Proyecto Aplicaciones de Biometría y Medio Ambiente

Sprint 0 – Documentación Técnica

Nombre del Alumno: Alejandro Vázquez Remes
Fecha: 10 de octubre de 2025
Curso: 2025 / 2026
Enlace GitHub: https://github.com/exby04/**Proyecto-Beacons**
Enlace Trello:
https://trello.com/invite/b/68c9247abcb37b54c03d9a7a/ATTIf4b76b794f03c17c3281edce0e717d32D4649957/sprint-0

ÍNDICE / TABLA DE CONTENIDOS

1. 1. Introducción
2. 2. HolaMundoIBeacon (archivo principal)
3. 3. Clase EmisoraBLE
4. 4. Clase LED
5. 5. Clase Medidor
6. 6. Clase Publicador
7. 7. Clase PuertoSerie
8. 8. Clase ServicioEnEmisora
9. 9. Código Java (Android)
10. 10. Código SQL (Base de datos)
11. 11. Tabla de Ilustraciones / Diseños

Introducción

Este documento describe la estructura modular del proyecto de Biometría y Medio Ambiente. En el Sprint 0 se desarrollaron las clases principales en C++ para Arduino BLE, las clases Java para la aplicación Android, y la base de datos SQL utilizada por el servidor Node.js. Cada módulo tiene responsabilidades definidas que permiten la escalabilidad y el mantenimiento del sistema.

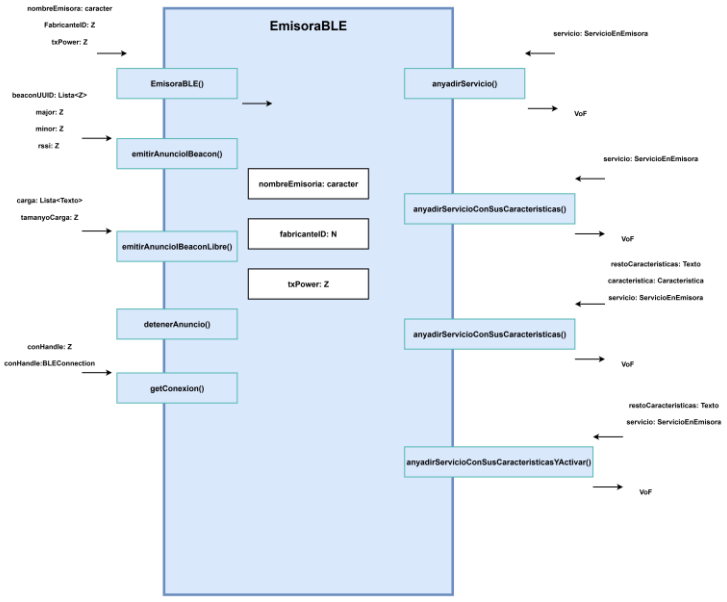
HolaMundoIBeacon

Contiene las funciones setup() y loop(), que coordinan la inicialización del sistema, la lectura de sensores y el envío de datos mediante beacons BLE. Es el punto de entrada del programa.

Clase EmisoraBLE

Gestiona el chip Bluetooth (nRF52) y controla la emisión de anuncios BLE tipo iBeacon.

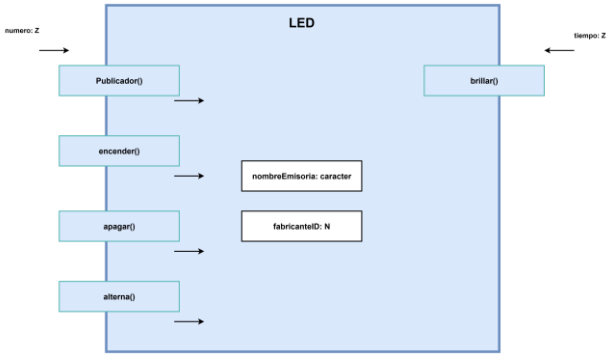
Métodos destacados:
- encenderEmisora(): inicializa el módulo BLE.
- detenerAnuncio(): detiene la transmisión.
- emitirAnuncioIBeacon(): emite una trama estándar (UUID, major, minor, RSSI).



Clase LED

Controla el LED físico de la placa para indicar el estado del dispositivo.

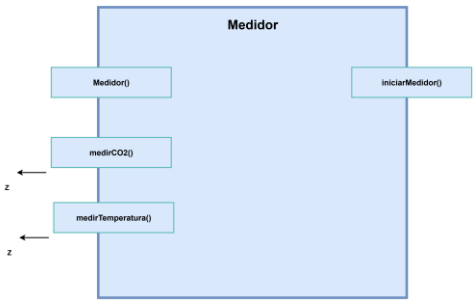
Métodos principales:
- encender(), apagar(), alternar(), brillar().



