



Tesis

NAVEGACIÓN Y LOCALIZACIÓN VIRTUAL 3D EN DISPOSITIVOS MÓVILES

Para obtener el grado de Ingeniería en Computación

Presenta:

Omar Roberto Cruz Ortega

Director de Tesis:

Ing. Erik Germán Ramos Pérez

Instituto de Computación

Huajuapán de León, Oaxaca, 7 de Noviembre de 2014

Agradecimientos

En el presente trabajo de tesis me gustaría agradecer primeramente a Dios por las bendiciones que me ha dado, permitiéndome llegar hasta donde me encuentro actualmente.

A mis padres Roberto Cruz Jiménez y Oliva Ortega de Cruz por apoyarme en todo momento, por los valores inculcados, por darme la oportunidad de tener una excelente educación en el transcurso de mi vida y afrontar los retos diarios, así como a nunca rendirme ante las adversidades y seguir siempre adelante.

A mis hermanos Noé, Adrián, Alicia y Patricia por ser parte importante de mi vida, así como a mi tío Martín y mis abuelos que se adelantaron y ya no se encuentran entre nosotros, pero sin los cuales no hubiera tenido la oportunidad de ser la persona que soy hoy.

A mi director de tesis Erik Ramos y su pareja Lissete Rosales, los cuales han sido un gran apoyo en mi vida personal y profesional desde que los conozco.

A mis profesores y amigos Moisés, Hugo, Mónica, Jorge, Esperanza, Wendy, entre muchos otros con los cuales he pasado grandes momentos tanto en lo personal como en lo académico.

A mi jefe, compañeros y miembros de la División de Estudios de Posgrado de la Universidad Tecnológica de la Mixteca por el apoyo brindado todo este tiempo que me he encontrado trabajando junto a ellos.

A la Universidad Tecnológica de la Mixteca así como a sus profesores que ayudaron en mi formación académica y profesional.

Índice General

Índice de Figuras	I
Índice de Tablas.....	VI
CAPITULO I: Introducción	1
Objetivos	3
Objetivo General.....	3
Objetivos Especificos.....	3
Organización de capitulos.....	4
CAPITULO II: Marco Teórico	5
Sistemas de Localización.....	5
Características de los sistemas de localización	5
La localización física y la localización simbólica	5
Absoluto contra relativo	6
Cómputo del cálculo de localización.....	6
La exactitud y la precisión.....	6
La escala	7
El reconocimiento	7
El costo	7
Las limitaciones	7
Clasificación de los Sistemas de Localización	8
Sistemas y métodos de Localización en Ambientes Externos.....	8
Sistema y Métodos de Localización en Ambientes Internos.....	11
Ejemplos de Sistema de Localización Implementados.....	18
Técnicas de localización más usuales en los sistemas para interiores basados en radiofrecuencia	26
Ejemplos de Sistemas de Localización para interiores basados en radiofrecuencia.....	31
Utilidades y Aplicaciones de los Sistemas de Localización para Interiores.....	35
Lenguajes de Programación para Dispositivos Móviles y Embebidos	36
Perfil Bluetooth	38
APIs Bluetooth para desarrollo en C/C++.....	39
APIs Bluetooth para desarrollo en C#	40
APIs Bluetooth para desarrollo en Java	41
Implementaciones JSR-82 en J2SE.....	41
Conceptos básicos de modelado 3D	43
Espacio tridimensional.....	43
Elementos básicos del diseño 3D	43
Modelado	43
Iluminación	49
Renderizado	51

APIs de Gráficos 3D	51
APIs de Gráficos 3D para Dispositivos Móviles	51
Selección del API de gráficos 3D a usar en el modelado para el sistema de localización para ambientes internos	53
CAPITULO III: Metodología de desarrollo	55
¿Por qué el uso de una metodología de desarrollo dentro de un proyecto de desarrollo software?	55
Selección de la metodología de desarrollo a usar dentro del proyecto de tesis	55
Extreme Programming (XP)	58
CAPITULO IV: Compilación de máquina virtual J2ME para Windows Mobile 5.0	68
Fines perseguidos con la compilación de la máquina virtual	68
Fines obtenidos y no obtenidos con la compilación de la máquina virtual	68
CAPITULO V: Toma de medidas para el modelado 3D del Instituto de Electrónica y Computación (IEC)	70
Obtención de medidas del IEC	70
Modelado 3D del IEC	75
CAPITULO VI: Desarrollo de la aplicación	82
Selección de tipo de Sistema y Métodos de Localización en Ambientes Internos	82
Selección de la técnica de localización a usar en el sistema localización basado en radiofrecuencia	83
Ejemplo de localización mediante el método KNN	84
Selección del lenguaje de programación para dispositivos móviles y embebidos a usar para el desarrollo del sistema de localización para ambiente internos	85
Bluetooth	86
Por qué usar la tecnología Bluetooth y no Wi-Fi	86
Selección de la implementación JSR-82 en J2SE a usar para el desarrollo del sistema de localización para ambiente internos	88
Exploración	89
Planificación de la Entrega e Iteraciones	92
Estándar de Codificación	94
Pruebas	95
CAPITULO VII: Resultados y conclusiones	115
Infraestructura del sistema	115
Etapa de entrenamiento-recolección de datos	117
Pruebas de funcionalidad y su evaluación	118
Limitaciones del sistema	122
Conclusiones	123
Bibliografía	127

Índice de Figuras

Figura 1 Configuración satelital del sistema GPS.	8
Figura 2 Infraestructura de CELL-ID.....	9
Figura 3 Funcionamiento de CELL-ID mas sectorización.....	9
Figura 4 Sistema de localización por dirección de ángulo de llegada.....	10
Figura 5 Sistema de localización por retardo de tiempo.....	10
Figura 6 Cálculo de la distancia entre transmisor y receptor en TDOA.....	11
Figura 7 Sistema de localización basado en infrarrojo.	12
Figura 8 Ejemplo de a) Tag IR y b) Beacon IR.....	12
Figura 9 Componentes de un sistema basado en ultrasonido a) Tag ultrasónico y b) Listener ultrasónico.	13
Figura 10 Patrón de recepción del sistema basado en ultrasonido Hexamite e implementación con micrófonos WM-61 de Panasonic.....	13
Figura 11 Ejemplo de cálculo de orientación relativa en un sistema de localización basado en ultrasonido.....	13
Figura 12 Infraestructura que compone a un sistema de localización basado en ultrasonido.....	14
Figura 13 Transponder, compuesto por una antena férrea, micro ATmega88, receptor MAS9180 y transceptor CC2500.	15
Figura 14 Ejemplo de funcionamiento de sistema de localización basado en visión computacional.	16
Figura 15 Ejemplo de cálculo del ángulo de visión α de una cámara donde L es la distancia de barrido y H es la distancia entre la cámara y la superficie.....	16
Figura 16 Ejemplos de sensores de presión: a) Sensor minuatoura y b) Sensor fleixble ultradelgado.	16
Figura 17 Arquitectura general de un sistema de localización basado en radiofrecuencia.....	17
Figura 18 Arquitectura de un sistema de localización basado en radiofrecuencia y usando Tags RFID.	17
Figura 19 Satélite Navstar GPS.....	18
Figura 20 Ubicación de las estaciones de control, la estación principal y las antenas terrestres del segmento de control del sistema GPS.....	19
Figura 21 Ejemplo de localización de un receptor GPS.....	19
Figura 22 Esquema del Sistema SLVT.....	20
Figura 23 Composición de un Active Badge.	20
Figura 24 Composición de un oscilador Beacon.	21
Figura 25 Infraestructura del Sistema Active Badge.....	21

Figura 26 Ejemplo de un dispositivo Bat.....	22
Figura 27 Ejemplo de dispositivo Listener y Beacon usado por el sistema Cricket.	22
Figura 28 Ejemplo de la ubicación de Beacons del sistema Cricket dentro de un edificio.....	23
Figura 29 Ejemplo de sensor de presión del sistema Smart Floor.	24
Figura 30 Ejemplo de los 10 elementos del perfil de una muestra tomada de un usuario por el sistema Smart Floor.....	24
Figura 31 Ejemplo del uso de un Marcador Fiduciario en un cartel de MEDC 2007.....	25
Figura 32 Marcador de Marco Der) Colocado en un cartel de Microsoft's TechReady7 e Izq) Identificación y lectura del marcador por medio del sistema Singpost.	26
Figura 33 Visualización de un problema de Trilateración tridimensional con 3 puntos de referencia conocidos.	26
Figura 34 Ejemplo del cálculo de localización en 2D por medio de Lateración.	27
Figura 35 Ejemplo del cálculo de localización 3D por medio de Angulación.	28
Figura 36 Visualización de un problema de Triangulación tridimensional con 3 puntos de referencia conocidos.	28
Figura 37 Arquitectura general del sistema PinPoint 3D-iD.....	31
Figura 38 Mapa de un edificio que implementa el sistema RADAR con 3 estaciones base señales por una estrella.	32
Figura 39 Cuarto principal de conferencias del edificio FXPAL donde se implemento el sistema Nibble.	33
Figura 40 Diagrama del 2 piso del CIC donde el area de pruebas esta encerrada en un cuadro verde.	34
Figura 41 Perfiles Bluetooth.	38
Figura 42 Ejemplo de un tubo generado en Blender mediante Modelado con NURBS.....	44
Figura 43 Diseño de una perilla en Blender mediante Modelado por Estructuras Predefinidas (aplicado a los objetos UVshpere y Cylinder).....	44
Figura 44 Ejemplo de la forma final de una perilla generada en Blender mediante Modelado por Estructuras Predefinidas (aplicado a los objetos UVshpere y Cylinder).....	45
Figura 45 Diseño de una dona en Blender mediante Modelado por Extrude (aplicado a un plano tipo Circle).	45
Figura 46 Ejemplo de la forma final de una dona generada en Blender mediante Modelado por Extrude (aplicado a un plano tipo Circle).	45
Figura 47 Diseño de un arco en Blender mediante Modelado por Operaciones Booleanas (aplicando a objetos tipo Cube una resta booleana de un objeto Cylinder).....	46
Figura 48 Ejemplo de la forma final de un arco generada en Blender mediante Modelado por Operaciones Booleanas (aplicando a objetos tipo Cube una resta booleana de un objeto Cylinder).	46

Figura 49 Diseño de un abanico en Blender mediante Modelado por Malla de Polígonos (aplicado a un plano tipo Circle).	46
Figura 50 Ejemplo de la forma final de un abanico generada en Blender mediante Modelado por Malla de Polígonos (aplicado a un plano tipo Circle).....	47
Figura 51 Diseño de una L en Blender mediante Modelado por B-Rep (usando planos como representación de fronteras).	47
Figura 52 Ejemplo de la forma final de una L generada en Blender mediante Modelado por B-Rep (usando planos como representación de fronteras).	47
Figura 53 Ejemplo de un Cubo-Seccionado generada en Blender mediante Modelado con Texturas (usando objeto Cube con un material color rojo, canal Alpha activo y flexión).....	48
Figura 54 Ejemplo de un Cubo-Seccionado generada en Blender mediante Modelado con Texturas (usando objeto Cube con una textura Stucci y canal Alpha activo).....	48
Figura 55 Ejemplo en Blender de Fuente Emisora de Luz (Luz de Ambiente Global).	49
Figura 56 Ejemplo en Blender de Fuente Emisora de Luz (Luz Puntual).....	49
Figura 57 Ejemplo en Blender de Fuente Emisora de Luz (Luz Direccional).	50
Figura 58 Ejemplo en Blender de Fuente Emisora de Luz (Luz de Lugar).....	50
Figura 59 Ejemplo en Blender de Fuente Reflectora de Luz (Luz Direccional).	50
Figura 60 Modelado de un cubo con texturas y visualizacion de algunas tecnicas de los motores de render.	51
Figura 61 Reforzamiento entre sí de las 12 practica de XP.....	66
Figura 62 Ejemplo de visibilidad entre estación total y prisma (ausencia de obstáculos que bloqueen el láser).....	70
Figura 63 Marca de punto de control.	70
Figura 64 Nivelación de trípode sobre punto de control.	71
Figura 65 Instalación de equipo GPS sobre trípode.	71
Figura 66 Medición de antena GPS con respecto al piso.	71
Figura 67 Pantalla Modo.	72
Figura 68 Pantalla Medir.	72
Figura 69 Pantalla Menú Medir.....	72
Figura 70 Pantalla Atributo Punto.	72
Figura 71 Pantalla Unidad Medida.....	72
Figura 72 Ejemplo de recepción de señal de los satélites accesibles al dispositivo GPS. Izq) Configuración de satélites con menor error de ubicación, Der) Configuración de satélites con mayor error de ubicación.	72
Figura 73 Instalación de equipo de la estación total sobre trípode.	73
Figura 74 Marca de referencia de la estación total para medir su altura.....	73
Figura 75 Conexión de la estación total a la batería.	73
Figura 76 Medición con prisma mediante el uso de menú "Direct Reflez".	73

Figura 77 Ejemplo de toma de medidas dentro del IEC usando la estación total.....	74
Figura 78 Nivelación de trípode de primas.....	74
Figura 79 Ajuste de la altura del primas.....	74
Figura 80 Mirilla y perillas de ajuste vertical y horizontal de la estación total.....	74
Figura 81 Medidas obtenidas de un cubículo del IEC.	75
Figura 82 Ejemplo de renderizado M3G con Immediate Mode en celular Motorola a1200.....	76
Figura 83 Ejemplo de renderizado M3G con Retained Mode en celular Motorola a1200.....	76
Figura 84 Ejemplo de renderizado M3G con Immediate Mode en emulador.....	77
Figura 85 Ejemplo de renderizado M3G con Retained Mode en emulador.....	77
Figura 86 Modelo final de la estructura interna del IEC (pasillos).	78
Figura 87 Visión de las líneas que forman la estructura del modelo de un cubículo del IEC.....	79
Figura 88 Visión de las líneas y vértices que forman la estructura del modelo de un cubículo del IEC.	79
Figura 89 Renderizado del modelo del cubículo del IEC ya con la aplicación de las texturas y efectos necesarios.	80
Figura 90 Renderizado del modelo completo del IEC.	80
Figura 91 Captura del modelo del cubículo del IEC ya con la aplicación de las texturas y efectos necesarios a) desde el Emulador Wireless Toolkit 2.5.2 y b) desde Celular Motorola a1200.....	81
Figura 92 Imagen de un Smartphone Motorola Symbol MC 35.	82
Figura 93 Imagen de un Smartphone Motorola A1200r.....	83
Figura 94 Visión general de las tecnologías usadas por los sistemas de localización.....	88
Figura 95 Cronograma de actividades del proyecto de Tesis.	91
Figura 96 Sección del cronograma de actividades correspondiente a la fase de Exploración de la primera versión del sistema.....	92
Figura 97 Pantalla de la aplicación con las pruebas unitarias correspondientes a la aplicación para la localización del dispositivo móvil.....	97
Figura 98 Pantalla de las pruebas unitarias correspondientes a la clase Bluetooth.	98
Figura 99 Pantalla de las pruebas unitarias correspondientes a la clase CamaraMovil.	99
Figura 100 Pantalla de las pruebas unitarias correspondientes a la clase CargarCanvas.....	100
Figura 101 Pantalla de las pruebas unitarias correspondientes a la clase ModeloM3G.....	101
Figura 102 Pantalla de la prueba unitaria correspondiente a la clase PantallaInicio.	101
Figura 103 Diagrama a bloques del funcionamiento de las aplicaciones para la Terminal Móvil y Fija del sistema.....	116
Figura 104 Imágenes correspondientes a la pantalla de la terminal móvil que muestra: a) El nombre del sistema b) La presentación del funcionamiento de la aplicación para dispositivos móviles que usa el sistema, c) Petición de permiso al usuario para que la terminal móvil realice conexión con otro dispositivo vía Bluetooth y d) Vista de la posición actual de la terminal móvil a través de su indicador predefinido, dentro de edificio representado por el modelo 3D precargado del mismo.	119

Figura 105 Image que muestra la ubicación de la terminal móvil: Derecha) dentro del edificio e Izquierda) dentro del modelo 3D.	120
Figura 106 Cambio de modalidad de la aplicación para la terminal móvil que permite navegar con la cámara dentro de un modelo 3D con transparencias del edificio, sin alterar los resultados de la localización (absoluta y relativa) del propio dispositivo.	120
Figura 107 Imagen correspondiente a la pantalla de la aplicación de la terminal móvil en la cual se muestra la información adicional con que se cuenta al momento que el sistema realiza la localización del propio dispositivo.....	121

Índice de Tablas

Tabla 1 Comparativa de implementaciones JSR-82 para J2SE	42
Tabla 2 Comparativa de las APIs de gráficos 3D disponibles para el lenguaje J2ME.	53
Tabla 3 Resultados de las pruebas de SPMark04 sobre los teléfonos Nokia N-Gage, 6600 y 6630.	54
Tabla 4 Resultados de las pruebas de JBenchmark3D sobre los teléfonos Nokia N-Gage, 6600 y 6630.	54
Tabla 5 Comparativa entre herramientas de diseño 3D para la creación de archivos binarios M3G.	54
Tabla 6 Diferencias entre Metodologías Tradicionales y. Metodologías Ágiles.	57
Tabla 7 Comparación hecha entre los modos de diseño del API JSR-184.	76
Tabla 8 Comparativa entre lenguajes de programación para dispositivos móviles y embebidos....	86
Tabla 9 Comparativa de características de la tecnología Bluetooth y Wi-Fi.....	86
Tabla 10 Clasificación de resultados observables de la Tarea 1.	102
Tabla 11 Caminos posibles de ejecución de la Tarea 1.	102
Tabla 12 Valores posibles para los posibles caminos de ejecución de la Tarea 1.	103
Tabla 13 Clasificación de resultados observables de la Tarea 2.	103
Tabla 14 Nueva Clasificación de resultados observables de la Tarea 2.	104
Tabla 15 Caminos posibles de ejecución de la Tarea 2.	104
Tabla 16 Valores posibles para los posibles caminos de ejecución de la Tarea 2.	104
Tabla 17 Nuevos caminos posibles de ejecución de la Tarea 2.	105
Tabla 18 Nuevos valores posibles para los posibles caminos de ejecución de la Tarea 2.	105
Tabla 19 Clasificación de resultados observables de la Tarea 4.	105
Tabla 20 Valores posibles para los posibles caminos de ejecución de la Tarea 3.	106
Tabla 21 Valores posibles para los posibles caminos de ejecución de la Tarea 3.	107
Tabla 22 Clasificación de resultados observables de la Tarea 4.	107
Tabla 23 Clasificación de resultados observables de la Tarea 4.	108
Tabla 24 Valores posibles para los posibles caminos de ejecución de la Tarea 4.	108
Tabla 25 Clasificación de resultados observables de la Tarea 5.	109
Tabla 26 Caminos posibles de ejecución de la Tarea 5.	109
Tabla 27 Valores posibles para los posibles caminos de ejecución de la Tarea 5.	109
Tabla 28 Clasificación de resultados observables de la Tarea 6.	110
Tabla 29 Caminos posibles de ejecución de la Tarea 6.	110
Tabla 30 Valores posibles para los posibles caminos de ejecución de la Tarea 6.	111
Tabla 31 Nuevos caminos posibles de ejecución de la Tarea 6.	111
Tabla 32 Nuevos valores posibles para los posibles caminos de ejecución de la Tarea 6.	111
Tabla 33 Clasificación de resultados observables de la Tarea 7.	112

Tabla 34 Valores posibles para los posibles caminos de ejecución de la Tarea 7.	113
Tabla 35 Valores posibles para los posibles caminos de ejecución de la Tarea 7	114
Tabla 36 Tiempos necesarios para recolectar muestras de RSSI, promediar las muestras y enviar el valor promedio obtenido desde cada módulo Bluetooth conectado a una terminal fija.....	117

CAPITULO I: Introducción

Los sistemas de localización y ubicación permiten localizar (buscar un objeto o entidad específica) y ubicar (desplegar de forma visual o por coordenadas dónde se encuentra cierto objeto o entidad específica) a determinados dispositivos electrónicos de forma absoluta o relativa por medio de diferentes tipos de tecnologías (GPS, Bluetooth, Wi-Fi, etc.) y señales (radio, ultrasonidos, acústica, etc.), ya sea en ambientes externos (ciudades, bosques, montañas, etc.) o internos (cuartos, laboratorios, edificios, etc.). **“Los sistemas de localización son uno de los campos más prometedores en la computación móvil. Estos sistemas permiten desarrollar nuevas aplicaciones basadas en el contexto.”** (Rodríguez, Escudero y Pece)

Actualmente los sistemas de localización y ubicación tienen un gran impacto dentro de diferentes áreas, las cuales van desde su uso para un simple proyecto escolar de rastreo en un determinado espacio, hasta el uso de los mismos por empresas y/o gobiernos con el fin de obtener la localización de un objeto o persona para que, en base a la información obtenida, se recupere, guíe o notifique al mismo sobre cierta información que éste requiera.

Un sistema de localización y ubicación debe tomar en cuenta y cumplir de alguna forma con las siguientes características:

- **La localización física y la simbólica:** Representa la forma en que el sistema entrega la información referente a la localización de una entidad u objeto.
- **Absoluto contra relativo:** Indica el tipo de marco de referencia usado por el sistema.
- **Cómputo del cálculo de localización:** Forma en la cual el sistema se apoyará para realizar los cálculos computacionales de la información referente a la localización de la entidad u objeto.
- **La exactitud y la precisión:** Información referente a la variación existente entre la localización real y la calculada de una entidad u objeto, así como con qué frecuencia coinciden ambos valores.
- **La escala:** Indica el rango de cobertura del sistema.
- **El reconocimiento:** Capacidad de un sistema para identificar de forma única a las entidades u objetos a localizar y ubicar, con el fin de poder realizar una serie de acciones o eventos de acuerdo a la identificación realizada.
- **Limitaciones:** Indica las características bajo las cuales un sistema funcionará o no de forma óptima.

Los sistemas de localización se pueden clasificar principalmente dentro de dos grupos: Sistemas de Localización en Ambientes Internos y Sistemas de Localización en Ambientes Externos. Dentro de los mismos se pueden tener otras sub-clasificaciones, las cuales están de acuerdo a los tipos de tecnologías, señales y/o métodos que usan para la localización y ubicación.

Dentro de los sistemas de localización para ambientes externos el que mayor auge, desarrollo, soporte y uso ha tenido en todo el mundo es el GPS, a tal grado que actualmente muchos dispositivos electrónicos incluyen la tecnología que lo soporta para poder usarlo en una infinidad de aplicaciones entre las cuales se pueden encontrar principalmente aquellas para empresas de seguridad, constructoras para el rastreo de vehículos y toma de mediciones respectivamente.

“La problemática de la localización en interiores ha sido objeto de un intenso estudio e investigación durante los últimos años. Hasta ahora, ninguna de las soluciones propuestas ha conseguido el éxito que han alcanzado los sistemas de localización y navegación análogos empleados en exteriores, sobre todo el GPS.” (Sánchez Vitores)

CAPITULO I: Introducción

Aunque actualmente existen una variedad de propuestas de sistemas de localización para ambientes internos, ninguno ha dado el resultado esperado. Algunos de estos sistemas son mejoras de otros, sin embargo la mayor parte son independientes unos de otros y se han desarrollado usando diferentes tipos de tecnologías, señales y métodos, al igual que lenguajes de programación y plataformas que los soportan.

Gracias a las mejoras e incremento de las capacidades computacionales de los dispositivos móviles (PDA, Laptops, Celulares, etc.) y las tecnologías con las que cuentan los mismos (Bluetooth, Wi-Fi, GSM, etc.) en la actualidad, el desarrollo de sistemas de localización para interiores se ha inclinado por los sistemas basados en radiofrecuencia. Esto se debe a que es posible el desarrollo de los mismos con muchos de los actuales dispositivos móviles con que cuenta las personas, lo cual conlleva a una reducción en la inversión requerida por este tipo de sistemas ya que posee una gran escalabilidad y no es necesario el uso de equipo e infraestructura específica por parte del mismo.

La presente tesis persigue como objetivo principal desarrollar un sistema de localización para interiores 3D basado en la tecnología Bluetooth usando los lenguajes J2SE y J2ME de Oracle. Para ello los dispositivos (Terminales Móviles) a los que dará soporte la infraestructura del sistema deben contar con soporte para el lenguaje J2ME de Sun, así como los APIs de Java JSR-82 (Comunicación Bluetooth) y JSR-184 (Modelo 3D con M3G). La infraestructura estará compuesta por PCs (Terminales Fijas) que tengan soporte para el lenguaje J2SE de Sun y la API AvetanaBT implementación de JSR-82 para J2SE.

La decisión de usar la tecnología Bluetooth se basa en el bajo costo, rango de cobertura y gran disponibilidad en diferentes dispositivos móviles de la misma, de igual forma se optó por el lenguaje J2SE y J2ME de Oracle debido a que es OpenSource para desarrollos (no requiere el pago de una licencia para desarrollo), al elevado número de desarrolladores de software que lo usan y la gran disponibilidad en la mayor parte de las plataformas para dispositivos móviles del mercado existentes en el momento que se inició el desarrollo del sistema propuesto.

Aun cuando el lenguaje J2ME de Oracle sigue vigente y con mejoras dentro del mismo como se observa en ORACLE JAVA ME SDK 8¹, el cual incorpora soporte para el Connected Limited Device Configuration Ver. 8 (CLDC-1.8) y el ME Embedded Profile Ver. 8 (MEEP-8.0), se encuentra actualmente en desuso en dispositivos móviles como teléfonos inteligentes y tabletas, tendiendo más su utilización en dispositivos embebidos como las de placa computadora (SBC por sus siglas en inglés) e internet de las cosas (IoE por sus siglas en inglés).

Es por ello que una alternativa viable es la migración de la aplicación para las terminales móviles al sistema operativo y lenguaje de programación Android, el cual, es uno de los más usados actualmente en lo que se refiere a dispositivos móviles², por lo tanto la migración entre lenguajes no requiere de un gran esfuerzo ya que J2ME es una versión reducida de J2SE, mientras que el SDK de Android se basa en el lenguaje Java³. Esto puede observarse con MicroEmu, una implementación pura de Java para J2ME⁴ que permite portar aplicaciones J2ME básicas hacia Android; al igual que los emuladores de J2ME para Android como son Netmite App Runner⁵, JBED⁶, JBlend⁷ o phoneME⁸.

Dos de los requisitos especiales para el funcionamiento de la aplicación para las terminales móviles son el soporte para la comunicación Bluetooth y modelos 3D. Como se mencionó previamente, en J2ME para el manejo de estos requisitos existen las APIs de Java JSR-82 y JSR-184

¹ <http://docs.oracle.com/javame/8.0/index.html>

² <http://gs.statcounter.com/#mobile+tablet+console-os-ww-monthly-201309-201409>

³ <http://androidos.readthedocs.org/en/latest/data/caracteristicas/>

⁴ <https://code.google.com/p/microemu/>

⁵ <http://www.netmite.com/>

⁶ <http://jbed.malavida.com/>

⁷ http://www.unet.me/android-app/JBlend-android_13203

⁸ <http://davy.preuveneers.be/phoneme/>

CAPITULO I: Introducción

respectivamente, por otro lado aunque Android no cuenta con soporte para estas APIs existen dos alternativas viables: el API Bluetooth propio de Android⁹ y el motor 3D jPCT-AE¹⁰.

En lo referente a la API Bluetooth de Android, a pesar de basarse en Java no implementa el estándar JSR-82 al 100%, la forma de programación y funcionamiento son similares. Tomando como base la similitud que comparten las APIs es posible realizar una comparativa de ambas ^(Vazquez-Briseno), implementar una misma aplicación tanto en J2ME como en Android¹¹ o bien implementar un sistema de comunicación entre las dos APIs ^(Neerunjun, Bhiwoo y Nagowah).

Por otra parte el motor 3D jPCT-AE aun cuando no se deriva del estándar JSR-184 de Java comparte ciertas similitudes como se observa en ^(Alvarez Toro y De la Hoz Sánchez) y ^(Bargo Rodríguez). El motor 3D jPCT-AE es una implementación de alto nivel de gráficos 3D basados en OpenGL-ES, lo cual es equivalente al modelo para gráficos de alto nivel del JSR-184 denominado Retained Mode. En ambos casos existe la capacidad para cargar un modelo encapsulado dentro de un archivo binario. Además al estar basado jPCT-AE en OpenGL-ES permite crear soluciones basadas solamente en código funcionamiento similar al provisto por el modelo denominado Immediate Mode que incorpora el JSR-184.

Objetivos

Objetivo General

Desarrollar un sistema de navegación y localización 3D para dispositivos móviles (Smartphones, PDAs, Laptops, etc) en ambientes internos, basado en la tecnología Bluetooth. Se pretende generar un sistema portable y facil de adaptar a cualquier espacio cerrado, mediante la modificación únicamente del modelo 3D del mismo, la lista de MACs de los modulos Bluetooth y el radio mapa patrón correspondienet al espacio donde se implementara el sistema.

Objetivos Especificos

- Estudiar las características, capacidades y funcionalidades que proporciona la tecnología Bluetooth, así como el estándar IEEE 802.15.1 que presentan las redes WPAN basadas en esta tecnología.
- Estudiar las características, capacidades y funcionalidades que proporcionan las APIs de Gráficos 3D para el desarrollo de aplicaciones móviles.
- Estudiar la factibilidad de trabajar la comunicación Bluetooth por medio del lenguaje de programación J2SE para PC y J2ME para dispositivos móviles, además del modelado 3D para este último; considerando que se trata de lenguajes bajo la licencia GNU GPL.
- Estudiar las posibles técnicas de localización para interiores existentes y su uso o adaptación de acuerdo con el hardware y software con el que se cuenta.
- Simular de manera tridimensional el Instituto de Electrónica y Computación (IEC) de la Universidad Tecnológica de la Mixteca.
- Desarrollar un sistema de localización para ambientes internos portable en su uso para espacios cerrados. Para ello el sistema será capaz de adaptarse a diferentes espacios cerrados mediante la modificación del modelo 3D y la base de datos del radio mapa patrón del mismo, para que estos corresponderán a cada espacio cerrado donde se desee implementar el sistema de localización.

⁹ <http://developer.android.com/guide/topics/connectivity/bluetooth.html>

¹⁰ <http://www.jpct.net/jpct-ae/>

¹¹ http://wiki.mikrokoetter.de/en/DUBwise#via_Bluetooth

Organización de capítulos

El Capítulo I aborda una introducción sobre la temática en la que se enfoca la tesis: el desarrollo de un sistema de localización para interiores, por lo que se da una breve explicación de los sistemas de localización y ubicación, las características que deben cumplir, así como su clasificación. De igual forma se da una breve explicación de estado actual de los sistema de localización para interiores, el objetivo principal y los objetivos específicos que persigue la tesis, las tecnologías y leguajes seleccionados para el desarrollo de los objetivos perseguidos, así como el porqué de los mismos.

El Capítulo II da una explicación del estado del arte en el que se apoya la tesis para cumplir con los objetivos de la misma. Dentro de la información que se incluye en ese capítulo se menciona con mayor amplitud los sistemas de localización, su clasificación, ejemplos de los mismos, así como el porqué de desarrollar un sistema de localización para interiores basado en radiofrecuencia y que se apoya en el método de reconocimiento de patrones KNN. Se da una explicación de los lenguajes de programación para dispositivos móviles así como las plataformas más usadas que los soportan, explicando de igual forma la razón por la cual se optó por el uso de J2SE y J2ME para el desarrollo de la tesis. También se mencionará la tecnología Bluetooth, su historia, estado actual de la misma, el porqué de tal selección, así como las opciones para su programación bajo el lenguaje seleccionado para la tesis. Por último se da una explicación de Gráficos 3D, conceptos básicos, opciones disponibles para la programación de éstos de acuerdo al lenguaje seleccionado para la tesis así como el motivo por el cual se optó por el uso de modelado 3D basado en M3G.

El Capítulo III aborda la problemática de la metodología de desarrollo de software que se usa para el desarrollo de la tesis, la cual se enfoca en las metodologías ágiles, haciendo enfoque en XP y el esquema de trabajo que se debe cumplir para el uso de la misma en los desarrollos de software.

El Capítulo IV detalla por qué se decidió tratar de migrar la aplicación de las terminales clientes a una plataforma que no incluye como lenguaje nativo J2ME, cómo se trató de conseguir esto mediante la compilación de una Máquina Virtual de J2ME para la plataforma Windows Mobile 5.0, los pasos para compilar la misma, así como la incorporación de la API Bluecove para permitir el uso de JSR-82 dentro de esta plataforma, la causa de que no se lograra este objetivo y el aporte que se obtuvo de forma no intencional al lograr la compilación de la Máquina Virtual.

El Capítulo V explica la forma en que obtuvieron las mediciones del IC, esto con el fin de poder realizar un modelo 3D con mediciones lo más reales posibles, el cual se usa como modelo de pruebas para la tesis, de igual forma se da una explicación de cómo se hizo uso de las mediciones, así como las técnicas de modelo empleadas para recrear el modelo por medio de la herramienta de diseño 3D denominada Blender, el uso de la misma para exportarlo a un archivo binario en formato M3G compatible con el API JSR-184 por medio de un plugin de desarrollo independiente.

El Capítulo VI da una explicación de la forma en que se usó la metodología de desarrollo ágil XP dentro de la tesis, los estándares, tiempos y prácticas que se siguieron durante el desarrollo del software.

El Capítulo VII muestra la información que se obtuvo una vez concluido el desarrollo del objetivo perseguido por el trabajo de tesis, es decir, la creación de un Sistema de Localización para Interiores, para lo cual se llevó a cabo la implementación del mismo dentro del IC para posteriormente realizar una serie de pruebas del mismo, con lo cual se obtuvo información de suma importancia para la mejora y posibles modificaciones en trabajos futuros del Sistema propuesto en la tesis.

El Capítulo VIII en base a la información obtenida con el desarrollo del Sistema y puesta en marcha del mismo en un ambiente real, da algunas ideas de posibles trabajos futuros que incluyan, mejoren modifiquen o reestructuren el sistema de forma total o parcial creado como parte de la tesis.

CAPITULO II: Marco Teórico

Sistemas de Localización

En general se puede describir a un sistema de localización como un sistema formado por una infraestructura de nodos (PCs, Dispositivos Móviles, HW especializado, etc.), en la cual algunos de ellos (por lo general móviles) emiten un tipo de señal (radio, ultrasónica, acústica, etc.) al medio y son recibidas por otros nodos en específico (generalmente fijos), después de que la señal se propaga a cierta distancia por el medio. Una vez que los nodos captan la señal se realiza una serie de mediciones de algún parámetro (rangos, ángulos, potencia de la señal, etc.) en particular y a partir de los cuales se estima la posición geográfica (aproximada) de ciertos nodos con apoyo de algunos valores conocidos y toman como referencia generalmente a los nodos específicos que se encargan de recibir las señales de los nodos a los cuales se desea localizar. Posteriormente esta información es enviada ya sea al nodo al cual se calculó su posición o a los nodos que apoyaron el cálculo de la posición, para que alguno de estos realice alguna tarea o evento en específico.

Estos sistema de localización pueden ser capaces de ubicar objetos o entidades de forma global (mundial), dentro de un área metropolitana, en un campus, y demás., esto depende en particular del tipo de ambiente donde se desea implementar el sistema (ambientes internos o externos) y de la tecnología a usar (GPS, GSM, Bluetooth, Wi-Fi, etc.).

Actualmente los sistemas de localización tienen una infinidad de aplicaciones dentro de las cuales se pueden destacar: seguridad, control, consulta y mejora de servicios en general.

Características de los sistemas de localización

La localización física y la localización simbólica

Es de gran importancia considerar en los sistemas de localización una característica común a todos ellos la cual es el grado de granularidad¹² con respecto a la información de la localización realizada por el sistema. “Un sistema de localización puede proporcionar dos clases de información: física y simbólica. (Borraz Gómez)” Ambas clases de información representan los dos extremos del grado de granularidad de los sistemas de localización, la información física representa una granularidad fina debido a que la información trata de ser lo más precisa y real posible; por otro lado la información simbólica representa una granularidad gruesa ya que el grado de precisión y realismo de la información es muy ambigua.

La localización física proporciona posiciones físicas, es decir, coordenadas de latitud, longitud y/o altura. Por ejemplo los GPS permiten dar una localización física de un punto específico, de la forma: 47,39° 17" N – 122,18° 23" O, a una elevación de 20.5 metros.

En contraste, la localización simbólica se basa en ideas abstractas acerca de dónde se encuentra algo, es decir, da una posición basando su referencia con respecto a cierto objeto o entidad. Por ejemplo, una aplicación puede proporcionar una localización simbólica de un punto específico donde se sitúe el dispositivo que ejecuta una aplicación de la siguiente forma: Te encuentras ubicado en el cubículo 5, estás cerca del servidor 4, estás aproximándote a la antena 3, etc.

De igual forma, es posible usar la localización física para determinar localizaciones simbólicas.

¹² Es la especificidad con respecto al nivel de detalle de la información proporcionada por un sistema.

Absoluto contra relativo

Los sistemas de localización deben de tomar especial importancia a las coordenadas de localización para poder hacer un uso adecuado y correcto de ellas, y así obtener una mayor exactitud al cálculo de localización de un objeto o entidad, por ello es de gran importancia seleccionar el marco de referencia más adecuado para el sistema de localización, es posible definir dos tipos de sistema de localización de acuerdo al marco de referencia usado: absoluto y relativo.

Un sistema de localización absoluto utiliza un marco de referencia que comparte con todas las partes interesadas del sistema (objetos localizados). Por ejemplo, el sistema GPS, sin importar el dispositivo usado, tiene un marco de referencia común el cual está compuesto por coordenadas en x,y,z (latitud, longitud y altitud).

En un sistema de localización relativo, cada objeto o entidad puede tener su propio marco de referencia. Por ejemplo un sistema para localización y ubicación en interiores basado en uso de cámaras de celulares y marcadores de seguimiento¹³ (Mulloni, Barakonyi y Wagner) tiene diferentes marcos de referencia de acuerdo a los marcadores que sean usados y el lugar a donde se puede dirigir dentro de un edificio.

Cómputo del cálculo de localización

De acuerdo con las necesidades y capacidades con que se cuenta para desarrollar e implementar un sistema de localización, es necesario tomar en cuenta de qué forma y dónde se realiza el cálculo de la posición del objeto o entidad que el sistema deberá localizar. Por lo cual un sistema de localización puede realizar el cómputo del cálculo de localización de dos formas: privado (también denominado método centralizado) o público (también denominado método distribuido).

Se dice que el cómputo del cálculo de localización es privado cuando los sistemas que proporcionan capacidad de localización insisten en que el objeto o entidad que se está localizando calcule su propia posición, con lo cual se asegura la privacidad, garantizando que ninguna otra entidad pueda saber dónde está el objeto, a no ser que éste específicamente tome la decisión de divulgar esa información y que tanto la infraestructura como el sistema lo soporte, un ejemplo de este tipo es el uso por él GPS.

Por otro lado el cómputo del cálculo de localización es público cuando los sistemas que proporcionan capacidad de localización son los que computan la posición del objeto que se está localizado teniendo como principal desventaja que otros objetos que interactúen con el sistema tengan acceso a la información sobre la localización de este objeto, un ejemplo de este tipo es el uso por el CELL-ID. Sin embargo, este tipo de cómputo de cálculo de localización presenta la ventaja de que disminuye la carga de trabajo en lo referente al cómputo interno que realiza el propio objeto para su localización.

La exactitud y la precisión

Para conocer qué tan útil y exitoso es un sistema de localización existen dos puntos de gran importancia que siempre deben considerars: la exactitud y presión que proporciona un sistema. A pesar de que es posible hacer una estimación de estos, la mejor forma de conocer sus valores reales es mediante la puesta en práctica del sistema en un ambiente real.

“Un sistema de localización debe informar sobre localizaciones con exactitud y consistencia de medida a medida.” (Borraz Gómez) La variación (error en metros) existente entre la posición real y la calculada por el sistema denotan la exactitud. Los porcentajes denotan la precisión, es decir, cuán a menudo se puede esperar conseguir exactitud entre la posición real y la posición calculada. Por ejemplo, los receptores GPS económicos pueden ubicar posiciones a menos de 10 metros para

¹³ Patrones 2D que incluyen dentro del mismo un código de barras con ID único que informan sobre la posición del mismo, posibles caminos a seguir u otro tipo de información que el programador desee agregar.

CAPITULO II: Marco Teórico

aproximadamente el 95% de las mediciones, las unidades más costosas alcanzan exactitudes de 1 a 3 metros en un 99 %.

La escala

Un sistema de localización es capaz de proveer un rango de cobertura tanto en tiempo, espacio y número de elementos a soportar de acuerdo con el tipo de señal y tecnología usada por el mismo, el cual en ciertos casos puede ser mixto; pero además se ve afectado de forma negativa por la sensibilidad de estas, a ciertos factores.

Para determinar la escala de un sistema de localización, se debe considerar su área de cobertura por unidad de infraestructura¹⁴, y el número de objetos que el sistema puede ubicar de igual forma por unidad de infraestructura en un determinado intervalo de tiempo. El tiempo es un factor importante a considerar, ya que el ancho de banda está limitado para la detección de objetivos.

El reconocimiento

Un sistema de localización, como su propio nombre lo dice, se encarga de ubicar un objeto o entidad dentro de un ambiente, para realizar una determinada acción o evento de acuerdo al mismo, por lo cual es de gran importancia poder identificar al mismo. Algunas aplicaciones reconocen o clasifican objetos o entidades, para posteriormente ejecutar una acción específica en base a la localización de estos, para ello es necesario un mecanismo de identificación automático. Generalmente una técnica usada para el suministro de la capacidad de reconocimiento es la asignación de nombres o identificadores globalmente únicos (GUID) para los objetos que el sistema localiza, los cuales pueden ser usados para obtener información como: nombre de usuario, privilegios que posee dentro de la infraestructura, entre otros datos; esto mediante el uso de una base de datos que deberá poseer la infraestructura que soporta a la aplicación.

El costo

Implícitamente el desarrollo e implementación de un sistema conlleva una serie de costos, el cual abarcará varios puntos de importancia para el correcto funcionamiento y uso del mismo de acuerdo a las necesidades que se desean cubrir con el uso del sistema.

Se puede evaluar el costo de un sistema de localización en función de diferentes factores, algunos de los costos más importantes a considerar son los siguientes:

- **Costos de tiempo:** son los que incluyen factores como el tiempo de instalación y administración del sistema.
- **Costos de espacio:** los cuales involucran la cantidad de infraestructura necesaria a instalar y el tamaño del hardware.
- **Costos de capital:** incluyen factores como el precio de infraestructura (terminal o unidad móvil) y los salarios establecidos para el personal de soporte.

Las limitaciones

Como se mencionó anteriormente, un sistema de localización debe de considerar bajo qué limitaciones es posible el uso correcto del mismo, debido a que esto implica una disminución o no del rango de cobertura de éste. De acuerdo a la o las señales y tecnologías ocupadas por el sistema éste se verá afectado por las mismas y cualquier persona o empresa que desea usar este sistema debe de conocer de antemano las limitaciones del mismo, para así saber si es útil o no para su propósito.

Algunos sistemas no funcionarán en ciertos ambientes o bajo ciertas condiciones. Una desventaja con el GPS es que los receptores por lo general no pueden detectar u obtener las transmisiones de

¹⁴ Elementos o equipo fijo en el cual se soporta toda la infraestructura del sistema a usar, tales como Servidores, Antenas, Routers, etc.

los satélites en interiores (dentro de estructuras cerradas como edificios). En ocasiones una de las problemáticas que se presenta es que los sistemas de localización usan la misma frecuencia de operación que otros sistemas, por lo cual experimentan interferencia, por ejemplo la tecnología Wi-Fi y Bluetooth usan el mismo rango de frecuencia y por lo cual puede existir una interferencia entre ambas tecnologías. En general, se analizan las limitaciones funcionales del sistema de localización, en función de las características de las tecnologías usadas en la implementación del mismo.

Clasificación de los Sistemas de Localización

Los sistemas y métodos de localización en general pueden clasificarse de dos formas: sistemas y métodos de localización para ambientes externos (sistemas de largo alcance) y sistemas y método de localización para ambientes internos (sistemas de corto alcance).

La principal razón por la cual los sistemas y métodos de localización se dividen en estos dos grupos se debe a la falta de capacidad y funcionalidad de los sistemas y métodos de localización para exteriores, en lo referente al uso de estos mismos en interiores ya que fueron los primeros en surgir. “La investigación destinada a la localización en interiores se origina tras el fracaso de los sistemas GPS de localizar dentro de un edificio de manera adecuada. (Ambrona Tabernilla)” Esto se debe a que el GPS se considera el más grande logro y el mejor sistema de localización para ambientes externos, debido a que posee la mayor exactitud y precisión con respecto a otros sistemas del mismo género.

Sistemas y métodos de Localización en Ambientes Externos

Sistema de Posicionamiento Global (GPS, *Global Position System*): Es un sistema de navegación mundial basado en satélites, consta de 24 satélites ubicados en 6 orbitas planas, con 4 satélites en cada orbita con una distancia igual entre cada uno (Figura 1). Los satélites están ubicados a una altura de 20,200 kilómetros sobre la tierra, estos transmiten señales portadoras de código, unas para uso civil y otras para uso militar-gobierno. Los receptores GPS procesan la señal para determinar la posición en 3D (latitud, longitud y altitud), con una exactitud de 10 metros o menos (dependiendo del receptor). Sin embargo, para su buen funcionamiento se necesita una línea de recepción de los satélites libre (no funciona bien dentro de edificios) y recepción de señal de por lo menos 3 o 4 satélites, a medida que se incrementa el número de satélites encontrados será mejor la estimación en cuanto a localización del receptor GPS.

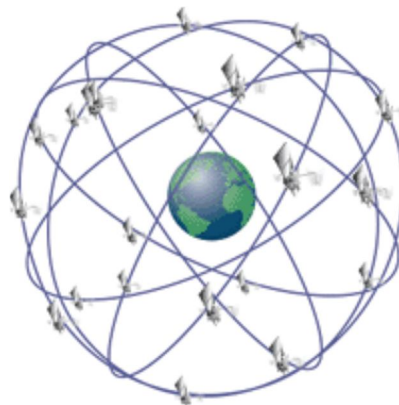


Figura 1 Configuración satelital del sistema GPS.

Sus competidores equivalentes serían el ruso Glonass, el chino Beidoru y el europeo Galileo cuya funcionalidad e infraestructura que lo soporta son equivalentes, sólo son diferentes en lo referente a las bandas de frecuencias usadas por cada uno, la vocación del mismo (ya sea de uso comercial, gratuita o ambas), a la exactitud y la presión que proveen, aunque actualmente se trata de crear una homogeneidad entre todos ellos con respecto a GPS con el fin de poder brindar el servicio a todo tipo de receptor (en especial los GPS ya que es el más comercial, barato y usado).

Identificación de Células (CELL-ID): Este método toma en cuenta el hecho que redes móviles pueden dar un aproximado de la posición (exactitud de 250 m para microceldas urbanas a 2.5 km para macroceldas urbanas) de un móvil conociendo la célula (Figura 2) en la que se encuentra en ese momento (siempre que el móvil tenga acceso a la red celular). Sin embargo la exactitud del método depende del tamaño de la célula, y ésta generalmente es de varios kilómetros (en el caso de telefonía celular).

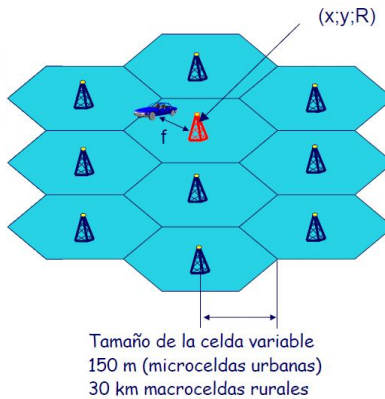


Figura 2 Infraestructura de CELL-ID.

“La ventaja de este método es su sencillez y que no supone ningún impacto para la red o la terminal.”
(Martínez Gens y Urios de Heras) En el caso de la telefonía celular, es posible incluir el sector de la célula donde se encuentra el teléfono celular, con lo cual se reduce el área de localización mejorando la exactitud del sistema. En entornos urbanos es posible que hasta 6 celdas puedan tomar la señal del móvil, además de la que le presta servicio al mismo, con ellos es posible obtener mejor la exactitud de la localización, a esto se le denomina sectorización (Figura 3).

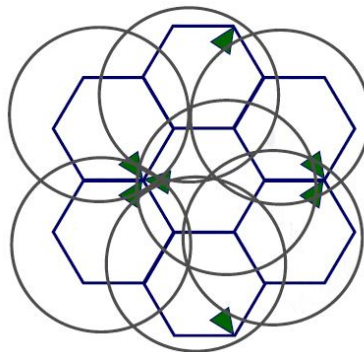


Figura 3 Funcionamiento de CELL-ID mas sectorización.

Dirección del ángulo de llegada (AOA): Esta técnica se basa en la idea de uso de arreglos de antenas direccionales (por lo menos 2) o antenas omnidireccionales (por lo menos 3), de forma que se pueda calcular el ángulo de llegada de la señal, para ello la idea es moverse en el espacio de las antenas hasta que la dirección de la máxima intensidad de la señal o que la “fase coherente”¹⁵ sea detectada, para posteriormente hacer el cálculo de la localización por medio de la triangulación de los ángulos obtenidos por cada antena (Figura 4).

¹⁵ Situación en la que dos señales mantienen una relación de fase fija entre sí o con una tercera señal que puede servir de referencia para cada una.

CAPITULO II: Marco Teórico

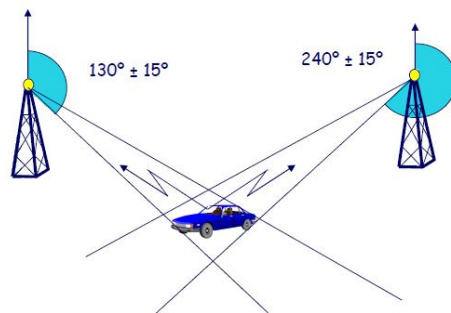


Figura 4 Sistema de localización por dirección de ángulo de llegada.

Normalmente los dispositivos móviles no incluyen un arreglo de antenas sino sólo una o dos antenas, por lo que es necesario la adquisición de hardware especializado para su uso, lo que conlleva una inversión adicional. En los sistemas satelitales móviles, la movilidad en la dirección de la antena es difícil de implementar, aunque ésta es fundamental para obtener una exactitud adecuada en aplicaciones móviles consientes del contexto.

Esta técnica resulta solamente efectiva cuando no existen multitrayectos¹⁶, ya que en caso de existir, se puede obtener un ángulo de llegada erróneo al recibir una o más veces la misma señal reflejada. Por lo cual los sistema de localización de este género son más adecuados para lugares despejados y que provoquen variaciones en las antenas emisoras por el viento o tormentas, además es frecuente el uso de un Sistema de Información Geográfica (GIS)¹⁷ como apoyo para tener una mejor exactitud en el cálculo de la localización.

Retardo de Tiempo: Esta técnica se basa en el hecho de que las ondas electromagnéticas viajan a una velocidad constante en el espacio, es decir, las ondas electromagnéticas tardan un tiempo constante en llegar de un emisor a un receptor. Por lo cual la distancia entre dos puntos puede ser determinada en función del retardo de tiempo que existen entre las ondas de radio transmitidas, debido a que existe una relación lineal entre tiempo de propagación y distancia recorrida.

Dentro de estos sistemas es necesario la recepción por el dispositivo móvil de al menos 3 antenas emisoras (conociendo de antemano la ubicación de las mismas), para que una vez obtenido el retardo de tiempo de la señal de cada antena se usen técnicas de triangulación (intersección de hiperboloides) para hacer el cálculo de la localización del móvil (Figura 5).

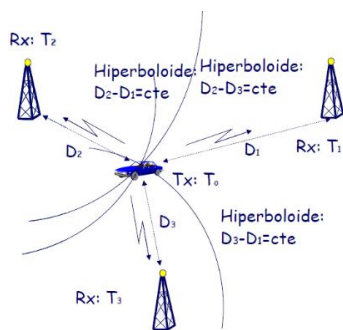


Figura 5 Sistema de localización por retardo de tiempo.

Este método es usado por los sistemas de satélites. Existen dos tipos de métodos de retardo:

¹⁶ Fenómeno que se da debido a la reflexión, difracción y/o dispersión que sufre una señal en el medio debido a diversos factores y que provoca la propagación de la onda correspondiente a la señal por varios caminos diferentes.

¹⁷ Sistema integro cuyo software y hardware está especialmente diseñado con el fin capturar, almacenar, manipular, analizar y desplegar la información geográfica necesaria para resolver cualquier necesidad o problema referente a los datos geográficos contenidos por el sistema.

CAPITULO II: Marco Teórico

- *Medición Absoluta del Tiempo de Llegada (TOA)*: La posición se estima del tiempo absoluto que la onda tardó en viajar entre el transmisor y el receptor o viceversa (se conoce exactamente el tiempo de transmisión), por lo cual debe existir una sincronía entre el transmisor y el receptor.
- *Medición del Tiempo Diferencial de Llegada (TDOA)*: Elimina la necesidad de sincronización entre el transmisor y el receptor utilizando muchos transmisores sincronizados a un tiempo común base y midiendo la diferencia del tiempo de llegada en el receptor de cada una de las señales de los transmisores (Figura 6).

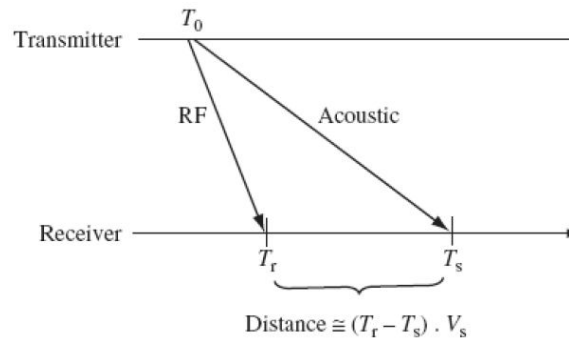


Figura 6 Cálculo de la distancia entre transmisor y receptor en TDOA.

En base a la información anterior, es posible observar que TDOA es el que resulta con mayores ventajas, debido a que al medir el tiempo de forma relativa y no absoluta se disminuye el error usando una resolución temporal mayor basada en el uso del tiempo común al que se sincronizan los transmisores.

Este tipo de sistema al igual que los basados en AOA necesita de un GIS¹⁸ para poder mejorar la exactitud en el cálculo de la localización, e igual que estos se enfrentan al problema de multitrayectos en espacios no despejados, ya que provoca que el tiempo de llegada sea más elevado al esperado y por lo cual disminuirá la exactitud en los cálculos de localización. Otros factores a considerar son el uso de relojes de alta exactitud (en general manejados por GPS), errores proporcionales al ancho de banda de la señal usada, y que generalmente es diseñado para procesamiento centralizado, por lo cual resulta obvio el uso de una infraestructura altamente especializada y costosa.

Sistema y Métodos de Localización en Ambientes Internos

Sistemas basados en Infrarrojos (IR): La tecnología infrarroja fue la primera en uso para la creación de sistemas de localización para ambientes internos, estos sistemas utilizan la proximidad entre dispositivos electrónicos con tecnología infrarroja. “Se utilizan *Tags*¹⁹ que emiten radiación infrarroja en modo difuso, es decir, de forma radial, no en modo punto a punto como es habitual en los sistemas IR empleados en comunicaciones.” (Sánchez Vitores)

¹⁸ Sistema de Información Geográfica (GIS, Geographic Information Systems), es un sistema de información que integra, almacena, edita, analiza, comparte y muestra información geográfica referenciada.

¹⁹ Etiquetas electrónicas pasivas (sin pila) o activas (con pila) que se pueden adherir a un objeto o persona, y que contiene una antena para poder recibir y/o enviar señales de RF hacia un emisor-receptor de RF.

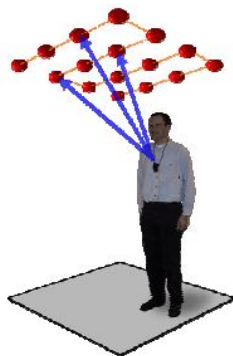


Figura 7 Sistema de localización basado en infrarrojo.

Este tipo de sistema puede considerarse más un sistema de detección que de localización, hace uso de la posición conocida de los Beacons²⁰ o sensores IR. Por lo cual, es necesario tener instalado una serie de Beacons, los cuales enviarán ráfagas de señales IR que serán leídas por los Tags, para que posteriormente esta información sea enviada al nodo central de la infraestructura, el cual se encargará del cálculo de la localización de cada Tag detectado.

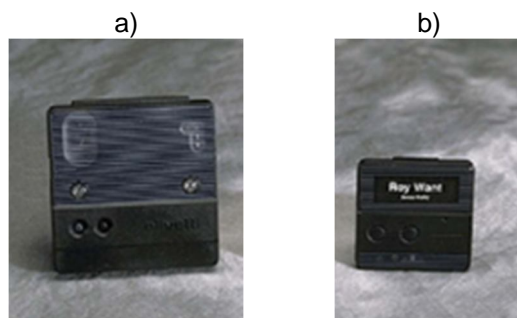


Figura 8 Ejemplo de a) Tag IR y b) Beacon IR.

