

Proyecto3A_Android

Generado por Doxygen 1.12.0

1 Proyecto3A_Android	1
1.1 Funcionalidades	1
1.2 Requisitos	1
1.3 Estructura del Proyecto	1
1.3.1 Estructura de la Base de Datos	2
1.4 Instalación	2
1.5 Uso	2
1.6 Contribuciones	2
1.7 Licencia	2
2 Índice de espacios de nombres	3
2.1 Lista de espacios de nombres	3
3 Índice jerárquico	5
3.1 Jerarquía de clases	5
4 Índice de clases	7
4.1 Lista de clases	7
5 Índice de archivos	9
5.1 Lista de archivos	9
6 Documentación de espacios de nombres	11
6.1 Paquete com.example.usuario_upv.proyecto3a	11
7 Documentación de clases	13
7.1 Referencia de la clase com.example.usuario_upv.proyecto3a.MainActivity	13
7.1.1 Descripción detallada	14
7.1.2 Documentación de funciones miembro	14
7.1.2.1 actualizarIP()	14
7.1.2.2 botonBuscarDispositivosBTLEPulsado()	14
7.1.2.3 botonBuscarNuestroDispositivoBTLEPulsado()	14
7.1.2.4 botonDetenerBusquedaDispositivosBTLEPulsado()	15
7.1.2.5 onCreate()	15
7.1.2.6 onRequestPermissionsResult()	15
7.1.2.7 procesarBeacon()	16
7.2 Referencia de la clase com.example.usuario_upv.proyecto3a.RetrofitClient	16
7.2.1 Descripción detallada	16
7.2.2 Documentación de funciones miembro	16
7.2.2.1 getClient()	16
7.3 Referencia de la interface com.example.usuario_upv.proyecto3a.SensorApi	17
7.3.1 Descripción detallada	17
7.3.2 Documentación de funciones miembro	17
7.3.2.1 checkConnection()	17

7.3.2.2 createSensorData()	18
7.3.2.3 createUser()	18
7.3.2.4 deleteUser()	18
7.3.2.5 resetTables()	19
7.3.2.6 setupTables()	19
7.4 Referencia de la clase com.example.usuario_upv.proyecto3a.SensorData	19
7.4.1 Descripción detallada	20
7.4.2 Documentación de constructores y destructores	20
7.4.2.1 SensorData()	20
7.4.3 Documentación de funciones miembro	20
7.4.3.1 getTimestamp()	20
7.4.3.2 getType()	20
7.4.3.3 getUserId()	21
7.4.3.4 getValue()	21
7.4.3.5 setTimestamp()	21
7.4.3.6 setType()	21
7.4.3.7 setUserId()	21
7.4.3.8 setValue()	21
7.5 Referencia de la clase com.example.usuario_upv.proyecto3a.TramaBeacon	22
7.5.1 Descripción detallada	22
7.5.2 Documentación de constructores y destructores	22
7.5.2.1 TramaBeacon()	22
7.5.3 Documentación de funciones miembro	23
7.5.3.1 getAdvFlags()	23
7.5.3.2 getAdvHeader()	23
7.5.3.3 getCompanyId()	23
7.5.3.4 getBeaconLength()	23
7.5.3.5 getBeaconType()	23
7.5.3.6 getLosBytes()	23
7.5.3.7 getMajor()	23
7.5.3.8 getMinor()	23
7.5.3.9 getPrefijo()	23
7.5.3.10 getTxPower()	24
7.5.3.11 getUUID()	24
7.5.3.12 setMajor()	24
7.5.3.13 setMinor()	24
7.6 Referencia de la clase com.example.usuario_upv.proyecto3a.User	24
7.6.1 Descripción detallada	24
7.6.2 Documentación de constructores y destructores	24
7.6.2.1 User()	24
7.6.3 Documentación de funciones miembro	25
7.6.3.1 getUsername()	25

7.6.3.2 setUsername()	25
7.7 Referencia de la clase com.example.usuario_upv.proyecto3a.Utilidades	25
7.7.1 Descripción detallada	26
7.7.2 Documentación de funciones miembro	26
7.7.2.1 bytesToHexString()	26
7.7.2.2 bytesToInt()	26
7.7.2.3 bytesToIntOK()	27
7.7.2.4 bytesToLong()	27
7.7.2.5 bytesToString()	27
7.7.2.6 dosLongToBytes()	28
7.7.2.7 stringToBytes()	28
7.7.2.8 stringToUUID()	28
7.7.2.9 uuidToHexString()	29
7.7.2.10 uuidToString()	29
8 Documentación de archivos	31
8.1 Referencia del archivo app/src/main/java/com/example/usuario_upv/proyecto3a/MainActivity.java	31
8.2 Referencia del archivo app/src/main/java/com/example/usuario_upv/proyecto3a/RetrofitClient.java	31
8.3 Referencia del archivo app/src/main/java/com/example/usuario_upv/proyecto3a/SensorApi.java	32
8.4 Referencia del archivo app/src/main/java/com/example/usuario_upv/proyecto3a/SensorData.java	32
8.5 Referencia del archivo app/src/main/java/com/example/usuario_upv/proyecto3a/TramalBeacon.java	32
8.6 Referencia del archivo app/src/main/java/com/example/usuario_upv/proyecto3a/User.java	33
8.7 Referencia del archivo app/src/main/java/com/example/usuario_upv/proyecto3a/Utilidades.java	33
8.8 Referencia del archivo README.md	33
Índice alfabético	35

Capítulo 1

Proyecto3A_Android

Este proyecto es una aplicación Android que utiliza beacons para medir niveles de CO2 y temperatura. La aplicación se conecta a un servidor para almacenar y recuperar datos de sensores a través de una API REST.