Clase Medidor

Simula sensores ambientales (CO_2 y temperatura).

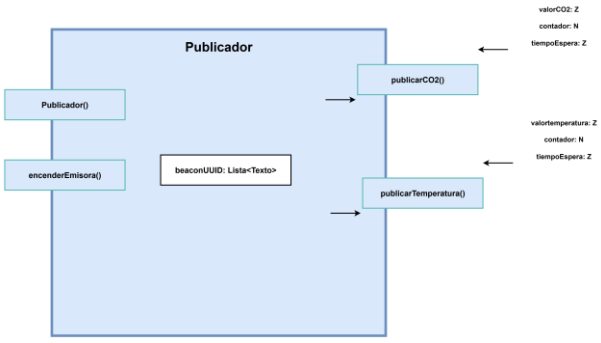
Métodos:
- medirCO2(): devuelve un valor entero simulado.
- medirTemperatura(): devuelve temperatura simulada.



Clase Publicador

Empaqueta y envía los valores medidos usando la emisora BLE.

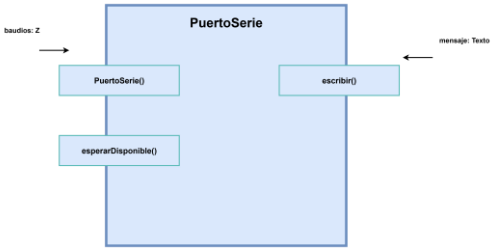
Métodos:
- publicarCO2(), publicarTemperatura().



Clase PuertoSerie

Gestiona la comunicación serie para depuración y monitoreo.

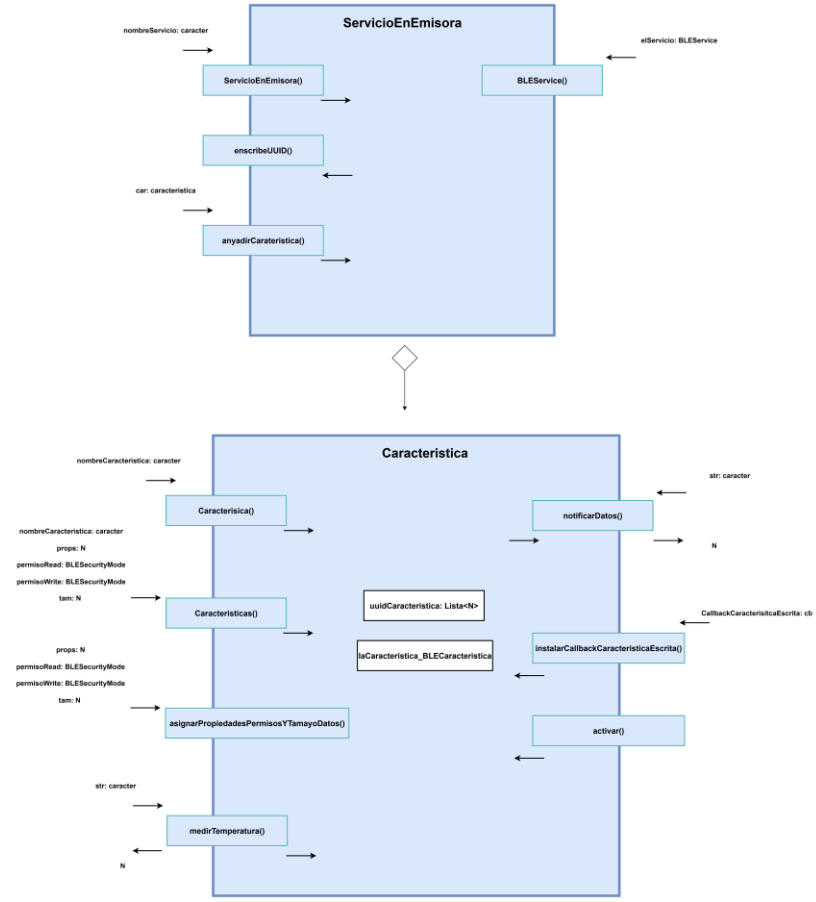
Métodos:
- escribir(), esperarDisponible().



Clase ServicioEnEmisora

Permite crear servicios BLE personalizados con características asociadas.

Incluye una clase interna Caracteristica con métodos para escribir, notificar y activar características BLE.

