

Documentacion

1

Generado por Doxygen 1.12.0

1 Índice de espacios de nombres	1
1.1 Lista de paquetes	1
2 Índice jerárquico	1
2.1 Jerarquía de clases	1
3 Índice de clases	2
3.1 Lista de clases	2
4 Índice de archivos	2
4.1 Lista de archivos	2
5 Documentación de espacios de nombres	3
5.1 Paquete com.example.smariba_upv.btle_sento	3
5.2 Paquete com.example.smariba_upv.btle_sento.API	3
5.3 Paquete com.example.smariba_upv.btle_sento.LOGIC	3
5.4 Paquete com.example.smariba_upv.btle_sento.POJO	3
6 Documentación de clases	4
6.1 Referencia de la interface com.example.smariba_upv.btle_sento.API.ApiService	4
6.1.1 Descripción detallada	4
6.1.2 Documentación de funciones miembro	4
6.2 Referencia de la clase com.example.smariba_upv.btle_sento.MainActivity	5
6.2.1 Descripción detallada	7
6.2.2 Documentación de funciones miembro	7
6.3 Referencia de la clase com.example.smariba_upv.btle_sento.POJO.Medicion	8
6.3.1 Descripción detallada	9
6.3.2 Documentación de constructores y destructores	9
6.3.3 Documentación de funciones miembro	9
6.4 Referencia de la clase com.example.smariba_upv.btle_sento.API.RetrofitClient	11
6.4.1 Descripción detallada	11
6.4.2 Documentación de funciones miembro	12
6.5 Referencia de la clase com.example.smariba_upv.btle_sento.POJO.TramalBeacon	12
6.5.1 Descripción detallada	13
6.5.2 Documentación de constructores y destructores	13
6.5.3 Documentación de funciones miembro	13
6.6 Referencia de la clase com.example.smariba_upv.btle_sento.LOGIC.Utilidades	16
6.6.1 Descripción detallada	17
6.6.2 Documentación de funciones miembro	17
7 Documentación de archivos	22
7.1 Referencia del archivo ApiService.java	22
7.1.1 Descripción detallada	22
7.2 ApiService.java	23

7.3 Referencia del archivo RetrofitClient.java	23
7.3.1 Descripción detallada	24
7.4 RetrofitClient.java	24
7.5 Referencia del archivo Utilidades.java	24
7.5.1 Descripción detallada	25
7.6 Utilidades.java	25
7.7 Referencia del archivo MainActivity.java	26
7.7.1 Descripción detallada	27
7.8 MainActivity.java	28
7.9 Referencia del archivo Medicion.java	33
7.9.1 Descripción detallada	33
7.10 Medicion.java	33
7.11 Referencia del archivo TramalBeacon.java	34
7.11.1 Descripción detallada	34
7.12 TramalBeacon.java	35
Índice alfabético	37

1. Índice de espacios de nombres

1.1. Lista de paquetes

Estos son los paquetes con breves descripciones (si están disponibles):

com.example.smariba_upv.btle_sento	3
com.example.smariba_upv.btle_sento.API	3
com.example.smariba_upv.btle_sento.LOGIC	3
com.example.smariba_upv.btle_sento.POJO	3

2. Índice jerárquico

2.1. Jerarquía de clases

Este listado de herencia está ordenado de forma general pero no está en orden alfabético estricto:

com.example.smariba_upv.btle_sento.API.ApiService	4
com.example.smariba_upv.btle_sento.POJO.Medicion	8
com.example.smariba_upv.btle_sento.API.RetrofitClient	11
com.example.smariba_upv.btle_sento.POJO.TramalBeacon	12

<code>com.example.smariba_upv.btle_sento.LOGIC.Utilidades</code> AppCompatActivity	16
<code>com.example.smariba_upv.btle_sento.MainActivity</code>	5

3. Índice de clases

3.1. Lista de clases

Lista de clases, estructuras, uniones e interfaces con breves descripciones:

<code>com.example.smariba_upv.btle_sento.API.ApiService</code> Interfaz que define las operaciones que se realizarán contra el servidor usando Retrofit	4
<code>com.example.smariba_upv.btle_sento.MainActivity</code> La clase principal que gestiona la interfaz y las funciones BTLE	5
<code>com.example.smariba_upv.btle_sento.POJO.Medicion</code> Clase que encapsula los datos de una medición de gas, incluyendo el lugar, el tipo de gas y su valor	8
<code>com.example.smariba_upv.btle_sento.API.RetrofitClient</code> Clase que proporciona la instancia de Retrofit configurada para realizar llamadas a la API REST	11
<code>com.example.smariba_upv.btle_sento.POJO.TramalBeacon</code> Clase que modela los datos de una trama iBeacon y permite acceder a sus diferentes componentes, tales como UUID, Major, Minor y más	12
<code>com.example.smariba_upv.btle_sento.LOGIC.Utilidades</code> Clase con métodos estáticos para la manipulación de datos en forma de bytes, cadenas, y UUIDs	16

4. Índice de archivos

4.1. Lista de archivos

Lista de todos los archivos con breves descripciones:

<code>ApiService.java</code> Interfaz para la API REST utilizada en la aplicación, donde se definen las llamadas GET y POST	22
<code>RetrofitClient.java</code> Clase encargada de gestionar la instancia de Retrofit para interactuar con el servidor	23
<code>Utilidades.java</code> Clase con métodos utilitarios para manipulación de bytes, UUIDs y conversión de tipos	24
<code>MainActivity.java</code> Clase principal de la aplicación desde la que se inicia el escaneo de dispositivos BTLE	26
<code>Medicion.java</code> Clase que representa una medición de gas en un lugar determinado	33

TramalBeacon.java

Clase que representa una trama iBeacon, extrayendo sus diferentes campos a partir de un array de bytes

34

5. Documentación de espacios de nombres

5.1. Paquete com.example.smariba_upv.btle_sento

Paquetes

- package [API](#)
- package [LOGIC](#)
- package [POJO](#)

Clases

- class [MainActivity](#)
La clase principal que gestiona la interfaz y las funciones BTLE.

5.2. Paquete com.example.smariba_upv.btle_sento.API

Clases

- interface [ApiService](#)
Interfaz que define las operaciones que se realizarán contra el servidor usando Retrofit.
- class [RetrofitClient](#)
Clase que proporciona la instancia de Retrofit configurada para realizar llamadas a la [API](#) REST.

5.3. Paquete com.example.smariba_upv.btle_sento.LOGIC

Clases

- class [Utilidades](#)
Clase con métodos estáticos para la manipulación de datos en forma de bytes, cadenas, y UUIDs.

5.4. Paquete com.example.smariba_upv.btle_sento.POJO

Clases

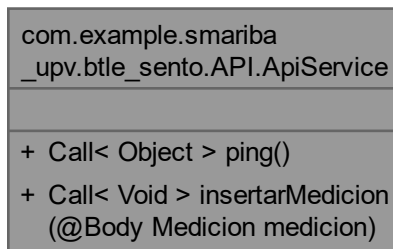
- class [Medicion](#)
Clase que encapsula los datos de una medición de gas, incluyendo el lugar, el tipo de gas y su valor.
- class [TramalBeacon](#)
Clase que modela los datos de una trama iBeacon y permite acceder a sus diferentes componentes, tales como UUID, Major, Minor y más.

6. Documentación de clases

6.1. Referencia de la interface com.example.smariba_upv.btle_sento.API.ApiService

Interfaz que define las operaciones que se realizarán contra el servidor usando Retrofit.

Diagrama de colaboración de com.example.smariba_upv.btle_sento.API.ApiService:



Métodos públicos

- Call< Object > [ping](#) ()
Método que realiza una petición GET al servidor para verificar su disponibilidad.
- Call< Void > [insertarMedicion](#) (@Body [Medicion](#) medicion)
Método que realiza una petición POST para insertar una medición en el servidor.

6.1.1. Descripción detallada

Interfaz que define las operaciones que se realizarán contra el servidor usando Retrofit.

Definición en la línea 20 del archivo [ApiService.java](#).

6.1.2. Documentación de funciones miembro

insertarMedicion()

```
Call< Void > com.example.smariba_upv.btle_sento.API.ApiService.insertarMedicion (
    @Body Medicion medicion)
```

Método que realiza una petición POST para insertar una medición en el servidor.

Parámetros

in	<i>medicion</i>	Objeto de tipo Medicion que contiene los datos de la medición a insertar.
----	-----------------	---

Devuelve

Retorna un objeto de tipo Call<Void> que representa el resultado de la operación.

ping()

```
Call< Object > com.example.smariba_upv.btle_sento.API.ApiService.ping ()
```

Método que realiza una petición GET al servidor para verificar su disponibilidad.

Devuelve

Retorna un objeto de tipo Call que contiene la respuesta del servidor.

La documentación de esta interface está generada del siguiente archivo:

- [ApiService.java](#)

6.2. Referencia de la clase com.example.smariba_upv.btle_sento.MainActivity

La clase principal que gestiona la interfaz y las funciones BTLE.

