



## Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorios de docencia

# Laboratorio de Computación Salas A y B

*Profesor(a):* Calvillo Pérez Juan Ángel

*Asignatura:* Programación Orientada a Objetos

*Grupo:* 09

*No de Práctica(s):* 02

*Integrante(s):* Durán Rendón Santiago

*No. de lista o  
brigada:* 07

*Semestre:* 2026-1

*Fecha de entrega:* 04 de septiembre de 2025

*Observaciones:*

CALIFICACIÓN: \_\_\_\_\_

# Índice

<b>Objetivo .....</b>	<b>1</b>
<b>Introducción.....</b>	<b>1</b>
<b>Desarrollo.....</b>	<b>2</b>
<b>Resultados.....</b>	<b>7</b>
<b>Conclusión.....</b>	<b>11</b>
<b>Referencias .....</b>	<b>11</b>
<b>Recurso adicional.....</b>	<b>11</b>

## Objetivo

Crear programas que implementen variables y constantes de diferentes tipos de datos, expresiones y estructuras de control de flujo.

## Introducción

En el lenguaje de programación Java hay reglas muy estrictas sobre la forma en que se organiza el código fuente para poder tener claridad y estructura dentro de los programas. Una de estas reglas es que el archivo en el que se tiene que escribir el código fuente debe tener el mismo nombre de la clase pública que contiene, ya que el compilador de Java (javac) establece una relación uno a uno entre el nombre de la clase y el nombre del archivo.

En el caso de que el nombre de una clase pública no coincida con el nombre del archivo del que se habla, el compilador genera un error, ya que no puede identificar correctamente cuál es el punto de entrada o el lugar donde empieza la definición de la clase. Este mecanismo es el que garantiza una mayor organización dentro de los proyectos, evita que haya ambigüedades y hace más sencilla la comprensión de cada archivo, que debe tener la forma de una entidad bien definida.

Así, cuando se crean programas en Java con el modelo orientado a objetos, el hecho de usar siempre archivos fuente que tengan el mismo nombre que la clase ofrece la posibilidad de que la definición de las clases sea compilada adecuadamente y que los proyectos sigan una cierta estandarización, además de que se logre la comprensión del código por parte de otros programadores.

## Desarrollo

1. Tipos de datos y declaración de variables (int, double, boolean, char, String y comparación):

- a. Entrada:

```
public static void main(String args[]) {  
  
    int edad;  
    edad = 10;  
    double pi = 3.1416;  
    boolean bool = true;  
    char unCaracter = 'a';  
    //Integer myInteger = 5;  
    String cadena = "daniel va reprobar";  
    String cadena2 = "yael tambien va a reprobar";  
    System.out.println("hola mundo");  
    System.out.println(edad);  
    System.out.println(pi);  
    System.out.println(bool);  
    System.out.println(unCaracter);  
    System.out.println(cadena);  
    System.out.println(cadena2);  
  
    String alumno1 = "Gustavo";  
    String alumno2 = "gustavo";  
    String alumno3 = "Gustavo ";  
    String alumno4 = "Gustavo";  
    System.out.println(alumno1.equals(alumno2));  
    System.out.println(alumno1.equals(alumno3));  
    System.out.println(alumno3.equals(alumno2));  
    System.out.println(alumno1.equals(alumno4));  
}
```

- b. Salida:

```

hola munde
10
3.1416
true
a
daniel va reprobar
yael tambien va a reprobar
false
false
false
true

```

## 2. Bucles (for, else if, while, do while):

### a. Entrada:

```

if (edad >= 18){
    System.out.println("ponte a chambear");
} else {
    System.out.println("Eres casi un adulto chiquito");
}

float salario = 22000;

if (salario < 8000){
    System.out.println("Ni a chalan llega");
} else if (salario < 16000){
    System.out.println("Sigues casi igual");
} else {
    System.out.println("Sigue asi a los 40 te vas a independizar");
}

int i = 0;
for(; i<7; i++){ //1 += 2 para aumentar de 2 en 2
    System.out.println("alo?");
}

int inicio = 0;
int fin = 10;
while(inicio < fin){
    System.out.println("hola?");
    inicio++;
    fin -= 2; //fin = fin - 2
}

int num1 = 10;
int num2 = 9;
do {
    System.out.println("Entro?");
} while (num1 < num2);

