 Icono

El contenido generado por IA puede ser incorrecto.

|  |
| --- |
| **Diseños Backend**  **SPRINT 0** |

**PROYECTO APLICACIONES DE BIOMETRÍA Y MEDIO AMBIENTE**

**Nombre del Alumno:** Alan Guevara Martínez

**Enlace Git:** [**https://github.com/ALANGMupv/ProyectoBiometria2025.git**](https://github.com/ALANGMupv/ProyectoBiometria2025.git)

**Enlace trello:** [**https://trello.com/invite/b/68c9273bf08b76ab09c96593/ATTI19e0d5a059994ed24df6c6455b577bae3E1C4F40/gtiproyectobiometria**](https://trello.com/invite/b/68c9273bf08b76ab09c96593/ATTI19e0d5a059994ed24df6c6455b577bae3E1C4F40/gtiproyectobiometria)

**Curso:** 2025/2026

# 

**ÍNDICE / TABLA DE CONTENIDOS**

[Introducción 3](#_Toc210163751)

[MainActivity (archivo principal) 3](#_Toc210163752)

[Clase Utilidades 4](#_Toc210163753)

[Clase TramaIBeacon 5](#_Toc210163754)

[LogicaFake (clase auxiliar creada por mí) 7](#_Toc210163755)

**TABLA DE ILUSTRACIONES / DISEÑOS**

[Ilustración 1: Diseño MainAcitivity 4](#_Toc210163756)

[Ilustración 2: Diseño Utilidades 5](#_Toc210163757)

[Ilustración 3: Diseño TramaIBeacon 6](#_Toc210163758)

[Ilustración 4: Diseño LogicaFake 7](#_Toc210163759)

##### Introducción

En este documento se presentan los diseños y la documentación del backend en **Node.js con Express** y del **front-end en JavaScript (cliente)** que consume la API REST.  
El flujo completo conecta los dispositivos BLE → móvil → API REST → lógica de negocio → base de datos → interfaz web.

**Flujo simplificado:**

1. El cliente hace una petición → API REST.
2. ReglasREST valida la entrada → (controlador).
3. Llama a Logica.guardarMedida() o Logica.listarMedidas() → (lógica de negocio).
4. La lógica ejecuta SQL sobre MySQL.
5. Devuelve los datos al cliente en JSON.

Los diagramas completos de diseño pueden consultarse en el archivo de Figma correspondiente a **Diseño Backend y Frontend**:

<https://www.figma.com/board/OhXGiBzKXEmR9dCqy3RwvB/Dise%C3%B1o-Backend-y-Frontend?node-id=0-1&t=TAxnJyZzxQ48UPCf-1>

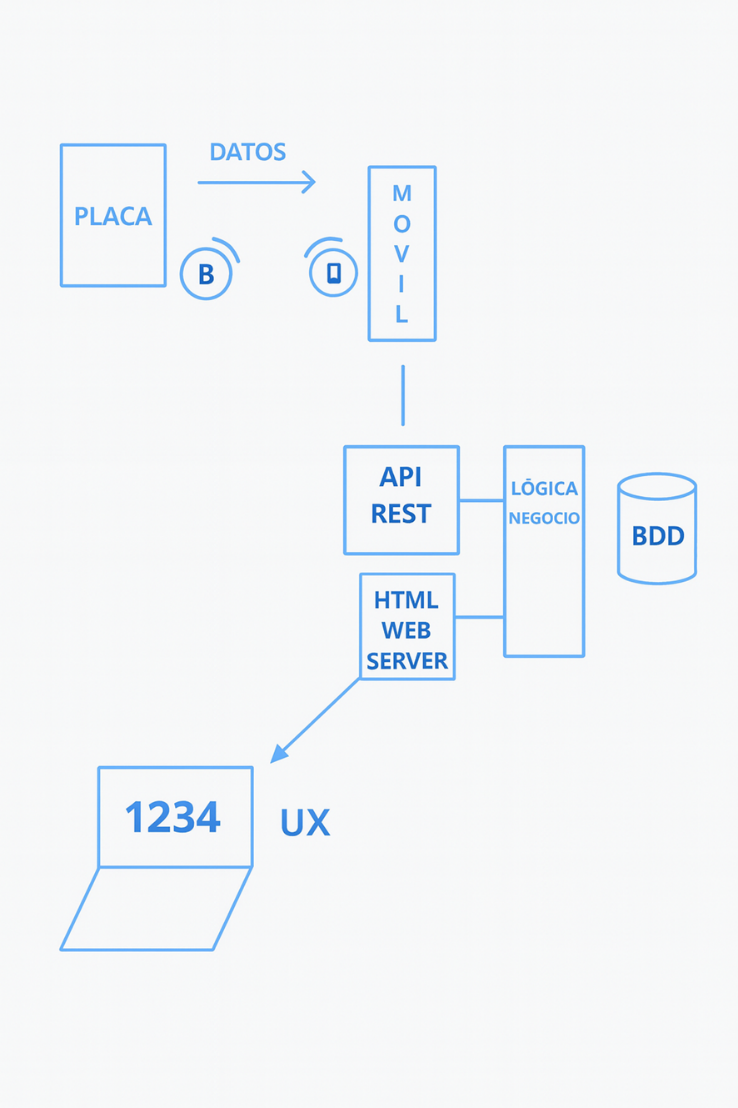


Ilustración 1: Diseño Flujo Completo

##### mainServidorREST.js (punto de entrada del servidor)

Archivo principal que configura y arranca el servidor REST con Express.  
Se encarga de inicializar la infraestructura, aplicar middlewares y montar las rutas.

**Responsabilidades principales:**

* Configura el servidor Express.
* Aplica middlewares (express.json(), cors()).
* Instancia la lógica de negocio (Logica).
* Monta las rutas definidas en ReglasREST.
* Arranca el servidor en el puerto definido (en Plesk lo asigna el sistema).

##### ReglasREST.js (controladores / endpoints)

Archivo que define los endpoints REST de la API.  
No accede directamente a la base de datos, sino que delega en la lógica de negocio.

**Endpoints principales:**

* **POST /medida:**
  + Recibe un JSON con { uuid, gas, valor, contador }.
  + Valida los campos obligatorios.
  + Llama a logica.guardarMedida(...).
  + Devuelve confirmación en JSON con la medida guardada o error.
* **GET /medidas:**
  + Permite obtener las últimas medidas registradas (?limit=N).
  + Llama a logica.listarMedidas(limit).
  + Devuelve un array JSON con las medidas.