1.1. Funcionalidades

- **Detección de Dispositivos BTLE:** La aplicación busca y se conecta a beacons Bluetooth Low Energy.
- **Medición de Sensores:** Recopila datos de sensores (CO2 y temperatura) y los envía al servidor.
- **Gestión de Usuarios:** Permite la creación y eliminación de usuarios en el sistema.
- **Interfaz de Usuario:** Interfaz sencilla para interactuar con la aplicación y visualizar datos.

1.2. Requisitos

- Android Studio
- SDK de Android (versión mínima recomendada: 21)
- Dependencias:
 - Retrofit para la comunicación con la API REST
 - Gson para la conversión de JSON

1.3. Estructura del Proyecto

```
/app
  /src
    /main
      /java
        /com.example.usuario_upv.proyecto3a
          - MainActivity.java
          - RetrofitClient.java
          - SensorApi.java
          - SensorData.java
          - User.java
          - Utilidades.java
          - TramaIBeacon.java
      /res
        /layout
        /drawable
        /values
```

1.3.1. Estructura de la Base de Datos

- **Tabla `sensors`:**
 - `id`: Identificador único.
 - `type`: Tipo de sensor (CO2, temperatura).
 - `value`: Valor medido.
 - `timestamp`: Marca de tiempo de la medición.
 - `user_id`: Identificador del usuario.
- **Tabla `users`:**
 - `id`: Identificador único.
 - `username`: Nombre de usuario.

1.4. Instalación

1. Clona este repositorio:

```
git clone https://github.com/Javitax47/Proyecto3A_Android.git
```
2. Abre el proyecto en Android Studio.
3. Asegúrate de tener configuradas las dependencias necesarias en tu archivo `build.gradle`.
4. Ejecuta la aplicación en un dispositivo Android.

1.5. Uso

1. Abre la aplicación en tu dispositivo Android.
2. Usa los botones de la interfaz para buscar dispositivos BTLE y detener la búsqueda.
3. Se mostrarán los datos de los sensores en la interfaz.

1.6. Contribuciones

Si deseas contribuir a este proyecto, por favor, crea un fork del repositorio y envía un pull request con tus cambios.

1.7. Licencia

Este proyecto está bajo la Licencia MIT. Consulta el archivo `LICENSE` para más detalles.

Capítulo 2

Índice de espacios de nombres

2.1. Lista de espacios de nombres

Lista de los espacios de nombres documentados, con breves descripciones:

com.example.usuario_upv.proyecto3a	11
--	----

Capítulo 3

Índice jerárquico

3.1. Jerarquía de clases

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

com.example.usuario_upv.proyecto3a.RetrofitClient	16
com.example.usuario_upv.proyecto3a.SensorApi	17
com.example.usuario_upv.proyecto3a.SensorData	19
com.example.usuario_upv.proyecto3a.TramalBeacon	22
com.example.usuario_upv.proyecto3a.User	24
com.example.usuario_upv.proyecto3a.Utilidades	25
AppCompatActivity	
com.example.usuario_upv.proyecto3a.MainActivity	13

Capítulo 4

Índice de clases

4.1. Lista de clases

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

com.example.usuario_upv.proyecto3a.MainActivity	
MainActivity de la aplicación	13
com.example.usuario_upv.proyecto3a.RetrofitClient	
Clase para gestionar la instancia de Retrofit	16
com.example.usuario_upv.proyecto3a.SensorApi	
Interfaz para definir los endpoints de la API de sensores	17
com.example.usuario_upv.proyecto3a.SensorData	
Clase que representa los datos de un sensor	19
com.example.usuario_upv.proyecto3a.TramaIbeacon	
Clase que representa la trama de un beacon iBeacon	22
com.example.usuario_upv.proyecto3a.User	
Clase que representa un usuario	24
com.example.usuario_upv.proyecto3a.Utilidades	
Clase utilitaria para conversiones entre diferentes tipos de datos	25

Capítulo 5

Índice de archivos

5.1. Lista de archivos

Lista de todos los archivos con breves descripciones:

app/src/main/java/com/example/usuario_upv/proyecto3a/ MainActivity.java	31
app/src/main/java/com/example/usuario_upv/proyecto3a/ RetrofitClient.java	31
app/src/main/java/com/example/usuario_upv/proyecto3a/ SensorApi.java	32
app/src/main/java/com/example/usuario_upv/proyecto3a/ SensorData.java	32
app/src/main/java/com/example/usuario_upv/proyecto3a/ TramaBeacon.java	32
app/src/main/java/com/example/usuario_upv/proyecto3a/ User.java	33
app/src/main/java/com/example/usuario_upv/proyecto3a/ Utilidades.java	33

Capítulo 6

Documentación de espacios de nombres

6.1. Paquete com.example.usuario_upv.proyecto3a

Clases

- class [MainActivity](#)
MainActivity de la aplicación.
- class [RetrofitClient](#)
Clase para gestionar la instancia de Retrofit.
- interface [SensorApi](#)
Interfaz para definir los endpoints de la API de sensores.
- class [SensorData](#)
Clase que representa los datos de un sensor.
- class [TramalBeacon](#)
Clase que representa la trama de un beacon iBeacon.
- class [User](#)
Clase que representa un usuario.
- class [Utilidades](#)
Clase utilitaria para conversiones entre diferentes tipos de datos.

Capítulo 7

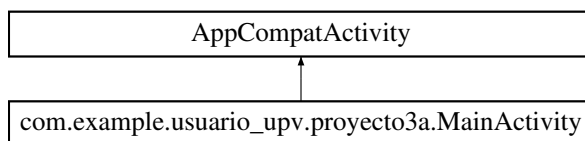
Documentación de clases

7.1. Referencia de la clase

com.example.usuario_upv.proyecto3a.MainActivity

[MainActivity](#) de la aplicación.

