SPRINT 0

Documento técnico: Lógica de negocio, API y base de datos.

Autor: Greysy Burgos Salazar

1. Visión General

El proyecto implementa un sistema IoT distribuido capaz de captar, transmitir, almacenar y visualizar datos ambientales (principalmente CO_2) mediante una arquitectura modular basada en:

- Dispositivo físico: beacon BLE nRF52840 (Arduino)
- Aplicación móvil: Android (Java)
- Servidor REST: Node.js + Express
- Base de datos: MySQL local
- Interfaz web: HTML + CSS + JS

Cada componente cumple una función concreta dentro del flujo de datos: la placa mide y emite, la app detecta y publica, el backend guarda, y la web muestra.

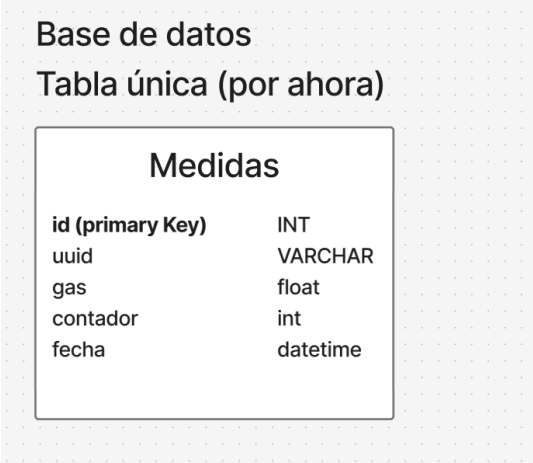
Tambien se presentaran los diseños del backend y el frontend que consume la API REST

El flujo completo del sistema conecta todas las capas de comunicación:

Dispositivo BLE → Aplicación móvil → API REST → Lógica de negocio → Base de datos → Interfaz web, desarrollado en un servidor local.

2. Base de Datos

El sistema usa una base de datos MySQL con una única tabla principal `medidas`.



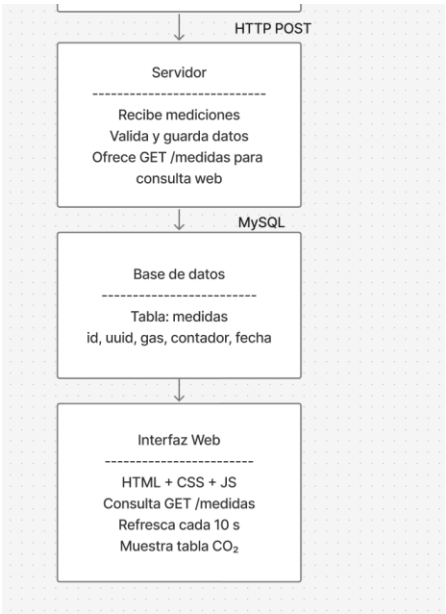
Campos principales:
- id: identificador autoincremental.
- uuid: nombre o identificador del sensor BLE.
- gas: valor del CO_2.
- contador: número de emisión.
- fecha: registro automático de la hora de inserción.

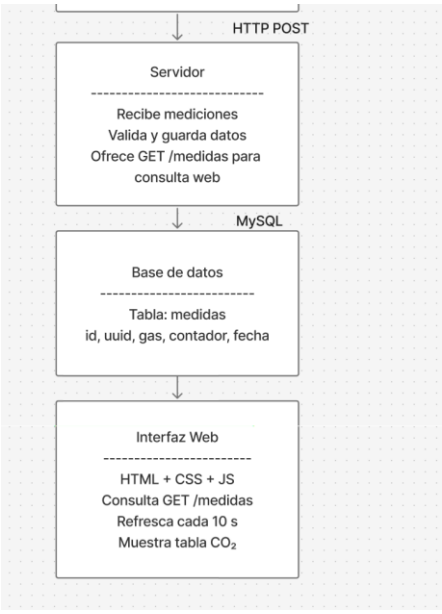
3. Lógica de Negocio (Servidor)

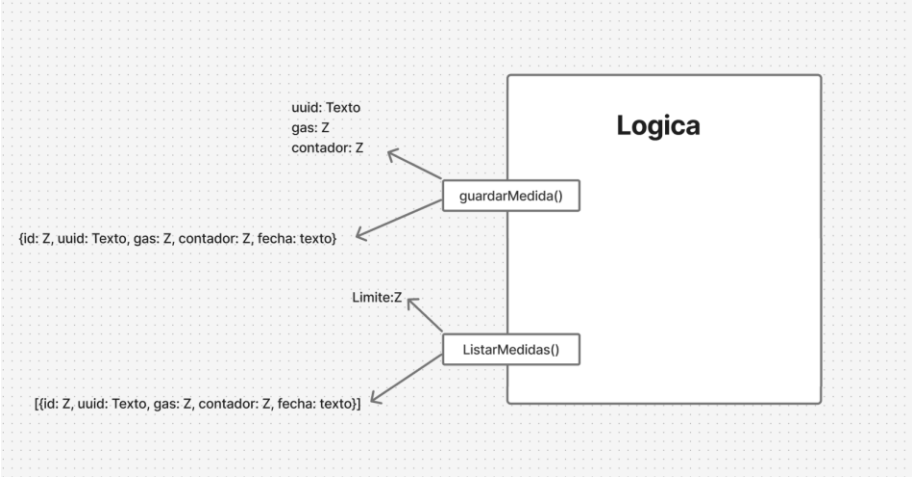
La clase `Logica.js` gestiona todas las operaciones MySQL usando `mysql2/promise`. Actúa como capa de acceso a datos, separada del resto del sistema.

Métodos principales:
- guardarMedida(uuid, gas, contador): inserta una nueva medición.
- listarMedidas(limit): devuelve las últimas mediciones.

Arquitectura en capas:
• Logica.js → Modelo (acceso a datos)
• ReglasREST.js → Controlador (API)
• mainServidorREST.js → inicia el servidor







4. API REST

Desarrollada con Express.js. Ofrece dos rutas principales:

POST /medida → inserta una nueva medición enviada por Android.
GET /medidas → devuelve las últimas mediciones almacenadas.

Ejemplo JSON:
{ uuid: 'GTI-3A', gas: 420.5, contador: 1 }

5. Aplicación Android

La app Android escanea dispositivos BLE con BluetoothLeScanner. Filtra la placa por nombre, decodifica la trama iBeacon (UUID, Major, Minor, TxPower) y envía los datos al servidor REST.

Relación de campos:
- Minor → gas (CO_2)
- Major → contador (número de emisión)

6. Interfaz Web

Construida con HTML + CSS + JS. Consulta la API REST cada 10 segundos mediante fetch() y actualiza la tabla con los datos más recientes (ID, UUID, gas, fecha, hora).