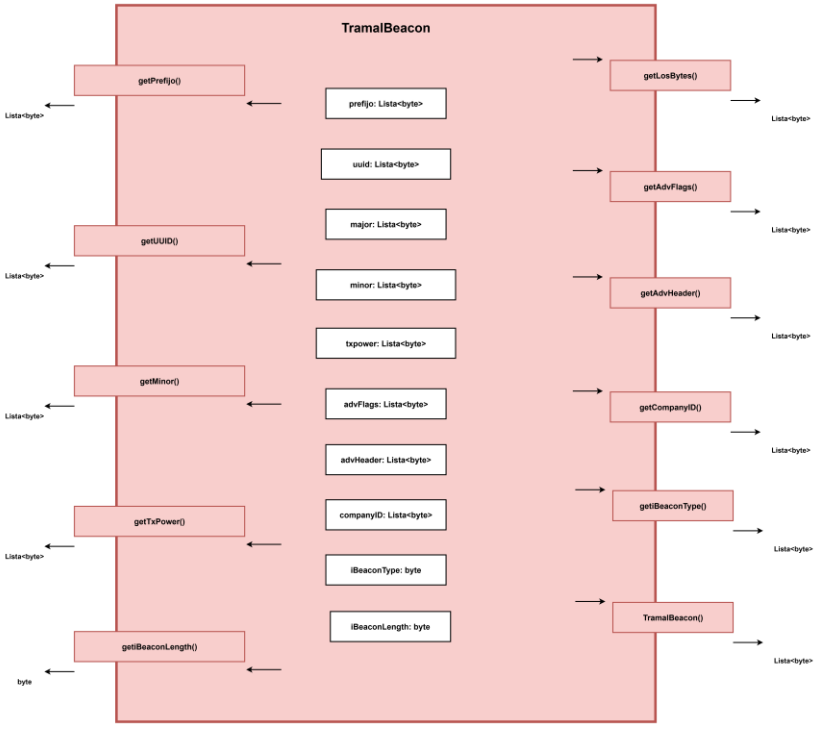


Código Java (Android)

El módulo Android se encarga de escanear los beacons emitidos por Arduino, procesar los datos y enviarlos al servidor mediante peticiones HTTP. A continuación, se describen las clases principales:

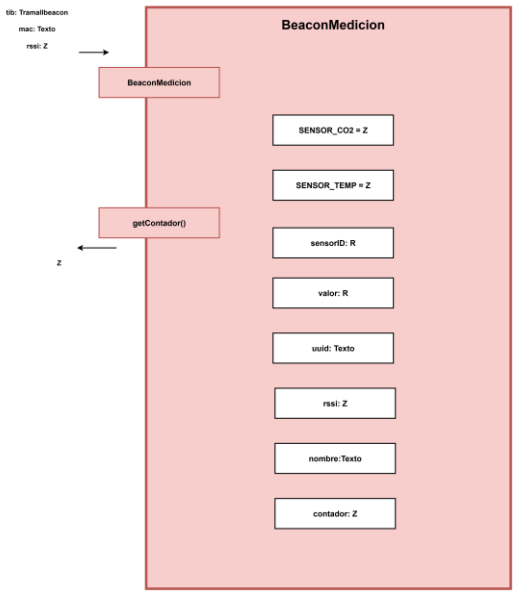
TramaIBeacon.java

Interpreta los bytes recibidos por BLE y extrae el UUID, major, minor y potencia (txPower).