Diagrama de herencia de com.example.smariba_upv.btle_sento.MainActivity

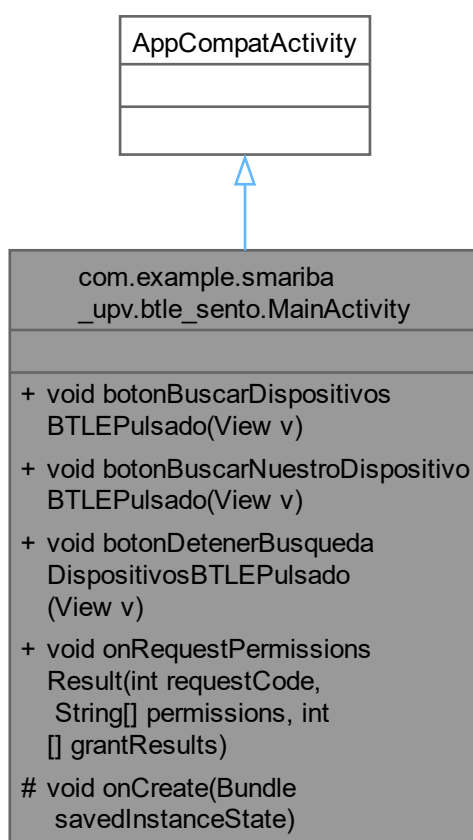
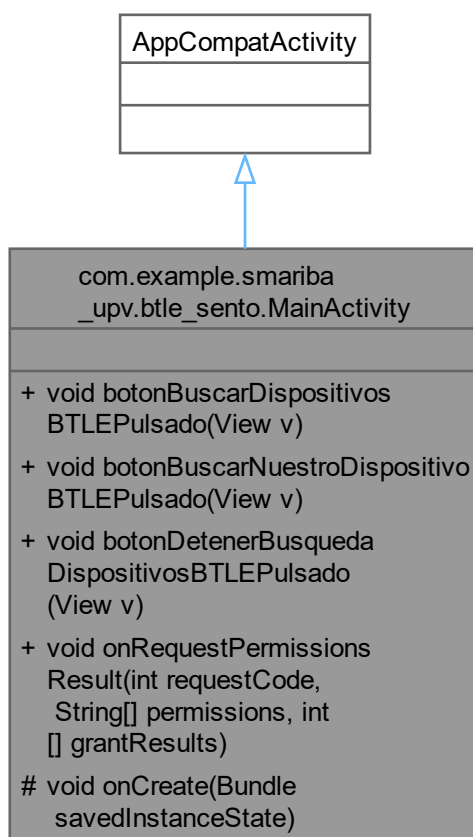


Diagrama de colaboración de com.example.smariba_upv.btle_sento.MainActivity:



Métodos públicos

- void [botonBuscarDispositivosBTLEPulsado](#) (View v)
Método que se ejecuta al pulsar el botón para buscar dispositivos BTLE.
- void [botonBuscarNuestroDispositivoBTLEPulsado](#) (View v)
Método que se ejecuta al pulsar el botón para buscar un dispositivo BTLE específico.
- void [botonDetenerBusquedaDispositivosBTLEPulsado](#) (View v)
Método que se ejecuta al pulsar el botón para detener la búsqueda de dispositivos BTLE.
- void [onRequestPermissionsResult](#) (int requestCode, String[] permissions, int[] grantResults)
Método que se ejecuta al solicitar permisos.

Métodos protegidos

- void [onCreate](#) (Bundle savedInstanceState)
Método que se ejecuta al crear la actividad.

6.2.1. Descripción detallada

La clase principal que gestiona la interfaz y las funciones BTLE.

Definición en la línea 43 del archivo [MainActivity.java](#).

6.2.2. Documentación de funciones miembro

botonBuscarDispositivosBTLEPulsado()

```
void com.example.smariba_upv.btle_sento.MainActivity.botonBuscarDispositivosBTLEPulsado (
    View v)
```

Método que se ejecuta al pulsar el botón para buscar dispositivos BTLE.

Parámetros

	v	the v
in	v	View asociada al botón.

Definición en la línea 271 del archivo [MainActivity.java](#).

botonBuscarNuestroDispositivoBTLEPulsado()

```
void com.example.smariba_upv.btle_sento.MainActivity.botonBuscarNuestroDispositivoBTLEPulsado
(
    View v)
```

Método que se ejecuta al pulsar el botón para buscar un dispositivo BTLE específico.

Parámetros

	v	the v
in	v	View asociada al botón.

Definición en la línea 282 del archivo [MainActivity.java](#).

botonDetenerBusquedaDispositivosBTLEPulsado()

```
void com.example.smariba_upv.btle_sento.MainActivity.botonDetenerBusquedaDispositivosBTLEPulsado
(
    View v)
```

Método que se ejecuta al pulsar el botón para detener la búsqueda de dispositivos BTLE.

Parámetros

	v	the v
in	v	View asociada al botón.

Definición en la línea 297 del archivo [MainActivity.java](#).

onCreate()

```
void com.example.smariba_upv.btle_sento.MainActivity.onCreate (
    Bundle savedInstanceState) [protected]
```

Método que se ejecuta al crear la actividad.

Parámetros

in	<i>savedInstanceState</i>	Estado de la instancia guardada.
----	---------------------------	----------------------------------

Definición en la línea 395 del archivo [MainActivity.java](#).

onRequestPermissionsResult()

```
void com.example.smariba_upv.btle_sento.MainActivity.onRequestPermissionsResult (
    int requestCode,
    String[] permissions,
    int[] grantResults)
```

Método que se ejecuta al solicitar permisos.

Parámetros

in	<i>requestCode</i>	Código de solicitud.
in	<i>permissions</i>	Permisos solicitados.
in	<i>grantResults</i>	Resultados de los permisos.

Definición en la línea 416 del archivo [MainActivity.java](#).

La documentación de esta clase está generada del siguiente archivo:

- [MainActivity.java](#)

6.3. Referencia de la clase com.example.smariba_upv.btle_sento.POJO.Medicion

Clase que encapsula los datos de una medición de gas, incluyendo el lugar, el tipo de gas y su valor.

Diagrama de colaboración de com.example.smariba_upv.btle_sento.POJO.Medicion:

com.example.smariba_upv.btle_sento.POJO.Medicion
+ Medicion(String Lugar, String Gas, int Valor) + String getLugar() + void setLugar(String lugar) + String getGas() + void setGas(String gas) + int getValor() + void setValor(int valor)

Métodos públicos

- **Medicion** (String Lugar, String Gas, int Valor)
*Constructor de la clase **Medicion**.*
- String **getLugar** ()
Obtiene el lugar de la medición.
- void **setLugar** (String lugar)
Establece el lugar de la medición.
- String **getGas** ()
Obtiene el tipo de gas medido.
- void **setGas** (String gas)
Establece el tipo de gas medido.
- int **getValor** ()
Obtiene el valor de la medición del gas.
- void **setValor** (int valor)
Establece el valor de la medición del gas.

6.3.1. Descripción detallada

Clase que encapsula los datos de una medición de gas, incluyendo el lugar, el tipo de gas y su valor.

Definición en la línea 14 del archivo [Medicion.java](#).

6.3.2. Documentación de constructores y destructores

Medicion()

```
com.example.smariba_upv.btle_sento.POJO.Medicion.Medicion (
    String Lugar,
    String Gas,
    int Valor)
```

Constructor de la clase [Medicion](#).

Parámetros

<i>Lugar</i>	Nombre del lugar donde se realiza la medición.
<i>Gas</i>	Tipo de gas medido.
<i>Valor</i>	Valor medido del gas.

Definición en la línea 25 del archivo [Medicion.java](#).

6.3.3. Documentación de funciones miembro

getGas()

```
String com.example.smariba_upv.btle_sento.POJO.Medicion.getGas ()
```

Obtiene el tipo de gas medido.

Devuelve

Un String que representa el tipo de gas.

Definición en la línea 51 del archivo [Medicion.java](#).

getLugar()

```
String com.example.smariba_upv.btle_sento.POJO.Medicion.getLugar ()
```

Obtiene el lugar de la medición.

Devuelve

Un String que representa el lugar.

Definición en la línea 35 del archivo [Medicion.java](#).

getValor()

```
int com.example.smariba_upv.btle_sento.POJO.Medicion.getValor ()
```

Obtiene el valor de la medición del gas.

Devuelve

Un entero que representa el valor medido.

Definición en la línea 67 del archivo [Medicion.java](#).

setGas()

```
void com.example.smariba_upv.btle_sento.POJO.Medicion.setGas (  
    String gas)
```

Establece el tipo de gas medido.

Parámetros

<i>gas</i>	String que representa el tipo de gas.
------------	---------------------------------------

Definición en la línea 59 del archivo [Medicion.java](#).

setLugar()

```
void com.example.smariba_upv.btle_sento.POJO.Medicion.setLugar (  
    String lugar)
```

Establece el lugar de la medición.

Parámetros

<i>lugar</i>	String que representa el lugar.
--------------	---------------------------------

Definición en la línea 43 del archivo [Medicion.java](#).

setValor()

```
void com.example.smariba_upv.btle_sento.POJO.Medicion.setValor (  
    int valor)
```

Establece el valor de la medición del gas.

Parámetros

valor	Entero que representa el valor medido.
-------	--

Definición en la línea 75 del archivo [Medicion.java](#).

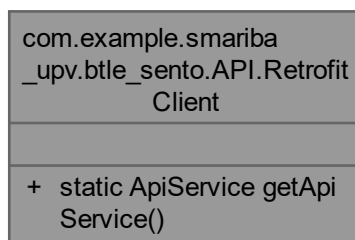
La documentación de esta clase está generada del siguiente archivo:

- [Medicion.java](#)

6.4. Referencia de la clase com.example.smariba_upv.btle_sento.API.RetrofitClient

Clase que proporciona la instancia de Retrofit configurada para realizar llamadas a la [API](#) REST.

