

**UNIVERSIDAD PRIVADA DEL NORTE**

FACULTAD DE INGENIERÍA



**CARRERA:** INGENIERÍA DE SISTEMAS COMPUTACIONALES

**CURSO:** TÉCNICAS DE PROGRAMACIÓN ORIENTADA A OBJETOS

**DOCENTE:** ING. MARTIN EDUARDO TORRES RODRIGUEZ

**TEMA:** PRACTICA DE CAMPO 2 – DESARROLLO DE CASOS PROPUESTOS

**INTEGRANTES:**

- Sergio Daniel Castro Flores N00400192
- Fabiola Timoteo Huarcaya N00417614
- Alexander Gabino Andrade N00462137
- Kevin Jarlin Castro Palacios N00471823
- Daniel Huatay Lazaro N00391194

2025

LIMA - PERU

## PRACTICA DE CAMPO 2

Link repositorio: <https://github.com/Fabiola322/Proyecto1>

### CASO 1: Lectura de datos simples con Scanner

Consigna: Desarrolla un programa en Java que solicite datos por teclado usando la clase Scanner. El programa debe ser guardado en la carpeta `caso1/` y subido al repositorio con commits explicativos.

Descripción del caso:

En este caso se busca practicar la lectura de datos desde teclado utilizando la clase Scanner en Java. El programa solicitará al usuario que ingrese información básica, como su nombre, edad, peso o cualquier otro dato necesario, y luego mostrará esos valores en pantalla de manera ordenada. Este ejercicio tiene como objetivo familiarizarse con la entrada y salida de datos, reforzar el uso de tipos de datos primitivos y comprender cómo funciona la interacción entre el usuario y el programa a través de la consola.

```

1 // CASO 2: Lectura de datos simples con Scanner
2 // Programa que calcula el Índice de Masa Corporal (IMC) a partir del peso y la estatura del usuario.
3
4 package com.upn.caso2;
5
6 import java.util.Scanner;
7
8 public class LecturaDatos {
9     public static void main(String[] args) {
10         // Crear objeto Scanner para leer datos desde el teclado
11         Scanner sc = new Scanner(System.in);
12
13         System.out.println("== CÁLCULO DE IMC ==");
14
15         System.out.print("Ingrese su nombre: ");
16         String nombre = sc.nextLine();
17
18         System.out.print("Ingrese su peso en kilogramos (kg): ");
19         double peso = sc.nextDouble();
20
21         System.out.print("Ingrese su estatura en metros (m): ");
22         double estatura = sc.nextDouble();
23
24         // Calcular el IMC
25         double imc = peso / (estatura * estatura);
26
27         // Mostrar resultados
28         System.out.println("\n--- RESULTADO ---");
29         System.out.println("Nombre: " + nombre);
30         System.out.printf("IMC: %.2f\n", imc);
31
32         // Clasificación según el valor del IMC
33         if (imc < 18.5) {
34             System.out.println("Estado: Bajo peso");
35         } else if (imc < 24.9) {
36             System.out.println("Estado: Peso normal");
37         } else if (imc < 29.9) {
38             System.out.println("Estado: Sobrepeso");
39         } else {
40             System.out.println("Estado: Obesidad");
41         }
42
43         sc.close();
44     }
45 }
```

1. Estas primeras líneas definen la estructura básica: nos indica el paquete donde se encuentra la clase; permite usar la clase Scanner para leer datos y declarar la clase principal

```

4 package com.upn.caso2;
5
6 import java.util.Scanner;
7
8 public class LecturaDatos {

