

**INFORME TÉCNICO:
SISTEMA DE MONITOREO DE SENSOR DE ACELERACIÓN
DE UN CARRO DE CARRERAS**

Juan Camilo Arias Ospina

Ingeniería de Software
Semestre III

Hugo Hernan Henao Hernandez
Programación II
Corporación Universitaria Empresarial Alexander Von Humboldt

Armenia, Quindío
2024

SISTEMA DE MONITOR CON FLASK Y FIREBASE

El sistema desarrollado es una solución integral de monitoreo de sensores que combina captura, almacenamiento y visualización de datos en tiempo real. Este sistema está diseñado para trabajar con un sensor simulado que emula el comportamiento de la aceleración de un automóvil. El sensor genera datos continuamente y los comunica mediante una API REST desarrollada con el framework Flask, proporcionando una interfaz eficiente para la transferencia y gestión de información.

La información generada por el sensor es almacenada de forma segura en una base de datos Firebase, aprovechando sus capacidades de escalabilidad, confiabilidad y acceso remoto en tiempo real. La arquitectura del sistema asegura que los datos sean consistentes y estén siempre disponibles para los usuarios.

Para facilitar la interacción y comprensión de los datos recopilados, se ha desarrollado una interfaz web intuitiva y responsiva que permite a los usuarios consultar la información histórica almacenada, así como visualizarla gráficamente en tiempo real. La representación gráfica incluye opciones interactivas, como selección de períodos específicos, filtros por tipo de datos y vistas dinámicas para analizar tendencias de aceleración.

1. COMPONENTES PRINCIPALES

Simulador de Sensor: Un script en Python que genera y envía datos de aceleración de forma realista y periódica.

API REST: Un servicio backend desarrollado con Flask que recibe, valida y almacena los datos en Firebase.

Interfaz Web: Una página web interactiva que permite consultar y visualizar los datos en gráficos de línea y tablas.

Base de Datos Firebase: Un sistema en la nube para almacenar los datos de sensores con alta disponibilidad y rendimiento.

2. OBJETIVOS

Objetivo General

Desarrollar un sistema completo de monitoreo de sensores que permita la captura, almacenamiento y visualización de datos utilizando tecnologías web modernas.

Objetivos Específicos

1. Simular el funcionamiento de un sensor que envía datos a una API desarrollada con Flask.
2. Implementar una API que registre los datos enviados por el sensor en Firebase.
3. Diseñar una interfaz web para consultar y graficar los datos registrados, filtrados por sensor y rango de fechas.

3. IMPLEMENTACIÓN DEL SISTEMA

3.1. Simulador de Sensor

El simulador de sensor es un componente clave desarrollado en Python, diseñado para emular el comportamiento de un sensor real que mide la aceleración de un vehículo. Este simulador cumple con las siguientes características:

Identificador del Sensor:

Cada simulador está asociado con un identificador único, generado combinando las iniciales del estudiante y un número secuencial. Esto permite diferenciar fácilmente los datos provenientes de múltiples simuladores en caso de una implementación a mayor escala o de pruebas con diversos dispositivos virtuales.

Datos Generados:

Los datos de aceleración se expresan en metros por segundo al cuadrado (m/s^2) y varían dentro de un rango realista de -10 a 10. Este rango refleja comportamientos típicos de aceleración y desaceleración en un contexto vehicular.

Para añadir mayor realismo, los valores generados incorporan ligeras fluctuaciones aleatorias que simulan condiciones de conducción variadas, como cambios repentinos en la aceleración debido al frenado, aceleración brusca o terrenos irregulares.

Periodicidad y Configuración:

Cada dato generado se envía automáticamente a través de la API REST hacia la base de datos Firebase, asegurando una transmisión continua y confiable.

3.2. API REST

La API REST, desarrollada con Flask, expone un endpoint para recibir datos de sensores y almacenarlos en Firebase:



- **Validación de datos:** Se asegura que los datos contengan todos los campos requeridos antes de su almacenamiento.
- **Almacenamiento en Firebase:** Integración con Firebase Admin SDK para registrar los datos en la base de datos.

3.3. Interfaz Web

Se implementó una página web como interfaz principal para la consulta y visualización de los datos generados por el simulador de sensor. Este componente permite a los usuarios interactuar con la información de manera intuitiva y visualmente clara. Las principales funcionalidades y características de la interfaz son las siguientes:

1. Formulario de Consulta:

La página incluye un formulario interactivo que permite a los usuarios filtrar los datos almacenados en la base de datos Firebase.

