



Universidad del Azuay

Facultad de Ciencias de la Administración

Escuela de Ingeniería en Sistemas y Telemática

**PROTOTIPO DE APLICACIÓN MOVIL DE  
LOCALIZACIÓN EN INTERIORES**

Autor:

**Marco Antonio Quezada Cabrera**

Director:

**Ing. Chester Andrew Sellers Walden m. sC.**

**Cuenca - Ecuador**

**2018**

## **DEDICATORIA**

Dedico esta tesis a mi familia y a todas las personas cercanas que me han apoyado de forma incondicional durante estos cinco años para poder culminar exitosamente mis estudios.

## **AGRADECIMIENTO**

Quiero agradecer principalmente a Dios, después a mi familia que con su apoyo me han ayudado a alcanzar esta meta.


También a los ingenieros Chester Seller y Lenin Erazo por haberme prestado su guía durante el desarrollo de esta tesis.

## ABSTRACT:

Computer systems have evolved from simple scientific calculators to the point of providing information based on the environment that surrounds the user, taking as a reference a user profile and his or her current location. In this way, an indoor location system is essential to generate context sensitive systems. Having a mobile application prototype, through which it is possible to estimate the location of a user by means of the owner's Smartphone, will serve as a basis for future development, in which the application can be integrated with other applications that require to know the location of the user in indoor environments, as part of its operation.



UNIVERSIDAD DEL  
AZUAY  
Dpto. Idiomas



Translated by,  
Lic. Lourdes Crespo

## **RESUMEN:**

Los sistemas informáticos han evolucionado desde simples calculadoras científicas hasta llegar a proporcionar información en base al entorno que rodea al usuario, teniendo como referencia un perfil de usuario y la ubicación actual del mismo. De esta manera un sistema de localización en interiores es indispensable para poder generar sistemas que sean sensibles al contexto.

El poder contar con un prototipo de aplicación móvil, mediante la cual se pueda llegar a estimar la ubicación de un usuario por medio de su *Smartphone*, servirá como base para un futuro desarrollo, en el cual se la pueda integrar con otras aplicaciones que requieran como parte de su funcionamiento conocer la localización del usuario en ambientes de interiores.

## ÍNDICE

DEDICATORIA .....	II
AGRADECIMIENTO .....	III
ABSTRACT:.....	IV
RESUMEN: .....	V
ÍNDICE.....	VI
Índice de Figuras.....	X
Índice de Tablas .....	XII
Índice de Ecuaciones.....	XIII
Introducción:.....	1
Capítulo 1.....	2
1. Estado del Arte.....	2
1.1. Introducción .....	2
1.2. Localización en Exteriores.....	2
1.3. Localización en Interiores.....	2
1.4. Generalidades de la Localización en Interiores .....	3
1.4.1. Posicionamiento Físico y Ubicación Simbólica.....	3
1.4.2. Localización Absoluta y Relativa .....	5
1.4.3. Precisión y Exactitud .....	6
1.4.4. Escala .....	6
1.4.5. Arquitecturas básicas de localización .....	7
1.5. Técnicas de Localización en Interiores .....	8
1.5.1. Triangulación .....	8
1.5.2. Técnicas basadas en Triangulación.....	9
1.5.2.1. Tiempo de Llegada (TOA, <i>Time of Arrival</i> ) .....	9
1.5.2.2. Diferencia de Tiempo de Llegada (TDOA, <i>Time Difference of Arrival</i> ).....	9
1.5.2.3. Método Basado en la Atenuación de la Señal.....	10
1.5.2.4. Tiempo de Retorno de la Señal (ROTF, <i>Return Time of Flight</i> ).....	10
1.5.2.5. Fase de la Señal Recibida ( <i>Received Signal Phase Method</i> ) .....	10
1.5.2.6. Ángulo de Llegada (AOA, <i>Angle of Arrival</i> ).....	10
1.5.3. <i>Fingerprint</i> (Huella Digital).....	12
1.5.3.1. Método kNN (k vecinos más próximos) .....	13
1.5.3.2. Redes Neuronales.....	13
1.5.3.3. <i>Smallest M-Vertex Polygon</i> .....	13

1.5.3.4.	Vector de Potencia .....	13
1.5.4.	Técnicas de Proximidad.....	14
1.5.5.	Heurística de Movimiento.....	14
1.6.	Tecnologías para Localización en Interiores.....	14
1.6.1.	Sistema de Posicionamiento Global (GPS).....	15
1.6.2.	WiFi .....	15
1.6.2.1.	Localización en interiores por WiFi mediante <i>fingerprint</i> .....	16
1.6.3.	Sistema Global de Comunicaciones Móviles (GSM) .....	17
1.6.3.1.	Localización en Interiores basado en tecnología GSM.....	17
1.6.4.	Bluetooth.....	19
1.6.4.1.	Localización en Interiores basada en <i>iBeacons</i> .....	19
1.6.4.2.	Bluetooth y Redes Neuronales .....	20
1.6.5.	Campo Magnético .....	21
1.6.6.	Localización en interiores mediante tecnología VLC.....	22
1.6.7.	<i>ZigBee</i> .....	23
1.6.7.1.	Localización de Interiores basada en tecnología <i>ZigBee</i> .....	23
1.6.8.	Comparación entre Bluetooth <i>Low Energy</i> (BTLE) y WiFi. ....	24
1.7.	Productos comerciales de localización en interiores.....	27
1.7.1.	Situm.....	27
1.7.2.	Indoor Atlas .....	29
1.7.3.	Google Maps en Interiores.....	30
1.7.4.	Ekahau.....	30
1.7.5.	<i>Estimote Beacons</i> .....	30
1.8.	Conclusiones .....	31
	Capítulo 2.....	32
2.	Tecnología WiFi .....	32
2.1.	Introducción .....	32
2.2.	Infraestructura de Localización.....	32
2.3.	Método <i>Fingerprint</i> .....	33
2.4.	Algoritmo kNN ( <i>k-nearest neighbors</i> ).....	35
2.5.	Levantamiento de Requerimientos de Software .....	37
2.5.1.	Introducción .....	37
2.5.1.1.	Propósito .....	37
2.5.1.2.	Ámbito del Sistema.....	37
2.5.1.3.	Definiciones, Acrónimos y Abreviaturas .....	38
2.5.1.4.	Referencias.....	38

2.5.1.5.	Visión general del documento.....	38
2.5.2.	Descripción General.....	38
2.5.2.1.	Perspectiva del Producto.....	38
2.5.2.2.	Funciones del Producto.....	39
2.5.2.3.	Características de los Usuarios .....	39
2.5.2.4.	Restricciones .....	39
2.5.2.5.	Suposiciones y Dependencias .....	39
2.5.3.	Requisitos Específicos .....	40
2.5.3.1.	Interfaces de Usuario .....	40
2.5.3.2.	Interfaces de Hardware .....	40
2.5.3.3.	Interfaces de Software.....	40
2.5.3.4.	Interfaces de Comunicación.....	40
2.5.4.	Requisitos Funcionales .....	40
2.5.5.	Requisitos no Funcionales .....	43
2.6.	Servidor de Aplicaciones .....	44
2.6.1.	Arquitectura .....	44
2.6.2.	Base de Datos.....	44
2.6.3.	Servicios Web .....	45
2.7.	Aplicación Móvil .....	46
2.7.1.	Diseño de la aplicación móvil.....	46
2.7.2.	<i>Login</i> .....	47
2.7.3.	Escaneo de RSSI.....	47
2.7.4.	Visualización de la localización en interiores.....	48
2.7.5.	Configuraciones .....	49
2.8.	Conclusiones.....	50
	Capítulo 3.....	51
3.	Tecnología Bluetooth.....	51
3.1.	Introducción .....	51
3.2.	Bluetooth <i>Low Energy</i> (BLE) .....	51
3.3.	<i>Beacons</i> .....	52
3.3.1.	Descripción .....	52
3.3.2.	Especificaciones.....	54
3.3.3.	Configuración .....	55
3.4.	Infraestructura de Localización.....	57
3.5.	Método de Localización.....	59
3.5.1.	Cálculo de Distancia entre <i>Smartphone</i> y <i>Beacon</i> .....	60



3.5.2.	Trilateración.....	64
3.6.	Algoritmo de Localización.....	67
3.7.	Levantamiento de Requerimientos de Software .....	69
3.7.1.	Introducción .....	69
3.7.1.1.	Propósito .....	69
3.7.1.2.	Ámbito del Sistema.....	69
3.7.1.3.	Definiciones, Acrónimos y Abreviaturas .....	70
3.7.1.4.	Referencias.....	70
3.7.1.5.	Visión general del documento.....	70
3.7.2.	Descripción General.....	70
3.7.2.1.	Perspectiva del Producto .....	70
3.7.2.2.	Funciones del Producto.....	71
3.7.2.3.	Características de los Usuarios .....	71
3.7.2.4.	Restricciones .....	71
3.7.2.5.	Suposiciones y Dependencias .....	71
3.7.3.	Requisitos Específicos .....	72
3.7.3.1.	Interfaces de Usuario .....	72
3.7.3.2.	Interfaces de Hardware .....	72
3.7.3.3.	Interfaces de Software.....	72
3.7.3.4.	Interfaces de Comunicación.....	72
3.7.4.	Requisitos Funcionales .....	72
3.7.5.	Requisitos no Funcionales .....	75
3.8.	Servidor de Aplicaciones .....	76
3.8.1.	Arquitectura .....	76
3.8.2.	Base de Datos.....	76
3.8.3.	Servicios Web .....	77
3.9.	Aplicación Móvil .....	78
3.9.1.	Diseño de la aplicación móvil.....	78
3.9.2.	Base de Datos <i>SQLite</i> .....	80
3.9.3.	Módulo Bluetooth .....	80
3.9.4.	Localización y visualización.....	81
3.9.5.	Información contextual .....	83
3.10.	Conclusiones .....	85
Capítulo 4.....		86
4.	Resultados y Discusión .....	86
4.1.	Introducción .....	86

4.2. Resultados de la tecnología WiFi.....	86
4.3. Resultados de la tecnología Bluetooth.....	87
4.4. Matriz FODA de la tecnología WiFi.....	89
4.5. Matriz FODA de la tecnología Bluetooth.....	90
4.6. Comparación entre tecnologías.....	91
Conclusiones.....	94
Trabajos Futuros.....	94
Bibliografía.....	96

## Índice de Figuras

Figura 1: Ejemplificación de Posicionamiento Físico (Autoría Propia, 2017). .....	3
Figura 2: Ejemplificación de una Ubicación Simbólica (Autoría Propia, 2017). .....	4
Figura 3: TOA (Núñez, 2009) (tartigues, 2014) .....	9
Figura 4: Atenuación de la Señal (tartigues, 2014). .....	10
Figura 5: AOA (Días-Ambrona Tabernilla, 2008). .....	11
Figura 6: Grafico de Propagación de la señal (Zhao, Xiao, Markham, Trigoni, & Ren, 2014). .....	26
Figura 7: Trayectorias estimadas (Zhao, Xiao, Markham, Trigoni, & Ren, 2014). .....	26
Figura 8: App de Situm (SITUM, 2017). .....	28
Figura 9: App de IndoorAtlas (IndoorAtlas, 2017). .....	29
Figura 10: Location Beancos (Estimote, 2017) .....	31
Figura 11: Distribución de Laboratorios (Autoría Propia, 2017). .....	32
Figura 12: Distribución de puntos de acceso y sus MAC (Autoría Propia, 2017). .....	33
Figura 13: Puntos de Referencia (Autoría Propia, 2017). .....	34
Figura 14: Recepción de Valores RSSI (Autoría Propia, 2017). .....	35
Figura 15: Diagrama de Funcionamiento del algoritmo kNN (Autoría Propia, 2017). .....	36
Figura 16: Diagrama de Casos de Uso (Autoría Propia, 2017). .....	39
Figura 17: Arquitectura del sistema (Autoría Propia, 2017). .....	44
Figura 18: Modelo Entidad/Relación (Autoría Propia, 2017). .....	45
Figura 19: Estructura del Servicio Web para Login (Autoría Propia, 2017). .....	45
Figura 20: Estructura del servicio web para localización (Autoría Propia, 2017). .....	46
Figura 21: Diagrama de Clases (Autoría Propia, 2017). .....	46
Figura 22: Login (Autoría Propia, 2017). .....	47
Figura 23: Escaneo de puntos de acceso (Autoría Propia, 2017). .....	47
Figura 24: Laboratorio 5 (Autoría Propia, 2017). .....	48
Figura 25: Laboratorio 6 (Autoría Propia, 2017). .....	48
Figura 26: Laboratorio 4 (Autoría Propia, 2017). .....	48
Figura 27: Laboratorio 7 (Autoría Propia, 2017). .....	48
Figura 28: Pasillo (Autoría Propia, 2017). .....	49
Figura 29: Menú de Configuración (Autoría Propia, 2017). .....	49

Figura 30: Configuración de tiempo (Autoría Propia, 2017).....	49
Figura 31: BLE (areatecnologica, 2017).....	51
Figura 32: Beacon (NYVIDA, 2017).....	52
Figura 33: Rango de un Beacon (Cisco, 2017) .....	53
Figura 34: Hardware de un Beacon (NYVIDA, 2017) .....	54
Figura 35: App BeaconCFG (GooglePlay, 2017).....	55
Figura 36: Pantalla de Configuración (Autoría Propia, 2017). .....	55
Figura 37: Parámetros por defecto (Autoría Propia, 2017).....	55
Figura 38: Configuración de Nombre (Autoría Propia, 2017).....	56
Figura 39: Configuración de Serial ID (Autoría Propia, 2017). .....	56
Figura 40: Configuración de Broadcasting Interval (Autoría Propia, 2017).....	56
Figura 41: Reinicio de Beacon Autoría Propia, 2017).....	57
Figura 42: Beacons Configurados (Autoría Propia, 2017).....	57
Figura 43: Infraestructura (Autoría Propia, 2017). .....	57
Figura 44: Primer Escenario (Autoría Propia, 2017). .....	58
Figura 45: Segundo Escenario (Autoría Propia, 2017). .....	58
Figura 46: Distribución de Beacons del primer escenario (Autoría Propia, 2017). .....	59
Figura 47: Distribución de Beacons en escenario dos (Autoría Propia, 2017). .....	59
Figura 48: Resultados de Regresión (Autoría Propia, 2017). .....	62
Figura 49: Trilateración con Beacons (iArtHis_LAB, 2017).....	64
Figura 50: Plano de coordenadas de primer escenario (Autoría Propia, 2017).....	64
Figura 51: Plano de coordenadas de segundo escenario (Perez, 2013).....	65
Figura 52: Distancias reales entre Beacons del primer escenario (Autoría Propia, 2017). ....	66
Figura 53: Distancias reales entre Beacons del segundo escenario (Autoría Propia, 2017). .	67
Figura 54: Diagrama de Flujo del algoritmo del primer escenario (Autoría Propia, 2017). ..	68
Figura 55: Diagrama de Flujo del algoritmo del segundo escenario (Autoría Propia, 2017). 68	
Figura 56: Diagrama de Casos de Uso (Autoría Propia, 2017).....	71
Figura 57: Diagrama de la arquitectura del sistema (Autoría Propia, 2017).....	76
Figura 58: Modelo Entidad/Relación .....	77
Figura 59: Estructura de servicio web de localización (Autoría Propia, 2017). .....	78
Figura 60: Estructura de servicio web de contexto (Autoría Propia, 2017). .....	78
Figura 61: Diagrama de Clases (Autoría Propia, 2017). .....	79
Figura 62: Logo de SQLite (Filein, 2017). .....	80
Figura 63: Proceso de Localización (Autoría Propia, 2017).....	81
Figura 64: Proceso de Localización (Autoría Propia, 2017). .....	81
Figura 65: Localización parte izquierda de pasillo (Autoría Propia, 2017). .....	82
Figura 66: Laboratorio 4 (Autoría Propia, 2017).	
Figura 67: Laboratorio 5 (Autoría Propia, 2017). .....	82
Figura 68: Configuración de tiempo de actualización (Autoría Propia, 2017). .....	83
Figura 69: Mensaje de Error (Autoría Propia, 2017). .....	83
Figura 70: Información contextual del pasillo (Autoría Propia, 2017). .....	84
Figura 71: Información contextual de la clase (Autoría Propia, 2017).....	84
<i>Figura 72: Información contextual del laboratorio (Autoría Propia, 2017).....</i>	<i>84</i>
Figura 73: Resultados de tecnología WiFi (Autoría Propia, 2017).....	87
Figura 74: Pruebas de Precisión (Autoría Propia, 2017).....	88

## Índice de Tablas

Tabla 1: Resumen arquitecturas de localización en interiores (Autoría Propia, 2017).....	8
Tabla 2: Ventajas y Desventajas de las Técnicas de Triangulación (Barba, 2012).....	11
Tabla 3: Comparación características técnicas .....	25
Tabla 4: Parámetros derivados de la localización en interiores (Zhao, Xiao, Markham, Trigoni, & Ren, 2014).....	26
Tabla 5: Codificación de direcciones MAC (Autoría Propia, 2017).....	34
Tabla 6: Vectores de puntos de referencia (Autoría Propia, 2017).....	34
Tabla 7: Definiciones, Acrónimos y Abreviaturas (Autoría Propia, 2017). .....	38
Tabla 8: Referencias (Autoría Propia, 2017). .....	38
Tabla 9: Características de Usuario (Autoría Propia, 2017). .....	39
Tabla 10: Descripción de Caso de Uso - CDU01 (Autoría Propia, 2017). .....	41
Tabla 11: Descripción de Caso de Uso - CDU02 (Autoría Propia, 2017). .....	41
Tabla 12: Descripción de Caso de Uso - CDU03 (Autoría Propia, 2017). .....	41
Tabla 13: Descripción de Caso de Uso - CDU04 (Autoría Propia, 2017). .....	42
Tabla 14: Descripción de Caso de Uso - CDU05 (Autoría Propia, 2017). .....	42
Tabla 15: Descripción de Caso de Uso - CDU06 (Autoría Propia, 2017). .....	43
Tabla 16: Descripción de Requisito No Funcional - RNF01 (Autoría Propia, 2017).....	43
Tabla 17: Descripción de Requisito No Funcional - RNF02 (Autoría Propia, 2017).....	43
Tabla 18: Descripción de Requisito No Funcional - RNF03 (Autoría Propia, 2017).....	43
Tabla 19: Descripción de Requisito No Funcional - RNF04 (Autoría Propia, 2017).....	43
Tabla 20: Características de los Beacons (NYVIDA, 2017).....	53
Tabla 21: Especificaciones de Beacon (NYVIDA, 2017) .....	54
Tabla 22: Mediciones RSSI (Autoría Propia, 2017).....	61
Tabla 23: Cálculo de Ratio para cada RSSI.....	61
Tabla 24: Calculo de distancia aproximada con A y B (Autoría Propia, 2017).....	63
Tabla 25: Calculo de distancia aproximada con los tres coeficientes (Autoría Propia, 2017). .....	63
Tabla 26: Nomenclaturas de cada Beacon (Autoría Propia, 2017).....	66
Tabla 27: Coordenadas de Beacons (Autoría Propia, 2017).....	67
Tabla 28: Definiciones, Acrónimos y Abreviaturas (Autoría Propia, 2017). .....	70
Tabla 29: Referencias (Autoría Propia, 2017). .....	70
Tabla 30: Características de Usuario (Autoría Propia, 2017). .....	71
Tabla 31: Descripción de Caso de Uso - CDU01 (Autoría Propia, 2017). .....	73
Tabla 32: Descripción de Caso de Uso - CDU02 (Autoría Propia, 2017). .....	73
Tabla 33: Descripción de Caso de Uso - CDU03 (Autoría Propia, 2017). .....	74
Tabla 34: Descripción de Caso de Uso - CDU04 (Autoría Propia, 2017). .....	74
Tabla 35: Descripción de Caso de Uso - CDU05 (Autoría Propia, 2017). .....	74
Tabla 36: Descripción de Caso de Uso - CDU06 (Autoría Propia, 2017). .....	75
Tabla 37: Descripción de Requisito No Funcional - RNF01 (Autoría Propia, 2017).....	75
Tabla 38: Descripción de Requisito No Funcional - RNF02 (Autoría Propia, 2017).....	75
Tabla 39: Descripción de Requisito No Funcional - RNF03 (Autoría Propia, 2017).....	75
Tabla 40: Descripción de Requisito No Funcional - RNF04 (Autoría Propia, 2017).....	76
Tabla 41: Matriz FODA de tecnología WiFi (Autoría Propia, 2017). .....	89
Tabla 42: Matriz FODA de tecnología Bluetooth (Autoría Propia, 2017).....	90

## Índice de Ecuaciones

Ecuación 1: Distancia Euclidiana (Wikipedia, 2016) .....	35
Ecuación 2: Distancia en algoritmo kNN (Rios, 2012).....	35
Ecuación 3: Calculo de distancia en metros (Networks, 2017) .....	60
Ecuación 4: Cálculo de Ratio (Networks, 2017).....	61
Ecuación 5: Distancia calculada con A y B (Networks, 2017). .....	62
Ecuación 6: Diferencia de distancias de B y C (Perez, 2013).....	65
Ecuación 7: Diferencia de distancias de B y A (Perez, 2013). .....	65
Ecuación 8: Calculo de coordenada y (Perez, 2013).....	65
Ecuación 9: Calculo de coordenada x (Perez, 2013).....	65

## **Introducción:**

El presente documento describe el análisis e implementación de un sistema de localización, el cual consiste en revisar el estado de arte de las tecnologías que conjuntamente con un respectivo método y algoritmo permitan desarrollar un sistema de localización en interiores, el análisis está enfocado específicamente en las tecnologías WiFi y Bluetooth, para cada una de las tecnologías planteadas se desarrolló un prototipo de aplicación móvil en el sistema operativo *Android* con un servidor de aplicaciones. Este documento está constituido en cuatro capítulos que serán brevemente explicados a continuación.

Capítulo 1, llamado Estado de Arte; serán tratados los conceptos fundamentales sobre la localización en interiores, donde se detallará más a fondo las diferentes tecnologías con las cuales se puede llegar a realizar la localización, se especifican también los distintos métodos que pueden ser aplicados.

Capítulo 2, llamado Tecnología WiFi; se detalla la implementación del sistema de localización en interiores implementado mediante la tecnología WiFi, detallándose, el levantamiento de requerimientos, diseño del sistema y la implementación del mismo. También incluye la selección de la infraestructura de localización sobre la que están implementadas ambas tecnologías.

Capítulo 3, llamado Tecnología Bluetooth; se detalla la realización del sistema de localización en interiores implementado mediante la tecnología Bluetooth, detallándose, el levantamiento de requerimientos, diseño del sistema y la implementación del mismo.

Finalmente en el Capítulo 4, llamado Resultados y Discusión; están expuestos los resultados conseguidos luego de la realización de pruebas para las dos tecnologías, concluye con una comparación entre las dos tecnologías, considerando los puntos más importantes, los cuales permitan determinar cuál es la tecnología más apta para implementar un sistema de localización en interiores a una escala mayor.

# Capítulo 1

## 1. Estado del Arte

### 1.1. Introducción

El campo de la localización en ambientes de interiores ha tenido un gran crecimiento en los últimos años, este hecho se debe principalmente al incremento en el uso de varias tecnologías, como sensores, dispositivos que utilizan bluetooth, infrarrojo, radiofrecuencia, etc. Y aparte de todas estas tecnologías también el auge de los *Smartphone* ha ocasionado que la localización en interiores sea más fácil de lograr.

Debido a la existencia de numerosas alternativas de tecnologías y métodos que se han desarrollado dentro de esta modalidad de localización, se plantea como primer capítulo de este trabajo de titulación realizar una revisión de la literatura (estado del arte), donde se expondrán las modalidades de localización en interiores más relevantes.

### 1.2. Localización en Exteriores

La localización geográfica de cualquier objeto a lo largo de todo el globo terráqueo, es un área que ha crecido muy notablemente en los últimos tiempos sobre todo por el incremento de los servicios que se prestan mediante Internet.

Ha evolucionado de simplemente establecer una posición dentro de un plano bidimensional o tridimensional hasta llegar a convertirse en una estrategia de negocio, como actualmente lo es la gestión de rutas vehiculares, o las aplicaciones enfocadas en el entorno de contexto, donde se presenta al usuario diferentes ofertas de ciertos establecimientos al momento que el usuario se encuentra físicamente cerca del lugar.

### 1.3. Localización en Interiores

En la actualidad se encuentran múltiples aportes sobre las alternativas existentes para poder realizar localización en interiores, algunas de ellas, en su gran mayoría utilizan dispositivos hardware que necesitan estar instalados en la edificación para poder realizar

una estimación de su posición, otras en cambio no requieren tener instalado ningún hardware.

Existen métodos que utilizan dispositivos (*Smartphone, Laptop, Tablet, etc.*) que han emergido actualmente en el mercado, como también hay alternativas (radio frecuencia, infrarrojo, etc.) que continúan utilizando las tecnologías tradicionales pero con un distinto enfoque o con un mejoramiento de las técnicas propuestas anteriormente.

En este apartado se describirá algunas de las técnicas o métodos que se han investigado para poder tener un marco teórico de las posibles opciones a tomar en cuenta para llegar al desarrollo de un aplicativo propio adoptando uno de los métodos.

## 1.4. Generalidades de la Localización en Interiores

### 1.4.1. Posicionamiento Físico y Ubicación Simbólica

Dentro de un ámbito de interiores existen dos formas de poder realizar la localización. La primera es mediante un posicionamiento físico del objeto, tal y como funciona la tecnología GPS y la segunda manera es con una localización simbólica (Hightower & Borriello, 2001).

Una localización basada en posicionamiento físico utiliza valores de coordenadas mediante los cuales se indica la ubicación del objeto o usuario, llegando a establecer coordenadas X-Y dentro de un plano bidimensional, o tridimensional si se considera la elevación (Hightower & Borriello, 2001; Liu, Darabi, Banerjee, & Liu, 2007).

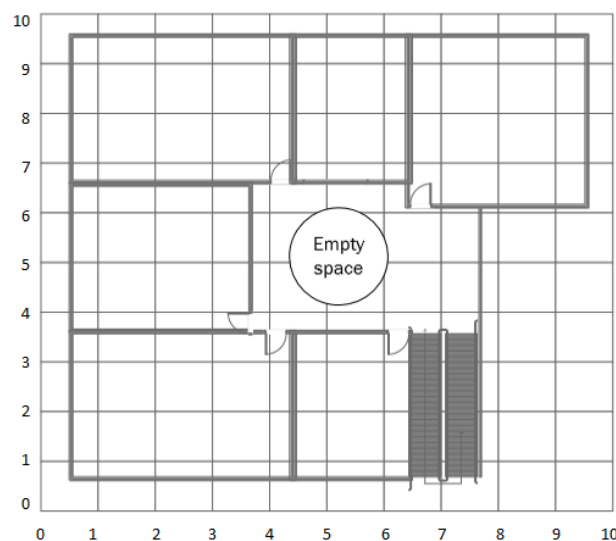


Figura 1: Ejemplificación de Posicionamiento Físico (Autoría Propia, 2017).



Una complicación que se da al realizar una representación física real se da al elegir un sistema de coordenadas. En una localización en exteriores como con el sistema GPS se utiliza latitud y longitud, estas referencias podrían usarse para ambientes de interiores, sin embargo el problema está que se manejaría con números muy pequeños, puesto que las distancias que se recorren muchas de las veces no superarían los 15 metros (Barba, 2012).

Al tratarse de una ubicación simbólica se brinda ideas abstractas de donde se encuentra el usuario, en un cuarto, sala, laboratorio de informática, aula de inglés, etc., expresando la ubicación mediante lenguaje natural (Liu, Darabi, Banerjee, & Liu, 2007). Un sistema que se base en posicionamiento físico puede llegar a proporcionar ubicación simbólica también, esto se consigue teniendo conocimiento de las coordenadas de cada salón o habitación del edificio (Hightower & Borriello, 2001).

Un sistema de ubicación simbólica sería suficiente si el objetivo del sistema de localización en interiores es de servir como guía, un servicio de guiado dentro de un edificio no requiere conocer la posición física real sino únicamente en que sala se halla y presentar una ruta hasta donde necesita llegar (Ruiz, 2012).

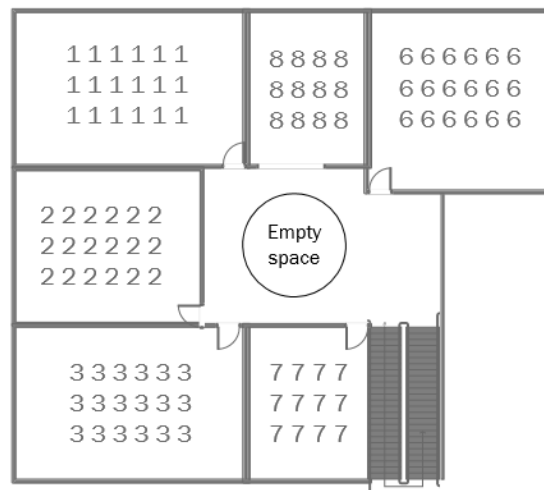


Figura 2: Ejemplificación de una Ubicación Simbólica (Autoría Propia, 2017).

Con una ubicación simbólica no se puede llegar a un posicionamiento físico, pero puede ser tomado como punto de partida, puesto que si se conoce ya la sub-área donde está el usuario se puede discriminar a las demás (Hightower & Borriello, 2001). Al realizarse inicialmente un posicionamiento físico y proseguir con una ubicación simbólica existe un gran riesgo de precisión, ya que muchos de los métodos que se utilizan para llegar a determinar un coordenada tienen asociado un error de aproximación, y si se considera un

error de 10 metros de precisión esto puede ocasionar una ubicación errónea, y la única forma de reducir este error es con más dispositivos de hardware instalados en la edificación.

#### **1.4.2. Localización Absoluta y Relativa**

La forma de ubicar a un objeto se puede realizar de manera absoluta o relativa. Al tratarse de una localización de manera absoluta el sistema reconoce una referencia para identificar la posición de todos los objetos, tal es el caso de sistema GPS que presenta la localización en base a longitud, latitud y elevación, si se hiciera una solicitud desde el mismo punto con dos dispositivos GPS el resultado sería el mismo en ambos (Hightower & Borriello, 2001).

Con una localización relativa se llega a estimar la ubicación en base al entorno que rodea al objeto, no se mantiene una referenciación en común para situar a todos los puntos (Ruiz, 2012). El ejemplo más común de esta forma de localizar es al realizarse por medio de proximidad a ciertos puntos conocidos, sin calcular una coordenada para después inferir el lugar, sino de acuerdo al punto de referencia que el dispositivo a localizar marque mayor intensidad (Liu, Darabi, Banerjee, & Liu, 2007).

Al igual que con los conceptos de posicionamiento físico y simbólico dentro de este tema también se puede llegar a integrar ambos métodos, trabajando con una localización absoluta se puede llegar a obtener una posición relativa, si se conocen las coordenadas del lugar en cuestión y de los que lo rodean se puede indicar que el salón B se halla en la segunda planta, pasillo izquierdo a lado de los salones A y C. Pero inclusive para llegar a tener un sistema que obtenga posiciones absolutas, este deberá estar basado en un modelo relativo y conocer de antemano las coordenadas de los puntos de referencia.

El proceso de triangulación es un claro ejemplo de que se parte de un modelo relativo, al momento de la ejecución de esta técnica el dispositivo a localizarse establece conexión con los puntos de referencia más cercanos, es decir con los que mayor intensidad de señal tenga contacto, al igual que en un sistema relativo, con la diferencia de que para una triangulación se conocen las coordenadas de las puntos de medición, y se obtiene una coordenada aproximada sobre donde está el objeto.

### **1.4.3. Precisión y Exactitud**

Estos dos conceptos son mencionados en múltiples ocasiones dentro de la computación, debido a que están ligados comúnmente al momento de trabajar con simulaciones, dichas simulaciones se enfocan para el análisis de eventos del mundo real, pero como se la realiza mediante una computadora la cual tiene un límite de memoria y procesamiento jamás se podrá obtener un resultado que sea 100% certero.

La localización de objetos tampoco está libre de estas limitaciones, ya sea en ambientes de exteriores o interiores, es decir que siempre se obtendrá una coordenada o una distancia aproximada a la que es realmente. Existen sistemas GPS más baratos los cuales dan un error de medición de 10 metros, lo que representaría a un 95% de aproximación del valor real y en los más sofisticados el error varía de 1 a 3 metros, esto representaría el 99% de aproximación (Barba, 2012).

La precisión dentro de los sistemas de localización viene representada por el porcentaje de aproximación o de confianza que este brinda, la precisión está latente para cualquier medición o estimación que se haga mediante esta tecnología, es decir que no varía respecto al punto que se desea localizar, siempre se contara con un 95%, 99%, 99.99%, etc. de precisión (Liu, Darabi, Banerjee, & Liu, 2007).

La exactitud por otro lado está representada por la distancia de error aproximada en la medición, representada por lo general en metros. Estos dos indicadores sirven para poder realizar una evaluación de cuanta precisión se necesita que el sistema brinde de acuerdo a la aplicación que se desarrollará, pudiera darse el caso que con una precisión del 95% y una exactitud de alrededor de 10 metros es suficiente para cumplir el objetivo de la aplicación, siempre se debe tener esto en cuenta para diseñar este tipo de sistemas, ya que una precisión más alta requiere un costo en infraestructura más alto también.

### **1.4.4. Escala**

Los sistemas de localización tienen la variante de cuantos objetos pueden ser localizados en un intervalo de tiempo, a esto se lo denomina escala de sistema de localización, la escala está determinada en varias ocasiones por la infraestructura que se tiene implementada (Hightower & Borriello, 2001).

Al tener un sistema de localización cuyo método de localización es a través de radiofrecuencia existe la limitante del número máximo de dispositivos que se pueden conectar a los equipos que brindan el servicio de transmisión, es necesario tener especificado cuantos dispositivos podrán hacer uso del servicio para no llegar a tener una saturación en la red, ya que una saturación conlleva a una disminución de la precisión de localización.

#### 1.4.5. Arquitecturas básicas de localización

En estudios realizados en años anteriores por Liu, Darabi, Banerjee, & Liu (2007) se menciona que existen cuatro topologías básicas que permiten realizar localización en interiores, dichas topologías son las siguientes:

- **Auto posicionamiento:** este método consiste en que todo el procesamiento para determinar la ubicación es realizado por el dispositivo a localizarse (*Smartphone* para este caso), el sistema contempla que únicamente el dispositivo mide la señal hacia los puntos de acceso y el propiamente evalúa el posicionamiento.
- **Posicionamiento remoto:** se incorpora un tercer elemento dentro de esta topología, el cual trata de una unidad de procesamiento remota, la forma de funcionamiento consiste en que este tercero recibe las señales tomadas por los puntos de referencia, procesa los datos y entrega como resultado la ubicación del *Smartphone*.
- **Auto posicionamiento indirecto:** las mediciones de señal las realizan los puntos de referencia, estos los envía al dispositivo a localizarse y el propio dispositivo móvil realiza el proceso de ubicación.
- **Posicionamiento remoto indirecto:** el *Smartphone* se encarga de medir las señales hacia los puntos de referencia, para posteriormente enviar estos datos hacia una tercera unidad para el procesamiento y establecimiento de la ubicación.

La siguiente tabla muestra un resumen de las cuatro opciones:

<b>Topología</b>	<b>Medición de Señal</b>	<b>Proceso de localización</b>
Auto posicionamiento	<i>Smartphone</i>	<i>Smartphone</i>
Posicionamiento remoto	Punto de referencia	Unidad de procesamiento
Auto posicionamiento indirecto	Punto de referencia	<i>Smartphone</i>
Posicionamiento remoto indirecto	<i>Smartphone</i>	Unidad de procesamiento

Tabla 1: Resumen arquitecturas de localización en interiores (Autoría Propia, 2017).

## 1.5. Técnicas de Localización en Interiores

Existen múltiples formas para llegar a determinar la localización de un objeto que se han propuesto y desarrollado a lo largo de los años, estas técnicas se pueden agrupar en dos grupos principales, los métodos o algoritmos que permiten realizar un ubicación basada en triangulación y los métodos o algoritmos que se basan en análisis del escenario de cobertura o llamado también *fingerprint*.

### 1.5.1. Triangulación

Dentro de la localización para poder llegar a tener un posicionamiento en coordenadas del objeto a ubicarse existen dos conceptos: la trilateración y triangulación.

En la trilateración para determinar la posición del objeto se miden las distancias desde el objeto hacia los puntos de referencia, necesitándose de al menos 3 de ellos (García Polo, 2008) y con la triangulación es similar, solamente en lugar de usar las distancias como referente se toman los ángulos (García Polo, 2008). Dentro de la investigación realizada por Chico Ciprián (2009) se generaliza el proceso de triangulación en base a potencia de señales, través de las siguientes cinco fases:

1. Con las potencias de tres puntos de referencia se crea un sistema de ecuaciones.
2. Se resuelve el sistema de ecuaciones llegándose a obtener los denominados puntos de triangulación.
3. Cada punto de triangulación se considera el vértice de un triángulo.

4. Se forman todos los triángulos posibles y se calculan sus áreas para una futura comparación.
5. La estimación de la ubicación se la realiza primero seleccionado el triángulo cuya área se la menor y el punto medio del triángulo es considerado como la ubicación del usuario.

### 1.5.2. Técnicas basadas en Triangulación

A continuación se presenta un recopilación de información tomada de (Barba, 2012; Ruiz, 2012; Liu, Darabi, Banerjee, & Liu, 2007; González Merino, 2013) donde se exponen las diversas técnicas de triangulación para realizar una localización en interiores.

#### 1.5.2.1. Tiempo de Llegada (TOA, *Time of Arrival*)

Es un método que utiliza la trilateración, se obtiene las medidas de distancia con respecto a los puntos de referencia midiendo el tiempo que tarda en llegar la señal hasta el receptor, formándose así un círculo alrededor de cada punto para después hallar la intersección entre los círculos y estimar la localización.

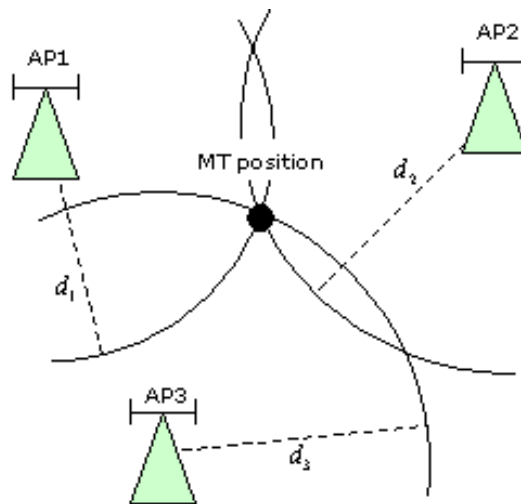


Figura 3: TOA (Núñez, 2009) (tartigues, 2014)

#### 1.5.2.2. Diferencia de Tiempo de Llegada (TDOA, *Time Difference of Arrival*)

La funcionalidad de este algoritmo es casi igual que en TOA, solamente cambia el sentido de la señal, ya que en este caso la distancia es tomada al momento de que la señal proveniente del receptor llega al *Smartphone*.

### 1.5.2.3. Método Basado en la Atenuación de la Señal

Método que está basado en la trilateración, utiliza la potencia de señal de cada punto de referencia y las pérdidas de propagación de cada señal, donde se determina la distancia entre los puntos, para conseguir tener un proceso similar como en TOA.



Figura 4: Atenuación de la Señal (tartigues, 2014).

### 1.5.2.4. Tiempo de Retorno de la Señal (ROTF, *Return Time of Flight*)

Este algoritmo emplea una forma de medición de distancia casi igual a la TOA, la diferencia recae en que mediante este método se analiza el tiempo que tarda la señal en llegar al receptor más el tiempo que tarda en retornar de vuelta al dispositivo a localizarse.

### 1.5.2.5. Fase de la Señal Recibida (*Received Signal Phase Method*)

Denominado también como *Phase of Arrival* (POA) es un método que usa las diferencias de fase que se presentan dentro de la señal, dentro de este método se considera que todas las señales son sinusoidales<sup>1</sup>, con la misma frecuencia. Para la recepción de las señales se necesita que cada una tenga un retraso, el cual consiste en un rango de desplazamiento determinado por la longitud de la onda, el cual debe ser evaluado para cada caso.

### 1.5.2.6. Ángulo de Llegada (AOA, *Angle of Arrival*)

El algoritmo propone que se puede conocer la ubicación de un objeto calculando la intersección de dos líneas prolongadas de sus ángulos, estas líneas representarían los

---

<sup>1</sup> Señal cuya grafica representa la señal de la función seno.

radios de circunferencias formadas alrededor de los puntos de referencia, para ello se necesita tener al menos dos puntos de referencia y obtener los ángulos de los mismos con respecto al *Smartphone*.

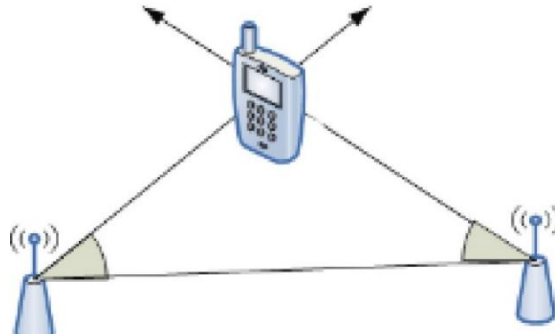


Figura 5: AOA (Días-Ambrona Tabernilla, 2008).

A continuación se presenta una tabla donde se establecen las principales ventajas y desventajas que Barba (2012) planteó:

<b>Técnica</b>	<b>Ventajas</b>	<b>Desventajas</b>
<b>TOA</b>	Algoritmo con carga computacional baja	Requiere buena sincronización de unidades de medida.
		Las mediciones de los tiempos se las envía en el paquete de datos
<b>TDOA</b>	Algoritmo con carga computacional baja	Requiere buena sincronización de unidades de medida.
		Las mediciones de los tiempos se las envía en el paquete de datos
<b>Atenuación de la Señal</b>	Algoritmo con carga computacional baja.	Se necesita de una línea de visión directa con el menor multitrayecto.
<b>ROTF</b>	Algoritmo con carga computacional baja.	Utilizar métodos precisos de sincronización.
		La medición del tiempo depende del modelo de cada <i>Smartphone</i> .
<b>POA</b>	Algoritmo con carga computacional baja.	Se necesita de una línea de visión directa con el menor multitrayecto.
		Limitación en el desplazamiento de fase.
<b>AOA</b>	Solamente se necesita dos puntos de referencia para un posicionamiento 2D	La estimación se reduce al presentarse un alejamiento de los puntos de referencia
		Infraestructura de hardware compleja

Tabla 2: Ventajas y Desventajas de las Técnicas de Triangulación (Barba, 2012).



### 1.5.3. *Fingerprint* (Huella Digital)

La técnica de *fingerprint* está orientada para una localización de posicionamiento físico como también para una simbólica, partiendo del hecho de que la propagación de la señal es diferente en cada salón dentro del edificio. Pudiendo llegar a conocer en donde se halla el usuario realizando una comparación entre los valores leídos y los tomados como referencia (Núñez, 2009).

RSS, por su abreviatura en inglés *Received Signal Strength*, o en español Intensidad de la Señal Recibida. Es un término muy común dentro de las redes de radiofrecuencia y se refiere a la medición de la potencia de una señal de radio. Esta medición está representada en dB (decibelios), cuyos valores oscilan entre los 0dB (muy buena señal) y los -110dB (señal pobre) (Inc., 2015).

El valor RSS resultante dependerá de varios factores, siendo el más impórtate la distancia que existe entre el dispositivo y los puntos de acceso, aunque es el más influyente no es el único, puesto que también está determinada por la calidad de los dispositivos móviles y puntos de acceso, la cantidad de paredes dentro del edificio, existencia de objetos compuestos de materiales que causan interferencia y el número de usuarios que estén presentes en la red (Inc., 2015)

Un *fingerprint* está compuesto por un vector que almacena generalmente los valores RSS de los puntos de referencia que puede escuchar, por lo que el tamaño del vector que se maneje está directamente ligado al número de puntos de referencia o acceso que se coloquen en la infraestructura (Núñez, 2009).

El método en general se descompone en dos fases: de calibración (*off-line*) y de localización (*online*). En la fase de calibración se toman muestras de la señal en diversos puntos dentro del área que se utilizara, almacenando la *Media Access Control* (MAC) del dispositivo de referencia y el valor RSS medido en dB. Para la segundo fase que es la de localización propiamente, existen varios algoritmos propios de *fingerprint* que se pueden emplear para llegar a la estimación de la posición del usuario.

A continuación se detallarán algunos de los algoritmos que más se utilizan para llegar a realizar una ubicación mediante *fingerprint*.

### **1.5.3.1. Método kNN (k vecinos más próximos)**

kNN (*k-Nearest-Neighbor*) o bien método de los vecinos más cercanos en español, dentro de este método se hace uso de un valor k el cual permite determinar los vecinos que se encuentren más cerca del *Smartphone* y seleccionar el vector de potencias RSS para poder realizar una evaluación de los valores (Barba, 2012; Navarro, Peuker, & Quan, 2010).

### **1.5.3.2. Redes Neuronales**

Los valores que se generan dentro de la fase de calibración sirven como entradas y salidas del sistema de redes neuronales, para continuar con una generación de pesos para cada medición, dentro de este sistema se maneja un árbol de decisión que se encarga de estimar la localización durante la segunda fase (Barba, 2012).

### **1.5.3.3. *Smallest M-Vertex Polygon***

Las mediciones que se toman a través de los puntos de referencia se las analiza por separado. Dentro de este algoritmo se considera un M número de equipos de conexión y cada uno de ellos posee una posición candidata, cada posición candidata es considerada como el vértice dentro de un polígono de las M unidades y para poder conocer la localización del usuario se toma como referencia el polígono con menor perímetro, y se procede a realizar un promedio de las posiciones candidatas (González Merino, 2013).

### **1.5.3.4. Vector de Potencia**

Esta técnica está descrita por Chico Ciprián (2009) está comprendida en tres fases:

1. Recolectar información sobre los puntos de acceso (AP) o puntos de referencia.
2. Construir una base de datos donde se almacenen las potencias tomadas desde cada punto de referencia.
3. Para la localización generar un vector con las potencias de cada AP al cual tenga acceso para después compararlo con los almacenados y encontrar el más parecido.

#### **1.5.4. Técnicas de Proximidad**

Las técnicas de proximidad son fáciles de comprender e implementar ya que se centran exclusivamente en una localización simbólica y relativa, no engloba algoritmos con diferentes mecanismos como en los métodos anteriores puesto que es el mismo y lo único que cambia es la tecnología con la que se implementa.

Las técnicas de proximidad poseen una única forma de ubicación, esta consiste en establecer la localización en base a la baliza o sensor con la que más potencia de señal reciba, es decir se colocan balizas en las distintas salas, y cuando el usuario ingrese a una de ellas establecerá conexión con la más cercana, que a su vez es la que genera señal más potente y asumirá que el usuario se encuentra en la habitación donde está dicha baliza (Barba, 2012).

Existen varias tecnologías con las que se puede implementar una localización por proximidad, entre ellas están: infrarrojo, RFID, bluetooth, etc. Aunque la desventaja que presenta es que brinda una exactitud pobre y para poder mejorarla se necesitan de muchos más puntos de conexión.

#### **1.5.5. Heurística de Movimiento**

El método consiste en tener la posibilidad de conocer si el usuario está en una posición estática o si se halla en movimiento, para realizar determinar el estado de usuario la potencia de la señal recibida también sirve de ayuda, puesto que se conoce que existen más picos en la señal cuando el dispositivo de medición se halla en movimiento (Chico Ciprián, 2009).

### **1.6. Tecnologías para Localización en Interiores**

Han existido numerosas tecnologías que a lo largo de los últimos años se han utilizado para la implementación de infraestructuras que permitan realizar la localización de objetos o usuarios. Dentro de las tecnologías más comunes se encuentran: la tecnología GPS, dispositivos que trabajan con una red de radiofrecuencia (WiFi, bluetooth, *ZigBee*, etc.), redes GSM, sensores de *Smartphone*, etc.

### **1.6.1. Sistema de Posicionamiento Global (GPS)**

El sistema de posicionamiento global (GPS), es el método más empleado en todo el mundo al momento de hablar sobre geolocalización, trabaja mediante coordenadas geográficas o esféricas.

El sistema GPS está conformado de tres partes, los satélites que orbitan alrededor de la tierra en una posición preestablecida, estaciones que se encuentran ya en la tierra, con la misión de controlar el correcto funcionamiento de los satélites, y finalmente los usuarios receptores de la señal (Olaya, 2014).

La base con la cual funciona el sistema GPS es la triangulación, haciendo uso de señales emitidas por una red de satélites, además de la señal del dispositivo receptor del usuario, para poder determinar las coordenadas X-Y-Z. (Días-Ambrona Tabernilla, 2008).

En el proceso de triangulación se requiere al menos de tres satélites conjuntamente con una estación terrena, los que permiten calcular la coordenada X-Y del receptor, pero es necesario de un cuarto satélite para conseguir estimar la posición Z, pero con la utilización de un mayor número de satélites como referencia se podrá reducir errores, mejorando la precisión, los GPS actuales tienen la posibilidad de trabajar con 12 satélites o más (Olaya, 2014).

### **1.6.2. WiFi**

La tecnología WiFi es la técnica más común para la localización en interiores, y esto se debe a que no se necesita la instalación adicional de hardware dentro la edificación, ya que en la actualidad todas las instalaciones poseen una red WiFi. Apoyado por el enorme incremento del uso del *Smartphone*, lo que lo hace un opción muy viable para la localización en interiores.

Como lo expresa Gómez Ruiz (2015) la localización mediante esta tecnología inalámbrica presenta múltiples retos e inconvenientes, uno de ellos consiste que mediante una red WiFi las ondas de radiofrecuencia son propagadas por múltiples rutas, esto se da

por los fenómenos físicos que experimenta la señal WiFi como: difracción<sup>2</sup>, absorción<sup>3</sup> y reflexión<sup>4</sup> de ondas electromagnéticas.

#### **1.6.2.1. Localización en interiores por WiFi mediante *fingerprint***

Esta tecnología fue aplicada en el sistema realizado por Chico Ciprián (2009) y a continuación se la describirá en términos generales.

El sistema de localización implementado hace uso de una red WiFi, el sistema se enfoca para un posicionamiento físico del usuario. La estimación se la realiza por medio de la huella digital o *fingerprint*.

La arquitectura del sistema no incluye una tercera unidad de procesamiento, esto se debe a que en la funcionalidad del sistema no es posible tener más de una línea de comunicación por la misma tarjeta de red.

El *Smartphone* es el medio con el cual se realizó la localización, por lo que tanto los puntos de acceso y el teléfono deben estar en a la misma red WiFi. El algoritmo no presenta una carga computacional alta, por dicho motivo el *Smartphone* es el único elemento encargado del proceso de localización.

La información que se recopila sobre las potencias de la señal consiste en la dirección MAC (*Media Access Control*) del punto de acceso y el nivel de potencia medido en decibelios (dB), se toman todos los datos de cada punto de acceso y se los almacena en un archivo de extensión .TXT, esto a fin de almacenarlos en la memoria del propio teléfono y no necesitar tenerlos en una base de datos externa.

El algoritmo implementado dentro del sistema consiste en el Vector de Potencia<sup>5</sup>, donde se realizan estimaciones del vector recogido por el dispositivo con los vectores almacenados dentro del archivo de texto.

---

<sup>2</sup> La desviación de la señal al pasar alrededor de un obstáculo.

<sup>3</sup> Cuando parte de la energía de una señal se pierde con un obstáculo.

<sup>4</sup> Cuando una señal continua propagándose luego de chocar con un obstáculo, aunque con una disminución de energía.

<sup>5</sup> Estructura de datos en la cual se almacenan los valores RSSI de cada medición.

La aplicación del usuario está realizada en *Android* y está compuesta de dos partes o actividades, la primera que corresponde a un calibrador y la segunda correspondiente a un localizador.

La primera está orientada a los encargados de levantar la infraestructura de localización, dentro de esta aplicación el usuario encargado debe medir el potencial de la señal y pulsar dentro del mapa digital la posición desde la cual está realizando la medición, y consecutivamente se almacenaran los datos dentro del archivo de texto.

La aplicación del localizador tiene el objetivo de solicitar las direcciones MAC de los puntos de acceso y calcular la potencia de la señal para continuar con la estimación de la posición dependiendo de la mayor o menor señal recibida.

### **1.6.3. Sistema Global de Comunicaciones Móviles (GSM)**

La tecnología GSM tuvo sus inicios desde el siglo XXI en Europa y seguidamente en todo el mundo. La tecnología está basada en el concepto de celdas, asociando a cada celda una cobertura de la zona geográfica y teniendo una estación base en cada celda. Cada teléfono móvil tiene asignada una tarjeta SIM (módulo de identificación de suscripción), esta permite identificar a cada usuario mediante un único número. Todas las estaciones base son gestionadas por un subsistema de estaciones base y estos a su vez están conectadas al Centro de Conmutación Móvil (CCM, 2008).

#### **1.6.3.1. Localización en Interiores basado en tecnología GSM**

Ruiz Cuesta (2012) desarrollo un sistema de localización en interiores mediante esta tecnología y es expuesta a continuación.

El sistema de localización fue implementando mediante tecnología GSM, haciendo hincapié en el aprovechamiento de que esta red de comunicación está presente en la mayoría del mundo y por lo tanto no se necesita adquirir ni instalar hardware adicional.

El algoritmo utilizado en el sistema está basado en el método de *fingerprint*, solamente que almacenando muestras de la potencia de señal GSM, la estimación de posicionamiento se lo realiza mediante el cálculo de la distancia euclidiana de los valores tomados por el dispositivo móvil y los que se hayan guardado anteriormente.

El sistema fue diseñado para determinar la ubicación simbólica del usuario y no su posicionamiento físico. Dentro de este trabajo se planteó tres maneras de poder inferir la ubicación simbólica del usuario, las cuales son las siguientes:

- Mediante puntuaciones, esta técnica consiste en clasificar a todas las mediciones obtenidas a modo de después calcular el total de cada una y el de menor valor es la zona estimada.
- Media geométrica, se realiza la media geométrica a todas las distancias de cada zona, la menor es la que se elige.
- Media aritmética, se realiza la media aritmética a todas las distancias de cada zona, la menor es la que se elige.

