UNEMI
UNIVERSIDAD ESTATAL DE MILAGRO

FACULTAD DE CIENCIAS E INGENIERÍA

CARRERA DE INGENIERÍA DE SOFTWARE

TEMA:

Desarrollar Segundo Proyecto

AUTOR:

NATANAEL ABRAHAM PÉREZ CHEVEZ

JOSÉ ANDRÉS GÓMEZ MOLINA

ADÁN ALÍ ESCANDÓN ROCA

ASIGNATURA:

MODELOS MATEMÁTICOS Y SIMULACIÓN

DOCENTE:

FABRICIO ISIDRO TORRES MORALES

PERIODO:

Abril 2025 a Julio 2025

MILAGRO-ECUADOR

Contenido

1. Información General del Proyecto 3

2. Descripción General del Sistema 3

3. Arquitectura del Sistema 4

4. Especificaciones Técnicas por Módulo 4

4.1. Modulo de Arduino 4

4.2. Módulo de Detección del Puerto Arduino 6

..... 7

4.3. Módulo de Lectura de Datos Seriales 7

4.4. Módulo de Reconexión Automática 8

..... 9

4.5. Módulo del Dashboard con Dash 9

4.6. Funciones Auxiliares 10

5. Base de Datos 10

6. Interfaces y Comunicación 10

7. Implementación de la Interfaz de Usuario 11

8. Manejo de Errores y Excepciones 11

9. Pruebas 11

10. Seguridad 11

11. Despliegue e Instalación 11

12. Mantenimiento y Escalabilidad 13

13. Referencias Técnicas 13

1. Información General del Proyecto

- **Título del proyecto:** Sistema de Monitoreo de Humedad con Arduino y Dash
- **Autor/Desarrollador:** José Andrés Gómez Molina, Adán Alí Escandón Roca, Natanael Abraham Pérez Chevez
- **Institución:** Universidad Estatal de Milagro, Facultad de Ciencias e Ingeniería, Carrera de Ingeniería de Software
- **Periodo de desarrollo:** Abril 2025 a Julio 2025
- **Versión del software:** 1.0

2. Descripción General del Sistema

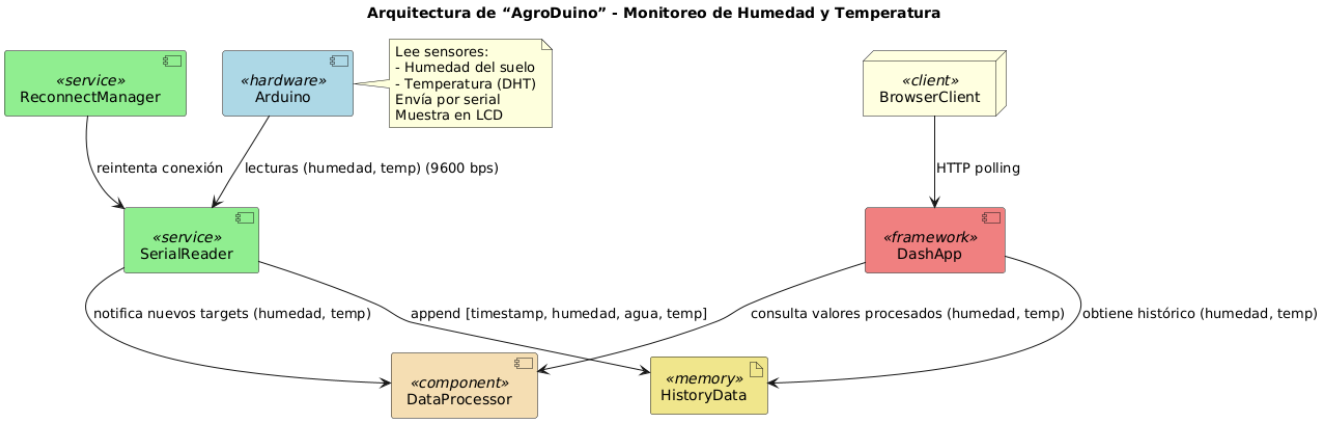
- **Descripción técnica:** El sistema monitorea la humedad del suelo y la temperatura ambiente mediante sensores conectados a un Arduino. Los datos son enviados a través del puerto serial a una aplicación en Python, que utiliza Dash para mostrarlos en tiempo real en un dashboard web interactivo. Incluye gauges para humedad, nivel de agua y temperatura, una tabla de historial, gráficos de tendencia y alertas dinámicas.
- **Objetivos del software:** Proporcionar una herramienta visual para monitorear en tiempo real la humedad del suelo y la temperatura ambiente, alertando sobre necesidades de riego o condiciones críticas de temperatura.
- **Alcance técnico:** Integra comunicación serial con Arduino para adquirir datos de humedad y temperatura, y usa Dash para visualización dinámica en un entorno web.
- **Tecnologías utilizadas:**
    - Python 3.x
    - pyserial (comunicación serial)
    - Dash (interfaz web)
    - Plotly (gráficos interactivos)
    - NumPy (cálculos numéricos)
    - Bootstrap y Font Awesome (estilos y íconos)
- **Requisitos de hardware y software:**
    - **Hardware:** Arduino con sensores de humedad y temperatura conectados vía USB.
        1. **Arduino Uno:** El microcontrolador principal que lee los datos de los sensores (FC-28-B y DHT11), los procesa y los envía a la pantalla LCD y, mediante el puerto serial, a una aplicación externa.