Los filtros disponibles incluyen:

- **Identificador del sensor:** Permite seleccionar registros específicos asociados con un sensor en particular, facilitando el análisis segmentado.
- **Rango de fechas:** Los usuarios pueden definir un período de tiempo específico para limitar los datos consultados, agilizando la búsqueda de información relevante y mejorando la experiencia de usuario.

2. Gráficos Interactivos:

- Los datos filtrados se visualizan mediante gráficos de línea interactivos generados con la biblioteca chart.js para que la gráfica se vea mucho mejor.
- El gráfico permiten:
- **Actualización en tiempo real:** La interfaz es capaz de reflejar nuevos datos que se almacenen en Firebase, brindando una experiencia de monitoreo actualizada.

3. Tabla de Datos

Además de los gráficos, la interfaz presenta los datos consultados en una tabla ordenada y legible.

Las características principales de la tabla incluyen:

Columnas detalladas: Muestra información como el identificador del sensor, el valor de aceleración (en m/s^2) y la marca temporal del registro.

3.4 Base de Datos Firebase

- **Estructura:** Cada registro contiene los campos: fecha, hora, identificador del sensor, y valor.
- **Ventajas:** Escalabilidad y alta disponibilidad de Firebase para almacenamiento de datos en tiempo real.

4. USO DE LA API

- **Endpoint:** /api/sensor_data
- **Método HTTP:** POST

```

1  {
2    "data": [
3      {
4        "fecha": "2024-12-01",
5        "hora": "20:22:39",
6        "id": "-OD3xaL4P52UaY3ctQmy",
7        "idSensor": "JCAO4275",
8        "valor": 1.92
9      },
10     {
11       "fecha": "2024-12-01",
12       "hora": "20:22:39",
13       "id": "-OD3xiBbXgzfAaHtseHX",
14       "idSensor": "JCAO4275",
15       "valor": 4.2
16     },
17     {
18       "fecha": "2024-12-01",
19       "hora": "20:22:48",
20       "id": "-OD3y77vKaAwdAZHjrhy",
21       "idSensor": "JCAO4275",
22       "valor": 6.11
23     },
24     {
25       "fecha": "2024-12-01",
26       "hora": "20:23:11",
27       "id": "-OD3yEzmDwMxxlaPamNj",
28       "idSensor": "JCAO4275",
29       "valor": 6.46
30     },
31     {
32       "fecha": "2024-12-01",
33       "hora": "20:24:58",
34       "id": "-OD3yMwzJG7EvG7b0bt9",
35       "idSensor": "JCAO4275",
36       "valor": 6.2
37     },
38     {
39       "fecha": "2024-12-01",
40       "hora": "20:25:30",
41       "id": "-OD3yMwzJG7EvG7b0bt9",
42       "idSensor": "JCAO4275",
43       "valor": 4.46
44     },
45     {
46       "fecha": "2024-12-01",
47       "hora": "20:26:02",
48       "id": "-OD3yMwzJG7EvG7b0bt9",
49       "idSensor": "JCAO4275",
50       "valor": 2.57
51     },
52     {
53       "fecha": "2024-12-01",
54       "hora": "20:26:02",
55       "id": "-OD3yMwzJG7EvG7b0bt9",
56       "idSensor": "JCAO4275",
57       "valor": 2.57
58     }
59   ]
60 }

