



UNIVERSIDAD NACIONAL AUTÓNOMA  
DE MÉXICO

FACULTAD DE INGENIERÍA  
DIVISION DE INGENIERÍA ELÉCTRICA

DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE  
LOCALIZACIÓN EN REDES INALÁMBRICAS  
HÍBRIDAS WI-FI Y AD-HOC

T E S I S

QUE PARA OBTENER EL TÍTULO DE  
INGENIERO EN TELECOMUNICACIONES  
E  
INGENIERO EN COMPUTACIÓN  
PRESENTA

MARÍA FERNANDA HERNÁNDEZ SANDOVAL  
YAZMANI OMAR GALINDO MARTÍNEZ

DIRECTOR DE TESIS: DR. JAVIER GÓMEZ CASTELLANOS





Universidad Nacional  
Autónoma de México



**UNAM – Dirección General de Bibliotecas**  
**Tesis Digitales**  
**Restricciones de uso**

**DERECHOS RESERVADOS ©**  
**PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

## DEDICATORIAS

---

---

A **Dios**, que me permitió llegar hasta este momento en compañía de mi familia y amigos.

A mis padres, **Paulina Sandoval Vega** y **Fernando Fabián Hernández Velasco**, como muestra de mi amor y aprecio hacia ellos, ya que por el cuidado y atenciones que tuvieron conmigo, logré finalizar mi carrera y continuar en la carrera de la vida. Además de que con su amor y cariño me mostraron el mejor camino y me enseñaron que en la vida siempre obtienes lo que mereces.

A mis hermanos, **Alma Alejandra** y **Luis Daniel Hernández Sandoval**, por lo grandes detalles que siempre han tenido conmigo.

A mis abuelos y abuelas, tíos y tías, primos y primas tanto de parte de la familia de mi papá como la de mi mamá, por todo el cariño y apoyo brindado en los momentos difíciles, por los consejos y recuerdos inolvidables.

A **Yazmani Omar Galindo Martínez**, por enseñarme que la vida siempre tiene un lado bueno y sólo basta con cambiar la perspectiva para encontrarlo, además de su apoyo incondicional a cada momento.

**Fernanda Hernández**

## **DEDICATORIAS**

---

---

*A Dios por ser mi compañero en todo momento y permitir que las cosas se dieran.*

*A mi mamá por su cariño, tiempo y devoción que tuvo conmigo, siempre estuvo aunque no la llamara, siempre me enseñó que yo puedo lograr cualquier meta que me proponga, gracias por tenerme amor y confianza, recuerde que este logro es de los dos.*

*A mi padre, por dedicarme parte de su tiempo, admiración y enseñarme que no importa que tan difícil estén las cosas siempre existe un camino, si en verdad quieres llegar a él.*

*A mis hermanos para que les sirva como ejemplo de superación, pues yo se que ustedes serán mejores.*

*A Fernanda por enseñarme el lado recto de las cosas, gracias por todo lo que me has dado y hemos compartido.*

**Yazmani Galindo**

## AGRADECIMIENTOS

---

---

A la **Universidad Nacional Autónoma de México** por brindarme una educación sólida e integral durante mi estancia en ella.

Al **Dr. Javier Gómez Castellanos** por la confianza y apoyo proporcionado durante la elaboración de este trabajo de tesis y el estímulo para continuar con otros estudios profesionales.

A la **DGAPA/PAPIIT** por el apoyo económico y el material brindado para llevar a cabo este trabajo de tesis

Al **Dr. José Ismael Martínez López** por la amistad y apoyo basado en la palabra y enseñanzas de Dios.

A mi amigo e ingeniero **Rodrigo Atahualpa Sánchez Telésforo**, que me brindó su amistad apoyo incondicional en todos los sentidos.

A mis amigas **Adriana Arriaga Mendoza** y **Graciela Alvarado Martínez** por brindarme su amistad durante 20 y 10 años respectivamente, además de mostrarme que el tiempo no puede borrarla.

A todos mis amigos, **Fabiola Alavez Alvarez, Anabel Sánchez Acosta, Guadalupe Vázquez Pérez, Nuria Rangel Rivera, Yolanda Zeferino, Alma Iliana Ramos, Nancy Nanliby Sánchez, Roberto Alfonso Gallardo Pacheco, Hugo Alfonso Gómez Cruz, Miriam Calvillo, Carlos Villanueva Zúñiga, Juan Luis Almaráz, Guadalupe Calvillo, Homero Hernández Díaz y David García** porque gracias a ellos he conocido el valor de la amistad y como ser una mejor persona.

**Fernanda Hernández**

## **AGRADECIMIENTOS**

---

---

*A la UNAM, mi segundo hogar, gracias por brindarme una educación de gran nivel, agradezco todas las experiencias y lo más importante, gracias por enseñarme que no todo es estudio y libros.*

*Al Dr. Javier Gómez Castellanos por la enorme paciencia y confianza que depositó en este trabajo, gracias por su apoyo.*

*A la DGAPA/PAPIIT por apoyarnos con el material y los recursos necesarios para llevar a buen fin este trabajo de tesis.*

*A mis profesores, la mayoría excelentes algunos no tanto, sin embargo, cada uno me enseñó algo diferente que hasta la fecha me ha servido.*

*A mis amigos, pues cada uno hizo que los momentos en la universidad fueran diferentes, gracias por su apoyo, pues hicieron mas llevadera la estancia en ella.*

**Yazmani Galindo**

## ÍNDICE

ÍNDICE.....	1
ÍNDICE DE FIGURAS .....	5
ÍNDICE DE TABLAS.....	8
INTRODUCCIÓN .....	9
OBJETIVOS.....	11
CAPÍTULO I .....	12
TECNOLOGÍAS DE LOCALIZACIÓN .....	12
1.1 Introducción.....	12
1.2 Localización mediante redes móviles y satelitales.....	15
1.2.1 Tecnologías basadas en la identidad celular.....	16
1.2.1.1 Cell ID.....	16
1.2.1.2 Cell Global Identity.....	16
1.2.1.3 Enhanced Cell ID.....	17
1.2.2 Técnicas basadas en la red.....	18
1.2.2.1 Direction of Arrival (DOA).....	18
1.2.2.2 Time of Arrival (TOA).....	20
1.2.2.3 Time Diference of Arrival (TDOA).....	21
1.2.2.4 Huella Multirrayecto (MF).....	23
1.2.3 Técnicas basadas en la modificación del sistema móvil.....	25
1.2.3.1 Enhanced Observed Time Difference (E-OTD).....	25
1.2.3.2 Advanced Forward Link Trilateration (A-FLT).....	28
1.3 Localización mediante redes inalámbricas.....	28
1.3.1 Bluetooth.....	29
1.3.2 Wireless Fidelity (Wi-Fi).....	29
1.3.3 Banda ultra-ancha.....	31
1.4 Global Positioning System (GPS).....	32
1.4.1 Introducción.....	32
1.4.2 Funcionamiento.....	33
1.4.2.1 Trilateración.....	33
1.4.2.2 Medición de distancias.....	34
1.4.2.3 Control de Tiempo.....	35
1.4.2.4 Ubicación de satélites en el espacio.....	35
1.4.2.5 Corrección de errores.....	36
1.4.3 Localización mediante GPS y señal de Televisión.....	38
CAPÍTULO II.....	39
WIRELESS LOCAL AREA NETWORK (WLAN).....	39
2.1 Introducción.....	39
2.2 Componentes que conforman una WLAN.....	40
2.2.1 Terminales.....	40
2.2.2 Equipos Dedicados.....	41
2.2.3 Access Point.....	41
2.2.4 Antenas.....	44
2.2.4.1 Antenas Direccionales.....	44

2.2.4.2 Antenas Sectoriales.....	44
2.3 Arquitectura Física.....	45
2.4 Topologías.....	46
2.4.1 Redes Ad-hoc.....	48
2.4.2 Redes Infraestructura.....	49
2.5 Roaming.....	50
2.6 Los estándares 802.11x.....	52
2.6.1 Introducción.....	52
2.6.2 El estándar 802.11.....	53
2.6.3 Evolución del estándar 802.11.....	54
2.6.3.1 802.11a.....	54
2.6.3.2 802.11b.....	55
2.6.3.3 802.11g.....	56
2.6.3.4 Otras variantes del 802.11.....	56
2.6.4 Capa física.....	57
2.6.4.1 Descripción.....	57
2.6.4.2 Estándares de difusión de frecuencias para Capa I (Tipos de Modulación).....	59
2.6.5 Capa Media Access Control (MAC).....	64
2.6.5.1 Descripción.....	64
2.6.5.2 Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA).....	65
2.6.5.3 Formato del Frame.....	67
CAPÍTULO III.....	69
LA APLICACIÓN WIRELESS NETWORK POSITION SYSTEM (WINPS).....	69
3.1 Introducción.....	69
3.2 Propuesta de solución.....	70
3.2.1 Consideraciones Iniciales.....	71
3.2.1.1 Elementos Físicos.....	73
3.2.1.2 Elementos de Hardware.....	74
3.2.1.3 Elementos de Software.....	76
3.2.2 Delimitar el área de trabajo de los APs.....	78
3.2.3 Calibración del modelo de posicionamiento.....	79
3.2.4 Creación e implementación de un software.....	80
3.2.5 Verificación y realimentación.....	81
CAPÍTULO IV.....	82
EL DESARROLLO DE WINPS.....	82
4.1 Plano de la edificación.....	82
4.2 Mediciones de niveles en factores de transmisión.....	86
4.2.1 Mapa de la distribución de factores.....	86
4.2.2 Relación de los factores con la estructura de la edificación.....	87
4.3 Características del sistema de cómputo.....	90
4.3.1 Estructura del Sistema de Archivos de Linux.....	90
4.3.2 Herramientas de monitoreo de una red Wi-Fi en Linux.....	93
4.3.2.1 Wireless Extensions.....	93
4.3.2.2 KWiFiManager.....	101
4.3.2.3 KwaveControl.....	102
4.3.2.4 WaveLANApplet.....	103



4.3.3 Programación Gráfica en Linux.....	104
4.3.3.1 Qt.....	105
4.3.3.2 GTK+.....	106
4.3.3.3 Borland Kylix, el entorno de desarrollo para WiNPS.....	107
4.3.4 Características utilizadas en el sistema del cliente y el servidor.....	110
4.4 Documentación del software.....	113
4.4.1 Diagramas de Flujo de Datos.....	113
4.4.2 Diseño de interfaz.....	116
4.4.2.1 Launcher.....	116
4.4.2.2 Módulo Monitoreo.....	117
4.4.2.3 Módulo de Graficación.....	118
4.4.2.4 Módulo de Mapeo.....	119
4.4.2.5 Módulo de Localización.....	120
4.4.2.6 Módulo de Ayuda.....	121
4.4.3 Diagramas de casos de uso.....	122
4.4.4 Programación de la aplicación.....	131
4.4.4.1 Módulo Launcher.....	131
4.4.4.2 Función BuscarString.....	132
4.4.4.3 Módulo Monitoreo.....	133
4.4.4.4 Módulo de Graficación.....	135
4.4.4.5 Módulo de Mapping.....	136
4.4.4.6 Módulo de Localización.....	140
4.4.4.7 El algoritmo de localización.....	143
4.4.5 Pruebas preliminares del sistema.....	150
4.5 Consideraciones finales sobre el programa.....	154
CAPÍTULO V.....	155
APLICACIÓN Y PRUEBAS FINALES.....	155
5.1 Introducción.....	155
5.2 Prueba del módulo Launcher.....	156
5.2.1 Descripción.....	156
5.2.2 Resultados.....	156
5.3 Prueba del módulo graficación.....	157
5.3.1 Descripción.....	157
5.3.2 Resultados.....	157
5.4 Prueba del módulo de monitoreo.....	158
5.4.1 Descripción.....	158
5.4.2 Resultados.....	158
5.5 Prueba del módulo de mapeo.....	160
5.5.1 Descripción.....	160
5.5.2 Resultados.....	160
5.6 Prueba del módulo de ayuda.....	161
5.6.1 Descripción.....	161
5.6.2 Resultados.....	161
5.7 Prueba del módulo de localización.....	162
5.7.1 Descripción.....	162
5.7.2 Resultados.....	162

CAPÍTULO VI .....	167
CONCLUSIONES .....	167
6.1 Acerca del proceso de diseño e implementación de WinPS. ....	167
6.2 Sobre nuestro algoritmo de localización basado en el mapeo. ....	169
6.3 Mejoras que se le pueden hacer a WinPS. ....	170
6.4 Sobre la caracterización de la red. ....	170
APÉNDICES .....	173
A. NDISWRAPPER. ....	173
B. Instalación de la tarjeta PCMCIA Linksys WPC54G version 1.3. ....	174
C. El intervalo de estabilización. ....	176
ACRÓNIMOS Y ABREVIATURAS .....	178
REFERENCIAS Y BIBLIOGRAFÍA .....	183

## ÍNDICE DE FIGURAS

FIGURA 1.1 AOA CON 2 ESTACIONES BASE. ....	18
FIGURA 1.2 AOA CON 3 ESTACIONES BASE. ....	19
FIGURA 1.3 REPRESENTACIÓN DEL ARREGLO DE ANTENAS. ....	20
FIGURA 1.4 REPRESENTACIÓN GRÁFICA DE LA LOCALIZACIÓN POR TRIANGULACIÓN. ....	21
FIGURA 1.5 REPRESENTACIÓN DEL SISTEMA DE LOCALIZACIÓN TDOA. ....	22
FIGURA 1.6 SISTEMA DE LOCALIZACIÓN TDOA CON VARIANTE PARA PRECISIÓN. ....	23
FIGURA 1.7 EJEMPLO DEL REPORTE EN UN SISTEMA MF. ....	24
FIGURA 1.8 MÉTODO E-OTD HIPERBÓLICO. ....	26
FIGURA 1.9 MÉTODO E-OTD CIRCULAR. ....	27
FIGURA 1.10 EJEMPLO DE GDOP. ....	36
FIGURA 1.11 FUENTES DE ERROR EN LA TRANSMISIÓN DE LA SEÑAL DE UN SATÉLITE GPS. ....	37
FIGURA 1.12. SISTEMA DE LOCALIZACIÓN MEDIANTE TV. ....	38
FIGURA 2.1. EJEMPLO DE UNA RED INALÁMBRICA. ....	39
FIGURA 2.2 AP EN MODO ROOT. ....	42
FIGURA 2.3 AP EN MODO BRIDGE. ....	43
FIGURA 2.4 AP EN MODO REPETIDOR. ....	43
FIGURA 2.5 ARQUITECTURA FÍSICA DE UNA WLAN. ....	45
FIGURA 2.6 EJEMPLO DEL MODO DE OPERACIÓN BSS Y BSA. ....	46
FIGURA 2.7 EJEMPLO DEL MODO DE OPERACIÓN IBSS. ....	47
FIGURA 2.8 EJEMPLO DEL MODO DE OPERACIÓN ESS. ....	47
FIGURA 2.9 EJEMPLO DE UNA RED AD-HOC. ....	48
FIGURA 2.10 EJEMPLO DE UNA RED DE INFRAESTRUCTURA. ....	49
FIGURA 2.11 REUTILIZACIÓN DE CANALES. ....	50
FIGURA 2.12 EJEMPLO DE ROAMING. ....	50
FIGURA 2.13 ROAMING DE UN USUARIO ESTÁTICO. ....	51
FIGURA 2.14 EL 802.11 Y MODELO OSI. ....	53
FIGURA 2.15 ESQUEMA CARACTERÍSTICO DEL ESPECTRO DISPERSO. ....	59
FIGURA 2.17 CANALES SIN TRASLAPE UTILIZADOS POR EL ESTÁNDAR 802.11. ....	61
FIGURA 2.18 BARKER CODE. ....	61
FIGURA 2.19 LA TÉCNICA DE GESTIÓN FDM. ....	62
FIGURA 2.20 ESQUEMA GENERAL DE OFDM. ....	62
FIGURA 2.21 ORTOGONALIDAD DE OFDM. ....	63
FIGURA 2.22 ARQUITECTURA MAC. ....	65
FIGURA 2.23 TRAMA MAC GENERAL. ....	67
FIGURA 2.24 TRAMA MAC DETALLADA. ....	67
FIGURA 2.25 TRAMA DE CONTROL DETALLADA. ....	68
FIGURA 3.1 DIAGRAMA DE BLOQUES REPRESENTATIVO DEL SISTEMA. ....	71
FIGURA 3.2 DIAGRAMA DE BLOQUES REPRESENTATIVO DEL SISTEMA. ....	72
FIGURA 3.3 EVOLUCIÓN DE WINPS. ....	73
FIGURA 3.4 ESQUEMA DE LA PRIMER BARRERA TÉCNICA. ....	73
FIGURA 3.5 ESQUEMA DE LA SEGUNDA BARRERA TÉCNICA. ....	75
FIGURA 3.6 ESQUEMA DE LA TERCER BARRERA TÉCNICA. ....	76
FIGURA 3.7 FLUJO DE DATOS EN WINPS. ....	80
FIGURA 4.1 DISTRIBUCIÓN ACTUAL DEL WORKSPACE. ....	82
FIGURA 4.2 PERÍMETRO DEL WORKSPACE. ....	83
FIGURA 4.3 DIVISIONES DE MUROS REALES EN EL WORKSPACE. ....	83
FIGURA 4.4 DIVISIONES DE MUROS FALSOS EN EL WORKSPACE. ....	84
FIGURA 4.5 DISTRIBUCIÓN DE LOS APs EN EL WORKSPACE. ....	85
FIGURA 4.6 MAPA DE DISTRIBUCIÓN DE LA SEÑAL SUPERPUESTA AL PERÍMETRO DEL WORKSPACE. ....	86
FIGURA 4.7 MAPA DE DISTRIBUCIÓN VS ESTRUCTURA DE CONCRETO. ....	87
FIGURA 4.8 MAPA DE DISTRIBUCIÓN VS ESTRUCTURA DE FALSO CONCRETO. ....	88
FIGURA 4.9 MAPA DE DISTRIBUCIÓN VS ESTRUCTURA TOTAL DEL WORKSPACE. ....	89

FIGURA 4.10 EJEMPLIFICACIÓN DEL PATH. ....	91
FIGURA 4.11 PARTES EN LA QUE SE IMPLEMENTAN LAS WIRELESS EXTENSIONS. ....	93
FIGURA 4.12 GUI DE KWAVECONTROL. ....	103
FIGURA 4.13 GUI DE WAVELANAPPLET. ....	104
FIGURA 4.14 RELACIÓN DE X WINDOW CON EL OS LINUX. ....	104
FIGURA 4.15 LOGOTIPO DE QT. ....	105
FIGURA 4.16 LOGOTIPO DE GTK. ....	106
FIGURA 4.17 LOGOTIPO DE KYLIX 3.0. ....	107
FIGURA 4.18 INTERFAZ DE USUARIO DE KYLIX 3.0. ....	108
FIGURA 4.19 EL INSPECTOR DE OBJETOS Y OBJECT TREE VIEW. ....	108
FIGURA 4.20 INFRAESTRUCTURA DE LA RED SOBRE LA QUE SE MONTA WINPS. ....	112
FIGURA 4.21 DIAGRAMA DE FLUJO DE DATOS NIVEL 1 DE WINPS. ....	113
FIGURA 4.22 DIAGRAMA DE FLUJO DE DATOS DE WINPS. ....	114
FIGURA 4.23 DIAGRAMA DE FLUJO DE DATOS NIVEL 2 DE WINPS. ....	115
FIGURA 4.24 INTERFAZ DEL LAUNCHER. ....	116
FIGURA 4.25 MENÚ QUE CONFORMA AL LAUNCHER. ....	116
FIGURA 4.26 INTERFAZ DEL MÓDULO MONITOREO. ....	117
FIGURA 4.27 INTERFAZ DEL MÓDULO DE GRAFICACIÓN. ....	118
FIGURA 4.28 INTERFAZ DEL MÓDULO DE MAPEO. ....	119
FIGURA 4.29 INTERFAZ DEL MÓDULO DE LOCALIZACIÓN. ....	120
FIGURA 4.30 INTERFAZ DEL WORKSPACE DEL MÓDULO DE LOCALIZACIÓN. ....	120
FIGURA 4.31 INTERFAZ DEL MÓDULO DE AYUDA. ....	121
FIGURA 4.32 DEPENDENCIA FUNCIONAL DE LOS MÓDULOS DE WINPS. ....	122
FIGURA 4.33 DIAGRAMA GENERAL DE CASOS DE USO PARA WINPS. ....	123
FIGURA 4.34 DIAGRAMA DE CASOS DE USO PARA MONITOREO_SCANNING. ....	123
FIGURA 4.35 DIAGRAMA DE CASO DE USO PARA MONITOREO_GRAFICA. ....	126
FIGURA 4.36 DIAGRAMA DE CASOS DE USO PARA MAPPING. ....	127
FIGURA 4.37 DIAGRAMA DE CASOS DE USO PARA MAPPING_LOCALIZACIÓN. ....	128
FIGURA 4.38 DIAGRAMA DE CASOS DE USO PARA AYUDA. ....	130
FIGURA 4.39 DIAGRAMA DE FLUJO PARA EL MÓDULO LAUNCHER. ....	131
FIGURA 4.40 DIAGRAMA DE FLUJO DE LA FUNCIÓN BUSCARSTRING. ....	132
FIGURA 4.41 DIAGRAMA DE FLUJO DEL MÓDULO DE MONITOREO. ....	134
FIGURA 4.42 DIAGRAMA DE FLUJO DEL MÓDULO DE GRAFICACIÓN. ....	135
FIGURA 4.43 DIAGRAMA DE FLUJO DEL SUBMÓDULO 1 PARA EL MÓDULO DE GRAFICACIÓN. ....	136
FIGURA 4.44 DIAGRAMA DE FLUJO DEL MÓDULO DE MAPEO. ....	137
FIGURA 4.45 DIAGRAMAS DE FLUJO DE LOS SUBMÓDULOS 1 Y 2 DE LA HERRAMIENTA DE MAPEO. ....	138
FIGURA 4.46 TRIANGULACIÓN DEL DEPARTAMENTO DE TELECOMUNICACIONES DE LA DIE. ....	139
FIGURA 4.47 MAPPING PARA AP2. ....	140
FIGURA 4.48 DIAGRAMA DE FLUJO DEL MÓDULO DE LOCALIZACIÓN. ....	141
FIGURA 4.49 DIAGRAMA DE FLUJO DE LA FUNCIÓN DE CAPTURASENAL. ....	142
FIGURA 4.50 DIAGRAMA DE FLUJO DE LA FUNCIÓN DE NORMALIZACIÓN. ....	143
FIGURA 4.51 DIAGRAMA DE FLUJO DEL ALGORITMO DE LOCALIZACIÓN. ....	144
FIGURA 4.52 DIAGRAMA DE FLUJO DE LA FUNCIÓN DE VERIFICA_SL. ....	145
FIGURA 4.53 EJEMPLO DE LA PROPIEDAD GEOMÉTRICA DEL VALOR PROMEDIO DE COORDENADAS. ....	146
FIGURA 4.54 REGIONES DEL DEPARTAMENTO DE TELECOMUNICACIONES DE LA DIE. ....	147
FIGURA 4.55 DIAGRAMA DEL MÓDULO VERIFICA_REGION. ....	147
FIGURA 4.56 EJEMPLO DE CÓMO SE DEFINEN LOS LIMITES DE LAS REGIONES. ....	148
FIGURA 4.57 DIAGRAMA DE FLUJO DE LA FUNCIÓN CONVIERTEXY. ....	148
FIGURA 4.58 UBICACIÓN DEL SISTEMA CARTESIANO. ....	149
FIGURA 4.59A RESULTADO DE LA COMPILACIÓN DE WINPS. ....	150
FIGURA 4.59B CARGA DE WINPS EN LA MEMORIA DE LA PC. ....	150
FIGURA 4.60 ARCHIVO EJECUTABLE DE WINPS CONSTRUIDO. ....	151
FIGURA 4.61 CARGA DE LA APLICACIÓN DE GRAFICACIÓN. ....	151
FIGURA 4.62A CARGA DEL MAPA PARA LA APLICACIÓN DE LOCALIZACIÓN. ....	152
FIGURA 4.62B CARGA DE LA APLICACIÓN DE LOCALIZACIÓN. ....	152
FIGURA 4.63 CARGA DE LA APLICACIÓN DE MAPEO. ....	153

FIGURA 4.64 CARGA DE LA APLICACIÓN DE MONITOREO. ....	153
FIGURA 4.65 CARGA DE LA APLICACIÓN DE AYUDA. ....	154
FIGURA 5.1 IMAGEN DEL LAUNCHER DURANTE LA EJECUCIÓN DE WINPS. ....	156
FIGURA 5.2 IMAGEN DE LA HERRAMIENTA DE GRAFICACIÓN DURANTE LA EJECUCIÓN DE WINPS. ....	157
FIGURA 5.3 IMAGEN DE LA HERRAMIENTA DE MONITOREO DURANTE LA EJECUCIÓN DE WINPS. ....	159
FIGURA 5.4 IMAGEN DE LA HERRAMIENTA DE MAPPING DURANTE LA EJECUCIÓN DE WINPS. ....	160
FIGURA 5.5 IMAGEN DE LA HERRAMIENTA DE MAPPING DURANTE LA EJECUCIÓN DE WINPS. ....	161
FIGURA 5.6 ARCHIVOS BASE PARA LA EJECUCIÓN DEL SISTEMA DE POSICIONAMIENTO. ....	162
FIGURA 5.7 PETICIÓN DEL DIRECTORIO EN DONDE SE ALOJA EL MAPA DEL WORKSPACE. ....	162
FIGURA 5.8 GUI DEL SISTEMA DE LOCALIZACIÓN EN TIEMPO DE EJECUCIÓN. ....	163
FIGURA 5.9 GUI DEL SISTEMA DE LOCALIZACIÓN CON EL PRIMER OBJETO HALLADO. ....	164
FIGURA 5.10 GUI DEL SISTEMA DE LOCALIZACIÓN CON EL SEGUNDO OBJETO HALLADO. ....	165
FIGURA 5.11 GUI DEL SISTEMA DE LOCALIZACIÓN CON EL TERCER OBJETO HALLADO. ....	166
FIGURA 6.1 EVOLUCIÓN DE WINPS. ....	168
FIGURA 6.2 DISTRIBUCIÓN DE LA INTENSIDAD DE SEÑAL NORMALIZADA. ....	171
FIGURA 6.3 DISTRIBUCIÓN TRIDIMENSIONAL DE LA INTENSIDAD DE SEÑAL NORMALIZADA. ....	172
FIGURA A.1 FUNCIONAMIENTO DE NDIS WRAPPER. ....	173
FIGURA C.1 MEDICIÓN DE LA INTENSIDAD DE SEÑAL EN UN MISMO PUNTO (DÍA 1). ....	176
FIGURA C.2 MEDICIÓN DE LA INTENSIDAD DE SEÑAL EN UN MISMO PUNTO (DÍA 2). ....	176
FIGURA C.3 SUPERPOSICIÓN DE LAS DOS MEDICIONES. ....	177

## ÍNDICE DE TABLAS

TABLA 1.1. TÉCNICAS DE LOCALIZACIÓN MEDIANTE REDES DE TELEFONÍA MÓVIL Y SATELITAL. ....	15
TABLA 1.2 ESTÁNDARES PARA WIRELESS LAN .....	30
TABLA 1.3 COMPARACIÓN DE TECNOLOGÍAS. ....	31
TABLA 2.1 VENTAJAS Y DESVENTAJAS DE LAS REDES AD-HOC. ....	48
TABLA 2.2 VENTAJAS Y DESVENTAJAS DE LAS REDES DE INFRAESTRUCTURA .....	49
TABLA 2.3 COMPARACIÓN DE MEDIO VS TASA DE TX. ....	55
TABLA 2.4 COMPARACIÓN DE MEDIO VS TASA DE TX. ....	55
TABLA 2.5 PRINCIPALES CARACTERÍSTICAS DEL ESTÁNDAR 802.11x.....	57
TABLA 2.6 CAPA FÍSICA DEL ESTÁNDAR 802.11. ....	58
TABLA 2.7 MODULACIÓN Y CODIFICACIÓN DE ACUERDO AL ESTÁNDAR 802.11 .....	58
TABLA 4.1 ESTRUCTURA BÁSICA DEL SISTEMA DE ARCHIVOS EN UNIX. ....	92
TABLA 4.2 PROPIEDADES QUE PUEDEN SER OBTENIDAS MEDIANTE KWAVECONTROL .....	103
TABLA 4.3 CASO DE USO BUSCARSTRING. ....	124
TABLA 4.4 CASO DE USO DE BUSCAR INFORMACIÓN. ....	125
TABLA 4.5 CASO DE USO DE DESPLIEGA INFORMACIÓN. ....	125
TABLA 4.6 CASO DE USO DE GRAFICAR DATOS XY. ....	126
TABLA 4.7 CASO DE USO DE MAPPING.....	128
TABLA 4.8 CASO DE USO DE MAPPING_LOCALIZACIÓN. ....	129
TABLA 4.9 CASO DE USO DE BUSCAR TEMA. ....	130
TABLA 4.10 CASO DE USO DE DESPLIEGA INFORMACIÓN PARA LA AYUDA. ....	130
TABLA 5.1 INFORMACIÓN GENERAL DE LA PRUEBA PARA EL LAUNCHER. ....	156
TABLA 5.2 INFORMACIÓN GENERAL DE LA PRUEBA PARA EL MÓDULO DE GRAFICACIÓN.....	158
TABLA 5.3 INFORMACIÓN GENERAL DE LA PRUEBA PARA EL MÓDULO DE MONITOREO. ....	159
TABLA 5.4 INFORMACIÓN GENERAL DE LA PRUEBA PARA EL MÓDULO DE MAPPING. ....	160
TABLA 5.5 INFORMACIÓN GENERAL DE LA PRUEBA PARA EL MÓDULO DE AYUDA. ....	161
TABLA 5.6 INFORMACIÓN GENERAL DE LA PRUEBA PARA EL MÓDULO DE LOCALIZACIÓN .....	166
TABLA 6.1 VARIACIONES DE INFORMACIÓN GENERAL DE LA PRUEBA PARA EL MÓDULO DE MAPPING. ....	170

## INTRODUCCIÓN

---

El presente trabajo tiene como objetivo fundamental conocer el proceso de diseño e implementación de un sistema de localización en redes inalámbricas Wi-Fi (*Wireless Fidelity* por sus siglas en inglés) y Ad-Hoc. Esto se debe a que tipo de infraestructura de interconectividad ha cobrado mucha fuerza en los últimos años, y en consecuencia, está presente en la mayoría de los sitios públicos de diversas ciudades de México.

Más adelante conoceremos el método de mapeo o *mapping*, el cual, es un modo de localización bastante preciso, no obstante, públicamente no hay un algoritmo que especifique su funcionamiento, todas las descripciones son muy superficiales. Así pues, lo que aparece en el presente trabajo es:

- ✦ *Nuestra propuesta de un algoritmo de localización basado en el mapping, y*
- ✦ *La implementación del mismo, dentro de un sistema de cómputo.*

Esta decisión no fue sencilla, pues había que tomar en cuenta diversos factores, entre ellos, está el estudio sobre los diversos métodos de localización que existen en la actualidad; además se realizó una investigación sobre los principales elementos que conforman el estándar de comunicación del 802.11x.

Así como un análisis de la infraestructura de las condiciones, a lo que hemos llamado nuestro *workspace* o área de trabajo, donde está situada nuestra *Wireless Local Area Network (WLAN)*; y por último había que observar el comportamiento de los factores de comunicación en la red.

Por lo anterior, la estructura que seguimos para el desarrollo de este trabajo de tesis es el siguiente:

- ✦ *Capítulo I.* En este capítulo, se muestran los sistemas de localización que existen en la actualidad, se definen los elementos que lo conforman y se explica el método que se sigue para la localización. Para que el estudio fuese lo más general posible, se acordó dividirlo en tres modelos, el primero de ellos es el referente a las tecnologías basadas en la identidad celular, el segundo es la localización mediante redes inalámbricas y por último el llamado *Global Positioning System (GPS)*.
- ✦ *Capítulo II.* Para este capítulo, se mencionan los elementos más sobresalientes de las redes inalámbricas locales. Pues a partir de estos conceptos, así como toda la tecnología que se involucra alrededor de la capa de hardware, podremos entender la vinculación existente entre la terminal y el *Access Point (AP)*, la cual es donde se centra la relación entre el *workspace - sistema de localización - usuario*.
- ✦ *Capítulo III.* Aquí es donde aterrizamos las ideas de los capítulos I y II y se propone una solución para el diseño e implementación del sistema de localización, tomando en cuenta los elementos de hardware, software, así como los elementos de transmisión en la red inalámbrica.

- ✦ *Capítulo IV.* Este capítulo presenta el desarrollo que se hizo para la implementación del sistema de localización, primero se analizan los elementos externos del *workspace* (plano de la edificación y mediciones de niveles en factores de transmisión), luego se analizan las características del sistema de cómputo y por último se procede a la construcción del software, lo cual implica la documentación básica, así como un proceso de pruebas preliminares.
- ✦ *Capítulo V.* Este capítulo es consecuencia del cuarto, pues solo se realizan algunas pruebas más sobre la funcionalidad así como la corroboración de los resultados de localización.
- ✦ *Capítulo VI.* En este se menciona el alcance que tiene el proyecto, se vierten las experiencias y se resaltan algunos de los elementos más importantes que se encontraron durante el desarrollo del presente trabajo.
- ✦ *Apéndices.* Estos son algunos compendios breves que fueron de utilidad en el momento del desarrollo del sistema. Se decidieron incluir por la falta de información que existe sobre el tema, así como de la disponibilidad en nuestro idioma.
- ✦ *Acrónimos y Abreviaturas.* A lo largo del trabajo, muchas son las siglas o pequeños conceptos que son importantes, para no perder continuidad en la estructura que se siguió, estas abreviaturas y acrónimos fueron ubicadas al final del presente escrito.
- ✦ *Bibliografía y referencias.* Son los vínculos, libros y otros artículos que nos fueron de gran utilidad en el desarrollo de este escrito.