2. **Sensor de Humedad del Suelo FC-28-B:** Un sensor analógico que mide la humedad del suelo detectando la conductividad eléctrica.
3. **Sensor de Temperatura DHT11:** Un sensor digital que mide la temperatura ambiente (en °C). En este proyecto, se usa exclusivamente para temperatura y está conectado al pin A1 del Arduino.
4. **Pantalla LCD 16x2 con Módulo I2C:** Una pantalla que muestra los datos de humedad y temperatura localmente.

- **Software:** Python 3.x con las bibliotecas pyserial, dash, dash-bootstrap-components, numpy y plotly instaladas. Compatible con Windows, macOS y Linux.

3. Arquitectura del Sistema

- **Diagrama de arquitectura:** El sistema consta de dos componentes principales:
  - **Arduino:** Lee datos de sensores de humedad y temperatura, enviándolos por el puerto serial.
  - **Aplicación Python:** Procesa los datos seriales y los muestra en un dashboard web usando Dash.
- **Estructura del código:** El código Python se organiza en funciones modulares para detectar el puerto Arduino, leer datos seriales, reconexión automática y actualización de la interfaz. Usa hilos (threading) para tareas en segundo plano.



4. Especificaciones Técnicas por Módulo

4.1. Modulo de Arduino

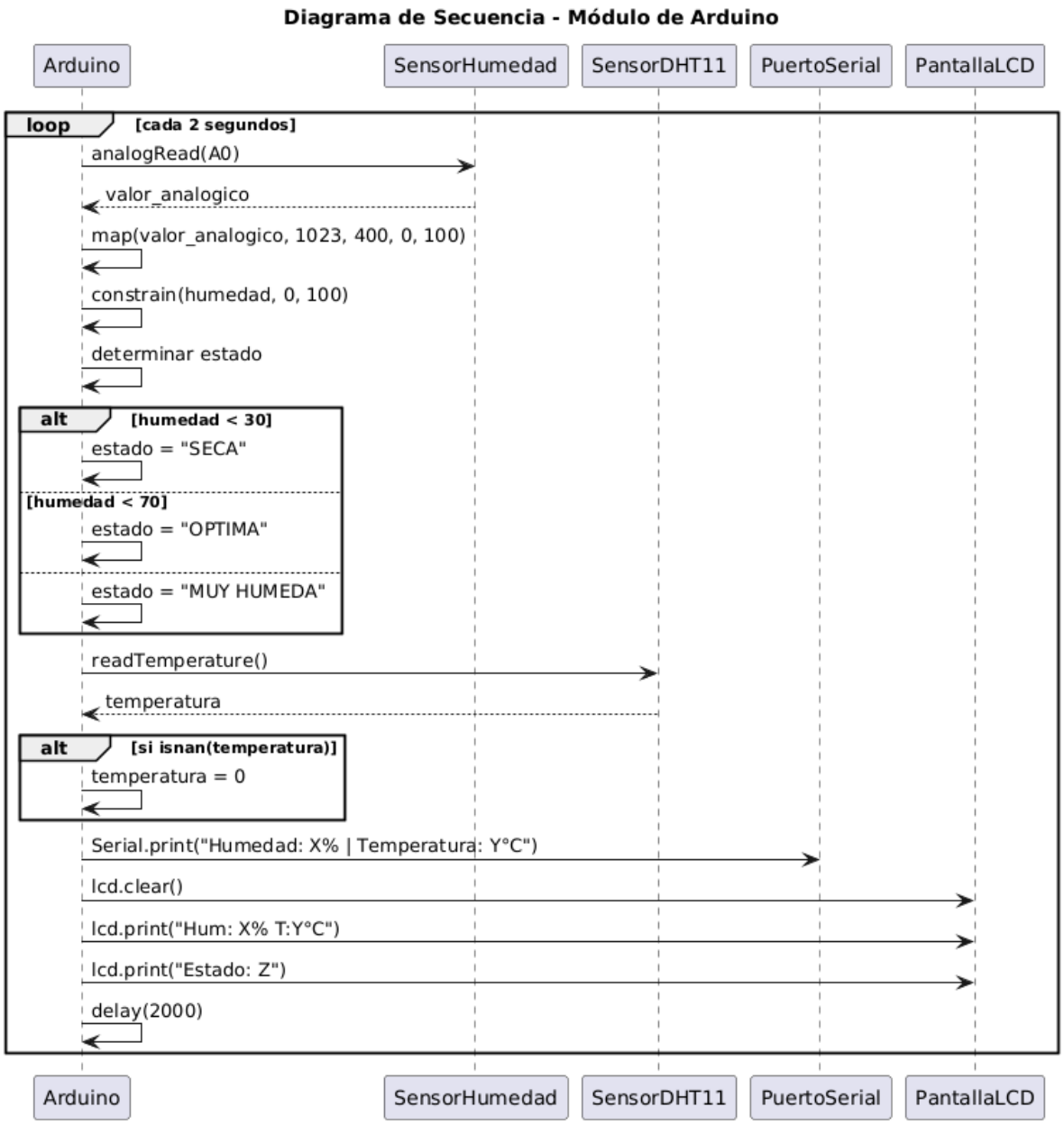
Descripción: Este módulo utiliza un Arduino para leer datos de sensores de hu- medad del suelo y temperatura, enviarlos por el puerto serial y mostrarlos en una pantalla LCD.

Componentes:

- Sensor de humedad del suelo conectado al pin A0.
- Sensor DHT11 (temperatura) conectado al pin A1.
- Pantalla LCD I2C para visualización local.

