

SEMINARIO DE PRÁCTICA

UNIVERSIDAD  
**SIGLO**



La educación evoluciona

CATEDRA - E - INF275 - EDH - PR - TUTORIAS

CUATRIMESTRALES 2/24

Trabajo Práctico N° 3

“Situación Estación Meteorológica Convencional del INTA”

**Alumno:**

Leandro Ariel GONZALEZ - VINF02351

**Profesor:**

Titular Experto Ing. (Msc) Hugo Fernando FRIAS

**Nota: El tp3 lo hacemos corto para ensamblar todo en el tp4 final**

[Escriba aquí]

## Contenido

<b>Título</b> .....	2
<b>Introducción</b> .....	2
Código Main:.....	3
Imagen N° 1 .....	3
<b>Inicio del Sistema</b> .....	4
Imagen N° 2 .....	4
Menú Principal.....	4
Imagen N° 3 .....	4
Imagen N° 4 .....	4
<b>Modulo Termometría</b> .....	5
Código Registro Temperatura: .....	5
Imagen N° 5 .....	5
Imagen N° 6 .....	6
Imagen N° 7 .....	6
Imagen N° 8 .....	7
Imagen N° 9 .....	8
<b>Modulo Pluviometría</b> .....	8
Código Registro Pluviometría: .....	8
Imagen N° 10 .....	8
Imagen N° 11 .....	9
Imagen N° 12 .....	10
Imagen N° 13 .....	11
<b>Prototipo de Video</b> .....	12
<b>GitHub</b> .....	12
<b>Conclusión personal sobre el proyecto:</b> .....	12

### Título

Desarrollo de un Sistema de Registro de datos agrometeorológicos de estaciones con instrumentos convencionales del **INTA**, Unidad **Colonia Benítez**.

### Introducción

Para esta etapa del prototipado haremos un **mock ups** del sistema mostrando una interfaz consola retro, simulando monitores monocromáticos, no realizaremos conexión con la base de datos y nos permitimos algunos cambios en los modificadores de acceso diagramados con anterioridad para tratar de conseguir un mejor uso del código y aplicar conceptos básicos de UML, algunas líneas son redundantes y se podrían crear métodos para evitar esto, se intentará corregir para la última etapa del proyecto. -

### Explicación del Código

Sistema de Registro de Datos Agrometeorológicos - Leandro Ariel Gonzalez – **pág. 2**

Solo se realizó el código de algunos módulos para los cuales usaremos como modelo de la función del software, en todas se crearon métodos **getters** and **setters** y **constructores** por defectos, se utiliza 7 clases, más la clase **Main**, si bien se crearon más estos carecen de código

Las clases:

- RegistroTemperatura
- RegistroPluviometria
- RegistroEvaporimetria
- Pluviometria
- Termometria
- Evaporimetria (No se creó todavía la consola de acceso a esta parte)
- Usuarios

### Código Main:

```
package tp3;
import java.util.Scanner;

public class Tp3 {
    public static final String ANSI_GREEN = "\u001B[32m"; //Para dar un sensación de pantalla de fosforo
    public static final String ANSI_RESET = "\u001B[0m"; // Para resetear el color

    public static void main(String[] args) {

        //Creo la clase scanner para poder leer lo ingresado por pantalla
        Scanner scanner = new Scanner(System.in);
        boolean salir = false;

        System.out.println(ANSI_GREEN + "===== BIENVENIDO AL SISTEMA REGISTRO DE DATOS AGROMETEOROLOGICOS
=====");
        System.out.println(ANSI_GREEN + "Desea gestionar usuarios? 1.SI | 2.No");
        //Intento capturar excepciones ante cualquier ingreso fuera de los valores permitidos
        try{ int gestionar = scanner.nextInt();
        if (gestionar == 1){
            Usuario.gestionarUsuario(); // SI bien es una maqueta, se intenta mostrar el módulo de gestión
            de usuario
        } else if (gestionar == 2) {
            //Inicio de un bucle para gestionar el menú
            while (!salir) {

                System.out.println(ANSI_GREEN + "1.TERMOMETRIA | 2.EVAPORIMETRICA | 3.PLUVIOMETRIA |
4.ANEMOMETRIA | 5.NUBOSIDAD | 6.PSICROMETRIA | 7.FENOMENOS | 8.SALIR");
                System.out.println(ANSI_GREEN + "Elige una opción: ");

                int opcion = scanner.nextInt();

                switch (opcion) {
                    case 1:
                        RegistroTemperatura.ConsolaTermometria();
                        break;
                    case 2:
                        RegistroEvaporimetria.ConsolaEvaporimetria();
                        break;
                    case 3:
                        RegistroPluviometria.ConsolaPluviometria();
                        break;
                    case 4:
                        System.out.println("===== ANEMOMETRIA =====");
                        break;
                    case 5:
                        System.out.println("===== NUBOSIDAD =====");
                        break;
                    case 6:
                        System.out.println("===== PSICROMETRIA =====");
                        break;
                    case 7:
                        System.out.println("===== FENOMENOS METEOROLOGICOS =====");
                        break;
                    case 8:
                        System.out.println("Saliendo del sistema...");
                        salir = true;
                        break;
                    default:
                        System.out.println("Opción no válida, elige nuevamente.");
                }
            }
        } else{System.out.println(ANSI_GREEN + "Ingrese una opción valida...Saliendo del Sistema");}
        }catch (Exception e) {
            System.out.println(ANSI_GREEN + "Error inesperado: se espera el ingreso de un 1 o 2 " /*
+ e.getMessage()*/);
            scanner.nextLine(); // Limpiar el buffer del scanner
        }
        scanner.close();
    }
}
```