## OBJETIVOS

---

De acuerdo a lo mencionado en la introducción, el presente trabajo tiene como objetivos fundamentales:

- ✦ *Conocer el proceso de diseño e implementación de un sistema de localización en redes inalámbricas Wi-Fi y Ad- Hoc.*
- ✦ *Mostrar nuestro algoritmo de localización basado en el principio de mapeo o mapping.*
- ✦ *Dar a conocer los principales detalles referentes a las diversas tecnologías de localización que se tienen en la actualidad.*
- ✦ *Implementar el sistema de localización creando una herramienta de mapeo y monitoreo para así dar lugar a lo que hemos definido como sistema de posicionamiento en una red inalámbrica (WiNPS - Wireless Network Position System).*
- ✦ *Definir un funcionamiento independiente al servidor de la red, a modo de que WiNPS funcione sobre las intensidades de señal captadas en la terminal de los puntos de acceso.*

## CAPÍTULO I

### TECNOLOGÍAS DE LOCALIZACIÓN

---

#### 1.1 Introducción.

Diversas circunstancias han impulsado el desarrollo de los sistemas de posicionamiento alrededor del mundo. En el caso de Estados Unidos estas tecnologías cobraron especial interés a raíz de un mandato legislativo promulgado por la *Federal Communications Commission (FCC)*, el cual mencionaba que a partir de diciembre de 2005 las operadoras de telefonía tendrían que ser capaces de localizar automáticamente a cualquier persona que efectuara una llamada de emergencia con una precisión de 50 a 100 metros.

En la actualidad los operadores de telefonía móvil han introducido servicios basados en el posicionamiento, esto con la finalidad de diferenciarse de sus competidores, incrementar sus ingresos y reducir costos. En España, la penetración de la telefonía móvil representa el 80.1% de la población<sup>1</sup>, lo que indica que la saturación del mercado es inminente.

En este marco, donde la diferenciación por servicios es clave, las aplicaciones de localización generarán una gran parte de los ingresos de las operadoras. Es por ello, que no resulta difícil de entender el motivo por el cual los operadores y fabricantes de los servicios y dispositivos de telecomunicaciones, han tenido un seguimiento constante a todas las tecnologías de localización.

Para ofrecer estos servicios existen en la actualidad varias técnicas que permiten conocer la posición de una terminal móvil. La precisión, el costo y la dificultad de implementación son parámetros que el prestador de servicios valora previamente antes de decidirse por una u otra opción. La posibilidad de localización de una terminal móvil ya ha dado lugar a numerosos servicios de información, rastreo, selección de rutas y gestión de recursos.

Una de las aplicaciones más importantes de un sistema de localización, es el sistema *Wireless Enhanced 911 (E911)*; el cual enruta la llamada de emergencia al centro de asistencia más próximo; que a diferencia del servicio básico, despliega automáticamente en una pantalla, la dirección y número telefónico de la persona que solicita el servicio, sin presentar retardos que podrían hacer la diferencia entre una asistencia exitosa y una que no lo sea. Otros países tienen sistemas similares, pero utilizan otra combinación de números; por ejemplo China utiliza el 110, Inglaterra el 999 y Francia el 17 por citar algunos.

Otro ejemplo claro de esta evolución se puede observar en la telefonía celular; ya que inicialmente, esta solo fue creada para realizar llamadas; actualmente un teléfono celular puede hacer más que eso; presenta funciones de otros dispositivos, como los *Personal Digital Assistants (PDAs)*, laptops y cámaras. El número de servicios disponibles y su complejidad se incrementa día con día; al igual que su uso entre la población. Es así como sur-

---

<sup>1</sup> Dato proporcionado por la CMT 2002 (Conferencia Mundial de Telecomunicaciones realizada en el 2002).

ge la idea de un servicio que pueda proporcionar la ubicación aproximada de cualquiera de los usuarios.

Así pues, de manera global, encontramos que existen cuatro tipos de servicios de localización móvil, estos son:

- ✦ *Servicios por activación automática (Trigger Services)*: Este tipo de servicio se inicia cuando el usuario entra en un área determinada, su aplicación más adecuada es en aplicaciones publicitarias o de facturación.
- ✦ *Servicios de información basados en la posición (Location-based Information Services)*: En este caso, el usuario del servicio demanda información de algún tipo, que varía según su posición. Las operadoras de telefonía móvil son las que ofertan este tipo de servicios, los cuales, en su mayoría permiten encontrar establecimientos cercanos al demandante de información.
- ✦ *Servicios de seguimiento por terceros (Third Part Tracking Services)*: Estos contemplan tanto aplicaciones corporativas como de consumidor, donde la información de la localización es requerida por un tercero. Se pueden utilizar para gestión de flotas, búsqueda de personas, información bursátil, asesoramiento rápido, etc.
- ✦ *Servicios de asistencia al usuario final (End User Assistance Services)*: Están diseñados para proveer al usuario de condiciones de red segura, si este se encuentra en dificultades, además de servicios de asistencia en carretera u otros servicios de emergencia están dentro de este grupo.

Hablando un poco más sobre el proceso de localización, nos encontramos que debe llevarse a cabo independientemente que la terminal móvil se encuentre en el exterior o interior de un edificio. Las técnicas que se emplean para este propósito son muy diversas y dependen en gran medida de la precisión con la que se deba encontrar al usuario.

Por este motivo, en este primer capítulo se comentarán las técnicas de localización más empleadas para espacios abiertos, y en segundo lugar las basadas en redes inalámbricas, enfocadas a la oferta de servicios en interiores. De cualquier forma, tanto unas como otras son capaces de encontrar una terminal móvil si la infraestructura es la adecuada, aunque la precisión con que lo harán variará sensiblemente dependiendo del entorno.

Cabe mencionar, que actualmente muchas de las aplicaciones al transporte, pueden ser soportadas por sistemas de navegación que utilizan las redes de comunicación, sistemas de localización centralizados y otras infraestructuras que, junto con el equipo que se encuentra a bordo del vehículo; realizan la localización y navegación del mismo.

Para estas aplicaciones, se incluyen dos tipos de sistemas, esto es, existe un sistema de telemática automotriz que generalmente realiza la comunicación a través de una red de telecomunicaciones, también se utiliza un sistema de tránsito el cual es la aplicación de un sistema automático de localización de vehículos.

Ahora bien, volviendo a las tecnologías de localización, tenemos que las más utilizadas pueden dividirse en tres grupos:

- ✦ *Stand – alone.*
- ✦ *Satellite – Based.*
- ✦ *Terrestrial radio – base.*

Una aplicación de la primera categoría es “*Dead reckoning*”, también conocido como “Rumbo estimado”, esta ha sido utilizada durante mucho tiempo para obtener la posición estimada a partir de la velocidad, tiempo y dirección proveniente de un punto conocido; posee aplicaciones en la aeronáutica realizando el envío periódico de reportes de la posición a un sitio remoto por medio de un enlace de radio; monitoreo de prisioneros y pacientes a través de un podómetro; además de seguridad y logística, por medio del rastreo de los usuarios dentro de un medio urbano; y aplicaciones en la navegación, cuando se presentan condiciones climáticas no favorables; donde GPS presenta problemas de operación.

Los datos recopilados de “*Dead reckoning*”, pueden ser procesados con el fin de obtener información secundaria de los rastreos; la cual puede ser el trazado de un túnel, la posición y orientación de las puertas que rodean una construcción; inclusive realizar mapas de las zonas de acceso a un complejo.

Una aplicación típica de la segunda categoría, es GPS; el cual se discutirá a profundidad más adelante.

La tercera categoría contempla a E911, redes de telefonía celular, sistemas de comunicación terrestre y de navegación, etc; es por ello que esta utiliza las señales de la red para ayudar a determinar la localización de la unidad móvil.

En general, la mayoría de los métodos de localización que utilizan esta tecnología, requieren de radio transmisores, receptores y/o transeceptores. En otras palabras, todos ellos dependen de la emisión y recepción de señales de radio, para determinar la localización de un objeto que contenga al radio receptor o al transeceptor. La ubicación puede realizarse en cualquier extremo del sistema, asumiendo que uno de los extremos del sistema es fijo, mientras que el otro es móvil. Generalmente es decisión del diseñador del sistema la residencia del extremo capaz de realizar la ubicación.

Existe también una tecnología de localización denominada *terrestrial radio-based*, la cual utiliza, dispositivos que emiten señales de *Radio Frequency (RF)*, las cuales tienen como destino, receptores móviles, para así obtener, por ende, la posición del usuario; en este caso se dice que la tecnología es basada *en la red*. Sin embargo, las señales pueden ser también emitidas desde la terminal móvil a la base, siendo esta, el dispositivo receptor; en este caso la tecnología se dice que se basa *en la modificación del sistema móvil o en el handset*; inclusive el transmisor puede difundir su identidad, con el fin de que todos los radio receptores la conozcan y así se pueda determinar su ubicación aproximada; esta variante de la tecnología se conoce como basada en la *identidad célula*. No obstante con el fin de mejorar los sistemas, se han implementado métodos híbridos, es decir, se realiza la combinación de diferentes técnicas en una sola.

De lo anteriormente mencionado, se puede decir que la tecnología *terrestrial radio-based* se puede subdividir en tres categorías:

- ◆ Tecnologías basadas en la identidad celular.
- ◆ Técnicas basadas en la red.
- ◆ Técnicas basadas en la modificación del sistema móvil.

En este capítulo, nosotros analizamos con más detalle dichas categorías. Se tomó la decisión de incluirlas ya que en el sistema que se desarrollará, está basado en algunos de

los conceptos de esta tecnología de localización, aunado a ello, tenemos que la naturaleza del ambiente y las herramientas, propician el uso de esta tecnología.

## **1.2 Localización mediante redes móviles y satelitales.**

En esta sección se pretende describir aquellos métodos basados en la tecnología para redes móviles; en nuestra investigación, encontramos que algunos de ellos se pueden implementar directamente, otros necesitan que se efectúen modificaciones en la red, mientras que el resto requieren de una ampliación en las funcionalidades en la terminal del usuario.

Es por todos sabido que la información sobre localización siempre ha estado presente en redes celulares *Global System for Mobile Communications (GSM)*, ya que es necesaria para establecer y mantener la comunicación, no obstante, esto inicialmente no era accesible fuera de los nodos de red; en la actualidad se han introducido nuevos elementos que son necesarios para poder corregir estas fallas. Asimismo, en *Universal Mobile Telecommunications System (UMTS)* este sistema es ya un elemento integrante de la red, también podemos decir que el núcleo de red incluye todo lo necesario para que se pueda conocer la posición tanto interna como externa.

Las operadoras de telefonía móvil suelen utilizar una combinación o variación de uno o más sistemas, dependiendo de la aplicación que se prevea ofertar. De esta manera, las técnicas de localización más comunes se pueden agrupar, como se muestra en la siguiente tabla.

<ul style="list-style-type: none"><li>⊕ <b>Técnicas basadas en la identidad celular:</b><ul style="list-style-type: none"><li>⊕ <i>Identidad Celular Global (Cell Global Identity, CGI).</i></li><li>⊕ <i>Identidad Celular Perfeccionada (Enhanced Cell ID).</i></li></ul></li><li>⊕ <b>Técnicas basadas en la red:</b><ul style="list-style-type: none"><li>⊕ <i>Ángulo de llegada (Angle of arrival, AOA).</i></li><li>⊕ <i>Tiempo de llegada (Time of Arrival, TOA).</i></li><li>⊕ <i>Diferencia en el tiempo de llegada (Time Difference of Arrival, TDOA).</i></li><li>⊕ <i>Huella multitrayecto (Multipath Fingerprint, MF).</i></li></ul></li><li>⊕ <b>Técnicas basadas en la modificación de la terminal móvil:</b><ul style="list-style-type: none"><li>⊕ <i>Diferencia en el tiempo de llegada perfeccionada (Enhanced Observed Time Difference, E-OTD o TDOA con terminal modificado).</i></li><li>⊕ <i>Triangulación avanzada de enlace hacia delante (Advanced Forward Link Trilateration, A-FLT).</i></li><li>⊕ <i>Sistema de Posicionamiento Global (Global Positioning System, GPS).</i></li></ul></li></ul>
-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

*Tabla 1.1. Técnicas de localización mediante redes de telefonía móvil y satelital.*

### 1.2.1 Tecnologías basadas en la identidad celular.

Las tecnologías de localización basadas en la identidad celular, pueden ser implementadas sin necesidad de realizar alguna inversión, modificación en la red o terminal; debido a que las estaciones base difunden su identidad a todos los dispositivos móviles que se encuentran en la célula y de esta manera pueden ser localizados. De acuerdo al tipo de tecnología, se pueden llegar a utilizar otro tipo de parámetros que ayuden a mejorar la precisión al momento de ubicar la terminal móvil.

#### 1.2.1.1 *Cell ID*.

Esta es una tecnología de posicionamiento basada en la red, sin embargo, dado que la mayor parte del método se basa en la información de la célula, se puede clasificar como un método basado en la identidad celular.

Para poder obtener la localización aproximada de los usuarios, utiliza la información del ID de su célula, también conocido como *Cell ID*. Esta información se obtiene de las estaciones base y se distribuye a las terminales móviles a través de la tecnología de red existente que se encuentre dentro del área celular.

En las redes celulares actuales, la cobertura está provista por una serie de estaciones base distribuidas que delimitan las unidades de área geográfica, llamadas células; cada una, se divide en tres sectores y su tamaño puede variar de 1 a 3 [km] en áreas urbanas y aumenta de 3 a 20 [km], en áreas rurales o suburbanas; sin embargo, el sector de la célula donde se localiza el usuario solo se conocerá durante una llamada activa, ya sea de voz o de datos. Con este método, no se necesita una interfaz de aire ni modificaciones de hardware o software en la unidad móvil, para obtener la localización del usuario.

No obstante, la principal desventaja de este método reside en que la precisión, esto es, esta técnica presenta muy baja exactitud en comparación con otras tecnologías; ya que la ubicación del usuario será más aproximada, si el sector de la célula es más pequeño; en caso contrario el error será mayor, es decir, depende del área del sector de la célula en la que se encuentre el usuario.

La falta de precisión en este método, propició el rechazo de la FCC, por no cumplir con los estándares mínimos. Sin embargo, este problema puede ser solventado fácilmente, solo se añade un parámetro denominado *Timing Advance (TA)*; convirtiéndose entonces en una técnica perfeccionada de identidad celular denominada como *Enhanced Cell-ID*.

#### 1.2.1.2 *Cell Global Identity*.

Esta técnica de localización puede ser utilizada sin realizar ninguna inversión ni modificación en la red o terminal, pues la posición se obtiene mediante la identidad de la celda en la que se encuentra la terminal móvil; sin embargo, a diferencia de Cell ID, esta técnica resulta ser la unión de Cell ID y *Location Area Identity (LAI)*; donde el segundo se refiere al parámetro que identifica de manera única a un área dentro de cualquier red pú-

blica móvil, también denominada *Public Land Mobile Network (PLMN)*, ya sea nacional o internacional.

La localización incrementa su precisión gracias a ciertos parámetros que se introducen al sistema por parte de LAI; dentro de éstos se puede mencionar el *Mobile Country Code (MCC)*, el cual es un código conformado de tres dígitos que identifica a cada país, el *Mobile Network Code (MNC)* que es un código de igual longitud que el anterior, encargado de identificar la red que está siendo utilizada; por último se incluye *Location Area Code (LAC)*, el cual determina el área en la que se encuentra el dispositivo móvil.

La precisión de este método; al igual que en *Cell ID*, depende estrictamente del radio de la célula, sin embargo debido a los parámetros considerados, puede aumentar, ya que en áreas urbanas puede variar hasta cientos de metros, en áreas urbanas y de 3 ó 4 [km] en áreas rurales. Con esto se logra una disminución significativa con respecto a su antecesor.

Éste es el sistema de localización más utilizado por los operadores, pues es suficiente para ofrecer al usuario cierto tipo de servicios en entornos urbanos, con penetración en el mercado inmediata del 100%. Su función es ubicar todo tipo de dispositivos móviles en redes *General Packet Radio Service (GSM)*, GPRS, UMTS y *Code Division Multiple Access (CDMA)*.

Cuando las redes de tercera generación se encuentren ya en operación, es muy probable que se empleen otros mecanismos más precisos como los que se enunciarán más adelante, en este capítulo.

#### *1.2.1.3 Enhanced Cell ID.*

Es una tecnología basada en la identidad multicelular que combina el principio de operación de *Cell ID* incluyendo un parámetro denominado, TA.

El parámetro añadido al sistema, está en función del retardo de acceso entre el principio de un intervalo de tiempo y la llegada de las ráfagas en la terminal móvil. Por medio de este se puede obtener de manera indirecta, el valor de la distancia de la unidad móvil con respecto a su estación base; a parte de la célula activa, pueden utilizarse otras más e inclusive, tomar en cuenta la potencia de las señales que difunden, con el fin de mejorar la precisión del sistema.

La potencia puede ser obtenida por el método *Received Signal Level (RSL)* el cual es el encargado de medir el nivel promedio de la intensidad de la señal recibida.

El rango de precisión momento de localizar a la unidad móvil es aún mayor que CGI y dado que utiliza *Multi Cell ID*; ahora la unidad móvil a localizar, se encuentra en un rango poco mayor a las decenas de metros.

### 1.2.2 Técnicas basadas en la red.

En los últimos años, la FCC ha aumentado la demanda de operadores que proporcionen información posicional a los servicios de emergencia; para aumentar el interés, ha provisto de estímulos a los desarrolladores de las aplicaciones basadas en tecnologías de posicionamiento inalámbrico.

Hay varias tecnologías alternativas para localizar una unidad móvil, como lo es un teléfono celular. GPS es una de las opciones más solicitadas, sin embargo, otras tecnologías pueden realizar la ubicación sin necesitar de una unidad muy especializada; ya que éstas en general, solo requieren una red inalámbrica y puntos que sirvan de referencia (estaciones base) para realizar la ubicación de la estación móvil.

Así pues, encontramos que existen tres métodos principales para determinar la localización de una unidad móvil remota dentro de un sistema de comunicaciones:

- ⊕ *Direction of Arrival (DOA).*
- ⊕ *Time of Arrival (TOA).*
- ⊕ *Time Diference of Arrival (TDOA)*

En las secciones subsecuentes, nosotros describiremos de manera breve cada uno de estos métodos, y en su momento iremos mencionando las principales ventajas y desventajas que presentan.

#### 1.2.2.1 *Direction of Arrival (DOA).*

También denominado AOA, fue el primer sistema de localización desarrollado por organizaciones militares y del gobierno; determina la posición de la unidad móvil basándose en la medición del ángulo con el que llegan las señales con respecto a cada una de las estaciones base o *Position Determining Entity (PDE)*, es decir, por medio de la intersección de dos o más líneas radiales de apoyo que definen una posición única en un plano bidimensional o tridimensional; en la figura 1.1 se muestra la tecnología AOA bidimensional.

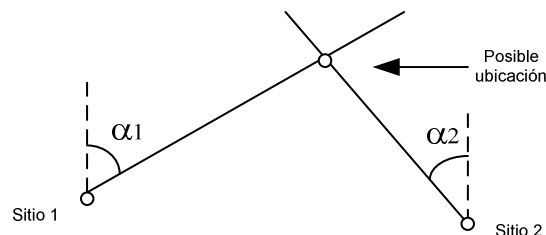


Figura 1.1 AOA con 2 estaciones base.

Las ventajas principales de esta técnica es que no modifica las características del dispositivo móvil, solo requiere de un par de estaciones base como mínimo para determinar la ubicación, sin embargo, en caso de contar con más de un par, la precisión aumenta, debido a que la información de las diferentes estaciones puede ser triangulada para determinar la posición. Como se puede ver, este método es independiente de las referencias temporales y de los sistemas de sincronía, por lo que lo hace un sistema fácil, pero caro de implementar.



En la figura 1.2, se observa la implementación de AOA en el plano tridimensional.

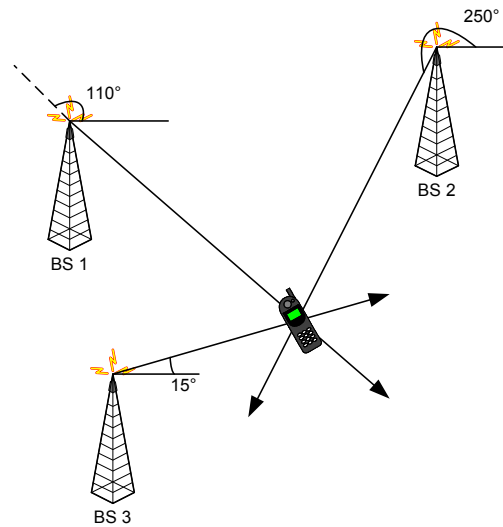


Figura 1.2 AOA con 3 estaciones base.

Cabe mencionar, que este sistema presenta disponibilidad muy alta, independientemente de las condiciones bajo las cuales se opere, lo que lo hace una buena opción, si la precisión máxima no es fundamental.

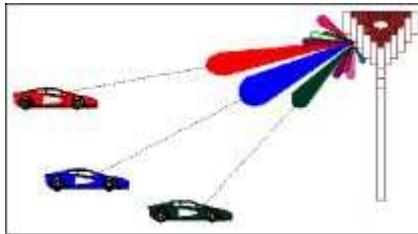
Algunas de sus principales desventajas radican en el uso necesario de numerosas antenas de gran tamaño por célula, de cuatro a doce unidades, ya sean direccionales o de arreglos en fase; este factor incrementa el costo de mantenimiento del sistema a largo plazo. Sin embargo, en caso de no contar con *Line Of Sight (LOS)*, los arreglos de antenas bloquearán las señales que sean reflejadas de otros dispositivos que no sean el que se encuentra bajo localización.

El hecho de contar con una gran cantidad de antenas necesarias, dificulta su implementación en áreas donde se requieran muchas células debido al gran número de usuarios; además de que un pequeño error en la medición del ángulo, se traduce en un error considerable, cuando la estación móvil se encuentra alejada con respecto a la estación base; tal y como sucede en las zonas rurales, lo que provoca que la precisión del método sea altamente cuestionable. Es por ello, que este método debe implementarse en células de tamaño medio o pequeño.

Una posible solución, en el caso de las áreas rurales, es la ubicación de la estación base en un lugar elevado con respecto al resto del terreno; los efectos de los objetos que provocan la reflexión de las ondas que se encuentran en las cercanías de la unidad móvil, podrán ser atenuados; a consecuencia, las señales llegarán en una forma de un rayo relativamente estrecho a la estación base. Incluso podría complementarse con otros sistemas, como TDOA, para mejorar la precisión del sistema.

Las señales recibidas en cada estación base pueden ser caracterizadas electrónicamente por medio de las diferencias en amplitud o fase captadas por un sistema de antenas de arreglos en fase adaptables a la dirección de la señal proveniente del móvil; para después relacionarlas con el ángulo con el que es captada la señal.

Este sistema adaptable consiste en un arreglo de antenas en fase con elementos de censado y un procesador de señales en tiempo real, el cual ajusta automáticamente el patrón de radiación de las antenas; con la posibilidad de formar múltiples lóbulos para rastrear el ángulo de llegada de más de una estación móvil; la representación de efecto se muestra en la figura 1.3; la respuesta en frecuencia y algunos otros parámetros que permitan mejorar el rendimiento del sistema; combatiendo las imprecisiones introducidas por los efectos de propagación en multitrayectorias; por lo que podría ser implementado en sistemas de espectro disperso.



*Figura 1.3 Representación del arreglo de antenas.*

En caso de que las señales sean captadas por antenas simplemente direccionales, el efecto de multitrayectorias no podría ser atenuado y deberá ser traducido como un error en la exactitud del sistema.

#### *1.2.2.2 Time of Arrival (TOA).*

Esta técnica está basada en la medición del tiempo de llegada de una señal transmitida por una terminal móvil a diferentes estaciones base a través de una *Location Measurement Unit (LMU)*, también denominado como servidor de localización, que será el encargado de realizar la recopilación y análisis de los datos recibidos.

Todas las estaciones deben estar sincronizadas, ya que la unidad encargada de obtener las mediciones se comporta como un transceptor capaz de procesar retardos con gran precisión, esto lo realiza mediante el uso de métodos matemáticos; con esto se eliminan una gran parte de los errores provocados por la ausencia de LOS; estos errores son los generados por la señal al ser dispersada por los objetos que se presentan en su trayectoria. Todo esto se lleva a cabo pues un microsegundo de error puede provocar un desfase de 300[m] en la localización del equipo.

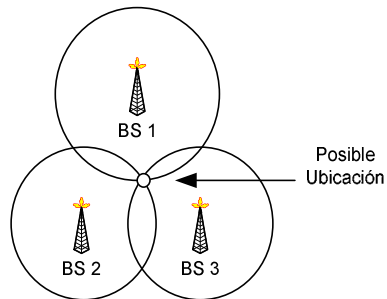
Vale la pena mencionar que existen otro tipo de errores que no pueden ser eliminados totalmente, estos son los relacionados al tiempo de procesado en la terminal, este tipo de fallas dependen del fabricante y también de la situación de carga del dispositivo en un momento determinado.

Ahora bien, volviendo a la explicación del método, tenemos que utilizando una combinación de los tiempos de retardo y conociendo la localización exacta de cada estación base, se puede obtener la posición del móvil por triangulación<sup>2</sup>, la cual se puede calcular a partir del tiempo y velocidad de propagación que presentan las ondas de radio desde el transmisor hasta el receptor. Sin embargo para obtener una posición única deberá

---

<sup>2</sup> Este método se basa en la intersección de los círculos con centro en cada una de las estaciones base y radio aproximado a la distancia entre la unidad móvil y el centro de cada circunferencia;

contarse con más de 3 lecturas, tal y como se muestra en la figura 1.4; de lo contrario la localización del dispositivo móvil desembocará en una falsa estimación.



*Figura 1.4 Representación gráfica de la localización por triangulación.*

Actualmente con la introducción de sistemas digitales de gran ancho de banda, la obtención de la información relativa a los tiempos se facilitó al realizar la correlación con secuencias piloto implícitas en el receptor; esta proporciona una mayor precisión, de acuerdo a los tiempos de muestreo utilizados.

Se puede decir que esta tecnología es funcional en interiores y exteriores, adaptándose a cualquier unidad móvil; la precisión llega a ser de 50 a 150[m]; manteniéndose por debajo de los 125[m] el 67[%] del tiempo; con estos valores, se cumplen los requerimientos mínimos solicitados por la FCC para un sistema de localización de este tipo.

La principal ventaja de este método radica en que no necesita LOS, sin embargo no es capaz de eliminar los efectos multitrayectoria en zonas urbanas; además es necesario modificar las unidades móviles y contar con un sistema de transmisión de tipo dúplex; incluyendo una inversión inicial significativa en la infraestructura de la red.

### *1.2.2.3 Time Difference of Arrival (TDOA).*

El principio de este sistema de posicionamiento es diferente al de TOA, ya que TDOA emplea la diferencia entre los tiempos de llegada de la señal procedente del terminal móvil a distintos pares de estaciones base para calcular la posición.

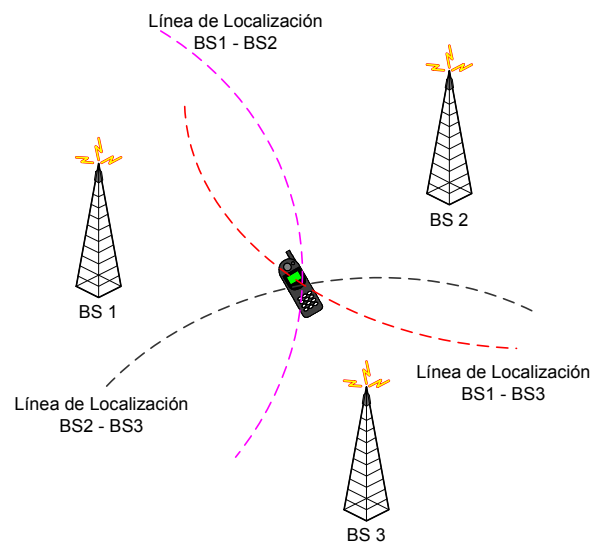
Así pues, TDOA mide el tiempo relativo de llegada de una señal proveniente de la unidad móvil, a cada una de al menos tres estaciones base en un instante dado; es por ello que se requiere de sincronización entre cada una de las bases, tanto móviles como estacionarias. El máximo tiempo de muestreo, dependerá del receptor, por lo que las estaciones base deberán estar equipadas con dispositivos que sean capaces de realizar esta función de forma eficiente.

Al igual que en TOA, la sincronización entre estaciones base es muy importante, pues la falta de sincronía se traduce en errores de precisión. La principal ventaja de esta técnica es que puede funcionar incluso sin señal de visión directa, ya que la diferencia de tiempos cancela posibles errores por reflexiones. Pero en entornos favorables al multitrayecto (áreas urbanas) a veces es necesario efectuar las medidas respecto a cuatro estaciones base para compensar los efectos de las reflexiones. En entornos rurales se puede combinar con la AOA para proporcionar mayor precisión, de tal forma que en entornos suburbanos se puede conseguir una precisión de hasta 125 [m].

Cabe mencionar que a diferencia de TOA, este método es menos sensible a errores del orden de milisegundos y es por ello que resulta ser más utilizado en aplicaciones de localización para sistemas comerciales.

Ahora bien, hablando más del método, se tiene que la curva cuyos puntos satisfacen la condición en la que su distancia a dos referencias (en este caso un par de estaciones base) es una constante; geoméricamente puede ser descrita por medio de una hipérbola, y al calcular esta correlación para varias estaciones base, la intersección de al menos dos hipérbolas muestra el punto donde se encuentra la terminal móvil. Con ello se puede decir que la ubicación estimada del dispositivo móvil, puede ser calculada por la intersección de dos hipérbolas, tal es el caso mostrado en la figura 1.5.

Las ecuaciones representativas de las hipérbolas, pueden ser resueltas por métodos empíricos e iterativos<sup>3</sup>; logrando una precisión estimada menor a 200[m] el 67[%] del tiempo.



*Figura 1.5 Representación del sistema de localización TDOA.*

Usualmente se le asocia a un sistema hiperbólico, porque la diferencia de tiempo es convertida a una distancia que resulta ser una diferencia constante con respecto a dos de las estaciones base, esto forma una hipérbola.

Por lo tanto, se utilizan dos pares de estaciones base, es decir, al menos se necesitan tres de éstas para definir un sistema bidimensional y de esta forma, la posición de la estación móvil.

La precisión del sistema es una función de la ubicación geométrica relativa de cada una de las estaciones base, esto es, entre mejor ubicadas se encuentren las estaciones base y su número sea mayor o igual a tres; la precisión del sistema será mayor.

Otros métodos de localización pueden complementar el sistema, como es el caso de las tecnologías basada en la identidad celular, donde el dispositivo móvil puede ser localizado con un error no mayor al área de la célula predeterminada por el sistema.

---

<sup>3</sup> Por cuestiones del alcance de este presente trabajo de tesis, no se desarrollaran estos métodos.

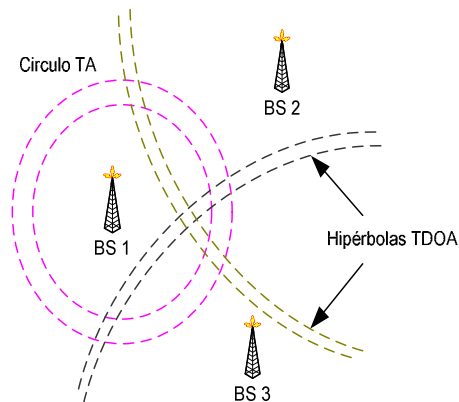


Figura 1.6 Sistema de localización TDOA con variante para precisión.

A partir de la figura 1.6, se puede ver que si se cuenta con una célula muy pequeña podría tener 150[m] de radio, con lo que el error no podría ser mayor a esas dimensiones, con esto, la precisión del sistema sería aceptable en términos generales. En cambio si se tuviera una célula grande con aproximadamente 30,000[m], la posición obtenida no tendría relación con la posición real.

Se puede concluir que aún con esta mejora y en las mejores condiciones, la precisión no podría ser menor a 100[m]; ya que se encuentra limitado por las zonas de disponibilidad del sistema, además de altos costos para la manutención y evolución del sistema.

Resulta importante mencionar que este método no necesita de arreglos de antenas como TOA, sin embargo requiere de unidades adicionales para determinar la localización de los equipos, además se requiere de modificaciones en la red sobre la que se implementa el sistema, ya que red deberá poseer la capacidad de recibir las señales de al menos tres estaciones base de manera simultánea.

Estos requerimientos crean la necesidad de cambios simples en el software adaptables de acuerdo al modelo de cada uno de los dispositivos móviles; además las características de las unidades móviles y la localización geográfica de las estaciones base afectarán la precisión y disponibilidad al momento de determinar la posición del dispositivo móvil.

#### 1.2.2.4 Huella Multitrayecto (MF).

También conocido como Huella RF, este método de localización aprovecha una de las perturbaciones más molestas que se presentan al momento de localizar una terminal móvil, es decir, las señales multitrayectoria.

Este sistema fundamenta su operación en el hecho de que las ondas RF sufren interferencia multitrayecto que se manifiesta por repetidas reflexiones durante su propagación en un medio abundante en obstáculos. Esto hace que la señal procedente de una unidad móvil, no solo llegará en línea recta; sino también a través de trayectorias secundarias.

Así pues, cuando la unidad móvil, que se desea localizar, emita señales iguales en diferentes trayectorias, una de ellas será recibida directamente por la estación base más próxima, mientras que las demás serán reflejadas por los edificios, árboles, colinas y otros

obstáculos que se encuentren en los alrededores; presentando una variación en los parámetros de retardo, trayectoria y atenuación al llegar a la estación.

Para cuantificar la variación de cada uno de los parámetros en la señales, se envían unidades de prueba a distintos puntos, con el fin de que las estaciones base graben las huellas multitrayecto y creen una base de datos.

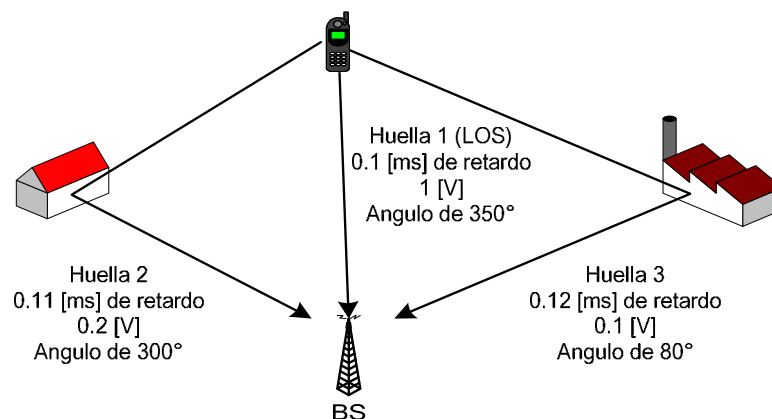
Una de sus principales ventajas, es que el sistema presenta una gran confiabilidad y precisión en áreas extremadamente urbanizadas y congestionadas debido a la existencia de una gran diversidad de objetos que reflejan las ondas de radio; además, dado que dos puntos en el espacio no pueden presentar la misma huella o perfil, los patrones obtenidos en estos lugares son caracterizados de manera única. Permitiendo obtener la posición de una terminal móvil en ambientes donde otros sistemas de localización son limitados, tal y como es el caso de GPS, del cual se hablará más adelante.