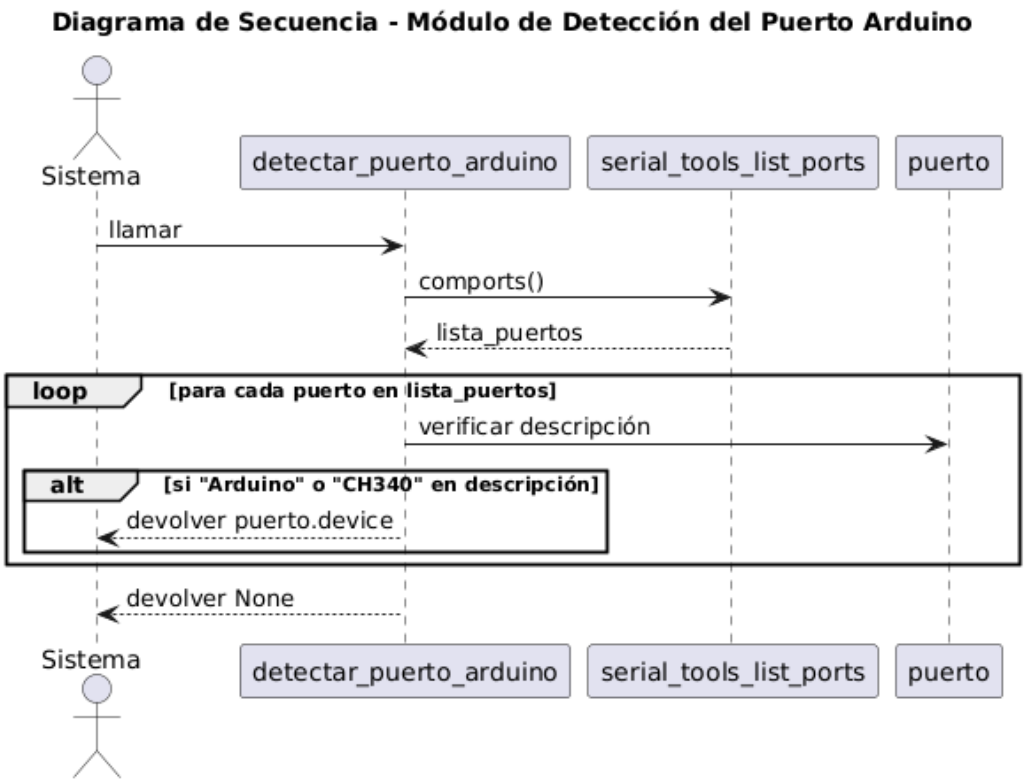
Funcionalidad:

- Lee el valor analógico del sensor de humedad y lo convierte a un porcentaje (0-100 %) usando una calibración (1023 = seco, 400 = agua).
- Determina el estado del suelo: ”SECA”(<30 %), .OPTIMA”(30-69 %), ”MUY HUMEDA”(70 %).
- Lee la temperatura ambiente con el sensor DHT11.
- Envía los datos al puerto serial en el formato ”Humedad: X % | Temperatura: Y°C”.
- Muestra los datos (humedad, temperatura y estado) en la pantalla LCD.