```

2. Estas líneas constituyen la parte del programa encargada de recibir los datos del usuario. Dentro del método principal **main**, se crea un objeto **Scanner** para permitir la lectura desde el teclado. Luego, el programa muestra mensajes en pantalla y solicita al usuario que ingrese su nombre, su peso en kilogramos y su estatura en metros, almacenando cada dato en variables específicas. Esta sección prepara la información necesaria para realizar el cálculo del Índice de Masa Corporal (IMC) en las siguientes etapas del programa.

```

9⊕   public static void main(String[] args) {
10      // Crear objeto Scanner para leer datos desde el teclado
11      Scanner sc = new Scanner(System.in);
12
13      System.out.println("== CÁLCULO DE IMC ==");
14
15      System.out.print("Ingrese su nombre: ");
16      String nombre = sc.nextLine();
17
18      System.out.print("Ingrese su peso en kilogramos (kg): ");
19      double peso = sc.nextDouble();
20
21      System.out.print("Ingrese su estatura en metros (m): ");
22      double estatura = sc.nextDouble();

```

3. este bloque del programa realiza el cálculo y la interpretación del Índice de Masa Corporal (IMC) a partir de los datos ingresados por el usuario. Primero, se aplica la fórmula  $imc = peso / (estatura^2)$  para obtener el valor del IMC. Luego, el programa muestra los resultados en pantalla, incluyendo el nombre del usuario y el valor del IMC con dos decimales. A continuación, se emplea una estructura condicional if-else para clasificar el resultado: si el IMC es menor a 18.5 se indica "Bajo peso"; entre 18.5 y 24.9 "Peso normal"; entre 25.0 y 29.9 "Sobrepeso"; y si es 30 o más "Obesidad". Finalmente, se cierra el objeto Scanner con `sc.close()` para liberar los recursos utilizados. En conjunto, este bloque completa el proceso lógico del programa, mostrando al usuario su diagnóstico basado en el cálculo del IMC.

```

// Mostrar resultados
System.out.println("\n--- RESULTADO ---");
System.out.println("Nombre: " + nombre);
System.out.printf("IMC: %.2f\n", imc);

// Clasificación según el valor del IMC
if (imc < 18.5) {
    System.out.println("Estado: Bajo peso");
} else if (imc < 24.9) {
    System.out.println("Estado: Peso normal");
} else if (imc < 29.9) {
    System.out.println("Estado: Sobrepeso");
} else {
    System.out.println("Estado: Obesidad");
}

sc.close();
}

```

## Resultados

- Llenamos los datos

```
== CÁLCULO DE IMC ==
Ingrese su nombre: PEDRO
Ingrese su peso en kilogramos (kg): 75
Ingrese su estatura en metros (m): 1,75
```

- Obtenemos como resultado

```
--- RESULTADO ---
Nombre: PEDRO
IMC: 24,49
Estado: Peso normal
```

- Otro ejemplo

```
== CÁLCULO DE IMC ==
Ingrese su nombre: LUCHO
Ingrese su peso en kilogramos (kg): 90
Ingrese su estatura en metros (m): 1,60
|
--- RESULTADO ---
Nombre: LUCHO
IMC: 35,16
Estado: Obesidad
```

**CASO 2: Clase Estudiante interactivo**

Consigna: Implementa una clase Java llamada `Estudiante` con atributos privados, constructor y métodos `get/set`. El estudiante debe ser creado desde el método `main()` y sus datos deben ser mostrados con uso de Scanner. Publica tu código en `caso2/`.

Descripción del caso:

Se crea la clase **Estudiante**, una clase que guarda datos básicos de un alumno, y define métodos constructores, getters y setters, así como métodos para agregar y eliminar cursos, y mostrar todos los datos de manera familiar.

```
public class EstudianteInteractivo {  
    // Creación de clase Estudiante  
    public static class Estudiante{  
        // Atributos privados  
        private String nombre;  
        private String carrera;  
        private int edad;  
        private ArrayList<String> cursos; // lista de cursos (dinámica)  
  
        // Constructor  
        public Estudiante(String nombre, String carrera, int edad){  
            this.nombre = nombre;  
            this.carrera = carrera;  
            this.edad = edad;  
            this.cursos = new ArrayList<>(); // Inicializamos la lista vacía.  
        }  
  
        // Getters  
        public String retornarNombre(){  
            return nombre;  
        }  
  
        public String retornarCarrera(){  
            return carrera;  
        }  
  
        public int retornarEdad(){  
            return edad;  
        }  
  
        public String retornarCursos(){  
            String cursos_str = "";  
            for (String curso : cursos){  
                cursos_str += curso + " ";  
            }  
            return cursos_str.trim();  
        }  
    }  
}
```

```
// Setters
public void cambiarNombre(String nombre){
    this.nombre = nombre;
}

public void cambiarCarrera(String carrera){
    this.carrera = carrera;
}

public void cambiarEdad(int edad){
    this.edad = edad;
}

// Otros métodos
public void agregarCurso(String curso){
    cursos.add(curso);
}

public void eliminarCurso(String curso){
    cursos.remove(curso);
}

public void mostrarDatos(){
    System.out.println("Nombre: " + nombre);
    System.out.println("Carrera: " + carrera);
    System.out.println("Edad: " + String.valueOf(edad));
    System.out.println("Lista de cursos: " + retornarCursos());
}

}
```

La clase **main** se encarga de correr el programa principal, que hace uso de la clase Scanner para capturar datos del alumno de manera interactiva y luego confirmando la recepción de datos.

```
public static void main(String[] args){  
    // Creación del objeto Scanner  
    Scanner input = new Scanner(System.in);  
  
    // Recopilación de datos  
    System.out.print("Ingresa tu nombre: ");  
    String nombre = input.nextLine();  
  
    System.out.print("Ingresa tu carrera: ");  
    String carrera = input.nextLine();  
  
    System.out.print("Ingresa tu edad: ");  
    int edad = Integer.parseInt(input.nextLine());  
  
    // Registro de cursos  
    System.out.print("Ingresa la cantidad de cursos a inscribir: ");  
    int cant_cursos = Integer.parseInt(input.nextLine());  
  
    ArrayList<String> lista_cursos = new ArrayList<>();  
  
    for (int i=1 ; i<=cant_cursos; i++){  
        String formatted_message = "Ingrese curso %d : ".formatted(i);  
        System.out.print(formatted_message);  
        String curso = input.nextLine();  
        lista_cursos.add(curso);  
    }  
  
    // Creación de instancia  
    Estudiante estudiante = new Estudiante(nombre, carrera, edad);  
  
    // Agregamos los cursos a la instancia  
    for (String curso : lista_cursos){  
        estudiante.agregarCurso(curso);  
    }  
  
    input.close();  
  
    // Mostramos mensaje de registro exitoso y datos:  
    System.out.println("Registro exitoso!");  
    System.out.println("Datos: ");  
    estudiante.mostrarDatos();  
}
```

Resultado:

```
Igresa tu nombre: Sergio
Ingresa tu carrera: Sistemas
Ingresa tu edad: 31
Ingresa la cantidad de cursos a inscribir: 2
Ingresé curso 1 : Objetos
Ingresé curso 2 : Algoritmos

Registro exitoso!
```

Datos:

```
Nombre: Sergio
Carrera: Sistemas
Edad: 31
Lista de cursos: Objetos Algoritmos
```

### CASO 3: Clase CuentaBancaria con validación

Consigna: Crea una clase java llamada “CuentaBancaria” que permita operaciones básicas como depositar y retirar, validando que no se retire más del saldo. Usa Scanner para el ingreso de datos. El código debe estar en la carpeta “caso3/” con commits bien documentados.