Otro de los aspectos importantes que se tienen, es el ahorro en infraestructura, ya que solo requiere de una estación base para localizar al dispositivo móvil. Al momento de que este realiza una llamada, se realiza una comparación entre el patrón de la señal emitida, con los que se encuentran previamente cargados a la base de datos y así poder definir la posible ubicación de este; en caso de no encontrarla, se obtendrá una estimación aproximada.

Sin embargo, al igual que todos los sistemas analizados anteriormente, el sistema posee desventajas, una de ellas se presenta al momento de realizar la recopilación de los datos, ya que la precisión reside en el tipo de terreno donde se llevará a cabo la ubicación.

Las áreas rurales situadas en altiplanicies o valles, carecen de obstáculos significativos para caracterizar cada uno de los puntos del lugar; permitiendo que los únicos factores capaces de alterar las huellas de las ondas de radio sean las condiciones atmosféricas y la flora propia del lugar, principalmente.

En la figura 1.7 se puede observar que cualquier alteración en el terreno, tales como la construcción o demolición de algún objeto, dará lugar a la actualización inminente de la base de datos, en caso de omitirlo, el sistema carecerá de confiabilidad.



*Figura 1.7 Ejemplo del reporte en un sistema MF.*

Para finalizar, es necesario mencionar que existen también técnicas "híbridas", las cuales resultan de combinar algunas de las anteriores. Estas técnicas mejoran la precisión sin modificar notablemente las características de costo y complejidad. Así es posible im-

plementar un sistema híbrido que adopte la estimación mediante AOA para cada estación base y las estimaciones TDOA para estaciones múltiples. También aparece una combinación de AOA y TOA, procedimiento que determina la posición con una precisión similar a MF con una única estación base.

### 1.2.3 Técnicas basadas en la modificación del sistema móvil.

Este tipo de técnicas son una variante de las mencionadas anteriormente, su nombre hace referencia a la modificación de la unidad móvil, es decir, una buena parte del procesamiento se realiza en este, con ellos, las mediciones de diversos parámetros son realizadas y reportadas para su propia localización. En la actualidad se cuenta con dos métodos que entran en esta clasificación, en las secciones siguientes se verá más a detalle cada una de ellas.

#### 1.2.3.1 Enhanced Observed Time Difference (E-OTD).

E-OTD es una técnica avanzada de posicionamiento basada en el principio de operación de TDOA, es decir, utiliza las diferencias de tiempos observados entre las señales procedentes de las estaciones base más cercanas a la unidad móvil. Este método es adaptable a estaciones base con transmisión de tramas síncronas o asíncronas.

En E-OTD, el sistema necesita de la instalación de unidades que proporcionen una medida de la posición, también conocidas como LMU, de modo que sean balizas que sirvan de referencia en puntos dispersos geográficamente. La densidad de las LMUs determinará la precisión del sistema, y por ello normalmente es necesario instalar en toda la red una LMU por cada una o dos *Base Transceiver Station (BTS)*.

En cuanto a las unidades móviles, estas son habilitadas con un software especializado, ya que éstas se encargan de realizar las medidas de las señales procedentes de tres o más estaciones base de forma periódica. Las diferencias temporales de llegada de la señal a las LMU y a las estaciones base se combinan para producir líneas hiperbólicas que se intersecan en el lugar donde está la terminal móvil, ofreciendo de esta manera, localización en dos dimensiones.

Para redes síncronas, la estación móvil mide el tiempo relativo de llegada de las señales procedentes desde varias estaciones base, esto lo realiza sin la necesidad de recurrir a algún otro dispositivo. Sin embargo, cuando la transmisión de tramas de las estaciones base no están sincronizadas, como GSM, las señales son además recibidas por la LMU, en los puntos fijos geográficamente conocidos.

La posición de la estación móvil es determinada por deducción de las componentes geométricas de las diferencias de tiempo real, ya sean de tipo relativo o absoluto entre las distintas estaciones. En este caso son necesarias al menos tres medidas procedentes de las tres estaciones base más cercanas a la estación móvil. Los valores de las diferencias de tiempo son recopilados de forma inteligente entre parejas de estaciones base, para finalmente obtener la localización de la unidad móvil mediante triangulación. De acuerdo al caso, la sincronización de las BTS o la precisión de las mediciones del *Real Time Difference (RTD)*, deben ser del orden de 100[ns]; para tener un error menor a 50 [m].



La localización de la estación móvil puede ser calculada tanto en la red o, si toda la información está disponible, en la estación móvil; de esta forma al primer caso se le denomina *mobile-assisted* y al segundo *mobile-based*.

En *mobile-assisted*, la terminal móvil envía los valores de las diferencias de tiempo al procesador de la red para que este realice la triangulación. Mientras que en *mobile-based*, la estimación de la localización es realizada en el dispositivo móvil.

En cualquiera de los dos casos anteriores, la estimación de la localización es realizada por una *Position Computation Function (PCF)* y puede basarse en cualquiera de los dos tipos de cálculo de localización E-OTD, denominados *Método Hiperbólico (E-OTD-H)* y *Método Circular (E-OTD-C)*.

Para el *Método Hiperbólico*, la diferencia de tiempo observada es medida por el dispositivo móvil, como la diferencia entre los tiempos de llegada (real y geométrica) entre las señales provenientes de un par de estaciones base. La RTD es medida por las estaciones base como las diferencias de tiempo entre sus relojes, asimismo, la diferencia de tiempo geométrica, *Geometric Time Difference (GTD)* es la diferencia entre los tiempos de llegada de la señal a la *Mobile Station (MS)* a consecuencia de la geometría y además, es igual a la diferencia entre *Observed Time Difference (OTD)* y RTD.

Para obtener un valor constante de GTD, la unidad móvil debe ser posicionada en una hipérbola, de esta manera la intersección de las dos hipérbolas debido a tres estaciones base, determina la posición estimada del dispositivo móvil tal y como se puede ver en la figura 1.8.

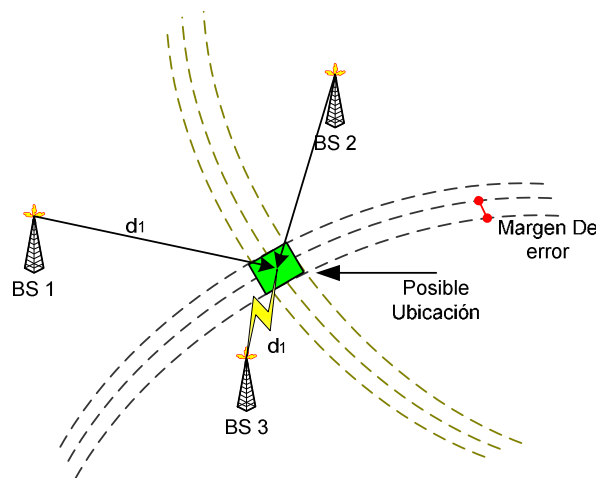


Figura 1.8 Método E-OTD Hiperbólico.

En el *Método Circular*, los tiempos de llegada de las señales a la unidad móvil son medidos individualmente; la distancia del dispositivo proveniente de la estación base es calculado a partir de este valor, lo cual produce un círculo de probabilidad para cada una de las estaciones base.

Al tratar de generar un modelo para este método, tenemos que la distancia geométrica de la unidad móvil a la estación base, la distancia geométrica entre la LMU y estación base y el RTD entre las estaciones base pueden ser considerado como incógnitas del sistema; por lo que se trata de un sistema de tres ecuaciones con tres incógnitas.



Con esta información se puede encontrar la posición estimada de la unidad móvil, así, solo se necesitará la intersección de tres círculos asociados a tres estaciones base. A continuación, se presentan las variables más significativas asociadas a E-OTD-C:

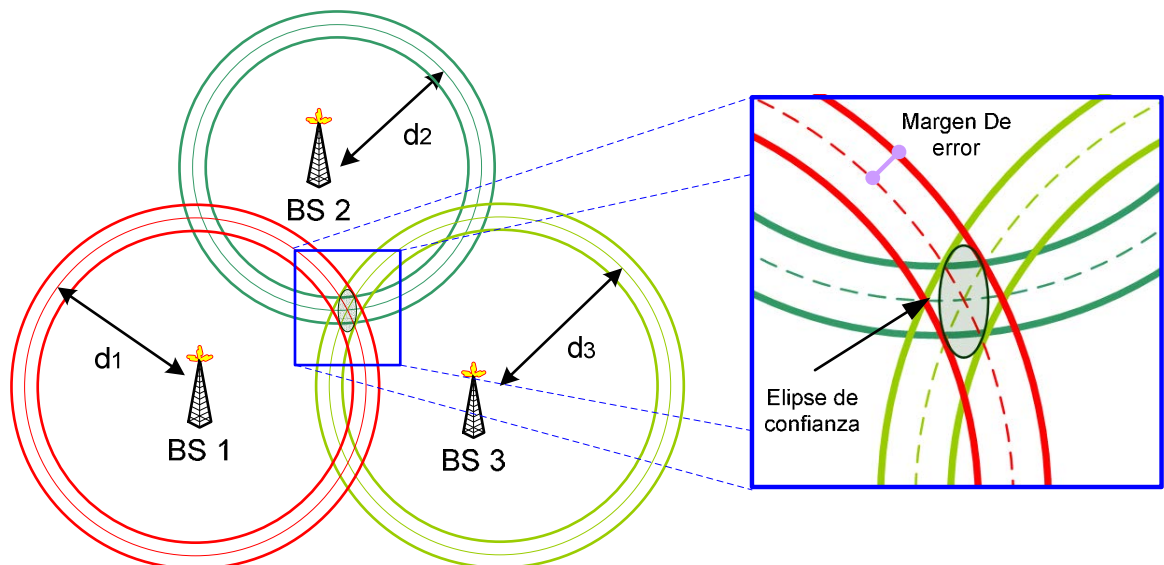
- ⊕ El *Observed Time at the MS (MOT)*, es el tiempo en el cual una señal llega de la BTS, este a su vez es comparado con el reloj interno de la MS.
- ⊕ El *Observed Time at the LMU (LOT)*, es el tiempo en el cual una señal llega de la BTS, este a su vez es comparado con el reloj interno de la LMU.
- ⊕ La diferencia de tiempos de sincronía ( $\epsilon$ ), es la diferencia entre los relojes internos de la LMU y la MS.
- ⊕ La *Geometrical Distance from MS to BTS (DMB)*.
- ⊕ La *Geometrical Distance from LMU to BTS (DLB)*.

Todas estas variables se pueden relacionar por medio de la siguiente expresión:

$$DML - DLB = V (MOT - LOT + \epsilon)$$

En donde,  $V$  es la velocidad de las señales de RF.

Como se había analizado anteriormente, por cada BTS, se tiene una ecuación con tres incógnitas; las cuales se refieren a la posición de la MS en el plano  $XY$  y el tiempo de sincronía. Para realizar la ubicación de la MS principalmente, es necesario contar con al menos tres BTSs; de esta manera la posición estimada de la MS es definida por la intersección de los círculos con centro en cada una de las BTSs, esto resulta ser una analogía de las observaciones hechas por la MS y las LMU's cuando miden los tiempos de llegada y los tiempos observados (LOT y MOT).



*Figura 1.9 Método E-OTD Circular.*

En la figura 1.9 se muestra que este método no presenta una exactitud del 100%, para este caso, la incertidumbre presentada se asocia con el radio de los círculos que rodean a las BTSs, también es conocido como margen de error de la medición. La región en donde

se traslapan los radios de las BTSs, incluyendo su margen de error, se denomina elipse de confianza, ya que dentro de ellas es en donde se localiza la MS.

En E-OTD-C, la precisión es de aproximadamente 150[m], pero en condiciones ideales, es capaz de identificar la posición con pocos metros de error.

Uno de los líderes en el campo de E-OTD es *Cambridge Positioning Systems* con *Cursor™* basado en E-OTD, el cual puede entregar una precisión de hasta 50 [m] para las redes GSM; inclusive posee una gran precisión para las redes 3G. Actualmente *Cursor™* es utilizado por el *Mobile Positioning System (MPS)* de Ericsson para determinar la ubicación de sus equipos. E-OTD es actualmente utilizado sobre redes GSM y GPRS, incluyéndose como una nueva tecnología tanto en la terminal móvil como en la red.

La principal ventaja del E-OTD, es que presenta una precisión de 50[m] o menos, con una red bien planeada. Vale la pena mencionar que las LMU's pueden ser integradas o no en las estaciones base, pero con el fin de obtener mejores mediciones en cada uno de los tiempos, las LMU's contienen un receptor de GPS, que proporcionan una mayor precisión a la sincronía en sus relojes.

#### *1.2.3.2 Advanced Forward Link Trilateration (A-FLT).*

La técnica A-FTL es exclusiva para redes CDMA, pues éstas son síncronas en cuanto a la operación se refiere. El método es muy similar al TDOA, básicamente consiste en efectuar la medida del retardo de fase entre señales enviadas a un par de estaciones base, y compararla con la medida de otro par.

Los datos procedentes de tres estaciones base permiten localizar una terminal móvil. También existe otra técnica mejorada con los mismos fundamentos que AFLT, que es EFLT (Enhanced Forward Link Trilateration).

### **1.3 Localización mediante redes inalámbricas.**

A pesar de las limitaciones e inconvenientes de las técnicas de localización en exteriores, éstas son suficientes para ofrecer numerosos servicios al usuario final. Ahora bien, el reto es ahora la localización en espacios cerrados, lugares donde las tecnologías basadas en la red o en la modificación de la terminal móvil pierden su precisión debido a obstáculos como paredes y techos, provocando un error inaceptable.

En la actualidad existen varias tecnologías que pueden utilizarse en interiores de edificios. Las más destacadas son Bluetooth, Wi-Fi (estándar 802.11b) y la denominada Banda Ultra-Ancha (Ultrawideband). A continuación analizaremos de manera somera cada una de ellas.

### 1.3.1 Bluetooth.

Es una especificación que surge en 1994, bajo la iniciativa de *Ericsson Mobile Communications* e *Intel*, la cual ofrece conectividad a las redes *Wireless Personal Area Network (WPAN)* mediante un dispositivo móvil de manera económica.

En realidad, no se plantea como una alternativa real a las redes WLAN, sino más bien como un sustituto del cable en las conexiones a corta distancia de dispositivos como computadoras portátiles, teléfonos móviles, PDAs, por mencionar algunos; además de ofrecer conexión a una LAN o WAN a través de un punto de acceso.

Bluetooth define un canal de comunicación de 721[kbps] en *downstream* y de 56 [kbps] en *upstream*, basándose en la tecnología *Frequency Hopped Spread Spectrum (FHSS)* emplea frecuencias en el rango de 2.4 a 2.8 [GHz] que cubren un radio de 10[m] a 100[m]. Adicionalmente, ofrece un bajo costo por ser una tecnología fácil de implementar, también presenta un consumo de potencia mínimo, ya que para transmitir a una distancia de 10[m] emplea 1[mW] y para 100[m] solo 100[mW]. Asimismo posee la capacidad de funcionar en entornos radioeléctricos ruidosos y sobretodo, con buenas tasas de transmisión.

El sistema de localización basado en esta tecnología se denomina Red de Localización Bluetooth o de sus siglas en inglés *BLN*. Este sistema transmite información de la posición del terminal móvil a los servidores, sin la participación del usuario. No es objeto de restricciones debido a la pérdida de línea de vista y funciona con dispositivos comerciales ya existentes que sean o puedan ser compatibles con Bluetooth a través de una tarjeta de expansión.

El BLN está compuesto por pequeños nodos Bluetooth que establecen una topología de red espontánea con la inicialización del sistema. Puede cohabitar con dispositivos Bluetooth que no son parte del sistema de localización, como impresoras y auriculares.

### 1.3.2 Wireless Fidelity (Wi-Fi).

Wi-Fi surge como resultado de una decisión unánime tomada en 1985 por la FCC, con el propósito de habilitar varias bandas del espectro inalámbrico, sin necesidad de una licencia gubernamental; sin embargo para trabajar en ellas, los dispositivos necesitan utilizar la tecnología de "espectro disperso", ya que esta propaga una señal de radio en un amplio rango de frecuencias permitiendo que la señal sea difícil de interceptar y menos susceptible a la interferencia.

En 1990, se configuró un nuevo comité IEEE, llamado 802.11, destinado a iniciar un estándar; pero no fue hasta 1997, casi una década más tarde, cuando se realizó su publicación y en los años sucesivos, se ratificaron variantes de este, como el 802.11b que funciona en la banda *Industry, Science and Technology (ISM)*, el 802.11a y el 802.11g, por mencionar algunos.

A continuación se muestra una tabla con una descripción general de las variantes más difundidas y utilizadas, posteriormente se ampliará esta información.

<p><b>IEEE 802.11:</b> Estándar original de las WLAN que es capaz de soportar velocidades de transmisión entre 1 y 2 [Mbps].</p> <p><b>IEEE 802.11a:</b> Estándar de alta velocidad que soporta hasta 54[Mbps] en la banda de 5[GHz].</p> <p><b>IEEE 802.11b:</b> Estándar dominante de WLAN, denominado como "Wi-Fi", llega a obtener velocidades de hasta 11[Mbps] en la banda de 2.4[GHz].</p> <p><b>IEEE 802.11g:</b> Estándar aprobado en junio del 2003, incrementa la tasa de transmisión hasta 54 [Mbps], al igual que el estándar anterior, se localiza en la banda de 2.4[GHz].</p>
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

*Tabla 1.2 Estándares para Wireless LAN*

Hoy en día, las tecnologías inalámbricas de área local se encuentran bajo una gran demanda, y en respuesta, una gran cantidad de distribuidores realizan su instalación en lugares estratégicos; con esto se prevé proporcionar, además de acceso casi instantáneo a la red mundial, servicios de localización altamente confiables.

Esto es posible gracias a la cobertura que presentan, ya que puede ser de hasta 75[m] en interiores y de hasta 300[m] en exteriores, esta última pueden llegar a ampliarse a varios kilómetros a la redonda, mediante arreglos de antenas.

La localización mediante este tipo de redes, se puede llevar a cabo de diferentes maneras, la más fácil de realizar, es a través del punto de acceso más cercano a la terminal del usuario y así delimitar la posible zona en donde pueda encontrarse.

Sin embargo, en ocasiones puede llegar a causar confusión, ya que la zona delimitada puede describirse en tres dimensiones, es decir, tener una geométrica similar a la de una esfera y así poder abarcar más de un piso dentro de un edificio. Por otra parte, este método es altamente vulnerable a las interferencias del medio y por consecuencia, impreciso.

Existe otro método basado en la propuesta de *EkaHau*, compañía que comercializa un motor de posicionamiento denominado *EkaHau Positioning Engine (EPE)*; este sistema basa su funcionamiento en el almacenamiento de medidas de potencia de una señal en diferentes puntos delimitados por un lugar cerrado y la técnica conocida como Wi-Fi mapping.

A través de estas dos herramientas, se pueden obtener resultados más exactos en comparación a los que se tienen en métodos de triangulación celular, ya que su precisión se encuentra en el rango de las unidades de metro, esto depende en gran medida de las condiciones bajo las cuales se realice la medición, tales como la proximidad de los puntos que conforman la rejilla de mapeo; esto ofrece una precisión de 1 a 20 metros. Además hay que mencionar que es sensible a los cambios de altura, por lo que provee una mejor precisión con respecto al primer método descrito.

Finalmente, otro de los métodos que utilizan a Wi-Fi para la localización de un usuario, es el que proponemos en este trabajo de tesis, sin embargo este será descrito a profundidad en capítulos posteriores.

### 1.3.3 Banda ultra-ancha.

De sus siglas en inglés, *Ultra Wide Band (UWB)* es una tecnología basada en radio frecuencias desarrollada por el Departamento de Defensa de los Estados Unidos en 1989 con el propósito de evitar la inseguridad en sus comunicaciones. Ofrece una transmisión de datos de alta velocidad y potencias mil veces más bajas que las de un teléfono celular bajo la técnica de espectro disperso.

Puede utilizarse para transmitir voz, video u otro tipo de datos digitales; esto es gracias a la inmunidad que presenta ante las señales ajenas al sistema, tales como las señales multitrayectoria.

En comparación con otro tipo de tecnologías inalámbricas, UWB proporciona una mayor velocidad de transmisión con una gran eficiencia en potencia, facilidad de implementación e infraestructura poco costosa; lo que permite el desarrollo de dispositivos portátiles autónomos. Sin embargo, debido a las limitaciones de potencia, el alcance se reduce a una decena de metros.

Asimismo se tiene que el principal campo de aplicación de UWB se orienta hacia la electrónica doméstica, por ejemplo en la interconexión de periféricos tales como monitores, teclados, monitores e impresoras con una *Personal Computer (PC)*, o en la distribución de señales *High Definition Televisión (HDTV)* a distintos receptores de televisión. Estudios de mercado predicen que UWB tendrá un gran auge durante el 2007, especialmente en los sectores de redes y electrónica de consumo, donde se prevé unas ventas en torno a 50 millones de unidades.

Sin embargo, el éxito de la tecnología UWB depende en gran medida de la adopción de un estándar. En la actualidad, existen diferentes soluciones propietarias que no pueden interactuar entre sí debido al uso de formas de pulso y técnicas de modulación distintas. El estándar se encuentra prediseñado para esta tecnología es el desarrollado bajo el nombre de IEEE 802.15.3a.

La principal barrera para la implantación de UWB dependerá del nivel de grado de penetración que haya alcanzado Wi-Fi en el momento en que se proceda a su lanzamiento comercial masivo. Aunque en realidad, UWB aparece como un contendiente de Bluetooth y un complemento de Wi-Fi, ya que al hacer una analogía, el 802.11 sería un router inalámbrico perteneciente a una red, y UWB sería la red de área local.

La siguiente tabla muestra un resumen comparativo de las tecnologías analizadas en este tema. Como se puede observar, UWB presenta características muy superiores con respecto a las demás; sin embargo con una potencia de emisión insignificante, está limitada en alcance.

<i>Tecnología</i>	<i>Tasa de transmisión [Mbps]</i>	<i>Potencia [mW]</i>	<i>Alcance [m]</i>	<i>Frecuencia [Ghz]</i>
<i>Bluetooth</i>	1-2	100	100	2.4
<i>UWB</i>	100-500	1	10	3.1-10.6
<i>IEEE 802.11a</i>	54	40-800	20	5
<i>IEEE 802.11b</i>	11	200	100	2.4
<i>IEEE 802.11c</i>	54	65	50	2.4

*Tabla 1.3 Comparación de tecnologías.*

La localización de terminales móviles mediante esta tecnología es similar a TOA, ya que se basa en la transmisión de una señal digital de alta frecuencia que será captada por tres o más receptores diferentes; a su vez, esta será caracterizada con el fin de cuantificar el tiempo de llegada de las señales y al utilizar la trilateración se obtiene la posición del dispositivo con una precisión en el orden de centímetros<sup>4</sup>.

Una de sus desventajas, en cuanto a la localización se refiere, reside en el hecho de utilizar una infraestructura masiva para un área relativamente pequeña; sin embargo la precisión que se obtiene es en mucho, superior a la de otros sistemas.

## **1.4 Global Positioning System (GPS).**

### **1.4.1 Introducción.**

El Sistema de posicionamiento global, de sus siglas en inglés GPS, es un sistema de navegación idealizado por la armada y la fuerza aérea norteamericanas en el primer lustro de los años 60, grosso modo, GPS le permite conocer la ubicación precisa y continua de su armamento, sin depender del lugar geográfico o condiciones meteorológicas.

Este sistema surge en 1973 con los problemas que experimentaron las tropas norteamericanas en el conflicto con Vietnam. En esa época se utilizaba un sistema llamado LORAN y debido a sus deficiencias, como las de cualquier frecuencia de radio, se buscó otra alternativa.

Estados Unidos desarrolló entonces lo que ahora es el sistema GPS, pero solo con cuatro satélites y lo llamaron *Transit*. Este sistema era de uso limitado debido a la insuficiencia de satélites.

Después de numerosas inversiones millonarias, en 1989 se consolida la idea enviando los primeros satélites. No obstante, debido a un incidente internacional, el entonces presidente de E.U.A, Ronald Reagan, permitió el uso del GPS a la comunidad civil internacional, bajo la consigna de una precisión menor.

No obstante, fue hasta en 1990 cuando un sistema con una mayor red satelital llamado NavStar quedó finalmente funcionado al 100% con una red de 21 satélites. En 1991, el sistema demuestra su eficiencia durante la guerra del Golfo Pérsico, siendo considerado el más importante de los elementos de equipamiento de esa época. Y finalmente en el 2000, Bill Clinton elimina la limitante de precisión para uso civil, logrando precisiones de hasta  $15[m]$  en condiciones normales, ya que los receptores fueron diseñados para limitar su funcionamiento cuando se encuentran a más de  $18000[m]$  de altitud y/o una velocidad

mayor a  $1.667 \left[ \frac{km}{h} \right]$ .

El sistema GPS está formado por tres segmentos o áreas: *el segmento espacial*, *el segmento de control* y *el segmento de usuario*. El primero engloba los satélites del sistema, el segundo abarca las infraestructuras terrestres necesarias para el control de la constelación de satélites. Por último, el segmento de usuario está constituido por los equipos de recepción

---

<sup>4</sup> Se dice que son centímetros por el alcance predefinido del sistema.

y el software para el procesamiento de señales. Los satélites son capaces de producir dos tipos de señales: una muy precisa de uso militar y otra de libre uso a la que las estaciones de tierra introducen un error aleatorio controlado.

El sistema cuenta principalmente con 5 estaciones fijas y una red de 24 satélites denominados NAVSTAR. Las estaciones fijas se encuentran repartidas en cinco puntos estratégicos en el planeta tierra: Hawái, Diego García, Isla de Ascensión, Colorado Springs y Atolón de Kwajalein. Estas estaciones se encargan de vigilar el estado operativo de los satélites y su correcta posición en el espacio. Una de ellas cumple las funciones de estación principal y transmite las correcciones a los satélites.

Los satélites; con tiempo de vida útil de 10 años; se encuentran distribuidos en 6 trayectorias orbitales situadas a 20,200[km] aproximadamente sobre la superficie terrestre; realizando dos translaciones con respecto a la tierra en 24 horas. De toda la constelación, 3 satélites se encuentran de reserva y 21 en servicio; cada uno de ellos transmite de manera continua una señal igual o menor a 50[W], sin importar las condiciones meteorológicas que se presenten.

La señal que transmite cada satélite consta de un código pseudo aleatorio, datos *ephemeris* y posicionales. El primero, también denominado *Pseudo Random Code (PRC)*, por sus siglas en inglés; es un código de identificación del satélite asignado de forma pseudo aleatoria por medio de un número en el rango de 1 a 32, que se visualiza en el receptor e identifica al satélite que transmite en cada posición geográfica. La asignación del rango, facilita el mantenimiento de los propios satélites; ya que en caso de falla; uno de los satélites de reserva, entrará en funcionamiento y adquirirá un PRC que no haya sido asignado.

Los datos *ephemeris* son datos enviados de forma constante por cada satélite con información sobre su propio estado y datos sobre la fecha y la hora; mientras que los posicionales indican al receptor, la posición de cada satélite en cada momento del día. De esta forma se concluye que la información que envía el satélite es un número identificador propio junto con su posición y la hora a la que fue enviada la información.

La manera de determinar la posición del dispositivo móvil es por medio de la diferencia de tiempos de transmisión y recepción; de esta manera se obtiene la distancia entre los puntos; además con las mediciones proporcionadas por otros satélites se puede obtener información adicional con respecto a la unidad móvil, es decir, su velocidad y altura con respecto al nivel del mar.

## 1.4.2 Funcionamiento.

### 1.4.2.1 Trilateración.

El principio básico para determinar la posición de un receptor GPS es la trilateración y no la triangulación; ya que la primera, es la medida obtenida de los lados de los triángulos en lugar de los ángulos para determinar la posición, como ocurre en la segunda.

Para llevar a cabo la trilateración, el receptor GPS calcula la distancia hasta el satélite midiendo el tiempo que tarda la señal en llegar hasta él. Para ello, debe de contar con un sistema muy preciso para medir los tiempos, de lo contrario se generarán errores en la

ubicación del receptor. Para minimizar estos efectos negativos es necesario corregir las señales recibidas, en el mismo receptor.

Una vez que el receptor GPS recibe la posición de al menos cuatro satélites y conoce su distancia hasta cada uno de ellos, puede determinar su posición superponiendo las esferas imaginarias que generan.

Está comprobado matemáticamente que se necesitan cuatro mediciones de distancia a los satélites para determinar la posición exacta. En la práctica, la posición correcta aparente se obtiene con solo tres mediciones, por medio de la eliminación de supuestos absurdos. Sin embargo, a pesar de su aparente falta de utilidad, este cuarto satélite tendrá una función crucial en la obtención de la posición, como se verá más adelante.

#### *1.4.2.2 Medición de distancias.*

La distancia a cada uno de los 4 satélites necesarios para determinar la posición del receptor GPS, se obtiene a partir del tiempo que le toma a una señal RF desde que es emitida por el satélite hasta que es recibida en la unidad terrestre; en este caso, se utiliza un PRC.

El código es una secuencia digital con un grado de complejidad elevado; para su transmisión, emplea una señal portadora a una frecuencia de  $1575.42 [MHz]$  en el caso de uso civil y de  $1227.60 [MHz]$  para uso militar. La señal se asimila a un ruido eléctrico generado aleatoriamente; sin embargo esta dificultad se explica fácilmente cuando el propósito es evitar que el receptor GPS llegara a sintonizarse por equivocación con algún otro dispositivo emisor y/o pudiera confundirse entre el ruido ambiental.

Sin embargo esta medición se complica debido a la brevedad del parámetro a obtener; por lo que debe considerarse una perfecta sincronía entre ambos dispositivos terminales; es decir, se parte del supuesto de que los dispositivos generan la señal en el mismo instante. Así, al comparar el retardo existente entre la señal que proviene del satélite y la generada en el receptor; se puede obtener la distancia recorrida al multiplicar el tiempo medido con una velocidad aproximada a la de la luz.

Cabe mencionar que el código pseudo aleatorio es único para cada satélite, por lo que garantiza la confiabilidad de los datos; evitando equivocaciones al momento de ser recibidas en el receptor. De esta forma, los satélites pueden utilizar la misma frecuencia para transmitir sin que se interfieran; además de proporcionar seguridad al sistema y un completo control al Departamento de Defensa de E.U.A.

Otras de las ventajas de la aplicación del código es que permite la amplificación y recuperación de las señales, sin necesidad de utilizar antenas o dispositivos costosos en el receptor; por lo que hace a GPS un sistema relativamente económico y de fácil implementación.



#### *1.4.2.3 Control de Tiempo.*

Con el fin de obtener una sincronía casi perfecta, los satélites GPS son equipados con relojes atómicos de gran precisión; ya que un error de  $1[\mu s]$  se traduce en un error de 300[m].

Sin embargo, los receptores GPS, por el gran costo que implicaría, no alojan relojes de tipo atómico; ello no impide que entre ambas terminales se obtenga la sincronía deseada, ya que por medio de una cuarta medición satelital es posible conseguir en esencia, un reloj atómico por su precisión.

Los diseñadores del sistema se percataron de que tres mediciones perfectas pueden posicionar un punto en el espacio; mientras que cuatro mediciones imperfectas pueden lograr el mismo resultado.

La cuarta medición no necesariamente intersectará en un punto a las tres mediciones anteriores; de manera que el sistema del receptor hallará una diferencia en tiempo que será adjudicada a la falta de sincronía con respecto a la hora universal. Es entonces cuando este, aplicará un factor de corrección a cada una de las mediciones obtenidas siendo capaz de unificarlas en un punto; propiciando su reajuste a la hora universal y brindando su posición precisa.

En la práctica, el receptor GPS puede llegar a captar de manera simultánea, la señal proveniente de 6, y hasta 12 satélites; con el fin de minimizar al máximo los errores concebidos por la falta de un reloj de alta precisión en él.

#### *1.4.2.4 Ubicación de satélites en el espacio.*

Para que la trilateración sea realizada con éxito, es necesario que el receptor conozca con exactitud la posición en tiempo real de cada uno de los satélites confortantes de la constelación. Esto es posible por medio de una base de datos que se encuentra cargada en el sistema del mismo y que se actualiza constantemente.

Los satélites fueron colocados por la Fuerza Aérea de E.U.A a una altura aproximada de 20000 [km], donde las órbitas resultan ser predecibles por medio de ecuaciones matemáticas de fácil resolución; gracias a su falta de contacto con la atmósfera terrestre.

A pesar de su estabilidad y exactitud, cada órbita requiere de un continuo monitoreo realizado por radares proporcionados por el Departamento de Defensa; siendo éstos, los encargados de controlar la altura, velocidad y posición de cada satélite.

Cada vez que se determina la posición exacta de un satélite, la información corregida le es reenviada; para que a su vez, la retransmita junto con la señal de RF a los receptores en tierra. De aquí se puede concluir que, el código pseudo aleatorio no solo presenta utilidad para calcular la distancia existente entre el satélite y el receptor; sino que también es el portador de un mensaje de navegación o estado que comprende toda la información referente a la órbita exacta del satélite.

#### 1.4.2.5 Corrección de errores.

Por medio de una sincronía casi perfecta y la ubicación exacta de cada uno de los satélites podría obtenerse una posición muy aproximada del receptor; sin embargo, es necesario eliminar los efectos de los errores ocurridos durante la trayectoria de la señal para lograr una posición “perfecta”.

A continuación se enumerarán y describirán cada una de las fuentes que pueden ocasionar los posibles errores de posicionamiento en el sistema, así como su respectiva corrección.

