



Información General

Proyecto

SPRINT 0

PROYECTO APLICACIONES DE BIOMETRÍA Y MEDIO AMBIENTE

Nombre del Alumno: Alan Guevara Martínez

Enlace Git: <https://github.com/ALANGMupv/ProyectoBiometria2025.git>

Enlace trello:

<https://trello.com/invite/b/68c9273bf08b76ab09c96593/ATTI19e0d5a059994ed24df6c6455b577bae3E1C4F40/gtiproyectobiometria>

Curso: 2025/2026

Información/Resumen General del Proyecto

El proyecto consiste en desarrollar un sistema de **monitorización ambiental distribuida** para medir la calidad del aire en tiempo real mediante un enfoque de **crowdsensing participativo**.

Componentes principales:

1. **Nodo sensor portátil (Bluetooth LE)**
 - Sensor electroquímico para un gas específico (CO, NO₂, O₃, SO₂).
 - Microcontrolador que procesa datos y los transmite vía **Bluetooth LE**.
 - Batería recargable, carcasa 3D y electrónica de soporte.
2. **Aplicación móvil (Android)**
 - Recibe datos del sensor vía BLE.
 - Envía las mediciones al servidor usando **HTTP (JSON)**.
 - Muestra mapas de contaminación, tips, noticias e información ambiental.
3. **Servidor en la nube / UPV**
 - Recibe, procesa y almacena datos.
 - Aplica filtros y correlaciones con estaciones oficiales.
 - Genera mapas de polución en alta resolución mediante interpolación geoestadística.

Posibles objetivos

- Empoderar a la ciudadanía convirtiéndola en estaciones móviles de medición.
- Crear **mapas colaborativos de polución** más precisos y en tiempo real.
- Sensibilizar a la población sobre la contaminación y su impacto en la salud.

Conexiones del Proyecto

- **Placa sensor (SparkFun nRF52832)** → emite *beacons* por Bluetooth LE
- **Móvil Android** → recibe beacons, envía datos vía HTTP POST (JSON)
- **API REST** → recibe y pasa datos a la lógica de negocio
- **Lógica de negocio** → procesa, guarda en la BBDD y publica resultados
- **Servidor Web HTML** → expone mapas e información
- **Usuario final (UX)** → consulta datos desde su app o navegador