Diagrama de colaboración de com.example.smariba_upv.btle_sento.API.RetrofitClient:



Métodos públicos estáticos

- static [ApiService](#) [getApiService](#) ()

Método estático que devuelve una instancia de [ApiService](#) para hacer llamadas a la [API](#).

6.4.1. Descripción detallada

Clase que proporciona la instancia de Retrofit configurada para realizar llamadas a la [API](#) REST.

Definición en la línea 17 del archivo [RetrofitClient.java](#).

6.4.2. Documentación de funciones miembro

getApiService()

```
static ApiService com.example.smariba_upv.btle_sento.API.RetrofitClient.getApiService () [static]
```

Método estático que devuelve una instancia de [ApiService](#) para hacer llamadas a la [API](#).

Este método comprueba si la instancia de Retrofit ya ha sido creada. Si no lo ha sido, la crea y configura. Utiliza el patrón Singleton para asegurarse de que solo se cree una instancia de Retrofit.

Devuelve

Retorna una instancia de [ApiService](#) que contiene las definiciones de las operaciones [API](#).

Definición en la línea 34 del archivo [RetrofitClient.java](#).

La documentación de esta clase está generada del siguiente archivo:

- [RetrofitClient.java](#)

6.5. Referencia de la clase com.example.smariba_upv.btle_sento.POJO.TramaBeacon

Clase que modela los datos de una trama iBeacon y permite acceder a sus diferentes componentes, tales como UUID, Major, Minor y más.

Diagrama de colaboración de com.example.smariba_upv.btle_sento.POJO.TramaBeacon:

```
com.example.smariba
_upv.btle_sento.POJO.Trama
IBeacon

+ TramaBeacon(byte
[] bytes)
+ byte[] getPrefijo()
+ byte[] getUUID()
+ byte[] getMajor()
+ byte[] getMinor()
+ byte getTxPower()
+ byte[] getLosBytes()
+ byte[] getAdvFlags()
+ byte[] getAdvHeader()
+ byte[] getCompanyID()
+ byte getiBeaconType()
+ byte getiBeaconLength()
```

Métodos públicos

- `TramalBeacon` (`byte[] bytes`)

Constructor de la clase `TramalBeacon`.

- `byte[] getPrefijo ()`
- `byte[] getUUID ()`
- `byte[] getMajor ()`
- `byte[] getMinor ()`
- `byte getTxPower ()`

Obtiene el valor de Tx Power de la trama `iBeacon`.

- `byte[] getLosBytes ()`
- `byte[] getAdvFlags ()`
- `byte[] getAdvHeader ()`
- `byte[] getCompanyID ()`
- `byte getiBeaconType ()`

Obtiene el tipo de `iBeacon`.

- `byte getiBeaconLength ()`

Obtiene la longitud de los datos `iBeacon`.

6.5.1. Descripción detallada

Clase que modela los datos de una trama `iBeacon` y permite acceder a sus diferentes componentes, tales como UUID, Major, Minor y más.

Definición en la línea 16 del archivo `TramalBeacon.java`.

6.5.2. Documentación de constructores y destructores

`TramalBeacon()`

```
com.example.smariba_upv.btle_sento.POJO.TramaIBeacon.TramaIBeacon (
    byte[] bytes)
```

Constructor de la clase `TramalBeacon`.

Parámetros

<code>bytes</code>	Array de bytes que contiene los datos de la trama <code>iBeacon</code> .
--------------------	--

Definición en la línea 35 del archivo `TramalBeacon.java`.

6.5.3. Documentación de funciones miembro

`getAdvFlags()`

```
byte[] com.example.smariba_upv.btle_sento.POJO.TramaIBeacon.getAdvFlags ()
```

Definición en la línea 110 del archivo `TramalBeacon.java`.

getAdvHeader()

```
byte[] com.example.smariba_upv.btle_sento.POJO.TramaIBeacon.getAdvHeader ()
```

Definición en la línea 119 del archivo [TramaIBeacon.java](#).

getCompanyID()

```
byte[] com.example.smariba_upv.btle_sento.POJO.TramaIBeacon.getCompanyID ()
```

Definición en la línea 128 del archivo [TramaIBeacon.java](#).

getiBeaconLength()

```
byte com.example.smariba_upv.btle_sento.POJO.TramaIBeacon.getiBeaconLength ()
```

Obtiene la longitud de los datos iBeacon.

Devuelve

Un byte que representa la longitud del iBeacon.

Definición en la línea 146 del archivo [TramaIBeacon.java](#).

getiBeaconType()

```
byte com.example.smariba_upv.btle_sento.POJO.TramaIBeacon.getiBeaconType ()
```

Obtiene el tipo de iBeacon.

Devuelve

Un byte que representa el tipo de iBeacon.

Definición en la línea 137 del archivo [TramaIBeacon.java](#).

getLosBytes()

```
byte[] com.example.smariba_upv.btle_sento.POJO.TramaIBeacon.getLosBytes ()
```

Definición en la línea 101 del archivo [TramaIBeacon.java](#).

getMajor()

```
byte[] com.example.smariba_upv.btle_sento.POJO.TramaIBeacon.getMajor ()
```

Definición en la línea 74 del archivo [TramaIBeacon.java](#).

getMinor()

```
byte[] com.example.smariba_upv.btle_sento.POJO.TramaIBeacon.getMinor ()
```

Definición en la línea 83 del archivo [TramaIBeacon.java](#).

getPrefijo()

```
byte[] com.example.smariba_upv.btle_sento.POJO.TramaIBeacon.getPrefijo ()
```

Definición en la línea 56 del archivo [TramaIBeacon.java](#).

getTxPower()

```
byte com.example.smariba_upv.btle_sento.POJO.TramaIBeacon.getTxPower ()
```

Obtiene el valor de Tx Power de la trama iBeacon.

Devuelve

Un byte que representa el valor de Tx Power.

Definición en la línea 92 del archivo [TramaIBeacon.java](#).

getUUID()

```
byte[] com.example.smariba_upv.btle_sento.POJO.TramaIBeacon.getUUID ()
```

Definición en la línea 65 del archivo [TramaIBeacon.java](#).

La documentación de esta clase está generada del siguiente archivo:

- [TramaIBeacon.java](#)

6.6. Referencia de la clase com.example.smariba_upv.btle_sento.LOGIC.Utilidades

Clase con métodos estáticos para la manipulación de datos en forma de bytes, cadenas, y UUIDs.

Diagrama de colaboración de com.example.smariba_upv.btle_sento.LOGIC.Utilidades:

com.example.smariba _upv.btle_sento.LOGIC.Utilidades
<ul style="list-style-type: none"> + static byte[] stringToBytes (String texto) + static UUID stringToUUID (String uuid) + static String uuidToString (UUID uuid) + static String uuidToHexString (UUID uuid) + static String bytesToString (byte[] bytes) + static byte[] dosLongToBytes (long masSignificativos, long menosSignificativos) + static int bytesToInt (byte[] bytes) + static long bytesToLong (byte[] bytes) + static int bytesToIntOK (byte[] bytes) + static String bytesToHexString (byte[] bytes)

Métodos públicos estáticos

- static byte[] [stringToBytes](#) (String texto)
- static UUID [stringToUUID](#) (String uuid)
Convierte un string de 16 caracteres en un objeto UUID.
- static String [uuidToString](#) (UUID uuid)
Convierte un objeto UUID en un string.
- static String [uuidToHexString](#) (UUID uuid)
Convierte un objeto UUID en un string hexadecimal.
- static String [bytesToString](#) (byte[] bytes)
Convierte un array de bytes en un string.
- static byte[] [dosLongToBytes](#) (long masSignificativos, long menosSignificativos)

- static int `bytesToInt` (byte[] bytes)
Convierte un array de bytes en un entero.
- static long `bytesToLong` (byte[] bytes)
Convierte un array de bytes en un valor `long`.
- static int `bytesToIntOK` (byte[] bytes)
Convierte un array de bytes en un entero, con manejo de signos.
- static String `bytesToHexString` (byte[] bytes)
Convierte un array de bytes en un string hexadecimal.

6.6.1. Descripción detallada

Clase con métodos estáticos para la manipulación de datos en forma de bytes, cadenas, y UUIDs.

Definición en la línea 18 del archivo [Utilidades.java](#).

6.6.2. Documentación de funciones miembro

`bytesToHexString()`

```
static String com.example.smariba_upv.btle_sento.LOGIC.Utilidades.bytesToHexString (  
    byte[] bytes) [static]
```

Convierte un array de bytes en un string hexadecimal.

Parámetros

<code>bytes</code>	El array de bytes a convertir.
--------------------	--------------------------------

Devuelve

El string hexadecimal generado a partir del array de bytes.

Definición en la línea 162 del archivo [Utilidades.java](#).

Gráfico de llamadas a esta función:



`bytesToInt()`

```
static int com.example.smariba_upv.btle_sento.LOGIC.Utilidades.bytesToInt (  
    byte[] bytes) [static]
```

Convierte un array de bytes en un entero.

Parámetros

<i>bytes</i>	El array de bytes a convertir.
--------------	--------------------------------

Devuelve

El valor entero resultante de la conversión.