```

4.1 Uso de la Interfaz Web

1. Abrir el navegador y acceder a la URL de la interfaz.

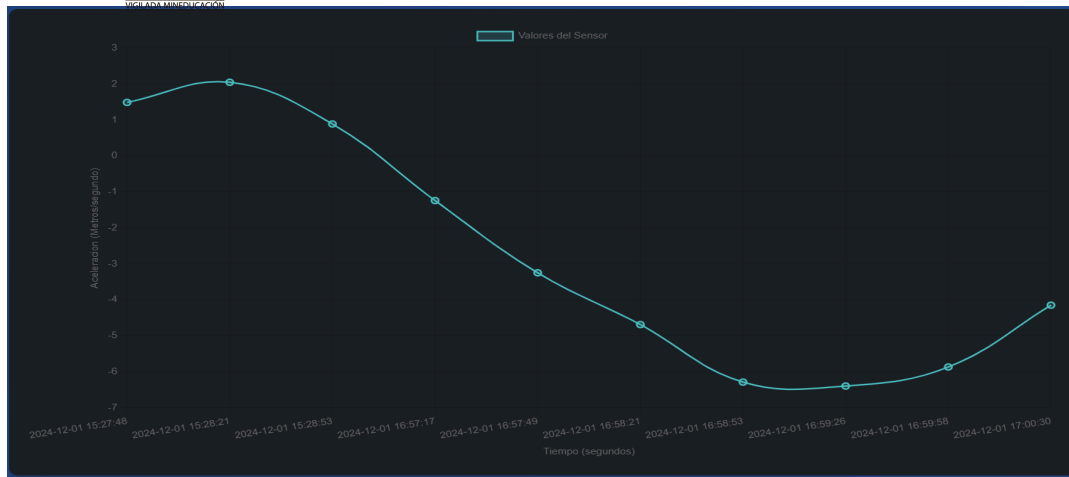


2. Seleccionar el identificador del sensor y rango de fechas.

ID del Sensor: Fecha: Hora:

3. Visualizar los datos graficados y en formato de tabla.

ID	Fecha	Hora	ID del Sensor	Valor Aceleración
-OD3xaL4P52UaY3ctQmy	2024-12-01	20:22:39	JCAO4275	3.94
-OD3xiBbXgzfAaHtseHX	2024-12-01	20:23:11	JCAO4275	5.44
-OD3y77vKaAwdAZHjrhy	2024-12-01	20:24:58	JCAO4275	-1.89
-OD3yEzmDwMxxlaPamNj	2024-12-01	20:25:30	JCAO4275	-3.84
-OD3yMwzJG7EvG7b0bt9	2024-12-01	20:26:02	JCAO4275	-5.75