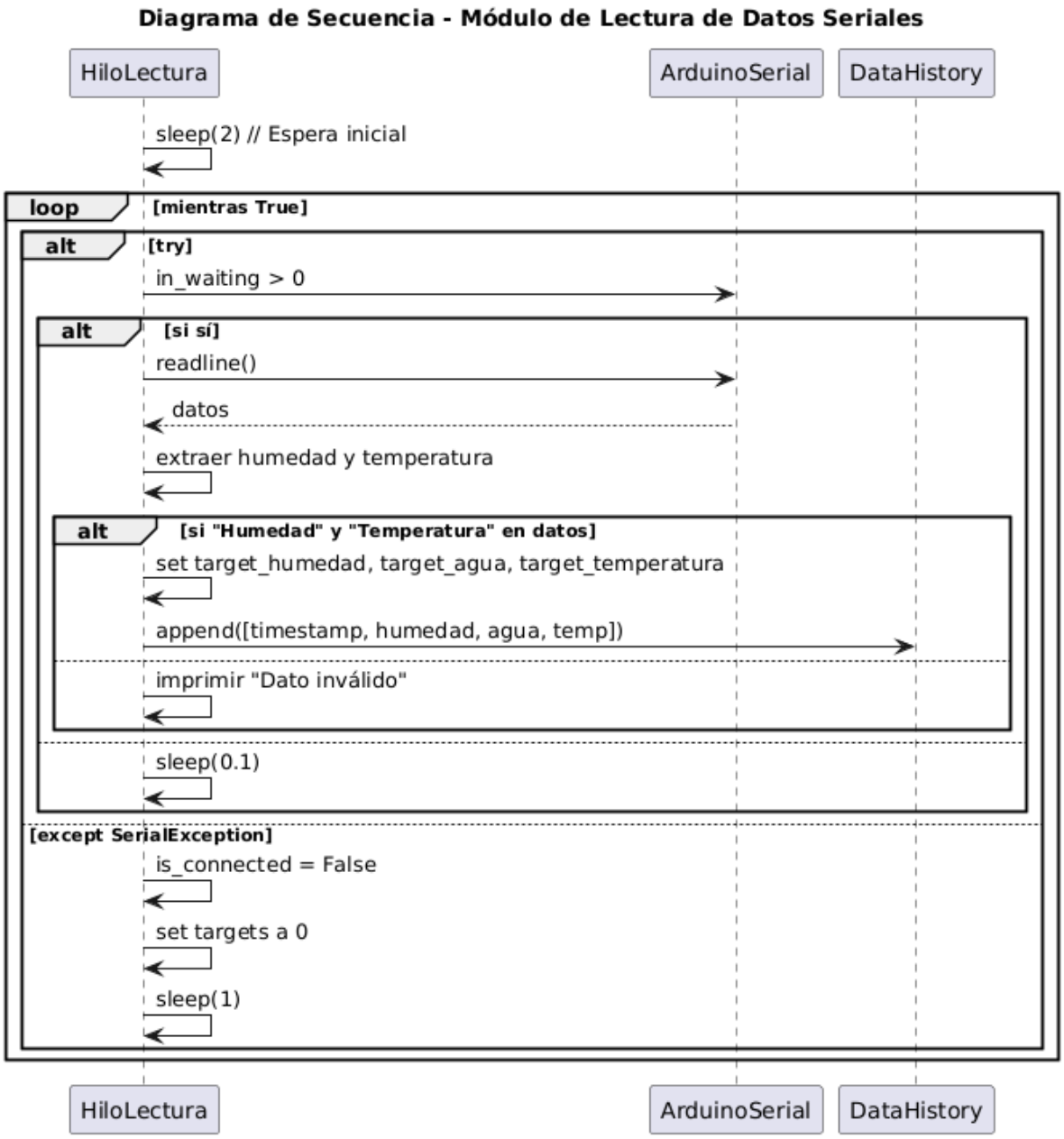
4.2. Módulo de Detección del Puerto Arduino

- **Función**: detectar_puerto_arduino()
- **Descripción**: Identifica automáticamente el puerto serial donde está conectado el Arduino usando serial.tools.list_ports.