Diagrama de herencia de `com.example.usuario_upv.proyecto3a.MainActivity`



Métodos públicos

- void [actualizarIP](#) (View view)
Actualiza la dirección IP del servidor para la conexión con la API.
- int [procesarBeacon](#) (int major, int minor)
Procesa la información del beacon detectado.
- void [botonBuscarNuestroDispositivoBTLEPulsado](#) (View v)
Maneja el evento de pulsación del botón para buscar un dispositivo BLE específico.
- void [botonBuscarDispositivosBTLEPulsado](#) (View v)
Maneja el evento de pulsación del botón para buscar todos los dispositivos BLE.
- void [botonDetenerBusquedaDispositivosBTLEPulsado](#) (View v)
Maneja el evento de pulsación del botón para detener la búsqueda de dispositivos BLE.
- void [onRequestPermissionsResult](#) (int requestCode, String[] permissions, int[] grantResults)
Maneja el resultado de las solicitudes de permisos.

Métodos protegidos

- void [onCreate](#) (Bundle savedInstanceState)
Método llamado cuando se crea la actividad.

7.1.1. Descripción detallada

[MainActivity](#) de la aplicación.

Esta clase es la actividad principal de la aplicación que gestiona la interfaz de usuario y las interacciones con los sensores a través de Bluetooth LE.

7.1.2. Documentación de funciones miembro

7.1.2.1. actualizarIP()

```
void com.example.usuario_upv.proyecto3a.MainActivity.actualizarIP (
    View view) [inline]
```

Actualiza la dirección IP del servidor para la conexión con la API.

Este método es llamado cuando el usuario introduce una IP y presiona el botón para actualizarla. Valida que la IP no esté vacía, construye la URL base para las peticiones HTTP mediante Retrofit y actualiza la instancia de la API. Muestra un mensaje en pantalla indicando la nueva IP del servidor.

Parámetros

<i>view</i>	La vista que desencadena este método, generalmente el botón de envío de la IP.
-------------	--

7.1.2.2. botonBuscarDispositivosBTLEPulsado()

```
void com.example.usuario_upv.proyecto3a.MainActivity.botonBuscarDispositivosBTLEPulsado (
    View v) [inline]
```

Maneja el evento de pulsación del botón para buscar todos los dispositivos BLE.

Este método se invoca cuando se pulsa el botón correspondiente en la interfaz de usuario. Cambia la visibilidad de una imagen para indicar que la búsqueda está en curso y llama al método que inicia la búsqueda de todos los dispositivos BLE disponibles.

Parámetros

<i>v</i>	La vista que ha sido pulsada (el botón).
----------	--

7.1.2.3. botonBuscarNuestroDispositivoBTLEPulsado()

```
void com.example.usuario_upv.proyecto3a.MainActivity.botonBuscarNuestroDispositivoBTLEPulsado
(
    View v) [inline]
```

Maneja el evento de pulsación del botón para buscar un dispositivo BLE específico.

Este método se invoca cuando se pulsa el botón correspondiente en la interfaz de usuario. Cambia la visibilidad de una imagen y llama al método para buscar un dispositivo BLE con un UUID específico.

Parámetros

<i>v</i>	La vista que ha sido pulsada (el botón).
----------	--

7.1.2.4. `botonDetenerBusquedaDispositivosBTLEPulsado()`

```
void com.example.usuario_upv.proyecto3a.MainActivity.botonDetenerBusquedaDispositivosBTLEPulsado
(
    View v) [inline]
```

Maneja el evento de pulsación del botón para detener la búsqueda de dispositivos BLE.

Este método se invoca cuando se pulsa el botón correspondiente en la interfaz de usuario. Registra en el log que se ha pulsado el botón y llama al método que detiene la búsqueda activa de dispositivos BLE.

Parámetros

<i>v</i>	La vista que ha sido pulsada (el botón).
----------	--

7.1.2.5. `onCreate()`

```
void com.example.usuario_upv.proyecto3a.MainActivity.onCreate (
    Bundle savedInstanceState) [inline], [protected]
```

Método llamado cuando se crea la actividad.

Este método inicializa la actividad, configurando el layout y asignando las referencias a los elementos de la interfaz de usuario, como los `TextView`, `ImageView`, `EditText` y el contenedor de los beacons. También se inicializa el Bluetooth.

Parámetros

<i>savedInstanceState</i>	Estado previamente guardado de la actividad, si existe.
---------------------------	---

7.1.2.6. `onRequestPermissionsResult()`

```
void com.example.usuario_upv.proyecto3a.MainActivity.onRequestPermissionsResult (
    int requestCode,
    String[] permissions,
    int[] grantResults) [inline]
```

Maneja el resultado de las solicitudes de permisos.

Este método es llamado cuando el usuario responde a una solicitud de permisos. Verifica si los permisos solicitados han sido concedidos y registra el resultado.

Parámetros

<i>requestCode</i>	El código de la solicitud de permisos.
<i>permissions</i>	Un arreglo de permisos solicitados.
<i>grantResults</i>	Un arreglo de resultados correspondientes a cada permiso.

7.1.2.7. procesarBeacon()

```
int com.example.usuario_upv.proyecto3a.MainActivity.procesarBeacon (
    int major,
    int minor) [inline]
```

Procesa la información del beacon detectado.

Este método determina el tipo de medición a partir del valor 'major' del beacon y registra el valor 'minor'. Los tipos de medición reconocidos son CO2 y Temperatura, identificados por sus respectivos códigos.

Parámetros

<i>major</i>	El valor 'major' del beacon, que contiene el tipo de medición y un contador.
<i>minor</i>	El valor 'minor' del beacon, que representa el dato medido (por ejemplo, CO2 o temperatura).

Devuelve

Un entero que representa el tipo de medición:

- 1 si el dato es de CO2
- 2 si el dato es de temperatura
- 0 si el tipo de dato no es reconocido.

