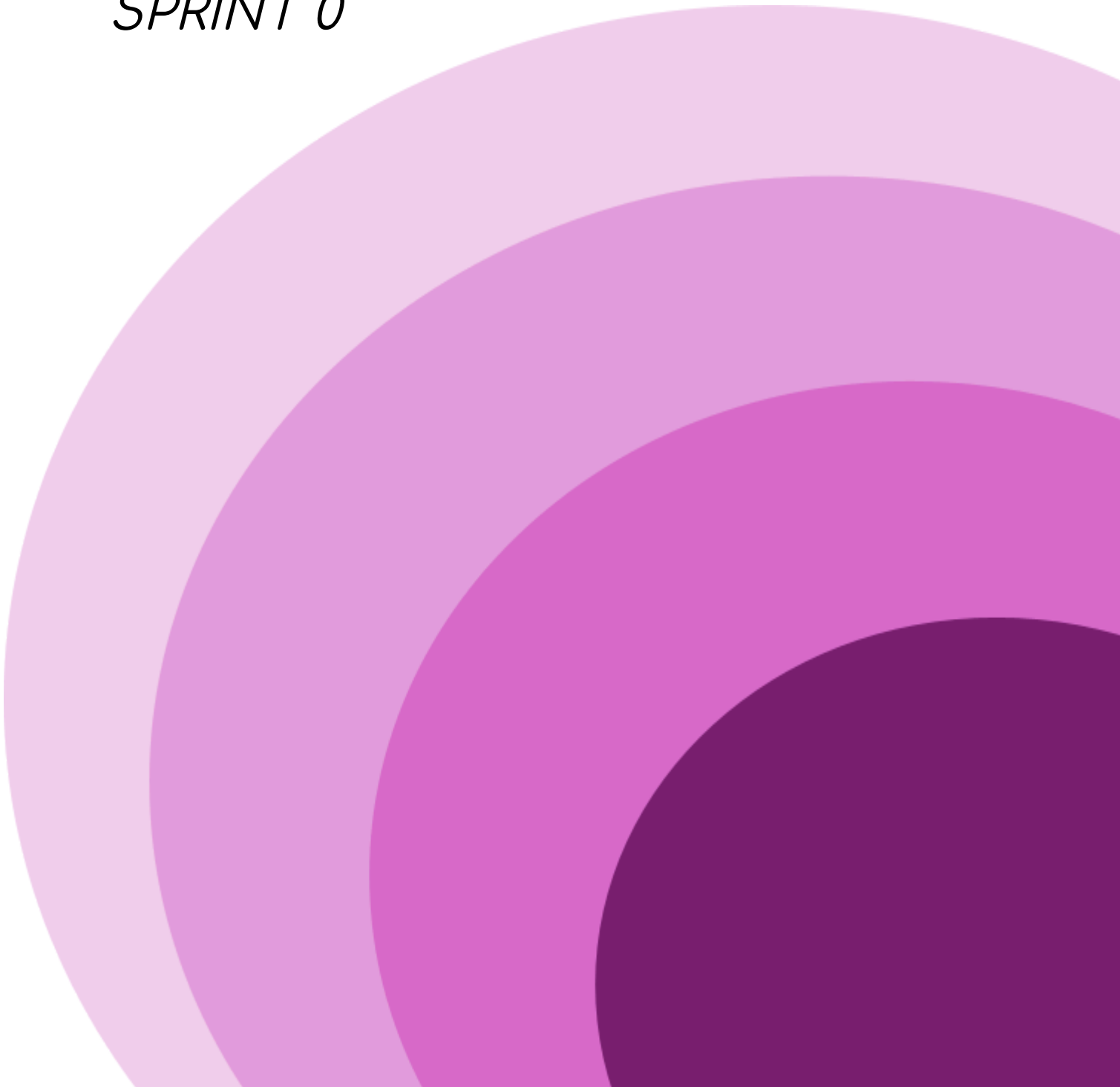


DOCUMENTO DE DISEÑO

SPRINT 0

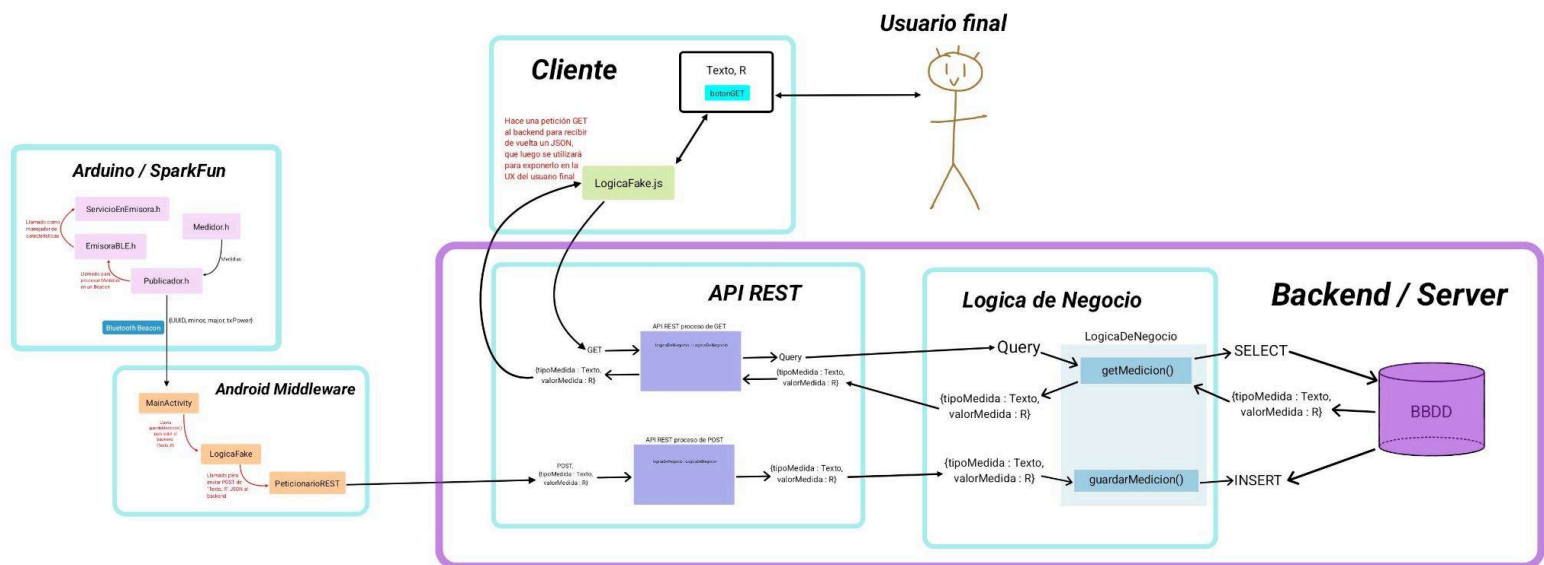


INDEXACIÓN.....	1
INTRODUCCIÓN.....	3
ARQUITECTURA DEL SISTEMA.....	3
• Arduino / Sparkfun.....	4
• Android Middleware.....	8
• Backend / Server.....	11
◦ API REST.....	12
◦ Lógica de Negocio.....	13
• Cliente y Usuario final.....	13

