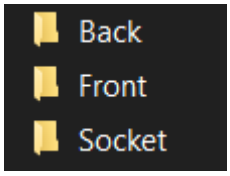
Visualización de datos en tiempo real con tecnologías Web

Instrucciones de instalación y ejecución

- Descomprima el .zip o ejecute el comando:
  - *git clone* https://github.com/alfonsoleonm/Reto-FullStack-JS-GS.git
- 3 carpetas se encontrarán en el directorio => Back, Front y Socket:
  - 

- Por favor ejecute el comando “npm install” en cada una de ellas para instalar las dependencias necesarias.

Ejecución:

- Ejecute los comandos siguientes:
  - En la carpeta *Socket:* “node app.js”
  - En la carpeta *Back:* “node src/index.js”
  - En la carpeta Front: “ng serve -o” NOTA: esto abrirá la aplicación angular en su navegador automáticamente en http://localhost:4200/

Tecnologías utilizadas:

Angular CLI: 14.2.1

Angular Material “^14.2.0”

Node: 18.12.1

Chart.js 4.4.1

socket.io-client 4.7.2

Package Manager: npm 8.19.2

OS: win32 x64

MongoDB Atlas versión de prueba (**Nota**: Dada la limitación de almacenamiento en la base de datos y considerando que estamos utilizando la versión de prueba de MongoDB Atlas, esta base de datos se ha empleado exclusivamente para este proyecto, cumpliendo satisfactoriamente con su propósito. Como una mejora significativa en esta evaluación, se podría perfeccionar el acceso a la base de datos mediante la implementación de un método más seguro de autenticación.)

Nota sobre Seguridad:

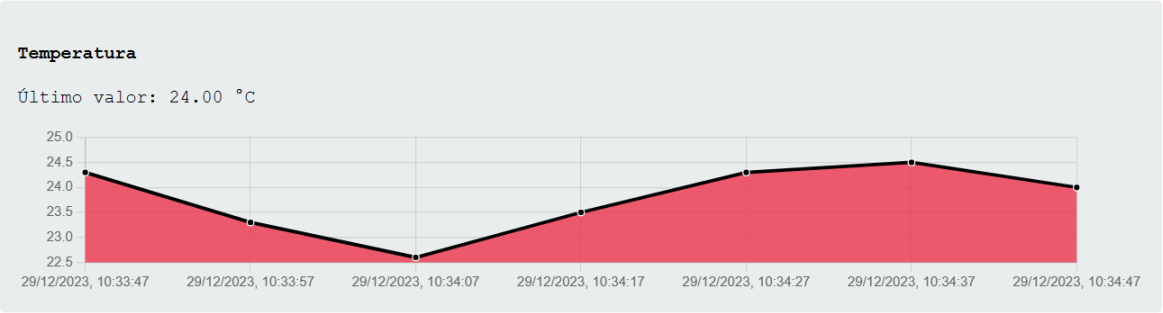
Para esta prueba, se ha omitido la consideración de aspectos de seguridad, como la gestión de credenciales para la API REST o el broker Socket.io, de acuerdo con los requisitos establecidos. Sin embargo, es importante destacar que, en un entorno de producción, es crucial abordar ciertos problemas de seguridad para garantizar la integridad y la confidencialidad de los datos. A continuación, se mencionan algunos puntos clave que deberían considerarse:

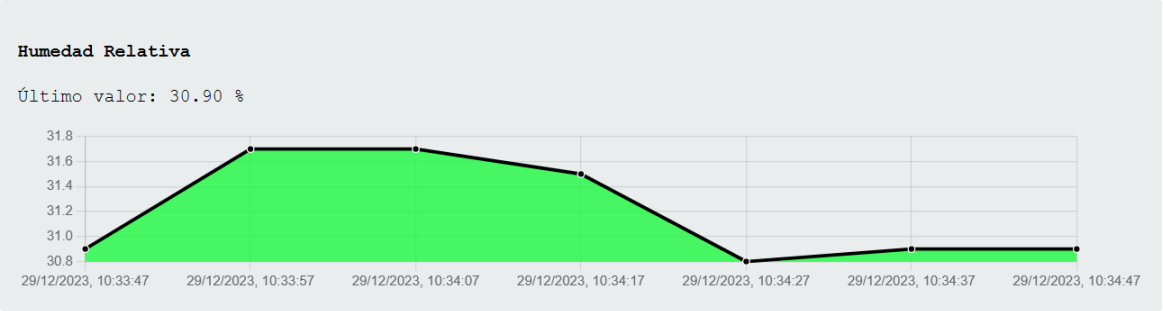
1. **Gestión de Credenciales:**
    - **Problema:** La falta de consideración de credenciales puede dejar expuesta la aplicación a posibles amenazas de seguridad.
    - **Solución:** Implementar una gestión segura de credenciales, como el uso de variables de entorno o soluciones de gestión de secretos, para proteger las credenciales de API REST y Socket.io.
2. **Seguridad de la Capa de Transporte:**

- **Problema:** La transmisión de datos sin cifrar puede exponer información sensible durante la comunicación.
- **Solución:** Utilizar protocolos seguros como HTTPS para la API REST y WSS (WebSocket Secure) para Socket.io, garantizando la encriptación de extremo a extremo.

3. **Validación de Entradas:**
    - **Problema:** La falta de validación de datos de entrada puede abrir la puerta a ataques de inyección.
    - **Solución:** Implementar una estricta validación de entrada y sanitización para prevenir posibles ataques de inyección, como SQL injection o Cross-Site Scripting (XSS).

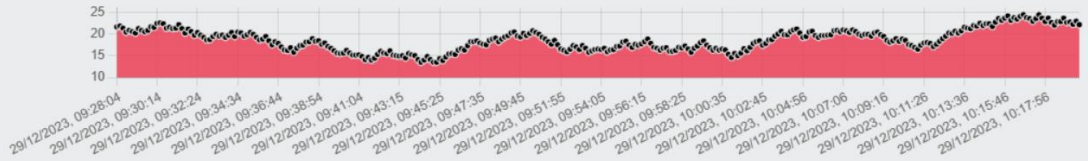
Screenshots:





Temperatura

Último valor: 22.10 °C

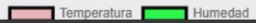


Humedad Relativa

Último valor: 32.60 %

Datos Históricos

Seleccione el tiempo: 15 minutos



Datos Históricos

Seleccione el tiempo: 15 minutos