Diseño del Proyecto

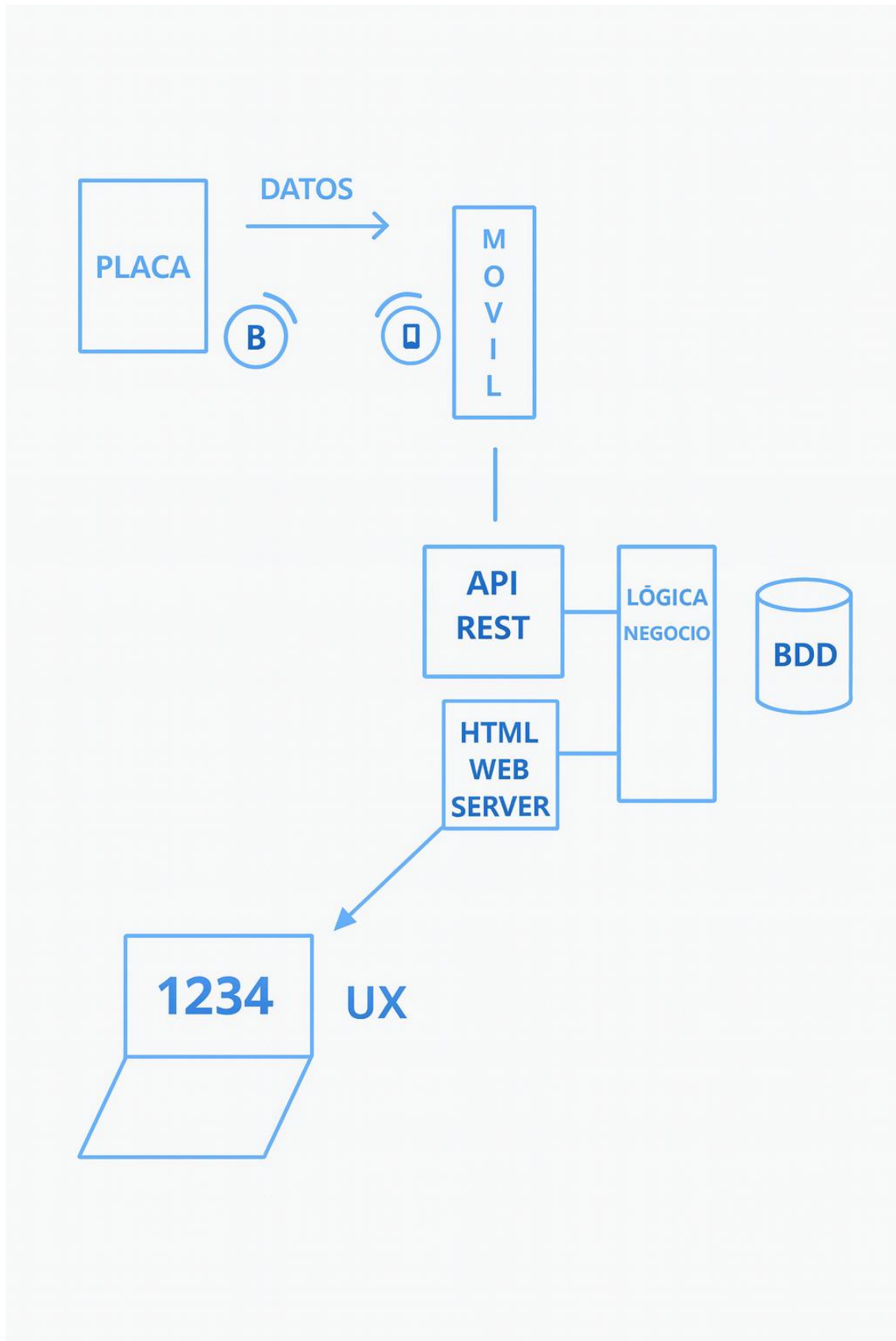


Ilustración 1: Diseño del Proyecto

iBeacon

iBeacon es una tecnología que permite que un dispositivo, siguiendo el estándar “bluetooth low energy BTLE”, emita información “en pulsos periódicos” (como un faro). También permite conexión vinculada entre dos dispositivos.

Es un dispositivo que **emite** (sin conectarse) pequeños anuncios por Bluetooth Low Energy a intervalos regulares. El móvil los “oye”, lee el **identificador** que viene en el anuncio y, con el **RSSI** (potencia recibida), estima proximidad. No hay emparejamiento ni sesión: son **paquetes de difusión**.

Apple definió un formato concreto llamado **iBeacon** que viaja en un campo “Manufacturer Specific Data”.

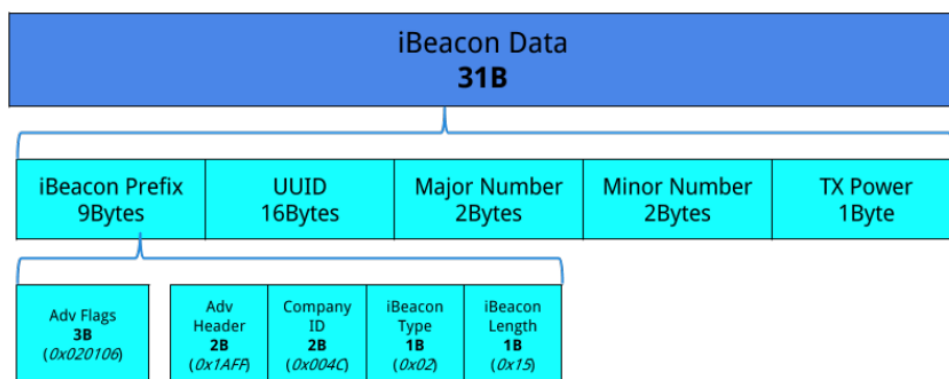


Ilustración 2: iBeacon

Estructura del paquete iBeacon (31 Bytes)

iBeacon Prefix (9 Bytes)

Contiene información fija necesaria para que un receptor entienda que el mensaje es de tipo iBeacon.

- **Adv Flags (3B):** indican características de la transmisión (ej. uso de BLE).
- **Adv Header (2B):** especifica el tipo de anuncio.
- **Company ID (2B):** identifica al fabricante (en este caso 0x004C = Apple).
- **iBeacon Type (1B):** valor fijo (0x02 → iBeacon).
- **iBeacon Length (1B):** indica el tamaño de los datos que siguen (0x15 = 21 bytes).

UUID (16 Bytes)

Identificador único global. Permite distinguir una aplicación o proyecto concreto.

Aquí es donde meteremos nuestro valor de CO2 (por ejemplo).

Ejemplo: EPSG-GTI-PROY-3A.

Major Number (2 Bytes)

Subdivisión del UUID. Sirve para identificar un grupo dentro de un mismo proyecto o instalación.

Ejemplo: edificio, planta o área.

Minor Number (2 Bytes)

Identificador más específico dentro del **major**.

Ejemplo: un beacon concreto en una sala o pasillo.

TX Power (1 Byte)

Potencia de transmisión medida a 1 metro.

Se usa para estimar la distancia al beacon comparando este valor con la intensidad real recibida (**RSSI**). Cuanto más negativo, más lejos está el dispositivo.