INTRODUCCIÓN

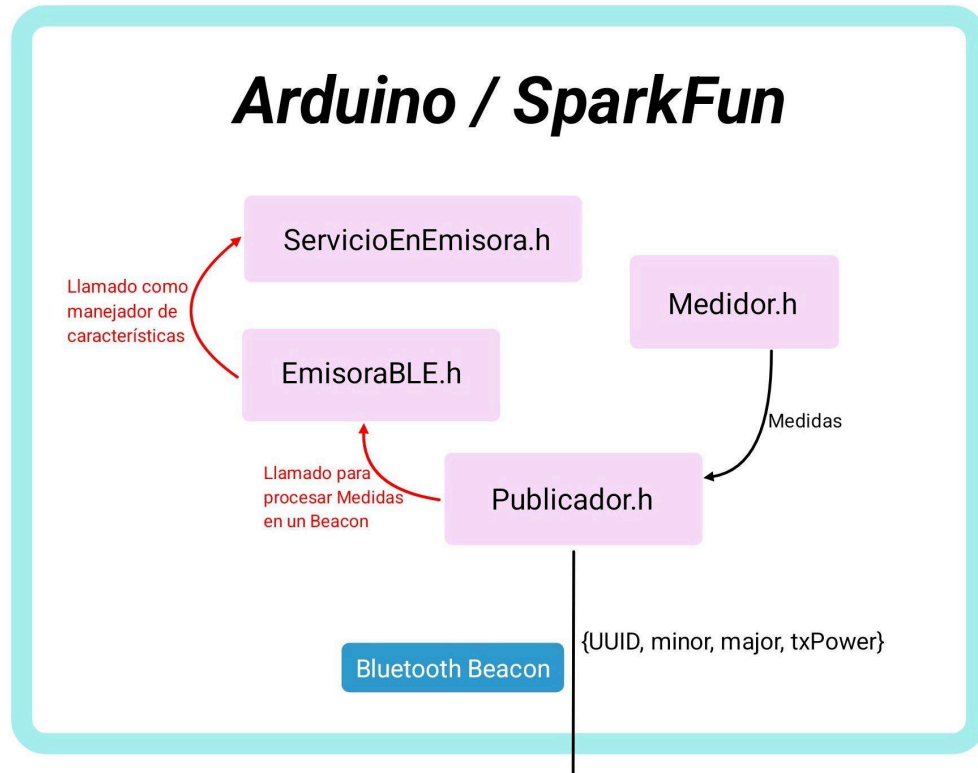
Este documento trata de presentar, de manera ordenada y simple, los diseños de los programas utilizados en el Sprint 0 del proyecto, Proyecto Aplicaciones de Biometría y Medio Ambiente.

ARQUITECTURA DEL SISTEMA



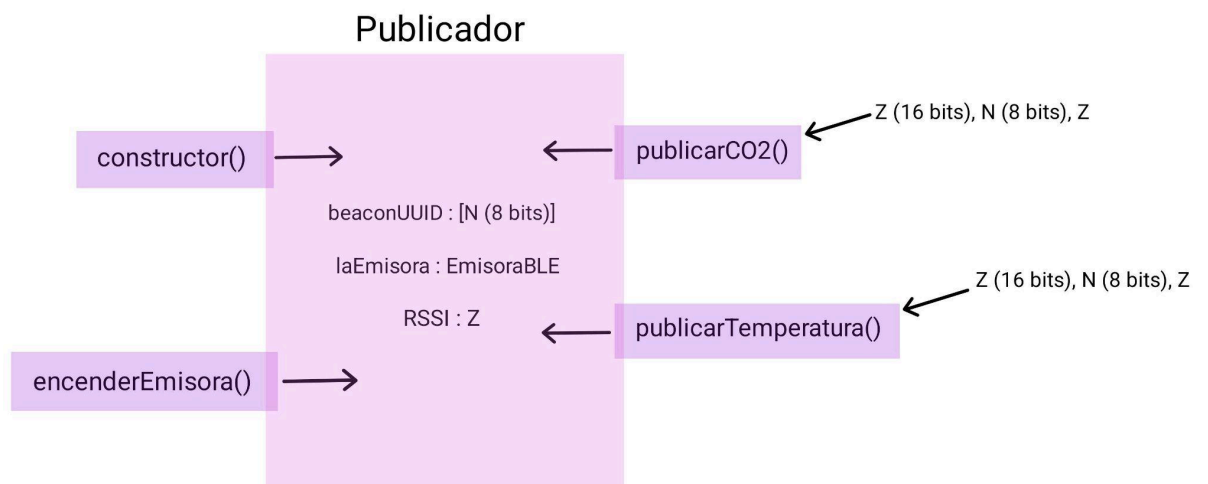
Este sistema se comporta de las siguientes partes :