##### ✦ *Disponibilidad Selectiva.*

La Selective Availability (SA) es la primera y más importante de los aspectos que pueden introducir un nivel de error en las mediciones realizadas por el receptor GPS. Se trata de una limitación intencionada, impuesta a los receptores civiles por disposición del Departamento de Defensa de los Estados Unidos; con esta se puede generar un error máximo en la posición de 100[m], siendo el error promedio de 30[m].

Este error se puede generar de dos maneras principalmente; la primera consiste en introducir pequeñas alteraciones en los datos del reloj satelital, mientras que la segunda se basa en el envío de datos sutilmente equívocos a los receptores, con respecto a las posiciones orbitales.

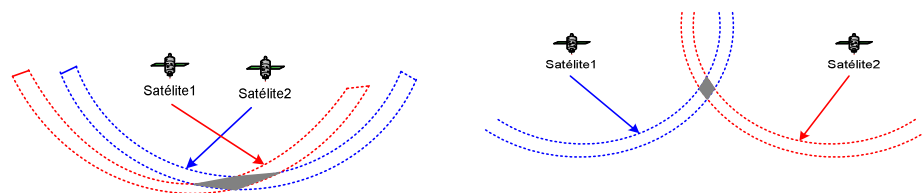
En el caso de los receptores exclusivos al uso militar, se utiliza una clave encriptada para eliminar la SA y así obtener una precisión en el orden de las unidades de metros.

El motivo de existencia de esta política radica en mantener la seguridad nacional; ya que evita que alguna fuerza de tipo hostil o terrorista utilice el sistema para fabricar armas de gran precisión. Es por ello que este error no puede ser minimizado.

##### ✦ *Dilución geométrica de la precisión (GDOP).*

Dado que un receptor puede tener acceso a cualquier satélite que forma parte de la constelación, es necesario que la elección de aquellos que le proporcionen su posición, sea la que le brinde el menor *Geometric Dilution of Precision (GDOP)*; es decir, aquellos que se encuentren a una distancia tal, que la intersección de los anillos de circunferencias que tienen como radio, la distancia que existe entre los satélites y el receptor; se produzca aproximadamente a 90[°] y delimite la zona más pequeña en la que se pueda localizar.

En la figura 1.10, se puede observar que entre más cerrado es el ángulo, la región es mayor (izquierda), mientras que a un ángulo mayor la zona de localización es más estrecha. En caso de que el GDOP no cumpla con la característica requerida, los errores ocasionados por desfases en los relojes o desvíos en las trayectorias podrían incrementarse. Para solucionar este problema se hace uso de receptores de mayor costo, que proveen de una mayor capacidad para determinar un GDOP menor.



*Figura 1.10 Ejemplo de GDOP.*

✦ *Efectos multitraectoria.*

Una vez que la señal proveniente del satélite llega a la superficie terrestre; sufre de múltiples reflexiones provocadas por los objetos que se localizan en su trayecto. De esta manera, la señal y sus reflejos, llegan al receptor en intervalos diferentes, ocasionando la afectación de los parámetros dependientes del tiempo de llegada de la señal.

Para eliminar este efecto, los receptores son diseñados para que excluyan las señales que presenten las mismas características de una señal procedente de un satélite que haya llegado primero.

✦ *Efectos atmosféricos.*

Debido a que el planeta Tierra está rodeado por la atmósfera, los errores que en ella se producen afectan de manera considerable la recepción de las señales; con el fin de minimizar sus efectos se han realizado las siguientes acciones:

*Modelo matemático:* Se trata de un modelo capaz de ser actualizado de manera continua, a partir de la información que se recibe por parte de los satélites y que representa de manera efímera el comportamiento de la atmósfera terrestre.

*Medición dual de frecuencias:* Este sistema es uso exclusivo militar y se basa en el manejo de dos señales emitidas simultáneamente; las cuales serán comparadas en el receptor con el fin de eliminar los posibles errores obtenidos durante su trayectoria. Esta medición de frecuencia dual es muy sofisticada y solo es posible en receptores GPS muy avanzados.

✦ *Otros.*

Dentro de los errores más comunes pero menos perceptibles, se encuentran los relacionados con las efemérides o evolución orbital, así como los generados por influencias gravitacionales del sol y de la luna.

Por otra parte, aunque los relojes atómicos utilizados en los satélites son muy precisos, no son perfectos y los monitoreos orbitales se realizan de manera constante, pero no por fracción de segundo; estas discrepancias originan errores muy sutiles en los parámetros donde interviene el tiempo. Si se desea una gran exactitud en el sistema, éstos deberán de tomarse en cuenta.

Finalmente, se deben de tomar en cuenta todas las posibles fuentes de error para minimizar sus efectos. En la figura 1.11 se muestran ejemplos de los factores que alteran la señal durante su transmisión.

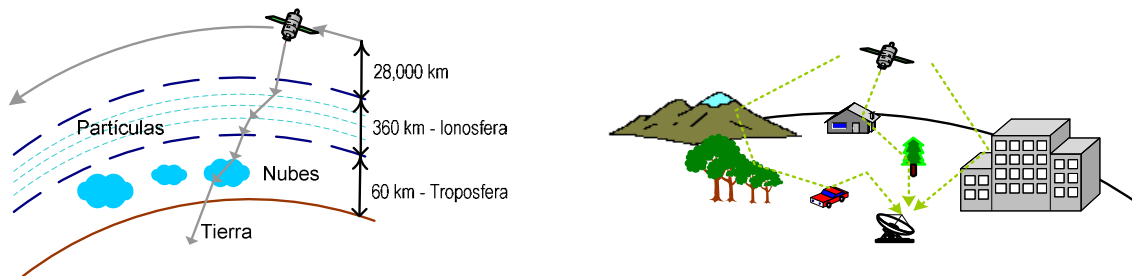


Figura 1.11 Fuentes de error en la transmisión de la señal de un satélite GPS.

La precisión del sistema GPS para uso civil es de  $18.29[m]$  a  $68.6[m]$ ; mientras que para el uso militar se obtiene en el orden de los centímetros o unidades de metros; obviamente la variación obtenida en ambos casos, dependerá de las condiciones climatológicas del lugar en donde se encuentre el receptor.

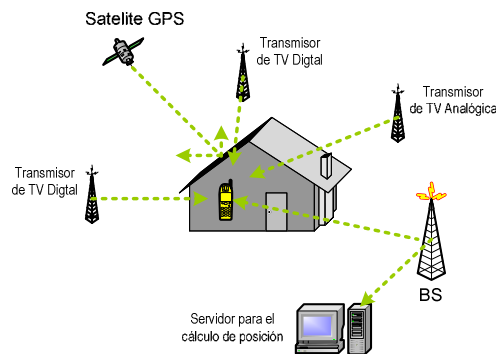
Actualmente existe una variación del sistema GPS tradicional, denominado GPS Diferencial, este permite la misma precisión de uso militar para uso civil. Esto es posible, por medio de la utilización de un segundo receptor para balizas del sistema diferencial y que es acoplado al receptor GPS, con el fin de calcular las correcciones de las mediciones obtenidas por el receptor original. A grandes rasgos, este es el funcionamiento del sistema GPS actual.

### 1.4.3 Localización mediante GPS y señal de Televisión.

Como se ha analizado hasta el momento, ninguna de las tecnologías anteriores funciona con un rendimiento óptimo en todos los entornos. Para minimizar esta limitación se crearon una serie de soluciones híbridas; como es el caso de la unión del sistema GPS con una técnica capaz de posicionar un dispositivo móvil mediante señales de sincronización de televisión que son transmitidas por estaciones televisivas analógicas o digitales, tal y como se muestra en la figura 1.12.

A diferencia del GPS tradicional, este sistema híbrido es útil para la localización en interiores, ya que las señales utilizadas son de alta potencia y de gran ancho de banda.

En términos de cobertura, las señales de TV son un complemento a las satelitales, ya que están concentradas en áreas urbanas y pueden penetrar fácilmente en los edificios.



*Figura 1.12. Sistema de localización mediante TV.*

Con la utilización invariable de la infraestructura de TV digital, es posible crear un sistema de localización fiable, preciso y rápido. Dado que GPS opera eficientemente en áreas rurales remotas y la tecnología de posicionamiento mediante TV resulta eficaz en emplazamientos urbanos, la unión entre los dos sistemas puede dar lugar a una buena solución que unifique las técnicas de localización, además de ser disponibles en áreas externas y recintos cerrados. A pesar de todo, la precisión del sistema dependerá de factores como la cobertura de las estaciones de televisión, los diferentes niveles de atenuación y los efectos multitrayectoria sufridos por la señal.

## CAPÍTULO II

### WIRELESS LOCAL AREA NETWORK (WLAN)

#### 2.1 Introducción.

WLAN fue desarrollada en 1979 a partir de enlaces infrarrojos que en conjunto formaban una red de comunicaciones dentro de un espacio bien delimitado; y a partir de ese momento ha evolucionado hasta definirla como una red de tipo local que se basa en la tecnología de radiofrecuencia, principalmente, para comunicar los equipos confortantes de la red, en lugar de los medios alámbricos.

Así pues, WLAN surge como una solución ante la necesidad de movilidad universal, enfocándose a una filosofía *seamless* o sin discontinuidades, es decir, se buscó que la comunicación pudiera realizarse a través de diferentes entornos sin tomar en cuenta la mayor parte de las limitaciones que presentan las redes alámbricas.

Sin embargo, en sus inicios, no existía un estándar que determinara su correcta implementación; por lo que los dispositivos elaborados por los diferentes fabricantes, resultaban ser incompatibles y no existía la interoperabilidad. Estas deficiencias aunadas a la inversión inicial, fueron los factores que aplazaron su introducción al mercado.

Y es hasta 1997, con la publicación del estándar IEEE 802.11, WLAN comienza a ganar terreno en el campo de las comunicaciones inalámbricas; atrayendo miles de usuarios, que con el transcurso de algunos años, se convertirían en millones. Además con el creciente mercado de los dispositivos que permiten el acceso inalámbrico, como los PDAs y Laptops, el empleo de una red inalámbrica, que proporcione Internet de banda ancha y alta velocidad, cobra pleno sentido.

Luego entonces, podemos decir que para considerar a una red como tipo WLAN, debe cumplir con alguna de las diferentes versiones predefinidas en el estándar IEEE 802.11, cuyas características se enunciarán en un tema posterior, tal es el caso de la figura 2.1 que muestra una configuración básica de una WLAN.

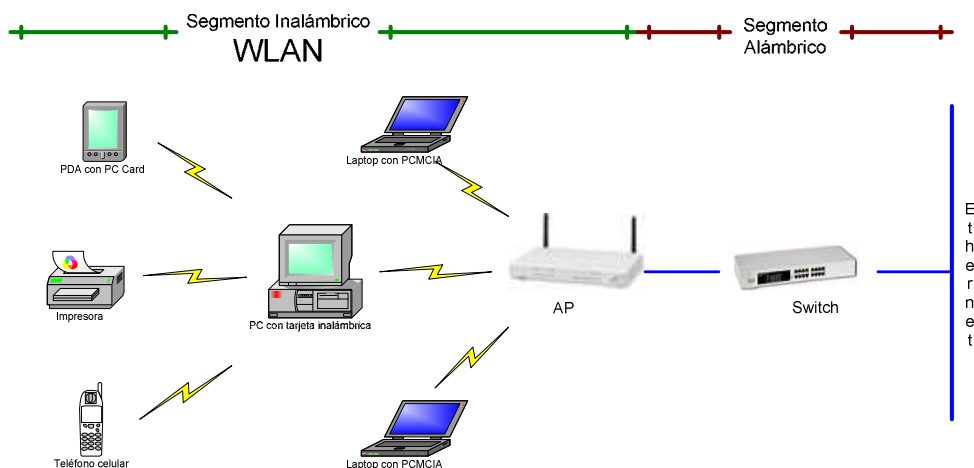


Figura 2.1. Ejemplo de una red inalámbrica.

## 2.2 Componentes que conforman una WLAN.

Retomando el segmento inalámbrico de la figura 2.1 podemos ver que son muchos los elementos que conforman a una WLAN, no obstante, los podemos agrupar en cuatro principales conjuntos, estos son las terminales, los equipos dedicados, los Access Point's (APs) y las antenas.

### 2.2.1 Terminales.

En el caso de las terminales, encontramos que una terminal es un dispositivo móvil inalámbrico o bien puede ser una PC de escritorio o una estación de trabajo con su respectiva interfaz de hardware que le permite conectarse a la WLAN. Un sinónimo de terminal, podría ser el cliente, pues básicamente lo que hace es ocupar la infraestructura o los servicios de la red para operar a través de ellos.

Muchos son los dispositivos que sirven como terminales en una WLAN, esto se debe a que en la actualidad el desarrollo de este tipo de dispositivos ha crecido, sin embargo, encontramos principalmente:

- ⊕ *Laptop*. Actualmente se les puede considerar como una pequeña computadora o *notebook* personal móvil, pues posee componentes similares, pero miniaturizados a modo de optimizar el consume de energía.
- ⊕ *Subnotebooks*. También conocidas como *kneetop* o *ultraportable*, estas terminales son similares a las laptops, solo que son extremadamente más pequeñas y ligeras, pero con una capacidad de procesamiento muy similar; su tamaño ha llegado a ser menor a una hoja de papel A4 (entre tamaño carta y oficio).
- ⊕ *Personal Digital Assistants*. También conocidas por sus siglas como PDA, estos son dispositivos portátiles, que caben en la palma de la mano; inicialmente fueron diseñados como organizadores, sin embargo, últimamente se les han agregado otras funcionalidades como calculadoras, reproductores de MP3 y video, GPS, e inclusive se les puede conectar a una redes Wi-Fi o redes inalámbricas WAN.
- ⊕ *Portable Data Terminal*. Este es un dispositivo también conocido como *Portable Data Terminal (PDT)*, el cual es usado para transmitir o recibir datos vía inalámbrica, se utilizan como etiquetadoras o lectoras de códigos de barras de los centros comerciales, hospitales o algunas fábricas, así como terminales financieras, en donde se tiene una gran base de datos con la información pertinente a cada artículo.
- ⊕ *Tablet PC*. Este dispositivo es básicamente una *notebook*, la cual opera a través de una pantalla *touch screen* y una pluma que permite al usuario operar la computadora en vez del teclado y el ratón, su forma ofrece mayor movilidad y facilidad de interacción, a diferencia de la PC.
- ⊕ *SmartPhones*. Se define como *smartphone* o *sphone* al dispositivo inalámbrico portátil, y que cabe en la palma de la mano, que integra la funcionalidad de un celular y una PDA.

### 2.2.2 Equipos Dedicados.

En una red alámbrica pueden existir equipos dedicados a la conexión de las computadoras, estos equipos son obligados a fin de garantizar un servicio de conectividad, compartir datos, o bien, proporcionar seguridad en el entorno de trabajo.

Ahora bien, en una red WLAN no necesariamente ocurre lo anterior, esto es, no se requiere un equipo dedicado para que se establezca la conectividad, pues esta se puede obtener a través de otra terminal, con esto se compartirían los recursos de la terminal *huésped*.

Sin embargo, para que exista un correcto y seguro funcionamiento, se requieren de ciertos elementos que encontramos en una LAN, y estos son los llamados equipos dedicados, principalmente se requieren dos tipos de ellos:

- ⊕ *Puertas de Enlace*. Este puede ser un AP, y básicamente sirve como medio de acceso directo a la red, sin este elemento, no podríamos acceder a Internet o bien, no se podría acceder a un segmento alámbrico LAN.
- ⊕ *FireWall*. Este equipo rara vez está presente en los clientes pequeños de una WLAN, sin embargo existe en las grandes empresas que prestan algún servicio de conectividad de datos, como en el caso de las compañías de Internet. Básicamente se compone de una capa de seguridad (Netfilter, IPtables y Detección Intrusos), NAT (Network Address Translation) y Reglas de Enrutamiento.
- ⊕ *Puentes Inalámbricos*. Este tipo de equipos dedicados rara vez existen en WLAN de corto tamaño, sin embargo, juegan un papel importante en *Wireless Wide Area Network (WWAN)* para evitar pérdidas de conectividad al cambiar de AP y problemas de encaminamiento, asimismo, se llegan a usar repetidores en el caso de mayor tráfico e interferencias.

### 2.2.3 Access Point.

Las redes inalámbricas, en sus diferentes topologías, se conectan a la red mundial a través de un punto de acceso o AP. Su principal función es fungir como un punto centralizador, además de ser un gestor de tráfico de todos los equipos que dependan de él.

De acuerdo al estándar IEEE 802.11, un AP puede proporcionar servicio de acceso a 30 usuarios, en un radio de 100[m], sin embargo, ambos parámetros pueden variar de acuerdo al lugar en donde se ubique; ya que factores como la altura existente entre el techo y el piso, obstáculos cercanos de gran tamaño, interferencia producida por otros dispositivos, tipos de antena, potencia de salida, entre otros.

El AP generalmente se utiliza como un puente que permite la conexión entre dos interfaces, la inalámbrica y la alámbrica (representada por Ethernet); esta forma de operación, trabaja al nivel de capa Media Access Control (MAC), ya que con solo manipular los encabezados MAC, es posible enviar, filtrar, etc., los paquetes emitidos.

También es posible emplearlos como repetidores, esto se realiza con la finalidad de extender su cobertura por medio del re-envío de información; su principal desventaja reside en la pérdida de eficiencia, ya que puede disminuir hasta un 50[%]. Para evitar esta



pérdida, es posible utilizar una topología de red tipo *lily-pad*, la cual consiste en colocar una serie de APs dentro de un área extensa, de tal forma que cada uno se conecta a una red diferente.

Asimismo, pueden ser utilizados como enrutadores y/o puertas de enlace, dada su capacidad de enrutamiento de los paquetes a nivel de capa 3 y 4 del modelo de referencia OSI. Sin embargo, la mayor parte de ellos, solo son utilizados como una ruta a seguir para conectarse a la red mundial, red de acceso o red local, proviniendo de una red inalámbrica.

Finalmente, otra de sus aplicaciones menos utilizadas es como interventor, para este caso, su función primordial es designar el instante en que los dispositivos que conforman la red inalámbrica puedan transmitir. A pesar de presentar esta alternativa, la mayoría de las redes que operan bajo el protocolo IEEE 802.11 no lo aplican, porque prefieren utilizar algoritmos pseudo aleatorios distribuidos que pueden cumplir con las mismas funciones y presenten mayor simplicidad al momento de la implementación.

El bajo costo de este tipo de equipos, así como la facilidad de instalación y ventajas de movilidad sobre las redes alámbricas, les has permitido gran aceptación entre la población de las grandes ciudades. En el contexto industrial y comercial, las redes alámbricas han tenido un gran impacto en el desarrollo de las operaciones, ya que los empleados pueden tener acceso a dispositivos inalámbricos portátiles, como son los PDT, permitiéndoles actualizar y desarrollar sus actividades laborales en tiempo real y por consecuencia, con mayor eficiencia.

En resumen, los APs pueden soportar tres modos generales de operación:

- ⊕ *Modo Root*: El AP es conectado a la red cableada y todos los dispositivos inalámbricos acceden a la red y mantienen comunicación entre ellos vía el AP; este modo es la configuración inicial de la mayoría de los APs (Figura 2.2).

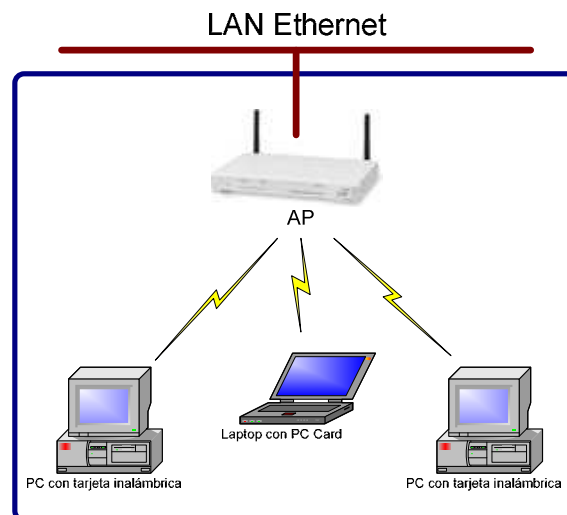


Figura 2.2 AP en Modo Root.



- ✦ **Modo Bridge:** Permite la interconexión inalámbrica de dos redes cableadas, este modo solo puede implementarse en APs prediseñados para soportarlo, lo que los hace significativamente más caros (Figura 2.3).

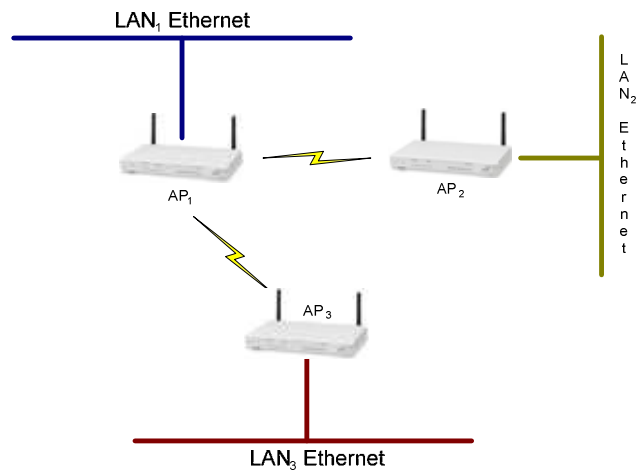


Figura 2.3 AP en Modo Bridge.

- ✦ **Modo Repetidor:** Un AP configurado en este modo presenta un funcionamiento intermedio entre los usuarios de una red inalámbrica y un AP en modo root, es decir, permite que el área de cobertura se amplíe; sin embargo, al momento de la implementación, el *throughput* se reduce significativamente, por lo que no se recomienda, a menos que sea muy necesario (Figura 2.4).

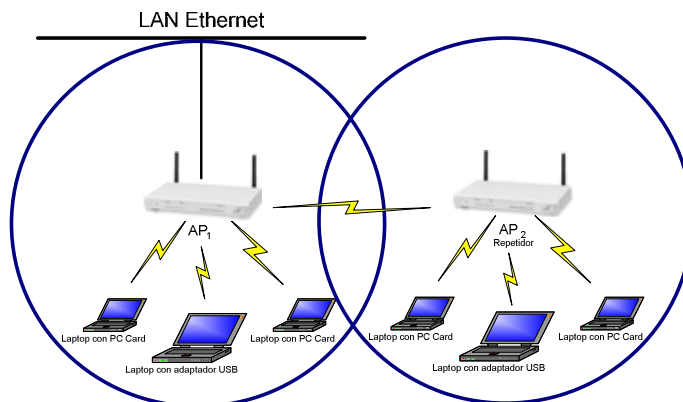


Figura 2.4 AP en Modo Repetidor.

## 2.2.4 Antenas.

El envío inalámbrico de datos y voz, necesita que tanto el receptor como el transmisor estén provistos de una antena. La selección apropiada de la antena tendrá una gran influencia en el desempeño del sistema, ya que de esta dependerá su calidad.

A continuación se realizará una descripción general de los tipos de antenas que se utilizan en los dispositivos confortantes de una WLAN.

### 2.2.4.1 Antenas Direccionales.

A pesar de que la mayor parte de las WLANs utilizan antenas omnidireccionales, WLAN puede ser compatible con las de tipo direccional. Una de sus ventajas principales reside en que favorecen la recepción de la señal en ciertos ángulos y con un ancho del haz variable. Para un haz angosto, la ganancia es mayor; ya que elimina la mayor parte de las interferencias que se encuentran en otras direcciones.

Con este tipo de antenas, es posible tener un alcance de algunos kilómetros para dispositivos que operen en la banda de ISM; siempre y cuando, cuenten con LOS. Una de las desventajas que se puede encontrar al utilizar este tipo de antenas, es que deberán estar posicionadas de tal forma, que una de ellas apunte a la otra y viceversa. De acuerdo al ancho del haz, será el error permitido; ya que entre más angosto sea el haz, la precisión al momento de posicionarlas deberá ser mayor. Por otra parte, entre más directivas son, sus dimensiones aumentan; lo que dificulta su colocación y mantenimiento.

Es por ello, que las antenas direccionales solo son utilizadas en enlaces punto a punto, como es el caso de puentes inalámbricos, también denominados *Wireless Bridge (WB)*, por sus siglas en inglés. Para la mayoría de las redes, en donde los nodos necesitan comunicarse entre ellos al momento que pueden o no cambiar de posición; las antenas omnidireccionales son más prácticas y fáciles de implementar.

### 2.2.4.2 Antenas Sectoriales.

Estas antenas, poseen características similares a las que presentan las direccionales; ya que una está conformada por un conjunto de direccionales, siendo ensambladas verticalmente, de tal forma que cada una cubra un sector hasta completar 360[°].

De tal manera que cuando se desea recibir la señal de un nodo específico, bastará que la estación base seleccione el sector de la antena, que cubra la región deseada; por otra parte, resulta ser altamente directiva sin necesidad de sacrificar cobertura.

Actualmente, se encuentra bajo investigación un tercer tipo de antena, denominada *beam forming*; la cual consiste en utilizar un grupo de antenas direccionales junto con un interferómetro, con el propósito de mejorar la calidad de la señal y al compensar los retardos de la propagación; de esta forma es posible apuntar el sistema hacia una dirección muy específica y obtener los mismos beneficios de las antenas direccionales; con la ventaja principal de ser adaptable y dinámico; resultando una propuesta viable para WLAN.

### 2.3 Arquitectura Física.

De acuerdo al estándar 802.11 la configuración básica de arquitectura física de una WLAN la podemos ver en la figura 2.5.

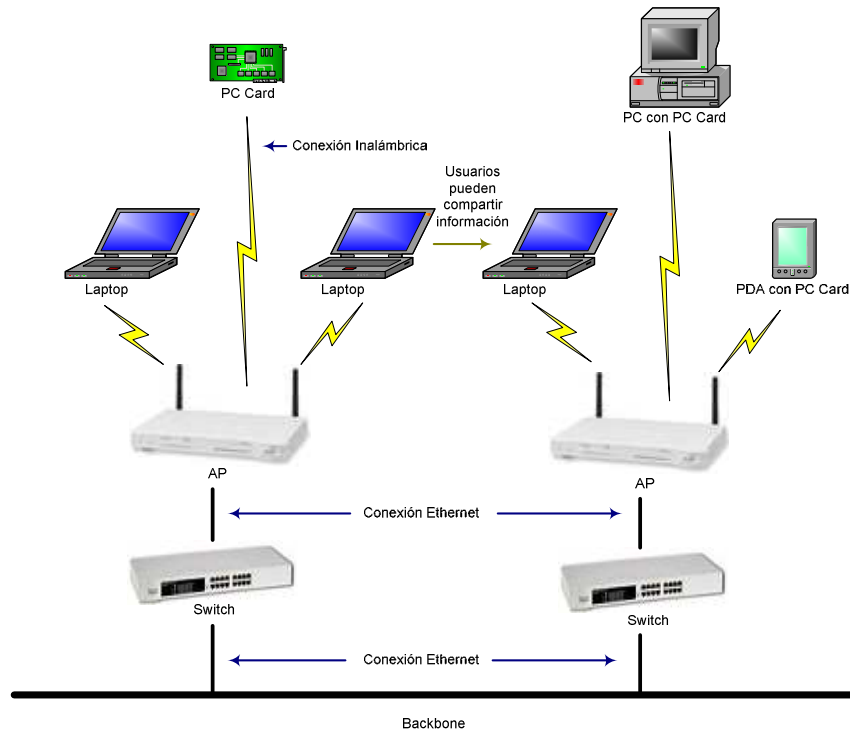


Figura 2.5 Arquitectura Física de una WLAN.

A partir de esta imagen encontramos que se conforma principalmente de al menos una estación *wireless* y dispositivos transmisores/receptores (transceptores), que como veíamos, son los denominados APs, que actúan como un puente entre las redes *wireless* y las cableadas, además de ser una estación base para la red *wireless*, agregando acceso para múltiples estaciones *wireless* a la red cableada. Éstos, a su vez, se enlazan a través de una conexión tipo *Ethernet* a *switches* que les brindaran acceso a la red mundial o a cualquier otra predefinida.

Los dispositivos de los usuarios finales de WLAN acceden a través de adaptadores que proveen de una interfaz entre el sistema operativo de la red del cliente (NOS) y las ondas que son recibidas en la antena; en caso de que no los contengan previamente, éstos les son implementados, tales como PC Cards, *Universal Serial Buses (USBs)* y PCI. Cabe mencionar que la naturaleza de las conexiones inalámbricas es transparente al NOS.

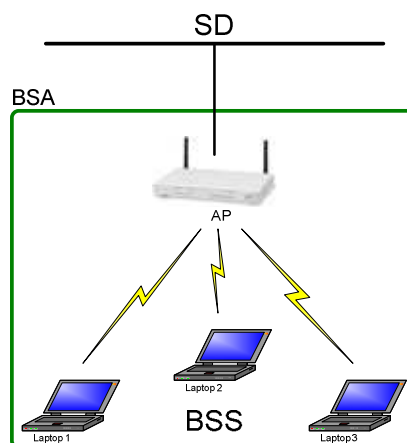
## 2.4 Topologías.

Dentro de la arquitectura, se puede definir el modo en que se llevará a cabo el funcionamiento y conexión entre los APs y los usuarios. Para esto, es necesario definir una serie de topologías que ayudarán a cumplir el objetivo.

Tomando en cuenta que la arquitectura física del estándar IEEE 802.11 es modular, se puede partir de un bloque básico o mínimo denominado BSS (Conjunto básico de servicios), el cual es un modo de operación que define un espacio geográfico en el cual una serie de dispositivos se interconectan por una interfaz de aire.

Cada BSS está controlada por medio de un AP, el cual tiene la función de ser la puerta de enlace entre la red inalámbrica y la cableada. Además, posee la capacidad de gestionar el tráfico de las estaciones móviles, así como mantener la comunicación con otras celdas y/o redes. El BSS, es por tanto, una entidad independiente que puede vincularse con otros BSS a través de los AP mediante un *sistema de distribución (DS)*, ya sea de forma alámbrica o inalámbrica; esto dependerá de la forma en que se comunica con los demás BSS.

La interacción de estos dispositivos tiene lugar dentro de un área determinada, llamada *Basic Service Area (BSA)*, si un dispositivo móvil no se encuentra en esta, no podrá comunicarse con los otros miembros del BSS. En la figura 2.6 se puede apreciar un ejemplo de ambos conceptos.



*Figura 2.6 Ejemplo del modo de operación BSS y BSA.*

Debido a que un BSS es el concepto más básico de operación, se pueden tener las siguientes variantes:

- ⊕ *Independent Basic Service Set (IBSS).*
- ⊕ *Ad-Hoc.*
- ⊕ *Infraestructura.*
- ⊕ *Extended Service Set (ESS).*

A continuación se realizará una descripción general de IBSS y ESS, los modos restantes se explicarán en temas posteriores debido a la importancia que presentan en este trabajo.

- ✦ *IBSS*, también llamado BSS independiente, es un módulo inalámbrico que no presenta DS, por lo que no puede interactuar con otras redes. Los dispositivos móviles se comunican directamente entre sí, sin necesidad de utilizar un AP; esta configuración es útil cuando se requiere instalar una nueva red privada y sin conexiones alámbricas. (Figura 2.7)

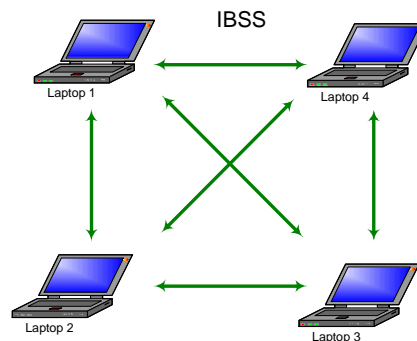


Figura 2.7 Ejemplo del modo de operación IBSS.

- ✦ *ESS*, es un conjunto de dos o más BSS conectados a través de un AP, los cuales a su vez, se encuentran interconectados mediante el SD y así pueden formar una sola red.

Cada ESS tiene un identificador de red inalámbrica conocido como *Service Set Identifier (SSID)*, el cual debe ser el mismo para todos los APs que conforman el ESS (Figura 2.8).

Teniendo en cuenta que la mayoría de las WLAN exhibe la necesidad de conectarse a las redes LAN, este resulta ser un modo de operación eficiente, por lo que es generalmente adoptado al momento de estructurar una nueva red.

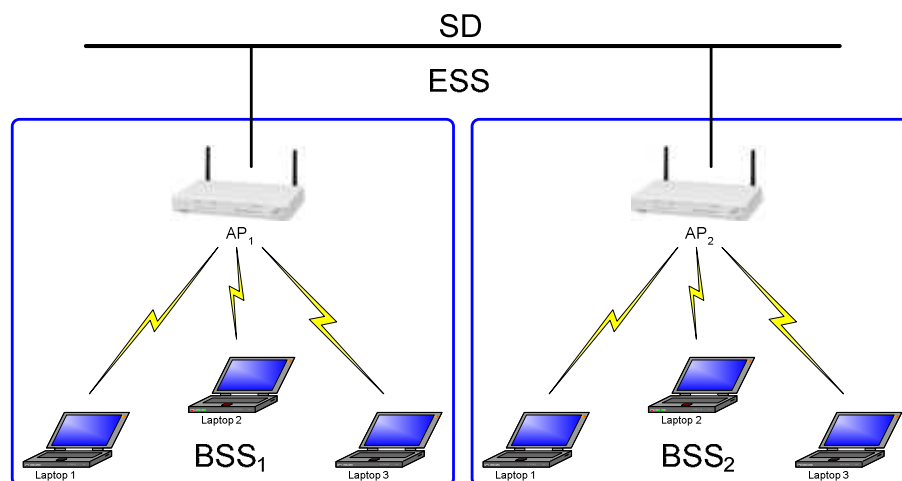


Figura 2.8 Ejemplo del modo de operación ESS.

### 2.4.1 Redes Ad-hoc.

Las redes inalámbricas basadas en el modo de operación *Ad-hoc*; como la configuración que se muestra en la figura 2.9; también reciben el nombre de *Peer-to-peer* o *Point-to-point*; generalmente son redes de área local independiente que se componen de un grupo de dispositivos inalámbricas regidos bajo el estándar 802.11, que permite la interconexión grupal de manera directa vía señales RF, sin necesidad de utilizar un AP o conexión a una red cableada. Para obtener una cobertura total, es preciso que la información distribuida a través de la red, llegue a todos los usuarios.

Para que la comunicación se lleve a cabo, es necesario que todo el grupo de dispositivos tengan configurado el mismo canal y SSID en modo *Ad-hoc*, además es preciso contar con un identificador único que caracterice a cada unidad, es decir, una dirección MAC (48 bits), el cual es generado de manera aleatoria por el adaptador inalámbrico.