El sistema hizo uso de las tres operaciones y tomando como ubicación la zona que dé como resultado en dos de las tres operaciones.

El sistema fue desarrollo en *Android*, el cual posee funciones que permiten conocer cuáles son las potencias de señal y las antenas con las que tiene conexión (Developers, CellSignalStrengthGsm, 2017), la información suministrada consta de los números de antenas y la potencia de señal recibida en dB, la potencia que esta red maneja oscila entre -113 dB y -51 dB.

Ya que la localización fue realizada por *fingerprint* se debió analizar los dos proceso macro que están presente en este mecanismo, las fases *offline* y *online*. El sistema se compuso de dos aplicaciones móviles, la primera denominada *Activity Reading* y la segunda *Activity Position*.

*Activity Reading* fue la aplicación encargada de tomar las muestras de la señal GSM, dentro de esta sección se distinguen dos mediciones, la primera se realiza en un periodo de tiempo para después almacenar únicamente la media de estas mediciones, y la segunda donde se almacenan un número determinado de valores cada segundo, toda esta información es guardada dentro de un archivo de texto.

La segunda aplicación está orientada directamente hacia el usuario final ya que es la que determinaba el lugar donde está ubicado, su funcionamiento es simple, únicamente toma un conjunto de valores que equivalen a las potencias que alcanzo a tomar para luego hacer la comparación en base a la información calibrada.

#### 1.6.4. Bluetooth

Bluetooth es una tecnología de comunicación inalámbrica para la transmisión de datos mediante radiofrecuencia, las señales se transmiten a una frecuencia de 2.4 GHz, está basado en el estándar IEEE 802.15.1. El uso de este medio de comunicación se incrementó de gran medida por la simplicidad de conexión y por la velocidad de transmisión, llegando a alcanzar velocidades de 32 Mbits/s en su última versión que es la número 4 (Bluetooth, 2017; Pérez Porto & Merino, 2017).

##### 1.6.4.1. Localización en Interiores basada en *iBeacons*

Esta tecnología fue aplicada por Li, Xu, Wang, & Muhammad (2016), la cual se describe a continuación.

Los *iBeacons* son dispositivos de comunicación inalámbrica que utiliza el protocolo de Bluetooth *Low Energy* (BLE) en su última versión, los *iBeacons* mantienen una gran ventaja frente a los dispositivos habituales de bluetooth ya que consumen menor energía y no requiere una vinculación entre nodos para transferir datos, además de ello provienen información como el código UUID (*Universally Unique Identifier*) y el nivel RSSI.

El sistema se basó en posicionamiento físico con coordenadas en base a la potencia de las señales obtenidas de los *iBeacons*. La aplicación fue realizada para *Android*, empleando las librerías pertinentes para obtener información sobre conexión Bluetooth (Developers, BluetoothDevice, 2017) y utilizo como repositorio de la información de cada baliza una base de datos desarrollada en *SQLite*.

El procedimiento de localización para el sistema se lo puede resumir de la siguiente manera: inicia en la toma de los valores RSSI de las balizas, ejecuta algoritmos para filtrar y purificar los datos recibidos y representa la información mediante una representación 3D en el *Smartphone*.

El sistema consta de 36 *iBeacons* para poder realizar el posicionamiento y un total de 15 puntos de referencia en los cuales se toma los valores RSSI, los mismos que servirán como muestra de comparación.

Como se mencionó anteriormente los *iBeacons* no necesitan realizar una vinculación para transmitir datos, solamente envían señales *broadcast* para que cualquier



dispositivo que esté presente en el entorno lo escuche, y de esta manera siempre estará en modo activo y será el terminal móvil que capte esta señal posteriormente a una solicitud del usuario.

Las medidas de RSSI son almacenadas en un vector de datos para poder luego ser tratadas, lo que comúnmente se realizaría para comparar valores es obtener la media de los mismos pero dada la naturaleza de una señal de radiofrecuencia este método no es muy apropiado, por lo que se requiere de procesos adicionales para mejorar la precisión.

Se necesitó realizar una purificación de la medición de la señal mediante el modelo Gaussiano, para poder tener una mayor seguridad en cuanto a la validez de la muestra tomada, dicho algoritmo da como resultado un valor numérico, donde se consensuó el valor de 0.6 como criterio mínimo de aceptación, es decir si con la ejecución del algoritmo se obtenía un valor mayor a 0.6 y menor a 1 se tenía una medición con muy buena sin muchas variaciones.

Previamente se tiene almacenada información de las balizas, como también muestras de las señales en vectores y además la coordenada respectiva de cada una. Con esta información y el conjunto de datos tomados en tiempo real se procede a obtener la distancia euclidiana considerando el promedio del vector medido por el *Smartphone* y el promedio de los vectores almacenados en la base de datos.

Finalmente se busca el *iBeacon* dentro de la distancia calculada y se puede obtener la coordenada promedio de la baliza dentro de la base de datos.

#### **1.6.4.2. Bluetooth y Redes Neuronales**

El uso de redes neuronales como algoritmo de localización también es posible, y este es el mecanismo propuesto por Ali Hassan (2016). La localización se la planteó realizar por *fingerprint* y realiza una ubicación simbólica dentro de la zona de implementación.

El sistema se desarrolló con dispositivos bluetooth conectados con *Arduino*, y estas se distribuyeron en cada habitación y en el pasillo, para tener una cobertura de toda la zona. Dentro de la primera fase, la de calibración, se toma muestras de la señal de bluetooth y se almacena dentro de una matriz la distancia euclidiana a cada baliza,

teniendo una matriz de referencia de m muestras por n puntos de referencia (Ali Hassan, 2016).

La localización se la realiza mediante una ecuación de descenso de gradiente, donde se analiza la señal captada desde el dispositivo a ser localizado y los datos de la matriz. La red neuronal está compuesta en un inicio de 6 neuronas, las cuales resuelven la ecuación y dan como resultado un número binario especificando un 1 en la habitación ubicada según su orden, es decir si se tuviese 1000 el usuario estaría localizado en la primera habitación (Ali Hassan, 2016).

### **1.6.5. Campo Magnético**

Este es un trabajo propuesto por García Vázquez, Galván Tejada, & Galván Tejada (2013), el cual está únicamente planteado, más no implementado en un escenario real. La tecnología de localización será expuesta a continuación.

La propuesta consiste en hacer uso del campo magnético de la tierra, basándose en el hecho de que cada pulgada cuadrada dentro del globo terráqueo tiene una medida de magnetismo única, y este es el criterio con el cual se pudiese determinar la posición en la que se encuentra.

La metodología con la cual el posible sistema está planteada es mediante *fingerprint*, donde se tomaría como muestra la medición del magnetismo de ciertos puntos de referencia. La medición del campo magnético se ha facilitado con la incorporación de un sensor de magnetismo en los *Smartphone* de últimas generaciones, es por ello que el artículo plantea esta alternativa.

Las fases del sistema son las mismas que se estudiaron para una localización por radiofrecuencia, de calibración (*offline*) y de localización (*online*). El hecho de poseer en la actualidad un *Smartphone* con un sensor de magnetismo, y que el magnetismo sea un elemento natural de la tierra aporta de gran manera ya que se evita todo el proceso de instalar una infraestructura de localización como en los casos de WiFi, bluetooth, GSM, etc.

La fase de calibración podría tener dos opciones de realización, una de ellas sería realizar un proceso exhaustivo y largo de toma de muestras, capturando valores en zonas

muy específicas y detalladas, lo que conllevaría a tener una mayor precisión, pero tiene una desventaja en que ciertos cambios en el ambiente obligarían a una nueva toma de datos. Sin embargo se tiene la segunda opción denominada enfoque siguiendo al líder, la cual es más sencilla que la anterior, donde se toma muestras de los alrededores de cada habitación generando un perímetro, y para la localización se debería verificar si la medición realizada esta dentro del perímetro para determinar su localización.

Estos dos enfoques dependen del objetivo para el cual el sistema se desarrollara, aquellos que necesiten de una localización física mediante coordenadas deberán tener la calibración más rigurosa, pero si lo que se busca es tener un ubicación simbólica la segunda opción es la más viable.

#### **1.6.6. Localización en interiores mediante tecnología VLC**

Esta alternativa fue realizada por Muñoz Arcentales, Marin Garcia, Calero Bravo, & Chavez Burbano (2014).

La tecnología VLC (*Visible Light Communications*) consiste en tener como medio de comunicación la óptica no guiada por luz visible, lo principal de la tecnología VLC es la eliminación de interferencias del entorno, que en una red de radio comunicación es muy complicado evitarlas.

La tendencia actual esta hacia el uso de dispositivos de luminosidad LED, cuyas ondas ofrecen ciclos de vida más largos y eficiencia de conversión más alta. VLC puede ser utilizada como tecnología complementaria a la de radiofrecuencia, ya que poseen un menor costo, media alta velocidad, robustez a las interferencias electromagnéticas, etc.

Aprovechando esta tecnología se plantea utilizarla para la localización en interiores, apoyada por el hecho de que cada vez más se adquieren este tipo de lámparas y de que los *Smartphone* actuales traen incorporados un sensor de luminosidad.

El sistema descrito plantea realizar una triangulación de datos haciendo uso del algoritmo TDoA. Se tienen colocados lámparas SSL dentro de una amplia habitación para que sirvan como balizas, en lugar de puntos de acceso con WiFi, *iBeacons*, etc. La aplicación del usuario tomaría los datos enviados por las balizas y ejecutarían la triangulación para calcular las coordenadas.

La funcionalidad del sistema consiste de dos componentes, lámparas SSL y un receptor de ultrasonido, este receptor es necesario para recibir la orden de activación por parte del usuario, por lo que el terminal móvil debería tener un emisor de ultrasonido.

El sistema maneja un retardo progresivo en el envío de los códigos por parte del emisor de ultrasonido, esto se debe a que pueden dos lámparas estar a la misma distancia del usuario causando una colisión entre las señales enviadas.

### **1.6.7. ZigBee**

*ZigBee* es un sistema de comunicación inalámbrico que se centra en usar la menor tasa de comunicación entre dispositivos, a fin de tener un menor consumo energético. *ZigBee* fue diseñado hacia una mayor orientación al área domótica. Esta tecnología funciona mediante radiodifusión, se basa en el estándar IEEE 802.15.4, y adopta un medio de canal en 2.4 GHz. (Gutiérrez, 2017).

Esta tecnología es mucho más sencilla de utilizar y más barato comparado con bluetooth, la creación de un dispositivo *ZigBee* requiere el 10% de lo necesario para un dispositivo bluetooth. En cuanto al software del dispositivo es necesario un 50% más que con bluetooth (Gutiérrez, 2017).

Al tener una red de dispositivos *ZigBee* existen tres tipos de nodos o terminales, un nodo controlador para la gestión de la red, un nodo que emula a un router de WiFi dedicado a transferencia de datos entre los nodos de la red, y el nodo final que se encarga de comunicarse con los componentes que se controlan (Gutiérrez, 2017).

#### **1.6.7.1. Localización de Interiores basada en tecnología *ZigBee***

Una aplicación de la tecnología *ZigBee* fue desarrollada e implementada por Álvarez & Las Heras (2016).

El sistema planteado consiste en tener una red censada de dispositivos *ZigBee*, ya que esta red opera bajo radiofrecuencia al igual que WiFi o bluetooth la teoría aplicada es la misma, se tiene como medición referente el nivel RSS.

La ubicación del usuario se la realiza mediante posicionamiento físico, es decir estableciendo y usando un sistema de coordenadas propio para la zona donde se

implemente la infraestructura. La infraestructura se compone de 6 nodos *ZigBee* o nodos estáticos como se los denomina, y cubre una zona de 65 m<sup>2</sup> con 5 habitaciones internas.

La red de nodos se compone por un nodo principal, al cual los usuarios se conectan para solicitar la ubicación. Seis nodos estáticos los cuales sirven como balizas o puntos de referencia para el sistema. Y los nodos repetidores, los cuales representan a los nodos usuarios que serán localizados.

El algoritmo de localización consiste en tomar muestras de la señal de radio en cada punto de referencia realizando de esta manera una calibración previa. Se propone utilizar la ecuación de propagación en espacio libre, y dentro de ella existe una variable que engloba a los datos de las coordenadas, tanto del objeto de referencia como de los objetos estáticos. Pero esta ecuación no es muy precisa debido al multitrayecto en una red de radio, dado esto se planteó una mejora de la ecuación incluyendo más variantes en la misma para minimizar este inconveniente.

Como el sistema está compuesto de varios nodos estáticos se debe realizar esta operación con todos ellos y luego poder estimar la coordenada más apropiada, para poder tener un criterio de selección se realiza una función por minimización de coste, dentro de esta función se analiza cada punto de referencia y al final en el punto donde se haya obtenido el menor valor se lo considera como la mejor aproximación.

El sistema realiza la localización en base a computadoras personales puesto que el *ZigBee* se conecta vía *USB (Universal Serial Bus)* y todo el proceso de posicionamiento es realizado por la propia *laptop*, lamentablemente la tecnología *ZigBee* no ha sido implementada para *Smartphone* por lo que este tipo de sistema solo puede ser realizado teniendo un computadora como nodo a localizar.

#### 1.6.8. Comparación entre Bluetooth *Low Energy* (BTLE) y WiFi.

	WiFi	BLE
<b>Topología de la red</b>	Ad-hoc Estrella	Estrella Bus
<b>Frecuencia</b>	2,4/5 GHz	2,6 GHz
<b>Tasa de transmisión</b>	11/54 Mbps	1 Mbps
<b>Rango</b>	Hasta 100m	Hasta 30m
<b>Consumo de Energía</b>	Alto	Muy bajo
<b>Duración de batería</b>	Varias horas	Varios meses
<b>Costo</b>	Alto	Bajo

<i>Smartphone</i>	Soportado	Soportado
<b>Desarrollado para la localización</b>	No	Si
<b>Complejidad de instalación</b>	Media	Baja
<b>Precisión</b>	5-10m	1-3m
<b>Aplicaciones típicas</b>	WLAN, comunicación	Sensores, localización en interiores
<b>Seguridad</b>	Alto riesgo	Bajo riesgo

Tabla 3: Comparación características técnicas

Para el análisis de la comparación es necesario conocer lo que es y cómo funciona el RSSI (*Received Signal Strength Indicator*), lo cual se explica a continuación.

El RSSI o por sus siglas en español Indicador de Fuerza de Señal Recibida, es utilizado únicamente para indicar la señal que reciben los dispositivos receptores consiste en una escala que va desde 0 hasta -100 dBm.

El RSSI es el mecanismo de medición más empleado en la localización en interiores tomando de referencia una red de radiofrecuencia, aunque posee una desventaja y es que muchas de las veces este valor depende del fabricante del dispositivo receptor.

Zhao, X., Xiao, Z., Markham, A., Trigoni, N., & Ren, Y (2014) presentan un artículo donde exponen una comparativa entre las tecnologías más comunes en la radiofrecuencia, las cuales son WiFi y bluetooth BLE.

Dentro de la comparativa se utilizó el método de la *fingerprint* para la localización, fueron planteadas pruebas tanto en ambientes de interiores como de exteriores, se colocaron las balizas o puntos de acceso en las mismas pociones y se consideraron los mismos puntos de referencia con ambas tecnologías.

El objetivo es determinar cuál tecnología posee una mejor relación entre RSSI y distancia, cuál de ellos es el más confiable y presenta una mejor precisión en la localización. El sistema se propuso tomando muestras cada 50 cm en un espacio de 20 metros, lo que dio como resultado un conjunto de 200 muestras, esto se lo realizo exactamente para ambas tecnologías.

Las pruebas fueron realizadas considerando los dos modelos de propagación de señal, las cuales son: *Line of Sight* (LOS) y *Non Line of Sight* (NLOS), obteniendo

resultados en los cuales la tecnología Bluetooth *Low Energi* (BLTE) mantiene una mejor relación entre el nivel RSSI y la distancia en ambos modelos de propagación.

Technology	Setup	$n$	$RSSI(d_0)$ (dBm)	$R^2$	$var(dBm^2)$
WiFi	LOS	2.13	-36.1	0.492	20.02
WiFi	NLOS	3.33	-42.7	0.363	5.55
BTLE	LOS	1.98	-52.0	0.775	16.75
BTLE	NLOS	1.35	-72.3	0.872	4.68

Tabla 4: Parámetros derivados de la localización en interiores (Zhao, Xiao, Markham, Trigoni, & Ren, 2014).

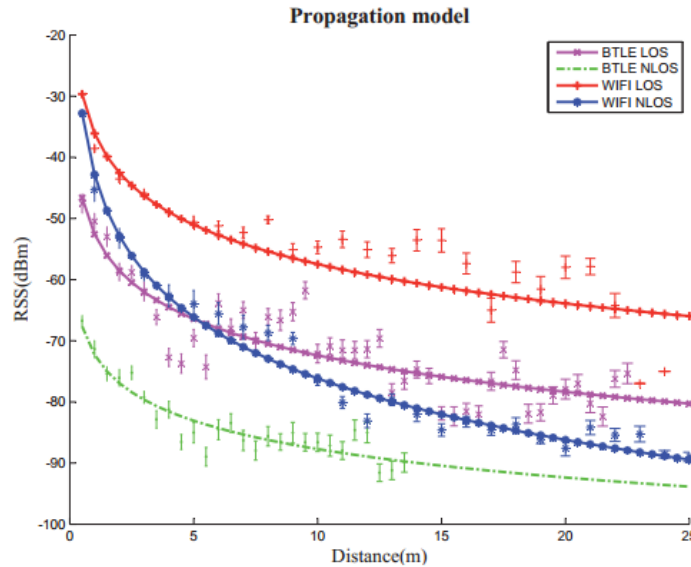


Figura 6: Gráfico de Propagación de la señal (Zhao, Xiao, Markham, Trigoni, & Ren, 2014).

Ya con la implementación de la localización en interiores existe una mayor precisión con BTLE, obteniéndose un error promedio de 3.8m con BTLE y de 5.3m con WiFi, inclusive se realizó una fusión entre las tecnologías pero no mejora la precisión obtenida únicamente con Bluetooth *Low Energi* (BLTE).



Figura 7: Trayectorias estimadas (Zhao, Xiao, Markham, Trigoni, & Ren, 2014).

El análisis concluye que existen 3 razones por las cuales la tecnología BTLE es mejor que WiFi en localización en interiores. Por el mecanismo de saltos del canal, donde si un canal de comunicación posee mucha interferencia se salta a otro canal, lo que no ocurre con WiFi. Transmisión de bajo costo, el hecho de necesitar baja energía para transmitir datos ayuda a contribuir a evitar la visibilidad de todos los dispositivos, esto beneficia al descartar ciertas balizas que no se encuentran cerca del usuario. Y por una mayor tasa de frecuencia de muestreo, ya que con BTLE es de 50Hz y con WiFi es de 1Hz.

## **1.7. Productos comerciales de localización en interiores**

### **1.7.1. Situm**

Situm es un producto desarrollado y comercializado por Situm Technologies (SITUM, 2017) una empresa española. El producto consiste en una aplicación móvil de localización en interiores desarrollada para las plataformas *Android* e *IOS*.

La aplicación cuenta con una gran variedad de funciones que van desde una aplicación estándar gratuita y una Premium la cual requiere pago. Entre las múltiples funciones que la aplicación oferta se encuentran:

- Gestión de cartografía, donde permite generar o montar espacios sobre los planos o mapas del área.
- Uso de rutas, esta opción permite crear rutas internas dentro de los edificios, tal y como funciona las rutas de *Google Maps*, permitiendo también incluir gradas o ascensores en la ruta.
- Tener visión en tiempo real del usuario que está siendo localizado.
- Presentación de informes estadísticos sobre los usuarios y las localizaciones, facilitando cualquier análisis que requiera esta información.
- SDK<sup>6</sup> para desarrolladores, paquete de APIs<sup>7</sup> para que se puedan aplicar las funciones que proporciona el sistema dentro de cualquier ámbito.

---

<sup>6</sup> Conjunto de herramientas y programas de desarrollo que permite al programador crear aplicaciones para un determinado software.

<sup>7</sup> Conjunto de funciones y procedimientos con el fin de ser utilizadas por otro software.



La localización con Situm es realizada únicamente con un *Smartphone*, dicho teléfono debe tener incluido en su hardware el sensor de posicionamiento en 3D llamado giroscopio, y además todo el proceso de localización deber ser hecho conectado a Internet y por medio de datos no en una red WiFi.

El sistema de localización consiste en cuatro pasos, los cuales se describen a continuación:

1. Crear una cuenta en el sitio <https://dashboard.situm.es>, consecuentemente se debe seleccionar el país, ciudad y edificación donde se implementara la localización en interiores, una vez seleccionada se cargar los planos digitales del edificio planta por planta.
2. Calibración de edificio, para este paso se debe descargar la aplicación de Situm, el proceso consiste en caminar por los sectores marcados dentro del plano, de esta manera el sistema aprende las localizaciones y las rutas, almacenando toda esta información en la nube.
3. El tercer paso consiste ya en la navegación dentro del edificio, dentro de este apartado ya se consigue la localización en interiores marcando como el camino que el usuario sigue, como si se tratase de un GPS en exteriores.
4. El último paso es el correspondiente a los desarrolladores, dentro del cual se descarga el SDK para *Android* o *IOS*.

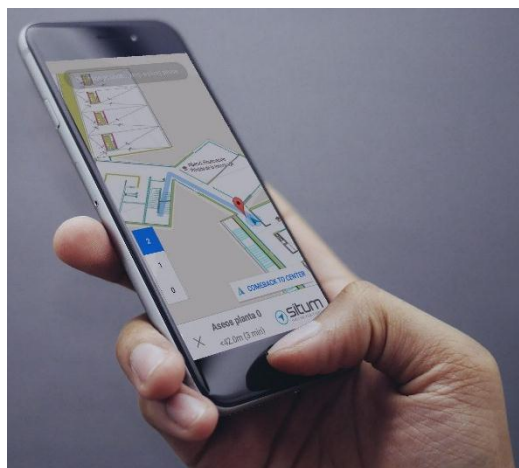


Figura 8: App de Situm (SITUM, 2017).

### 1.7.2. Indoor Atlas

La aplicación desarrollada por IndoorAtlas Ltd (IndoorAtlas, 2017), consiste en un sistema de localización en interiores que usa como referencia de ubicación el campo magnético de la tierra, basándose en que la posición geográfica del edificio y la interacción del campo magnético con el acero y otros materiales generan muestras de magnetismo únicas para cada piso y habitación dentro de un edificio.

El sistema maneja almacenamiento en la nube para guardar toda la información de cada zona de localización. El campo magnético no es el único referente que emplean, también suelen integrar la red WiFi para obtener mejores precisiones. Así como poseen un SDK que permite desarrollar aplicaciones reutilizando las funciones de localización en interiores.

El sistema está contemplado para que no se requiera la instalación de ninguna infraestructura de hardware, lo que aporta para evitar costos de instalación. El sistema brinda información en tiempo real de la ubicación de usuario, algo que con WiFi o bluetooth es más complicado de realizar.

La funcionalidad del sistema es muy similar a la de Situm, primero se necesita crear un usuario, después subir los planos del edificio y realizar una calibración con la aplicación propia de la empresa, necesitando claramente un sensor de magnetismo en el *Smartphone*.

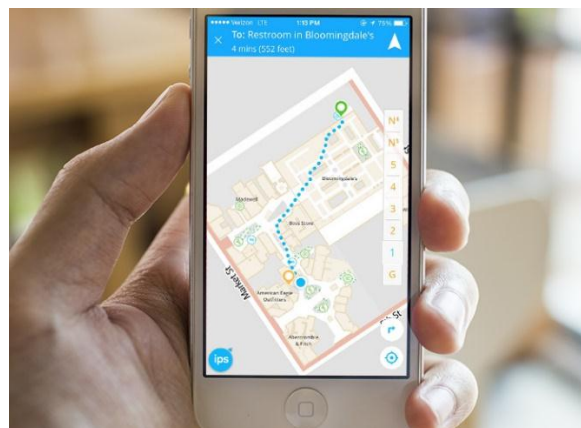


Figura 9: App de IndoorAtlas (IndoorAtlas, 2017).

### **1.7.3. Google Maps en Interiores**

Los mapas de interiores de las edificaciones era un objetivo que se perseguía con la inclusión de *Google Maps 6.0*. *Google Maps* es la herramienta de cartografía digital web más utilizada para las ubicaciones y rutas en exteriores, y la inclusión de mapas de interiores es una gran ayuda que puede potenciar el uso de la herramienta (Google, 2017).

El proceso de subir los planos de cualquier edificio es de igual manera que con las tecnologías anteriores, únicamente se ingresa a la cuenta de Google y ya se puede cargar los planos (Google, 2017).

Google además ofrece la API de los mapas de interiores para el desarrollo de herramientas, de la misma manera que lo hace con los de exteriores. Lamentablemente Ecuador no está habilitado aun para poder tener este servicio (Google, 2017).

### **1.7.4. Ekahau**

Es un software que permite realizar localización en interiores empleado las redes *Wireless*, hace uso de la *fingerprint* para la ubicación y permite también realizar una localización en exteriores si se mantiene una buena cobertura de la red (Ekahau, 2017).

Consiste en una localización física con coordenadas X-Y, incluyendo si se desea la información simbólica de la ubicación. Permite realizar una localización a cualquier dispositivo que sea capaz de conectarse con la red inalámbrica (Chico Ciprián, 2009).

Se requiere que los usuarios tengan instalado el software Ekahau Client para poder usar el servicio, adicional a ello se necesita el EPE (*Ekahau Positioning Engine*), que es un software encargado de controlar y gestionar la localización. El sistema utiliza el vector de potencia como algoritmo de localización (Chico Ciprián, 2009).

### **1.7.5. Estimote Beacons**

Son dispositivos *Beacons* de Bluetooth BLE destinados para localización en interiores, desarrollados por la empresa *Estimote, Inc*, la cual está centrada en estos y otros productos tecnológicos enfocados en el posicionamiento físico (Estimote, 2017).

Existen dos tipos de *Beacons* para la localización en interiores: *Location Beacons* y *Location Beacons with UWB*<sup>8</sup>, para ambos tipos de dispositivos se proporciona un SDK de desarrollo, únicamente para dispositivos IOS por el momento.



Figura 10: Location Beacons (Estimote, 2017)

## 1.8. Conclusiones

Es necesario hacer énfasis en la gran cantidad de tecnologías existentes hoy en día para realizar localización en interiores. Pero de entre todas ellas las más convenientes son WiFi y Bluetooth debido a la gran gama de servicios que emplean estos tipos de redes.

Lo que respecta a las técnicas y métodos que existen para realizar la localización dentro de la literatura revisada, varias referencias utilizan el método de *fingerprint* para su implementación, esto debido a que se puede captar de mejor manera la señal en los diversos ambientes, a diferencia de la triangulación, donde las interferencias son el principal inconveniente por las que puede llegar a obtenerse errores muy considerables.

Los productos terminados representan una muy buena opción, puesto a que la mayoría disponen de librerías para ampliar la localización en interiores, las balizas bluetooth además poseen diferentes sensores que a más de realizar la localización proporciona más información del entorno.

---

<sup>8</sup> Tecnología de radio que usa un ancho de banda mayor de 500 MH.

## Capítulo 2

### 2. Tecnología WiFi

#### 2.1. Introducción

Según la literatura revisada acerca de la localización en interiores, la cual fue descrita en el capítulo anterior, la tecnología WiFi es la más empleada en un sistema de localización en interiores. Una red WiFi resulta muy apropiada por varias razones, como el costo nulo para implementar una nueva infraestructura, facilidad de acceso por parte de dispositivos, etc.

Por todo lo antes mencionado es conveniente incluir a esta tecnología para la realización del prototipo de la aplicación móvil. A continuación dentro de este capítulo se detallará la infraestructura, algoritmo y desarrollo del prototipo de la aplicación móvil, basándose en la tecnología WiFi.

#### 2.2. Infraestructura de Localización

La zona designada para realizar el levantamiento de la infraestructura de localización es último piso de la Facultad de Filosofía de la Universidad del Azuay. Esta planta posee un total de 7 laboratorios, con 3 oficinas adicionales, los cuales están dividido en áreas izquierda y derecha. La infraestructura será levantada en la parte izquierda, donde están presentes cuatro laboratorios, una parte del pasillo y una oficina.

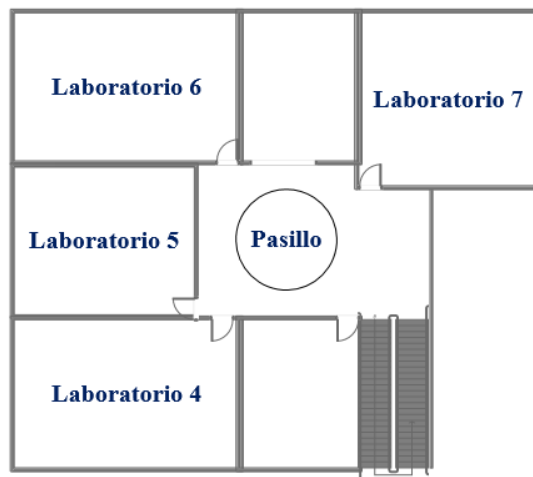


Figura 11: Distribución de Laboratorios (Autoría Propia, 2017).

El método que se empleará para la aplicación con la tecnología WiFi es el método *fingerprint*, para lo cual se necesita conocer los puntos de acceso a la red WiFi (WiFi-UDA). La distribución de los mismos está organizada por los pisos del edificio, dentro de cada planta existen 3 o 4 puntos de acceso, en la última planta (área de implementación) existen 4 puntos de acceso, cuya distribución está representada en Figura 12, adicionalmente se consideró un quinto punto de acceso, el cual no está ubicado en la cuarta planta, pero sirve para realizar una mejor estimación.

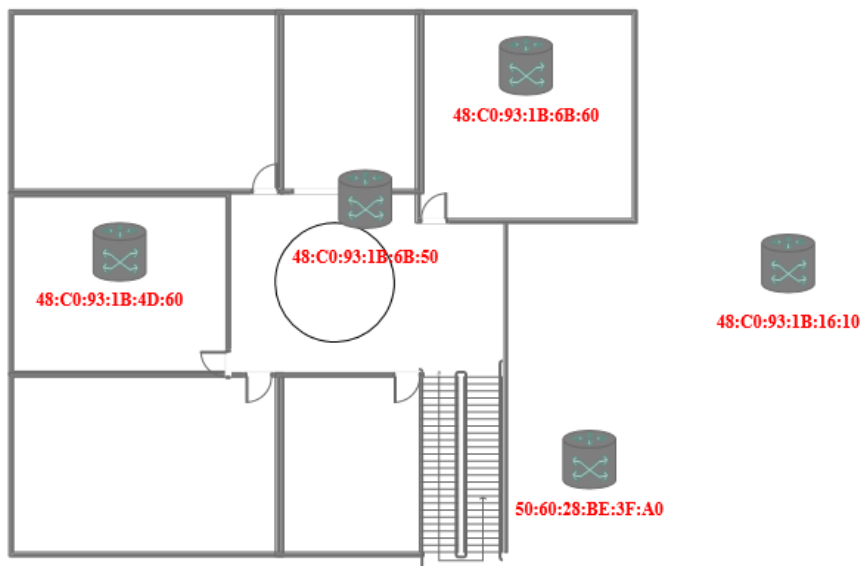


Figura 12: Distribución de puntos de acceso y sus MAC (Autoría Propia, 2017).

### 2.3. Método *Fingerprint*

El método que se aplicará es *fingerprint*, el cual consiste en tomar los niveles de potencia con respecto a los puntos de acceso, es decir medir cuál es la potencia de la señal en la red WiFi para cada una de los puntos identificados en la Figura 12.

Para poder realizar la toma de los niveles de potencia se necesita establecer puntos de referencia dentro de los cuatro laboratorios y el pasillo. Para cada laboratorio se consideró 5 puntos referenciales y para el pasillo 4 puntos.



Figura 13: Puntos de Referencia (Autoría Propia, 2017).

Para cada uno de los puntos de referencia es necesario conocer los niveles de potencia (RSSI) con respecto a cada punto de acceso, por lo que se almacena en un vector el número del punto, los cinco niveles de RSSI medidos y un identificador del laboratorio físico al cual pertenece el punto de referencia. Debido a que los puntos de acceso están identificados por sus respectivas direcciones MAC, y dado que están en numeración hexadecimal, es necesario tener una codificación propia para referenciar a la dirección MAC, dicha codificación está realizada en literales, como se muestra en la Tabla 5.

MAC	Código
48:c0:93:1b:6d:50	a
48:c0:93:1b:6b:60	b
48:c0:93:1b:4d:60	c
50:60:28:be:3f:a0	d
48:c0:93:1b:16:10	e

Tabla 5: Codificación de direcciones MAC (Autoría Propia, 2017).

Un ejemplo de la estructura de los vectores donde se almacena la información se presenta en la Tabla 6:

Punto	a	b	c	d	e	Salón
1	-59	-56	-90	-90	-90	Laboratorio 7
2	-68	-59	-90	-90	-90	Laboratorio 7
3	-46	-40	-90	-84	-90	Laboratorio 7
4	-54	-44	-90	-90	-90	Laboratorio 7
5	-57	-44	-90	-90	-90	Laboratorio 7

Tabla 6: Vectores de puntos de referencia (Autoría Propia, 2017).

## 2.4. Algoritmo kNN (*k-nearest neighbors*)

El algoritmo kNN fue utilizado para realizar la localización en interiores, el método consiste la comparación de los valores almacenados en vectores, con un vector que posea los valores tomados en la posición a ubicarse, con referencia a los mismos puntos de acceso que están almacenados.

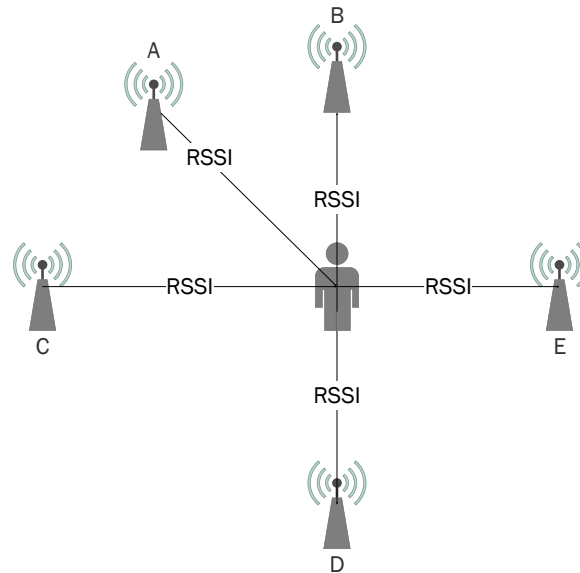


Figura 14: Recepción de Valores RSSI (Autoría Propia, 2017).

Para realizar esta comparación el algoritmo emplea la fórmula de la distancia euclidiana, con los respectivos valores RSSI.

$$De(P, Q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \dots + (p_n - q_n)^2} = \sqrt{\sum_{i=1}^n (p_i - q_i)^2}$$

Ecuación 1: Distancia Euclidiana (Wikipedia, 2016)

El funcionamiento del algoritmo kNN consiste en recorrer cada punto de referencia realizando una comparación de valores RSSI, la operación de la distancia euclidiana se la debe realizar en el orden de los puntos de acceso, es decir el valor del punto “a” del vector almacenado con el valor del punto “a” del vector tomado en este momento.

$$De(P, Q) = \frac{1}{M} * \sqrt{\sum_{i=1}^n (RSSI_i(x, y) - RSSI_i(x_i, y_i))^2}$$

donde  $M$  = número de elementos de ptos\_escaneo

donde  $RSSI_j$  = fuerza de la señal recibido del nodo baliza<sub>j</sub>

Ecuación 2: Distancia en algoritmo kNN (Rios, 2012)



Está claro que casi nunca los valores que se midan utilizando el *Smartphone* al momento de realizar la ubicación coincidan con los valores almacenados anteriormente, ya que existe una gran variabilidad del RSSI, sin embargo, el algoritmo trata de buscar similitud entre el vector medido y el vector de un punto de referencia almacenado, tomando como comparación el punto hacia el cual exista menor distancia, asumiendo que el usuario se halla presente en el laboratorio al cual pertenece dicho punto de referencia más cercano.

El funcionamiento esquemático del algoritmo es el siguiente:

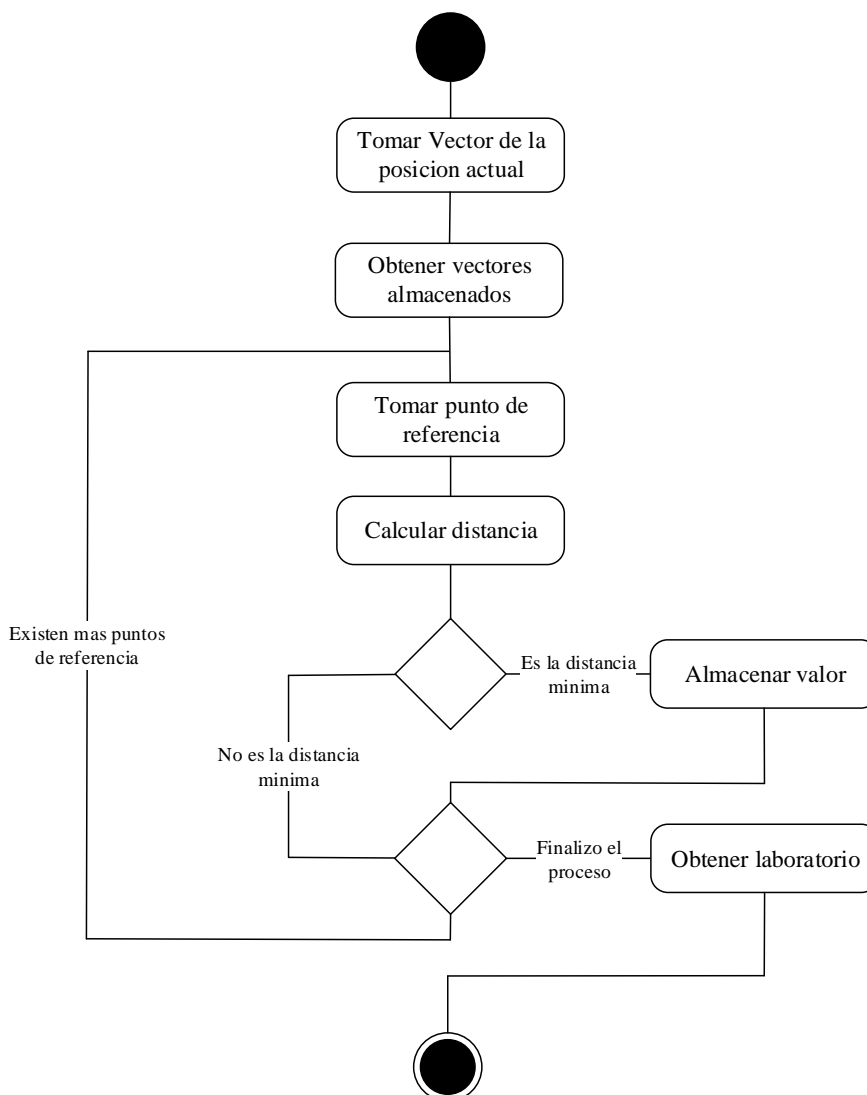


Figura 15: Diagrama de Funcionamiento del algoritmo kNN (Autoría Propia, 2017).

## **2.5. Levantamiento de Requerimientos de Software**

### **2.5.1. Introducción**

En el presente documento se detallará las especificaciones del prototipo de la aplicación móvil, las cuales servirán como base para un futuro desarrollo de la misma. Este documento también establecerá el alcance que tendrá en software.

El análisis se lo presentará empleando el Lenguaje de Modelado Unificado (UML), haciendo uso del diagrama de casos de uso y siguiendo el estándar IEEE 830.

#### **2.5.1.1. Propósito**

El propósito de este segmento es analizar mediante diagramación UML la funcionalidad de la aplicación móvil, la cual en términos generales consistiría en ingresar como usuario de la universidad, cargar un plano de la zona de cobertura y determinar la localización del usuario.

#### **2.5.1.2. Ámbito del Sistema**

El software tendrá por nombre de “Prototipo de Aplicación Móvil para Localización en Interiores”. La aplicación está contemplada únicamente para proporcionar al usuario información respecto al lugar donde se encuentra en ese momento, especificando únicamente el nombre del aula o salón. La aplicación no realizará seguimientos en tiempo real, no generará rutas para guiar a los usuarios, no proporcionará información sobre el entorno donde se halla, y tampoco llevara un registro del historial sobre los salones en donde hayan estado los usuarios.

La aplicación tiene varios beneficios, entre los que están: poder mejorar la localización en interiores, para la cual un sistema GPS tiene muy poca precisión, llegar a tener mediante consultas al *Smartphone* el salón o pasillo donde se está ubicado, aportar al desarrollo de un futuro proyecto en cual se llegará a proporcionar información contextual.

El objetivo único que la aplicación persigue es la de proporcionar la ubicación del usuario, el proceso de ubicación consiste en un plano de referencia, el cual estará seccionado de acuerdo a los salones internos y un icono que represente la ubicación del usuario en dicho plano y una etiqueta que recalque la ubicación donde se encuentra.

### 2.5.1.3. Definiciones, Acrónimos y Abreviaturas

Nombre	Descripción
Usuario	Persona que usará el sistema para gestionar procesos
ERS	Especificación de Requisitos Software
RF	Requerimiento Funcional
RNF	Requerimiento No Funcional
ILS	<i>Indoor Location System</i> (Sistema de Localización en Interiores)

Tabla 7: Definiciones, Acrónimos y Abreviaturas (Autoría Propia, 2017).

### 2.5.1.4. Referencias

Título del Documento	Referencia
Standard IEEE 830 - 1998	IEEE

Tabla 8: Referencias (Autoría Propia, 2017).

### 2.5.1.5. Visión general del documento

El documento posee tres partes. En la primera de ella se da una breve introducción del documento, brindando una visión general sobre el software que se está tratando.

La segunda parte del documento consiste en una descripción generalizada sobre el sistema, facilitando mediante diagramas las diversas funciones que la aplicación posee, algunas restricciones, supuestos y factores ligados directamente con el desarrollo.

La última parte de documento consiste en la definición detallada de todos los requerimientos que el sistema deberá cumplir.

## 2.5.2. Descripción General

### 2.5.2.1. Perspectiva del Producto

El prototipo para el ILS consiste en una aplicación para dispositivos móviles, dicho prototipo está enfocado para trabajar en el sistema operativo *Android*. El sistema de localización esta implementado en una arquitectura de dos capas. La capa del servidor que se encarga de receptor los datos enviados por el *Smartphone*, procesar la ubicación y retornar el resultado. Y la capa del usuario o del dispositivo móvil, el cual tendrá las funciones de toma de datos, envió al servidor y representación del resultado.

### 2.5.2.2. Funciones del Producto

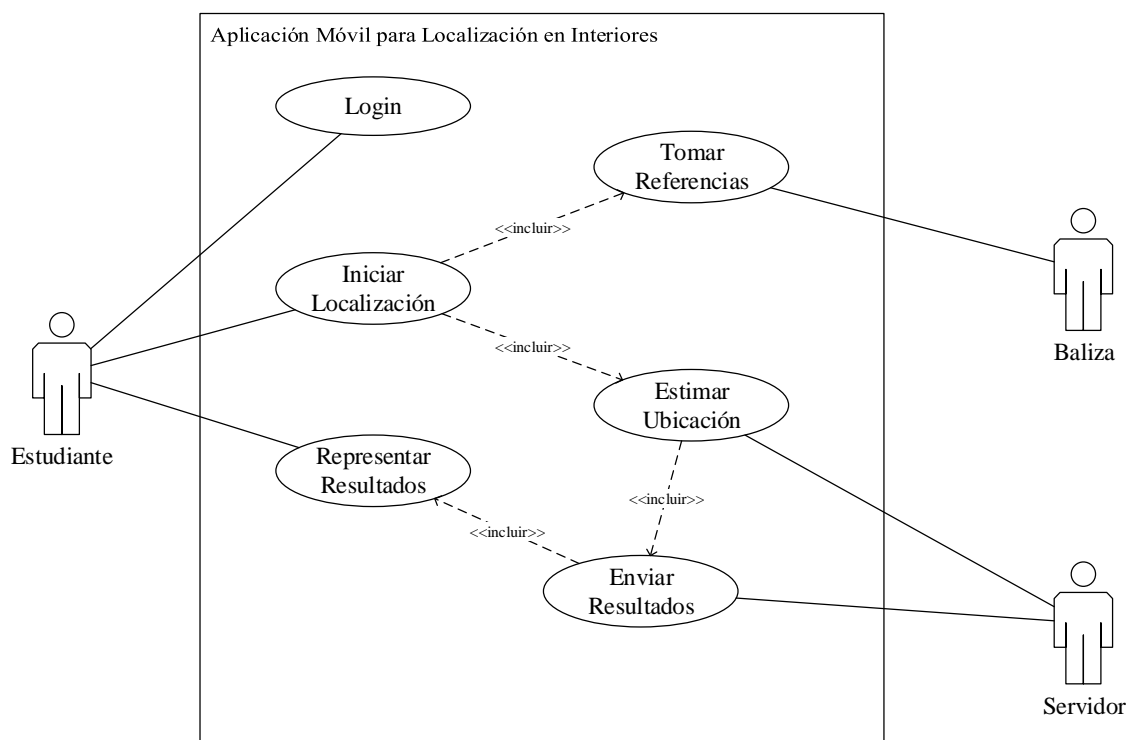


Figura 16: Diagrama de Casos de Uso (Autoría Propia, 2017).

### 2.5.2.3. Características de los Usuarios

<b>Tipo de usuario</b>	Estudiante
<b>Formación</b>	Conocimiento en manejo de aplicaciones móviles
<b>Actividades</b>	Solicitar ubicación

Tabla 9: Características de Usuario (Autoría Propia, 2017).

### 2.5.2.4. Restricciones

- La aplicación necesita estar conectada a Internet para ser usada.
- Lenguajes de programación: JAVA.
- El sistema se diseñará según un modelo cliente/servidor.
- El sistema deberá tener un diseño de interfaz sencilla.
- Utilizar distintas librerías de acuerdo al tipo de baliza que se emplee.

### 2.5.2.5. Suposiciones y Dependencias

- Estabilidad en los requerimientos planteados en este documento.
- Los dispositivos que se adquieran como balizas estén en óptimas condiciones.
- Que la alternativa de localización que se haya adoptado sea la más adecuada.

## 2.5.3. Requisitos Específicos

### 2.5.3.1. Interfaces de Usuario

La interfaz de la aplicación móvil estará contenida principalmente por el plano de la planta en la cual se implementó la localización en interiores, dentro de dicho plano existirá también un ícono que represente la figura del usuario. Existirán también botones para poder solicitar la ubicación por parte del usuario final. Y también una ventana de *login* para el inicio de sesión de los estudiantes universitarios.

### 2.5.3.2. Interfaces de Hardware

- Balizas de acuerdo a la tecnología que se haya seleccionado.
- *Smartphone* con sensores de la tecnología seleccionada.
- Servidor que resuelva las peticiones realizadas por los usuarios.

### 2.5.3.3. Interfaces de Software

- Framework de desarrollo para aplicaciones *Android*.
- Sistema operativo *Android* para los *Smartphone*.
- Controladores de las balizas.

### 2.5.3.4. Interfaces de Comunicación

Los usuarios de la aplicación móvil y el servidor se comunicaran mediante mensajes empleando *web services* SOAP.

## 2.5.4. Requisitos Funcionales

<b>Identificación de caso de uso</b>	CDU01
<b>Nombre</b>	<i>Login</i>
<b>Actores</b>	Usuario a Ubicar
<b>Descripción del caso de uso</b>	1. La App solicitara el ingreso del usuario y contraseña. 2. La App obtendrá los datos del estudiante.
<b>Condiciones previas</b>	
<b>Resultado de la terminación</b>	1. Estudiante logeado.
<b>Excepciones</b>	1. Usuario o contraseña incorrecta. 1.1 Volver a solicitar ingreso.
<b>Asociaciones de casos de uso</b>	

<b>Resumen de entradas</b>	1. Usuario. 2. Contraseña.
<b>Resumen de salidas</b>	1. Ingreso correcto

Tabla 10: Descripción de Caso de Uso - CDU01 (Autoría Propia, 2017).

<b>Identificación de caso de uso</b>	CDU02
<b>Nombre</b>	Iniciar Localización
<b>Actores</b>	Estudiante, Servidor
<b>Descripción del caso de uso</b>	1. El usuario solicita visualizar su ubicación. 2. La App llamara a la función CDU03 "Tomar Referencias". 3. La App armara los valores según una estructura predefinida. 4. La App envía los datos al servidor por <i>web service</i> mediante la función CDU04 "Estimar Ubicación"
<b>Condiciones previas</b>	1. CDU01.
<b>Resultado de la terminación</b>	1. Paquete de datos de las balizas enviados al servidor.
<b>Excepciones</b>	1. Servidor desconectado. 1.1 Cancelar proceso de localización.
<b>Asociaciones de casos de uso</b>	1. CDU03. 2. CDU04.
<b>Resumen de entradas</b>	1. Solicitud de usuario.
<b>Resumen de salidas</b>	1. Información estructurada de las balizas

Tabla 11: Descripción de Caso de Uso - CDU02 (Autoría Propia, 2017).

<b>Identificación de caso de uso</b>	CDU03
<b>Nombre</b>	Tomar Referencias
<b>Actores</b>	Baliza
<b>Descripción del caso de uso</b>	1. La App receptorá la información de conexión proveniente las balizas. 2. La App obtendrá los parámetros necesarios de dicha conexión 3. Enviar los datos de los parámetros al caso de uso CDU02.
<b>Condiciones previas</b>	1. CDU02. 2. Balizas funcionando correctamente.
<b>Resultado de la terminación</b>	1. Datos de la conexión hacia las balizas.
<b>Excepciones</b>	1. Baliza desconectada. 1.1 Continuar con las balizas activas.
<b>Asociaciones de casos de uso</b>	1. CDU02.
<b>Resumen de entradas</b>	1. Solicitud desde CDU02.
<b>Resumen de salidas</b>	Datos de parámetros de conexión

Tabla 12: Descripción de Caso de Uso - CDU03 (Autoría Propia, 2017).

<b>Identificación de caso de uso</b>	CDU04
<b>Nombre</b>	Estimar Posición
<b>Actores</b>	Servidor

<b>Descripción del caso de uso</b>	1. El servidor receptorá la petición por parte de la App. 2. El servidor aplicará el algoritmo en base a los datos proporcionados. 3. El servidor enviará el resultado a la función CD05.
<b>Condiciones previas</b>	1. CDU03.
<b>Resultado de la terminación</b>	1. Resultado de ubicación.
<b>Excepciones</b>	1. Errores en el algoritmo de localización. 1.1 Reintentar proceso de localización.
<b>Asociaciones de casos de uso</b>	1. CDU02. 2. CDU05.
<b>Resumen de entradas</b>	1. Datos de las balizas.
<b>Resumen de salidas</b>	1. Resultado de ubicación.

Tabla 13: Descripción de Caso de Uso - CDU04 (Autoría Propia, 2017).

<b>Identificación de caso de uso</b>	CDU05
<b>Nombre</b>	Enviar Resultados
<b>Actores</b>	Servidor
<b>Descripción del caso de uso</b>	1. El servidor receptorá el resultado proveniente de la función CDU04. 2. El servidor enviará mediante <i>web service</i> el resultado a la App.
<b>Condiciones previas</b>	1. CDU04. 2. App solicitante conectada a la red.
<b>Resultado de la terminación</b>	1. Resultado de ubicación.
<b>Excepciones</b>	1. App desconectada de la red. 1.1 Cancelar proceso de localización.
<b>Asociaciones de casos de uso</b>	1. CDU04. 2. CDU06.
<b>Resumen de entradas</b>	1. Resultado de ubicación.
<b>Resumen de salidas</b>	1. Envió de resultado.

Tabla 14: Descripción de Caso de Uso - CDU05 (Autoría Propia, 2017).

<b>Identificación de caso de uso</b>	CDU06
<b>Nombre</b>	Representar Resultados
<b>Actores</b>	Estudiante
<b>Descripción del caso de uso</b>	1. La App tomará el resultado enviado por el servidor. 2. La App cargará el mapa de la planta. 3. La App interpretará el resultado recibido e identificara la ubicación en el mapa digital 4. La App cargará el nombre de la ubicación lógica. 5. La App colocará el ícono en la zona del plano.
<b>Condiciones previas</b>	1. CDU05. 2. Resultado erróneo.
<b>Resultado de la terminación</b>	1. Ubicación en el plano digital. 2. Nombre de la ubicación simbólica.
<b>Excepciones</b>	1. Resultado erróneo. 1.1 Informar al usuario y dar opción de realizar un reproceso.

<b>Asociaciones de casos de uso</b>	1. CDU05.
<b>Resumen de entradas</b>	1. Recepción del resultado.
<b>Resumen de salidas</b>	1. Ubicación en el plano digital. 2. Nombre de la ubicación simbólica.

Tabla 15: Descripción de Caso de Uso - CDU06 (Autoría Propia, 2017).

### 2.5.5. Requisitos no Funcionales

<b>Identificación del requerimiento</b>	RNF01
<b>Nombre</b>	Interfaz de la App
<b>Descripción</b>	La aplicación presentará una interfaz fácil de entender, la cual muestre un plano intuitivo para el usuario y se use un icono fácil de distinguir
<b>Prioridad</b>	Alta

Tabla 16: Descripción de Requisito No Funcional - RNF01 (Autoría Propia, 2017).

<b>Identificación del requerimiento</b>	RNF02
<b>Nombre</b>	Disponibilidad
<b>Descripción</b>	Mantener disponible al servidor el máximo de tiempo para prestar el servicio de localización, y procurando tiempos cortos de respuesta.
<b>Prioridad</b>	Media/Alta

Tabla 17: Descripción de Requisito No Funcional - RNF02 (Autoría Propia, 2017).