Imagen N° 1: Código Main – *fuentes:* Elaboración propia

## Inicio del Sistema

```
===== BIENVENIDO AL SISTEMA REGISTRO DE DATOS AGROMETEOROLOGICOS =====
Desea gestionar usuarios? 1.Si | 2.No
1
```

**Imagen N° 2:** Inicio del sistema – *fuentes:* Elaboración propia

En la primera parte se podrá elegir si ingresar a la parte de *administración de usuario* o la carga de datos, para esto usamos una estructura condicional del estilo **IF – ELSE**, el cual permitirá llamar un método dentro de la clase usuario para gestionarlo. -

## Menú Principal

```
===== BIENVENIDO AL SISTEMA REGISTRO DE DATOS AGROMETEOROLOGICOS =====
Desea gestionar usuarios? 1.Si | 2.No
2
1.TERMOMETRIA | 2.EVAPORIMETRICA | 3.PLUVIOMETRIA | 4.ANEMOMETRIA | 5.NUBOSIDAD | 6.PSICROMETRIA | 7.FENOMENOS | 8.SALIR
Elige una opción:
```

**Imagen N° 3:** Menú principal del sistema – *fuentes:* Elaboración propia

Una vez pasada la etapa del inicio del sistema, nos encontramos con la parte *menú principal*, donde podemos gestionar los módulos del software, para esto usamos una estructura cíclica **WHILE**, también utilizamos una estructura condicional **SWITCH – CASE**. -

```
switch (opcion) {
    case 1:
        RegistroTemperatura.ConsolaTermometria();
        break;
    case 2:
        RegistroEvaporimetrica.ConsolaEvaporimetrica();
        break;
    case 3:
        RegistroPluviometria.ConsolaPluviometria();
        break;
    case 4:
        System.out.println("===== ANEMOMETRIA =====");
        break;
    case 5:
        System.out.println("===== NUBOSIDAD =====");
        break;
    case 6:
        System.out.println("===== PSICROMETRIA =====");
        break;
    case 7:
        System.out.println("===== FENOMENOS METEOROLOGICOS =====");
        break;
    case 8:
        System.out.println("Saliendo del sistema...");
        salir = true;
        break;
    default:
        System.out.println("Opción no válida, elige nuevamente.");
}

} else{System.out.println(ANSI_GREEN + "Ingrese una opción valida...Saliendo del Sistema");}
}catch (Exception e) {
    System.out.println(ANSI_GREEN + "Error inesperado: se espera el ingreso de un 1 o 2 " /*
+ e.getMessage()*/);
    scanner.nextLine(); // Limpiar el buffer del scanner
}
scanner.close();
}
```

**Imagen N° 4:** Código fuente del Switch Case, Menú Principal – *fuentes:* Elaboración propia

Como vimos usamos para el manejo de excepciones un **TRY - CATCH** al bloque donde capturamos el ingreso por teclado.

**Nota:** usamos una variable estática para poner las letras en verde para aparentar que es una pantalla de fósforo o monitor monocromo, hay que gestionar mejor esto, acá hay código redundante. –

