	Carátula para entrega de prácticas	
Facultad de Ingeniería	Laboratorios de docencia	

Laboratorio de Computación Salas A y B

Profesor(a): JOSE ANTONIO AYALA BARBOSA

Asignatura: Programación Orientada a Objetos

Grupo: 02

No de Práctica(s): 02 Fundamentos y sintaxis del lenguaje.

Integrante(s): Hernández Reyes Rebeca Sarai

No. de lista o brigada:

Semestre: 2026-1

Fecha de entrega: 03 – 09 – 2025

Observaciones:

CALIFICACIÓN: _____

PRÁCTICA 2 Fundamentos y sintaxis del lenguaje.

PROGRAMACIÓN ORIENTADA A OBJETOS

Hernández Reyes Rebeca Sarai

OBJETIVO

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

INTRODUCCIÓN

En esta práctica se busca reforzar el entendimiento de los fundamentos de la programación orientada a objetos mediante la creación de variables y constantes de distintos tipos de datos, así como el uso de expresiones con operadores y declaraciones básicas. Además, se implementarán estructuras de control de flujo que permitan la toma de decisiones y la repetición de procesos. Con ello, se pretende afianzar los conocimientos iniciales y sentar las bases para el desarrollo de programas más complejos a lo largo del curso.

ACTIVIDADES

- Crear variables y constantes de diferentes tipos de datos.
- Crear diversas expresiones (operadores, declaraciones, etc.).
- Implementar estructuras de control de flujo.

CÓDIGO FUENTE Y DESARROLLO

Se desarrolló un programa en el que se declararon variables y constantes de diferentes tipos de datos (enteros, decimales, cadenas, booleanos, entre otros), y se aplicaron expresiones mediante el uso de operadores aritméticos, relacionales y lógicos.

Posteriormente, se implementaron estructuras de control como sentencias condicionales if-else y switch, además de ciclos de repetición while, do-while y for, con el fin de observar cómo cambian los resultados según las condiciones y valores definidos. A continuación, se desarrolló el siguiente código:

```
public static void main(String[] args) {  
    System.out.println("Hola mundo <3 ");  
    System.out.println("#####if#####");  
    int a;  
    a=5;  
    int b =3;  
    if (a<b){  
        System.out.println("a es menor que b");  
    }else if (a==b){  
        System.out.println("a es igual que b");  
    }else{  
        System.out.println("a es menor que b");  
    }  
}
```

[Imagen 2. Código de "Hola Mundo".]

Finalmente, se incluyeron ejemplos prácticos con arreglos y estructuras iterativas como el ciclo for-each, permitiendo reforzar la comprensión del recorrido de colecciones de datos dentro de un programa en Java, y se desarrolló el siguiente código:

```
System.out.println("#####if en funcion / metodo#####");  
if(menor(a,b)){  
    System.out.println("metodo retorna true");  
}else{  
    System.out.println("metodo retorna false");  
}
```

[Imagen 2. If.]

```
System.out.println("#####switch#####");  
int dia = 6;  
switch (dia) {  
    case 1:  
        System.out.println("Lunes");  
        break;  
    case 2:  
        System.out.println("Martes");  
        break;  
    case 3:  
        System.out.println("Miercoles");  
        break;  
    case 4:  
        System.out.println("Jueves");  
        break;  
    case 5:  
        System.out.println("Viernes");  
        break;  
    case 6:  
        System.out.println("Sabado");  
        break;  
    case 7:  
        System.out.println("Domingo");  
        break;  
    default:  
        System.out.println("No es un dia de la semana");  
        //throw new AssertionError();  
}
```

[Imagen 3 y 4. Switch.]

```

System.out.println("##### while #####");
int n=0;
while (n<10) {
    System.out.println("Contando hacia arriba n= "+n);
    n++;
}
System.out.println("");
do {
    System.out.println("Contando hacia abajo n= "+n);
    n--;
} while (n>0);
System.out.println("n="+n);

System.out.println("");

```

[Imagen 5. While.]

```

System.out.println("##### for #####");
for (int i = 0; i < 10; i++) {
    System.out.println("Contando hacia arriba i= "+i);
}
System.out.println("");
for (int i = 10; i > 0; i--) {
    System.out.println("Contando hacia abajo i= "+i);
}
System.out.println("");
System.out.println("#####for#####");
int [] arreglo = {1,2,3,4,5};
int tamaño = arreglo.length;
System.out.println(tamaño);
for (int i = 0; i < arreglo.length; i++) {
    System.out.println("arreglo["+i+"]= "+arreglo[i]);
}

```

[Imagen 6. For.]

```

System.out.println("");
System.out.println("#####for-each#####");
for (int temp:arreglo) {
    System.out.println("Elemento en arreglo= "+temp);
}

```

[Imagen 7. For each.]