Definición en la línea 111 del archivo [Utilidades.java](#).

bytesToIntOK()

```
static int com.example.smariba_upv.btle_sento.LOGIC.Utilidades.bytesToIntOK (  
    byte[] bytes) [static]
```

Convierte un array de bytes en un entero, con manejo de signos.

Parámetros

<i>bytes</i>	El array de bytes a convertir.
--------------	--------------------------------

Devuelve

El valor entero resultante de la conversión.

Excepciones

<i>Error</i>	Si el array de bytes es demasiado largo para un entero.
--------------	---

Definición en la línea 134 del archivo [Utilidades.java](#).

bytesToLong()

```
static long com.example.smariba_upv.btle_sento.LOGIC.Utilidades.bytesToLong (  
    byte[] bytes) [static]
```

Convierte un array de bytes en un valor `long`.

Parámetros

<i>bytes</i>	El array de bytes a convertir.
--------------	--------------------------------

Devuelve

El valor `long` resultante de la conversión.

Definición en la línea 122 del archivo [Utilidades.java](#).

Gráfico de llamadas a esta función:



bytesToString()

```
static String com.example.smariba_upv.btle_sento.LOGIC.Utilidades.bytesToString (  
    byte[] bytes) [static]
```

Convierte un array de bytes en un string.

Parámetros

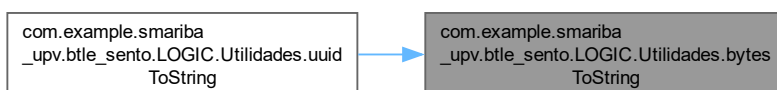
<i>bytes</i>	El array de bytes a convertir.
--------------	--------------------------------

Devuelve

El string generado a partir del array de bytes.

Definición en la línea 77 del archivo [Utilidades.java](#).

Gráfico de llamadas a esta función:

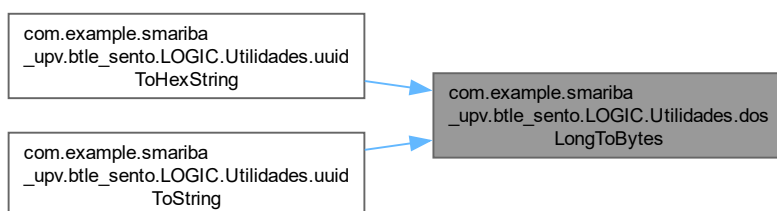


dosLongToBytes()

```
static byte[] com.example.smariba_upv.btle_sento.LOGIC.Utilidades.dosLongToBytes (  
    long masSignificativos,  
    long menosSignificativos) [static]
```

Definición en la línea 97 del archivo [Utilidades.java](#).

Gráfico de llamadas a esta función:



stringToBytes()

```
static byte[] com.example.smariba_upv.btle_sento.LOGIC.Utilidades.stringToBytes (  
    String texto) [static]
```

Definición en la línea [27](#) del archivo [Utilidades.java](#).

stringToUUID()

```
static UUID com.example.smariba_upv.btle_sento.LOGIC.Utilidades.stringToUUID (  
    String uuid) [static]
```

Convierte un string de 16 caracteres en un objeto UUID.

Parámetros

<i>uuid</i>	El string que se desea convertir a UUID.
-------------	--

Devuelve

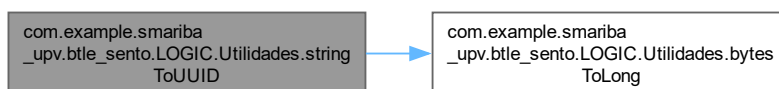
El objeto UUID generado a partir del string.

Excepciones

<i>Error</i>	Si el string no tiene exactamente 16 caracteres.
--------------	--

Definición en la línea [39](#) del archivo [Utilidades.java](#).

Gráfico de llamadas de esta función:



uuidToHexString()

```
static String com.example.smariba_upv.btle_sento.LOGIC.Utilidades.uuidToHexString (  
    UUID uuid) [static]
```

Convierte un objeto UUID en un string hexadecimal.

Parámetros

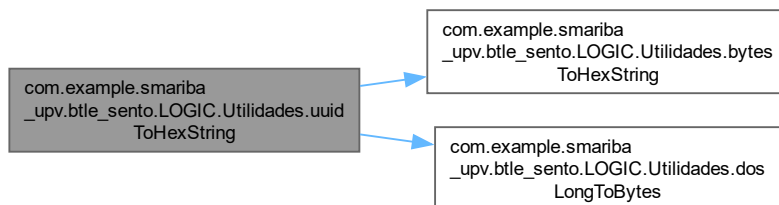
<i>uuid</i>	El objeto UUID a convertir.
-------------	-----------------------------

Devuelve

El string hexadecimal que representa al UUID.

Definición en la línea 66 del archivo [Utilidades.java](#).

Gráfico de llamadas de esta función:

**uuidToString()**

```
static String com.example.smariba_upv.btle_sento.LOGIC.Utilidades.uuidToString (
    UUID uuid) [static]
```

Convierte un objeto UUID en un string.

Parámetros

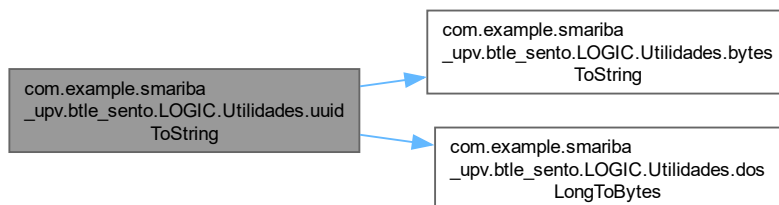
<i>uuid</i>	El objeto UUID a convertir.
-------------	-----------------------------

Devuelve

El string que representa al UUID.

Definición en la línea 55 del archivo [Utilidades.java](#).

Gráfico de llamadas de esta función:



La documentación de esta clase está generada del siguiente archivo:

- [Utilidades.java](#)

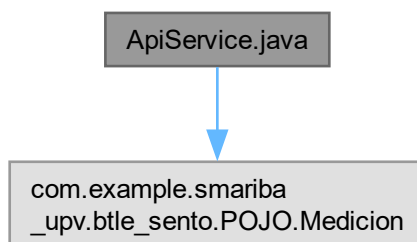
7. Documentación de archivos

7.1. Referencia del archivo ApiService.java

Interfaz para la API REST utilizada en la aplicación, donde se definen las llamadas GET y POST.

```
import com.example.smariba_upv.btle_sento.POJO.Medicion;
```

Gráfico de dependencias incluidas en ApiService.java:



Clases

- interface [com.example.smariba_upv.btle_sento.API.ApiService](#)
Interfaz que define las operaciones que se realizarán contra el servidor usando Retrofit.

Paquetes

- package [com.example.smariba_upv.btle_sento.API](#)

7.1.1. Descripción detallada

Interfaz para la API REST utilizada en la aplicación, donde se definen las llamadas GET y POST.

Versión

1.0

Fecha

2024

Definición en el archivo [ApiService.java](#).

7.2. ApiService.java

[Ir a la documentación de este archivo.](#)

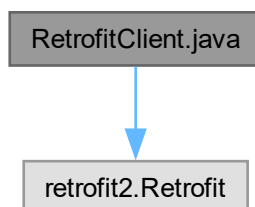
```
00001
00008 package com.example.smariba_upv.btle_sento.API;
00009
00010 import com.example.smariba_upv.btle_sento.POJO.Medicion;
00011 import retrofit2.Call;
00012 import retrofit2.http.Body;
00013 import retrofit2.http.GET;
00014 import retrofit2.http.POST;
00015
00016
00020 public interface ApiService {
00021
00022
00027     @GET("/ping")
00028     Call<Object> ping(); // Aquí puedes usar una clase específica en lugar de Object para mapear la
                                respuesta
00029
00030
00036     @POST("/insertar")
00037     Call<Void> insertarMedicion(@Body Medicion medicion);
00038 }
```

7.3. Referencia del archivo RetrofitClient.java

Clase encargada de gestionar la instancia de Retrofit para interactuar con el servidor.

```
import retrofit2.Retrofit;
```

Gráfico de dependencias incluidas en RetrofitClient.java:



Clases

- class [com.example.smariba_upv.btle_sento.API.RetrofitClient](#)

Clase que proporciona la instancia de Retrofit configurada para realizar llamadas a la [API](#) REST.

Paquetes

- package [com.example.smariba_upv.btle_sento.API](#)

7.3.1. Descripción detallada

Clase encargada de gestionar la instancia de Retrofit para interactuar con el servidor.

Versión

1.0

Fecha

2024

Definición en el archivo [RetrofitClient.java](#).

7.4. RetrofitClient.java

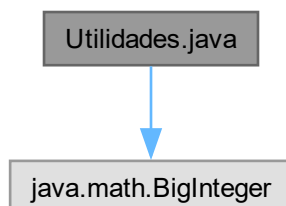
[Ir a la documentación de este archivo.](#)

```
00001
00008 package com.example.smariba_upv.btle_sento.API;
00009
00010 import retrofit2.Retrofit;
00011 import retrofit2.converter.gson.GsonConverterFactory;
00012
00013
00017 public class RetrofitClient {
00018
00019
00023     private static Retrofit retrofit = null;
00024
00025
00034     public static ApiService getApiService() {
00035         if (retrofit == null) {
00036             // Construye la instancia de Retrofit con la URL base y el convertidor Gson
00037             retrofit = new Retrofit.Builder()
00038                 .baseUrl("http://192.168.1.19:3000/") // Cambia la IP por la de tu servidor
00039                 .addConverterFactory(GsonConverterFactory.create())
00040                 .build();
00041         }
00042         // Retorna la instancia de ApiService creada a partir de Retrofit
00043         return retrofit.create(ApiService.class);
00044     }
00045 }
```

7.5. Referencia del archivo Utilidades.java

Clase con métodos utilitarios para manipulación de bytes, UUIDs y conversión de tipos.

```
import java.math.BigInteger;
Gráfico de dependencias incluidas en Utilidades.java:
```



Clases

- class [com.example.smariba_upv.btle_sento.LOGIC.Utilidades](#)

Clase con métodos estáticos para la manipulación de datos en forma de bytes, cadenas, y UUIDs.