Descripción del caso:

La clase **CuentaBancaria** representa una cuenta de un cliente en un banco. Contiene atributos privados para el titular y el saldo, los cuales se inicializan mediante un **constructor**. Incluye métodos para **depositar** y **retirar dinero** validando los montos ingresados, así como un método para **mostrar el saldo actual del cliente**. También, cuenta con métodos **getters** que permiten obtener el nombre del titular y el saldo disponible de forma segura.

```
public class CuentaBancaria {
    private String titular;
    private double saldo;

    //CONSTRUCTOR
    public CuentaBancaria(String titular, double saldoInicial) {
        this.titular = titular;
        this.saldo = saldoInicial;
    }
    //Método para depositar dinero
    public void depositar(double monto){
        if(monto > 0){
            saldo += monto;
            System.out.println("Depósito realizado correctamente");
        } else {
            System.out.println("El monto debe ser mayor que cero.");
        }
    }

    // Método para retirar dinero
    public void retirar(double monto){
        if(monto > saldo){
            System.out.println("No se puede retirar más del saldo
disponible.");
        } else if (monto <= 0) {
            System.out.println("El monto debe ser mayor que cero.");
        }else {
            saldo -= monto;
            System.out.println("Retirado realizado correctamente");
        }
    }
    //Método para mostrar saldo
    public void mostrarSaldo(){
        System.out.println("Saldo actual: S/ " + saldo);
    }
    //Getter para titular
    public String getTitular() {
        return titular;
    }
    //Getter para el saldo
```

```

public double getSaldo() {
    return saldo;
}
}
  
```

La clase **BancoApp** actúa como el programa principal permitiendo la interacción entre el usuario y la cuenta bancaria.

Usamos la clase Scanner para solicitar al usuario el nombre del titular de la cuenta y el saldo inicial, validando los datos ingresados para evitar errores. También se implementó un menú interactivo que ofrece opciones para **depositar, retirar, consultar el saldo o salir del sistema**, usando los métodos implementados en la clase **CuentaBancaria**.

Además, incorporamos el manejo de excepciones (InputMismatchException) para garantizar que las entradas sean numéricas y válidas, brindando una experiencia más segura y controlada.

```

// Sirve para manejar errores cuando el usuario ingresa un tipo de dato
incorrecto con Scanner
import java.util.InputMismatchException;
import java.util.Scanner;

public class BancoApp {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in); // Instanciamos el objeto sc de la
//clase Scanner para poder usarla

        System.out.println("Ingrese el nombre del Titular: ");
        String nombre = sc.nextLine();

        // Validación del ingreso del saldo inicial
        double saldoInicial = 0;
        boolean entradaValida = false;

        // Se repetirá hasta que el usuario ingrese un saldo válido (numérico y
//no negativo)
        do {
            try {
                System.out.print("Ingrese el saldo inicial: ");
                saldoInicial = sc.nextDouble();

                if (saldoInicial < 0) {
                    System.out.println(" El saldo no puede ser negativo.");
                } else {
                    entradaValida = true; // Salida del bucle
                }
            } catch (InputMismatchException e) {
                System.out.println(" Error: Debe ingresar un número válido.");
                sc.nextLine(); // Limpia el buffer del Scanner
            }
        } while (!entradaValida);

        // Instanciamos la clase CuentaBancaria con los datos ingresados
        CuentaBancaria cuenta = new CuentaBancaria(nombre, saldoInicial);
    }
}
  