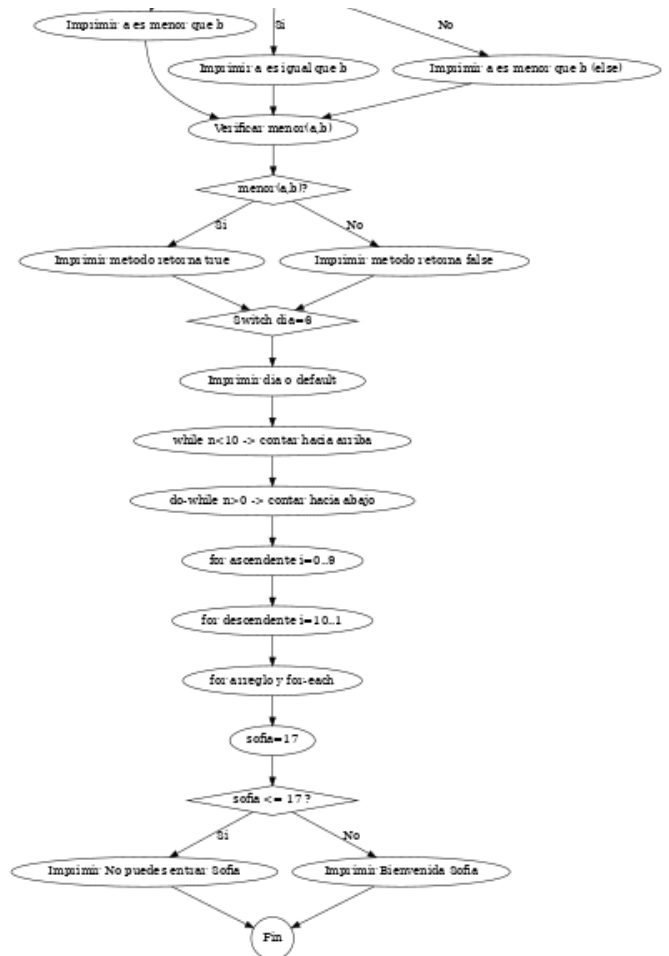
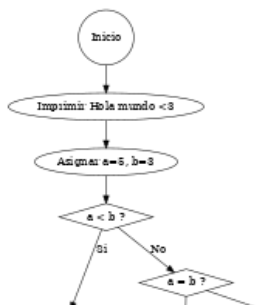
```

int sofia = 17;
System.out.println("Cuantos años tiene Sofia?");
System.out.println(sofia);
if (sofia<=17) {
    System.out.println("No puedes entrar Sofia");
}else{
    System.out.println("Bienvenida Sofia");
}
}

private static boolean menor(int a, int b){
    return a<b;
    //throw new UnsupportedOperationException("Not supported yet");
}

```

[Imagen 8. Últimos ejercicios.]



[Imagen 9 y 10. Diagrama de flujo]

CONCLUSIONES

Durante esta práctica se logró afianzar el uso de variables y constantes de distintos tipos de datos, así como la construcción de expresiones utilizando operadores básicos. Además, se comprendió el papel de las estructuras de control de flujo para la toma de decisiones y la repetición de procesos en un programa.

REFERENCIAS

- Navarro, L. (2008b, mayo 6). *Plantillas o abreviaturas de código en NetBeans 6.1*. <https://monillo007.blogspot.com/2008/05/plantillas-o-abreviaturas-de-codigo-en.html>
- Vélez, J. (2025, 25 agosto). ▷ ¿Qué es un Snippet, para qué sirve y cómo mejorarlo? Jonathan Vélez. <https://jonathanvelez.com/que-es-snippet/>