## Modulo Termometría

### Código Registro Temperatura:

Esta parte creamos una consola con **While** y un **Switch Case** e intentamos con algunos métodos simular el **ABMCL**(Alta – Baja - Modificación – Consulta - Listar) a la base de datos.

```
package test;

import java.time.LocalDate;
import java.time.LocalDateTime;
import java.time.LocalTime;
import java.util.Scanner;

// Clase para almacenar datos
public class RegistroTemperatura {
    public static final String DB_NAME = "termometria"; // Nombre de la base de datos en formato de archivo
    public static final String DB_PATH = "termometria.db"; // Ruta completa al archivo

    private LocalDate fecha;
    private LocalDateTime hora;
    private float temperatura;
    private int tipoTermometro;

    public RegistroTemperatura() {}

    // Constructor con parámetros
    public RegistroTemperatura(LocalDate fecha, LocalDateTime hora, int tipoTermometro, float grado) {
        this.fecha = fecha;
        this.hora = hora;
        this.temperatura = tipoTermometro;
        this.grado = grado;
        this.observacion = "";
    }

    public LocalDate getFecha() {
        return fecha;
    }

    public void setFecha(LocalDate fecha) {
        this.fecha = fecha;
    }

    public LocalDateTime getHora() {
        return hora;
    }

    public void setHora(LocalDateTime hora) {
        this.hora = hora;
    }

    public int getTipoTermometro() {
        return tipoTermometro;
    }

    public void setTipoTermometro(int tipoTermometro) {
        this.tipoTermometro = tipoTermometro;
    }

    public void getGrado() {
        System.out.println("Grado: " + grado);
    }

    public void setGrado(float grado) {
        this.grado = grado;
    }

    public String getObservacion() {
        return observacion;
    }

    public void setObservacion(String observacion) {
        this.observacion = observacion;
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        boolean salir = false;

        while (!salir) {
            System.out.println("Bienvenido al sistema de registro de datos");

            // Menú principal
            System.out.println("1. Registrar temperatura");
            System.out.println("2. Consultar temperatura");
            System.out.println("3. Modificar temperatura");
            System.out.println("4. Listar temperaturas");
            System.out.println("5. Salir");

            int opcion = scanner.nextInt();

            switch (opcion) {
                case 1:
                    // Registrar temperatura
                    System.out.println("Ingrese la fecha de la toma de datos, formato YYYY-MM-DD");
                    LocalDate fecha = LocalDate.parse(scanner.next());

                    System.out.println("Ingrese la hora de la toma de datos, formato HH:MM:SS");
                    LocalDateTime hora = LocalDateTime.parse(scanner.next());

                    System.out.println("Ingrese el tipo de termómetro");
                    int tipoTermometro = scanner.nextInt();

                    System.out.println("Ingrese la temperatura en grados");
                    float temperatura = scanner.nextFloat();

                    RegistroTemperatura registro = new RegistroTemperatura(fecha, hora, tipoTermometro, temperatura);
                    registro.setObservacion("");

                    System.out.println("Se ha registrado la temperatura exitosamente.");

                    // Guardar en archivo
                    try {
                        FileWriter writer = new FileWriter(DB_PATH, true);
                        writer.write(registro.toString() + "\n");
                        writer.close();
                    } catch (Exception e) {
                        System.out.println("Error al guardar los datos. Por favor, intente de nuevo.");
                    }

                    break;

                case 2:
                    // Consultar temperatura
                    System.out.println("Ingrese la fecha de la toma de datos, formato YYYY-MM-DD");
                    LocalDate fecha = LocalDate.parse(scanner.next());

                    System.out.println("Ingrese la hora de la toma de datos, formato HH:MM:SS");
                    LocalDateTime hora = LocalDateTime.parse(scanner.next());

                    System.out.println("Ingrese el tipo de termómetro");
                    int tipoTermometro = scanner.nextInt();

                    System.out.println("Ingrese la temperatura en grados");
                    float temperatura = scanner.nextFloat();

                    RegistroTemperatura registro = new RegistroTemperatura(fecha, hora, tipoTermometro, temperatura);

                    System.out.println("Se ha consultado la temperatura exitosamente.");

                    // Mostrar en consola
                    System.out.println("Fecha: " + fecha);
                    System.out.println("Hora: " + hora);
                    System.out.println("Tipo de termómetro: " + tipoTermometro);
                    System.out.println("Temperatura: " + temperatura);
                    System.out.println("Observación: " + registro.getObservacion());

                    break;

                case 3:
                    // Modificar temperatura
                    System.out.println("Ingrese la fecha de la toma de datos, formato YYYY-MM-DD");
                    LocalDate fecha = LocalDate.parse(scanner.next());

                    System.out.println("Ingrese la hora de la toma de datos, formato HH:MM:SS");
                    LocalDateTime hora = LocalDateTime.parse(scanner.next());

                    System.out.println("Ingrese el tipo de termómetro");
                    int tipoTermometro = scanner.nextInt();

                    System.out.println("Ingrese la temperatura en grados");
                    float temperatura = scanner.nextFloat();

                    RegistroTemperatura registro = new RegistroTemperatura(fecha, hora, tipoTermometro, temperatura);

                    System.out.println("Se ha modificado la temperatura exitosamente.");

                    // Guardar en archivo
                    try {
                        FileWriter writer = new FileWriter(DB_PATH, true);
                        writer.write(registro.toString() + "\n");
                        writer.close();
                    } catch (Exception e) {
                        System.out.println("Error al guardar los datos. Por favor, intente de nuevo.");
                    }

                    break;

                case 4:
                    // Listar temperaturas
                    System.out.println("Ingrese la fecha de la toma de datos, formato YYYY-MM-DD");
                    LocalDate fecha = LocalDate.parse(scanner.next());

                    System.out.println("Ingrese la hora de la toma de datos, formato HH:MM:SS");
                    LocalDateTime hora = LocalDateTime.parse(scanner.next());

                    System.out.println("Ingrese el tipo de termómetro");
                    int tipoTermometro = scanner.nextInt();

                    System.out.println("Ingrese la temperatura en grados");
                    float temperatura = scanner.nextFloat();

                    RegistroTemperatura registro = new RegistroTemperatura(fecha, hora, tipoTermometro, temperatura);

                    System.out.println("Se han listado las temperaturas exitosamente.");

                    // Mostrar en consola
                    System.out.println("Fecha: " + fecha);
                    System.out.println("Hora: " + hora);
                    System.out.println("Tipo de termómetro: " + tipoTermometro);
                    System.out.println("Temperatura: " + temperatura);
                    System.out.println("Observación: " + registro.getObservacion());

                    break;

                case 5:
                    // Salir
                    System.out.println("¡Gracias por usar el sistema!");
                    salir = true;
                    break;
            }
        }
    }
}
```

