Valor Obtenido
<b>I. Contenido y Estructura</b>			
<b>1</b>	Incluye el <b>Título Principal</b> , Curso y Carrera.	2	
<b>2</b>	Presenta <b>6 Subtemas</b> claramente definidos.	8	
<b>3</b>	Los subtemas son <b>relevantes</b> al tema "Introducción a Java" y al nivel.	8	
<b>4</b>	Cada subtema contiene al menos <b>3 puntos clave</b> o conceptos.	2	
<b>II. Calidad y Claridad</b>			
<b>5</b>	El lenguaje es <b>claro</b> , preciso y adecuado para el nivel de Bachillerato.	3	
<b>6</b>	La información está <b>organizada</b> de forma lógica y secuencial (de básico a avanzado).	3	
<b>7</b>	Utiliza <b>negritas</b> para resaltar términos clave.	2	
<b>III. Formato</b>			
<b>8</b>	El formato es limpio y facilita la lectura.	2	
<b>IV. Actividad Complementaria</b>			
<b>9</b>	El programa compila y se ejecuta correctamente	7	
<b>10</b>	Las notas adicionales explican cada parte del programa	3	
<b>TOTAL</b>		<b>40 puntos</b>	