Paquetes

- package [com.example.smariba_upv.btle_sento.LOGIC](#)

7.5.1. Descripción detallada

Clase con métodos utilitarios para manipulación de bytes, UUIDs y conversión de tipos.

Versión

1.0

Fecha

2024

Definición en el archivo [Utilidades.java](#).

7.6. Utilidades.java

[Ir a la documentación de este archivo.](#)

```
00001
00008 package com.example.smariba_upv.btle_sento.LOGIC;
00009
00010 import java.math.BigInteger;
00011 import java.nio.ByteBuffer;
00012 import java.util.UUID;
00013
00014
00018 public class Utilidades {
00019
00020
00027     public static byte[] stringToBytes(String texto) {
00028         return texto.getBytes();
00029     }
00030
00031
00039     public static UUID stringToUUID(String uuid) {
00040         if (uuid.length() != 16) {
00041             throw new Error("stringUUID: string no tiene 16 caracteres");
00042         }
00043         String masSignificativo = uuid.substring(0, 8);
00044         String menosSignificativo = uuid.substring(8, 16);
00045         return new UUID(Utilidades.bytesToLong(masSignificativo.getBytes()),
00046             Utilidades.bytesToLong(menosSignificativo.getBytes()));
00047     }
00048
00055     public static String uuidToString(UUID uuid) {
00056         return bytesToString(dosLongToBytes(uuid.getMostSignificantBits(),
00057             uuid.getLeastSignificantBits()));
00058     }
00059
00066     public static String uuidToHexString(UUID uuid) {
00067         return bytesToHexString(dosLongToBytes(uuid.getMostSignificantBits(),
00068             uuid.getLeastSignificantBits()));
00069     }
00070 }
```

```

00077     public static String bytesToString(byte[] bytes) {
00078         if (bytes == null) {
00079             return "";
00080         }
00081
00082         StringBuilder sb = new StringBuilder();
00083         for (byte b : bytes) {
00084             sb.append((char) b);
00085         }
00086         return sb.toString();
00087     }
00088
00089
00097     public static byte[] dosLongToBytes(long masSignificativos, long menosSignificativos) {
00098         ByteBuffer buffer = ByteBuffer.allocate(2 * Long.BYTES);
00099         buffer.putLong(masSignificativos);
00100         buffer.putLong(menosSignificativos);
00101         return buffer.array();
00102     }
00103
00104
00111     public static int bytesToInt(byte[] bytes) {
00112         return new BigInteger(bytes).intValue();
00113     }
00114
00115
00122     public static long bytesToLong(byte[] bytes) {
00123         return new BigInteger(bytes).longValue();
00124     }
00125
00126
00134     public static int bytesToIntOK(byte[] bytes) {
00135         if (bytes == null) {
00136             return 0;
00137         }
00138
00139         if (bytes.length > 4) {
00140             throw new Error("demasiados bytes para pasar a int");
00141         }
00142
00143         int res = 0;
00144         for (byte b : bytes) {
00145             res = (res « 8) + (b & 0xFF); // Asegurarse de obtener solo los 8 bits menos
significativos de cada byte
00146         }
00147
00148         if ((bytes[0] & 0x8) != 0) { // Si el número es negativo
00149             res = ~(byte) res) - 1; // Realiza el complemento a 2
00150         }
00151
00152         return res;
00153     }
00154
00155
00162     public static String bytesToHexString(byte[] bytes) {
00163         if (bytes == null) {
00164             return "";
00165         }
00166
00167         StringBuilder sb = new StringBuilder();
00168         for (byte b : bytes) {
00169             sb.append(String.format("%02x", b));
00170             sb.append(':');
00171         }
00172         return sb.toString();
00173     }
00174
00175 } // class

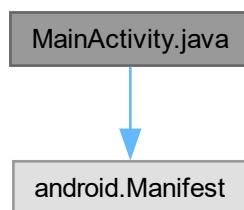
```

7.7. Referencia del archivo MainActivity.java

Clase principal de la aplicación desde la que se inicia el escaneo de dispositivos BTLE.

```
import android.Manifest;
```

Gráfico de dependencias incluidas en MainActivity.java:



Clases

- class [com.example.smariba_upv.btle_sento.MainActivity](#)
La clase principal que gestiona la interfaz y las funciones BTLE.

Paquetes

- package [com.example.smariba_upv.btle_sento](#)

7.7.1. Descripción detallada

Clase principal de la aplicación desde la que se inicia el escaneo de dispositivos BTLE.

Versión

1.0

Fecha

2024

Definición en el archivo [MainActivity.java](#).

7.8. MainActivity.java

[Ir a la documentación de este archivo.](#)

```

00001 package com.example.smariba_upv.btle_sento;
00002
00008 import android.Manifest;
00009 import android.bluetooth.BluetoothAdapter;
00010 import android.bluetooth.BluetoothDevice;
00011 import android.bluetooth.le.BluetoothLeScanner;
00012 import android.bluetooth.le.ScanCallback;
00013 import android.bluetooth.le.ScanFilter;
00014 import android.bluetooth.le.ScanResult;
00015 import android.content.pm.PackageManager;
00016 import android.os.Bundle;
00017 import android.util.Log;
00018 import android.view.View;
00019 import android.widget.Button;
00020 import android.widget.TextView;
00021
00022 import androidx.appcompat.app.AppCompatActivity;
00023 import androidx.core.app.ActivityCompat;
00024 import androidx.core.content.ContextCompat;
00025
00026
00027 import com.example.smariba_upv.btle_sento.API.RetrofitClient;
00028 import com.example.smariba_upv.btle_sento.LOGIC.Utilidades;
00029 import com.example.smariba_upv.btle_sento.POJO.Medicion;
00030 import com.example.smariba_upv.btle_sento.POJO.TramaIBeacon;
00031
00032 import java.util.List;
00033
00034 import retrofit2.Call;
00035 import retrofit2.Callback;
00036 import retrofit2.Response;
00037
00038
00039
00043 public class MainActivity extends AppCompatActivity {
00044
00045
00051     private static final String ETIQUETA_LOG = "»»»";
00052     private static final int CODIGO_PETICION_PERMISOS = 11223344;
00053
00054     private BluetoothLeScanner elEscanner;
00055     private ScanCallback callbackDelEscaneo = null;
00056     private TextView txt_datos;
00057
00058
00059
00064     private void buscarTodosLosDispositivosBTLE() {
00065         Log.d(ETIQUETA_LOG, " buscarTodosLosDispositivosBTL(): empieza ");
00066
00067         Log.d(ETIQUETA_LOG, " buscarTodosLosDispositivosBTL(): instalamos scan callback ");
00068
00069         this.callbackDelEscaneo = new ScanCallback() {
00070             @Override
00071             public void onScanResult(int callbackType, ScanResult resultado) {
00072                 super.onScanResult(callbackType, resultado);
00073                 Log.d(ETIQUETA_LOG, " buscarTodosLosDispositivosBTL(): onScanResult() ");
00074
00075                 mostrarInformacionDispositivoBTLE(resultado);
00076             }
00077
00078             @Override
00079             public void onBatchScanResults(List<ScanResult> results) {
00080                 super.onBatchScanResults(results);
00081                 Log.d(ETIQUETA_LOG, " buscarTodosLosDispositivosBTL(): onBatchScanResults() ");
00082             }
00083
00084             @Override
00085             public void onScanFailed(int errorCode) {
00086                 super.onScanFailed(errorCode);
00087                 Log.d(ETIQUETA_LOG, " buscarTodosLosDispositivosBTL(): onScanFailed() ");
00088             }
00089         };
00090
00091     };
00092
00093     Log.d(ETIQUETA_LOG, " buscarTodosLosDispositivosBTL(): empezamos a escanear ");
00094
00095     if (ActivityCompat.checkSelfPermission(this, Manifest.permission.BLUETOOTH_SCAN) !=
    PackageManager.PERMISSION_GRANTED) {
00096         Log.d(ETIQUETA_LOG, " buscarTodosLosDispositivosBTL(): NO tengo permisos para escanear ");
00097         ActivityCompat.requestPermissions(
00098             MainActivity.this,
00099             new String[]{Manifest.permission.BLUETOOTH_SCAN},