La documentación de esta clase está generada del siguiente archivo:

- [app/src/main/java/com/example/usuario_upv/proyecto3a/MainActivity.java](#)

7.2. Referencia de la clase com.example.usuario_upv.proyecto3a.RetrofitClient

Clase para gestionar la instancia de Retrofit.

Métodos públicos estáticos

- static Retrofit [getClient](#) (String baseUrl)
Obtiene la instancia de Retrofit.

7.2.1. Descripción detallada

Clase para gestionar la instancia de Retrofit.

Esta clase proporciona un método para obtener una instancia de Retrofit configurada con una URL base y un convertidor de JSON a objetos Java.

7.2.2. Documentación de funciones miembro

7.2.2.1. getClient()

```
static Retrofit com.example.usuario_upv.proyecto3a.RetrofitClient.getClient (
    String baseUrl) [inline], [static]
```

Obtiene la instancia de Retrofit.

Este método verifica si la instancia de Retrofit ya existe o si la URL base ha cambiado. Si es así, crea una nueva instancia de Retrofit.

Parámetros

<code>baseUrl</code>	La URL base que se utilizará para las solicitudes de la API.
----------------------	--

Devuelve

La instancia de Retrofit configurada.

La documentación de esta clase está generada del siguiente archivo:

- `app/src/main/java/com/example/usuario_upv/proyecto3a/RetrofitClient.java`

7.3. Referencia de la interface `com.example.usuario_upv.proyecto3a.SensorApi`

Interfaz para definir los endpoints de la API de sensores.

Métodos públicos

- `Call< Void > checkConnection ()`
Verifica la conexión con el servidor.
- `Call< Void > setupTables ()`
Crea las tablas necesarias en la base de datos.
- `Call< Void > createUser (@Body User user)`
Inserta un nuevo usuario en la base de datos.
- `Call< Void > createSensorData (@Body SensorData sensorData)`
Inserta una medición de sensor en la base de datos.
- `Call< Void > deleteUser (@Path("id") int userId)`
Elimina un usuario de la base de datos por su ID.
- `Call< Void > resetTables ()`
Resetea las tablas en la base de datos.

7.3.1. Descripción detallada

Interfaz para definir los endpoints de la API de sensores.

Esta interfaz contiene métodos para interactuar con el servidor que gestiona los datos de los sensores y los usuarios.

7.3.2. Documentación de funciones miembro

7.3.2.1. `checkConnection()`

```
Call< Void > com.example.usuario_upv.proyecto3a.SensorApi.checkConnection ()
```

Verifica la conexión con el servidor.

Este método envía una solicitud GET al servidor para comprobar si está disponible.

Devuelve

Un objeto Call que contiene la respuesta de la solicitud.

7.3.2.2. createSensorData()

```
Call< Void > com.example.usuario_upv.proyecto3a.SensorApi.createSensorData (
    @Body SensorData sensorData)
```

Inserta una medición de sensor en la base de datos.

Este método envía una solicitud POST con la información de la medición en el cuerpo de la solicitud.

Parámetros

<i>sensorData</i>	El objeto SensorData que contiene la información de la medición.
-------------------	--

Devuelve

Un objeto Call que contiene la respuesta de la solicitud.

7.3.2.3. createUser()

```
Call< Void > com.example.usuario_upv.proyecto3a.SensorApi.createUser (
    @Body User user)
```

Inserta un nuevo usuario en la base de datos.

Este método envía una solicitud POST con la información del usuario en el cuerpo de la solicitud.

Parámetros

<i>user</i>	El objeto User que contiene la información del nuevo usuario.
-------------	---

Devuelve

Un objeto Call que contiene la respuesta de la solicitud.

7.3.2.4. deleteUser()

```
Call< Void > com.example.usuario_upv.proyecto3a.SensorApi.deleteUser (
    @Path("id") int userId)
```

Elimina un usuario de la base de datos por su ID.

Este método envía una solicitud DELETE al endpoint correspondiente para eliminar al usuario especificado.

Parámetros

<i>user</i> ↔ <i>Id</i>	El ID del usuario que se desea eliminar.
----------------------------	--

Devuelve

Un objeto Call que contiene la respuesta de la solicitud.

7.3.2.5. `resetTables()`

```
Call< Void > com.example.usuario_upv.proyecto3a.SensorApi.resetTables ()
```

Resetea las tablas en la base de datos.

Este método envía una solicitud DELETE al endpoint para reiniciar las tablas.

Devuelve

Un objeto Call que contiene la respuesta de la solicitud.

7.3.2.6. `setupTables()`

```
Call< Void > com.example.usuario_upv.proyecto3a.SensorApi.setupTables ()
```

Crea las tablas necesarias en la base de datos.

Este método envía una solicitud GET al endpoint de configuración para crear las tablas requeridas.

Devuelve

Un objeto Call que contiene la respuesta de la solicitud.

La documentación de esta interface está generada del siguiente archivo:

- `app/src/main/java/com/example/usuario_upv/proyecto3a/SensorApi.java`

7.4. Referencia de la clase `com.example.usuario_upv.proyecto3a.SensorData`

Clase que representa los datos de un sensor.

Métodos públicos

- `SensorData` (String type, float value, int userId)
Constructor de la clase `SensorData`.
- String `getType` ()
- void `setType` (String type)
- float `getValue` ()
- void `setValue` (float value)
- String `getTimestamp` ()
- void `setTimestamp` (String timestamp)
- int `getUserId` ()
- void `setUserId` (int userId)

7.4.1. Descripción detallada

Clase que representa los datos de un sensor.

Esta clase se utiliza para almacenar la información de las mediciones de los sensores, incluyendo el tipo, el valor, la marca de tiempo y el ID del usuario asociado.

7.4.2. Documentación de constructores y destructores

7.4.2.1. `SensorData()`

```
com.example.usuario_upv.proyecto3a.SensorData.SensorData (
    String type,
    float value,
    int userId) [inline]
```

Constructor de la clase [SensorData](#).

