

## Práctica 02

# Estructuras de Control

### Objetivos:

Esta tercera práctica de la asignatura persigue los siguientes objetivos:

- Practicar las sentencias de control del flujo de un programa.
- Reforzar la comprensión sobre cómo usar lazos y ramas.

### Instrucciones y Actividades:

#### 1. Marco Teórico

La instrucción `if-else` permite ejecutar un bloque de código si una condición es verdadera y otro bloque si la condición es falsa.

Los operadores lógicos (`&&`, `||`, `!`) se utilizan para combinar condiciones en las instrucciones `if-else`.

La instrucción `switch` permite ejecutar diferentes bloques de código según el valor de una expresión.

El lazo `while` ejecuta un bloque de código mientras una condición sea verdadera.

El lazo `do-while` ejecuta un bloque de código al menos una vez y luego repite mientras una condición sea verdadera.

El lazo `for` ejecuta un bloque de código un número específico de veces.

La instrucción `break` termina un lazo o una instrucción `switch`.

La instrucción `continue` salta a la siguiente iteración de un lazo.

Las etiquetas se utilizan para referirse a un bucle específico desde dentro de otro bucle.

#### 2. Uso sentencias de control de flujo

1. Crea una clase llamado `ControlFlujo`. Agrega el método `main` y dentro del método `main` realiza los siguientes pasos.

```
public class ControlFlujo {  
    public static main(String[] args) {  
        // aqui va el código  
    }  
}
```

```
}
```

2. Agrega el código que permita solicitar se ingrese el nombre del usuario y su nota del primer bimestre (la nota del primer bimestre debe ser sobre 20 puntos).

No olvides agregar en la línea 1 del archivo, fuera de la clase

```
import java.util.Scanner;
```

Agrega el código en la siguiente línea del main:

```
Scanner gestorIngreso = new Scanner(System.in);
```

```
// Solicitar al usuario ingresar su nombre
System.out.print("Ingresa tu nombre: ");
String nombre = gestorIngreso.nextLine();
// Solicitar al usuario ingresar su calificación
System.out.print("Ingresa tu nota1 (entre 0 y 20): ");
double nota1 = gestorIngreso.nextDouble();
```

3. Agrega el código que permita validar que la calificación sea un número real entre 0 y 20.

Actualiza el código de inicialización de calificación con un valor inválido.

```
// Validar la calificación
while (nota1 < 0 || nota1 > 20) {
    System.out.print("Ingresa tu nota1 (entre 0 y 20 con dos decimales): ");
    // Verificar si la entrada es un número real
    if (gestorIngreso.hasNextDouble()) {
        nota1 = gestorIngreso.nextDouble();
        // Verificar si la calificación está dentro del rango válido
        if (nota1 < 0 || nota1 > 20) {
            System.out.println("Error: La nota debe estar entre 0 y 20");
        }
    } else {
        // Limpiar el buffer del Scanner
        System.out.println("Error: Ingresa un valor válido");
        gestorIngreso.next();
    }
}
```

4. Desarrolla el código que permita solicitar se ingrese la nota del segundo bimestre (la nota del segundo bimestre debe ser sobre 20 puntos), agrega el código para validar la segunda nota.

5. Obtén la nota final, la cual corresponde a la suma con las dos notas.

```
double fin = nota1 + nota2;
```

6. Genera código para presentar el equivalente de la nota, considerando lo definido en el Reglamento de Régimen Académico de la EPN (la nota es sobre 40 puntos).

```
String equivalencia;
```

```
// Determinar la calificación
if (fin >= 39) {
    equivalencia = "Excelente, con honores";
} else if (fin >= 36 && fin < 38.99) {
    equivalencia = "Excelente";
} else if (fin >= 32 && fin < 35.99) {
    equivalencia = "Muy bueno";
} else if (fin >= 30 && fin < 31.99) {
    equivalencia = "Bueno";
} else if (fin >= 28 && fin < 29.99) {
    equivalencia = "Aprobado";
} else {
    equivalencia = "Fallido";
}
```

7. Incluye el código para presentar en pantalla el nombre del usuario, la nota1, la nota2 y la nota final y su equivalencia, así como cierra el Scanner.

```
System.out.println("El estudiante " + nombre + "
obtuvo la nota 1: " + nota1 + ", obtuvo la nota 2:
" + nota2 + ", su nota final es de: " + fin + "
que equivale a: " + equivalencia);
gestorIngreso.close();
```

8. Agrega un do-while para que el código se ejecute al menos una vez. Pregunta al usuario si le gustaría volver a ejecutar el programa, si indica que sí, vuelve a ejecutar el código.