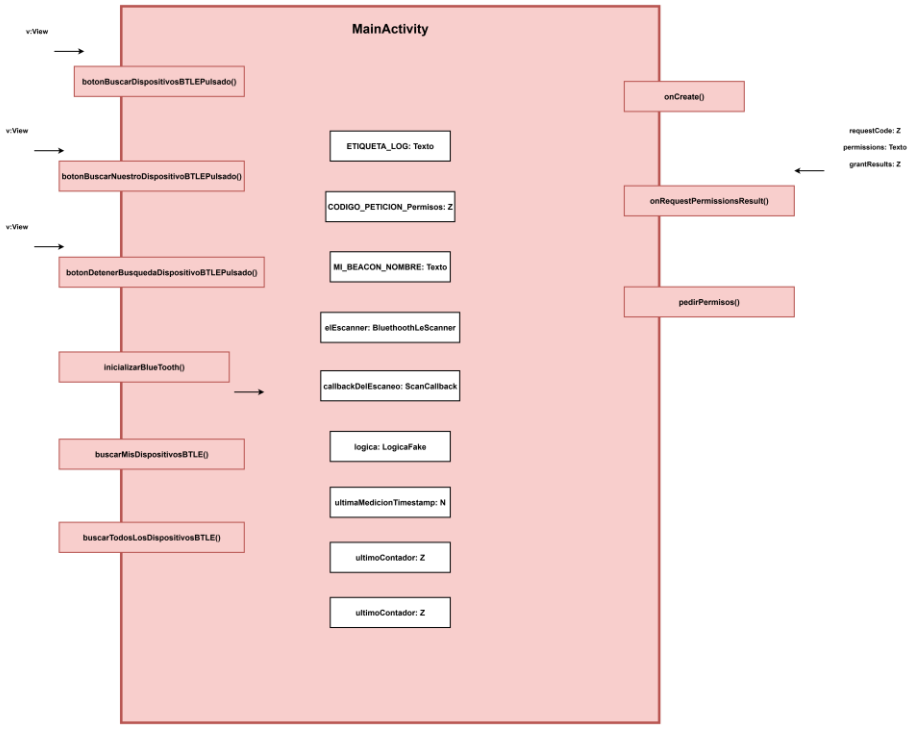
BeaconMedicion.java

Modela una medición BLE con atributos: uuid, mac, sensorId, valor, rssi y timestamp.



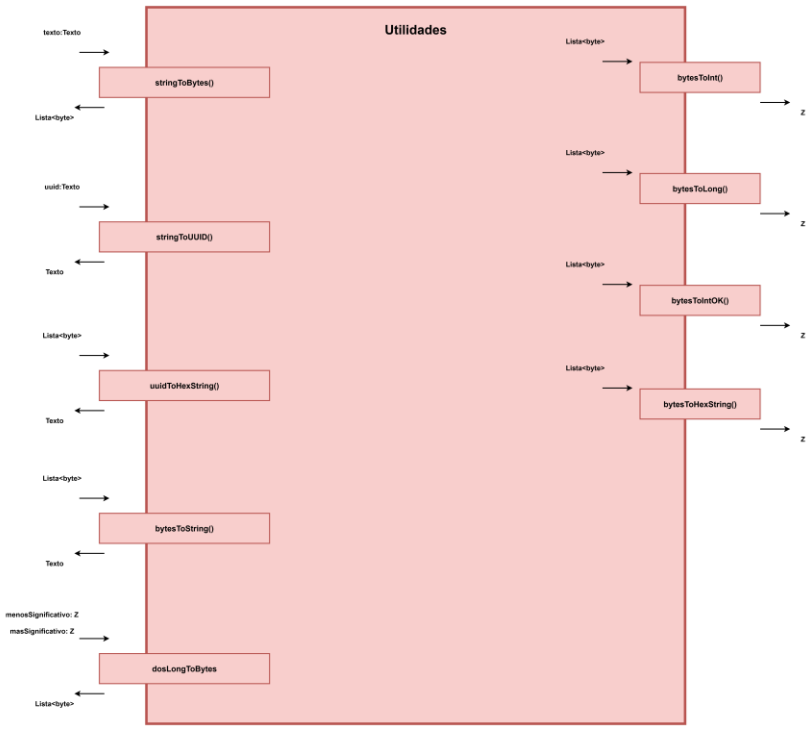
MainActivity.java

Gestiona el escaneo BLE en segundo plano y la comunicación con el servidor REST.



Utilidades.java

Proporciona métodos de conversión (bytes → enteros, UUID → string, etc.).



Código SQL (Base de datos)

La base de datos MySQL almacena las mediciones recibidas desde la aplicación Android. Consta de una tabla principal 'mediciones' y puede ampliarse con otras relacionadas (usuarios, sensores, etc.).

Estructura principal de la tabla:

Campo	Tipo	Descripción
id	INT AUTO_INCREMENT	Identificador único de la medición
uuid	VARCHAR(50)	Identificador del beacon
sensorId	INT	Tipo de sensor (CO2, temperatura, etc.)
valor	FLOAT	Valor medido
rssi	INT	Potencia de la señal

fecha	TIMESTAMP	Momento de recepción

Ejemplo de inserción:

```
INSERT INTO mediciones (uuid, sensorId, valor, rssi, fecha)
VALUES ('EPSG-GTI-PROY-3A', 11, 235, -53, NOW());
```

Conclusión

El Sprint 0 establece la base del sistema completo: sensores BLE simulados en Arduino, aplicación Android para lectura y envío de datos, y servidor MySQL para almacenamiento. Este diseño modular permite extender fácilmente nuevas funcionalidades en los siguientes sprints.