Los principales inconvenientes que tienen estos sistemas son su corto alcance (unos cuantos metros), las señales IR no pueden atravesar paredes, son sensibles a la luz solar y fluorescente, al uso de una gran cantidad de Beacons (para mejorar la exactitud y cobertura, lo que incrementa el costo), además de que el cuerpo portador del Tag puede producir un efecto pantalla que produce un error en la detección del mismo, lo que conlleva al uso de más de un Tag por la entidad y objeto a detectar.

A pesar de esto, las ventajas que ofrecen estos sistemas son evitar interferencias y falsas detecciones por otro tipo de sensores (error común que sucede en RF).

Sistemas basados en Ultrasonido: Se trata de sistemas que están basadas en Tags o etiquetas para los elementos a controlar, estos Tags emiten o reciben señales ultrasónicas, dentro de la infraestructura del sistema los encargados de la detección y comunicación con los Tags son una serie de listeners²¹ ultrasónicos ubicados alrededor del ambiente donde se implementa el sistema.

²⁰ Pequeño dispositivo que envía señales RF en una determinada frecuencia, el cual se ubica de forma fija para cubrir un determinado espacio geográfico, este dispositivo se comunica por medio de mensaje con los Beacons sirviendo de apoyo para el cálculo de la ubicación de los mismos.

²¹ Pequeño dispositivo recibe los mensajes de los Beacons, usando esto para calcular la posición donde se encuentre este dispositivo.

CAPITULO II: Marco Teórico



Figura 9 Componentes de un sistema basado en ultrasonido a) Tag ultrasonico y b) Listener ultrasonico.

La estructura interna del Tag incluye: un transceptor radio de banda de 433 Mhz, una lógica de control con un ID único de 48 bits y un emisor de ultrasonidos. El método más sencillo para calcular la orientación, está basado en el cálculo de la orientación relativa (Figura 10 y Figura 11).

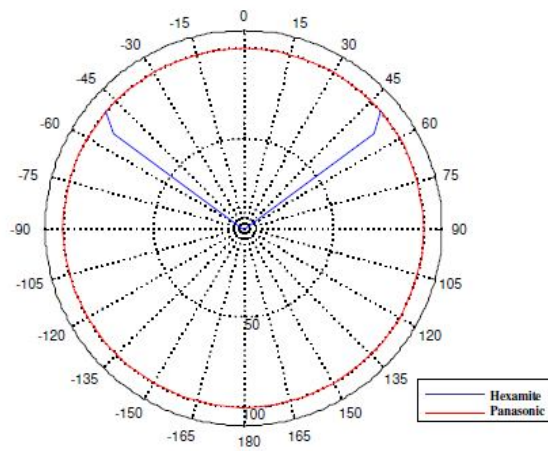


Figura 10 Patrón de recepción del sistema basado en ultrasonido Hexamite e implementación con micrófonos WM-61 de Panasonic.

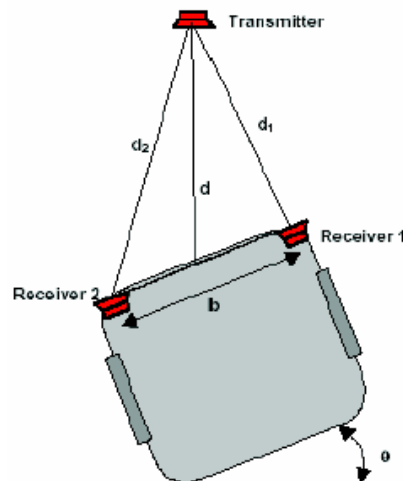


Figura 11 Ejemplo de cálculo de orientación relativa en un sistema de localización basado en ultrasonido.

Por lo general, la infraestructura que compone a estos sistemas está integrada por Tags ultrasónicos que forman una malla de puntos conocidos por lo general colocados en el techo, estaciones base RF que controlan a los Listeners ultrasónicos y un sistema central de gestión que se encarga de localizar y ubicar los Tags (Figura 12).

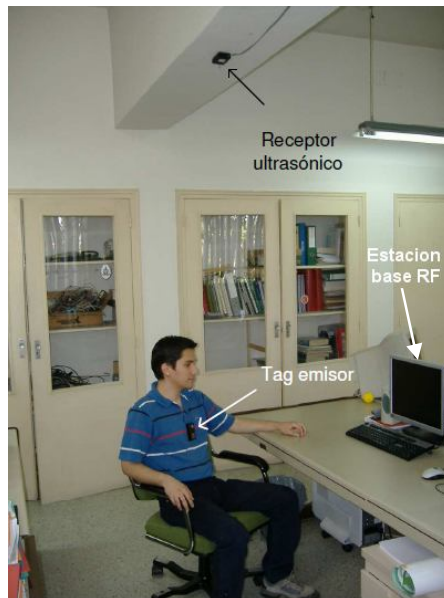


Figura 12 Infraestructura que compone a un sistema de localización basado en ultrasonido.

Una estación base transmite periódicamente un mensaje que contiene el identificador del Tag que desea activar y al recibir el mensaje con su identidad, el Tag aludido se despierta y emite un pulso de muy corta duración. Además, se resetea el reloj de los sensores del área de influencia, los cuales comienzan a contar el tiempo que transcurre hasta que reciben la señal del Tag. A partir de este retardo y de la velocidad de propagación del sonido en el aire, se calcula de forma inmediata la distancia al sensor. Cabe destacar que con las distancias del Tag a varios sensores (mínimo tres), puede conocerse la posición del Tag en 3 dimensiones. (Sánchez Vitores)

Las desventajas de este tipo de sistemas radican en el uso de equipo altamente especializado (con un elevado costo), corto alcance (hasta 8 metros máximo) y la susceptibilidad a equipo y ondas ultrasónicas que trabajen bajo la misma frecuencia que el sistema. Sin embargo, sus principales ventajas radican en la alta exactitud (con un error máximo de 3 cm) y precisión (95% de veces) del mismo.

Sistemas basados en Electromagnetismo: Estos sistemas se basan en el uso de ondas electromagnéticas de baja frecuencia para posicionamiento, es un método clásico que se auxilia de TDOA para el cálculo de la localización. Por lo general se producen campos electromagnéticos bajos y negativos (de entre -4 y -20 dB μ A/m) y los dispositivos usados por el sistema trabajando en el rango de frecuencia de 40 a 150 kHz.

Los elementos que componen a estos sistemas son dos principalmente: el transponder²² (Figura 13) y el lector/escritor²³. Estos sistemas generan campos magnéticos de forma constante desde los lectores/escritores, una vez que el transponder entra en el campo de acción del lector/escritor, absorbe la energía de éste, la cual se convierte en energía eléctrica que usa para poder generar un pulso electromagnético que envía el ID del propio transponder. El sistema calcula la posición y la

²² Componente pasivo (sin pila interna) o activo (con pila interna) compuesto por un chip y una antena que transmiten y reciben pulsos electromagnéticos, puede ser de sólo lectura (la información de éste es única y personalizada durante la fabricación) o de lectura/escritura (la información de éste puede modificarse por el lector/escritor).

²³ Componente integrado por un chip que emite y recibe energía electromagnética a través de una antena, con una electrónica especialmente diseñada para la recepción y codificación de la información enviada por los transponders y la cual envía al sistema de captura de datos.

CAPITULO II: Marco Teórico

orientación de las antenas de los lectores/escritores mediante la medición de la respuesta en los tres ejes ortogonales (x,y,z) del pulso transmitido.



Figura 13 Transponder, compuesto por una antena férrea, micro ATmega88, receptor MAS9180 y transceptor CC2500.

La ventaja de estos sistemas es su alta precisión, en el orden de menos de 1 mm de resolución espacial. Sin embargo, tiene un alto costo de implementación y la necesidad de que el objeto a ser localizado lleve consigo una unidad de control, además de que los sensores deben estar de uno a tres metros del transmisor y la precisión se ve afectada por la presencia de objetos metálicos en el ambiente.

Sistemas basados en visión computacional: El uso de la tecnología de visión computacional para calcular la ubicación de un objeto o entidad ha sido ampliamente tratada por diversos grupos de investigación, en la actualidad la tendencia de estos sistemas se encamina hacia el uso de técnicas de realidad aumentada y marcadores de seguimiento.

Estos sistemas se basan en la utilización de cámaras de alta precisión, funcionan en tiempo real, posicionadas en diferentes puntos, capturan secuencias de imágenes, que posteriormente son analizadas para identificar y localizar entidades y objetos específicos, para lo cual pueden o no auxiliarse de marcadores (etiquetas gráficas). El tipo de análisis que se hace principalmente a las imágenes incluyen estimación de profundidad, segmentación del color, clasificación de patrones de intensidad y reconocimiento de etiquetas (Figura 14). Tanto para el análisis de las imágenes como para el cálculo de la localización se hace uso de algoritmos altamente especializados, los cuales pueden incluir el uso de redes neuronales, lógica difusa y probabilidad, con el fin de obtener mayor exactitud y precisión. Además de esto, es necesario, determinar el mejor campo de visión de las cámaras, el cual se obtiene por lo general de forma experimental modificando el ángulo (Figura 15) de éstas hasta obtener el que proporcione una distorsión radial mínima.

Los inconvenientes de este tipo de sistemas radican en su alto consumo de recursos de cómputo necesario para analizar las imágenes, el rango de visión está limitado al número de cámaras usadas y sólo trabajan con un reducido número de objetos o entidades dentro del espacio. Sin embargo, el uso de técnicas de realidad aumentada y marcadores de seguimiento junto con el aumento de capacidad de cómputo de los teléfonos celulares, han hecho posible su uso en dispositivos móviles y la reducción de infraestructura que soporta al sistema, de igual forma en la actualidad el avance en cuanto al hardware de cámaras ha sido tal que sus precios han disminuido, sus capacidades han aumentado y existe una gran variedad de alternativas en el mercado de forma considerable.

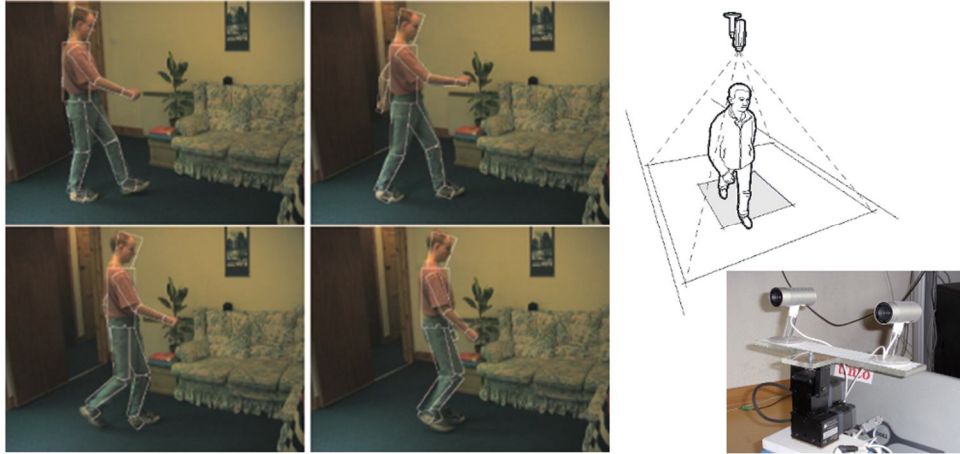


Figura 14 Ejemplo de funcionamiento de sistema de localización basado en visión computacional.

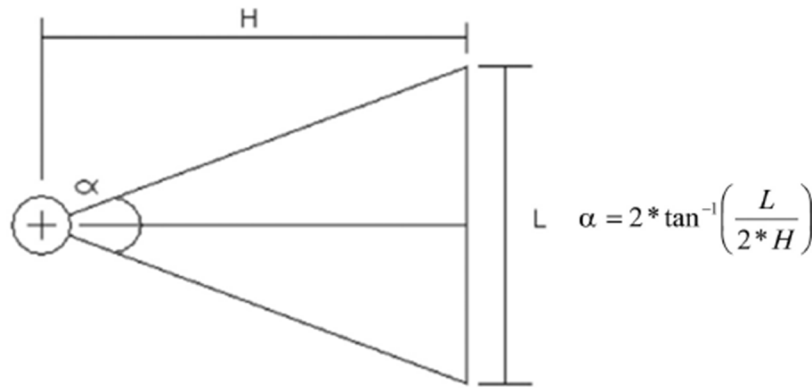


Figura 15 Ejemplo de cálculo del ángulo de visión α de una cámara donde L es la distancia de barrido y H es la distancia entre la cámara y la superficie.

Sistemas basados en contacto físico: Estos sistemas se basan en el contacto físico entre un objeto o entidad con el sistema. En general se usan sensores de presión (Figura 16) que capturan la existencia de algún tipo de presión específica sobre los mismos, para ser usados posteriormente en el rastreo y localización del objeto o entidad que se encuentra dentro del rango de cobertura del sistema. En general estos sistemas están más orientados hacia la detección e identificación de entidades y objetos que para su propia localización, logrando una exactitud del 93% en la identificación de los mismos.



Figura 16 Ejemplos de sensores de presión: a) Sensor minuatoura y b) Sensor flexible ultradelgado.

CAPITULO II: Marco Teórico

La ventaja de estos sistemas es que no se requiere del uso de algún dispositivo móvil para su localización. Las desventajas de estos sistemas son su pobre escalabilidad y un alto costo de instalación de los sensores debido a la modificación del ambiente mismo.

Sistemas basados en radiofrecuencia: Se utilizan dispositivos receptores-emisores que leen y envían señales RF, los cuales pueden ser fijos o móviles. Los dispositivos emiten radio-señales por lo general de 2.4 GHz o 5.8 GHz (según la tecnología usada). También pueden usarse Tags especiales denominados "RFID" ya sean pasivos (sin pila) o activos (con pila) que funcionan bajo frecuencias mayores a los 800 MHz, los cuales se comunican con lectores especialmente diseñados para el uso en conjunto con estos (Figura 18), su funcionamiento es similar al de los Tags y Lectores/Escritores usados en sistemas basados en electromagnetismo, sólo que en vez de enviar ondas electromagnéticas se envían señales de RF, tienen una exactitud de hasta 5m y un rango de alcance de hasta 10 metros para Tags pasivos y de 30 a 100 metros en Tags activos.

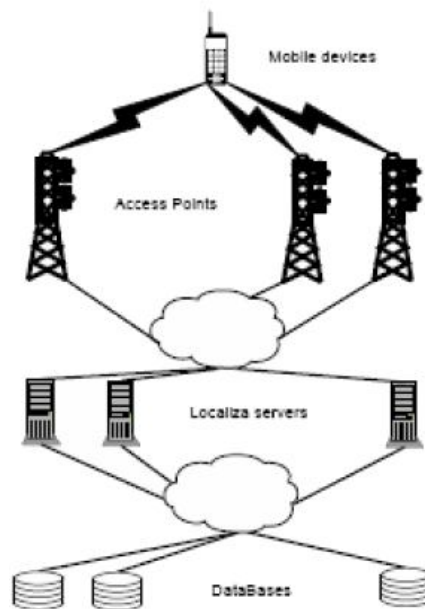


Figura 17 Arquitectura general de un sistema de localización basado en radiofrecuencia.

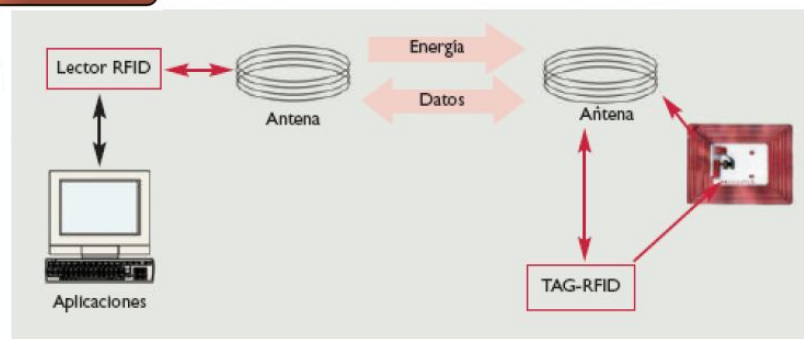
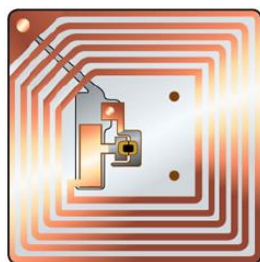


Figura 18 Arquitectura de un sistema de localización basado en radiofrecuencia y usando Tags RFID.

CAPITULO II: Marco Teórico

Estos sistemas se apoyan en diversas técnicas de localización que son particularmente interesantes porque no requieren de una infraestructura física especial y son alternativas variables para espacios cerrados, además de que el crecimiento de las tecnologías que las soporta (Wi-Fi, Bluetooth, GSM) ha ido en aumento en los últimos años.

Las ventajas que proporcionan este tipo de sistemas radican en la baja inversión de capital y uso de equipo especializado (a excepción de los basados en Tags RFID), variedad de tecnologías compatibles con los mismos, rango de cobertura de varios metros o decenas de metros (ideal para interiores) de acuerdo a la tecnología usada, exactitud en general de hasta 10 m y la capacidad de uso de una infraestructura centralizada o descentralizada para la localización de acuerdo a las necesidades del sistema.

Las desventajas de este tipo de sistemas radican en su falta de estandarización en general (debido al uso de diversa tecnologías) y a los errores de exactitud y precisión (debido a la atenuación y multitrayectos de las señales) generados por el medio ambiente y/o equipo que funcione bajo la misma frecuencia que el sistema.

Ejemplos de Sistema de Localización Implementados

GPS

Sus orígenes se atribuyen al gobierno francés y belga, aunque el desarrollo, implementación y funcionamiento se debe al Departamento de Defensa (DoD) de los Estados Unidos. El sistema GPS está compuesto por de 3 elementos denominados segmentos:

- **Segmento espacial:** formado por los satélites que son parte de la infraestructura.
- **Segmento de control:** formado por las estaciones de control que son parte de la infraestructura.
- **Segmento del usuario:** formador por los receptores GPS que desean conocer su posición apoyados de la infraestructura del sistema.

“El segmento espacial (SS, *Space Segment*) está formado por una constelación de 24 satélites (Figura 19), llamados SV (*Space Vehicle*). Circundando la Tierra a 20,200 kilómetros de altitud y formando 6 orbitas diferentes con 4 satélites cada una.”^(Correia) En promedio, al día se llega observar un máximo de 8 satélites, es importante destacar que para que un dispositivo GPS pueda ser ubicado necesita como mínimo entablar comunicación con 3 satélites.

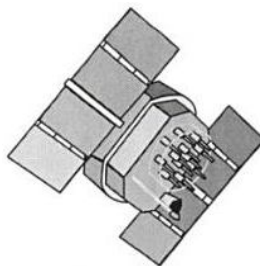


Figura 19 Satelite Navstar GPS.

El segmento de control (CS, *Control Segment*) está compuesto por: 5 estaciones de vigilancia (recolectan los mensaje satelitales y los envían a la estación principal), una estación principal (denominada MCS, *Master Control Station*, aplica las correcciones a los mensajes de los satélites) y 3 antenas terrestres (transmiten las correcciones hechas a los satélites, ubicadas alrededor del mundo) (Figura 20), en general cada satélite es observado durante más del 95% del tiempo.

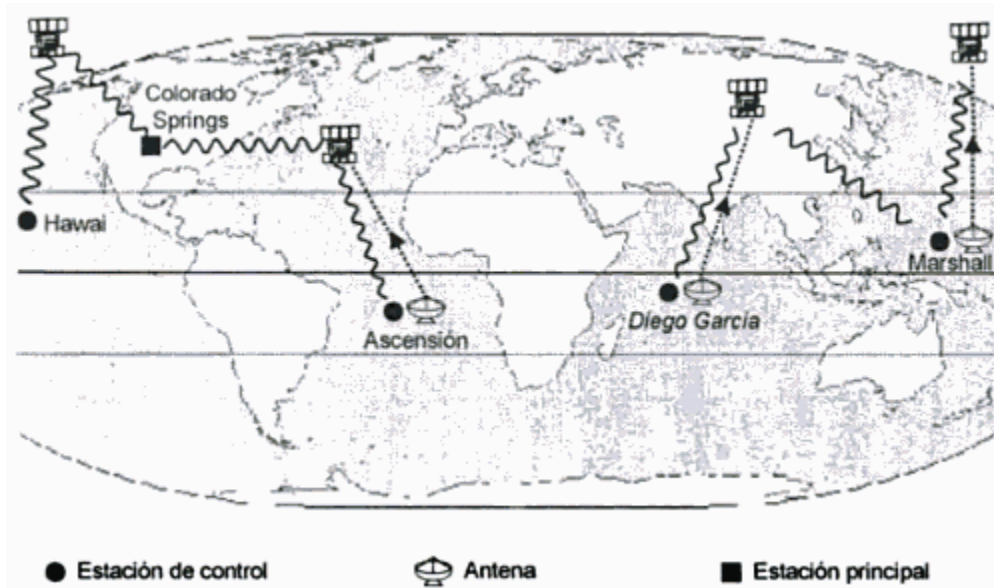


Figura 20 Ubicación de las estaciones de control, la estación principal y las antenas terrestres del segmento de control del sistema GPS.

El segmento del usuario está compuesto por los receptores GPS que desean obtener la posición con ayuda de la infraestructura que compone al sistema GPS, a partir de los mensajes recibidos de los satélites. Además es posible obtener información adicional sobre velocidad, ruta, hora, fecha, entre otra información útil para la navegación. El sistema GPS proporciona dos tipos de localizaciones: Posicionamiento Preciso (PPS) y Posicionamiento Estándar (SPS), debido a esto, se proporcionan para su uso dos bandas de comunicación, una para uso civil (para SPS) y la otra para uso militar (para PPS).

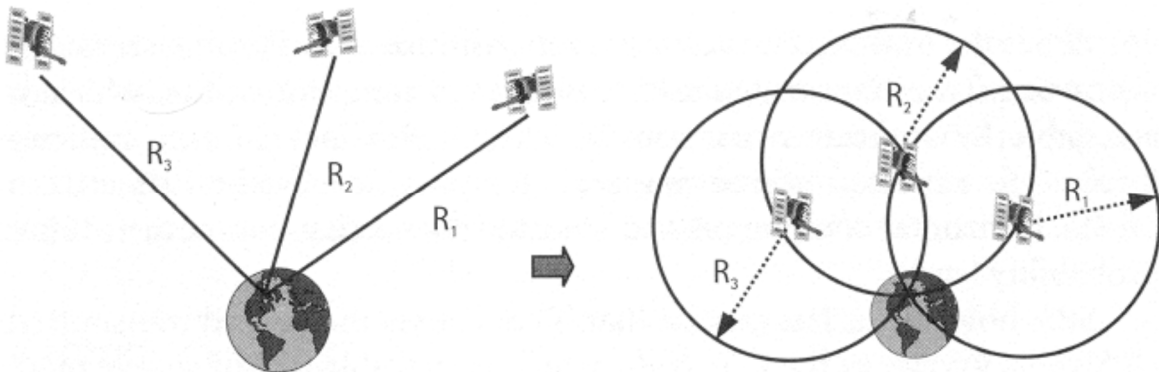


Figura 21 Ejemplo de localización de un receptor GPS.

En Julio de 2003 se presentó en el Instituto Politécnico Nacional, México, la tesis denominada “Sistema de Localización y Seguimiento de Móviles Terrestres Utilizando el Sistema de Posicionamiento Global” cuyo objetivo fue el de diseño, construcción e implementación de un prototipo de sistema de posicionamiento geográfico de móviles terrestres usan la tecnología GPS para ubicación dentro de un mapa virtual en tiempo real, denominado Sistema de Localización de Vehículos Terrestres (SLVT). “Permite conocer por medio de un programa computacional las características principales del recorrido como son: velocidad promedio, distancia, distancia total, velocidad promedio del viaje, punto de inicio, tiempo del recorrido, etc., así mismo mostrar en la pantalla por medio de un mapa por donde se desplaza el móvil.” (Rojano Romero)

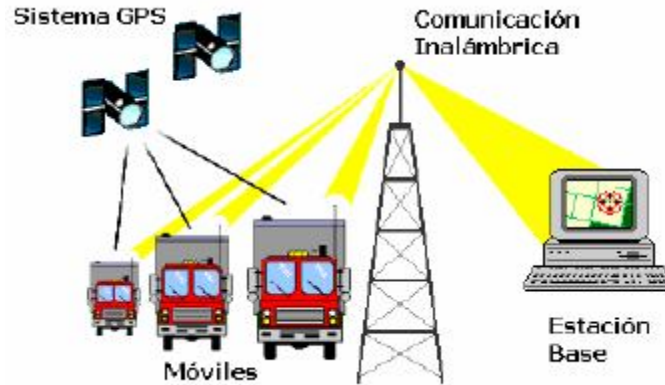


Figura 22 Esquema del Sistema SLVT.

Active Badge

Sistema basado en infrarrojo y desarrollado por los laboratorios Olivetti, el cual usa dispositivos metálicos con tecnología infrarroja (Figura 23) denominados Active Badge (con una dimensión de 55x55x7mm y peso de 40 gr.), envían un único identificador por un periodo 1/10 de segundo cada 15 segundos o cuando es requerido y sirven de apoyo a la infraestructura, la cual usa un único servidor que recolecta la información de todos los sensores fijos (Osciladores Beacons), para procesar la información y dar una ubicación simbólica de los Active Badge que sean detectados.

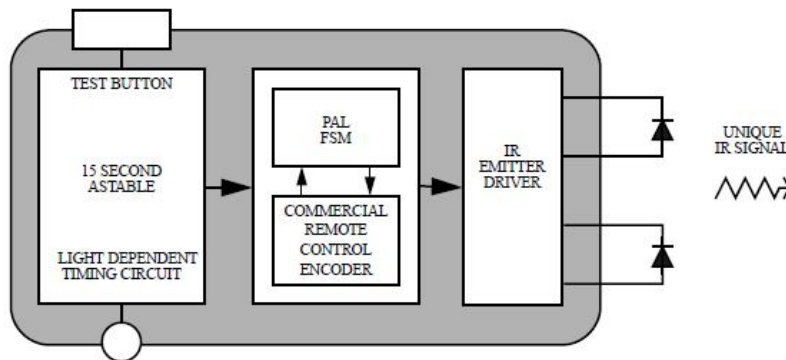


Figura 23 Composición de un Active Badge.

El sistema usa Modulación por Ancho de Pulso²⁴ (MPA o PWM, Pulse-Width Modulation) de IR durante la comunicación entre el Active Badge y el o los Osciladores Beacons (Figura 24), permitiendo un rango de cobertura de hasta 6m en la práctica. Los Activos Badges se diseñaron de tal forma que el funcionamiento de la comunicación de los mismos sea de forma periódica y no constante, esto tiene 2 propósitos: el ahorro de energía (auto-apagado es oscuridad total y una vida útil de la batería interna de hasta 1 año) y evitar colisiones entre 2 señales de diferentes dispositivos (teniendo 2/100 probabilidad de colisión entre 2 dispositivos colocados en el mismo lugar). En general las señales enviadas desde los Active Badge hacia los Osciladores Beacons no es directa, sino por medio de reflexión ya que las señales IR no pueden atravesar paredes y son reflejadas por muchos tipos de superficies incluyendo la tela de la ropa, por lo cual es fundamental que los Active Badge sean colocados de forma visible y no estén cubiertos por algún tipo de obstáculo como por ejemplo: telas, plásticos, etc.

²⁴ Técnica que consisten en la modificación del ciclo de trabajo de una señal periódica (una sinusoidal o una cuadrada).

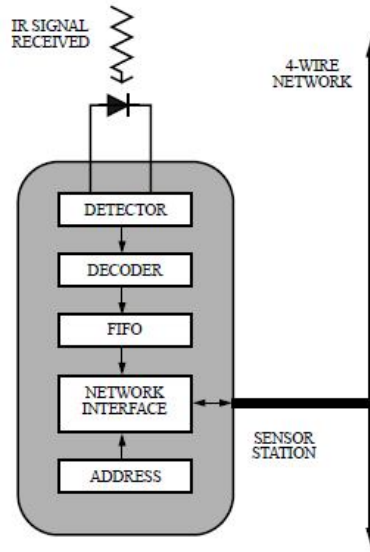


Figura 24 Composición de un oscilador Beacon.

El sistema cuenta con un diseño que permite una red independiente formada hasta por 128 Osciladores Beacons, donde cada Oscilador Beacon puede ser controlado por el puertos serie RS232 de una estación de trabajo normal, pudiendo apoyarse de la existencia previa de una red Ethernet de propósito general o implementando una de uso específico para el sistema, para entablar comunicación con el servidor principal el cual se ocupará del cálculo de la localización de cada Active Badge que sea detectado por algún Oscilador Beacon.

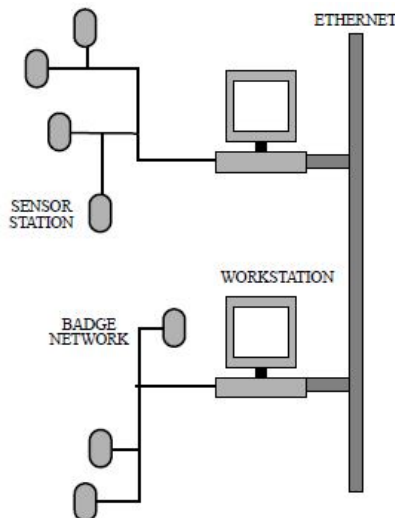


Figura 25 Infraestructura del Sistema Active Badge.

Active Bat

Sistema basado en ultrasonido y desarrollado por AT&T, usa una serie de Tags o etiquetas denominadas Bat que se colocan al nodo móvil o fijo del cual se desea conocer su ubicación, para ello el cálculo de la ubicación se hace por medio de sensores ultrasónicos (Listeners) que forma parte de la infraestructura que soporta al sistema. "Los Tags cuentan con un transceptor radio (banda de 433 MHz), una lógica de control que contiene un identificador único de 48 bits y un emisor de ultrasonidos." (Sánchez Vitores)



Figura 26 Ejemplo de un dispositivo Bat.

Como se mencionó anteriormente los Listeners se encuentran conectados a estaciones bases, los cuales se encargan de entablar comunicación con los Bats para realizar el cálculo de la posición, para el cálculo se hace uso de TOA, midiendo el tiempo entre la señal de reset enviada por los Listener (esto sucede cuando el Listener envía el ID de radio frecuencia hacia el Bat) y la llegada de la respuesta del Bat al Listener.

Dentro del sistema, cada estación base puede comunicar, por medio del Listener designado a la misma, un máximo de 3 Bats simultáneamente, con una frecuencia de refresco de 50 veces por segundo. En general en la práctica este sistema tiene una precisión de 6 cm, una exactitud del 95%. La vida útil de la batería interna del Bat es en promedio de 15 meses.

Cricket

Sistema basado en ultrasonido y desarrollado en el MIT, en el cual el nodo fijo o móvil se encarga del propio cálculo de su localización. "Permite que las aplicaciones se ejecuten en los nodos móviles y estáticos para conocer su ubicación física mediante el uso de *Listeners* que escuchan y analizan la información desde los *Beacons* repartidas por todo el edificio." (Priyantha, Chakraborty y Balakrishnan)

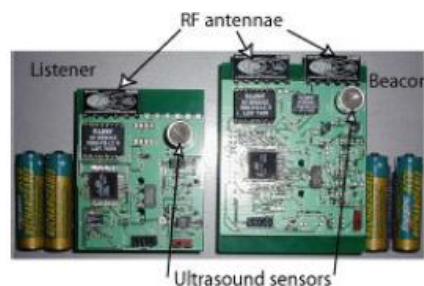


Figura 27 Ejemplo de dispositivo Listener y Beacon usado por el sistema Cricket.

Este sistema hace uso de señales RF y ultrasónicas con el fin de mejorar su rendimiento, para ello los Beacons realizan balizas de señales RF y ultrasónica simultáneamente (una a la vez, para evitar colisiones con balizas vecinas), donde las señales RF son anuncios de la existencia próxima del Beacon, además de incluir datos que proporcionan la ubicación del mismo; mientras que las señales ultrasónicas envían paquetes de datos. Una vez que el Listener recibe la señal RF y ultrasónica de un Beacon, procede al cálculo de la localización por medio del uso de TOA de las señales RF y ultrasónicas, además de usar la información de la ubicación propia del Beacon proporcionada por el mismo.

Al no requerir un sistema central para hacer el cálculo de la localización, los Beacons realizan de forma independiente y asíncrona la propagación de las balizadas (no necesitan estar conectados a ninguna estación de trabajo), por lo cual para evitar colisiones e interferencias, hacen uso de

CAPITULO II: Marco Teórico

planificación aleatoria para las balizadas (cada Beacon funciona con tiempos aleatorios en intervalo de 150 a 350 ms y 4 Hz). Los *Listener* usan un algoritmo de Estadística Mínima para superar los problemas de múltiples ultrasonidos e interferencia de RF durante la decodificación de las señales. Además, el *Listener* posee una API para programas ejecutados en el nodo que permite al mismo tiempo el aprendizaje de la ubicación donde se encuentran, con lo cual, este puede mejorar la forma de anunciarse y la ubicación de su servicio de descubrimiento de recursos de forma más óptima.



Figura 28 Ejemplo de la ubicación de Beacons del sistema Cricket dentro de un edificio.

En general el rango de cobertura del sistema es de 1.2 m de radio para cada Beacon, el sistema cuenta con 2 versiones. En la versión 1, la precisión es de 40 cm usando un detector de tonos en la señal ultrasónica, por otro lado, en la versión 2 la precisión es de 2 cm usando un detector de umbral en la señal ultrasónica (150 ms de emisión) y una mejor estimación mediante el uso de un filtro de Kalman²⁵.

SmartFloor

Sistema basado en contacto físico y desarrollado por el Colegio de Computación y el Centro GVU del Instituto de Georgia, Atlanta. Este sistema permite el reconocimiento de objetos y entidades mediante el uso de sensores de presión (Figura 29) instalados en posiciones específicas de una superficie y que conecta a un servidor, el cual no sólo determina la posición de la entidad a ubicar, sino que además permite identificarlo biométricamente²⁶ en base a un modelo de pasos (forma del pie) de usuarios, no se requiere del uso de ningún dispositivo móvil para su funcionamiento.

El modelo de pasos de cada usuario está compuesto por 10 elementos del perfil (Figura 30) que son: valor medio del perfil, desviación estándar del perfil, la longitud del perfil (número de muestras), el área total bajo la curva del perfil, la coordenada del punto máximo en la primera mitad del perfil (x_{max1} , y_{max2}), la coordenada del punto máximo en la última mitad del perfil (x_{min} , y_{min}) y la coordenada del punto mínimo entre los dos puntos como máximo (x_{max2} , y_{max2}).

²⁵ Algoritmo que sirve para poder identificar el estado oculto (no medible) de un sistema dinámico lineal, aun cuando el sistema está sometido a ruido blanco aditivo.

²⁶ Estudio encaminado a los métodos automáticos de reconocimiento único de personas en función de rasgos de conductas y/o físicos.

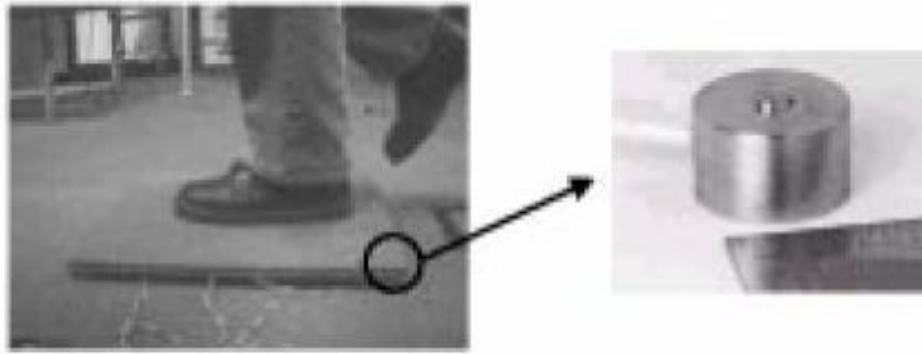


Figura 29 Ejemplo de sensor de presión del sistema Smart Floor.

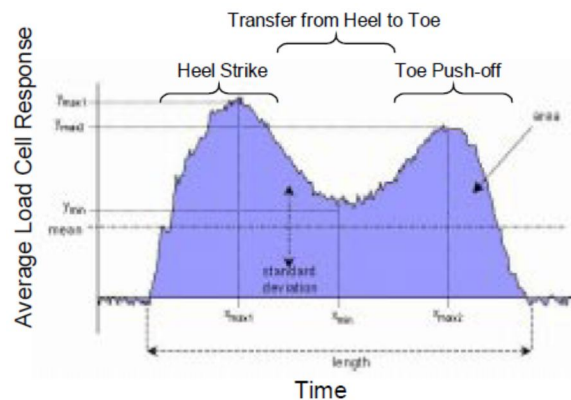


Figura 30 Ejemplo de los 10 elementos del perfil de una muestra tomada de un usuario por el sistema Smart Floor.

El sistema ha sido probado con grupos pequeños de entrenamiento (en promedio 15 personas) teniendo un 95% de exactitud a la hora de identificar usuarios. Se ha demostrado que el sistema es capaz de identificar a una persona a pesar de llevar calzado. El sistema identifica al usuario usando la Fuerza de Reacción del Suelo²⁷ (GRF, *Ground Reaction Force*), obtenida por medio 3 componentes de hardware: un sensor de presión, una placa de acero que recubre al sensor y el hardware de recolección de datos del sensor de presión, estos interactúan entre si y en conjunto con una terminal que envía la información al servidor para el tratado de la información y la identificación de la persona. Para identificar a un usuario que lleva un tipo de calzado diferente al que se usó inicialmente, se auxilia de la búsqueda del vecino más cercano que llegue a coincidir con éste por medio del cálculo de la distancia Euclídiana.

Singpost

Sistema basado en el uso de técnicas de realidad aumentada y marcadores de seguimiento, fue financiado por el Fondo de Ciencia Australiano (FWF, *Austrian Science Fund*). Fue presentado y usado por primera vez en la Conferencia de Desarrolladores Móviles y Embebidos de Microsoft (MEDC, *Microsoft Mobile and Embedded Developers Conference*) del 2007. Su funcionamiento se basa en la colocación de patrones denominados "marcadores de seguimiento", los cuales son ubicados de forma visual (impresa o digital) y posteriormente son reconocidos por medio de las cámaras de los móviles que implementan la aplicación correspondiente del sistema. Una vez que el marcador es reconocido, se muestra la posición actual dentro del edificio y los posibles caminos a seguir de acuerdo con el destino al que desea llegar el usuario y a los marcadores que se hayan reconocido con anterioridad.

²⁷ Consiste del peso y la inercia del cuerpo medidas.

CAPITULO II: Marco Teórico

Este sistema, se auxilia del uso de Marcadores Fiduciarios,²⁸ es un mecanismo bien establecido en aplicaciones móviles y fáciles de distinguir por el usuario en ambientes altamente saturados de imágenes debido a su particular apariencia, en comparación con otros tipos de marcadores de seguimiento existentes. "A diferencia de las características naturales de seguimiento, la detección y decodificación de marcadores artificiales es muy robusta y funciona bien sobre variaciones de las condiciones de iluminación y recursos mínimos computacionales." (Mulloni, Barakonyi y Wagner) Estos marcadores permiten 6 grados de libertad (DOF, *Degrees of Freedom*) de la cámara de dispositivo móvil con respecto a la orientación y posición del marcador (se ha demostrado en la práctica que con sólo 3 DOF es suficiente), con el fin de realizar la identificación y recolección de datos.



Figura 31 Ejemplo del uso de un Marcador Fiduciario en un cartel de MEDC 2007.

Posteriormente, se introdujo una menor obstrucción en la visibilidad de los Marcadores Fiduciarios por medio de los Marcadores de Marco (Frame Markers), los cuales permiten la misma cantidad de datos que sus predecesores (36 bits), para ello los datos se decodifican en forma de marco (cada lado contiene 9 bits de datos), este marcador organiza el código en el sentido de las manecillas del reloj, con lo cual sólo una de las 4 formas de observar el código es válida, lo que facilita la identificación y recolección de datos del mismo. Con las modificaciones hechas al sistema, es capaz de realizar la identificación y lectura de los marcadores en tiempo real como con otros tipos de marcadores de seguimiento y sistemas ya existentes, tiene una tolerancia de ángulos de inclinación de hasta 70 grados y rotaciones de los ejes ópticos de la cámara del móvil con respecto a los marcadores, para el buen funcionamiento cada cuadro (que representa 1 bit) debe de ocupar como mínimo 2 pixeles en la imagen de la cámara (lo cual resulta sumamente óptimo). Es importante destacar que el sistema usa internamente una base de datos (que se auto-actualiza constantemente mientras se leen marcadores) y mapas 2D (se hicieron evaluaciones en mapas 3D, pero no fueron óptimas desde la visión del usuario y alentaron al sistema) pre-cargados para la localización y ubicación.

²⁸ Etiqueta o diseño con símbolos gráficos usado en el campo de visión computacional para uso como punto de referencia, de medida o identificador, cada cuadro blanco o negro representan 1 bit de datos.



Figura 32 Marcador de Marco Der) Colocado en un cartel de Microsoft's TechReady7 e Izq) Identificación y lectura del marcador por medio del sistema Singpost.

Técnicas de localización más usuales en los sistemas para interiores basados en radiofrecuencia

Geométricos: Se auxilian de propiedades geométricas para tratar un problema de localización, como un problema de intersección de círculos o esferas en una determinada posición, mediante distancias y ángulos conocidos. Por ello, este tipo de técnicas suponen que el tiempo y/o la potencia de una señal guarda una relación proporcional con la distancia existente entre el emisor y el receptor, además que consideran de ante mano que el patrón de radiación generado por una antena es de forma esférica con centro en la misma. Ejemplos de estas técnicas:

- **Trilateración:** Es un método para determinar la posición entre un receptor-emisor móvil, basado en la distancia existente entre éste y otros receptores-emisores fijos colocados en determinadas posiciones estáticas. Por lo general, se necesitan al menos 3 puntos de referencia conocidos para calcular de forma única la localización relativa de un punto en un plano bidimensional. La derivación de la solución de un problema de trilateración tridimensional se puede obtener a partir de las ecuaciones de tres esferas, que al igualarlas se puede reducir a un problema de intersección de una línea y una esfera. En la Figura 33 se puede observar un ejemplo de cómo se solucionaría un problema de Trilateración tridimensional donde $p_i = (x_i, y_i, z_i)$, $i = 1, 2, 3$ son las coordenadas de los puntos de referencia conocidos i (emisores-receptores fijos), l_i es la distancia existente entre la posición de punto desconocido (emisor-receptor móvil) y la referencia conocida i , además de que $a = \|p_2 - p_1\|$ y $b = \|p_3 - p_1\|$.

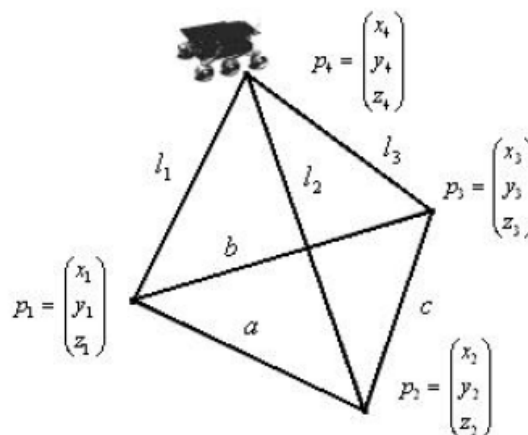


Figura 33 Visualización de un problema de Trilateración tridimensional con 3 puntos de referencia conocidos.

Para resolver el problema se obtiene el siguiente sistema de ecuaciones:

$$(x - x_1)^2 + (y - y_1)^2 + (z - z_1)^2 = l_1^2$$

$$(x - x_2)^2 + (y - y_2)^2 + (z - z_2)^2 = l_2^2$$

$$(x - x_3)^2 + (y - y_3)^2 + (z - z_3)^2 = l_3^2$$

El cual se puede simplificar a un problema de intercepción de una línea y una ecuación cuadrática.

$$(x - x_1)^2 + (y - y_1)^2 + (z - z_1)^2 = l_1^2$$

$$x_2 (x - x_1)^2 + y_2 (y - y_1)^2 + z_2 (z - z_1)^2 = (l_1^2 - l_2^2 + a^2) / 2$$

$$x_2 (x - x_1)^2 + y_2 (y - y_1)^2 + z_2 (z - z_1)^2 = (l_1^2 - l_3^2 + b^2) / 2$$

- *Triangulación.* La técnica de triangulación utiliza las propiedades de los triángulos para determinar la localización de objetos. La triangulación se puede dividir en subcategorías de:
 - Lateración: La lateración calcula la posición de un objeto midiendo su distancia relativa a múltiples puntos de referencia cuyas posiciones son conocidas y no alineadas. Para el cálculo de la localización de un objeto en 2D se requiere de por lo menos 3 puntos de referencia (Figura 34), mientras que para 3D son necesarios 4 puntos de referencia.

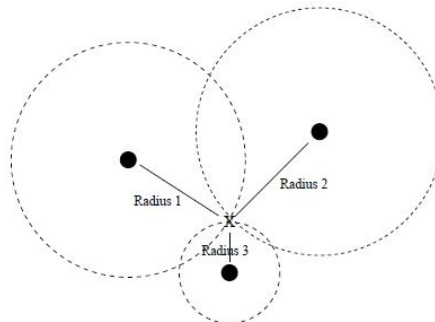


Figura 34 Ejemplo del cálculo de localización en 2D por medio de Lateración.

Existen 3 enfoques generales para que un sistema obtenga las mediciones de distancia para la técnica de lateración:

- **Directo:** Se hace la medición de la distancia utilizando una acción física o movimiento (de nula utilidad para los sistemas basados en RF, aunque suele ser usado en robótica).
- **Retardo de tiempo:** Utiliza el retardo de tiempo que existe entre la emisión de una señal desde un punto de referencia conocido, hasta la llegada de esta, a un punto desconocido, para ello se puede usar TOA o TDOA (en general, el retardo de tiempo de una señal se ve afectado por objetos en el ambiente y la temperatura del mismo).
- **Atenuación:** Usa la intensidad de una señal para obtener la distancia existente desde un punto de referencia origen hasta el punto de llegada desconocido, para ello considera que existe un factor de atenuación de la señal proporcional a la distancia de recorrido, es decir, por cada cierta distancia recorrida se produce una unidad de atenuación en la intensidad original de la señal (igual que en el caso del retardo de tiempo, los objetos en el ambiente y la temperatura del mismo afectan la intensidad de la señal).

CAPITULO II: Marco Teórico

- o Angulación: Similar a la lateración, a excepción, de que en lugar de las distancias, son usados los ángulos para determinar la posición de un objeto. Por lo general, para angulación bidimensional, se requieren dos mediciones: ángulo y longitud, mientras que para la angulación tridimensional se requiere por lo menos una medición de longitud, una de “acimut”²⁹, y dos de ángulo para especificar una posición precisa (Figura 35).

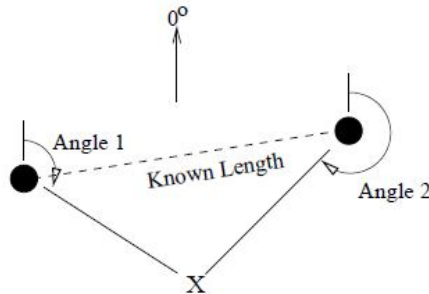


Figura 35 Ejemplo del cálculo de localización 3D por medio de Angulación.

La triangulación por radiofrecuencia puede basarse en: el tiempo de propagación, la potencia de la señal recibida, o el ángulo de llegada. En la Figura 36 se observa un ejemplo de cómo se solucionaría un problema de Triangulación tridimensional donde se detectaron 3 puntos de referencia conocidos, denotados por z_o , z_i , y z_j ; se conocen los ángulos formados entre cada par de ellos denotados por φ_{oi} y φ_{ij} , las distancias entre cada una denotadas por $|v_{oi}|$, $|v_{ij}|$ y $|v_{oj}|$; el punto desconocido esta dado por $p = (p_x, p_y)$; y las distancias no conocidas desde el punto desconocido hasta cada punto conocido se denominan $|z_o^{(r)}|$, $|z_i^{(r)}|$ y $|z_j^{(r)}|$.

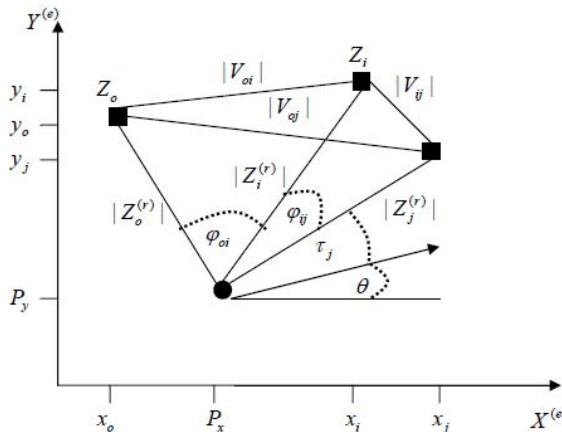


Figura 36 Visualización de un problema de Triangulación tridimensional con 3 puntos de referencia conocidos.

Usando la ley de coseno la relación entre z_i y z_j genera la siguiente ecuación:

$$|v_{ij}| = |z_i^{(r)}| + |z_j^{(r)}| - 2|z_i^{(r)}| |z_j^{(r)}| \cos \varphi_{ij}$$

Con base a la ecuación anterior, y aplicando el mismo procedimiento a cada par de puntos de referencias conocidos se obtiene el siguiente sistema de ecuaciones:

²⁹ Es el ángulo de una dirección contado en el sentido de las agujas del reloj a partir del norte geográfico. El acimut de un punto hacia el este es de 90 grados y hacia el oeste de 270 grados sexagesimales. El término acimut sólo se usa cuando se trata del norte geográfico.

$$|v_{0i}| = |z_0^{(r)}| + |z_i^{(r)}| - 2|z_0^{(r)}| |z_i^{(r)}| \cos \varphi_{i0}$$

$$|v_{0j}| = |z_0^{(r)}| + |z_j^{(r)}| - 2|z_0^{(r)}| |z_j^{(r)}| \cos \varphi_{0j}$$

$$|v_{ij}| = |z_i^{(r)}| + |z_j^{(r)}| - 2|z_i^{(r)}| |z_j^{(r)}| \cos \varphi_{ij}$$

Expresando las coordenadas del punto desconocido en función de las distancias conocidas se obtiene el siguiente sistema de ecuaciones:

$$|z_0^{(r)}|^2 = (x_0 - p_x)^2 + (y_0 - p_y)^2$$

$$|z_i^{(r)}|^2 = (x_i - p_x)^2 + (y_i - p_y)^2$$

$$|z_j^{(r)}|^2 = (x_j - p_x)^2 + (y_j - p_y)^2$$

Igualando y sustituyendo los dos sistemas de ecuaciones obtenidos anteriormente se determina el siguiente sistema de ecuaciones soluble y con el cual es posible estimar la posición de punto desconocido p:

$$|z_0^{(r)}|^2 - |z_i^{(r)}|^2 = x_0^2 - x_i^2 + 2p_x(x_i - x_0) + y_0^2 - y_i^2 + 2p_y(y_i - y_0)$$

$$|z_0^{(r)}|^2 - |z_j^{(r)}|^2 = x_0^2 - x_j^2 + 2p_x(x_j - x_0) + y_0^2 - y_j^2 + 2p_y(y_j - y_0)$$

La ventaja de estos métodos es su aplicación directa en cualquier entorno, con la suposición que el modelo empleado en cada caso se adapte a la propagación real.

La desventaja es que si la propagación real difiere mucho de la teórica, es necesario la obtención de un modelo adaptado del entorno por lo que la generalidad se pierde (ventaja principal de este método). Otro inconveniente de este enfoque es que al tratar los Niveles de Potencia Recibida (RSS, *Received Signal Strength*) como distancia, no tienen en cuenta obstáculos (paredes, mobiliario, suelos, etc.) por lo que no se consideran muy apropiados para interiores (entornos cerrados).

Reconocimiento de patrones: Las técnicas de reconocimiento de patrones (*fingerprinting*), intentan acoplar el vector de RSS obtenido de varios puntos de acceso en un determinado momento, con una muestra de calibración previamente definida, sin la necesidad de usar algoritmos geométricos. Para ello, primero es necesario obtener un radio mapa patrón, este puede ser implementado mediante una base de datos que contiene la siguiente información: posiciones en el edificio y vectores RSS de cada punto de acceso en esas posiciones. Ejemplos de estas técnicas son:

- **Los métodos K-Nearest Neighbors (K-NN):** La idea sobre la que se basa el método es que, aunque la RSS recibida en el emisor-receptor móvil no depende linealmente de la distancia al receptor-emisor fijo, sino que guarda alguna relación; al menos se puede asegurar que si el receptor-emisor fijo está muy cerca del receptor-emisor móvil la potencia recibida será alta, y viceversa. Con esta idea base, se crea un radio mapa, el cual es una base de datos de localizaciones (conjunto de vectores de prueba), que contiene la posición de cada localización (coordenada de referencia) y la RSS de los puntos de acceso. Cuando se desea localizar el emisor-receptor móvil, se obtiene un vector de potencias de las coordenadas de éste, posteriormente se compara con la base de datos de localizaciones y se trata de estimar la localización observando cuáles de todos los vectores de prueba son más parecidos al vector obtenido. Una vez calculados los vectores de prueba más similares al vector obtenido, mediante votación se puede identificar cuál es la ubicación más probable. La distancia existente entre el vector obtenido y cada uno de los vectores de prueba puede

ser calculada, por medio de la distancia euclidiana (entre otras medidas de distancia). La desventaja principal de este método es la necesidad de calcular la distancia a cada uno de los vectores de prueba, lo cual puede llevar bastante tiempo si hay muchos puntos de acceso y/o muchas localizaciones posibles.

- **Métodos Bayesianos:** Esta clase de método se auxilia de la construcción de una red bayesiana en la fase de entrenamiento, de tal forma que el usuario determina los puntos de interés, para posteriormente entrenar durante un tiempo al sistema en cada uno de los puntos de interés, recogiendo periódicamente muestras de RSS. Durante la fase de entrenamiento como primer paso se debe establecer la probabilidad a priori al estar en una localización $p(L)$, posteriormente son definidas las probabilidades a posteriori, al ser estas las probabilidades condicionadas de recibir una RSS desde un punto de acceso cuando se está en una localización determinada $p(E|L)$. Estos métodos también modelan transiciones imposibles de una localización a otra, por medio de un modelo similar al Modelo Oculto de Markov (HMM).
- **Máquinas de Soporte Vectorial (SVM, *Support Vector Machines*):** Este método es un paradigma de las Redes Neuronales, en el cual se procesan los vectores de observación, basándose en el uso de un espacio de hipótesis de funciones lineales en un espacio de mayor dimensión (espacio de características alto-dimensional) que la dimensión de las observaciones inducido por funciones núcleo (kernel), con el fin de obtener un hiperplano que separe linealmente las observaciones y permita la localización de la forma más fiable como sea posible (vector de soporte). Esto se logra basándose en hiperplanos que separan el conjunto de datos de entrenamiento en 2 subgrupos que poseen cada uno una etiqueta propia, de entre todos los posibles planos de separación existe un único hiperplano de separación óptima (OSH), el cual es calculado al maximizar la distancia entre el hiperplano de separación óptima y el patrón de entrenamiento más cercano, es decir la maximización del margen (sesgo inductivo particular de las SVM). Una de las características más importantes de las SVM es que al ser la separación entre conjuntos máxima, en teoría este método tiene una gran tolerancia a errores, considerando que el entrenamiento de este se haya hecho con suficientes observaciones. Un punto importante a considerar de las SVM radica en que para la realización del entrenamiento se requiere un gran conjunto (con un mínimo de 50,000 datos) de datos de observación (vectores de observación) lo cual implica a un alto consumo de recursos computacionales y de tiempo durante la fase de entrenamiento.
- **Métodos basados en Redes Neuronales:** En este método se usa una Red Neuronal para calcular la localización, por medio de la observación de la RSS de los distintos emisores-receptores fijos. La Red Neuronal usada es un Perceptrón Multicapa (Red Neuronal Artificial formada por múltiples capas), en el cual las entradas pueden ser la RSS de cada uno de los emisores-receptores fijos, y las salidas son las probabilidades de estar en cada una de las localizaciones. Un factor importante a considerar de este método es la gran cantidad de tiempo que lleva el entrenamiento del mismo y la necesidad de enormes cantidades de datos de prueba, además de correr el riesgo de sobreajustarse (*Overfitting*³⁰).

En general las ventajas de estos métodos frente a los geométricos es que se requiere menos cálculos. Debido a la discretización del entorno se puede determinar el grado de precisión de este método, es decir, cuanto más se parta el espacio del entorno, mayor es el grado de precisión. Los cálculos que permiten la estimación de la localización en los algoritmos de reconocimiento de patrones son sencillos y cambian según sea el método a emplear.

³⁰ Efecto de sobreentrenar un algoritmo de aprendizaje con unos ciertos datos para los que se conoce el resultado deseado, el algoritmo de aprendizaje puede quedar ajustado a unas características muy específicas de los datos de entrenamiento que no tienen relación causal con la función objetivo.