```

### b. Salida:

```
Eres casi un adulto chiquito
Sigue asi a los 40 te vas a independizar
alo?
alo?
alo?
alo?
alo?
alo?
alo?
alo?
hola?
hola?
hola?
hola?
Entro?
```

## 3. Arrays:

### a. Entrada:

```
/* ARRAYS O ARREGLOS*/
int[] miArreglo; //Declaración
miArreglo = new int[5]; //Inicialización
miArreglo[0] = 4;
miArreglo[1] = 8;
miArreglo[2] = 8;
miArreglo[3] = 5;
miArreglo[4] = 9;

int[] miArreglo2 = {4, 8, 8, 5, 9};
System.out.println(miArreglo[3]);
System.out.println(miArreglo2[4]);

// OJO: En un array, una vez establecido el tamaño no puede ser alterado.

// ¿Cómo recorrer un array?

for (int indice = 0; indice < miArreglo.length; indice++){
    System.out.println(miArreglo[indice]);
}

System.out.println(miArreglo[miArreglo.length - 1]);
```

### b. Salida:

5  
9  
4  
8  
8  
5  
9  
9

#### 4. Void y llamado:

##### a. Entrada:

```
Saludo();  
Saludo();  
Saludo();  
  
saludo("Elena de troya");  
  
}  
  
/*  
Permisos funcion -> public, private, protected.  
static -> De clase (Detalle en OOP).  
Tipo de valor de retorno -> String, int, double. No regresa nada: void  
nombre de función  
[argumentos]  
*/  
public static void saludo() {  
    System.out.println(";Hola mundo!");  
}  
  
/* SOBRECARGA */  
public static void saludo(String nombre) {  
    System.out.println("Hola "+nombre);  
}
```

##### b. Salida:

◆Hola mundo!  
◆Hola mundo!  
◆Hola mundo!

##### c. Hola Elena de troya

#### 5. Biblioteca Scanner:

##### a. Entrada:

```

// Importar fuera de las clases
import java.util.Scanner; //Entrada y salida de datos

public class bibliotecaScanner {

    public static void main(String[] args){
        Scanner sc = new Scanner(System.in);

        System.out.println("Menú de opciones:");
        System.out.println("1. Saludar");
        System.out.println("2. Mostrar la fecha");
        System.out.println("3. Calcular el doble de un número");
        System.out.println("4. Salir");
        System.out.println("Elige una opcion:");

        int option = sc.nextInt(); // Input gonna be interger

        switch(option) {
            case 1:
                System.out.println("¿Cómo te llamas?");
                sc.nextLine(); // clean buffer ??
                String nombre = sc.nextLine();
                System.out.println("Hola, " + nombre + ":");
                break;
            case 2:
                java.util.Date fecha = new java.util.Date(); // call date
                System.out.println("La fecha actual es: " + fecha);
                break;
            case 3:
                System.out.println("Inserte un número: ");
                int num = sc.nextInt();
                System.out.println("El doble de " + num + " es " + (num * 2));
                break;
            case 4:
                System.out.println("Saliendo del programa...");
                break;
            default:
                System.out.println("Opción no válida");
        }
        sc.close();
    }
}

```

## b. Salida:

```

Menú de opciones:
1. Saludar
2. Mostrar la fecha
3. Calcular el doble de un número
4. Salir
Elige una opcion:
1
¿Cómo te llamas?

```

# Resultados

## 1. FOR:

### a. Entrada:

```
public static void main(String args[]) {  
    for(int i = 0; i<7; i++){ //1 += 2 para aumentar de 2 en 2  
        System.out.println("alo? "+i);  
    }  
}
```

### b. Salida:

```
alo? 0  
alo? 1  
alo? 2  
alo? 3  
alo? 4  
alo? 5  
alo? 6
```

## 2. While:

### a. Entrada:

```
int inicio = 0;  
int fin = 10;  
while(inicio < fin){  
    System.out.println("hola? "+fin);  
    inicio++;  
    fin -= 2; //fin = fin - 2  
}
```

### b. Salida:

```
hola? 10  
hola? 8  
hola? 6  
hola? 4
```

## 3. Do while:

### a. Entrada:



```

int num1 = 0;
int num2 = 9;
int count = 0;
do {
    count++;
    num1++;
    System.out.println("Alo "+count);
} while (num1 < num2);