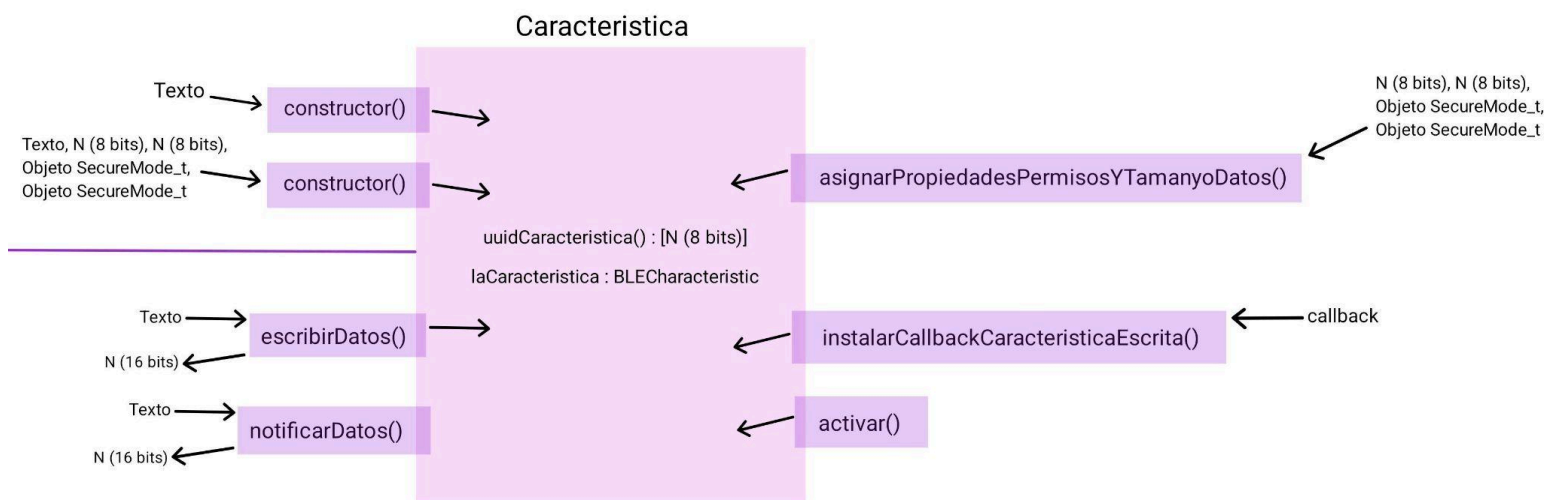
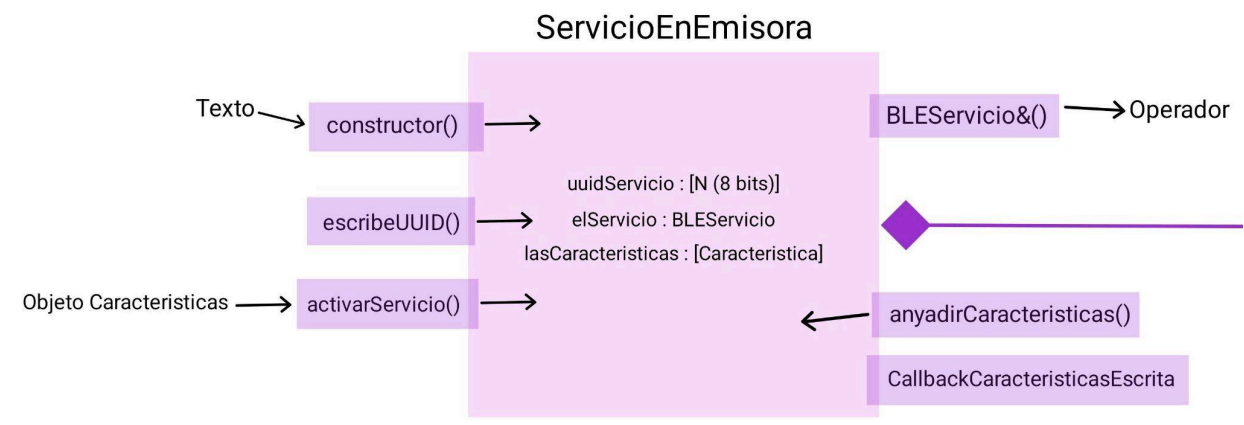
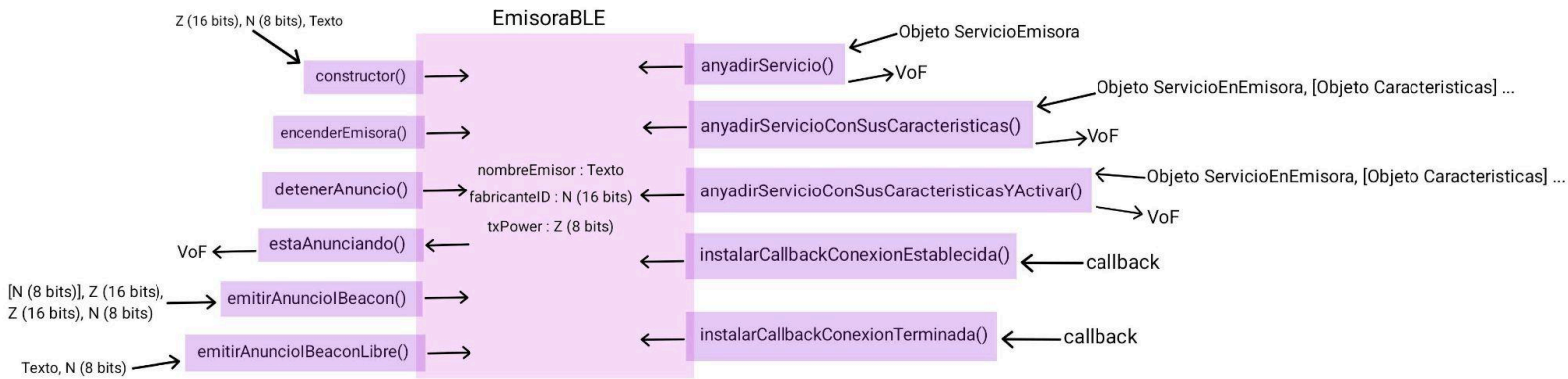
- Arduino / Sparkfun