En general las desventajas de estos métodos radican, en que debido a la discretización del entorno hace que no sea posible obtener una localización que no se haya definido a diferencia de los geométricos que puede localizar en cualquier punto del espacio por no haberse discretizado el entorno. El tiempo de cómputo y de almacenamiento está en función del grado de discretización, a mayor discretización mayor será el tiempo de cómputo y almacenamiento, y viceversa.

Ejemplos de Sistemas de Localización para interiores basados en radiofrecuencia

PinPoint 3D-iD

Es un sistema de localización para interiores desarrollado por el Departamento de las Ciencias de la Computación de la Universidad de Maryland, usa estaciones fijas, células controladoras, antenas y Tags activos RFID de 2.4 y 5.8 GHz, con la tecnología L3RF (*Low rang, Long life, Low cost*) propietaria (Figura 37) y aplica el método geométrico de triangulación y TOA de ida y vuelta de señal para la localización.

En la etapa de comunicación entre las antenas y los Tags, las primeras envían una señal en la frecuencia de 2.4 GHz a los Tags, estos responden con una señal en la frecuencia de 5.8 GHz que se envía en intervalos de tiempo definidos y constantes la cual incluye su ID, la repuesta de los Tags es recibida por las antenas y canalizada a las células controladoras que se encargan del cálculo de la localización por medio de una estación base.

En general este sistema proporciona una precisión de entre 1 a 3 m, no necesita de la sincronización entre las antenas cuando estas envían sus señales, el Tag puede ser configurado para poder retransmitir a 3 estaciones simultáneamente con lo que se resuelve la posición en X-Y y para Z se fija un valor, y cada estación fija trabaja con una célula controladora, la cual a su vez trabaja hasta con 16 antenas conectadas por medio de cable coaxial. Sin embargo, el área de cobertura está limitada por el número de antenas debido a que las mismas son muy directivas y es necesario la creación de una red *ad hoc* de propósito único para la comunicación entre las estaciones base y células controladoras.

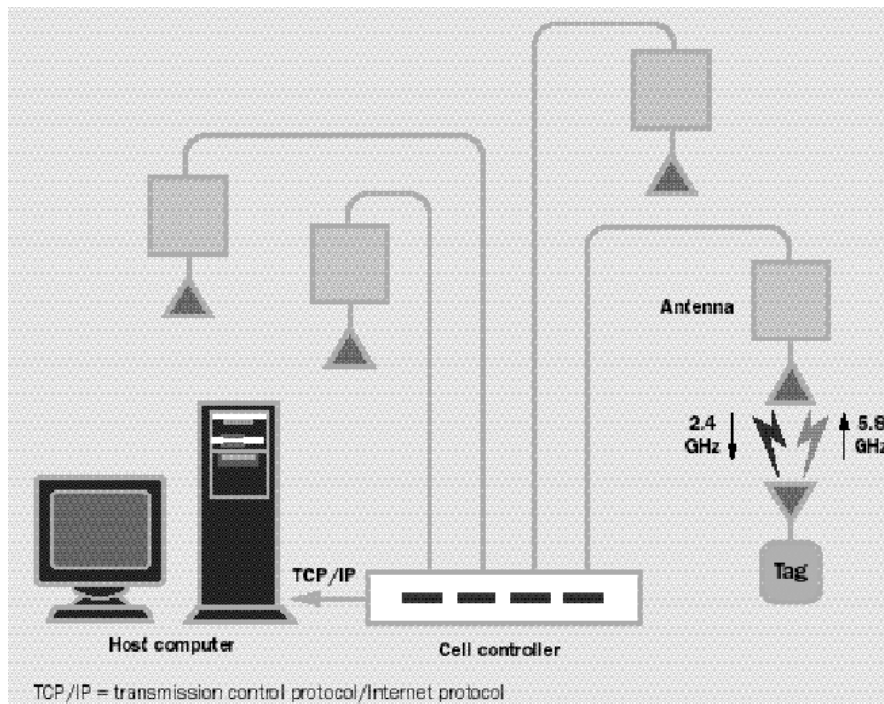


Figura 37 Arquitectura general del sistema PinPoint 3D-iD.

RADAR

Es un sistema de localización para interiores desarrollado por Microsoft Research en el 2000, hace uso de estaciones bases y Tags que cuentan con la tecnología Wi-Fi (estándar IEEE 802.11). Usa el método de reconocimiento de patrones *K-NN* para la localización por medio de la creación de un radio mapa patrón basado en un modelo matemático que toma en cuenta los obstáculos del medio ambiente en una de las dos versiones, mientras que en la otra usa el método de triangulación por lateración.

Durante la etapa de comunicación las estaciones bases y los Tags cuentan con una frecuencia de refrescado de 50 veces por segundo, para reducir el costo computacional el sistema sólo calcula distancias a localizaciones que se consideren posibles en función de la existencia de redes inalámbricas durante la última búsqueda en el medio ambiente de las mismas.

El sistema se crea para uso en modelos 2D (Figura 38) debido a que su implementación dentro de un edificio de dos o más pisos produce problemas difíciles de solucionar por las atenuaciones y reflexiones que se producen cuando las señales atraviesan los diferentes pisos del edificio, estas alteraciones sufridas por las señales distorsionan enormemente las mediciones, lo que llevará a ubicar al Tag en un piso erróneo. Cuenta con una precisión de 3 a 4 m, una exactitud del 50% y se puede apoyar para su implementación de la preexistencia de redes WLAN, sin embargo, los dispositivos Wi-Fi tienen un costo moderado y un alto consumo de energía.

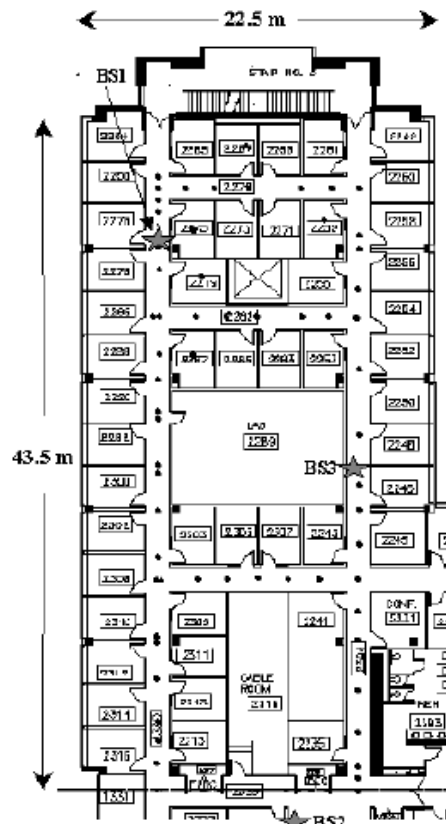


Figura 38 Mapa de un edificio que implementa el sistema RADAR con 3 estaciones base señales por una estrella.

Nibble

Sistema de localización para interiores desarrollado por el Departamento de las Ciencias de la Computación de la Universidad de California, se basa en el uso de puntos de acceso y móviles que cuentan con la tecnología Wi-Fi (estándar IEEE 802.11), para la localización usa el método de reconocimiento de patrones con Red Bayesiana. “Nibble es una versión independiente de un servicio de localización Wi-Fi que se deriva de un servicio de localización en interiores prototipo dentro de la infraestructura generalizada de sensores MUSE.” (Castro, Chiu y Kremenek)

Para obtener una buena localización de un móvil, usando como parámetro el RSSI obtenido entre este y los diferentes puntos de acceso de la infraestructura que forma al sistema, Nibble se auxilia del uso de un servicio de fusión³¹ para evitar los posibles cambios que sufre de la RSSI y así obtener una buena aproximación de la localización real, esto lo logra el sistema por medio de una Red Bayesiana denominada Red de Creencias o Red de Causalidad.

Este sistema proporciona una gran flexibilidad en cuanto al agregado o eliminación de puntos de acceso para una reconfiguración del sistema, sin embargo esto implica una re-calibración del sistema en sí. Es decir, es necesario realizar nuevamente la fase de entrenamiento para la Red Bayesiana. La calibración del sistema en sí lleva un largo periodo de tiempo ya que se necesita de la toma de muestras en las localizaciones deseadas durante diferentes días para lograr un buen entrenamiento de la Red.

En teoría el sistema es capaz de distinguir n^s diferentes lugares dentro de una red Wi-Fi, donde n es el número de valores de salida de un sensor y s es el número de puntos de acceso. Sin embargo, la implementación del sistema en dos edificios del campus de UCLA (Figura 39) arrojó como resultado en un caso $n=4$ y $s=10$ (para el edificio denominado Boelter Hall), mientras que en el otro caso el resultado fue de $n=4$ y $s=4$ (para el edificio denominado FXPAL), esto debido a la arquitectura de los edificios que limitan e interfieren con las señales usadas por el sistema.



Figura 39 Cuarto principal de conferencias del edificio FXPAL donde se implemento el sistema Nibble.

El sistema realiza la localización por medio del establecimiento de los valores de la probabilidad a priori $p(L)$ de estar en una localización L y probabilidad condicional $p(E|L)$ de recibir un vector de RSS E estando en la localización L . En general, el sistema tiene una exactitud del 90%.

³¹ Es un servicio forma por dos componentes: 1) un modelo de razonamiento probatorio que agrega e interpreta la información de los sensores (fusibles); y 2) un modelo de optimización de recursos para minimizar el costo computacional de la recopilación de datos

Posicionamiento Basado en Redes Inalámbricas para la determinación de Funcionalidades en un GIS

Sistema de localización prototipo para interiores y exteriores (siempre que el usuario esté dentro del rango de alcance) desarrollado en el Laboratorio de Geoprocesamiento del Centro de Investigación en Computación en el Instituto Politécnico Nacional (IPN), se basa en el uso puntos de acceso y móviles que cuentan con la tecnología Wi-Fi (estándar IEEE 802.11) y usa para la localización, el método de reconocimiento de patrones con redes neuronales (Memoria Asociativa³²).

Este sistema utiliza los RSSI provenientes de diversos puntos de acceso que estén al alcance de un dispositivo móvil para obtener la localización por contextos del usuario, es decir proporciona la localización, identifica y describe las características de las diferentes áreas sobre las que se ésta posicionado.

Se basa en la integración de diversas tecnologías, como son redes inalámbricas, dispositivos móviles, Sistemas de Información Geográfica (SIG), sistemas multiagente (Multi Agent Systems, MAS) y algoritmos de aprendizaje con el fin de desarrollar un Sistema de Cómputo Sensible al Contexto Geográfico con lo que es posible ubicar geográficamente dispositivos móviles utilizando las señales de radiofrecuencia (RF) de una red inalámbrica. (Montiel Hernández)

Para el entrenamiento del sistema, hace uso del aprendizaje supervisado en memorias asociativas, usando como conjunto fundamental de entrenamiento una serie de clases (ubicaciones) e intensidades RF de un número definido de ubicaciones conocidas por el usuario, lo cual permitirá construir un modelo que especifique la memoria de manera adecuada, una vez que el modelo ha sido construido las mediciones obtenidas se asocian a una clase con lo cual es posible hacer el posicionamiento local.

Como caso de prueba el sistema se implementó en el último piso del edificio del Centro de Investigaciones en Computación (CIC) del IPN (Figura 40), los resultados arrojados por el sistema muestran una exactitud del 41.6%.

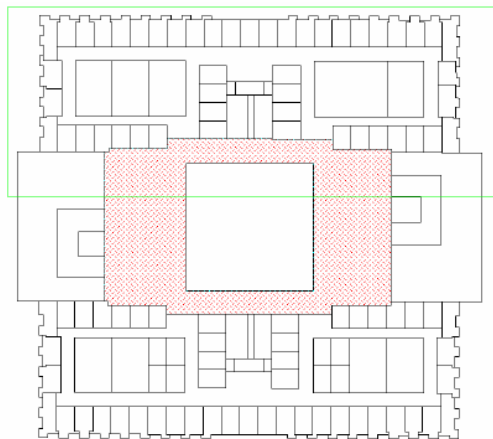


Figura 40 Diagrama del 2º piso del CIC donde el área de pruebas está encerrada en un cuadro verde.

³² Es una de las redes neuronales artificiales más importantes, tiene un amplio rango de aplicaciones en áreas tales como: Memorias de acceso por contenido, identificación de patrones y control inteligente. Puede almacenar información y recuperarla cuando sea necesario, es decir, una red retroalimentada, cuya salida se utiliza repetidamente como una nueva entrada hasta que el proceso converge, se pueden encontrar dos tipos de estas memorias asociativas: Memorias heteroasociativas y Memorias autoasociativas.

Utilidades y Aplicaciones de los Sistemas de Localización para Interiores

Los sistemas de localización para interiores en la actualidad cuentan con una diversidad aplicaciones y usos dentro de diferentes áreas del mercado, algunas de las más importantes son:

- **Industria y transporte:** Permiten la organización de almacenes e inventarios, así como estrategias comerciales de servicios de mensajería y publicidad interactiva dentro del transporte público. Por ejemplo, se puede hacer un seguimiento de los empleados y de los activos con el fin de hacer más eficiente la producción y supervisión, mejorar la seguridad de los empleados y reducir los errores en las entregas. Por otra parte se puede localizar a usuarios o compradores por medio de un subsistema incorporado al transporte público con el fin de orientar procesos de marketing o prestación de servicios para los consumidores de un mercado en específico y con ello lograr mayores ventas y una fidelidad del consumidor al proporcionar estos servicios preferentes.
- **Gestión de seguimientos:** Permiten analizar un seguimiento pasivo de rutas con el fin de observar las acciones y caminos seguidos por un usuario o empleado, durante la realización de una tarea ya sea en tiempo real o posteriormente, con lo cual se puede mejorar el desempeño y reducción del tiempo empleado en la misma. Esto se puede lograr mediante eliminación de acciones no esenciales y/o reducción de las distancias seguidas por medio de caminos alternos y/o reubicación de equipo y material.
- **Logística:** Permiten el seguimiento de equipo y personal dentro de almacenes, centros logísticos y otras áreas con una gran variedad de productos y de información que se mueven constantemente adentro y fuera del recinto, con lo cual se puede conseguir una ventaja competitiva. Esto se puede lograr al disminuir el tiempo que toman los inventarios, aumentando la productividad y seguridad de los empleados, mediante la localización de los mismos.
- **Prestación de servicios adicionales:** Permiten prestar servicios al cliente o personal de una empresa que por lo general no son provistos por otras empresas, o de ser así se requiere del uso de un área o personal específico, permitiendo un plus para la empresa que mejora la funcionalidad de la misma y la fidelidad y preferencia por parte del cliente. Por ejemplo, cuando un recepcionista recibe una llamada importante para su jefe y no se encuentra, lo puede ubicar por medio de su PDA, Smartphone o cualquier otro dispositivo electrónico que lleve consigo y que interactúe con el sistema permitiendo su ubicación, informarle sobre la situación y transferir la llamada al teléfono más cercano. Otro ejemplo es el uso del sistema dentro de un almacén, tienda, supermercado o lugar de compras que permite a los clientes mediante la instalación del sistema en sus PDA, Smartphone u otro dispositivo electrónico portátil, conocer la ubicación, precio, características e incluso garantías de un artículo, permitiendo incluso la creación de perfiles para clientes frecuentes que permitan realizar sugerencias al cliente sobre nuevos productos y descuentos de acuerdo a su perfil.
- **Servicios sanitarios y de emergencia:** Permiten el seguimiento automático y localización de pacientes, personal de salud y equipo médico dentro de los edificios. De esta forma los hospitales aumenta el rendimiento en el cuidado del paciente mejorando la información sanitaria del mismo, al mismo tiempo se obtiene una respuesta más rápida y eficiente de los servicios de emergencia.
- **Servicios de evacuación y rescates:** Permiten alertar al personal de un edificio o construcción sobre un desastre ocurrido dentro del mismo, con el fin de facilitar la evacuación y de ser necesario la ubicación de personal y su equipo. Por ejemplo, en caso de que ocurra un incendio, temblor, ruptura de una tubería de gas, u otro tipo de desastre dentro de un edificio el sistema serviría de apoyo a sistemas de alarma convencionales como son timbres; en caso de que esos sistemas no fueran escuchados por el personal, el sistema de apoyo permitiría informar sobre el desastre a otras personas por medio de dispositivos electrónicos, y en caso de ser necesario se podría ubicar a personal atrapado dentro del edificio, de igual forma se podría localizar equipo muy costo o difícil de reemplazar durante o después del desastre.

- **Servicios de activación automática:** Permiten activar y/o usar aplicaciones de forma automática cuando es localizado un usuario dentro del área de cobertura del sistema. Estos servicios automatizados se utilizan en los edificios inteligentes o automatizados donde la domótica³³ sobresale por encima de todo, aportan más rapidez y más eficiencia en las acciones necesarias por parte de los edificios así como los usuarios.

Lenguajes de Programación para Dispositivos Móviles y Embebidos

Dentro de la programación orientada a dispositivos móviles y embebidos existen principalmente los siguientes lenguajes y plataformas de programación:

- **C/C++:** “C evolucionó de dos lenguajes de programación anteriores, BCPL, y B.” (Deitel y Deitel) El desarrollo del lenguaje C fue dentro de los Laboratorios Bell de AT&T, Dennis Ritchie fue el encargado de esto y cuyo trabajo fue implementado en 1972. “El lenguaje C++ apareció oficialmente en 1983, fecha de las primeras utilidades fuera del laboratorio que lo vio nacer, AT&T.” (Guérin) Este lenguaje es uno de los más conocidos y usados en general por los programadores en todo el mundo, para la programación estructural se cuenta con ambos variantes de los lenguajes C/C++, por otro lado, el verdadero potencial de C++ se observa en la programación orientada objetos. Por lo general para evitar problemas con las diferentes Sistemas Operativos que dan soporte al lenguaje, la programación se basa en el estándar ANSI. A continuación se mencionan algunas de las ventajas y desventajas que presenta este lenguaje en el sentido de la programación de móviles:
 - **Ventajas:**
 - Amplio conocimiento del lenguaje por parte de los programadores.
 - Optimización en tiempo de ejecución de las aplicaciones desarrolladas debido a su alto nivel de desempeño y bajo consumo de recursos de cómputo por parte del lenguaje.
 - **Desventajas:**
 - Falta de estandarización de APIs (APIs de GUI's, Bluetooth, Wi-Fi, etc).
 - Limitaciones que presentan los diversos compiladores existentes (Codewarrior, Microsoft Visual C++ .Net, Microsoft eMbedded Visual C++, PocketGCC, etc) en la actualidad (los comerciales son los más completos).
 - Generación de ejecutables por lo general sólo para un tipo de Sistema Operativo (Symbian OS Nokia, Windows CE, Mobile Linux o Palm OS) en específico.
- **C#:** “Es un lenguaje de programación orientado a objetos y visual desarrollado por Microsoft como parte de su iniciativa .NET. Microsoft basó C# en C++ y Java y fue diseñado para combinar potencia y velocidad de programación.” (Hernández Sánchez y Blázquez Román) Fue presentado por Microsoft en el año 2000 como lenguaje nativo de su plataforma .Net siendo Scott Wiltamuth y Anders Hejlsberg sus principales colaboradores y a los que se reconoce como padres del mismo. En el año 2001 se obtuvo la aprobación como un estándar de programación por parte de la ECMA y en el 2003 por parte de ISO. En lo referente a la programación de dispositivos móviles y embebidos con este lenguaje, se logra a partir de la arquitectura de software denominada .Net Compact Framework derivada de .Net Framework desarrollada por Microsoft. La cual dispuso de la capacidad de programación para su primera versión por medio del entorno de programación Visual Studio .Net 2003. A continuación se mencionan algunas de las ventajas y desventajas que presenta este lenguaje en el sentido de la programación de móviles:
 - **Ventajas:**
 - Facilidad de uso por parte de los programadores C# debido a la estructuración del .Net Compact Framework con dos componentes principales: el Tiempo de Ejecución en Lenguaje Común (CLR³⁴, *Common*

³³ Sistemas capaces de automatizar un edificio y pueden estar integrados por medio de redes interiores y/o exteriores de comunicación, cableadas y/o inalámbricas. Aportan servicios de gestión energética, seguridad, bienestar y/o comunicación dentro de un edificio hacia el usuario.

³⁴ Es el entorno de ejecución en el que se cargan las aplicaciones desarrolladas en los distintos lenguajes, ampliando el conjunto de servicios que ofrece el sistema operativo estándar Win32. Además de la integración de lenguajes, también se

- Language Runtime*) y las API's .Net (de las cuales se implementan un 30%) por lo que su desarrollo es similar al desarrollo de aplicaciones .Net para equipos de escritorios.
- Lenguaje visual que facilita la creación de todo tipo de interfaces basándose en el uso de sus componentes gráficos (con sólo unas pequeñas modificaciones).
 - Desventajas:
 - Ejecutables que solamente trabajan con Sistemas Operativos (Symbian S60 2ª Edición o superior y Windows Embedded, Mobile, Automotives o CE) que implementan el .Net Compact Framework lo que hace que su uso sea más reducido por los fabricantes de móviles debido al costo que conllevan.
 - Java: Surge en 1991 como un proyecto de investigación de la compañía Sun denominado Green que se enfoca en la programación de dispositivos electrónicos inteligentes de uso doméstico. "El proyecto desembocó en el desarrollo de un lenguaje basado en C y C++, al cual, James Gosling llamó Oak, debido a un roble que tenía a la vista desde su ventana en la oficinas de Sun." (Deitel y Deitel) Sin embargo, posteriormente Sun descubriría la existencia de un lenguaje con este nombre, por lo que cuando un grupo de la gente de Sun visitaba una cafetería se le ocurrió denominarlo Java. A pesar de lo cual, el lenguaje no tuvo el éxito esperado sino hasta que se vio la popularidad del Red Global Mundial (WWW, *World Wide Web*) en 1993 y que se ajusta de forma óptima para el desarrollo en el lenguaje Java. En Mayo del mismo año Sun lanza oficialmente Java junto con el Kit de Desarrollo Java Ver. 1.02 (JDK, *Java Development Kit*). Debido a la creciente popularidad de los dispositivos móviles en los años posteriores Sun lanza en 1999 la primera versión de Java 2 Micro Edition con una máquina virtual denominada KVM con una única API (diseñada para Palm OS), sin embargo, al ser una solución ineficiente por la gran variedad de dispositivos móviles en el año 2000 se genera una configuración estándar denominada Configuración de Dispositivos de Conexión Limitada Ver 1.0 (CLDC, *Connected Limited Device Configuration*). A continuación se mencionan algunas de las ventajas y desventajas que presenta este lenguaje en el sentido de la programación de móviles:
 - Ventajas:
 - Reducción en la complejidad del desarrollo de aplicaciones por parte del programador debido al nivel de abstracción con que cuenta el lenguaje J2ME.
 - Amplia gama de Sistemas Operativos (Palm OS, Symbian OS Nokia, Mobile Linux, Windows Mobile o CE y Solaris Operating Environment) para dispositivos móviles que le dan soporte a lenguaje J2ME.
 - Por tratarse de un lenguaje gratuito y con tendencias OpenSource se cuenta una gama de entornos de desarrollo y compiladores comerciales y gratuitos (NetBeans, Eclipse Europa, Sun Java (TM) Wireless Toolkit, MOTODEV Studio for JAVA ME, etc).
 - Se cuenta con un conjunto estándar de API's propias bien definida y documentada.
 - Diversidad de APIs complementarias en desarrollo continuo debido a la gran aceptación por parte de los programadores, tanto por proyectos de grupos de desarrollo independientes como por empresas particulares del desarrollo de software entre las cuales podemos destacar los proyectos McObject's y phoneME.

CAPITULO II: Marco Teórico

- Desventajas:
 - Reducido número de Interfaces y Clases con que cuenta J2ME respecto a Java, lo cual conlleva a un reducido número de posibles caminos para el programador.
 - Las reducciones implementadas por J2ME provoca que muchas acciones o eventos sean ocultados para la programación de móviles y por lo cual se hace difícil o imposible su acceso.
 - Ejecutables que solamente trabajan con los Sistemas Operativos que contengan una implementación CDC o CLCD de J2ME.

Perfil Bluetooth

Son una serie de normas emitidas por el grupo Bluetooth SIG que permiten garantizar la interoperabilidad y proporcionar apoyo a aplicaciones específicas, con el fin de que los diversos fabricantes de dispositivos Bluetooth las implementen y así evitar que surjan conflictos entre la gama de dispositivos existentes en el mercado, cuando se desee crear un canal de comunicación entre los mismos.

Un perfil puede describirse como una rejilla vertical entre las capas de la pila de protocolos Bluetooth, una interfaz entre dos dispositivos, define qué opciones son obligatorias en cada protocolo de acuerdo al propio perfil usado. Un nuevo perfil puede ser implementado sobre los ya existentes, permitiendo la reutilización de protocolos y procedimientos existentes.

Para todos los perfiles las definiciones fundamentales, recomendaciones y requerimientos relativos a los modos de operación e instalación de conexiones y canales se basan en el Perfil de Acceso Genérico (GAC, por sus siglas en inglés). Así los perfiles definen los protocolos necesarios y otros perfiles de apoyo (tales como la forma de obtener el acceso general a otros dispositivos Bluetooth) obligatorios para ofrecer a los usuarios tipos de servicios específicos o determinadas características.

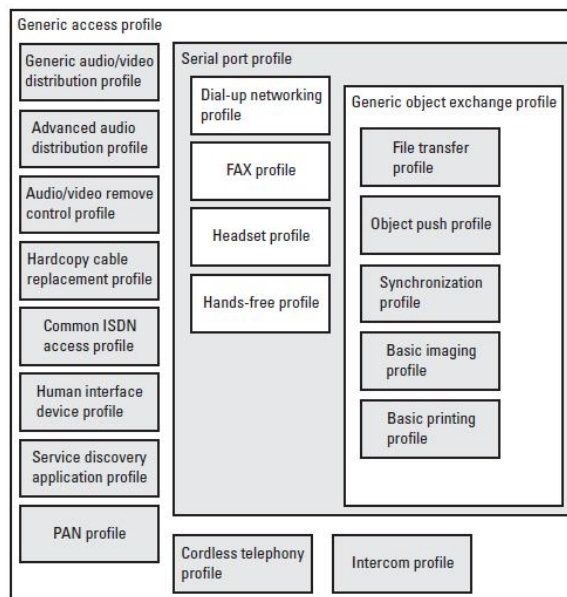


Figura 41 Perfiles Bluetooth.

Aunque diferentes servicios requieren distintos requisitos de seguridad, la mayoría de los perfiles se benefician del uso de las funciones de seguridad de la Banda Base, por lo cual es de gran importancia que los desarrolladores de aplicaciones entiendan y comprendan las fuerzas y limitaciones de los servicios de seguridad a nivel del Manejador de Enlace.

APIs Bluetooth para desarrollo en C/C++

El desarrollo de aplicaciones basadas en C/C++ para dispositivos móviles y/o embebidos que permitan la comunicación por medio de la tecnología Bluetooth ha tenido una gran discrepancia en cuanto a su portabilidad para diversos sistemas operativos y aparición de las API's de desarrollo para los mismos. Dentro de las APIs Bluetooth para desarrollo en C/C++ se encuentran las siguientes:

- **Bluetooth Egin API:** Una API estándar abierta para programación en lenguaje C/C++ para los dispositivos cuyo sistema operativo base es Symbian S60 2ª edición o superior que da soporte a los protocolos Bluetooth RFCOMM, L2CAP y OBEX, la primera especificación de esta API se dio a conocer el 27 de Agosto de 2004. La API está diseñada conforme a la especificación de los protocolos Bluetooth y su estructura general de uso y programación está dividida en los siguientes módulos:
 - *Sockets Bluetooth: Encapsulan el acceso a L2CAP Y RFCOMM a través de una interfaz de socket semejante a TCP/IP.*
 - *Base de Datos de Descubrimiento de Servicios Bluetooth: Encapsula un lado SDP. Un servicio local utilizado para generar sus atributos, de modo que los dispositivos remotos puedan descubrir su presencia y determinar si puede ser utilizado.*
 - *Agente de Descubrimiento de Servicios Bluetooth: Encapsula el otro lado de SDP. Permite al usuario descubrir qué servicios están disponibles en un dispositivo remoto, así como los atributos de esos servicios.*
 - *Manejador de Seguridad Bluetooth: Permite establecer los requisitos de servicios de seguridad adecuada, que deben de cumplir las conexiones entrantes.*
 - *UI Bluetooth (también llama Selección de Dispositivos de IU Bluetooth). Provee de una API que mediante un cuadro de diálogo que pide a los usuarios la información del dispositivo seleccionado que puede ser llamado.* (Nokia Corporation)

Las funciones principales de la pila de esta API tanto de sus clases como interfaces están agrupadas en un conjunto de librerías dentro de las cuales se pueden encontrar *bteng.lib*, *bluetooth.lib* y *btdevice.lib* principalmente, de uso genérico para las diversas versiones disponibles de la API, además de contar con una serie de librerías extras con mayores funcionalidades de acuerdo con la versión de API soportada por el sistema operativo.

- **Microsoft Bluetooth API:** Está formado por un conjunto de APIs estándares propietarias (tanto independientes como extendidas) que proveen de una serie de Interfaces y Clases que se usan para el desarrollo de aplicaciones Bluetooth en el lenguaje C/C++ y que interactúan únicamente con las Pilas Bluetooth Microsoft y Widcomm/Broadcom, los dispositivos que las implementan poseen como mínimo las versiones de los Sistemas Operativo Windows CE 3.0, Windows Mobile 2003 o Symbain S60 2ª edición. Por lo general para el manejo de la pila Bluetooth en las aplicaciones a desarrollar se hace a través de las APIs extendidas tanto de la API Winsock como de la API IrDA que se encuentran soportadas principalmente por las librerías Winsock.lib, ws2.lib y Ircomm.lib. Un punto importante a destacar es que si se opta por el desarrollo a través de la API extendida de Winsock se debe de tomar en cuenta la versión (Ver. 1.1 que trabaja con Winsock.lib o 2.0 que trabaja con Ws2.lib) instalada sea la misma en el Sistemas Operativo ya que ciertas funcionalidades son exclusivas de la versión más reciente de Winsock. El desarrollo de las aplicaciones están estrechamente relacionado con el SDK que sea usado por el programador, lo que conlleva a la incompatibilidad entre las aplicaciones desarrolladas, ya que se encuentran ligadas al tipo de pila Bluetooth (Microsoft o Widcomm) usada por el SDK. Para tratar de dar una solución a este problema Microsoft creó una API Blueooth que se incluye a partir de los SDK para desarrollo de las versiones de los Sistemas Operativos Windows CE/Mobile 5.0 soportada principalmente por las librerías Btd.lib, Bthguid.lib, Bthutil.lib, Coredll.lib y Ws2.lib, esta última como librería de apoyo que diferencia con los SDK anteriores, sólo interfiere en algunas etapas de la comunicación por Bluetooth con lo que se evita la dependencia que existía entre la pila Bluetooth y el SDK.

- **API Bluetooth Library (BtLib):** Es API estándar abierto diseñado por Palm Inc. de acuerdo con la especificación 1.1 de protocolo Bluetooth, surge a partir de la decisión de Palm Inc. de incluir en Palm OS 4.0 el soporte para los 5 perfiles Bluetooth y que fue dado a conocer el 19 de Marzo de 2001. Esta API es asíncrona, por lo que API retorna inmediatamente en lugar de bloquearse a esperar una respuesta del dispositivo remoto o del módulo de radio local.^[26] La API esta principal soportada por la librería BtLib.h que está diseñada para trabajar sobre el lenguaje C/C++ y que nos permite trabajar con los protocolos Bluetooth RFCOMM, L2CAP y OBEX. El desarrollo de aplicaciones para móviles basadas en el lenguaje C/C++ solamente cuenta con entornos de desarrollo y compiladores comerciales (Palm OS SDK 4.0 o superior, Codewarrior IDE, etc) que dan soporte a esta API. Existen dos forma principalmente para desarrollar aplicaciones con base en este API:
 - Utilizando el Controlador Bluetooth Virtual Serial permite a las aplicaciones basadas en serial existente rápidamente ponerse al día para tomar ventaja de la tecnología Bluetooth, y es una manera fácil de crear nuevas aplicaciones Bluetooth activas.^[8] Para esto como parte de los paquetes soportados con Palm OS se incluye el Bluetooth VDRV para interacción con Manteador Serial, en base a este paquete se pueden desarrollar aplicaciones que funcionan sobre el protocolo Bluetooth RFCOMM o L2CAP.
 - “Si se está interesado en utilizar la tecnología Bluetooth para transferir archivos, o si tiene un flujo de datos constante esto no es importante para su aplicación (como en un juego basado en turnos), la biblioteca de Intercambio de Bluetooth podría ser la herramienta perfecta para su uso.” (Kammer, McNutt y Senese) Esta librería permite el envío de bloques de dato sin tomar demasiada importancia al transporte fundamental, para ello se cuenta con un Manejador de Intercambios concepto único de Palm OS que permite la comunicación entre dispositivos por medio del protocolo OBEX, esto se debe a que es el protocolo que mejor se adecúa al ancho de banda que se necesita.

APIs Bluetooth para desarrollo en C#

En lo referente al desarrollo de aplicaciones Bluetooth con C#, al ser este el lenguaje nativo del entorno de desarrollo .Net, este entorno proporciona el API propietario Win32 Wrapper para el desarrollo de las aplicaciones deseadas por medio del programa y API complementario gratuito *Windows Embedded Source Tools for Bluetooth Technology*. Las aplicaciones desarrolladas bajo esta API pueden ser generadas y compiladas por medio del entorno comercial Microsoft Visual Studio, ejecutándose bajo la arquitectura .Net Compact Framework lo que permite una compatibilidad en dispositivos con Sistemas Operativos a partir de las versiones Windows CE 5.0 o Windows Mobile 5.0. Esta API reduce la complejidad del manejo del protocolo Bluetooth para el programador mediante las clases envoltura que nos provee y que se encuentran principalmente soportadas por la librería dinámica Bluetooth.dll. Las clases podemos vivirlas principalmente en 3 tipos:

- Clases para crear servicios Bluetooth.
- Clases para enumerar dispositivos Bluetooth.
- Clases para conectarse a un dispositivo o servicio Bluetooth existente.

La API provee la capacidad de manejo de los protocolos de comunicación RFCOMM, L2CAP y OBEX; sin embargo hay algunas etapas (por ejemplo la búsqueda de dispositivos Bluetooth cercanos) de las aplicaciones que no pueden ser implementadas directamente con este API y requieren del uso de la API de sockets de .Net soportada principalmente por la librería dinámica ws2.dll. El requisito principal para el funcionamiento de las aplicaciones Bluetooth desarrolladas bajo esta API son la implementación de la pila Bluetooth de Microsoft en el dispositivo en el que se desea ejecutar la aplicación, ya que no existe compatibilidad entre esta API y la pila Bluetooth Widcomm a pesar de ser la otra opción de pila usada por los Sistemas Operativos Windows CE y Windows Mobile. Por otra parte existen una serie de programas y APIs complementarios alternativos tanto comerciales como gratuitos, desarrollados por empresas y grupos ligados a Microsoft o independientes del mismo, que permiten el desarrollo basado en la pila de Microsoft, Widcomm o ambas, entre las más destacadas se tienen las siguiente:

CAPITULO II: Marco Teórico

- 32feet.Net y desarrolloMobile.NET para desarrollo basado en la pila Microsoft.
- BTAccess y BTW-CE SDK para Windows Mobile Development Platform desarrollo basado en la pila Widcomm.
- BlueTools para desarrollo basado tanto en la pila Microsoft como Widcomm.

APIs Bluetooth para desarrollo en Java

El hardware Bluetooth ha avanzado mucho a un ritmo acelerado desde su aparición en 1998, pero no fue sino hasta hace relativamente poco que surgió alguna forma de desarrollar aplicaciones Java-Bluetooth; esto se logró con el surgimiento del JSR-82 en Marzo del 2002, que estandarizó la forma de desarrollar aplicaciones Bluetooth usando Java. Éste estándar permite ocultar la complejidad del protocolo Bluetooth, por medio de APIs de alto nivel que permiten centrarse en el desarrollo, en lugar de poner atención en los detalles de bajo nivel del Bluetooth. Estos APIs para Bluetooth están orientados hacia los dispositivos que poseen como mínimo las siguientes características:

- Al menos 512K de memoria libre (ROM y RAM, las aplicaciones necesitan memoria adicional).
- Conectividad a la red inalámbrica Bluetooth.
- Que tengan una implementación del J2ME CLDC.

JSR-82: El objetivo que perseguía esta especificación era el definir un API estándar abierto basado en los APIs J2ME y el entorno de desarrollo J2ME. Los APIs JSR 82 permiten una gran flexibilidad, trabajar con aplicaciones nativas Bluetooth y Java Bluetooth sin ningún problema.

Este API intenta ofrecer las siguientes capacidades:

- Registro de servicios.
- Descubrimiento de dispositivos y servicios.
- Conexiones RFCOMM, L2CAP y OBEX entre dispositivos.
- Usar las conexiones establecidas para mandar y recibir datos (las comunicaciones de voz no están soportadas).
- Manejar y controlar las conexiones de comunicación.
- Ofrecer seguridad en todas las actividades anteriormente establecidas.

Los APIs Java para Bluetooth están definidos dentro de dos paquetes que dependen del paquete CLDC javax.microedition.io:

- javax.bluetooth
- javax.obex

Implementaciones JSR-82 en J2SE

Las implementaciones JSR-82 existentes se dividen principalmente en dos grupos (Tabla 1): las APIs que implementan completamente la pila de Bluetooth y las que utilizan pilas de Bluetooth externas. Las APIs que implementan completamente la pila de Bluetooth ofrecen la ventaja de funcionar en cualquier sistema operativo pero tienen la desventaja de tomar el control del dispositivo Bluetooth, lo cual conlleva al bloqueo del acceso a otras aplicaciones. Por su parte las APIs que utilizan pilas de Bluetooth externas tienen la desventaja de que dependen del sistema operativo, sin embargo, tienen la ventaja de no bloquear el acceso al dispositivo Bluetooth para otras aplicaciones.

Tabla 1 Comparativa de implementaciones JSR-82 para J2SE

Nombre	Licencia	SO	Java	javax.bluetooth	javax.obex	Comentarios
<i>APIs de Java con pila de Bluetooth incorporada</i>						
Harald	Gratuito	Todas	Todas	No	No	Requiere javax.comm. Sólo soporta L2CAP
AveLink	Comercial	Todas	Todas	Si	Si	
JavaBluetooth	GNU	Todas	Todas	Si	No	Requiere javax.comm. Sólo soporta L2CAP
<i>APIs de Java que se ejecutan sobre pila de Bluetooth Externas</i>						
Avetana Bluetooth	Comercial	Windows XP, Linux, MacOS	J2SE	Si	Si	25 € las versiones de MacOS y Windows. Version GPL para Linux
BlueCove	GNU	Windows XP, Linux	J2SE	Si	No	Soporta L2CAP y RFCOMM
JBlueZ	GNU	Linux	J2SE	No	No	Sólo soporta descubrimiento básico
Impronto SDK	Comercial	Linux	J2SE	Si	Si	Gratis para uso académico

Harald: Es una pequeña pila de Bluetooth para la comunicación inalámbrica entre aplicaciones Java, que le permite el acceso al chip a través del puerto serie y, por lo cual, requiere la instalación del paquete JavaComm. La pila funciona en cualquier sistema donde se ejecuta Java, además es gratuito para cualquier uso y para su distribución.

AveLink: AveLink BT solutions y SDK es una implementación completa para protocolo de la pila Bluetooth, con perfiles para sistemas integrados Windows/WinCE y Java. Esta implementación promete a los fabricantes de hardware la capacidad de diseñar y producir un amplio grupo de productos móviles para el mundo real en un periodo de tiempo inferior y con un costo reducido.

JavaBluetooth: Es una pila en un 100% (no nativa) una implementación Java de la especificación Bluetooth versión 1.1 (en la actualidad soporta HCI, L2CAP y SDP). Fue desarrollado originalmente como parte de un proyecto de tesis para la realización de un Grado B. Sc en la Universidad Internacional en Alemania. Sin embargo, la pila es independiente del sistema operativo, ya que utiliza el paquete javax.comm para hacer frente a los chips Bluetooth.

AvetanaBT: Esta implementación del API JSR82, encapsula por medio de librerías java las llamadas al API de la pila Bluetooth del sistema operativo, por ejemplo, en Linux esto se consigue con el uso de JNI (Java Native Interface) y las librerías de la pila Bluetooth oficial de Linux (BlueZ). Al estar la pila BlueZ bajo licencia GPL, se puede obtener todo el código de AvetanaBT para Linux de forma gratuita. Esta implantación tienen varias mejoras al borde de las especificaciones JSR 82, también es posible su uso en los sistemas operativos Windows, Windows Mobile 2003, Windows Mobile 5.0 y MacOS X, por medio de una licencia de paga que se puede adquirir desde la página oficial de esta iniciativa³⁵.

BlueCove: Es una librería Java para Bluetooth (implementación JSR-82), en la actualidad trabaja con Mac OS X, Widcomm, BlueSoleil y la pila Microsoft Bluetooth que se encuentre en el SP2 de Windows XP o Windows Vista y en Windows Mobile. Recientemente se ha incluido BlueCove-gpl que es módulo adicional de BlueCove para apoyar a ésta en Linux, en este módulo se distribuyen como un JAR diferente por las diferencias de la licencia.

JBlueZ: Es un paquete Java que está en total interfaz con el protocolo de pila Bluetooth BlueZ para Linux, proporcionando un procedimiento sencillo para añadir funcionalidad Bluetooth a aplicaciones Java. Está diseñado para el soporte de Bluetooth de código abierto para J2SE y J2EE sin ningún tipo de dependencia de propiedad de terceros protocolos de pilas. JBlueZ fue desarrollado originalmente por la Appliance Studio en el Reino Unido. A medida que el proyecto y el interés en él desde fuera de la empresa crecieron, se decidió convertirlo en una actividad de código abierto.

³⁵ <http://www.avetana-gmbh.de>

Impronto Developer Kit: Es una herramienta de diseño basada en estándares Java para hacer fácil la construcción de aplicaciones de Bluetooth. Impronto Framework esconde complejos protocolos Bluetooth detrás de las API estándar de Java, permitiendo a los desarrolladores concentrarse en la escritura de aplicaciones inalámbricas en lugar de la creación de redes Bluetooth de bajo nivel, teniendo como resultado la creación más rápida y más fácil de aplicaciones de Bluetooth.

El Simulador de Impronto ejecuta aplicaciones Java-Bluetooth en un ambiente Bluetooth simulado, permitiendo a los desarrolladores probar y configurar fácilmente aplicaciones antes de implementar en los dispositivos Bluetooth. Dado que el Simulador Impronto ofrece total emulación por software, los desarrolladores pueden iniciar la programación JSR 82 sin la molestia de la adquisición y configuración de múltiples dispositivos Bluetooth dentro de su ambiente de desarrollo.^(Jode)

Conceptos básicos de modelado 3D

Espacio tridimensional

El espacio tridimensional (3D) es un espacio matemático virtual que es originado por el programa de diseño 3D, el cual está definido por un sistema cartesiano de tres ejes (X, Y, Z). El punto donde salen las líneas virtuales que definen los ejes (cruce de los 3 ejes) se denomina origen y sus coordenadas son: (0, 0, 0). Este espacio virtual será usado para crear, modificar y colocar los diferentes objetos tridimensionales que se usarán para armar la escena.

Elementos básicos del diseño 3D

En un principio para todo diseño 3D se cuenta con 4 elementos básicos, los cuales también son conocidos como dimensiones. A continuación se dará una breve descripción de cada uno de los elementos y la dimensión que representa:

- **Punto (Dimensión 0)**: Elemento fundamental de construcción para todo programa de diseño 3D, cuya posición está definida por un grupo de coordenadas 3D (X,Y,Z).
- **Línea (1^{ra} Dimensión)**: Elemento definido por 2 o más puntos siendo el recorrido existente entre estos, sin llegar a cerrarse entre si y crear una superficie.
- **Plano (2^{da} Dimensión)**: Elemento definido por la unión de 3 o más líneas que se unen entre y forman una superficie que posee área y volumen.
- **Objeto (3^{ra} Dimensión)**: También denominado volumen, es el elemento definido por la unión de 3 o más planos que se unen entre sí para formar un figura que posee volumen.

Modelado

El modelado consiste en ir dando forma a objetos individuales que posteriormente integrarán la escena. Existen una diversidad de técnicas de modelado, algunas de estas son:

- **Modelado con NURBS³⁶ (Non Uniform Rational B-Splines)**: Consiste en crear una malla³⁷ compleja que presenta a un objeto con un aspecto más curvado u orgánico a partir de splines³⁸ por medio de diferentes métodos (Figura 42).
- **Modelado por Estructuras Predefinidas**: Consiste en hacer uso de aquellos objetos (estructuras predefinidas) dados por el sistema con el fin de crear objetos (modelos) o escenas más complejas a partir de éstas (Figura 43 y Figura 44).
- **Modelado por Extrude**: Se basa en el hecho de expandir o contraer replicas que se generan de los puntos, líneas o planos que forman una malla en alguna de las coordenadas cartesianas (X, Y, Z) con el fin de obtener como resultado una malla más compleja (Figura

³⁶ Representaciones matemáticas de geometría en 3D capaces de describir cualquier forma con precisión, usadas para generar y representar curvas y superficies.

³⁷ Conjunto o redes de puntos interconectados entre sí.

³⁸ En el campo del diseño asistido por computadora, de los gráficos por computadora y en el modelado 3D, hace referencia a una curva (paramétrica) o línea flexible definida en porciones mediante polinomios.

CAPITULO II: Marco Teórico

45 y Figura 46), por lo general esta técnica es más usada para proveer de profundidad a un spline.

- **Modelado con Operaciones Booleanas:** Se basa en la idea de aplicar a dos mallas entre si una operación booleana (resta, intersección o unión) obteniendo como resultado una malla nueva y diferente a las originales (Figura 47 y Figura 48).
- **Modelado por Mallas de Polígonos:** Consiste en el diseño de mallas que cumple con la regla de que cada línea es compartida a lo máximo por dos planos (Figura 49 y Figura 50).
- **Modelador por B-Rep (*Boundary Representation*):** Es la representación de un objeto mediante la frontera del mismo, es decir, este objeto esta descrito en términos de los elementos de su superficie (puntos, líneas y planos) con lo cual queda separado el interior del exterior del objeto (Figura 51 y Figura 52).
- **Modelado con Texturas:** Especie de modelado, que consiste en el uso de imágenes como texturas y el canal Alpha, en vez de la aplicación de materiales sobre las mallas con el fin de tener un objeto simple que a la vista simule una estructura o forma más compleja (Figura 53 y Figura 54). Con esto ya no es necesario la deformación y/o generación de mallas complejas para tener efectos de cortes, relieve, rasgados, entre otros sobre un objeto, por lo que este tipo de modelado resulta sencillo de implementar

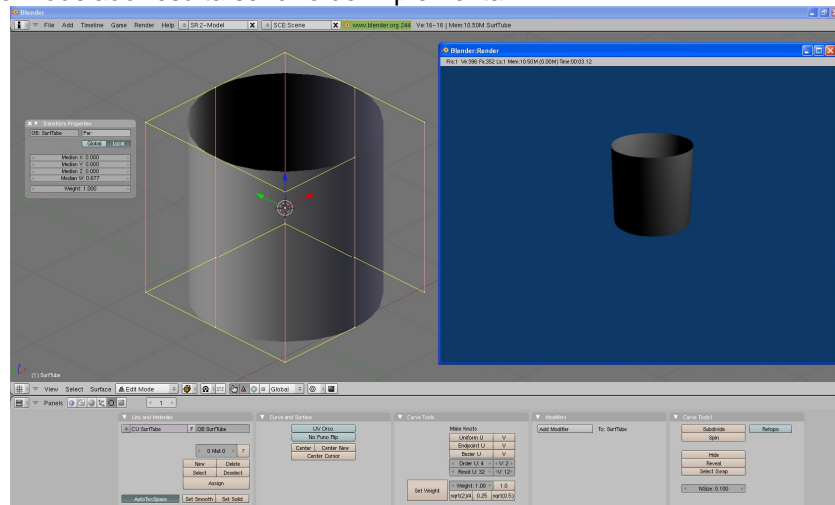


Figura 42 Ejemplo de un tubo generado en Blender mediante Modelado con NURBS.

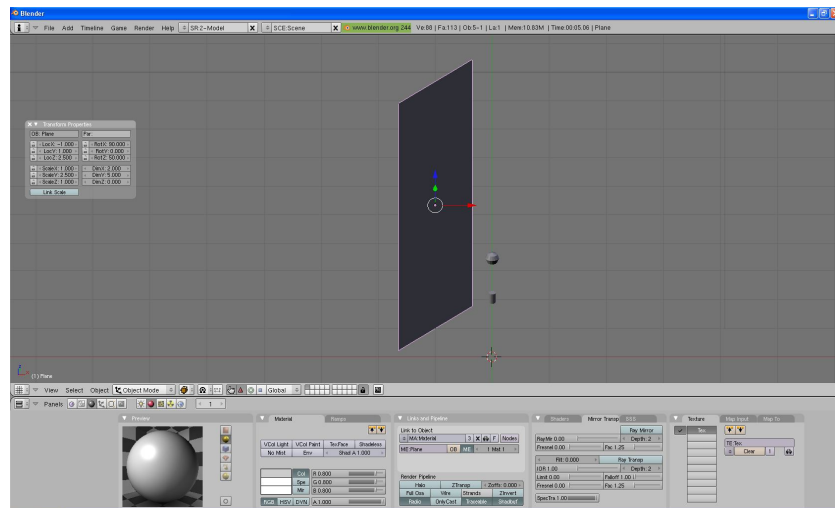


Figura 43 Diseño de una perilla en Blender mediante Modelado por Estructuras Predefinidas (aplicado a los objetos UVsphere y Cylinder).

CAPITULO II: Marco Teórico

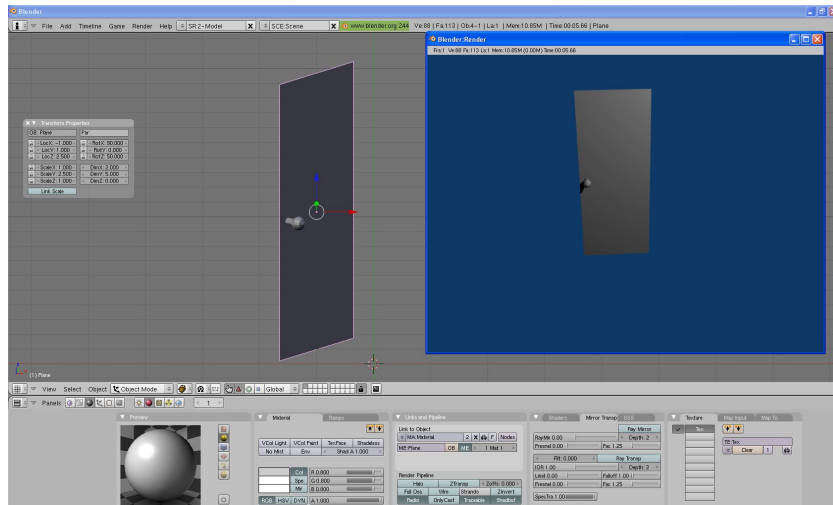


Figura 44 Ejemplo de la forma final de una perilla generada en Blender mediante Modelado por Estructuras Predefinidas (aplicado a los objetos UVsphere y Cylinder).

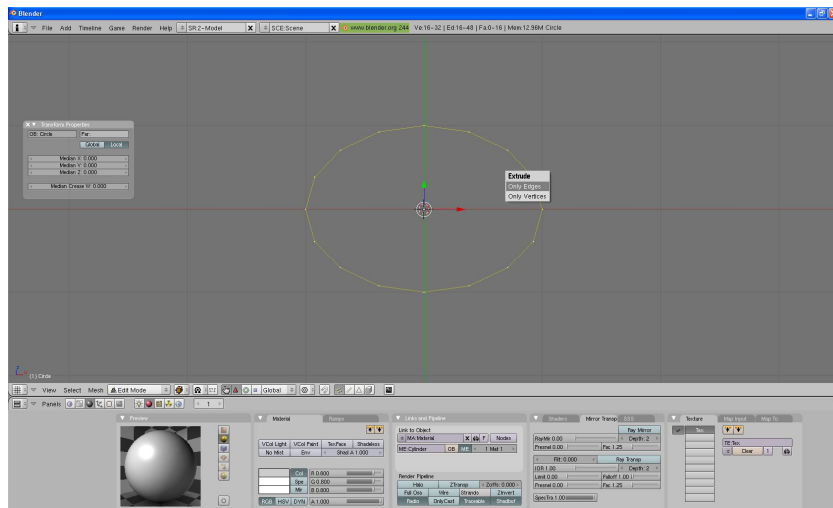


Figura 45 Diseño de una dona en Blender mediante Modelado por Extrude (aplicado a un plano tipo Circle).

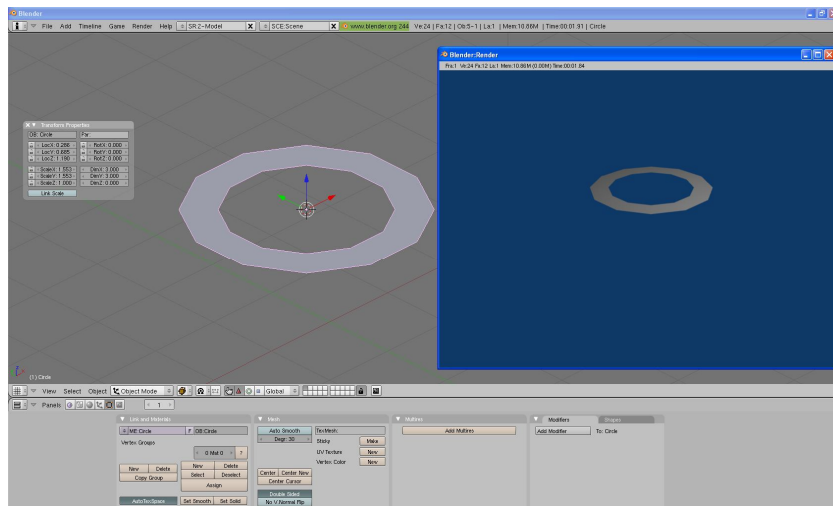


Figura 46 Ejemplo de la forma final de una dona generada en Blender mediante Modelado por Extrude (aplicado a un plano tipo Circle).

CAPITULO II: Marco Teórico

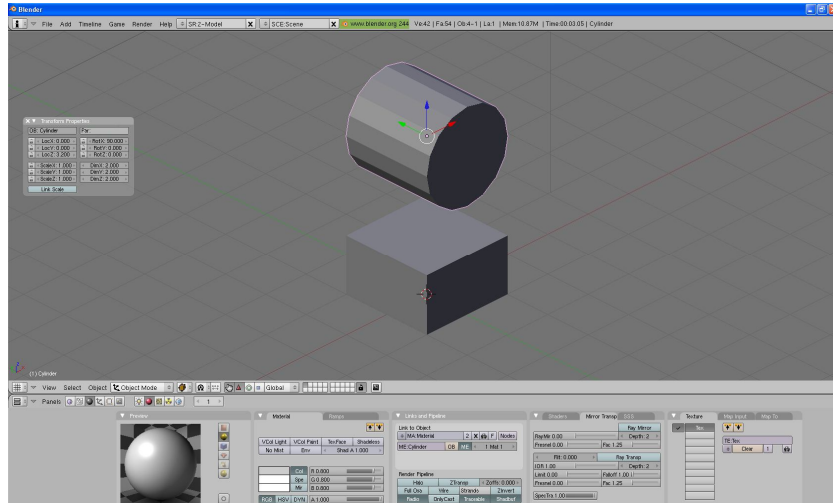


Figura 47 Diseño de un arco en Blender mediante Modelado por Operaciones Booleanas (aplicando a objetos tipo Cube una resta booleana de un objeto Cylinder).

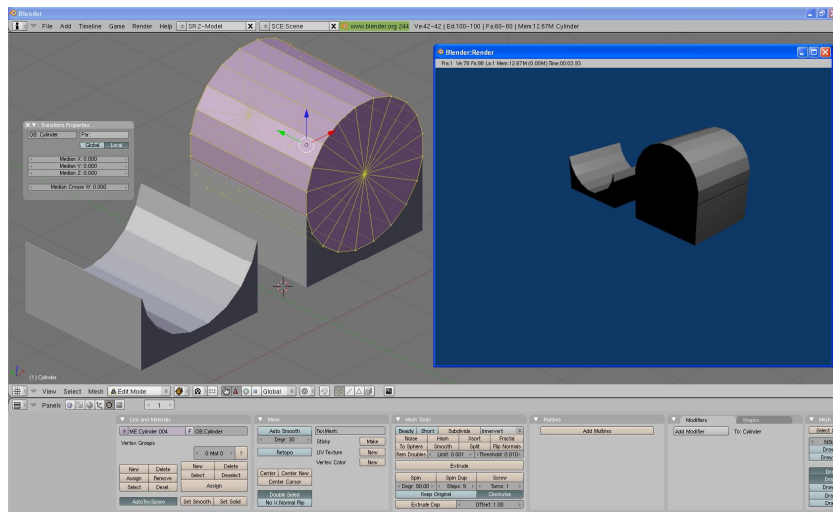


Figura 48 Ejemplo de la forma final de un arco generada en Blender mediante Modelado por Operaciones Booleanas (aplicando a objetos tipo Cube una resta booleana de un objeto Cylinder).