Parámetros

<i>type</i>	Tipo de medición (ej. "CO2", "temperatura").
<i>value</i>	Valor de la medición.
<i>userId</i>	ID del usuario que realiza la medición.

7.4.3. Documentación de funciones miembro

7.4.3.1. `getTimestamp()`

```
String com.example.usuario_upv.proyecto3a.SensorData.getTimestamp () [inline]
```

Devuelve

Marca de tiempo de la medición.

7.4.3.2. `getType()`

```
String com.example.usuario_upv.proyecto3a.SensorData.getType () [inline]
```

Devuelve

Tipo de medición.

7.4.3.3. `getUserId()`

```
int com.example.usuario_upv.proyecto3a.SensorData.getUserId () [inline]
```

Devuelve

ID del usuario asociado a la medición.

7.4.3.4. `getValue()`

```
float com.example.usuario_upv.proyecto3a.SensorData.getValue () [inline]
```

Devuelve

Valor de la medición.

7.4.3.5. `setTimestamp()`

```
void com.example.usuario_upv.proyecto3a.SensorData.setTimestamp (  
    String timestamp) [inline]
```

Parámetros

<i>timestamp</i>	Marca de tiempo de la medición.
------------------	---------------------------------

7.4.3.6. `setType()`

```
void com.example.usuario_upv.proyecto3a.SensorData.setType (  
    String type) [inline]
```

Parámetros

<i>type</i>	Tipo de medición.
-------------	-------------------

7.4.3.7. `setUserId()`

```
void com.example.usuario_upv.proyecto3a.SensorData.setUserId (  
    int userId) [inline]
```

Parámetros

<i>userId</i>	ID del usuario asociado a la medición.
---------------	--

7.4.3.8. `setValue()`

```
void com.example.usuario_upv.proyecto3a.SensorData.setValue (  
    float value) [inline]
```

Parámetros

<i>value</i>	Valor de la medición.
--------------	-----------------------

La documentación de esta clase está generada del siguiente archivo:

- `app/src/main/java/com/example/usuario_upv/proyecto3a/SensorData.java`

7.5. Referencia de la clase

com.example.usuario_upv.proyecto3a.TramaIBeacon

Clase que representa la trama de un beacon iBeacon.

Métodos públicos

- `byte[] getPrefijo ()`
- `byte[] getUUID ()`
- `byte[] getMajor ()`
- `byte[] getMinor ()`
- `byte getTxPower ()`
- `byte[] getLosBytes ()`
- `byte[] getAdvFlags ()`
- `byte[] getAdvHeader ()`
- `byte[] getCompanyID ()`
- `byte getiBeaconType ()`
- `byte getiBeaconLength ()`
- `void setMajor (byte[] major)`
- `void setMinor (byte[] minor)`
- `TramaIBeacon (byte[] bytes)`

Constructor que crea una instancia de [TramaIBeacon](#).

7.5.1. Descripción detallada

Clase que representa la trama de un beacon iBeacon.

Esta clase se encarga de interpretar los datos de un beacon iBeacon a partir de un array de bytes. Extrae información relevante como el UUID, el major, el minor y el TxPower.

7.5.2. Documentación de constructores y destructores**7.5.2.1. TramaIBeacon()**

```
com.example.usuario_upv.proyecto3a.TramaIBeacon.TramaIBeacon (
    byte[] bytes) [inline]
```

Constructor que crea una instancia de [TramaIBeacon](#).

Parámetros

<i>bytes</i>	Array de bytes que contiene los datos del beacon.
--------------	---

7.5.3. Documentación de funciones miembro

7.5.3.1. getAdvFlags()

```
byte[] com.example.usuario_upv.proyecto3a.TramaIBeacon.getAdvFlags () [inline]
```

7.5.3.2. getAdvHeader()

```
byte[] com.example.usuario_upv.proyecto3a.TramaIBeacon.getAdvHeader () [inline]
```

7.5.3.3. getCompanyID()

```
byte[] com.example.usuario_upv.proyecto3a.TramaIBeacon.getCompanyID () [inline]
```

7.5.3.4. getiBeaconLength()

```
byte com.example.usuario_upv.proyecto3a.TramaIBeacon.getiBeaconLength () [inline]
```

7.5.3.5. getiBeaconType()

```
byte com.example.usuario_upv.proyecto3a.TramaIBeacon.getiBeaconType () [inline]
```

7.5.3.6. getLosBytes()

```
byte[] com.example.usuario_upv.proyecto3a.TramaIBeacon.getLosBytes () [inline]
```

7.5.3.7. getMajor()

```
byte[] com.example.usuario_upv.proyecto3a.TramaIBeacon.getMajor () [inline]
```

7.5.3.8. getMinor()

```
byte[] com.example.usuario_upv.proyecto3a.TramaIBeacon.getMinor () [inline]
```

7.5.3.9. getPrefijo()

```
byte[] com.example.usuario_upv.proyecto3a.TramaIBeacon.getPrefijo () [inline]
```

7.5.3.10. getTxPower()

```
byte com.example.usuario_upv.proyecto3a.TramaIBeacon.getTxPower () [inline]
```

7.5.3.11. getUUID()

```
byte[] com.example.usuario_upv.proyecto3a.TramaIBeacon.getUUID () [inline]
```

7.5.3.12. setMajor()

```
void com.example.usuario_upv.proyecto3a.TramaIBeacon.setMajor (  
    byte[] major) [inline]
```

7.5.3.13. setMinor()

```
void com.example.usuario_upv.proyecto3a.TramaIBeacon.setMinor (  
    byte[] minor) [inline]
```

La documentación de esta clase está generada del siguiente archivo:

- `app/src/main/java/com/example/usuario_upv/proyecto3a/TramaIBeacon.java`

7.6. Referencia de la clase `com.example.usuario_upv.proyecto3a.User`

Clase que representa un usuario.