5. RESULTADOS.

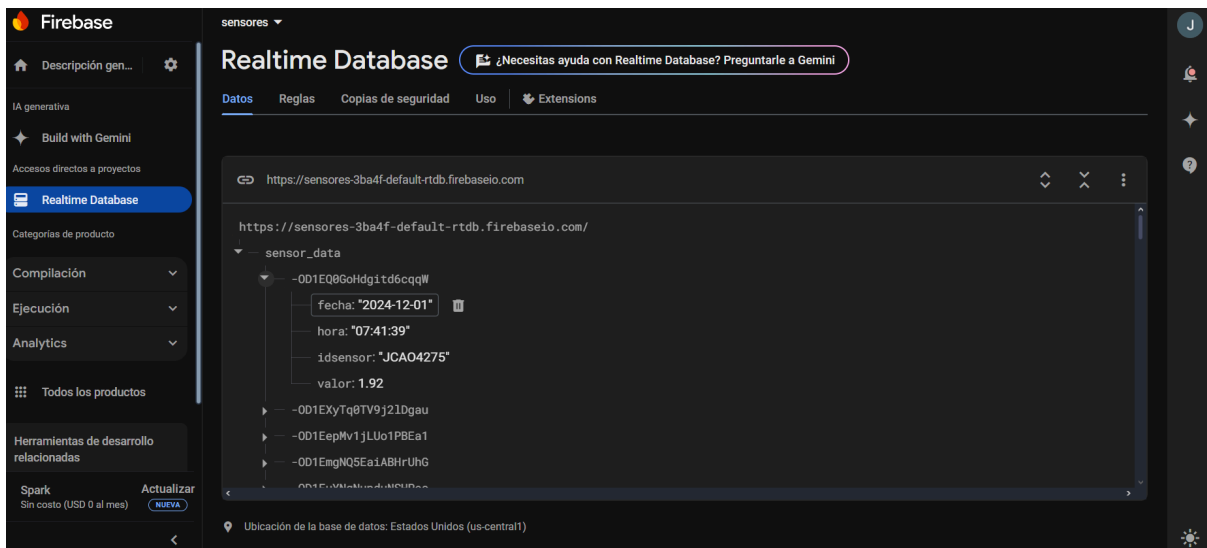
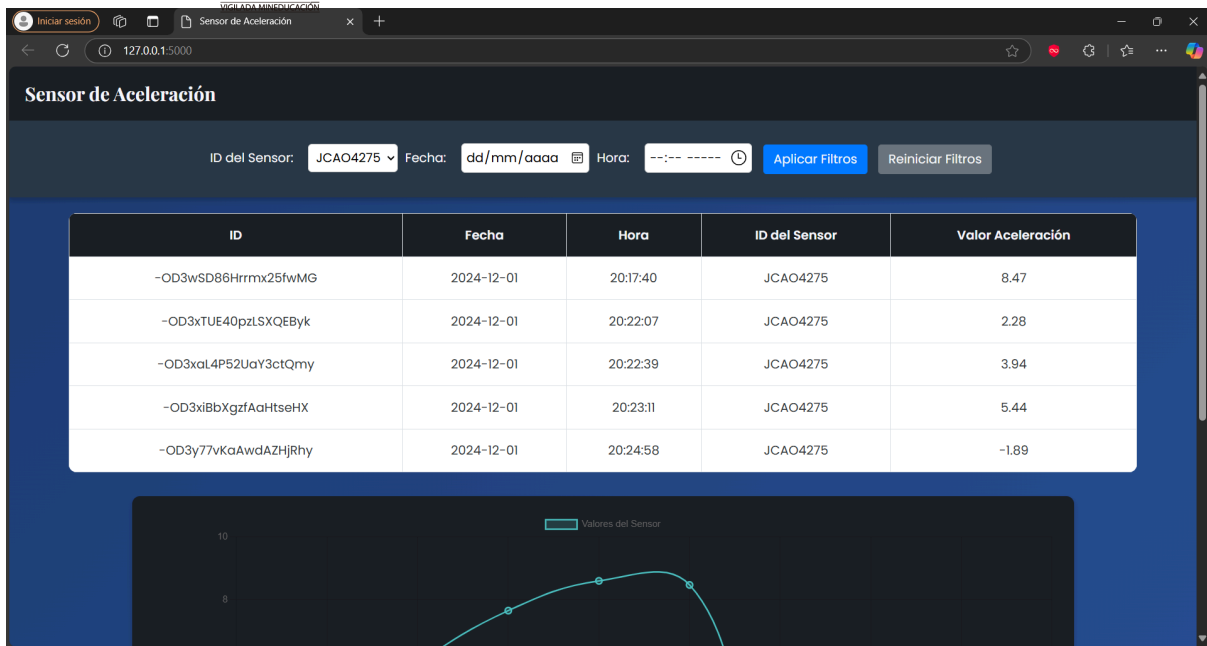
```
Run  api x  sensorAceleracion_simulator x

127.0.0.1 - - [01/Dec/2024 16:56:26] "GET /api/sensor_data HTTP/1.1" 200 -
127.0.0.1 - - [01/Dec/2024 16:56:31] "GET /api/sensor_data HTTP/1.1" 200 -
127.0.0.1 - - [01/Dec/2024 16:56:36] "GET /api/sensor_data HTTP/1.1" 200 -
127.0.0.1 - - [01/Dec/2024 16:56:41] "GET /api/sensor_data HTTP/1.1" 200 -
127.0.0.1 - - [01/Dec/2024 16:56:46] "GET /api/sensor_data HTTP/1.1" 200 -
127.0.0.1 - - [01/Dec/2024 16:56:51] "GET /api/sensor_data HTTP/1.1" 200 -
127.0.0.1 - - [01/Dec/2024 16:56:56] "GET /api/sensor_data HTTP/1.1" 200 -
127.0.0.1 - - [01/Dec/2024 16:57:01] "GET /api/sensor_data HTTP/1.1" 200 -
127.0.0.1 - - [01/Dec/2024 16:57:07] "GET /api/sensor_data HTTP/1.1" 200 -
```

```
Run  api x  sensorAceleracion_simulator x

C:\Users\Rafael\IdeaProjects\proyecto_sensores\.venv\Scripts\python.exe C:\Users\Rafael\IdeaProjects\proyecto_sensores\simulator\sensorAceleracion_simulator.py
Iniciando simulación de sensor de aceleración para el sensor JCA04275
Lectura enviada correctamente: {'idsensor': 'JCA04275', 'valor': -1.24}
Lectura enviada correctamente: {'idsensor': 'JCA04275', 'valor': -3.25}
Lectura enviada correctamente: {'idsensor': 'JCA04275', 'valor': -4.69}

proyecto_sensores > docs 22184 CRLF UTF-8 4 spaces
```



6. CONCLUSIONES

En resumen, este sistema representa una solución integral que combina simulación, almacenamiento en la nube, desarrollo backend y visualización interactiva, sentando una base sólida para aplicaciones más complejas en el ámbito del monitoreo de datos en tiempo real.