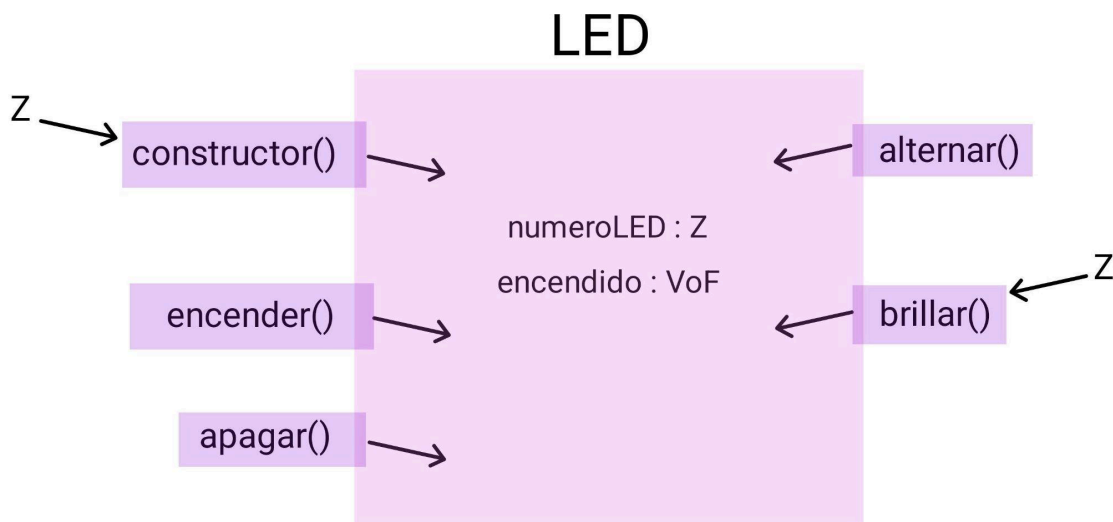
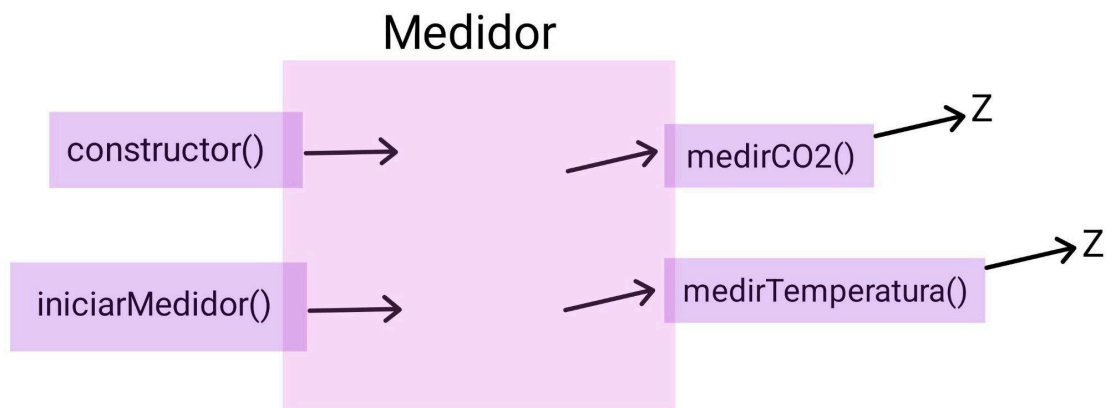
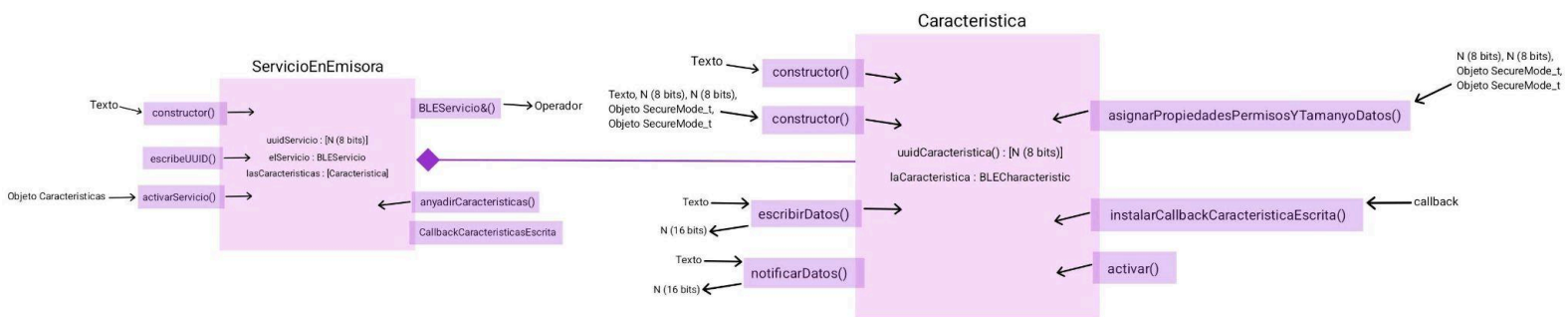


En esta parte del proyecto, aplicamos C++ para poder programar el SparkFun Mini, que estará encargado de enviarnos medidas a través de un IBeacon (o mejor dicho, Beacon de Bluetooth), enviando medidas por medio de Bluetooth.

Este se comporta de las siguientes clases, los representaremos simplemente por sus diseños :







PuertoSerie

constructor()

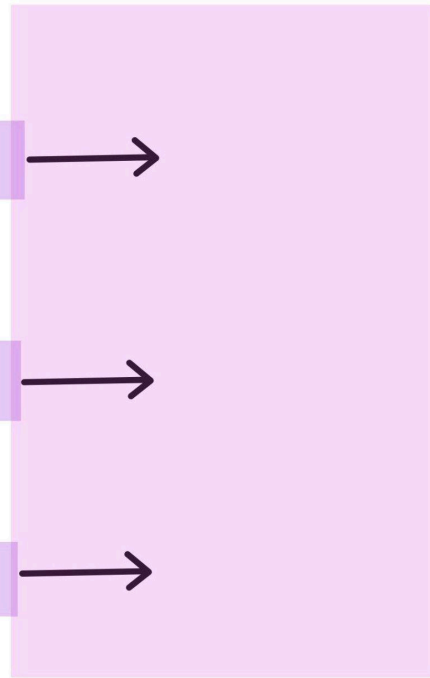


esperaDisponible()