Cuando el adaptador inalámbrico es puesto en operación, entra en un periodo de detección, en el cual, realiza la búsqueda de algún usuario que se encuentre activo en ese momento, creando la posibilidad de anexarse a esa red; en caso de no ser así, crea una sub-red para que otros usuarios puedan comunicarse con él.

Este modo de operación, resulta muy útil al momento de montar una WLAN en espacios geográficos que no posean una infraestructura *wireless* o no se requieran servicios avanzados de valor agregado, además de lugares donde el acceso a la red cableada se encuentre restringido.

La diferencia principal entre IBSS y Ad-hoc, es que el primero es la configuración en la que se basa el segundo.

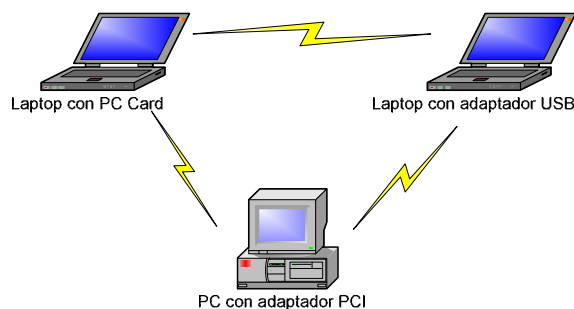


Figura 2.9 Ejemplo de una red Ad-hoc.

En la siguiente tabla se resumen algunas de las ventajas y desventajas que presenta esta topología.

RED AD-HOC	
VENTAJAS	DESVENTAJAS
Configuración sencilla	Alcance limitado
Conexión <i>point-to-point</i> sin utilizar un elemento intermedio.	No se integra a otras redes existentes
Instalación rápida y a bajo costo	Número de usuarios limitado

Tabla 2.1 Ventajas y desventajas de las redes Ad-hoc.

### 2.4.2 Redes Infraestructura.

En el modo de infraestructura definido por el 802.11, los usuarios de una WLAN se conectan al menos a un AP que servirá para gestionar y coordinar el tráfico entre los usuarios, así para que un dispositivo inalámbrico pueda comunicarse con otro dentro de la misma red, es necesario enviar la información al AP para que este realice el re-envío. La cobertura que proporcione el AP, definirá el área de servicio en la que operará este modo, un ejemplo de este tipo de red se visualiza en la figura 2.10.

La comunicación entre varias redes puede ser posible a través de un DS y cada uno de los APs correspondientes a cada BSS, es decir, se necesita constituir un ESS.

Al igual que en las redes *Ad-hoc*, es importante contar con un identificador único para cada BSS, en el caso de las redes infraestructura, se asigna una cadena de longitud máxima de 32 caracteres que se denomina SSID y un canal en el cual operará. Por otra parte, si se desea restringir el acceso a la red, los dispositivos móviles deberán ser configurados con el SSID que corresponda a su BSS; en caso contrario, existe la posibilidad de que un dispositivo elija una BSS y adopte su SSID para conectarse a ella.

Por su facilidad de integración a redes de cableadas ya existentes, este modo de operación se utiliza en la mayoría de las redes de área local inalámbricas corporativas.

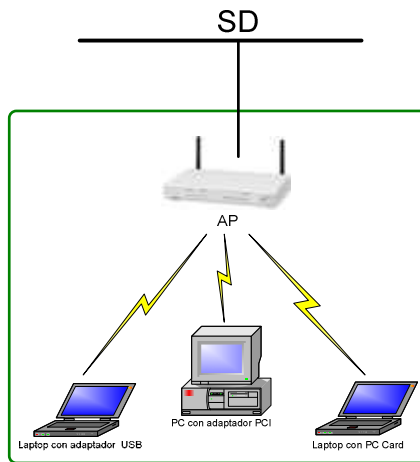


Figura 2.10 Ejemplo de una red de Infraestructura.

En la siguiente tabla se resumen algunas de las ventajas y desventajas que presenta esta topología.

RED INFRAESTRUCTURA	
VENTAJAS	DESVENTAJAS
Los dispositivos no requieren <i>LOS</i> para comunicarse con otros.	Número de usuarios limitado.
Posibilidad de integración en redes alámbricas.	Costo considerable del equipo.
Instalación rápida y a bajo costo.	Cobertura limitada a la proporcionada por el AP.

Tabla 2.2 Ventajas y desventajas de las redes de infraestructura

## 2.5 Roaming.

Uno de los servicios que proporciona WLAN, es la posibilidad de realizar *roaming* entre los APs de un ESS, siendo análogo a la telefonía celular, un dispositivo inalámbrico puede desplazarse de un BSS a otro sin perder continuidad en los procesos que se encontraba realizando, lo cual refuerza la idea de utilizar una red inalámbrica, dado que no se tienen limitaciones de movimiento en el interior del área de cobertura.

Conforme un dispositivo móvil sale del área de cobertura de un AP y se aproxima a otro, el nivel de la señal con respecto al primero disminuye, al mismo tiempo que se incrementa cuando se aproxima al segundo. Estas variaciones en la señal, permiten el cambio de un AP a otro, tal y como se observa en la figura 2.11.

Con el propósito de evitar la interferencia entre los APs, a cada uno se deberá asignar un canal en donde puedan operar. Si los canales se encuentran correctamente distribuidos, se pueden reutilizar evitando la interferencia entre ellos.

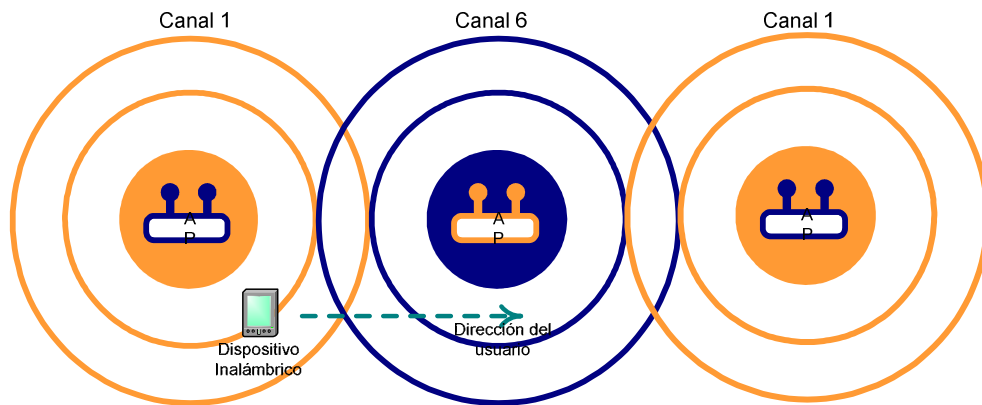


Figura 2.11 Reutilización de canales.

El *roaming* también puede llevarse a cabo, si existe de por medio una negociación entre los APs para compartir o balancear su carga, un ejemplo de esta modalidad se aprecia en la figura 2.12.

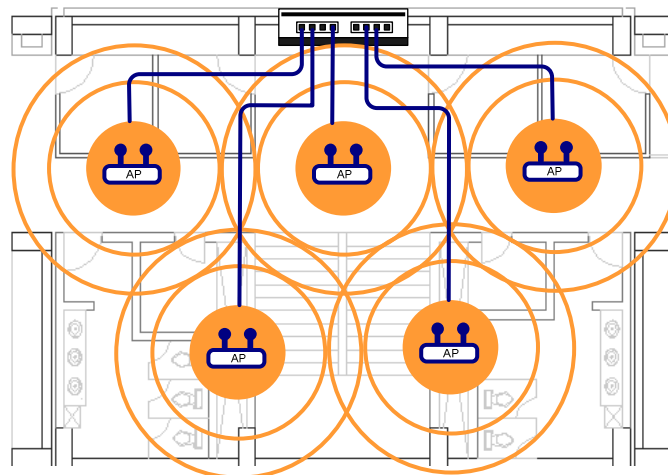


Figura 2.12 Ejemplo de Roaming



Dado que en una WLAN los recursos se comparten, hay que tomar en cuenta que a mayor número de usuarios, menor cantidad de recursos habrá para cubrir sus necesidades. Es por ello, que los APs evaluarán factores como el número de clientes asociados, calidad e intensidad de la señal y tráfico al momento de transferir a un usuario a otro AP, sin importar si el usuario se encuentra estático, así como se muestra en la figura 2.13.

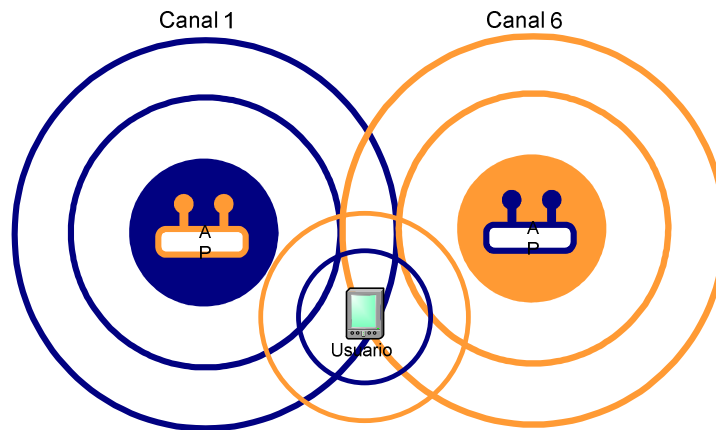


Figura 2.13 Roaming de un usuario estático.

El protocolo que hace posible el *roaming* se denomina *Inter-Access Point Protocol (IAPP)*, su primera versión fue diseñada y publicada por Airones (antes Cisco), Lucent Technologies y Digital Ocean. Actualmente forma parte del estándar 802.11 tomando el nombre de 802.11f y permite la interoperabilidad entre los equipos de los diversos fabricantes que se encuentran en el mercado.

En resumen, para que el *roaming* pueda ser realizado, es necesario que los APs cumplan con las siguientes características:

- ✦ Deben estar conectados a la misma sub-red, de tal manera que un usuario no necesite cambiar de la dirección *Internet Protocol (IP)* que se le asignó en un principio.
- ✦ Tener configurado el mismo SSID para identificar de manera única a la sub-red a la cual pertenecen.
- ✦ Presentar los mismos esquemas de autenticación y encriptación, a modo de que sean del conocimiento del usuario, para que este pueda utilizarlos.

## **2.6 Los estándares 802.11x.**

### **2.6.1 Introducción.**

Publicado en junio de 1997, el 802.11 es un estándar del protocolo de comunicaciones de la IEEE que define el uso de las dos capas más bajas del sistema de referencia OSI (física y de enlace de datos), define las normas de funcionamiento para las redes inalámbricas de área local.

En un inicio, las tasas de transferencia definidas por el estándar eran demasiado bajas (1 a 2[Mbps]) y no eran suficientes para suplir las necesidades que de los usuarios, por lo que la demanda de las WLANs disminuyó notablemente.

Preocupada por el desempeño del nuevo estándar, en septiembre de 1999 la IEEE ratifica el estándar 802.11b para transmisiones de 5 y hasta 11[Mbps], operando en la banda ISM de 2.4 [GHz] y el estándar 802.11a para transmisiones de 54[Mbps] como máximo y que opera en las bandas de la *Unlicensed National Information Infrastructure (UNII)* de 5.3 [GHz] y 5.8 [GHz]. Con la aplicación de esta medida logra un mejor desempeño y por consecuencia una mayor aceptación en el mercado.

Cabe mencionar que la arquitectura básica y servicios del estándar 802.11b son basados en el estándar 802.11 ya existente, por lo que las especificaciones del estándar tienen afección únicamente a la capa física, al añadir una conectividad más robusta y velocidades mayores.

Desafortunadamente, las nuevas dos especificaciones eran incompatibles, la causa: operaban en diferentes frecuencias. Esto significa que las *Network Interface Cards (NICs)* y APs pertenecientes a una red bajo el estándar 802.11a no podían comunicarse con los elementos de otra red que estuviera regida por el 802.11b. Esta incompatibilidad obligó a la creación de un nuevo estándar conocido como 802.11g, el cual soporta hasta 54[Mbps] y dado que opera en la misma banda de frecuencia que el 802.11b, resulta ser interoperable con las redes y dispositivos ya existentes.

Posteriormente se crearon otros estándares del 802.11, que permitieron la ampliación progresiva de la cobertura, además de proporcionar una mayor seguridad durante la transmisión de la información con el propósito de brindar *Quality of Service (QoS)*.

A continuación se realizará una descripción más detallada de cada uno de los estándares anteriormente expuestos.

## 2.6.2 El estándar 802.11.

En 1997, la IEEE aprobó este estándar para definir el funcionamiento e interoperatividad de las redes LAN inalámbricas (*Ad-hoc* e infraestructura) a través de las especificaciones proporcionadas para el uso de los dos niveles más bajos de la arquitectura OSI, en el interior de esta última, se encuentra la sub-capa MAC (figura 2.14), que utiliza un estándar de difusión de la señal por *Direct-Sequence Spread Spectrum (DSSS)*, contemplando la opción de FHSS que limita su velocidad de transmisión a 2 [Mbps] debido a las restricciones establecidas por la FCC.

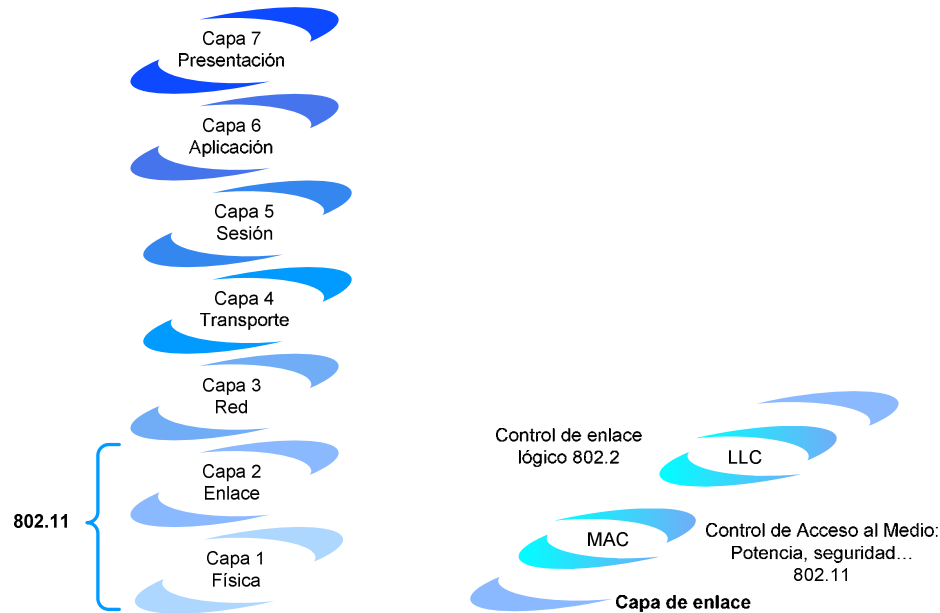


Figura 2.14 El 802.11 y modelo OSI

El método de acceso al medio que utiliza es el *Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA)*, la dificultad en detectar la portadora en el acceso WLAN consiste básicamente en que la tecnología utilizada se basa en espectro disperso con acceso CDMA, lo que conlleva a que el medio radioeléctrico sea compartido, ya sea por DSSS o FHSS. El acceso por código CDMA implica que pueden coexistir dos señales en el mismo espectro utilizando códigos diferentes, y eso para un receptor de radio implicará que detectaría la portadora inclusive con señales distintas de las de la propia red WLAN.

Las tecnologías que se basan en este estándar permiten a los administradores crear nuevas redes que sean compatibles con otras tecnologías para satisfacer sus necesidades, ya que funcionan con cualquier aplicación LAN, en red o protocolo, incluyendo *Transmission Control Protocol/Internet Protocol (TCP/IP)* y Novell Netware corren sobre 802.11 de la misma forma que lo hacen sobre *Ethernet*.

Por otra parte, el estándar 802.11 define dos componentes, una estación *wireless*, que normalmente es una PC equipada con una tarjeta de red (NIC) y un Punto de Acceso (AP), que actúa como un puente entre las redes *wireless* y las cableadas, como ya se mencionó en el tema correspondiente.

Esta especificación está en desuso, ya que gradualmente fue sustituida por otros estándares mejorados a partir de éste.

### 2.6.3 Evolución del estándar 802.11.

El estándar 802.11 a través del tiempo ha sufrido mejoras que se han plasmado en variantes del estándar original, cada una de ellas identifican a través de una letra del alfabeto. A continuación se realizará una breve pero bien caracterizada descripción de cada una de ellas.

#### 2.6.3.1 802.11a.

También conocido como Wi-Fi5 y siendo una evolución del 802.11, fue ratificado en 1999 por la IEEE al mismo tiempo que el 802.11b. Operando en la banda de frecuencia de 5 [GHz], es decir de (5.150 a 5.350) [GHz] y de (5.725 a 5.825) [GHz], se encuentra aislada de interferencias producidas por otras tecnologías. En este estándar se definen 8 canales para la primera banda y 4 para la segunda.

Utiliza el estándar de difusión *Orthogonal Frequency Division Multiplexing (OFDM)* como técnica de gestión de frecuencias; este método consiste en dividir una señal de información de alta velocidad en múltiples señales de información y transmitir las en paralelo utilizando portadoras ortogonales de baja capacidad con la misma amplitud y fase, moduladas a su vez, utilizando *Phase Shift Keying (PSK)* o *Quadrature Amplitude Modulation (QAM)*.

La ortogonalidad de esta técnica, se debe a la relación matemática precisa entre las frecuencias portadoras, las cuales son linealmente independientes.

Dentro de las ventajas que presenta este estándar de difusión, se pueden enunciar las siguientes:

- ⊕ Proporciona una mayor robustez contra la interferencia de banda angosta.
- ⊕ La velocidad de transmisión puede llegar hasta 54 [Mbps] y ser aplicable en entornos con un elevado requerimiento de ancho de banda, tales como la transmisión de audio y video digital, *Voice over Internet Protocol (VoIP)*, etc.
- ⊕ Alta eficiencia espectral (la ortogonalidad de las portadoras permite el traslape del espectro).
- ⊕ Menor distorsión por propagación multitrayectoria (dado que las múltiples señales se transmiten a una velocidad menor, el ancho de los pulsos transmitidos es mayor, permitiendo que el retardo no afecte al receptor).
- ⊕ Permite un máximo de 12 canales sin sufrir traslape, lo que permite un mayor número de usuarios.

A pesar de todas las ventajas que presenta, existe un mayor nivel de consumo de energía con respecto a otros estándares, por lo que el costo aumenta y no resulta atractivo para su implementación en computadoras portátiles o PDAs.

En Europa, algunos países utilizan la banda de 5[GHz] para cuestiones militares o el desarrollo de otras tecnologías.

Presentando una mayor vulnerabilidad a los obstáculos, su cobertura en interiores se ven reducidas significativamente y debido a que se manejan mayores frecuencias, la atenuación es mayor.

En la tabla 2.3 se realiza una pequeña comparación entre la velocidad de transmisión, la distancia con respecto al AP de acuerdo al medio en el que se encuentra.

<b>Medio</b> <b>Tasa de TX</b>	<b>Interior</b>	<b>Exterior</b>
<b>54 [Mbps]</b>	12 [m]	30 [m]
<b>6 [Mbps]</b>	90 [m]	300 [m]

*Tabla 2.3 Comparación de Medio vs Tasa de TX.*

Como puede observarse, el alcance del sistema y la velocidad de transmisión varían de acuerdo a diversos parámetros, sin embargo estos pueden ser controlados a través de los parámetros de la modulación, como la tasa de codificación, los bits por portadora, etc.

### 2.6.3.2 802.11b.

Este estándar es una variante del IEEE 802.11 también conocido como Wi-Fi o *Wireless Ethernet* que define los niveles físico y de acceso al medio (MAC); creado por un consorcio de empresas fabricantes como Ericsson, IBM, Nokia, Motorola e Intel, y ratificado en 1999 como la extensión del estándar original, de manera que las especificaciones de este afectan únicamente a la capa 1 del modelo OSI añadiendo velocidades mayores y una conectividad más robusta. Opera en la banda ISM de 2.4 a 2.4835 [GHz] que no requiere licencia en EUA.

A diferencia de su antecesor, en general utiliza la técnica de transmisión DSSS que le permite ofrecer hasta 11 [Mbps] utilizando 3 canales de 22 [MHz] de ancho de banda sin que sufran algún traslape y su reducido nivel de consumo lo hace una buena opción para aplicaciones en computadoras portátiles o PDAs por su bajo costo.

Para incrementar la tasa de datos de este estándar, se desarrollaron técnicas de codificación avanzadas como CCK<sup>5</sup>, que consiste en un conjunto de 64 palabras código de 8 bits. Como conjunto, estas palabras código tienen propiedades matemáticas únicas que les permiten distinguirse correctamente una de otra por un receptor incluso en presencia de un ruido significativo e interferencia ocasionada por multirayectorias.

Todas estas ventajas han permitido su rápida aceptación y expansión dentro de las WLANs, ya que un poco más del 70[%] de los productos operan bajo este estándar.

La cobertura que presenta dependerá del número de usuarios conectados y del lugar en donde opere; tal y como se muestra en la tabla 2.4.

<b>Medio</b> <b>Tasa de TX</b>	<b>Interior</b>	<b>Exterior</b>
<b>11 [Mbps]</b>	30 [m]	120 [m]
<b>1 [Mbps]</b>	90 [m]	460 [m]

*Tabla 2.4 Comparación de Medio vs Tasa de TX.*

---

<sup>5</sup> La modulación CCK utiliza un conjunto de funciones complejas conocidas como *códigos complementarios* con el fin de aumentar la cantidad de información transmitida, por lo que reemplaza al código Barker.

Dentro de sus principales desventajas se tienen mayor número de fuentes de interferencia como teléfonos inalámbricos y hornos de microondas que operan en las mismas frecuencias. A diferencia del 802.11, no es compatible con los sistemas que utilizan FHSS.

### 2.6.3.3 802.11g.

El estándar 802.11g, basándose en la norma del 802.11b fue aprobado a mediados del 2003. Opera sobre la misma frecuencia del 802.11b y es capaz de utilizar a DSSS y OFDM; por este hecho, resulta ser compatible con el 802.11a al mantener una velocidad de transmisión de 54[MHz] y características propias del 802.11b en cuanto a niveles de consumo y número de canales que no sufren traslape.

Dentro de otras de las ventajas más importantes que presenta este estándar, se pueden enunciar las siguientes:

- ⊕ Cobertura de zonas amplias.
- ⊕ Compatibilidad con la infraestructura del estándar 802.11b.
- ⊕ Menor absorción y atenuación de la señal en la banda de 2.4 [GHz].

Sin embargo, al igual que el 802.11b, es vulnerable a la interferencia de otros dispositivos que trabajan en la misma banda además que el alcance teórico de 802.11g no es aprovechable cuando la densidad de usuarios o el tráfico generado es alto o cuando existen problemas de interferencia entre redes cercanas.

### 2.6.3.4 Otras variantes del 802.11.

Las tres versiones del 802.11 (a, b y g) tienen problemas de seguridad, carencia de calidad de servicio y limitaciones de movilidad. Varios grupos del IEEE están trabajando en soluciones para ellos. Estos grupos, denominados como los estándares actuales WLAN IEEE, pero no pretenden definir estándares de WLAN alternativos a los tres citados, sino solo extensiones de los mismos para solucionar los problemas anteriores.

El estándar 802.11b permite cifrar la información con un procedimiento llamado *Wired Equivalent Privacy (WEP)*, sin embargo este presenta cierta vulnerabilidad que permite descifrar la información en muy poco tiempo.

Los grupos IEEE 802.11x y 802.11i están trabajando en varias soluciones que permitan ofrecer soluciones interoperables y un patrón robusto que permitan mejorar la seguridad del cifrado y añadan funciones de autenticación, además se están estudiando tanto soluciones compatibles hacia atrás con WEP, como soluciones más robustas basadas en algoritmos nuevos, como el *Advanced Encryption Standard (AES)*. Por su parte, el grupo 802.11e pretende introducir QoS y multimedia en redes 802.11b y 802.11a cambiando el mecanismo de acceso al medio basado en contienda por otro más controlado basado en *Time Division Multiple Access (TDMA)*, con calidad de servicio.

El grupo 802.11f, creó esta especificación que funciona bajo el estándar 802.11g y que se ocupa de definir funciones de traspaso normalizadas que permitan el *roaming*; es decir que permitan a un usuario cambiar de un canal radio a otro o de un punto de acceso a otro sin perder la comunicación.

Otros grupos y actividades en curso son: el 802.11d, el cual está definiendo versiones de 802.11b en otras bandas de frecuencia; el 802.11h, que se ocupa de mejorar el control de potencia transmitida y la selección dinámica de canales de radio y control automático del 802.11a en línea con los requisitos de *European Telecommunications Standards Institute (ETSI)* en Europa; y el 802.11j, el más reciente, ocupado de la coexistencia de 802.11a con el estándar europeo HiperLAN2. El grupo 802.11n trabaja en el estándar que permitirá alcanzar velocidades binarias por encima de los 100 [Mbps], y operar en el rango de frecuencias del 802.11a.

A continuación se presenta la tabla 2.5 que resume los aspectos más importantes de cada uno de los estándares anteriormente citados.

ESTÁNDAR	DESCRIPCIÓN
802.11	1 a 2 [Mbps] en la banda de 2.4[GHz], DSSS y FHSS.
802.11a	Hasta 54 [Mbps] en la banda de 5[GHz], OFDM.
802.11b	Hasta 11 [Mbps] en la banda de 2.4[GHz], DSSS.
802.11c	Configuración de "Wireless Bridge" entre APs.
802.11d	Aspectos reglamentarios en países sin normativa vigente sobre el 802.11.
802.11e	Define niveles de la calidad de Servicio.
802.11f	Roaming entre APs a través de IAPP.
802.11g	Hasta 54 [Mbps] en la banda de 2.4[GHz], DSSS y OFDM.
802.11h	Selección dinámica de frecuencia y control de la potencia transmitida para 802.11a.
802.11i	Desarrollo de mecanismos de seguridad a través de AES.
802.11j	Resuelve la adición del canal 4.9 [GHz] al de 5[GHz] para el estándar 802.11 <sup>a</sup> en Japón.
802.11n	Hasta 100[Mbps]

*Tabla 2.5 Principales características del estándar 802.11x.*

## 2.6.4 Capa física.

### 2.6.4.1 Descripción.

La capa física también conocida como capa *physical (PHY)*, es la interfaz encargada de definir los métodos de difusión de la señal en el medio. Esta a su vez, se divide en dos subcapas denominadas *Physical Medium Dependent (PMD)* y *Physical Layer Convergence Procedure (PLCP)*; la primera es la encargada de la difusión de la señal, mientras que la segunda realiza la conversión de los datos a un formato compatible con el medio físico, añadiendo preámbulo y cabecera del nivel físico.

La definición del 802.11 para el estándar de la capa física incluye, además de la conectividad inalámbrica, las técnicas de difusión utilizadas por los estándares 802.11 y 802.11b que se apoyan en la transmisión de Espectro Disperso, desarrollada por la milicia para la seguridad y confiabilidad de los sistemas de comunicaciones. Este se refiere a los esquemas de señalización que son basados en algunas formas de codificación; lo cual es independiente a la información transmitida; y que utiliza una función matemática para diseminar la energía de la señal sobre un amplio rango de frecuencias para atenuar los efectos de la interferencia sobre los datos transmitidos.



Inicialmente se establecieron tres estándares de difusión para la capa física, que buscaban optimizar la eficiencia espectral:

- ⊕ FHSS.
- ⊕ *Direct Sequence Spread Spectrum (DHSS)*.
- ⊕ *Infrared (IR)*.

Posteriormente en 1999, se agregaron 2 nuevos estándares:

- ⊕ OFDM.
- ⊕ *High Rate DSSS (HR-DSSS)*.

En la tabla 2.6, se detalla el estándar a nivel de capa física que utiliza el 802.11 y sus principales extensiones.

MODELO ISO	MODELO 802.11	TÉCNICA DE DIFUSIÓN DE 802.11					
Capa física	PLCP	OFDM	DSSS	FHSS	IR	HR-DSSS	OFDM
	PMD	802.11a	802.11	802.11	802.11	802.11b	802.11g

*Tabla 2.6 Capa física del estándar 802.11.*

Sin embargo, cada estándar de acuerdo a su velocidad máxima de transmisión tendrá un tipo de modulación y una codificación. A continuación en la tabla 2.7 se presenta el detalle de estos dos parámetros para el estándar 802.11 y su principal extensión, el 802.11b.

ESTÁNDAR	VELOCIDAD DE TRANSMISIÓN [MBPS]	MODULACIÓN	CODIFICACIÓN
802.11	1	BPSK	Barker Sequence
	2	QPSK	Barker Sequence
802.11b	5.5	QPSK	CCK
	11	QPSK	CCK

*Tabla 2.7 Modulación y codificación de acuerdo al estándar 802.11.*

Para poder realizar la transmisión de una señal RF, es necesario definir un método de difusión y un método de modulación de la señal. La modulación consiste en modificar una señal RF (portadora) en amplitud, frecuencia, fase o alguna combinación de éstas, para incorporar la información que se desea transmitir.

Entre mayor es la velocidad a la que se desea transmitir, la técnica de modulación será más compleja. Una vez que se ha modulado la señal y se ha enviado, la función del receptor es sincronizar el código de difusión y demodular la información; los sistemas DSSS son más sencillos de sincronizar, ya que en estos solo se sincroniza el tiempo, mientras que en los sistemas FDSS también se requiere sincronizar la frecuencia.

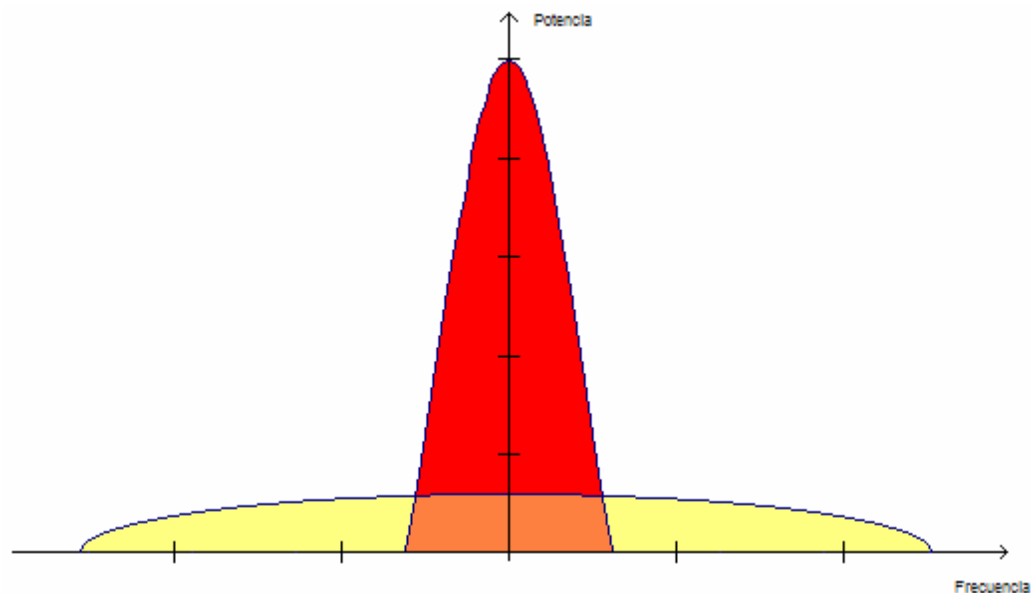


#### 2.6.4.2 Estándares de difusión de frecuencias para Capa I (Tipos de Modulación).

##### ⊕ Espectro Disperso.

Tradicionalmente las comunicaciones de RF buscaban concentrar la máxima cantidad de potencia o energía de una señal en bandas de frecuencia lo más angosto posible, sin embargo el usuario final debía obtener una licencia por parte de la FCC por cada sitio en donde se empleara, además de hacer uso de filtros altamente selectivos y costosos; por lo que se decidió adoptar sistemas de espectro disperso, una representación básica de ellos se observa en la figura 2.15.

Como se ha visto, la mayoría de los sistemas WLAN utilizan la tecnología de espectro disperso para obtener una mayor seguridad, confiabilidad e integridad de la información. Basan su funcionamiento en el consumo de un mayor ancho de banda a diferencia de otros sistemas de banda angosta, además de reducir el nivel de potencia máximo a un nivel constante de alrededor de 100[mW]; permite la disminución de la interferencia en la señal que es producida por el medio. Existen dos tipos de tecnologías que basan su operación en espectro disperso: FHSS y DSSS.



*Figura 2.15 Esquema característico del Espectro Disperso.*

##### ⊕ FHSS.

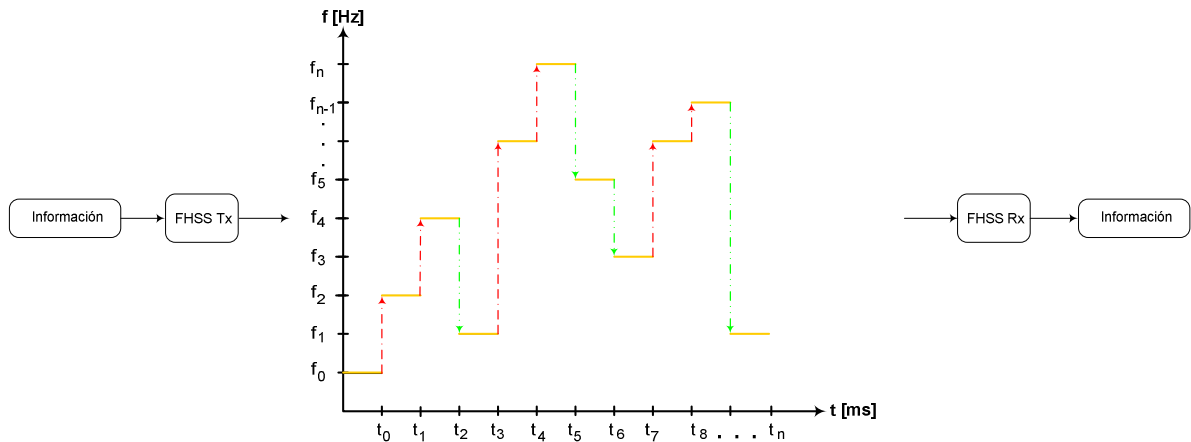
Esta tecnología utiliza una portadora de banda estrecha que cambia la frecuencia en un patrón dado por la codificación y que es conocido tanto por el transmisor como el receptor que se encuentran perfectamente sincronizados. Existen al menos 26 secuencias diferentes que pueden ser definidas o seleccionadas para que una de ellas opere en el transmisor.