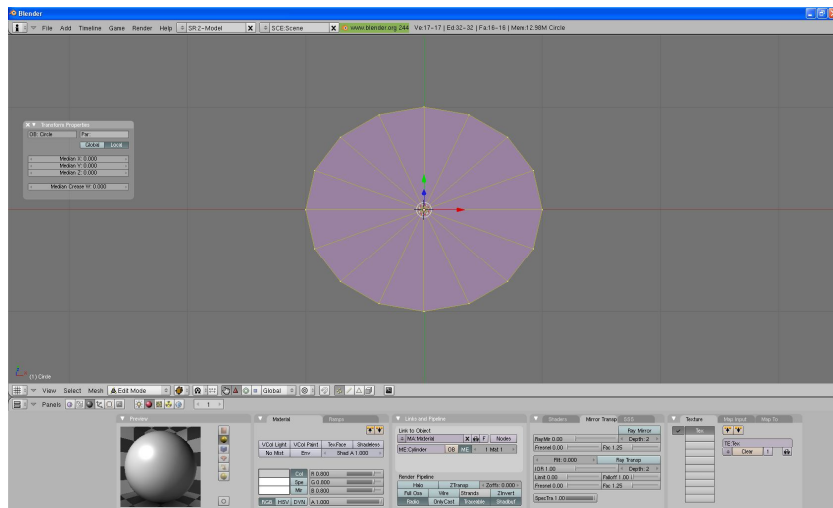


Figura 49 Diseño de un abanico en Blender mediante Modelado por Malla de Polígonos (aplicado a un plano tipo Circle).

CAPITULO II: Marco Teórico

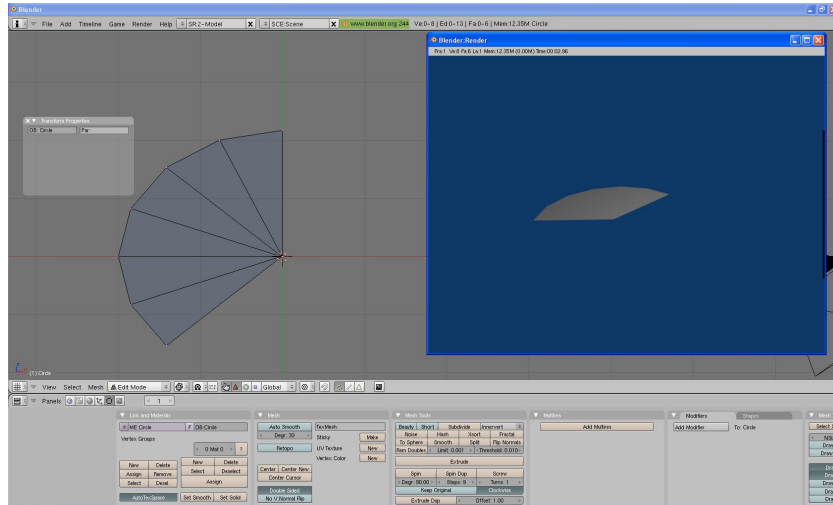


Figura 50 Ejemplo de la forma final de un abanico generada en Blender mediante Modelado por Malla de Polígonos (aplicado a un plano tipo Circle).

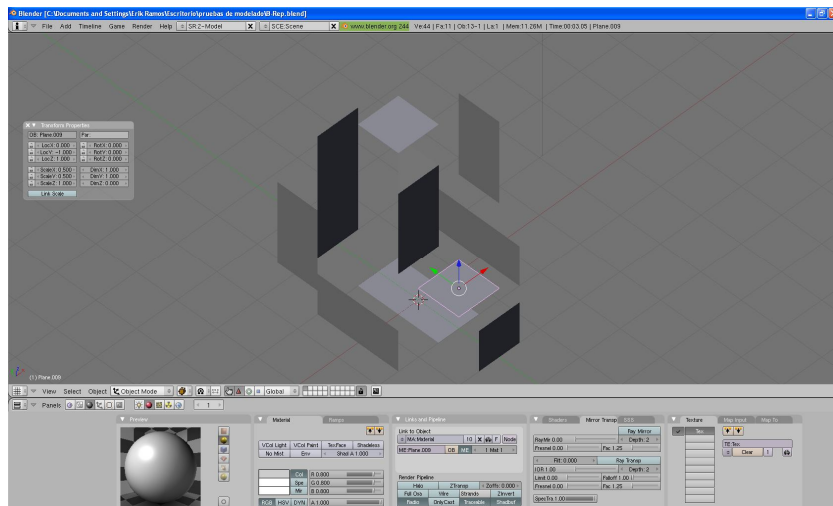


Figura 51 Diseño de una L en Blender mediante Modelado por B-Rep (usando planos como representación de fronteras).

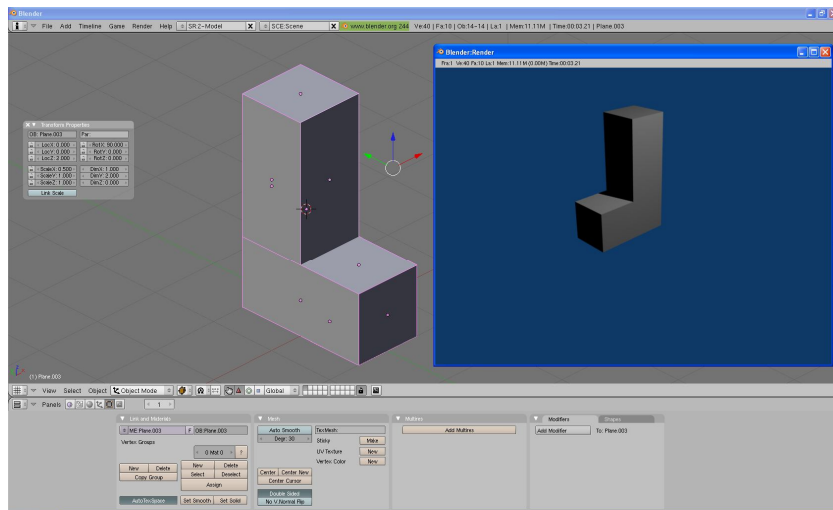


Figura 52 Ejemplo de la forma final de una L generada en Blender mediante Modelado por B-Rep (usando planos como representación de fronteras).

CAPITULO II: Marco Teórico

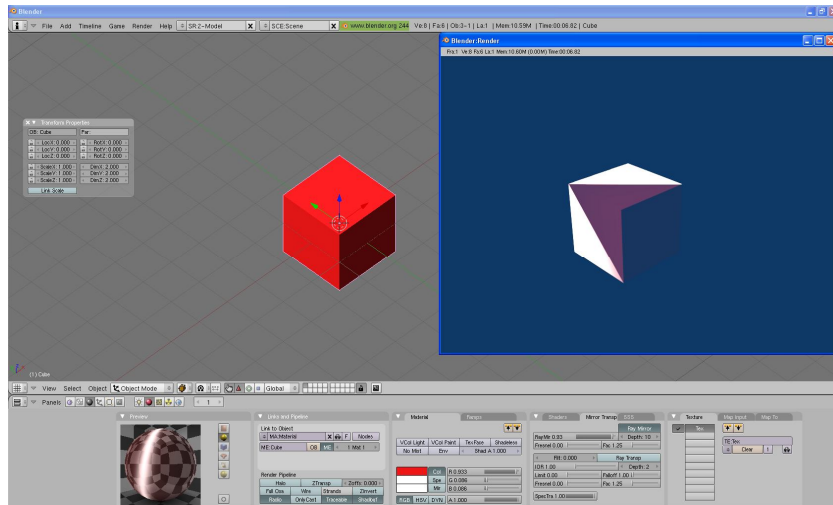


Figura 53 Ejemplo de un Cubo-Seccionado generada en Blender mediante Modelado con Texturas (usando objeto Cube con un material color rojo, canal Alpha activo y flexión).

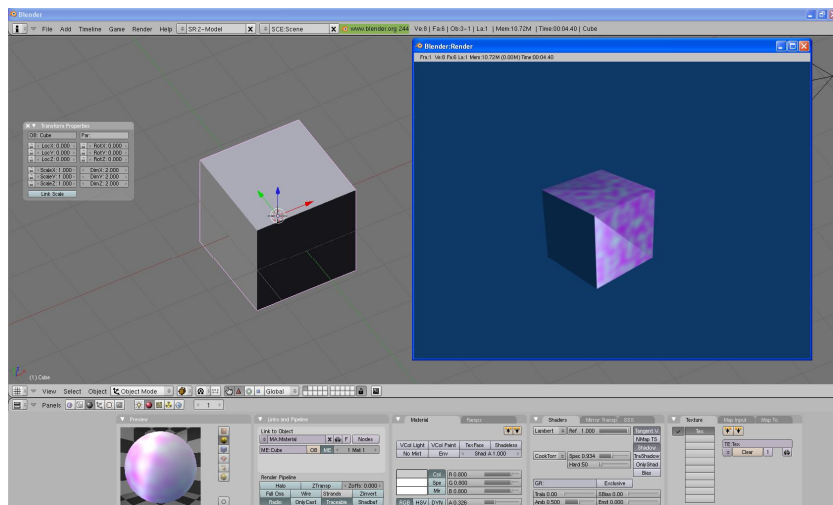


Figura 54 Ejemplo de un Cubo-Seccionado generada en Blender mediante Modelado con Texturas (usando objeto Cube con una textura Stucci y canal Alpha activo).

Algunas de las actividades que se incluyen durante los procesos de modelado son: la edición de la superficie del objeto o las propiedades del material (color, luminosidad, transparencia/opacidad, entre otras), agregado de texturas, el uso de mapas de relieve (bump-maps), entre otras características.

Durante el proceso de modelado se puede incluir actividades relacionadas con la preparación del modelo 3D para su animación, tal como la asignación de un esqueleto (estructura central con capacidad de modificar forma y movimientos del objeto al que se le asigna), esto ayuda al posterior proceso de animación, ya que el movimiento del esqueleto afectará de forma automática las secciones correspondientes del objeto. El modelado puede ser llevado a cabo por medio de una de las siguientes herramientas:

- Programas dedicados: Blender, Lightwave 3D, Rhinoceros 3D, Moray o Maya.
- Componente de una aplicación: Shaper o Lofter en 3D Studio.
- Lenguaje de descripción de escenas: POV-Ray.

Iluminación

Cada objeto que integra la escena tiene una intensidad de iluminación que depende del tipo de luz cercana al mismo y las características de la superficie del objeto (superficie brillante, opaca, transparente, etc), lo cual es de suma importancia si lo que se quiere es dar mayor realismo a las superficies por medio de un modelado de sombreado, ya que cuando un objeto es observado lo que se percibe es la cantidad de luz reflejada por la superficie del mismo. Las fuentes de luz que actúan sobre el objeto de una escena son de dos tipos:

- **Fuentes emisoras de luz:** Son las que generan su propia luz, entre las que se incluyen los focos y el sol. Por lo general en los programas de diseño y APIs 3D se incluyen 4 tipos de luces básicas: luz de ambiente global (simulación de la luz del sol, Figura 55), luz puntual (Figura 56), luz direccional (Figura 57) y luz de lugar (simulación de la luz de un foco, Figura 58).
- **Fuentes reflectoras de luz:** Son las generadas por las reflexiones de las superficies iluminadas por fuentes emisoras de luz de otros objetos cercanos al objeto a observar (Figura 59). Cuando las multiplex reflexiones de luz provenientes de objetos cercanos se combinan para producir una iluminación uniforme se dice que hay una luz ambiente o de fondo.

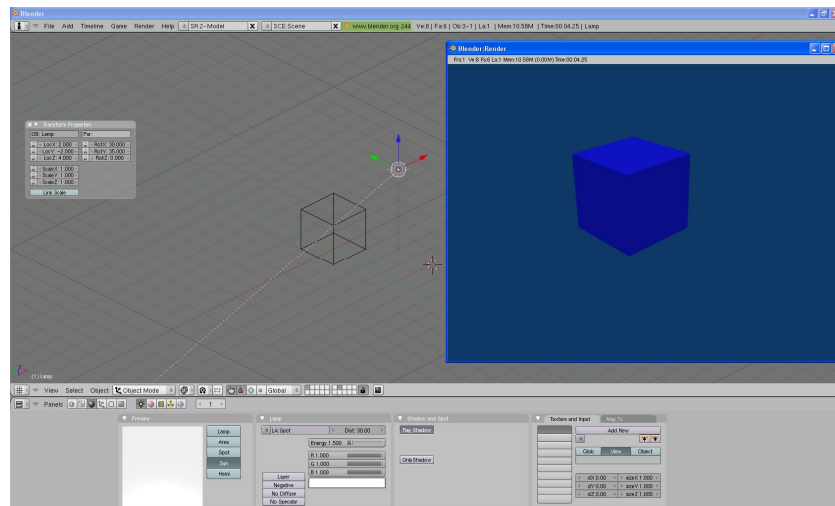


Figura 55 Ejemplo en Blender de Fuente Emisora de Luz (Luz de Ambiente Global).

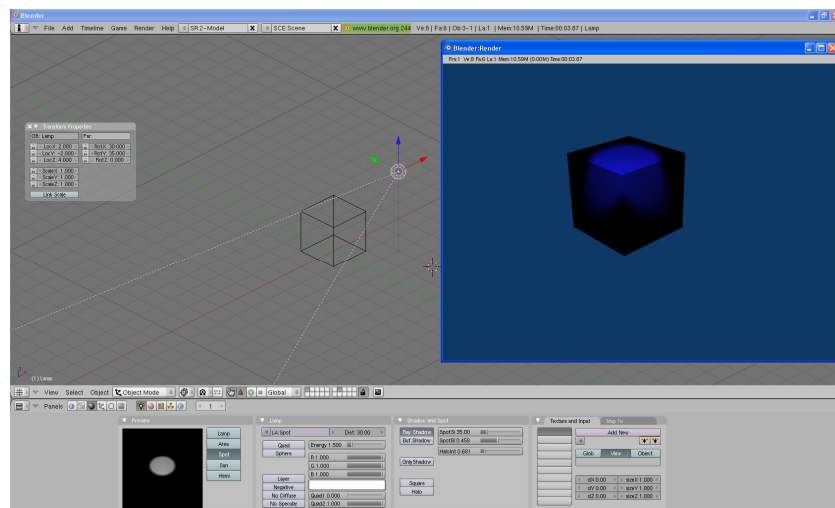


Figura 56 Ejemplo en Blender de Fuente Emisora de Luz (Luz Puntual).

CAPITULO II: Marco Teórico

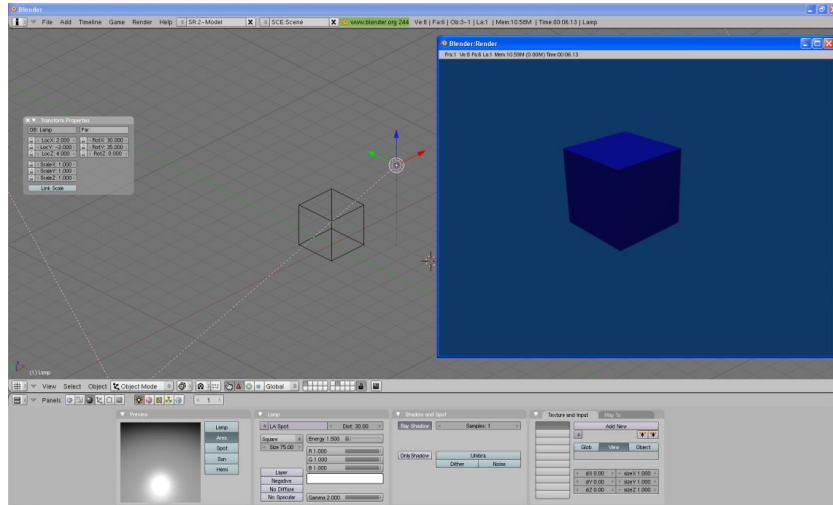


Figura 57 Ejemplo en Blender de Fuente Emisora de Luz (Luz Direccional).

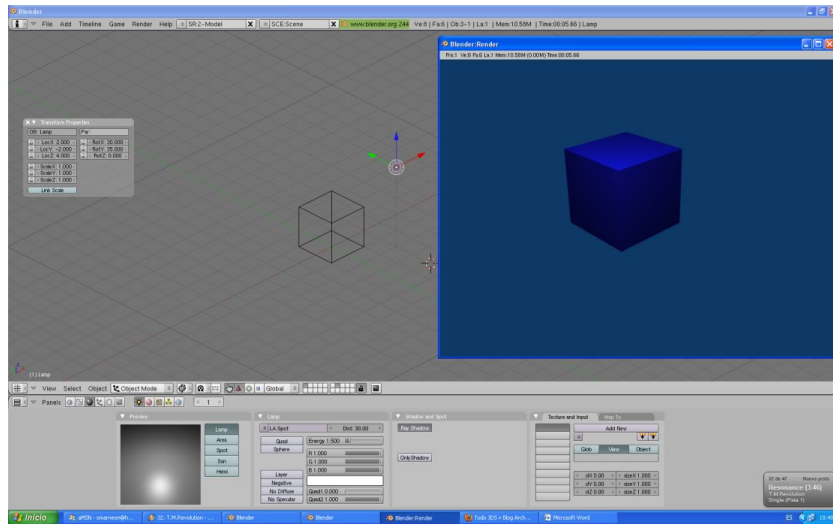


Figura 58 Ejemplo en Blender de Fuente Emisora de Luz (Luz de Lugar).

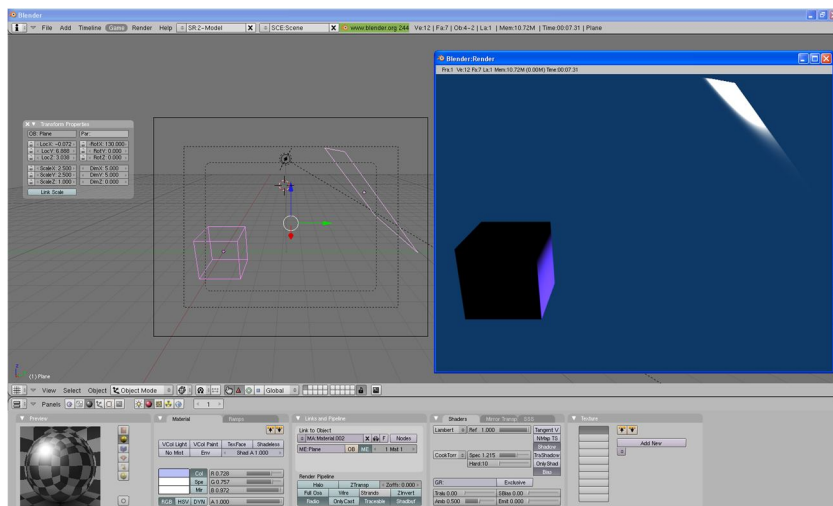


Figura 59 Ejemplo en Blender de Fuente Reflectora de Luz (Luz Direccional).

Renderizado

El renderizado es un proceso de cálculo complejo llevado a cabo por una computadora cuyo propósito es generar una imagen 2D a partir de una escena 3D, en otras palabras se puede decir que durante el proceso de renderización, la computadora representa la escena 3D, plasmada en una imagen 2D en la pantalla.

La renderización se aplica a los gráficos generados por computadora, siendo más usual en el proceso denominado infografía, este proceso se desarrolla con el fin de imitar un espacio 3D constituido por estructuras poligonales, luces, texturas, materiales, animación, simulado de ambientes, estructuras físicas realistas, entre otras.

Una de las partes de mayor relevancia en los programas dedicados a la infografía, son los motores de render, estos son capaces de realizar técnicas complejas como radiosity, raytrace (trazador de rayos), canal alpha, reflexión, refracción, iluminación global, entre otras muchas actividades.

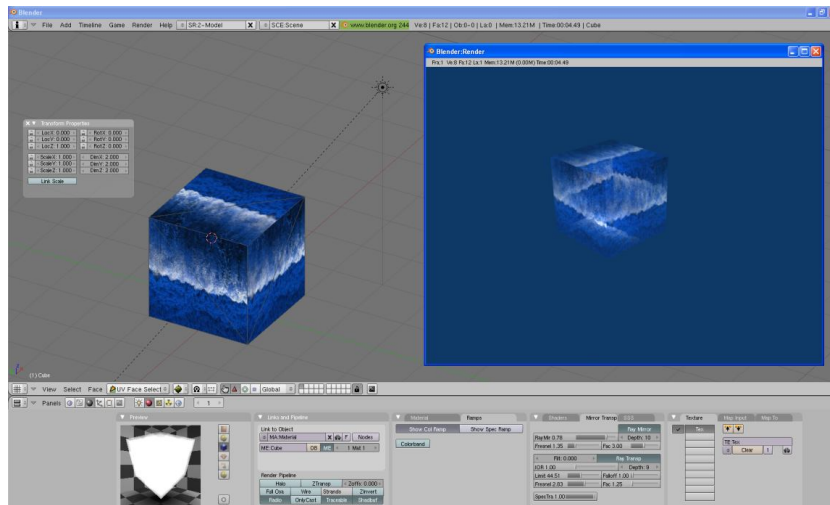


Figura 60 Modelado de un cubo con texturas y visualización de algunas tecnicas de los motores de render.

APIs de Gráficos 3D

“Los gráficos 3D se han convertido en algo muy popular, particularmente en juegos de computadora, al punto que se han creado APIs especializadas para facilitar los procesos en todas las etapas de la generación de gráficos por computadora.” (Corchado) Un API se define como la especificación o descripción de un conjunto de clases y/o funciones cuyo uso tienen un fin determinado, la creación de gráficos 3D. Estas APIs son de gran importancia para los desarrolladores de hardware para gráficos por computadora, ya que permiten al programador un fácil acceso al hardware y sus propiedades de manera abstracta, con lo que se permite sacar el máximo provecho de las ventajas que nos ofrecen cada una de las diversas tarjetas de video que existen actualmente.

APIs de Gráficos 3D para Dispositivos Móviles

- **Manager Direct3D Mobile (MD3DM):** Forma parte de iniciativa DirectX de Microsoft, y proporciona soporte para aplicaciones gráficas 3D en plataformas basadas en Windows CE y Windows Mobile, deriva del API Direct3D de los sistemas Windows de escritorio, pero con una optimización para sistemas embebidos. Los ventajas más importantes que presentan son:

- El uso de recursos es menor comparado con los sistemas de escritorio como resultado de eliminar el soporte para aquellas capacidades gráficas que no están disponibles en dispositivos móviles (debido a su limitado hardware y batería).
- Su arquitectura permite implementar soluciones basadas sólo en software, sólo en hardware, o con una mezcla de ambas.
- Está construida de forma que de soporte a diferentes arquitecturas, tal que, además de usar números en coma flotante puede operar también con valores en coma fija.
- **Mobile 3D Graphics (M3G):** M3G También denominada JSR-184, fue diseñada para crear gráficos 3D en dispositivos móviles, se basa en las APIs de Java 3D y OpenInventor, de tal forma que los gráficos 3D creados con esta API son escalables, interactivos y eficientes en el uso de los limitados recursos. Esta API se propone como una extensión de J2ME, al integrarse en este como un conjunto adicional de clases, entre sus características las siguientes:
 - Posibilidad de permitir el uso de 3D en diferentes aplicaciones.
 - Sacar el máximo rendimiento de las capacidades de hardware de alto nivel al no asumir la existencia de este.
 - No imponer límites arbitrarios a la complejidad o tamaño del contenido del modelo 3D.
 - Soporte de características avanzadas en plataformas con pocas prestaciones o mantener una reducción en el uso de memoria RAM y ROM.
- **OpenGL-ES:** Es una versión compacta de la librería estándar OpenGL para ser usado en entornos embebidos (teléfonos móviles o PDAs), esta es la API oficial escogida por el sistema operativo Symbian. La han adoptado diversos fabricantes de dispositivos móviles (Nokia, Sony Ericsson, LG, entre otros), también existe soporte para esta en los sistemas Windows Mobile y Linux. En la actualidad existen varias versiones y perfiles de uso:
 - OpenGL-ES 1.0: La primera versión de OpenGL-ES, destinado a proporcionar una API muy compacta sin sacrificar características: tenía que ser plenamente aplicable en el software en menos de 50kB código, al mismo tiempo que se debe adaptar bien para la aceleración de hardware. (Pulli, Aarnio y Roimela) Está basada en la especificación 1.5 de OpenGL y debido a que está diseñada para ser usada en procesadores embebidos con capacidad limitada, se han quitado algunas funcionalidades (manejo de texturas 3D, operaciones complejas con polígonos, etc) y se han añadido funciones capaces de operar en coma fija para aumentar la velocidad, reducir el ancho de banda y minimizar el consumo de energía (la versión OpenGL-ES 1.1 presenta algunas mejoras).
 - OpenGL-ES 2.0: OpenGL ES 2.0, proporciona una API totalmente renovada, y el apoyo a un alto nivel de lenguaje de sombreado (GLSL ES): sustituye varias etapas de la tradicional función denominada tubería de gráficos programables con sombreadores de vértices y fragmentos, y por lo tanto no es compatible hacia atrás con la serie 1.x. (Pulli, Aarnio y Roimela) Está basada en la especificación OpenGL 2.0, lo cual permite usar un lenguaje de definición de shaders derivados del estándar original, programar la línea de transformación y algunas operaciones con texturas.
 - OpenGL-SC: El perfil de OpenGL SC es parte de una familia más amplia de derivados OpenGL de interfaces de programación de aplicaciones. (Hall y Knaus) Está basada en la especificación 1.3 de OpenGL, está diseñada para uso en sistemas de tiempo real, por lo cual se han simplificado algunos aspectos del estándar original para que sea más claro y menos complejo su implementación.
- **Java building for OpenGL-ES:** También denominada JSR-239, proporciona a los programadores de Java acceso lo más directamente posible a la implementación de OpenGL-ES y EGL que haya por debajo. Esta se diferencia de la JSR-184 en que no está orientada a objetos, ni estructurada en clases, ni ligada a ninguna extensión de Java. Se creó para facilitar la traducción a Java de programas ya existentes que usasen OpenGL, o para desarrolladores que ya estén familiarizados con esa API.

Las diferentes versiones de OpenGL-ES se presentan como una jerarquía de las interfaces de Java. La base de la interfaz GL se amplía con nuevas funciones y fichas en GL10 y GL11, para las versiones de OpenGL-ES 1.0 y 1.1, respectivamente. Varias extensiones de OpenGL-ES también están expuestas en la API, a fin de que las características más allá de la funcionalidad básica se puedan acceder. (Pulli, Aarnio y Roimela)

Selección del API de gráficos 3D a usar en el modelado para el sistema de localización para ambientes internos

En base a la información mostrada anteriormente y con la selección del lenguaje de programación para dispositivos móviles y embebidos hecho en los temas anteriores de antemano se descarta la posibilidad de las APIs de gráficos 3D OpenGL-ES (versiones no compatibles con J2ME) y Managed Direct3D Mobile, por la falta de compatibilidad directa con el lenguaje J2ME el cual fue seleccionado anteriormente como el lenguaje de desarrollo para el sistema. Por lo tanto se cuenta solamente con APIs de gráficos 3D las cuales son: Java binding for OpenGL-ES (JSR 239) y Mobile 3D Graphics (M3G o JSR-184), de las cuales se optó por el API de Mobile 3D Graphics para el modelado en base a las prestaciones (Tabla 2) con que cuenta cada una de las APIs.

Además de las prestaciones mencionadas anteriormente cabe desatacar que la combinación de gráficos 2D con 3D es mucho más sencilla y factible dentro del API JSR-184 en comparación con JSR-239, y de igual forma es más factible la combinación de esta API con otras APIs destinadas a otros fines (en especial para el diseño de juegos) como video, audio, conexiones inalámbricas, entre otras. Aunado a esto en un estudio hecho sobre la evolución de los juegos 3D en dispositivos móviles (Chehimi, Coulton y Edwards) se observó el rendimiento de las APIs JSR-184 y JSR-239 en algunos teléfonos Nokia por medio del Benchmarking software (software de evaluación comparativa) SPMark04 de Futuremark y JBenchmark3D de JBenchmark, observado un mejor rendimiento (Tabla 3 y Tabla 4) de la aplicación basada en JSR-184 en comparación con la implementación basada en JSR-239 (a pesar de la falta de compatibilidad de JSR-184 en 2 de los 3 modelos de celulares usados).

Tabla 2 Comparativa de las APIs de gráficos 3D disponibles para el lenguaje J2ME.

<i>Prestaciones \ API</i>	Java binding for OpenGL-ES (JSR-239)	Mobile 3D Graphics (M3G o JSR-184)
<i>Modo de modelado</i>	Basado sólo en código	Basado en código (Immediate Mode) o con ayuda de herramientas de diseño 3D (Retained Mode)
<i>Tamaño máximo del modelado</i>	Solución de 50 KB máximo en código	Solución de 150 KB máximo en código y archivos binarios M3G de máximo 2 MB cada uno.
<i>Manejo de multitexturas</i>	Sólo con la interfaz EGL11	Si
<i>Soporte para animaciones</i>	No	Si
<i>Operaciones sobre valores numéricos</i>	Punto flotante	Punto flotante
<i>Integración e interacción de imágenes 2D con el modelo 3D</i>	No	Si
<i>Conocimientos requeridos para inicialización de una escena</i>	Sólo se necesita conocer la creación de gráficos.	Se requiere inicializar una máquina de estados configurada con OpenGL (conocimientos previos del HW del dispositivo).
<i>Nivel de abstracción de las clases e interfaces</i>	Medio (se basa en los modelos de OpenGL-ES 1.0 y 1.1)	Alta (funciona como OpenGL-ES en el Immediate Mode y como Java3D en el Retained Mode)

CAPITULO II: Marco Teórico

	N-Gage	6600	6630
OS v.	6.1	7.0	8.0
CPU (MHz)	104	104	220
OGL ES support	No	No	Yes
Frame/sec	8.15	7.79	14.59
M texel/s	1.80	1.67	3.2
K triangle/s	28.10	27.1	66.41
Running time	4h 10m	5h 28m	5h 42m

Tabla 3 Resultados de las pruebas de SPMark04 sobre los teléfonos Nokia N-Gage, 6600 y 6630.

	N-Gage	6600	6630
M3G support	No	No	Yes
Frame/sec	-	-	11.79
M texel/s	-	-	2.472
K triangle/s	-	-	48.608
Running time	-	-	N/A

Tabla 4 Resultados de las pruebas de JBenchmark3D sobre los teléfonos Nokia N-Gage, 6600 y 6630.

Otro punto importante que se debe contemplar es la capacidad de uso de herramientas de diseño 3D que permiten con mayor facilidad la creación del modelo 3D que usará el sistema de localización. Ya que como se ha visto el API de Mobile 3D Graphics permite el cargar modelos 3D almacenados en archivos binarios con extensión M3G en la aplicaciones donde se implemente ésta. Por ello se cuenta con una serie de herramientas de diseño 3D de las cuales se optó por el uso de Blender, proporciona una ciertas ventajas (Tabla 5) con respecto a otras.

Tabla 5 Comparativa entre herramientas de diseño 3D para la creación de archivos binarios M3G.

Prestaciones \ Herramienta	Blender	Autodesk 3Ds Max	Autodesk Maya
Costo	Gratuito	De paga	De paga
Sistema Operativo soportado	Windows, Linux, Mac Os, Solaris	Windows	Windows y Linux
Facilidad de uso	Media (interfaz no muy fácil hecha en el lenguaje Python)	Alta (interfaz similar las usadas en Windows)	Alta (interfaz similar a las usadas en Windows)
Capacidad de creación de archivos binarios M3G	Si (plugin externo gratuito)	Si (plugin propio o externo de paga)	Si (plugin propio gratuito limitado o completo de paga)
Consideraciones especiales para los crear los modelos M3G	Mapeo UV de texturas y instalación del lenguaje Python	Ángulos y posiciones de la iluminación	Ángulos y posiciones de la iluminación
Mejoras continuas para la creación de archivos M3G	Alta (el creador del plugin para exportación lo mejora continuamente)	Baja (los plugins usados mejoran sólo en ocasiones o con el cambio de versión de la herramienta)	Baja (el plugin sólo cambia en ocasiones)
Observaciones hechas	Obtención de una versión modificada de plugin por parte del creador para solucionar problemas con transparencias	Consumo y configuraciones específicas para poder exportar el modelo a un archivo binario M3G	Consumo y configuraciones específicas para poder exportar el modelo a un archivo binario M3G

CAPITULO III: Metodología de desarrollo

¿Por qué el uso de una metodología de desarrollo dentro de un proyecto de desarrollo software?

Para que un proyecto de desarrollo de software tenga éxito, es necesario el uso de una metodología de desarrollo que involucre tanto a los desarrolladores como a los clientes, llevando a éstos a seguir ciertas pautas predefinidas para el desarrollo del software, es decir, llevar un comportamiento metódico que garantice tener una mayor seguridad y calidad dentro del proyecto de desarrollo software, fin que persigue el uso de una metodología.

Sin embargo a través del tiempo hasta la actualidad se observa claramente la falta de uso de metodologías de desarrollo dentro proyectos de desarrollo de software, un concepto casi desconocido tanto por profesionales del área como por estudiantes de la misma. La razón de esta falta de uso de una metodología de desarrollo se debe a un sin fin de razones provenientes tanto de los desarrolladores (falta de conocimientos, preferencia por la codificación inmediata, tiempos de cortos de entrega del producto, etc.) como de los clientes (ambigüedad de los requerimientos, preferencia por un costo menor del producto, rápido desarrollo del producto, etc.), aunado a esto, la constante evolución tecnológica hace cada vez más necesario el uso de nuevas metodologías que se adapten mejor a estas innovaciones y sin embargo dentro de los planes de estudios y libros de referencia se hace hincapié en los uso de metodologías antiguas y pensadas en los viejos tiempos, aunque no son malas ni inferiores a las nuevas metodologías, no contemplan de forma adecuada las necesidades actuales tanto de desarrolladores como de clientes.

Para entender mejor la relevancia del uso de una metodología de desarrollo dentro de un proyecto de desarrollo software es necesario conocer a que se refiere este término, entendiendo de manera general que:

Todo software contempla un ciclo de vida, el cual a su vez contempla una serie de tareas con el fin de obtener un producto final, los diversos componentes del software deben de pasar por una serie de fases o etapas dentro de propio ciclo de vida, estas mismas tareas pueden ser resueltas por medio de diferentes técnicas y herramientas, a la vez que se debe de contemplar cuando una tarea finaliza, quien debe realizarla, que tarea es anterior y posterior a la misma, que documentación es necesaria para la tarea, entre otros muchos requerimientos. Todo lo mencionado anteriormente, son detalles organizados, es decir, un estilo de hacer las cosas, pero si a ese estilo se le añade normas y rigurosidad se dice que se tiene una metodología, por lo cual, cuando todos los integrantes de un equipo de desarrollo de software (incluyendo desarrolladores y clientes) siguen un criterio común a la hora de realizar las tareas del ciclo de vida de un software, este criterio es denominado metodología de desarrollo.

Selección de la metodología de desarrollo a usar dentro del proyecto de tesis

Anteriormente se menciona el porqué de la importancia del uso de una metodología de desarrollo dentro de un proyecto de desarrollo de software, ya con esto sólo queda el seleccionar la metodología más acorde con el proyecto, es decir, la metodología que cubra tanto las necesidades de los programadores como de los clientes proporcionando un producto de software de calidad y seguro.

Dentro de las metodologías de desarrollo se puede observar la existencia de una gran variedad de éstas que han sido propuestas a lo largo de varios años, escritas por autores del ámbito académico, autores del ámbito empresarial del desarrollo de software, por administraciones públicas, por gurús del software libre, entre otros. De igual forma cada metodología puede tener una referencia bibliográfica basada en incontables páginas o tomos de un libro (describiendo con detalle todo lo

CAPITULO III: Metodología de desarrollo

relacionado con la metodología), mientras que otras están descritas en apenas algunas páginas (la descripción de la metodología de forma más genérica). Dentro de las propias metodologías es posible encontrar diferentes enfoques, algunas hacen hincapié en los datos, en los usuarios, en la documentación o en cualquier combinación de aspectos que se den dentro del desarrollo de software.

De acuerdo con la información mencionada hasta el momento se observa la diversidad de metodologías de desarrollo existentes y los aspectos que contempla cada una, por lo cual es de suma importancia tener clara cuál es la clasificación de estas para poder hacer una selección adecuada de la metodología de desarrollo a usar. Dentro de las metodologías de desarrollo existen principalmente dos clasificaciones: las metodologías de desarrollo tradicionales y las metodologías de desarrollo ágiles.

Las metodologías de desarrollo tradicionales son metodologías escritas con anterioridad a los 90's en su mayoría, se centran en la realización de pautas secuenciales acorde con cada una de las tareas de desarrollo de software y una extensiva documentación por parte de cada una a lo largo de todo el proyecto, acorde con un plan de proyecto; estas metodologías surgen a raíz de la ausencia de lineamientos claros y procesos formales dentro del desarrollo de software siendo casi artesanal en su totalidad, de ahí que se optara por importar los conceptos y fundamentos de metodologías existentes en otras aéreas y que posteriormente fueron adaptadas al desarrollo software. Otras de las características propias de estas metodologías son:

- Procesos basados en planeación exhaustiva, cuyo resultado de cada uno se espera sea determinístico y predecible, en la realidad sólo por medio de la recolección y estudio de métricas de desarrollo se logran estimaciones acertadas en contextos de desarrollo repetibles.
- Altos costos que conllevan el implementar un cambio, siendo aún mayor en el caso de que éste sea en una etapa ya superada dentro del ciclo de vida del proyecto de software, y que en muchas ocasiones es necesario la documentación extra del cambio de acuerdo al manejo que hace la metodología para el mismo.
- Falta de flexibilidad en proyectos en donde el entorno es altamente volátil, es decir, la metodología no puede ser fácilmente adaptable a los cambios que sufre el entorno como nuevos requisitos por parte del cliente, cambios en la tecnología base del proyecto, entre otros.
- Las personas involucradas en el desarrollo de software son partes reemplazables, es decir, las personas son como los recursos, estando disponibles dentro de varios tipos (programadores, diseñadores, analistas, etc.), los individuos no son tan importantes, sólo los roles lo son, de tal forma que el plan del proyecto será afectado por el número de individuos dentro del rol asignado, por lo cual no deben ser tratados como individuos ya que esto afecta la moral y la productividad.

Las metodologías de desarrollo ágiles son metodologías escritas desde los 90's; estas se centran en el uso de procesos adaptables promoviendo su formalización, son conscientes de los cambios y amplitud del mundo software, considerando que una metodología debe ser lo suficientemente precisa para que todo el mundo la siga y sea útil como pauta común, al mismo tiempo que sea adaptable para su uso en diversos proyectos, debe contar con una complejidad con el fin de que su utilización sea provechosa para el equipo involucrado en el proyecto, y que al mismo tiempo sea sencilla para que no resulte complicada su utilización por parte de los involucrados; estas metodologías surgen a partir de varias opiniones a favor y en contra de las metodologías tradicionales, en muchas ocasiones son mal entendidas ya que no están en contra de administrar procesos de desarrollo como muchos creen, se fundamenta en dos aspectos retrasar las decisiones y la planificación adaptativa, potenciando el desarrollo de software a gran escala al perfilar como su objetivo valores y principios que permitan a los equipos desarrollar software y responder a los cambios que surjan durante el proyecto de forma rápida. Otros aspectos que considera un modelo de desarrollo ágil son el uso de procesos incrementales que incluyen pequeñas y frecuentes entregas en ciclos rápidos, además de la cooperación constante entre cliente y desarrolladores por medio de una comunicación precisa y constante.

CAPITULO III: Metodología de desarrollo

Con las definiciones y conceptos dados hasta el momento sobre las dos principales clasificaciones dentro de las metodologías de desarrollo software, se puede resumir las principales diferencias que existen entre ellas por medio de la siguiente tabla:

Tabla 6 Diferencias entre Metodologías Tradicionales y Metodologías Ágiles.

Metodologías Tradicionales	Metodologías Ágiles
Basada en normas provenientes de estándares seguidos por el entorno de desarrollo.	Basadas en heurísticas provenientes de prácticas de producción de código.
Cierta resistencia a los cambios.	Especialmente preparadas para cambios durante el proyecto.
Impuestas externamente (fuera del equipo de desarrollo).	Impuestas internamente (por el equipo de desarrollo).
Procesos muy controlados, con numerosas políticas/normas.	Proceso controlado moderadamente, con pocos principios.
Existe en contrato prefijado.	No existe un contrato tradicional o por lo menos es bastante flexible.
El cliente interactúa con el equipo de desarrollo mediante reuniones.	El cliente es parte del equipo de desarrollo.
Grupos grandes y por lo general distribuidos.	Grupos pequeños (de no más de 10 integrantes) y que trabajan conjuntamente en un sólo lugar.
Muchos roles.	Pocos roles.
La arquitectura del software es fundamental y se expresa mediante modelos.	Menor énfasis en la arquitectura del software.

Conforme a la información recopilada y mostrada hasta el momento sobre las metodologías de desarrollo de software y de acuerdo a los fines perseguidos por el proyecto de Tesis, el tipo de metodología que se considera más adecuado para su uso sería la rama concerniente a las Metodologías Ágiles. Sin embargo dentro de esta rama existe una gran variedad de metodologías de desarrollo, entre las cuales se encuentran las siguientes:

- **Evolutionary Project Management** (Evo, presentada en 1976 por Gilb).
- **Microsoft Solutions Framework** (MSF, presentada en 1994 por Microsoft).
- **Scrum** (presentada en 1995 por Sutherland y Schwaber).
- **Rapid Development** (RAD, presentada en 1996 por McConnell).
- **Rational Unified Process** (RUP, presentada en 1996 por Kruchten).
- **Dynamic Solutions Delivery Model** (DSSM, presentada en 1997 por Stapleton).
- **Agile RUP** (dX, presentada en 1998 por Booch, Martin y Newkirk).
- **Crystal Methods** (CM, presentada en 1998 por Cockburn).
- **Feature-driven development** (FDD, presentada en 1998 por De Luca y Coad).
- **Extreme Programming** (XP, presentada en 1999 por Beck).
- **Adaptive Software Development** (ASD, presentada en el 2000 por Highsmith).
- **Lean Development** (LD, presentada en el 2001 por Charette, Mary y Tom Poppendieck).
- **Agile Modeling** (AM, presentada en 2002 por Ambler).

Conforme a una pequeña revisión de cada una de las metodologías ágiles mencionadas anteriormente, de acuerdo con comentarios recibidos y conforme a los involucrados en el proyecto de tesis (sólo un tesista), se decidió que la opción más viable como metodología ágil de desarrollo de software a usar sería XP, ya que cubre los requerimientos de desarrollo y las expectativas propias del equipo involucrado, todo esto respaldado por las dos promesas principales que propone esta metodología de desarrollo y que son:

- Para los programadores se promete la idea de que sólo se trabajaran con cosas realmente importantes todos los días, permitiendo que estos no sólo se enfrenten a situaciones de terror, sino que tomaran decisiones que sean las mejores que ellos mismos puedan lograr y no tomaran decisiones que no sean las más calificadas para hacer, logrando de esta forma hacer todo lo posible para que el sistema tenga éxito.

- Para los clientes y los administradores se promete ver un avance sustancial en código, obteniendo el valor máximo posible de cada semana de programación, siendo posible ver en pocas semanas avances concretos en los objetivos perseguidos de interés, haciendo posible el cambio de dirección del proyecto en sí durante el desarrollo del mismo, sin que se produzcan grandes costos.

Extreme Programming (XP)

Es una metodología ágil que se basa en llevar a cabo todos los principios y prácticas usadas con sentido común pero al extremo, una metodología ligera especialmente pensada para equipos de mediano a pequeño tamaño (equipos de 2 hasta 10 integrantes) que no están fuertemente limitados al entorno informático existente, adecuada para proyectos con requisitos imprecisos y muy cambiantes con un alto riesgo técnico, ya que se considera que los cambios sobre la marcha son un aspecto natural, inevitable y en algunas ocasiones deseables. Fue propuesta por Kent Beck en 1999 en su libro titulado “abrazando el cambio”, las ideas en las que se basa no son nuevas, la mayoría tan antiguas como la propia programación, hasta cierto punto conservadora, describiendo un enfoque de desarrollo que combina las prácticas utilizadas por muchos desarrolladores de éxito que fueron sepultadas bajo la inmensa literatura existente de los métodos de software y de proceso, algunas de estas: las pruebas de unidad, la programación en parejas, y la refactorización. Estas prácticas dentro de la metodología se combinan de tal forma que se complementan y a menudo se controlan unas a otras, centrando su atención entre la interacción que existe dentro de las diferentes prácticas, es en sí una disciplina de desarrollo de software que da lugar a un tiempo razonable (a tal grado que se pueden hacer en una fracción del día en ocasiones) de la ejecución de las pruebas necesarias.

XP es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. XP se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. (Canós, Letelier y Penadés)

Dentro de XP se define cuatro variables para proyectos software:

- *Costo: Más dinero puede ayudar, pero demasiado dinero muy pronto crea más problemas que los que resuelve. Por otro lado, el dar a un proyecto muy poco dinero conlleva a no ser capaz de resolver el problema del cliente.*
- *Tiempo: Más tiempo para entregar puede mejorar la calidad y aumentar el alcance. Dado que la retroalimentación de los sistemas de calidad en la producción es mucho más alta que cualquier otro tipo de retroalimentación, dar a un proyecto demasiado tiempo es dañino. Dar a un proyecto muy poco tiempo y la calidad disminuye, al alcance, tiempo y costo les pasará lo mismo.*
- *Calidad: La calidad es terrible como una variable de control. se puede tener muchas ganancias a corto plazo (días o semanas) deliberadamente sacrificando la calidad, pero el costo -negocio humanos, y técnicos- es enorme.*
- *Alcance: Menos alcance permite ofrecer una mejor calidad (mientras el problema de negocio del cliente sea aún resuelto). También le permite entregarlo más pronto o más barato. (Kent)*

Como se ha dicho los cambios son algo común, no quedando exento el ciclo de vida de desarrollo de un proyecto de software, por lo cual es necesario que los desarrolladores involucrados en el proyecto se den cuenta que lo importantes no son los cambios en sí, ya que estos van a suceder de una u otra forma, lo realmente relevante es cómo se van a enfrentar éstos. Por lo cual como sucede en cualquier otra actividad humana es necesario el establecimiento de valores compartidos que guíen a aprender a conducir un conjunto de prácticas de desarrollo de software, teniendo en cuenta ciertos criterios que indiquen si se va en la dirección correcta, XP propone los siguientes 4 valores para ello:

- **Comunicación:** El primer valor para XP es la comunicación, todo problema que surge dentro de un proyecto puede remontarse a la nula o mala comunicación entre alguno de todos los involucrados en el mismo, por lo que se hace énfasis en que la comunicación debe involucrar a todos los interesados, debe ser libre, sincera y efectiva. La mala o nula comunicación no ocurre por casualidad, hay muchas circunstancias que la provocan (el jefe del proyecto que se enoja cuando el programador le dice una mala noticia, aquella idea importante que le comunica un cliente al programador y que sea la ignorada, etc). XP mantiene, ayuda y fomenta la fluidez correcta de la comunicación mediante el empleo de prácticas en sentido de corto plazo (como las pruebas unitarias, la programación en parejas, y la estimación de tareas) que no pueden ser llevadas a cabo sin la comunicación, pero no se debe olvidar que a pesar de esto la comunicación es en ocasiones ocultada (la gente se distrae, se asusta, comete errores, entre otros problemas) dentro de un proyecto de XP, para tratar de dar una solución a esto XP cuenta con un monitor (Entrenador) cuya tarea es advertir que la gente no se está comunicando y reintroducirlos dentro de la misma.
- **Simplicidad:** El segundo valor para XP es la simplicidad, es de gran importancia nunca perder de vista que el objetivo de todo proyecto es el de proporcionar un valor al cliente, no demostrar que tanto sabe o puede hacer el equipo de desarrollo, ya que en muchas ocasiones esto provoca que se desarrolle una aplicación que resuelva más problemas que los que el cliente necesita, al hacer esto se producen retrasos, errores y todo tipo de problemas que se pudieron evitar si sólo se responde a esta pregunta “¿Qué es lo más simple que pueda funcionar?”, dentro de XP existe un monitor (Entrenador) encargado de esta tarea.

Lograr la simplicidad de algo no es tarea fácil por lo general siempre se tiende a pensar que se debe hacer mañana, la próxima semana o el próximo mes. Esto sucede mucho dentro del desarrollo de software, se tiene la vaga idea que puede ser que en futuro se ocupe y que realmente si no está dentro de los requisitos futuros esto se traduce en pérdida de tiempo en muchas ocasiones. Debido a esto XP apuesta a que se haga una cosa sencilla el día de hoy y mejor pagar un poco más mañana si es necesario, en vez de hacer una cosa complicada hoy y que mañana no tenga utilidad alguna.

- **Retroalimentación:** El tercer valor para XP es la retroalimentación, por lo general se recurre al monitor (Entrenador) involucrado en los 2 primeros valores de XP para entender sobre el progreso y fiabilidad del sistema, sin embargo esto no puede responder en su totalidad, por eso es común que éste comente la frase “No me preguntes a mí, pregúntale al sistema”. Para ello es necesario la creación de pruebas funcionales del software, lo que permite conocer el grado del progreso y fiabilidad del sistema, esta información proporciona una retroalimentación que puede provenir de las herramientas de automáticas, los clientes, los programadores, los encargados de pruebas o todo aquello relacionado con las pruebas del sistema.

La retroalimentación real de un sistema por lo general es recibida por los clientes, los encargados de pruebas y los encargados de seguimiento, estos individuos involucrados principalmente dentro de las pruebas. Este valor junto con la sencillez y la comunicación actúan de forma conjunta, a tal grado que una mayor retroalimentación se ve reflejada en una fácil más comunicación, lo que lleva a que al escribir pruebas el sistema comienza a simplificarse y al simplificarse el sistema es más fácil probarlo, un sistema simple, tienes menos que comunicar, lo que conduce a una comunicación más completa, lo que al final de cuentas lleva a un ciclo cuya meta final es un sistema simple y completo que cumpla con todas las necesidades para las que fue diseñado.

CAPITULO III: Metodología de desarrollo

- **Valentía:** El cuarto y último valor de XP es la valentía, esta se ve puesta a prueba en muchas ocasiones al igual que sucede en la vida diaria de la personas, de tal forma que tenemos que tomar una decisión que conlleva muchos riesgos y sólo se tienen dos caminos: apostar por ese camino (valentía) o dejarlo e irse por el camino más sencillo (cobardía), lo mismo sucede dentro del desarrollo software, ya que hacer lo correcto requiere mucho valor, se deben asumir retos y tener el valor de afrontar los problemas que conllevan. Esto se puede ver dentro del desarrollo de software cuando se está diseñando una aplicación durante días o semanas y, una vez que se acaba con esta o estando casi por terminar se tiene el valor de tirar el código y volver a hacerlo, esto se puede deber a que el código está muy forzado (parchado), se tiene una idea que podría reducir algún parámetro (código, tiempo de ejecución o requerimiento por parte de las misma), se ha alcanzado el límite de escalabilidad, no cumple con los casos de prueba, o por alguna otra razón que signifique una mejora de la aplicación o de las necesidades a satisfacer por la misma, siendo lo opuesto una cobardía dejarla tal y como está. La valentía junto la comunicación, la sencillez y la información concreta se convierte en algo extremadamente valioso.

Existen ciertas funciones que tiene que ser llevadas a cabo por algún integrante en especial del equipo de desarrollo, a esto se le conoce como rol. En general un equipo consigue más y mejores resultados cuando ciertos roles son asumidos con responsabilidad. Ya que cuando un integrante de un equipo asume un rol específico, lo hace con un conjunto de responsabilidades para con sus compañeros de equipo, siendo algunos de estos roles casi solitarios o por el contrario en interacción constante con otros. La mejor forma de optimizar un esquema de trabajo es mediante la introducción o eliminación de roles, del mismo modo e igualmente importante es la adaptación del esquema de trabajo para con los integrantes del equipo (por ejemplo no es mala idea ni practica que un cliente sea un programador, siempre y cuando se sepan separar las decisiones técnicas de las de negocio), en vez del sentido inverso. En base a lo anterior se observó que los roles más apropiados dentro de XP de acuerdo con la propuesta original de Beck son:

- **Programador:** El programador es el corazón de XP, a tal grado que si un programador pudiera tomar decisiones cuidadosamente equilibradas a corto plazo y las prioridades a largo plazo, no sería necesaria la existencia de ningún otro rol técnico. En esencia se encarga de escribir pruebas unitarias y produce el código del sistema, sin embargo no son las tareas a las que se encuentra limitado, ya que el trabajo de éste no termina cuando el equipo entiende que se tiene que hacer. Tiene que tener ciertas habilidades como es la comunicación (debe existir una comunicación y coordinación estrecha con otros programadores y con los cliente), la programación en parejas, simplicidad (no sólo en el código, sino que también se debe discutir que es realmente necesario y hasta que nivel), capacidad de refactorizar, creación de pruebas unitarias, a aprender que propiedad del código es colectiva (confiar en los cambios hechos a su código por terceros y si están equivocados asumir la responsabilidad de corregirlos o mejorarlos) y sobre todo reconocer los miedos propios (se gasta demasiado tiempo tratando desesperadamente de no fallar) y tener coraje para con estos.
- **Cliente:** El cliente es la otra mitad de la dualidad que forma el corazón de XP, el programador sabe programar y el cliente que es necesario programar. La tareas de éstos son las de escribir las historias de usuario (debe comprender que terreno cubrir, que incluir y que excluir) y las pruebas funcionales (que cosa es útil para poner a prueba y que pruebas son redundantes) para validar su implementación, al igual que se encarga de asignar la prioridad a las historias de usuario y decide cuáles se implementan en cada iteración con el fin de dar un mayor valor al negocio. Los mejores clientes son los que realmente van a usar el sistema y que tienen un cierta perspectiva sobre el problema a resolver, hay ciertas cualidades que debe cubrir: saber escribir buenas historias (en un principio puede tener apoyo técnico por parte del equipo de desarrollo y con el tiempo poder realizarlas por si sólo), tener una actitud de éxito en las tareas a realizar (demostrar coraje) y confianza al tomar decisiones (debe ser de mente abierta capaz de ver los beneficios de tomar un rumbo diferente al acostumbrado).

- **Encargado de pruebas:** El trabajo del encargado de pruebas se centra en el cliente ayudándolo a elegir y escribir las pruebas funcionales, ya que en gran medida la responsabilidad de las pruebas recae sobre el programador. Otras tareas que realiza son las de ejecutar las pruebas regularmente, publicando y difundiendo los resultados en el equipo, e igualmente es el responsable de las herramientas de soporte para pruebas, en muchas ocasiones este rol es mal visto por los programadores ya que estos piensan que es una persona independiente y que su único trabajo es el de humillarlo al destruir el funcionamiento del sistema, sin embargo su trabajo tiene gran utilidad, ya que este tiene como responsabilidad probar todos los posibles caminos e informar el resultado, al mismo tiempo que verificar el buen funcionamiento de las herramientas de prueba.

Encargado de seguimiento: Se podría decir que es el equivalente a un rastreador de un equipo de caza, este se encarga de proporcionar retroalimentación al equipo. Realiza un conjunto de cálculos, verifica el grado de acierto entre las estimaciones realizadas (hacer buenas estimaciones es una cuestión de práctica y retroalimentación) y como estas se ajustan a la realidad (tiempo real dedicado), para mejorar futuras estimaciones (pudiendo indicar en el próximo ciclo si una estimación es muy alta o baja, y en qué grado es correcta). Realiza el seguimiento del progreso de cada iteración, teniendo la capacidad de indicar a mitad de la misma si el equipo va por el camino correcto o es necesario cambiar el rumbo, y con el tiempo indicar si la siguiente versión requiere de grandes cambios o no, esto se hace posible mediante los comentarios al cierre del ciclo.

Otro papel importante que realiza el encargado de seguimiento es el de historiador, ya que registra los resultados de las pruebas funcionales y las notificaciones de defectos, acepta la responsabilidad de cada uno, así como que los casos de prueba se han añadido conforme a cada defecto. La habilidad más necesaria para un encargado de seguimiento es la de recopilar información sin perturbar un proceso más de lo necesario (esto es un poco necesario para que las personas sean conscientes del tiempo dedicado a cada tarea).

- **Entrenador:** El papel que juega el entrenador varía dentro de cada proyecto de desarrollo de software, siendo en algunas ocasiones de vital importancia, mientras que en otras la relevancia es mínima (esto depende en muchas ocasiones de la autosuficiencia técnica del equipo y la experiencia que posea el mismo con respecto a XP), su responsabilidad recae en el proceso global, cada integrante del equipo es responsable de la función de su rol dentro de XP, por otro lado el entrenador debe contar con una comprensión más profunda de las prácticas alternativas dentro de XP que pueden ayudar a los problemas que se presenten dentro del equipo, proveyendo de guías y con la revisión del seguimiento del proceso correctamente.