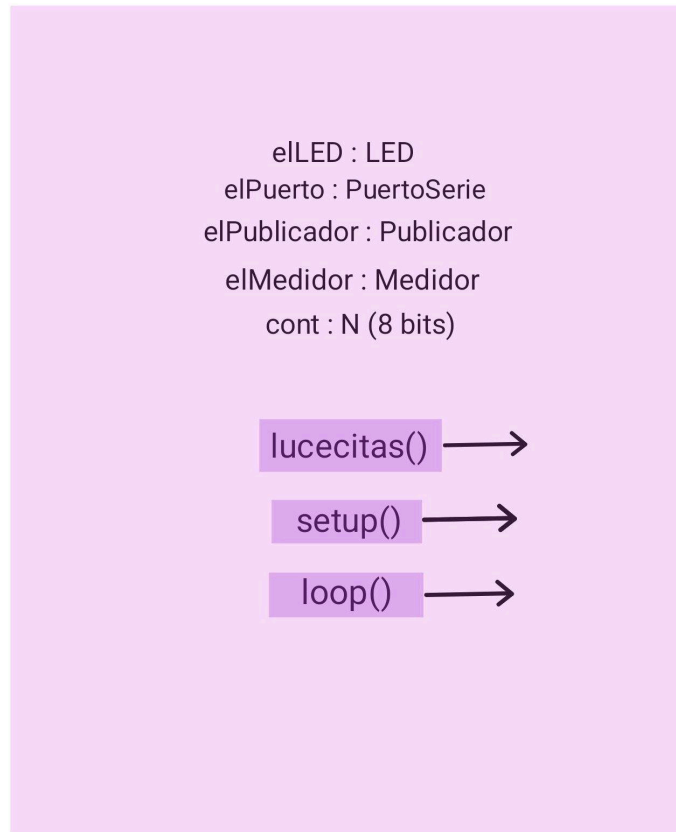


Cualquier input →

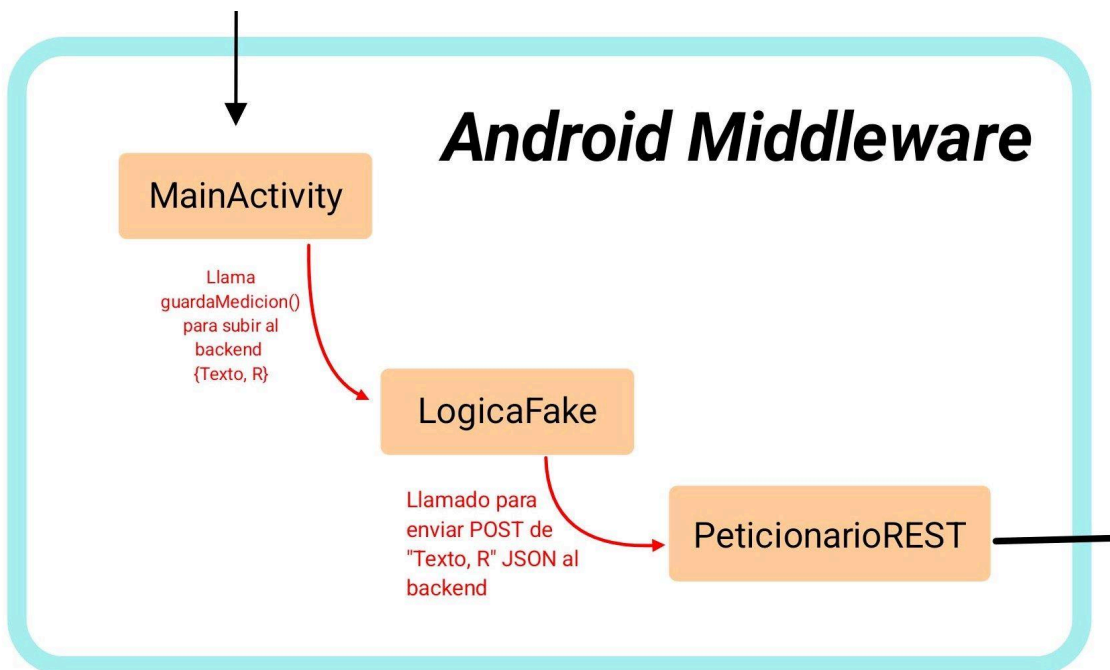
escribir()