Métodos públicos

- `User` (String username)
Constructor que crea una instancia de `User`.
- String `getUsername` ()
Obtiene el nombre de usuario.
- void `setUsername` (String username)
Establece el nombre de usuario.

7.6.1. Descripción detallada

Clase que representa un usuario.

Esta clase contiene información sobre un usuario, incluyendo su nombre de usuario.

7.6.2. Documentación de constructores y destructores

7.6.2.1. User()

```
com.example.usuario_upv.proyecto3a.User.User (  
    String username) [inline]
```

Constructor que crea una instancia de `User`.

Parámetros

<code>username</code>	Nombre de usuario.
-----------------------	--------------------

7.6.3. Documentación de funciones miembro**7.6.3.1. getUsername()**

```
String com.example.usuario_upv.proyecto3a.User.getUsername () [inline]
```

Obtiene el nombre de usuario.

Devuelve

Nombre de usuario.

7.6.3.2. setUsername()

```
void com.example.usuario_upv.proyecto3a.User.setUsername (  
    String username) [inline]
```

Establece el nombre de usuario.

Parámetros

<code>username</code>	Nombre de usuario a establecer.
-----------------------	---------------------------------

La documentación de esta clase está generada del siguiente archivo:

- `app/src/main/java/com/example/usuario_upv/proyecto3a/User.java`

**7.7. Referencia de la clase
`com.example.usuario_upv.proyecto3a.Utilidades`**

Clase utilitaria para conversiones entre diferentes tipos de datos.

Métodos públicos estáticos

- static byte[] [stringToBytes](#) (String texto)
Convierte una cadena a un arreglo de bytes.
- static UUID [stringToUUID](#) (String uuid)
Convierte una cadena de 16 caracteres a un UUID.
- static String [uuidToString](#) (UUID uuid)
Convierte un UUID a una cadena.
- static String [uuidToHexString](#) (UUID uuid)
Convierte un UUID a una cadena hexadecimal.
- static String [bytesToString](#) (byte[] bytes)
Convierte un arreglo de bytes a una cadena.
- static byte[] [dosLongToBytes](#) (long masSignificativos, long menosSignificativos)
Convierte dos longitudes a un arreglo de bytes.
- static int [bytesToInt](#) (byte[] bytes)
Convierte un arreglo de bytes a un entero.
- static long [bytesToLong](#) (byte[] bytes)
Convierte un arreglo de bytes a un long.
- static int [bytesToIntOK](#) (byte[] bytes)
Convierte un arreglo de bytes a un entero, manejando excepciones.
- static String [bytesToHexString](#) (byte[] bytes)
Convierte un arreglo de bytes a una cadena hexadecimal.

7.7.1. Descripción detallada

Clase utilitaria para conversiones entre diferentes tipos de datos.

Esta clase contiene métodos para convertir entre cadenas, UUIDs y arreglos de bytes, así como otros métodos de utilidad.

7.7.2. Documentación de funciones miembro

7.7.2.1. [bytesToHexString\(\)](#)

```
static String com.example.usuario_upv.proyecto3a.Utilidades.bytesToHexString (
    byte[] bytes) [inline], [static]
```

Convierte un arreglo de bytes a una cadena hexadecimal.

Parámetros

<i>bytes</i>	Arreglo de bytes a convertir.
--------------	-------------------------------

Devuelve

La representación hexadecimal del arreglo de bytes.

7.7.2.2. [bytesToInt\(\)](#)

```
static int com.example.usuario_upv.proyecto3a.Utilidades.bytesToInt (
    byte[] bytes) [inline], [static]
```

Convierte un arreglo de bytes a un entero.

Parámetros

<i>bytes</i>	Arreglo de bytes a convertir.
--------------	-------------------------------

Devuelve

El entero correspondiente.

7.7.2.3. bytesToIntOK()

```
static int com.example.usuario_upv.proyecto3a.Utilidades.bytesToIntOK (  
    byte[] bytes) [inline], [static]
```

Convierte un arreglo de bytes a un entero, manejando excepciones.

Parámetros

<i>bytes</i>	Arreglo de bytes a convertir.
--------------	-------------------------------

Devuelve

El entero correspondiente.

Excepciones

<i>Error</i>	Si el arreglo de bytes es nulo o tiene más de 4 bytes.
--------------	--

7.7.2.4. bytesToLong()

```
static long com.example.usuario_upv.proyecto3a.Utilidades.bytesToLong (  
    byte[] bytes) [inline], [static]
```

Convierte un arreglo de bytes a un long.

Parámetros

<i>bytes</i>	Arreglo de bytes a convertir.
--------------	-------------------------------

Devuelve

El long correspondiente.

7.7.2.5. bytesToString()

```
static String com.example.usuario_upv.proyecto3a.Utilidades.bytesToString (  
    byte[] bytes) [inline], [static]
```

Convierte un arreglo de bytes a una cadena.

Parámetros

<i>bytes</i>	Arreglo de bytes a convertir.
--------------	-------------------------------

Devuelve

La cadena correspondiente al arreglo de bytes.

7.7.2.6. dosLongToBytes()

```
static byte[] com.example.usuario_upv.proyecto3a.Utilidades.dosLongToBytes (  
    long masSignificativos,  
    long menosSignificativos) [inline], [static]
```

Convierte dos longitudes a un arreglo de bytes.

Parámetros

<i>masSignificativos</i>	Longitud más significativa.
<i>menosSignificativos</i>	Longitud menos significativa.

Devuelve

Arreglo de bytes que representa las dos longitudes.

7.7.2.7. stringToBytes()

```
static byte[] com.example.usuario_upv.proyecto3a.Utilidades.stringToBytes (  
    String texto) [inline], [static]
```

Convierte una cadena a un arreglo de bytes.