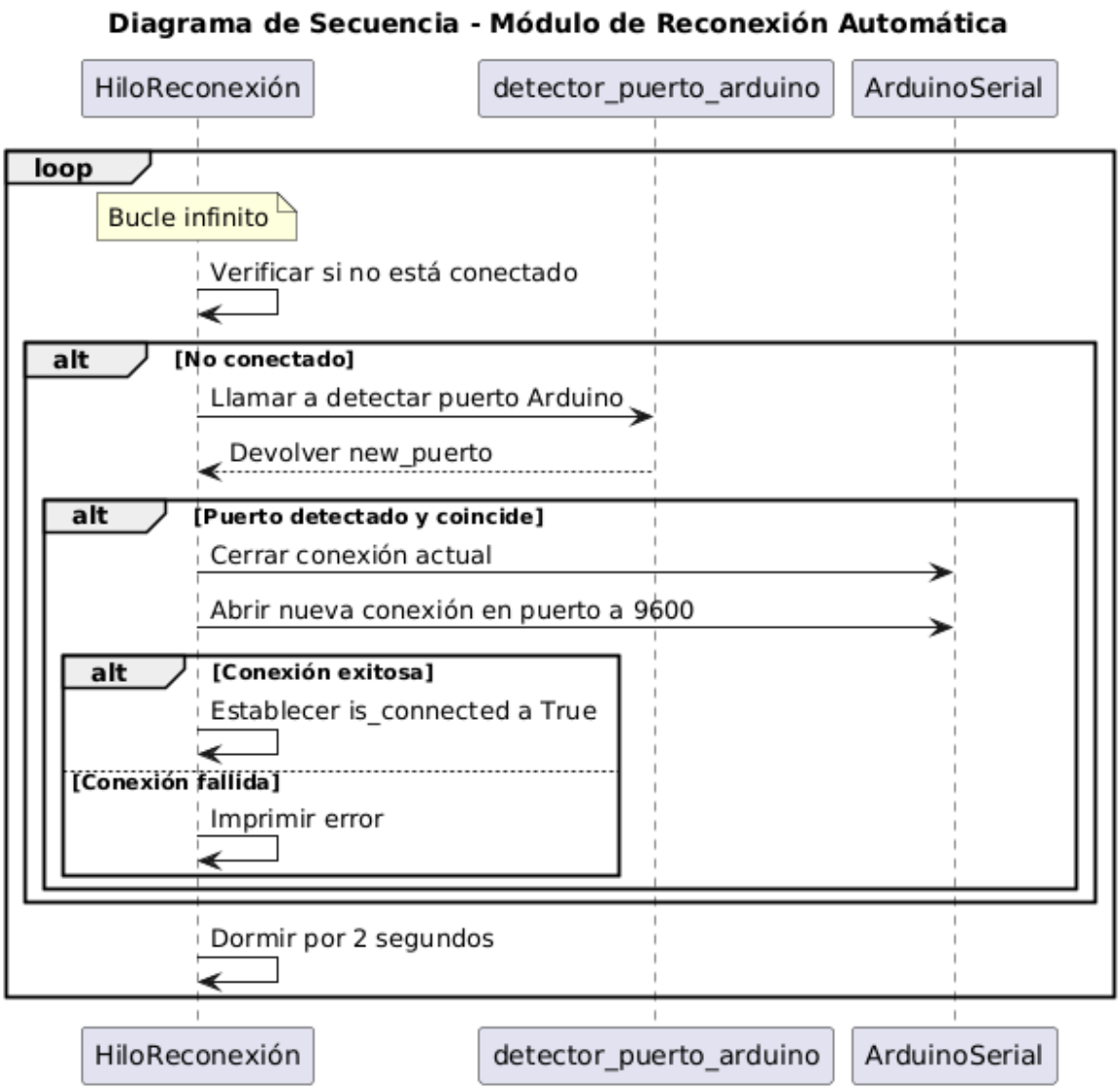
4.3. Módulo de Lectura de Datos Seriales

- **Función**: leer_serial()
- **Descripción**: Lee datos del Arduino en un hilo separado, almacenando valores de humedad y temperatura en variables globales y una lista de historial.
- **Estructura de datos**: Lista data_history con tuplas de timestamp, humedad, nivel de agua y temperatura.