<b>Identificación del requerimiento</b>	RNF03
<b>Nombre</b>	Escalabilidad
<b>Descripción</b>	Diseñar la aplicación móvil para que pueda ser integrada a otras aplicaciones que necesiten el servicio
<b>Prioridad</b>	Media

Tabla 18: Descripción de Requisito No Funcional - RNF03 (Autoría Propia, 2017).

<b>Identificación del requerimiento</b>	RNF04
<b>Nombre</b>	Rendimiento
<b>Descripción</b>	Realizar la localización mediante un algoritmo eficiente en tiempo y carga computacional
<b>Prioridad</b>	Alta

Tabla 19: Descripción de Requisito No Funcional - RNF04 (Autoría Propia, 2017).



## 2.6. Servidor de Aplicaciones

### 2.6.1. Arquitectura

La arquitectura que se utiliza en el sistema de localización en interiores empleando tecnología WiFi consiste en un modelo cliente/servidor de dos capas, donde la primera capa corresponde al nodo terminal que es teléfono móvil de sistema operativo *Android*, y el segundo es el servidor de aplicaciones desarrollado en el lenguaje Java.

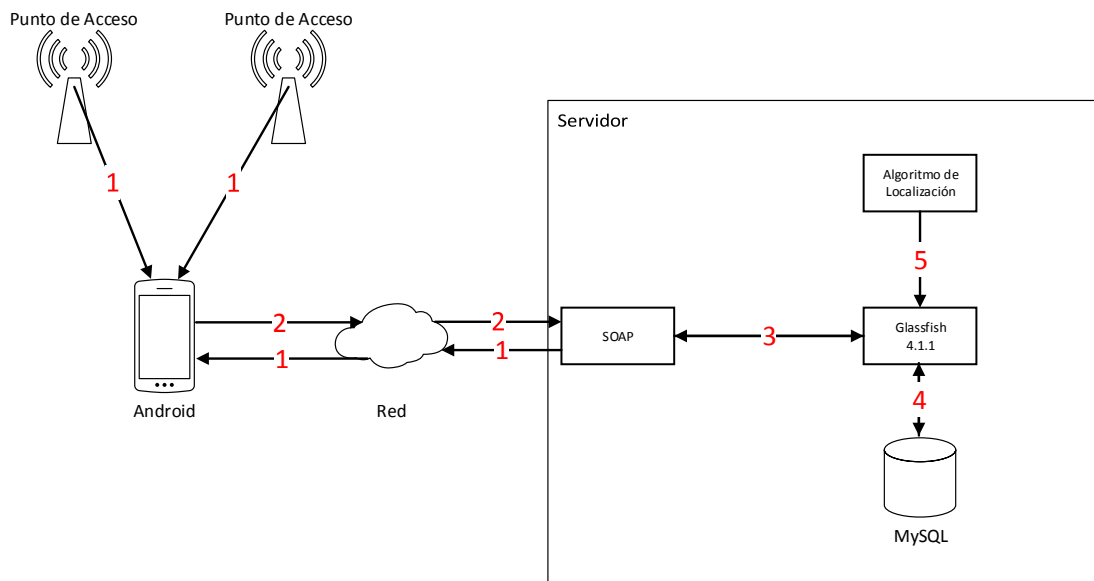


Figura 17: Arquitectura del sistema (Autoría Propia, 2017).

### 2.6.2. Base de Datos

Dentro del servidor de aplicaciones es necesario poseer una base de datos, en la cual se almacene toda la información de los vectores de niveles RSSI, la información de los laboratorios, plantas, facultades, etc.

La base de datos está construida en el gestor *MySQL*, el diagrama entidad/relación se presenta a continuación:

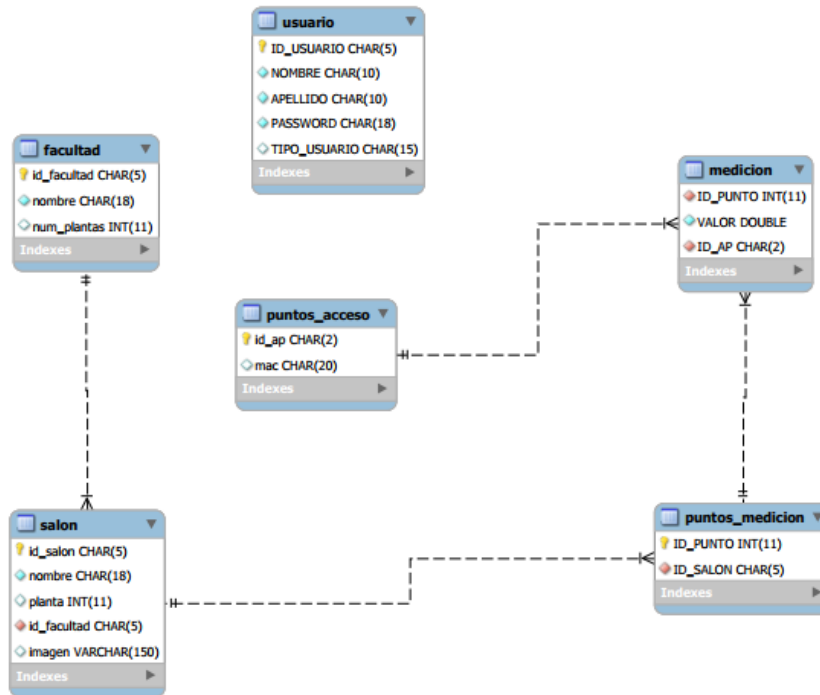


Figura 18: Modelo Entidad/Relación (Autoría Propia, 2017).

### 2.6.3. Servicios Web

La forma de comunicación entre el usuario final y el servidor de aplicaciones es a través de *web services*, el estándar empleado es SOAP. Se poseen dos servicios web, el primero encargado de recibir nombre de usuario y contraseña, para validar la autenticación correcta del usuario. Y el segundo para enviar la cadena correspondiente al vector de niveles RSSI tomado y conocer el posicionamiento del usuario. El WSDL (*Web Services Description Language*) correspondiente al servicio web es el siguiente:

```
<xsd:import namespace=http://localizacion/
schemaLocation="http://localhost:8080/ServiciosWeb/Localizacion?xsd=1"/>
```

**SOAP Request**

```
<?xml version="1.0" encoding="UTF-8"?><S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
<SOAP-ENV:Header/>
<S:Body>
<ns2:login xmlns:ns2="http://localizacion/">
<user>67926</user>
<pass>root</pass>
</ns2:login>
</S:Body>
</S:Envelope>
```

**SOAP Response**

```
<?xml version="1.0" encoding="UTF-8"?><S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
<SOAP-ENV:Header/>
<S:Body>
<ns2:loginResponse xmlns:ns2="http://localizacion/">
<return>Marco;Quezada </return>
</ns2:loginResponse>
</S:Body>
</S:Envelope>
```

Figura 19: Estructura del Servicio Web para Login (Autoría Propia, 2017).

### SOAP Request

```
<?xml version="1.0" encoding="UTF-8"?><S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header/>
  <S:Body>
    <ns2:localizar xmlns:ns2="http://localizacion/">
      <rsi>48:c0:93:1b:4d:60;-40/48:c0:93:1b:6b:50;-90/48:c0:93:1b:6b:60;-90/50:60:28:be:3f:a0;-90/48:c0:93:1b:16:10;-90</rsi>
    </ns2:localizar>
  </S:Body>
</S:Envelope>
```

### SOAP Response

```
<?xml version="1.0" encoding="UTF-8"?><S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header/>
  <S:Body>
    <ns2:localizarResponse xmlns:ns2="http://localizacion/">
      <return>Laboratorio 5 </return>
      <return>4</return>
      <return>Filosofia </return>
      <return>0</return>
    </ns2:localizarResponse>
  </S:Body>
</S:Envelope>
```

Figura 20: Estructura del servicio web para localización (Autoría Propia, 2017).

## 2.7. Aplicación Móvil

La aplicación móvil fue desarrollada para el sistema operativo *Android*, empleando en IDE de desarrollo *Android Studio* en el lenguaje de programación *Java*.

### 2.7.1. Diseño de la aplicación móvil

La aplicación móvil consta de la función principal de localización y visualización, además de las funciones de inicio de sesión y la solicitud de la localización especificando un tiempo de ciclo.

A continuación se presenta un diagrama de clases UML (*Unified Modeling Language*), dentro del cual se establecen las principales clases que componen el desarrollo de la aplicación (Anexo II).

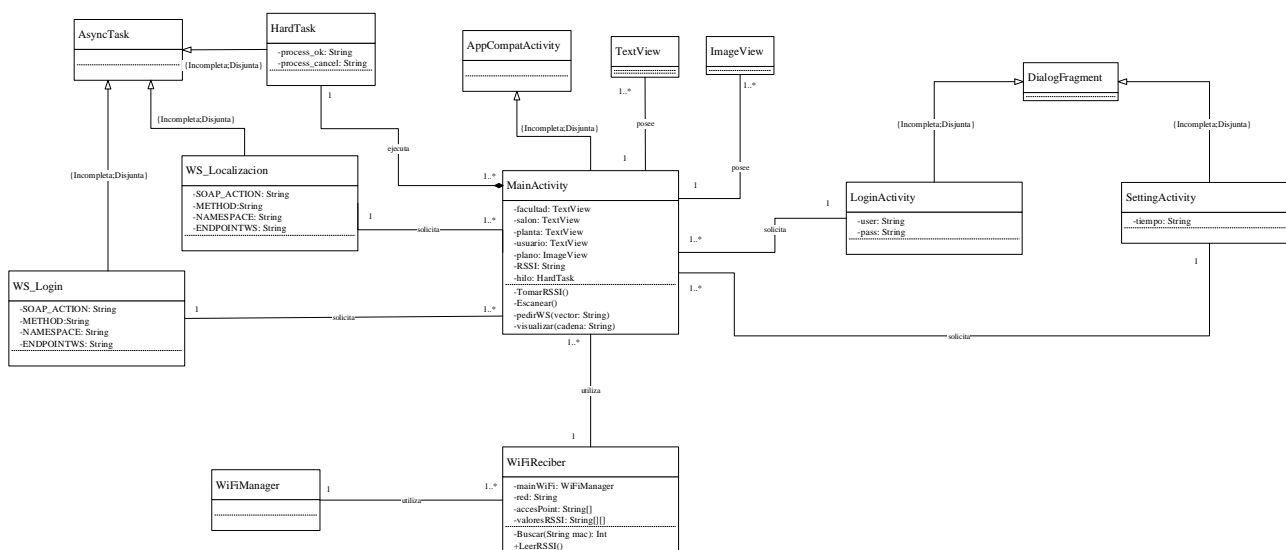


Figura 21: Diagrama de Clases (Autoría Propia, 2017).

### 2.7.2. Login

La aplicación móvil permite realizar un inicio de sesión, ya sea para profesores o estudiantes, o cualquier otro rol de usuario que necesite usar el servicio proveniente de la aplicación. Únicamente es necesario proveer el usuario y contraseña para buscar en la base de datos al usuario. Esta función debe ser realizada solamente al iniciarse por primera vez la aplicación, para las veces siguientes la aplicación guarda los datos del usuario.

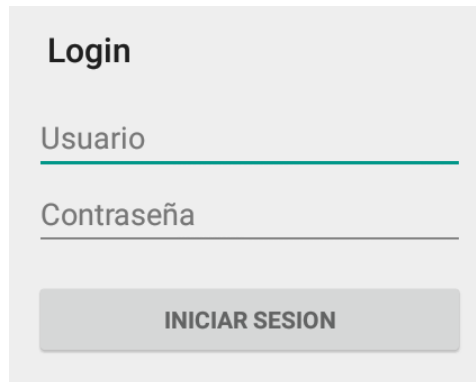


Figura 22: Login (Autoría Propia, 2017).

### 2.7.3. Escaneo de RSSI

Previamente para la localización es necesario obtener los niveles RSSI de los puntos de acceso, este proceso es un escaneo que tarda 8 segundos, tiempo en el cual se estabiliza la captación de la señal WiFi por parte del *Smartphone*.

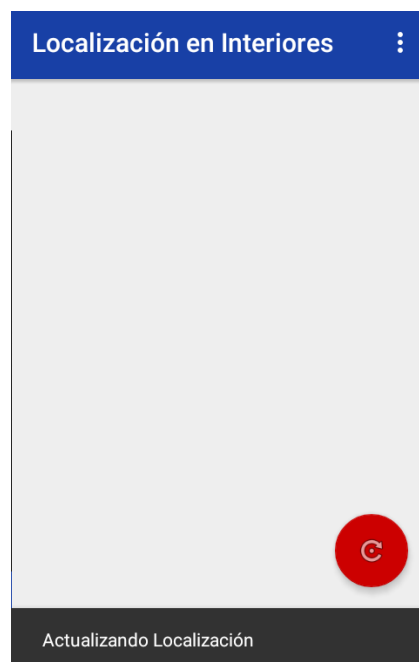


Figura 23: Escaneo de puntos de acceso (Autoría Propia, 2017).

## 2.7.4. Visualización de la localización en interiores

Una vez que se haya realizado el escaneo de los niveles RSSI, se envía el vector generado hacia el servidor, el cual lo tomará para ejecutar el algoritmo de localización y devolver otra cadena con el resultado. A continuación se presentará los posibles resultados:

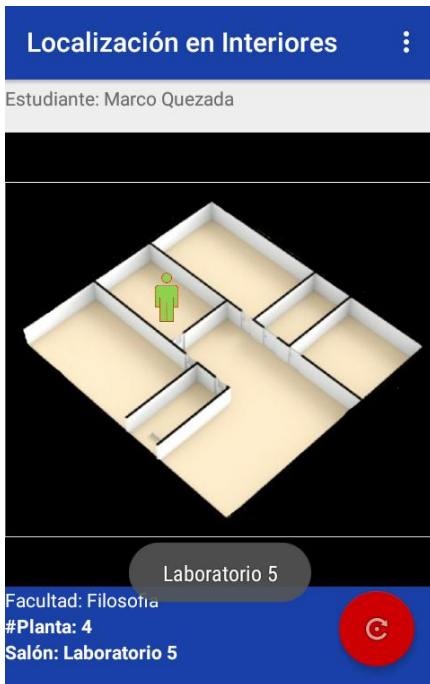


Figura 24: Laboratorio 5 (Autoría Propia, 2017).

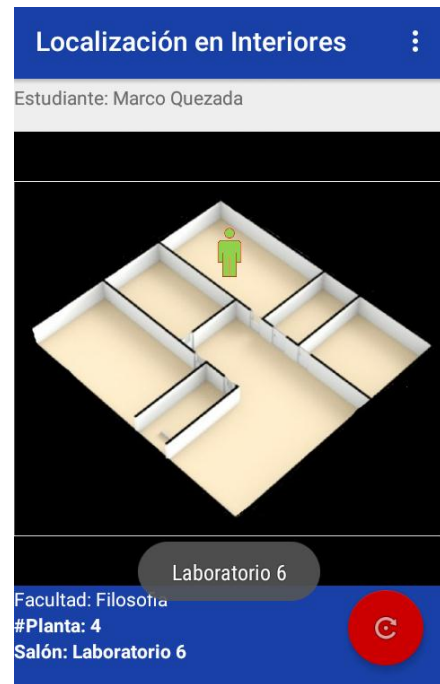


Figura 25: Laboratorio 6 (Autoría Propia, 2017).

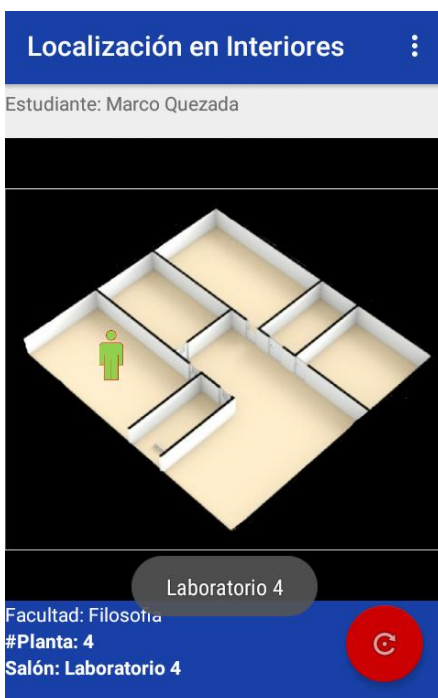


Figura 26: Laboratorio 4 (Autoría Propia, 2017).

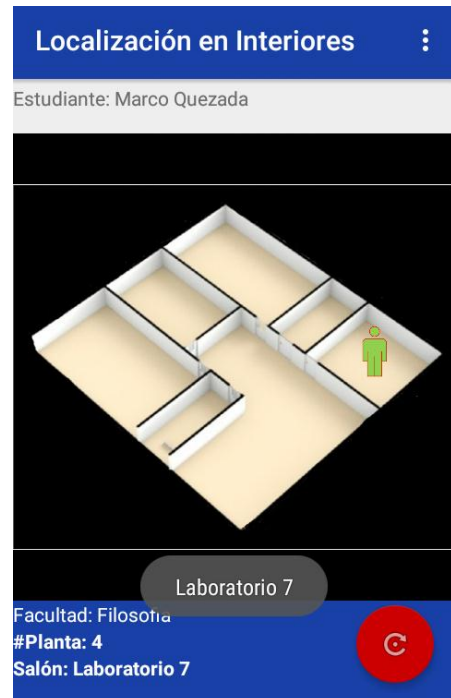


Figura 27: Laboratorio 7 (Autoría Propia, 2017).

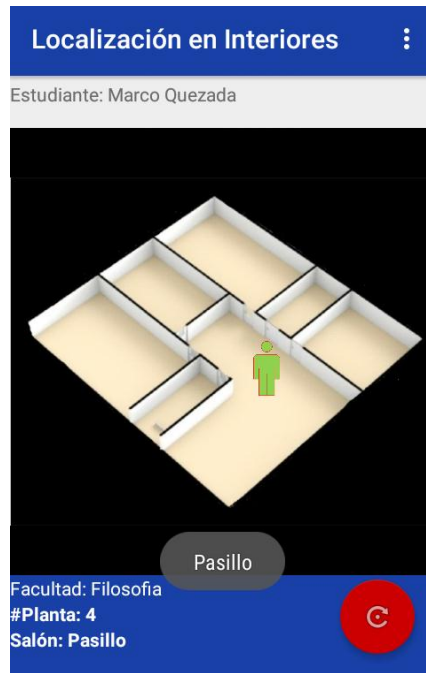


Figura 28: Pasillo (Autoría Propia, 2017).

### 2.7.5. Configuraciones

Dentro de la aplicación móvil también está la opción para configurar el tiempo en el que se solicitara la ubicación, esto se da cuando la aplicación no está activa, llevándolo en un proceso en segundo plano. También se tiene la opción de cambiar de usuario.

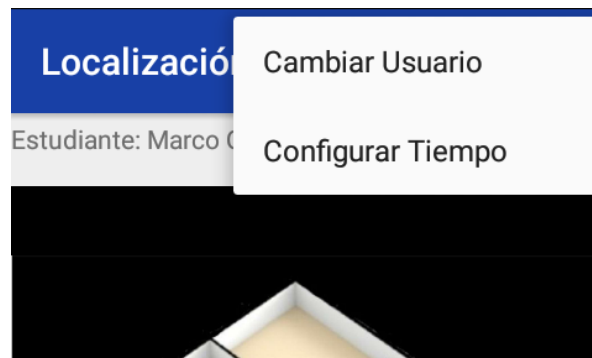


Figura 29: Menú de Configuración (Autoría Propia, 2017).

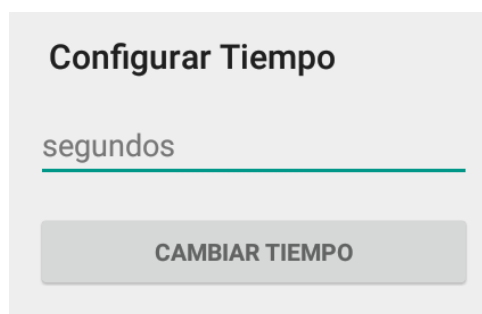


Figura 30: Configuración de tiempo (Autoría Propia, 2017).

En la configuración de usuario la pantalla es la misma que se presentó al inicio en la Figura 22. El botón flotante es para realizar una recarga del servicio, es decir para volver a solicitar la localización.

## 2.8. Conclusiones

Se logró satisfactoriamente la implementación de un sistema de localización de interiores, tomando el escenario con los cuatro laboratorios descritos anteriormente. Obteniendo una respuesta correcta de la ubicación simbólica en cada uno de los laboratorios.

El algoritmo kNN cumple con lo planteado, ya que se llegó a estimar el punto más cercano al vector de potencia tomado en el escaneo. El método de *fingerprint* se logró adaptar correctamente al entorno propuesto, donde los vectores tomados en cada laboratorio marcan una diferencia, la cual conlleva a tener una buena estimación de localización.

Tener cinco puntos de referencia dentro de cada laboratorio es la cantidad base debido al tamaño de los laboratorios, si estos fuesen más grandes se necesitaría tomar más puntos intermedios.

En lo que concierne al pasillo, los resultados fueron que en la parte más amplia a la derecha del espacio vacío (Figura 11) la respuesta es correcta y coherente con la topología utilizada, caso contrario alrededor del espacio vacío la aplicación presenta fallos de precisión.

Tener toda la información de los vectores de potencia almacenados dentro de un servidor externo, como se lo ha realizado dentro de este sistema, es la estrategia más adecuada que tenerla en el teléfono, esto se debe principalmente a la facilidad de modificación de dichos datos, y a que un servidor tiene mucha más capacidad de almacenamiento que un *Smartphone*.

## Capítulo 3

### 3. Tecnología Bluetooth

#### 3.1. Introducción

La segunda tecnología que será analizada para localización en interiores, es la tecnología bluetooth, ya que al igual que WiFi es también ampliamente utilizada en muchos ambientes además de la localización en interiores.

Emplear una red bluetooth resulta muy factible, puesto que los costos de implementación no son demasiado elevados y la mayoría de dispositivos (*Smartphone*, *Laptop*, *Tablet*, etc.) traen incluido una unidad de bluetooth. Esta tecnología será implementada en las mismas condiciones que con WiFi, para al final obtener un prototipo de aplicación móvil.

Se hará uso del *Smartphone* como el medio para realizar la localización, la infraestructura estará implementada con dispositivos emisores de señal bluetooth y todo será realizado en el mismo espacio físico.

#### 3.2. Bluetooth *Low Energy* (BLE)



Figura 31: BLE (areatecnologica, 2017)

Bluetooth es un estándar mediante el cual se puede establecer una red personal (PAN), integrando varios dispositivos, bluetooth permite establecer conexiones de hasta 30m de distancia llegando a tener una transmisión de datos de hasta 24 Mbps, pero estas conexiones necesitan ser continuas y persistentes, lo que implica un mayor consumo de batería del *Smartphone* (Velasco, 2017).



BLE se integra por primera vez en el estándar de Bluetooth 4.0 en el año 2010, BLE tiene como objetivo principal disminuir el consumo de energía, disminuyendo la potencia en la transmisión de la señal de radio. (Velasco, 2017)

BLE es conocido también como Bluetooth *Smart*, trabaja a una frecuencia de 2.4 GHz al igual que el bluetooth clásico. BLE difiere del clásico al trabajar a una tasa de transmisión de 1Mbps consumiendo entre 0.01 y 0.5 vatios.

Utilizar BLE es la mejor opción para la localización en interiores con tecnología bluetooth, puesto que no necesita establecer un vínculo continuo entre emisor y receptor, además debido a la poca potencia únicamente serán visibles los dispositivos más cercanos facilitando de gran manera la localización.

### 3.3. Beacons

#### 3.3.1. Descripción



Figura 32: Beacon (NYVIDA, 2017)

Los *Beacons* son productos desarrollados bajo el estándar BLE, están enfocados para desenvolverse dentro de ambientes de interiores. Estos dispositivos son independientes entre sí, y continuamente envían su UUID (*Universally Unique Identifier*) mediante una señal *broadcast* (señal que reciben todos los dispositivos) en un intervalo de tiempo para dar aviso de su presencia a los dispositivos más cercanos. (Cisco, 2017)

Los *Beacons* funcionan con una batería reemplazable, la cual depende del fabricante, que puede llegar a durar varios meses. Los *Beacons* son adaptables con numerosos equipos como: *Smartphone* (IOS y *Android*), *Smart TV*, relojes, etc.

La funcionalidad de estos dispositivos está dada en dos partes: descubrimiento y conexión. En la primera fase, la de descubrimiento, consiste en un mecanismo donde el *Beacon*, el cual es el dispositivo a ser descubierto, envía señales continuamente cada

cierto tiempo, en un intervalo que puede variar de los 20 ms hasta los 10 segundos dependiendo del modelo y de la configuración. (Cisco, 2017)

Mientras el intervalo de transmisión de señal sea menor, la duración de la batería también será menor, aunque un intervalo corto ayuda a que los *Beacons* sean descubiertos más rápidamente.

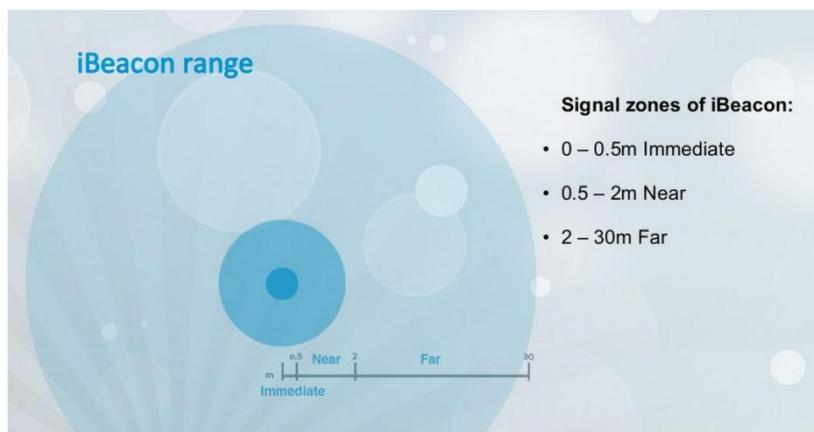


Figura 33: Rango de un Beacon (Cisco, 2017)

Una vez detectado el *Beacon* se continúa con la fase de conexión, en la cual se toma la información que el *Beacon* envía, comúnmente son las características de BLE, de las cuales algunas pueden ser únicamente leídas y otras si pueden ser modificadas.

Entre la información que un *Beacon* puede presentar al usuario están las siguientes:

Característica	Descripción
Batería	Nivel de batería
Nombre	Nombre asociado al dispositivo
UUID	<i>Universal Unique Identifier</i>
Mayor	Identificador del grupo al cual pertenece el <i>Beacon</i>
Minor	Identificador individual de <i>Beacon</i>
<i>Transmission Power</i>	Valor de la potencia para mejorar el alcance
<i>Intervalo Broadcast</i>	Cada cuanto tiempo el <i>Beacon</i> envía su señal
RSSI	Indicador de nivel de potencia de la señal

Tabla 20: Características de los Beacons (NYVIDA, 2017)

De las características presentadas en la Tabla 20 no todas pueden ser escritas o modificadas, como es el caso del UUID, nivel de batería o RSSI, las cuales son únicamente de lectura.

Ya que el sistema de localización se lo realizará con un *Smartphone* es necesario especificar que los *Beacons* son compatibles con la mayoría de teléfonos que han surgido desde el 2013 en adelante, es decir los dispositivos que tienen integrado Bluetooth 4.0 o

superior también son compatibles con BLE y por ende con los *Beacons* también (Cisco, 2017). En los dispositivos *Android* esta funcionalidad se la puede utilizar desde la versión 4.3 del sistema operativo y en los *iPhone* desde la versión 7 de IOS.

El hardware del *Beacon* está constituido por un microcontrolador con un chip de BLE, encargado de enviar la señal, y una batería denominada comúnmente como batería de botón, la cual proporciona desde 240mA hasta 1000mA.



Figura 34: Hardware de un Beacon (NYVIDA, 2017)

Para desarrollar el sistema de localización en interiores con tecnología bluetooth se adquirieron cinco *Beacons* de marca NYVIDA, los cuales estarán distribuidos en el escenario de localización.

### 3.3.2. Especificaciones

Especificación	Descripción
<b>Compatibilidad</b>	Compatible con IOS 7 en adelante y <i>Android</i> 4.3 en adelante
	Utiliza el protocolo de <i>iBeacon</i> de Apple
	Compatible con todos los dispositivos que tengan Bluetooth 4.0
<b>Batería</b>	Modelo CR2477 (reemplazable)
	Duración promedio de 12 meses
	Presenta un luz LED de color roja cuando la batería esta baja
<b>Parámetros de Configuración</b>	UUID
	<i>Major</i>
	<i>Minor</i>
	Nombre de dispositivo
	Contraseña
	<i>Transmission Power</i>
	Intervalo <i>broadcast</i>
<b>Rango/<i>Transmission power</i></b>	8 niveles ajustables, rango de 100ms – 800ms
	Rango de transmisión de poder de -30dBm a +4dBm
	Transmite hasta 100m en espacio abierto
<b>Programación/Modos de conexión</b>	Actualización vía <i>Over-The-Air</i> (sin necesidad del usuario)
	Modo de descubrimiento sin conexión
	Modo de configuración con conexión

Tabla 21: Especificaciones de Beacon (NYVIDA, 2017)

### 3.3.3. Configuración

Para realizar el proceso de configuración de los *Beacons*, el fabricante provee de una aplicación móvil, tanto para *IOS* y *Android*, la cual se llama *BeaconCFG*, dentro de esta aplicación pueden modificarse todos los parámetros editables que presentaron en la Tabla 21.

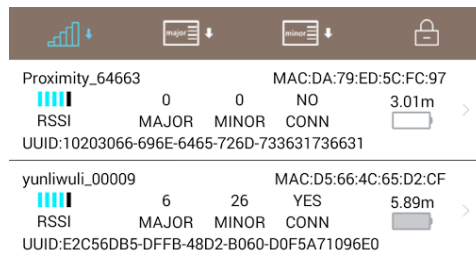


Figura 35: App BeaconCFG (GooglePlay, 2017)

Para realizar cambios de confirmación únicamente se debe seleccionar el *Beacon*, y se abrirá una pantalla con todos los parámetros que pueden ser editados, y después seleccionar *Save* para que se guarden los cambios.

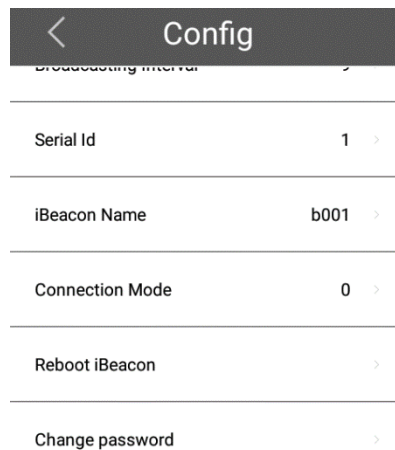


Figura 36: Pantalla de Configuración (Autoría Propia, 2017).

Los *Beacons* llegan por defecto con un nombre que corresponde a su número de serie, antepuesto por el texto “MiniBeacon”, por lo que lo primero que se necesitó configurar es el nombre de cada uno de los dispositivos.

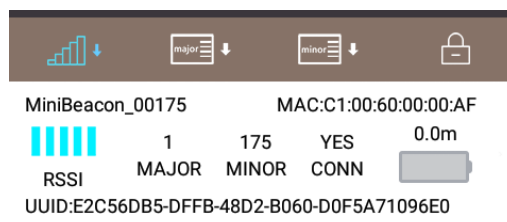


Figura 37: Parámetros por defecto (Autoría Propia, 2017).

Los *Beacons* serán nombrados con la letra b minúscula, seguido de una numeración según la cantidad de *Beacons*, es decir cómo se tiene cinco *Beacons* los nombre serian: b001, b002,..., b005.

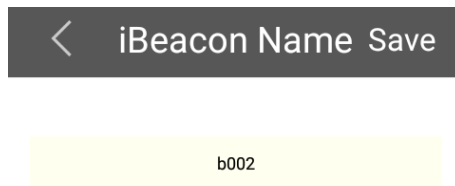


Figura 38: Configuración de Nombre (Autoría Propia, 2017).

A continuación la siguiente configuración fue la del Serial ID, este parámetro es un identificador numérico asignado a cada *Beacon*, al tener cinco *Beacons*, el valor de este campo será del 1 al 5 respectivamente para cada *Beacon*.

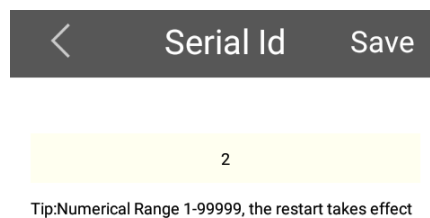


Figura 39: Configuración de Serial ID (Autoría Propia, 2017).

Finalmente la última configuración, es la configuración del intervalo *broadcast*, por defecto en el *Beacon* está establecido un intervalo de 800ms, para el desarrollo del sistema se decidió configurar un intervalo de 300ms, para que el *Smartphone* pueda detectar más rápido a los *Beacons*.

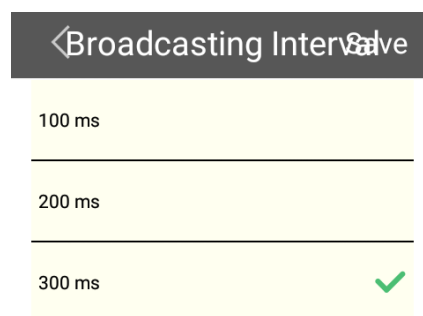


Figura 40: Configuración de Broadcasting Interval (Autoría Propia, 2017).

Después de realizarse todas las configuraciones se concluye reiniciando al *Beacon*, el reinicio se lo realiza ingresando la contraseña, la cual para este caso es “nyvida20”. Esta contraseña viene por defecto pero también puede ser modificada.

## < Reboot iBeacon Save

Please enter a new password

Tip: reboot iBeacon after modify

Figura 41: Reinicio de Beacon Autoría Propia, 2017).

Name	MAC	RSSI	MAJOR	MINOR	CONN	Distance
b00500005	MAC:C1:00:60:00:A2	1	162	YES	0.0m	
b00400004	MAC:C1:00:60:00:6B	1	107	YES	0.0m	
b00200002	MAC:C1:00:60:00:AF	1	175	YES	0.0m	

Figura 42: Beacons Configurados (Autoría Propia, 2017).

### 3.4. Infraestructura de Localización

El espacio físico donde se implementó el sistema localización, es al igual que con la tecnología WiFi, en el último piso de la facultad de Filosofía en la Universidad del Azuay, donde existen 7 laboratorios, 4 oficinas y un baño.



Figura 43: Infraestructura (Autoría Propia, 2017).

Se desarrollaron dos escenarios para este sistema con tecnología bluetooth, en ambos se empleó el mismo algoritmo de localización, que será descrito posteriormente y en ambos escenarios se utilizarán los cinco *Beacons* adquiridos.

El primer escenario consiste únicamente en el pasillo completo del piso, en el cual se implementó el sistema para mediante los *Beacons* diferenciar si se está presente en las zonas izquierda, central o derecha.

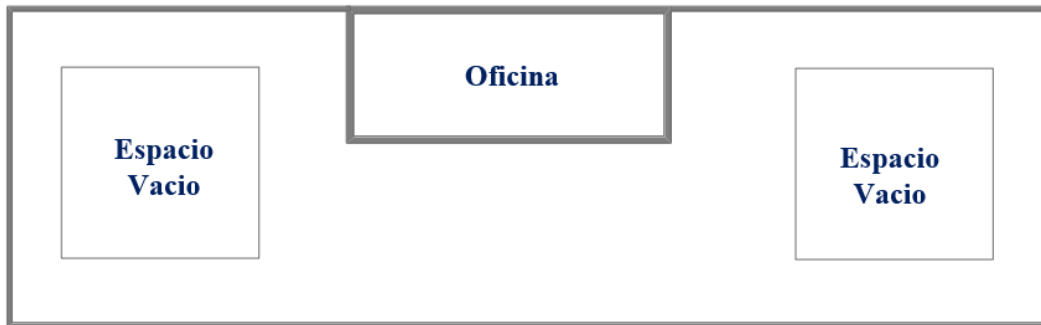


Figura 44: Primer Escenario (Autoría Propia, 2017).

En el segundo escenario se tomó únicamente la zona izquierda del pasillo, y además se incluye dos laboratorios, los laboratorios número cuatro y cinco.

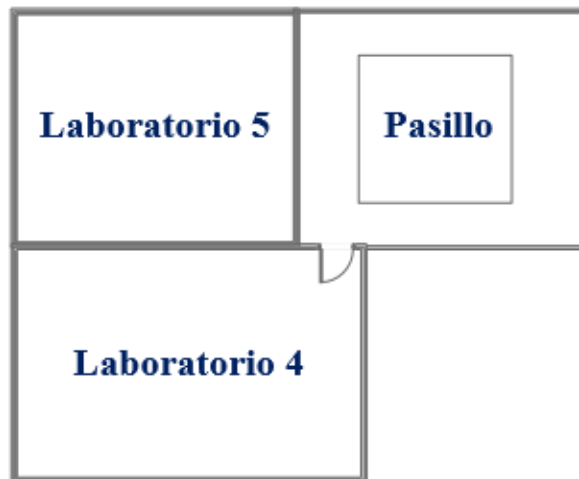


Figura 45: Segundo Escenario (Autoría Propia, 2017).

El posicionamiento de los *Beacons*, para ambos escenarios, está realizado en la forma de triángulos, teniendo un *Beacon* en cada vértice, para el primer escenario existe un *Beacon* en la zona central y dos *Beacons* en la zona izquierda y derecha.

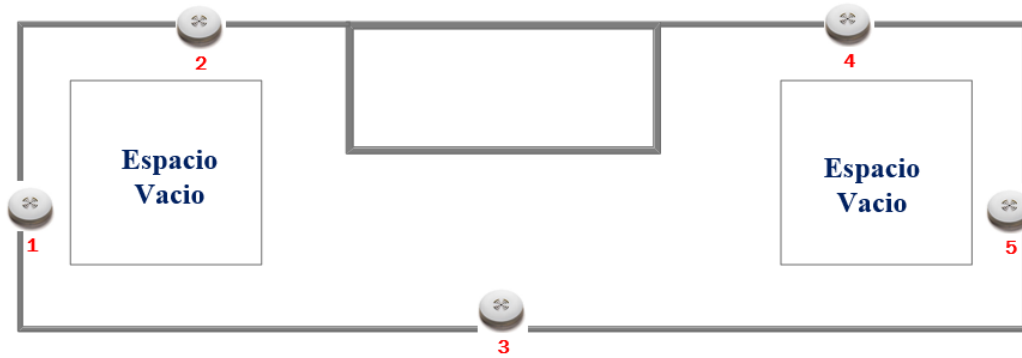


Figura 46: Distribución de Beacons del primer escenario (Autoría Propia, 2017).

En el segundo escenario ya que se tiene solo el pasillo del lado izquierdo, se necesitó solo tres *Beacons* para realizar la localización en el pasillo, igualmente estos tres *Beacons* forman un triángulo para poder localizar al usuario. Los otros dos *Beacons* están instalados en cada uno de los laboratorios 4 y 5, para que el sistema realice una diferenciación de si se está presente en el pasillo o si ya se ingresó dentro de uno de los laboratorios.

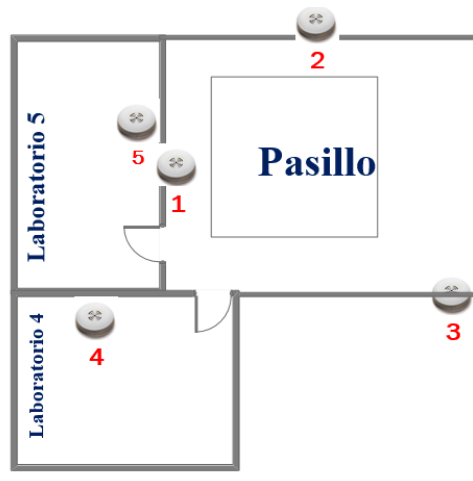


Figura 47: Distribución de Beacons en escenario dos (Autoría Propia, 2017).

### 3.5. Método de Localización

El método con el cual se implementó el sistema de localización con tecnología bluetooth es el de trilateración, este método consiste en obtener las coordenadas X-Y dentro de un plano cartesiano establecido anteriormente. Para conseguir determinar las coordenadas es necesario conocer cuál es la distancia en metros desde el *Smartphone* hacia cada *Beacon*, por ello primero se debe calcular dicha distancia antes de realizar la trilateración.



### 3.5.1. Cálculo de Distancia entre *Smartphone* y *Beacon*

La estimación de la distancia se la realiza teniendo como indicador el RSSI obtenido desde el *Smartphone*, una vez tenido ese valor es necesario tener también el valor de tres coeficientes fijos y un valor de referencia. La ecuación para determinar la distancia es la siguiente:

$$d = A * \left(\frac{r}{t}\right)^B + C$$

Ecuación 3: Calculo de distancia en metros (Networks, 2017)

Donde:

d = distancia en metros.  
A, B y C = coeficientes.  
r = valor RSSI.  
t = valor referencial.

Los coeficientes que la Ecuación 3 requiere están directamente relacionados con el modelo del *Smartphone*, por lo que los coeficientes variarán, esto se da principalmente en los *Smartphone* con sistema operativo *Android*, ya que los modelos son diferentes en términos de hardware, teniendo también diferente modelo de antena bluetooth.

Debido a estas diferencias, es necesario realizar un cálculo previo de dichos coeficientes, para poder realizar este cálculo es necesario un valor referencial, el cual es el valor RSSI medido a la distancia de un metro con un *iPhone 5s* (Networks, 2017).

El cálculo se lo realiza en seis pasos, partiendo de un conjunto de datos de los valores RSSI en un rango de distancias desde los 0.25m hasta los 40m (Networks, 2017), pero para el caso de este sistema considerando que el espacio es pequeño, se decidió realizar la medición hasta 12m.

El *Smartphone* empleado para este sistema es el mismo que se utilizó con la tecnología WiFi, *Samsung J1*, y es el *Smartphone* con el cual se realizó la toma de valores RSSI para todas las distancias.

Con respecto al valor que se utiliza como referencia, el cual representa el valor RSSI medido a un metro de distancia con un teléfono *iPhone 5s*, será de -41dBm<sup>9</sup> (Networks, 2017), el cual será utilizado tanto para obtener los coeficientes como para obtener la distancia hacia los *Beacons*.

---

<sup>9</sup> dBm el acrónimo de decibelios por milivatio

El primer paso como se menciono es obtener las mediciones RSSI desde distintas distancias entre *Beacon* y *Smartphone*, los resultados están en la Tabla 22.

Distancia (m)	RSSI (dBm)
0,25	-25
0,5	-27
1	-29
2	-30
3	-32
4	-33
5	-33
6	-37
7	-38
8	-41
9	-42
10	-41
12	-45

Tabla 22: Mediciones RSSI (Autoría Propia, 2017).

El segundo paso consiste obtener el resultado de la división entre de cada RSSI medido para el valor referencial de -41dBm, a este valor se lo denomina Ratio.

$$ratio = (RSSI)/-41$$

Ecuación 4: Cálculo de Ratio (Networks, 2017).

iPhone 1m RSSI	-41dBm	
Distancia (m)	RSSI (dBm)	Ratio
0,25	-25	0,609756098
0,5	-27	0,658536585
1	-29	0,707317073
2	-30	0,731707317
3	-32	0,780487805
4	-33	0,804878049
5	-33	0,804878049
6	-37	0,902439024
7	-38	0,926829268
8	-41	1
9	-42	1,024390244
10	-41	1
12	-45	1,097560976

Tabla 23: Cálculo de Ratio para cada RSSI

El tercer paso consiste en realizar un proceso de regresión con los datos de ratio de la Tabla 23 como datos independientes y los datos RSSI de la Tabla 22 como datos

dependientes, este proceso se lo realizó en una herramienta web (<http://www.xuru.org/rt/powr.asp>) (Networks, 2017), cuyos resultados corresponden a las variables A y B de la Ecuación 3.

	<b>x</b>	<b>y</b>	<b>Calculated y</b>	<b>Error</b>
1.	0.609756098	0.25	4.508686676·10 <sup>-1</sup>	2.008686676·10 <sup>-1</sup>
2.	0.658536585	0.5	7.312410314·10 <sup>-1</sup>	2.312410314·10 <sup>-1</sup>
3.	0.707317073	1	1.145663407	1.456634074·10 <sup>-1</sup>
4.	0.731707317	2	1.417644212	-5.823557882·10 <sup>-1</sup>
5.	0.780487805	3	2.126568148	-8.734318519·10 <sup>-1</sup>
6.	0.804878049	4	2.580172201	-1.419827799
7.	0.804878049	5	2.580172201	-2.419827799
8.	0.902439024	6	5.294811257	-7.051887432·10 <sup>-1</sup>
9.	0.926829268	7	6.260694657	-7.393053433·10 <sup>-1</sup>
10.	1	8	10.09188024	2.09188024
11.	1.024390244	9	11.74165296	2.741652961
12.	1	10	10.09188024	9.188024006·10 <sup>-2</sup>
13.	1.097560976	12	18.11324583	6.113245833

**Calculate**

**Result: y = 10.09188024 x<sup>6.283270601</sup>**

Residual Sum of Squares: **rss = 59.40524154**

Coefficient of Determination: **R<sup>2</sup> = 6.629122043·10<sup>-1</sup>**

Figura 48: Resultados de Regresión (Autoría Propia, 2017).

Como se aprecia en la Figura 47, el valor de los coeficientes son: A = 10,09188024 y B = 6,629122067, esto es importante porque son los valores con los que se configuró la aplicación.

El cuarto paso consiste en emplear la Ecuación 5, con la cual se puede tener una aproximación a la distancia, empleado los coeficientes A y B, los resultados permitirán obtener el valor del coeficiente C.

$$d = A * \left(\frac{r}{t}\right)^B$$

Ecuación 5: Distancia calculada con A y B (Networks, 2017).

Donde:

d = distancia en metros.

A y B = coeficientes.

r = valor RSSI.

t = valor referencial.

<b>RSSI (dBm)</b>	<b>Ratio</b>	<b>Distancia Real (m)</b>	<b>Distancia Aproximada (m)</b>
-25	0,609756098	0,25	0,3799671
-27	0,658536585	0,5	0,63287239
-29	0,707317073	1	1,016356041
-30	0,731707317	2	1,272471721
-32	0,780487805	3	1,951883845
-33	0,804878049	4	2,393565421
-33	0,804878049	5	2,393565421
-37	0,902439024	6	5,110126943
-38	0,926829268	7	6,098307729
-41	1	8	10,09188024
-42	1,024390244	9	11,83991884
-41	1	10	10,09188024
-45	1,097560976	12	18,70589924

Tabla 24: Calculo de distancia aproximada con A y B (Autoría Propia, 2017).

El quinto paso es en el cual se obtiene el valor del coeficiente C, el cual consiste en una variable de intersección, ya que el modelo de regresión asume una intersección de cero, esta variable optimiza la estimación a un metro, el cálculo se lo realiza restando 1 menos la distancia aproximada de 1m, en la Tabla 24 se observa que el valor aproximado a un metro es de: 1,016356041, por lo tanto el valor de  $C = (1 - 1,016356041)$ , es decir  $= -0,016356041$ .

Finalmente en el sexto paso se obtiene nuevamente las distancias aproximadas, solo que esta vez contando con los 3 coeficientes y la Ecuación 3.

<b>RSSI (dBm)</b>	<b>Ratio</b>	<b>Distancia Real (m)</b>	<b>Distancia Aproximada (m)</b>
-25	0,609756098	0,25	0,36361106
-27	0,658536585	0,5	0,61651635
-29	0,707317073	1	1
-30	0,731707317	2	1,25611568
-32	0,780487805	3	1,9355278
-33	0,804878049	4	2,37720938
-33	0,804878049	5	2,37720938
-37	0,902439024	6	5,0937709
-38	0,926829268	7	6,08195169
-41	1	8	10,0755242
-42	1,024390244	9	11,8235628
-41	1	10	10,0755242
-45	1,097560976	12	18,6895432

Tabla 25: Calculo de distancia aproximada con los tres coeficientes (Autoría Propia, 2017).

### 3.5.2. Trilateración

Como se mencionó en el primer capítulo la trilateración es un método en el cual se estima la posición de una coordenada X-Y, esto se lo hace conociendo las distancias a las cuales se encuentra cada *Beacon* con respecto al punto a localizarse, es decir el *Smartphone*.

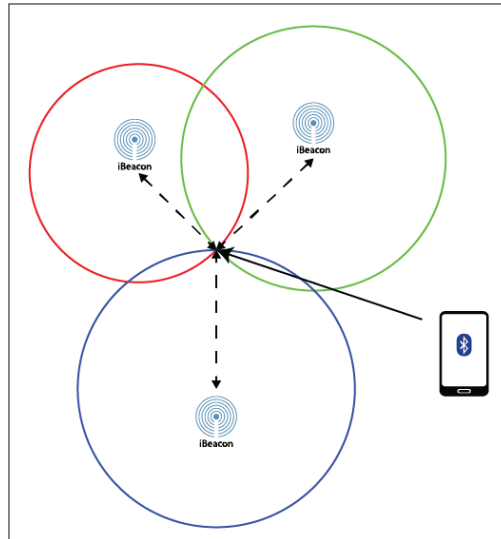


Figura 49: Trilateración con Beacons (iArtHis\_LAB, 2017).

En ambos escenarios se aplicó la trilateración, en el primer escenario con los cinco *Beacons* (Figura 46), y en el segundo escenario con tres *Beacons* (Figura 47). Pero para la aplicación del algoritmo es necesario previamente establecer un plano cartesiano con el cual se pueda llegar a estimar la localización y para establecer las coordenadas de cada *Beacon*.

El plano cartesiano fue establecido para toda la zona del pasillo, la cual mide 5.6m por 17.8m, también se consideró la oficina intermedia y los espacios vacíos, además este plano cartesiano es utilizado principalmente para el primer escenario.

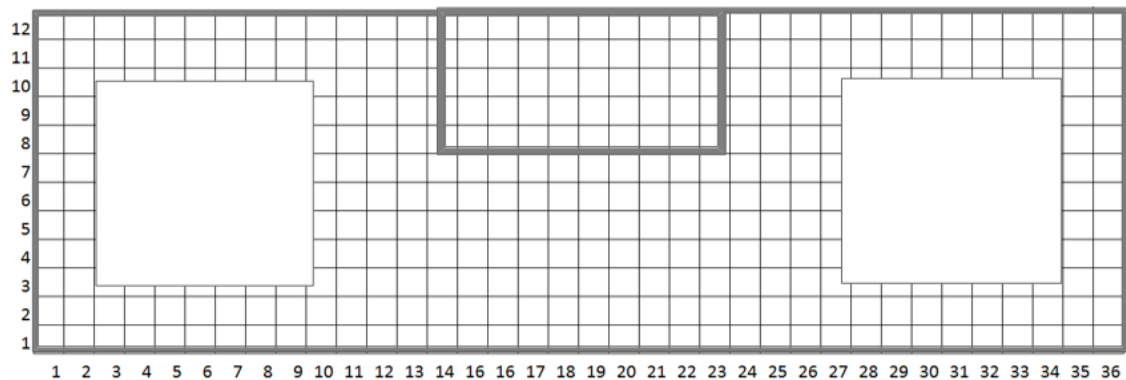


Figura 50: Plano de coordenadas de primer escenario (Autoría Propia, 2017).

En el segundo escenario al trabajar únicamente con la zona izquierda del pasillo cuyas medidas son de 5.6m x 6m, también se tomara únicamente la parte izquierda del plano cartesiano.

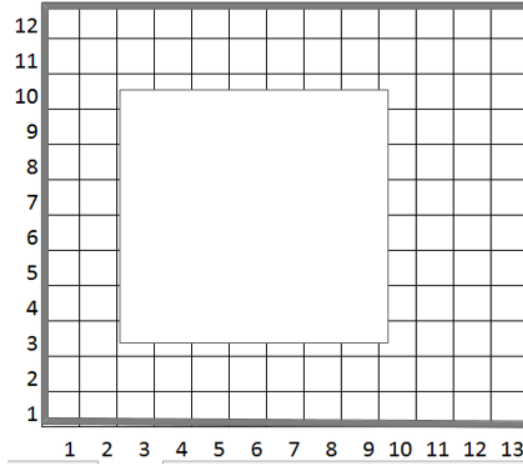


Figura 51: Plano de coordenadas de segundo escenario (Perez, 2013).

Para realizar el cálculo de las coordenadas mediante la trilateración se necesita conocer las coordenadas fijas de los *Beacons*. Para mayor facilidad al identificar a los *Beacons*, estos fueron etiquetados con las letras A, B y C.

La trilateración se la realiza empleando las distancias con respecto a los *Beacons*, tanto como sus coordenadas fijas, por lo que se necesita tener almacenadas todas las coordenadas y previamente haber obtenido las distancias con respecto a todos los *Beacons* que se detecten.

$$Va = db^2 - dc^2$$

Ecuación 6: Diferencia de distancias de B y C (Perez, 2013).

$$Vb = db^2 - da^2$$

Ecuación 7: Diferencia de distancias de B y A (Perez, 2013).

Las Ecuaciones 6 y 7 son agrupadas en dos variables para facilitar la resolución de las Ecuaciones 8 y 9.

$$y = \frac{Vb(xb - xc) - Va(xb - xa)}{(ya - yb)(xb - xc) - (yc - yb)(xb - xc)}$$

Ecuación 8: Calculo de coordenada y (Perez, 2013)

$$x = \frac{y(ya - yb) - Vb}{xb - xc}$$

Ecuación 9: Calculo de coordenada x (Perez, 2013)

Los *Beacons* llevan una nomenclatura de A, B y C para poder facilitar el cálculo, y para tener una correcta referenciación a las coordenadas X-Y en las Ecuaciones 8 y 9.

<b>Beacon</b>	<b>Nomenclatura</b>
b001	C
b002	B
b003	A
b004	C
b005	B

Tabla 26: Nomenclaturas de cada Beacon (Autoría Propia, 2017).