El transmisor FHSS es un sintetizador de frecuencia de pseudo-ruido controlado, de manera que la frecuencia de salida instantánea salta de un valor a otro basándose en una entrada pseudo-aleatoria del generador de códigos. Al variar la frecuencia ins-

tantánea se tiene una salida de espectro disperso sobre el rango de frecuencias asignado, tal y como se observa en la figura 2.16.

Utiliza aproximadamente 80 canales de frecuencia, distribuidos en el intervalo de 2.4-2.483[GHz], que facilitan al transmisor los cambios de frecuencia en cada intervalo de tiempo, predefiniendo 400[ms] como máximo y permitiendo la transmisión simultánea de diferentes frecuencias, provenientes de estaciones diferentes, en un mismo intervalo de tiempo.

El FHSS, basándose en la diversidad de frecuencias para combatir la interferencia, es utilizado en aplicaciones a corta distancia con una gran cantidad de receptores distribuidos en el área donde se localice el punto de acceso.



*Figura 2.16 Proceso de transmisión para FHSS.*

#### ⊕ DSSS.

DSSS es la tecnología de la capa PHY más difundida debido a que puede soportar velocidades mayores a 2[Mbps], gracias a que genera un patrón o secuencia de alta velocidad para cada bit transmitido.

Esta secuencia de alta velocidad es la que controla la modulación y al mismo tiempo es la causa directa del esparcimiento del ancho de banda de la señal transmitida, de tal manera que para aquellos receptores que no soporten esta tecnología, la señal parece ser un ruido de banda ancha con una potencia muy baja, esta característica permite que la señal presente una mayor inmunidad con respecto a la que se presenta en una señal de banda angosta.

Por otra parte, entre mayor sea la longitud de la secuencia, la probabilidad de recuperar de manera exitosa la información aumenta.

Para el estándar 802.11b, se tienen definidos 14 canales tipo DSSS, distribuidos uniformemente en el segmento de frecuencias en el que opera, cada uno presenta un ancho de banda de 22[MHz] aproximadamente y una separación entre los inicios de cada canal de 5[Mhz], lo que indica que los canales se encuentran traslapados uno con otro.

En un sistema DS de 11 canales, como el que utiliza Estados Unidos y la mayor parte de los equipos comerciales, solo se disponen de 3 canales (1,6 y 11) que no se trasla-

pan uno con otro (figura 2.17), propiciando que entre ellos no exista interferencia, lo que implica que solo tres puntos de acceso pueden cohabitar en la misma celda. Por otra parte la banda de los 5 [GHz], permite utilizar hasta 8 canales sin interferencia en una misma celda.

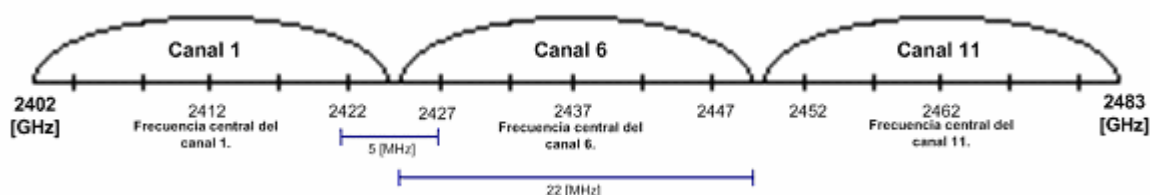


Figura 2.17 Canales sin traslape utilizados por el estándar 802.11.

✦ *Barker Code.*

El estándar 802.11, utiliza el estándar DSSS a nivel de capa PHY y una codificación de tipo *Barker Sequence* también denominada 11bit *chipping*, que es independiente de la velocidad de transmisión y la modulación que utiliza.

En este tipo de codificación, cada bit de información es procesado a través de una función XOR junto con una secuencia numérica pseudo-aleatoria de 11 bits por cada bit de información; dicha representación se observa en la figura 2.18; resultando en un flujo digital de mayor velocidad.

Cuando la secuencia codificada llega al receptor, un filtro correlacionador elimina la secuencia y recupera la información original.

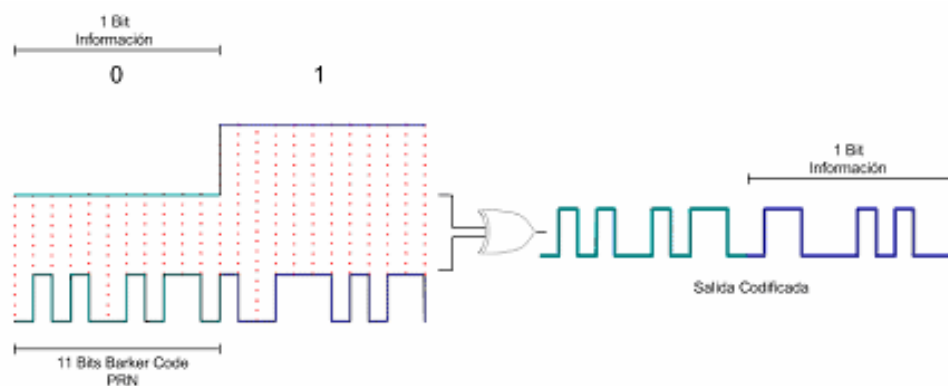


Figura 2.18 Barker Code.

✦ *OFDM.*

OFDM es una técnica de gestión de frecuencias y uno de los factores clave de para los estándares 802.11 en sus extensiones a y g, que a diferencia de su antecesor *Frequency Division Multiplexing (FDM)*, evita el desperdicio de hasta el 50% del ancho de banda disponible en bandas de guarda, sin limitar la velocidad de transmisión a la de de una subportadora (ver figura 2.19).

Ya que en la mayor parte de los sistemas que utilizan FDM, los usuarios son repartidos en una subportadora en particular, de esta forma, la velocidad de transmisión no puede exceder la capacidad de la subportadora asignada, de tal forma que en caso de que una subportadora vecina se encuentre ocupada, no podrá compartir el tráfico con otra que se encuentra disponible, como se aprecia en la figura 2.19.

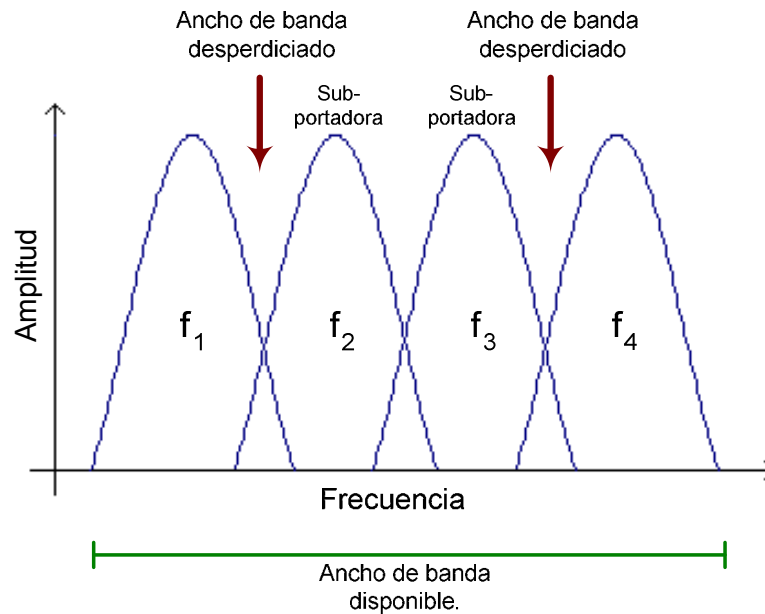


Figura 2.19 La técnica de gestión FDM.

La técnica de OFDM dispersa los datos transmitidos sobre una sucesión de portadoras cada una con una frecuencia única que operan en forma paralela (figura 2.20).

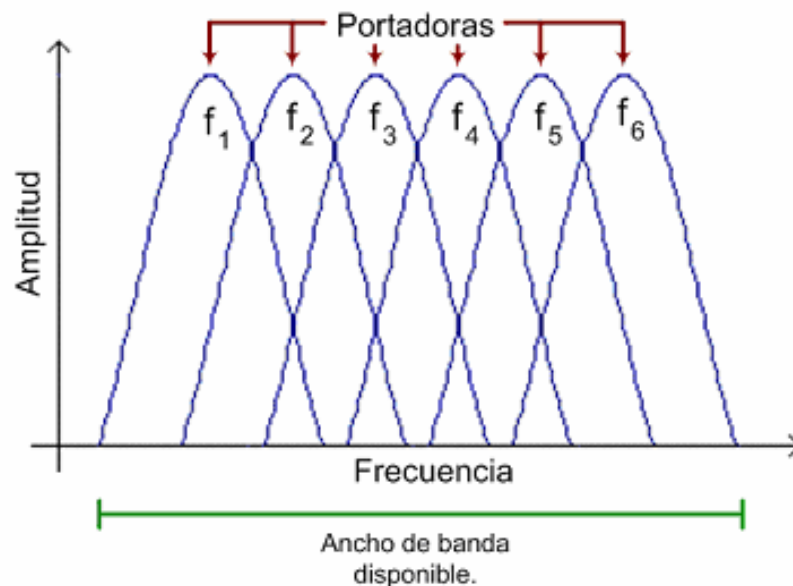


Figura 2.20 Esquema general de OFDM.

La ortogonalidad de esta tecnología, como se había mencionado en un tema previo, es debida a la independencia existente entre las portadoras, ya que cada una de ellas puede ser fácilmente identificada y las bandas de guarda se pueden eliminar (Figura 2.21).

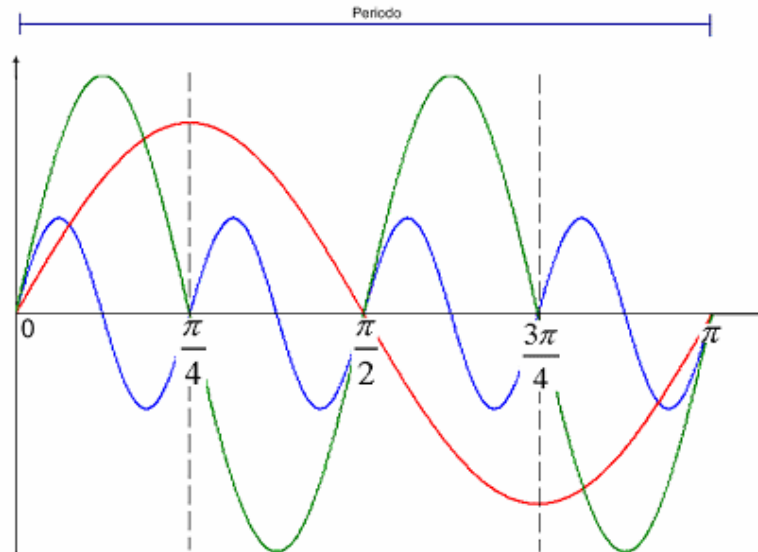


Figura 2.21 Ortogonalidad de OFDM.

Estas características permiten que solo el receptor y demodulador adecuados realicen su función. La combinación de todas estas ventajas, facilitan al sistema una mejor tolerancia al ruido e interferencias de tipo multirayectoria, optimizando el ancho de banda disponible, además de ayudar a disminuir el retraso de la señal que se traduce en una limitante de la velocidad de transmisión.

#### ✦ IR.

La luz infrarroja, tercera tecnología de la capa PHY, es un tipo de radiación electromagnética invisible para el ojo humano, pero al igual que la luz visible, no puede penetrar objetos opacos y necesita de LOS para realizar la transmisión de datos, por lo que se prefiere utilizar en *Personal Area Networks (PANs)* y ocasionalmente en aplicaciones muy específicas de WLAN, sobretodo en sistemas de corto alcance, como la implementación de sub-redes fijas que se limitan a áreas muy pequeñas, como la que es capaz de proporcionar una habitación.

Los sistemas de comunicaciones que basan su intercambio de información en esta tecnología, se dividen en dos categorías:

- ✦ *Haz difuso*: El haz no necesita de LOS, ya que posee la potencia necesaria, como para alcanzar su objetivo a pesar de haber sufrido múltiples refracciones.
- ✦ *Haz directo*: El haz necesita de LOS para poder llegar a su objetivo.

Las ventajas del infrarrojo residen en su bajo costo e inmunidad a la interferencia provocada por sistemas de radio de alta frecuencia, mientras que las desventajas radican en el hecho de que resulta ser impráctica para exteriores por la gran susceptibili-

dad a fenómenos naturales como la lluvia o la niebla, además de cubrir áreas muy pequeñas por su corto alcance.

El estándar 802.11 contempla el uso de infrarrojos, pero nunca se desarrolló debido a los inconvenientes que esta tecnología presenta. No obstante, es el sistema utilizado, no solo por millones de mandos a distancia, sino por millones de equipos electrónicos, como PDAs, teléfonos móviles, etc.

*Infrared Data Association (IrDA)* es una asociación, cuyo objetivo es crear y promover el uso de sistemas de comunicaciones que utilicen infrarrojo, actualmente ha creado dos estándares, los cuales se mencionan a continuación.

- ✚ *IrDa-Data*: Protocolo orientado a establecer redes de corto alcance con velocidades de transmisión en el rango de 9.6[Kbps] a 16[Mbps].
- ✚ *IrDa-Control*: Protocolo de baja velocidad optimizado para ser utilizado en los dispositivos de control remoto inalámbrico.

Las características de ambos protocolos permiten ser compatibles con tecnologías como Bluetooth, además de ofrecer la ventaja adicional de seguridad, ya que los entornos en donde se realiza el intercambio de información, son más privados en comparación con los que utilizan ondas RF.

## 2.6.5 Capa Media Access Control (MAC).

### 2.6.5.1 Descripción.

La capa MAC del 802.11 es similar a Ethernet en el sentido de ambos utilizan un mecanismo que controla el acceso a un medio compartido mediante el principio de *escuchar antes de hablar*, pero a diferencia de un medio cableado, un medio inalámbrico presenta adversidades únicas, como ostentar una mayor susceptibilidad a la interferencia e interceptación, resultando ser menos confiable.

Está diseñada para proveer comunicaciones robustas y seguras sobre un medio inalámbrico a través del monitoreo del medio al momento de transmitir, sin embargo no son sus únicas funciones, ya que también se encarga de todos los procedimientos relacionados con la reservación del medio, formato del frame, control de errores, tareas de autenticación, seguridad, fragmentación y reestructuración de los paquetes de información.

A nivel de esta capa, cuando una estación transmite un frame, este tiene que contener por el medio iniciando con un censado del medio y esperar hasta que el medio se encuentre disponible para poder realizar el envío de información.

Si el frame es de tipo “unicast” y es recibido de manera correcta por la estación destino, esta inmediatamente envía al origen un frame de acuse de recibo, en caso contrario, el destino asume que una colisión o algún otro evento propició la pérdida del paquete de información y selecciona un intervalo aleatorio de espera dentro de un rango constante que es duplicado por cada pérdida experimentada por el sistema, hasta que se alcanza un máximo y es entonces cuando se solicita de nueva cuenta la información perdida. Para transmisiones de tipo multicast o broadcast, no se generan frames de acuse de recibo, ya que otro tipo de mecanismos proveen la protección y control de los frames.

Por otro lado, cuando el medio censado está ocupado al momento de realizar el primer intento de transmisión, se elige un tiempo de espera aleatorio y hasta que finalice se realizará un intento más. Este mecanismo reduce la probabilidad de contención<sup>6</sup> cuando la posibilidad de que ocurra aumenta, es decir inmediatamente después de una transmisión.

Además del mecanismo básico de contención descrito anteriormente, el estándar 802.11 ofrece un método opcional de acceso sin contención, que solo es aplicable a los sistemas al tipo de infraestructura BSS y que actualmente no es muy utilizado, pero se espera que ayude a mejorar el parámetro QoS en los sistemas. En este método, los APs ganan y mantienen el control del medio inalámbrico por largos periodos a través de una portadora de censado virtual, de tal manera que durante los periodos sin contención, el sondeo hecho por el AP es usado para conceder el acceso al medio a las unidades móviles.

La capa MAC también presenta una extensión que especifica el soporte de servicios síncronos y asíncronos, los primeros también conocidos como *Distributed Time Bounded Services (DTBS)* se refieren a la transmisión de voz y video, los cuales generan un tráfico que está limitado por retardos específicos y que bajo el esquema de *Puntual Coordination Function (PCF)*, todas las peticiones del sistema son controladas por un AP.

Por otro lado, los servicios asíncronos como el correo electrónico y la transferencia de archivos, en donde todos los usuarios tienen la misma oportunidad de acceder a la red inalámbrica, definen su transporte bajo un esquema *Distributed Coordination Function (DCF)* que es similar al transporte de paquetes en redes bajo el esquema de "Best effort". DCF es el soporte de PCF y está basado en el mecanismo fundamental de acceso para la capa MAC, es decir, CSMA/CA; cada uno de los niveles se muestra gráficamente en la figura 2.22.

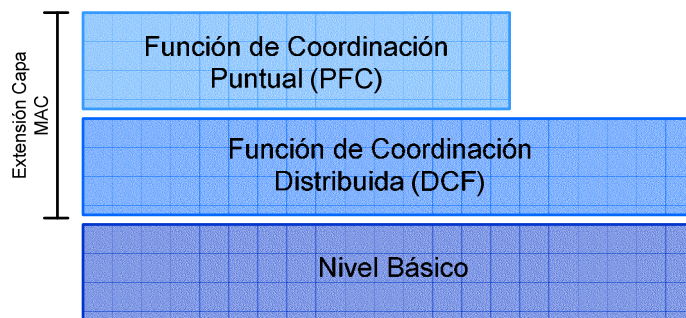


Figura 2.22 Arquitectura MAC.

#### 2.6.5.2 Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA).

El protocolo CSMA más conocidos en la industria es el de *Ethernet*, el cual es un protocolo *Carrier Sense Multiple Access with Collision Detection (CSMA/CD)*. Un protocolo CSMA trabaja de forma muy breve y resumida, de la siguiente manera:

*Cuando una estación desea realizar una transmisión, censa primero el medio, si este está ocupado, la estación aplazará por un tiempo su transmisión; en caso contrario la transmisión se llevará a cabo.*

<sup>6</sup> En este caso, contención es la acción de múltiples estaciones de contender por el acceso al medio, bajo el principio de FIFO (First Input, First Output).



Este tipo de protocolo es eficiente cuando el medio no se encuentra muy saturado, ya que permiten que las estaciones transmitan con un tiempo mínimo de retraso; sin embargo no pueden evitar colisiones, ya que si dos estaciones censan el medio y determinan su disponibilidad, ambas decidirán transmitir al mismo tiempo.

Cuando las colisiones son detectadas, la capa MAC retransmite el paquete solo a nivel de capa dos y no a otros niveles, lo que provoca un retardo significativo, sin embargo, estos mecanismos de detección de colisiones solo son aplicables a redes cableadas, ya que no pueden ser utilizados en WLAN, por las siguientes razones:

- ⊕ La implementación de un mecanismo de esta naturaleza, necesita de una transmisión Full Duplex y su implementación inalámbrica incrementaría considerablemente el costo de este servicio.
- ⊕ En un medio inalámbrico no se puede asumir que todas las estaciones se “escuchan” una a otra (lo cual es la suposición básica de un esquema de detección de una colisión) y el hecho de que una estación cense previamente un medio, no necesariamente implica que el medio se encuentre disponible.

Con el propósito de superar estos detalles, el estándar 802.11 utiliza un mecanismo de prevención de colisiones (CA) junto con un esquema de reconocimiento, que funciona de la siguiente manera:

- ⊕ Una estación dispuesta a transmitir censa el medio, si el medio se encuentra ocupado, pospone esta acción; en caso de que el medio se encuentre libre por cierto tiempo, denominado *Distributed Inter Frame Space (DIFS)* por el estándar, la estación transmite. La estación receptora revisa el *Cyclic Redundancy Check (CRC)* de los paquetes recibidos y envía paquete de acuse de recibo de toda la información, denominado *acknowledgement (ACK)*. La recepción de este paquete, indicará que no se produjo ninguna colisión, en caso contrario, se propiciará la retransmisión del fragmento perdido.

El mecanismo físico de detección de colisión es muy eficiente, pero no es eficaz cuando dos estaciones de una misma red que “no se ven entre ellas” transmiten simultáneamente. Para evitar estos problemas, se cuenta con un sistema lógico de detección de colisión que consiste en intercambiar información acerca de la disponibilidad del medio a través de tramas de control.

Cuando una estación se dispone a transmitir, primero envía una pequeña serie de paquetes de control llamada *Request To Send (RTS)*, en esta se incluyen la fuente, destino y la duración de la transacción (el envío del paquete y su respectivo ACK), en caso del que medio esté libre, la estación responderá con un paquete de control denominado *Clear To Send (CTS)*. Todas las estaciones receptoras de paquetes RTS o CTS, mantendrán su indicador de censado por portadora virtual y utilizarán esta información junto con una portadora de censado físico, al momento de determinar la disponibilidad del medio.

Este mecanismo reduce la posibilidad de que se presente colisión, ya que en el momento de que la estación emisora reciba el paquete CTS, reservará el medio hasta que se finalice la transacción. Debido a que RTS y CTS son paquetes cortos, reduce considerablemente el número de colisiones, ya que son identificados en un tiempo menor en compa-



ración con el que se emplea en los paquetes de información y no necesitan de otros paquetes de acuse de recibo.

### 2.6.5.3 Formato del Frame.

La trama MAC para WLAN's especificadas por el protocolo 802.11 poseen los siguientes componentes básicos:

- ✦ *Cabecera MAC*: Comprende campos de control, duración, direccionamiento y control de secuencia.
- ✦ *Cuerpo de trama*: Segmento de longitud variable, que contiene información específica del tipo de trama.
- ✦ *Secuencia checksum (FCS)*: Parte de la trama que contiene un código de redundancia (CRC) de 32 bits.

Cada uno de estos bloques se muestra de forma muy general en la figura 2.23.

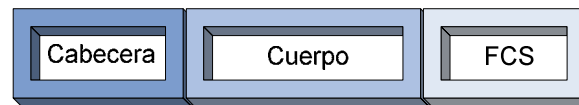


Figura 2.23 Trama MAC general.

Las tramas MAC se pueden clasificar según tres tipos:

- ✦ *Tramas de datos*: Contienen la información que se desea transmitir.
- ✦ *Tramas de control*: Los ejemplos de tramas de este tipo son los reconocimientos o ACKs, las tramas para multiacceso RTS y CTS, y las tramas libres de contienda.
- ✦ *Tramas de gestión*: Como ejemplo podemos citar los diferentes servicios de distribución, como el servicio de Asociación, las tramas de *Beacon* o portadora y las tramas *Traffic Indication Message (TIM)* o de tráfico pendiente en el punto de acceso.

El formato de la trama MAC de manera más detallada se aprecia en la figura 2.24.

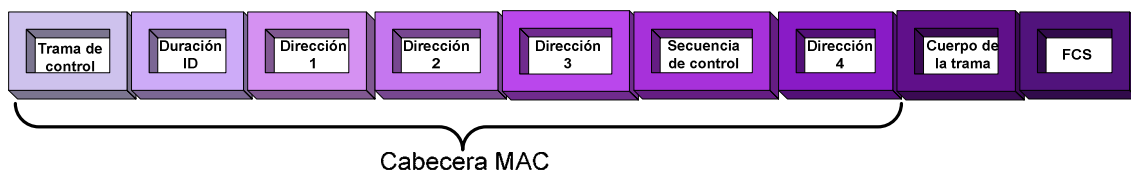


Figura 2.24 Trama MAC detallada.

Donde cada uno de los campos se detalla a continuación.

- ✦ *Trama de control*. Se constituye de 11 campos que contribuyen a la función de ésta.
  - ✦ *Versión*: Versión del protocolo que se está utilizando.
  - ✦ *Tipo/Subtipo*: El campo tipo, en colaboración con el campo subtipo identifica si la trama en cuestión transporta datos, es de control o gestión.

- ✚ *To DS/From DS*: Ambos campos identifican con un 1, si la trama se envía o se recibe al sistema de distribución (DS) respectivamente. En redes de tipo Ad-Hoc ambos campos poseen un cero. En casos más complejos, donde se realice el envío entre dos estaciones a través del DS, ambos campos se sitúan en 1.
- ✚ *Más Fragmentos*: Este campo se activa si y solo si se hace uso de la fragmentación.
- ✚ *Re tx*: Se activa cuando la trama es una retransmisión de una anterior; así se evita la duplicación de información
- ✚ *Admón. Energía*: Cuando la estación transmisora utiliza el modo de economía de potencia, este campo se activa. Un 1 indica su empleo.
- ✚ *Más datos*: Este campo se utiliza en caso de que la estación tenga tramas pendientes en un AP. Este campo solo se utiliza en tramas de datos y administración.
- ✚ *WEP*: En caso de hacer uso de algún mecanismo de autenticación o encriptado, este campo es activado.
- ✚ *Orden*: Cuando se utiliza un servicio de ordenamiento estricto, se indica con este campo.

La representación gráfica de la trama de control se puede observar en la figura 2.25.

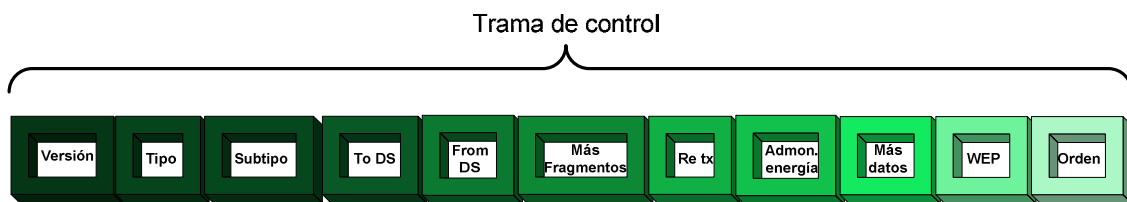


Figura 2.25 Trama de control detallada.

- ✚ *Duración ID*: En tramas del tipo PS o Power-Save para dispositivos con limitaciones de potencia, contiene el identificador o AID de estación. En el resto, se utiliza para indicar la duración del periodo que se ha reservado una estación.
- ✚ *Dirección 1-4*: Contiene direcciones de 48 bits donde se incluirán las direcciones de la estación que transmite, la que recibe, el punto de acceso origen y el punto de acceso destino.
- ✚ *Secuencia de control*: Contiene tanto el número de secuencia como el número de fragmento en la trama que se está enviando.
- ✚ *Cuerpo de la trama*: Varía según el tipo de trama que se quiere enviar.
- ✚ *FCS*: Contiene un código de redundancia de 32 bits, este valor se calcula utilizando el encabezado y cuerpo de la trama, a través de un polinomio de grado 32.

## CAPÍTULO III

### LA APLICACIÓN WIRELESS NETWORK POSITION SYSTEM (WINPS)

---

#### 3.1 Introducción.

En el capítulo I se describió un poco de las tecnologías de localización, ahora bien, se puede recapitular que en comunicaciones inalámbricas es posible detectar la posición de la llamada MS con ayuda de las BTS.

En este mismo capítulo se describieron algunas de las muchas técnicas existentes para la localización, algunas de las cuales pueden ser más eficientes que otras, o bien pueden tener aplicaciones más locales. Sin embargo, para este capítulo retomaremos el concepto de *mapping*, pues de acuerdo a las dimensiones del área de trabajo o *workspace*, nos es más útil que un método de trilateración como el que se usa en el GPS.

Asimismo, en el segundo capítulo se describieron los elementos de la llamada red WLAN, se vieron las unidades, los protocolos y algunas de las características de este tipo de comunicación inalámbrica.

Ahora bien, retomando lo visto en el capítulo II, podemos centrar nuestra atención en la posibilidad de realizar un ejercicio de localización muy similar al descrito en el capítulo I. Así pues, la parte medular de este trabajo se basa en la extrapolación de los conceptos de localización aplicados ahora a una red WLAN.

Es por ello que en este y en el capítulo siguiente se describirán las características de la aplicación del método de trilateración para detectar la posición de una PC (la cual hará las veces de MS) dentro de una red inalámbrica Wi-Fi controlada por tres Access Points (los que representan nuestras tres BTS).

En primera instancia, aterrizaremos los conceptos en una idea principal, formulando una propuesta para lograr el objetivo planteado al inicio de este presente trabajo de tesis. Posteriormente enunciaremos los pros y contras que teóricamente nos encontraremos en el desarrollo, y además se planteará la manera en que dichos factores puedan ser solventados a fin de llegar a nuestro objetivo.

### 3.2 Propuesta de solución.

En el capítulo I se mencionaba que la localización mediante redes locales inalámbricas se puede realizar de diversas maneras, muchos de ellos presentaban ciertos errores. Sin embargo en este presente trabajo de tesis se propone un método para la localización que en teoría debería presentar un error mínimo, y que además en México no ha sido adoptado totalmente, e inclusive, no está abierto a todo el público.

Como respuesta a esto, nosotros proponemos la creación de un motor de localización llamado WiNPS, este nombre resulta ser el acrónimo de *Wireless Network Position System*, y como objetivo fundamental, pretende cubrir la necesidad de llenar el hueco existente en las herramientas de posicionamiento existentes en México.

WiNPS se propone sea desarrollado en un sistema operativo que nos permita modificar y acceder con facilidad a los recursos de la capa de hardware. Es por ello que se propone como base a Linux, pues este es un *Operative System (OS)* que nos permite acceder de manera sencilla a una parte del sistema de archivos en donde se almacena toda aquella información del hardware.

Asimismo en base a los estándares actuales que han tenido mucho auge en México, y tomando en cuenta la red en donde va a ser implementada, debemos de tomar en cuenta que WiNPS debe de ser desarrollado para ser compatible con redes que utilizan el IEEE 802.11 b y g.

Cuando se puso a discusión el método que utilizaría WiNPS, se acordó en que el procedimiento de localización debía basarse en un *Wi-Fi mapping*<sup>7</sup>, para llevar a cabo esto, llegamos a la conclusión de que requerimos dos herramientas importantes:

- ✦ Herramienta para realizar el mapeo
- ✦ Herramienta para la localización

La herramienta para realizar el *mapping* no es más que una herramienta de monitoreo de la red, se investigó que en Linux existen diversas herramientas para monitorear el estatus de una red, sin embargo estas herramientas nos arrojan un conjunto de datos, es decir, no nos proporcionan por si solo el dato específico que requerimos. Es por ello que se debe de crear un programa que recoja y almacene los datos requeridos para el *mapping*.

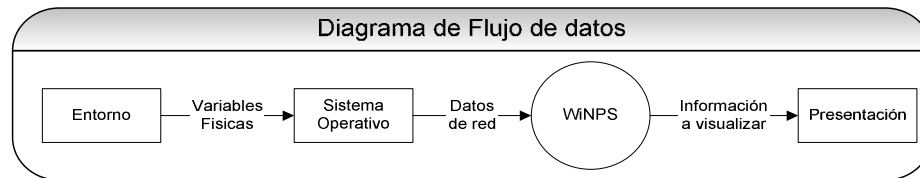
La herramienta de localización hace referencia a la programación del módulo que realiza la trilateración entre los datos mapeados y la intensidad de señal que presenta el móvil en un punto. Ahora bien, hablando del método, se proponen las siguientes etapas en el desarrollo de la aplicación.

- ✦ *Consideraciones Iniciales.*
- ✦ *Delimitar el área de trabajo de los APs.*
- ✦ *Calibración del modelo de posicionamiento.*
- ✦ *Implementación del software.*
- ✦ *Verificación y realimentación.*

---

<sup>7</sup> basada en el almacenamiento de medida de potencia de señal en diferentes puntos del recinto cubierto

En los siguientes temas se desarrollaran con más detalle cada una de estas etapas, lo que podemos adelantar es el flujo de datos que se tiene planeado para el desarrollo e implementación de WiNPS. Para ello se cuenta con la figura 3.1, en la que se puede ver lo que el OS procesa del exterior mediante el adaptador inalámbrico.



*Figura 3.1 diagrama de bloques representativo del sistema.*

Vale la pena mencionar que es muy importante recordar este diagrama pues en muchos de los temas tratados se hará referencia a la relación existente entre Red-Hardware y las interfaces Hardware-Software (OS) y Software (WiNPS)-Usuario.

### 3.2.1 Consideraciones Iniciales.

Como es de esperarse para dar solución a un problema, lo primero que tenemos que hacer es *pensar* y con ello realizar una lista de consideraciones iniciales. Siguiendo este camino, llegamos a una serie de cuestionamientos, dichas preguntas, nos llevan a darnos una idea más sólida de los problemas que pudieran surgir al momento de desarrollar a WiNPS.

Así pues las preguntas que nos hicimos en la propuesta del presente trabajo fueron:

- ✦ *¿Qué vamos a hacer?*
- ✦ *¿Cómo lo vamos a hacer?*
- ✦ *¿En dónde se va a implementar?*
- ✦ *¿Con qué elementos contamos?*
- ✦ *¿Qué elementos esperamos que cubra el software?*
- ✦ *¿Qué elementos pueden cubrirse de inmediato?*
- ✦ *¿Cuál es el capital con el que se dispone?*
- ✦ *¿Cuánto tiempo va a demorar el desarrollo del sistema?*

Quizás las preguntas no estén acomodadas en algún tipo especial de prioridad, sin embargo, se puede ver que todas están orientadas a aterrizar la idea que se tiene desde el planteamiento del tema de esta tesis.

Con base en dichos cuestionamientos nosotros inicialmente pensamos en un sistema como el que se muestra en la figura 3.2.

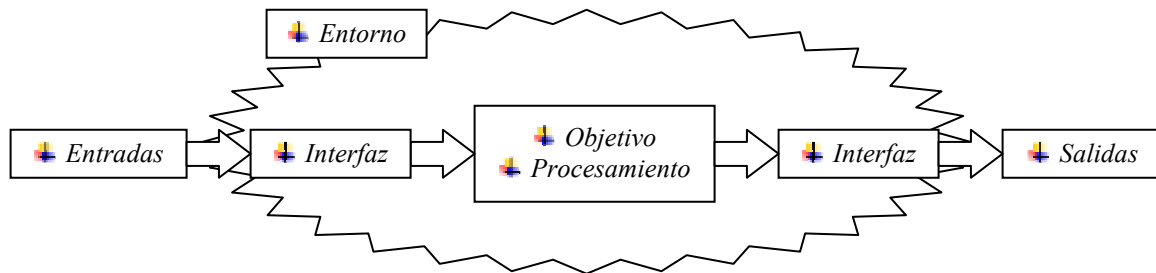


Figura 3.2 diagrama de bloques representativo del sistema.