Algunas habilidades con las que debe contar un entrenador es la de mantener la calma cuando el mundo es presa del pánico (debe contar con un don de mando) e igualmente entender cuando es el momento apropiado para interferir cuando se presente un problema (debe influenciar al equipo de tal forma que ellos vean lo que él ve, en ocasiones es necesario llegar a una intromisión directa o toma de control), pero tampoco se debe olvidar que éste tiene que guiar al equipo sólo el tiempo suficiente (con el fin de que poco a poco el equipo se vuelva autosuficiente y no caiga en la dependencia), un buen entrenador es aquel que funciona a distancia (a tal grado que el papel de este disminuya a medida que madura el equipo).

- **Consultor:** En si los proyectos basados en XP no generan especialistas, sin embargo suelen suceder ocasiones en las que se les necesita, siendo esto una fortaleza (ya que hace que el equipo posea una gran flexibilidad) y a la vez una debilidad (de vez en cuando el equipo necesita de conocimientos técnicos especializados) dentro de XP.

Cuando llega el momento de la necesidad de un alto grado de conocimientos técnicos, es cuando se necesita de un consultor el cual es un miembro externo del equipo con un conocimiento específico en algún tema concerniente a un problema que surge dentro del proyecto. Si se trata de un consultor no habituado a XP, este ve con algo de escepticismo el trabajo del equipo, sin embargo estos mismos le encaminarán sobre el problema que necesita ser resuelto, ofreciendo pruebas de ello (por lo general se insiste en las pruebas), por lo cual se le pedirá al consultor que diseñe una solución y posteriormente se le dejará ir. Esto no es una grosería, más bien se debe al enfoque que toma el equipo que trabaja con

CAPITULO III: Metodología de desarrollo

XP, ya que una vez que tienen una solución diseñada por el consultor, el objetivo que persiguen es que se les enseñe a resolver sus propios problemas (por lo general sólo uno o dos miembros del equipo se sientan con el consultor para resolver el problema), cuestionando el diseño de la solución para tratar de encontrar una alternativa más sencilla (estirando todo lo posible la solución original) a la misma.

- **Gestor.** La función del gestor es la de coordinar, por lo cual es necesario que posea cualidades como el valor y la confianza, además, de vez en cuando debe insistir en que los integrantes del equipo hagan lo que dicen que van a hacer, debido a todo esto, en un principio el integrante que asume este rol, le es difícil el trabajar con el equipo. Es el vínculo entre clientes y programadores, ya que ayuda a que el equipo trabaje efectivamente creando las condiciones adecuadas, por ello con frecuencia los propios integrantes le pedirán que controle al equipo. Esto se debe a que los integrantes desean saber lo antes posible si las cosas son diferentes del plan, por lo cual confían en que el gestor reconocerá y aprobará las ideas que permitan mejorar el proyecto (haciendo fuerte énfasis en las pruebas) especialmente si este tiene experiencia en el desarrollo de software. Por lo cual el gestor debe tener cuidado de lo que el equipo produzca, si se logra esto se tiene un equipo que trabaja productivamente, satisface al cliente y hace lo posible de no provocar sorpresas que conlleven a un problema (tanto con el gestor, el sistema o el cliente). Para lograr esto es necesario que el gestor observe lo que está haciendo el equipo, si no tiene sentido para este, debe pedir que se le explique y si aun así sigue sin sentido, el equipo debe detenerse inmediatamente y echar un vistazo para saber si se confía o no en el producto final de la tarea que se está realizando, en general el equipo desea que esta habilidad propia del gestor este a su disposición cuando lo necesiten o fuera del camino de no serles útil.

El ciclo de desarrollo XP pueden verse como un proceso repetitivo y en mejora continua, donde tanto el cliente (debe mejorar el ámbito de entrega del producto) como el programador (no debe trabajar más de lo estimado) aprende y que a grandes rasgos puede resumirse en los siguientes pasos:

1. *El cliente define el valor de negocio a implementar.*
2. *El programador estima el esfuerzo necesario para su implementación.*
3. *El cliente selecciona qué construir, de acuerdo con sus prioridades y las restricciones de tiempo.*
4. *El programador construye ese valor de negocio.*
5. *Vuelve al paso 1. (Canós, Letelier y Penadés)*

El ciclo de vida ideal para un proyecto de XP (no olvidando que dos proyectos pueden ser iguales) está compuesto por las siguientes 6 fases:

- **Exploración:** En esta fase los clientes se encarga de escribir las historias de usuarios en donde se especifica que es lo que los programadores necesitan con el fin de obtener un producto para la primera entrega que cubra puntos principales que den un valor de negocio. Por su parte los programadores revisan y utilizan cada pieza de tecnología, herramienta y practica que intervendrá durante la producción, para ello deben experimentar con los límites de esta, deben de experimentar con ideas arquitectónicas mediante prototipos, deben de estimar las tareas de programación necesarias durante la exploración y de ser necesario deben considerarse la posibilidad de consultar especialista en la tecnología. Esta fase puede tomar pocas semanas o pocos meses dependiendo de acuerdo con la familiaridad y conocimiento de la tecnología a ser usada, por ello es necesario tener confianza en las herramientas que cree pueden ser usadas para obtener un programa terminado, que el código que se hace puede ser ejecutado día con día, puede manejar o aprender las habilidades necesarias y que todos los miembros del equipo confíen unos en otros.
- **Planificación de la entrega:** En esta fase los programadores y clientes llegan a un acuerdo sobre un cronograma y el contenido de la primera entrega, de acuerdo a la priorización hecha por el cliente de cada historia de usuario y las estimaciones de esfuerzo para cada una hecha por los programadores.

Por lo general esta fase sólo dura unos pocos días y el plan para la primera entrega debe ser de entre 3 a 6 meses, para entregas posteriores se estima que no debe excederse de los 3 meses, para ello los programadores se valen de una medida denominada punto, la cual equivale a una semana ideal de programación (40 horas) la cual se usa para valorar las historias de usuario. Por otro lado el equipo de desarrollo valora la velocidad de desarrollo por medio de puntos por iteración, es decir, cuantas historias de usuario y su respectiva valoración se cumplieron en la última iteración antes de un tiempo específico, ya con la suma de puntos se puede indicar cuantos puntos se pueden completar por iteración (es importante recordar que cada iteración produce un conjunto casos de prueba funcional).

- **Iteraciones:** En esta fase se diseña el plan de entrega dividido en una o máximo cuatro iteraciones por semana (tratando de escoger una arquitectura para el sistema en la primera iteración), no excediendo las 3 semanas, se debe tomar en cuenta para su elaboración las historias de usuario no abordadas, velocidad del proyecto, pruebas de aceptación no superadas en la iteración anterior y tareas no terminadas en la iteración anterior. Todo lo mencionado es útil para detectar desviaciones en el plan haciendo los cambios necesarios en las historias de usuarios (agregar, quitar o modificar su alcance), al final de la última iteración, estará listo el sistema para entrar en la etapa de producción (para ello es importante que el cliente complete las pruebas funcionales y todas se halla ejecutado al final de cada iteración). Todo el trabajo de la iteración se expresa en tareas de programación, cada una asignada a parejas.
- **Producción:** Esta fase es el juego final de una liberación, ya que antes de que el sistema sea llevado al entorno del cliente es necesario que se cumpla con un proceso de certificación de software (realización de pruebas adicionales como pruebas en paralelo, revisión y ajustes al del rendimiento). Es el momento ideal para la toma de decisiones sobre el sistema ya que se conoce que se ha integrado en el sistema, estimaciones más realistas de la carga de trabajo del mismo y hasta probablemente dispondrá del hardware de producción, por lo cual es posible incluir características a la versión actual y de no ser posible serán documentadas para posteriores implementaciones (como por ejemplo en la fase de mantenimiento). Es importante destacar que durante esta fase el proceso de evolución del sistemas se alenta, no porque el sistema haya llegado a los límites su evolución, sino más bien porque cualquier cambio que merezca entrar en esta versión es un riesgo importante en la propia evolución.
- **Mantenimiento:** Dentro de XP esta es la fase normal de un proyecto, ya que mientras la primera versión se encuentra en producción se debe de mantener al sistema actual en funcionamiento a la par que se entra en el desarrollo de nuevas iteraciones que integran nuevas funcionalidades, además de incorporar nuevos miembros al equipo de desarrollo (cambios en el equipo poco a poco, pudiendo cambiar al mismo en su totalidad a lo largo de un año) y que el mismo siga adelante (no interrumpiendo los cambios al apoyo a la producción o el desarrollo en curso). Durante cada lanzamiento es normal que se produzca una nueva fase de exploración, verificando y utilizando nuevas tecnologías o mejoras a las ya usadas por el sistema, e igualmente se experimente con nuevas ideas arquitectónicas. Es importante recordar que el desarrollo de un sistemas en producción no es lo mismo que el que no está todavía en producción, se debe ser cuidadoso de los cambios realizados, se debe detener el desarrollo para reaccionar a los problemas de producción, se debe considerar con mucho cuidado al migrar datos reales cuando se cambia el diseño y se debe tener presente el cambio de la estructura del equipo para hacer frente a la producción, todo esto influye en la producción disminuyendo la velocidad de desarrollo del equipo. No se debe dejar el código por ahí en más de una iteración, manteniendo una sincronía entre el código base de producción y de desarrollo, se tendrán señales de advertencia de problemas de integración tempranamente y no se debe olvidar que el tiempo dependerá de la cantidad y costo de la migración.
- **Muerte:** Esta fase es la culminación del proyecto, algo natural en el ciclo de vida de las personas e igualmente sucede los mismo en los proyectos, no puede evitarse ya que tarde o temprano llegará ese momento, lo que se trata de hacer es atrasarlo lo más posible para así obtener los mayores beneficios posibles del proyecto. Estos sucede cuando el cliente está satisfecho con el sistema cumpliendo con sus expectativas, no teniendo más historias de usuario que agregar, es decir, no puede pensar en algo que se pueda agregar en un

futuro previsible. Otra posibilidad para que esto suceda es cuando características deseadas o necesarias por el cliente no pueden ser agregadas por razones económicas, es decir, ya no cumple con lo necesario siendo la tasa de errores intolerable y como consecuencia económicamente no sustentable. Se detienen los cambios en la arquitectura generando la documentación final, se aprovecha para buscar las razones de la caída del sistema y se entiende mejor que buscar para el futuro.

Con el fin de que todo lo mencionado hasta el momento funcione, permitiendo disminuir la mítica curva exponencial de costo que se presenta a lo largo de un proyecto y que el diseño evolutivo funcione, XP relaciona una serie de 12 prácticas de ingeniería de software llevadas al extremo y aplicadas disciplinadamente:

- **El juego de la planificación:** Esta práctica consiste en dividir la funcionalidad de un proyecto por medio de las historias de usuario determinando un balance entre lo que se desea y lo que se puede hacer, para ello el cliente y equipo de desarrollo determinan el nivel de importancia de cada historia y se hace una estimación del tiempo requerido para cada uno.

De esta forma los clientes determinan el ámbito de la aplicación (cuando es suficiente y cuanto es deseable), prioridades (que es lo que se quiere en primer lugar), composición de las entregas (cuanto se necesita del software para mejorar el negocio) y fechas de entrega (el cliente acuerda con el equipo la fecha de entrega que más convenga a ambos). Por otra parte el equipo de desarrollo realiza estimaciones (cuál es el costo en tiempo y esfuerzo requerido para cada tarea), consecuencias (informa de las consecuencias técnicas de una decisión de negocio), el proceso (la organización y la forma de trabajo del equipo) y la programación detalla (conforme a las prioridades de negocio y de las historias de usuario se informa el orden a realizar cada historia).

- **Entregas frecuentes:** Se trata de entregar al cliente lo antes posible un nuevo valor de negocio mediante la entrega de una nueva versión, para ello esta debe ser lo más pequeña posible (es mejor una entrega en un plazo de 1 a 2 meses que una de 6 a 12) y que a la vez incluya requisitos que den valores de negocio. Con base en esto es posible tener un sistema sencillo rápidamente en producción, versiones nuevas en ciclos cortos, se tiene un mayor y más fácil control del proyecto, y se maximiza la retroalimentación, sin embargo no se debe caer en la mala práctica de disminuir indiscriminadamente las funcionalidades para acelerar las entregas, ya que toda entrega tiene que tener sentido.
- **Metáfora:** El objetivo de esta práctica es el de la visión común del sistema tanto para los clientes como para el equipo de desarrollo, para ello se auxilia de una metáfora global única que sirve para la comunicación en términos comunes para con todos los involucrados en el proyecto, guiando a estos a través de todo el desarrollo como una historia compartida simple de cómo funciona el sistema en conjunto. La metáfora ayuda a los involucrados en el proyecto a comprender sus elementos básicos y sus relaciones, aunque en algunas ocasiones se necesita de una pequeña explicación ya que no tienen en sí un sentido literario. Se considera que la metáfora dentro de XP reemplaza mucho de lo que las metodologías tradicionales llaman arquitectura, ayudando a obtener una arquitectura fácil de comunicar y elaborar.
- **Diseño simple:** El objetivo de esta práctica es obtener un diseño más simple posible en todo momento para el sistema y que cumpla con las especificaciones (pruebas de aceptación), para ello toda complejidad adicional que surja a lo largo de proyecto se debe retirar o simplificar en cuanto es descubierta.

No tiene sentido un diseño complejo que pueda incluir necesidades hipotéticas ya que como sucede con todo en la vida los requerimientos están en constante cambio. Un diseño simple permite entregar un prototipo al cliente, pudiendo obtener una retroalimentación lo antes posible, proveyendo una idea más clara de los futuros requisitos del cliente a cubrir por el sistema. En general durante la creación de un diseño es normal que se reestructure posteriormente (borrando, modificando o eliminando partes del mismo), siempre que no afecte ninguna información ni a los valores de negocio deseados por el cliente. Para un diseño correcto de software y en el cual cada pieza de diseño debe de cumplir con los siguientes términos:

1. *Ejecutar todas las pruebas.*
 2. *No tiene lógica duplicada. Tenga cuidado con la duplicación oculta como las clases jerarquías en paralelo.*
 3. *Todos los estados son de importante atención los programadores.*
 4. *Tiene el menor número posible de clases y métodos.* ^(Kent)
- **Pruebas:** El objetivo de esta práctica es la de crear pruebas automáticas que indiquen si funciona y durante qué tiempo una características del sistema, para ello intervienen tanto los programadores (escriben pruebas unitarias para verificar el funcionamiento del código) como los clientes (escriben pruebas para verificar el funcionamiento de una característica finalizada). No es necesario escribir una prueba para cada método, sólo debe hacerse para aquellos métodos de producción que podrían romperse, ya que si este falla es necesario eliminar el código, volviendo a escribir el método y nuevas pruebas para el mismo. Lo mejor es no dejar las pruebas para el final y realizarlas al mismo tiempo que se escribe el código o incluso antes para evitar complicaciones mayores que pueden ocurrir posteriormente.
 - **Refactorización:** Esta práctica consiste en modificar el diseño del sistema sin modificar el funcionamiento externo del mismo, ya sea estructurando, eliminando la duplicación, mejorando la comunicación, simplificando o añadiendo flexibilidad por medio de funciones o características simples. Llevar a cabo la refactorización es posible gracias a que se pueden hacer las verificaciones del sistema mediante las pruebas. Debe tener en cuenta que esto conlleva más trabajo del necesario, y se hace cuando el sistema lo requiere obteniendo una característica de funcionalidad.
 - **Programación en parejas:** Esta práctica consiste en poner a producir código con sólo dos programadores por computadora, colaborando tanto en código como en pruebas del mismo, con lo que se estimula la comunicación y la transmisión de conocimientos, produciendo código de calidad y detección temprana de errores. Siendo una actividad dinámica a tal grado que las parejas pueden cambiar en el transcurso del día, y si alguno de los miembros de las parejas tiene que realizar una tarea en un área desconocida por el mismo puede preguntarle a alguno de los integrantes con el que se haya emparejado recientemente. Dentro de las parejas existen dos funciones, el que trabaja directamente con la computadora y piensa la forma de implementar el código, mientras que el otro piensa estratégicamente sobre si el enfoque tomado funcionará, que caso de prueba no puede funcionar, simplificar el sistema mientras aún resuelve el problema deseado, entre otras actividades similares.
 - **Propiedad colectiva del código:** Esta práctica trata de que todos los integrantes del equipo de desarrollo asuman la responsabilidad total del sistema, de tal forma que cualquier integrante puede modificar el código (no todos conocen a la perfección cada parte, pero si sabe algo) en cualquier parte del sistema y en cualquier momento si este ve una oportunidad (por lo general esto sucede durante la refactorización) siempre y cuando esto mejore de alguna forma al sistema. Para conseguir este fin un apoyo son las pruebas automáticas, lo que provee de valentía ante esta situación, con ello es posible el buen funcionamiento de los equipos dinámicos dentro de XP.
 - **Integración continua:** Esta es otra práctica llevada al extremo en la cual el sistema se integra y construye varias veces (cada hora o cada día) cada vez que se completa una tarea con lo cual es posible que surjan errores y se puedan solucionar de forma más rápida éstos. Para lograr esto el proceso de integración debe estar automatizado (por lo general se tiene una computadora dedicada a esta tarea) y debe verificarse todo esto mediante las pruebas (deben ser superadas al 100%), ya que si una prueba no consigue superarse hay que deshacerse de lo que hemos hecho y empezar de nuevo.
 - **Semanas de 40 horas de trabajo:** Esta práctica hace hincapié en no realizar más de 40 horas a la semana de trabajo y nunca trabajar horas extras más allá de una sola semana, este hecho se basa en que un programador cansado se equivoca más. De forma general cada personas tolera de diferente forma el trabajo, algunos sólo están concentrados 35 horas a las semanas, mientras otros hasta 45, pero es imposibles que alguien trabaje 60 horas a la semana a lo largo de varias semanas, igualmente es importante verificar que si la norma de trabajo es de 40 horas por semana a los largo de la semana, es que algo va mal en el proyecto o el equipo.

- **Ciente en el equipo:** Esta práctica incita a que un cliente que realmente va a ocupar el sistema debe estar lo más cerca del posible del equipo (en situaciones reales debe estar dentro del equipo), con la finalidad de responder a ciertas preguntas, resolver disputas que surjan, establecer prioridades a pequeña escala y ayudar en la escritura de pruebas de aceptación.
Una desventaja se presenta a la hora de que los administradores de la empresa deciden si es conveniente para ésta la salida temporal de un empleado con el fin de ser integrado en el equipo, sin embargo ni los programadores pueden generar 40 horas semanales de trabajo por lo cual aunque el empleado no esté físicamente en la empresa este puede encargarse de cierta actividades que le corresponde de forma distante.
Otra desventaja se presenta cuando un empleado que se ha integrado al equipo lleva ya cientos de hora trabajando con él mismo y el proyecto se cancela, esto se traduce en pérdidas para la empresa, sin embargo un proyecto en XP trata de evitar a toda costa esto. La ventaja de que el empleado de una empresa sea el cliente dentro del equipo de desarrollo es que es posible obtener un mayor valor de negocio para la empresa por medio del sistema desarrollo y a la vez un más rápido desarrollo del mismo.
- **Estándares de codificación:** Esta práctica indica que para que la refactorización y la programación en parejas lleve a cabo su cometido es necesario hacer hincapié en la comunicación mediante el código con el fin de que otros programadores puedan entender y modificarlo, para ello no puede permitirse el lujo de tener diferentes conjuntos de prácticas de codificación. Para lograr este punto es necesario establecer estándares de codificación que todos los programadores deben de seguir voluntariamente cuando escriben código obteniendo una constante en la codificación y manteniendo la menor cantidad de trabajo posible (no habrá código duplicado).

En a lo expuesto puede verse la gran cohesión e interacción que existen entre las 12 practicas (Figura 61) propias de XP.

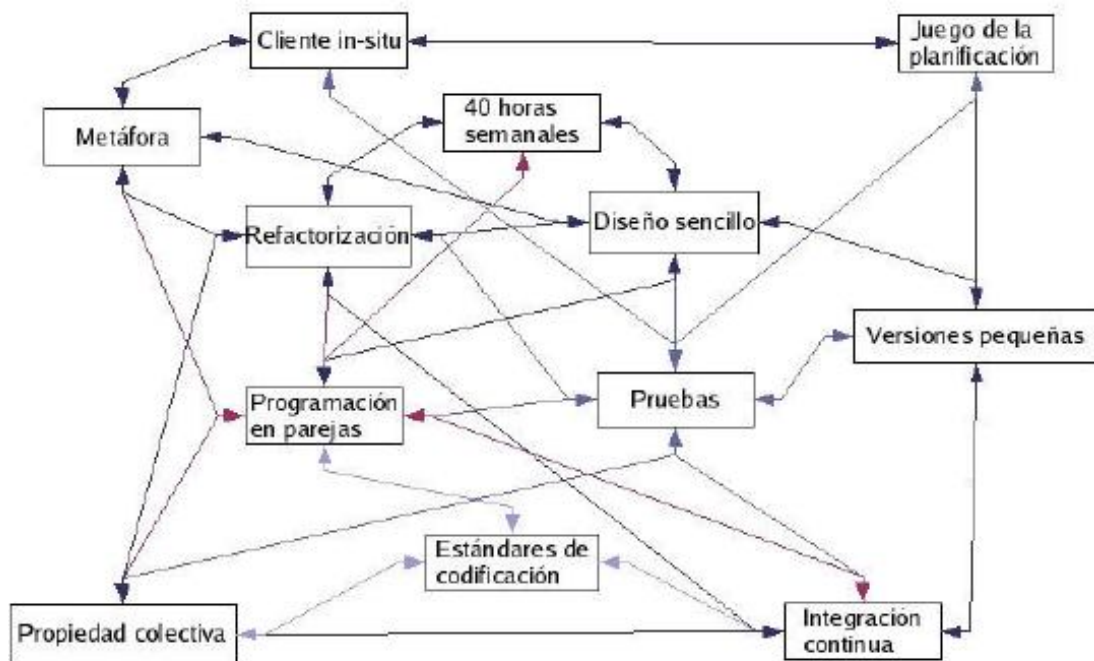


Figura 61 Reforzamiento entre sí de las 12 practica de XP.

CAPITULO III: Metodología de desarrollo

Además es posible distinguir las ventajas y desventajas que presenta esta metodología:

- **Ventajas**
 - Apropiado para entornos volátiles
 - Estar preparados para el cambio, significa reducir su coste.
 - Planificación más transparente para los clientes, conocen las fechas de entrega de funcionalidades vitales para su negocio
 - Permitirá definir en cada iteración cuales son los objetivos de la siguiente
 - Permite tener realimentación de los usuarios muy útil.
 - La presión está a lo largo de todo el proyecto y no en una entrega final
- **Desventajas**
 - Delimitar el alcance del proyecto con el cliente

CAPITULO IV: Compilación de máquina virtual J2ME para Windows Mobile 5.0

Fines perseguidos con la compilación de la máquina virtual

La idea de compilar la máquina virtual J2ME se basa en el hecho, como se mencionó en Capítulo II, que el lenguaje J2ME no sólo es de los más extendidos para la programación de móviles y embebidos, sino que además es multiplataforma, haciendo posible su uso en algunos S.O. no nativos de este lenguaje, tal como el S.O. Windows Mobile 5.0 el cual está disponible a través de uno de los dos dispositivos móviles (Smartphone Motorola Symbol MC 35) con lo que se cuenta para las pruebas. Mediante una pequeña búsqueda y recomendaciones hechas se observó que existen algunas máquinas virtuales para J2ME y J2SE compatibles con Windows Mobile 5.0 disponibles en el mercado, tanto gratuitas (Open Source) como comerciales, sin embargo existe un proyecto que llamó mucho la atención, cuya denominación es phoneME, este proyecto involucra directamente a Sun Microsystems, Inc. y la comunidad Open Source dedicada a difundir el uso del lenguaje J2ME y su tecnología en el mercado.

El objetivo perseguido por este proyecto es el de difundir el uso del lenguaje J2ME y su tecnología en la comunidades dedicadas al desarrollo software, en especial a las que se encaminan en la programación de dispositivos con bajos recursos (dispositivos móviles y embebidos) con el fin de reducir la variación existente entre códigos, aplicaciones y recursos usados por las mismas, tratando de obtener un lenguaje común y estándar disponible en el mayor número de dispositivos y S.O. existentes en el mercado (ver Anexo A).

Fines obtenidos y no obtenidos con la compilación de la máquina virtual

Hasta el momento mediante la compilación del proyecto phoneME Advanced Software se ha obtenido una máquina virtual J2ME para Windows Mobile 5.0 compatible con MIDP 2.0 y CLCD 1.1, con una arquitectura base CDC 1.1, que incorpora las clases y métodos básicos de estas mismas arquitecturas y con la posibilidad de agregar JSR's (120, 135, 172, 177, 211, 239 y 280) opcionales liberados hasta el momento por Sun Microsystems, Inc., además de la incorporación de una de las variantes de la implementación JSR-82 para J2SE denominada BlueCove, lo que permite trabajar con aplicaciones Bluetooth, diseñadas en un principio con la propia implementación JSR-82 de Sun Microsystems, Inc.

Sin embargo uno de los fines perseguidos es la de demostrar la portabilidad de la aplicación desarrolla por el proyecto de tesis, para el sistema de localización en interiores mediante un modelo pre-cargado 3D por medio de la tecnología Bluetooth.

A pesar de los esfuerzos hechos hasta el momento la máquina virtual J2ME obtenida sólo cubre 2/3 de los requisitos para la ejecución de la aplicación, ya que sólo pueden trabajarse con los componentes mencionados anteriormente, entre los cuales no se contempla ninguna forma de obtener o sustituir las APIs para modelado 3D (Java building for OpenGL-ES también denominada JSR-239 ni Mobile 3D Graphics también denominada JSR-184 o M3G) propias del lenguaje J2ME, esto se debe principalmente a que diferencia de JSR-82 que cuenta con una licencia de dominio público (por lo cual existen variantes e implementaciones del mismo) tanto JSR-84 como JSR-239 tiene licencias propietarias (por lo cual no es posible la generación de variantes o implementaciones de las mismas), y por lo que no pueden ser obtenida hasta que Sun Microsystems, Inc. las libere.

CAPITULO IV: Compilación de máquina virtual J2ME para Windows Mobile 5.0

En base a lo anterior no es posible hacer uso de esta máquina virtual J2ME obtenida de la compilación del proyecto phoneME Advanced Software, debido a los propios requerimientos (en lo referente al modelado 3D) de la aplicación del proyecto de tesis, sin embargo, esta queda de antecedente para aquellos proyectos o aplicaciones que contemplen los requisitos que si fueron cubiertos hasta el momento. De igual forma es posible que en un futuro sean liberadas más JSR's propietarias del Sun Microsystems, Inc. con lo que se podría obtener una máquina virtual mucho más completa para proyectos o aplicaciones futuros. Es importante mencionar que como se menciona al principio del capítulo existen otras máquinas virtuales J2ME tanto gratuitas (Open Source) como comerciales, sin embargo se hizo un análisis de las más comunes y se observó que si bien algunas contemplan alguna implementación JSR-82 necesaria para aplicaciones Bluetooth, tan bien se encuentran con el problema de la falta de alguna implementación JSR-184 y/o JSR-239 para las aplicaciones con modelados 3D.

CAPITULO V: Toma de medidas para el modelado 3D del Instituto de Electrónica y Computación (IEC)

Obtención de medidas del IEC

Para efectos de prueba del proyecto de tesis se optó por realizar un modelo 3D del edificio del IEC de la UTM, fue necesario la obtención de las medidas del mismo de la forma más precisa y real posible. Fueron necesarias dos formas para obtener dichos datos, la primera fue el uso de un receptor GPS y una estación total para la medición de la estructura del edificio, la segunda forma fue usar una medición manual por medio de un flexómetro, de un cubículo con medidas estándar del IEC, el grosor de las paredes de edificio, barandales, ventanas, pilares, puertas, etc. Con el fin tener un modelo 3D con las dimensiones lo más reales posibles.

Para realizar las mediciones por medio del GPS y la estación total (ver Anexo B) se tuvo que realizar la siguiente serie de pasos:

- Paso 1. Se seleccionan los puntos de control a ser usados, para los fines necesarios fueron suficientes tres puntos de control, estos puntos deben de asegurarse entre sí, tener visibilidad con los otros puntos (Figura 62) y un piso adecuado (para uso del trípode), se marca cada punto de control (Figura 63). Es importante mencionar que estos puntos deben estar visibles para el GPS.



Figura 62 Ejemplo de visibilidad entre estación total y prisma (ausencia de obstáculos que bloqueen el láser).



Figura 63 Marca de punto de control.

CAPITULO V: Toma de medidas para el modelado 3D del Instituto de Electrónica y Computación (IEC)

Paso 2. Para obtener las medidas 3D de los puntos de control se instala un trípode sobre cada marca (Figura 64), se hace un nivelado de éste, se instala el equipo GPS correspondiente (adaptador para base nivelante, barra de extensión vertical, antenas GPS y abrazadera/receptor) teniendo cuidado de no desestabilizar el trípode (Figura 65), por último se realiza la conexión entre el receptor y la antena GPS.



Figura 64 Nivelación de trípode sobre punto de control. Figura 65 Instalación de equipo GPS sobre trípode.

Paso 3. Antes de comenzar con la toma de medidas de los puntos de control es necesario obtener la altura con respecto al piso a la que se encuentra la antena GPS, este valor será de utilidad para las medidas posteriores.



Figura 66 Medición de antena GPS con respecto al piso.

Paso 4. Para comenzar las mediciones se enciende el equipo presionando el botón rojo de encendido y apagado, ahora se pueden realizar las configuraciones necesarias del equipo, aparecerá una pantalla de entrada, seguida de una pantalla Modo (Figura 67) dentro de la cual se selecciona la opción "Medir" y se presiona la tecla "Enter", posteriormente dentro de la pantalla Medir (Figura 68) se selecciona la opción "Configuración", ahora en la pantalla Menú Medir (Figura 69) se selecciona la opción "Atributo Punto", en la pantalla Atributo punto se agrega el "ID del sitio" y "Descripción del sitio", ahora dentro de los parámetros de altura de antena para el punto se ingresa el valor obtenido con anterioridad con respecto a la altura de la misma, para ello se tiene la opción Inclinada si está midiendo la altura de la antena hasta el extremo externo de la antena GPS, o Vertical si está midiendo la altura de la antena hasta el fondo de la antena GPS, se da clic sobre "Guardar", se regresará a la pantalla del Menú Medir, ahora se selecciona la opción "Unidades", dentro de la pantalla Unidad de Medida, selecciona la unidad de medición más adecuada (para este caso metros), se da clic sobre "Guardar", ya con esto se terminan las configuraciones necesarias.



Figura 67 Pantalla Modo.



Figura 68 Pantalla Medir.



Figura 69 Pantalla Menú Medir.



Figura 70 Pantalla Atributo Punto.



Figura 71 Pantalla Unidad Medida.

Paso 5. Para comenzar a guardar las mediciones de los puntos de control, se presiona la tecla “Esc.”, para salir del Menú Medir, regresará a la pantalla Modo (Figura 67), donde se selecciona la opción “Grabar datos” para comenzar con la medición, por lo general es necesario esperar de 10-15 minutos antes de parar la medición y apagar el GPS, se termina la toma de datos del punto de control (la secuencia de pasos del 2-5 se repiten para cada punto de control).

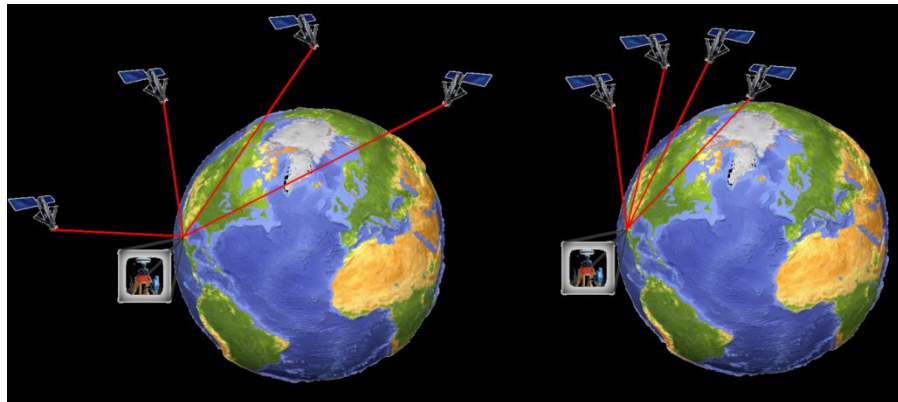


Figura 72 Ejemplo de recepción de señal de los satélites accesibles al dispositivo GPS. Izq) Configuración de satélites con menor error de ubicación, Der) Configuración de satélites con mayor error de ubicación.

Nota: La explicación de la figura anterior se basa en el concepto PDOP³⁹, cuando los satélites están más dispersos por el espacio el valor de PDOP es bajo, dando como resultado el cálculo de posiciones más exactas, por el contrario si los satélites están más cerca unos de otros el PDOP es mayor y las posiciones calculadas son imprecisas

³⁹ Medida sin unidades manejada por los sistemas GPS para una imprecisión tridimensional, la cual indica cuando la geometría satelital provee los resultados más precisos. Es inversamente proporcional al área del polígono delimitado por las intersecciones de las líneas de visión a los satélites con una esfera centrada en el observador

CAPITULO V: Toma de medidas para el modelado 3D del Instituto de Electrónica y Computación (IEC)

Paso 6. Una vez guardados los puntos de control, estos serán usados para obtener las dimensiones del edificio mediante puntos de medición de coordenadas del equipo de una estación total (Ver Anexo B), para ello es necesario (al igual que en caso de equipo GPS) ubicar el trípode sobre un punto de referencia, nivelando al mismo (Figura 64), se instala el equipo de la estación total (adaptador para base nivelante, estación total, unidad de control y batería; (Ver Anexo B) teniendo cuidado de no desestabilizar el trípode (Figura 73), es de gran importancia que la estación total quede a la misma altura que la usada para el punto de referencia (Figura 74), por último se hace la conexión entre la batería y la estación total (Figura 75).



Figura 73 Instalación de equipo de la estación total sobre trípode.



Figura 74 Marca de referencia de la estación total para medir su altura.



Figura 75 Conexión de la estación total a la batería.

Paso 7. Antes de comenzar las mediciones de coordenadas es necesario introducir las medidas 3D (x,y,z) del punto de referencia a ser usado por la estación control, para ello se enciende la estación total por medio de la tecla "ON/OFF" y se oprime la tecla "A/M" con lo que comenzará un barrido de 360° para calibrarse, una vez terminado esto se creará un programa para la recopilación donde se introducirán las coordenadas del punto de referencia donde se ubica la estación total y posteriormente se obtendrán los puntos de medición de coordenadas mediante un prisma y el menú "Direct Reflez" de la estación total.

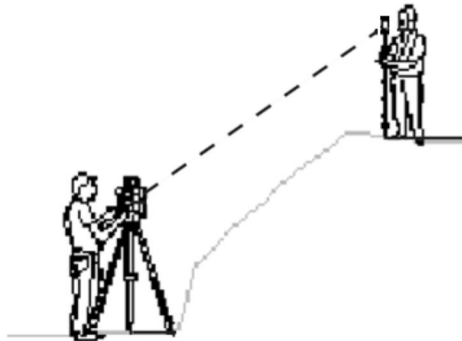


Figura 76 Medición con prisma mediante el uso de menú "Direct Reflez".

Paso 8. Para obtener las dimensiones del edificio, se usaron puntos de medición de coordenadas que se ubican en la estructura del edificio (Figura 77), se auxilió de un prisma y su respectivo trípode, los cuales se ubicaron en los puntos de medición de las coordenadas deseados de tal forma que el trípode del prisma quede nivelado (Figura 78), el prisma quede a la altura adecuada (Figura 79) y visible para la estación total, la cual apuntará al mismo y emitirá un ruido que junto con la mirilla y las perillas de movimiento vertical y horizontal de la estación total (Figura 80) permitirán centrarlo.



Figura 77 Ejemplo de toma de medidas dentro del IEC usando la estación total.



Figura 78 Nivelación de trípode de primas.



Figura 79 Ajuste de la altura del primas.



Figura 80 Mirilla y perillas de ajuste vertical y horizontal de la estación total.

Paso 9. Se procede a obtener las coordenadas 3D (x,y,z) del punto donde se ubica el prisma, por medio de la estación presionando la tecla "A/M", estas se guardan en el programa generado y a continuación se procede con el o los siguientes puntos de medición de coordenadas deseados o necesarios (para ello se realizan los pasos 8 y 9 en cada punto).

Cuando se necesita obtener los valores de un punto de medición de coordenadas que no es visible para la estación total, se puede hacer uso de las coordenadas de uno los puntos ya obtenidos o determinar uno nuevo, se procede a re-ubicar a la estación total en el mismo de tal forma que el punto de medición de coordenadas del que se desea obtener sus valores sean visible para la estación total. Para ello se repiten los pasos 7 y 8 con la única modificación de que en lugar de ingresar las coordenadas del nuevo punto de referencia, se le indica que éste ya está en el programa generado y que el mismo será usado como el nuevo marco referencia para las nuevas mediciones.

Nota: Para mayor información sobre el funcionamiento y las acciones necesarias a realizar en los paso 6-9 con respecto al equipo de la estación total ver Anexo B.

Ya con los puntos de medición de coordenadas es posible obtener las dimensiones lo más precisa y real posible del edificio del IEC, ahora como segunda acción se procedió a medir por medio de un flexómetro⁴⁰ el grosor y las dimensiones de las paredes que constituyen un cubículo del IEC, realizando lo mismo con las ventanas, marcos y puerta correspondientes (Figura 81).

⁴⁰ Instrumento de medición similar a una cinta métrica, construido en chapa metálica flexible debido su escaso espesor, dividida en unidades de medición, y que se enrolla dentro de una carcasa metálica o plástica.

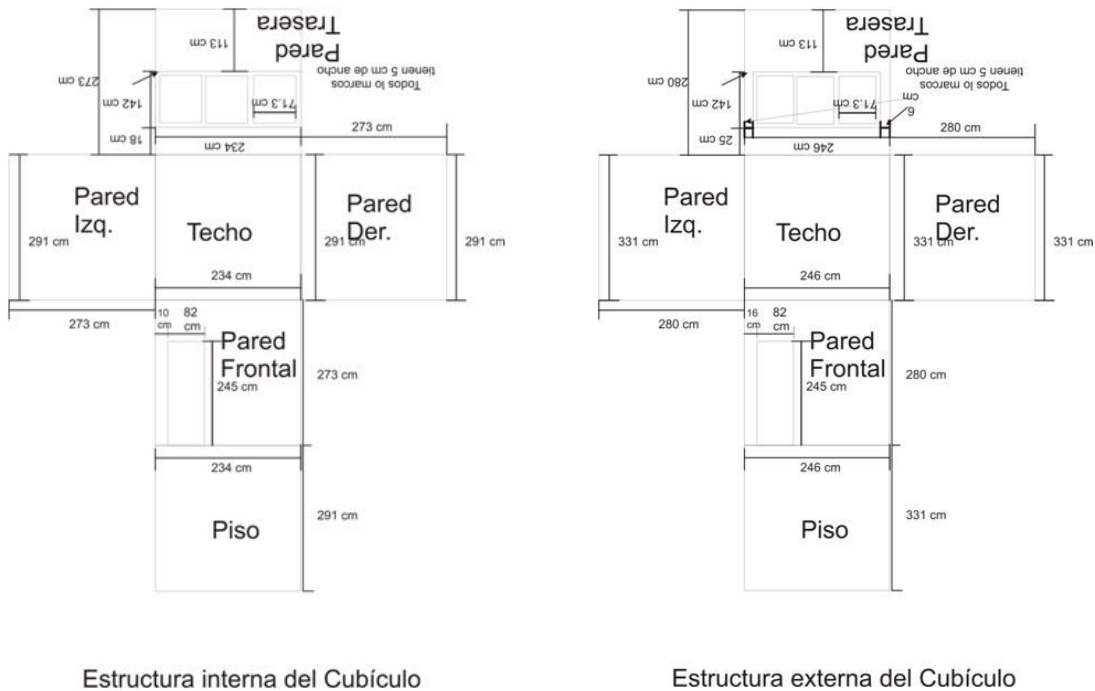


Figura 81 Medidas obtenidas de un cubículo del IEC.

Este procedimiento se realizó con algunas ventanas de pasillos, pilares, tragaluces, escaleras, etc., con el fin de tener las mediciones de las diferentes partes que forman al IEC lo más precisa y real posible, para la realización del modelo 3D correspondiente.

Modelado 3D del IEC

Tal y como se comentó en capítulos anteriores el modelado 3D se realizó por medio del API JSR-184 (Mobile 3D Graphics o M3G), el cual permite realizar el diseño mediante dos modos: Immediate Mode, se basa en código para la realización de todo lo referente al mismo; y Retained Mode, se auxilia del uso de herramienta de diseño 3D para generar un archivo binario en formato M3G que contendrá todo lo referente al modelo.

Existe la posibilidad de trabajar el modelado 3D en los dos modos con los que cuenta el API-JSR-184, teniendo ambos una serie de ventajas y desventajas. A continuación se dará una breve descripción de las ventajas y desventajas del diseño 3D basado en cada modo:

- **Immediate Mode**
 - Ventajas
 - Reducción en el tamaño final de la aplicación que usará el modelo 3D (el diseño se basa únicamente en código con un peso máximo de 150 Kb).
 - Mayor facilidad en la optimización y realismo del modelado (uso de texturas con dimensiones reducidas sin perder calidad durante su aplicación).
 - Desventajas
 - Alta complejidad aún en diseños muy simples (todo el diseño es basado en código y el trazo debe ser en forma de triángulos consecutivos, como si hiciera sin alzar el lápiz).
 - Alta complejidad en lo referente a aplicación de efectos y animaciones dentro el modelado (se debe tener conocimientos muy precisos del código y la forma correcta de aplicarlos).

- **Retained Mode**

- Ventajas

- Mayor facilidad en todo tipo de diseño (el modelado se realiza por medio de una herramienta de diseño 3D).
 - Mayor facilidad en cuanto a la aplicación de efectos y animaciones dentro del modelado (estos se realizan en su mayor parte dentro de la herramienta de diseño 3D y se accede a ellos mediante pequeñas rutinas de código).
 - Amplia gama de herramientas 3D (tanto comerciales como gratuitas) que permiten la generación de modelos en archivos binarios con extensión M3G.
 - Simplicidad en el código en lo referente al modelo 3D dentro de la aplicación (el modelado se basa en herramientas de diseño 3D y sólo se usan pequeñas secciones de código para interactuar con éste).

- Desventajas

- Mayor dificultad en la optimización y realismo del modelado (uso de texturas con dimensiones mayores, que sólo pueden ser reducidas y aplicadas correctamente mediante técnicas de modelado dentro de la herramienta de diseño 3D).
 - Tamaño considerable de la aplicación que usará el modelo 3D (el archivo binario con extensión M3G que contiene el modelo puede tener un peso máximo de 2 Mb).

Se realizaron una serie de pruebas a los modos de diseño provistos por el API, lo que arrojó los siguientes resultados:

Tabla 7 Comparación hecha entre los modos de diseño del API JSR-184.

Actividad \ Modo	Retained Mode	Immediate Mode
<i>Diseño del modelo 3D de un cubo (tiempo en mins.)</i>	40 Mins. (tiempo usado para definir los arreglos a usar para crear el modelo, así como el trazo y texturizado del mismo)	10 Mins. (tiempo usado para crear el modelo en Blender, así como para el texturizado y optimización del mismo)
<i>Codificación (secuencias de código)</i>	29 diferentes secuencias de código necesarias para generar y mostrar el modelo	39 diferentes secuencias de código necesarias para generar y mostrar el modelo
<i>Renderizado (tiempo en mseg.)</i>	339 mseg. (mediante celular Motorola a1200, Figura 82) y 90 mseg. (mediante emulador, Figura 84).	209 mseg. (mediante celular Motorola a1200, Figura 83) y 40 mseg. (mediante emulador, Figura 85).



Figura 82 Ejemplo de renderizado M3G con Retained Mode en celular Motorola a1200.



Figura 83 Ejemplo de renderizado M3G con Immediate Mode en celular Motorola a1200.

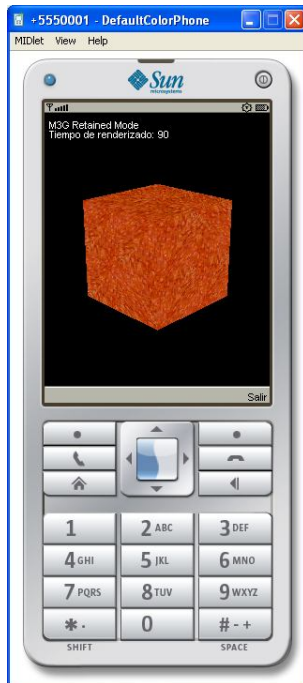


Figura 84 Ejemplo de renderizado M3G con Retained Mode en emulador.

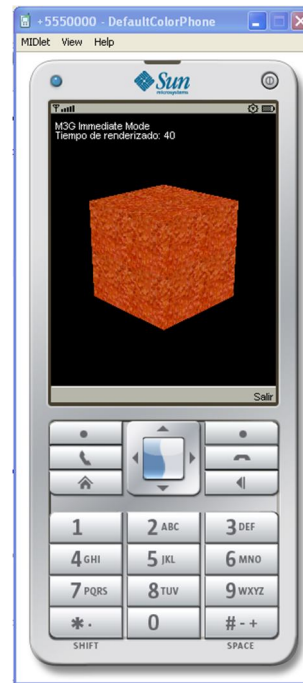


Figura 85 Ejemplo de renderizado M3G con Immediate Mode en emulador.

Nota: Para los tiempos mencionados en la tabla comparativa en lo referente al diseño del modelo cabe destacar que la persona que los realizó contaba con conocimientos previos de la programación en los modos, así como de la herramienta de diseño 3D usada.

Con base en lo anterior se optó por realizar el modelado 3D del edificio mediante el Retained Mode, ya que permite un diseño mucho más rápido y preciso, y aun cuando se sacrifica el realismo, esta pérdida puede ser reducida mediante el uso de técnicas de modelado que se aconsejan tanto por diseñadores como por programadores de aplicaciones 3D basadas en el API JSR-184⁴¹ y que fueron puestas a prueba para obtener el mayor realismo y a la vez la mayor optimización posible (para el modelo contenido dentro del archivo binario con extensión M3G).

Antes de comenzar a modelar es necesario tomar en cuenta que las coordenadas 3D (x,y,z) obtenidas con el GPS y usadas como referencia por la estación total para obtener puntos de medición de coordenadas, se encuentran dados en latitud, longitud y altitud del globo terráqueo por lo cual es necesario hacer una pequeña adaptación para ser usadas dentro de las coordenadas del modelado 3D, esto se logra obteniendo el mínimo valor de cada coordenada (para x,y,z) de entre todos los puntos obtenidos anteriormente para determinar las dimensiones del edificio y cuyos valores serán usados posteriormente para ser restados a cada punto de medición de coordenadas obtenidos hasta el momento.

Una vez adaptadas las coordenadas 3D (x, y, z) de los puntos de referencia del edificio, se procede a modelar al mismo mediante una herramienta de diseño 3D, la opción seleccionada fue la herramienta denominada “Blender” de acuerdo a las características y ventajas que proporciona ésta y que fueron mostradas en capítulos anteriores.

En primer lugar se procedió a construir la estructura interna del edificio (pasillos y escaleras) con base en los puntos de medición de coordenadas obtenidos y adaptados anteriormente, para ello se usaron 2 técnicas de modelado⁴² muy comunes denominadas “Modelado Geométrico” y “Modelado por Malla de Polígonos”, las cuales se intercalan entre sí. Como primer paso se agregará una

⁴¹ <http://www.oracle.com/technetwork/server-storage/ts-6258-158857.pdf>

⁴² http://www.futureworkss.com/tecnologicos/informatica/tutoriales/Manual_de_Blender.pdf

CAPITULO V: Toma de medidas para el modelado 3D del Instituto de Electrónica y Computación (IEC)

primitiva (objeto básico de malla) siendo la más adecuada para el caso la denominada “Plano”, se procede a ubicar el centro de la misma en el centro del espacio tridimensional del programa, posteriormente se modificará la malla del objeto con el propósito de eliminar la cara y líneas que unen a los vértices o puntos (Modelado por Malla de Polígonos), a continuación se duplicarán los vértices y se re-ubicará cada uno de acuerdo con los puntos de medición de coordenadas obtenidos repitiendo el proceso cuantas veces sea necesario (Modelado Geométrico), ya con esto se procura a crear las líneas que unirán cada vértice e igualmente las caras necesarias (considerando que cada cara es un plano formado por 3 o 4 vértices) con lo que finalmente se tiene construida la parte interna (Figura 86) del edificio (Modelado por Malla de Polígonos).

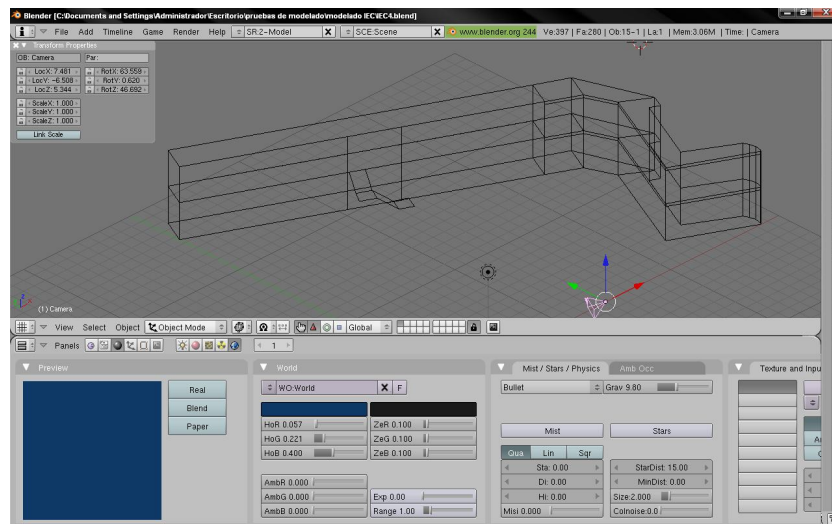


Figura 86 Modelo final de la estructura interna del IEC (pasillos).

En segundo lugar de acuerdo con las mediciones manuales obtenidas con el flexómetro de un cubículo y las observaciones hechas de la forma y ubicación de cada uno dentro del edificio, así como de las ventanas, pilares, puertas, tragaluces, escalares, etc., se procedió al diseño de cada uno mediante las técnicas usadas anteriormente aunado con las técnicas de “Modelado de Superficies de Subdivisión”, “Modelado por Triángulos”, “Modelado por Superficies Paramétricas” y “Modelador por Extrucciones”. Para ello en primer lugar las caras de los pasillos se subdividirán conforme al número y ubicación de cada cubículo (Modelador por Superficie de Subdivisión) y posteriormente se extruirán en las 3D dimensiones (x, y, z) con el fin de obtener la forma de los cubículos de acuerdo con las dimensiones obtenidas manualmente de los cubículos y caras generadas a partir de la estructura interna creada anteriormente (Modelado por Extrucciones). Ahora se procederá a crear la ventana y puerta del cubículo, para ello se extruirán las caras del cubículo donde se ubicara cada una de ellas (Modelado por Extrucciones), se re-ubicaran los vértices que formen las estructuras de la ventana y puerta de tal forma que sean lo más parecidas a las reales con respecto a la profundidad y formas de éstas (Modelado por Triángulos y Superficies Paramétricas), además de esto se incluirá un modelo de una mesa en cada cubículo (Figura 87 y Figura 88) la cual será creada a partir de planos, uniones, subdivisiones y extrucciones (Modelado por Malla de Polígonos, Geométrico, Superficies de Subdivisión, y Extrucciones). Por último se observarán las ubicaciones y dimensiones del resto de objetos que forman parte del edificio (ventanas, puertas, paredes, pilares, tragaluces, etc.), extruyendo y subdividiendo las caras de la estructura interna del edificio (Modelado por Extrucciones y Superficies de Subdivisión), uniéndolos con las primitivas necesarias (Planos y Círculos), eliminando y modificando las líneas y vértices necesarios dando forma y profundidad (Modelado por Malla de Polígonos, Geométrico, Triángulos y Superficies Paramétricas).

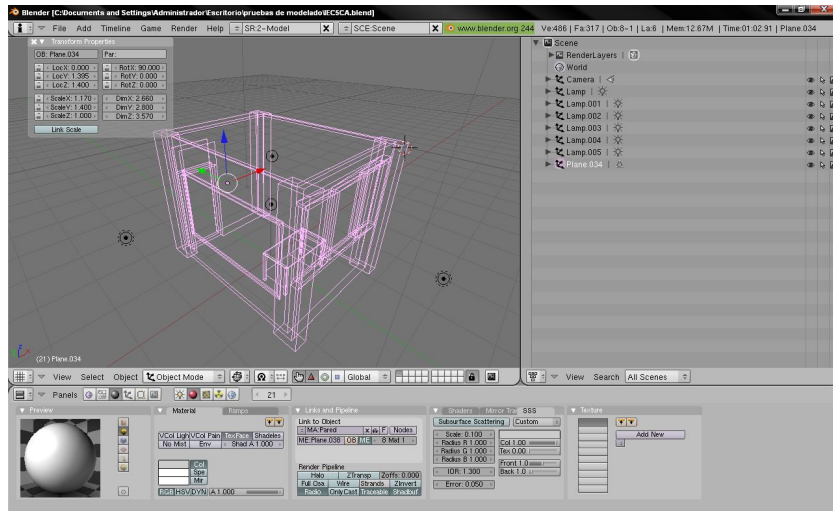


Figura 87 Visión de las líneas que forman la estructura del modelo de un cubículo del IEC.

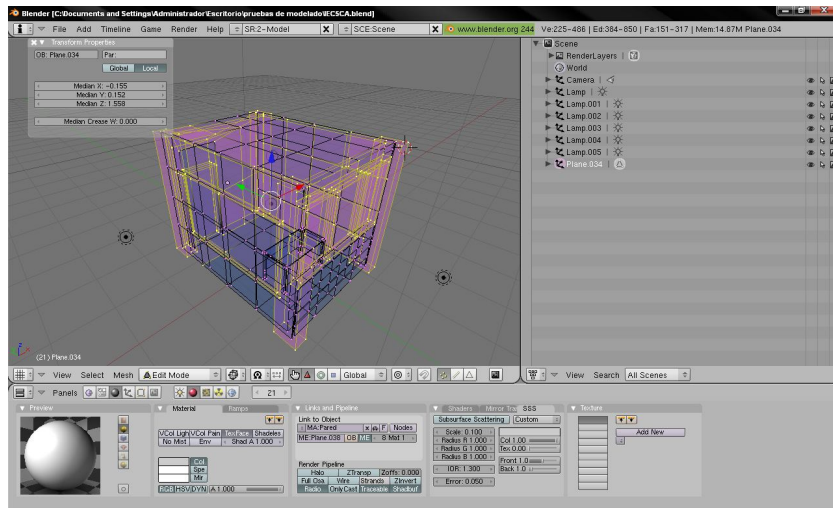


Figura 88 Visión de las líneas y vértices que forman la estructura del modelo de un cubículo del IEC.

Con las acciones anteriores se tiene completo el modelado 3D del edificio, sin embargo, el diseño no está completo aún, es necesario aplicar las texturas apropiadas con el fin de tener una mayor similitud con el edificio real, para ello se hizo una serie de tomas de fotografías de cada textura que se encuentra dentro del edificio, para posteriormente adaptarlas, con el fin de tener el mayor realismo y a la vez la mayor optimización posible (Figura 89, Figura 90 y Figura 91) del modelado 3D⁴³. Es importante mencionar que la exportación del modelo 3D desde la herramienta Blender a un archivo binario de formato M3G, soporta sólo algunos de los efectos propios de esta herramienta tales como todos los tipos de luces, multi-texturas, transparencia (canal Alpha), animaciones, mapeado de texturas UV, entre otros. La exportación se hace mediante un Plugin de desarrollo independiente⁴⁴ que fue incorporado en Blender a partir de la su versión 2.49a y cuya versión más actual del Plugin es la 0.91.