Las nomenclaturas de los *Beacons* b001, b002 y b003 descritos en la Tabla 26 aplican para ambos escenarios, pero los *Beacons* b004 y b005 descritos como C y B respectivamente, aplican solo para el primer escenario. En el segundo escenario los *Beacons* b004 y b005 estarán situados en los laboratorios 4 y 5 respectivamente, por lo que no es necesario darles una nomenclatura, ya que dentro de los laboratorios no se realizará la trilateración.

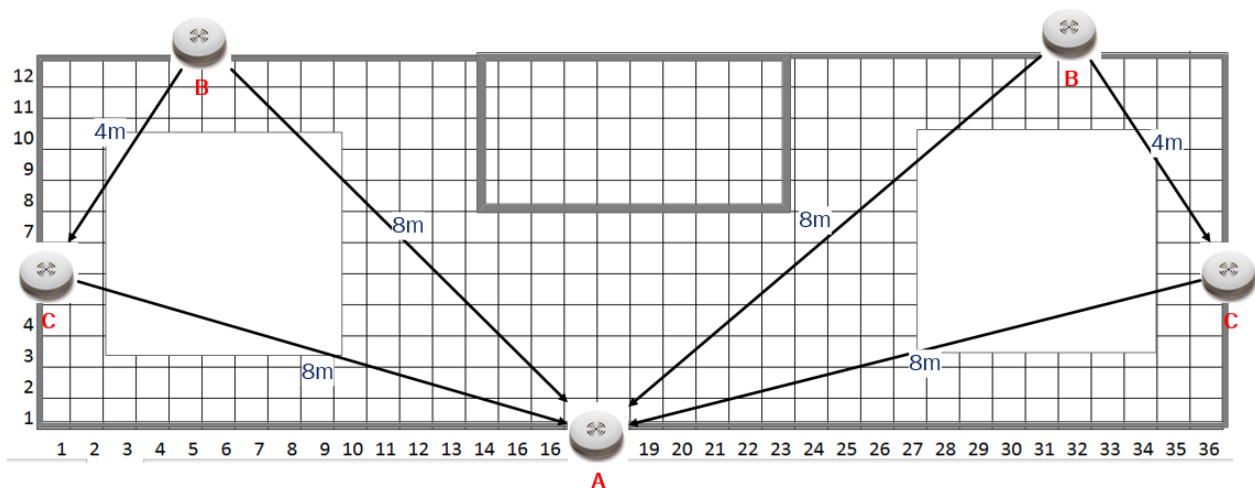


Figura 52: Distancias reales entre Beacons del primer escenario (Autoría Propia, 2017).

También es necesario conocer las distancias reales que existen entre los *Beacons*, obviamente no las distancias entre todos los dispositivos, únicamente las distancias que conforman los lados de los triángulos que permitirán realizar la trilateración, como se muestra en la Figura 52 (primer escenario).

En el segundo escenario únicamente se cuenta con un triángulo, el de la izquierda del primer escenario (Figura 52), variando únicamente las medidas con respecto al punto A en un metro.

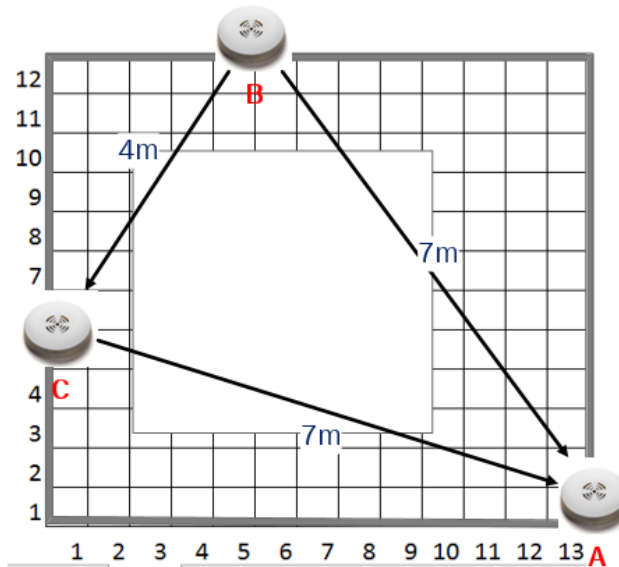


Figura 53: Distancias reales entre Beacons del segundo escenario (Autoría Propia, 2017).

Los valores de las coordenadas X-Y de los *Beacons* en ambos escenarios se describen a continuación en la Tabla 27.

<i>Beacons</i>	Primer escenario		Segundo escenario	
	X	Y	X	Y
b001 (C)	0	6	0	6
b002 (B)	5	12	5	12
b003 (A)	17	0	13	0
b004 (C)	36	6	-	-
b005 (B)	32	12	-	-

Tabla 27: Coordenadas de Beacons (Autoría Propia, 2017).

### 3.6. Algoritmo de Localización

El algoritmo que se implementó para ambos escenarios es muy parecido, únicamente varía al inicio, ya que en el escenario uno se debe determinar a cuál de los dos triángulos el usuario está más cerca para realizar los cálculos en base a los *Beacons* que conformen dicho triángulo.

Para el segundo escenario al estar solamente en el lado izquierdo esta clasificación ya no es necesaria y se toman los tres *Beacons* por defecto. Pero en este segundo escenario están presentes los laboratorios, así que al final si habría una clasificación, ya no para determinar la zona en la cual está, sino para determinar si se halla en el pasillo o dentro de uno de estos dos laboratorios.



### Primer Escenario:

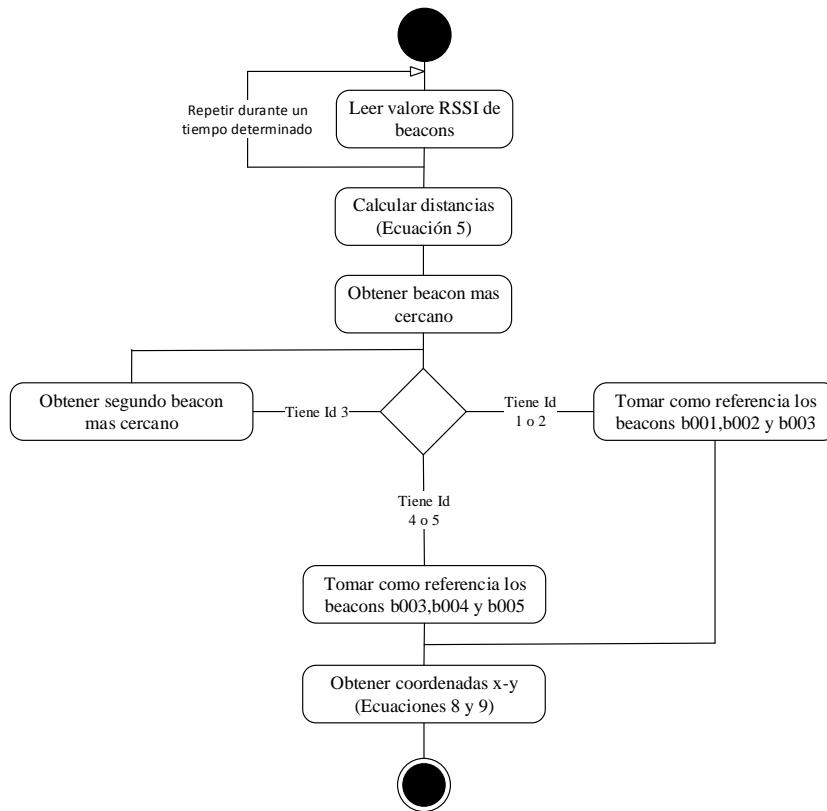


Figura 54: Diagrama de Flujo del algoritmo del primer escenario (Autoría Propia, 2017).

### Segundo Escenario:

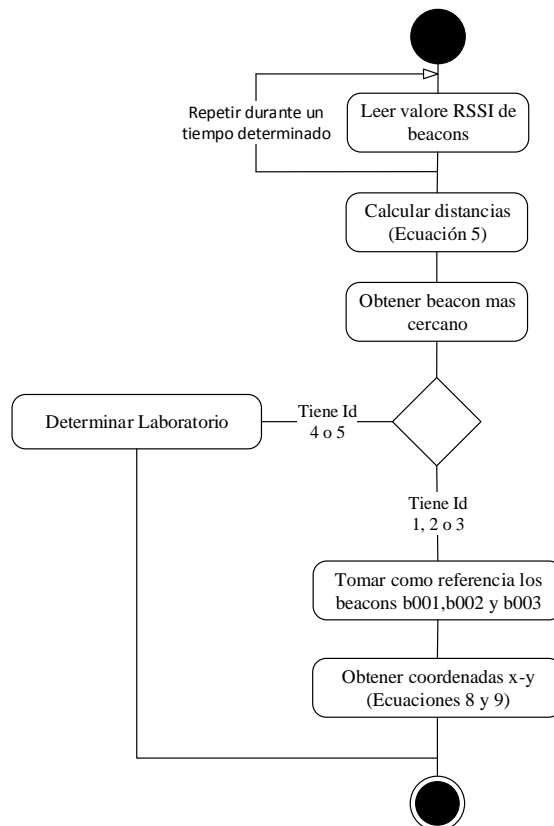


Figura 55: Diagrama de Flujo del algoritmo del segundo escenario (Autoría Propia, 2017).

## **3.7. Levantamiento de Requerimientos de Software**

### **3.7.1. Introducción**

En el presente documento se detallarán las especificaciones del sistema de localización en interiores con tecnología bluetooth, las cuales servirán como base para un futuro desarrollo del mismo. Este documento también establecerá el alcance que tendrá el software.

En las secciones anteriores se establecieron dos escenarios de localización mediante *Beacons*, de los dos escenarios se decidió desarrollar el sistema tomando el segundo escenario, puesto que en este se incluyen los laboratorios 4 y 5, y se presenta para una clasificación entre pasillo y laboratorio, la cual estará acompañada al proporcionarse información contextual tras haber ingresado en uno de los laboratorios.

El análisis se lo presentará empleando el Lenguaje de Modelado Unificado (UML), haciendo uso del diagrama de casos de uso y siguiendo el estándar IEEE 830.

#### **3.7.1.1. Propósito**

El propósito de este documento es analizar mediante diagramación UML la funcionalidad del sistema y de la aplicación móvil, la cual en términos generales consistiría en estimar la localización leyendo los niveles RSSI de los *Beacons*, cargar un plano de la zona de cobertura y proporcionar información contextual.

#### **3.7.1.2. Ámbito del Sistema**

El software tendrá por nombre de “Prototipo de Aplicación Móvil para Localización en Interiores”. La aplicación está contemplada para proporcionar al usuario información respecto a su ubicación, ya sea en el pasillo o de lo contrario en el laboratorio que se encuentre, especificando el nombre del lugar, e información contextual sobre la actividad que se está desarrollando (esto solo de estar en un laboratorio) e información descriptiva del lugar (tanto para el pasillo o laboratorio).

La aplicación tiene varios beneficios, entre los que están: poder mejorar la localización en interiores, para la cual un sistema GPS tiene muy poca precisión, llegar a tener mediante consultas desde el *Smartphone* el laboratorio o las coordenadas dentro del pasillo donde esté ubicado, y aportar al desarrollo de un futuro proyecto.

### 3.7.1.3. Definiciones, Acrónimos y Abreviaturas

<b>Nombre</b>	<b>Descripción</b>
Usuario	Persona que usará la aplicación móvil
ERS	Especificación de Requisitos Software
RF	Requerimiento Funcional
RNF	Requerimiento No Funcional
ILS	<i>Indoor Location System</i> (Sistema de Localización en Interiores)
<i>Beacon</i>	Dispositivo que maneja señal de bluetooth LE

Tabla 28: Definiciones, Acrónimos y Abreviaturas (Autoría Propia, 2017).

### 3.7.1.4. Referencias

<b>Título del Documento</b>	<b>Referencia</b>
Standard IEEE 830 - 1998	IEEE

Tabla 29: Referencias (Autoría Propia, 2017).

### 3.7.1.5. Visión general del documento

El documento posee tres partes. En la primera de ella se da una breve introducción del documento, brindando una visión general sobre el software que se está tratando.

La segunda parte del documento consiste en una descripción generalizada sobre el sistema, facilitando mediante diagramas las diversas funciones que la aplicación posee, algunas restricciones, supuestos y factores ligados directamente con el desarrollo.

La última parte de documento es la cual consiste en la definición detallada de todos los requerimientos que el sistema deberá cumplir.

## 3.7.2. Descripción General

### 3.7.2.1. Perspectiva del Producto

El prototipo para el ILS consiste en una aplicación para dispositivos móviles, dicho prototipo está enfocado para trabajar en el sistema operativo *Android*. El sistema de localización estará implementado en una arquitectura de dos capas. La capa del servidor que se encargará de recibir los datos enviados por el *Smartphone*, procesar la ubicación y retornar el resultado. Y la capa del usuario o del dispositivo móvil, el cual tendrá las funciones de toma de datos, envío al servidor y representación del resultado.

### 3.7.2.2. Funciones del Producto

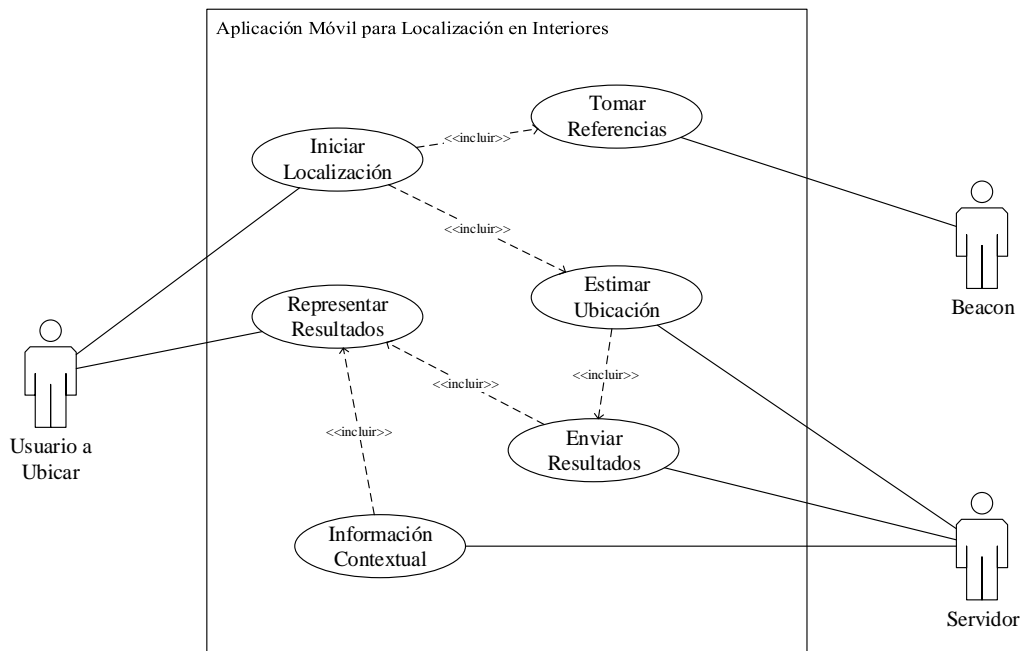


Figura 56: Diagrama de Casos de Uso (Autoría Propia, 2017).

### 3.7.2.3. Características de los Usuarios

<b>Tipo de usuario</b>	Estudiante/Profesor
<b>Formación</b>	Conocimiento en manejo de aplicaciones móviles
<b>Actividades</b>	Solicitar ubicación

Tabla 30: Características de Usuario (Autoría Propia, 2017).

### 3.7.2.4. Restricciones

- La aplicación necesita estar en conectada a Internet para ser usada.
- Lenguajes de programación: JAVA.
- El sistema se diseñará según un modelo cliente/servidor.
- El sistema deberá tener un diseño de interfaz sencilla.
- Utilizar distintas librerías de acuerdo al tipo de *Beacon* que se emplee.

### 3.7.2.5. Suposiciones y Dependencias

- Estabilidad en los requerimientos planteados en este documento.
- Los *Beacon* que se adquieran estén en óptimas condiciones.
- Que el método de localización que se haya adoptado sea la más adecuada.

### 3.7.3. Requisitos Específicos

#### 3.7.3.1. Interfaces de Usuario

La interfaz de la aplicación móvil estará contenida principalmente por el plano de la planta en la cual se implementó la localización en interiores, dentro de dicho plano existirá también un icono que represente la figura del usuario. Y botones para solicitar la información contextual del pasillo o laboratorio al que se haya ingresado.

#### 3.7.3.2. Interfaces de Hardware

- *Beacons* con el estándar Bluetooth LE.
- *Smartphone* con Bluetooth 4.0 en adelante.
- Servidor que resuelva las peticiones realizadas por los usuarios.

#### 3.7.3.3. Interfaces de Software

- Framework de desarrollo para aplicaciones *Android*.
- Sistema operativo *Android* para los *Smartphone*.
- App de configuración de los *Beacons*.

#### 3.7.3.4. Interfaces de Comunicación

Los usuarios de la aplicación móvil y el servidor se enviarán mensajes empleando *web services SOAP*.

### 3.7.4. Requisitos Funcionales

<b>Identificación de caso de uso</b>	CDU01
<b>Nombre</b>	Iniciar Localización
<b>Actores</b>	Usuario, Servidor
<b>Descripción del caso de uso</b>	<ol style="list-style-type: none"><li>1. El usuario solicita visualizar su ubicación.</li><li>2. La App verificara que el modulo bluetooth se halle activo y de no estarlo solicitara su activación.</li><li>3. La App llamará a la función CDU02 "Tomar Referencias".</li><li>4. La App tomara los valores de la toma de referencias.</li><li>5. La App envía los datos al servidor por <i>web service</i> mediante la función CDU03 "Estimar Ubicación"</li></ol>
<b>Condiciones previas</b>	Tener conexión a Internet
<b>Resultado de la terminación</b>	1. Paquete de datos para ser enviados al servidor.

<b>Excepciones</b>	1. Servidor desconectado. 1.1 Cancelar proceso de localización.
<b>Asociaciones de casos de uso</b>	1. CDU02. 2. CDU03.
<b>Resumen de entradas</b>	1. Solicitud de usuario.
<b>Resumen de salidas</b>	1. Información de la señales de los <i>Beacons</i>

Tabla 31: Descripción de Caso de Uso - CDU01 (Autoría Propia, 2017).

<b>Identificación de caso de uso</b>	CDU02
<b>Nombre</b>	Tomar Referencias
<b>Actores</b>	<i>Beacon, Smartphone</i>
<b>Descripción del caso de uso</b>	1. La App receptorá la información de los <i>Beacons</i> que detecte. 2. La App obtendrá los parámetros necesarios de la señal de cada <i>Beacon</i> . 3. La App almacenará los parámetros en una tabla dentro una base <i>SQLite</i> . 3. La App tendrá un tiempo de tres segundos para repetir los pasos 1 y 2. 3. Enviar los datos de la tabla al caso de uso CDU01.
<b>Condiciones previas</b>	1. CDU01. 2. <i>Beacons</i> funcionando correctamente. 3. Módulo de bluetooth del <i>Smartphone</i> encendido.
<b>Resultado de la terminación</b>	1. Tabla con información almacenada.
<b>Excepciones</b>	1. Beacon desconectado. 1.1 Continuar con los activos.
<b>Asociaciones de casos de uso</b>	1. CDU01.
<b>Resumen de entradas</b>	1. Solicitud desde CDU01. 2. Parámetros de cada <i>Beacon</i> .
<b>Resumen de salidas</b>	Tabla de información

Tabla 32: Descripción de Caso de Uso - CDU02 (Autoría Propia, 2017).

<b>Identificación de caso de uso</b>	CDU03
<b>Nombre</b>	Estimar Posición
<b>Actores</b>	Servidor
<b>Descripción del caso de uso</b>	1. El servidor receptorá la petición por parte de la App. 2. El servidor aplicará el algoritmo en base a los datos proporcionados. 3. El servidor pasara el resultado a la función CD04.
<b>Condiciones previas</b>	1. CDU02.
<b>Resultado de la terminación</b>	1. Coordenadas de ubicación o Laboratorio.
<b>Excepciones</b>	1. Errores en el algoritmo de localización. 1.1 Reintentar proceso de localización.
<b>Asociaciones de casos de uso</b>	1. CDU01. 2. CDU04.
<b>Resumen de entradas</b>	1. Datos de las <i>Beacons</i> .

<b>Resumen de salidas</b>	<ol style="list-style-type: none"> <li>1. Resultado de ubicación.</li> <li>2. Numero de laboratorio.</li> </ol>
---------------------------	---

Tabla 33: Descripción de Caso de Uso - CDU03 (Autoría Propia, 2017).

<b>Identificación de caso de uso</b>	CDU04
<b>Nombre</b>	Enviar Resultados
<b>Actores</b>	Servidor
<b>Descripción del caso de uso</b>	<ol style="list-style-type: none"> <li>1. El servidor receptorá el resultado proveniente de la función CDU03.</li> <li>2. El servidor enviará mediante <i>web service</i> SOAP el resultado a la App.</li> </ol>
<b>Condiciones previas</b>	<ol style="list-style-type: none"> <li>1. CDU04.</li> <li>2. App solicitante conectada a Internet.</li> </ol>
<b>Resultado de la terminación</b>	1. Resultado de ubicación.
<b>Excepciones</b>	<ol style="list-style-type: none"> <li>1. App sin conexión a Internet. <ol style="list-style-type: none"> <li>1.1 Cancelar proceso de localización.</li> </ol> </li> </ol>
<b>Asociaciones de casos de uso</b>	<ol style="list-style-type: none"> <li>1. CDU03.</li> <li>2. CDU05.</li> </ol>
<b>Resumen de entradas</b>	1. Resultado de ubicación.
<b>Resumen de salidas</b>	1. Envío de resultado.

Tabla 34: Descripción de Caso de Uso - CDU04 (Autoría Propia, 2017).

<b>Identificación de caso de uso</b>	CDU05
<b>Nombre</b>	Representar Resultados
<b>Actores</b>	Usuario
<b>Descripción del caso de uso</b>	<ol style="list-style-type: none"> <li>1. La App tomará el resultado enviado por el servidor.</li> <li>2. La App cargar el mapa de la planta.</li> <li>3. La App interpretará el resultado recibido e identificara la ubicación en el mapa digital</li> <li>4. La App cargará las coordenadas o el nombre del laboratorio.</li> <li>5. La App colocará el icono en la zona del plano.</li> <li>6. La App colocará la información contextual.</li> </ol>
<b>Condiciones previas</b>	<ol style="list-style-type: none"> <li>1. CDU04.</li> <li>2. CDU006</li> </ol>
<b>Resultado de la terminación</b>	<ol style="list-style-type: none"> <li>1. Ubicación en el plano digital.</li> <li>2. Colocación de número de laboratorio.</li> <li>3. Colocación de coordenadas</li> </ol>
<b>Excepciones</b>	<ol style="list-style-type: none"> <li>1. Resultado erróneo. <ol style="list-style-type: none"> <li>1.1 Informar al usuario y dar opción de realizar un reproceso.</li> </ol> </li> </ol>
<b>Asociaciones de casos de uso</b>	1. CDU04.
<b>Resumen de entradas</b>	1. Recepción del resultado.
<b>Resumen de salidas</b>	<ol style="list-style-type: none"> <li>1. Ubicación en el plano digital.</li> <li>2. Nombre del laboratorio.</li> </ol>

Tabla 35: Descripción de Caso de Uso - CDU05 (Autoría Propia, 2017).

<b>Identificación de caso de uso</b>	CDU06
<b>Nombre</b>	Información contextual
<b>Actores</b>	Servidor
<b>Descripción del caso de uso</b>	<ol style="list-style-type: none"> <li>1. El servidor obtendrá la fecha y hora actual.</li> <li>2. El servidor buscará los horarios de las clases, nombres de asignaturas y profesores, en base a la hora y fecha actuales.</li> <li>3. El servidor enviara la descripción del lugar donde se halle el usuario.</li> </ol>
<b>Condiciones previas</b>	CDU03
<b>Resultado de la terminación</b>	1. Información contextual.
<b>Excepciones</b>	<ol style="list-style-type: none"> <li>1. Resultados de localización erróneos. <ol style="list-style-type: none"> <li>1.1 No presentar ninguna información.</li> </ol> </li> </ol>
<b>Asociaciones de casos de uso</b>	
<b>Resumen de entradas</b>	1. Localización actual.
<b>Resumen de salidas</b>	1. Información contextual.

Tabla 36: Descripción de Caso de Uso - CDU06 (Autoría Propia, 2017).

### 3.7.5. Requisitos no Funcionales

<b>Identificación del requerimiento</b>	RNF01
<b>Nombre</b>	Interfaz de la App
<b>Descripción</b>	La aplicación presentará una interfaz fácil de entender, la cual muestre un plano intuitivo para el usuario y se use un icono fácil de distinguir
<b>Prioridad</b>	Alta

Tabla 37: Descripción de Requisito No Funcional - RNF01 (Autoría Propia, 2017).

<b>Identificación del requerimiento</b>	RNF02
<b>Nombre</b>	Disponibilidad
<b>Descripción</b>	Mantener disponible al servidor el máximo de tiempo para prestar el servicio de localización, y procurando tiempos cortos de respuesta.
<b>Prioridad</b>	Media/Alta

Tabla 38: Descripción de Requisito No Funcional - RNF02 (Autoría Propia, 2017).

<b>Identificación del requerimiento</b>	RNF03
<b>Nombre</b>	Escalabilidad
<b>Descripción</b>	Diseñar la aplicación móvil para que pueda ser integrada a otras aplicaciones que necesiten el servicio
<b>Prioridad</b>	Media

Tabla 39: Descripción de Requisito No Funcional - RNF03 (Autoría Propia, 2017).



<b>Identificación del requerimiento</b>	RNF04
<b>Nombre</b>	Rendimiento
<b>Descripción</b>	Realizar la localización mediante un algoritmo eficiente en tiempo y carga computacional
<b>Prioridad</b>	Alta

Tabla 40: Descripción de Requisito No Funcional - RNF04 (Autoría Propia, 2017).

### 3.8. Servidor de Aplicaciones

#### 3.8.1. Arquitectura

La arquitectura que se adopta en el sistema de localización en interiores empleando tecnología bluetooth consiste en un modelo cliente/servidor de dos capas, al igual que con la tecnología WiFi, donde la primera capa corresponde al nodo terminal que *Smartphone* de sistema operativo *Android*, y el segundo es el servidor de aplicaciones desarrollado en el lenguaje Java.

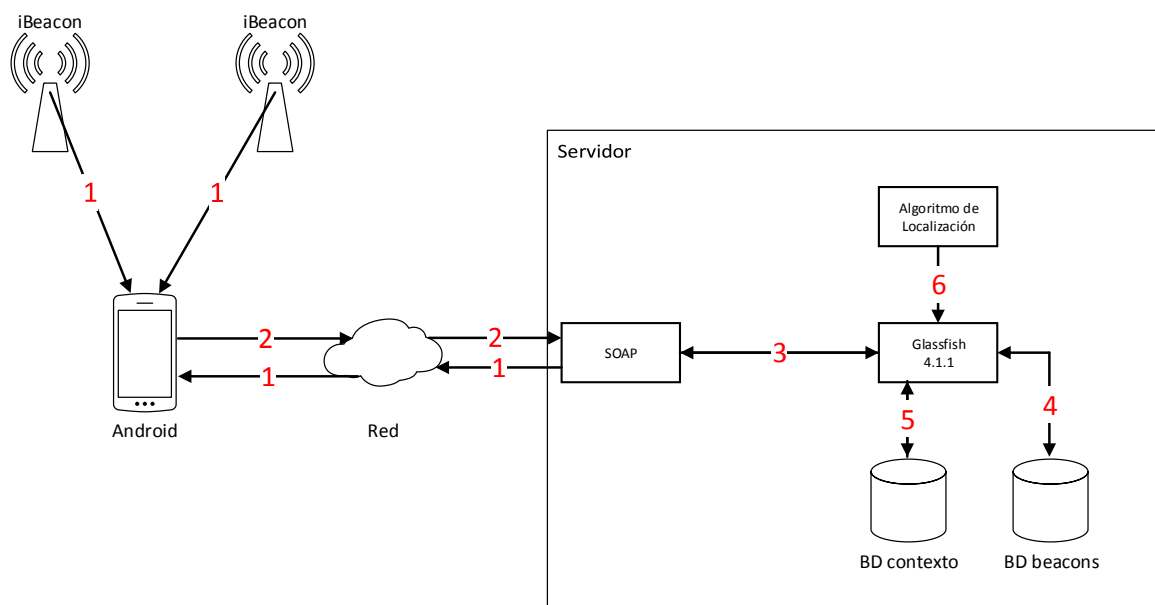


Figura 57: Diagrama de la arquitectura del sistema (Autoría Propia, 2017).

#### 3.8.2. Base de Datos

La base de datos esta implementada en el sistema de gestión MySQL, las tablas que contiene la base de datos es referente a la información de los *Beacons*, de las clases, asignaturas, de los lugares y los horarios, ya que en este sistema de localización también

se incluyó la entrega de información contextual basándose en el lugar donde se halle actualmente el usuario.

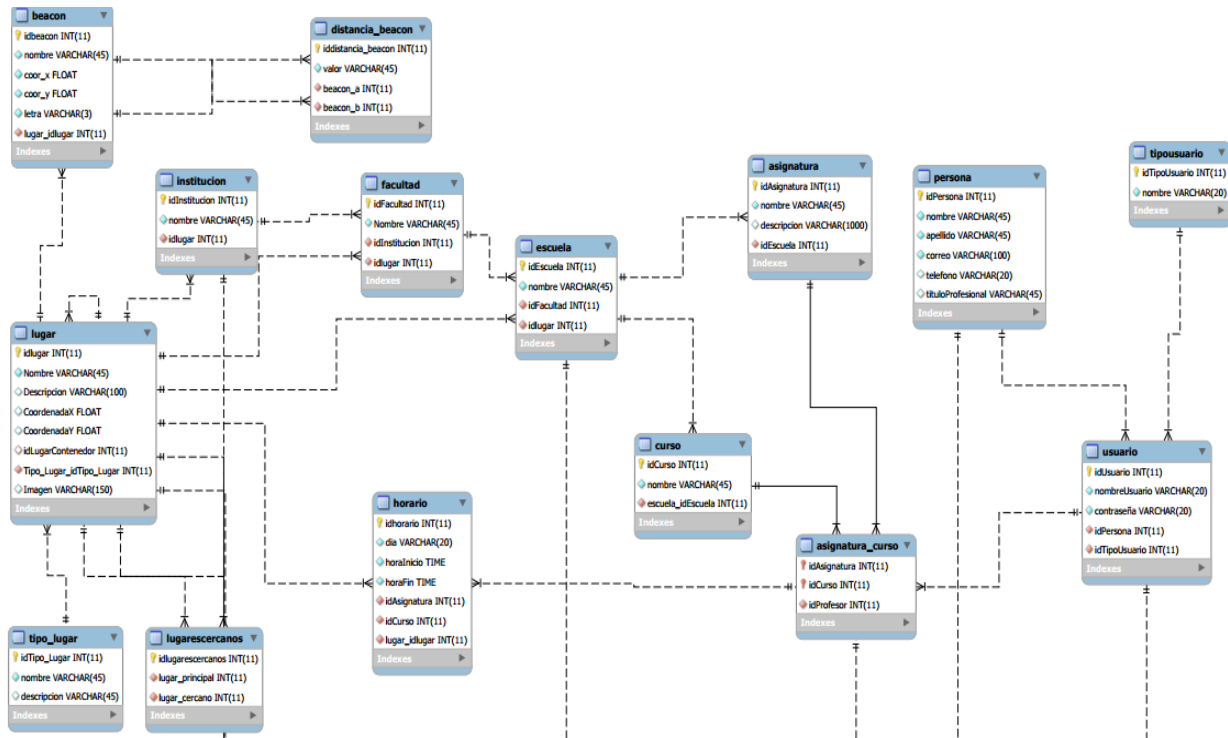


Figura 58: Modelo Entidad/Relación

Las tablas empleadas para el sistema de localización son las que se observa en el diagrama entidad/relación de la Figura 58, esta misma base de datos es compartida con en el proyecto “Recomendador de Eventos y Noticias”, cuyo diagrama entidad/relación se lo puede observar en el Anexo V, el cual se complementa con que el esquema presentado en la Figura 58 la cual muestra únicamente las tablas necesarias para el sistema de localización en interiores.

### 3.8.3. Servicios Web

La forma de comunicación entre el usuario final y el servidor de aplicaciones es a través de web services, y al igual que con la tecnología WiFi, el estándar del servicio es SOAP. Se poseen dos servicios web, el primero encargado de recibir la información de los *Beacons* conjuntamente con las distancias estimadas hacía el *Smartphone*, y retornar las coordenadas de ser el caso o el laboratorio en el cual se halla. Y el segundo para enviar

el id de un laboratorio y retornar la información contextual. El WSDL correspondiente al servicio web es el siguiente:

[<xsd:importnamespace=http://localizacion/schemaLocation="http://localhost:8080/ServidorIndoorBluetooth/ServiciosLocalizacion?xsd=1"/>](http://localhost:8080/ServidorIndoorBluetooth/ServiciosLocalizacion?xsd=1/)

**SOAP Request**

```
<?xml version="1.0" encoding="UTF-8"?><S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header/>
  <S:Body>
    <ns2:Localizar xmlns:ns2="http://servicios/">
  </S:Body>
</S:Envelope>
```

**SOAP Response**

```
<?xml version="1.0" encoding="UTF-8"?><S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header/>
  <S:Body>
    <ns2:LocalizarResponse xmlns:ns2="http://servicios/">
      <return>12.67</return>
      <return>5.43</return>
      <return>cuatro</return>
    </ns2:LocalizarResponse>
  </S:Body>
</S:Envelope>
```

Figura 59: Estructura de servicio web de localización (Autoría Propia, 2017).

**SOAP Request**

```
<?xml version="1.0" encoding="UTF-8"?><S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header/>
  <S:Body>
    <ns2:ObtenerClases xmlns:ns2="http://servicios/">
      <Datos>L4:12:00:00:2</Datos>
    </ns2:ObtenerClases>
  </S:Body>
</S:Envelope>
```

**SOAP Response**

```
<?xml version="1.0" encoding="UTF-8"?><S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header/>
  <S:Body>
    <ns2:ObtenerClasesResponse xmlns:ns2="http://servicios/">
      <return xmlns:x="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="xs:string">Laboratorio 4;SIG:Juan 1
    </ns2:ObtenerClasesResponse>
  </S:Body>
</S:Envelope>
```

Figura 60: Estructura de servicio web de contexto (Autoría Propia, 2017).

### 3.9. Aplicación Móvil

La aplicación móvil fue desarrollada para el sistema operativo *Android*, empleando en IDE de desarrollo *Android Studio* en el lenguaje de programación Java.

#### 3.9.1. Diseño de la aplicación móvil

La aplicación móvil consta de la función principal de localización y visualización, además de las funciones de obtener información adicional respecto al contexto, es decir información sobre las asignaturas que actualmente se dictan y una descripción de los laboratorios o pasillo.

Para la función de localización se poseen librerías que permiten tanto la obtención de los parámetros de los *Beacons*, así como también para realizar el llamado a los servicios web.

La base de datos que maneja propiamente en la App, consiste en una sola tabla donde se almacenan los valores actuales de cada *Beacon*, como son el nombre y la distancia calculada.

A continuación se presenta un diagrama de clases UML (*Unified Modeling Language*), dentro del cual se establecen las principales clases que componen el desarrollo de la aplicación (Anexo IV).

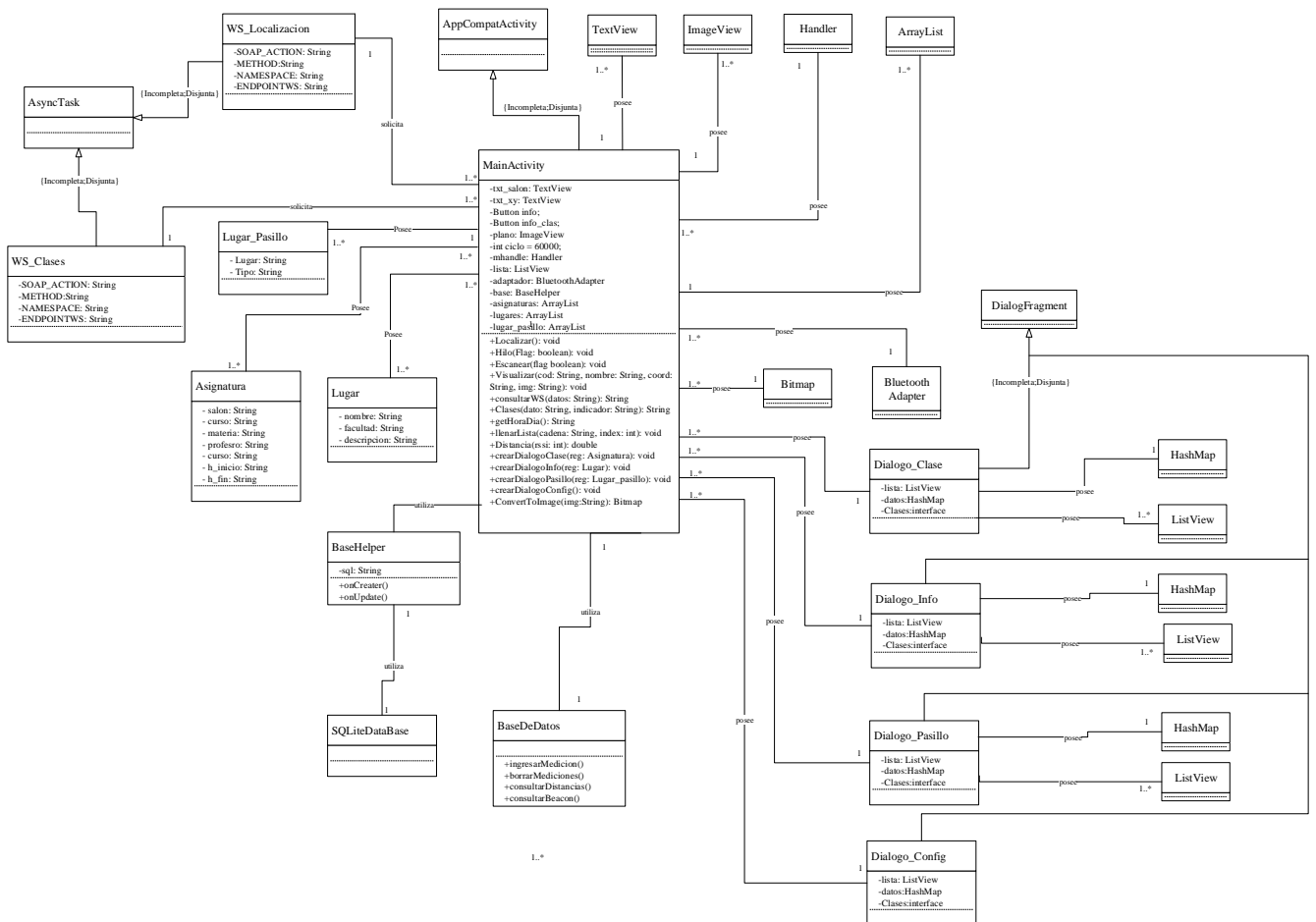


Figura 61: Diagrama de Clases (Autoría Propia, 2017).

### 3.9.2. Base de Datos *SQLite*



Figura 62: Logo de *SQLite* (Filein, 2017).

*SQLite* funciona de forma similar a un motor de base de datos SQL (*Structured Query Language*), se diferencia de los demás motores de bases de datos en que *SQLite* no necesita tener un servidor corriendo todo el tiempo. *SQLite* funciona como una biblioteca de archivos, destinado a dispositivos con baja capacidad como los *Smartphone* (Filein, 2017).

*SQLite* pone a disposición todas las sentencias SQL básicas, las cuales no tienen ningún cambio con respecto a los gestores comunes. *SQLite* tiene librerías disponibles para varios lenguajes de programación, entre ellos Java, lenguaje empleado para programar aplicaciones *Android* (Filein, 2017).

Se implementó una base de datos *SQLite* en la aplicación con el objetivo de almacenar datos temporales, únicamente cuenta con una tabla de dos campos, el primer campo es el nombre de cada Beacon que es captado por el *Smartphone* y segundo campo la distancia en metros estimada. De cada Beacon existen múltiples registros ya que estos están configurados para emitir una señal cada 300ms.

Esta única tabla es para un almacenamiento temporal, y una vez enviado los datos hacia el servidor la tabla es vaciada, pues ya no son necesarios los registros.

### 3.9.3. Módulo Bluetooth

En muchas ocasiones el módulo bluetooth del *Smartphone* se encuentra desactivado, por lo que es necesario antes de realizar cualquier acción verificar si encuentra o no activo este módulo.

Si el módulo de bluetooth está apagado es necesario lanzar una solicitud de autorización, para que el usuario permita la activación.

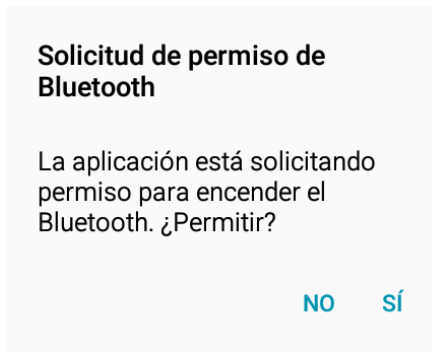


Figura 63: Proceso de Localización (Autoría Propia, 2017).

### 3.9.4. Localización y visualización

Durante la etapa de lectura de los niveles RSSI de los *Beacons* más cercanos, se mantiene un tiempo de tres segundos para la realización del escaneo, ya que todos los *Beacons* están configurados para enviar una señal cada 300ms, tiempo apropiado para el escaneo, puesto que se podrá tener aproximadamente 10 lecturas de los *Beacons*.

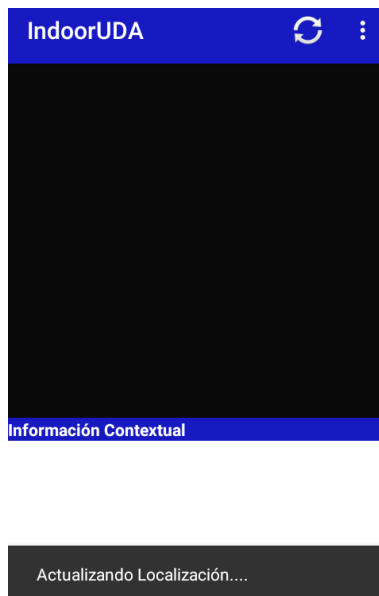


Figura 64: Proceso de Localización (Autoría Propia, 2017).

La visualización de la localización dependerá del resultado que entregue el servidor de aplicaciones, ya que si el resultado es una coordenada indicando que el usuario está ubicado en el pasillo, se mostrarán las coordenadas y un punto dentro del plano al cual pertenecen dichas coordenadas.

En la parte superior también posee un botón para actualizar manualmente la localización y una opción para configurar un tiempo de actualización automática.



Figura 65: Localización parte izquierda de pasillo (Autoría Propia, 2017).

Cuando el usuario ya ha ingresado a cualquiera de los dos laboratorios la aplicación únicamente presentará el plano del laboratorio en cual se encuentre, más no las coordenadas ya que solamente se posee un *Beacon* por laboratorio.

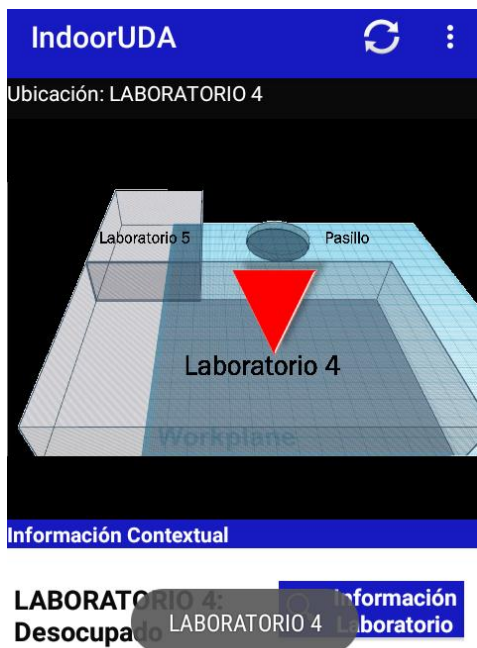


Figura 66: Laboratorio 4 (Autoría Propia, 2017).

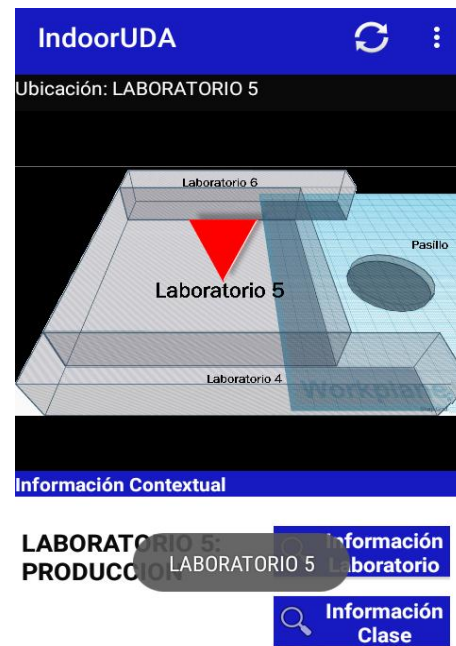
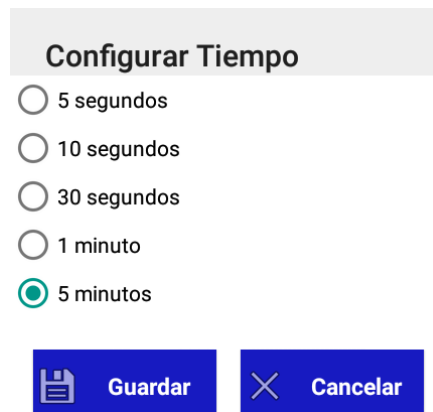


Figura 67: Laboratorio 5 (Autoría Propia, 2017).

Adicionalmente como ya se mencionó la aplicación posee una opción de menú para configuración de tiempo de actualización, esta opción permite al usuario establecer el tiempo para que la aplicación móvil actualice la localización.



**Configurar Tiempo**

5 segundos

10 segundos

30 segundos

1 minuto

5 minutos



 **Guardar**  **Cancelar**

Figura 68: Configuración de tiempo de actualización (Autoría Propia, 2017).

De ocurrir cualquier eventualidad durante la ejecución de la aplicación móvil, se presentara un mensaje de que no se pudo completar la estimación de la localización.

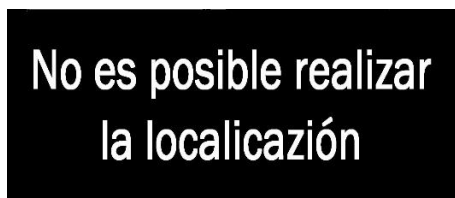


Figura 69: Mensaje de Error (Autoría Propia, 2017).

### 3.9.5. Información contextual

La información de contexto que se presenta al usuario al estar dentro de los laboratorios, corresponde a la clase que se está dando en este momento. Dentro de la información proporcionada está: el nombre de la asignatura, el nombre del profesor encargado, el nombre del curso, el laboratorio en la que se está dando la clase y las horas de inicio y finalización. De no estar el laboratorio ocupado en el instante al cual el usuario ingreso se notificará que el laboratorio se halla desocupado.

Así como también información acerca del laboratorio, como es el nombre, facultad y piso donde se halla, proyector, capacidad para estudiantes, número de equipos, etc.

Al estar el usuario ubicado en el pasillo la información proporcionada corresponderá a los lugares que estén cercanos al pasillo, como puede ser los mismos laboratorios, oficinas, aulas, etc.



## Lugares Cercanos

### LABORATORIO 6

Área de Informática

---

### LABORATORIO 4

Área de Informática

---

### LABORATORIO 7

Área de Informática

[Regresar](#)

Figura 70: Información contextual del pasillo (Autoría Propia, 2017).

## Clase Actual

### Asignatura:

PRODUCCION

---

### Horario:

16:00:00/21:00:00

---

### Profesor:

JUAN PEREZ

**CURSO.**

OCTAVO A

---

### Lugar:

LABORATORIO 5



[Regresar](#)

Figura 71: Información contextual de la clase (Autoría Propia, 2017)

## Datos del Lugar

### Descripcion:

Cuarto Piso

Posee Proyector

Capacidad para 25 alumnos

Posee 15 computadoras

---

### Facultad:

FILOSOFIA

---

### Lugar:

[Regresar](#)

Figura 72: Información contextual del laboratorio (Autoría Propia, 2017).

### 3.10. Conclusiones

Se logró realizar un sistema de localización en interiores basado en tecnología Bluetooth, las pruebas fueron exitosas en ambos escenarios, teniéndose una precisión de entre 1m a 3m en la zona del pasillo.

Los *Beacons* son dispositivos que permiten ubicar a un usuario en base a su *Smartphone*, ya que trabajan con tecnología bluetooth, siendo esta compatible con la mayoría de teléfonos móviles, es fácil adaptarlo a cualquier ambiente de interiores y al ser de bajo consumo no se requiere una conexión directa, facilitando la comunicación.

Colocar solo un *Beacon* por laboratorio fue suficiente para conseguir distinguir si se halla dentro del mismo o fuera en el pasillo, y también es suficiente para que sea reconocido desde cualquier punto del laboratorio.

La trilateración es método que conjuntamente combinado con los *Beacons*, logró realizar una buena estimación sobre la localización del usuario en el pasillo, de igual modo solamente requiere de cuatro operaciones matemáticas sencillas sin complicaciones de computo.

El incluir información contextual sobre las clases que se están dictando en el momento y sobre el lugar, es un adicional que mejora al sistema y da mayores prestaciones, ya que no únicamente indica al usuario su posición actual sino que también le entrega información adicional sobre lo que ocurre en los salones cercanos.

## Capítulo 4

### 4. Resultados y Discusión

#### 4.1. Introducción

Tras haber desarrollado dos sistemas de localización en interiores, uno con tecnología WiFi y el otro con tecnología Bluetooth, es necesario conocer cuál de las dos tecnologías se adecua más para un sistema de localización en interiores.

Se describirán los resultados que arrojaron las pruebas de precisión en la localización con cada tecnología.

En este capítulo también se plantea una discusión entre ambas tecnologías, partiendo de una matriz FODA (fortalezas, oportunidades, debilidades y amenazas) para cada una de las tecnologías y también una comparación sobre los resultados obtenidos.

#### 4.2. Resultados de la tecnología WiFi

Fueron realizadas pruebas en nueve puntos diferentes en cada laboratorio, obteniéndose una precisión de localización en cada uno, esta precisión se estableció por medio de una matriz comparativa entre los valores localización real y los estimados por la aplicación (matriz de prueba y error). Se obtuvo una precisión de 100% en los laboratorios 5 y 7, principalmente a que en estos existe un punto de acceso a la red en cada laboratorio (Anexo I).

En los laboratorios 4 y 6 mediante las pruebas realizadas se llega a tener un 88% de precisión, esto se debe a que en estos laboratorios no hay un punto de acceso, y en la parte posterior las lecturas de los niveles RSSI son similares, por lo que en ciertas ocasiones cuando el usuario está en el laboratorio 4 el sistema da como resultado el laboratorio 6 y viceversa (Anexo I).

En la zona del pasillo en lo que corresponde alrededor del espacio vacío, con las pruebas obtuvieron resultados incorrectos de localización, esto debido a que son espacios angostos y los puntos de referencia que se toman son muy parecidos a los puntos de las

esquinas de los laboratorios cercanos, únicamente en las zonas alejadas del espacio vacío se obtuvo resultados de localización correctos.

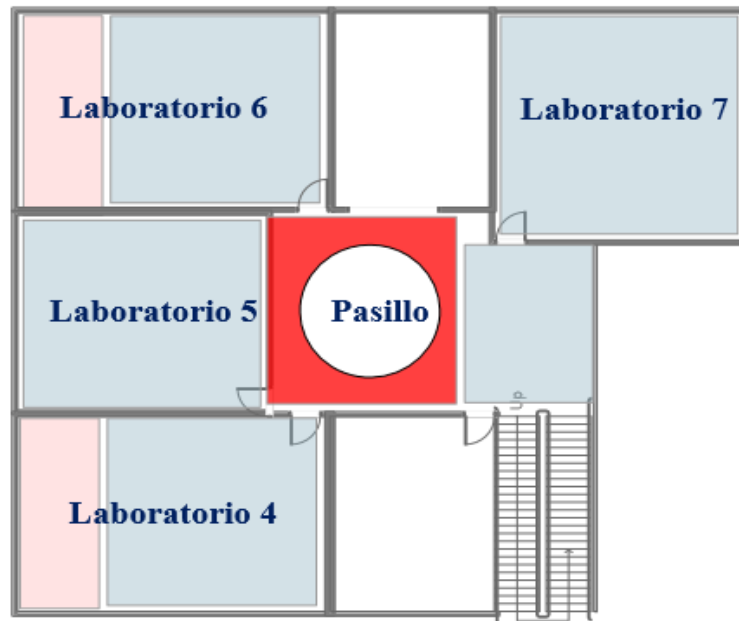


Figura 73: Resultados de tecnología WiFi (Autoría Propia, 2017).

Como se puede apreciar en la Figura 73, las marcas con color rosado representan las partes posteriores de los laboratorios 4 y 6 en las cuales existe una disminución de la precisión en la localización, las marcas de color azul colocadas en los laboratorios 5, 7 y la parte derecha al espacio vacío representan que los resultados de la localización son correctos.

La marca de color rojo alrededor del espacio vacío representa que las pruebas fallaron al estimar la localización, cuando se realizaban pruebas alrededor de este espacio los resultados daban que se hallaba dentro de uno de los laboratorios.

Todas las pruebas se las realizó con las puertas abiertas, pero al permanecer cerradas los resultados no varían, únicamente cambia el valor de los valores RSSI en una o dos unidades en los laboratorios que no poseen un punto de acceso a la red.

### 4.3. Resultados de la tecnología Bluetooth

Como el sistema de localización con tecnología bluetooth está basado en el método de trilateración para la zona del pasillo en ambos escenarios, fue necesario que las pruebas se las realice en determinadas coordenadas las cuales son conocidas de antemano

(planificación de la red), para después de la estimación de la coordenada, obtener la precisión en metros que el sistema maneja.