**Imagen N° 5:** Código del módulo termometría, registro de temperatura – *fuentes:* Elaboración propia

```
===== BIENVENIDO AL SISTEMA REGISTRO DE DATOS AGROMETEOROLOGICOS =====
Desea gestionar usuarios? 1.Si | 2.No
2
1.TERMOMETRIA | 2.EVAPORIMETRICA | 3.PLUVIOMETRIA | 4.ANEMOMETRIA | 5.NUBOSIDAD | 6.PSICROMETRIA | 7.FENOMENOS | 8.SALIR
Elige una opción:
1
===== TERMOMETRIA =====
1.Agregar Tipo de Termómetro
2.Agregar Lectura
3.Eliminar Lectura
4.Modificar Lectura
5.Consultar Lectura
6.Listar Lecturas
7.Salir
```

**Imagen N° 6:** Pantalla del módulo termometría, registro de temperatura – *fuentes:* Elaboración propia

```
===== BIENVENIDO AL SISTEMA REGISTRO DE DATOS AGROMETEOROLOGICOS =====
Desea gestionar usuarios? 1.Si | 2.No
2
1.TERMOMETRIA | 2.EVAPORIMETRICA | 3.PLUVIOMETRIA | 4.ANEMOMETRIA | 5.NUBOSIDAD | 6.PSICROMETRIA | 7.FENOMENOS | 8.SALIR
Elige una opción:
1
===== TERMOMETRIA =====
1.Agregar Tipo de Termómetro
2.Agregar Lectura
3.Eliminar Lectura
4.Modificar Lectura
5.Consultar Lectura
6.Listar Lecturas
7.Salir
1
===== Termómetro =====
1.Agregar Termómetro
2.Eliminar Termómetro
3.Modificar Termómetro
4.Consultar Termómetro
5.Listar Termómetro
6.Salir
```