Ahora bien se puede decir que WiNPS es el conjunto de todos estos bloques, sin embargo, en un sentido más estricto, nosotros lo hemos concebido como esos tres bloques centrales (interfaces y procesamiento), esto viene a razón de que WiNPS por sí solo no puede funcionar, o mejor dicho no arrojaría resultados coherentes, pues requiere de otros más elementos tales como el entorno.

Uno se podría preguntar *¿Si es un programa, entonces porque no funcionaría?* Bueno pues la respuesta subyace en que es un programa basado en *Wi-Fi mapping* lo que lo hace fungir como una herramienta de monitoreo, así pues, la relación Ambiente-Software resulta necesaria.

Con este esbozo, fue sencillo darse cuenta que existen ciertos elementos cruciales y definitivos en la solución, a estos elementos se les denominó *barreras técnicas*. Se les definió como barreras por hacer referencia a los obstáculos, esto es, al ir desarrollando el sistema, habrá ciertos elementos que hay que implementar o adaptar para llegar al óptimo desarrollo de la aplicación.

Para nosotros haber definido estos puntos consideramos no solo el esquema y las preguntas presentadas con anterioridad, sino que se tomaron en cuenta los alcances que tienen las respuestas a los anteriores cuestionamientos, esto es, se consideraron los elementos físicos de la red, elementos de red-hardware, las interfaces hardware-software y software-usuario.

Luego entonces, lo que se obtuvo fue una clasificación tripartita de los elementos, llegando a la conclusión de que existen cinco barreras, las cuales pueden ser clasificadas en tres grupos importantes. En la figura 3.3 se pueden ver la evolución de WiNPS y de cómo están relacionadas las barreras en su desarrollo; los círculos son los momentos en los que nosotros deseamos ver resultados parciales de la aplicación, y con base en ello, poder darle forma a WiNPS.

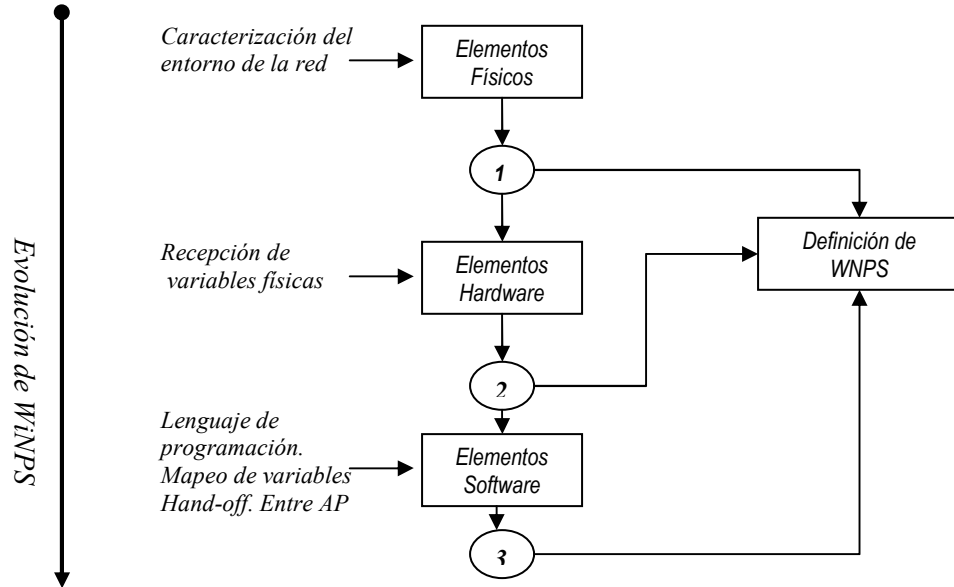


Figura 3.3 evolución de WiNPS.

A continuación explicaremos con más detalle en que consisten estas barreras, vale la pena mencionar que a cada una de las barreras se colocó en su elemento correspondiente para que así pudiesen ser mejor estudiadas y en el momento indicado poder resolverse.

### 3.2.1.1 Elementos Físicos.

Por elementos físicos no hacemos referencia al hardware, simplemente se hace alusión al entorno en el que vamos a trabajar, este elemento pretende resolver la pregunta *¿En dónde se va a implementar?* De acuerdo a la figura este es el primer elemento con el que nos topamos, y esto se debe a que si no tenemos definido el entorno en donde se va a desarrollar WiNPS, simplemente estaríamos desarrollando a ciegas.

Además, y como ya se decía al inicio de esta sección, existe una relación inalterable entre el espacio de trabajo y WiNPS, pues la aplicación se adapta a las características de la distribución de la potencia en un *workspace* delimitado previamente.

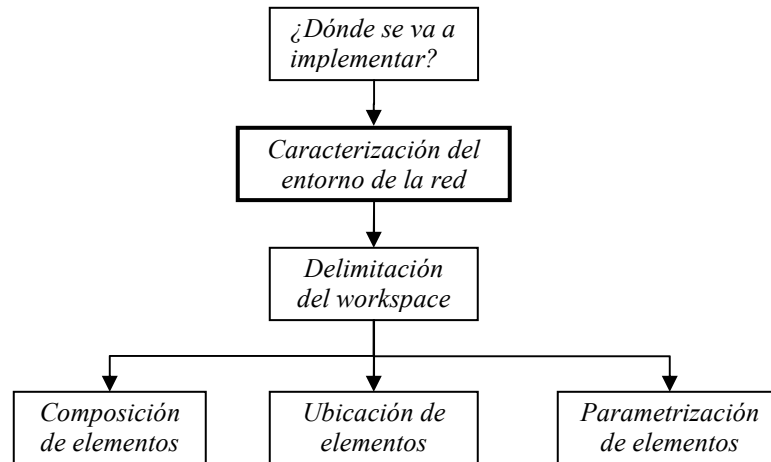


Figura 3.4 esquema de la primera barrera técnica.

Como se puede ver en la figura 3.4, con la limitación del espacio de trabajo, es posible relacionar la ubicación y parametrización de los elementos del ambiente, tales como la intensidad de la señal, el nivel de ruido, los paquetes transmitidos y recibos, entre otras más.

Ahora bien, ahondando más en la parte medular de este elemento, encontramos a la barrera correspondiente, de la cual se puede decir lo siguiente:

✦ *Caracterización del entorno de la red.*

Como bien se mencionaba, esta barrera se refiere a ubicar físicamente a la red, por ubicación nos referimos no solo conocer donde vamos a trabajar, si el objetivo de esta barrera fuera ese, realmente no se hubiera presentado una dificultad.

La barrera va más allá del conocimiento superficial del entorno, pues para el correcto desarrollo del sistema se requiere de una definición detallada del lugar en donde va a trabajar el sistema. Asimismo, y con base a la figura, tenemos tres elementos importantes, ubicación, composición y parametrización, estos elementos nos permitirán definir:

✦ *Dimensiones de los límites perimetrales de la locación.*

✦ *Definir los elementos internos en la locación (composición de paredes, en donde están ubicadas paredes falsa y de concreto, donde existen cristales, donde puertas, etc.)*

✦ *Dimensiones de los elementos internos.*

✦ *Definir la ubicación de los Access Points*

Esto resulta de vital importancia, pues el objetivo de WiNPS, y por ende el del presente trabajo de tesis, es conocer la ubicación de un móvil (persona con una terminal con acceso a una red Wi-Fi) dentro de una locación definida.

Si no se llegara a pasar esta barrera no podríamos saber si los resultados que obtenemos son coherentes, pues quizá la aplicación pudiera localizar a la terminal fuera de los límites de la planta del edificio.

### *3.2.1.2 Elementos de Hardware.*

Por elementos de hardware nosotros hacemos alusión a los elementos físicos que nos permiten interpretar el entorno, así por ejemplo para la creación de WiNPS el entorno está compuesto por una señal, la cual va a ser transmitida y emitida, es por ello que este elemento surge de la necesidad de contar con una interfaz física que interprete la intensidad con la que se recibe la señal y que esta pueda ser leída o interpretada por el sistema operativo.

Este elemento pretende solucionar las preguntas *¿Cómo lo vamos a hacer?*, *¿Con qué elementos contamos?* y parte de *¿Qué elementos esperamos que cubra el software?* Con ello, el sistema WiNPS se incorpora al ambiente de trabajo en el sentido de que encuentre el camino para interactuar con él.



Así pues, de acuerdo a la figura 3.3, este es el segundo elemento con el que nos topamos al momento de realizar el desarrollo del sistema, básicamente este elemento hace referencia a la interrelación que existe entre el ambiente y el dispositivo móvil.

De este elemento se desprende la barrera de la recepción de las variables físicas, la cual resulta muy importante, pues si no existe un modo en como recibirlo, el móvil quedaría aislado del entorno, y como consecuencia no se podría contar con una manera de localizarlo.

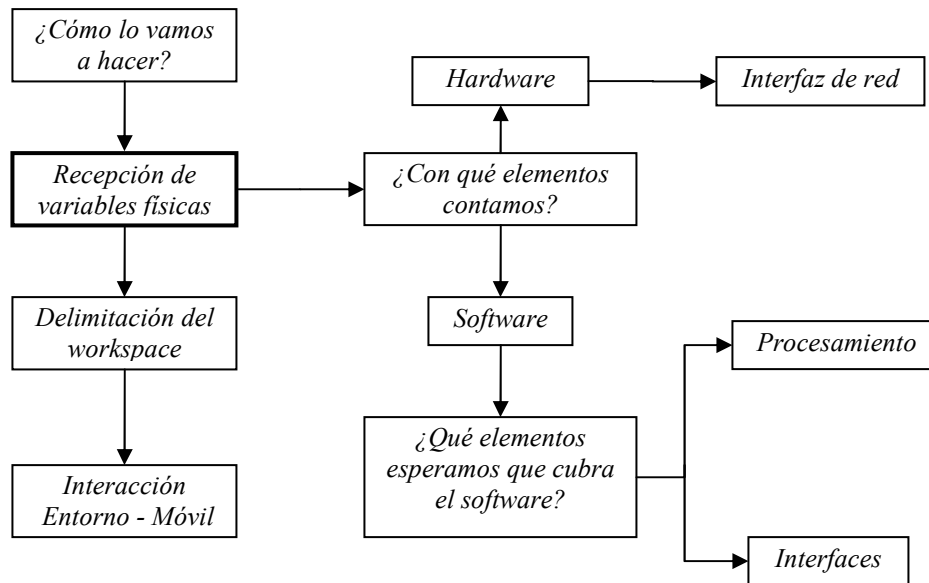


Figura 3.5 Esquema de la segunda barrera técnica.

Luego entonces, en la figura 3.5 se puede ver el camino que podemos seguir para solventar el problema que plantea la barrera. Se puede observar como un elemento importante es el hardware y el software, pues son un auxiliares en la interpretación de Entorno - Móvil, sino que en dado caso también lo será en la relación Móvil - Usuario. Así, ampliando un poco más la descripción de esta barrera técnica, encontramos lo siguiente:

✦ *Recepción de variables físicas.*

Como bien se mencionaba en los párrafos anteriores, esta segunda barrera se refiere a elegir una interfaz de hardware y software que permita las interacciones antes mencionadas. Consolidando esto, y en base a las condiciones con las que encontramos al analizar nuestro *workspace*, tenemos que nuestra interfaz Entono - Móvil, deberá contar con las siguientes características

- a) *Cumplir con los estándares 802.11 b y/o g.*
- b) *Ser compatible con el sistema operativo Linux.*
- c) *Permita acceder a las variables físicas que recibe del entorno.*

Como se puede ver, la recepción de variables físicas no solo debe cumplirse, sino que además nos permita extraerlas del dispositivo para que el usuario tenga la capacidad de visualizarlas, y que también nos permita manipular dichas variables para implementar nuestro algoritmo de localización.

### 3.2.1.3 Elementos de Software.

Este elemento es básicamente una consecuencia del punto anterior, pues también intenta satisfacer las mismas cuestiones planteadas, sin embargo dicho elemento trata la interacción final, la cual es la que tiene que ver con el usuario. También se puede considerar una parte importante, pues mediante este elemento se busca la presentación de resultados, pues la orientación del presente trabajo es obtener resultados tangibles y físicamente comprobables.

En esta parte básicamente se busca procesar los datos interpretados por el hardware, para así poder aplicar un algoritmo de localización previamente diseñado. Además, otra de las finalidades de este elemento es proporcionar al usuario una representación de la interpretación que el OS le hace a los datos.

Con base en esto llegamos a que para esta parte de los elementos de software encontramos tres partes muy importantes, las cuales deben de ser consideradas en la evolución del software. Así tenemos lo siguiente:

- ✚ *Procesamiento.* En esta parte nosotros debemos de elegir el OS que más nos convenga y que por ende, nos brinde más facilidades para la manipulación, esto para que el desarrollo no se vuelva tan complejo.
- ✚ *Interpretación.* Los resultados interpretados por el OS deberán de ser fácilmente procesados por un lenguaje de programación de alto nivel, se eligió programación de alto nivel, pues en la actualidad muchas de las aplicaciones están desarrollados en estos tipos de lenguajes, también nos presentan una mayor facilidad al momento de probar y modificar los algoritmos de procesamiento.
- ✚ *Presentación.* Esta la parte de la estética, y básicamente hacemos alusión a que el elemento de software debe de proporcionarnos una ambiente de trabajo agradable a la vista del usuario, además debe de proporcionarle la facilidad de interacción.

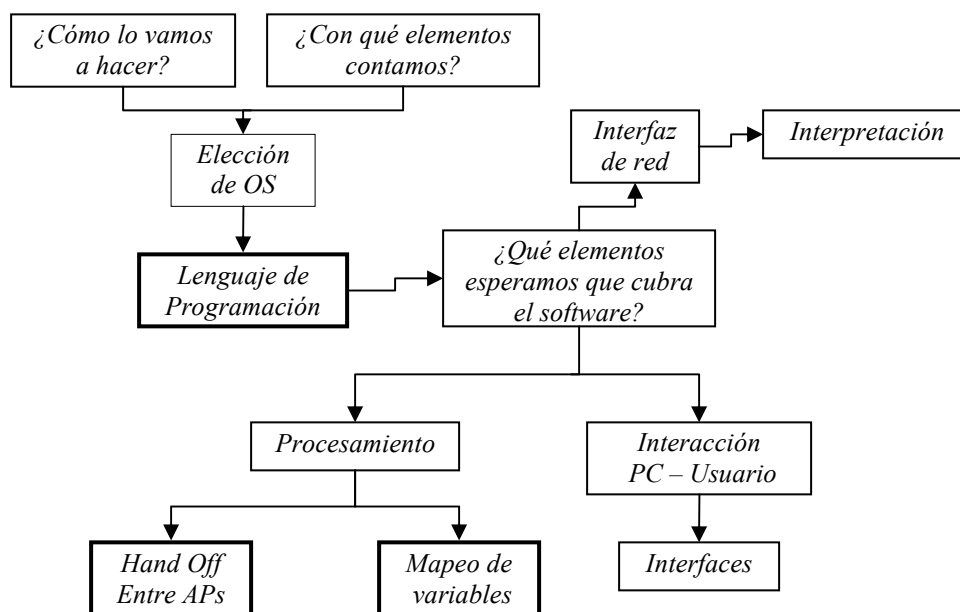


Figura 3.6 esquema de la tercer barrera técnica.

Como se puede ver en la figura 3.6, el elemento de software debe de satisfacer la interacción de estas tres rutinas principales. En negro se encuentran marcadas las tres últimas barreras que se derivan de este elemento, a continuación se da una explicación de lo que consisten estas barreras técnicas y del porque las hemos elegido como tales.

✦ *Lenguaje de Programación.*

El objetivo de esta barrera es detenerse a pensar en las capacidades de los lenguajes de programación existentes, sin embargo habría que definir la plataforma en la que se va a ejecutar la aplicación.

Como parte inicial de este trabajo de tesis, se propone que la aplicación este abierta para que en un futuro cualquier usuario pueda modificarla, es por ello que se eligió un OS de tipo libre tal como lo seria Linux o Solaris.

Así pues, ahora viene lo complicado de la barrera, esto es, las condiciones que debe de cumplir el lenguaje de programación, así, nosotros en base al esquema anterior llegamos a la conclusión que se requiere:

- a) *Contar con una interfaz gráfica.*
- b) *Debe de ser una aplicación de desarrollo sencilla.*
- c) *Debe permitir el acceso a comandos del sistema para manipular información.*
- d) *Permitir un rápido procesamiento de escritura a la memoria.*

Para solucionar el punto uno y dos se propone utilizar un lenguaje de tipo *Rapid Application Development (RAD)*, el cual a parte de flexibilidad nos permite crear *Graphic User Interfaces (GUIs)* con mucha facilidad. En tanto para dar solución a los dos últimos puntos podemos utilizar un lenguaje basado en C o C++ que nos permita acceder a los comandos del sistema en el cual nosotros tendremos captada la interpretación de lo que recibe el hardware mediante el procesamiento del Sistema Operativo.

Como se puede ver, esta barrera es la parte medular del sistema, se podría pensar que es la que menos trabajo cueste definir, sin embargo la definición se debe de hacer de manera conciente, pues en ella recae el procesamiento y la presentación final.

✦ *Mapeo de variables físicas.*

Una vez que se ha definido el lenguaje de programación, encontramos otro obstáculo inmediato, este consiste el mapeo de variables físicas. Ahora bien, ¿Porqué se plantea como obstáculo?, pues bien, el programa debe ser capaz de leer la información del ambiente, procesarla y almacenarla en un mapa de distribución, luego entonces, resulta sencillo ver que este es uno de los puntos más difíciles de implementar.

Así pues, se tiene que este es un punto clave para el desarrollo de WiNPS, pues al ser un motor de localización que se basa en el *Wi-Fi mapping*; entonces para sobrepasar esta barrera debemos de ser capaces que con las facilidades que proporciona el lenguaje de programación, podamos realizar las siguientes tareas primordiales:

- ✦ *Recopilar la información del ambiente.* Con ayuda del OS, el programa debe de poder acceder al sistema de archivos en el que estén almacenadas las variables físicas que el adaptador es capaz de recibir.

- ✚ *Procesar la información.* El programa debe de ser capaz de analizar las variaciones existentes en la señal<sup>8</sup> y entregar los datos que el usuario requiere.
- ✚ *Almacenamiento de datos.* WiNPS debe de poder almacenar en memoria secundaria todos aquellos resultados del mapeo, y de otras variables de interés para poder ejecutar el algoritmo de posicionamiento.
- ✚ *Ejecución del Algoritmo de posicionamiento.* El motor de posicionamiento debe ejecutar el algoritmo propuesto por nosotros cuando el usuario de la indicación correspondiente.
- ✚ *Visualización de la información procesada.* El programa debe presentar de manera amigable los datos requeridos por el usuario, además, cuando se trate de la ejecución del motor de posicionamiento, el programa deberá presentar la ubicación en tiempo real.
- ✚ *Hand-off entre Access Point's.*

Esta barrera resulta ser también muy importante, básicamente es una variante de los tres primeros puntos tratados en el apartado anterior, esto es, debe de realizar de manera simultánea todo el procesamiento antes mencionado para cada uno de los APs.

Así pues, el programa deberá ser capaz de recopilar la información del medio ambiente, procesarla y almacenarla en memoria para así tener un *mapping* de tres APs. Pues ahí es donde reside la dificultad de esta barrera, y es que el programa debe de realizar todo esto de manera simultánea.

Como se puede dar uno cuenta, estas consideraciones han sido creadas para formar lo que más adelante será el motor de localización WiNPS, quizás parezcan contemplaciones parciales del problema, sin embargo, estas consideraciones nos sirvieron para ver la evolución del proyecto, así como las áreas en las que se debía de dedicar más tiempo de desarrollo.

### 3.2.2 Delimitar el área de trabajo de los APs.

Esta etapa de la propuesta implica la construcción del plano del área de trabajo, es decir, nos dedicaremos a conocer el entorno en donde se va a desenvolver nuestro motor de localización.

Como bien se mencionaba en algunas de las secciones previas esto nos sirve para ubicar al programa en entorno espacial, este paso resulta importante, pues con el podremos:

- ✚ *Conocer la conformación del terreno.*
- ✚ *Ubicar espacialmente a nuestros AP.*
- ✚ *Ubicar los puntos en donde se tomaran las mediciones.*
- ✚ *Predecir el comportamiento de las intensidades de señal.*

---

<sup>8</sup> Se verá en el capítulo siguiente que las mediciones no son repetibles en un mismo punto.

A continuación discutiremos estos cuatro puntos. Pues bien, el conocimiento de conformación del terreno nos sirve para saber dos cuestiones, la primera de ellas es ubicar espacialmente a nuestro motor, esto es, nos sirve para valorar si las coordenadas que arroja son ubicadas dentro de nuestro *workspace*; la segunda es, si el WiNPS se utilizará como un sistema comercial, sería importante conocer la ubicación de locaciones importantes, como baños, recepciones, salas de exposición o locales comerciales.

El segundo punto hace referencia no solo a la posibilidad de ubicar a los *Access Points* (APs) a modo de saber que tan cerca estamos, sino que también nos permitiría saber el comportamiento de la señal, pues hay que recordar que cuanto más cercano estemos de un AP, la intensidad de señal aumentara en uno y disminuirá en los restantes.

El tercer punto resulta lógico, pues con la delimitación del *workspace* así como el conocimiento del mismo, nosotros podemos tomar la decisión de donde es mejor tomar los puntos para poder conformar el *mapping*.

Para finalizar, el último punto resulta de suma importancia, pues con ello podemos saber si el *mapping* fue realizado de manera correcta. Es decir, con la conformación del *workspace* sabremos los materiales con los que se encuentra distribuido su interior, y con ello partiendo de lo declarado en el capítulo anterior, tenemos que la existirá una mayor atenuación en materiales como concreto, y en menor medida con muros falsos y cristales; con ello sabremos en donde la intensidad será menor o mayor.

Como se puede ver, estas consideraciones son importantes, y además hacen que esta sea una etapa importante en el desarrollo de WiNPS, pues lo orientan hacia la interacción Ambiente – Aplicación antes mencionada.

### 3.2.3 Calibración del modelo de posicionamiento.

Básicamente el objetivo de este punto consiste en recorrer el *workspace* en los puntos previamente marcados, grabando muestras de la potencia de la señal en cada uno de ellos. Hay que mencionar que hasta esta etapa no se requiere de información sobre la localización de los APs.

Como era de esperarse, al momento de llegar a este punto, ya debemos de haber solventado la primer barrera técnica que habíamos mencionado; además de contar con el espacio delimitado, y tomando en cuenta los elementos que pueden causar una atenuación, nosotros hemos concluido que para contar con una distribución de los lugares en donde se va a tomar la medición correspondiente se deberán de tomar en cuenta las siguientes consideraciones:

✦ *Distancia entre puntos.*

Se propone que la distancia sea mayor a dos metros, pero menor a cinco metros, la razón de esto subyace en la precisión. Es decir, cuanto más cercanos sean los puntos la precisión será mayor, sin embargo, llegará un punto en el que la sensibilidad del hardware no sera capaz de distinguir variaciones en puntos muy cercanos.

Experimentalmente hemos visto que a una distancia de 1.5 [m] los valores aún son equivalente, es por ello que se eligió el valor de dos metros, asimismo, se considero que para tener una resolución razonable en las búsquedas esta distancia era la óptima.

El límite de cinco metros se eligió por analogía del párrafo anterior, esto es, encontramos que en segmentos rodeados de elementos que producen atenuación mínima, los valores de intensidad de señal eran muy similares hasta una distancia de 4.3 [m]; es por ello que se tomó la decisión de contar con puntos distante de hasta cinco metros.

✦ *Densidad de puntos Medidos.*

Como se pudo notar, tenemos dos tipos de distancias entre puntos, la distribución entre cada uno de ellos dependerá del material atenuante dentro del que se encuentren. Es decir, en lugares con espacios más abiertos o con cristales, la distancia entre un punto y otro podrá llegar a ser de hasta cinco metros. En tanto en lugares más cerrados y con muchas divisiones, la distancia logrará ser menor o igual a dos metros.

Estas dos consideraciones son muy importantes para el correcto mapeo de las intensidades de señal; con ello esperamos tener una perspectiva clara de la distribución de la potencia. Asimismo, y como consecuencia, esperamos mejorar el desarrollo del motor de localización

### 3.2.4 Creación e implementación de un software.

Para la etapa del desarrollo del software debemos de tomar en cuenta las dos condiciones previamente mencionadas; la primera se refiere a la herramienta de aplicación que se encargará de la recopilación de la información del ambiente, que es con la que nosotros realizamos el *mapping*.

La segunda, es la herramienta que se encargará de ser el motor de localización, en donde residirá nuestro algoritmo de localización y la interfaz de usuario necesaria para desplegar en pantalla la posición del usuario en tiempo real.

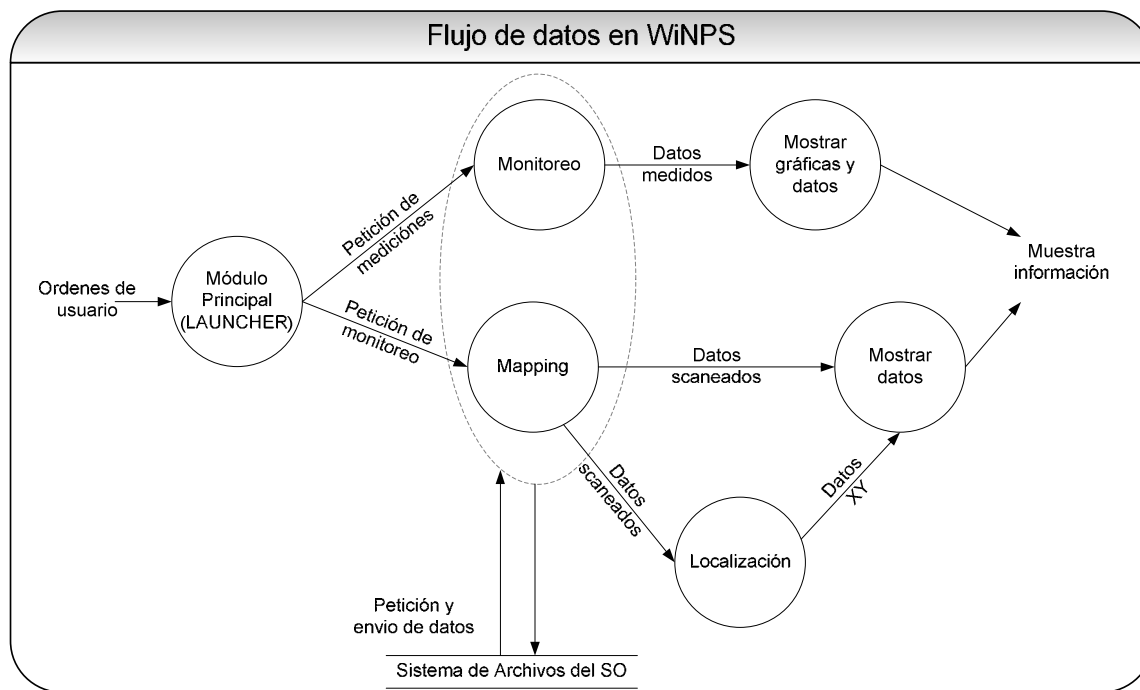


Figura 3.7 Flujo de datos en WinPS.

Retomando la figura 3.2, creamos el diagrama mostrado en la figura 3.7, el cual nos muestra el flujo de datos, y por ende el funcionamiento de la aplicación, en este punto no tiene caso explicar a fondo la estructura de WiNPS, consideramos que la información contenida en la figura 3.7 es suficiente para darse una idea del funcionamiento general del software.

Esta decisión la tomamos, pues hay aun hay que definir muchos conceptos y elementos del OS, esto se definirá con más detalle en el capítulo siguiente. Lo único que no hay que perder de vista es como se está presentando la dirección e interacción de los datos.

### 3.2.5 Verificación y realimentación.

En este punto se pretende corroborar todas nuestras expectativas planteadas a lo largo de este capítulo. Con las primeras ejecuciones del motor de localización, buscamos tener una visión general del comportamiento de la red y del comportamiento de las estimaciones de posición del usuario.

Con los datos arrojados generaremos conclusiones parciales, las cuales utilizaremos para calibrar nuestra aplicación. El paso final será retomar todas esas conclusiones parciales y realizar un compendio de todas ellas con su respectivo análisis.

Y bien, eso es de lo que trata, a grandes rasgos, el capítulo número cinco, veremos las conclusiones que se pueden sacar no solo del sistema WiNPS, sino de todo el análisis de la distribución de potencia y de la manera en que resolvimos los problemas existentes con el sistema operativo.

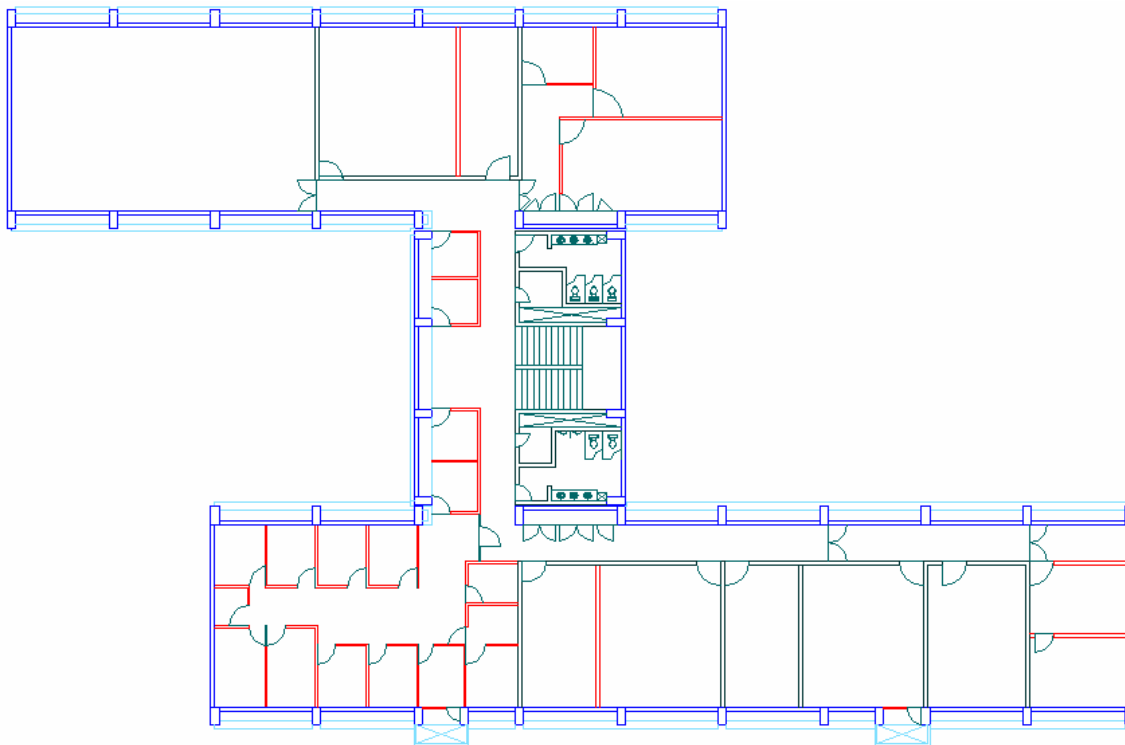
## CAPÍTULO IV EL DESARROLLO DE WINPS

---

### 4.1 Plano de la edificación.

Siguiendo la propuesta de solución planteada en el capítulo anterior, resulta necesario presentar un esquema representativo del área de trabajo en donde operan los APs. Sabemos que el área de trabajo será el tercer piso de la división de ingeniería eléctrica, el cual corresponde al departamento de Ingeniería en Telecomunicaciones.

Pues bien, la distribución de la planta de este edificio ha sufrido diversos cambios a partir de que se comenzó el desarrollo de WinPS, es por ello que el desarrollo del presente trabajo de tesis presentó un cierto retraso. No obstante, se pudo obtener una representación fidedigna de los elementos representativos de la planta.



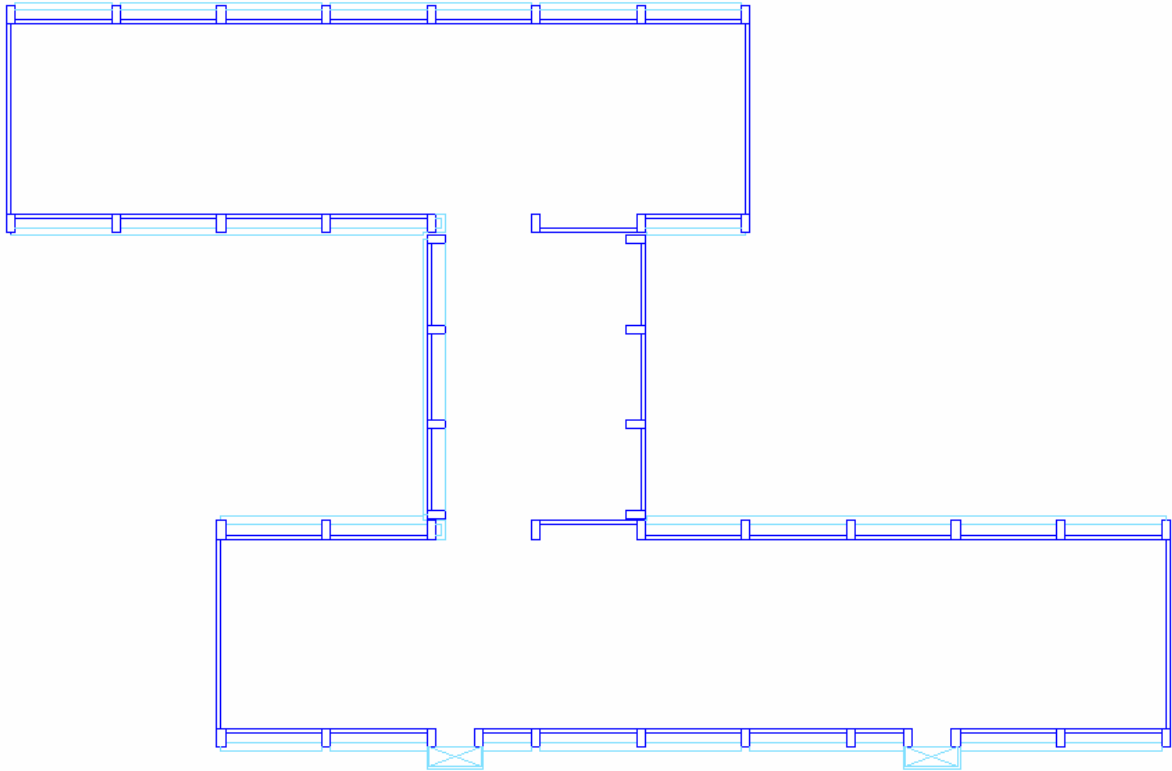
*Figura 4.1 Distribución actual del workspace.*

De la figura 4.1 resulta importante mencionar que las divisiones que se pueden ver están constituidas de diversos materiales, entre los materiales que encontramos para las divisiones tenemos:

- ⊕ Cristal de 0.4 [mm] y 0.5 [mm]
- ⊕ Muros falsos de 20 y 10 [cm]
- ⊕ Muros de concreto de 15 y 20[cm].