```

```

00100         CODIGO_PETICION_PERMISOS);
00101         return;
00102     }
00103     this.elEscanner.startScan(this.callbackDelEscaneo);
00104
00105 } // ()
00106
00107
00112     private void mostrarInformacionDispositivoBTLE(ScanResult resultado) {
00113
00114         BluetoothDevice bluetoothDevice = resultado.getDevice();
00115         byte[] bytes = resultado.getScanRecord().getBytes();
00116         int rssi = resultado.getRssi();
00117
00118         Log.d(ETIQUETA_LOG, " *****");
00119         Log.d(ETIQUETA_LOG, " ***** DISPOSITIVO DETECTADO BTLE *****");
00120         Log.d(ETIQUETA_LOG, " *****");
00121         if (ActivityCompat.checkSelfPermission(this, Manifest.permission.BLUETOOTH_CONNECT) !=
PackageManager.PERMISSION_GRANTED) {
00122             Log.d(ETIQUETA_LOG, " mostrarInformacionDispositivoBTLE(): NO tengo permisos para
conectar ");
00123             ActivityCompat.requestPermissions(
00124                 MainActivity.this,
00125                 new String[]{Manifest.permission.BLUETOOTH_CONNECT},
00126                 CODIGO_PETICION_PERMISOS);
00127             return;
00128         }
00129         Log.d(ETIQUETA_LOG, " nombre = " + bluetoothDevice.getName());
00130         Log.d(ETIQUETA_LOG, " toString = " + bluetoothDevice.toString());
00131
00132         /*
00133         ParcelUuid[] puuids = bluetoothDevice.getUuids();
00134         if ( puuids.length >= 1 ) {
00135             //Log.d(ETIQUETA_LOG, " uuid = " + puuids[0].getUuid());
00136             // Log.d(ETIQUETA_LOG, " uuid = " + puuids[0].toString());
00137         }*/
00138
00139         Log.d(ETIQUETA_LOG, " dirección = " + bluetoothDevice.getAddress());
00140         Log.d(ETIQUETA_LOG, " rssi = " + rssi);
00141
00142         Log.d(ETIQUETA_LOG, " bytes = " + new String(bytes));
00143         Log.d(ETIQUETA_LOG, " bytes (" + bytes.length + ") = " + Utilidades.bytesToHexString(bytes));
00144
00145         TramaIBeacon tib = new TramaIBeacon(bytes);
00146
00147         Log.d(ETIQUETA_LOG, " -----");
00148         Log.d(ETIQUETA_LOG, " prefijo = " + Utilidades.bytesToHexString(tib.getPrefijo()));
00149         Log.d(ETIQUETA_LOG, "         advFlags = " + Utilidades.bytesToHexString(tib.getAdvFlags()));
00150         Log.d(ETIQUETA_LOG, "         advHeader = " +
Utilidades.bytesToHexString(tib.getAdvHeader()));
00151         Log.d(ETIQUETA_LOG, "         companyID = " +
Utilidades.bytesToHexString(tib.getCompanyID()));
00152         Log.d(ETIQUETA_LOG, "         iBeacon type = " + Integer.toHexString(tib.getiBeaconType()));
00153         Log.d(ETIQUETA_LOG, "         iBeacon length 0x = " +
Integer.toHexString(tib.getiBeaconLength()) + " ( "
+ tib.getiBeaconLength() + " )");
00154         Log.d(ETIQUETA_LOG, "         uuid = " + Utilidades.bytesToHexString(tib.getUUID()));
00155         Log.d(ETIQUETA_LOG, "         uuid = " + Utilidades.bytesToString(tib.getUUID()));
00156         Log.d(ETIQUETA_LOG, "         major = " + Utilidades.bytesToHexString(tib.getMajor()) + " ( "
+ Utilidades.bytesToInt(tib.getMajor()) + " )");
00157         Log.d(ETIQUETA_LOG, "         minor = " + Utilidades.bytesToHexString(tib.getMinor()) + " ( "
+ Utilidades.bytesToInt(tib.getMinor()) + " )");
00158         Log.d(ETIQUETA_LOG, "         txPower = " + Integer.toHexString(tib.getTxPower()) + " ( " +
tib.getTxPower() + " )");
00159         Log.d(ETIQUETA_LOG, " *****");
00160
00161     } // ()
00162
00163
00164
00165
00166
00171     private void buscarEsteDispositivoBTLE(final String dispositivoBuscado) {
00172         //Log.d(ETIQUETA_LOG, " buscarEsteDispositivoBTLE(): empieza ");
00173
00174         //Log.d(ETIQUETA_LOG, " buscarEsteDispositivoBTLE(): instalamos scan callback ");
00175
00176
00177         // super.onScanResult(ScanSettings.SCAN_MODE_LOW_LATENCY, result); para ahorro de energía
00178
00179         this.callbackDelEscaneo = new ScanCallback() {
00180             @Override
00181             public void onScanResult(int callbackType, ScanResult resultado) {
00182                 super.onScanResult(callbackType, resultado);
00183                 //Log.d(ETIQUETA_LOG, " buscarEsteDispositivoBTLE(): onScanResult() ");
00184
00185                 if (ActivityCompat.checkSelfPermission(MainActivity.this,
Manifest.permission.BLUETOOTH_CONNECT) != PackageManager.PERMISSION_GRANTED) {
00186                     // TODO: Consider calling
00187                     // ActivityCompat#requestPermissions

```

```

00188         // here to request the missing permissions, and then overriding
00189         // public void onRequestPermissionsResult(int requestCode, String[] permissions,
00190         //                                         int[] grantResults)
00191         // to handle the case where the user grants the permission. See the documentation
00192         // for ActivityCompat#requestPermissions for more details.
00193         return;
00194     }
00195
00196     byte[] bytes = resultado.getScanRecord().getBytes();
00197     TramaIBeacon tib = new TramaIBeacon(bytes);
00198     if (Utilidades.bytesToString(tib.getUUID()).equals(dispositivoBuscado)) {
00199         mostrarInformacionDispositivoBTLE(resultado);
00200         txt_datos.setText("Major: " + (int) getMedicionsBeacon(resultado));
00201         insertarMedicion((int) getMedicionsBeacon(resultado));
00202     }
00203
00204     else {
00205         //Log.d(ETIQUETA_LOG, " buscarEsteDispositivoBTLE(): onScanResult(): no es el
00206         dispositivo buscado ");
00207     }
00208 }
00209
00210 @Override
00211 public void onBatchScanResults(List<ScanResult> results) {
00212     super.onBatchScanResults(results);
00213     //Log.d(ETIQUETA_LOG, " buscarEsteDispositivoBTLE(): onBatchScanResults() ");
00214 }
00215
00216 @Override
00217 public void onScanFailed(int errorCode) {
00218     super.onScanFailed(errorCode);
00219     //Log.d(ETIQUETA_LOG, " buscarEsteDispositivoBTLE(): onScanFailed() ");
00220 }
00221
00222 };
00223
00224 ScanFilter sf = new ScanFilter.Builder().setDeviceName(dispositivoBuscado).build();
00225
00226 //Log.d(ETIQUETA_LOG, " buscarEsteDispositivoBTLE(): empezamos a escanear buscando: " +
00227 dispositivoBuscado);
00228 //Log.d(ETIQUETA_LOG, " buscarEsteDispositivoBTLE(): empezamos a escanear buscando: " +
00229 dispositivoBuscado
00230 // + " -> " + Utilidades.stringToUUID( dispositivoBuscado ) );
00231
00232 if (ActivityCompat.checkSelfPermission(this, Manifest.permission.BLUETOOTH_SCAN) !=
00233 PackageManager.PERMISSION_GRANTED) {
00234     //Log.d(ETIQUETA_LOG, " buscarEsteDispositivoBTLE(): NO tengo permisos para escanear ");
00235     ActivityCompat.requestPermissions(
00236         MainActivity.this,
00237         new String[]{Manifest.permission.BLUETOOTH_SCAN},
00238         CODIGO_PETICION_PERMISOS);
00239     return;
00240 }
00241 this.elEscanner.startScan(this.callbackDelEscaneo);
00242 } // ()
00243
00244 private void detenerBusquedaDispositivosBTLE() {
00245     if (this.callbackDelEscaneo == null) {
00246         return;
00247     }
00248
00249     if (ActivityCompat.checkSelfPermission(this, Manifest.permission.BLUETOOTH_SCAN) !=
00250 PackageManager.PERMISSION_GRANTED) {
00251         Log.d(ETIQUETA_LOG, " detenerBusquedaDispositivosBTLE(): NO tengo permisos para escanear
00252 ");
00253         ActivityCompat.requestPermissions(
00254             MainActivity.this,
00255             new String[]{Manifest.permission.BLUETOOTH_SCAN},
00256             CODIGO_PETICION_PERMISOS);
00257         return;
00258     }
00259     this.elEscanner.stopScan(this.callbackDelEscaneo);
00260     this.callbackDelEscaneo = null;
00261 } // ()
00262
00263 public void botonBuscarDispositivosBTLEPulsado(View v) {
00264     Log.d(ETIQUETA_LOG, " boton buscar dispositivos BTLE Pulsado");
00265     this.buscarTodosLosDispositivosBTLE();
00266 } // ()