Parámetros

<i>texto</i>	Cadena a convertir.
--------------	---------------------

Devuelve

Arreglo de bytes correspondiente a la cadena.

7.7.2.8. stringToUUID()

```
static UUID com.example.usuario_upv.proyecto3a.Utilidades.stringToUUID (  
    String uuid) [inline], [static]
```

Convierte una cadena de 16 caracteres a un UUID.

Parámetros

<i>uuid</i>	Cadena de 16 caracteres que representa un UUID.
-------------	---

Devuelve

El UUID correspondiente.

Excepciones

<i>Error</i>	Si la cadena no tiene 16 caracteres.
--------------	--------------------------------------

7.7.2.9. uuidToHexString()

```
static String com.example.usuario_upv.proyecto3a.Utilidades.uuidToHexString (  
    UUID uuid) [inline], [static]
```

Convierte un UUID a una cadena hexadecimal.

Parámetros

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

Devuelve

La representación hexadecimal del UUID.

7.7.2.10. uuidToString()

```
static String com.example.usuario_upv.proyecto3a.Utilidades.uuidToString (  
    UUID uuid) [inline], [static]
```

Convierte un UUID a una cadena.

Parámetros

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

Devuelve

La representación de cadena del UUID.

La documentación de esta clase está generada del siguiente archivo:

- app/src/main/java/com/example/usuario_upv/proyecto3a/[Utilidades.java](#)

Capítulo 8

Documentación de archivos

8.1. Referencia del archivo `app/src/main/java/com/example/usuario_upv/proyecto3a/MainActivity.java`

```
import android.Manifest;
```

Clases

- class `com.example.usuario_upv.proyecto3a.MainActivity`
MainActivity de la aplicación.

Paquetes

- package `com.example.usuario_upv.proyecto3a`

8.2. Referencia del archivo `app/src/main/java/com/example/usuario_upv/proyecto3a/RetrofitClient.java`

```
import retrofit2.Retrofit;
```

Clases

- class `com.example.usuario_upv.proyecto3a.RetrofitClient`
Clase para gestionar la instancia de Retrofit.

Paquetes

- package `com.example.usuario_upv.proyecto3a`

8.3. Referencia del archivo `app/src/main/java/com/example/usuario_upv/proyecto3a/SensorApi.java`

```
import retrofit2.Call;
```

Clases

- interface `com.example.usuario_upv.proyecto3a.SensorApi`
Interfaz para definir los endpoints de la API de sensores.

Paquetes

- package `com.example.usuario_upv.proyecto3a`

8.4. Referencia del archivo `app/src/main/java/com/example/usuario_upv/proyecto3a/SensorData.java`

```
import java.text.SimpleDateFormat;
```

Clases

- class `com.example.usuario_upv.proyecto3a.SensorData`
Clase que representa los datos de un sensor.

Paquetes

- package `com.example.usuario_upv.proyecto3a`

8.5. Referencia del archivo `app/src/main/java/com/example/usuario_upv/proyecto3a/TramalBeacon.java`

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

Clases

- class `com.example.usuario_upv.proyecto3a.TramalBeacon`
Clase que representa la trama de un beacon iBeacon.

Paquetes

- package `com.example.usuario_upv.proyecto3a`

8.6. Referencia del archivo app/src/main/java/com/example/usuario_upv/proyecto3a/User.java

Clases

- class [com.example.usuario_upv.proyecto3a.User](#)
Clase que representa un usuario.

Paquetes

- package [com.example.usuario_upv.proyecto3a](#)

8.7. Referencia del archivo app/src/main/java/com/example/usuario_upv/proyecto3a/Utilidades.java

```
import java.math.BigInteger;
```

Clases

- class [com.example.usuario_upv.proyecto3a.Utilidades](#)
Clase utilitaria para conversiones entre diferentes tipos de datos.

Paquetes

- package [com.example.usuario_upv.proyecto3a](#)

8.8. Referencia del archivo README.md

Índice alfabético

actualizarIP
 com.example.usuario_upv.proyecto3a.MainActivity, 14

app/src/main/java/com/example/usuario_upv/proyecto3a/MainActivity.java, 31

app/src/main/java/com/example/usuario_upv/proyecto3a/RetrofitClient.java, 31

app/src/main/java/com/example/usuario_upv/proyecto3a/SensorApi.java, 32

app/src/main/java/com/example/usuario_upv/proyecto3a/SensorData.java, 32

app/src/main/java/com/example/usuario_upv/proyecto3a/TramaBeacon.java, 32

app/src/main/java/com/example/usuario_upv/proyecto3a/User.java, 33

app/src/main/java/com/example/usuario_upv/proyecto3a/Utilidades.java, 33

botonBuscarDispositivosBTLEPulsado
 com.example.usuario_upv.proyecto3a.MainActivity, 14

botonBuscarNuestroDispositivoBTLEPulsado
 com.example.usuario_upv.proyecto3a.MainActivity, 14

botonDetenerBusquedaDispositivosBTLEPulsado
 com.example.usuario_upv.proyecto3a.MainActivity, 15

