

Estructuras de Programación

Objetivo

Desarrollar programas que implementen estructuras de control condicionales y repetitivas, con un enfoque en la declaración y uso de variables, operadores, y la creación de un menú interactivo usando la estructura `switch`.

Actividad 1: Declaración y uso de variables

Instrucciones: Crea un programa que declare las siguientes variables:

- Nombre del usuario (`String`)
- Edad (`int`)
- Saldo de una cuenta bancaria (`double`)
- Estado civil (`boolean`: `true` si está casado, `false` si no lo está)

Imprime los valores de cada una con el formato:

```
Nombre: [nombre]
Edad: [edad]
Saldo: [saldo]
Casado: [estado_civil]
```

Desarrollo:

```
import java.util.Scanner;

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

        System.out.print("Ingresa tu nombre: ");
        String nombre = scanner.nextLine();

        System.out.print("Ingresa tu edad: ");
        int edad = scanner.nextInt();

        System.out.print("Ingresa tu saldo bancario: ");
        double saldo = scanner.nextDouble();

        System.out.print("¿Estás casado? (true/false): ");
        boolean casado = scanner.nextBoolean();

        System.out.println("\nInformación ingresada:");
        System.out.println("Nombre: " + nombre);
```

```

        System.out.println("Edad: " + edad);
        System.out.println("Saldo: " + saldo);
        System.out.println("Casado: " + casado);

        scanner.close();
    }
}

```

Actividad 2: Operadores aritméticos, relacionales y lógicos

Instrucciones: Escribe un programa que realice las siguientes operaciones:

1. Solicita dos números al usuario.
2. Realiza las operaciones básicas: suma, resta, multiplicación, división.
3. Determina si el primer número es mayor que el segundo usando operadores relacionales.
4. Verifica si ambos números son positivos y mayores que 10 usando operadores lógicos.
5. Imprime los resultados de todas las operaciones.

Desarrollo:

```

import java.util.Scanner;

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

        System.out.print("Ingresa el primer número: ");
        double num1 = scanner.nextDouble();

        System.out.print("Ingresa el segundo número: ");
        double num2 = scanner.nextDouble();

        System.out.println("\nResultados:");
        System.out.println("Suma: " + (num1 + num2));
        System.out.println("Resta: " + (num1 - num2));
        System.out.println("Multiplicación: " + (num1 * num2));
        System.out.println("División: " + (num1 / num2));

        System.out.println("Primer número > Segundo número: " + (num1 > num2));
        System.out.println("Ambos números son positivos y mayores que 10: " +
            (num1 > 10 && num2 > 10));

        scanner.close();
    }
}

```

Actividad 3: Estructuras condicionales

Instrucciones: Desarrolla un programa que solicite al usuario su edad y realice las siguientes verificaciones:

1. Si la edad es menor de 18, imprimir: "Eres menor de edad."
2. Si la edad está entre 18 y 65, imprimir: "Eres un adulto."
3. Si la edad es mayor o igual a 65, imprimir: "Eres una persona de la tercera edad."

Desarrollo:

```
import java.util.Scanner;

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

        System.out.print("Ingresa tu edad: ");
        int edad = scanner.nextInt();

        if (edad < 18) {
            System.out.println("Eres menor de edad.");
        } else if (edad < 65) {
            System.out.println("Eres un adulto.");
        } else {
            System.out.println("Eres una persona de la tercera edad.");
        }

        scanner.close();
    }
}
```

Actividad 4: Estructuras repetitivas

Instrucciones: Crea un programa que imprima los números del 1 al 10 utilizando las siguientes estructuras:

1. Un bucle `for`
2. Un bucle `while`
3. Un bucle `do-while`

Desarrollo:

```
public class Bucles {
    public static void main(String[] args) {
        System.out.println("Bucle for:");
        for (int i = 1; i <= 10; i++) {
            System.out.println(i);
        }

        System.out.println("\nBucle while:");
        int j = 1;
```

```

        while (j <= 10) {
            System.out.println(j);
            j++;
        }

        System.out.println("\nBucle do-while:");
        int k = 1;
        do {
            System.out.println(k);
            k++;
        } while (k <= 10);
    }
}

```

Actividad 5: Creación de un menú interactivo usando `switch`

Instrucciones: Crea un programa que despliegue un menú con las siguientes opciones:

1. Calcular el cuadrado de un número
2. Determinar si un número es par o impar
3. Salir

Desarrollo:

```

import java.util.Scanner;

public class MenuInteractivo {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        int opcion;

        do {
            System.out.println("\nMenú Principal:");
            System.out.println("1. Calcular el cuadrado de un número");
            System.out.println("2. Determinar si un número es par o impar");
            System.out.println("3. Salir");
            System.out.print("Selecciona una opción: ");
            opcion = scanner.nextInt();

            switch (opcion) {
                case 1:
                    System.out.print("Ingresa un número: ");
                    int num = scanner.nextInt();
                    System.out.println("El cuadrado de " + num + " es " + (num *
num));
                    break;
                case 2:
                    System.out.print("Ingresa un número: ");

```

```
        int numero = scanner.nextInt();
        if (numero % 2 == 0) {
            System.out.println("El número es par.");
        } else {
            System.out.println("El número es impar.");
        }
        break;
    case 3:
        System.out.println("Saliendo del programa...");
        break;
    default:
        System.out.println("Opción no válida. Intenta de nuevo.");
    }
} while (opcion != 3);

scanner.close();
}
```