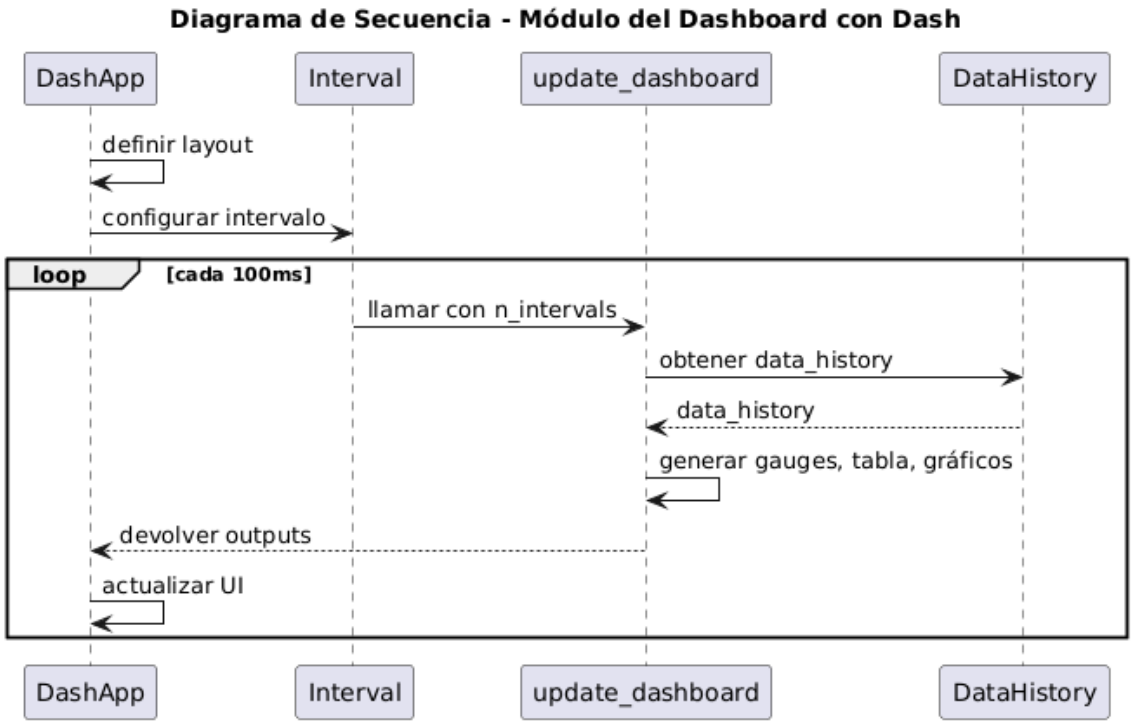
4.4. Módulo de Reconexión Automática

- **Función**: check_and_reconnect()
- **Descripción**: Verifica el estado de la conexión y reconecta automáticamente si el Arduino se desconecta.



4.5. Módulo del Dashboard con Dash

- **Descripción**: Define el layout del dashboard y actualiza componentes mediante un callback activado por un intervalo.
- **Componentes clave:**
  - Gauges para humedad, nivel de agua y temperatura.
  - Tabla de historial con las últimas 10 lecturas.
  - Gráfico de barras para tendencia de humedad.
  - Gráfico de líneas para tendencia de temperatura.
  - Alertas dinámicas basadas en valores de humedad y temperatura.



4.6. Funciones Auxiliares

- **Función**: obtener_valores_ultima_hora(data_history)
- **Descripción**: Filtra los datos históricos para devolver solo los registrados en la última hora, utilizados para gráficos de tendencia.
- **Función**: semicircular_gauge(current_value, target_value, colors, title, suffix=" %", is_active=True, max_value=100)
- **Descripción**: Genera un gauge semicircular para visualizar valores de humedad, nivel de agua y temperatura de manera intuitiva.

5. Base de Datos

- No aplica. Los datos se almacenan en memoria durante la ejecución.

6. Interfaces y Comunicación

- **Comunicación serial:** El Arduino envía datos en el formato ”Humedad: % , y Temperatura: en °C”, procesados por la aplicación Python.
- **Actualización del dashboard:** Un intervalo de 100 ms activa un callback para refrescar la interfaz.

7. Implementación de la Interfaz de Usuario

- **Estructura del layout**: Usa componentes de Dash con estilos de Bootstrap para un diseño responsivo.
- **Componentes**:
  - Gauges semicirculares para humedad, nivel de agua y temperatura.
  - Tabla de historial con las últimas 10 lecturas.
  - Gráfico de barras para tendencia de humedad.
  - Gráfico de líneas para tendencia de temperatura.
  - Alertas dinámicas sobre el estado del suelo y temperatura.