Se realizaron 18 pruebas para el primer escenario, es decir se tomó 18 coordenadas diferentes para realizar la comparación con las que el sistema da como resultado, y para el segundo escenario al estar reducido a la parte izquierda del pasillo de realizaron pruebas en 8 puntos distintos.

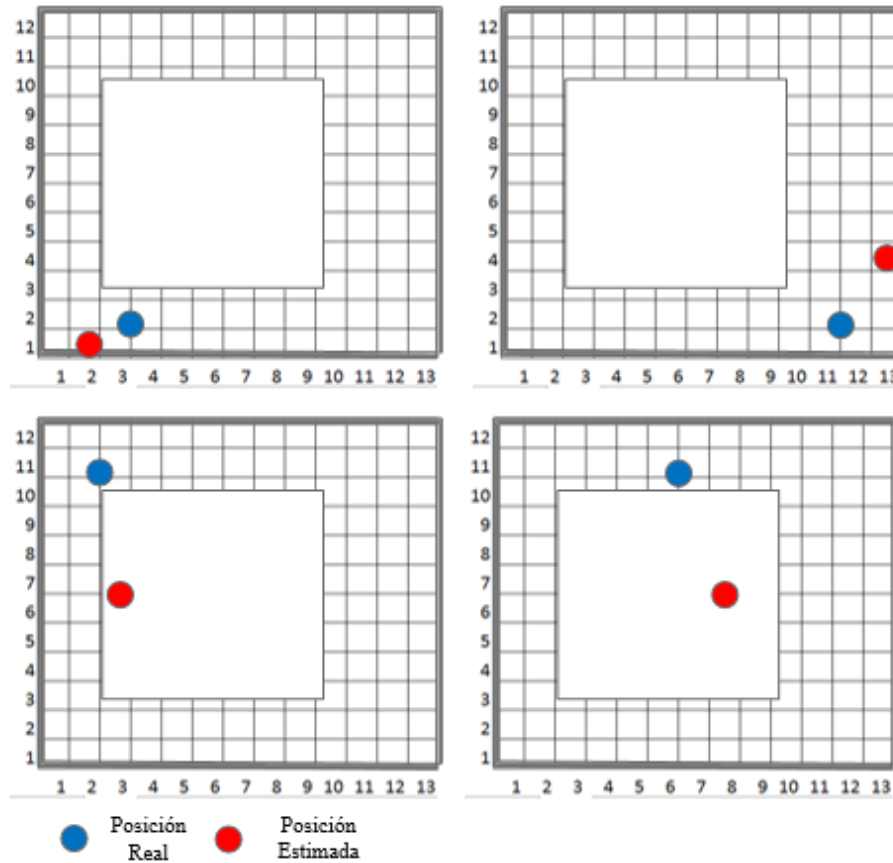


Figura 74: Pruebas de Precisión (Autoría Propia, 2017).

Como resultado de las pruebas se conoce que la precisión promedio del sistema para el primer escenario es de 3.54m (Anexo II) y para el segundo escenario de 3.70m (Anexo II), pero como se consideró solamente al segundo escenario para la implementación del sistema, la precisión que el sistema maneja es de 3.7m.

#### 4.4. Matriz FODA de la tecnología WiFi

<b>Fortalezas</b>	<b>Debilidades</b>
Tener una red WiFi en la mayoría de los casos ya implementada	Costos altos si se necesitase incrementar el número de puntos de acceso
Algoritmo con baja carga computacional	Necesidad de estar conectado específicamente a dicha red WiFi
Rapidez de escaneo de nivel RSSI	Variedad de antenas WiFi en los <i>Smartphone</i>
Complejidad media en el desarrollo de la App	Mientras existan más salones se debe establecer más puntos de referencia
Facilidad en el uso de librerías enfocadas a WiFi	El número de puntos de referencia incrementa dependiendo del tamaño de los salones
Posibilidad de mejorar el algoritmo al incluir más puntos de referencia	Necesidad de recoger mediciones en los puntos de referencia por cada modelo de <i>Smartphone</i>
Alta precisión para determinar el salón donde se encuentra	Ralentización del tiempo de respuesta al tener más usuarios
Los puntos de acceso son visibles desde la mayoría de los salones	Lectura similar de nivel RSSI en la salones diferentes y por lo tanto respuestas erróneas
Aplicable a cualquier <i>Smartphone</i>	Baja precisión para localización en los pasillos
	Ligera diferencia en la medición de nivel RSSI al estar con las puertas de los salones cerradas
	Aumento de la complejidad de desarrollo al ampliarse el sistema a infraestructuras de más de un piso
	Complejidad para realizar cambios en la red WiFi
	Alto consumo energético de los equipos de la red WiFi
	Complejidad para adaptarse a cambios en el ambiente físico
	Alteración en la lectura de nivel RSSI al estar el <i>Smartphone</i> en movimiento
<b>Oportunidades</b>	<b>Amenazas</b>
Implementación de un punto de acceso en cada salón	Continua desconexión del <i>Smartphone</i> a la red
Entregar información contextual al ingresar en un salón	Cambios en la arquitectura de la red
Obtener valores de mediciones genéricas que sea aplicables a la mayoría de los <i>Smartphone</i>	Reemplazo de puntos de acceso
Servidor de aplicaciones con mayor capacidad	Caída de red WiFi
Buscar una nueva variante que permita esclarecer la lectura RSSI	Saturación de servidor debido al aumento de usuarios
	Ralentización de escaneo debido al número de equipos conectados a la red
	Salones en los que no llega la señal WiFi

Tabla 41: Matriz FODA de tecnología WiFi (Autoría Propia, 2017).

#### 4.5. Matriz FODA de la tecnología Bluetooth

<b>Fortalezas</b>	<b>Debilidades</b>
Facilidad de adquisición e instalación de <i>Beacons</i>	Variedad de antenas Bluetooth en los <i>Smartphone</i>
Facilidad de reemplazo de <i>Beacons</i>	Realizar medición de niveles RSSI para cada modelo de <i>Smartphone</i>
Facilidad de configuración de <i>Beacons</i>	Ralentización del tiempo de respuesta al tener más usuarios
Alta precisión para localizar en los salones y distinguir en que parte del pasillo se halla	Aumento de la complejidad de desarrollo al ampliarse el sistema a infraestructuras de más de un piso
Solamente es necesario un <i>Beacon</i> por salón	Necesidad de implementar un sistema de coordenadas interno
Rapidez de escaneo de RSSI	Niveles RSSI muy parecidos entre <i>Beacons</i> al estar en zonas pequeñas
Algoritmo con baja carga computacional	Alteración en la lectura de nivel RSSI al estar el <i>Smartphone</i> en movimiento
No es necesario estar conectado a una red es específico, basta con tener acceso a Internet	
Facilidad para adaptarse a cambios en el ambiente físico	
Complejidad media en el desarrollo de la App	
Facilidad en el uso de librerías enfocadas a Bluetooth LE	
Bajo consumo energético de los <i>Beacons</i> y con batería reemplazable	
Aplicable a cualquier <i>Smartphone</i>	
No es necesario establecer una conexión estable entre <i>Smartphone</i> y <i>Beacon</i>	
<b>Oportunidades</b>	<b>Amenazas</b>
Adquirir más <i>Beacons</i>	Demasiados equipos con bluetooth activo puede volver más lento el escaneo
Entregar información contextual al ingresar en un salón o al estar en el pasillo	Obstrucción de la línea de visión de los <i>Beacons</i>
Servidor de aplicaciones con mayor capacidad	<i>Beacons</i> defectuosos al momento de adquirirlos
Implementar más funcionalidades con otros dispositivos que use bluetooth LE	Caída de redes WiFi
Adquirir <i>Beacons</i> con sensores adicionales para prestar más servicios	Obsolescencia de <i>Beacons</i>
Establecer algoritmos auxiliares para mejorar la estimación de nivel RSSI	

Tabla 42: Matriz FODA de tecnología Bluetooth (Autoría Propia, 2017).

## 4.6. Comparación entre tecnologías

Las tecnologías WiFi y Bluetooth funcionan dentro de una red de radio frecuencia, por lo que el parámetro empleado para estimar la localización en ambos casos es el nivel RSSI, y por ende no se realizará la comparación entre las señales que estos equipos emiten, sino que la comparación se da en la implementación de las tecnologías.

El implementar una red bluetooth en cualquier ambiente físico supone menor esfuerzo y menor costo al compararlo con la implementación de una red WiFi, principalmente se da por el costo de cada uno de los equipos. El sistema implementado con tecnología bluetooth está compuesto por *Beacons*, los cuales cuestan \$25 por unidad y en caso de la red WiFi, los puntos de acceso cuestan alrededor de \$100 por unidad, sin contar con los demás equipos necesarios para su implementación. Sin embargo, en la mayoría de los escenarios ya existe implementada una red WiFi, lo que supone un gasto de \$0, siendo esta la mayor ventaja con respecto a la tecnología bluetooth, aunque este hecho repercute en otros aspectos.

El consumo energético en ambas tecnologías es muy diferente, con una red WiFi aparte del costo de adquisición también está el costo por consumo energético, ya que estos equipos se hallan conectados a una fuente de energía eléctrica de manera permanente. Con los *Beacons* este costo por la energía no es de manera permanente, debido a que los *Beacons* cuentan con baterías, las cuales pueden llegar a durar meses, y cuando la batería esté por terminarse solamente se la reemplaza por otra.

La ventaja que presentan los *Beacons* es el hecho de contar con una fácil configuración de los mismos, la cual tarda menos de cinco minutos por cada dispositivo, esto en contraposición con la tecnología WiFi, a la cual muchas de las veces no se tiene permiso para realizar configuraciones (como fue en este caso), y si se tuviese la posibilidad de realizarlas, estas requieren más tiempo en comparación con un *Beacon*.

En lo concerniente a los métodos de localización ambos comparten la ventaja de tener una baja carga computacional, y los tiempos de cálculo son muy similares, únicamente difieren en la capacidad de almacenamiento de información que requieren, puesto que en el método de *fingerprint* es necesario guardar más datos que con el método de trilateración.



Es en la precisión en donde el método implementado con la tecnología bluetooth (trilateración) es mejor, ya que al tener *Beacons* en el pasillo el sistema puede interpretar en base al cálculo de coordenadas que el usuario se halla en el pasillo, lo que con el sistema de tecnología WiFi no se consiguió del todo (Figura 73), puesto que alrededor del espacio vacío el sistema siempre entrega un resultado erróneo y únicamente funciona al estar situado en la parte más amplia del pasillo, es decir a la derecha de espacio vacío.

Lo que respecta a la precisión para localizar dentro de los laboratorios, ambas tecnologías cumplen con este objetivo, obviamente la tecnología bluetooth es más confiable para los laboratorios en los cuales no poseen un punto de acceso, pero si se llegase a tener las dos rendirían con una precisión del 100% en este aspecto. Aunque si se desea instalar un punto de acceso por cada salón únicamente para un sistema de localización en interiores, es más recomendable optar por la compra de *Beacons*, debido al costo que tendría el adquirirlos.

Un aspecto importante es la facilidad de adaptar las tecnologías a diversas topologías dentro del entorno físico en el cual serán implementadas. Esta facilidad de adaptabilidad no siempre es posible con una red WiFi, ya que como se mencionó antes muchas de las veces esta red ya se halla implementada, y no es fácil ni recomendable cambiar su configuración, teniendo en cuenta además que la distribución de equipos en esta red está orientada para brindar la mayor cobertura de señal, y no para facilitar la localización en interiores.

Aquí la red de *Beacons* tiene la ventaja, puesto que son más fáciles para establecer topologías, debido a su tamaño pequeño y facilidad de traslado, pero la principal razón es que la topología que se llegase a establecer está directamente enfocada para la localización en interiores, teniendo como única condición que los *Beacons* tenga línea de visión hacia el *Smartphone*.

Otra ventaja que presenta el sistema con tecnología bluetooth es que puede detectar cuando el usuario se encuentra dentro de un laboratorio o fuera en el pasillo, esto no sería posible en el sistema con tecnología WiFi, ya que este tiene mala precisión para localizar al usuario en el pasillo, y no se puede observar esa transición de salir del pasillo e ingresar a un laboratorio, solamente se pudiera ver una transición tras haberse cambiado de un laboratorio a otro, misma que también se puede observar en el sistema con bluetooth.

Ambas tecnologías comparten la desventaja de que sus métodos de localización propuestos necesitan recoger información para cada modelo de *Smartphone*, debido a la variedad de antenas en los teléfonos móviles tanto para la captación de señal WiFi como de bluetooth. Anuqué esto es más desfavorable para el método *fingerprint* planteado con la tecnología WiFi, puesto que se debería repetir el proceso de calibración cada vez que se desee agregar a un nuevo modelo, es decir tener que tomar las lecturas de nivel RSSI en cada uno de los salones, en cambio con el método de la tecnología bluetooth se debe realizar un proceso de medición de niveles RSSI ubicándose a ciertas distancias del *Beacon*, lo que requiere poco tiempo.

El desarrollo de la aplicación conllevó una complejidad media para ambos casos, así que en este punto ninguna tecnología es más favorable que la otra. Teniendo en cuenta también que las librerías o repositorios de código fuente para las dos tecnologías están al alcance.

## Conclusiones

Es posible desarrollar un sistema de localización en interiores, y las dos tecnologías WiFi y bluetooth son aptas para la implementación del mismo, cada uno con sus ventajas y desventajas.

Se logró desarrollar y probar ambos sistemas de localización, basándose en los requerimientos establecidos al inicio de cada tecnología, cada uno con una alta precisión en la localización, con el correcto funcionamiento de un prototipo de aplicación móvil desarrollada en *Android* y con un servidor de aplicaciones desarrollado en Java.

Los métodos de localización empleados resultaron ser eficaces para la estimación de la localización, aunque está el hecho de que la trilateración conjuntamente con los *Beacons* proporciona información sobre la localización del usuario con mayor precisión, lo cual no pudo conseguirse con la tecnología WiFi.

Tras haber realizado la comparación con los resultados, ventajas y desventajas entre las tecnologías WiFi y bluetooth, se llegó a la determinación de que la tecnología bluetooth mediante los *Beacons* es la alternativa más viable para un sistema de localización en interiores.

En muchos puntos de comparación la tecnología bluetooth es mejor que la tecnología WiFi, como en el costo en tiempo y dinero que requiere la implantación de la red, en la facilidad para configurar los dispositivos, la mayor adaptabilidad hacia el entorno físico que presentan los *Beacons*, etc.

El sistema de localización con tecnología bluetooth puede ser ampliado hacia zonas más grandes, para conjuntamente entregar más información contextual, y únicamente serían necesario más *Beacons* y un periodo más amplio para el desarrollo.

## Trabajos Futuros

Desarrollar una aplicación móvil completa tanto para los sistemas operativos *Android* y *IOS*.

Adquirir *Beacons* con mejores capacidades para que mejoren la precisión en el sistema de localización y permitan entregar más servicios.

Ampliar más la información contextual proporcionada en el sistema de localización con tecnología bluetooth.

Llegar al desarrollo e implementación de lo que se llama una Universidad inteligente donde se incluya la localización en interiores y la prestación de información contextual, en base al posicionamiento del usuario.

Aplicar el sistema de localización a la mayoría de modelos *Smartphone* más comúnmente usados por los usuarios.

Ampliar el uso de los *Beacons* a entornos de exteriores, llegándose a conocer la transición de interior a exterior y viceversa.

## Bibliografía

- Ali Hassan, A. M. (2016). Indoor Location Tracking System Using Neural Network Based on Bluetooth. *International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT)*, (págs. 73-78). Egipto.
- Alvarez, Y., & Las Heras, F. (2016). ZigBee-based Sensor Network for Indoor Location and Tracking Applications. *IEEE LATIN AMERICA TRANSACTIONS*, 14(7), 3208-3214.
- areatecnologica. (19 de Septiembre de 2017). *Tecnología*. Obtenido de BLUETOOTH LE O LOW ENERGY: <http://www.areatecnologia.com/nuevas-tecnologias/bluetooth-le.html>
- Barba, F. (2012). *ESTUDIO DE ALGORITMOS DE LOCALIZACIÓN EN INTERIORES, PARA TECNOLOGÍAS MÓVILES DE ÚLTIMA GENERACIÓN*. Madrid: Universidad Politécnica de Madrid.
- Bluetooth. (2017). *Bluetooth*. Recuperado el 26 de Abril de 2017, de Wikipedia, La enciclopedia libre.: <https://es.wikipedia.org/wiki/Bluetooth>
- CCM. (16 de Octubre de 2008). *CCM - Comunidad Informática*. Recuperado el 26 de Abril de 2017, de Estándar GSM (Sistema global de comunicaciones móviles): <http://es.ccm.net/contents/681-estandar-gsm-sistema-global-de-comunicaciones-moviles>
- Chico Ciprián, A. (2009). *Diseño y desarrollo de un sistema de posicionamiento en interiores basado en Wi-Fi con tecnología Android*. Madrid: Universidad Carlos III de Madrid.
- Cisco. (20 de Septiembre de 2017). *Cisco Networking Academy*. Obtenido de iBeacon - Frequently Asked Questions: <https://www.netacad.com/es/>
- Developers, A. (30 de Mayo de 2017). *BluetoothDevice*. Obtenido de Developers: <https://developer.android.com/reference/android/bluetooth/BluetoothDevice.html>
- Developers, A. (30 de Mayo de 2017). *CellSignalStrengthGsm*. Obtenido de Developers: <https://developer.android.com/reference/android/telephony/CellSignalStrengthGsm.html>
- Días-Ambrona Tabernilla, L. (2008). *Requiere buena sincronización de unidades de medida*. Madrid: Universidad Politécnica de Madrid.
- Ekahau. (28 de Abril de 2017). *Wi-Fi Design, Wi-Fi Planning and Ekahau Site Survey Software, Wi-Fi Spectrum Analysis*. Obtenido de Ekahau: <https://www.ekahau.com/>
- Estimote. (11 de Junio de 2017). *About Estimote*. Obtenido de estimote: <https://estimote.com/about/>
- Filein, R. (23 de Septiembre de 2017). *SQLite: La Base de Datos Embebida*. Obtenido de SG Buzz: <https://sg.com.mx/revista/17/sqlite-la-base-datos-embebida>
- García Polo, E. (2008). *Técnicas de Localización en Redes Inalámbricas de Sensores*. Ciudad Real: Universidad de Castilla-La Mancha.
- García Vázquez, J. P., Galván Tejada, J. I., & Galván Tejada, C. E. (2013). Uso del campo magnetico de la tierra para localizar a las personas en interiores. *ISSN 2007-3585*, 32-36.

- Gómez Ruiz, C. A. (2015). Localización en espacios interiores de dispositivos móviles por medio de tecnologías WIFI, base para el desarrollo de aplicaciones sociales. *Encuentro Internacional de Educación en Ingeniería*, (págs. 1-9). Bogotá.
- González Merino, C. (2013). *ARQUITECTURA DE UN SISTEMA DE LOCALIZACIÓN EN REDES DE FEMTOCELDA EN INTERIORES*. Madrid: UNIVERSIDAD POLITÉCNICA DE MADRID.
- Google. (28 de Abril de 2017). *Entra en los edificios con los Mapas de interiores*. Obtenido de Google Maps: <https://www.google.com/maps/about/partners/indoormap/>
- GooglePlay. (22 de Septiembre de 2017). *Google Play*. Obtenido de BeaconCFG: [https://play.google.com/store/apps/details?id=com.yunliwuli.beaconcfg&hl=es\\_419](https://play.google.com/store/apps/details?id=com.yunliwuli.beaconcfg&hl=es_419)
- Gutiérrez, M. (27 de Abril de 2017). *El Androide Libre*. Obtenido de Todo sobre ZigBee, la tecnología ultrabarata para comunicación inalámbrica: <https://elandroidelibre.lespanol.com/2015/08/todo-sobre-zigbee-la-tecnologia-ultrabarata-para-comunicacion-inalambrica.html>
- Hightower, J., & Borriello, G. (2001). *A Survey and Taxonomy of Location Systems for Ubiquitous Computing*. Seattle: University of Washington.
- iArtHis\_LAB. (23 de Septiembre de 2017). *Trilateración*. Obtenido de Diccionario técnico para humanistas digitales: <http://iris.hdplus.es/dictionary/trilateracion/>
- Inc., A. (2015). *What is the RSS (Received Signal Strength)?* Recuperado el 25 de Abril de 2017, de Accuware Support: <https://www.accuware.com/support/knowledge-base/what-is-the-signal-strength-rss/>
- IndoorAtlas, L. (27 de Abril de 2017). *Making indoor worlds discoverable*. Obtenido de IndoorAtlas.com: <http://www.indooratlas.com/>
- Li, X., Xu, D., Wang, X., & Muhammad, R. (2016). Design and Implementation of Indoor Positioning System based on Ibeacon. *2016 International Conference on Audio, Language and Image Processing*, (págs. 126-130). Dubai.
- Liu, H., Darabi, H., Banerjee, P., & Liu, J. (2007). Survey of Wireless Indoor Positioning Techniques and Systems. *IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS*, 1067-1080.
- Moon, S. Y., & Kim, H. J. (2015). A Modeling Framework for Sensor Data in Indoor Location-based Services. *2015 4th International Conference on Advanced Information Technology and Sensor Application*, (págs. 67-70). Harbin.
- Muñoz Arcentales, A., Marín García, I., Calero Bravo, V., & Chávez Burbano, P. (2014). Sistema de Localización en Interiores Basado en Comunicaciones Ópticas no Guiadas por Luz Visible. Guayaquil.
- Navarro, E., Peuker, B., & Quan, M. (2010). *Wi-Fi Localization Using RSSI Fingerprinting*. San Luis Obispo: California Polytechnic State University.
- Networks, R. (22 de Septiembre de 2017). *Distance estimates*. Obtenido de Android Beacon Library: <http://altbeacon.github.io/android-beacon-library/distance-calculations.html>
- Núñez, M. (2009). *Sistema de localización en redes Wi-Fi basado en Fingerprint*. Madrid: Universidad Carlos III de Madrid.

- NYVIDA. (20 de Septiembre de 2017). *BEACON CONFIGURATION*. Obtenido de NYVIDA: <https://www.nyvida.com/start>
- Olaya, V. (2014). *Sistemas de Información Geográfica*. España: CreateSpace Independent Publishing Platform.
- Perez, R. (2013). *DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE POSICIONAMIENTO TERRESTRE UTILIZANDO PROTOCOLO ZIGBEE PARA PROVEER DE UN SISTEMA DE SEGURIDAD EN EL BARRIO SANTA LUCIA CENTRO DEL CANTÓN TISALEO*. LATACUNGA: ESPE.
- Rios, M. (2012). *SISTEMA DE LOCALIZACIÓN DE OBJETOS EN ESPACIOS CERRADOS POR MEDIO DE RF*. MÉXICO D. F: UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO.
- Ruiz, M. (2012). *Posicionamiento en Interiores Basado en Dispositivos Móviles*. Madrid: Universidad Autonoma de Madrid.
- SITUM, T. (27 de Abril de 2017). *El "GPS" para interiores*. Obtenido de Situm Indoor Positioning: <https://situm.es/es>
- tartigues. (13 de Octubre de 2014). *Calculadora online*. Recuperado el 20 de 08 de 2017, de CompartirWiFi: <http://www.compartirwifi.com/blog/calculadora-online-para-saber-el-alcance-de-un-router-o-antena-wifi-y-saber-la-distancia-maxima-a-la-que-llega/>
- Velasco, J. (19 de Septiembre de 2017). *¿En qué consiste Bluetooth LE?* Obtenido de Hipertextual: <https://hipertextual.com/2013/12/que-es-bluetooth-le>
- Wikipedia. (26 de Enero de 2016). *Distancia euclidiana*. Recuperado el 03 de Agosto de 2017, de Wikipedia: [https://es.wikipedia.org/wiki/Distancia\\_euclidiana](https://es.wikipedia.org/wiki/Distancia_euclidiana)
- Zhao, X., Xiao, Z., Markham, A., Trigoni, N., & Ren, Y. (2014). Does BTLE measure up against WiFi? A comparison of indoor location performance. *European Wireless 2014*, (págs. 263-268). Barcelona.

## Anexo I: Pruebas de los sistemas de localización en interiores con tecnología WiFi

Punto de Acceso (dirección MAC)	Punto
48:c0:93:1b:6d:50	a
48:c0:93:1b:6b:60	b
48:c0:93:1b:4d:60	c
50:60:28:be:3f:a0	d
48:c0:93:1b:16:10	e

Vector de Potencia									Localización Real	Localización Estimada	
a	-53	b	-50	c	-90	d	-90	e	-90	Laboratorio 7	Laboratorio 7
a	-53	b	-63	c	-90	d	-90	e	-90	Laboratorio 7	Laboratorio 7
a	-63	b	-59	c	-90	d	-86	e	-90	Laboratorio 7	Laboratorio 7
a	-60	b	-55	c	-90	d	-90	e	-90	Laboratorio 7	Laboratorio 7
a	-45	b	-42	c	-90	d	-82	e	-90	Laboratorio 7	Laboratorio 7
a	-53	b	-59	c	-90	d	-91	e	-90	Laboratorio 7	Laboratorio 7
a	-55	b	-57	c	-90	d	-91	e	-90	Laboratorio 7	Laboratorio 7
a	-52	b	-45	c	-90	d	-90	e	-90	Laboratorio 7	Laboratorio 7
a	-42	b	-48	c	-90	d	-90	e	-90	Laboratorio 7	Laboratorio 7
a	-90	b	-90	c	-69	d	-83	e	-90	Laboratorio 4	Laboratorio 4
a	-90	b	-90	c	-65	d	-89	e	-90	Laboratorio 4	Laboratorio 4
a	-90	b	-90	c	-74	d	-80	e	-90	Laboratorio 4	Laboratorio 4
a	-90	b	-90	c	-70	d	-90	e	-90	Laboratorio 4	Laboratorio 6
a	-90	b	-90	c	-67	d	-88	e	-90	Laboratorio 4	Laboratorio 4
a	-90	b	-90	c	-76	d	-78	e	-90	Laboratorio 4	Laboratorio 4
a	-90	b	-90	c	-69	d	-81	e	-90	Laboratorio 4	Laboratorio 4
a	-90	b	-90	c	-62	d	-88	e	-90	Laboratorio 4	Laboratorio 4
a	-90	b	-90	c	-70	d	-85	e	-90	Laboratorio 4	Laboratorio 4
a	-90	b	-90	c	-58	d	-90	e	-90	Laboratorio 5	Laboratorio 5
a	-90	b	-90	c	-52	d	-90	e	-90	Laboratorio 5	Laboratorio 5
a	-90	b	-90	c	-54	d	-90	e	-90	Laboratorio 5	Laboratorio 5
a	-90	b	-90	c	-56	d	-90	e	-90	Laboratorio 5	Laboratorio 5
a	-90	b	-90	c	-46	d	-92	e	-90	Laboratorio 5	Laboratorio 5
a	-90	b	-90	c	-54	d	-90	e	-90	Laboratorio 5	Laboratorio 5
a	-90	b	-90	c	-54	d	-93	e	-90	Laboratorio 5	Laboratorio 5
a	-90	b	-90	c	-46	d	-95	e	-90	Laboratorio 5	Laboratorio 5
a	-90	b	-90	c	-50	d	-90	e	-90	Laboratorio 5	Laboratorio 5
a	-87	b	-75	c	-69	d	-85	e	-90	Laboratorio 6	Laboratorio 6
a	-82	b	-80	c	-65	d	-81	e	-90	Laboratorio 6	Laboratorio 6
a	-83	b	-80	c	-62	d	-85	e	-90	Laboratorio 6	Laboratorio 6
a	-84	b	-82	c	-63	d	-79	e	-90	Laboratorio 6	Laboratorio 6
a	-84	b	-84	c	-67	d	-90	e	-90	Laboratorio 6	Laboratorio 6
a	-80	b	-87	c	-65	d	-90	e	-90	Laboratorio 6	Laboratorio 6
a	-86	b	-81	c	-64	d	-93	e	-90	Laboratorio 6	Laboratorio 6
a	-90	b	-81	c	-74	d	-95	e	-90	Laboratorio 6	Laboratorio 4
a	87	b	-82	c	-72	d	-90	e	-90	Laboratorio 6	Laboratorio 4
a	-71	b	-79	c	-90	d	-75	e	-90	Pasillo	Pasillo



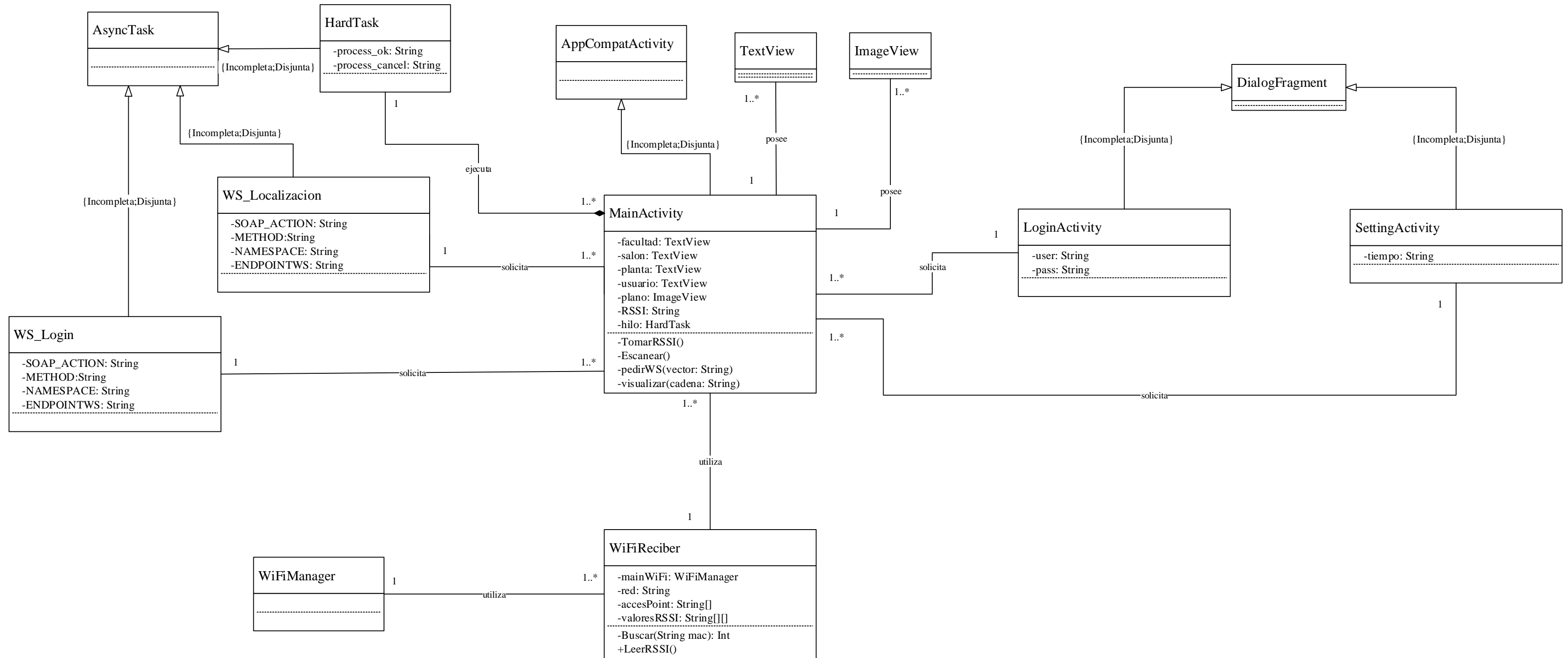
a	-68	b	-74	c	-90	d	-63	e	-90	Pasillo	Pasillo
a	-68	b	-61	c	-90	d	-70	e	-90	Pasillo	Laboratorio 7
a	-68	b	-76	c	-90	d	-64	e	-90	Pasillo	Pasillo

## Anexo II: Pruebas de los sistemas de localización en interiores con tecnología bluetooth

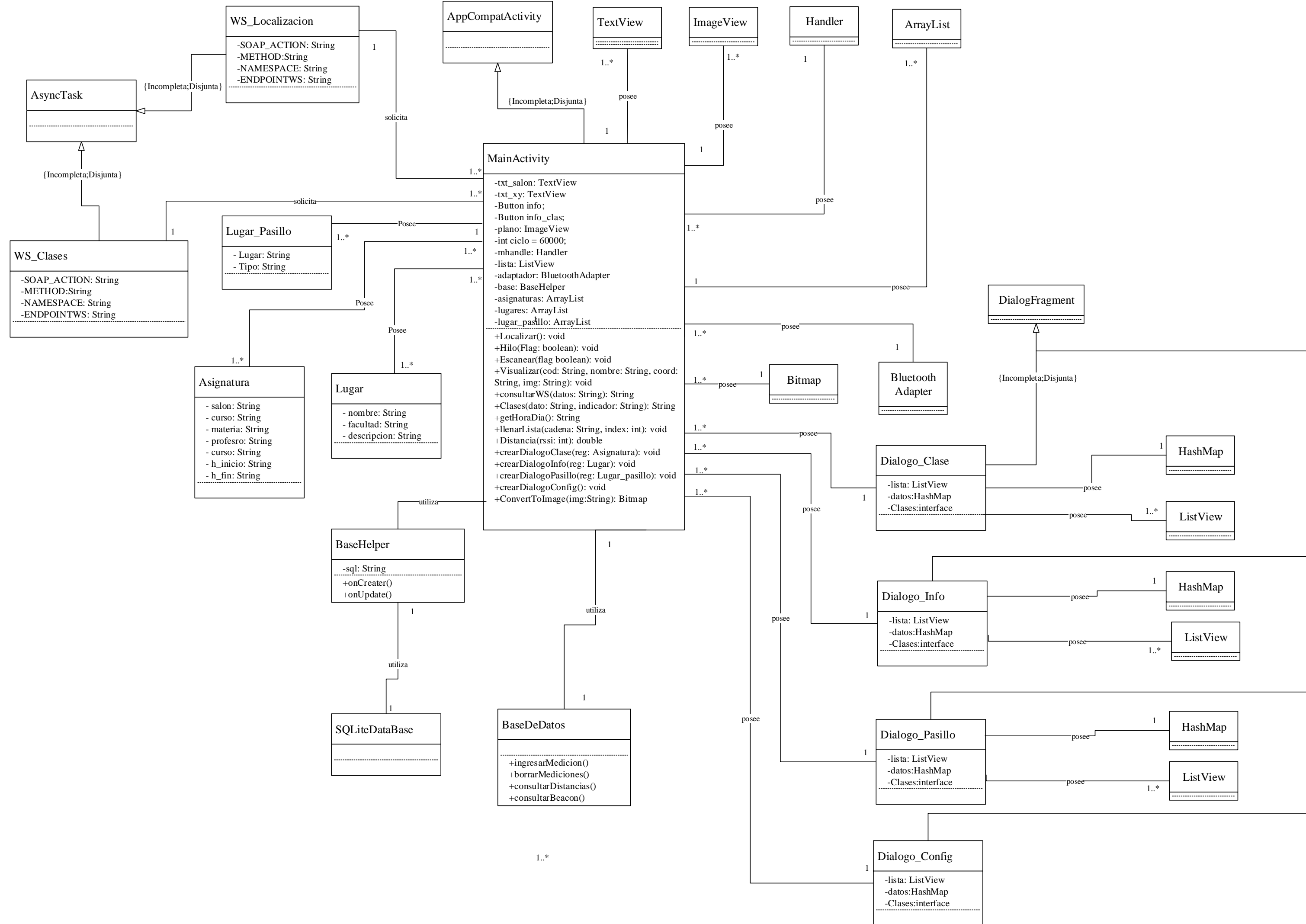
Primer Escenario				
Coordenada x-Real	Coordenada y-Real	Coordenada x-Estimada	Coordenada y-Estimada	Diferencia
2	11	6,75	12	4,85
6	11	3,74	10,64	2,29
11	11	16,8	8,7	6,24
3	6	9	7	6,08
3	2	1	0,97	2,25
6	2	5,4	0,85	1,30
11	2	12	2,7	1,22
11	6	14,36	7,68	3,76
13	2	12	0,7	1,64
17	2	12,74	6,07	5,89
23	2	26,2	1,95	3,20
25	2	23,8	5,12	3,34
25	6	26,79	2,85	3,62
25	11	25,56	10,68	0,64
30	11	27,74	10,75	2,27
35	11	28,65	11,37	6,36
35	6	36	12	6,08
35	2	36	4,55	2,74
<b>Promedio</b>				<b>3,54</b>

Segundo Escenario				
Coordenada x-Real	Coordenada y-Real	Coordenada x-Estimada	Coordenada y-Estimada	Diferencia
2	11	2	7,5	3,50
6	11	7,23	4,61	6,51
11	11	7,23	6,7	5,72
3	6	0,53	2,73	4,10
3	2	2,16	0,3	1,90
6	2	3,18	3,74	3,31
11	2	12,96	3,68	2,58
11	6	9,12	5,49	1,95
<b>Promedio</b>				<b>3,70</b>

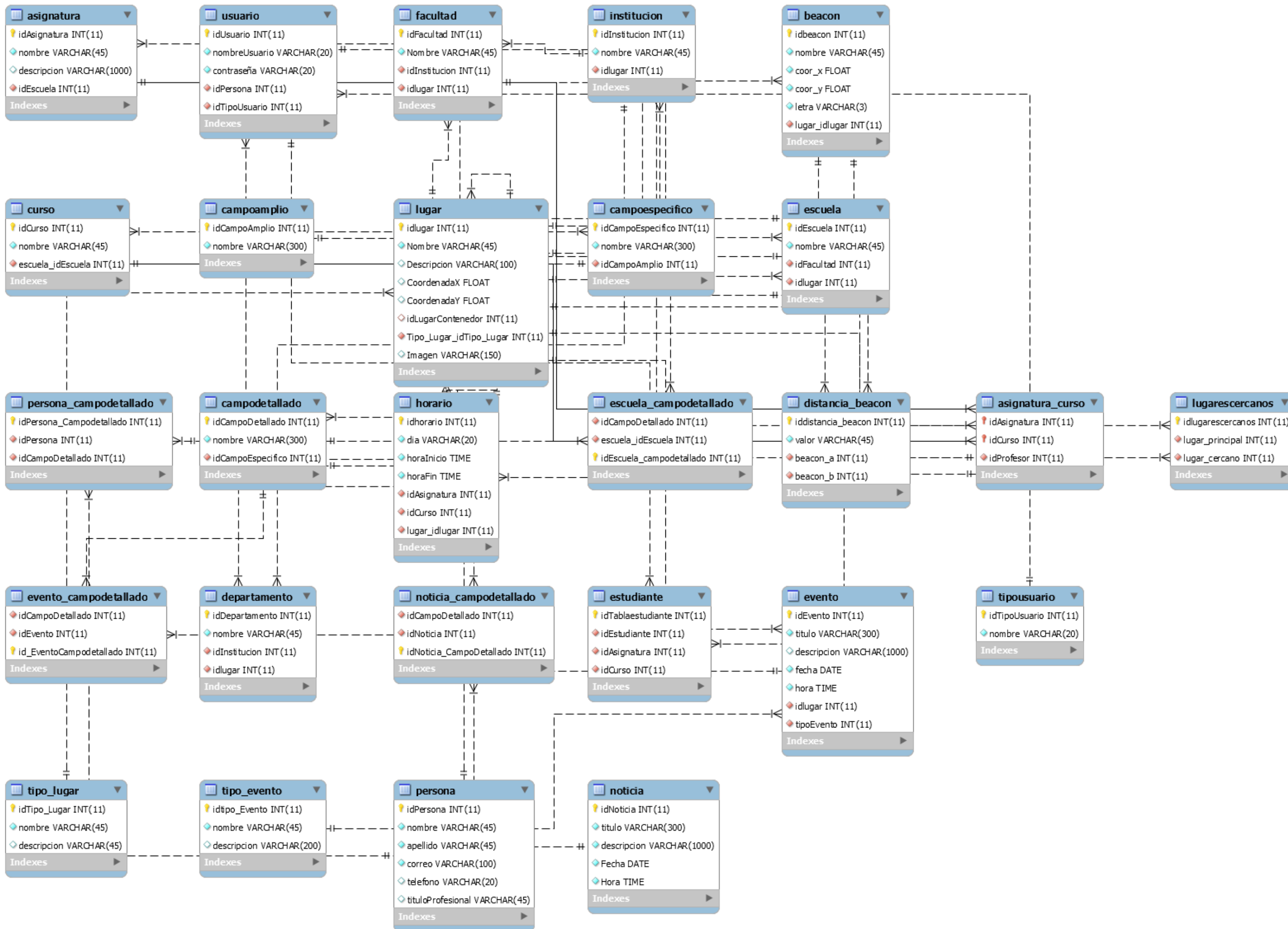
Anexo III: Diagrama de Clases del sistema localización en interiores con tecnología WiFi



**Anexo IV: Diagrama de Clases del sistema localización en interiores con tecnología bluetooth**



Anexo V: Diagrama de Clases del sistema localización en interiores con tecnología bluetooth



## Anexo VI: Código fuente del sistema de localización en interiores con tecnología WiFi

### Servidor de Aplicaciones

```
import javax.jws.WebService;
import javax.jws.WebMethod;
import javax.jws.WebParam;
import javax.swing.JOptionPane;

@WebService(serviceName = "Localizacion")

public class Localizacion {

    /*Servicio Web que llama al método consultar de la clase Localizador, tomando como
    parámetro una
    cadena con los las direcciones MAC de los access point y el valor RSSI para cada
    uno
    */

    @WebMethod(operationName = "localizar")
    public String[] localizar(@WebParam(name = "rssi") String rssi) {
        Localizador gps = new Localizador();
        String datos[] = gps.consultar(rssi);
        return datos;
    }

    /*Servicio Web que llama al método login de la clase Localizador, tomando como
    parámetro una
    cadena con el nombre de usuario y la contraseña para validar el inicio de
    sesion
    */

    @WebMethod(operationName = "login")
    public String login(@WebParam(name = "user") String user, @WebParam(name = "pass")
String pass) {
        Localizador gps = new Localizador();
        String datos = gps.login(user, pass);
        return datos;
    }
}

import java.io.File;
import java.io.FileInputStream;
import java.io.InputStream;
import java.sql.*;
import java.util.ArrayList;
import java.util.Base64;
import java.util.logging.Level;
import java.util.logging.Logger;

public class Localizador {

    ArrayList<String> lista = new ArrayList();
    String conexionBD = "jdbc:mysql://localhost:3306/localizacion";
    Connection conexion = null;
    String con;
    ResultSet rs;
    Statement s;
    String res = "";

    /*
    Constructor que inicializa la clase Localizador y crea la conexión hacia la base
    de datos
    */

    public Localizador() {
        try {
```

```

        Class.forName("com.mysql.jdbc.Driver");
        conexion = DriverManager.getConnection(conexionBD, "root", "root");
        s = conexion.createStatement();
    } catch (Exception e) {
        System.err.println(e.toString());
    }
}

/*
Método que genera la cadena de resultado que sera enviada al Smartphone,
llama a un ordenamiento de la cadena que llega de los niveles RSSI,
llama a la clase que implementa el algoritmo kNN y busca la información del lugar
que dio como resultado
*/

public String[] consultar(String rssi) {
    String respuesta[] = new String[4];
    KNN algoritmo = new KNN();
    res = algoritmo.Localizar(armarCadena(rssi));
    con = "select s.nombre,s.planta,f.nombre,s.imagen \n"
        + "from facultad f,salon s \n"
        + "where s.id_salon='" + res + "' \n"
        + "and s.id_facultad = f.id_facultad";
    try {
        rs = s.executeQuery(con);
        while (rs.next()) {
            respuesta[0] = rs.getString(1);
            respuesta[1] = rs.getString(2);
            respuesta[2] = rs.getString(3);
            respuesta[3] = establecerImg(rs.getString(4).trim());
        }
    } catch (SQLException ex) {
        Logger.getLogger(Localizador.class.getName()).log(Level.SEVERE, null, ex);
    }
    return respuesta;
}

/*
Realiza una compresión en Base64 de la imagen asociada al lugar de ubicación
para enviarlo al Smartphone como cadena de texto
*/

public String establecerImg(String img) {
    Base64.Encoder encoder = Base64.getEncoder();
    File file = new File(img.trim());
    byte[] fileArray = new byte[(int) file.length()];
    InputStream inputStream;
    String encodedFile = "";
    try {
        inputStream = new FileInputStream(file);
        inputStream.read(fileArray);
        encodedFile = encoder.encodeToString(fileArray);
    } catch (Exception e) {
        System.err.println(e.toString());
    }
    return encodedFile;
}

/*
Método que reemplaza la dirección MAC con su literal correspondiente, para facilitar
al algoritmo kNN.
*/

private String armarCadena(String rssi) {
    String aux = "";
    String cadenas[] = rssi.split("/");
    for (int i = 0; i < cadenas.length; i++) {
        if (i > 0) {
            aux += "/";
        }
    }
}

```

```

        con = "select id_ap from puntos_acceso where mac='" +
cadenas[i].split(";")[0] + "'";
        try {
            rs = s.executeQuery(con);
            while (rs.next()) {
                aux += rs.getString(1).trim() + ";" + (cadenas[i].split(";")[1]);
            }
        } catch (SQLException ex) {
            Logger.getLogger(Localizador.class.getName()).log(Level.SEVERE, null,
ex);
        }
    }
    return aux;
}

/*
Método encargado de realizar el inicio de sesión del usuario
*/

public String login(String user, String pass) {
    String log = "";
    try {
        con = "select nombre, apellido, tipo_usuario from usuario where
id_usuario='" + user + "' and password='" + pass + "'";
        rs = s.executeQuery(con);
        while (rs.next()) {
            log += rs.getString(3).trim() + ";" + rs.getString(1).trim() + ";" +
rs.getString(2);
        }
    } catch (Exception e) {
        Logger.getLogger(Localizador.class.getName()).log(Level.SEVERE, null, e);
    }
    return log;
}
}

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.ArrayList;
import java.util.logging.Level;
import java.util.logging.Logger;

public class KNN {

    ArrayList<String> lista = new ArrayList();
    String conexionBD = "jdbc:mysql://localhost:3306/localizacion";
    Connection conexion = null;
    String con;
    ResultSet rs;
    Statement s;
    String res = "";

    /*
    Constructor que inicializa la clase y crea la conexión con la base de datos
    */

    public KNN() {
        try {
            Class.forName("com.mysql.jdbc.Driver");
            conexion = DriverManager.getConnection(conexionBD, "root", "root");
            s = conexion.createStatement();
        } catch (Exception e) {
            System.err.println(e.toString());
        }
    }

    /*

```

```

Método encargado de obtener el total de puntos de referencia que se han tomado
*/

public int TotalPuntos() {
    int res = 0;
    con = "select count(ID_PUNTO) from puntos_medicion";
    try {
        rs = s.executeQuery(con);
        while (rs.next()) {
            res = Integer.parseInt(rs.getString(1));
        }
    } catch (SQLException ex) {
        Logger.getLogger(Localizador.class.getName()).log(Level.SEVERE, null, ex);
    }
    return res;
}

/*
Método que aplica el algoritmo kNN
*/

public String Localizar(String rssi) {
    String res = "";
    String linea[][] = ordenado(rssi);
    String aux[][] = new String[5][2];
    double min = 100;
    int j, i_min = 0;
    try {
        int puntos = TotalPuntos();
        for (int i = 1; i <= puntos; i++) {
            con = "select * from medicion where id_punto=" + i;
            rs = s.executeQuery(con);
            j = 0;
            while (rs.next()) {
                aux[j][0] = rs.getString(3);
                aux[j][1] = rs.getString(2);
                j++;
            }
            double dist = distancia(linea, aux);
            if (min > dist) {
                min = dist;
                i_min = i;
            }
        }
        con = "select id_salon from puntos_medicion where id_punto=" + i_min;
        rs = s.executeQuery(con);
        while (rs.next()) {
            res = rs.getString(1);
        }
    } catch (SQLException ex) {
        Logger.getLogger(Localizador.class.getName()).log(Level.SEVERE, null, ex);
    }
    return res.trim();
}

/*
Método que determina el valor de la distancia euclidiana entre dos puntos
*/

private double distancia(String[][] a, String[][] b) {
    double dist = 0;
    for (int i = 0; i < 5; i++) {
        dist += Math.pow((Double.parseDouble(a[i][1]) -
Double.parseDouble(b[i][1])), 2);
    }
    dist = Math.sqrt(dist) / 5;
    return dist;
}

/*

```



```

        Ordena la cadena suministrada al inicio, basandose en el literal asociado a cada
dirección MAC de
        cada access point
        */

private static String[][] ordenado(String linea) {
    String id[] = new String[]{"a", "b", "c", "d", "e"};
    int letras = 0;
    String res[][] = new String[5][2];
    String cadenas[] = linea.split("/");
    while (letras < 5) {
        for (int i = 0; i < cadenas.length; i++) {
            if (id[letras].equals(cadenas[i].split(";")[0])) {
                res[letras][0] = id[letras];
                res[letras][1] = cadenas[i].split(";")[1];
                letras++;
            }
        }
    }
    return res;
}
}
}

```

## Aplicación Móvil

```

import android.app.ProgressDialog;
import android.content.DialogInterface;
import android.content.SharedPreferences;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.os.AsyncTask;
import android.os.Bundle;
import android.support.design.widget.FloatingActionButton;
import android.support.design.widget.Snackbar;
import android.support.v7.app.AppCompatActivity;
import android.support.v7.widget.Toolbar;
import android.util.Base64;
import android.view.View;
import android.view.Menu;
import android.view.MenuItem;

import java.io.ByteArrayInputStream;
import java.io.InputStream;
import java.util.ArrayList;
import java.util.List;

import android.app.Activity;
import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.content.IntentFilter;
import android.net.wifi.ScanResult;
import android.net.wifi.WifiManager;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.AdapterView;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.ListAdapter;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity implements LoginActivity.Login,
SettingActivity.Setting{

    TextView facultad,salon,planta,usuario;

```

```

    ImageView imagen;
    String RSSI="";
    boolean flag=false;
    FloatingActionButton fab,aux;
    int inicial=0;

    /*
    Metodo sobrescrito que inicializa la app, conjuntamente con sus elementos graficos y
    otros metodo mas
    */

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
        facultad = (TextView) findViewById(R.id.facultad);
        salon = (TextView) findViewById(R.id.salon);
        planta = (TextView) findViewById(R.id.planta);
        imagen = (ImageView) findViewById(R.id.imageView);
        usuario = (TextView) findViewById(R.id.usuario);

        setSupportActionBar(toolbar);
        fab = (FloatingActionButton) findViewById(R.id.fab);
        aux = (FloatingActionButton) findViewById(R.id.fab2);
        fab.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                TomarRSSI();
            }
        });
    }

    /*
    Metodo asociado al evento click para presentar mensaje de actualizacion
    */

    aux.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            Snackbar.make(view, R.string.msg_act, Snackbar.LENGTH_LONG)
                .setAction("Action", null).show();
        }
    });

    /*
    Obtencion de los valores almacenados como preferencias, como son el nombre del
    usuario y
    el valor para realizar la actualizacion de la localizacion
    */

    SharedPreferences pref = getSharedPreferences("datos",Context.MODE_PRIVATE);
    if((pref.getString("tipo","def").equals("def"))){
        crearDialogoLogin();
        flag=true;
    }
    else{
        String inf = pref.getString("tipo", "valor") + ": " +
        pref.getString("nombre", "valor") + " " + pref.getString("apellido", "valor");
        usuario.setText(inf);
        flag=false;
        inicial=0;
        this.TomarRSSI();
    }

    }

    /*
    Metodo que llama al hilo encargado de leer los valores RSSI de los access pont
    */

    private void TomarRSSI() {

```

```

        EstadoOriginal(false);
        aux.callOnClick();
        final HardTask Task = new HardTask(MainActivity.this);
        Task.execute("some_param");
    }

    /*
    Metodo que inicializa la clase WifiReciber para leer los valores RSSI
    */

    private void Escanear() {
        Intent abreLista = new Intent(MainActivity.this, WifiReceiver.class);
        startActivityForResult(abreLista, 2);
    }

    // Broadcast receiver class called its receive method
    // when number of wifi connections changed

    /*
    Metodo llamado al terminar de llamarse a la clase WiFiReciber
    */

    protected void onActivityResult(int requestCode, int resultCode, Intent data) {
        switch (requestCode) {
            case 2:
                if (resultCode == Activity.RESULT_OK) {
                    String val=data.getExtras().getString("valores");
                    RSSI=val;
                } else {
                    finish();
                }
                break;
        }
    }

    /*
    Metodo sobrescrito que crea el meno de la app
    */

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.menu_main, menu);
        return true;
    }

    /*
    Metodo que limpia los textarea de la app y limpia el espacio de la imagen
    */

    public void EstadoOriginal(boolean err){
        facultad.setText("");
        planta.setText("");
        salon.setText("");
        if(err){
            imagen.setImageResource(R.mipmap.no_licalizado);
        }else{
            imagen.setImageResource(R.color.colorPrimaryDark);
        }
    }

    /*
    Metodo sobrescrito que llama al servicio web de localizacion
    */
    private void pedirWS(String maq){
        String res="";
        WS_Localizacion gps = (WS_Localizacion) new WS_Localizacion().execute(maq);
        try {
            res = gps.get();
            visualizar(res);
        }
    }