⁴³ http://mosaic.uoc.edu/wp-content/uploads/Manual_de_Introduccion_a_Blender.pdf

⁴⁴ <http://www.nelson-games.de/bl2m3g/default.html>

CAPITULO V: Toma de medidas para el modelado 3D del Instituto de Electrónica y Computación (IEC)

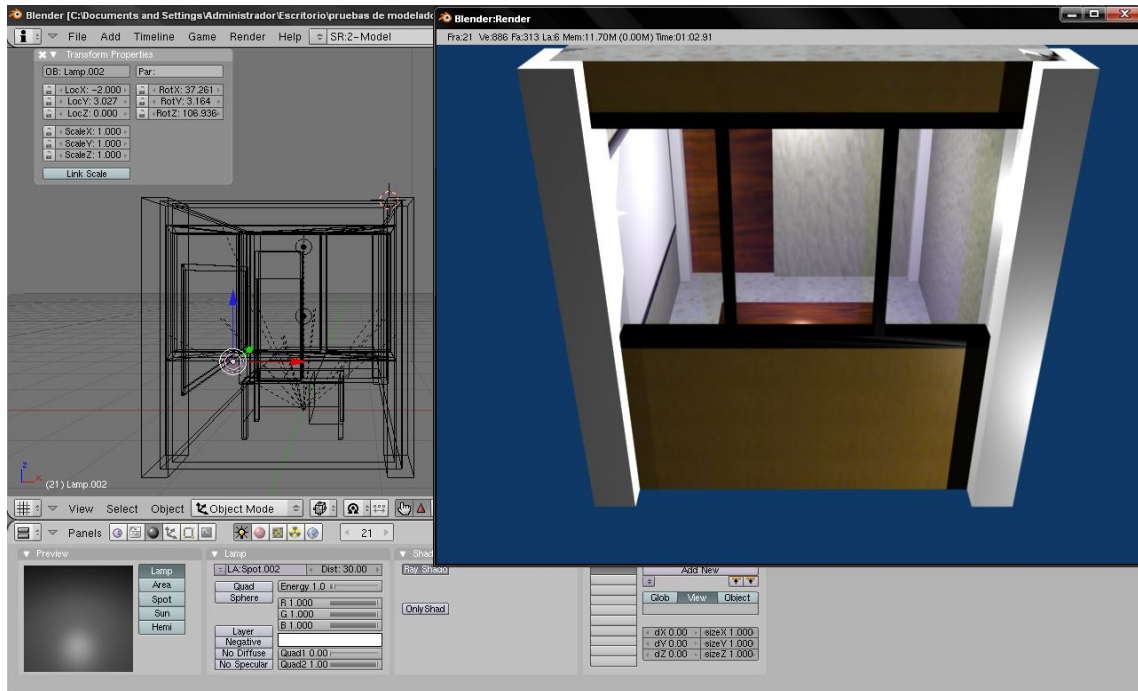


Figura 89 Renderizado del modelo del cubículo del IEC ya con la aplicación de las texturas y efectos necesarios.

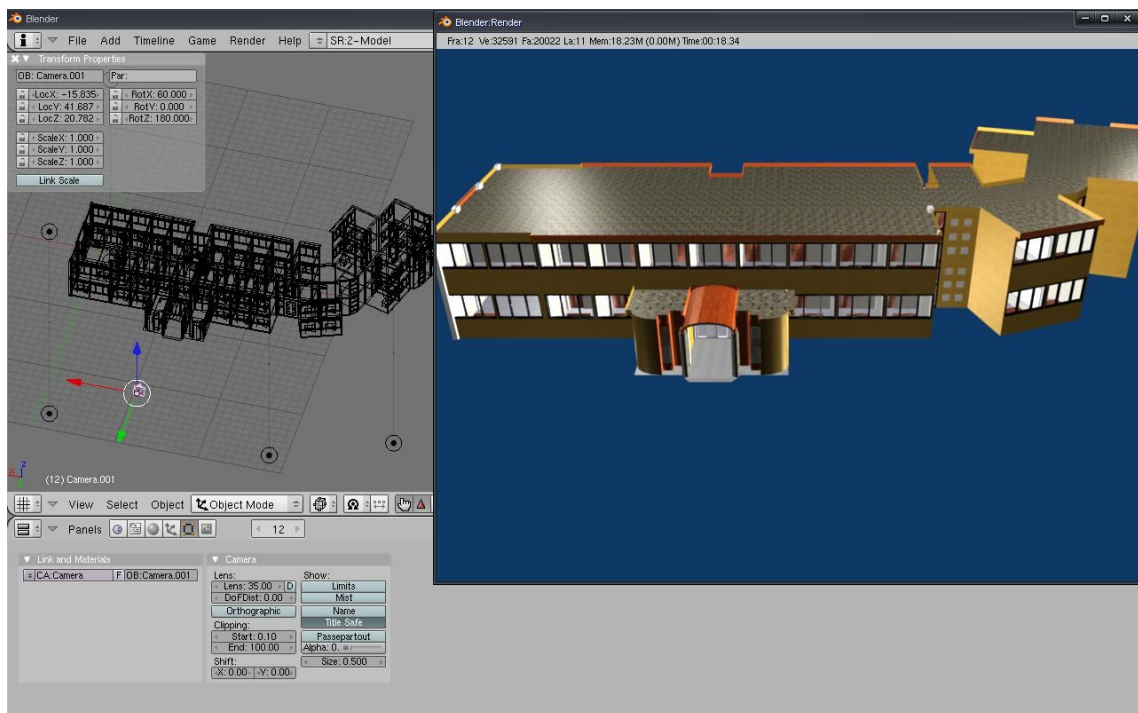


Figura 90 Renderizado del modelo completo del IEC.



Figura 91 Captura del modelo del cubículo del IEC ya con la aplicación de las texturas y efectos necesarios a) desde el Emulador Wireless Toolkit 2.5.2 y b) desde Celular Motorola a1200.

Nota: Durante el desarrollo del proyecto de tesis la versión con la que se contaba del Plugin era la 0.9, la cual no soportaba efectos probados como: la combinación de materiales/texturas en un mismo objeto (Mesh), suavizado de un objeto (Smoothing) y la transparencia de materiales/texturas, este último se solucionó cuando se tuvo contacto con el desarrollador del Plugin vía e-mail y el cual mandó una nueva versión 0.9 (modificada) que ya exportaba la transparencia.

CAPITULO VI: Desarrollo de la aplicación

Selección de tipo de Sistema y Métodos de Localización en Ambientes Internos

Con base en la información y ejemplos de los Sistemas y Métodos de Localización para Ambientes Internos mencionados hasta ahora y considerando el tipo de orientación que se desea dar al sistema de acuerdo a las características del mismo, se ha optado por realizar un sistema basado en radiofrecuencias (sin el uso de Tags RFID). Esta decisión se basa principalmente en el hecho que en la actualidad el estudio de los sistemas de localización para ambientes internos se ha inclinado por este tipo de sistemas (en ocasiones se pueden observar sistemas híbridos que usan otro tipo de sistema como apoyo) ya que el tipo de señal usadas por estos son los menos susceptibles a obstáculos y objetos comunes que se pueden encontrar en un ambiente interno, además, que el rango de cobertura es mayor que el de otros tipo de señales usadas por otros sistemas (aunque la exactitud y precisión varían de acuerdo a la señal y tecnología usada).

Como se comentó anteriormente, algunas de las características más importantes que se pueden destacar de este tipo de sistemas son:

- No se requiere de una infraestructura física especializada.
- Las tecnologías (en especial el Wi-Fi y el Bluetooth) que lo soportan son incluidas en la mayoría de los dispositivos móviles actuales.
- Es una alternativa variable para espacios cerrados.
- El uso de las tecnologías que lo soportan por parte del programador se ha hecho más sencilla en la actualidad.

Aunado a todos esto, para la implementación y pruebas del sistema se cuenta de ante mano con equipo que se orienta aún más a este tipo de sistemas. Las características del equipo móvil con que se cuenta son las siguientes:

- **Smartphone Motorola Symbol MC 35:**
 - *Sistema Operativo Windows:* Mobile 5.0 Phone Edition.
 - *Interfaz / Comunicaciones:* Bluetooth Ver. 1.2, IrDA, Wi-Fi 802.11 b/g, GSM/GPRS 850/900, 1800/1900 MHz) WAN y GPS.
 - *Pantalla:* Display TouchScreen color QVGA (320x240 píxeles).
 - *Procesador/Memoria:* Intel XScale PXA270 a 416 MHz y 64 MB SDRAM / 128 MB ROM (expansible por medio de SD hasta 4 GB).

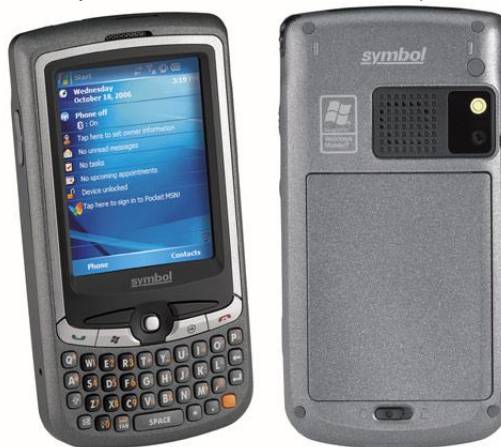


Figura 92 Imagen de un Smartphone Motorola Symbol MC 35.

- **Smartphone Motorola A1200r:**
 - *Sistema Operativo:* Mobile Linux 2.4.20.
 - *Interfaz / Comunicaciones:* Bluetooth Ver. 2.0 y GSM/GPRS (850/900, 1800/1900 MHz)
 - *Pantalla:* Display TouchScreen color QVGA (320x240 píxeles).
 - *Procesador/Memoria:* Intel XScale PXA270 a 312MHz y 64 MB SDRAM / 8 MB ROM (expandible por medio de microSD hasta 2 GB).



Figura 93 Imagen de un Smartphone Motorola A1200r.

Selección de la técnica de localización a usar en el sistema localización basado en radiofrecuencia

En base a las técnicas de localización usadas por los sistemas de localización basados en radio frecuencia y los ejemplos que implementan las mismas hasta ahora vistos, se ha optado por usar el método de reconocimiento de patrones KNN en el sistema de localización para ambientes internos que se desea implementar. Esto se debe principalmente al tipo de orientación y características que se desea, cumpla el sistema, destacando principalmente que se busca implementar un sistema en el que las tareas a realizar por la infraestructura sea mínima, es decir, las terminales fijas sólo servirán de antenas de detección y transmisión de algunos valores, así las terminales móviles serán las encargadas de realizar el cálculo de su localización y ubicación.

Por ello se optó por este tipo de técnica ya que en comparación con el resto, tiene la capacidad de proporcionar un cómputo del cálculo de localización de forma privada, una mayor exactitud y precisión sin requerir de gran capacidad de cómputo (ideal para terminales móviles), una mayor escala por unidad de infraestructura, así como una reducción en costos (capital, tiempo, espacio, etc.), cubriendo las características propias de un sistema de localización.

Además de las ventajas anteriores se facilitan otras como son, una mayor facilidad durante el desarrollo al no requerir de conocimientos altamente especializados y una reducción en el manejo de uno de los parámetros más problemáticos e importante en lo que se refiere a los sistemas de localización para ambientes internos: la alta variabilidad que tiene el uso de la radiofrecuencia para el cálculo de la localización. Esta variabilidad que afecta el RSS de señales de radiofrecuencia debido a diversos factores dentro de estos ambientes es muy importante que se considere al desarrollar un sistema de localización para ambientes internos, esta problemática se logra evitar, al hacer una aproximación de los posibles cambios en este parámetro por medio de métodos de reconocimiento de patrones en comparación con los métodos geométricos, que no consideran este problema o que si lo hacen es muy difícil de solucionar; por lo cual como se mencionó anteriormente entre los métodos de reconocimiento vistos hasta ahora se ha optado por del método KNN (que discretiza y mide previamente el parámetro en el ambiente donde se implementará el sistema), al ser una solución viable, no compleja y que con requerimientos computacionales mínimos.

CAPITULO VI: Desarrollo de la aplicación

Es importante destacar que en comparación con los demás métodos de reconocimiento de patrones no es necesario una etapa de entrenamiento sino tan sólo la etapa de recolección de datos en las localizaciones definidas por el usuario, con lo que se crea el radio mapa patrón que necesita el sistema, lo cual reduce como se comentó anteriormente, el costo en tiempo de implementación del sistema. Su falta de flexibilidad en cuanto al agregado o eliminación de terminales fijas para la implementación del sistema se ve compensado con la facilidad de modificación del código para incluir o eliminar estas terminales fijas; de igual forma, en lo referente a la reestructuración del radio mapa patrón sólo requiere del costo en tiempo de la realización nuevamente de la etapa de recolección de datos en comparación con otros métodos más flexibles equivale al mismo tiempo necesario para que estos sistemas se reajusten en la etapa de reentrenamiento.

Ejemplo de localización mediante el método KNN

A continuación se puede observar un ejemplo de cómo se solucionaría un problema de localización tridimensional mediante el uso del método KNN, para ello primero se obtiene el radio mapa patrón o mapa de intensidades que discretizan el espacio donde se hará la localización:

Localización	x (m)	y (m)	z (m)	RSS ₁ (dBm)	RSS ₂ (dBm)
1	0	0	0	-28	-88
2	5	0	0	-35	-70
3	10	0	0	-51	-62
4	0	5	0	-41	-85
5	5	5	0	-48	-68
6	10	5	0	-60	-50
7	10	10	0	-72	-30

Se asume que sólo existe un vecino que minimiza la distancia ($k=1$), por ello dado un vector de potencias obtenido, sólo existe un vector de prueba dentro del radio mapa patrón que minimiza la distancia. Por lo tanto, los k -vecinos más cercanos a y son $N_k(y)$, y $c(z)$ son la etiqueta de la clase z , la cardinalidad de $N_k(y)$ es igual a k y el número de clases es l . Entonces el subconjunto de los vecinos más cercanos dentro de la clase $j \in \{1, \dots, l\}$ es:

$$N_k^j(y) = \{z \in N_k(y) : c(z) = j\}$$

El resultado de las clasificación $\hat{j} \in \{1, \dots, l\}$ se define como el más votado:

$$\hat{j} = \arg \max_j |N_k^j(y)|$$

Ajustando la distancia Euclidiana para el método KNN, se tiene que la siguiente ecuación:

$$d(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\| = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$$

En donde $i=1, \dots, n$, n es el número de emisores-receptores fijos, x_i representa el valor RSS_i (valor del emisor-receptor fijo i) seleccionado del radio mapa patrón y y_i representa el valor RSS_i (valor del emisor-receptor fijo i) obtenido por el emisor-receptor móvil.

Suponiendo que el vector de potencias obtenido está compuesto por RSS₁=-40 y RSS₂=-62, al calcular la distancia Euclidiana del radio mapa patrón se obtiene lo siguiente:

CAPITULO VI: Desarrollo de la aplicación

Localización	x (m)	y (m)	z (m)	RSS ₁ (dBm)	RSS ₂ (dBm)	distancia
1	0	0	0	-28	-88	28,6356421
2	5	0	0	-35	-70	9,43398113
3	10	0	0	-51	-62	11
4	0	5	0	-41	-85	23,0217289
5	5	5	0	-48	-68	10
6	10	5	0	-60	-50	23,3238076
7	10	10	0	-72	-30	45,254834

Se puede observar que el vector de prueba (vecino) que minimiza la distancia es el de la localización 5, por lo tanto, se deduce que las coordenadas con mayor probabilidad donde puede encontrarse el receptor-emisor móvil son $x=5$, $y=5$ y $z=0$:

En ocasiones suele suceder que más de un vector de prueba minimizan la distancia (tienen la misma distancia euclidiana), cuando sucede este caso, se realiza un promedio de las coordenadas de los vectores de prueba para así hacer un estimado de la localización del emisor-receptor móvil.

Una modificación clásica de este método es denominado pesos de los K-vecinos más cercanos (WKNN, por sus siglas en inglés), se introducen pesos una vez que se obtiene los k vecinos, para combinarlos y obtener una localización intermedia entre estos.

Con esto la ecuación de la distancia Euclidiana se modificaría quedando de la siguiente forma:

$$d(X,Z) = \sqrt{\sum_{i=1}^m w_i (x_i - z_i)^2}$$

Donde w_i , representado el peso w de $i=1, \dots, n$, siendo n es el número de emisores-receptores fijos.

Selección del lenguaje de programación para dispositivos móviles y embebidos a usar para el desarrollo del sistema de localización para ambiente internos

Con base en la información mostrada anteriormente y de acuerdo a los requerimientos del sistema de localización para ambientes internos que se desea desarrollar, se observa (Tabla 8) que el lenguaje más acorde y con mayores prestaciones con que se cuenta es Java en su versión J2ME. Es importante mencionar que uno de los dos dispositivos móviles (Smartphone Motorola A1200r) con que se cuenta tiene S.O. nativo de este lenguaje, por lo cual se prestará interés a este dispositivo durante la programación e implementación, así como durante la etapa de pruebas.

Entre las características que se pueden considerar de mayor relevancia por lo cual fue seleccionado el lenguaje Java en su versión J2ME se puede mencionar la adaptabilidad que posee este lenguaje para ser llevado el ejecutable a una diversidad de S.O. para dispositivos móviles o embebidos, esto se debe principalmente a que el lenguaje tiene una mayor variedad de S.O. que dan soporte al mismo y que con el paso del tiempo ha sido incluido por los mismos desarrolladores del S.O. o por otros desarrolladores que desean llevar aplicaciones de este lenguaje a estos S.O. ya sea, con el fin de ver el potencial del lenguaje, el rendimiento del mismo, lograr una mayor estandarización, entre otras muchas razones.

CAPITULO VI: Desarrollo de la aplicación

Tabla 8 Comparativa entre lenguajes de programación para dispositivos móviles y embebidos.

Prestaciones \ Lenguaje	Java (J2ME)	C++ (Móviles/Embebidos)	C# (Móviles/Embebidos)
<i>S.O. compatible</i>	Palm OS, Symbian OS Nokia, Mobile Linux, Windows Mobile o CE y Solaris Operating Environment	Palm OS, Symbian OS Nokia, Mobile Linux y Windows CE	Symbian S60 2ª Edición o superior y Windows Embedded, Mobile, Automotives o CE
<i>Compilación</i>	Única (puede usar en cualquier S.O con implementación requerida CLDC/CDC de J2ME)	Para cada S.O.	Única (puede usar en cualquier S.O con implementación requerida .Net Compact Framework)
<i>Costo de compilador y ambiente de desarrollo</i>	Gratuito	De paga (compiladores con APIs mas completo) o gratuito (compiladores con APIs básicas)	Gratuito
<i>Estandarización de APIs</i>	Alta	Baja (casi no existe)	Media (algunas APIs dependen del S.O o hardware del dispositivo)
<i>Consumo de recursos de hardware</i>	Media	Baja	Alta
<i>Dificultad del lenguaje</i>	Media (para programadores Java)	Baja (para todo programadores)	Media (para programadores C# y visuales)
<i>Desarrollo de complementos y mejoras continuas del J2ME</i>	Alta (diversidad de desarrollos comerciales o gratuitos)	Baja (pocos desarrollos casi exclusivos de Palm Inc.)	Media (diversidad de desarrollos comerciales o gratuitos)

Bluetooth

Por qué usar la tecnología Bluetooth y no Wi-Fi

En base a estudios hechos por todo el mundo sobre las características, dispersión, usos y aplicaciones que se pueden dar a las tecnologías Bluetooth y Wi-Fi, y concordé al objetivo y características propias del sistema de localización para ambientes internos que se desea implementar se ha llegado a la conclusión de que la tecnología Bluetooth es la más adecuada para este propósito algunos de los factores que incurrieron en esta decisión pueden ser observados en la siguiente tabla:

Tabla 9 Comparativa de características de la tecnología Bluetooth y Wi-Fi.

Características \ Tecnología	Bluetooth	Wi-Fi
<i>Alcance aproximado (potencia máxima en mW)</i>	Clase 1: 100 m (100 mW) Clase 2: 10 m (2.5 mW) Clase 3: 1 m (1 mW)	100 m (100 mW)
<i>Consumo aproximado de potencia</i>	40 ma transmitiendo y 0.2 ma en reposo	400 ma transmitiendo y 20 ma en reposo
<i>Dificultad de la conexión entre dispositivos</i>	Fácil	Media (requiere de configuración de los nodos)
<i>Numero de dispositivos móviles que usan la tecnología</i>	Alta (cerca del 75% de los dispositivos móviles en el mundo poseen esta tecnología)	Media (cerca del 45% de los dispositivos móviles en el mundo poseen esta tecnología)
<i>Grado de uso de la tecnología en sistemas de localización</i>	Baja-Media	Media-Alta
<i>Precisión en localización</i>	1 – 10 m	10 – 100 m
<i>Principal ventaja</i>	Interpolaridad	Gran ancho de banda
<i>Velocidad de transferencia</i>	Media	Alta
<i>Complejidad de programación de aplicaciones basadas en la tecnología</i>	Media	Alta
<i>Mejora de la tecnología</i>	Constante	Casi nunca

Otras características que se consideraron para la selección de la tecnología Bluetooth sobre la tecnología Wi-Fi, es que por medio de estudio realizado (Golmie, Van Dyck y Soltanian) y (Patil, Kim y Ni) se puede observar que aunque Bluetooth y Wi-Fi trabajan en la misma banda de radiofrecuencia (2.4 GHz), las señales enviadas por dispositivos Wi-Fi son más susceptibles a interferencia debido a señales de

CAPITULO VI: Desarrollo de la aplicación

dispositivos Bluetooth, lo que no sucede en el sentido contrario, esto se debe a que Bluetooth usa la técnica de modulación denominada Espectro Ensanchado por Salto de Frecuencia (FHSS, Frequency Hopping Spread Spectrum), lo que permite a los dispositivos Bluetooth dividir el rango de frecuencia sobre el trabajan en 79 pequeños canales, estos dispositivos cambian el canal sobre el trabajan 1600 veces por segundo de acuerdo al patrón establecido cuando se inició la red, lo que permite no sólo reducir la interferencia de señales provenientes de dispositivos Wi-Fi, sino también de otros dispositivos que emitan señales sobre la misma banda (2.4 GHz). Aunado a todo esto la tecnología Bluetooth provee de 3 clases diferentes de dispositivos de acuerdo a la potencia, permitiendo tomar una decisión sobre la clase más adecuada en términos de potencias de salida y rango, que permite de igual forma reducir la interferencia de otros dispositivos. Los estudios de interferencia mutua, arrojaron en el peor escenario que Wi-Fi llega a tener una pérdida de 65% de sus paquetes, mientras que para Bluetooth es de tan sólo del 8%.

Aunque la tecnología Wi-Fi provee de un mayor ancho de banda con una tasa de transferencia de hasta 54 Mb/s y en comparación con Bluetooth que sólo proporciona 700 Kb/s en promedio, no es posible procesar en totalidad la gran cantidad de datos entrantes por muchos de los dispositivos móviles y embebidos (en especial los Celulares), debido a esto no se hace un óptimo aprovechamiento del potencial de la tecnología Wi-Fi y por lo que se puede decir la tecnología Bluetooth es más que suficiente para cualquier aplicación de red inalámbrica orientada a esta clase de dispositivos, esto en conjunto con la cantidad de potencia consumida por cada tipo de tecnología da un claro panorama de el por qué el uso predominante de la tecnología Bluetooth sobre la tecnología Wi-Fi en lo que se refiere a dispositivos móviles y embebidos, los cuales tienen un tiempo limitado de funcionamiento en base a la carga que provee la pila que usa cada uno.

Otros estudios (Elnahrawy, Li y Martin) se enfocan en el error medio cometido durante la localización basada en técnicas de reconocimiento de patrones usan WLAN (tecnología Wi-Fi). Estos estudios se enfocan en el uso de algoritmos de localización que se pueden clasificar principalmente dentro de dos tipos:

- **Algoritmos basados en puntos:** Devuelve el punto probable de localización estimada para el dispositivo móvil. Dentro de los cuales tenemos: Simple Punto de Coincidencia, Área Base de Probabilidad y Red Bayesiana.
- **Algoritmos basados en áreas:** Devuelve un área o volumen probable de localización estimada donde se puede encontrar ubicado el dispositivo móvil, algunos de los algoritmos usados normalmente pueden ser: punto bayesiano, promedio bayesiano, radar, radar promediado, radar cuadrículado, probabilidad más alta, promedio más alto de probabilidad y mayor probabilidad de rejillas.

El aporte principal de estos estudios es que no importando la complejidad del algoritmo ni la densidad de sensores instalados, el error medio en la localización no baja de 3m, es decir se tiene una exactitud de hasta 3m con una precisión de 95% usando un algoritmo de localización altamente complejo y una red muy densa de sensores, lo que en la vida real no es factible debido a los grandes costos que conlleva este tipo de sistema de localización. En general un sistema de localización factible usando una red de sensores de tamaño óptimo y cualquier algoritmo de localización anteriormente mencionado arroja una precisión de 10m con una exactitud del 97%. Estos estudios indican que el error medio no proviene en sí de los algoritmos de localización usados, sino de la propia incertidumbre existen en cuanto a la medición de RSS de la señal de los dispositivos WLAN.

Otros estudios (Kotanen, Hännikäinen y Leppäkoski) y (Bandara, Hasegawa y Inoue) enfocados en la localización usando tecnología Bluetooth e igual que en los estudios sobre WLAN, usando técnicas de reconocimiento de patrones (algoritmo de Niveles de Potencia de Posicionamiento Basado en RX), aportan información relevante sobre la precisión, la cual se encuentra entre 3 y 4 m con una exactitud del 92%, e igual que en caso anterior la mejora en la precisión de sistema de localización no está enfocada principalmente en la complejidad de los algoritmos de localización, sino que esta se enfoca en la mejora del hardware, es decir para tener una mayor precisión en un sistema de localización se debe mejorar el hardware encargado de medir el RSS de la señales Bluetooth para eliminar o reducir la incertidumbre en las mediciones.

CAPITULO VI: Desarrollo de la aplicación

Lo interesante de todos estos estudios es que proveen información relevante con respecto al porqué de la precisión de los sistemas de localización tanto para los que usan tecnología Wi-Fi como para los que usan la tecnología Bluetooth (Figura 94), e igualmente es importante destacar que estos resultados obtenidos a través de estos estudios no sólo han sido calculados matemáticamente y por simulación, sino que han sido llevados a la implementación en ambientes reales, como laboratorios y oficinas de 1 y 2 pisos, proveyendo de toda clase de obstáculos e interferencias a las que se enfrentará todo sistema de localización implementado en cualquier ambiente real, lo que da un gran respaldo a los resultados y conclusiones obtenidas.

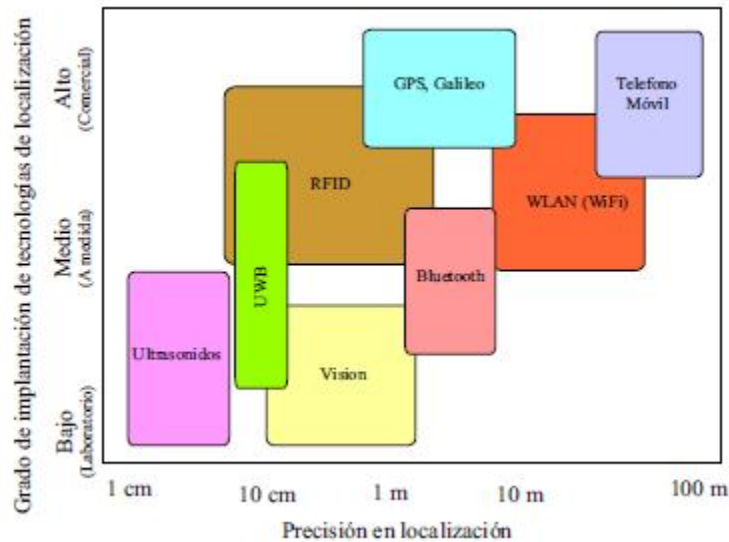


Figura 94 Visión general de las tecnologías usadas por los sistemas de localización.

Selección de la implementación JSR-82 en J2SE a usar para el desarrollo del sistema de localización para ambiente internos

De las implementaciones JSR-82 en J2SE vistas hasta el momento, el tipo de API más adecuado para el proyecto de tesis son las APIs de Java que se ejecutan sobre pila de Bluetooth Externas ya que aunque éstas dependen del Sistema Operativo, no bloquean el acceso al dispositivo Bluetooth para otras aplicaciones, además de que algunas de estas APIs pueden trabajar sobre varios Sistemas Operativos, un objetivo del proyecto de tesis es poder realizar una aplicación portable a diferentes dispositivos móviles donde se incluye a las Laptop y otros dispositivos que tienen Sistemas Operativos que trabajan con el lenguaje Java en su versión J2SE.

Además de lo anteriormente mencionado, es necesario centrarse en la arquitectura y funcionamiento de la aplicación a desarrollar para el proyecto de tesis, la funcionalidad principal que se persigue de las implementaciones JSR-82 en J2SE, es obtener el RSS de un dispositivo Bluetooth ya que en sí la implementación JSR-82 (desarrollada originalmente para Java en su versión J2ME) trabaja de forma reducida, ocultando al desarrollador mucha información por medio de la interfaces y clases que ofrece una mayor sencillez para el desarrollo de aplicaciones basadas en Bluetooth, y que como consecuencia algunas acciones deseadas por el desarrollador no son de acceso posible, entre ellas la obtención del RSS.

Por tal motivo se recurre a las implementaciones JSR-82 en J2SE que sigue el mismo estándar dentro del API (es decir tienen las mismas clases e interfaces que JSR-82 en J2ME), pero que además incluyen clases e interfaces extras.

En base al análisis hecho sobre las APIs Bluetooth y de acuerdo a los requerimientos anteriormente mencionado, es importante centrarse principalmente en dos APIs. BlueCove y AvetanaBT.

CAPITULO VI: Desarrollo de la aplicación

La primera API está disponible en su totalidad (para todos los Sistemas Operativos sobre los que trabaja) de forma gratuita, posee una total compatibilidad con JSR-82 de J2ME (permitiendo la reutilización de aplicaciones y código diseñado para J2ME) y trabaja tanto sobre L2CAP como con RFCOMM, no es posible la obtención de RSS desde ninguna de las clases e interfaces extras que incluye.

La segunda API al igual que la primera es totalmente compatible con JSR-82 de J2ME e igualmente trabaja sobre L2CAP y RFCOMM, sin embargo, sólo se tiene acceso de forma gratuita a la implementación compatible con el Sistema Operativo Linux (además de que es necesario compilarlo ya que se proporciona el código de compilación), para los otros Sistemas Operativos se puede obtener un versión de prueba (que trabaja sólo por 15 días) vía correo electrónico o una versión comercial por el precio de 25 € (tanto la versión de prueba como la comercial sólo trabajan con 3 MAC de dispositivos por programa), a pesar de esta desventaja esta API es una de las completas que además de incluir las clases e interfaces propias de JSR-82 de J2ME incluye un mayor número de clases e interface extras entre las cuales existen algunas que dan acceso al RSS de un dispositivo Bluetooth remoto. Por lo cual en conclusión la API seleccionada para el desarrollo de la aplicación a desarrollar en el proyecto de tesis es AvetanaBT.

Exploración

En todo proyecto antes de comenzar a codificar es necesario tener una idea clara de lo que necesita el cliente, es decir, cuales son los valores de negocio que el cliente desea obtener a través del sistema, es necesario como primer punto del proyecto, obtener los requerimientos del cliente, para ello en base a lo comentado en capítulos anteriores sobre la metodología ágil XP se cuenta con esta fase, cuyo objetivo es la de recolectar los requisitos del cliente, esto se puede hacer mediante las historias de usuarios, las cuales son una representación rápida, fácil y entendible de lo que el usuario desea que el sistema satisfaga de forma adecuada.

Historia de Usuario	
Numero: 1	Nombre: Obtención de RSSI
Usuario: Usuario de dispositivo móvil	
Modificación de historia numero:	Iteración asignada: 1
Prioridad en negocio: Alta (Alta/Media/Baja)	Puntos estimados: 5
Riegos en desarrollo: Alto (Alto/Medio/Bajo)	Puntos reales: 5
Descripción: El usuario inicia la aplicación desde el dispositivo móvil, obteniendo desde los módulos Bluetooth conectados a las terminales fijas los valores del RSSI correspondientes. Los valores son enviados al dispositivo móvil, el cual los muestra en pantalla.	
Observaciones: Esta tarea necesita la realización de dos aplicaciones una para el dispositivo móvil y otra para terminal fija. La aplicación de la terminal móvil realiza una búsqueda de los módulos Bluetooth de una lista precargada.	

Historia de Usuario	
Numero: 2	Nombre: Carga y Visualización del Modelado 3D
Usuario: Usuario de dispositivo móvil	
Modificación de historia numero:	Iteración asignadas: 2
Prioridad en negocio: Alta (Alto/Medio/Bajo)	Puntos estimados: 5
Riegos en desarrollo: Medio (Alto/Medio/Bajo)	Puntos reales: 5
Descripción: El usuario iniciará la aplicación, se procederá a la carga y visualización del modelado 3D en el dispositivo móvil.	
Observaciones: Antes de proceder al cargado y visualización del modelado 3D, es necesario desplegar en pantalla una presentación visual (gráfica y textual) que explica el funcionamiento de la aplicación.	

CAPITULO VI: Desarrollo de la aplicación

Historia de Usuario	
Numero: 3	Nombre: Localización en interiores
Usuario: Usuario de dispositivo móvil	
Modificación de historia numero: 1 y 2	Iteración asignadas: 3
Prioridad en negocio: Alta (Alto/Medio/Bajo)	Puntos estimados: 4
Riegos en desarrollo: Alta (Alto/Medio/Bajo)	Puntos reales: 4
Descripción: El usuario iniciará la aplicación, se procederá a cargar y posteriormente visualizar el modelado 3D, enseguida se obtendrán los valores RSSI de los módulos Bluetooth conectados a las terminales fijas, se podrá hacer el cálculo de la posible localización del dispositivo móvil.	
Observaciones: Esta aplicación es la unión y mejora de las historias de usuario 1 y 2. Para el cálculo de la localización se hace uso de un radio mapa patrón que es pre-cargado cuando se inicia la aplicación y que se encuentra en una base de datos del manejador Perst Lite.	

Historia de Usuario	
Numero: 4	Nombre: Obtención de RSSI
Usuario: Usuario de dispositivo móvil	
Modificación de historia numero: 1	Iteración asignadas: 4
Prioridad en negocio: Alta (Alto/Medio/Bajo)	Puntos estimados: 3
Riegos en desarrollo: Medio (Alto/Medio/Bajo)	Puntos reales: 4
Descripción: El usuario inicia la aplicación desde el dispositivo móvil, obteniendo desde los módulos Bluetooth conectados a las terminales fijas los valores del RSSI correspondientes. Los valores son enviados al dispositivo móvil, los muestra en pantalla. En caso de que no sea posible obtener el valor del RSSI de alguno de los módulos Bluetooth, el parámetro se completa con un valor de referencia.	
Observaciones: Esta tarea necesita la realización de dos aplicaciones una para la terminal móvil, y otra para terminal fija. La aplicación de la terminal móvil intentará establecer comunicación con los módulos Bluetooth de la lista precargada, no será necesario realizar la búsqueda de los mismos.	

Historia de Usuario	
Numero: 5	Nombre: Carga, visualización y recorrido del modelado 3D
Usuario: Usuario de dispositivo móvil	
Modificación de historia numero: 2	Iteración asignada: 5
Prioridad en negocio: Alta (Alto/Medio/Bajo)	Puntos estimados: 3
Riegos en desarrollo: Bajo (Alto/Medio/Bajo)	Puntos reales: 4
Descripción: El usuario iniciará la aplicación, procederá a la carga y visualización del modelado 3D en el dispositivo móvil.	
Observaciones: Se podrá navegar dentro del modelado 3D por medio de las teclas de dirección del dispositivo móvil.	

Historia de Usuario	
Numero: 6	Nombre: Carga, visualización y recorrido del modelado 3D
Usuario: Usuario de dispositivo móvil	
Modificación de historia numero: 5	Iteración asignada: 6
Prioridad en negocio: Alta (Alto/Medio/Bajo)	Puntos estimados: 3
Riegos en desarrollo: Bajo (Alto/Medio/Bajo)	Puntos reales: 3
Descripción: El usuario iniciará la aplicación, se procederá a la carga y visualización del modelado 3D en el dispositivo móvil.	
Observaciones: Se podrá navegar dentro del modelado 3D por medio de las teclas de dirección del dispositivo móvil, el modelo mostrará siempre de forma gráfica la posición estimada del dispositivo móvil por medio de una figura geométrica 3D y un rótulo con las coordenadas locales X,Y,Z en la parte superior en la pantalla.	

CAPITULO VI: Desarrollo de la aplicación

Historia de Usuario	
Numero: 7	Nombre: Localización en interiores
Usuario: Usuario de dispositivo móvil	
Modificación de historia numero: 3, 4 y 6	Iteración asignada: 7
Prioridad en negocio: Alta (Alto/Medio/Bajo)	Puntos estimados: 2
Riegos en desarrollo: Baja (Alto/Medio/Bajo)	Puntos reales: 3
Descripción: El usuario iniciará la aplicación la cual mostrará una pantalla inicial explicando el funcionamiento de la aplicación, una vez terminada la presentación se procederá a cargar y posteriormente visualizar el molado 3D, enseguida se obtendrán los valores RSSI de los módulos Bluetooth conectados a las terminales fijas con lo que se podrá hacer un cálculo de la localización probable del dispositivo móvil, concluido el cálculo se mostrara de forma gráfica la posición donde se encuentra por medio de una figura geométrica 3D y un rótulo con las coordenadas locales X,Y,Z en la parte superior de la pantalla. Al salir de la aplicación todos los procesos correspondientes a la misma (tanto por el dispositivo móvil como por la terminal fija) se detendrán para evitar cualquier tipo de error posterior.	
Observaciones: Esta aplicación es la unión y mejora de las historias de usuario 3, 4 y 6. Para el cálculo de la localización se hace uso de un radio mapa patrón que es pre-cargado cuando se inicia la aplicación y que se encuentra en una base de datos de Perst Lite.	

Con base en lo mostrado hasta el momento se puede observar que los requisitos del cliente para una primera versión del sistema se pueden agrupar en tres aplicaciones principales que fueron desarrolladas y mejoradas dentro del proyecto a través de 7 iteraciones, representada cada una con una historia de usuario (la cual refleja los valores de negocios deseados por el usuario y los cambios en éstos que este percibió como necesarios).

Aunado a todo esto, otra parte importante de la fase de exploración es la de revisar y utilizar las tecnologías, herramientas y prácticas que intervendrán durante el todo desarrollo del proyecto, en un principio, este punto no tuvo gran problemática ya que se tenían conocimientos básicos sobre éstas (ya se contaba con conocimientos previos del lenguaje J2ME, la tecnología Bluetooth, el estándar JSR-82 y la metodología de desarrollo XP), por lo cual durante la realización de las primeras historias de usuario y su respectivas iteraciones no se presentaron grandes dificultades, sin embargo, otras historias necesitan de cubrir aspectos totalmente desconocidos (el manejo del estándar JSR-184, la herramienta de diseño 3D Blender y la base de datos orientada a objetos Perst Lite) y que se fueron conociendo y experimentando con ellos a medida que la necesidad de su uso fueron surgiendo a través de cada iteración. Debido a esto la fase de exploración requirió un poco más de 6 meses (158 días) para la entrega de primera versión del sistema, esto se puede observar por medio del cronograma de actividades (Figura 95 y Figura 96) correspondiente al proyecto.

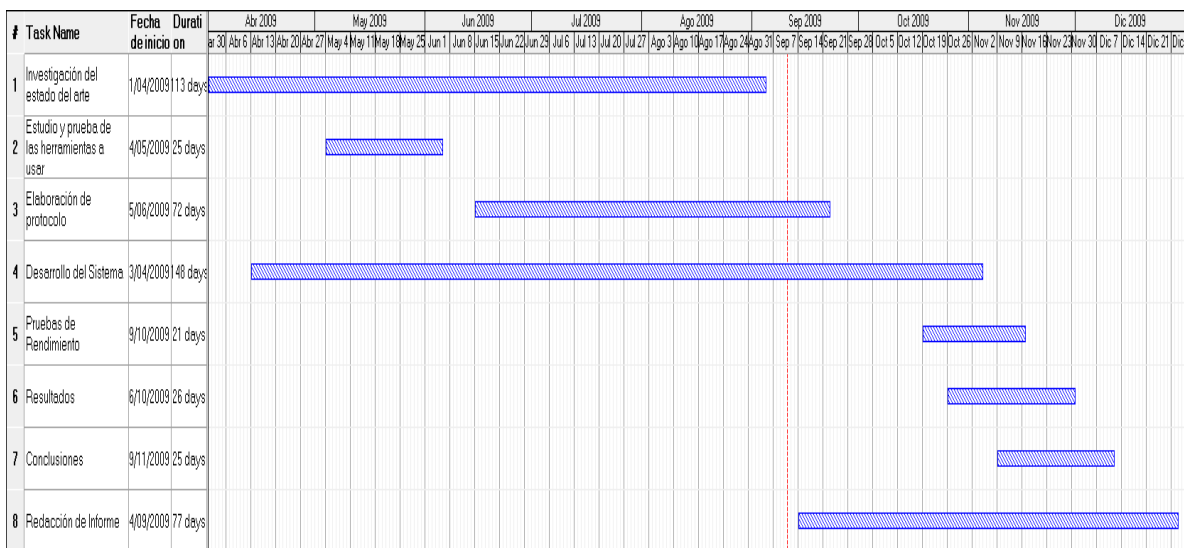


Figura 95 Cronograma de actividades del proyecto de Tesis.

CAPITULO VI: Desarrollo de la aplicación

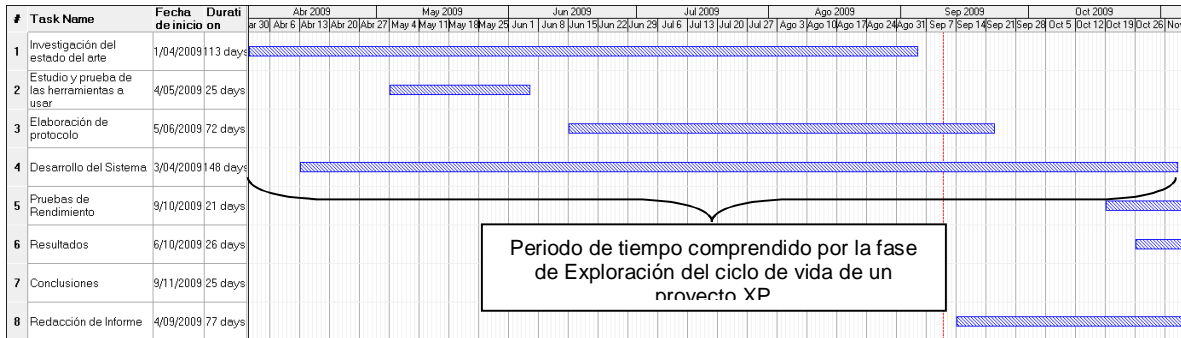


Figura 96 Sección del cronograma de actividades correspondiente a la fase de Exploración de la primera versión del sistema.

Es importante recordar que aunque dentro del ciclo de vida de un proyecto XP se percibe como si esta fase sólo se llevara a cabo una sola vez dentro del mismo, es común que esta se lleve a cabo varias veces, esto se debe al propio enfoque dado por XP para el ciclo de desarrollo y por tratarse de una metodología ágil, un punto principal de estas metodológicas es la de adaptarse a los cambios que surjan a través del proyecto.

Planificación de la Entrega e Iteraciones

En la fase anterior se pudo observar las historias de usuarios que intervinieron dentro del desarrollo de la primera versión del sistema correspondiente al proyecto de tesis, sin embargo es importante mencionar como fueron planteadas las entregas y prioridades de cada historia de usuario, así como las estimaciones del tiempo requerido para el desarrollo de cada una de ellas.

En un principio se tomó en cuenta tres historias de usuario que comprendía los componentes principales del sistema, estas historias de usuarios son la 1, 2 y 3, tomando en cuenta el tiempo calculado para cada una de ellas, se optó por la asignación de una iteración (iteraciones de tiempo variable sin exceder 5 semanas ideales de programación) para cada una de ellas. En este caso particular se optó por ampliar el tamaño de la iteración hasta 5 semanas a diferencia de las 4 semanas máximas que recomienda la metodología XP, debido a que el equipo de desarrollo del proyectos de tesis está formado por una sola persona la cual desempeña los diversos papeles necesarios dentro de la propia metodología.

Conforme a las necesidades y requerimientos en el desarrollo de cada una de las historias de usuario mencionadas previamente se obtuvieron las siguientes tareas:

Tarea	
Numero de tarea: 1	Numero de historia: 1
Nombre de tarea: Programación de parámetro RSSI	
Tipo de tarea: Desarrollo (Desarrollo/Corrección/Mejora/Otra)	Puntos estimados: 5
Fecha de inicio: 03/04/09	Fecha fin: 07/05/09
Programador responsable: Omar Roberto Cruz Ortega	
Descripción: Durante esta tarea se desarrollarán las aplicaciones tanto para la terminal móvil como fija, necesarias para cubrir las necesidades del cliente especificadas en la historia de usuario #1:	

CAPITULO VI: Desarrollo de la aplicación

Tarea	
Numero de tarea: 2	Numero de historia: 2
Nombre de tarea: Programación de modelado 3D	
Tipo de tarea: Desarrollo (Desarrollo/Corrección/Mejora/Otra)	Puntos estimados: 5
Fecha de inicio: 08/05/09	Fecha fin: 11/06/09
Programador responsable: Omar Roberto Cruz Ortega	
Descripción: Durante esta tarea se desarrollará la aplicación de la terminal móvil necesaria para cubrir las necesidades del cliente especificadas en la historia de usuario #2.	

Tarea	
Numero de tarea: 3	Numero de historia: 3
Nombre de tarea: Programación de la localización en interiores	
Tipo de tarea: Desarrollo (Desarrollo/Corrección/Mejora/Otra)	Puntos estimados: 4
Fecha de inicio: 12/06/09	Fecha fin: 09/07/09
Programador responsable: Omar Roberto Cruz Ortega	
Descripción: Durante esta tarea se mejora la aplicación de la terminal móvil para cubrir las necesidades del cliente especificadas en la historia de usuario #3.	

Para el momento en que completo la tarea 3, se observó que se necesitan ciertas mejoras a las aplicaciones desarrolladas, surgiendo en ese momento las historias de usuarios 4 y 5, se procedió con las siguientes tareas:

Tarea	
Numero de tarea: 4	Numero de historia: 4
Nombre de tarea: Mejora del parámetro RSSI	
Tipo de tarea: Mejora (Desarrollo/Corrección/Mejora/Otra)	Puntos estimados: 3
Fecha de inicio: 10/07/09	Fecha fin: 06/08/09
Programador responsable: Omar Roberto Cruz Ortega	
Descripción: Durante esta tarea se mejora la aplicación de la terminal móvil que cubrió en un principio las necesidades de la historia de usuario #1 conforme a las nuevas necesidades del cliente especificadas en la historia de usuario #4.	

Tarea	
Numero de tarea: 5	Numero de historia: 5
Nombre de tarea: Mejora del modelado 3D	
Tipo de tarea: Mejora (Desarrollo/Corrección/Mejora/Otra)	Puntos estimados: 3
Fecha de inicio: 07/08/09	Fecha fin: 03/09/09
Programador responsable: Omar Roberto Cruz Ortega	
Descripción: Durante esta tarea se mejora la aplicación de la terminal móvil que cubrió en un principio las necesidades de la historia de usuario #2 conforme a las nuevas necesidades del cliente especificadas en la historia de usuario #5.	

Durante la realización de las tareas 4 y 5 se tuvo un error de 1 punto en cada una de ellas, debido a esto en las fechas de desarrollo de cada una de ellas se observa el incremento de 1 semana ideal de programación. Para cuando la tarea de 5 había finalizado se observó la necesidad de una mejora en la misma por lo cual nuevamente surgió una nueva historia de usuario siendo esta la número #6, cuya tarea correspondiente es la siguiente:

CAPITULO VI: Desarrollo de la aplicación

Tarea	
Numero de tarea: 6	Numero de historia: 6
Nombre de tarea: Segunda mejora del modelado 3D	
Tipo de tarea: Mejora (Desarrollo/Corrección/Mejora/Otra)	Puntos estimados: 3
Fecha de inicio: 04/09/09	Fecha fin: 24/09/09
Programador responsable: Omar Roberto Cruz Ortega	
Descripción: Durante esta tarea se mejora la aplicación de la terminal móvil que cubrió en un principio las necesidades de la historia de usuario #5 conforme a las nuevas necesidades del cliente especificadas en la historia de usuario #6.	

En esta ocasión los puntos estimados para la historia de usuario fueron correctos y considerando que estas dos aplicaciones bases no necesitaban de más mejoras se tenía casi completo el sistema, sin embargo surgieron ciertas mejoras, se realizó una nueva historia de usuario siendo la #7, cuya tarea correspondiente es:

Tarea	
Numero de tarea: 7	Numero de historia: 7
Nombre de tarea: Mejora de la localización en interiores	
Tipo de tarea: Desarrollo (Desarrollo/Corrección/Mejora/Otra)	Puntos estimados: 2
Fecha de inicio: 25/09/09	Fecha fin: 15/10/09
Programador responsable: Omar Roberto Cruz Ortega	
Descripción: Durante esta tarea se mejora la aplicación de la terminal móvil para cubrir las necesidades del cliente especificadas en la historia de usuario #7.	

Durante esta última tarea el tiempo estimado para la historia de usuario fue incorrecto, por lo cual, fue necesario para la terminación de la misma una semana ideal de programación más, esto se puede observar a través de las fechas de desarrollo correspondientes.

Además como se puede observar en el cronograma de actividades (Figura 95) correspondiente al proyecto de tesis la etapa de desarrollo acaba el 04/11/09 fecha correspondiente a una parte de la fase de producción, esto se debe a que para ese momento se llega a la etapa donde el programa a pesar de estar considerado como finalizado en lo referente a la programación (se han cumplido con las tareas que cubren las historias de usuarios y sean realizado las pruebas necesarias), es necesario realizar pequeñas modificaciones, agregados y eliminaciones en el código que permitan mejorar al sistema y que cumplan con los requisitos del cliente, sin embargo estas acciones no necesitan de historias de usuarios y sus correspondiente tareas para ser llevadas a cabo ya que son ideas que surgen en el momento por parte del cliente o el equipo de desarrollo y que se probaran para observar si son fiables y aportan una mejora, y en el caso de no serlo, se darán por olvidado; por esta razón es que se observan en el cronograma un par más de semanas ideales de programación.

Estándar de Codificación

El objetivo de realizar un estándar de codificación es lograr la homogeneidad entre los componentes realizados por cada desarrollador así como su integración, es decir, que los programas sean más estándares y fáciles de leer, ya que cualquier programador está acostumbrado a tratar con código escrito de forma semejante. Además, proporcionan información sobre la funcionalidad (ver Anexo C).

CAPITULO VI: Desarrollo de la aplicación

Las palabras a usar serán en español, tendiendo como excepción aquellas de uso extendido por los programadores y que por lo general son en inglés. Ejemplos:

- canvas hace referencia a la interfaz gráfica de bajo nivel
- storage espacio donde se almacena la base de datos
- iterator interfaz para acceder y recorre elementos

De igual forma se respetarán los nombres de las métodos e interfaces necesarios para que el tipo de clase a usar funcione correctamente (que lo general son en inglés). Ejemplos:

- `public void readObject(InputStream entrada)` método usado por la base de datos Perst Lite para obtener los valores de un registro de una tabla definido como una clase
- `public void writeObject(OutputStream salida)` método usado por la base de datos Perst Lite para obtener los valores de un registro de una tabla definido como una clase
- `public void servicesDiscovered(int transID, ServiceRecord[] servRecord)` método usado por el Bluetooth para cuando se buscan los servicios de un dispositivo Bluetooth remoto.
- `public void serviceSearchCompleted(int transID, int respCode)` método usado por el Bluetooth para cuando la búsqueda de servicios se ha completado

En caso de que las palabras sean muy largas, existe la posibilidad de usar abreviaturas, de una extensión de 2 a 5 letras, pudiendo usar abreviaturas ya conocidas (incluyendo aquellas escritas en otro idioma). Ejemplos:

- bt abreviatura de Bluetooth
- bd abreviatura de Base de Datos
- m3g abreviatura del API de Gráficos 3D Móvil de Java Micro Edition
- coord abreviatura de coordenada
- cmd abreviatura de comando

Pruebas

En lo referente a las pruebas hechas del sistema, en base a la metodología XP se realizaron dos tipos pruebas: unitarias y de aceptación. Conforme a lo mencionado en capítulos previos sobre la metodología XP, las pruebas unitarias son las pruebas desarrolladas por los programadores para verificar el funcionamiento del código diseñado; por otro lado las pruebas de aceptación son diseñadas por el cliente para verificar la funcionalidad del sistema al final de cada interacción.

Para la implementación del sistema correspondiente al proyecto de tesis, se cuentan con 7 historias de usuario, las cuales a su vez, se relacionan con una tarea específica; cada tarea representa una iteración. Para el caso específico del sistema propuesto el diseño de las pruebas de aceptación está íntimamente relacionado con las historias de usuario, estas indican la funcionalidad que el cliente desea obtener del sistema.

Como se mencionó las pruebas unitarias se basan en el funcionamiento del código del sistema, se intenta automatizarlas. Con la finalidad de que en el caso de que exista una posterior modificación del código, esta modificación no afecte el resultado esperado por su correspondiente prueba unitaria y así evitar el rediseño, reestructuración y/o reimplementación de esta misma.

En general para la realización de las pruebas automáticas, los programadores que trabajan con la metodología XP se auxilian del uso de herramientas (tanto comerciales como gratuitas) específicamente diseñadas con este propósito. Existe un framework muy utilizado cuya denominación es xUnit del cual existen implementaciones para diferentes lenguajes de programación, una de las más conocidas JUnit que como su nombre lo indica esta diseñadas para pruebas automáticas de códigos en el lenguaje de programación Java ^(Berrueta).

A pesar de que existe J2MEUnit para la realización de pruebas automáticas de código en el lenguaje de programación J2ME, este framework presenta muchas inconvenientes en su uso, como los mismos autores lo mencionan los frameworks basados xUnit hacen uso generalmente de la reflexión

CAPITULO VI: Desarrollo de la aplicación

para el fácil desarrollo de las pruebas automáticas. Sin embargo a pesar de que J2MEUnit está basado en JUnit no cuenta con esta característica, por lo cual es necesario la implementación específica de cada clase de prueba, junto con todas las pruebas explícitas que conllevan la mismas, las cuales se relacionan directamente con la(s) funcionalidad(es) del código al cual se le desea aplicar la prueba. Existen otras alternativas como Mockito otro framework que persigue este mismo fin, sin embargo este proyecto se encuentra aún muy inmaduro a diferencias de J2MEUnit.

Debido a estos inconvenientes se optó por la implementación propia de las pruebas automáticas para el código en lenguaje J2ME de la aplicación para la terminal móvil del sistema correspondiente al proyecto de tesis. A pesar de que es posible realizar las pruebas automáticas del código en lenguaje Java de la aplicación para la terminal fija del sistema, se optó igualmente por la implementación propia de las pruebas automáticas. Esto se debe a que se buscaba una homogeneidad entre las pruebas automáticas, además de que se tomó en cuenta que el código de la terminal fija es pequeño dentro del sistema. Se toma en consideración que el sistema está constituido por las siguientes aplicaciones:

- Aplicación para la terminal fija que servirá como servidor para estimar el valor de RSSI del dispositivo móvil con respecto al módulo Bluetooth conectado a la terminal fija.
- Aplicación para el dispositivo móvil que se encargará de obtener la información necesaria para formar el radio mapa patrón del edificio usando como base los valores RSSI de uno o más servidores.
- Aplicación para la terminal fija que se encargará de generar el archivo del radio mapa patrón del edificio, mediante la información recolectada por la aplicación del dispositivo móvil diseñada para este propósito.
- Aplicación para la localización del dispositivo móvil dentro del edificio usando como base las señales recogidas (en un momento específico) de los servidores y que es contrastada con el radio mapa patrón (generado previamente).

Con la finalidad de que las pruebas unitarias sean de utilidad para futuras modificaciones del sistema, éstas se centraron solamente en la parte correspondiente a la aplicación para la localización del dispositivo móvil. Esto en base a lo mencionado previamente, además, se considera que es la parte medular del sistema (comparado con el resto de las aplicaciones que forman al sistema) y que en caso de modificaciones la estructura interna puede variar pero no su funcionalidad.

Para el diseño de las pruebas unitarias se debe considerar el uso de clases de tipo mocks, estas clases se usan cuando el funcionamiento de un método se ve comprometido por los datos obtenidos de otra clase mediante el uso de un objeto de la misma, es aquí cuando intervienen las clases tipo mocks. Las clases mocks están diseñadas de tal forma que emulan el funcionamiento de otra clase, con el fin de obtener siempre el dato correcto que debería entregar la clase que está emulando, su uso dentro de las pruebas unitarias consiste en poder aplicar las pruebas unitarias a los métodos de la clase correspondiente de forma aislada, es decir evita que el funcionamiento de la prueba unitaria se vea alterado por datos erróneos, los cuales se obtienen a partir de un objeto de otra clase.

Las pruebas sólo se aplican a los métodos relevantes de cada clase que conforman a la aplicación misma. A continuación se dará una breve explicación de las pruebas aplicadas a los métodos y las clases a la que pertenece cada método.