bytesToHexString
 com.example.usuario_upv.proyecto3a.Utilidades, 26

bytesToInt
 com.example.usuario_upv.proyecto3a.Utilidades, 26

bytesToIntOK
 com.example.usuario_upv.proyecto3a.Utilidades, 27

bytesToLong
 com.example.usuario_upv.proyecto3a.Utilidades, 27

bytesToString
 com.example.usuario_upv.proyecto3a.Utilidades, 27

checkConnection
 com.example.usuario_upv.proyecto3a.SensorApi, 17

com.example.usuario_upv.proyecto3a, 11

com.example.usuario_upv.proyecto3a.MainActivity, 13

 actualizarIP, 14

 botonBuscarDispositivosBTLEPulsado, 14

 botonBuscarNuestroDispositivoBTLEPulsado, 14

 botonDetenerBusquedaDispositivosBTLEPulsado, 15

 onCreate, 15

 onRequestPermissionsResult, 15

 RetrofitClient, 15

 com.example.usuario_upv.proyecto3a.RetrofitClient, 16

 SensorApi, 16

 com.example.usuario_upv.proyecto3a.SensorApi, 17

 SensorData, 17

 createSensorData, 17

 TramaBeacon, 18

 deleteUser, 18

 resetTables, 18

 setupTables, 19

 Utilidades, 19

 com.example.usuario_upv.proyecto3a.SensorData, 19

 getTimestamp, 20

 getType, 20

 getUserId, 20

 getValue, 21

 SensorData, 20

 setTimestamp, 21

 setType, 21

 setUserId, 21

 setValue, 21

 com.example.usuario_upv.proyecto3a.TramaBeacon, 22

 getAdvFlags, 23

 getAdvHeader, 23

 getCompanyID, 23

 getBeaconLength, 23

 getBeaconType, 23

 getLosBytes, 23

 getMajor, 23

 getMinor, 23

 getPrefijo, 23

 getTxPower, 23

 getUUID, 24

 setMajor, 24

 setMinor, 24

 TramaBeacon, 22

 com.example.usuario_upv.proyecto3a.User, 24

 getUsername, 25

 setUsername, 25

 User, 24

 com.example.usuario_upv.proyecto3a.Utilidades, 25

 bytesToHexString, 26

 bytesToInt, 26

 bytesToIntOK, 27

bytesToLong, [27](#)
 bytesToString, [27](#)
 dosLongToBytes, [28](#)
 stringToBytes, [28](#)
 stringToUUID, [28](#)
 uuidToHexString, [29](#)
 uuidToString, [29](#)
 createSensorData
 com.example.usuario_upv.proyecto3a.SensorApi,
 [17](#)
 createUser
 com.example.usuario_upv.proyecto3a.SensorApi,
 [18](#)
 deleteUser
 com.example.usuario_upv.proyecto3a.SensorApi,
 [18](#)
 dosLongToBytes
 com.example.usuario_upv.proyecto3a.Utilidades,
 [28](#)
 getAdvFlags
 com.example.usuario_upv.proyecto3a.TramalBeacon,
 [23](#)
 getAdvHeader
 com.example.usuario_upv.proyecto3a.TramalBeacon,
 [23](#)
 getClient
 com.example.usuario_upv.proyecto3a.RetrofitClient,
 [16](#)
 getCompanyID
 com.example.usuario_upv.proyecto3a.TramalBeacon,
 [23](#)
 getiBeaconLength
 com.example.usuario_upv.proyecto3a.TramalBeacon,
 [23](#)
 getiBeaconType
 com.example.usuario_upv.proyecto3a.TramalBeacon,
 [23](#)
 getLosBytes
 com.example.usuario_upv.proyecto3a.TramalBeacon,
 [23](#)
 getMajor
 com.example.usuario_upv.proyecto3a.TramalBeacon,
 [23](#)
 getMinor
 com.example.usuario_upv.proyecto3a.TramalBeacon,
 [23](#)
 getPrefijo
 com.example.usuario_upv.proyecto3a.TramalBeacon,
 [23](#)
 getTimestamp
 com.example.usuario_upv.proyecto3a.SensorData,
 [20](#)
 getTxPower
 com.example.usuario_upv.proyecto3a.TramalBeacon,
 [23](#)
 getType
 com.example.usuario_upv.proyecto3a.SensorData,
 [20](#)
 getUserId
 com.example.usuario_upv.proyecto3a.SensorData,
 [20](#)
 getUsername
 com.example.usuario_upv.proyecto3a.User, [25](#)
 getUUID
 com.example.usuario_upv.proyecto3a.TramalBeacon,
 [24](#)
 getValue
 com.example.usuario_upv.proyecto3a.SensorData,
 [21](#)
 onCreate
 com.example.usuario_upv.proyecto3a.MainActivity,
 [15](#)
 onRequestPermissionsResult
 com.example.usuario_upv.proyecto3a.MainActivity,
 [15](#)
 procesarBeacon
 com.example.usuario_upv.proyecto3a.MainActivity,
 [15](#)
 Proyecto3A_Android, [1](#)
 README.md, [33](#)
 resetTables
 com.example.usuario_upv.proyecto3a.SensorApi,
 [18](#)
 SensorData
 com.example.usuario_upv.proyecto3a.SensorData,
 [20](#)
 setMajor
 com.example.usuario_upv.proyecto3a.TramalBeacon,
 [24](#)
 setMinor
 com.example.usuario_upv.proyecto3a.TramalBeacon,
 [24](#)
 setTimestamp
 com.example.usuario_upv.proyecto3a.SensorData,
 [21](#)
 setType
 com.example.usuario_upv.proyecto3a.SensorData,
 [21](#)
 setupTables
 com.example.usuario_upv.proyecto3a.SensorApi,
 [19](#)
 setUserId
 com.example.usuario_upv.proyecto3a.SensorData,
 [21](#)
 setUsername
 com.example.usuario_upv.proyecto3a.User, [25](#)
 setValue
 com.example.usuario_upv.proyecto3a.SensorData,
 [21](#)
 stringToBytes

com.example.usuario_upv.proyecto3a.Utilidades,
28

stringToUUID
com.example.usuario_upv.proyecto3a.Utilidades,
28

TramaIBeacon
com.example.usuario_upv.proyecto3a.TramaIBeacon,
22

User
com.example.usuario_upv.proyecto3a.User, 24

uuidToHexString
com.example.usuario_upv.proyecto3a.Utilidades,
29

uuidToString
com.example.usuario_upv.proyecto3a.Utilidades,
29