```

```

        // Toast.makeText(this,res,Toast.LENGTH_LONG).show();
    } catch (Exception e) {
        Toast.makeText(this,e.toString(),Toast.LENGTH_LONG).show();
    }
}

/**
Metodo encargado de visualizar los resultados obtenidos desde el servicio web
*/
private void visualizar(String cadena){
    String aux = cadena.substring(1,cadena.length()-1);
    String datos[] = aux.split(",");
    salon.setText("SalÃ³n: "+datos[0].trim());
    Toast.makeText(this,datos[0].trim(),Toast.LENGTH_LONG).show();
    planta.setText("#Planta: "+datos[1].trim());
    facultad.setText("Facultad: "+datos[2].trim());
    imagen.setImageBitmap(ConvertToImage(datos[3].trim()));
}

/**
Metodo sobrescrito asociado al evento de seleccion de un elemento del menu
*/
@Override
public boolean onOptionsItemSelected(MenuItem item) {

    int id = item.getItemId();
    if (id == R.id.opt_user) {
        crearDialogoLogin();
        return true;
    }else if(id == R.id.opt_conf){
        crearDialogoSetting();
        return true;
    }
    return super.onOptionsItemSelected(item);
}

/**
Metodo que genera el dialogo de login
*/
private void crearDialogoLogin(){
    LoginActivity login = new LoginActivity();
    login.show(getSupportFragmentManager(),"ingreso");

    android.app.Fragment frag = getSupportFragmentManager().findFragmentByTag("ingreso");

    if (frag != null) {
        getSupportFragmentManager().beginTransaction().remove(frag).commit();
    }
}

/**
Metodo que genera el dialogo para configurar el tiempo de actualizacion
*/
private void crearDialogoSetting(){
    SettingActivity login = new SettingActivity();
    login.show(getSupportFragmentManager(),"tiempo");

    android.app.Fragment frag = getSupportFragmentManager().findFragmentByTag("tiempo");

    if (frag != null) {
        getSupportFragmentManager().beginTransaction().remove(frag).commit();
    }
}

/**
Metodo que se llama despues de finalizado el dialogo de login
*/
@Override
public void FinalizaLogin(String nombre) {
    String cadena[]=nombre.split(",");
    SharedPreferences pref = getSharedPreferences("datos",Context.MODE_PRIVATE);

```

```

        SharedPreferences.Editor edit = pref.edit();
        edit.putString("tipo",cadena[0]);
        edit.putString("nombre",cadena[1]);
        edit.putString("apellido",cadena[2]);
        edit.putString("tiempo","30");
        edit.commit();
        String inf = pref.getString("tipo", "valor") + ": " + pref.getString("nombre",
"valor") + " " + pref.getString("apellido", "valor");
        usuario.setText(inf);
        if(flag){
            this.TomarRSSI();
        }
    }

    /*
    Metodo que se llama despues de finalizado el dialogo de configuracion
    */
    @Override
    public void FinalizaSetting(String cant) {
        SharedPreferences pref = getSharedPreferences("datos",Context.MODE_PRIVATE);
        SharedPreferences.Editor edit = pref.edit();
        edit.putString("tiempo",cant);
        edit.commit();
    }
    /*
    Metodo que descomprime la imagen que llega como cadena de texto
    */
    public Bitmap ConvertToImage(String image){
        try{
            InputStream stream = new
            ByteArrayInputStream(Base64.decode(image.getBytes(), Base64.DEFAULT));
            Bitmap bitmap = BitmapFactory.decodeStream(stream);
            Log.v("Ben", "Image Converted");
            return bitmap;
        }
        catch (Exception e) {
            EstadoOriginal(true);
            return null;
        }
    }
    /*
    Clase que implementa un hilo encargado de leer los valores RSSI
    */
    private class HardTask extends AsyncTask<String, Integer, String> {

        private Context context;
        private static final String PROCESS_OK = "PROCESS_OK";
        private static final String PROCESS_ERROR = "PROCESS_ERROR";

        public HardTask(Context context) {
            this.context = context;
        }
    }
    /*
    Metodo que llama al Escaneo durante 8 segundos
    */
    @Override
    protected String doInBackground(String... params) {
        for (int i = 0; i <= 80; i++) {
            publishProgress(i);
            try {
                Escanear();
                Thread.sleep(100);
            } catch (InterruptedException e) {
                e.printStackTrace();
                return PROCESS_ERROR;
            }
        }
        return PROCESS_OK;
    }
    /*
    Metodo sobrescrito que se ejecuta despues del hilo

```

```

    */
    @Override
    protected void onPreExecute() {
        super.onPreExecute();
    }
    /*
    Metodo sobrescrito que se ejecuta durante el hilo
    */
    @Override
    protected void onProgressUpdate(Integer... progress) {
        super.onProgressUpdate(progress);
    }
    /*
    Metodo sobrescrito que se ejecuta despues el hilo
    */
    @Override
    protected void onPostExecute(String result) {
        if (PROCESS_OK.equals(result)) {
            pedirWS(RSSI);
        } else {
            Toast.makeText(context, "Process ERROR " + result,
                Toast.LENGTH_LONG).show();
        }
    }
}

import android.animation.Animator;
import android.animation.AnimatorListenerAdapter;
import android.annotation.TargetApi;
import android.support.v4.app.DialogFragment;
import android.content.Context;
import android.content.SharedPreferences;
import android.content.pm.PackageManager;
import android.support.annotation.NonNull;
import android.support.design.widget.FloatingActionButton;
import android.support.design.widget.Snackbar;
import android.support.design.widget.TextInputLayout;
import android.support.v7.app.AppCompatActivity;
import android.app.LoaderManager.LoaderCallbacks;

import android.content.CursorLoader;
import android.content.Loader;
import android.database.Cursor;
import android.net.Uri;
import android.os.AsyncTask;

import android.os.Build;
import android.os.Bundle;
import android.provider.ContactsContract;
import android.support.v7.widget.Toolbar;
import android.text.TextUtils;
import android.view.KeyEvent;
import android.view.LayoutInflater;
import android.view.View;
import android.view.View.OnClickListener;
import android.view.ViewGroup;
import android.view.inputmethod.EditorInfo;
import android.view.inputmethod.InputMethodManager;
import android.widget.AdapterView;
import android.widget.AutoCompleteTextView;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;

import java.util.ArrayList;
import java.util.List;

import static android.Manifest.permission.READ_CONTACTS;

```

```

/**
 * A login screen that offers login via email/password.
 */
public class LoginActivity extends DialogFragment {

    EditText user,pass;
    Button boton;
    Login interfaz=null;

    public interface Login {
        void FinalizaLogin(String nombre);
    }
    /**
    Metodo sobrescrito que incializa el dialogo y sus elementos graficos
    */
    @Override
    public View onCreateView(final LayoutInflater inflater, ViewGroup container, Bundle
savedInstanceState) {
        View view = inflater.inflate(R.layout.activity_login, container);
        super.onCreate(savedInstanceState);
        user = (EditText) view.findViewById(R.id.user);
        pass = (EditText) view.findViewById(R.id.pass);
        boton = (Button) view.findViewById(R.id.button);

        boton.setOnClickListener(new OnClickListener() {
            /**
            Metodo sobrescrito asociado al evento click del boton
            */
            @Override
            public void onClick(View v) {
                WS_Login ingreso = (WS_Login) new
WS_Login().execute(user.getText().toString(),pass.getText().toString());
                try {
                    String res = ingreso.get();
                    if(!res.isEmpty()) {
                        interfaz = (LoginActivity.Login) getActivity();
                        interfaz.FinalizaLogin(res);
                        InputMethodManager inputMethodManager =
(InputMethodManager)v.getContext().getSystemService(Context.INPUT_METHOD_SERVICE);
inputMethodManager.hideSoftInputFromWindow(user.getWindowToken(), 0);

inputMethodManager.hideSoftInputFromWindow(pass.getWindowToken(), 0);
                        dismiss();
                    }
                    else
                        Toast.makeText(getActivity().getBaseContext(),"Usuario o
Contraseña incorrecta",Toast.LENGTH_LONG).show();
                } catch (Exception e) {
                    Toast.makeText(getActivity().getBaseContext(),e.toString(),Toast.LENGTH_LONG).show();
                }
            }
        });
        getDialog().setTitle("Login");
        return view;
    }
}

import android.os.Bundle;
import android.support.v4.app.DialogFragment;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

/**

```

```

* Created by Marco on 06/08/2017.
*/

public class SettingActivity extends DialogFragment {
    EditText tiempo;
    Button boton;
    LoginActivity.Login interfaz=null;

    public interface Setting {
        void FinalizaSetting(String cant);
    }
    /*
    Metodo sobrescrito que inicializa el dialgo y sus componentes
    */
    @Override
    public View onCreateView(final LayoutInflater inflater, ViewGroup container, Bundle
savedInstanceState) {
        View view = inflater.inflate(R.layout.activity_setting, container);
        super.onCreate(savedInstanceState);
        tiempo = (EditText) view.findViewById(R.id.tiempo);
        boton = (Button) view.findViewById(R.id.btn_tiempo);

        /*
        Metodo sobrescrito asociado al evento click del boton
        */
        boton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                interfaz = (LoginActivity.Login) getActivity();

interfaz.FinalizaLogin(Integer.parseInt(tiempo.getText().toString())+"");
                dismiss();
            }
        });
        getDialog().setTitle("Configurar Tiempo");
        return view;
    }
}

import android.app.Activity;
import android.content.Context;
import android.content.Intent;
import android.content.IntentFilter;
import android.net.wifi.ScanResult;
import android.net.wifi.WifiManager;
import android.os.Bundle;
import android.util.Log;
import android.widget.ArrayAdapter;
import android.widget.Toast;

import java.util.ArrayList;
import java.util.List;

/**
 * Created by Marco on 07/06/2017.
 */

public class WifiReceiver extends Activity {

    WifiManager mainWifi;
    List<ScanResult> wifiList;
    public static String res ="";
    String red="WiFi-UDA";
    String []
AC={"48:c0:93:1b:6b:50","48:c0:93:1b:6b:60","48:c0:93:1b:4d:60","50:60:28:be:3f:a0","48:
c0:93:1b:16:10"};
    private String [][]valores;

    public String[][] getValores(){
        return valores;
    }
}

```



```

/*
Metodo que inicializa la clase
*/
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    Log.i("acpe", "llego");
    mainWifi = (WifiManager) getSystemService(Context.WIFI_SERVICE);
    if (mainWifi.isWifiEnabled() == false) {
        // If wifi disabled then enable it
        Toast.makeText(getApplicationContext(), "wifi is disabled..making it
enabled", Toast.LENGTH_LONG).show();
        mainWifi.setWifiEnabled(true);
    }
    mainWifi.startScan();
    leerRSSI();
    Intent retornar = new Intent();
    retornar.putExtra("valores", res.substring(0, res.length()-1));
    setResult(RESULT_OK, retornar);
    finish();
}
/*
Metodo busca si los access point censados son los correctos
*/
private int Buscar(String mac) {
    int i=0;
    while (i<4){
        if(!AC[i].equals(mac)){
            i++;
        }else
            break;
    }
    if (i==4)
        return -1;
    else
        return i;
}
/*
Metodo que lee los valores RSSI
*/
public void leerRSSI() {
    ArrayList<String> li = new ArrayList<String>();
    wifiList = mainWifi.getScanResults();
    for (int i = 0; i < wifiList.size(); i++) {
        valores = new String[4][2];
        if (wifiList.get(i).SSID.equals(red)) {
            if (Buscar(wifiList.get(i).BSSID) != -1) {
                li.add(wifiList.get(i).BSSID + ";" + wifiList.get(i).level);
                Log.i("ac", wifiList.get(i).level + "");
            }
        }
        res = "";
        for (int j = 0; j < li.size(); j++) {
            res += li.get(j);
            res += "/";
        }
        for (int k=0;k<AC.length;k++){
            if(res.indexOf(AC[k])!=-1){
                res+=AC[k]+";-90/";
            }
        }
    }
}
}

import android.os.AsyncTask;
import org.ksoap2.SoapEnvelope;
import org.ksoap2.serialization.SoapObject;
import org.ksoap2.serialization.SoapSerializationEnvelope;
import org.ksoap2.transport.HttpTransportSE;

```

```

/**
 * Created by Marco on 25/07/2017.
 */

public class WS_Localizacion extends AsyncTask<String, Void, String> {

    final String SOAP_ACTION = "http://localizacion/localizar";
    final String METHOD = "localizar";
    final String NAMESPACE = "http://localizacion/";
    final String ENDPOINTWS = "http://172.16.1.47:8080/ServiciosWeb/Localizacion?xsd=1";
    String respuesta = null;
    /*
    Metodo que inicializa la clase y llama al servicio web
    */
    @Override
    protected String doInBackground(String... args) {
        SoapObject userRequest = new SoapObject(NAMESPACE, METHOD);
        userRequest.addProperty("rssi", args[0]);

        SoapSerializationEnvelope envelope = new
        SoapSerializationEnvelope(SoapEnvelope.VER11);
        envelope.setOutputSoapObject(userRequest);

        try{
            HttpTransportSE androidHttpTransport = new HttpTransportSE(ENDPOINTWS);
            androidHttpTransport.debug = true;
            androidHttpTransport.call(SOAP_ACTION, envelope);

            respuesta = envelope.getResponse().toString();
        }
        catch (Exception e){
            e.printStackTrace();
        }
        return respuesta;
    }
}

package com.example.marco.interiores;

import android.os.AsyncTask;

import org.ksoap2.SoapEnvelope;
import org.ksoap2.serialization.SoapObject;
import org.ksoap2.serialization.SoapSerializationEnvelope;
import org.ksoap2.transport.HttpTransportSE;

/**
 * Created by Marco on 25/07/2017.
 */

public class WS_Login extends AsyncTask<String, Void, String> {

    final String SOAP_ACTION = "http://localizacion/login";
    final String METHOD = "login";
    final String NAMESPACE = "http://localizacion/";
    final String ENDPOINTWS = "http://172.16.1.47:8080/ServiciosWeb/Localizacion?xsd=1";
    String respuesta = null;
    /*
    Metodo sobrescrito que inicializa la clase y llama al servicio web
    */
    @Override
    protected String doInBackground(String... args) {
        SoapObject userRequest = new SoapObject(NAMESPACE, METHOD);
        userRequest.addProperty("user", args[0]);
        userRequest.addProperty("pass", args[1]);

        SoapSerializationEnvelope envelope = new
        SoapSerializationEnvelope(SoapEnvelope.VER11);
        envelope.setOutputSoapObject(userRequest);

```

```

try{
    HttpTransportSE androidHttpTransport = new HttpTransportSE(ENDPOINTWS);
    androidHttpTransport.debug = true;
    androidHttpTransport.call(SOAP_ACTION, envelope);

    respuesta = envelope.getResponse().toString();
}
catch (Exception e){
    e.printStackTrace();
}
return respuesta;
}
}

```

```

<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context="com.example.marco.localizador.MainActivity">

```

```

<LinearLayout
    android:layout_width="336dp"
    android:layout_height="498dp"
    android:background="@color/colorPrimaryDark"
    android:orientation="vertical"
    android:weightSum="1"
    tools:layout_editor_absoluteX="8dp"
    tools:layout_editor_absoluteY="8dp">

```

```

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="@color/colorPrimaryDark"
    android:orientation="horizontal"
    android:weightSum="1">

```

```

<TextView
    android:id="@+id/txt_salon"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_weight="0.87"
    android:background="@color/colorPrimaryDark"
    android:textColor="@color/letra_blanca" />

```

```

<TextView
    android:id="@+id/txt_xy"
    android:layout_width="128dp"
    android:layout_height="wrap_content"
    android:background="@color/colorPrimaryDark"
    android:textColor="@color/letra_blanca" />

```

```

</LinearLayout>

```

```

<ImageView
    android:id="@+id/imagen"
    android:layout_width="324dp"
    android:layout_height="299dp"
    android:background="@color/letra_blanca"
    app:srcCompat="@mipmap/cero" />

```

```

<TextView
    android:id="@+id/textView3"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="@color/colorPrimary"
    android:text="Información Contextual"
    android:textColor="@color/letra_blanca"

```

```

        android:textStyle="bold" />

<ListView
    android:id="@+id/lista"
    android:layout_width="324dp"
    android:layout_height="match_parent"
    android:background="@color/letra_blanca"
    android:divider="@color/colorPrimary"
    android:dividerHeight="1dp"
    android:scrollbars="horizontal|vertical" />

</LinearLayout>
</android.support.constraint.ConstraintLayout>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/letra_blanca"
    android:orientation="vertical"
    tools:context="layout.Dialogo_Clase">

<!-- TODO: Update blank fragment layout -->

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal">

<LinearLayout
    android:layout_width="174dp"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:weightSum="1">

<TextView
    android:id="@+id/textView8"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="@color/letra_blanca"
    android:gravity="right"
    android:text="@string/salon"
    android:textAlignment="inherit"
    android:textColor="@color/colorPrimary"
    android:textStyle="bold" />

<TextView
    android:id="@+id/textView11"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="@color/letra_blanca"
    android:gravity="right"
    android:text="@string/materia"
    android:textColor="@color/colorPrimary"
    android:textStyle="bold" />

<TextView
    android:id="@+id/textView13"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="@color/letra_blanca"
    android:gravity="right"
    android:text="@string/prof"
    android:textColor="@color/colorPrimary"
    android:textStyle="bold" />

<TextView
    android:id="@+id/textView15"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="@color/letra_blanca"

```

```

        android:gravity="right"
        android:text="@string/ini"
        android:textColor="@color/colorPrimary"
        android:textStyle="bold" />

<TextView
    android:id="@+id/textView17"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="@color/letra_blanca"
    android:gravity="right"
    android:text="@string/fin"
    android:textColor="@color/colorPrimary"
    android:textStyle="bold" />

</LinearLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/letra_blanca"
    android:orientation="vertical">

    <TextView
        android:id="@+id/txt_lab"
        android:layout_width="143dp"
        android:layout_height="wrap_content"
        android:background="@color/letra_blanca"
        android:gravity="left"
        android:text="TextView"
        android:textStyle="bold" />

    <TextView
        android:id="@+id/txt_mat"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:gravity="left"
        android:text="TextView"
        android:textStyle="bold" />

    <TextView
        android:id="@+id/txt_prof"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:gravity="left"
        android:text="TextView"
        android:textStyle="bold" />

    <TextView
        android:id="@+id/txt_ini"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:gravity="left"
        android:text="TextView"
        android:textStyle="bold" />

    <TextView
        android:id="@+id/txt_fin"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:gravity="left"
        android:text="TextView"
        android:textStyle="bold" />

</LinearLayout>
</LinearLayout>

```

```

    <Button
        android:id="@+id/boton"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Aceptar"
        android:textStyle="bold" />

</LinearLayout>

```

## Anexo VII: Script de la Base de Datos del sistema de localización en interiores con tecnología WiFi.

```

CREATE DATABASE IF NOT EXISTS `localizacion` /*!40100 DEFAULT CHARACTER SET utf8 */;
USE `localizacion`;
-- MySQL dump 10.13 Distrib 5.7.17, for Win64 (x86_64)
--
-- Host: localhost Database: localizacion
-- -----
-- Server version 5.7.19-log

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8 */;
/*!40103 SET @OLD_TIME_ZONE=@@TIME_ZONE */;
/*!40103 SET TIME_ZONE='+00:00' */;
/*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0 */;
/*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0 */;
/*!40101 SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;
/*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;

--
-- Table structure for table `facultad`
--

DROP TABLE IF EXISTS `facultad`;
/*!40101 SET @saved_cs_client = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `facultad` (
  `id_facultad` char(3) NOT NULL,
  `nombre` char(18) DEFAULT NULL,
  `num_plantas` int(11) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `facultad`
--

LOCK TABLES `facultad` WRITE;
/*!40000 ALTER TABLE `facultad` DISABLE KEYS */;
INSERT INTO `facultad` VALUES ('f1','Filosofia',4);
/*!40000 ALTER TABLE `facultad` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Table structure for table `medicion`
--

DROP TABLE IF EXISTS `medicion`;
/*!40101 SET @saved_cs_client = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `medicion` (
  `ID_PUNTO` int(11) NOT NULL,
  `VALOR` double NOT NULL,
  `ID_AP` char(2) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
/*!40101 SET character_set_client = @saved_cs_client */;

```

```

--
-- Dumping data for table `medicion`
--

LOCK TABLES `medicion` WRITE;
/*!40000 ALTER TABLE `medicion` DISABLE KEYS */;
INSERT INTO `medicion` VALUES (1,-59,'a'), (1,-56,'b'), (1,-90,'c'), (1,-90,'d'), (1,-90,'e'), (2,-68,'a'), (2,-59,'b'), (2,-90,'c'), (2,-90,'d'), (2,-90,'e'), (3,-46,'a'), (3,-40,'b'), (3,-90,'c'), (3,-84,'d'), (3,-90,'e'), (4,-54,'a'), (4,-62,'b'), (4,-90,'c'), (4,-84,'d'), (4,-90,'e'), (5,-57,'a'), (5,-44,'b'), (5,-90,'c'), (5,-90,'d'), (5,-90,'e'), (6,-85,'a'), (6,-76,'b'), (6,-72,'c'), (6,-78,'d'), (6,-90,'e'), (7,-80,'a'), (7,-90,'b'), (7,-59,'c'), (7,-90,'d'), (7,-90,'e'), (8,-78,'a'), (8,-90,'b'), (8,-66,'c'), (8,-86,'d'), (8,-90,'e'), (9,-81,'a'), (9,-90,'b'), (9,-62,'c'), (9,-90,'d'), (9,-90,'e'), (10,-86,'a'), (10,-90,'b'), (10,-73,'c'), (10,-86,'d'), (10,-90,'e'), (11,-90,'a'), (11,-90,'b'), (11,-55,'c'), (11,-93,'d'), (11,-93,'e'), (12,-90,'a'), (12,-90,'b'), (12,-51,'c'), (12,-89,'d'), (12,-90,'e'), (13,-90,'a'), (13,-90,'b'), (13,-52,'c'), (13,-87,'d'), (13,-90,'e'), (14,-90,'a'), (14,-90,'b'), (14,-46,'c'), (14,-85,'d'), (14,-90,'e'), (15,-90,'a'), (15,-90,'b'), (15,-52,'c'), (15,-90,'d'), (15,-90,'e'), (16,-90,'a'), (16,-82,'b'), (16,-75,'c'), (16,-90,'d'), (16,-90,'e'), (17,-90,'a'), (17,-90,'b'), (17,-66,'c'), (17,-85,'d'), (17,-90,'e'), (18,-90,'a'), (18,-90,'b'), (18,-80,'c'), (18,-85,'d'), (18,-90,'e'), (19,-90,'a'), (19,-90,'b'), (19,-68,'c'), (19,-83,'d'), (19,-90,'e'), (20,-90,'a'), (20,-90,'b'), (20,-71,'c'), (20,-83,'d'), (20,-90,'e'), (21,-90,'a'), (21,-90,'b'), (21,-73,'c'), (21,-70,'d'), (21,-90,'e'), (22,-77,'a'), (22,-85,'b'), (22,-74,'c'), (22,-82,'d'), (22,-90,'e'), (23,-72,'a'), (23,-90,'b'), (23,-80,'c'), (23,-78,'d'), (23,-90,'e'), (24,-90,'a'), (24,-76,'b'), (24,-90,'c'), (24,-69,'d'), (24,-90,'e');
/*!40000 ALTER TABLE `medicion` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Table structure for table `puntos_acceso`
--

DROP TABLE IF EXISTS `puntos_acceso`;
/*!40101 SET @saved_cs_client = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `puntos_acceso` (
  `id_ap` char(2) NOT NULL,
  `mac` char(20) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `puntos_acceso`
--

LOCK TABLES `puntos_acceso` WRITE;
/*!40000 ALTER TABLE `puntos_acceso` DISABLE KEYS */;
INSERT INTO `puntos_acceso` VALUES ('a','48:c0:93:1b:6b:50'), ('b','48:c0:93:1b:6b:60'), ('c','48:c0:93:1b:4d:60'), ('d','50:60:28:be:3f:a0'), ('e','48:c0:93:1b:16:10');
/*!40000 ALTER TABLE `puntos_acceso` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Table structure for table `puntos_medicion`
--

DROP TABLE IF EXISTS `puntos_medicion`;
/*!40101 SET @saved_cs_client = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `puntos_medicion` (
  `ID_PUNTO` int(11) NOT NULL,
  `ID_SALON` char(5) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `puntos_medicion`
--

```

```

LOCK TABLES `puntos_medicion` WRITE;
/*!40000 ALTER TABLE `puntos_medicion` DISABLE KEYS */;
INSERT INTO `puntos_medicion` VALUES
(1,'17'),(2,'17'),(3,'17'),(4,'17'),(5,'17'),(6,'16'),(7,'16'),(8,'16'),(9,'16'),(10,'16'),(11,'15'),(12,'15'),(13,'15'),(14,'15'),(15,'15'),(16,'14'),(17,'14'),(18,'14'),(19,'14'),(20,'14'),(21,'10'),(22,'10'),(23,'10'),(24,'10');
/*!40000 ALTER TABLE `puntos_medicion` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Table structure for table `salon`
--

DROP TABLE IF EXISTS `salon`;
/*!40101 SET @saved_cs_client      = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `salon` (
  `id_salon` char(5) NOT NULL,
  `nombre` char(18) DEFAULT NULL,
  `planta` int(11) DEFAULT NULL,
  `id_facultad` char(5) NOT NULL,
  `imagen` varchar(150) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `salon`
--

LOCK TABLES `salon` WRITE;
/*!40000 ALTER TABLE `salon` DISABLE KEYS */;
INSERT INTO `salon` VALUES ('14','Laboratorio 4',4,'f1','D:\\Trabajos
Universidad\\Tesis\\Infraestructura\\WiFi\\ServiciosWeb\\web\\lab4.jpg'), ('15','Laborato
rio 5',4,'f1','D:\\Trabajos
Universidad\\Tesis\\Infraestructura\\WiFi\\ServiciosWeb\\web\\lab5.jpg'), ('16','Laborato
rio 6',4,'f1','D:\\Trabajos
Universidad\\Tesis\\Infraestructura\\WiFi\\ServiciosWeb\\web\\lab6.jpg'), ('17','Laborato
rio 7',4,'f1','D:\\Trabajos
Universidad\\Tesis\\Infraestructura\\WiFi\\ServiciosWeb\\web\\lab7.jpg'), ('10','Pasillo'
,4,'f1','D:\\Trabajos
Universidad\\Tesis\\Infraestructura\\WiFi\\ServiciosWeb\\web\\lab0.jpg');
/*!40000 ALTER TABLE `salon` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Table structure for table `usuario`
--

DROP TABLE IF EXISTS `usuario`;
/*!40101 SET @saved_cs_client      = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `usuario` (
  `ID_USUARIO` char(5) NOT NULL,
  `NOMBRE` char(10) NOT NULL,
  `APELLIDO` char(10) NOT NULL,
  `PASSWORD` char(18) NOT NULL,
  `TIPO_USUARIO` char(15) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `usuario`
--

LOCK TABLES `usuario` WRITE;
/*!40000 ALTER TABLE `usuario` DISABLE KEYS */;
INSERT INTO `usuario` VALUES
('67926','Marco','Quezada','root','Estudiante'), ('67927','Juan','Perez','root','Estudian
te'), ('67928','Carlos','Mora','root','Estudiante');
/*!40000 ALTER TABLE `usuario` ENABLE KEYS */;

```



```

UNLOCK TABLES;
/*!40103 SET TIME_ZONE=@OLD_TIME_ZONE */;

/*!40101 SET SQL_MODE=@OLD_SQL_MODE */;
/*!40014 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS */;
/*!40014 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS */;
/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
/*!40111 SET SQL_NOTES=@OLD_SQL_NOTES */;

```

## Anexo VIII: Código fuente del sistema de localización en interiores con tecnología bluetooth

### Servidor de Aplicaciones

```

import javax.jws.WebService;
import javax.jws.WebMethod;
import javax.jws.WebParam;
import java.util.ArrayList;

@WebService(serviceName = "ServiciosLocalizacion")
public class ServiciosLocalizacion {

    /**
     * Servicio Web encargado de devolver la cadena resultante al Smartphone y de llamar al
     * método localizar
     * de la clase LocalizacionBeacons
     */
    @WebMethod(operationName = "Localizar")
    public String[] Localizar(@WebParam(name = "beacons") String beacons) {
        LocalizacionBeacons registro = new LocalizacionBeacons(beacons.split(":"));
        return registro.Localizar();
    }

    /**
     * Servicio Web que devuelve la información contextual tomando como parametro el lugar
     Y
     un indicador que especifica si es información sobre el lugar o de las clases
     */

    @WebMethod(operationName = "ObtenerClases")
    public ArrayList ObtenerClases(@WebParam(name = "Datos") String Datos,
    @WebParam(name = "indicador") String indicador) {
        return new Horarios(Datos).getContexto(indicador);
    }
}

import javax.swing.JOptionPane;
import java.io.File;
import java.io.FileInputStream;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.nio.charset.StandardCharsets;
import java.util.Base64;

/**
 *
 * @author Marco
 */
public class LocalizacionBeacons {

    String beacons[];
    BaseDeDatos base = new BaseDeDatos();

    /**
     * Constructor de la clase
     */

```

```

public LocalizacionBeacons() {

}

/**
Constructor de la clase que inicializa los datos de los beancos y sus respectivos
valores de RSSI
*/

public LocalizacionBeacons(String[] aux) {
    beacons = aux;
}

/**
Método que obtiene la distancia media entre la obtenida de beacon a Smartphone y
la real entre los beacons
*/

private double DistanciaMedia(String min, String cadena) {
    double dist = 0;
    dist = Double.parseDouble(cadena);
    if (cadena.indexOf(min) >= 0) {
        if (dist < 1) {
            dist = 1;
        }
    } else {
        dist = (dist + base.ObtenerDistancia(min, cadena.split(";")[0]) / 2);
    }
    return dist;
}

/**
Método que comprime la imagen en Base64 para ser enviada al Smartphone como cadena
de texto
*/

public String establecerImg(String lugar) {
    Base64.Encoder encoder = Base64.getEncoder();
    String direccion = base.ObtenerImagen(lugar);
    File file = new File(direccion.trim());
    byte[] fileArray = new byte[(int) file.length()];
    InputStream inputStream;
    String encodedFile = "";
    try {
        inputStream = new FileInputStream(file);
        inputStream.read(fileArray);
        encodedFile = encoder.encodeToString(fileArray);
    } catch (Exception e) {
        System.err.println(e.toString());
    }
    return encodedFile;
}

// public String Minimo(String nom) {
//     String letra = base.ObtenerLiteral(nom);
//     return letra;
// }

/**
Metodo que encuentra el valor del nivel RSSI de un beacon en especefico
de toda la cadena general
*/

public String distArregloRecibido(String beacon) {
    String res = null;
    int cont = 0;
    boolean flag = false;
    while (!flag) {
        if (beacons[cont].split(";")[0].equals(beacon)) {
            res = beacons[cont].split(";")[1];
        }
    }
}

```

```

        flag = true;
    } else {
        cont++;
    }
}
return res;
}

/*
Metodo encargado de establecer la cadena que sera entregada al Smartphone,
llama al metodo que implementa la trilateracion y
llama al metodo que comprime la imagen
*/

public String[] Localizar() {
    String respuesta[] = respuesta = new String[5];
    String linea[] = new String[3];
    Trilateracion tri = null;
    String A = "", B = "", C = "", min = "";

    String lugar = base.ObtenerCodLugar(beacons[0].split(";")[0]);
    String nomlugar = base.ObtenerLugar(lugar);
    if (nomlugar.indexOf("L") == 0) {
        respuesta[0] = lugar;
        respuesta[1] = nomlugar;
        respuesta[2] = "null";
        respuesta[3] = "null";
        respuesta[4] = establecerImg(lugar);
    } else if (nomlugar.indexOf("P") == 0) {
        String pasillo_beacons[] = base.ObtenerBeaconsPasillo(lugar);
        A = pasillo_beacons[0] + ";" + base.ObtenerCoordendadas(pasillo_beacons[0])
+ ";" + DistanciaMedia(beacons[0].split(";")[1],
distArregloRecibido(pasillo_beacons[0]));
        B = pasillo_beacons[1] + ";" + base.ObtenerCoordendadas(pasillo_beacons[1])
+ ";" + DistanciaMedia(beacons[0].split(";")[1],
distArregloRecibido(pasillo_beacons[1]));
        C = pasillo_beacons[2] + ";" + base.ObtenerCoordendadas(pasillo_beacons[2])
+ ";" + DistanciaMedia(beacons[0].split(";")[1],
distArregloRecibido(pasillo_beacons[2]));
        linea[0] = A;
        linea[1] = B;
        linea[2] = C;
        tri = new Trilateracion(linea);
        tri.Localizar();
        respuesta[0] = lugar;
        respuesta[1] = nomlugar+"\n"+base.ObtenerDescrpPasillo(lugar);
        respuesta[2] = tri.coord_x+"";
        respuesta[3] = tri.coord_y+"";
        respuesta[4] = establecerImg(lugar);
    }
    return respuesta;
}
}
import java.text.DecimalFormat;
import java.text.DecimalFormatSymbols;
import java.util.Locale;

public class Trilateracion {

    public String RSSI[];
    public double distA, distB, distC;
    public Punto A, B, C;
    public String coord_x = null, coord_y = null;

    /*
    Constructor de la clase
    */

    public Trilateracion() {

    }
}

```

```

/*
Constructor de la clase que toma los valores RSSI y crea tres objetos de la clase
punto,
para realizar la trilateracion
*/

public Trilateracion(String coordenadas[]) {
    RSSI = coordenadas;
    for (int i = 0; i < coordenadas.length; i++) {
        String beacons[] = coordenadas[i].split(";");
        switch (i) {
            case 0:
                A = new Punto(beacons[0], Double.parseDouble(beacons[1]),
Double.parseDouble(beacons[2]));
                distA = Double.parseDouble(beacons[3]);
                break;
            case 1:
                B = new Punto(beacons[0], Double.parseDouble(beacons[1]),
Double.parseDouble(beacons[2]));
                distB = Double.parseDouble(beacons[3]);
                break;
            case 2:
                C = new Punto(beacons[0], Double.parseDouble(beacons[1]),
Double.parseDouble(beacons[2]));
                distC = Double.parseDouble(beacons[3]);
                break;
            default:
                break;
        }
    }
}

/*
Método que aplica el algoritmo de trilateracion
*/

public void Localizar() {
    DecimalFormatSymbols simbolos =
DecimalFormatSymbols.getInstance(Locale.ENGLISH);
    DecimalFormat decimas = new DecimalFormat("0.00", simbolos);
    double Va = ((Math.pow(distB, 2) - Math.pow(distC, 2)));
    double Vb = ((Math.pow(distB, 2) - Math.pow(distA, 2)));

    double x = 0, y = 0;
    y = (Vb * (B.cordX - C.cordX) - Va * (B.cordX - A.cordX)) / ((A.cordY - B.cordY)
* (B.cordX - C.cordX) - (C.cordY - B.cordY) * (B.cordX - C.cordX));
    x = (y * (A.cordY - B.cordY) - Vb) / (B.cordX - C.cordX);

    if (x < 0) {
        x = x * -1;
    }
    if (y < 0) {
        y = y * -1;
    }
    if (x > 14) {
        x = 13;
    }
    if (y > 12) {
        y = 12;
    }
    coord_x = decimas.format(x);
    coord_y = decimas.format(y);
}

/*
Clase interna para almacenar el nombre de cada punto, y sus coordenadas
*/

private class Punto {

```

```

    public String nombre;
    public double cordX, cordY, cordX2, cordY2;

    public Punto(String beacon, double x, double y) {
        nombre = beacon;
        cordY = y;
        cordX = x;
        cordY2 = Math.pow(y, 2);
        cordX2 = Math.pow(x, 2);
    }

    public Punto() {

    }
}

import java.util.ArrayList;
public class Horarios {

    String info;
    BaseDeDatos base = new BaseDeDatos();

    /*
    Constructor de la clase
    */

    public Horarios() {
    }

    /*
    Constructor de la clase con cadena de texto como paramentro
    */

    public Horarios(String info) {
        this.info = info;
    }

    /*
    Método que llama a obtener la informacion de las clase o del lugar dependiendo del
    indicador
    */

    public ArrayList getContexto(String indicador) {

        ArrayList res = new ArrayList();
        String campos[] = info.split(";");
        if (base.VerificarEsPasillo(campos[0])) {
            res = base.ObtenerLugaresPasillo(campos[0]);
        } else {
            if (indicador.equals("descripcion")) {
                res = base.ObtenerDescripcionLugar(campos[0]);
            } else {
                res = base.ObtenerHorariosSalon(campos[1], Integer.parseInt(campos[2]),
campos[0]);
            }
        }
        return res;
    }
}

import java.awt.Cursor;
import java.sql.Statement;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.logging.Level;
import java.util.logging.Logger;

```

```

public class BaseDeDatos {

    String conexionBD = "jdbc:mysql://localhost:3306/indoorbeacons";

    Connection conexion = null;
    ResultSet rs;
    Statement s;

    /*
    Constructor de la clase que crea la conexion con la base de datos
    */

    public BaseDeDatos() {
        try {
            Class.forName("com.mysql.jdbc.Driver");
            conexion = DriverManager.getConnection(conexionBD, "root", "root");
            s = conexion.createStatement();
        } catch (Exception e) {
            System.err.println(e.toString());
        }
    }

    /*
    Método que devuelve los nombres de todos los beacons
    */

    public String[] ObtenerNombreBeacon() {
        int cont = 0;
        String res[] = null;
        String sql = "select nombre from beacon";
        try {
            rs = s.executeQuery(sql);
            while (rs.next()) {
                res[cont] = rs.getString(1);
                cont++;
            }
        } catch (SQLException ex) {
            Logger.getLogger(BaseDeDatos.class.getName()).log(Level.SEVERE, null, ex);
        }
        return res;
    }

    /*
    Método que obtiene el nombre de un lugar en especifico en base a su id
    */

    public String ObtenerLugar(String cod) {
        String res = null;
        String sql = "select nombre from lugar where idlugar=" + cod;
        try {
            rs = s.executeQuery(sql);
            while (rs.next()) {
                res = rs.getString(1);
            }
        } catch (SQLException ex) {
            Logger.getLogger(BaseDeDatos.class.getName()).log(Level.SEVERE, null, ex);
        }
        return res;
    }

    /*
    Método que obtiene el codigo de un beacon en base a su nombre
    */

    public String ObtenerCodLugar(String beacon) {
        String res = "";
        String sql = "select lugar_idlugar from beacon "
            + "where nombre = '"
            + beacon
            + "'";
    }
}

```

```

    try {
        rs = s.executeQuery(sql);
        while (rs.next()) {
            res = rs.getString(1);
        }
    } catch (SQLException ex) {
        Logger.getLogger(BaseDeDatos.class.getName()).log(Level.SEVERE, null, ex);
    }
    return res;
}

/**
Método que obtiene los beacons asociados a un pasillo en específico
*/

public String[] ObtenerBeaconsPasillo(String lugar) {
    String res[] = new String[3];
    int cont = 0;
    String sql = "select nombre from beacon "
        + "where lugar_idlugar="
        + lugar
        + " order by letra asc";
    try {
        rs = s.executeQuery(sql);
        while (rs.next()) {
            res[cont] = rs.getString(1);
            cont++;
        }
    } catch (SQLException ex) {
        Logger.getLogger(BaseDeDatos.class.getName()).log(Level.SEVERE, null, ex);
    }
    return res;
}

/**
Método que obtiene el literal que identifica a un beacon en específico
*/

public String ObtenerLiteral(String beacon) {
    String res = null;
    String sql = "select letra from beacon "
        + "where nombre = '"
        + beacon
        + "'";
    try {
        rs = s.executeQuery(sql);
        while (rs.next()) {
            res = rs.getString(1);
        }
    } catch (SQLException ex) {
        Logger.getLogger(BaseDeDatos.class.getName()).log(Level.SEVERE, null, ex);
    }
    return res;
}

/**
Método que obtiene las coordenadas reales de un beacon
*/

public String ObtenerCoordenadas(String beacon) {
    String res = null;
    String sql = "select coor_x, coor_y from beacon "
        + "where nombre = '"
        + beacon
        + "'";
    try {
        rs = s.executeQuery(sql);
        while (rs.next()) {
            res = rs.getString(1) + ";" + rs.getString(2);
        }
    } catch (SQLException ex) {

```

```

        Logger.getLogger(BaseDeDatos.class.getName()).log(Level.SEVERE, null, ex);
    }
    return res;
}

/**
Método que obtiene la ruta de donde esta almacenada una imagen dentro del servidor
en base al lugar al que pertenece
*/

public String ObtenerImgLab(String lugar) {
    String res = null;
    String sql = "select imagen from ubicaciones "
        + "where posicion = '"
        + lugar
        + "'";
    try {
        rs = s.executeQuery(sql);
        while (rs.next()) {
            res = rs.getString(1);
        }
    } catch (SQLException ex) {
        Logger.getLogger(BaseDeDatos.class.getName()).log(Level.SEVERE, null, ex);
    }
    return res;
}

/**
Método que obtiene la ruta de donde esta almacenada una imagen dentro del servidor
en base al lugar al que pertenece
*/

public String ObtenerImagen(String lugar) {
    String res = null;
    String sql = "select Imagen \n"
        + "from lugar \n"
        + "where idlugar=" + lugar;
    try {
        rs = s.executeQuery(sql);
        while (rs.next()) {
            res = rs.getString(1).trim();
        }
    } catch (SQLException ex) {
        Logger.getLogger(BaseDeDatos.class.getName()).log(Level.SEVERE, null, ex);
    }
    return res;
}

/**
Método que obtiene la descripcion de un pasillo
*/

public String ObtenerDescrpPasillo(String pasillo) {
    String res = null;
    String sql = "select t.descripcion\n"
        + "from tipo_lugar t, lugar l\n"
        + "where l.idlugar="+pasillo+"\n"
        + "and l.`Tipo_Lugar_idTipo_Lugar`= t.`idTipo_Lugar`";
    try {
        rs = s.executeQuery(sql);
        while (rs.next()) {
            res = rs.getString(1).trim();
        }
    } catch (SQLException ex) {
        Logger.getLogger(BaseDeDatos.class.getName()).log(Level.SEVERE, null, ex);
    }
    return res;
}

/**
Método que verifica si el codigo de un lugar es un pasillo

```



```

*/

public boolean VerificarEsPasillo(String cod) {
    boolean res = false;
    String val = null;
    String sql = "select Nombre\n"
        + "from lugar\n"
        + "where idlugar=" + cod;
    try {
        rs = s.executeQuery(sql);
        while (rs.next()) {
            val = rs.getString(1).trim();
        }
        if (val.equals("PASILLO")) {
            res = true;
        } else {
            res = false;
        }
    } catch (SQLException ex) {
        Logger.getLogger(BaseDeDatos.class.getName()).log(Level.SEVERE, null, ex);
    }
    return res;
}

/*
Método que obtiene los lugares cernado de un pasillo
*/

public ArrayList ObtenerLugaresPasillo(String pasillo) {
    ArrayList res = new ArrayList();
    String sql = "select l.`Nombre`,t.descripcion\n"
        + "from lugar l, tipo_lugar t, lugaresceranos c\n"
        + "where c.lugar_principal = " + pasillo + "\n"
        + "and l.idlugar = c.lugar_cercano\n"
        + "and l.`Tipo_Lugar_idTipo_Lugar` = t.`idTipo_Lugar`";
    try {
        rs = s.executeQuery(sql);
        while (rs.next()) {
            res.add(rs.getString(1).trim() + ";" + rs.getString(2).trim());
        }
    } catch (SQLException ex) {
        Logger.getLogger(BaseDeDatos.class.getName()).log(Level.SEVERE, null, ex);
    }
    return res;
}

/*
Método que obtiene la informacion de la clase actual que se esta llevando cabo en un
laboratorio
*/

public ArrayList ObtenerHorariosSalon(String hora, int dia, String salon) {
    ArrayList res = new ArrayList();
    String sql = "select l.Nombre,a.nombre,c.nombre,
p.nombre,p.apellido,h.horaInicio, h.horaFin\n"
        + "from horario h, asignatura a, lugar l, asignatura_curso ac,persona
p,curso c\n"
        + "where (h.horaInicio)<=" + hora + "'\n"
        + "and (h.horaFin)>' + hora + "' \n"
        + "and h.lugar_idlugar=" + salon + "\n"
        + "and h.dia=" + dia + "\n"
        + "and l.idlugar=h.lugar_idlugar\n"
        + "and a.idAsignatura=h.idAsignatura\n"
        + "and ac.idAsignatura =h.idAsignatura\n"
        + "and p.idPersona =ac.idProfesor\n"
        + "and c.idCurso=ac.idCurso";
    try {
        rs = s.executeQuery(sql);
        while (rs.next()) {

```

```

        res.add(rs.getString(1).trim() + ";" + rs.getString(2).trim() + ";" +
rs.getString(3).trim() + ";" + rs.getString(4).trim() + " " + rs.getString(5).trim() +
";" + rs.getString(6).trim() + ";" + rs.getString(7).trim());
    }
    } catch (SQLException ex) {
        Logger.getLogger(BaseDeDatos.class.getName()).log(Level.SEVERE, null, ex);
    }
    return res;
}

/*
Método que obtiene la descripción de un lugar
*/

public ArrayList ObtenerDescripcionLugar(String lugar) {
    ArrayList res = new ArrayList();
    String sql = "SELECT l.`Nombre`,f.`Nombre`,l.`Descripcion`\n"
        + "from lugar l,lugar f\n"
        + "where l.idlugar=" + lugar + "\n"
        + "and l.`idLugarContenedor` = f.idlugar";
    try {
        rs = s.executeQuery(sql);
        while (rs.next()) {
            String cadena[] = rs.getString(3).split(";");
            String des = "";
            for (int i = 0; i < cadena.length; i++) {
                des += cadena[i] + "\n";
            }
            res.add(rs.getString(1).trim() + ";" + rs.getString(2).trim() + ";" +
des);
        }
    } catch (SQLException ex) {
        Logger.getLogger(BaseDeDatos.class.getName()).log(Level.SEVERE, null, ex);
    }
    return res;
}

/*
Método que obtiene la distancia real entre dos beacons
*/

public double ObtenerDistancia(String a, String b) {
    int cont = 0;
    int bA = 0, bB = 0;
    double res = 0;
    try {
        System.out.println(a + " " + b);
        String sql = "select idbeacon"
            + " from beacon"
            + " where nombre = '" + a
            + "' or nombre ='" + b + "'";
        rs = s.executeQuery(sql);
        while (rs.next()) {
            if (cont == 0) {
                bA = Integer.parseInt(rs.getString(1));
                rs.getString(1);
                cont++;
            } else {
                bB = Integer.parseInt(rs.getString(1));
                cont = 0;
            }
        }
        sql = "select valor"
            + " from distancia_beacon"
            + " where beacon_a = "
            + bA
            + " and beacon_b = "
            + bB;
        rs = s.executeQuery(sql);
        while (rs.next()) {
            res = Double.parseDouble(rs.getString(1));
        }
    }
}

```

```

    }
    } catch (SQLException ex) {
        Logger.getLogger(BaseDeDatos.class.getName()).log(Level.SEVERE, null, ex);
    }
    return res;
}
}

```

## Aplicación Móvil

```

import android.content.Context;
import android.content.SharedPreferences;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.os.Bundle;
import android.support.design.widget.FloatingActionButton;
import android.support.design.widget.Snackbar;
import android.support.v7.app.AppCompatActivity;
import android.support.v7.widget.Toolbar;
import android.util.Base64;
import android.util.Log;
import android.view.View;
import android.view.Menu;
import android.view.MenuItem;
import android.bluetooth.BluetoothAdapter;
import android.bluetooth.BluetoothDevice;
import android.content.Intent;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.ListView;
import android.os.Handler;
import android.widget.TextView;
import android.widget.Toast;

import java.io.ByteArrayInputStream;
import java.io.InputStream;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Date;
import java.util.List;
import android.content.pm.ActivityInfo;

public class MainActivity extends AppCompatActivity implements Dialogo_Clase.Clases,
Dialogo_Config.Config, Dialogo_Info.Info, Dialogo_Pasillo.Pasillo{

    FloatingActionButton fab;
    TextView txt_salon,txt_xy,clas_act;
    ImageView imagen;
    BluetoothAdapter miBTAdapter = null;
    Button info;
    Button info_clas;
    BaseHelper baseHelper;
    String dbNombre = "bluetooth",desc="";
    Handler mHandler;
    boolean estado=false,estado1=false,descargado=false;
    String resView = "";
    Cursor c;
    ArrayList<Asignatura> asignaturas = new ArrayList<Asignatura>();
    ArrayList<Lugar> lugares = new ArrayList<Lugar>();
    ArrayList<Lugar_Pasillo> lugar_pasillos = new ArrayList<Lugar_Pasillo>();
    int ciclo = 60000;
    int lab_pasillo=1;
    /*

```

```

Metodo sobrescrito que inicializa la app y sus elementos graficos
*/
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
    setSupportActionBar(toolbar);

    info = (Button) findViewById(R.id.info);
    info_clas = (Button) findViewById(R.id.btn_clase);
    txt_salon= (TextView) findViewById(R.id.txt_salon);
    txt_xy = (TextView) findViewById(R.id.txt_xy);
    imagen = (ImageView) findViewById(R.id.imagen);
    clas_act = (TextView) findViewById(R.id.msg_clase);
    fab = (FloatingActionButton) findViewById(R.id.fab);
    fab.setVisibility(View.INVISIBLE);

    /*
    Inicializacion de la base de datos SQLite
    */
    baseHelper = new BaseHelper(this,dbNombre,null,1);
    /*
    Llamado a la clase BluetoothAdapter
    */
    miBTAdapter = BluetoothAdapter.getDefaultAdapter();
    if (!miBTAdapter.isEnabled()) {
        Intent turnOnIntent = new Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);
        startActivityForResult(turnOnIntent, 0);
    }
    mHandler = new Handler();
    miBTAdapter = BluetoothAdapter.getDefaultAdapter();
    /*
    Metodo sobrescrito que asociado al evento click del boton para presentar el mensaje
    de actualizacion
    */
    fab.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            Snackbar.make(view, R.string.msg_act, Snackbar.LENGTH_LONG)
                .setAction("Action", null).show();
        }
    });
    /*
    Metodo sobrescrito asociado al evento click del boton para visualizar el dialogo de
    la informacion del lugar
    */
    info.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            if(lab_pasillo==1) {
                Lugar registro = (Lugar) lugares.get(0);
                crearDialogoInfo(registro);
            }else
                crearDialogoPasillo();
        }
    });
    /*
    Metodo sobrescrito asociado al evento click del boton que genera el dialogo para
    visualizar la informacion de la clase
    */
    info_clas.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Asignatura registro = (Asignatura) asignaturas.get(0);
            crearDialogo(registro);
        }
    });
    if (miBTAdapter.isEnabled()) {
        SharedPreferences pref = getSharedPreferences("datos",
Context.MODE_PRIVATE);
        ciclo = Integer.parseInt(pref.getString("tiempo", "60000"));

```

```

        Escanear(true);
    }
    else{
        EstadoOriginal(false);
    }
}
/*
Metodo sobrescrito que genera el menu
*/
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the action bar if it is present.
    getMenuInflater().inflate(R.menu.menu_main, menu);
    return true;
}
/*
Metodo sobrescrito asociado al la seleccion de un elemento del menu
*/
@Override
public boolean onOptionsItemSelected(MenuItem item) {
    // Handle action bar item clicks here. The action bar will
    // automatically handle clicks on the Home/Up button, so long
    // as you specify a parent activity in AndroidManifest.xml.
    int id = item.getItemId();

    //noinspection SimplifiableIfStatement
    if (id == R.id.tiempo) {
        crearDialogoConfig();
        return true;
    }else if(id == R.id.menu_actualizar){
        Reiniciar();
    }
    return super.onOptionsItemSelected(item);
}
/*
Metodo encargado de reiniciar la app
*/
public void Reiniciar(){
    BaseDeDatos.LimpiarTablaTemporal(baseHelper);
    Intent i = getBaseContext().getPackageManager()
        .getLaunchIntentForPackage( getBaseContext().getPackageName() );
    i.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
    startActivity(i);
}
/*
Metodo que aplica el hilo encargado de actualizar la localizacion segun el tiempo
programado
*/
public void Hilo(boolean enable){
    if (enable && !estadol) {
        mHandler.postDelayed(new Runnable() {
            @Override
            public void run() {
                estadol = false;
                Escanear(true);
                //Log.i("Ref2:",ArrayBLE.get(ArrayBLE.size()-1));
            }
        }, ciclo);
        estadol = true;
        //boton.setEnabled(false);
        //progressDialog.setProgress(cont);
    } else {
        estadol = false;
        // boton.setEnabled(true);
    }
}
/*
Metodo encargado de representar los resultados
*/
private void Localizar(){
    String coor[] = EstablecerArreglo();
}

```

```

String res,beacons = "";
for(int i=0;i<coor.length;i++) {
    beacons+= coor[i] + ":";
}
Log.i("Beacons: ",beacons.substring(0,beacons.length()-1));
WS_Localizacion servicio = (WS_Localizacion) new
WS_Localizacion().execute(beacons);
try {
    res = servicio.get();
    Log.i("Res: ",res);
    res=res.substring(1,res.length());
    res=res.substring(0,res.length()-1);
    String val[]=res.split(",");
    if(val[2].trim().equals("null")) {
        Visualizar(val[0].trim(), val[1].trim(), "null", val[4].trim());
    }
    else {
        Visualizar(val[0].trim(),val[1].trim(),"X: " + val[2].trim() + "-Y: " +
val[3].trim(), val[4].trim());
    }
    // Toast.makeText(this,res,Toast.LENGTH_LONG).show();
} catch (Exception e) {
    BaseDeDatos.LimpiarTablaTemporal(baseHelper);
    Toast.makeText(this,e.toString(),Toast.LENGTH_LONG).show();
    EstadoOriginal(true);
    e.printStackTrace();
}
}

/*
Metodo encargado del llamar al servicio web que obtiene la informacion de las clases
*/
private String Clases(String dato, String indicador){
    String res = null;
    WS_Clases servicio = (WS_Clases) new WS_Clases().execute(dato,indicador);
    try {
        res = servicio.get();
        Log.i("Clases: ",res);
        // Toast.makeText(this,res,Toast.LENGTH_LONG).show();
    } catch (Exception e) {
        BaseDeDatos.LimpiarTablaTemporal(baseHelper);
        Toast.makeText(this,e.toString(),Toast.LENGTH_LONG).show();
        EstadoOriginal(true);
        e.printStackTrace();
    }
    if(res.indexOf("[")<0){
        res="["+res+"]";
    }
    return res;
}

/*
Metodo que obtiene el dia y la hora actuales
*/
private String getHoraDia(){

    Date date=new Date();
    SimpleDateFormat dia = new SimpleDateFormat("EEEE");
    String dayOfTheWeek = dia.format(date);
    String horaActual;
    if(date.getHours()<10){
        horaActual="0"+date.getHours();
    }
    else{
        horaActual = date.getHours()+"";
    }

    if(date.getMinutes()<10){
        horaActual+=":0"+date.getMinutes();
    }else{
        horaActual+=":"+date.getMinutes();
    }
}