```

b. Salida:

```

Alo 1
Alo 2
Alo 3
Alo 4
Alo 5
Alo 6
Alo 7
Alo 8
Alo 9

```

4. IF:

a. Entrada:

```

// IFs
boolean casado = true;
boolean soltero = true;
int hijos = 1;
if (casado && hijos >= 1) {
    System.out.println("Huye!!!");
}
if (casado && hijos >= 1) {
    System.out.println("Quien tenga miedo a morir que no nazca");
}
if ((!casado && soltero) && hijos >= 1) {
    System.out.println("No puedes ni contigo, huye!!!");
}
if (soltero && hijos < 1) {
    System.out.println("No existe son los papás");
}

```

```

// IFs
boolean casado = false;
boolean soltero = true;
int hijos = 1;
if (casado && hijos >= 1) {
    System.out.println("Huye!!!");
}
if (casado && hijos >= 1) {
    System.out.println("Quien tenga miedo a morir que no nazca");
}
if ((!casado && soltero) && hijos >= 1) {
    System.out.println("No puedes ni contigo, huye!!!");
}
if (soltero && hijos < 1) {
    System.out.println("No existe son los papás");
}

// IFs
boolean casado = false;
boolean soltero = true;
int hijos = 0;
if (casado && hijos >= 1) {
    System.out.println("Huye!!!");
}
if (casado && hijos >= 1) {
    System.out.println("Quien tenga miedo a morir que no nazca");
}
if ((!casado && soltero) && hijos >= 1) {
    System.out.println("No puedes ni contigo, huye!!!");
}
if (soltero && hijos < 1) {
    System.out.println("No existe son los papás");
}

```

#### b. Salida:

```

Huye!!!
Quien tenga miedo a morir que no nazca

No puedes ni contigo, huye!!!

No existe son los papás

```

### 5. SWITCH:

#### a. Entrada:

```

// Switch case
Scanner sc = new Scanner(System.in);

System.out.println("Jalas o te rajas:");
System.out.println("1. Casado y con hijo??");
System.out.println("2. Casado y tiene hijos??");
System.out.println("3. Soltero y con hijos");
System.out.println("4. Soltero y sin hijos");
System.out.println("Elige una opción:");

int option = sc.nextInt();

switch(option) {
    case 1:
        System.out.println("Huye!!!");
        break;
    case 2:
        System.out.println("Quien tenga miedo a morir que no nazca");
        break;
    case 3:
        System.out.println("No puedes ni contigo, huye!!!");
        break;
    case 4:
        System.out.println("No existe, son los papás");
        break;
    default:
        System.out.println("Opción no válida");
}

sc.close();
}

```

b. Salida:

Jalas o te rajas:	Jalas o te rajas:
1. Casado y con hijo??	1. Casado y con hijo??
2. Casado y tiene hijos??	2. Casado y tiene hijos??
3. Soltero y con hijos	3. Soltero y con hijos
4. Soltero y sin hijos	4. Soltero y sin hijos
Elige una opción:	Elige una opción:
1	2
Huye!!!	Quien tenga miedo a morir que no nazca
Jalas o te rajas:	Jalas o te rajas:
1. Casado y con hijo??	1. Casado y con hijo??
2. Casado y tiene hijos??	2. Casado y tiene hijos??
3. Soltero y con hijos	3. Soltero y con hijos
4. Soltero y sin hijos	4. Soltero y sin hijos
Elige una opción:	Elige una opción:
3	4
No puedes ni contigo, huye!!!	No existe, son los papás

## Conclusión

La práctica me ayudó a entender el manejo de variables, constantes, estructuras de control y bibliotecas en Java. Sin embargo, no comprendí bien algunas cuestiones, en especial en el vídeo, en la parte que se menciona vaciar el búfer, porque me ha dejado confundido a la hora de manejar entradas a través de la clase Scanner. También me costó entrar en el aspecto de cómo utilizar directamente la instrucción y por qué no se necesita importar nada: `java.util.Date` fecha = new `java.util.Date()`. Estas áreas representan las oportunidades de mejora y profundización necesarias para repasar el manejo de la organización del código en Java y el buen uso de sus bibliotecas.

## Referencias

- Facultad de Ingeniería UNAM. (2025). *Manual de prácticas de Programación Orientada a Objetos (MADO)*. Material de curso no publicado.

## Recurso adicional

<https://github.com/SARD82>