

# PROYECTO FINAL

Juan Diego Támara Escobar

Profesor: Hugo Hernan Henao Hernandez



Programacion 2

Alexander von Humboldt

Facultad de ingenierías

Ingeniería de software

Armenia, Quindío

2024

## Descripción General del Sistema

El sistema desarrollado es un proyecto que simula el comportamiento de un sensor de temperatura, recopilando datos de forma constante y almacenándolos a una base de datos. Los datos recopilados por los usuarios a través de una interfaz web pueden ser consultados, graficados, y analizados por los usuarios en función de un sensor seleccionado. Adicionalmente, el dispositivo tiene la capacidad de simular el funcionamiento de un sensor en un congelador, y posee la facultad de realizar emisión de datos a un API.

### Componentes Principales

Simulador de Sensores (`simuladorCongelador.py`):

Este módulo simula sensores de temperatura ubicados en un congelador, generando datos de temperatura aleatorios dentro de un rango específico ( $-20^{\circ}\text{C}$  a  $-18^{\circ}\text{C}$ ).

Los datos simulados se envían al servidor API a través de solicitudes HTTP cada 3 segundos.

API (`api_app.py`):

Implementada en Flask, esta API gestiona la recepción y almacenamiento de los datos provenientes de los sensores.

Proporciona varios endpoints:

GET `/api/sensores`: Recupera los sensores registrados en la base de datos.

GET `/api/datos`: Permite filtrar los datos de sensores por fecha, hora y tipo de sensor.

POST `/api/datos`: Permite registrar nuevos datos provenientes de los sensores.

Base de Datos (Firestore):

Se utiliza Firestore (de Firebase) para almacenar los datos de los sensores. Cada entrada de datos incluye el identificador del sensor, la fecha, la hora y la temperatura medida.

Se emplean métodos como `agregar_dato`, `actualizar_dato` y `eliminar_dato` para manejar las operaciones CRUD (Crear, Leer, Actualizar, Eliminar) en la base de datos.

Interfaz Web (`index.html`):

La interfaz permite a los usuarios seleccionar un sensor y un rango de fecha y hora para visualizar los datos registrados en una tabla y gráficos interactivos mediante Chart.js.

Los usuarios pueden consultar información sobre los sensores, así como visualizar los datos históricos de temperatura en forma de gráficos dinámicos.

#### Gestión de Credenciales:

El sistema maneja credenciales de acceso a Firebase mediante un archivo JSON. Este archivo es sensible y no debe ser compartido en repositorios públicos.

Se implementaron medidas para evitar el compromiso de estas credenciales mediante el uso de .gitignore y la eliminación de credenciales del historial de Git antes de subir el proyecto a un repositorio en GitHub.

#### Conclusión

Este sistema de sensores es eficiente y seguro para el manejo de datos de temperatura en un entorno de simulación. La integración de simuladores, bases de datos, API y una interfaz web facilita la visualización y gestión de los datos de manera sencilla y accesible para el usuario. Además, las medidas tomadas para proteger las credenciales aseguran que el sistema pueda ser utilizado de manera segura en un entorno de producción sin comprometer la seguridad de la información.