```

```

int opcion;

// Menú principal
do {
    System.out.println("\n--- MENÚ BANCO ---");
    System.out.println("1. Depositar");
    System.out.println("2. Retirar");
    System.out.println("3. Ver saldo");
    System.out.println("4. Salir");
    System.out.print("Elija una opción: ");

    // Validamos que la opción sea numérica
    while (!sc.hasNextInt()) {
        System.out.println(" Debe ingresar un número del 1 al 4.");
        sc.next();
        System.out.print("Elija una opción: ");
    }

    opcion = sc.nextInt();

    switch (opcion) {
        case 1:
            System.out.print("Ingrese monto a depositar: ");
            while (!sc.hasNextDouble()) {
                System.out.println(" Ingrese un valor numérico
válido.");
                sc.next();
                System.out.print("Ingrese monto a depositar: ");
            }
            double montoDep = sc.nextDouble();
            cuenta.depositar(montoDep);
            cuenta.mostrarSaldo();
            break;

        case 2:
            System.out.print("Ingrese monto a retirar: ");
            while (!sc.hasNextDouble()) {
                System.out.println(" Ingrese un valor numérico
válido.");
                sc.next();
                System.out.print("Ingrese monto a retirar: ");
            }
            double montoRet = sc.nextDouble();
            cuenta.retirar(montoRet);
            cuenta.mostrarSaldo();
            break;

        case 3:
            cuenta.mostrarSaldo();
            break;

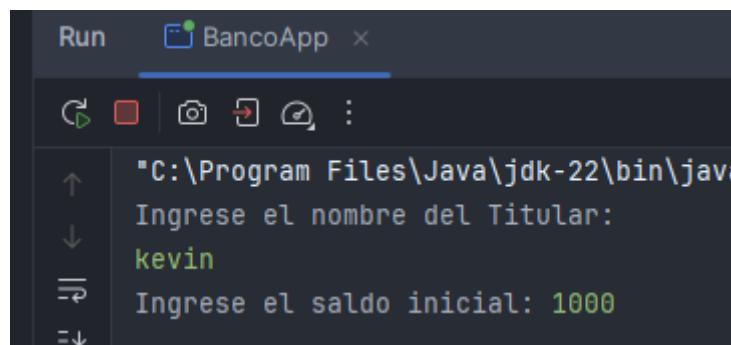
        case 4:
            System.out.println("\n Gracias por usar el sistema, " +

```

```
cuenta.getTitular() + ".");  
        break;  
  
    default:  
        System.out.println(" Opción inválida. Intente  
nuevamente.");  
    }  
  
} while (opcion != 4);  
  
sc.close(); // Cerramos el Scanner  
}  
}
```

**Resultados:**

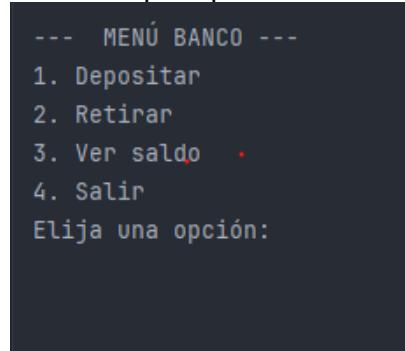
- 1. Ingreso de datos del titular.



The screenshot shows a terminal window titled "BancoApp". The user has entered "kevin" as the account holder's name and "1000" as the initial balance. The terminal output is as follows:

```
"C:\Program Files\Java\jdk-22\bin\java  
Ingresar el nombre del Titular:  
kevin  
Ingresar el saldo inicial: 1000
```

- Menú principal.



The screenshot shows a terminal window displaying the main menu of the application. The menu options are:

```
--- MENÚ BANCO ---  
1. Depositar  
2. Retirar  
3. Ver saldo .  
4. Salir  
Elija una opción:
```

- Ejemplo de depósito exitoso.

```
Elija una opción: 1
Ingrese monto a depositar: 500
Depósito realizado correctamente
Saldo actual: S/ 1500.0
```

- Ejemplo de retiro válido.

```
Elija una opción: 2
Ingrese monto a retirar: 300
Retirado realizado correctamente
Saldo actual: S/ 1200.0
```

- Validaciones.

- Monto negativo

```
Elija una opción: 1
Ingrese monto a depositar: -100
El monto debe ser mayor que cero.
Saldo actual: S/ 1200.0
```

- Retiro mayor al saldo

```
Elija una opción: 2
Ingrese monto a retirar: 200000
No se puede retirar más del saldo disponible.
Saldo actual: S/ 1200.0
```

## 6. Salida del sistema

```
Elija una opción: 4
Gracias por usar el sistema, kevin.
```