Pruebas unitarias de la aplicación para la localización del dispositivo móvil

La aplicación utilizada para la localización del dispositivo móvil está formada por 7 clases las cuales son:

- Bluetooth
- CamaraMovil
- CargarCanvas
- Midlet
- ModeloM3G
- PantallaInicio
- RadioMapa

Es importante mencionar que la clase Midlet es la clase principal y es una clase derivada de MIDlet (hereda los método y atributo de la clase MIDlet), por otro lado la clase Bluetooth deriva de la clase Thread, las clases CargarModelo y PantallaInicio derivan de la clase Canvas, mientras que la clase RadioMapa deriva de la clase Persitent.

RadioMapa es una clase que se usa para representar la tabla de consulta de la base de datos, esto se debe al hecho de Perts Lite es una base de datos orientada a objetos, por lo cual, para definir tablas se hace uso de clases y las consultas se hacen mediante objetos de las propias clases. Es por ello que no existen propiamente pruebas unitarias directas para esta la clase (Figura 97), aún así parte de su funcionalidad se prueba dentro de algunas de las otras clases que forman la aplicación, ya que se hace uso de consultas a esta clase mediante objetos de la misma.

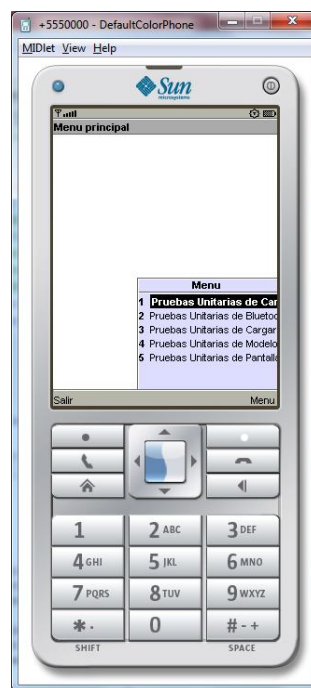


Figura 97 Pantalla de la aplicación con las pruebas unitarias correspondientes a la aplicación para la localización del dispositivo móvil.

Las pruebas unitarias aplicadas a la clase Blueotth son las siguientes (Figura 98):

1. Verificación del método "run" de la clase antes de establecer los parámetros mediante el método "setParametros". Si el método funciona correctamente ningún mensaje de error debe mostrarse en pantalla y el método "getEstado" de la clase debe devolver un valor booleano de "false".

CAPITULO VI: Desarrollo de la aplicación

2. Verificación del método “run” de la clase después de establecer los parámetros mediante el método “setParametros”. Si el método funciona correctamente ningún mensaje de error debe mostrarse en pantalla y el método “getEstado” de la clase debe devolver un valor booleano de “true”.
3. Verificación del método “metodoKNN” de la clase. Se establece una cadena de consulta que debe recibir el método, así como conocer de ante mano la cadena que debe devolver esta consulta. Se comparan las cadenas, si ambas cadenas coinciden y además el método “getEstado” de la clase devuelve un valor booleano de “true” entonces el método funciona correctamente.

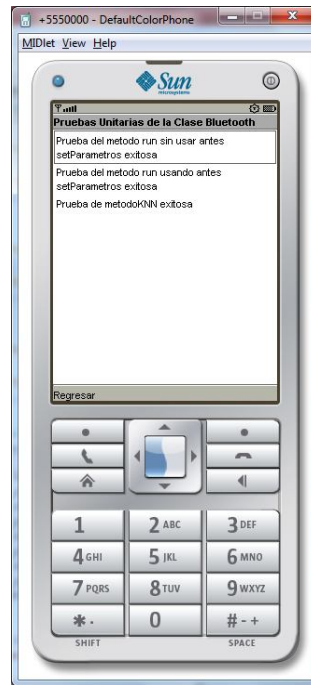


Figura 98 Pantalla de las pruebas unitarias correspondientes a la clase Bluetooth.

Las pruebas unitarias aplicadas a la clase CamaraMovil son las siguientes (Figura 99):

- Verificación del método “getCamara” de la clase. Si el método funciona correctamente ningún mensaje de error debe mostrarse en pantalla y el método debe devolver un valor diferente de “null”.
- Verificación del método “getGrupoCamara” de la clase. Si el método funciona correctamente ningún mensaje de error debe mostrarse en pantalla y el método debe devolver un valor diferente de “null”.
- Verificación del método “getPerspectivaCamara” de la clase. Se establecen los valores iniciales de la perspectiva de la cámara, se obtienen los valores de perspectiva de la cámara actuales y se comparan, si todos los valores coinciden entonces el método funciona correctamente.
- Verificación del método “getPosicion” de la clase. Se establecen los valores de las coordenadas iniciales de la cámara, se obtienen los valores de las coordenadas actuales de la cámara y se comparan, si todos los valores coinciden entonces el método funciona correctamente.
- Verificación del método “getRotacion” de la clase. Se establecen los valores de los ángulos iniciales de la cámara, se obtienen los valores de los ángulos actuales de la cámara y se comparan, si todos los valores coinciden entonces el método funciona correctamente.
- Verificación del método “movimientoTeclado” de la clase para desplazamiento-rotación de la cámara independiente de la posición especificada para el dispositivo móvil. Se establecen los valores de las coordenadas y los ángulos iniciales de la cámara independientes de la posición especificada para el dispositivo móvil, se obtienen los valores de las coordenadas

CAPITULO VI: Desarrollo de la aplicación

y los ángulos actuales de la cámara y se comparan, si todos los valores coinciden entonces el método funciona correctamente.

- Verificación del método “movimientoTeclado” de la clase para desplazamiento de la cámara a la posición especificada para el dispositivo móvil. Se establecen los valores de las coordenadas iniciales de la cámara a las coordenadas especificadas para el dispositivo móvil, se obtienen los valores de las coordenadas actuales de la cámara y se comparan, si todos los valores coinciden entonces el método funciona correctamente.

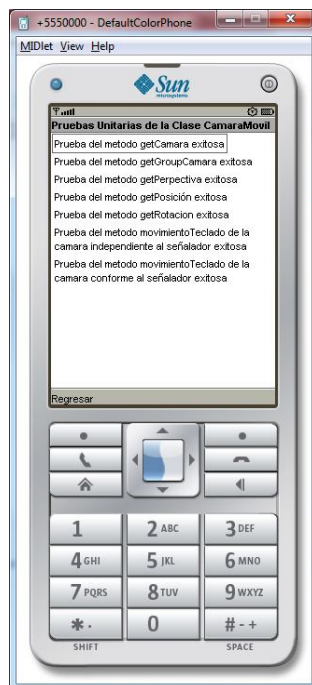


Figura 99 Pantalla de las pruebas unitarias correspondientes a la clase CamaraMovil.

Las pruebas unitarias aplicadas a la clase CargarCanvas son las siguientes (Figura 100):

- Verificación del método “constuirEscena” de la clase. Si el método funciona correctamente ningún mensaje de error debe mostrarse en pantalla y el método “getAlerta” de la clase debe devolver un valor booleano de “false”.
- Verificación del método “getModeloCoordXYZ” de la clase. Se establecen los valores de las coordenadas iniciales de la figura geométrica 3D que se usa como indicador visual de la posición del dispositivo móvil con respecto al modelo 3D del edificio, se obtienen los valores de las coordenadas de la figura geométrica 3D usada como indicador y se comparan, si todos los valores coinciden entonces el método funciona correctamente.
- Verificación del método “getPosicionModelo” de la clase. Se establece una cadena con los valores de las coordenadas iniciales de la figura geométrica 3D que se usa como indicador visual de la posición del dispositivo móvil con respecto al modelo 3D del edificio, se obtienen una cadena con los valores de las coordenadas de la figura geométrica 3D usada como indicador y se comparan, si todos los valores coinciden entonces el método funciona correctamente.
- Verificación del método “actualizaModeloCoordXYZ” de la clase. Se establece una cadena con los valores de las coordenadas nuevas de la figura geométrica 3D que se usa como indicador visual de la posición del dispositivo móvil con respecto al modelo 3D del edificio, se aplican las nuevas coordenadas a la figura geométrica 3D usada como indicador, se obtienen una cadena con los valores de las coordenadas de la misma y se comparan, si todos los valores coinciden entonces el método funciona correctamente.

CAPITULO VI: Desarrollo de la aplicación

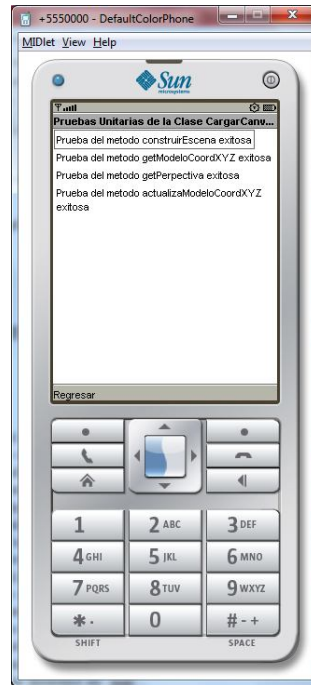


Figura 100 Pantalla de las pruebas unitarias correspondientes a la clase CargarCanvas.

Las pruebas unitarias aplicadas a la clase ModeloM3G son las siguientes (Figura 101):

- Verificación del método "inicio" de la clase. Si el método funciona correctamente ningún mensaje de error debe mostrarse en pantalla, el método "getInicioMidlet" de la clase debe devolver un valor booleano de "true" y el método "getAlerta" de la clase debe devolver un valor booleano de "false".
- Verificación del método "getMAC" de la clase. Si el método funciona correctamente ningún mensaje de error debe mostrarse en pantalla y el método debe devolver un Vector cuyo tamaño debe coincidir con el número de los módulos Bluetooth usados en las terminales fijas.
- Verificación del método "getRSSI" de la clase. Si el método funciona correctamente ningún mensaje de error debe mostrarse en pantalla y el método debe devolver un Vector cuyo tamaño debe coincidir con el número de los módulos Bluetooth usados en las terminales fijas.
- Verificación del método "getBaseDatos" de la clase. Si el método funciona correctamente ningún mensaje de error debe mostrarse en pantalla y el método debe devolver un valor diferente de "null".
- Verificación del método "getIterador" de la clase. Si el método funciona correctamente ningún mensaje de error debe mostrarse en pantalla y el método debe devolver un valor diferente de "null".
- Verificación del método "getDeposito" de la clase. Si el método funciona correctamente ningún mensaje de error debe mostrarse en pantalla y el método debe devolver un valor diferente de "null".

CAPITULO VI: Desarrollo de la aplicación

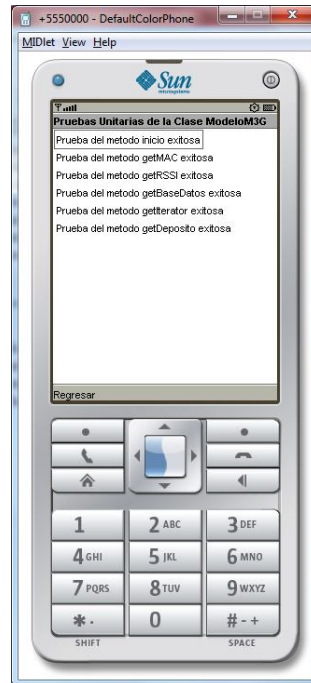


Figura 101 Pantalla de las pruebas unitarias correspondientes a la clase ModeloM3G.

Las pruebas unitarias aplicadas a la clase PantallaInicio es la siguiente (Figura 102):

- Verificación del método "cargarImagen" de la clase. Si el método funciona correctamente ningún mensaje de error debe mostrarse en pantalla, y el método "getAlerta" de la clase debe devolver un valor booleano de "false".

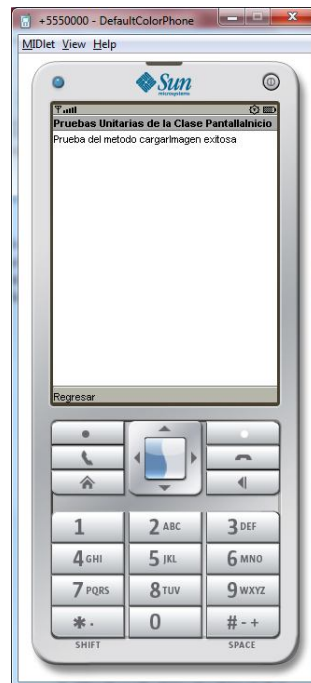


Figura 102 Pantalla de la prueba unitaria correspondiente a la clase PantallaInicio.

CAPITULO VI: Desarrollo de la aplicación

Pruebas de Aceptación de la Tarea 1

Ya que la Tarea 1 se relaciona con la historia de usuario 1 se debe observar lo que persigue, la obtención del RSSI de un dispositivo móvil a través de la búsqueda y conexión con los módulos Bluetooth conectados a las terminales fijas. Los resultados pueden ser definidos como terminales si dan por concluida la historia de usuario y no terminales si se sigue en la misma, la clasificación de los resultados se muestra en la siguiente tabla:

Tabla 10 Clasificación de resultados observables de la Tarea 1.

Resultados observables	Clasificación
Obtención del valor RSSI de cada una de los módulos Bluetooth conectados a las terminales fijas.	Terminal
Obtención del valor RSSI de sólo algunos de los módulos Bluetooth conectados a las terminales fijas.	Terminal
No se puede obtener el valor RSSI de ninguno de los módulos Bluetooth conectados a las terminales fijas.	Terminal

El hecho de que todos los resultados obtenidos sean terminales es requisito del cliente para el funcionamiento del sistema, este tipo información permite terminar con la tarea ya que puede usarse para indicar si en un momento determinado ninguno de los módulos Bluetooth está funcionando, el dispositivo móvil se encuentra fuera del rango de cobertura de alguno o todos los módulos Bluetooth y/o si el dispositivo móvil tiene o no soporte para del API necesario.

Ahora es necesario identificar todos los posibles caminos que llevan a los resultados observables vistos hasta el momento, estos caminos se muestran en la siguiente tabla:

Tabla 11 Caminos posibles de ejecución de la Tarea 1.

Caminos de ejecución	Resultados observables
Selección de los módulos Bluetooth mediante sus respectivas MACs, búsqueda y establecimiento de comunicación con todos ellos.	Obtención del valor RSSI de cada una de los módulos Bluetooth conectados a las terminales fijas.
Selección de los módulos Bluetooth mediante sus respectivas MACs, búsqueda y establecimiento de comunicación con todos ellos, con pérdida posterior de la comunicación de algunos de los módulos.	Obtención del valor RSSI de sólo algunos de los módulos Bluetooth conectados a las terminales fijas.
Selección de los módulos Bluetooth mediante sus respectivas MACs, búsqueda y establecimiento de comunicación con todos ellos, con pérdida posterior de la comunicación de todos los módulos.	No se puede obtener el valor RSSI de ninguno de los módulos Bluetooth conectados a las terminales fijas.
Selección de los módulos Bluetooth mediante sus respectivas MACs, búsqueda y establecimiento de comunicación con algunos de ellos.	Obtención del valor RSSI de sólo algunos de los módulos Bluetooth conectados a las terminales fijas.
Selección de los módulos Bluetooth mediante sus respectivas MACs, búsqueda y establecimiento de comunicación con algunos de ellos, con pérdida posterior de la comunicación de algunos de los módulos.	Obtención del valor RSSI de sólo algunos de los módulos Bluetooth conectados a las terminales fijas.
Selección de los módulos Bluetooth mediante sus respectivas MACs, búsqueda y establecimiento de comunicación con algunos de ellos, con pérdida posterior de la comunicación de todos los módulos.	No se puede obtener el valor RSSI de ninguno de los módulos Bluetooth conectados a las terminales fijas.
Selección de los módulos Bluetooth mediante sus respectivas MACs, búsqueda e imposibilidad del establecimiento de comunicación con todos ellos.	No se puede obtener el valor RSSI de ninguno de los módulos Bluetooth conectados a las terminales fijas.

Debido a que existen diversos caminos de ejecución es necesario definir el conjunto de valores al cual corresponde cada uno, contando con dos posibles resultados: correcto e incorrecto. Sin embargo como ya se mencionó previamente, para este caso todos los resultados son de utilidad, por lo tanto, todos queda clasificados como correctos, tal y como se muestra en la siguiente tabla:

CAPITULO VI: Desarrollo de la aplicación

Tabla 12 Valores posibles para los posibles caminos de ejecución de la Tarea 1.

Caminos de ejecución	Conjunto de valores
Selección de los módulos Bluetooth mediante sus respectivas MACs, búsqueda y establecimiento de comunicación con todos ellos.	Correcto
Selección de los módulos Bluetooth mediante sus respectivas MACs, búsqueda y establecimiento de comunicación con todos ellos, con pérdida posterior de la comunicación de algunos de los módulos.	Correcto
Selección de los módulos Bluetooth mediante sus respectivas MACs, búsqueda y establecimiento de comunicación con todos ellos, con pérdida posterior de la comunicación de todos los módulos.	Correcto
Selección de los módulos Bluetooth mediante sus respectivas MACs, búsqueda y establecimiento de comunicación con algunos de ellos.	Correcto
Selección de los módulos Bluetooth mediante sus respectivas MACs, búsqueda y establecimiento de comunicación con algunos de ellos, con pérdida posterior de la comunicación de algunos de los módulos.	Correcto
Selección de los módulos Bluetooth mediante sus respectivas MACs, búsqueda y establecimiento de comunicación con algunos de ellos, con pérdida posterior de la comunicación de todos los módulos.	Correcto
Selección de los módulos Bluetooth mediante sus respectivas MACs, búsqueda e imposibilidad del establecimiento de comunicación con todos ellos.	Correcto

En base la información recolectada hasta el momento se puede concluir que no existe redundancia en los caminos, por lo tanto, la tabla de caminos posibles con que se cuenta hasta el momento es definitiva.

Para terminar con las pruebas de aceptación se verifica que el sistema cumpla con los posibles caminos establecidos junto con sus respectivos conjuntos de valores. Una vez finalizado el desarrollo del código correspondiente a la Tarea 1 se observa que todos estos requisitos son válidos dentro del sistema.

Pruebas de Aceptación de la Tarea 2

Ya que la Tarea 2 se relaciona con la historia de usuario 2 se debe observar lo que persigue, la carga y visualización del modelado 3D en el dispositivo móvil, además de proveer una presentación gráfica inicial del funcionamiento de la aplicación. La clasificación de los resultados se muestra en la siguiente tabla:

Tabla 13 Clasificación de resultados observables de la Tarea 2.

Resultados observables	Clasificación
Se ejecuta la presentación al mismo tiempo que se carga el modelo en memoria y una vez terminada la presentación se visualiza el modelo en la pantalla del dispositivo móvil.	Terminal
Se ejecuta la presentación, pero se detiene ya que no se puede cargar el modelo en memoria.	Terminal
Se ejecuta la presentación, pero se detiene ya que el usuario cierra la tapa del dispositivo móvil, sin embargo se carga el modelo en memoria y posteriormente se visualiza el modelo en la pantalla del dispositivo móvil.	Terminal
Se ejecuta la presentación, pero se detiene ya que el usuario cierra la tapa del dispositivo móvil, además de que el no posible cargar el modelo en memoria.	Terminal
No se ejecuta la presentación, sin embargo se carga el modelo en memoria y se visualiza el mismo en la pantalla del dispositivo móvil.	Terminal
No se ejecuta la presentación, además de que no se puede cargar el modelo en memoria.	Terminal

Al igual que en lo sucedido en la pruebas de aceptación previas, todos los resultados obtenidos son terminales ya que permiten obtener al final un resultado que indica si es posible o no realizar la tarea. Posteriormente analizando los resultados se llegó a la conclusión que los últimos 4, eran posibilidades no deseables dentro de la aplicación. Se realizaron algunos cambios en el código de la tarea 2 de tal forma que no importa si el usuario cierra o no la tapa del dispositivo móvil, esto no debe afectar el funcionamiento de la aplicación. Además la información necesaria para la presentación (para este caso específico se trata de un conjunto de imágenes) ha quedado incluida dentro del instalador de la aplicación para el dispositivo móvil. Con estos cambios, la tabla de resultados observables queda de la siguiente forma:

CAPITULO VI: Desarrollo de la aplicación

Tabla 14 Nueva Clasificación de resultados observables de la Tarea 2.

Resultados observables	Clasificación
Se ejecuta la presentación al mismo tiempo que se carga el modelo en memoria y una vez terminada la presentación se visualiza el modelo en la pantalla del dispositivo móvil.	Terminal
Se ejecuta la presentación, pero está se detiene ya que no se puede cargar el modelo en memoria.	Terminal

Ahora es necesario identificar todos los posibles caminos que llevan a los resultados observables vistos hasta el momento, estos caminos se muestran en la siguiente tabla:

Tabla 15 Caminos posibles de ejecución de la Tarea 2.

Caminos de ejecución	Resultados observables
El archivo con el modelo se encuentra en la memoria externa, es accesible ya que tiene el nombre y extensión adecuados, se tiene suficiente memoria temporal y soporte para el API necesario.	Se ejecuta la presentación al mismo tiempo que se carga el modelo en memoria y una vez terminada la presentación se visualiza el modelo en la pantalla del dispositivo móvil.
El archivo con el modelo se encuentra en la memoria externa, es accesible ya que tiene el nombre y extensión adecuados, pero no se tiene suficiente memoria temporal.	Se ejecuta la presentación, pero se detiene ya que no se puede cargar el modelo en memoria.
El archivo con el modelo se encuentra en la memoria externa, es accesible ya que tiene el nombre y extensión adecuados, se tiene suficiente memoria temporal pero no soporte para el API necesario	Se ejecuta la presentación, pero se detiene ya que no se puede cargar el modelo en memoria.
El archivo con el modelo se encuentra en la memoria externa, pero no es accesible ya que no tiene el nombre adecuado.	Se ejecuta la presentación, pero se detiene ya que no se puede cargar el modelo en memoria.
El archivo con el modelo se encuentra en la memoria externa, pero no es accesible ya que no tiene la extensión adecuada.	Se ejecuta la presentación, pero se detiene ya que no se puede cargar el modelo en memoria.
El archivo con el modelo se encuentra en la memoria externa, pero no es accesible ya que no tiene el nombre ni la extensión adecuada.	Se ejecuta la presentación, pero se detiene ya que no se puede cargar el modelo en memoria.
El archivo con el modelo no se encuentra en la memoria externa.	Se ejecuta la presentación, pero se detiene ya que no se puede cargar el modelo en memoria.

De igual forma que ocurrió en las pruebas anteriores, todos los resultados son de utilidad, por lo tanto, todos quedan clasificados como correctos, tal y como se muestra en la siguiente tabla:

Tabla 16 Valores posibles para los posibles caminos de ejecución de la Tarea 2.

Caminos de ejecución	Conjunto de valores
El archivo con el modelo se encuentra en la memoria externa, es accesible ya que tiene el nombre y extensión adecuados, se tiene suficiente memoria temporal y soporte para el API necesario.	Correcto
El archivo con el modelo se encuentra en la memoria externa, es accesible ya que tiene el nombre y extensión adecuados, pero no se tiene suficiente memoria temporal.	Correcto
El archivo con el modelo se encuentra en la memoria externa, es accesible ya que tiene el nombre y extensión adecuados, se tiene suficiente memoria temporal pero no soporte para el API necesario	Correcto
El archivo con el modelo se encuentra en la memoria externa, pero no es accesible ya que no tiene el nombre adecuado.	Correcto
El archivo con el modelo se encuentra en la memoria externa, pero no es accesible ya que no tiene la extensión adecuada.	Correcto
El archivo con el modelo se encuentra en la memoria externa, pero no es accesible ya que no tiene el nombre ni la extensión adecuada.	Correcto
El archivo con el modelo no se encuentra en la memoria externa.	Correcto

En base la información recolectada hasta el momento se puede observar que existe redundancia en los caminos, es por esto que algunos caminos son redefinidos en un único camino.

Las tablas de caminos posibles y valores posibles para los mismos quedan de la siguiente forma:

CAPITULO VI: Desarrollo de la aplicación

Tabla 17 Nuevos caminos posibles de ejecución de la Tarea 2.

Caminos de ejecución	Resultados observables
El archivo con el modelo se encuentra en la memoria externa, es accesible ya que tiene el nombre y extensión adecuados, se tiene suficiente memoria temporal y soporte para el API necesario.	Se ejecuta la presentación al mismo tiempo que se carga el modelo en memoria y una vez terminada la presentación se visualiza el modelo en la pantalla del dispositivo móvil.
El archivo con el modelo se encuentra en la memoria externa, es accesible ya que tiene el nombre y extensión adecuados, pero no se tiene suficiente memoria temporal.	Se ejecuta la presentación, pero se detiene ya que no se puede cargar el modelo en memoria.
El archivo con el modelo se encuentra en la memoria externa, es accesible ya que tiene el nombre y extensión adecuados, se tiene suficiente memoria temporal pero no soporte para el API necesario	Se ejecuta la presentación, pero se detiene ya que no se puede cargar el modelo en memoria.
El archivo con el modelo se encuentra en la memoria externa, pero no es accesible ya que no tiene el nombre y/o la extensión adecuada.	Se ejecuta la presentación, pero se detiene ya que no se puede cargar el modelo en memoria.
El archivo con el modelo no se encuentra en la memoria externa.	Se ejecuta la presentación, pero se detiene ya que no se puede cargar el modelo en memoria.

Tabla 18 Nuevos valores posibles para los posibles caminos de ejecución de la Tarea 2.

Caminos de ejecución	Conjunto de valores
El archivo con el modelo se encuentra en la memoria externa, es accesible ya que tiene el nombre y extensión adecuados, se tiene suficiente memoria temporal y soporte para el API necesario.	Correcto
El archivo con el modelo se encuentra en la memoria externa, es accesible ya que tiene el nombre y extensión adecuados, pero no se tiene suficiente memoria temporal.	Correcto
El archivo con el modelo se encuentra en la memoria externa, es accesible ya que tiene el nombre y extensión adecuados, se tiene suficiente memoria temporal pero no soporte para el API necesario	Correcto
El archivo con el modelo se encuentra en la memoria externa, pero no es accesible ya que no tiene el nombre y/o la extensión adecuada.	Correcto
El archivo con el modelo no se encuentra en la memoria externa.	Correcto

Para terminar con las pruebas de aceptación se verifica que el sistema cumpla con los posibles caminos establecidos junto con sus respectivos conjuntos de valores. Una vez finalizado el desarrollo del código correspondiente a la Tarea 2 se observa que todos estos requisitos son válidos dentro del sistema.

Pruebas de Aceptación de la Tarea 3

Ya que la Tarea 3 se relaciona con la historia de usuario 3, se debe observar lo que persigue, es la combinación de las tareas 1 y 2, pero además se incorpora el uso de un radio mapa patrón para el cálculo de la posible localización del dispositivo móvil mediante el conjunto de RSSIs obtenidos. La clasificación de los resultados se muestra en la siguiente tabla:

Tabla 19 Clasificación de resultados observables de la Tarea 4.

Resultados observables	Clasificación
Se ejecuta la presentación al mismo tiempo que se carga el modelo en memoria y una vez terminada la presentación se visualiza el modelo en la pantalla del dispositivo móvil.	No terminal
Se ejecuta la presentación, pero se detiene ya que no se puede cargar el modelo en memoria.	Terminal
Una vez cargado el modelo se obtiene el valor RSSI de cada una de los módulos Bluetooth conectados a las terminales fijas.	Terminal
Una vez cargado el modelo se obtiene el valor RSSI de sólo algunos de los módulos Bluetooth conectados a las terminales fijas.	Terminal
Una vez cargado el modelo no se puede obtener el valor RSSI de ninguno de los módulos Bluetooth conectados a las terminales fijas.	Terminal

CAPITULO VI: Desarrollo de la aplicación

A diferencia de lo sucedido en las pruebas de aceptación para la Tarea 1 y 2, no todos los resultados obtenidos son terminales, conforme a los requisitos del cliente, la terminación de la tarea puede ser debido a un error o si se trata de obtener los RSSI de los módulos Bluetooth seleccionados, es en este caso específico que uno de los resultados observables no cumple con estas condiciones por lo cual queda clasificado como no terminal.

Ahora es necesario identificar todos los posibles caminos que llevan a los resultados observables vistos hasta el momento, estos caminos se muestran en la siguiente tabla:

Tabla 20 Valores posibles para los posibles caminos de ejecución de la Tarea 3.

Caminos de ejecución	Resultados observables
El archivo con el modelo se encuentra en la memoria externa, es accesible ya que tiene el nombre y extensión adecuados, se tiene suficiente memoria temporal y soporte para el API necesario.	Se ejecuta la presentación al mismo tiempo que se carga el modelo en memoria y una vez terminada la presentación se visualiza el modelo en la pantalla del dispositivo móvil.
El archivo con el modelo se encuentra en la memoria externa, es accesible ya que tiene el nombre y extensión adecuados, pero no se tiene suficiente memoria temporal.	Se ejecuta la presentación, pero se detiene ya que no se puede cargar el modelo en memoria.
El archivo con el modelo se encuentra en la memoria externa, es accesible ya que tiene el nombre y extensión adecuados, se tiene suficiente memoria temporal pero no soporte para el API necesario	Se ejecuta la presentación, pero se detiene ya que no se puede cargar el modelo en memoria.
El archivo con el modelo se encuentra en la memoria externa, pero no es accesible ya que no tiene el nombre y/o la extensión adecuada.	Se ejecuta la presentación, pero se detiene ya que no se puede cargar el modelo en memoria.
El archivo con el modelo no se encuentra en la memoria externa.	Se ejecuta la presentación, pero se detiene ya que no se puede cargar el modelo en memoria.
Selección de los módulos Bluetooth mediante sus respectivas MACs, búsqueda y establecimiento de comunicación con todos ellos.	Una vez cargado el modelo se obtiene el valor RSSI de cada una de los módulos Bluetooth conectados a las terminales fijas.
Selección de los módulos Bluetooth mediante sus respectivas MACs, búsqueda y establecimiento de comunicación con todos ellos, con pérdida posterior de la comunicación de algunos de los módulos.	Una vez cargado el modelo se obtiene el valor RSSI de sólo algunos de los módulos Bluetooth conectados a las terminales fijas.
Selección de los módulos Bluetooth mediante sus respectivas MACs, búsqueda y establecimiento de comunicación con todos ellos, con pérdida posterior de la comunicación de todos los módulos.	Una vez cargado el modelo no se puede obtener el valor RSSI de ninguno de los módulos Bluetooth conectados a las terminales fijas.
Selección de los módulos Bluetooth mediante sus respectivas MACs, búsqueda y establecimiento de comunicación con algunos de ellos.	Una vez cargado el modelo se obtiene el valor RSSI de sólo algunos de los módulos Bluetooth conectados a las terminales fijas.
Selección de los módulos Bluetooth mediante sus respectivas MACs, búsqueda y establecimiento de comunicación con algunos de ellos, con pérdida posterior de la comunicación de algunos de los módulos.	Una vez cargado el modelo se obtiene el valor RSSI de sólo algunos de los módulos Bluetooth conectados a las terminales fijas.
Selección de los módulos Bluetooth mediante sus respectivas MACs, búsqueda y establecimiento de comunicación con algunos de ellos, con pérdida posterior de la comunicación de todos los módulos.	Una vez cargado el modelo no se puede obtener el valor RSSI de ninguno de los módulos Bluetooth conectados a las terminales fijas.
Selección de los módulos Bluetooth mediante sus respectivas MACs, búsqueda e imposibilidad del establecimiento de comunicación con todos ellos.	Una vez cargado el modelo no se puede obtener el valor RSSI de ninguno de los módulos Bluetooth conectados a las terminales fijas.

De igual que en las pruebas anteriores todos los resultados quedan clasificados como correctos, tal y como se muestra en la siguiente tabla:

CAPITULO VI: Desarrollo de la aplicación

Tabla 21 Valores posibles para los posibles caminos de ejecución de la Tarea 3.

Caminos de ejecución	Conjunto de valores
El archivo con el modelo se encuentra en la memoria externa, es accesible ya que tiene el nombre y extensión adecuados, se tiene suficiente memoria temporal y soporte para el API necesario.	Correcto
El archivo con el modelo se encuentra en la memoria externa, es accesible ya que tiene el nombre y extensión adecuados, pero no se tiene suficiente memoria temporal.	Correcto
El archivo con el modelo se encuentra en la memoria externa, es accesible ya que tiene el nombre y extensión adecuados, se tiene suficiente memoria temporal pero no soporte para el API necesario	Correcto
El archivo con el modelo se encuentra en la memoria externa, pero no es accesible ya que no tiene el nombre y/o la extensión adecuada.	Correcto
El archivo con el modelo no se encuentra en la memoria externa.	Correcto
Selección de los módulos Bluetooth mediante sus respectivas MACs, búsqueda y establecimiento de comunicación con todos ellos.	Correcto
Selección de los módulos Bluetooth mediante sus respectivas MACs, búsqueda y establecimiento de comunicación con todos ellos, con pérdida posterior de la comunicación de algunos de los módulos.	Correcto
Selección de los módulos Bluetooth mediante sus respectivas MACs, búsqueda y establecimiento de comunicación con todos ellos, con pérdida posterior de la comunicación de todos los módulos.	Correcto
Selección de los módulos Bluetooth mediante sus respectivas MACs, búsqueda y establecimiento de comunicación con algunos de ellos.	Correcto
Selección de los módulos Bluetooth mediante sus respectivas MACs, búsqueda y establecimiento de comunicación con algunos de ellos, con pérdida posterior de la comunicación de algunos de los módulos.	Correcto
Selección de los módulos Bluetooth mediante sus respectivas MACs, búsqueda y establecimiento de comunicación con algunos de ellos, con pérdida posterior de la comunicación de todos los módulos.	Correcto
Selección de los módulos Bluetooth mediante sus respectivas MACs, búsqueda e imposibilidad del establecimiento de comunicación con todos ellos.	Correcto

Al observarse que no existe redundancia en los caminos, la tabla de caminos posibles con que se cuenta hasta el momento es definitiva.

Para terminar con las pruebas de aceptación se verifica que el sistema cumpla con los posibles caminos establecidos junto con sus respectivos conjuntos de valores. Una vez finalizado el desarrollo del código correspondiente a la Tarea 3 se observa que todos estos requisitos son válidos dentro del sistema.

Pruebas de Aceptación de la Tarea 4

La Tarea 4 se relaciona con la historia de usuario 4, la obtención del RSSI de un dispositivo móvil a través de las terminales fijas se realiza tal y como se hizo en la Tarea 1. Sin embargo por requerimientos del cliente también es necesario que en caso de no obtener el RSSI, éste se complete con un valor de referencia. La clasificación de los resultados se muestra en la siguiente tabla:

Tabla 22 Clasificación de resultados observables de la Tarea 4.

Resultados observables	Clasificación
Obtención del valor RSSI de cada una de los módulos Bluetooth conectados a las terminales fijas.	Terminal
Obtención del valor RSSI de sólo algunos de los módulos Bluetooth conectados a las terminales fijas, para el resto se usó un valor de referencia.	Terminal
No se puede obtener el valor RSSI de ninguno de los módulos Bluetooth conectados a las terminales fijas, por lo cual se hace uso de un valor de referencia para todos.	Terminal

Al igual que en la pruebas de aceptación para la Tarea 1, todos los resultados obtenidos son terminales debido a las circunstancias ya mencionadas. Además, la propia historia de usuario indica que la falta de obtención del RSSI se complementará con el uso de un valor de referencia, el cual se usará como parte de los datos necesarios para la localización del dispositivo móvil. Con este cambio la tabla de resultados observables queda de la siguiente forma:

CAPITULO VI: Desarrollo de la aplicación

Tabla 23 Clasificación de resultados observables de la Tarea 4.

Caminos de ejecución	Resultados observables
Selección de los módulos Bluetooth mediante sus respectivas MACs y establecimiento de comunicación con todos ellos.	Obtención del valor RSSI de cada uno de los módulos Bluetooth conectados a las terminales fijas.
Selección de los módulos Bluetooth mediante sus respectivas MACs y establecimiento de comunicación con todos ellos, con pérdida posterior de la comunicación de algunos de los módulos.	Obtención del valor RSSI de sólo algunos de los módulos Bluetooth conectados a las terminales fijas, para el resto se usó un valor de referencia.
Selección de los módulos Bluetooth mediante sus respectivas MACs y establecimiento de comunicación con todos ellos, con pérdida posterior de la comunicación de todos los módulos.	No se puede obtener el valor RSSI de ninguno de los módulos Bluetooth conectados a las terminales fijas, se hace uso de un valor de referencia para todos.
Selección de los módulos Bluetooth mediante sus respectivas MACs y establecimiento de comunicación con algunos de ellos.	Obtención del valor RSSI de sólo algunos de los módulos Bluetooth conectados a las terminales fijas, para el resto se usó un valor de referencia.
Selección de los módulos Bluetooth mediante sus respectivas MACs y establecimiento de comunicación con algunos de ellos, con pérdida posterior de la comunicación de algunos de los módulos.	Obtención del valor RSSI de sólo algunos de los módulos Bluetooth conectados a las terminales fijas, para el resto se usó un valor de referencia.
Selección de los módulos Bluetooth mediante sus respectivas MACs y establecimiento de comunicación con algunos de ellos, con pérdida posterior de la comunicación de todos los módulos.	No se puede obtener el valor RSSI de ninguno de los módulos Bluetooth conectados a las terminales fijas, por lo cual se hace uso de un valor de referencia para todos..
Selección de los módulos Bluetooth mediante sus respectivas MACs e imposibilidad del establecimiento de comunicación con todos ellos.	No se puede obtener el valor RSSI de ninguno de los módulos Bluetooth conectados a las terminales fijas, por lo cual se hace uso de un valor de referencia para todos.

Todos lo resultados de estas pruebas son de utilidad por lo cual todos quedan clasificados como correctos, tal y como se muestra en la siguiente tabla:

Tabla 24 Valores posibles para los posibles caminos de ejecución de la Tarea 4.

Caminos de ejecución	Conjunto de valores
Selección de los módulos Bluetooth mediante sus respectivas MACs y establecimiento de comunicación con todos ellos.	Correcto
Selección de los módulos Bluetooth mediante sus respectivas MACs y establecimiento de comunicación con todos ellos, con pérdida posterior de la comunicación de algunos de los módulos.	Correcto
Selección de los módulos Bluetooth mediante sus respectivas MACs y establecimiento de comunicación con todos ellos, con pérdida posterior de la comunicación de todos los módulos.	Correcto
Selección de los módulos Bluetooth mediante sus respectivas MACs y establecimiento de comunicación con algunos de ellos.	Correcto
Selección de los módulos Bluetooth mediante sus respectivas MACs y establecimiento de comunicación con algunos de ellos, con pérdida posterior de la comunicación de algunos de los módulos.	Correcto
Selección de los módulos Bluetooth mediante sus respectivas MACs y establecimiento de comunicación con algunos de ellos, con pérdida posterior de la comunicación de todos los módulos.	Correcto
Selección de los módulos Bluetooth mediante sus respectivas MACs e imposibilidad del establecimiento de comunicación con todos ellos.	Correcto

Al no existir redundancia en los caminos, la tabla de caminos posibles con que se cuenta hasta el momento es definitiva.

Para terminar con las pruebas de aceptación se verifica que el sistema cumpla con los posibles caminos establecidos junto con sus respectivos conjuntos de valores. Una vez finalizado el desarrollo del código correspondiente a la Tarea 4 se observa que todos estos requisitos son válidos dentro del sistema.

Pruebas de Aceptación de la Tarea 5

La Tarea 5 se relaciona con la historia de usuario 5, entonces, al igual que en la Tarea 2, se realiza la carga y visualización del modelado 3D en el dispositivo móvil junto con una presentación grafica inicial del funcionamiento de la aplicación, pero además se provee la capacidad para que el usuario

CAPITULO VI: Desarrollo de la aplicación

navegue dentro del modelo mediante las teclas de dirección del dispositivo. A partir de la historia de usuario 5 y las pruebas de aceptación de la Tarea 2 se puede observar que existen 2 posibles resultados. La clasificación de los resultados se muestra en la siguiente tabla:

Tabla 25 Clasificación de resultados observables de la Tarea 5.

Resultados observables	Clasificación
Se ejecuta la presentación al mismo tiempo que se carga el modelo en memoria, una vez terminada la presentación se visualiza el modelo en la pantalla del dispositivo móvil y se permite la navegación dentro del modelo mediante las teclas de dirección.	Terminal
Se ejecuta la presentación, pero se detiene debido a que no se puede cargar el modelo en memoria.	Terminal

Todos los resultados obtenidos en las pruebas de aceptación son terminales debido a las circunstancias ya comentadas, adicionando sólo el hecho de que ahora el usuario cuenta con la posibilidad de navegar dentro del modelo. Ya con este cambio la tabla de resultados observables queda de la siguiente forma:

Tabla 26 Caminos posibles de ejecución de la Tarea 5.

Caminos de ejecución	Resultados observables
El archivo con el modelo se encuentra en la memoria externa, es accesible ya que tiene el nombre y extensión adecuados, se tiene suficiente memoria temporal y soporte para el API necesario.	Se ejecuta la presentación al mismo tiempo que se carga el modelo en memoria, una vez terminada la presentación se visualiza el modelo en la pantalla del dispositivo móvil y se permite la navegación dentro del modelo mediante las teclas de dirección.
El archivo con el modelo se encuentra en la memoria externa, es accesible ya que tiene el nombre y extensión adecuados, pero no se tiene suficiente memoria temporal.	Se ejecuta la presentación, pero se detiene ya que no se puede cargar el modelo en memoria.
El archivo con el modelo se encuentra en la memoria externa, es accesible ya que tiene el nombre y extensión adecuados, se tiene suficiente memoria temporal pero no soporte para el API necesario	Se ejecuta la presentación, pero se detiene ya que no se puede cargar el modelo en memoria.
El archivo con el modelo se encuentra en la memoria externa, pero no es accesible ya que no tiene el nombre y/o la extensión adecuada.	Se ejecuta la presentación, pero se detiene ya que no se puede cargar el modelo en memoria.
El archivo con el modelo no se encuentra en la memoria externa.	Se ejecuta la presentación, pero se detiene ya que no se puede cargar el modelo en memoria.

De igual forma que lo sucedido en las pruebas anteriores de la Tarea 2 todos los resultados quedan clasificados como correctos, tal y como se muestra en la siguiente tabla:

Tabla 27 Valores posibles para los posibles caminos de ejecución de la Tarea 5.

Caminos de ejecución	Conjunto de valores
El archivo con el modelo se encuentra en la memoria externa, es accesible ya que tiene el nombre y extensión adecuados, se tiene suficiente memoria temporal y soporte para el API necesario.	Correcto
El archivo con el modelo se encuentra en la memoria externa, es accesible ya que tiene el nombre y extensión adecuados, pero no se tiene suficiente memoria temporal.	Correcto
El archivo con el modelo se encuentra en la memoria externa, es accesible ya que tiene el nombre y extensión adecuados, se tiene suficiente memoria temporal pero no soporte para el API necesario	Correcto
El archivo con el modelo se encuentra en la memoria externa, pero no es accesible ya que no tiene el nombre y/o la extensión adecuada.	Correcto
El archivo con el modelo no se encuentra en la memoria externa.	Correcto

Al eliminarse la redundancia desde las pruebas unitarias para la Tarea 2, la tabla de caminos posibles con que se cuenta hasta el momento es definitiva.

Para terminar con las pruebas de aceptación se verifica que el sistema cumpla con los posibles caminos establecidos junto con sus respectivos conjuntos de valores. Una vez finalizado el desarrollo del código correspondiente a la Tarea 4 se observa que todos estos requisitos son válidos dentro del sistema.

CAPITULO VI: Desarrollo de la aplicación

Pruebas de Aceptación de la Tarea 6

La Tarea 6 se relaciona con la historia de usuario 6, se debe observar lo que persigue, tratándose al igual que en la Tarea 5 pero, además como funcionalidad adicional se incorpora una figura geométrica 3D la cual sirve como indicador visual de la posición del dispositivo móvil dentro del modelo junto con un rotulo con las coordenadas locales X,Y,Z en la parte superior en la pantalla del propio dispositivo móvil. A partir de la historia de usuario 6 y las pruebas de aceptación de la Tarea 5 se puede observar que existen 2 posibles resultados. La clasificación de los resultados se muestra en la siguiente tabla:

Tabla 28 Clasificación de resultados observables de la Tarea 6.

Resultados observables	Clasificación
Se ejecuta la presentación al mismo tiempo que se cargan los modelos (la figura geométrica 3D usada como indicador y el modelo del edificio) en memoria, una vez terminada la presentación se visualiza los modelos en la pantalla del dispositivo móvil, se permite la navegación dentro del modelo mediante las teclas de dirección, mientras que el indicador visual se posiciona en las coordenadas especificadas.	Terminal
Se ejecuta la presentación, pero se detiene ya que no se puede cargar algunos o ninguno de los modelos memoria.	Terminal

Al igual que en las pruebas de aceptación para la Tarea 5, todos los resultados obtenidos son terminales debido a las circunstancias ya comentadas, adicionando sólo el hecho de que ahora se cuenta con un indicador visual junto con un rótulo con las coordenadas locales X,Y,Z que se usan para señalar la posible localización del dispositivo móvil dentro del modelo. Ya con este cambio la tabla de resultados observables queda de la siguiente forma:

Tabla 29 Caminos posibles de ejecución de la Tarea 6.

Caminos de ejecución	Resultados observables
Los archivos con los modelos se encuentran en la memoria externa, es accesible ya que tiene el nombre y extensión adecuados, se tiene suficiente memoria temporal y soporte para el API necesario.	Se ejecuta la presentación al mismo tiempo que se carga el modelo en memoria, una vez terminada la presentación se visualiza el modelo en la pantalla del dispositivo móvil y se permite la navegación dentro del modelo mediante las teclas de dirección.
Los archivos con los modelos se encuentran en la memoria externa, es accesible ya que tiene el nombre y extensión adecuados, pero no se tiene suficiente memoria temporal.	Se ejecuta la presentación, pero se detiene ya que no se puede cargar el modelo en memoria.
Los archivos con los modelos se encuentran en la memoria externa, es accesible ya que tiene el nombre y extensión adecuados, se tiene suficiente memoria temporal pero no soporte para el API necesario	Se ejecuta la presentación, pero se detiene ya que no se puede cargar el modelo en memoria.
Los archivos con los modelos se encuentran en la memoria externa, pero no es accesible ya que no tiene el nombre y/o la extensión adecuada.	Se ejecuta la presentación, pero se detiene ya que no se puede cargar el modelo en memoria.
Alguno de los archivos con los modelos no se encuentran en la memoria externa.	Se ejecuta la presentación, pero se detiene ya que no se puede cargar el modelo en memoria.
Los archivos con los modelos no se encuentran en la memoria externa.	Se ejecuta la presentación, pero se detiene ya que no se puede cargar el modelo en memoria.

De igual forma que lo sucedido en las pruebas anteriores de la Tarea 4 todos los resultados quedan clasificados como correctos, tal y como se muestra en la siguiente tabla:

CAPITULO VI: Desarrollo de la aplicación

Tabla 30 Valores posibles para los posibles caminos de ejecución de la Tarea 6.

Caminos de ejecución	Conjunto de valores
El archivo con el modelo se encuentra en la memoria externa, es accesible ya que tiene el nombre y extensión adecuados, se tiene suficiente memoria temporal y soporte para el API necesario.	Correcto
El archivo con el modelo se encuentra en la memoria externa, es accesible ya que tiene el nombre y extensión adecuados, pero no se tiene suficiente memoria temporal.	Correcto
El archivo con el modelo se encuentra en la memoria externa, es accesible ya que tiene el nombre y extensión adecuados, se tiene suficiente memoria temporal pero no soporte para el API necesario	Correcto
El archivo con el modelo se encuentra en la memoria externa, pero no es accesible ya que no tiene el nombre y/o la extensión adecuada.	Correcto
Alguno de los archivos con los modelos no se encuentran en la memoria externa.	Correcto
Los archivos con los modelos no se encuentran en la memoria externa.	Correcto

Al observar que existe redundancia en los caminos, es necesario redefinir algunos en uno sólo. Las tablas de caminos posibles y valores posibles para los mismos quedan de la siguiente forma:

Tabla 31 Nuevos caminos posibles de ejecución de la Tarea 6.

Caminos de ejecución	Resultados observables
Los archivos con los modelos se encuentran en la memoria externa, es accesible ya que tiene el nombre y extensión adecuados, se tiene suficiente memoria temporal y soporte para el API necesario.	Se ejecuta la presentación al mismo tiempo que se carga el modelo en memoria, una vez terminada la presentación se visualiza el modelo en la pantalla del dispositivo móvil y se permite la navegación dentro del modelo mediante las teclas de dirección.
Los archivos con los modelos se encuentran en la memoria externa, es accesible ya que tiene el nombre y extensión adecuados, pero no se tiene suficiente memoria temporal.	Se ejecuta la presentación, pero se detiene ya que no se puede cargar el modelo en memoria.
Los archivos con los modelos se encuentran en la memoria externa, es accesible ya que tiene el nombre y extensión adecuados, se tiene suficiente memoria temporal pero no soporte para el API necesario	Se ejecuta la presentación, pero se detiene ya que no se puede cargar el modelo en memoria.
Los archivos con los modelos se encuentran en la memoria externa, pero no es accesible ya que no tiene el nombre y/o la extensión adecuada.	Se ejecuta la presentación, pero se detiene ya que no se puede cargar el modelo en memoria.
Alguno o todos los archivos con los modelos no se encuentran en la memoria externa.	Se ejecuta la presentación, pero se detiene ya que no se puede cargar el modelo en memoria.

De igual forma que lo sucedido en las pruebas anteriores de la Tarea 4 todos los resultados quedan clasificados como correctos, tal y como se muestra en la siguiente tabla:

Tabla 32 Nuevos valores posibles para los posibles caminos de ejecución de la Tarea 6.

Caminos de ejecución	Conjunto de valores
El archivo con el modelo se encuentra en la memoria externa, es accesible ya que tiene el nombre y extensión adecuados, se tiene suficiente memoria temporal y soporte para el API necesario.	Correcto
El archivo con el modelo se encuentra en la memoria externa, es accesible ya que tiene el nombre y extensión adecuados, pero no se tiene suficiente memoria temporal.	Correcto
El archivo con el modelo se encuentra en la memoria externa, es accesible ya que tiene el nombre y extensión adecuados, se tiene suficiente memoria temporal pero no soporte para el API necesario	Correcto
El archivo con el modelo se encuentra en la memoria externa, pero no es accesible ya que no tiene el nombre y/o la extensión adecuada.	Correcto
Alguno o todos de los archivos con los modelos no se encuentran en la memoria externa.	Correcto

Para terminar con la prueba de aceptación se verifica que el sistema cumpla con los posibles caminos establecidos junto con sus respectivos conjuntos de valores. Una vez finalizado el desarrollo del código correspondiente a la Tarea 5 se observa que todos estos requisitos son válidos dentro del sistema.

Pruebas de Aceptación de la Tarea 7

La Tarea 7 se relaciona con la historia de usuario 7, busca la mejora de la Tarea 3 usando como bases la Tarea 4 y 6, agregando el reposicionamiento automático del indicador visual conforme al cálculo de la posible localización del dispositivo móvil mediante el conjunto de RSSIs obtenidos y el radio mapa pre-cargado. Además, se implementa en el sistema la capacidad para que una vez que terminada la aplicación en el dispositivo móvil, todos los procesos relacionados con el mismo lleguen a terminar correctamente tanto en el dispositivo móvil como en el servidor instalado en las terminales fijas. A partir de la historia de usuario se puede observar que existen 5 posibles resultados. La clasificación de los resultados se muestra en la siguiente tabla:

Tabla 33 Clasificación de resultados observables de la Tarea 7.

Resultados observables	Clasificación
Se ejecuta la presentación al mismo tiempo que se cargan los modelos (la figura geométrica 3D usada como indicador y el modelo del edificio) en memoria, una vez terminada la presentación se visualiza los modelos en la pantalla del dispositivo móvil, se permite la navegación dentro del modelo mediante las teclas de dirección, mientras que el indicador visual se posiciona y se actualiza el rótulo con las coordenadas locales X,Y,Z conforme a las coordenadas iniciales especificadas.	No terminal
Se ejecuta la presentación, pero se detiene ya que no se puede cargar algunos o ninguno de los modelos memoria.	Terminal
Una vez cargado el modelo se obtiene el valor RSSI de cada una de los módulos Bluetooth conectados a las terminales fijas.	No terminal
Una vez cargado el modelo se obtiene el valor RSSI de sólo algunos de los módulos Bluetooth conectados a las terminales fijas, para el resto se usó un valor de referencia.	No terminal
Una vez cargado el modelo no se puede obtener el valor RSSI de ninguno de los módulos Bluetooth conectados a las terminales fijas, por lo cual se hace uso de un valor de referencia para todos.	No terminal
Si el conjunto de RSSIs obtenidos contiene por lo menos un valor diferente del valor de referencia, se calcula la posible localización del dispositivo móvil dentro del modelo del edificio, usando esta localización para el reposicionamiento del indicar visual y la actualización del rótulo con las coordenadas locales X,Y,Z.	Terminal
Si el conjunto de RSSIs obtenidos contiene sólo valores de referencia, el indicador visual no cambia de posición y no se actualiza el rótulo con las coordenadas locales X,Y,Z.	Terminal

Al igual que en las pruebas de aceptación para la Tarea 3, la mayoría de los resultados obtenidos son no terminales conforme a los requisitos del cliente, la terminación de la tarea puede ser debido a un error o de calcular la posible localización del dispositivo móvil dentro del edificio mediante el conjunto de RSSIs obtenidos, en este caso específico sólo 3 de los resultados observables cumple con estas condiciones clasificando a estos mismo resultados como terminales.

Ahora es necesario identificar todos los posibles caminos que llevan a los resultados observables vistos hasta el momento, estos caminos se muestran en la siguiente tabla:

CAPITULO VI: Desarrollo de la aplicación

Tabla 34 Valores posibles para los posibles caminos de ejecución de la Tarea 7.

Caminos de ejecución	Resultados observables
Los archivos con los modelos se encuentran en la memoria externa, es accesible ya que tiene el nombre y extensión adecuados, se tiene suficiente memoria temporal y soporte para el API necesario.	Se ejecuta la presentación al mismo tiempo que se cargan los modelos (la figura geométrica 3D usada como indicador y el modelo del edificio) en memoria, una vez terminada la presentación se visualiza los modelos en la pantalla del dispositivo móvil, se permite la navegación dentro del modelo mediante las teclas de dirección, mientras que el indicar visual se reposiciona y se actualiza el rótulo con las coordenadas locales X,Y,Z conforme a las coordenadas iniciales especificadas.
Los archivos con los modelos se encuentran en la memoria externa, es accesible ya que tiene el nombre y extensión adecuados, pero no se tiene suficiente memoria temporal.	Se ejecuta la presentación, pero se detiene ya que no se puede cargar el modelo en memoria.
Los archivos con los modelos se encuentran en la memoria externa, es accesible ya que tiene el nombre y extensión adecuados, se tiene suficiente memoria temporal pero no soporte para el API necesario	Se ejecuta la presentación, pero se detiene ya que no se puede cargar el modelo en memoria.
Los archivos con los modelos se encuentran en la memoria externa, pero no es accesible ya que no tiene el nombre y/o la extensión adecuada.	Se ejecuta la presentación, pero se detiene ya que no se puede cargar el modelo en memoria.
Alguno o todos los archivos con los modelos no se encuentran en la memoria externa.	Se ejecuta la presentación, pero se detiene ya que no se puede cargar el modelo en memoria.
Selección de los módulos Bluetooth mediante sus respectivas MACs y establecimiento de comunicación con todos ellos.	Una vez cargado el modelo se obtiene el valor RSSI de cada una de los módulos Bluetooth conectados a las terminales fijas.
Selección de los módulos Bluetooth mediante sus respectivas MACs y establecimiento de comunicación con todos ellos, con pérdida posterior de la comunicación de algunos de los módulos.	Una vez cargado el modelo se obtiene el valor RSSI de sólo algunos de los módulos Bluetooth conectados a las terminales fijas, para el resto se usó un valor de referencia.
Selección de los módulos Bluetooth mediante sus respectivas MACs y establecimiento de comunicación con todos ellos, con pérdida posterior de la comunicación de todos los módulos.	Una vez cargado el modelo no se puede obtener el valor RSSI de ninguno de los módulos Bluetooth conectados a las terminales fijas, por lo cual se hace uso de un valor de referencia para todos.
Selección de los módulos Bluetooth mediante sus respectivas MACs y establecimiento de comunicación con algunos de ellos.	Una vez cargado el modelo se obtiene el valor RSSI de sólo algunos de los módulos Bluetooth conectados a las terminales fijas, para el resto se usó un valor de referencia.
Selección de los módulos Bluetooth mediante sus respectivas MACs y establecimiento de comunicación con algunos de ellos, con pérdida posterior de la comunicación de algunos de los módulos.	Una vez cargado el modelo se obtiene el valor RSSI de sólo algunos de los módulos Bluetooth conectados a las terminales fijas, para el resto se usó un valor de referencia.
Selección de los módulos Bluetooth mediante sus respectivas MACs y establecimiento de comunicación con algunos de ellos, con pérdida posterior de la comunicación de todos los módulos.	Una vez cargado el modelo no se puede obtener el valor RSSI de ninguno de los módulos Bluetooth conectados a las terminales fijas, por lo cual se hace uso de un valor de referencia para todos..
Selección de los módulos Bluetooth mediante sus respectivas MACs e imposibilidad del establecimiento de comunicación con todos ellos.	Una vez cargado el modelo no se puede obtener el valor RSSI de ninguno de los módulos Bluetooth conectados a las terminales fijas, por lo cual se hace uso de un valor de referencia para todos.
Obtención del RSSI de todos los módulos Bluetooth seleccionados mediante sus respectivas MACs.	Si el conjunto de RSSIs obtenidos contiene un valor diferente del valor de referencia, se calcula la posible localización del dispositivo móvil dentro del modelo del edificio, usando esta localización para el reposicionamiento del indicar visual y la actualización del rótulo con las coordenadas locales X,Y,Z.
Obtención del RSSI de sólo alguno(s) de los módulos Bluetooth seleccionados mediante sus respectivas MACs.	Si el conjunto de RSSIs obtenidos contiene por lo menos un valor diferente del valor de referencia, se calcula la posible localización del dispositivo móvil dentro del modelo del edificio, usando esta localización para el reposicionamiento del indicar visual y la actualización del rótulo con las coordenadas locales X,Y,Z.
No se obtiene ningún RSSI de los módulos Bluetooth seleccionados mediante sus respectivas MACs.	Si el conjunto de RSSIs obtenidos contiene sólo valores de referencia, el indicador visual no cambia de posición y no se actualiza el rótulo con las coordenadas locales X,Y,Z.