HolaMundoIBeacon

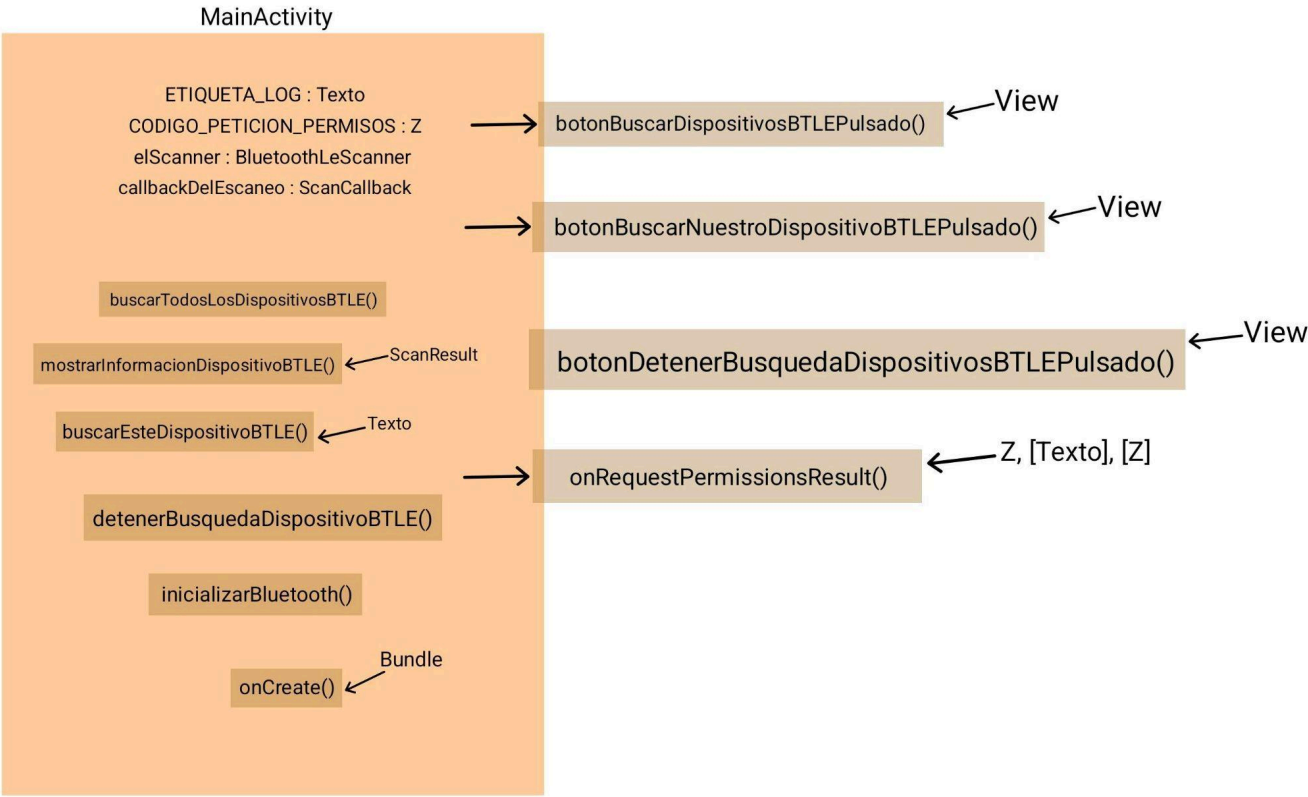


- Android Middleware

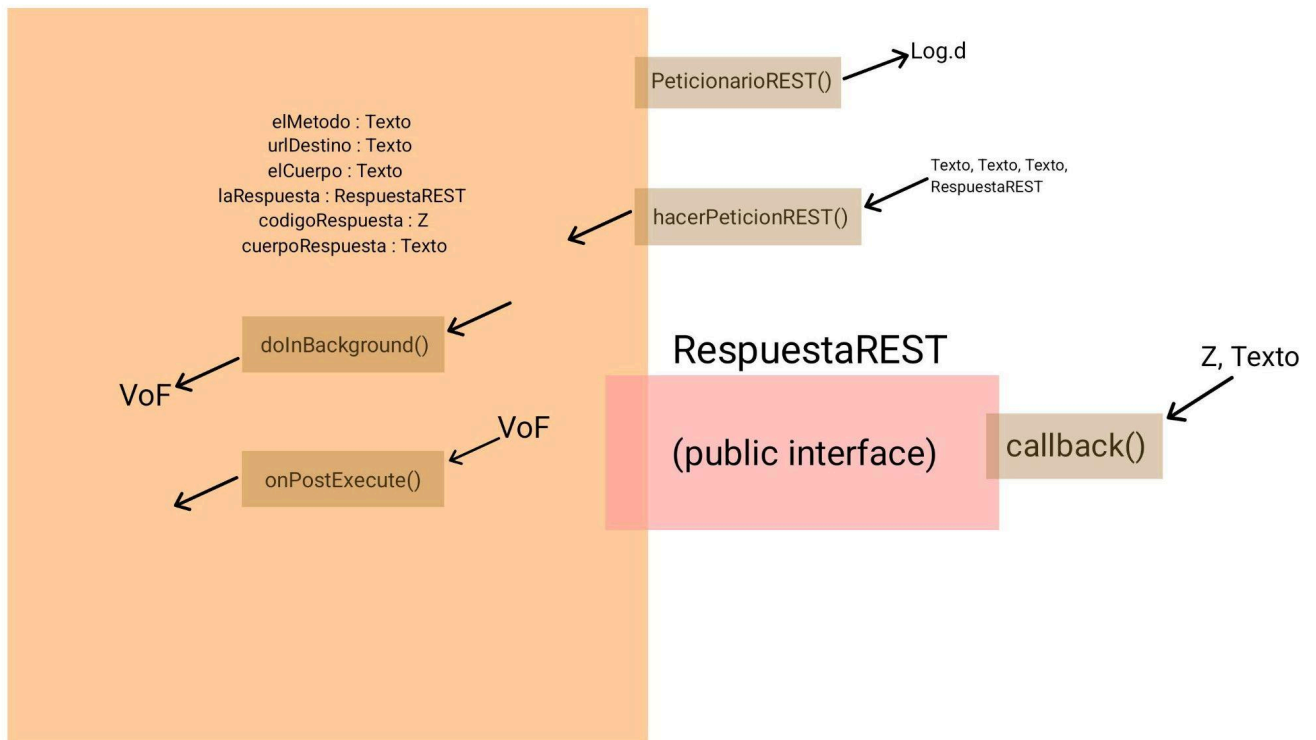


En esta parte del proyecto programamos una aplicación con Java, este mismo se encarga de detectar el Beacon Bluetooth enviado para poder procesar la información que esté enviado, en este caso las medidas. Después, la aplicación se encargará de subir esta información procesada hacia nuestro servidor con métodos de HTTP REST.

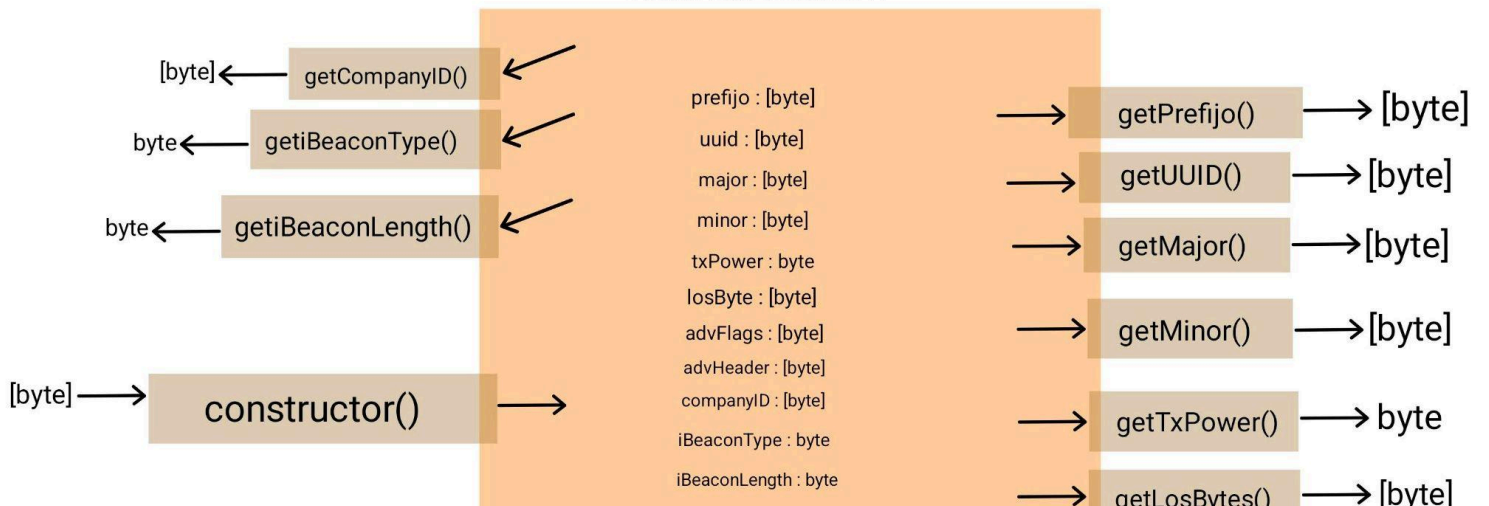
Este se comporta de las siguientes clases, los representaremos simplemente por sus diseños :