```



```

00282     public void botonBuscarNuestroDispositivoBTLEPulsado(View v) {
00283         Log.d(ETIQUETA_LOG, " boton nuestro dispositivo BTLE Pulsado");
00284         //this.buscarEsteDispositivoBTLE( Utilidades.stringToUUID( "EPSPG-GTI-PROY-3A" ) );
00285
00286         this.buscarEsteDispositivoBTLE( "EPSPG-GTI-PROY-3D" );
00287         // this.buscarEsteDispositivoBTLE("fistro");
00288
00289     } // ()
00290
00291
00297     public void botonDetenerBusquedaDispositivosBTLEPulsado(View v) {
00298         Log.d(ETIQUETA_LOG, " boton detener busqueda dispositivos BTLE Pulsado");
00299         this.detenerBusquedaDispositivosBTLE();
00300     } // ()
00301
00302
00306     private void inicializarBlueTooth() {
00307         Log.d(ETIQUETA_LOG, " inicializarBlueTooth(): obtenemos adaptador BT ");
00308
00309         BluetoothAdapter bta = BluetoothAdapter.getDefaultAdapter();
00310
00311         Log.d(ETIQUETA_LOG, " inicializarBlueTooth(): habilitamos adaptador BT ");
00312
00313         if (ActivityCompat.checkSelfPermission(this, Manifest.permission.BLUETOOTH_CONNECT) !=
00314             PackageManager.PERMISSION_GRANTED) {
00315             Log.d(ETIQUETA_LOG, " inicializarBlueTooth(): NO tengo permisos para conectar ");
00316             ActivityCompat.requestPermissions(
00317                 MainActivity.this,
00318                 new String[]{Manifest.permission.BLUETOOTH_CONNECT},
00319                 CODIGO_PETICION_PERMISOS);
00320             return;
00321         }
00322         bta.enable();
00323
00324         Log.d(ETIQUETA_LOG, " inicializarBlueTooth(): habilitado = " + bta.isEnabled() );
00325         Log.d(ETIQUETA_LOG, " inicializarBlueTooth(): estado = " + bta.getState() );
00326
00327         Log.d(ETIQUETA_LOG, " inicializarBlueTooth(): obtenemos escaner btle ");
00328
00329         this.elEscanner = bta.getBluetoothLeScanner();
00330
00331         if ( this.elEscanner == null ) {
00332             Log.d(ETIQUETA_LOG, " inicializarBlueTooth(): Socorro: NO hemos obtenido escaner btle
00333             !!!!");
00334         }
00335
00336         Log.d(ETIQUETA_LOG, " inicializarBlueTooth(): voy a pedir permisos (si no los tuviera)
00337         !!!!");
00338
00339         if (
00340             ContextCompat.checkSelfPermission(this, Manifest.permission.BLUETOOTH) !=
00341             PackageManager.PERMISSION_GRANTED
00342             || ContextCompat.checkSelfPermission(this,
00343             Manifest.permission.BLUETOOTH_ADMIN) != PackageManager.PERMISSION_GRANTED
00344             || ContextCompat.checkSelfPermission(this,
00345             Manifest.permission.ACCESS_FINE_LOCATION) != PackageManager.PERMISSION_GRANTED
00346             ) {
00347             ActivityCompat.requestPermissions(
00348                 MainActivity.this,
00349                 new String[]{Manifest.permission.BLUETOOTH, Manifest.permission.BLUETOOTH_ADMIN,
00350                 Manifest.permission.ACCESS_FINE_LOCATION},
00351                 CODIGO_PETICION_PERMISOS);
00352         }
00353         else {
00354             Log.d(ETIQUETA_LOG, " inicializarBlueTooth(): parece que YA tengo los permisos necesarios
00355             !!!!");
00356         }
00357     } // ()
00358
00359
00360     private void verificarPermisosBluetooth() {
00361         Log.d(ETIQUETA_LOG, " verificarPermisosBluetooth(): Verificando permisos");
00362
00363         // Verificamos si los permisos necesarios ya están concedidos
00364         if (ContextCompat.checkSelfPermission(this, Manifest.permission.BLUETOOTH) !=
00365             PackageManager.PERMISSION_GRANTED
00366             || ContextCompat.checkSelfPermission(this, Manifest.permission.BLUETOOTH_ADMIN) !=
00367             PackageManager.PERMISSION_GRANTED
00368             || ContextCompat.checkSelfPermission(this, Manifest.permission.ACCESS_FINE_LOCATION)
00369             != PackageManager.PERMISSION_GRANTED
00370             || ContextCompat.checkSelfPermission(this, Manifest.permission.BLUETOOTH_SCAN) !=
00371             PackageManager.PERMISSION_GRANTED

```

```

00368         || ContextCompat.checkSelfPermission(this, Manifest.permission.BLUETOOTH_CONNECT) !=
PackageManager.PERMISSION_GRANTED) {
00369
00370         // Si no están concedidos, los solicitamos
00371         Log.d(ETIQUETA_LOG, " verificarPermisosBluetooth(): Solicitando permisos necesarios");
00372         ActivityCompat.requestPermissions(
00373             MainActivity.this,
00374             new String[]{
00375                 Manifest.permission.BLUETOOTH,
00376                 Manifest.permission.BLUETOOTH_ADMIN,
00377                 Manifest.permission.ACCESS_FINE_LOCATION,
00378                 Manifest.permission.BLUETOOTH_SCAN,
00379                 Manifest.permission.BLUETOOTH_CONNECT
00380             },
00381             CODIGO_PETICION_PERMISOS
00382         );
00383     } else {
00384         Log.d(ETIQUETA_LOG, " verificarPermisosBluetooth(): Permisos ya concedidos");
00385         inicializarBlueTooth(); // Llamamos a la inicialización de Bluetooth si los permisos
están concedidos
00386     }
00387 } // ()
00388
00389
00394     @Override
00395     protected void onCreate(Bundle savedInstanceState) {
00396         super.onCreate(savedInstanceState);
00397         setContentView(R.layout.activity_main);
00398
00399         Log.d(ETIQUETA_LOG, " onCreate(): empieza ");
00400
00401         // Verificamos los permisos antes de cualquier otra operación
00402         verificarPermisosBluetooth();
00403         Button enviarMedicionButton = findViewById(R.id.btn_insertar);
00404         txt_datos = findViewById(R.id.Txt_datos);
00405
00406     } // onCreate()
00407
00408
00415     @Override
00416     public void onRequestPermissionsResult(int requestCode, String[] permissions, int[] grantResults)
{
00417         super.onRequestPermissionsResult(requestCode, permissions, grantResults);
00418
00419         switch (requestCode) {
00420             case CODIGO_PETICION_PERMISOS:
00421                 if (grantResults.length > 0 && grantResults[0] == PackageManager.PERMISSION_GRANTED) {
00422                     Log.d(ETIQUETA_LOG, " onRequestPermissionsResult(): Permisos concedidos");
00423                     inicializarBlueTooth(); // Llamamos a la inicialización si se conceden los
permisos
00424                 } else {
00425                     Log.d(ETIQUETA_LOG, " onRequestPermissionsResult(): Permisos NO concedidos");
00426                 }
00427                 return;
00428             }
00429         } // ()
00430
00431
00436     private void insertarMedicion(int major) {
00437         RetrofitClient.getApiService().insertarMedicion(new Medicion("Living Room", "CO2",
major)).enqueue(new Callback<Void>() {
00438             @Override
00439             public void onResponse(Call<Void> call, Response<Void> response) {
00440                 if (response.isSuccessful()) {
00441                     Log.d("API Response", "Medición insertada correctamente");
00442                 } else {
00443                     Log.d("API Error", "Error en la respuesta: " + response.code());
00444                 }
00445             }
00446
00447             @Override
00448             public void onFailure(Call<Void> call, Throwable t) {
00449                 Log.e("API Failure", "Error al conectar con el servidor", t);
00450             }
00451         });
00452
00453     }
00454
00460     private double getMedicionsBeacon (ScanResult resultado) {
00461         byte[] bytes = resultado.getScanRecord().getBytes();
00462         TramaIBeacon tib = new TramaIBeacon(bytes);
00463         return Utilidades.bytesToInt(tib.getMinor());
00464     }
00465
00466 } // class

```

7.9. Referencia del archivo Medicion.java

Clase que representa una medición de gas en un lugar determinado.

Clases

- class [com.example.smariba_upv.btle_sento.POJO.Medicion](#)

Clase que encapsula los datos de una medición de gas, incluyendo el lugar, el tipo de gas y su valor.

Paquetes

- package [com.example.smariba_upv.btle_sento.POJO](#)

7.9.1. Descripción detallada

Clase que representa una medición de gas en un lugar determinado.

Versión

1.0

Fecha

2024

Definición en el archivo [Medicion.java](#).

7.10. Medicion.java

[Ir a la documentación de este archivo.](#)

```
00001
00008 package com.example.smariba_upv.btle_sento.POJO;
00009
00010
00014 public class Medicion {
00015     private String Lugar;
00016     private String Gas;
00017     private int Valor;
00018
00019
00025     public Medicion(String Lugar, String Gas, int Valor) {
00026         this.Lugar = Lugar;
00027         this.Gas = Gas;
00028         this.Valor = Valor;
00029     }
00030
00031
00035     public String getLugar() {
00036         return Lugar;
00037     }
00038
00039
00043     public void setLugar(String lugar) {
00044         Lugar = lugar;
00045     }
00046
00047
00051     public String getGas() {
00052         return Gas;
00053     }
```

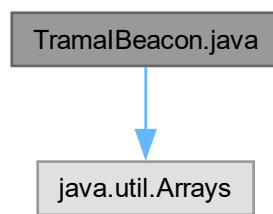
```
00054
00055
00059     public void setGas(String gas) {
00060         Gas = gas;
00061     }
00062
00063
00067     public int getValor() {
00068         return Valor;
00069     }
00070
00071
00075     public void setValor(int valor) {
00076         Valor = valor;
00077     }
00078 }
```

7.11. Referencia del archivo TramalBeacon.java

Clase que representa una trama iBeacon, extrayendo sus diferentes campos a partir de un array de bytes.

```
import java.util.Arrays;
```

Gráfico de dependencias incluidas en TramalBeacon.java:



Clases

- class [com.example.smariba_upv.btle_sento.POJO.TramalBeacon](#)
Clase que modela los datos de una trama iBeacon y permite acceder a sus diferentes componentes, tales como UUID, Major, Minor y más.