**Imagen N° 7:** Pantalla del módulo termometría, carga de termómetro – *fuentes:* Elaboración propia

```
package tp3;

import java.util.Scanner;

import java.time.LocalDate;
import java.time.LocalDateTime;
import java.time.LocalTime;

//
//
// descripcion general sensor
//

public class Termometro {

    //Se pone en una clase que tenga las constantes que se van a usar en el sistema
    public static final String ANSI_GREEN = "\033[32m"; //Para dar un sensación de pantalla de terminal
    public static final String ANSI_RESET = "\033[0m"; // Para restablecer el color
    protected int tipo_termometro;
    protected int total_grados;
    protected String descripcion;

    //constructor vacío
    public Termometro() {}

    //constructor con los atributos correspondientes
    public Termometro(int tipo_termometro, int total_grados, String descripcion) {
        this.tipo_termometro = tipo_termometro;
        this.total_grados = total_grados;
        this.descripcion = descripcion;
    }

    public int getTipoTermometro() {
        return tipo_termometro;
    }

    public void setTipoTermometro(int tipo_termometro) {
        this.tipo_termometro = tipo_termometro;
    }

    public int getTotalGrados() {
        return total_grados;
    }

    public void setTotalGrados(int grados) {
        this.total_grados = total_grados;
    }

    public String getDescripcion() {
        return descripcion;
    }

    public void setDescripcion(String descripcion) {
        this.descripcion = descripcion;
    }

    public static void consolaTermometro() {
        Scanner scanner = new Scanner(System.in);
        boolean salir = false;
        while (!salir) {
            System.out.println(ANSI_GREEN + "===== Terminómetro =====");
            System.out.println(ANSI_GREEN + "> Agregar Terminómetro");
            System.out.println(ANSI_GREEN + "> Eliminar Terminómetro");
            System.out.println(ANSI_GREEN + "> Modificar Terminómetro");
            System.out.println(ANSI_GREEN + "> Consultar Terminómetro");
            System.out.println(ANSI_GREEN + "> Salir.");
            int opcion = scanner.nextInt();
            scanner.nextLine(); // Consumir el carácter de nueva línea
            switch (opcion) {
                case 1:
                    //se solicita los datos para dar de alta
                    try {System.out.println(ANSI_GREEN + "Ingrese tipo de termómetro, se espera un entero");}
                    {int tipo_termometro = scanner.nextInt();}
                    {System.out.println(ANSI_GREEN + "Ingresé la capacidad total del termómetro, se espera un entero");}
                    {int total_grados = scanner.nextInt();}
                    {scanner.nextLine();}
                    {System.out.println(ANSI_GREEN + "Ingresé descripción de termómetro");}
                    {String descripcion = scanner.nextLine();}
                    {agregarTermometro(tipo_termometro, total_grados, descripcion);}
                    {break;}
                    {catch(Exception e){}
                        {System.out.println(ANSI_GREEN + "Error al ingresar los datos. Por favor, intente de nuevo.");}
                    }
                    break;
                case 2:
                    //se solicita los datos para eliminar
                    try {System.out.println(ANSI_GREEN + "Ingrese tipo de termómetro, se espera un entero");}
                    {int tipo_termometro = scanner.nextInt();}
                    {System.out.println(ANSI_GREEN + "Ingresé la capacidad total del termómetro, se espera un entero");}
                    {int total_grados = scanner.nextInt();}
                    {System.out.println(ANSI_GREEN + "Ingresé descripción de termómetro");}
                    {String descripcion = scanner.nextLine();}
                    {eliminarTermometro(tipo_termometro, total_grados, descripcion);}
                    {break;}
                    {catch(Exception e){}
                        {System.out.println(ANSI_GREEN + "Error al ingresar los datos. Por favor, intente de nuevo.");}
                    }
                    break;
                case 3:
                    //se solicita los datos para modificar
                    //se solicita los datos para eliminar
                    try {System.out.println(ANSI_GREEN + "Ingrese tipo de termómetro, se espera un entero");}
                    {int tipo_termometro = scanner.nextInt();}
                    {System.out.println(ANSI_GREEN + "Ingresé la capacidad total del termómetro, se espera un entero");}
                    {int total_grados = scanner.nextInt();}
                    {System.out.println(ANSI_GREEN + "Ingresé descripción de termómetro");}
                    {String descripcion = scanner.nextLine();}
                    {modificarTermometro(tipo_termometro, total_grados, descripcion);}
                    {break;}
                    {catch(Exception e){}
                        {System.out.println(ANSI_GREEN + "Error al ingresar los datos. Por favor, intente de nuevo.");}
                    }
                    break;
                case 4:
                    //se solicita los datos para consultar
                    //se solicita los datos para eliminar
                    try {System.out.println(ANSI_GREEN + "Ingrese tipo de termómetro, se espera un entero");}
                    {int tipo_termometro = scanner.nextInt();}
                    {System.out.println(ANSI_GREEN + "Ingresé la capacidad total del termómetro, se espera un entero");}
                    {int total_grados = scanner.nextInt();}
                    {System.out.println(ANSI_GREEN + "Ingresé descripción de termómetro");}
                    {String descripcion = scanner.nextLine();}
                    {consultarTermometro(tipo_termometro, total_grados, descripcion);}
                    {break;}
                    {catch(Exception e){}
                        {System.out.println(ANSI_GREEN + "Error al ingresar los datos. Por favor, intente de nuevo.");}
                    }
                    break;
                case 5:
                    //se solicita los datos para listar
                    //se solicita los datos para eliminar
                    try {System.out.println(ANSI_GREEN + "Ingrese tipo de termómetro, se espera un entero");}
                    {int tipo_termometro = scanner.nextInt();}
                    {System.out.println(ANSI_GREEN + "Ingresé la capacidad total del termómetro, se espera un entero");}
                    {int total_grados = scanner.nextInt();}
                    {System.out.println(ANSI_GREEN + "Ingresé descripción de termómetro");}
                    {String descripcion = scanner.nextLine();}
                    {listarTermometro(tipo_termometro, total_grados, descripcion);}
                    {break;}
                    {catch(Exception e){}
                        {System.out.println(ANSI_GREEN + "Error al ingresar los datos. Por favor, intente de nuevo.");}
                    }
                    break;
                default:
                    {System.out.println(ANSI_GREEN + "Opción no válida, elige nuevamente.");}
            }
        }
    }

    private static void agregarTermometro(int tipo_termometro,int total_grados, String descripcion){
        System.out.println(ANSI_GREEN + "Se agregó un termómetro "+ descripcion + " Con el total de grados "+ total_grados);
        Tipo de Termómetro "+tipo_termometro;
    }

    private static void eliminarTermometro(int tipo_termometro,int total_grados, String descripcion){
        System.out.println(ANSI_GREEN + "Se eliminó un termómetro "+ descripcion + " Con el total de grados "+ total_grados);
        Tipo de Termómetro "+tipo_termometro;
    }

    private static void consultarTermometro(int tipo_termometro,int total_grados, String descripcion){
        System.out.println(ANSI_GREEN + "Se consultó un termómetro "+ descripcion + " Con el total de grados "+ total_grados);
        Tipo de Termómetro "+tipo_termometro;
    }

    private static void listarTermometro(int tipo_termometro,int total_grados, String descripcion){
        System.out.println(ANSI_GREEN + "Se listó terminómetro "+ descripcion + " Con el total de grados "+ total_grados);
        Tipo de Termómetro "+tipo_termometro;
    }

}
```