8. Manejo de Errores y Excepciones

- **Desconexión del Arduino:** Muestra una alerta e intenta reconectar automáticamente.
- **Datos inválidos:** Ignora datos no numéricos y mantiene valores previos.

9. Pruebas

- **Estrategia:** Pruebas manuales para verificar conexión, lectura de datos y actualización del dashboard.

10. Seguridad

- **Validación de datos:** Verifica que los datos sean del tipo esperado (enteros para humedad y flotantes para temperatura) antes de procesarlos.

11. Despliegue e Instalación
Dependencias:

- Pyserial
- Dash
- dash-bootstrap-components
- numpy
- plotly

Instrucciones:

Configuración del Hardware:

El Protoboard se utiliza para distribuir la alimentación (5V) y la tierra (GND) del Arduino a los sensores y la pantalla LCD. Asegúrate de usar las **líneas de alimentación (positiva)** y **líneas de tierra (negativa)** del Protoboard para todas las conexiones.

1. **Conectar la alimentación y tierra del Arduino al Protoboard:**
    - Conecta el pin **5V** del Arduino a la **línea de alimentación** del Protoboard.
    - Conecta el pin **GND** del Arduino a la **línea de tierra** del Protoboard.
2. **Conectar el sensor de humedad del suelo:**
    - Conecta el pin **A0** del módulo del sensor de humedad (módulo azul de 4 pines) al pin **A0** del Arduino.
    - Conecta el pin **VCC** del módulo a la **línea de alimentación** del Protoboard.
    - Conecta el pin **GND** del módulo a la **línea de tierra** del Protoboard.
3. **Conectar el sensor de temperatura:**
    - Conecta el pin **(+)** del sensor de temperatura a la **línea de alimentación** del Protoboard.
    - Conecta el pin **(-)** del sensor de temperatura a la **línea de tierra** del Protoboard.
    - Conecta el pin **(out)** del sensor de temperatura al pin **A1** del Arduino.
4. **Conectar la pantalla LCD:**
    - Conecta el pin **GND** de la pantalla LCD a la **línea de tierra** del Protoboard.
    - Conecta el pin **VCC** de la pantalla LCD a la **línea de alimentación** del Protoboard.
    - Conecta el pin **SDA** de la pantalla LCD al pin **A4** del Arduino (para comunicación I2C).
    - Conecta el pin **SCL** de la pantalla LCD al pin **A5** del Arduino (para comunicación I2C).

Configuración del Software:

1. **Conectar el Arduino a la computadora:**
    - Usa un cable USB para conectar el Arduino al puerto USB de tu computadora.
2. **Programar el Arduino:**

- Descarga e instala el **Arduino IDE** si aún no lo tienes (disponible en arduino.cc).
- Abre el Arduino IDE y carga el sketch del proyecto al Arduino. (Se asume que tienes un archivo de código Arduino proporcionado para este proyecto).

3. **Instalar las dependencias de Python:**
   - Asegúrate de tener **Python** instalado en tu computadora (puedes descargarlo desde python.org).
   - Abre una terminal o línea de comandos y ejecuta el siguiente comando para instalar las dependencias: Pip install pyserial dash dash-bootstrap-components numpy ploty
4. **Ejecutar la aplicación Python:**
   - Navega al directorio donde está el archivo app.py usando la terminal o línea de comandos.
   - Ejecuta el siguiente comando: Python app.py

12. Mantenimiento y Escalabilidad

- **Extensibilidad:** Diseño modular permite agregar nuevos sensores o funcionalidades.
- **Estilo:** Seguir PEP 8 para mantener la legibilidad.

13. Referencias Técnicas

- **Bibliotecas:** Documentación de pyserial, Dash, Plotly, NumPy, Bootstrap, Font Awesome.