Paquetes

- package [com.example.smariba_upv.btle_sento.POJO](#)

7.11.1. Descripción detallada

Clase que representa una trama iBeacon, extrayendo sus diferentes campos a partir de un array de bytes.

Versión

1.0

Fecha

2024

Definición en el archivo [TramalBeacon.java](#).

7.12. TramaBeacon.java

[Ir a la documentación de este archivo.](#)

```
00001
00008 package com.example.smariba_upv.btle_sento.POJO;
00009
00010 import java.util.Arrays;
00011
00012
00016 public class TramaIBeacon {
00017     private byte[] prefijo = null; // 9 bytes
00018     private byte[] uuid = null; // 16 bytes
00019     private byte[] major = null; // 2 bytes
00020     private byte[] minor = null; // 2 bytes
00021     private byte txPower = 0; // 1 byte
00022
00023     private byte[] losBytes;
00024
00025     private byte[] advFlags = null; // 3 bytes
00026     private byte[] advHeader = null; // 2 bytes
00027     private byte[] companyID = new byte[2]; // 2 bytes
00028     private byte iBeaconType = 0; // 1 byte
00029     private byte iBeaconLength = 0; // 1 byte
00030
00031
00035     public TramaIBeacon(byte[] bytes) {
00036         this.losBytes = bytes;
00037
00038         prefijo = Arrays.copyOfRange(losBytes, 0, 8 + 1); // 9 bytes
00039         uuid = Arrays.copyOfRange(losBytes, 9, 24 + 1); // 16 bytes
00040         major = Arrays.copyOfRange(losBytes, 25, 26 + 1); // 2 bytes
00041         minor = Arrays.copyOfRange(losBytes, 27, 28 + 1); // 2 bytes
00042         txPower = losBytes[29]; // 1 byte
00043
00044         advFlags = Arrays.copyOfRange(prefijo, 0, 2 + 1); // 3 bytes
00045         advHeader = Arrays.copyOfRange(prefijo, 3, 4 + 1); // 2 bytes
00046         companyID = Arrays.copyOfRange(prefijo, 5, 6 + 1); // 2 bytes
00047         iBeaconType = prefijo[7]; // 1 byte
00048         iBeaconLength = prefijo[8]; // 1 byte
00049     }
00050
00051
00056     public byte[] getPrefijo() {
00057         return prefijo;
00058     }
00059
00060
00065     public byte[] getUUID() {
00066         return uuid;
00067     }
00068
00069
00074     public byte[] getMajor() {
00075         return major;
00076     }
00077
00078
00083     public byte[] getMinor() {
00084         return minor;
00085     }
00086
00087
00092     public byte getTxPower() {
00093         return txPower;
00094     }
00095
00096
00101     public byte[] getLosBytes() {
00102         return losBytes;
00103     }
00104
00105
00110     public byte[] getAdvFlags() {
00111         return advFlags;
00112     }
00113
00114
00119     public byte[] getAdvHeader() {
00120         return advHeader;
00121     }
00122
00123
00128     public byte[] getCompanyID() {
00129         return companyID;
00130     }
00131 }
```

```
00132
00137     public byte getiBeaconType() {
00138         return iBeaconType;
00139     }
00140
00141
00146     public byte getiBeaconLength() {
00147         return iBeaconLength;
00148     }
00149 } // class
```

Índice alfabético

ApiService.java, [22](#), [23](#)

botonBuscarDispositivosBTLEPulsado

com.example.smariba_upv.btle_sento.MainActivity,

[7](#)

botonBuscarNuestroDispositivoBTLEPulsado

com.example.smariba_upv.btle_sento.MainActivity,

[7](#)

botonDetenerBusquedaDispositivosBTLEPulsado

com.example.smariba_upv.btle_sento.MainActivity,

[7](#)

bytesToHexString

com.example.smariba_upv.btle_sento.LOGIC.Utilidades,

[17](#)

bytesToInt

com.example.smariba_upv.btle_sento.LOGIC.Utilidades,

[17](#)

bytesToIntOK

com.example.smariba_upv.btle_sento.LOGIC.Utilidades,

[18](#)

bytesToLong

com.example.smariba_upv.btle_sento.LOGIC.Utilidades,

[18](#)

bytesToString

com.example.smariba_upv.btle_sento.LOGIC.Utilidades,

[18](#)

com.example.smariba_upv.btle_sento, [3](#)

com.example.smariba_upv.btle_sento.API, [3](#)

com.example.smariba_upv.btle_sento.API.ApiService, [4](#)

insertarMedicion, [4](#)

ping, [4](#)

com.example.smariba_upv.btle_sento.API.RetrofitClient,

[11](#)

getApiService, [12](#)

com.example.smariba_upv.btle_sento.LOGIC, [3](#)

com.example.smariba_upv.btle_sento.LOGIC.Utilidades,

[16](#)

bytesToHexString, [17](#)

bytesToInt, [17](#)

bytesToIntOK, [18](#)

bytesToLong, [18](#)

bytesToString, [18](#)

dosLongToBytes, [19](#)

stringToBytes, [19](#)

stringToUUID, [20](#)

uuidToHexString, [20](#)

uuidToString, [21](#)

com.example.smariba_upv.btle_sento.MainActivity, [5](#)

botonBuscarDispositivosBTLEPulsado, [7](#)

botonBuscarNuestroDispositivoBTLEPulsado, [7](#)

botonDetenerBusquedaDispositivosBTLEPulsado,

[7](#)

onCreate, [7](#)

onRequestPermissionsResult, [8](#)

com.example.smariba_upv.btle_sento.POJO, [3](#)

com.example.smariba_upv.btle_sento.POJO.Medicion,

[8](#)

getGas, [9](#)

getLugar, [9](#)

getValor, [10](#)

Medicion, [9](#)

setGas, [10](#)

setLugar, [10](#)

setValor, [10](#)

com.example.smariba_upv.btle_sento.POJO.TramalBeacon,

[12](#)

getAdvFlags, [13](#)

getAdvHeader, [13](#)

getCompanyID, [14](#)

getiBeaconLength, [14](#)

getiBeaconType, [14](#)

getLosBytes, [14](#)

getMajor, [14](#)

getMinor, [14](#)

getPrefijo, [15](#)

getTxPower, [15](#)

getUUID, [15](#)

TramalBeacon, [13](#)

dosLongToBytes

com.example.smariba_upv.btle_sento.LOGIC.Utilidades,

[19](#)

getAdvFlags

com.example.smariba_upv.btle_sento.POJO.TramalBeacon,

[13](#)

getAdvHeader

com.example.smariba_upv.btle_sento.POJO.TramalBeacon,

[13](#)

getApiService

com.example.smariba_upv.btle_sento.API.RetrofitClient,

[12](#)

getCompanyID

com.example.smariba_upv.btle_sento.POJO.TramalBeacon,

[14](#)

getGas

com.example.smariba_upv.btle_sento.POJO.Medicion,

[9](#)

getiBeaconLength

com.example.smariba_upv.btle_sento.POJO.TramalBeacon,

[14](#)

getiBeaconType

com.example.smariba_upv.btle_sento.POJO.TramalBeacon,

[14](#)

getLosBytes

com.example.smariba_upv.btle_sento.POJO.TramalBeacon,

[14](#)

getLugar

com.example.smariba_upv.btle_sento.POJO.Medicion,

[9](#)

getMajor com.example.smariba_upv.btle_sento.POJO.TramalBeacon, 13
14 TramalBeacon.java, 34, 35

getMinor com.example.smariba_upv.btle_sento.POJO.TramalBeacon, 14
Utilidades.java, 24, 25
uuidToHexString

getPrefijo com.example.smariba_upv.btle_sento.LOGIC.Utilidades, 20
15 uuidToString

getTxPower com.example.smariba_upv.btle_sento.LOGIC.Utilidades, 21
15

getUUID com.example.smariba_upv.btle_sento.POJO.TramalBeacon, 15

getValor com.example.smariba_upv.btle_sento.POJO.Medicion, 10

insertarMedicion com.example.smariba_upv.btle_sento.API.ApiService, 4

MainActivity.java, 26, 28

Medicion com.example.smariba_upv.btle_sento.POJO.Medicion, 9
Medicion.java, 33

onCreate com.example.smariba_upv.btle_sento.MainActivity, 7

onRequestPermissionsResult com.example.smariba_upv.btle_sento.MainActivity, 8

ping com.example.smariba_upv.btle_sento.API.ApiService, 4

RetrofitClient.java, 23, 24

setGas com.example.smariba_upv.btle_sento.POJO.Medicion, 10

setLugar com.example.smariba_upv.btle_sento.POJO.Medicion, 10

setValor com.example.smariba_upv.btle_sento.POJO.Medicion, 10

stringToBytes com.example.smariba_upv.btle_sento.LOGIC.Utilidades, 19

stringToUUID com.example.smariba_upv.btle_sento.LOGIC.Utilidades, 20

TramalBeacon