**Imagen N° 8:** Código fuente del módulo termometría – *fuentes:* Elaboración propia  
Sistema de Registro de Datos Agrometeorológicos - Leandro Ariel Gonzalez – **pág. 7**

Como vemos en el código, se intenta aplicar estructuras básicas y simular el acceso de bases con la clase **Scanner** e impresiones por pantalla con **System.out.println()**, aplicando conceptos generales de la programación orientada a objetos como atributos, métodos y el tipo de acceso a estos

```
1.TERMOMETRIA | 2.EVAPORIMETRICA | 3.PLUVIOMETRIA | 4.ANEMOMETRIA | 5.NUBOSIDAD | 6.PSICROMETRIA | 7.FENOMENOS | 8.SALIR
Elige una opción:
1
===== TERMOMETRIA =====
1.Agregar Tipo de Termómetro
2.Agregar Lectura
3.Eliminar Lectura
4.Modificar Lectura
5.Consultar Lectura
6.Listar Lecturas
7.Salir
1
===== Termómetro =====
1.Agregar Termómetro
2.Eliminar Termómetro
3.Modificar Termómetro
4.Consultar Termómetro
5.Listar Termómetro
6.Salir
1
Ingrese tipo de Termómetro, se espera un entero
3
Ingrese la capacidad total del termometro, se espera un entero
34
Ingrese descripción de termómetro
prueba
Se agrego un termómetro prueba Con el total de grados 34 Tipo de Termómetro 3
===== Termómetro =====
```

**Imagen N° 9:** Pantalla del módulo termometría – *fuelle:* Elaboración propia

### Modulo Pluviometría

#### Código Registro Pluviometría:

Esta parte creamos una consola con **While** y un **Switch Case** e intentamos con algunos métodos simular el **ABMCL** a la base de datos

```
1.TERMOMETRIA | 2.EVAPORIMETRICA | 3.PLUVIOMETRIA | 4.ANEMOMETRIA | 5.NUBOSIDAD | 6.PSICROMETRIA | 7.FENOMENOS | 8.SALIR
Elige una opción:
3
===== PLUVIOMETRIA =====
1.Agregar Tipo de Pluviómetro
2.Agregar Lectura
3.Eliminar Lectura
4.Modificar Lectura
5.Consultar Lectura
6.Listar Lecturas
7.Salir
```

**Imagen N° 10:** Pantalla del módulo Pluviometría – *fuelle:* Elaboración propia



**Imagen N° 11:** Código fuente del módulo Pluviometría, registro pluviometría – *fuentes:* Elaboración propia

```
1
1.TERMOMETRIA | 2.EVAPORIMETRICA | 3.PLUVIOMETRIA | 4.ANEMOMETRIA | 5.NUBOSIDAD | 6.PSICROMETRIA | 7.FENOMENOS | 8.SALI
Elige una opción:
3
===== PLUVIOMETRIA =====
1.Agregar Tipo de Pluviómetro
2.Agregar Lectura
3.Eliminar Lectura
4.Modificar Lectura
5.Consultar Lectura
6.Listar Lecturas
7.Salir
1
===== PLUVIOMETRO =====
1.Agregar Pluviómetro
2.Eliminar Pluviómetro
3.Modificar Pluviómetro
4.Consultar Pluviómetro
5.Listar Pluviómetro
6.Salir
```