El código planteado debe ser incluido luego de la declaración del objeto Scanner, el cierre del objeto Scanner debe realizarse luego del do while:

```
char continuar;
do {
    // aquí va el código desarrollado

    // Consumir un salto de línea pendiente
    gestorIngreso.nextLine();

    // Preguntar al usuario si desea continuar
```

```
        System.out.print("¿Deseas ingresar otra  
calificación? (S/N): ");  
        continuar =  
gestorIngreso.nextLine().charAt(0);  
    } while (continuar == 'S' || continuar == 's');  
    // coloca acá el cierre del objeto Scanner
```

9. Modifica tu código, considera que, si `fin` está entre 18 y 27,99, el estudiante puede presentarse al examen de recuperación. Tu código debe preguntar, solo si `fin` está entre 18 y 27,99, si se desea ingresar nota del examen de recuperación (`recuperacion`). Si el usuario indica que desea ingresar nota del examen de recuperación, tu programa debe solicitar ingrese la nota de examen de recuperación y almacenarla en `recuperacion`, así también debe validar que el valor de `recuperacion` esté entre 0 y 40. Si se ingresa la nota del examen de recuperación, debes obtener `fin` como el promedio entre `nota1` + `nota2` con `recuperacion` ( $(fin+recuperacion)/2$ ). También modifica la sección en la que se presenta el resultado para incluir la calificación del examen de recuperación (solo si hay nota).

10. Agrega dos sentencias para presentar el indicador `estado`, el cual es un `char`:

- i. Una que permita asignar el valor 'E' o 'F', usando el operador condicional. 'E' se asigna si `fin` es mayor o igual a 28 y 'F' si es menor a 28.
- ii. Una que permita asignar el valor 'A' o 'F', usando el operador condicional. 'A' se asigna si dio el examen de recuperación y obtuvo una nota superior o igual a 24, y 'F' si es menor a 24.

11. Modifique el código para presentar en pantalla el nombre del usuario, la `nota1`, la `nota2`, la `nota final`, la `nota del examen de recuperación` (solo si la hay), su equivalencia y finalmente el estado.

12. Realiza las siguientes pruebas, los resultados de las pruebas deben ser colocados en la sección Resultados del encabezado.

Caso1:

```
nombre = "Juan Perez"  
nota1 = 0;  
nota2 = 0;  
resultado: XXXX
```

Caso2:

```
nombre = "Jenny Romo"
```

```
        nota1 = 10;
        nota2 = 10;
        recuperación = 40;
        resultado: XXXX
Caso3:
        nombre = "María de los Ángeles Piñeiros"
        nota1 = -1;
        nota1 = 38
        nota1 = 11,99;
        nota2 = -1
        nota2 = 12;
        resultado: XXXX
Caso4:
        nombre = "Jorge Sánchez"
        nota1 = 20;
        nota2 = 20;
        resultado: XXXX
Caso5:
        nombre = "TU NOMBRE"
        nota1 = 13,99;
        nota2 = 20;
        recuperación = 0;
        resultado: XXXX
Caso6:
        nombre = "NOMBRE DE TU COMPAÑERO"
        nota1 = 13,99;
        nota2 = 20;
        recuperación = 30;
        resultado: XXXX
```

### **3. Entregables**

Para esta práctica debes realizar lo siguiente:

1. Completa el código solicitado, completa la información faltante.
2. Ubica la carpeta del proyecto generada, en caso de que hayas usado Eclipse o IntelliJ, elimina la carpeta `bin` o `out` y procede a comprimir la carpeta del proyecto. Esta carpeta comprimida (.ZIP), con el nombre cumpliendo el formato indicado en clase, es la que debes subir en el Práctica 02. En caso de que uses Replit, descarga solamente el archivo `.java`, y ese archivo cómpralo y cárgalo en la plataforma.

**ITID433**

**Programación Avanzada**

**Facultad de Ingeniería Eléctrica y Electrónica**

#### **4. Realizado por:**

David Mejía N.