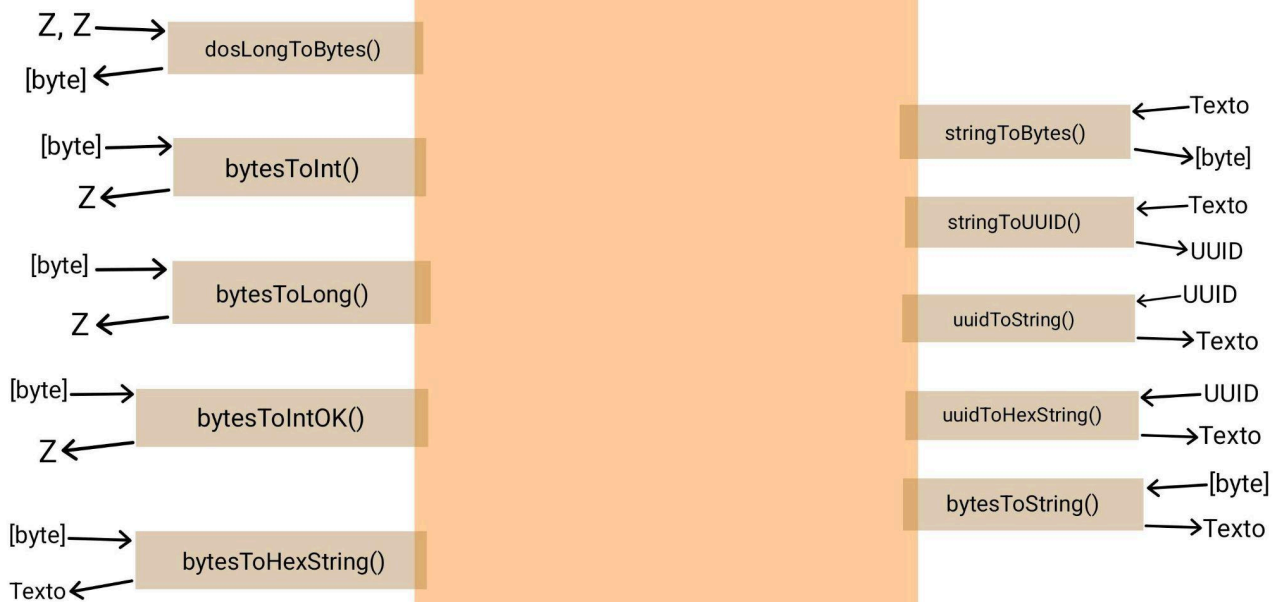
PeticionarioREST



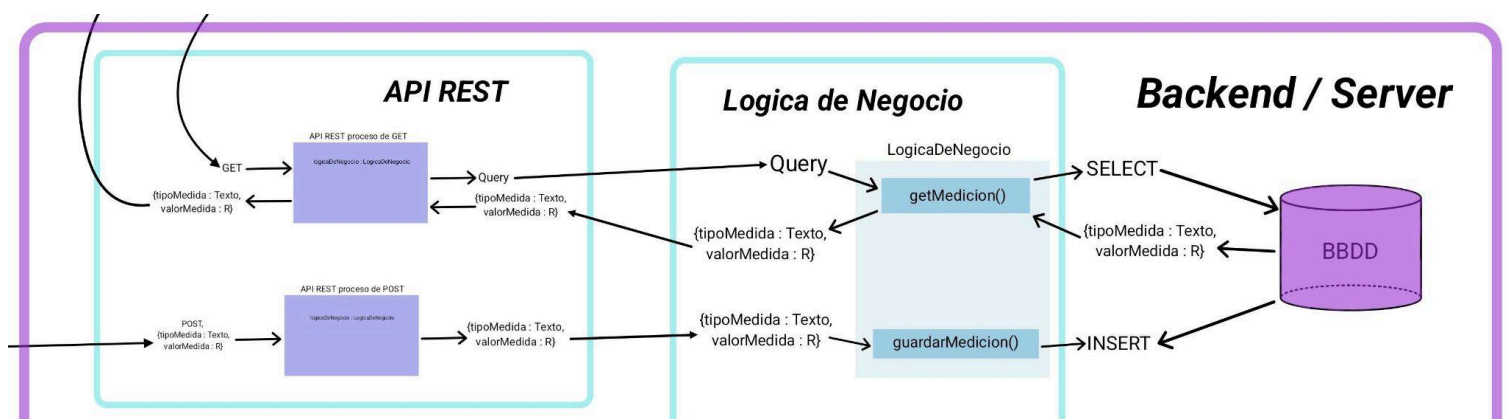
TramalBeacon



Utilidades



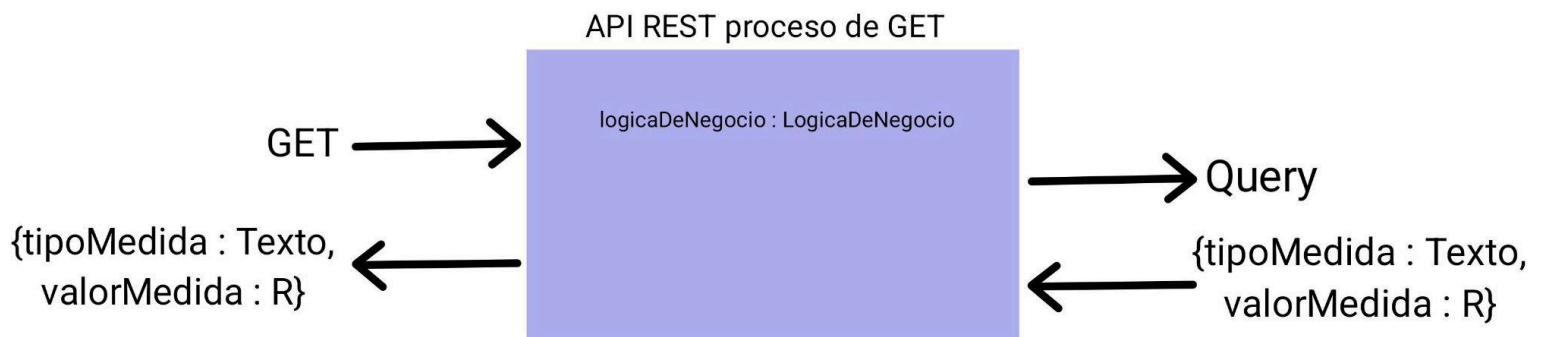
● Backend / Server



El backend, el servidor, se encarga de guardar y enviar los datos mediante peticiones de protocolos HTTP REST, en nuestro caso las medidas. Esta parte tiene más flexibilidad a la hora de programar, por tanto utilizaremos JavaScript y MySQL.