**Imagen N° 12:** Pantalla del módulo Pluviometría, carga pluviómetro – *fente:* Elaboración propia

```

package tp3;
import java.util.Scanner;
import java.time.LocalDate;
import java.time.LocalDateTime;
import java.time.LocalTime;
import java.time.OffsetTime;
import java.time.ZoneOffset;

/**
 * @author gonzaes.leandro
 */
public class Pluviometria {
    // Los datos se ingresan a través de los constantes que se van a usar en el sistema.
    public static final String ANSI_GREEN = "\u001B[32m"; // Para dar un formato de pantalla de texto
    public static final String ANSI_RESET = "\u001B[0m"; // Para resetear el color

    protected int tipo_pluviometro;
    protected int capacidad;
    protected String descripcion;
    public Pluviometria() {
    }
    //Constructor con los atributos correspondientes
    public Pluviometria(int tipo_pluviometro, int capacidad, String descripcion) {
        this.tipo_pluviometro = tipo_pluviometro;
        this.capacidad = capacidad;
        this.descripcion = descripcion;
    }
    public int getTipo_pluviometro() {
        return tipo_pluviometro;
    }
    public void setTipo_pluviometro(int tipo_pluviometro) {
        this.tipo_pluviometro = tipo_pluviometro;
    }
    public int getCapacidad() {
        return capacidad;
    }
    public void setCapacidad(int capacidad) {
        this.capacidad = capacidad;
    }
    public String getDescripcion() {
        return descripcion;
    }
    public void setDescripcion(String descripcion) {
        this.descripcion = descripcion;
    }
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        boolean salir = false;
        while (!salir) {
            System.out.println(ANSI_GREEN + "===== PLUVIOMETRO =====");
            System.out.println(ANSI_GREEN + "1. Agregar Pluviómetro");
            System.out.println(ANSI_GREEN + "2. Eliminar Pluviómetro");
            System.out.println(ANSI_GREEN + "3. Modificar Pluviómetro");
            System.out.println(ANSI_GREEN + "4. Consultar Pluviómetro");
            System.out.println(ANSI_GREEN + "5. Listar Pluviómetro");
            System.out.println(ANSI_GREEN + "6. Salir");
            int opcion = scanner.nextInt();
            scanner.nextLine(); // Consumir el carácter de nueva línea
            switch (opcion) {
                case 1:
                    //Se solicita los datos para dar de alta
                    try {
                        System.out.println(ANSI_GREEN + "Ingrese tipo de pluviómetro, se espera un entero");
                        int tipo_pluviometro = scanner.nextInt();
                        System.out.println(ANSI_GREEN + "Ingrese la capacidad total del pluviómetro, se espera un entero");
                        int capacidad = scanner.nextInt();
                        scanner.nextLine();
                        System.out.println(ANSI_GREEN + "Ingrese descripción de pluviómetro");
                        String descripcion = scanner.nextLine();
                        agregarPluviometro(tipo_pluviometro, capacidad, descripcion);
                    } catch (Exception e) {
                        System.out.println(ANSI_GREEN + "Error al ingresar los datos. Por favor, intente de nuevo.");
                    }
                    break;
                case 2:
                    //Se solicita los datos para eliminar
                    try {
                        System.out.println(ANSI_GREEN + "Ingrese tipo de pluviómetro, se espera un entero");
                        int tipo_pluviometro = scanner.nextInt();
                        System.out.println(ANSI_GREEN + "Ingrese la capacidad total del pluviómetro, se espera un entero");
                        int capacidad = scanner.nextInt();
                        System.out.println(ANSI_GREEN + "Ingrese descripción de pluviómetro");
                        String descripcion = scanner.nextLine();
                        eliminarPluviometro(tipo_pluviometro, capacidad, descripcion);
                    } catch (Exception e) {
                        System.out.println(ANSI_GREEN + "Error al ingresar los datos. Por favor, intente de nuevo.");
                    }
                    break;
                case 3:
                    //Se solicita los datos para modificar
                    //Se solicita los datos para eliminar
                    try {
                        System.out.println(ANSI_GREEN + "Ingrese tipo de pluviómetro, se espera un entero");
                        int tipo_pluviometro = scanner.nextInt();
                        System.out.println(ANSI_GREEN + "Ingrese la capacidad total del pluviómetro, se espera un entero");
                        int capacidad = scanner.nextInt();
                        System.out.println(ANSI_GREEN + "Ingrese descripción de pluviómetro");
                        String descripcion = scanner.nextLine();
                        modificarPluviometro(tipo_pluviometro, capacidad, descripcion);
                    } catch (Exception e) {