De igual que en las pruebas anteriores todos los resultados quedan clasificados como correctos, tal y como se muestra en la siguiente tabla:

CAPITULO VI: Desarrollo de la aplicación

Tabla 35 Valores posibles para los posibles caminos de ejecución de la Tarea 7

Caminos de ejecución	Conjunto de valores
Los archivos con los modelos se encuentran en la memoria externa, es accesible ya que tiene el nombre y extensión adecuados, se tiene suficiente memoria temporal y soporte para el API necesario.	Correcto
Los archivos con los modelos se encuentran en la memoria externa, es accesible ya que tiene el nombre y extensión adecuados, pero no se tiene suficiente memoria temporal.	Correcto
Los archivos con los modelos se encuentran en la memoria externa, es accesible ya que tiene el nombre y extensión adecuados, se tiene suficiente memoria temporal pero no soporte para el API necesario	Correcto
Los archivos con los modelos se encuentran en la memoria externa, pero no es accesible ya que no tiene el nombre y/o la extensión adecuada.	Correcto
Alguno o todos los archivos con los modelos no se encuentran en la memoria externa.	Correcto
Selección de los módulos Bluetooth mediante sus respectivas MACs y establecimiento de comunicación con todos ellos.	Correcto
Selección de los módulos Bluetooth mediante sus respectivas MACs y establecimiento de comunicación con todos ellos, con pérdida posterior de la comunicación de algunos de los módulos.	Correcto
Selección de los módulos Bluetooth mediante sus respectivas MACs y establecimiento de comunicación con todos ellos, con pérdida posterior de la comunicación de todos los módulos.	Correcto
Selección de los módulos Bluetooth mediante sus respectivas MACs y establecimiento de comunicación con algunos de ellos.	Correcto
Selección de los módulos Bluetooth mediante sus respectivas MACs y establecimiento de comunicación con algunos de ellos, con pérdida posterior de la comunicación de algunos de los módulos.	Correcto
Selección de los módulos Bluetooth mediante sus respectivas MACs y establecimiento de comunicación con algunos de ellos, con pérdida posterior de la comunicación de todos los módulos.	Correcto
Selección de los módulos Bluetooth mediante sus respectivas MACs e imposibilidad del establecimiento de comunicación con todos ellos.	Correcto
Obtención del RSSI de todos los módulos Bluetooth seleccionados mediante sus respectivas MACs.	Correcto
Obtención del RSSI de sólo alguno(s) de los módulos Bluetooth seleccionados mediante sus respectivas MACs.	Correcto
No se obtiene ningún RSSI de los módulos Bluetooth seleccionados mediante sus respectivas MACs.	Correcto

Al no existir redundancia en los caminos, la tabla de caminos posibles con que se cuenta hasta el momento es definitiva.

Para terminar con las pruebas de aceptación se verifica que el sistema cumpla con los posibles caminos establecidos junto con sus respectivos conjuntos de valores. Una vez finalizado el desarrollo del código correspondiente a la Tarea 7 se observa que todos estos requisitos son válidos dentro del sistema.

Al ser la Tarea 7 la última que forma parte del desarrollo de la primera versión del sistema para el proyecto de tesis, se dan por concluidas las fases de planificación de la entrega e iteraciones.

CAPITULO VII: Resultados y conclusiones

Se presenta un panorama general de la implementación del Sistema propuesto en la tesis, los resultados obtenidos de las pruebas hechas al mismo, así como las conclusiones a las que se puede llegar en base a los resultados.

En capítulos previos se hizo mención sobre el lenguaje de programación, la técnica de localización y el tipo de tecnología a usar para la implementación del sistema de localización y ubicación a desarrollar en la tesis, explicando el porqué de la selección de cada uno de estos. Es necesario ahora entrar en detalles sobre la infraestructura del sistema, la etapa de recolección de datos, las pruebas de funcionalidad, la forma de evaluación de las pruebas, las limitantes del sistema y las conclusiones a las que se pueden llegar sobre el mismo.

Infraestructura del sistema

En base a las APIs seleccionadas para la implementación del sistema de localización y ubicación a desarrollar en la tesis y las plataformas que dan soporte a las mismas, se cuenta con dos archivos uno de extensión .jad y otro de extensión .jar los cuales realizan la función de instaladores de la aplicación para las terminales móviles que forma parte del sistema. Además también se tiene un archivo de extensión .iso, el cual es la imagen de un Live-CD de la distribución Linux Xubuntu 8.14 para las terminales fijas creada mediante el programa Remastersys Backup. Este Live-CD cuenta con la pre-instalación del JDK de Java, la pre-compilación de la API AvetanaBluetooth (modificada con soporte hasta para 3 módulos Bluetooth) e incorpora la aplicación terminal fija que forma parte del sistema.

Este sistema se basa en conexiones punto a punto entres las terminales móviles y las terminales fijas (Figura 103). Como se mencionó previamente la terminal móvil ejecuta la aplicación del Sistema que se usa para la localización y ubicación, por lo tanto requiere establecer comunicación con al menos un módulo Bluetooth conectado a una de las terminales fijas. Cada módulo puede establecer comunicación simultánea hasta con 8 dispositivos (conforme a las especificaciones de Bluetooth para piconets). En caso de que más dispositivos requieran el establecimiento de la comunicación con un mismo modulo Bluetooth, estos quedan en espera hasta que uno de los primeros 8 dispositivos ya conectados termine la comunicación con el modulo Bluetooth.

Para la implementación física del sistema se utilizaron módulos Bluetooth clase 2, los cuales conforme a las especificaciones tienen un alcance de hasta 10 m en espacio libre, sin embargo, al utilizarse en interiores el alcance se reduce debido al ruido existente en el ambiente, el cual se debe principalmente a las paredes, pisos y muros que debe de atravesar la señal.

En base a pruebas de penetración de las señales hechas, se optó por colocar los módulos en el techo del pasillo del primer piso del IEC para disminuir la interferencia. De esta forma es posible realizar la localización y ubicación de los dispositivos móviles en cualquier posición dentro del edificio (planta alta o baja) con un mínimo de 3 módulos Bluetooth conectados a 2 Laptops que fungen como terminales fijas.

Los módulos Bluetooth fueron colocados con la finalidad de tener una forma esférica radial de la señal, cuyo radio mínimo de alcance (en condiciones de ruido) fue aproximadamente de 6m, lo cual garantiza que una terminal móvil pueda entablar comunicación mínimo con un módulo Bluetooth, en el peor caso cuando la terminal móvil se encuentra en uno de los extremos del edificio y dentro de un cubo; en el extremo opuesto se tiene que la comunicación se entabla con los 3 módulos Bluetooth, este el mejor caso, la terminal móvil se encuentra en el pasillo ubicado entre los 3 módulos Bluetooth. Para garantizar esto el modulo Bluetooth central se coloca aproximadamente a 12 metros entre cada módulo Bluetooth extremo, conforme a la forma y dimensiones del edificio se cumple que cada

CAPITULO VII: Resultados y conclusiones

módulo Bluetooth extremo se encuentra a una distancia aproximada de 6 metros de las paredes laterales del edificio.

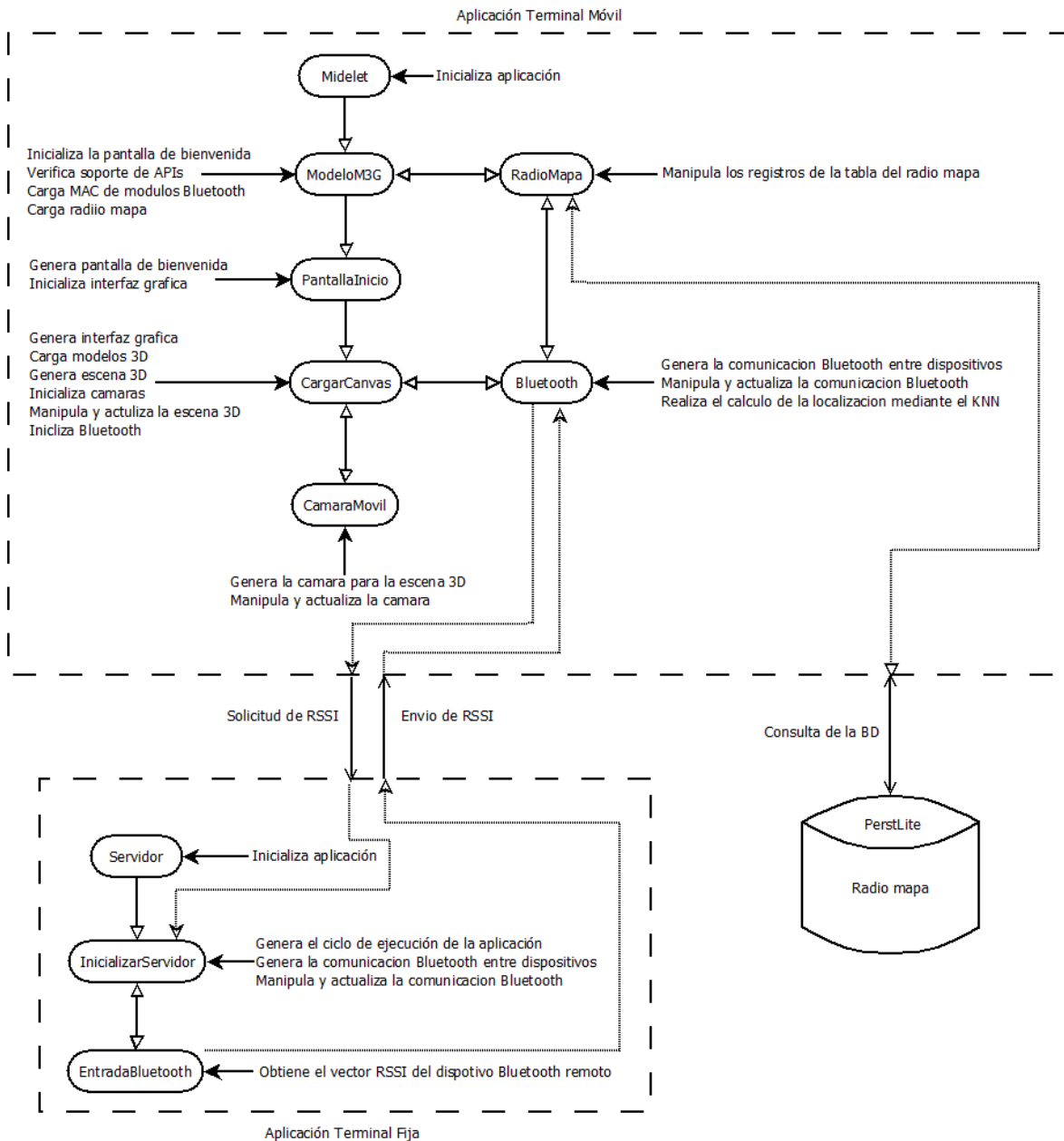


Figura 103 Diagrama a bloques del funcionamiento de las aplicaciones para la Terminal Móvil y Fija del sistema.

Cada módulo Bluetooth funciona mediante un puerto USB y debido a la forma en que se deben instalar los módulos físicamente en el edificio, no es posible conectarlos directamente a las terminales fijas. Por lo tanto fue necesario el uso de cables extensores USB, sin embargo los cables de tipo pasivo existentes en el mercado tienen un alcance máximo de 5 m (debido a especificaciones técnicas). En su lugar se optó por la construcción de cables extensores USB basados en cable UTP, que permite extensiones de este tipo de hasta 30 m⁴⁵.

⁴⁵ <http://www.instructables.com/id/How-to-Extend-yout-USB-using-UTP/>

Etapa de entrenamiento-recolección de datos

Una vez hecha la instalación física del Sistema se procede a la recolección de datos en las localizaciones definidas por el usuario, para ello como se mencionó en capítulos previos, se realizó la implementación de una aplicación para los dispositivos móviles usada para la generación del radio mapa del edificio. La aplicación permite la recolección de una gran cantidad de localizaciones (mientras exista espacio en memoria); cada localización está formada por los siguientes elementos:

- **Información física:** Son las coordenadas 3D (x, y, z) con respecto al edificio de la localización y es proporcionada por el usuario.
- **Información simbólica:** Es la posición cuya base de referencia es un cierto objeto o entidad (inmóvil) y es proporcionada por el usuario.
- **Vector de potencias:** Conjunto de RSSI medidos desde cada módulo Bluetooth y es proporcionado por la propia aplicación.

Para tener una correcta discretización del espacio y mejorar el resultado por parte del sistema de localización y ubicación, se optó por que en cada localización propuesta por el usuario se realizará la recolección de datos para 4 diferentes direcciones (norte, sur, este y oeste), siempre y cuando sea posible obtener al menos un valor RSSI de algunos de los módulos Bluetooth para cada dirección independiente. Además de que cada valor de RSSI obtenido de cada módulo Bluetooth se deriva de un número N de muestras definidas por el usuario, que son recolectadas y promediadas por la aplicación de la terminal fija.

Tomando como base pruebas hechas previamente al variar el valor N con respecto al tiempo que tarda la terminal móvil en recibir el valor RSSI desde un sólo módulo Bluetooth (Tabla 36), se decidió usar un valor de N=100 dentro del sistema teniendo así un tiempo de respuesta aceptable para este caso en específico.

Tabla 36 Tiempos necesarios para recolectar muestras de RSSI, promediar las muestras y enviar el valor promedio obtenido desde cada módulo Bluetooth conectado a una terminal fija.

Numero de muestras recolectas	Tiempo total para realizar las tareas (ms)
300	2000
200	1300
100	700
50	200
20	180
10	110

Cada localización propuesta debe cumplir con alguna de las siguientes condiciones:

- Para los pasillos debe ubicarse el dispositivo en medio, ya sea entre dos cubículos (si hay cubículos a ambos lados del pasillo), frente a un cubículo (si sólo hay un cubículo en alguno de los lados del pasillo), frente o en la puerta de entrada del edificio, frente a alguno de los laboratorios, frente a alguno de los baños, frente a la escalera o debajo de esta, en el espacio del escritorio de la secretaria, frente a la puerta de emergencia, frente a la venta del segundo piso ubicada sobre la puerta de emergencia del primer piso o en algunos de los dos escalones principales de la escalera.
- Para los baños, cubículos y laboratorios cuyas dimensiones se aproximan a las de un cubículo estándar, se toma en cuenta una sola localización, el centro. Para los de gran dimensión se realiza una segmentación conforme a las dimensiones aproximadas de un cubículo estándar, de tal forma que se trate de obtener un número de localizaciones proporcional a cubículos independientes (tratando de englobar un número entero de cubículos dentro del espacio).

CAPITULO VII: Resultados y conclusiones

Conforme a la distribución hecha de los módulos Bluetooth dentro del edificio, se obtuvo un área de cobertura en el primer piso de aproximadamente 36 m de largo, considerando el pasillo y los cubículos alrededor del mismo. Por otro lado en el segundo piso se tuvo un área de cobertura de 32 m de largo (por la atenuación de la señal al atravesar el techo del primer piso), ésta misma se vio reducida en los extremos con respecto al área de cobertura obtenida para el primer piso

El espacio de muestras está formado por 193 localizaciones de las cuales 169 corresponden con aquellas que cumplen con las condiciones para los pasillos del primer y segundo piso, las otras 35 corresponden con los cubículos, laboratorios y baños a los que se tuvo acceso tanto para la toma de muestras como para las pruebas del sistema.

En base a lo comentado en capítulos previos, al hacer uso de la técnica de localización para interiores basada en radiofrecuencia usando el algoritmo KNN, ésta incluye como característica propia el hecho de no requerir de una etapa de entrenamiento previa para su funcionamiento, es decir, sólo es necesaria una etapa de recolección de datos antes de su puesta en marcha. Es por ello que se puede decir que tanto la etapa de recolección de datos como la de entrenamiento están a la par una de otra.

Una vez terminada la recolección de datos se procede a la generación del archivo con el radio mapa patrón. Se da por terminado tanto el entrenamiento como la recolección de datos para el sistema de localización y ubicación a desarrollar en la tesis.

Pruebas de funcionalidad y su evaluación

Una vez generado el archivo con el radio mapa patrón y el archivo binario con extensión M3G del modelo 3D del IEC y el modelo 3D que representa la terminal móvil; son cargados en la memoria externa de la terminal móvil en la cual se instalará y ejecutará la aplicación que realizará la localización. Para ello se considera que la infraestructura física del sistema coincide con la que existía al momento de que se recolectó la información para la generación del radio mapa patrón.

Con las terminales fijas en funcionamiento se procede a la pruebas de funcionalidad del sistema completo, para ello se pone en ejecución la aplicación en la terminal, la cual como primer paso revisará que el dispositivo cuente con conexión Bluetooth. Si la terminal móvil cuenta con conexión Bluetooth se procede a la verificación de la existencia en memoria externa de los archivos con el radio mapa patrón, el archivo binario con extensión M3G que contiene el modelo 3D y el archivo con las direcciones MAC de los módulos Bluetooth conectados a las terminales fijas con los cuales se comunicará la terminal móvil.

Si se cumple con la existencia de estos archivos, la aplicación muestra inicialmente una presentación en la cual se da a conocer el nombre del sistema, el funcionamiento de la aplicación para dispositivos móviles que forma parte del mismo, así como el fin que ésta persigue (Figura 104). Una vez termina la presentación la aplicación procederá a realizar de forma automática la detección y conexión con las terminales fijas que se encuentran activos. Para ello la aplicación realiza la conexión con los módulos Bluetooth designados, mediante las direcciones MAC especificadas previamente.

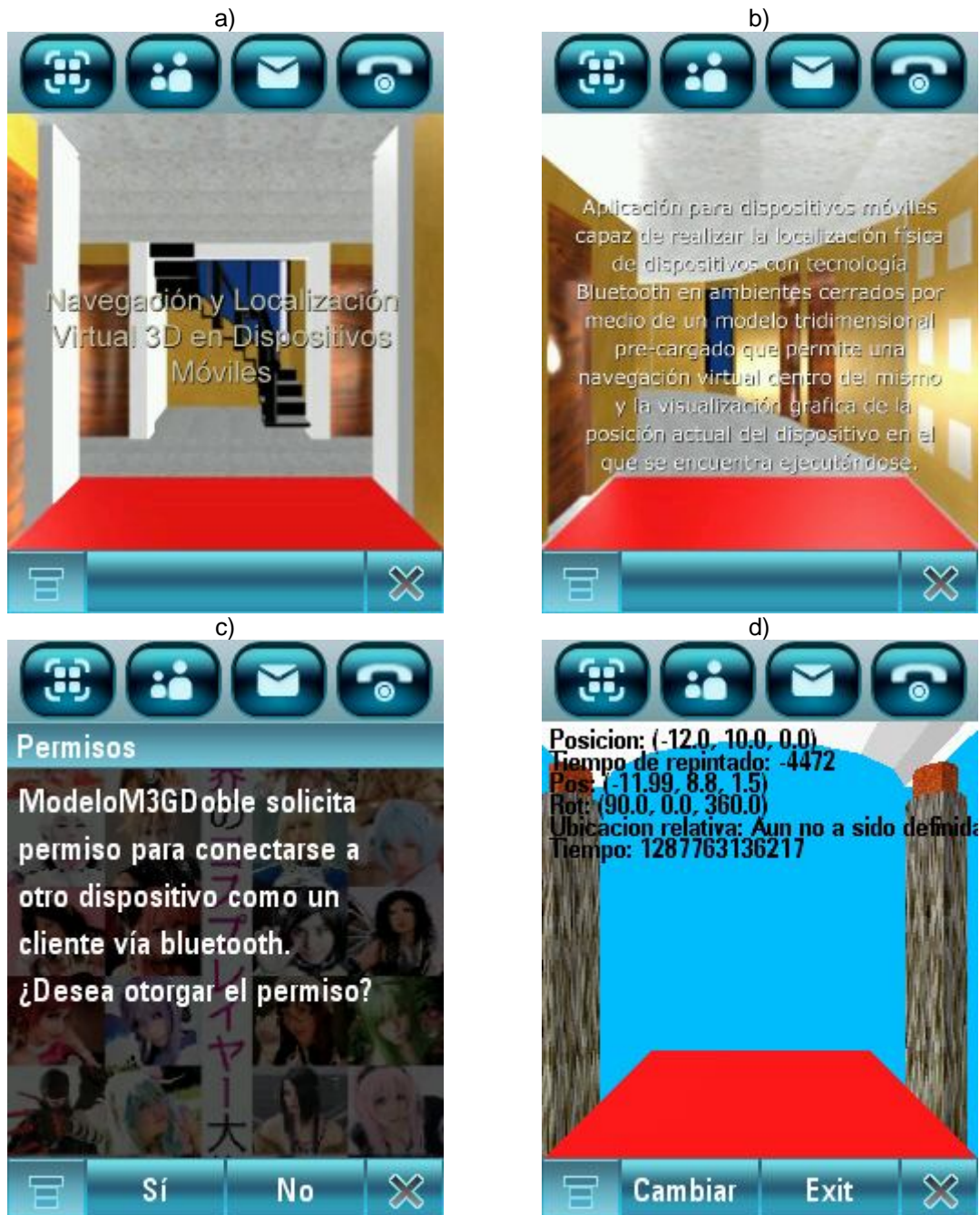


Figura 104 Imágenes correspondientes a la pantalla de la terminal móvil que muestra: a) El nombre del sistema b) La presentación del funcionamiento de la aplicación para dispositivos móviles que usa el sistema, c) Petición de permiso al usuario para que la terminal móvil realice conexión con otro dispositivo vía Bluetooth y d) Vista de la posición actual de la terminal móvil a través de su indicador predefinido, dentro de edificio representado por el modelo 3D precargado del mismo.

Una vez que la aplicación obtiene los RSSI de los módulos Bluetooth (con los que se pudo realizar una conexión exitosa), se procede a usar estos datos como entrada para la implementación hecha del KNN, la cual a su vez proporciona como salida la información necesaria para que la aplicación muestre la posición (absoluta y relativa) estimada de la terminal móvil dentro del edificio (Figura 105), permitiendo además la navegación virtual dentro del mismo (sin alterar los resultados) mediante un modelo 3D con transparencia del edificio (Figura 106).



Figura 105 Image que muestra la ubicación de la terminal móvil: Derecha) dentro del edificio e Izquierda) dentro del modelo 3D.

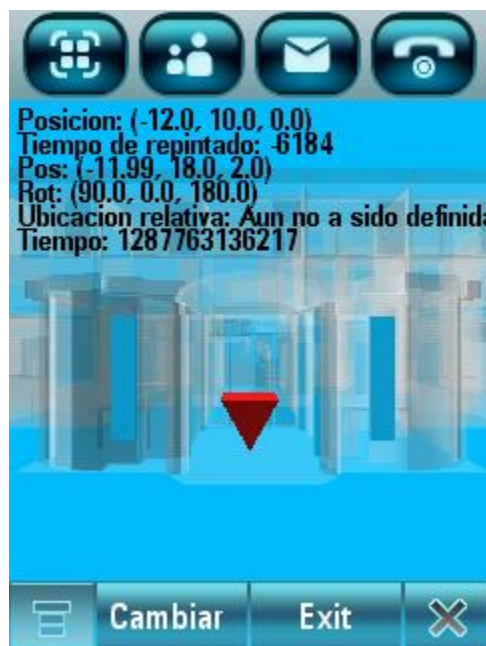


Figura 106 Cambio de modalidad de la aplicación para la terminal móvil que permite navegar con la cámara dentro de un modelo 3D con transparencias del edificio, sin alterar los resultados de la localización (absoluta y relativa) del propio dispositivo.

Es importante destacar que el algoritmo del KNN clásico asume que del conjunto de vecinos más cercanos obtenidos para el vector de potencias dado, siempre existe un sólo vecino que minimiza la distancia euclidiana. Sin embargo como se menciona en capítulos previos existen casos en los cuales esta suposición no es válida, por lo cual se optó por obtener un promedio de las coordenadas de los vecinos más cercanos que minimizan la distancia euclidiana y así obtener un mejor estimado de la localización de la terminal móvil.

Cuando la aplicación de la terminal móvil cuenta con la información de la localización del mismo, esta se muestra de forma visual en la pantalla del dispositivo incluyendo otros parámetros como son: la posición (coordenadas con respecto a los ejes x,y,z) y orientación (ángulos con respecto a los ejes

CAPITULO VII: Resultados y conclusiones

x,y,z) de la cámara en el espacio 3D, tiempo (en ms) que le toma al dispositivo repintar la pantalla y tiempo (en ms) usado para estimar y actualizar la posición del modelo 3D de la terminal móvil dentro espacio 3D (Figura 107).



Figura 107 Imagen correspondiente a la pantalla de la aplicación de la terminal móvil en la cual se muestra la información adicional con que se cuenta al momento que el sistema realiza la localización del propio dispositivo.

Tomando en cuenta los parámetros usados por el sistema, que fueron mencionadas en este capítulo, uno de los aspectos a evaluar dentro de la funcionalidad es el tiempo que le toma al sistema realizar el cálculo de la localización y visualización de la terminal móvil dentro del modelo 3D. Este tiempo de respuesta de 22000 ms en el mejor de los casos y de 57000 ms en el peor de los casos, con un promedio en general de 30000 ms.

Tal y como se menciona en el primer capítulo, la utilidad y éxito de un sistema de localización está dado principalmente por dos puntos: la exactitud y precisión del mismo. En general las pruebas de funcionalidad del Sistema muestran los siguientes resultados:

- Cuando la terminal móvil se encuentra dentro del área de muestreo en el primer piso, el cálculo en la localización tiene una exactitud de 2m con una precisión de entre el 90% y 80%.
- Cuando la terminal móvil se encuentra dentro del área de muestreo en el segundo piso, el cálculo en la localización tiene una exactitud de 3 m con una precisión de entre el 80% y 60%.
- Cuando la terminal móvil se encuentra fuera del área de muestreo pero cercana a la misma, el cálculo en la localización tiene una exactitud de entre 4 y 5 m con una precisión del 50%.

Limitaciones del sistema

El Sistema debe tomar en cuenta que cada módulo Bluetooth puede presentar un cuello de botella, en el caso de que más de 8 dispositivos móviles deseen establecer comunicación con este mismo. Esto se debe a las propias especificaciones de Bluetooth para piconets, lo que provocaría un aumento en el tiempo para el cálculo de la localización de los dispositivos móviles.

Otra limitación que presenta la infraestructura actual del Sistema es el rango de cobertura y el número de módulos Bluetooth usados. El Sistema fue implementado usando módulos Bluetooth clase 2 con un rango de cobertura (esférica) de 6 m (considerando el ruido del ambiente), usando 3 módulos sólo se tuvo una área de cobertura lineal (considerando el pasillo y cubículos) de aproximadamente 36 m para el primer piso y de 32 m para el segundo piso. Otro de los inconvenientes consiste en el tipo de cableado usado para conectar cada módulo Bluetooth a la terminal fija, aún cuando previamente se presentó una alternativa basada en el uso de cable UTP para extensiones comerciales USB pasivas, esta alternativa no permite tener una extensión mayor a 30 m., en dado caso de que se haga uso de módulos Bluetooth clase 1, puede ser necesario el uso de cables extensores USB activos lo que incrementaría el costo de la infraestructura del sistema.

Otra limitación en la infraestructura actual del Sistema, tiene que ver con limitaciones en la implantación de la especificación JSR-82 para J2ME, se relaciona con el hecho de que la aplicación desarrollada para el dispositivo es incapaz de acceder al RSSI de los módulos Bluetooth, por lo que fue necesario el uso de terminales fijas que ejecutan su propia aplicación para resolver este problema, tal y como se había comentado en capítulos previos. En el caso de que esta limitación no existiera se reduciría el software y hardware usado en la infraestructura del sistema, además de que algunas de las limitantes mencionadas previamente no existirían.

En cuanto a limitaciones de instalación de software del Sistema, se presenta el hecho de que los archivos de extensión .jad y .jar mencionados previamente tienen la finalidad de instalar la aplicación en la terminal móvil, siempre que la misma cuente con soporte para J2ME. Sin embargo esta aplicación necesita de archivos adicionales los cuales deben estar en el directorio raíz de la terminal móvil antes de poder ejecutarse. Estos archivos contienen las direcciones mac de los módulos Bluetooth (rssi.lst), el radio mapa patrón (radiomapa.lst), el modelo 3D del edificio (iec.m3g), el modelo 3D del edificio con transparencias (modelo.m3g) y el modelo 3D que sirve como indicador de la terminal móvil (indicador.m3g). Estos archivos deben precargarse en la ruta indicada de forma manual y en el caso de uno de los archivos falte y/o no tenga el nombre adecuado, la aplicación de la terminal móvil no podrá ser ejecutada.

Las limitaciones de ejecución de software del Sistema se ven relacionadas por una parte en lo referente a las aplicaciones de la terminal fija con la presencia de módulos Bluetooth, si no existe, no se podrán ejecutar. Por otro lado las aplicaciones para las terminales móviles, se ven afectadas por el soporte para J2ME. Además, es necesaria la presencia de alguna implementación de la especificación JSR-82 y del hardware de Bluetooth tanto para la ejecución de la aplicación que recolecta datos para generar el radio mapa como para la aplicación encargada de localizar la terminal fija. Es necesaria la existencia de alguna implementación de la especificación JSR-184 en el caso de la aplicación encargada de localizar la terminal móvil.

Las limitaciones de tiempo usado para la localización por parte del Sistema, se dan principalmente a las restricciones de software y hardware de los propios dispositivos implantadas por los fabricantes, así como por los desarrolladores de la propia tecnología Bluetooth (SIG) y de la implementación para la especificación JSR-82. Como se observó durante las pruebas funcionales del Sistema, las terminales móviles usadas no permiten conexiones simultáneas con diversos dispositivos (conforme a la especificación para piconets) a diferencia de las terminales fijas. Otros factores que afectan el tiempo de respuesta del sistema, son el tiempo usado por un par de dispositivos Bluetooth para establecer una conexión el cual se encuentra en un rango de entre 1500 a 3000 ms con un promedio general de 2200 ms. Mientras que en caso de que no pueda realizarse la conexión entre dispositivos, se recibe una respuesta de falta de conexión que no es inmediata y consume un tiempo de entre

CAPITULO VII: Resultados y conclusiones

6000 a 8000 ms con un promedio en general de 7000 ms. Además, la desconexión no es instantánea, consumiendo un tiempo de entre 200 a 300 ms con un promedio en general de 250 ms.

Una última limitante se relaciona con la exactitud y precisión en la localización por parte del Sistema, estos puntos se ven afectados por los tiempos ya mencionados, lo que provocan una falta de respuesta en tiempo real. Así como también por la técnica de localización y el método (KNN) usados, ya que dependen del nivel de discretización del entorno y que por la sencillez de los mismos, el Sistema no contempla el auto-aprendizaje por retroalimentación. Además de todo esto, es necesario tomar en consideración las propias limitantes en la infraestructura del Sistema que se refieren al área de cobertura de la misma. Todo esto en conjunto provoca que el Sistema tenga diferentes niveles de precisión y exactitud conforme a las condiciones que rodean a la terminal móvil, las cuales pueden quedar englobas de la siguiente forma:

- El cálculo de la localización para una terminal móvil ubicada dentro del área de recolección de muestras para el primer piso y que se encuentra estática o en movimiento muy lento, tiene una exactitud de 2 m con una precisión del 90%, mientras que si se encuentra en movimiento constante la precisión disminuye hasta un 80%.
- El cálculo de la localización para una terminal móvil ubicada dentro del área de recolección de muestras para el segundo piso y que se encuentra estática o en movimiento muy lento tiene una exactitud de 3 m con una precisión del 80%, mientras que si se encuentra en movimiento constante la precisión disminuye hasta un rango de entre el 70% y el 60%.
- El cálculo de la localización para una terminal móvil ubicada fuera del área de recolección de muestras pero cercana a la misma, tanto en el primer piso como en el segundo piso, ya sea que se encuentre estática o en movimiento tiene una exactitud de entre 4 y 5 m con una precisión del 50%.

Conclusiones

El Sistema desarrollado para la tesis demostró contar con parámetros de trabajo bastante aceptables en comparación con las implementaciones existentes para localización en ambientes internos. Se observó de forma practica la veracidad de los estudios hechos a la tecnología Bluetooth (ya mencionados en capítulos previos), los cuales analizan la susceptibilidad de esta tecnología al ruido producido por la presencia de diversos dispositivos con esta misma y/o otras tecnologías que trabajan de forma similar. Se proponía la idea de que al hacer uso de la tecnología Bluetooth se tendría una mejor precisión y exactitud en el cálculo de la localización de terminales móviles haciendo uso de técnicas y métodos de localización simples, lo cual se pudo observar en los resultados arrojados por las pruebas funcionales comentadas en este mismo capítulo. En base a lo ya mencionado también fue posible cumplir con el objetivo de tener una localización y navegación 3D lo cual muy pocos sistemas lo han llevado a cabo, ya que en general estos mismo trabajan con espacios 2D.

Otra de los objetivos que se tenía al momento de desarrollar el Sistema, fue la portabilidad y fácil adaptación del mismo para cualquier tipo de ambiente interno. Lo cual, aunque las aplicaciones que forman parte del Sistema no son totalmente autónomas tanto en instalación, ejecución y/o uso, cuentan con los siguientes puntos a favor:

- Son amigables e intuitivos.
- Requieren de bajos recursos para su ejecución.
- Los archivos adicionales para su funcionamiento no necesitan de ninguna modificación y en caso de requerirla no es necesario conocimientos especializados, además de que s{olo deben ser ubicados en las rutas indicadas.
- Las aplicaciones para la terminal móvil cuenta con un instalador, mientras que para la terminal fija se cuenta con un live-cd que permite su instalación o ejecución directa sobre el mismo.

CAPITULO VII: Resultados y conclusiones

En lo referente al hardware del Sistema este cumple con las siguientes características:

- Es de uso común.
- Su costo es accesible.
- No se necesita contar con ningún tipo de topología en específico.
- La cobertura de la señal se considera aceptable conforme al tipo de dispositivos usados durante la pruebas del Sistema, pudiendo mejorar si se cuenta con dispositivos más robustos. Es decir, si se cambian los módulos Bluetooth clase 2 por clase 1.

Además de lo ya mencionado hasta el momento, otra de las cualidades del Sistema a la ahora de ser implantado en un nuevo ambiente interno es la fácil y rápida generación del modelo 3D del mismo. Ya que sólo se necesita de una herramienta de diseño 3D para generar el archivo binario con extensión .M3G sin la necesidad de modificar código alguno. De igual forma por la sencillez de las técnicas y métodos de localización usadas no es necesaria de una etapa específica de entrenamiento, por lo que la propia etapa de recolección de muestras abarca este punto. Esto permite que el Sistema esté funcionando en un tiempo menor en comparación con otros sistemas más complejos.

Aún cuando la idea de tener un Sistema que permitiera la localización de terminales móviles en tiempo real no se alcanzó. Se puede considerar que aunque los tiempos de respuesta no son los mejores, si son de utilidad, basando esta idea en los niveles de exactitud y precisión obtenidos durante las pruebas de funcionalidad y que fueron expuestos previamente. Además es de gran importancia considerar el hecho de que estos tiempos se deben en gran medida por restricciones de software y hardware impuestos por los propios fabricantes y por los desarrolladores de las implementaciones para la especificación JSR-82. Probablemente si las condiciones cambiaran se aceleraría el tiempo necesario para el cálculo de la localización de las terminales móviles. Así como también se tendría una mejora en los niveles de exactitud y precisión de la misma.

Aparte de las ideas presentadas hasta el momento y el estado actual de las mismas con respecto al Sistema desarrollado para la tesis. No se debe olvidar el hecho de que los resultados obtenidos permitieron adicionalmente contar con una publicación sobre este tema de nominada "3D Indoor Location and Navigation System Based on Bluetooth". La publicación de hecho fue enviada y aceptada en el congreso denominado "CONIELCOM 2011: 21st International Conference on Electrical Communications and Computers" el cual se llevó a cabo en San Andres Cholula, Puebla, Mexico los días 28 Febrero a 2 de Marzo del 2011 y que gracias al mismo fue publicado en las páginas 271-277 de las memorias del congreso a través de la IEEE contando con el ISBN 978-1-4244-9558-0.

Es importante destacar el hecho de que este artículo ha sido citado dentro de diferentes artículos los cuales se muestran a continuación ordenados cronológicamente de forma descendente:

- Honggui Li, "**Low-Cost 3D Bluetooth Indoor Positioning with Least Square**", Wireless Personal Communications, Septiembre 2014, Volume 78, Issue 2, páginas 1331-1344.
- Pallavi Holey, Vishwa T. Gaikwad, "**Blue-fi Based Navigation System**", International Journal of Pure and Applied Research in Engineering and Technology, 2014, ISSN 2229-6107, factor de impacto 0.621.
- Ali Motamedi, "**Improving Facilities Lifecycle Management Using RFID Localization And BIM-Based Visual Analytics**", Tesis doctoral, 13 de Enero de 2014, Concordia University, Montreal, Quebec, Canada.
- Mohamed Adel, Serhani, Wassim, El-Hajj, "**Movement-aware and QoS-driven indoor location and mobile service discovery framework**", Journal International Journal of Wireless and Mobile Computing archive, Volume 6 Issue 5, October 2013, Pages 481-500.
- Mohammad Mostafa Soltani, "**Neighborhood Localization Method for Locating Construction Resources Based on RFID and BIM**", Tesis de maestría, 18 de Noviembre de 2013, Concordia University, Montreal, Quebec, Canada
- Luís Conceição, Marília Curado, "**Onto Scalable Wireless Ad Hoc Networks: Adaptive and Location-aware Clustering**", Ad Hoc Networks, Volume 11, Issue 8, Noviembre 2013, Páginas 2484-2499, ISSN 1570-8705.

CAPITULO VII: Resultados y conclusiones

- Rohit Agrawal and Ashesh Vasalya, “**Bluetooth Navigation System using Wi-Fi Access Points**”, The Computing Research Repository, 2012
- Yusong Pang, Hans Veeke, Gabriël Lodewijks, “**Design of tracing and tracking network for educational building utilization**”, Networking, Sensing and Control (ICNSC) 10th IEEE International Conference on, 10-12 de Abril, 2013, Páginas 276-281.
- Ankita Paul, Rohit Das, Abhik Sarka. “**Mall Navigation System Using Wi-Fi Router**”, International Journal of Advance Computer Research, ISSN 2249-7277, Volumen 2, Número 4, Issue 6, 2012
- Juraj Machaj, Peter Brida, and Michal Mlynka, “**Impact of Used Communication Technology on the Navigation System for Hybrid Environment**”, Journal of Computer Networks and Communications, Volume 2012 (2012), Article ID 731015, 10 pages
- Rohit Agrawal and Ashesh Vasalya, “**Bluetooth Navigation System using Wi-Fi Access Points**”, The Computing Research Repository, 08 de Abril, 2012.
- Eduardo Metola, Ana M. Bernardos, “**POSTER An Embedded Fusion System for Location Management**”, 8th International ICST Conference, MobiQuitous 2011, Copenhagen, Dinamarca, 6-9 de Diciembre, 2011. Página 233-237

Para finalizar, en lo que se refiere a trabajo a futuro, una de las primeras tareas a considerar es la migración de la aplicación para terminales móviles al sistema operativo Android, tal y como se comento en el capítulo I. Desde el punto de vista de la programación esto no sería muy difícil, debido a que comparte un lenguaje cuyo diseño de aplicaciones es similar al uso por J2ME y la sintaxis se hereda en términos generales de Java. La propuesta de migración se basa en el hecho de que el mercado de terminales móviles cuyo sistema operativo es Android ha crecido considerablemente y se encuentra en expansión. Además de que los dispositivos que implementan este sistema operativo se abaratan al no tener que pagar licencias por tratarse de un proyecto Open Source. Las capacidades con las que cuenta este sistema operativo han posibilitado el incorporar cada vez más elementos de hardware como son sensores electromagnéticos, cámaras, módulos WiFi, módulos Bluetooth, etc., los cuales con la tecnología actual se han visto una reducción en cuanto a tamaño y costos de producción. Además se puede decir que la programación necesaria para hacer funcionar el dispositivo Bluetooth incorporado en las terminales móviles que trabajan bajo Android es similar, a la usada para J2ME. Y al igual que éste, cuenta con su propia implementación denominada Bluetooth API equivalente a la especificación JSR-82 de Java. Por otro lado en lo que se refiere al API para gráficos 3D también se cuenta con implementaciones para OpenGL-ES similares a la especificación JSR-239 de Java, pero que a diferencia de ésta, no sólo permiten trabajar bajo OpenGL 1.0 Y 1.1, dependiendo de la versión de Android trabajan con OpenGL-ES 1.0 y 1.1 (Android 1.0 a 2.1), 2.0 (Android 2.2 a 4.2) y 3.0 (Android 4.3 o superior). Sin embargo al igual que se comentó en capítulos previos el desarrollo de modelos 3D bajo este enfoque es tedioso y necesita un gran número de líneas de código. Es por ello que se propone el uso de jPCT-AE un motor 3D que tiene una funcionalidad y codificación similar a la implementación para Mobile 3D Graphics o JSR-184, teniendo como única restricción la falta de soporte para multi-texturas. Como un plus adicional está implementación permite el manejo de archivos binarios con extensión .OBJ soportado por la herramienta para diseño 3D denominada Blender, la cual fue usada para generar el modelo 3D del IEC.

Otros cambios que se proponen al sistema se dividen en dos vertientes, una consiste en aumentar la infraestructura del Sistema al implementar un servidor Ethernet en el ambiente interno donde se desea instalar. Con la idea de automatizar la carga de los archivos adicionales que usa la aplicación para la terminal fija los cuales contendrán las versiones del modelo 3D a usar (con y sin transparencias), las direcciones mac de los módulos Bluetooth y el radio mapa. Para ello se propone el uso de la implementación propia de Android para el acceso a redes denominado java.net.* APIs, junto con la biblioteca Open Source denominada “ZXing” usada para procesamiento de imágenes con código de barras 1D/2D basado en java. Dando lugar a la posibilidad de que mediante un código QR se tenga la dirección URL del servidor donde se encuentran los archivos a usar y mediante una

CAPITULO VII: Resultados y conclusiones

conexión vía WiFi, permita descargar a memoria RAM o a la memoria de almacenamiento externa. Un cambio más dentro de esta misma vertiente consiste en conectar las terminales fijas al servidor Ethernet y usando la información enviada a cada terminal móvil se realice el cálculo de la localización, para que mediante una aplicación web adicional se pueda mostrar las áreas donde se ha implementado el Sistema, así como las terminales móviles que lo estén usando. Esto como un medio para buscar, ubicar y verificar la posición de las terminales móviles por algún otro usuario interesado en este tipo de información.

Por último, el cambio propuesto para la vertiente contraria a la mencionada consiste en reducir la infraestructura del Sistema, para ello se pretende cambiar las terminales fijas por módulos Bluetooth que cuente con la capacidad para ser manejados mediante micro-controladores. Eliminado la necesidad de uso de terminales fijas así como de la aplicación para las mismas, con lo que se tienen tendrían los siguientes puntos a favor:

- Reducción en costo de hardware.
- Reducción en consumo de energía por parte de hardware.
- Reducción en el área necesaria para la instalación del hardware.
- Eliminación de la necesidad de cableado dentro de la infraestructura física del Sistema.

Para ello se ha contemplado el uso de módulos Bluetooth con interfaz UART para las conexiones necesarias con el micro controlador. Proveyendo al micro controlador del software adecuado para el manejo de la información que necesite la terminal móvil para el cálculo de la localización. Para cumplir con esta meta se propone el uso del módulo RN-41 de Roving Networks, Inc., de clase 1 y que a diferencia de otros módulos permite conforme al estándar de Bluetooth la configuración en modo de operación maestro o esclavo (la mayoría de los módulos para micro controladores sólo trabajan como esclavos). Además, conforme a la hoja de especificaciones del módulo, este cuenta con la capacidad de obtener el RSSI del dispositivo conectado al mismo mediante comandos AT.

Bibliografía

(Alvarez Toro y De la Hoz Sánchez): Toro Alvarez, P. C., De la Hoz Sánchez, M. F., "Desarrollo de Juego 3D para Plataforma Android", Memoria para optar al título de Ingeniero Civil en Computación e Informática, Universidad del Bío-Bío, Departamento de Ciencias de la Computación y Tecnologías de Información, Diciembre 2010.

(Ambrona Tabernilla): Ambrona Tabernilla, L. D., "Sistema de Localización en Interiores", Director: Fernando Pérez Costoya, Proyecto de fin de Carrera, Universidad Politécnica de Madrid, Departamento de Arquitectura y Tecnología de Sistemas Informáticos, 27 de Abril de 2008.

(Archundia Papacetzi): Archundia Papacetzi, F. M., "Wireless Personal Area Network (WPAN) & Home Networking", Director: Luis Gerardo Guerrero Ojeda, Tesis profesional, Universidad de las Américas Puebla, Escuela de Ingeniería, Departamento de Ingeniería Electrónica, 16 de Diciembre de 2003.

(Bahl y Padmanabhan): P. Bahl and V. N. Padmanabhan, "RADAR: An In-Building RF-based User Location and Tracking System", in *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE* vol. 2, 2000, pp. 775-784 vol.2.

(Bandara, Hasegawa y Inoue): Bandara, U., Hasegawa, M., Inoue, M., Morikawa, H., Aoyama, T., "Design and implementation of a Bluetooth signal strength based location sensing system", in *Proceedings of IEEE Radio and Wireless Conference*, Atlanta, GA, 2004, pp. 319-322

(Bargo Rodríguez): Bargo Rodríguez, D., "Videojoc 3D per Android basat en Angry Birds", Director: Lluís Pérez Vidal, Tesis profesional, Universitat Politècnica de Catalunya, Facultat de Informàtica de Barcelona, 9 de Enero de 2012.

(Battiti, Le Nhat y Villani): R. Battiti, T. Le Nhat, A. Villani, "Location-Aware Computing: a Neural Network Model for Determining Location in Wireless Lans," 2002.

(Berrueta): Berrueta, Diego, "Programacion extrema y software libre." *VI Jornadas de Software Libre en Asturias*. Principado de Asturias, 22 de Enero de 2006, pp. 1-8.

(Borraz Gómez): Borraz Gómez, I. J., "Estudi i implementació d'una aplicació de localització en xarxes de sensors", Director: Carles Gómez, Treball de fi de carrera, Universitat Politècnica de Catalunya, Departament d'Enginyeria Telemàtica, 25 de Febrero de 2005.

(Bray y Sturman): Bray, J., Sturman, C. F., "Bluetooth 1.1: Connect Without Cables", 2nd Edition, Inglaterra: Prentice Hall, 2002.

(Canós, Letelier y Penadés): Canós, J. H., Letelier, P., Penadés, M. C., "Metodologías ágiles en el desarrollo de software". En *Actas de las VIII Jornadas de Ingeniería del software y Bases de Datos*, Alicante, España 2003.

(Castro, Chiu y Kremenek): P. Castro, P. Chiu, T. Kremenek, R. R. Muntz, "A probabilistic room location service for wireless networked environments," in *UbiComp '01: Proceedings of the 3rd international conference on Ubiquitous Computing*, London, UK: Springer-Verlag, 2001, pp. 18-34.

(Chehimi, Coulton y Edwards): Chehimi, F., Coulton, P., Edwards, R., "Evolution of 3D Games on Mobile Phones", in *International Conference on Mobile Business*, Washington, DC, USA, pp. 173-179.

(Corchado): Corchado, D. J., "Modelado en 3D y composición de objetos" Curso: Segundo Ciclo Ingeniería Informática, Universidad de Salamanca, Facultad de Ciencias, 2002.
(Correia): Correia, P., "Guía Práctica del GPS", 1ra Edición, España: MARCOMBO, S.A., 2002.

(D. A. Davison): Davison, D. A. "Killer Game Programming in Java", First Edition, USA: O'REELLY, 2005.

(Davison y Radenahmud): Davison, A. and Radenahmud, S., "Comparing M3G and JSR-239 for 3D Games Programming", In *CGames 2008*, Wolverhampton UK, 2008.

(Deitel y Deitel): Deitel, H. M., Deitel, P. J. "Como Programar en C, C++ y Java", 4ta Edición, Mexico: PEARSON, 2004.

(Elnahrawy, Li y Martin): E. Elnahrawy, X. Li, and R. P. Martin, "The limits of localization using signal strength: a comparative study," In: *Proceedings of The First IEEE International Conference on Sensor and Ad hoc Communications and Networks*, Santa Clara, CA, 2004, pp. 406-414.

(García Barrales): Cristian Ignacio G. B., "Monte Carlo aplicado a la auto-localización de robots", Director: Daniel Vallejo Rodríguez, Tesis profesional, Universidad de las Américas Puebla, Escuela de Ingeniería, Departamento de Ingeniería en Sistemas Computacionales, 17 de mayo de 2005.

(Golmie, Van Dyck y Soltanian): Golmie, N., Van Dyck, R. E., Soltanian, A., Tonnerre, A., Rébala, O., "Interference of Bluetooth and IEEE 802.11 simulation modeling and performance evaluation". In *Proceedings of the 4th ACM Intl. Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, Roma, Italia, 2001, pp. 11-18.

(González Seco): González Seco, J. A., "El lenguaje de programación C#", Publicación propia, 2001

(Guérin): Guérin, B. A., "Lenguaje C++: Programación Unix & Windows, Standard Template Library, Creación de un programa de archivado – Recurso Informáticos", Barcelona: Ediciones Eni, 2005.

(Hall y Knaus): Hall, C., Knaus, C. "Safety-Critical Profile Specification Version 1.0.1 (Difference Specification) (Annotated)". KHRONOS GROUP, 12 de Marzo de 2002.

(Henriksson): Henriksson, D., "Proximity Payments with Betala.se", Supervisor: Jerry Eriksson, Master's Thesis, Umea University, Department of Computing Science, 3 de Junio de 2007.

(Hernández Sánchez y Blázquez Román): Hernández Sánchez, S., Blázquez Román, F. ".Net Framework", Publicación propia, Universidad de Salamanca, Departamento de Informática y Automática, 2005.

(Hightower y Borriello): Hightower, J., Borriello, G. "Location Sensing Techniques", 2001.

(Huang y Rudolph): Huang, A. S., Rudolph, L., "Bluetooth Essentials for Programmers", New York: CAMBRIDGE UNIVERSITY PRESS, 2007.

(Jode): Jode, M. de, "Programming Java 2 Micro Edition on Symbian OS", Chichester: WILEY, 2004.

(Kammer, McNutt y Senese): Kammer, D., McNutt, G., Senese, B., Bray, J. "Bluetooth Application Developer's Guide: The Short Range Interconnect Solution", USA: SYNGRESS, 2002.

(Kent): Kent, B., "Extreme Programming Explained: embrace change", USA: Addison-Wesley, 2000.

(Kotanen, Hännikäinen y Leppäkoski): Kotanen, A., Hännikäinen. M., Leppäkoski, H., Hämäläinen, T. D., "Experiments on Local Positioning with Bluetooth," In: *Proceedings of the International Conference on Information Technology: Computers and Communications*, 2003, Las Vegas, Nevada, pp. 297-303.

(Maestre, Machuca y Navarro): Maestre, I. M., Machuca, M., Navarro, A., Velasco, J. R. "Un enfoque práctico para la localización de usuarios mediante Bluetooth en entornos domóticos", España: CEUR Workshop Proceedings, 2005.

(Martinez Gens y Urios de Heras): Martinez Gens, L. E., Urios de Heras, M., "Tecnologías de Localización y Posicionamiento para Servicios Basados en Localización (LBS)", Publicación BIT #154, Sección Tecnología y Sociedad, pp. 68-70, Diciembre 2005 – Enero 2006, <http://www.coit.es/modulos/anunciantes/index.php>.

(Montiel Hernández): Montiel Hernández, N. M., "Posicionamiento Basado en Redes Inalámbricas para la Determinación de Funcionalidades en un GIS", Director: Marco Antonio Moreno Ibarra, Tesis profesional, Instituto Politécnico Nacional, Centro de Investigación en Computación, Laboratorio de Geoprocusamiento, Junio de 2003.

(Muller): Muller, N. J., "Bluetooth Demystified", USA: MacGraw Hill, 2001.

(Mulloni, Barakonyi y Wagner): Mulloni, A., Wagner, D., Barakonyi, I., Schmalstieg, D., "Indoor Positioning and Navigation with Camera Phones" *IEEE Pervasive Computing*, Vol. 8, no. 2, pp. 22-31, April-June, 2009.

(Neerunjun, Bhiwoo y Nagowah): Neerunjun, S., Bhiwoo, C., Nagowah, L., "A Framework for Android and J2ME Bluetooth Communication." *International Journal of Computer Applications*, Vol. 57, No. 22, pp. 23-33. Noviembre 2012.

(Nokia Corporation): Nokia Corporation, "S60 Platform: Bluetooth API Developer's Guide. Version 2.1", Forum Nokia Global Web Site and S60 Web Site, 27 de Junio de 2008.

(Palm Incorporated): Palm Incorporated, "Palm OS® Bluetooth Support", Whitepaper, 12 de Diciembre de 2000.

(Patil, Kim y Ni): Patil, A.P., Kim, D.J., Ni, L.M., "A study of frequency interference and indoor location sensing with 802.11b and Bluetooth technologies", *International Journal of Mobile Communications*, Vol. 4, No. 6, pp.621-644, 2006.

(Priyantha, Chakraborty y Balakrishnan): N. B. Priyantha, A. Chakraborty, H. Balakrishnan, "The cricket location-support system," in *MobiCom '00: Proceedings of the 6th annual international conference on Mobile computing and networking*. New York, NY, USA: ACM Press, 2000, pp. 32-43.

(Pulli, Aarnio y Roimela): Pulli, K., Aarnio, T., Roimela, K., Vaarala, J., "Mobile 3D Graphics with OpenGL ES and M3G", USA: MORGAN KAUFFMAN, Noviembre del 2007.

(Resendiz Trejo): Resendiz Trejo, I. J., "Las maquinas de vectores de soporte para identificación en línea", Tesis profesional, Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional, Departamento de Control Automático, Septiembre de 2006.

(Rodríguez, Escudero y Pece): Rodríguez, M., Escudero, C. J., Pece, J. P., "Blueps: sistema de localización en interiores utilizando Bluetooth", Departamento de Electrónica e Sistemas Universidade da Coruña, U.R.S.I , Noviembre de 2005.

(Rojano Romero): Rojano Romero, A., "Sistema de Localización y Seguimiento de Móviles Terrestres Utilizando el Sistema de Posicionamiento Global", Tesis profesional, Instituto Politécnico Nacional, Escuela Superior de Ingeniería Mecánica y Eléctrica Unidad Profesional "Adolfo López Mateos", Julio de 2003.

(Sánchez Vítores): Sánchez Vítores, R., (Diciembre de 2004 - Enero de 2005). "ILS (Indoor Location Systems) Sistemas de Localización en Interiores", Publicación BIT #148, Sección Tendencias, pp. 57, Diciembre de 2004 – Enero 2005, <http://www.coit.es/modulos/anunciantes/index.php>.

(Torres): Torres, J. M., "Bluetooth en .Net", doNetManía #35, pp. 26-35, Marzo 2007, <http://www.dotnetmania.com/Articulos/index.html>.

(Vazquez-Briseno): Vazquez-Briseno, M., Mendez-Flores, M., Jimenez-Garcia, E., Nieto-Hipolito, J. I., "Evaluation of Java-Based Platforms for mHealth Solutions", International Journal of Computer Science and Mobile Applications, Vol. 2, No. 5,, pp. 20-35, Mayo 2014.

(Want, Hopper y Falcao): R. Want, A. Hopper, V. Falcao, and J. Gibbons, "The active badge location system" *ACM Trans. Inf. Syst.*, vol. 10, no. 1, pp. 91-102, January 1992.

(Werb y Lanzi): J. Werb and C. Lanzi, "Designing a positioning system for finding things and people indoors," *Spectrum, IEEE*, vol. 35, no. 9, pp. 71-78, 1998.