```

```

    }

    if(date.getSeconds()<10){
        horaActual+=":0"+date.getSeconds();
    }else{
        horaActual+=":"+date.getSeconds();
    }

    int diaSemana;
    switch (dayOfTheWeek){
        case "lunes":
            diaSemana=1;
            break;
        case "martes":
            diaSemana=2;
            break;
        case "miercoles":
            diaSemana=3;
            break;
        case "jueves":
            diaSemana=4;
            break;
        case "viernes":
            diaSemana=5;
            break;
        case "domingo":
            diaSemana=7;
            break;
        default:
            diaSemana=6;
            break;
    }
    Log.i("dia/hota ",horaActual+" "+diaSemana);
    return horaActual+" "+diaSemana;
}

/*
Metodo que carga la informacion contextual
*/
private void llenarLista(String cadena, int index){

    cadena = cadena.substring(1,cadena.length()-1);
    String clases[]=cadena.split(",");
    Log.i("index",index+"");
    switch (index){
        case 1:{
            for(int i=0;i<clases.length;i++){
                String dato[]=clases[i].split(";");
                Lugar_Pasillo registro = new Lugar_Pasillo(dato[0],dato[1]);
                lugar_pasillos.add(registro);
                info_clas.setVisibility(View.INVISIBLE);
                info.setVisibility(View.VISIBLE);
                info.setText(R.string.btn_pasillo);
                lab_pasillo=2;
                clas_act.setText(txt_salon.getText().toString().substring(10)+desc);
            }
            break;
        }
        case 2:{
            if(cadena.indexOf("Desocupado")>=0){
                clas_act.setText(txt_salon.getText().toString().substring(10)+"\n
"+cadena);

                info_clas.setVisibility(View.INVISIBLE);
            }else {
                for (int i = 0; i < clases.length; i++) {
                    String dato[] = clases[i].split(";");
                    Asignatura registro = new Asignatura(dato[0], dato[1], dato[2],
dato[3], dato[4],dato[5]);
                    asignaturas.add(registro);
                    clas_act.setText(" "+registro.salon+": \n"+"
"+registro.materia);
                }
            }
        }
    }
}

```

```

        lab_pasillo=1;
        info_clas.setVisibility(View.VISIBLE);
    }
    }
    break;
}
case 3:{
    if(cadena.indexOf("Sin InformaciÃ³n")>=0){
        info.setVisibility(View.INVISIBLE);
    }else {
        for (int i = 0; i < clases.length; i++) {
            String dato[] = clases[i].split(";");
            Lugar registro = new Lugar(dato[0], dato[1], dato[2]);
            lugars.add(registro);
            info.setVisibility(View.VISIBLE);
            lab_pasillo=1;
            info.setText(R.string.btn_lab);
            //clas_act.setText(" "+registro.salon+": \n"+"
"+registro.materia);
        }
    }
    break;
}
}

}

/*
Metodo que limpia la app
*/
public void EstadoOriginal(boolean err){
    txt_salon.setText("");
    txt_xy.setText("");
    if(err){
        imagen.setImageResource(R.mipmap.no_licalizado);
    }else{
        imagen.setImageResource(R.color.colorPrimaryDark);
    }
    info.setVisibility(View.INVISIBLE);
    info_clas.setVisibility(View.INVISIBLE);
    clas_act.setText("");
    asignaturas.clear();
    lugars.clear();
    lugar_pasillos.clear();
}

/*
Metodo que visualiza los resultados en los textview y la imagen
*/
private void Visualizar(String cod,String nom_lugar,String coor,String img){

    String horario = null;
    String clases=null,info_lab=null;
    Log.i("lugar",nom_lugar);
    txt_salon.setText(R.string.ubicacion);
    txt_salon.setText(txt_salon.getText().toString()+" "+nom_lugar.split("\n") [0]);
    horario=cod+"";
    Toast.makeText(this,nom_lugar.split("\n") [0],Toast.LENGTH_LONG).show();
    Bitmap myBitmap = ConvertToImage(img);
    imagen.setImageBitmap(myBitmap);
    horario+=getHoraDia();
    if(coor.equals("null")) {
        txt_xy.setText("");
    }else
        txt_xy.setText(coor);
    if(nom_lugar.indexOf("L")==0) {
        clases = Clases(horario,"clases");
        llenarLista(clases,2);
        info_lab = Clases(horario,"descripcion");
        llenarLista(info_lab,3);
    }else if(nom_lugar.indexOf("P")==0) {

```



```

        desc="\n"+nom_lugar.split("\n")[1];
        info_lab = Clases(horario,"descripcion");
        llenarLista(info_lab,1);
    }
}

/*
Metodo que implementa un hilo encargado de escanear los niveles RSSI de los beacons
durante 5 segundos
*/
private void Escanear(boolean enable){
    int cont=0;
    EstadoOriginal(false);
    if(miBTAdapter.isEnabled()) {
        fab.callOnClick();
        try {
            if (enable && !estado) {
                mHandler.postDelayed(new Runnable() {
                    @Override
                    public void run() {
                        estado = false;
                        miBTAdapter.stopLeScan(scan);
                        Localizar();
                        // boton.setEnabled(true);
                        resView = "";
                        BaseDeDatos.LimpiarTablaTemporal(baseHelper);
                        Hilo(true);
                    }
                }, 3000);
                estado = true;
                miBTAdapter.startLeScan(scan);

                //boton.setEnabled(false);
                ///progressDialog.setProgress(cont);
                cont++;
            } else {
                estado = false;
                miBTAdapter.stopLeScan(scan);
                // boton.setEnabled(true);
            }
        } catch (Exception e) {
            e.printStackTrace();
            BaseDeDatos.LimpiarTablaTemporal(baseHelper);
        }
    }else
        Toast.makeText(this,"Bluetooth Desactivado",Toast.LENGTH_LONG).show();
}

/*
Metodo que obtiene la distancia entre beacon y smartphone
*/
private double Distancia(double rssi){
    double res=0;
    double c1=10.09188024,c2=6.629122067,c3=(-0.016356041),txPower=-41;
    res = (c1)*Math.pow((rssi/txPower),c2) + c3;
    return res;
}

/*
Metodo sobrescrito que escanea los beacons
*/
private BluetoothAdapter.LeScanCallback scan = new BluetoothAdapter.LeScanCallback()
{
    @Override
    public void onLeScan(BluetoothDevice device, int rssi, byte[] scanRecord) {
        if (device.getName() != null) {
            if (resView.equals("") || resView.indexOf(device.getName()) < 0) {
                resView = resView + "\n" + device.getName() + ";" + Distancia(rssi);
                boolean flag = BaseDeDatos.IngresarTemporal(baseHelper,
                device.getName(), Distancia(rssi));
                if (!flag) {

```

```

        Toast.makeText(getApplicationContext(), "Error de DB",
Toast.LENGTH_LONG).show();
    }
}
};

/*
Metodo que arma la cadena con los nombres de los beacons y sus valores RSSI
*/
private String[] EstablecerArreglo(){
    String res[] = null;
    int cont=0;
    try {
        SQLiteDatabase db = baseHelper.getReadableDatabase();
        if (db != null) {
            //c=BaseDeDatos.ConsultaEspecifica(baseHelper,"b00500005");
            c = BaseDeDatos.ConsultaCompletaDistancia(baseHelper);
            if (c.moveToFirst()) {
                res = new String[c.getCount()];
                res[cont] = c.getString(0) + ";" + c.getString(1);
                cont++;
                while (c.moveToNext()) {
                    res[cont] = c.getString(0) + ";" + c.getString(1);
                    cont++;
                }
            } else
                Toast.makeText(this, "Error de BD", Toast.LENGTH_LONG).show();
        } else
            Toast.makeText(this, "Error de BD", Toast.LENGTH_LONG).show();
    }
    catch (Exception e){
        Toast.makeText(this, "Error de Ejecucion", Toast.LENGTH_LONG).show();
        BaseDeDatos.LimpiarTablaTemporal(baseHelper);
        EstadoOriginal(true);
        e.printStackTrace();
    }
    return res;
}

/*
Metodo sobrescrito para la finalizacion del dialogo de informacion de las clases
*/
@Override
public void FinalizaClases() {

}

/*
Metodo que crea el dialogo para la informacion de las clases
*/
public void crearDialogo(Asignatura as){
    Dialogo_Clase dialogo = new
Dialogo_Clase(as.salon,as.materia,as.curso,as.profesor,as.h_inicio,as.h_fin);
    dialogo.show(getSupportFragmentManager(), "ingreso");

    android.app.Fragment frag = getSupportFragmentManager().findFragmentByTag("ingreso");

    if (frag != null) {
        getSupportFragmentManager().beginTransaction().remove(frag).commit();
    }
}

/*
Metodo que crea el dialogo para la informacion del lugar
*/
public void crearDialogoInfo(Lugar lugar){
    Dialogo_Info dialogo = new
Dialogo_Info(lugar.nombre,lugar.facultad,lugar.descripcion);
    dialogo.show(getSupportFragmentManager(), "informacion");
}

```

```

        android.app.Fragment frag =
getFragmentManager().findFragmentByTag("informacion");

        if (frag != null) {
            getFragmentManager().beginTransaction().remove(frag).commit();
        }
    }

    /**
Metodo que crea el dialogo para la informacion de los lugares cercanos al pasillo
*/
    public void crearDialogoPasillo(){
        Dialogo_Pasillo dialogo = new Dialogo_Pasillo(lugar_pasillos);
        dialogo.show(getSupportFragmentManager(), "pasillo");

        android.app.Fragment frag = getFragmentManager().findFragmentByTag("pasillo");

        if (frag != null) {
            getFragmentManager().beginTransaction().remove(frag).commit();
        }
    }

    /**
Metodo que crea el dialogo de configuracion
*/
    public void crearDialogoConfig(){
        SharedPreferences pref = getSharedPreferences("datos", Context.MODE_PRIVATE);
        Dialogo_Config dialogo = new Dialogo_Config(pref.getString("tiempo", "60000"));
        dialogo.show(getSupportFragmentManager(), "config");
        android.app.Fragment frag = getFragmentManager().findFragmentByTag("config");

        if (frag != null) {
            getFragmentManager().beginTransaction().remove(frag).commit();
        }
    }

    /**
Metodo que descomprime la imagen de Base64
*/
    public Bitmap ConvertToImage(String image){
        try{
            InputStream stream = new
ByteArrayInputStream(Base64.decode(image.getBytes(), Base64.DEFAULT));
            Bitmap bitmap = BitmapFactory.decodeStream(stream);
            Log.v("Ben", "Image Converted");
            return bitmap;
        }
        catch (Exception e) {
            EstadoOriginal(true);
            return null;
        }
    }

    /**
Metodo sobrescrito para la finalizacion del dialogo de informacion de las
configuracion de tiempo
*/
    @Override
    public void FinalizaConfig(int time) {
        ciclo = time;
        if(ciclo!=-1){
            SharedPreferences pref = getSharedPreferences("datos",
Context.MODE_PRIVATE);
            SharedPreferences.Editor edit = pref.edit();
            edit.putString("tiempo", time + "");
            Log.i("config", time + "");
            edit.commit();
        }
    }
}

```

```

    /*
    Metodo sobrescrito para la finalizacion del dialogo de informacion de los lugares
    */
    @Override
    public void FinalizaInfo() {

    }

    /*
    Metodo sobrescrito para la finalizacion del dialogo de informacion de los lugares
    cercanos de un pasillo
    */
    @Override
    public void FinalizaPasillo() {

    }

    /*
    Clase interna que posee los datos de una asignatura
    */
    private class Asignatura{

        public String salon;
        public String curso;
        public String materia;
        public String profesor;
        public String h_inicio;
        public String h_fin;

        public Asignatura(String sa,String mat,String cu,String pro,String hi,String
hf) {
            salon=sa;
            materia=mat;
            profesor=pro;
            h_inicio=hi;
            h_fin=hf;
            curso=cu;
        }
    }

    /*
    Clase interna que posee los datos del lugar
    */
    private class Lugar{

        public String nombre;
        public String facultad;
        public String descripcion;

        public Lugar(String nom,String fac,String desc){
            nombre=nom;
            descripcion=desc;
            facultad=fac;
        }
    }
}

package com.example.marco.indooruda;

import android.support.v4.app.DialogFragment;
import android.content.Context;
import android.net.Uri;
import android.os.Bundle;
import android.support.v4.app.Fragment;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.ListView;
import android.widget.SimpleAdapter;
import android.widget.TextView;

```

```

import android.widget.Toast;

import com.example.marco.indooruda.R;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.Iterator;
import java.util.List;
import java.util.Map;

public class Dialogo_Clase extends DialogFragment {

    Button boton,desc;
    Clases interfaz=null;
    HashMap<String,String> datos = new HashMap<>();
    ListView lista;
    boolean descarga=false;
    String material=null;

    public interface Clases {
        void FinalizaClases();
    }

    public Dialogo_Clase(){

    }

    public Dialogo_Clase(String sa,String asi,String cur,String pro,String hi,String hf)
{

        datos.put("Profesor:",pro+"\n");
        datos.put("Horario:",hi+"/"+"hf+"\n");
        datos.put("Asignatura:",asi+"\n");
        datos.put("Lugar:",sa+"\n");
        datos.put("Curso: ",cur+"\n");
    }

    /*
    Metodo sobrescrito que inicializa el dialogo y sus componentes graficos
    */
    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle
savedInstanceState) {
        final View view = inflater.inflate(R.layout.fragment_dialogo__clase,container);

        lista = (ListView)view.findViewById(R.id.lista);
        boton = (Button)view.findViewById(R.id.boton);

        List<HashMap<String, String>> listItems = new ArrayList<>();
        SimpleAdapter adapter = new SimpleAdapter(view.getContext(), listItems,
R.layout.list_item,
            new String[]{"First Line", "Second Line"},
            new int[]{R.id.text1, R.id.text2});

        Iterator it = datos.entrySet().iterator();
        while (it.hasNext())
        {
            HashMap<String, String> resultsMap = new HashMap<>();
            Map.Entry pair = (Map.Entry)it.next();
            resultsMap.put("First Line", pair.getKey().toString());
            resultsMap.put("Second Line", pair.getValue().toString());
            listItems.add(resultsMap);
        }

        /*
        Metodo sobrescrito asociado al evento click de regresar
        */
        lista.setAdapter(adapter);
        boton.setOnClickListener(new View.OnClickListener() {
            @Override

```

```

        public void onClick(View v) {

            interfaz =(Clases) getActivity();
            interfaz.FinalizaClases();
            dismiss();
        }
    });

    getDialog().setTitle("Clase Actual");
    return view;
}
}

package com.example.marco.indooruda;

import android.os.Bundle;
import android.support.v4.app.DialogFragment;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.RadioButton;
import android.widget.TextView;

public class Dialogo_Config extends DialogFragment {
    RadioButton opt1,opt2,opt3,opt4,opt5,opt6;
    Button boton,cancelar;
    Config interfaz=null;
    String actual;

    public interface Config {
        void FinalizaConfig(int time);
    }

    public Dialogo_Config(String current){
        actual=current;
    }

    /*
    Metodo sobreescrito que inicializa el dialogo y sus componentes graficos
    */
    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle
savedInstanceState) {
        View view = inflater.inflate(R.layout.fragment_dialogo_config,container);

        opt1 = (RadioButton)view.findViewById(R.id.rd_a);
        opt2 = (RadioButton)view.findViewById(R.id.rd_b);
        opt3 = (RadioButton)view.findViewById(R.id.rd_c);
        opt4 = (RadioButton)view.findViewById(R.id.rd_d);
        opt5 = (RadioButton)view.findViewById(R.id.rd_0);
        opt6 = (RadioButton)view.findViewById(R.id.rd_e);
        boton = (Button)view.findViewById(R.id.btn);
        cancelar = (Button)view.findViewById(R.id.btn_can);

        switch (actual){
            case "10000":{
                opt1.setChecked(true);
                break;
            }
            case "30000":{
                opt2.setChecked(true);
                break;
            }
            case "60000":{
                opt3.setChecked(true);
                break;
            }
            case "300000":{
                opt4.setChecked(true);
            }
        }
    }
}

```

```

        break;
    }
    case "5000":{
        opt4.setChecked(true);
        break;
    }
    case "600000":{
        opt4.setChecked(true);
        break;
    }
    default:{
        opt1.setChecked(true);
        break;
    }
}

/*
Metodo sobreescrito asociado al evento click del boton de guardar
*/
    boton.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {

            interfaz =(Config) getActivity();
            interfaz.FinalizaConfig(seleccion());
            dismiss();
        }
    });

/*
Metodo sobreescrito asociado al evento click del boton regresar
*/
    cancelar.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            interfaz =(Config) getActivity();
            interfaz.FinalizaConfig(-1);
            dismiss();
        }
    });

    getDialog().setTitle("Configurar Tiempo");

    return view;
}

/*
Metodo encargado de obtener el tiempo en base a la opcion escogida
*/
public int seleccion(){
    int time = 0;
    if(opt1.isChecked()){
        time=10000;
    }else if(opt2.isChecked()){
        time=30000;
    }else if(opt3.isChecked()){
        time=60000;
    }else if(opt4.isChecked()){
        time=300000;
    }else if(opt5.isChecked()){
        time=5000;
    }else if(opt6.isChecked()){
        time=600000;
    }
    return time;
}
}

package com.example.marco.indooruda;

import android.os.Bundle;

```

```

import android.support.v4.app.DialogFragment;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.ListView;
import android.widget.SimpleAdapter;
import android.widget.TextView;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.Iterator;
import java.util.List;
import java.util.Map;

public class Dialogo_Info extends DialogFragment {
    TextView lab, fac, desc;
    Button boton;
    Info interfaz=null;
    HashMap<String,String> datos = new HashMap<>();
    ListView lista;

    public interface Info {
        void FinalizaInfo();
    }

    public Dialogo_Info(){

    }

    public Dialogo_Info(String sa, String fa, String des) {

        datos.put("Lugar:", sa+"\n");
        datos.put("Facultad:", fa+"\n");
        datos.put("DescripciÃ³n:", des);
    }

    /*
    Metodo sobrescrito que inicializa el dialogo y sus componentes graficos
    */
    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle
savedInstanceState) {
        View view = inflater.inflate(R.layout.fragment_dialogo_info, container);
        lista = (ListView)view.findViewById(R.id.lista);
        boton = (Button)view.findViewById(R.id.boton);

        List<HashMap<String, String>> listItems = new ArrayList<>();
        SimpleAdapter adapter = new SimpleAdapter(view.getContext(), listItems,
R.layout.list_item,
            new String[]{"First Line", "Second Line"},
            new int[]{R.id.text1, R.id.text2});

        Iterator it = datos.entrySet().iterator();
        while (it.hasNext())
        {
            HashMap<String, String> resultsMap = new HashMap<>();
            Map.Entry pair = (Map.Entry)it.next();
            resultsMap.put("First Line", pair.getKey().toString());
            resultsMap.put("Second Line", pair.getValue().toString());
            listItems.add(resultsMap);
        }

        lista.setAdapter(adapter);

    /*
    Metodo sobrescrito asociado al evento click del boton regresar
    */
        boton.setOnClickListener(new View.OnClickListener() {
            @Override

```



```

        public void onClick(View v) {

            interfaz =(Info) getActivity();
            interfaz.FinalizaInfo();
            dismiss();
        }
    });

    getDialog().setTitle("Datos del Lugar");
    return view;
}
}

package com.example.marco.indooruda;

import android.os.Bundle;
import android.support.v4.app.DialogFragment;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.ListView;
import android.widget.RadioButton;
import android.widget.SimpleAdapter;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.Iterator;
import java.util.List;
import java.util.Map;

public class Dialogo_Pasillo extends DialogFragment {

    Button boton;
    Pasillo interfaz=null;
    ListView nombres;
    HashMap<String,String> datos = new HashMap<>();

    public interface Pasillo {
        void FinalizaPasillo();
    }

    public Dialogo_Pasillo(ArrayList<Lugar_Pasillo> lugares){
        for(int i=0;i<lugares.size();i++){
            datos.put(lugares.get(i).lugar,lugares.get(i).tipo+"\n");
        }
    }

    /*
    Metodo sobreescrito que inicializa el dialogo y sus componentes graficos
    */
    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle
savedInstanceState) {
        View view = inflater.inflate(R.layout.fragment_dialogo_pasillo,container);

        nombres = (ListView)view.findViewById(R.id.lista);
        boton = (Button)view.findViewById(R.id.boton);

        List<HashMap<String, String>> listItems = new ArrayList<>();
        SimpleAdapter adapter = new SimpleAdapter(view.getContext(), listItems,
R.layout.list_item,
            new String[]{"First Line", "Second Line"},
            new int[]{R.id.text1, R.id.text2});

        Iterator it = datos.entrySet().iterator();
        while (it.hasNext())
        {
            HashMap<String, String> resultsMap = new HashMap<>();

```

```

        Map.Entry pair = (Map.Entry)it.next();
        resultsMap.put("First Line", pair.getKey().toString());
        resultsMap.put("Second Line", pair.getValue().toString());
        listItems.add(resultsMap);
    }

    nombres.setAdapter(adapter);

    /*
    Metodo sobreescrito asociado al evento click del boton de regreso
    */
    boton.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {

            interfaz =(Pasillo) getActivity();
            interfaz.FinalizaPasillo();
            dismiss();
        }
    });

    getDialog().setTitle("Lugares Cercanos");

    return view;
}
}

package com.example.marco.indooruda;

/**
 * Created by Marco on 15/10/2017.
 */

public class Lugar_Pasillo{

    public String lugar;
    public String tipo;

    /*
    Constructor de la clase
    */
    public Lugar_Pasillo(String lug,String tip){
        lugar=lug;
        tipo=tip;
    }
}

import android.os.AsyncTask;
import org.ksoap2.SoapEnvelope;
import org.ksoap2.serialization.SoapObject;
import org.ksoap2.serialization.SoapSerializationEnvelope;
import org.ksoap2.transport.HttpTransportSE;

public class WS_Clases extends AsyncTask<String, Void, String> {

    final String SOAP_ACTION = "http://servicios/ObtenerClases";
    final String METHOD = "ObtenerClases";
    final String NAMESPACE = "http://servicios/";
    final String ENDPOINTWS =
"http://172.16.1.47:8080/ServidorIndoorBluetooth/ServiciosLocalizacion?xsd=1";
    String respuesta = null;

    /*
    Metodo sobreescrito que inicializa la clase y llama al servicio web
    */
    @Override
    protected String doInBackground(String... args) {
        SoapObject userRequest = new SoapObject(NAMESPACE, METHOD);
        userRequest.addProperty("Datos", args[0]);
    }
}

```

```

        userRequest.addProperty("indicador", args[1]);

        SoapSerializationEnvelope envelope = new
        SoapSerializationEnvelope(SoapEnvelope.VER11);
        envelope.setOutputSoapObject(userRequest);

        try{
            HttpTransportSE androidHttpTransport = new HttpTransportSE(ENDPOINTWS);
            androidHttpTransport.debug = true;
            androidHttpTransport.call(SOAP_ACTION, envelope);

            if(envelope.getResponse()==null) {
                if(args[1].equals("descripcion")){
                    respuesta = "Sin Informaci3n";
                }else {
                    respuesta = "Desocupado";
                }
            }
            else
                respuesta = envelope.getResponse().toString();
        }
        catch (Exception e){
            e.printStackTrace();
        }
        return respuesta;
    }
}

package com.example.marco.indooruda;

import android.os.AsyncTask;

import org.ksoap2.SoapEnvelope;
import org.ksoap2.serialization.SoapObject;
import org.ksoap2.serialization.SoapSerializationEnvelope;
import org.ksoap2.transport.HttpTransportSE;

public class WS_Localizacion extends AsyncTask<String, Void, String> {

    final String SOAP_ACTION = "http://servicios/Localizar";
    final String METHOD = "Localizar";
    final String NAMESPACE = "http://servicios/";
    final String ENDPOINTWS =
"http://172.16.1.47:8080/ServidorIndoorBluetooth/ServiciosLocalizacion?xsd=1";
    String respuesta = null;

    /*
    Metodo sobreescrito que inicializa la clase y llama al servicio web
    */
    @Override
    protected String doInBackground(String... args) {
        SoapObject userRequest = new SoapObject(NAMESPACE, METHOD);
        userRequest.addProperty("beacons", args[0]);

        SoapSerializationEnvelope envelope = new
        SoapSerializationEnvelope(SoapEnvelope.VER11);
        envelope.setOutputSoapObject(userRequest);

        try{
            HttpTransportSE androidHttpTransport = new HttpTransportSE(ENDPOINTWS);
            androidHttpTransport.debug = true;
            androidHttpTransport.call(SOAP_ACTION, envelope);

            respuesta = envelope.getResponse().toString();
        }
        catch (Exception e){
            e.printStackTrace();
        }
        return respuesta;
    }
}

```

```

}

import android.content.Context;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;

public class BaseHelper extends SQLiteOpenHelper {

    String sql2 = "CREATE TABLE Temp (Nombre TEXT, distancia DOUBLE )";

    /*
    constructor de la clase
    */
    public BaseHelper(Context context, String name, SQLiteDatabase.CursorFactory
factory, int version) {
        super(context, name, factory, version);
    }

    /*
    Metodo que crea la base de datos
    */
    @Override
    public void onCreate(SQLiteDatabase db) {
        db.execSQL(sql2);
    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {

    }
}

import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.util.Log;
import android.widget.Toast;

public class BaseDeDatos {

    /*
    Metodo que ingresa en la tabla temporal el nombre del beacon y su distancia con el
    smartphone
    */
    public static boolean IngresarTemporal(BaseHelper baseHelper, String nom, double
dist) {
        boolean res;
        SQLiteDatabase db = baseHelper.getWritableDatabase();
        if (db != null) {
            if (db != null) {
                ContentValues values = new ContentValues();
                values.put("Nombre", nom);
                values.put("distancia", dist);
                long i = db.insert("Temp", null, values);
                if (i > 0) {
                    res = true;
                } else
                    res = false;
            } else
                res = false;
        } else
            res = false;
        db.close();
        return res;
    }

    /*
    Metodo que limpia la tabla temporal
    */

```

```

public static Cursor LimpiarTablaTemporal(BaseHelper baseHelper){
    SQLiteDatabase db = baseHelper.getReadableDatabase();
    Cursor c = null;
    if (db!=null) {
        String sql = "delete from Temp";
        c = db.rawQuery(sql, null);
        if (!c.moveToFirst()) {
            c = null;
        }
    }
    db.close();
    return c;
}

/*
Metodo que obtiene los beacons y sus distancias
*/
public static Cursor ConsultaCompletaDistancia(BaseHelper baseHelper){
    SQLiteDatabase db = baseHelper.getReadableDatabase();
    Cursor c = null;
    if (db!=null) {
        String sql = "select Nombre, distancia " +
            "from Temp " +
            "order by distancia ASC";
        c = db.rawQuery(sql, null);
        if (!c.moveToFirst()) {
            c = null;
            Log.i("Error:", "error");
        }
    }
    db.close();
    return c;
}

/*
Metodo que obtiene la distancia y nombre de un beacons especifico
*/
public static Cursor ConsultaEspecifica(BaseHelper baseHelper, String nom){
    SQLiteDatabase db = baseHelper.getReadableDatabase();
    Cursor c = null;
    if (db!=null) {
        String sql = "select Nombre, distancia " +
            "from Temp " +
            "where Nombre = '"+nom+"'";
        c = db.rawQuery(sql, null);
        if (!c.moveToFirst()) {
            c = null;
        }
    }
    db.close();
    return c;
}
}

```

```

<?xml version="1.0" encoding="utf-8"?>
<android.support.design.widget.CoordinatorLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context="com.example.marco.indooruda.MainActivity">

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">

    <android.support.design.widget.AppBarLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"

```

```

        android:theme="@style/AppTheme.AppBarOverlay">

        <android.support.v7.widget.Toolbar
            android:id="@+id/toolbar"
            android:layout_width="match_parent"
            android:layout_height="?attr/actionBarSize"
            android:background="?attr/colorPrimary"
            android:titleTextColor="@color/letra_blanca"
            app:popupTheme="@style/AppTheme.PopupOverlay" />

    </android.support.design.widget.AppBarLayout>

    <include
        layout="@layout/content_main"
        android:layout_width="match_parent"
        android:layout_height="452dp"
        android:layout_weight="1" />

</LinearLayout>

<android.support.design.widget.FloatingActionButton
    android:id="@+id/fab"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_gravity="bottom|end"
    android:layout_margin="@dimen/fab_margin"
    android:layout_weight="1"
    android:tint="@color/letra_blanca"
    app:srcCompat="@android:drawable/ic_popup_sync" />

</android.support.design.widget.CoordinatorLayout>

<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/letra_blanca"
    tools:context="com.example.marco.indooruda.MainActivity">

    <LinearLayout
        android:layout_width="336dp"
        android:layout_height="498dp"
        android:background="@color/letra_blanca"
        android:orientation="vertical"
        android:weightSum="1"
        tools:layout_editor_absoluteX="24dp"
        tools:layout_editor_absoluteY="16dp">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_weight="0.13"
            android:background="@color/colorPrimaryDark"
            android:orientation="horizontal"
            android:weightSum="1">

            <TextView
                android:id="@+id/txt_salon"
                android:layout_width="wrap_content"
                android:layout_height="match_parent"
                android:layout_weight="0.87"
                android:background="@color/colorPrimaryDark"
                android:textColor="@color/letra_blanca" />

            <TextView
                android:id="@+id/txt_xy"

```

```

        android:layout_width="128dp"
        android:layout_height="match_parent"
        android:layout_weight="0.08"
        android:background="@color/colorPrimaryDark"
        android:textColor="@color/letra_blanca" />

</LinearLayout>

<ImageView
    android:id="@+id/imagen"
    android:layout_width="324dp"
    android:layout_height="270dp"
    android:background="@android:color/black"
    app:srcCompat="@color/colorPrimaryDark" />

<TextView
    android:id="@+id/textView3"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="@color/colorPrimary"
    android:text="Información Contextual"
    android:textColor="@color/letra_blanca"
    android:textStyle="bold" />

<Space
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_weight="0.35" />

<LinearLayout
    android:layout_width="308dp"
    android:layout_height="125dp"
    android:orientation="horizontal">

    <TextView
        android:id="@+id/msg_clase"
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:layout_weight="0.21"
        android:textColor="@color/colorPrimaryDark"
        android:textSize="18sp"
        android:textStyle="bold" />

    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:layout_weight="0.03"
        android:orientation="vertical"
        android:weightSum="1">

        <Button
            android:id="@+id/info"
            style="@android:style/Theme.DeviceDefault.Dialog.MinWidth"
            android:layout_width="match_parent"
            android:layout_height="38dp"
            android:autoText="false"
            android:background="@color/colorPrimary"
            android:drawableLeft="@android:drawable/ic_menu_search"
            android:drawableTint="@color/letra_blanca"
            android:text="@string/btn_lab"
            android:textColor="@color/letra_blanca"
            android:textSize="15sp"
            android:textStyle="bold"
            tools:style="@style/Base.Theme.AppCompat.Dialog.FixedSize" />

        <Space
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_weight="0.30" />

        <Button

```

```

        android:id="@+id/btn_clase"
        style="@android:style/Theme.DeviceDefault.Dialog.MinWidth"
        android:layout_width="match_parent"
        android:layout_height="38dp"
        android:autoText="false"
        android:background="@color/colorPrimary"
        android:drawableLeft="@android:drawable/ic_menu_search"
        android:drawableTint="@color/letra_blanca"
        android:text="@string/btn_clase"
        android:textColor="@color/letra_blanca"
        android:textSize="15sp"
        android:textStyle="bold"
        tools:style="@style/Base.Theme.AppCompat.Dialog.FixedSize" />

    </LinearLayout>

</LinearLayout>

</LinearLayout>
</android.support.constraint.ConstraintLayout>

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/letra_blanca"
    tools:context=".Dialogo_Clase"
    android:orientation="vertical"
    android:weightSum="1">

    <!-- TODO: Update blank fragment layout -->

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="164dp"
        android:layout_weight="0.13"
        android:orientation="horizontal">

        <ListView
            android:id="@+id/lista"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:divider="@color/separador"
            android:dividerHeight="2px" />
    </LinearLayout>

    <Space
        android:layout_width="match_parent"
        android:layout_height="17dp"
        android:layout_weight="0.04" />

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        android:weightSum="1">

        <Space
            android:layout_width="13dp"
            android:layout_height="51dp"
            android:layout_weight="0.38" />

        <Button
            android:id="@+id/boton"
            style="@android:style/Theme.DeviceDefault.Dialog.MinWidth"
            android:layout_width="121dp"
            android:layout_height="wrap_content"
            android:autoText="false"
            android:background="@color/colorPrimary"
            android:drawableLeft="?attr/actionModeCloseDrawable"

```



```

        android:drawableTint="@android:color/background_light"
        android:text="@string/btn_retorno"
        android:textColor="@color/letra_blanca"
        android:textSize="15sp"
        android:textStyle="bold"
        tools:style="@style/Base.Theme.AppCompat.Dialog.FixedSize"
        android:layout_weight="0.23" />

    <Space
        android:layout_width="13dp"
        android:layout_height="match_parent"
        android:layout_weight="0.13" />

</LinearLayout>

</LinearLayout>

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/letra_blanca"
    tools:context=".Dialogo_Config"
    android:orientation="vertical">

    <!-- TODO: Update blank fragment layout -->

    <ScrollView
        android:id="@+id/ScrlView"
        android:layout_width="fill_parent"
        android:layout_height="164dp">

        <RadioGroup
            android:layout_width="match_parent"
            android:layout_height="164dp"
            android:backgroundTint="@color/colorPrimary"
            android:scrollbars="vertical">

            <RadioButton
                android:id="@+id/rd_0"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:text="@string/conf_nuno" />

            <RadioButton
                android:id="@+id/rd_a"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:backgroundTint="@color/colorPrimary"
                android:text="@string/conf_a" />

            <RadioButton
                android:id="@+id/rd_b"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:text="@string/conf_b" />

            <RadioButton
                android:id="@+id/rd_c"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:text="@string/conf_c" />

            <RadioButton
                android:id="@+id/rd_d"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:text="@string/conf_d" />

            <RadioButton

```

```

        android:id="@+id/rd_e"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="@string/conf_e" />

    </RadioGroup>
</ScrollView>

<Space
    android:layout_width="match_parent"
    android:layout_height="17dp" />

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:weightSum="1">

    <Space
        android:layout_width="13dp"
        android:layout_height="wrap_content"
        android:layout_weight="0.15" />

    <Button
        android:id="@+id/btn"
        style="@android:style/Theme.DeviceDefault.Dialog.MinWidth"
        android:layout_width="121dp"
        android:layout_height="wrap_content"
        android:layout_weight="0.01"
        android:autoText="false"
        android:background="@color/colorPrimary"
        android:drawableLeft="@android:drawable/ic_menu_save"
        android:drawableTint="@android:color/white"
        android:text="@string/btn_acpetar"
        android:textColor="@color/letra_blanca"
        android:textSize="15sp"
        android:textStyle="bold"
        tools:style="@style/Base.Theme.AppCompat.Dialog.FixedSize"
        tools:text="@string/btn_acpetar" />

    <Space
        android:layout_width="13dp"
        android:layout_height="wrap_content"
        android:layout_weight="0.17" />

    <Button
        android:id="@+id/btn_can"
        style="@android:style/Theme.DeviceDefault.Dialog.MinWidth"
        android:layout_width="121dp"
        android:layout_height="wrap_content"
        android:autoText="false"
        android:background="@color/colorPrimary"
        android:drawableLeft="@android:drawable/ic_menu_close_clear_cancel"
        android:drawableTint="@android:color/white"
        android:text="@string/btn_cancel"
        android:textColor="@color/letra_blanca"
        android:textSize="15sp"
        android:textStyle="bold"
        tools:style="@style/Base.Theme.AppCompat.Dialog.FixedSize"
        tools:text="@string/btn_cancel" />

</LinearLayout>

</LinearLayout>

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"

```

```

    android:background="@color/letra_blanca"
    android:orientation="vertical"
    android:weightSum="1"
    tools:context=".Dialogo_Clase">

<!-- TODO: Update blank fragment layout -->

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="164dp"
    android:orientation="horizontal">

    <ListView
        android:id="@+id/lista"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:divider="@color/separador"
        android:dividerHeight="2px" />

</LinearLayout>

<Space
    android:layout_width="match_parent"
    android:layout_height="17dp" />

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="horizontal">

    <Space
        android:layout_width="70dp"
        android:layout_height="48dp"
        android:layout_weight="0.00" />

    <Button
        android:id="@+id/boton"
        style="@android:style/Theme.DeviceDefault.Dialog.MinWidth"
        android:layout_width="40dp"
        android:layout_height="wrap_content"
        android:layout_weight="0.01"
        android:autoText="false"
        android:background="@color/colorPrimary"
        android:drawableTint="@android:color/white"
        android:text="@string/btn_retorno"
        android:textColor="@color/letra_blanca"
        android:textSize="20sp"
        android:textStyle="bold"
        tools:style="@style/Base.Theme.AppCompat.Dialog.FixedSize"
        tools:text="@string/btn_retorno" />

    <Space
        android:layout_width="70dp"
        android:layout_height="51dp"
        android:layout_weight="0.00" />
</LinearLayout>

</LinearLayout>

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="@color/letra_blanca"
    android:orientation="vertical"
    android:weightSum="1"
    tools:context=".Dialogo_Clase">

<!-- TODO: Update blank fragment layout -->

```

```

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="164dp"
    android:orientation="horizontal"
    android:weightSum="1">

    <ListView
        android:id="@+id/lista"
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:divider="@color/separador"
        android:dividerHeight="2px" />

</LinearLayout>

<Space
    android:layout_width="match_parent"
    android:layout_height="41dp" />

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="53dp"
    android:orientation="horizontal">

    <Space
        android:layout_width="70dp"
        android:layout_height="48dp"
        android:layout_weight="0.00" />

    <Button
        android:id="@+id/boton"
        style="@android:style/Theme.DeviceDefault.Dialog.MinWidth"
        android:layout_width="40dp"
        android:layout_height="wrap_content"
        android:layout_weight="0.01"
        android:autoText="false"
        android:background="@color/colorPrimary"
        android:drawableTint="@android:color/white"
        android:text="@string/btn_retorno"
        android:textColor="@color/letra_blanca"
        android:textSize="20sp"
        android:textStyle="bold"
        tools:style="@style/Base.Theme.AppCompat.Dialog.FixedSize"
        tools:text="@string/btn_retorno" />

    <Space
        android:layout_width="70dp"
        android:layout_height="51dp"
        android:layout_weight="0.00" />
</LinearLayout>

</LinearLayout>

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:layout_width="match_parent"
    android:layout_height="match_parent">

    <TextView
        android:id="@+id/text1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textColor="@color/colorPrimary"
        android:textSize="17sp"
        android:textStyle="bold" />

    <TextView
        android:id="@+id/text2"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"

```

```

        android:textColor="@android:color/black"
        android:textSize="12sp"
        android:textStyle="bold" />
</LinearLayout>

```

#### Anexo IX: Script de la base de datos del sistema de localización en interiores con tecnología bluetooth

```

-- MySQL dump 10.13  Distrib 5.7.17, for Win64 (x86_64)
--
--
-----
-- Server version      5.7.19-log

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8 */;
/*!40103 SET @OLD_TIME_ZONE=@@TIME_ZONE */;
/*!40103 SET TIME_ZONE='+00:00' */;
/*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0 */;
/*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0 */;
/*!40101 SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;
/*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;

--
-- Table structure for table `asignatura`
--

DROP TABLE IF EXISTS `asignatura`;
/*!40101 SET @saved_cs_client      = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `asignatura` (
  `idAsignatura` int(11) NOT NULL AUTO_INCREMENT,
  `nombre` varchar(45) NOT NULL,
  `descripcion` varchar(1000) DEFAULT NULL,
  `idEscuela` int(11) NOT NULL,
  PRIMARY KEY (`idAsignatura`),
  KEY `fk_asignatura_escuela1_idx` (`idEscuela`),
  CONSTRAINT `fk_asignatura_escuela1` FOREIGN KEY (`idEscuela`) REFERENCES `escuela`
(`idEscuela`) ON DELETE NO ACTION ON UPDATE NO ACTION
) ENGINE=InnoDB AUTO_INCREMENT=18 DEFAULT CHARSET=utf8;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `asignatura`
--

LOCK TABLES `asignatura` WRITE;
/*!40000 ALTER TABLE `asignatura` DISABLE KEYS */;
INSERT INTO `asignatura` VALUES
(5, 'PROGRAMACIÃ"N', 'Java', 3), (6, 'PROYECTOS', 'Scrum', 3), (7, 'SIG', 'Arcgis', 3), (8, 'ELECTRON
ICA', 'Arduino', 3), (9, 'BASE DE DATOS', 'Oracle', 3), (10, 'ANALISIS DE
SISTEMAS', 'UML', 3), (11, 'TELECOMUNICACIONES', 'Redes
', 3), (12, 'MATEMATICAS', 'Integrales', 3), (13, 'ESTADISTICA', 'Lenguajes R', 3), (14, 'SISTEMAS
OPERATIVOS', 'Linux', 3), (15, 'INTELIGENCIA ARTIFICIAL', 'Lenguaje Prolog', 3), (16, 'SEGURIDAD
DE SISTEMAS', 'Conceptos', 3), (17, 'PRODUCCION', 'Restricciones', 3);
/*!40000 ALTER TABLE `asignatura` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Table structure for table `asignatura_curso`
--

DROP TABLE IF EXISTS `asignatura_curso`;
/*!40101 SET @saved_cs_client      = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `asignatura_curso` (
  `idAsignatura` int(11) NOT NULL,
  `idCurso` int(11) NOT NULL,
  `idProfesor` int(11) NOT NULL,
  PRIMARY KEY (`idAsignatura`, `idCurso`),
  KEY `fk_asignaturaCurso_asignatural_idx` (`idAsignatura`),

```

```

    KEY `fk_asignaturaCurso_cursor1_idx` (`idCurso`),
    KEY `fk_asignaturaCurso_usuario1_idx` (`idProfesor`),
    CONSTRAINT `fk_asignaturaCurso_asignatura1` FOREIGN KEY (`idAsignatura`) REFERENCES
`asignatura` (`idAsignatura`) ON DELETE NO ACTION ON UPDATE NO ACTION,
    CONSTRAINT `fk_asignaturaCurso_cursor1` FOREIGN KEY (`idCurso`) REFERENCES `curso`
(`idCurso`) ON DELETE NO ACTION ON UPDATE NO ACTION,
    CONSTRAINT `fk_asignaturaCurso_usuario1` FOREIGN KEY (`idProfesor`) REFERENCES
`usuario` (`idUsuario`) ON DELETE NO ACTION ON UPDATE NO ACTION
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `asignatura_curso`
--

LOCK TABLES `asignatura_curso` WRITE;
/*!40000 ALTER TABLE `asignatura_curso` DISABLE KEYS */;
INSERT INTO `asignatura_curso` VALUES
(5,11,4),(10,17,4),(6,13,5),(7,18,6),(17,18,6),(8,15,7),(15,17,7),(9,19,8),(11,14,9),(12
,11,11),(13,20,12),(14,12,13),(16,19,14);
/*!40000 ALTER TABLE `asignatura_curso` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Table structure for table `beacon`
--

DROP TABLE IF EXISTS `beacon`;
/*!40101 SET @saved_cs_client = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `beacon` (
  `idbeacon` int(11) NOT NULL AUTO_INCREMENT,
  `nombre` varchar(45) NOT NULL,
  `coor_x` float NOT NULL,
  `coor_y` float NOT NULL,
  `letra` varchar(3) NOT NULL,
  `lugar_idlugar` int(11) NOT NULL,
  PRIMARY KEY (`idbeacon`),
  KEY `fk_beacon_lugar1_idx` (`lugar_idlugar`),
  CONSTRAINT `fk_beacon_lugar1` FOREIGN KEY (`lugar_idlugar`) REFERENCES `lugar`
(`idlugar`) ON DELETE NO ACTION ON UPDATE NO ACTION
) ENGINE=InnoDB AUTO_INCREMENT=6 DEFAULT CHARSET=utf8;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `beacon`
--

LOCK TABLES `beacon` WRITE;
/*!40000 ALTER TABLE `beacon` DISABLE KEYS */;
INSERT INTO `beacon` VALUES
(1,'b00100001',0,6,'C',7),(2,'b00200002',5,12,'B',7),(3,'b00300003',13,0,'A',7),(4,'b004
00004',-1,-1,'L4',3),(5,'b00500005',-2,-2,'L5',4);
/*!40000 ALTER TABLE `beacon` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Table structure for table `campoamplio`
--

DROP TABLE IF EXISTS `campoamplio`;
/*!40101 SET @saved_cs_client = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `campoamplio` (
  `idCampoAmplio` int(11) NOT NULL AUTO_INCREMENT,
  `nombre` varchar(300) NOT NULL,
  PRIMARY KEY (`idCampoAmplio`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
/*!40101 SET character_set_client = @saved_cs_client */;

--

```

```

-- Dumping data for table `campoamplio`
--

LOCK TABLES `campoamplio` WRITE;
/*!40000 ALTER TABLE `campoamplio` DISABLE KEYS */;
/*!40000 ALTER TABLE `campoamplio` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Table structure for table `campodetallado`
--

DROP TABLE IF EXISTS `campodetallado`;
/*!40101 SET @saved_cs_client      = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `campodetallado` (
  `idCampoDetallado` int(11) NOT NULL AUTO_INCREMENT,
  `nombre` varchar(300) NOT NULL,
  `idCampoEspecifico` int(11) NOT NULL,
  PRIMARY KEY (`idCampoDetallado`),
  KEY `fk_campodetallado_campoespecifico1_idx` (`idCampoEspecifico`),
  CONSTRAINT `fk_campodetallado_campoespecifico1` FOREIGN KEY (`idCampoEspecifico`)
REFERENCES `campoespecifico` (`idCampoEspecifico`) ON DELETE NO ACTION ON UPDATE NO
ACTION
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `campodetallado`
--

LOCK TABLES `campodetallado` WRITE;
/*!40000 ALTER TABLE `campodetallado` DISABLE KEYS */;
/*!40000 ALTER TABLE `campodetallado` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Table structure for table `campoespecifico`
--

DROP TABLE IF EXISTS `campoespecifico`;
/*!40101 SET @saved_cs_client      = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `campoespecifico` (
  `idCampoEspecifico` int(11) NOT NULL AUTO_INCREMENT,
  `nombre` varchar(300) NOT NULL,
  `idCampoAmplio` int(11) NOT NULL,
  PRIMARY KEY (`idCampoEspecifico`),
  KEY `fk_campoespecifico_campoamplio1_idx` (`idCampoAmplio`),
  CONSTRAINT `fk_campoespecifico_campoamplio1` FOREIGN KEY (`idCampoAmplio`) REFERENCES
`campoamplio` (`idCampoAmplio`) ON DELETE NO ACTION ON UPDATE NO ACTION
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `campoespecifico`
--

LOCK TABLES `campoespecifico` WRITE;
/*!40000 ALTER TABLE `campoespecifico` DISABLE KEYS */;
/*!40000 ALTER TABLE `campoespecifico` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Table structure for table `curso`
--

DROP TABLE IF EXISTS `curso`;
/*!40101 SET @saved_cs_client      = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `curso` (

```

```

`idCurso` int(11) NOT NULL AUTO_INCREMENT,
`nombre` varchar(45) NOT NULL,
`escuela_idEscuela` int(11) NOT NULL,
PRIMARY KEY (`idCurso`),
KEY `fk_curso_escuela1_idx` (`escuela_idEscuela`),
CONSTRAINT `fk_curso_escuela1` FOREIGN KEY (`escuela_idEscuela`) REFERENCES `escuela`
(`idEscuela`) ON DELETE NO ACTION ON UPDATE NO ACTION
) ENGINE=InnoDB AUTO_INCREMENT=21 DEFAULT CHARSET=utf8;
/*!40101 SET character_set_client = @saved_cs_client */;

```

```

--
-- Dumping data for table `curso`
--

```

```

LOCK TABLES `curso` WRITE;
/*!40000 ALTER TABLE `curso` DISABLE KEYS */;
INSERT INTO `curso` VALUES (11,'PRIMERO A',3),(12,'SEGUNDO A',3),(13,'TERCERO
A',3),(14,'CUARTO A',3),(15,'QUINTO A',3),(16,'SEXTO A',3),(17,'SEPTIMO
A',3),(18,'OCTAVO A',3),(19,'NOVENO A',3),(20,'DECIMO A',3);
/*!40000 ALTER TABLE `curso` ENABLE KEYS */;
UNLOCK TABLES;

```

```

--
-- Table structure for table `departamento`
--

```

```

DROP TABLE IF EXISTS `departamento`;
/*!40101 SET @saved_cs_client = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `departamento` (
  `idDepartamento` int(11) NOT NULL AUTO_INCREMENT,
  `nombre` varchar(45) NOT NULL,
  `idInstitucion` int(11) NOT NULL,
  `idlugar` int(11) NOT NULL,
  PRIMARY KEY (`idDepartamento`),
  KEY `fk_departamento_universidad1_idx` (`idInstitucion`),
  KEY `fk_departamento_lugar1_idx` (`idlugar`),
  CONSTRAINT `fk_departamento_lugar1` FOREIGN KEY (`idlugar`) REFERENCES `lugar`
(`idlugar`) ON DELETE NO ACTION ON UPDATE NO ACTION,
  CONSTRAINT `fk_departamento_universidad1` FOREIGN KEY (`idInstitucion`) REFERENCES
`institucion` (`idInstitucion`) ON DELETE NO ACTION ON UPDATE NO ACTION
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
/*!40101 SET character_set_client = @saved_cs_client */;

```

```

--
-- Dumping data for table `departamento`
--

```

```

LOCK TABLES `departamento` WRITE;
/*!40000 ALTER TABLE `departamento` DISABLE KEYS */;
/*!40000 ALTER TABLE `departamento` ENABLE KEYS */;
UNLOCK TABLES;

```

```

--
-- Table structure for table `distancia_beacon`
--

```

```

DROP TABLE IF EXISTS `distancia_beacon`;
/*!40101 SET @saved_cs_client = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `distancia_beacon` (
  `iddistancia_beacon` int(11) NOT NULL AUTO_INCREMENT,
  `valor` varchar(45) NOT NULL,
  `beacon_a` int(11) NOT NULL,
  `beacon_b` int(11) NOT NULL,
  PRIMARY KEY (`iddistancia_beacon`),
  KEY `fk_distancia_beacon_beacon1_idx` (`beacon_a`),
  KEY `fk_distancia_beacon_beacon2_idx` (`beacon_b`),
  CONSTRAINT `fk_distancia_beacon_beacon1` FOREIGN KEY (`beacon_a`) REFERENCES `beacon`
(`idbeacon`) ON DELETE NO ACTION ON UPDATE NO ACTION,

```



```

    CONSTRAINT `fk_distancia_beacon_beacon2` FOREIGN KEY (`beacon_b`) REFERENCES `beacon`
(`idbeacon`) ON DELETE NO ACTION ON UPDATE NO ACTION
) ENGINE=InnoDB AUTO_INCREMENT=7 DEFAULT CHARSET=utf8;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `distancia_beacon`
--

LOCK TABLES `distancia_beacon` WRITE;
/*!40000 ALTER TABLE `distancia_beacon` DISABLE KEYS */;
INSERT INTO `distancia_beacon` VALUES
(1,'4.0',1,2), (2,'8.0',1,3), (3,'4.0',2,1), (4,'8.0',2,3), (5,'4.0',3,1), (6,'8.0',3,2);
/*!40000 ALTER TABLE `distancia_beacon` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Table structure for table `escuela`
--

DROP TABLE IF EXISTS `escuela`;
/*!40101 SET @saved_cs_client      = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `escuela` (
  `idEscuela` int(11) NOT NULL AUTO_INCREMENT,
  `nombre` varchar(45) NOT NULL,
  `idFacultad` int(11) NOT NULL,
  `idlugar` int(11) NOT NULL,
  PRIMARY KEY (`idEscuela`),
  KEY `fk_escuela_facultad1_idx` (`idFacultad`),
  KEY `fk_escuela_lugar1_idx` (`idlugar`),
  CONSTRAINT `fk_escuela_facultad1` FOREIGN KEY (`idFacultad`) REFERENCES `facultad`
(`idFacultad`) ON DELETE NO ACTION ON UPDATE NO ACTION,
  CONSTRAINT `fk_escuela_lugar1` FOREIGN KEY (`idlugar`) REFERENCES `lugar` (`idlugar`)
ON DELETE NO ACTION ON UPDATE NO ACTION
) ENGINE=InnoDB AUTO_INCREMENT=4 DEFAULT CHARSET=utf8;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `escuela`
--

LOCK TABLES `escuela` WRITE;
/*!40000 ALTER TABLE `escuela` DISABLE KEYS */;
INSERT INTO `escuela` VALUES (3,'INGENIERIA EN SISTEMAS',3,1);
/*!40000 ALTER TABLE `escuela` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Table structure for table `escuela_campodetallado`
--

DROP TABLE IF EXISTS `escuela_campodetallado`;
/*!40101 SET @saved_cs_client      = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `escuela_campodetallado` (
  `idCampoDetallado` int(11) NOT NULL,
  `escuela_idEscuela` int(11) NOT NULL,
  `idEscuela_campodetallado` int(11) NOT NULL AUTO_INCREMENT,
  PRIMARY KEY (`idEscuela_campodetallado`),
  KEY `fk_escuela_campodetallado_campodetallado1_idx` (`idCampoDetallado`),
  KEY `fk_escuela_campodetallado_escuela1_idx` (`escuela_idEscuela`),
  CONSTRAINT `fk_escuela_campodetallado_campodetallado1` FOREIGN KEY
(`idCampoDetallado`) REFERENCES `campodetallado` (`idCampoDetallado`) ON DELETE NO
ACTION ON UPDATE NO ACTION,
  CONSTRAINT `fk_escuela_campodetallado_escuela1` FOREIGN KEY (`escuela_idEscuela`)
REFERENCES `escuela` (`idEscuela`) ON DELETE NO ACTION ON UPDATE NO ACTION
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
/*!40101 SET character_set_client = @saved_cs_client */;

--