                        System.out.println(ANSI_GREEN + "Error al ingresar los datos. Por favor, intente de nuevo.");
                    }
                    break;
                case 4:
                    //Se solicita los datos para consultar
                    //Se solicita los datos para eliminar
                    try {
                        System.out.println(ANSI_GREEN + "Ingrese tipo de pluviómetro, se espera un entero");
                        int tipo_pluviometro = scanner.nextInt();
                        System.out.println(ANSI_GREEN + "Ingrese la capacidad total del pluviómetro, se espera un entero");
                        int capacidad = scanner.nextInt();
                        System.out.println(ANSI_GREEN + "Ingrese descripción de pluviómetro");
                        String descripcion = scanner.nextLine();
                        consultarPluviometro(tipo_pluviometro, capacidad, descripcion);
                    } catch (Exception e) {
                        System.out.println(ANSI_GREEN + "Error al ingresar los datos. Por favor, intente de nuevo.");
                    }
                    break;
                case 5:
                    //Se solicita los datos para listar
                    //Se solicita los datos para eliminar
                    try {
                        System.out.println(ANSI_GREEN + "Ingrese tipo de pluviómetro, se espera un entero");
                        int tipo_pluviometro = scanner.nextInt();
                        System.out.println(ANSI_GREEN + "Ingrese la capacidad total del pluviómetro, se espera un entero");
                        int capacidad = scanner.nextInt();
                        System.out.println(ANSI_GREEN + "Ingrese descripción de pluviómetro");
                        String descripcion = scanner.nextLine();
                        listarPluviometro(tipo_pluviometro, capacidad, descripcion);
                    } catch (Exception e) {
                        System.out.println(ANSI_GREEN + "Error al ingresar los datos. Por favor, intente de nuevo.");
                    }
                    break;
                case 6:
                    System.out.println(ANSI_GREEN + "Saliedo del sistema...");
                    salir = true;
                    break;
                default:
                    System.out.println(ANSI_GREEN + "Opción no válida, elige nuevamente.");
            }
        }
    }
    public static void agregarPluviometro(int tipo_pluviometro, int capacidad, String descripcion) {
        System.out.println(ANSI_GREEN + "Se agrego un pluviómetro " + descripcion + " " + "Con el total de grados " + capacidad + " Tipo de Termómetro " + tipo_pluviometro);
    }
    public static void modificarPluviometro(int tipo_pluviometro, int capacidad, String descripcion) {
        System.out.println(ANSI_GREEN + "Se modifico un pluviómetro " + descripcion + " " + "Con el total de grados " + capacidad + " Tipo de Termómetro " + tipo_pluviometro);
    }
    public static void eliminarPluviometro(int tipo_pluviometro, int capacidad, String descripcion) {
        System.out.println(ANSI_GREEN + "Se elimino un pluviómetro " + descripcion + " " + "Con el total de grados " + capacidad + " Tipo de Termómetro " + tipo_pluviometro);
    }
    public static void consultarPluviometro(int tipo_pluviometro, int capacidad, String descripcion) {
        System.out.println(ANSI_GREEN + "Se consulto un pluviómetro " + descripcion + " " + "Con el total de grados " + capacidad + " Tipo de Termómetro " + tipo_pluviometro);
    }
    public static void listarPluviometro(int tipo_pluviometro, int capacidad, String descripcion) {
        System.out.println(ANSI_GREEN + "Se listo pluviómetro " + descripcion + " " + "Con el total de grados " + capacidad + " Tipo de Termómetro " + tipo_pluviometro);
    }
}

```

**Imagen N° 13:** Código fuente del módulo Pluviometría, clase pluviometría – *fente:* Elaboración propia  
Sistema de Registro de Datos Agrometeorológicos - Leandro Ariel Gonzalez – **pág. 11**

Como se dijo antes, si bien se hacen getter and setter, juntos con sus constructores por defecto, se lo hace para futuras necesidades.

#### **Prototipo de Video**

<https://www.youtube.com/watch?v=2AE-POHlcwk>

#### **GitHub**

<https://github.com/leandroariel14/TrabajoS21/tree/main/src>

#### **Conclusión personal sobre el proyecto:**

Como experiencia personal, ha sido un gran desafío para mí tener que dedicar el aprendizaje a un lenguaje totalmente diferente a los que manejaba, si bien la teoría de el paradigma de orientado de objetos la hemos visto, ponerlo en práctica ha sido un poco complejo, ha sido un nuevo desafío para mí, el lenguaje estructurado con el cual yo tuve mis pocas experiencias como programador me ha condicionado a pensar de otra forma, para mí esto fue tratar de comprender y ver de otra manera, si bien los tiempos son exigentes por cuestiones de agenda, ha sido gratificante. -