ServicioEscuchaBeacons

```
=>
constructor()
```

```
=>
parar()
```

```
onDestroy()
```

```
Intent : intent
            =>
onHandleIntent()
```

```
String : ETIQUETA_LOG
N : tiempoDeEspera
T/F : seguir
BluetoothLeScanner : elEscanner;
ScanCallback : callbackDelEscaneo
LogicaFake : logicaFake
GestionNotificaciones: gestorNotificaciones
N: idNotificacion
Intent: elIntentDelServicio
```

```
buscarTodosLosDispositivosBTLE()
```

```
            =>
detenerBusquedaDispositivosBTLE()
```

```
N: id
<=
quitarNotificacion()
```

```
ScanResult : resultado
=>
mostrarInformacionDispositivoBTLE()
```

```
String : dispositivoBuscado
=>
buscarEsteDispositivoBTLE()
                        <=
```

```
inicializarBluetooth()
                    <=
```

CheckUrl

```
Context : mContext
=>          =>
constructor()
<=          <=
String
```

```
String : strUrl
=>
downloadUrl()
<=          <=
String
```

```
onPostExecute()
```

```
Context : mContext
String : directionMode
```

```
[ String ] : strings
=>
doInBackground()
<=              <=
String
```

```
String : strUrl
=>
postUrl()
<=      <=
String
```

TramaBeacon

getPrefijo()
<=   <=
[byte]

getUUID()
<= <=
[byte]

getMajor()
<= <=
[byte]

getMinor()
<= <=
[byte]

getCompanyID()
<= <=
[byte]

getiBeaconLength()
<= <=
[byte]

[byte] : bytes
=>   =>
constructor()
<= <=
[byte]

[byte] : prefijo
[byte] : uuid
[byte] : major
[byte] : minor
byte : txPower
[byte] : losBytes
[byte] : advFlags
[byte] : advHeader
[byte] : companyID
byte : iBeaconType
byte : iBeaconLength

getTxPower()
<= <=
[byte]

getLosBytes()
<= <=
[byte]

getAdvFlags()
<= <=
[byte]

getAdvHeader()
<= <=
[byte]

getiBeaconType()
<= <=
[byte]

Utilidades

```
String : texto
=>            =>
stringToBytes()
<=
[byte]
```

```
UUID : uuid
=>            =>
stringToUUID()
<=
UUID
```

```
UUID: uuid
=>            =>
uuidToString()
<=
String
```

```
UUID: uuid
=>            =>
uuidToHexString()
<=
String
```

```
[byte] : bytes
=>            =>
bytesToString()
<=
String
```

```
R : masSignificativos
R : menosSignificativos
=>            =>
dosLongToBytes()
<=
[byte]
```

```
[byte] : bytes
=>            =>
bytesToInt()
<=
N
```

```
[byte] : bytes
=>            =>
bytesToLong()
<=
R
```

```
[byte] : bytes
=>            =>
bytesToIntOK()
<=
N
```

```
[byte] : bytes
=>            =>
bytesToHexString()
<=
String
```

```
                        LogicaFake

N : cuantas                              medida : Medida
=>                                                    <=
ObtenerUltimasMediciones()               InsertarMedicion()
<=                                                    =>
[ Medida ]

                          Text : ip

ObtenerTodasLasMediciones()
<=
[ Medida ]
```

MainActivity

```
Bundle : savedInstanceState
=>        =>
onCreate()
        <=
```

```
Text : ETIQUETA_LOG
Context : mContext
LogicaFake : logicaFake
N : CODIGO_PETICION_PERMISOS
TextView : t
GestionNotificaciones: gestorNotificaciones
Context: mContext
```

```
View : view
=>        =>
recibir()
        <=
```

```
View : view
=>        =>
cambiarBeacons()
        <=
```

```
          =>
recibir()
        <=
```

```
View : view
=>        =>
recibir()
        <=
```

```
getContext()
=>Context
```

BeaconActivity

```
Texto : ETIQUETA_LOG
Intent : elIntentDelServicio
Context : mContext
Activity : mActivity
EditText : e
TextView : t
ServicioEscuharBeacons : servicio
String : latitud
String : longitut
BeaconsActivity : activity
LocationManager : lm
boolean : activado
```

```
View : v
=>                                =>
botonDetenerBusquedaDispositivosBTLEPulsado()
                                         <=
```

```
Bundle : savedInstanceState
=>          =>
onCreate()
            <=
```

```
              =>
getLatitud()
<=          <=
String
```

```
              =>
getLongitut()
<=          <=
String
```

```
View : v
=>                                =>
botonBuscarDispositivosBTLEPulsado()
                                  <=
```

```
View : v
=>                                =>
botonBuscarNuestroDispositivoBTLEPulsado()
                                      <=
```

```
View : view
=>          =>
encender()
            <=
```

```
View : view
=>          =>
apagar ()
            <=
```

```
Location : location
=>          =>
onLocationChanged()
          <=
```

```
                          GestionNotificaciones
                                                            Intent: intent, Texto: titulo, Texto: descripcion
                                                            <=
 GestionNotificaciones()=>    NotificacionManagerCompat: notificationManager    crearNotificacionServicio()
                                                            =>
                                                            N
                              Texto: CHANNEL_ID

                                                            N: id
                                                            <=
                                                            quitarNotificacion()
 createNotificationChannel()


 N: color, Texto: titulo, Texto: descripcion    N: idNotificacion, Intent: intent, Texto: titulo, Texto: descripcion
 =>                                             <=
 crearNotificacionAlertas()                     actualizarNotificacoinServicio()
```