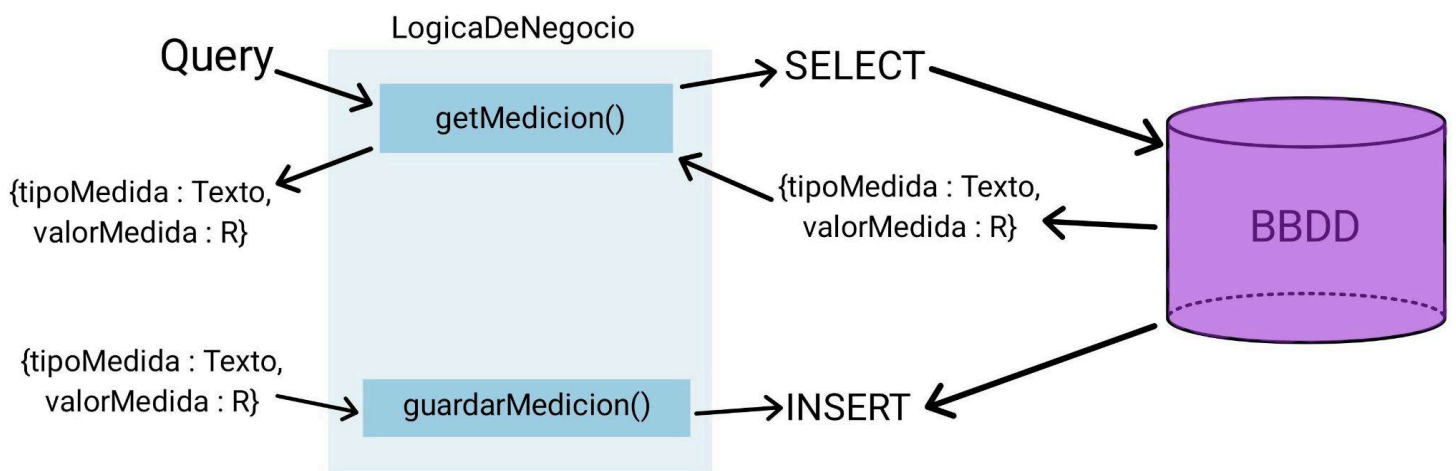
Este se comporta de las siguientes programaciones, además de la base de datos :

- **API REST**



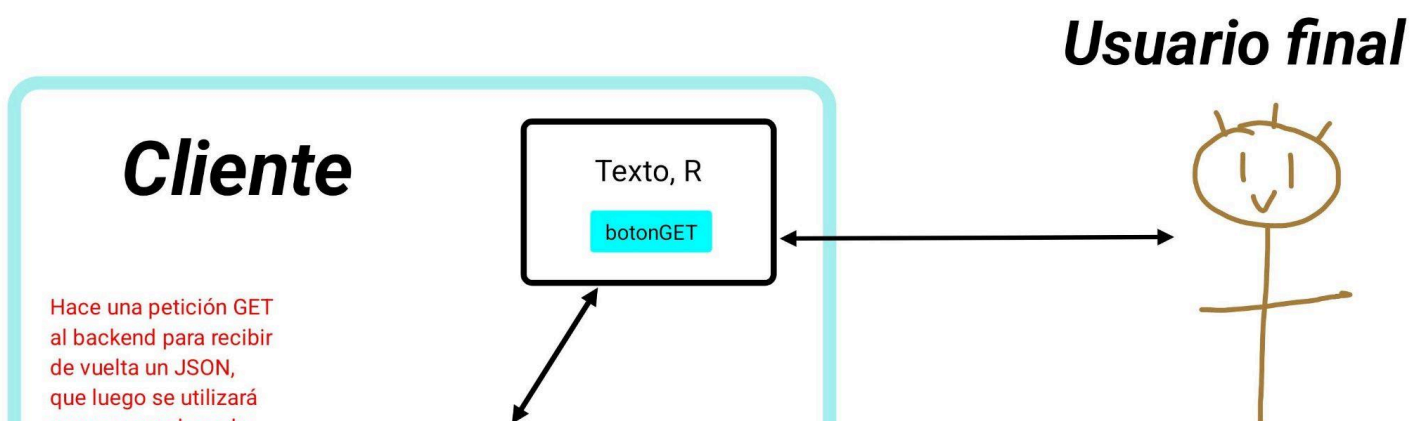
Esta parte del servidor nos permite traficar las medidas con la ayuda de los protocolos HTTP REST. Funciona como conexión entre las aplicaciones fuera del servidor, haciendo las peticiones, y la Lógica de Negocio, que se explicará en seguida.

- Lógica de Negocio



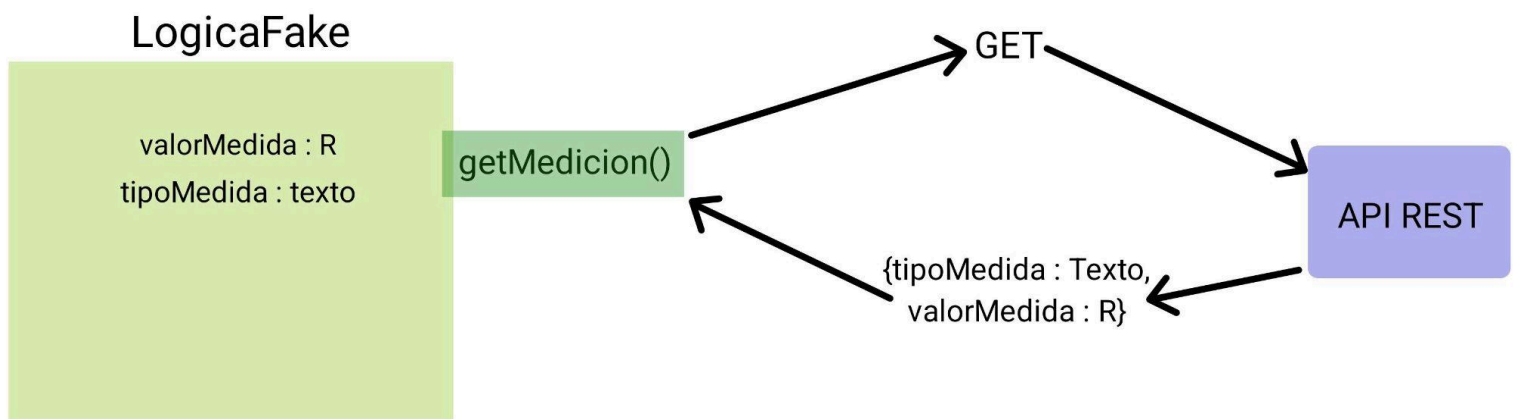
Aquí observamos la Lógica de Negocio, esta nos permitirá añadir y seleccionar las medidas, que nos solicita la API REST, desde la base de datos. Nos sirve como norma para filtrar el tráfico de información enviado por la API REST.

- Cliente y Usuario final



En esta parte final, e igualmente importante, del proyecto, llegamos al punto de presentación de las medidas mediante un HTML simple sin CSS. Con la ayuda de un simple botón que sirve para enviar una petición REST a través de la única clase de JavaScript, nos servirá como lógica fake.

El funcionamiento de esa clase de JavaScript está representado de la siguiente manera :



Y por fin nuestro usuario final, eres tu o cualquier tipo de ser que pueda entender de manera intuitiva la representación de las medidas que estén expuestas en el sitio web compuesto de HTML.