```

```

-- Dumping data for table `escuela_campodetallado`
--

LOCK TABLES `escuela_campodetallado` WRITE;
/*!40000 ALTER TABLE `escuela_campodetallado` DISABLE KEYS */;
/*!40000 ALTER TABLE `escuela_campodetallado` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Table structure for table `estudiante`
--

DROP TABLE IF EXISTS `estudiante`;
/*!40101 SET @saved_cs_client      = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `estudiante` (
  `idTablaestudiante` int(11) NOT NULL AUTO_INCREMENT,
  `idEstudiante` int(11) NOT NULL,
  `idAsignatura` int(11) NOT NULL,
  `idCurso` int(11) NOT NULL,
  PRIMARY KEY (`idTablaestudiante`),
  KEY `fk_estudiante_usuario1_idx` (`idEstudiante`),
  KEY `fk_estudiante_asignaturaCurso1_idx` (`idAsignatura`,`idCurso`),
  CONSTRAINT `fk_estudiante_asignaturaCurso1` FOREIGN KEY (`idAsignatura`,`idCurso`)
REFERENCES `asignatura_curso` (`idAsignatura`,`idCurso`) ON DELETE NO ACTION ON UPDATE
NO ACTION,
  CONSTRAINT `fk_estudiante_usuario1` FOREIGN KEY (`idEstudiante`) REFERENCES `usuario`
(`idUsuario`) ON DELETE NO ACTION ON UPDATE NO ACTION
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `estudiante`
--

LOCK TABLES `estudiante` WRITE;
/*!40000 ALTER TABLE `estudiante` DISABLE KEYS */;
/*!40000 ALTER TABLE `estudiante` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Table structure for table `evento`
--

DROP TABLE IF EXISTS `evento`;
/*!40101 SET @saved_cs_client      = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `evento` (
  `idEvento` int(11) NOT NULL AUTO_INCREMENT,
  `titulo` varchar(300) NOT NULL,
  `descripcion` varchar(1000) DEFAULT NULL,
  `fecha` date NOT NULL,
  `hora` time NOT NULL,
  `idlugar` int(11) NOT NULL,
  `tipoEvento` int(11) NOT NULL DEFAULT '1',
  PRIMARY KEY (`idEvento`),
  KEY `fk_evento_lugar1_idx` (`idlugar`),
  KEY `fk_evento_tipo_Evento1_idx` (`tipoEvento`),
  CONSTRAINT `fk_evento_lugar1` FOREIGN KEY (`idlugar`) REFERENCES `lugar` (`idlugar`)
ON DELETE NO ACTION ON UPDATE NO ACTION,
  CONSTRAINT `fk_evento_tipo_Evento1` FOREIGN KEY (`tipoEvento`) REFERENCES
`tipo_evento` (`idtipo_Evento`) ON DELETE NO ACTION ON UPDATE NO ACTION
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `evento`
--

LOCK TABLES `evento` WRITE;
/*!40000 ALTER TABLE `evento` DISABLE KEYS */;

```

```

/*!40000 ALTER TABLE `evento` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Table structure for table `evento_campodetallado`
--

DROP TABLE IF EXISTS `evento_campodetallado`;
/*!40101 SET @saved_cs_client      = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `evento_campodetallado` (
  `idCampoDetallado` int(11) NOT NULL,
  `idEvento` int(11) NOT NULL,
  `id_EventoCampodetallado` int(11) NOT NULL AUTO_INCREMENT,
  PRIMARY KEY (`id_EventoCampodetallado`),
  KEY `fk_evento_campodetallado_campodetallado1_idx` (`idCampoDetallado`),
  KEY `fk_evento_campodetallado_evento1_idx` (`idEvento`),
  CONSTRAINT `fk_evento_campodetallado_campodetallado1` FOREIGN KEY (`idCampoDetallado`)
REFERENCES `campodetallado` (`idCampoDetallado`) ON DELETE NO ACTION ON UPDATE NO
ACTION,
  CONSTRAINT `fk_evento_campodetallado_evento1` FOREIGN KEY (`idEvento`) REFERENCES
`evento` (`idEvento`) ON DELETE NO ACTION ON UPDATE NO ACTION
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `evento_campodetallado`
--

LOCK TABLES `evento_campodetallado` WRITE;
/*!40000 ALTER TABLE `evento_campodetallado` DISABLE KEYS */;
/*!40000 ALTER TABLE `evento_campodetallado` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Table structure for table `facultad`
--

DROP TABLE IF EXISTS `facultad`;
/*!40101 SET @saved_cs_client      = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `facultad` (
  `idFacultad` int(11) NOT NULL AUTO_INCREMENT,
  `Nombre` varchar(45) NOT NULL,
  `idInstitucion` int(11) NOT NULL,
  `idlugar` int(11) NOT NULL,
  PRIMARY KEY (`idFacultad`),
  KEY `fk_facultad_universidad1_idx` (`idInstitucion`),
  KEY `fk_facultad_lugar1_idx` (`idlugar`),
  CONSTRAINT `fk_facultad_lugar1` FOREIGN KEY (`idlugar`) REFERENCES `lugar` (`idlugar`)
ON DELETE NO ACTION ON UPDATE NO ACTION,
  CONSTRAINT `fk_facultad_universidad1` FOREIGN KEY (`idInstitucion`) REFERENCES
`institucion` (`idInstitucion`) ON DELETE NO ACTION ON UPDATE NO ACTION
) ENGINE=InnoDB AUTO_INCREMENT=4 DEFAULT CHARSET=utf8;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `facultad`
--

LOCK TABLES `facultad` WRITE;
/*!40000 ALTER TABLE `facultad` DISABLE KEYS */;
INSERT INTO `facultad` VALUES (2,'FILOSOFIA',2,1),(3,'ADMINISTRACION',2,1);
/*!40000 ALTER TABLE `facultad` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Table structure for table `horario`
--

DROP TABLE IF EXISTS `horario`;

```

```

/*!40101 SET @saved_cs_client      = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `horario` (
  `idhorario` int(11) NOT NULL AUTO_INCREMENT,
  `dia` varchar(20) NOT NULL,
  `horaInicio` time NOT NULL,
  `horaFin` time NOT NULL,
  `idAsignatura` int(11) NOT NULL,
  `idCurso` int(11) NOT NULL,
  `lugar_idlugar` int(11) NOT NULL,
  PRIMARY KEY (`idhorario`),
  KEY `fk_horario_asignaturaCurso1_idx` (`idAsignatura`,`idCurso`),
  KEY `fk_horario_lugar1_idx` (`lugar_idlugar`),
  CONSTRAINT `fk_horario_asignaturaCurso1` FOREIGN KEY (`idAsignatura`,`idCurso`)
REFERENCES `asignatura_curso` (`idAsignatura`,`idCurso`) ON DELETE NO ACTION ON UPDATE
NO ACTION,
  CONSTRAINT `fk_horario_lugar1` FOREIGN KEY (`lugar_idlugar`) REFERENCES `lugar`
(`idlugar`) ON DELETE NO ACTION ON UPDATE NO ACTION
) ENGINE=InnoDB AUTO_INCREMENT=34 DEFAULT CHARSET=utf8;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `horario`
--

LOCK TABLES `horario` WRITE;
/*!40000 ALTER TABLE `horario` DISABLE KEYS */;
INSERT INTO `horario` VALUES
(1,'1','07:00:00','09:00:00',5,11,3),(2,'1','09:00:00','12:00:00',6,13,3),(3,'1','14:00:
00','17:00:00',7,18,3),(4,'2','07:00:00','09:00:00',8,15,3),(5,'2','09:00:00','12:00:00'
,9,19,3),(6,'2','14:00:00','17:00:00',5,11,3),(7,'3','07:00:00','09:00:00',5,11,3),(8,'3
','09:00:00','12:00:00',12,11,3),(9,'3','14:00:00','17:00:00',8,15,3),(10,'4','07:00:00'
,'09:00:00',15,17,3),(11,'4','09:00:00','12:00:00',13,20,3),(12,'4','12:00:00','14:00:00'
,16,19,3),(13,'5','07:00:00','09:00:00',14,12,3),(14,'5','09:00:00','12:00:00',17,18,3)
,(15,'5','14:00:00','17:00:00',10,17,3),(16,'1','09:00:00','12:00:00',13,20,4),(17,'1','
14:00:00','17:00:00',6,13,4),(18,'1','07:00:00','09:00:00',12,11,4),(19,'2','07:00:00','
09:00:00',7,18,4),(20,'2','09:00:00','12:00:00',16,19,4),(21,'2','14:00:00','17:00:00',9
,19,4),(22,'3','09:00:00','12:00:00',8,15,4),(23,'3','14:00:00','17:00:00',17,18,4),(24,
'3','07:00:00','09:00:00',11,14,4),(25,'4','07:00:00','09:00:00',13,20,4),(26,'4','09:00:
00','12:00:00',15,17,4),(27,'4','14:00:00','17:00:00',7,18,4),(28,'5','07:00:00','09:00:
00',16,19,4),(29,'5','09:00:00','12:00:00',6,13,4),(30,'5','14:00:00','17:00:00',12,11,
4),(31,'4','17:00:00','20:00:00',10,17,3),(32,'5','16:00:00','21:00:00',17,18,4);
/*!40000 ALTER TABLE `horario` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Table structure for table `institucion`
--

DROP TABLE IF EXISTS `institucion`;
/*!40101 SET @saved_cs_client      = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `institucion` (
  `idInstitucion` int(11) NOT NULL AUTO_INCREMENT,
  `nombre` varchar(45) NOT NULL,
  `idlugar` int(11) NOT NULL,
  PRIMARY KEY (`idInstitucion`),
  KEY `fk_universidad_lugar1_idx` (`idlugar`),
  CONSTRAINT `fk_universidad_lugar1` FOREIGN KEY (`idlugar`) REFERENCES `lugar`
(`idlugar`) ON DELETE NO ACTION ON UPDATE NO ACTION
) ENGINE=InnoDB AUTO_INCREMENT=3 DEFAULT CHARSET=utf8;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `institucion`
--

LOCK TABLES `institucion` WRITE;
/*!40000 ALTER TABLE `institucion` DISABLE KEYS */;
INSERT INTO `institucion` VALUES (2,'UNIVERSIDAD DEL AZUAY',1);
/*!40000 ALTER TABLE `institucion` ENABLE KEYS */;

```

```

UNLOCK TABLES;

--
-- Table structure for table `lugar`
--

DROP TABLE IF EXISTS `lugar`;
/*!40101 SET @saved_cs_client      = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `lugar` (
  `idlugar` int(11) NOT NULL AUTO_INCREMENT,
  `Nombre` varchar(45) NOT NULL,
  `Descripcion` varchar(100) DEFAULT NULL,
  `CoordenadaX` float DEFAULT NULL,
  `CoordenadaY` float DEFAULT NULL,
  `idLugarContenedor` int(11) DEFAULT NULL,
  `Tipo_Lugar_idTipo_Lugar` int(11) NOT NULL,
  `Imagen` varchar(150) DEFAULT NULL,
  PRIMARY KEY (`idlugar`),
  KEY `fk_lugar_lugar1_idx` (`idLugarContenedor`),
  KEY `fk_lugar_Tipo_Lugar1_idx` (`Tipo_Lugar_idTipo_Lugar`),
  CONSTRAINT `fk_lugar_Tipo_Lugar1` FOREIGN KEY (`Tipo_Lugar_idTipo_Lugar`) REFERENCES
`tipo_lugar` (`idTipo_Lugar`) ON DELETE NO ACTION ON UPDATE NO ACTION,
  CONSTRAINT `fk_lugar_lugar1` FOREIGN KEY (`idLugarContenedor`) REFERENCES `lugar`
(`idlugar`) ON DELETE NO ACTION ON UPDATE NO ACTION
) ENGINE=InnoDB AUTO_INCREMENT=9 DEFAULT CHARSET=utf8;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `lugar`
--

LOCK TABLES `lugar` WRITE;
/*!40000 ALTER TABLE `lugar` DISABLE KEYS */;
INSERT INTO `lugar` VALUES (1,'UNIVERSIDAD DEL
AZUAY','Uda',NULL,NULL,NULL,6,''), (2,'FILOSOFIA','Filosofia',NULL,NULL,1,5,NULL), (3,'LAB
ORATORIO 4','Cuarto Piso;Posee Proyector;Capacidad para 30 alumnos;Posee 18
computadoras',NULL,NULL,2,1,'D:\\\\Trabajos
Universidad\\\\Tesis\\\\Infraestructura\\\\Bluetooth\\\\ServidorIndoorBluetooth\\\\web\\\\
\\Lab4.jpg'), (4,'LABORATORIO 5','Cuarto Piso;Posee Proyector;Capacidad para 25
alumnos;Posee 15 computadoras',NULL,NULL,2,1,'D:\\\\Trabajos
Universidad\\\\Tesis\\\\Infraestructura\\\\Bluetooth\\\\ServidorIndoorBluetooth\\\\web\\\\
\\Lab5.jpg'), (5,'LABORATORIO 6','Cuarto Piso;Posee Proyector;Capacidad para 30
alumnos;Posee 20 computadoras',NULL,NULL,2,1,NULL), (6,'LABORATORIO 7','Cuarto Piso;Posee
Proyector;Capacidad para 25 alumnos;Posee 15
computadoras',NULL,NULL,2,1,NULL), (7,'PASILLO','',NULL,NULL,2,2,'D:\\\\Trabajos
Universidad\\\\Tesis\\\\Infraestructura\\\\Bluetooth\\\\ServidorIndoorBluetooth\\\\web\\\\
\\Pasillo.jpg'), (8,'DEPARTAMENTO DE PROYECTOS','Proyectos de Ingenieria en
Sistemas',NULL,NULL,2,7,NULL);
/*!40000 ALTER TABLE `lugar` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Table structure for table `lugarescercanos`
--

DROP TABLE IF EXISTS `lugarescercanos`;
/*!40101 SET @saved_cs_client      = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `lugarescercanos` (
  `idlugarescercanos` int(11) NOT NULL AUTO_INCREMENT,
  `lugar_principal` int(11) NOT NULL,
  `lugar_cercano` int(11) NOT NULL,
  PRIMARY KEY (`idlugarescercanos`),
  KEY `fk_lugarescercanos_lugar1_idx` (`lugar_principal`),
  KEY `fk_lugarescercanos_lugar2_idx` (`lugar_cercano`),
  CONSTRAINT `fk_lugarescercanos_lugar1` FOREIGN KEY (`lugar_principal`) REFERENCES
`lugar` (`idlugar`) ON DELETE NO ACTION ON UPDATE NO ACTION,
  CONSTRAINT `fk_lugarescercanos_lugar2` FOREIGN KEY (`lugar_cercano`) REFERENCES
`lugar` (`idlugar`) ON DELETE NO ACTION ON UPDATE NO ACTION
) ENGINE=InnoDB AUTO_INCREMENT=5 DEFAULT CHARSET=utf8;

```

```

/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `lugarescercanos`
--

LOCK TABLES `lugarescercanos` WRITE;
/*!40000 ALTER TABLE `lugarescercanos` DISABLE KEYS */;
INSERT INTO `lugarescercanos` VALUES (1,7,6),(2,7,5),(3,7,4),(4,7,3);
/*!40000 ALTER TABLE `lugarescercanos` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Table structure for table `noticia`
--

DROP TABLE IF EXISTS `noticia`;
/*!40101 SET @saved_cs_client = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `noticia` (
  `idNoticia` int(11) NOT NULL AUTO_INCREMENT,
  `titulo` varchar(300) NOT NULL,
  `descripcion` varchar(1000) NOT NULL,
  `Fecha` date NOT NULL,
  `Hora` time NOT NULL,
  PRIMARY KEY (`idNoticia`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `noticia`
--

LOCK TABLES `noticia` WRITE;
/*!40000 ALTER TABLE `noticia` DISABLE KEYS */;
/*!40000 ALTER TABLE `noticia` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Table structure for table `noticia_campodetallado`
--

DROP TABLE IF EXISTS `noticia_campodetallado`;
/*!40101 SET @saved_cs_client = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `noticia_campodetallado` (
  `idCampoDetallado` int(11) NOT NULL,
  `idNoticia` int(11) NOT NULL,
  `idNoticia_CampoDetallado` int(11) NOT NULL AUTO_INCREMENT,
  PRIMARY KEY (`idNoticia_CampoDetallado`),
  KEY `fk_noticia_campodetallado_campodetallado1_idx` (`idCampoDetallado`),
  KEY `fk_noticia_campodetallado_noticial_idx` (`idNoticia`),
  CONSTRAINT `fk_noticia_campodetallado_campodetallado1` FOREIGN KEY
(`idCampoDetallado`) REFERENCES `campodetallado` (`idCampoDetallado`) ON DELETE NO
ACTION ON UPDATE NO ACTION,
  CONSTRAINT `fk_noticia_campodetallado_noticial` FOREIGN KEY (`idNoticia`) REFERENCES
`noticia` (`idNoticia`) ON DELETE NO ACTION ON UPDATE NO ACTION
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `noticia_campodetallado`
--

LOCK TABLES `noticia_campodetallado` WRITE;
/*!40000 ALTER TABLE `noticia_campodetallado` DISABLE KEYS */;
/*!40000 ALTER TABLE `noticia_campodetallado` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Table structure for table `persona`

```

```

--
DROP TABLE IF EXISTS `persona`;
/*!40101 SET @saved_cs_client      = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `persona` (
  `idPersona` int(11) NOT NULL AUTO_INCREMENT,
  `nombre` varchar(45) NOT NULL,
  `apellido` varchar(45) NOT NULL,
  `correo` varchar(100) NOT NULL,
  `telefono` varchar(20) DEFAULT NULL,
  `tituloProfesional` varchar(45) DEFAULT NULL,
  PRIMARY KEY (`idPersona`)
) ENGINE=InnoDB AUTO_INCREMENT=15 DEFAULT CHARSET=utf8;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `persona`
--

LOCK TABLES `persona` WRITE;
/*!40000 ALTER TABLE `persona` DISABLE KEYS */;
INSERT INTO `persona` VALUES
(4, 'LENIN', 'ERAZO', 'correo1@uazuay.edu.ec', '111111', 'Ingeniero'), (5, 'CHESTER', 'SELLERS',
'correo2@uazuay.edu.ec', '222222', 'Ingeniero'), (6, 'JUAN', 'PEREZ', 'correo3@uazuay.edu.ec',
'333333', 'Ingeniero'), (7, 'LUIS', 'LOPEZ', 'correo4@uazuay.edu.ec', '444444', 'Ingeniero'), (8
, 'MIGUEL', 'GARCIA', 'correo5@uazuay.edu.ec', '555555', 'Ingeniero'), (9, 'CARLOS', 'VELEZ', 'cor
reo6@uazuay.edu.ec', '1115111', 'Ingeniero'), (10, 'JUAN', 'ROMERO', 'correo7@uazuay.edu.ec', '
22515122', 'Ingeniero'), (11, 'XIMENA', 'BURBANO', 'correo8@uazuay.edu.ec', '352333', 'Ingenier
o'), (12, 'DAMIAN', 'RODR GUEZ', 'correo9@uazuay.edu.ec', '447444', 'Ingeniero'), (13, 'ROBERTO
', 'MORA', 'correo10@uazuay.edu.ec', '118421', 'Ingeniero'), (14, 'ALBERTO', 'ZAMORA', 'correoi1
@uazuay.edu.ec', '847565', 'Ingeniero');
/*!40000 ALTER TABLE `persona` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Table structure for table `persona_campodetallado`
--

DROP TABLE IF EXISTS `persona_campodetallado`;
/*!40101 SET @saved_cs_client      = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `persona_campodetallado` (
  `idPersona_Campodetallado` int(11) NOT NULL AUTO_INCREMENT,
  `idPersona` int(11) NOT NULL,
  `idCampoDetallado` int(11) NOT NULL,
  PRIMARY KEY (`idPersona_Campodetallado`),
  KEY `fk_persona_campodetallado_personal_idx` (`idPersona`),
  KEY `fk_persona_campodetallado_campodetallado1_idx` (`idCampoDetallado`),
  CONSTRAINT `fk_persona_campodetallado_campodetallado1` FOREIGN KEY
(`idCampoDetallado`) REFERENCES `campodetallado` (`idCampoDetallado`) ON DELETE NO
ACTION ON UPDATE NO ACTION,
  CONSTRAINT `fk_persona_campodetallado_personal` FOREIGN KEY (`idPersona`) REFERENCES
`persona` (`idPersona`) ON DELETE NO ACTION ON UPDATE NO ACTION
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `persona_campodetallado`
--

LOCK TABLES `persona_campodetallado` WRITE;
/*!40000 ALTER TABLE `persona_campodetallado` DISABLE KEYS */;
/*!40000 ALTER TABLE `persona_campodetallado` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Table structure for table `tipo_evento`
--

DROP TABLE IF EXISTS `tipo_evento`;

```

```

/*!40101 SET @saved_cs_client      = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `tipo_evento` (
  `idtipo_Evento` int(11) NOT NULL AUTO_INCREMENT,
  `nombre` varchar(45) NOT NULL,
  `descripcion` varchar(200) DEFAULT NULL,
  PRIMARY KEY (`idtipo_Evento`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `tipo_evento`
--

LOCK TABLES `tipo_evento` WRITE;
/*!40000 ALTER TABLE `tipo_evento` DISABLE KEYS */;
/*!40000 ALTER TABLE `tipo_evento` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Table structure for table `tipo_lugar`
--

DROP TABLE IF EXISTS `tipo_lugar`;
/*!40101 SET @saved_cs_client      = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `tipo_lugar` (
  `idTipo_Lugar` int(11) NOT NULL AUTO_INCREMENT,
  `nombre` varchar(45) NOT NULL,
  `descripcion` varchar(45) DEFAULT NULL,
  PRIMARY KEY (`idTipo_Lugar`)
) ENGINE=InnoDB AUTO_INCREMENT=8 DEFAULT CHARSET=utf8;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `tipo_lugar`
--

LOCK TABLES `tipo_lugar` WRITE;
/*!40000 ALTER TABLE `tipo_lugar` DISABLE KEYS */;
INSERT INTO `tipo_lugar` VALUES (1,'LABORATORIO','Área de
Informática'), (2,'PASILLO','Zona Izquierda'), (3,'PASILLO','Zona
Central'), (4,'AULA','Normal'), (5,'FACULTAD','Edificio'), (6,'UNIVERSIDAD','Principal'), (7
,'OFICINA','Investigación');
/*!40000 ALTER TABLE `tipo_lugar` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Table structure for table `tipousuario`
--

DROP TABLE IF EXISTS `tipousuario`;
/*!40101 SET @saved_cs_client      = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `tipousuario` (
  `idTipoUsuario` int(11) NOT NULL AUTO_INCREMENT,
  `nombre` varchar(20) NOT NULL,
  PRIMARY KEY (`idTipoUsuario`)
) ENGINE=InnoDB AUTO_INCREMENT=4 DEFAULT CHARSET=utf8;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `tipousuario`
--

LOCK TABLES `tipousuario` WRITE;
/*!40000 ALTER TABLE `tipousuario` DISABLE KEYS */;
INSERT INTO `tipousuario` VALUES (3,'PROFERSOR');
/*!40000 ALTER TABLE `tipousuario` ENABLE KEYS */;
UNLOCK TABLES;

```



```

--
-- Table structure for table `usuario`
--

DROP TABLE IF EXISTS `usuario`;
/*!40101 SET @saved_cs_client      = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `usuario` (
  `idUsuario` int(11) NOT NULL AUTO_INCREMENT,
  `nombreUsuario` varchar(20) NOT NULL,
  `contraseña` varchar(20) NOT NULL,
  `idPersona` int(11) NOT NULL,
  `idTipoUsuario` int(11) NOT NULL,
  PRIMARY KEY (`idUsuario`),
  KEY `fk_usuario_personal_idx` (`idPersona`),
  KEY `fk_usuario_tipoUsuariol_idx` (`idTipoUsuario`),
  CONSTRAINT `fk_usuario_personal` FOREIGN KEY (`idPersona`) REFERENCES `persona`
(`idPersona`) ON DELETE NO ACTION ON UPDATE NO ACTION,
  CONSTRAINT `fk_usuario_tipoUsuariol` FOREIGN KEY (`idTipoUsuario`) REFERENCES
`tipousuario` (`idTipoUsuario`) ON DELETE NO ACTION ON UPDATE NO ACTION
) ENGINE=InnoDB AUTO_INCREMENT=15 DEFAULT CHARSET=utf8;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `usuario`
--

LOCK TABLES `usuario` WRITE;
/*!40000 ALTER TABLE `usuario` DISABLE KEYS */;
INSERT INTO `usuario` VALUES
(4, 'lerazo', 'root', 4, 3), (5, 'csellers', 'root', 5, 3), (6, 'jperez', 'root', 6, 3), (7, 'llopez', 'r
oot', 7, 3), (8, 'mgarcia', 'root', 8, 3), (9, 'cvezel', 'root', 9, 3), (10, 'jromero', 'root', 10, 3), (1
1, 'xburbano', 'root', 11, 3), (12, 'drodriguez', 'root', 12, 3), (13, 'rmora', 'root', 13, 3), (14, 'az
amora', 'root', 14, 3);
/*!40000 ALTER TABLE `usuario` ENABLE KEYS */;
UNLOCK TABLES;
/*!40103 SET TIME_ZONE=@OLD_TIME_ZONE */;

/*!40101 SET SQL_MODE=@OLD_SQL_MODE */;
/*!40014 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS */;
/*!40014 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS */;
/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
/*!40111 SET SQL_NOTES=@OLD_SQL_NOTES */;

```

Doctora Jenny Ríos Coello, Secretaria de la Facultad de Ciencias de la Administración de la Universidad del Azuay

**CERTIFICA:**

Que, el Consejo de Facultad en sesión del 29 de mayo de 2017, conoció la petición del estudiante **MARCO ANTONIO QUEZADA CABRERA** con código **67926**, que presenta el diseño de su trabajo de titulación denominado: **"PROTOTIPO DE APLICACIÓN MÓVIL DE LOCALIZACIÓN EN INTERIORES"**, presentado previa a la obtención del título de Ingeniero de Sistemas y Telemática.- El Consejo de Facultad acogió el informe de la Junta Académica de Ingeniería de Sistemas y Telemática y resolvió aprobar el diseño. Designa como **Director al ingeniero Chester Sellers Walden** y como miembros del Tribunal Examinador a los ingenieros Francisco Salgado Arteaga, Ph.D. y Lenin Erazo Garzón.- En esta misma sesión el Consejo de Facultad fija como plazo para la entrega del trabajo de titulación, seis meses contados desde la fecha de su aprobación, esto es hasta el **29 de noviembre de 2017**, debiendo el Director presentar a la Junta Académica, dos informes bimensuales del desarrollo del trabajo de titulación.

Cuenca, mayo 30 de 2017


Dra. Jenny Ríos Coello  
Secretaria de la Facultad de  
Ciencias de la Administración

UNIVERSIDAD DEL AZUAY  
FACULTAD DE  
ADMINISTRACION  
SECRETARIA

## CONVOCATORIA

Por disposición de la Junta Académica de **Ingeniería de Sistemas y Telemática**, se convoca a los Miembros del Tribunal Examinador, a la sustentación del Protocolo del Trabajo de Titulación: "**PROTOTIPO DE APLICACIÓN MÓVIL DE LOCALIZACIÓN EN INTERIORES**", presentado por el estudiante **Marco Antonio Quézada Cabrera**, previa a la obtención del grado de **Ingeniero en Sistemas y Telemática**, para el día **MARTES 16 DE MAYO DE 2017 A LAS 11h40.** La sustentación se realizará en el **laboratorio del IERSE.**

Cuenca, 11 de mayo de 2017



Dra. Jenny Ríos Coello  
Secretaria de la Facultad

Ing. Chester Sellers Walden

Ing. Lenin Erazo Garzón

Dr. Francisco Salgado Arteaga



mjmr/

Comunicado OK.  
Hond  
12-05-2017

Oficio Nro. 067-2017-DIST-UDA

Cuenca, 12 de mayo de 2017

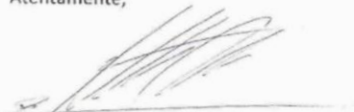
**Señor Ingeniero  
Oswaldo Merchán Manzano  
DECANO DE LA FACULTAD DE CIENCIAS DE LA ADMINISTRACIÓN  
Presente.-**

De nuestras consideraciones:

La Junta Académica de la Escuela de Ingeniería de Sistemas y Telemática, reunida el día 12 de mayo del 2017, recibió el proyecto de tesis titulado "Prototipo de aplicación móvil de localización en interiores", presentado por Marco Antonio Quezada Cabrera estudiante de la Escuela de Ingeniería de Sistemas y Telemática, y revisado por el Ing. Chester Sellers, previo a la obtención del título de Ingeniero de Sistemas y Telemática.

Por lo expuesto, y de conformidad con el Reglamento de Graduación de la Facultad, recomendamos como director y responsable de aplicar cualquier modificación al diseño del trabajo de graduación posterior al Ing. Chester Sellers y como miembros del Tribunal a Francisco Salgado Ph.D. e Ing. Lenin Erazo.

Atentamente,



Ing. Marcos Orellana Cordero  
Cordinador Escuela de Ingeniería de Sistemas y Telemática  
Universidad del Azuay



ACTA

SUSTENTACIÓN DE PROTOCOLO/DENUNCIA DEL TRABAJO DE TITULACIÓN

- 1.1 Nombre del estudiante: **Marco Antonio Quezada Cabrera**
- 1.2 Director sugerido: Ing. Chester Sellers Walden
- 1.3 Codirector (opcional):
- 1.4 Tribunal: Ing. Lenin Erazo Garzón/ Dr. Francisco Salgado Arteaga
- 1.5 Título propuesto: **"PROTOTIPO DE APLICACIÓN MÓVIL DE LOCALIZACIÓN EN INTERIORES"**
- 1.6 Resolución:

1.6.1 Aceptado sin modificaciones

1.6.2 Aceptado con las siguientes modificaciones:

---

---

---

1.6.3 Responsable de dar seguimiento a las modificaciones: Ing. Chester Sellers Walden

1.6.4 No aceptado  
• Justificación:

---

---

---

Ing. Chester Sellers Walden

Tribunal  
  
Ing. Lenin Erazo Garzón

Dr. Francisco Salgado Arteaga

Sr. Marco Antonio Quezada Cabrera

Dra. Jenny Ríos Coello  
Secretario de Facultad

Fecha de sustentación: día **MARTES 16 DE MAYO DE 2017 A LAS 11h40**



**RÚBRICA PARA LA EVALUACIÓN DEL PROTOCOLO DE TRABAJO DE TITULACIÓN**

**1.1 Nombre del estudiante: Marco Antonio Quezada Cabrera**

**1.2 Director sugerido: Ing. Chester Sellers Walden**

**1.3 Codirector (opcional):**

**1.4 Título propuesto: "PROTOTIPO DE APLICACIÓN MÓVIL DE LOCALIZACIÓN EN INTERIORES"**

**1.5 Revisores (tribunal): Ing. Lenin Erazo Garzón/ Dr. Francisco Salgado Arteaga**

**1.6 Recomendaciones generales de la revisión:**

	Cumple totalmente	Cumple parcialmente	No cumple	Observaciones (*)
<b>Línea de investigación</b>				
1. ¿El contenido se enmarca en la línea de investigación seleccionada?	✓			
<b>Título Propuesto</b>				
2. ¿Es informativo?	✓			
3. ¿Es conciso?	✓			
<b>Estado del arte</b>				
4. ¿Identifica claramente el contexto histórico, científico, global y regional del tema del trabajo?	✓			
5. ¿Describe la teoría en la que se enmarca el trabajo	✓			
6. ¿Describe los trabajos relacionados más relevantes?	✓			
7. ¿Utiliza citas bibliográficas?				
<b>Problemática y/o pregunta de investigación</b>				
8. ¿Presenta una descripción precisa y clara?	✓			
9. ¿Tiene relevancia profesional y social?	✓			
<b>Hipótesis (opcional)</b>				
10. ¿Se expresa de forma clara?	✓			
11. ¿Es factible de verificación?	✓			
<b>Objetivo general</b>				
12. ¿Concuerda con el problema formulado?	✓			
13. ¿Se encuentra redactado en tiempo verbal infinitivo?	✓			
<b>Objetivos específicos</b>				





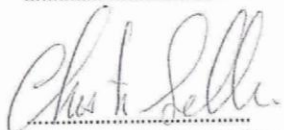
14.¿Concuerdan con el objetivo general?	✓			
15.¿Son comprobables cualitativa o cuantitativamente?	✓			
<b>Metodología</b>				
16.¿Se encuentran disponibles los datos y materiales mencionados?	✓			
17.¿Las actividades se presentan siguiendo una secuencia lógica?	✓			
18.¿Las actividades permitirán la consecución de los objetivos específicos planteados?	✓			
19.¿Los datos, materiales y actividades mencionadas son adecuados para resolver el problema formulado?	✓			
<b>Resultados esperados</b>				
20.¿Son relevantes para resolver o contribuir con el problema formulado?	✓			
21.¿Concuerdan con los objetivos específicos?	✓			
22.¿Se detalla la forma de presentación de los resultados?	✓			
23.¿Los resultados esperados son consecuencia, en todos los casos, de las actividades mencionadas?	✓			
<b>Supuestos y riesgos</b>				
24.¿Se mencionan los supuestos y riesgos más relevantes?	✓			
25.¿Es conveniente llevar a cabo el trabajo dado los supuestos y riesgos mencionados?	✓			
<b>Presupuesto</b>				
26.¿El presupuesto es razonable?				
27.¿Se consideran los rubros más relevantes?				
<b>Cronograma</b>				
28.¿Los plazos para las actividades son realistas?	✓			
<b>Referencias</b>				
29.¿Se siguen las recomendaciones de normas internacionales para citar?	✓			
<b>Expresión escrita</b>				
30.¿La redacción es clara y fácilmente comprensible?	✓			
31.¿El texto se encuentra libre de faltas ortográficas?	✓			

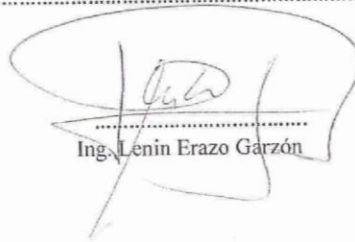
(\*) Breve justificación, explicación o recomendación.

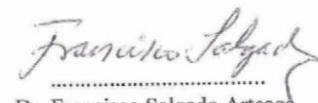


- Opcional cuando cumple totalmente,
- Obligatorio cuando cumple parcialmente y NO cumple.

.....  
.....  
.....

  
.....  
Ing. Chester Sellers Walden

  
.....  
Ing. Lenin Erazo Garzón

  
.....  
Dr. Francisco Salgado Arteaga





Escuela  
Sistemas y  
Telemática

**Oficio Estudiante: Solicitud aprobación de  
Protocolo de Trabajo de Titulación**

IST-RE-EST-02  
Versión 01  
04/04/2017  
Página 1 de 1

Lugar de Almacenamiento  
F: Archivo Secretaría de la Facultad

Retención  
5 años

Disposición Final  
Almacenar en archivo pasivo de la Facultad

Cuenca, 11 de Mayo de 2017

Ingeniero,  
Oswaldo Merchán Manzano  
**DECANO DE LA FACULTAD DE CIENCIAS DE LA ADMINISTRACIÓN**  
UNIVERSIDAD DEL AZUAY

De mi consideración,

Estimado Señor Decano, yo **Marco Antonio Quezada Cabrera** con C.I. **0104811062**, código estudiantil 67926; estudiante de la Carrera de Sistemas y Telemática, solicito muy comedidamente a usted y por su intermedio al Consejo de Facultad, la aprobación del protocolo de trabajo de titulación con el tema **"PROTOTIPO DE APLICACIÓN MÓVIL DE LOCALIZACIÓN EN INTERIORES"** previo a la obtención del título de Ingeniero en Sistemas y Telemática para lo cual adjunto la documentación respectiva.

Por la favorable acogida que brinde a la presente, anticipo mi agradecimiento.

Atentamente:

Marco Quezada

Estudiante de la Carrera de Sistemas y Telemática



Cuenca, 11 de Mayo de 2017

Ingeniero,  
Oswaldo Merchán Manzano  
**DECANO DE LA FACULTAD DE CIENCIAS DE LA ADMINISTRACIÓN**  
UNIVERSIDAD DEL AZUAY

De mi consideración,

Yo, Ing. Chester Andrew Sellers Walden informo que he revisado el protocolo de trabajo de titulación previo a la obtención del título de Ingeniero en Sistemas y Telemática, denominado "PROTOTIPO DE APLICACIÓN MÓVIL DE LOCALIZACIÓN EN INTERIORES", realizado por el estudiante Marco Antonio Quezada Cabrera, con código estudiantil 67926, protocolo que a mi criterio, cumple con los lineamientos y requerimientos establecidos por la carrera.

Por lo expuesto, me permito sugerir que sea considerado para la revisión y sustentación del mismo,

Sin otro particular, suscribo.

Atentamente

Ing. Chester Sellers



DOCTORA JENNY RIOS COELLO, SECRETARIA DE LA FACULTAD  
DE CIENCIAS DE LA ADMINISTRACION DE LA UNIVERSIDAD DEL  
AZUAY

**CERTIFICA:**

Que, el señor **QUEZADA CABRERA MARCO ANTONIO** con código **67926**, alumno  
de la escuela de **INGENIERIA DE SISTEMAS Y TELEMATICA**, tiene aprobado más  
del 80% de los créditos de su malla de estudios.

Que, al señor **QUEZADA CABRERA MARCO ANTONIO** le falta aprobar las  
siguientes asignaturas para finalizar sus estudios:

PRODUCCIÓN II  
SISTEMAS DE INFORMACIÓN GERENCIAL  
PROYECTOS TELEMÁTICOS  
CALIDAD DE SOFTWARE  
INGENIERÍA DE SOFTWARE II  
METODOLOGIA DE LA INVESTIGACION

Cuenca, 24 de marzo de 2017

Derecho No. 001-010-000116172  
mjmr



UNIVERSIDAD DEL AZUAY  
INGENIERIA EN SISTEMAS Y TELEMATICA  
DISEÑO DE TESIS

**1. DATOS GENERALES**

**1.1 Nombre del estudiante:** Quezada Cabrera Marco Antonio

**1.1.1 Código:** 67926

**1.1.2 Contacto:**

Teléfono convencional: 2827889

Celular: 0997300955

Correo electrónico: [marco78931@yahoo.com](mailto:marco78931@yahoo.com)

**1.2 Director sugerido:** Sellers Walden, Chester Ing.

**1.2.1 Contacto:**

Teléfono convencional: 2815237

Celular: 0992636061

Correo electrónico: [csellers@uazuay.edu.ec](mailto:csellers@uazuay.edu.ec)

**1.3 Co-director sugerido:** Erazo Garzón Lenin, Ing.

**1.4 Asesor metodológico:** Salgado Arteaga Francisco, PhD.

**1.5 Tribunal designado:**

**1.6 Aprobación:** Junta Académica y Consejo de Facultad.

**1.7 Línea de Investigación de la carrera:**

**1.7.1 Código UNESCO:** 1203 Informática de computadores.

1203.17 Informática.

**1.7.2 Tipo de trabajo:** Investigación aplicada.



**1.8 Área de estudio:** geolocalización, programación móvil.

**1.9 Título propuesto:** Prototipo de aplicación móvil de localización en interiores.

**1.10 Subtítulo:**

**1.11 Estado del proyecto:** El proyecto no abarca una investigación nueva, consiste en realizar una evaluación de tecnologías existentes, y en función de los resultados obtenidos llegar a generar un aplicativo propio.

## **2. CONTENIDO**

### **2.1 Motivación de la investigación:**

Los sistemas informáticos han evolucionado desde simples calculadoras científicas hasta llegar a proporcionar información en base al entorno que rodea al usuario, teniendo como referencia un perfil de usuario y la ubicación actual del mismo. De esta manera un sistema de localización en interiores es indispensable para poder generar sistemas que sean sensibles al contexto.

El poder contar con un prototipo de aplicación móvil, mediante la cual se pueda llegar a estimar la ubicación de un usuario por medio de su Smartphone, servirá como base para un futuro desarrollo, en el cual se la pueda integrar con otras aplicaciones que requieran como parte de su funcionamiento conocer la localización del usuario en ambientes interiores.

### **2.2 Problemática:**

La geolocalización es una temática que ha tenido un gran crecimiento dentro de los últimos años, con la inclusión del sistema GPS, se puede obtener la ubicación de cualquier objeto o persona en cualquier parte del mundo con una muy alta precisión. No obstante al enfocar el GPS hacia la localización en ambientes de interiores (edificios, hospitales, universidades, etc.), la precisión de

esta tecnología disminuye considerablemente, imposibilitando determinar la ubicación, dado esto es necesario tener alternativas que se enfoquen únicamente en localización de interiores, a manera de que puedan complementarse con la localización en exteriores, logrando tener un cobertura total de la ubicación de cualquier persona u objeto.

### 2.3 Pregunta de investigación:

¿Cuáles son las mejores alternativas para implementar un sistema de localización de interiores aplicado a un ambiente académico, y existen las capacidades técnicas y tecnológicas para desarrollar un prototipo de aplicación móvil para este tipo de localización?

### 2.4 Resumen:

El proyecto consiste en implementar un sistema para localizar en ambientes interiores a cualquier usuario, tomando como referente el Smartphone, que será el instrumento para poder detectar la ubicación. Así también el desarrollo de un prototipo de aplicación móvil, desde la cual se visualizara la ubicación en el plano del usuario.

### 2.5 Indagación Exploratoria:

Los sistemas que permiten realizar localización en ambientes de interiores han tenido un gran auge en los últimos años, esto debido a la necesidad de poseer una tecnología que pueda ser de gran utilidad, como lo es el sistema de posicionamiento global (GPS) para la localización en exteriores.

El GPS funciona de excelente manera para poder localizar a un objeto en ambientes de exteriores, pero no alcanza las misma precisión para una ubicación en interiores, produciendo en algunas ocasiones malas aproximaciones, con un error de alrededor de cien metros, lo cual no sirve

de ayuda para una localización dentro de edificaciones (García Vázquez, Galván Tejada, & Galván Tejada, 2013).

El surgimiento y gran incursión de los teléfonos inteligentes (Smartphone), han ayudado a potenciar el desarrollo de la localización de interiores, debido a la facilidad que estos presentan para una rápida recolección de datos, añadiendo el hecho de que actualmente llevan incorporados sensores que facilitan la interacción con el medio ambiente (Gómez Ruiz, 2015).

Existen múltiples maneras que permiten realizar una estimación sobre la ubicación de un teléfono inteligente y, a través de él, al usuario. Algunas de las técnicas son únicamente una propuesta, mientras que en otras si se ha llegado a una implementación en ambientes de prueba.

La localización mediante WIFI es utilizada en la mayoría de las veces, tomando como referente principal la recepción de la potencia de señal RSSI (Received Signal Strength Indication), debido a que este indicador permite posteriormente realizar un proceso de triangulación para hallar la ubicación de cualquier objeto. Un enfoque distinto de este método de localización consiste en tomar muestras de las mediciones RSSI, para luego armar una base de datos con esta información y con la obtención del nivel RSSI que entrega el Smartphone comparar con los datos almacenados e inferir la ubicación (Gómez Ruiz, 2015).

Una propuesta de localización en interiores empleando redes WIFI es la de (Descamps Vila, Pérez Navarro, & Conesa, 2013), donde se centran más exclusivamente en integrar la localización en interiores con una aplicación SIG (Sistemas de Información Geográfica), con la finalidad de tener una cobertura externa como interna. Para ello se hace uso de una aplicación para localización interna ya realizada.

La alternativa presentada por (Grzechca & Patryk Bielecki, 2014), se apoya de igual manera en redes WIFI para la localización en interiores, pero se complementa con la captura de





objetos por video, permitiendo conocer que objeto en específico se está localizando, lo que contribuye a elevar la localización en interiores a un nivel más alto.

Haciendo uso del campo magnético de la tierra (García Vázquez, Galván Tejada, & Galván Tejada, 2013) donde se considera el hecho de que cada pulgada cuadrada en el planeta tiene un valor de magnetismo distinto. La metodología es similar que con la tecnología WIFI; primero se recolecta datos sobre el valor de magnetismo en diversas zonas, haciendo uso del sensor de magnetismo que viene incluido en varios Smartphone, para luego pasar a una fase de localización, en la cual el usuario podrá ser localizado de acuerdo a la medición del magnetismo de su teléfono, y determinar el lugar donde se halla según los rangos conocidos.

La tecnología VCL (Visible Light Communications) ha tenido grandes avances durante los últimos tiempos, ligado a la utilización masiva de las lámparas LED, permitiendo que se genere una tendencia hacia usar la luz emitida por estas lámparas como medio de transmisión de datos, y esta es la técnica que presenta (Muñoz Arcentales, Marin Garcia, Calero Bravo, & Chavez Burbano, 2014).

El usar la luz que se emite de un LED como medio de comunicación reduce la interferencia, y podría ser una alternativa para la localización en interiores, en esta modalidad se usan lámparas como balizas o puntos referentes de localización y adicionalmente un receptor de ultrasonido, esto para conseguir enviar la señal de activación a las balizas (lámparas), y a su vez que devuelva un código para poder inferir a través de él en donde se halla el usuario (Muñoz Arcentales, Marin Garcia, Calero Bravo, & Chavez Burbano, 2014).



La técnica que ha sido implementada correctamente en distintos escenarios es mediante dispositivos bluetooth, más específicamente Bluetooth Low Energy (BLE), una versión optimizada para un menor consumo de energía. Sigue el mismo principio de la localización por WIFI, con la diferencia de que la red bluetooth debería ser implementada, ya que no está presente en la mayoría de casos; un análisis con esta tecnología es el de (Ali Hassan, 2016), donde se emplea redes neuronales para calcular la posición interna, teniendo el mismo mecanismo del RSSI como punto de partida.

Una implementación de BLE para la localización en interiores está realizada por (Li, Xu, Wang, & Muhammad, 2016), dentro de este sistema las balizas son Ibeacons, modelo de BLE desarrollado por Apple, de igual manera implementa con el RSSI, y empleando filtros para poder ofrecer una mejor precisión en la localización.

La tecnología ZigBee es una alternativa más que se ha aplicado, siendo también muy utilizado en el campo de la domótica, con la ventaja de un menor consumo energético que con WIFI y bluetooth, (Alvarez & Las Heras, 2016).

Sin embargo las tecnologías que más aceptación e implementación en este tipo de localización han tenido son mediante WIFI y bluetooth, el primero debido a una infraestructura ya establecida en la mayoría de lugares y el segundo porque esta tecnología es sencilla de usar y sigue presente en casi todos los dispositivos móviles, por ello han existido algunos análisis comparativos como el de (Zhao, Xiao, Markham, Trigoni, & Ren, 2014).

En dicho análisis se evalúan las dos técnicas, implementadas en el mismo escenario, y manteniendo los mismos puntos de referencia para ambas tecnologías, y con el RSSI como mecanismo de estimación, llegando a la conclusión de que con la tecnología bluetooth se puede



tener una mejor precisión para localización en interiores que con la red inalámbrica WIFI, (Zhao, Xiao, Markham, Trigoni, & Ren, 2014).

#### **2.6 Objetivo general:**

Desarrollar un prototipo de aplicación móvil para estimar la localización en interiores.

#### **2.7 Objetivos específicos:**

1. Realizar una investigación del estado de arte sobre las diferentes técnicas de localización en interiores.
2. Realizar un análisis comparativo de las técnicas de geolocalización en interiores a fin de seleccionar la mejor alternativa.
3. Implementar la infraestructura para la localización dentro de un escenario específico para la alternativa seleccionada.
4. Especificar los requerimientos para el prototipo de la aplicación móvil.
5. Desarrollar un prototipo de aplicación móvil para estimar la localización en interiores.

#### **2.8 Metodología:**

Se estudiará los fundamentos teóricos del tema y el estado del conocimiento tecnológico del mismo, apoyado en una búsqueda bibliográfica en diferentes fuentes: eventos relacionados con la temática, revistas, conferencias y repositorios de universidades. Se seleccionarán las fuentes

bibliográficas más sobresalientes y se procederá a crear un documento que refleje el estado del arte y un estudio comparativo sobre las técnicas de geolocalización para interiores.

Para el desarrollo del prototipo de aplicación móvil se utilizarán metodologías de desarrollo de software iterativas e incrementales. Primeramente, se elaborará un documento que contenga la especificación de los requerimientos funcionales y no funcionales del prototipo de aplicación móvil a desarrollar. Luego, se realizará el diseño arquitectónico, de procesos, de datos y de la interfaz gráfica y finalmente se realizará la codificación, instalación y pruebas del prototipo.

#### **2.9 Alcances y resultados esperados:**

Con la conclusión del proyecto se tendrá un prototipo de aplicación móvil para la localización en interiores, y durante el desarrollo del mismo se pretende cumplir con los siguientes resultados:

- Documento donde se describe el estado de arte de las técnicas y tecnologías que existen para la localización en interiores.
- Matriz comparativa donde se evalúen técnicas de localización en interiores y se concluya con la mejor alternativa a ser aplicada.
- Poseer una infraestructura que permita la localización en interiores, donde se haya implementado la técnica más adecuada.
- Documentación ERS donde se planteen los requerimientos de software para el desarrollo del prototipo de la aplicación.
- Aplicativo móvil de localización en interiores.



### 2.10 Supuestos y riesgos:

Riesgos	Probabilidad	Alternativas de solución
Falta de conocimientos en las áreas de manejo de señales sobre tecnologías de comunicación inalámbrica.	Media/Alta	Pedir asesoría a expertos en el tema.
Falta de disponibilidad de recursos tecnológicos que permitan implementar la alternativa de geolocalización en interiores seleccionada.	Alta	En el análisis comparativo de técnicas incluir un indicador del costo de implementación de la misma a fin de seleccionar la mejor alternativa según el costo.
Carencia de tiempo y recursos para realizar el análisis y evaluación de todas las técnicas mencionadas en el estado de arte.	Baja	Seleccionar únicamente las técnicas para las cuales se posee recursos y conocimientos suficientes, o realizar una evaluación únicamente teórica.
Inconvenientes para la implementación de la infraestructura de comunicación en la planta o zona designada.	Media/Baja	Solicitar el uso de otra área donde se disponga de mayor facilidad para la implementación de la infraestructura.
Falta de cumplimiento de las tareas en los tiempos propuestos.	Media	Compromiso de entrega de resultados en los tiempos fijados.
Problemas de horario con expertos que apoyan al proyecto.	Media	Buscar horarios donde sea más cómodo realizar las reuniones.

### 2.11 Esquema tentativo:

- Resumen.
- Abstract.
- Índices.
- Introducción.
- Objetivos.
- **Capítulo 1:** Estado del Arte.
- **Capítulo 2:** Análisis de requerimientos.
- **Capítulo 3:** Implementación de infraestructura para la localización en interiores.



### 2.13 Referencias:

- Ali-Hassan, A. M. (2016). Indoor Location Tracking System Using Neural Network Based on Bluetooth. *International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT)*, (págs. 73-78). Egipto.
- Alvarez, Y., & Las Heras, F. (2016). ZigBee-based Sensor Network for Indoor Location and Tracking Applications. *IEEE LATIN AMERICA TRANSACTIONS*, 14(7), 3208-3214.
- Descamps Vila, L., Pérez Navarro, A., & Conesa, J. (2013). Integración de un sistema de posicionamiento indoor en aplicaciones SIG para dispositivo móvil. *VII JORNADAS DE SIG LIBRE*, (págs. 1-12). Girona.
- García Vázquez, J. P., Galván Tejada, J. I., & Galván Tejada, C. E. (2013). Uso del campo magnético de la tierra para localizar a las personas en interiores. *ISSN 2007-3585*, 32-36.
- Gómez Ruiz, C. A. (2015). Localización en espacios interiores de dispositivos móviles por medio de tecnologías WIFI, base para el desarrollo de aplicaciones sociales. *Encuentro Internacional de Educación en Ingeniería*, (págs. 1-9). Bogotá.
- Grzechca, D., & Patryk Bielecki, T. W. (2014). Indoor location and identification of objects with video surveillance system and WiFi module. *2014 International Conference on Mathematics and Computers in Sciences and in Industry*, 171-174.
- Li, X., Xu, D., Wang, X., & Muhammad, R. (2016). Design and Implementation of Indoor Positioning System based on Ibeacon. *2016 International Conference on Audio, Language and Image Processing*, (págs. 126-130). Dubai.
- Moon, S. Y., & Kim, H. J. (2015). A Modeling Framework for Sensor Data in Indoor Location-based Services. *2015 4th International Conference on Advanced Information Technology and Sensor Application*, (págs. 67-70). Harbin.
- Muñoz Arcentales, A., Marín García, I., Calero Bravo, V., & Chavez Burbano, P. (2014). Sistema de Localización en Interiores Basado en Comunicaciones Ópticas no Guiadas por Luz Visible. Guayaquil.
- Zhao, X., Xiao, Z., Markham, A., Trigoni, N., & Ren, Y. (2014). Does BTLE measure up against WiFi? A comparison of indoor location performance. *European Wireless 2014*, (págs. 263-268). Barcelona.

### 2.14 Anexos.

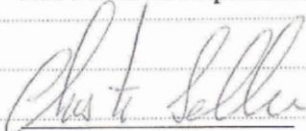


**2.15 Firma de responsabilidad (estudiante)**



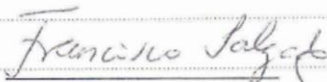
Marco Quezada

**2.16 Firma de responsabilidad (director sugerido)**



Ing. Chester Sellers

**2.17 Firma de responsabilidad (asesor metodológico)**



PhD. Francisco Salgado

**2.18 Fecha de entrega: 16 de Mayo de 2017.**