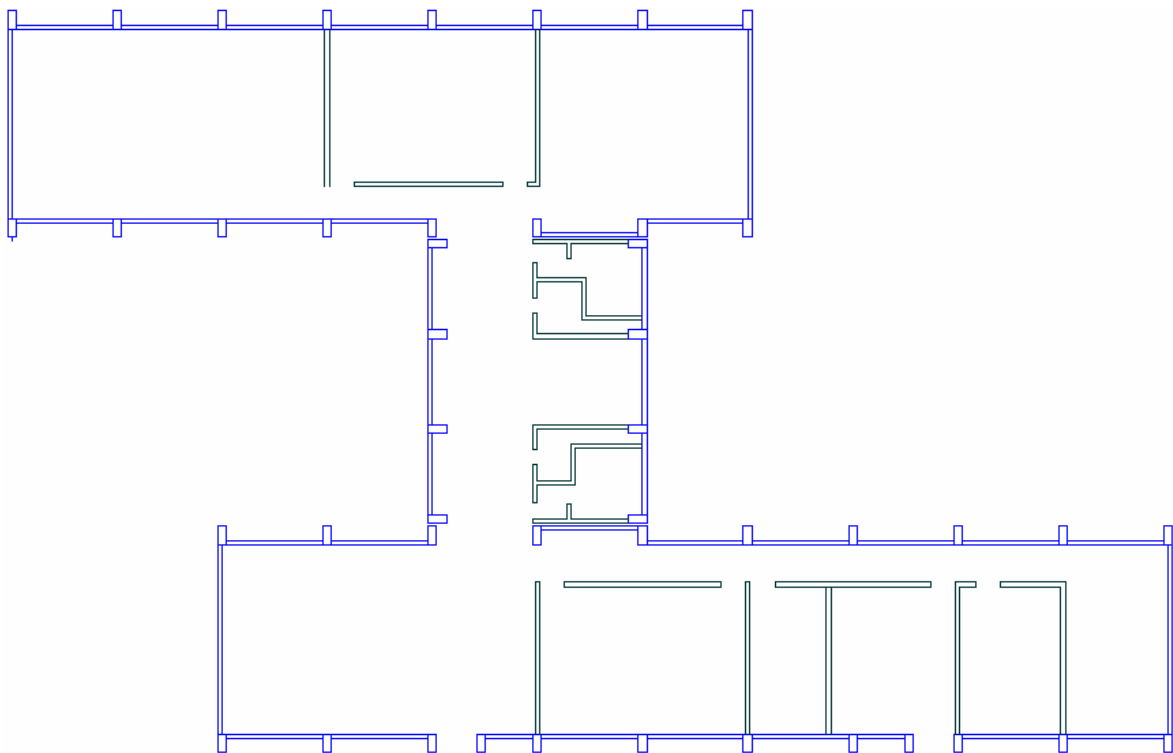
En la figura 4.2 se ve el perímetro del área de trabajo, hay que mencionar que esta es la parte que está compuesta de muros de concreto de hasta 20 [cm] de espesor.





*Figura 4.2 Perímetro del workspace.*

En la figura 4.3 se muestra las divisiones reales de la planta, su composición es de concreto con un espesor de hasta 15 [cm]; estas quizás fueron las divisiones originales del complejo.



*Figura 4.3 Divisiones de muros reales en el workspace.*

Para finalizar esta descomposición estructural, presentamos las divisiones falsas (figura 4.4) que se le han ido haciendo a la distribución del edificio, dichas divisiones tienen grosores de 10 [cm] y 15 [cm]. Podemos mencionar que estas adecuaciones son las de más reciente adición a la estructura de la planta.

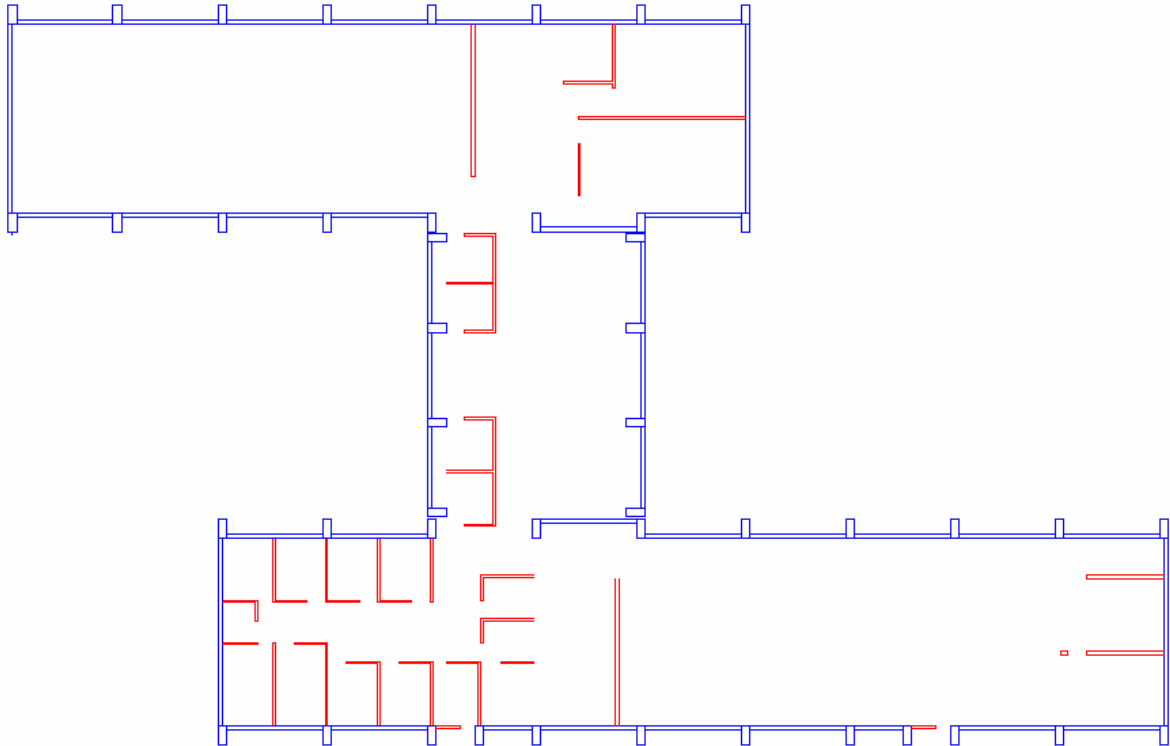


Figura 4.4 Divisiones de muros falsos en el workspace.

En este punto del desarrollo, una pregunta obligada sería *¿Porqué se realizan estas especificaciones?*, el motivo de hacer mención de esto, recae en un análisis *a priori* del *mapping*, es decir, en el capítulo II se mencionaba que la intensidad de señal varía de acuerdo al tipo de material en donde la señal se propaga, ahora bien, con base en ello generamos las siguientes suposiciones:

- ⊕ *La intensidad de señal aumentará cuanto más cerca se encuentre del AP.*
- ⊕ *En lugares rodeados de cristal o lugares abiertos, la intensidad de señal será superior que aquellos lugares rodeados de muros falsos y de concreto.*
- ⊕ *Como consecuencia de los puntos anteriores, la intensidad de señal será menor mientras el punto este más alejado de los AP y si además se encuentra rodeado de muros de concreto o bien se encuentre en espacios muy reducidos.*

Quizás estos puntos estén planteados muy a la ligera, sin embargo, en la sección dos del tema siguiente se discutirá con más detalle y además se pondrán como prueba los resultados del *mapping* para cada AP. Con ello lograremos no solo corroborar las suposiciones, sino que de cierto modo comprobaremos el estado del mapa de intensidades de señal.

Para finalizar, el último punto a tratar en esta sección es el correspondiente a la ubicación de los AP, pues bien, la figura 4.5 muestra la distribución de los APs en la planta del edificio.

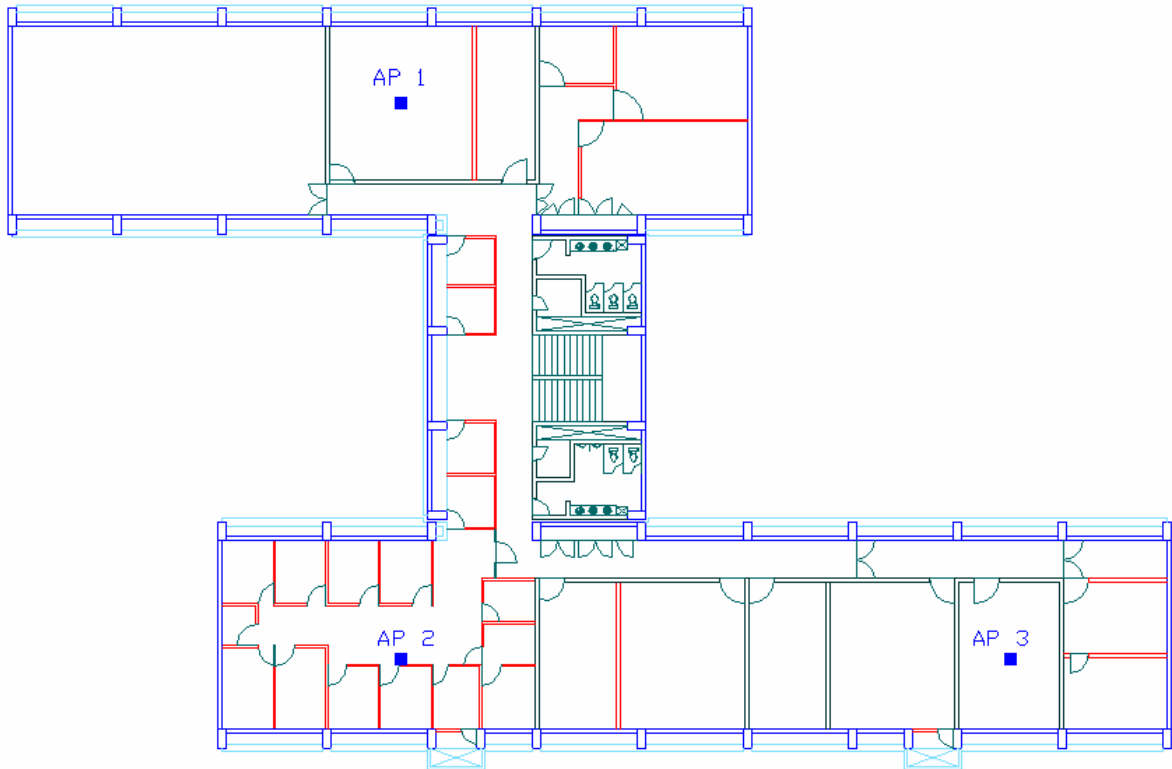


Figura 4.5 Distribución de los APs en el workspace.

De esta figura resulta importante mostrar las direcciones MAC de los AP, pues en secciones siguientes se hará referencia solo a  $AP_n$ , así pues, tenemos que:

- ✦  $AP_1$ : '00:0D:54:A9:D6:A8'
- ✦  $AP_2$ : '00:12:17:AA:44:0C'
- ✦  $AP_3$ : '00:12:17:7F:F2:AF'

Esta distribución resulta importante, cuando pusimos en marcha el desarrollo de WiNPS, todavía no se colocaban  $AP_2$  y  $AP_3$ . Se puso a discusión de la mejor forma en que debían de ser colocados los APs restantes, la conclusión a la que se llegó fue que éstos deberían ser colocados como si formasen un triángulo rectángulo y que además fuera isósceles. Existieron tres puntos que apoyaron esta decisión:

- ✦ *Las mediciones deben de ser controladas.* A distancias similares, las intensidades de señal en  $AP_1$  medidas en  $AP_2$  deberían de ser similares a las recibidas en  $AP_3$  medidas en  $AP_2$ .
- ✦ *Conveniencia para la trilateración.* Se espera que por lo menos por cada punto existan mediciones de intensidad recibidas por dos APs, con esto se espera que siempre se puedan tener dos mediciones distintas de cero en cualquier punto; el mejor de los casos sería contar con las tres.
- ✦ *Cubrir más terreno.* Si los tres APs se colocaran en la parte central del *workspace*, las partes suroeste y noreste quedarían sin cobertura de señal.

## 4.2 Mediciones de niveles en factores de transmisión.

En el capítulo anterior mencionábamos que para la caracterización del área de trabajo, necesitamos conocer el entorno, pero no solamente los elementos tangibles, sino que también hay que caracterizar el ambiente para ver si es factible poder trabajar sobre el Departamento de Telecomunicaciones.

Con la caracterización de los factores de transmisión (específicamente la intensidad de señal) podemos predecir el comportamiento de nuestro sistema de localización así como de la herramienta de monitoreo y *mapping*; también podemos tomar la decisión de donde es mejor tomar los puntos para poder conformar el mapeo.

Asimismo, pretendemos dar conocer el comportamiento que tiene en factor de intensidad de señal con respecto del entorno, esto es, con la conformación del *workspace* sabremos los materiales con los que se encuentra distribuido su interior comprobaremos que existirá una mayor atenuación en materiales como concreto, y en menor medida con muros falsos y cristales; con ello sabremos en donde la intensidad será menor o mayor.

### 4.2.1 Mapa de la distribución de factores.

Para llevar a cabo el mapa de distribución, nos auxiliamos de la herramienta de monitoreo de WinNPS con la que caracterizamos el *workspace*, posteriormente con ayuda de AutoCAD, se trazó el mapa de distribución que se presenta en la figura 4.6.

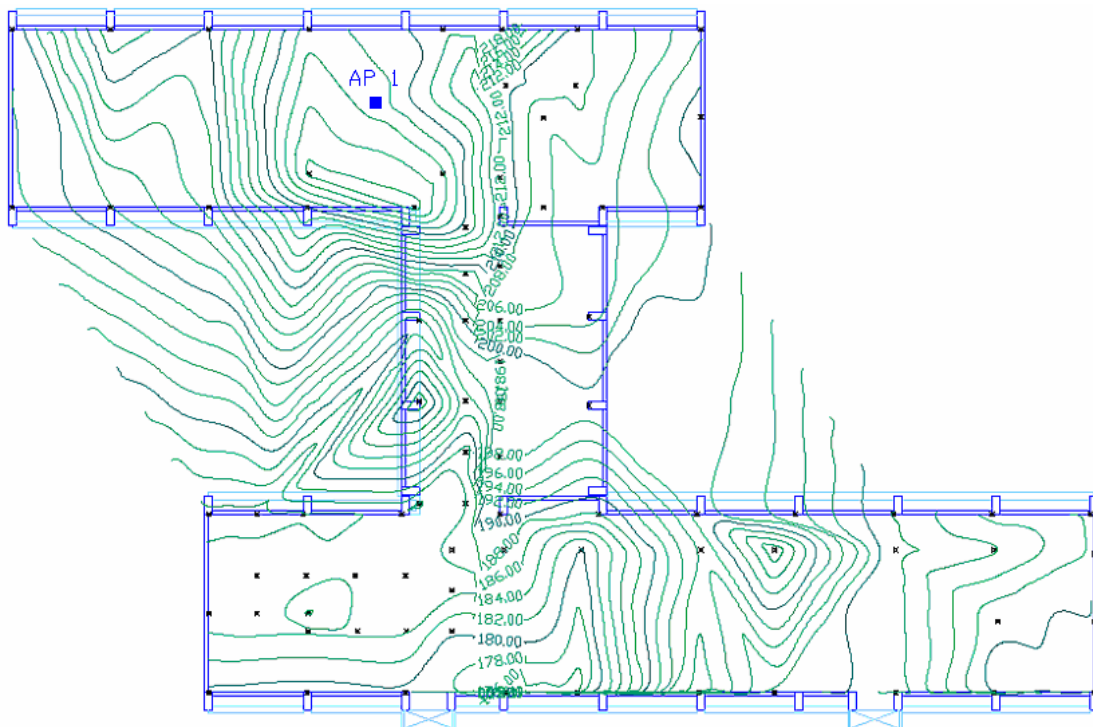


Figura 4.6 Mapa de distribución de la señal superpuesta al perímetro del workspace.

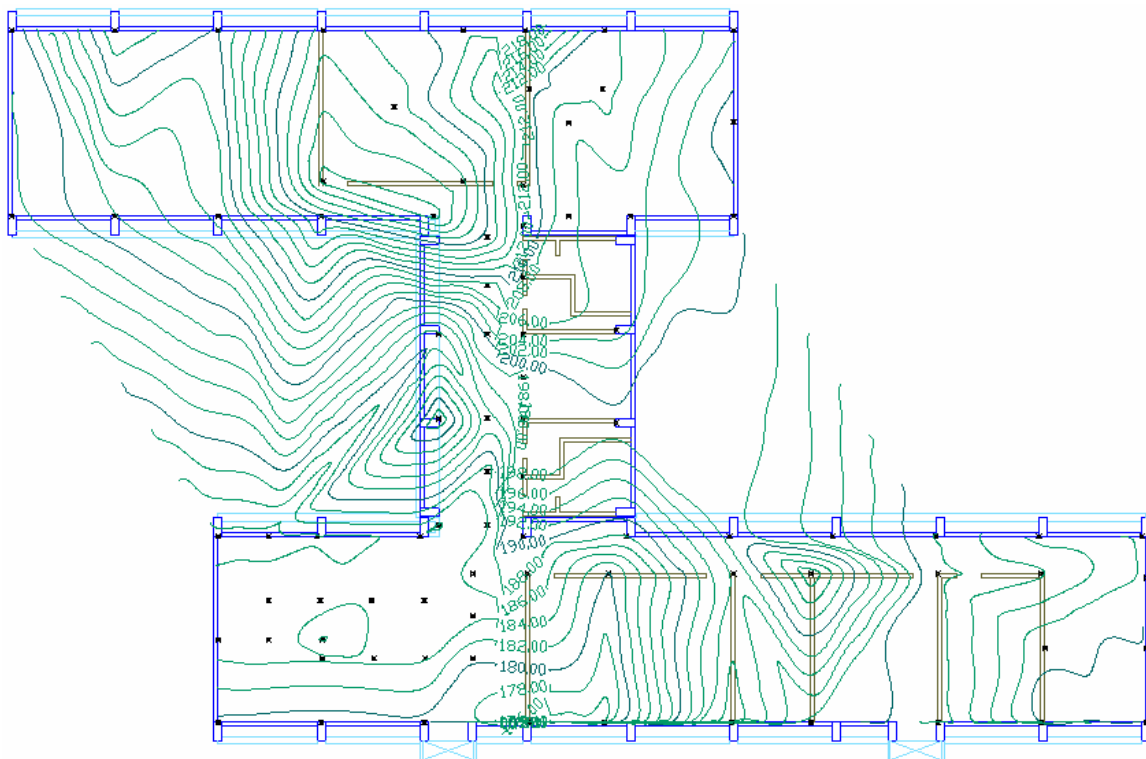
La distribución que se observa en la figura 4.6 se realizó con el AP que presentó una señal más uniforme durante las pruebas de la herramienta de monitoreo, este AP es el que se encuentra localizado en el laboratorio de redes. Ahora bien, realizaremos algunas observaciones sobre la figura 4.6, en esta se pueden ver una serie de elementos que carac-

terizan la intensidad de señal en el Departamento de Telecomunicaciones, luego entonces encontraremos lo siguiente:

- ✦ *Las curvas se obtuvieron mediante un proceso de triangulación utilizado por topógrafos o geólogos para realizar estudios de altimetría basándose en muestras de altura del terreno, solo que en nuestro caso, utilizamos intensidades de señal normalizadas.*
- ✦ *Cada curva representa una intensidad de señal diferente.*
- ✦ *Cuanto más cerrada es una curva (tiene un perímetro más pequeño) la intensidad de señal es mayor.*
- ✦ *Los números que se encuentran en la curva son las intensidades de señal normalizadas que le corresponden.*
- ✦ *Cuanto más cercanas son las curvas quiere decir que existe una mejor recepción de intensidad de señal.*
- ✦ *Como consecuencia del punto anterior cuanto más lejanas son las curvas una de otra, quiere decir que la intensidad de señal es menor.*

#### 4.2.2 Relación de los factores con la estructura de la edificación.

Con base en las observaciones realizadas en el punto anterior, estamos preparados para ver la relación existente entre el factor de intensidad de señal y la estructura del edificio; tal es el caso que presentamos en la figura 4.7, que básicamente es similar a la figura 4.6, solo que se han agregado las divisiones de paredes de concreto.



*Figura 4.7 Mapa de distribución vs Estructura de concreto.*

De la figura 4.7 se puede ver lo siguiente:

- ✦ La señal se atenúa en la parte centro-derecha, esto se debe a que en esa parte se encuentran los sanitarios y ahí existen muros de concreto de hasta 15 cm. de grosor.
- ✦ En la parte inferior-derecha, podemos ver que la señal crece, esto se debe a los rebotes de la señal en las paredes de concreto, es por ello que el pasillo de esta zona presenta una mayor intensidad de señal.
- ✦ Se puede ver que en el extremo derecho inferior más lejano, las líneas son más dispersas, por lo que la intensidad de señal se va perdiendo, este efecto se debe a la atenuación y la gran distancia existente.

Ahora corresponde el turno de mostrar el mapa de distribución contra las divisiones falsas (figura 4.8) esto es, son divisiones de entre 10 y 15 cm. de grosor de un material similar al concreto, solo que de menor densidad.

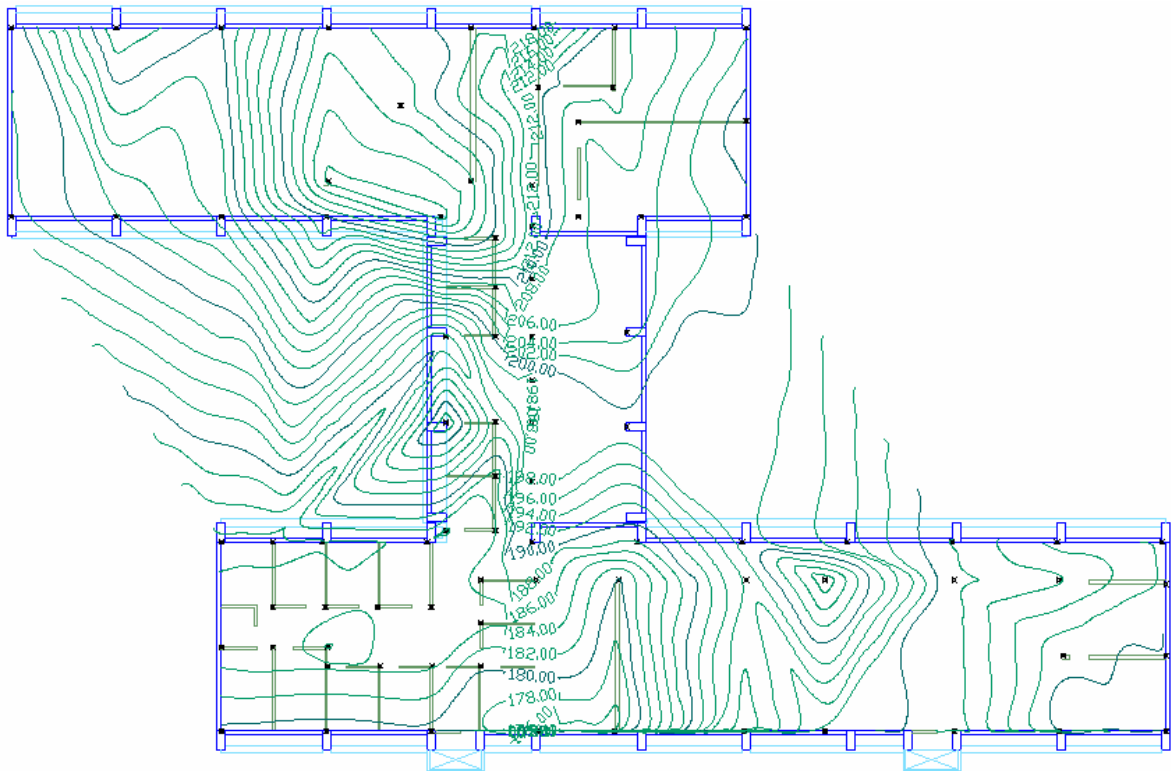
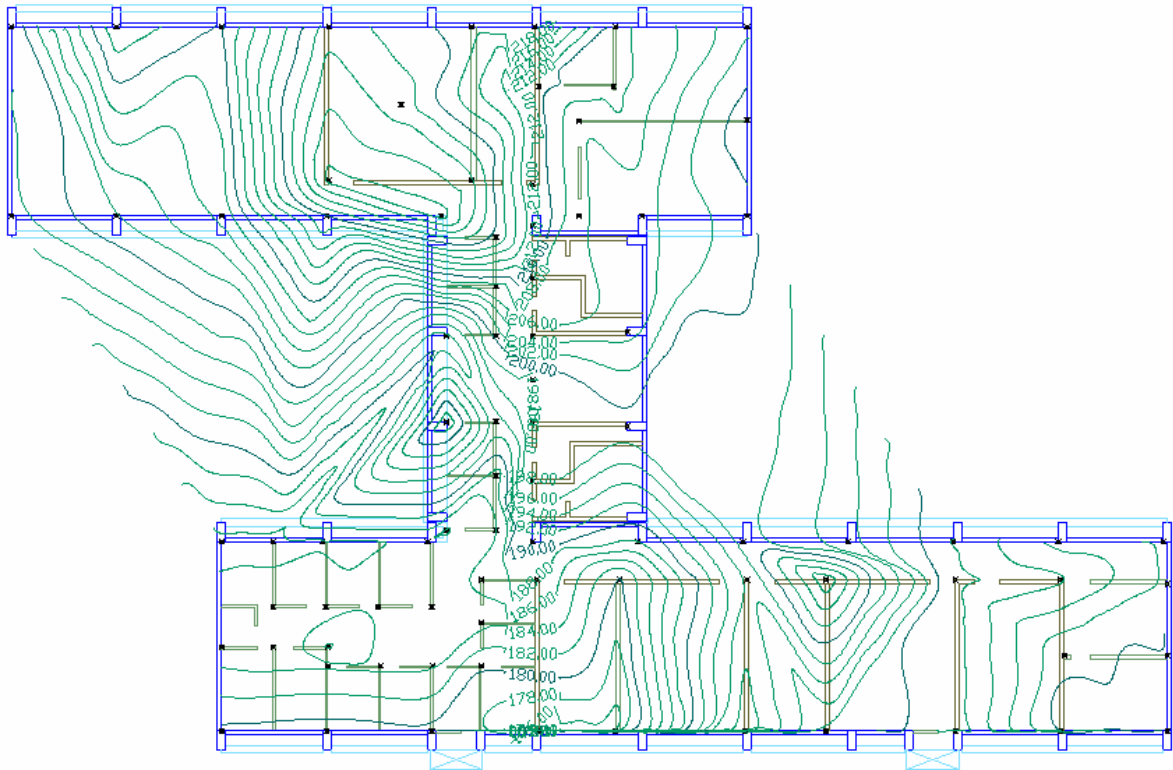


Figura 4.8 Mapa de distribución vs Estructura de falso concreto.

De esta figura se puede abstraer lo siguiente:

- ✦ En la zona inferior izquierda, la señal se ha perdido, esto se debe a que se forma una caja con la estructura de concreto y además esta caja tendría, por así llamarlo, un recubrimiento de muros falsos y puertas, esto hace que la atenuación crezca.
- ✦ Se ve que la intensidad de señal crece en la zona central izquierda, esto se debe a que el AP técnicamente tiene línea de vista con toda esa zona, además ayuda de que en esa parte existan amplios ventanales, así como el rebote de la señal en el pasillo con muros de concreto y paredes falsas.

Para finalizar este tema, presentamos el conglomerado de la estructura del *workspace* (figura 4.9) podemos darnos una idea generalizada de lo que ocurre con la señal dentro del área de trabajo.



*Figura 4.9 Mapa de distribución vs Estructura total del workspace.*

De la figura 4.9 no podemos obtener gran información pues mucha de ella ya fue vertida en la descripción de las dos anteriores, sin embargo, podemos concluir algunas cosas del *workspace* y de la herramienta de monitoreo.

- ⊕ *El área de trabajo puede caracterizarse sin temor a crear errores que nos impedirían la localización de usuario.*
- ⊕ *Por la particularidad de la forma del workspace, resulta importante mencionar que habrá lugares en los que no exista una intensidad de señal captada, esto ocurrirá en las zonas más alejadas, tal es el caso del área de los laboratorios de comunicaciones ópticas y la sala de juntas.*
- ⊕ *La distribución captada por nuestra herramienta de monitoreo cumplió con las expectativas teóricas de atenuación y distribución de señal.*
- ⊕ *Como consecuencia del punto anterior se puede extrapolar que la herramienta es totalmente funcional y que la herramienta de mapeo también lo será, pues esta realiza el mismo procedimiento de scanning, pero de manera simultánea a tres APs.*



### 4.3 Características del sistema de cómputo.

Continuando con las especificaciones del análisis propuesto para la creación de WiNPS que se mencionaba en el capítulo anterior, el programa WiNPS está formado de dos subprogramas:

- ✦ El primero de ellos es una herramienta de monitoreo de la intensidad de señal dentro de nuestro *workspace*;
- ✦ El segundo es propiamente el sistema de localización.

En este punto del desarrollo, resulta importante definir ciertos conceptos antes de dar paso a la descripción del programa. En las líneas subsecuentes, se describirán los elementos más importantes del sistema operativo y del lenguaje de programación, a modo de que se logre entender la interacción que existe entre ellos y WiNPS.

Así pues, se propuso que WiNPS fuese desarrollado sobre una plataforma Linux, es por esta razón que se requiere hacer mención de algunos conceptos básicos para poder entender, en mayor medida, el funcionamiento del motor de localización implementado.

#### 4.3.1 Estructura del Sistema de Archivos de Linux.

Para definir al sistema de archivos de Linux, resulta imperante conocer algunos conceptos fundamentales, pues a diferencia de Windows, este OS tiene algunas particularidades a las que no se está acostumbrado, de esta manera tenemos:

- ✦ *Archivos.*

La base del sistema de archivos de Linux, es obviamente el archivo, básicamente un archivo no es más que la estructura empleada por el sistema operativo para almacenar información en un dispositivo físico como un disco duro, un disquete, un CD-ROM o un DVD.

Todos sabemos que un archivo puede contener cualquier tipo de información, desde una imagen en formato *png* o *jpg* a un texto o una página web. Así podemos decir que el sistema de archivos es la estructura que permite que Linux maneje los archivos que contiene.

Como en Windows, se puede emplear un cierto criterio de *tipo* para “marcar” las distintas clases de ficheros empleando una serie de caracteres al final del nombre que indiquen el tipo de fichero del que se trata, a estas “marcas” se les conoce con el nombre de extensión. Así, los ficheros de texto, páginas Web, las imágenes *PNG* o *JPEG* tienen extensiones *.txt*, *.htm*, *.png* y *.jpg* respectivamente.

No obstante, en Linux ocurre algo diferente, pues este OS solo distingue tres tipos de archivos:

- ✦ Archivos o ficheros ordinarios, son los mencionados anteriormente.
- ✦ Directorios (o carpetas), es un archivo especial que agrupa otros ficheros de una forma estructurada.



- ✚ Archivos especiales, son la base sobre la que se asienta Linux, puesto que representan los dispositivos conectados a un ordenador, como puede ser una impresora. De esta forma introducir información en ese archivo equivale a enviar información a la impresora. Para el usuario estos dispositivos tienen el mismo aspecto y uso que los archivos ordinarios.

#### ✚ Enlaces.

Los enlaces son un tipo de archivo ordinario cuyo objetivo es crear un nuevo nombre para un archivo determinado. Una vez creado el enlace simbólico este permite acceder al fichero que enlaza de igual modo que si se hubiera copiado el contenido del mismo a otro fichero, con la ventaja de que este realmente no se ha copiado.

Los enlaces simbólicos son especialmente útiles cuando se quiere que un grupo de personas trabajen sobre un mismo fichero, puesto que permiten compartir el fichero pero centralizan las modificaciones.

#### ✚ Path.

En cualquier sistema operativo moderno la estructura de archivos es jerárquica y depende de los directorios. En general la estructura del sistema de archivos se asemeja a una estructura de árbol, estando compuesto cada nudo por un directorio o carpeta, que contiene otros directorios o archivos.

En Windows cada unidad de disco se identifica como una carpeta básica que sirve de raíz a otras, y cuyo nombre es especial a:, c:, d: etc. En los sistemas Unix, y por lo tanto en Linux, existe una única raíz llamada / de la que se derivan todos los ficheros y directorios, y que es independiente de qué dispositivos estén conectados al ordenador.

El camino o *path* de un fichero o directorio es la secuencia de directorios que se ha de recorrer para acceder a un determinado fichero separados por /. Así pues, supongamos la estructura de archivos tiene una organización como en la figura 4.10.

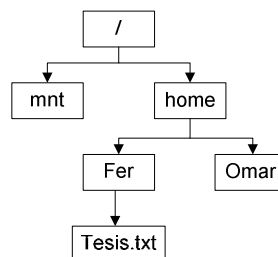


Figura 4.10 Ejemplificación del path.

Así pues, existen dos formas del path o camino:

- ✚ El camino absoluto que muestra toda la ruta a un fichero, /home/Fer/Tesis.txt
- ✚ El path relativo a un determinado directorio, por ejemplo si nos encontramos en el directorio /home, el path relativo al fichero Tesis.txt es Fer/Tesis.txt

Para complicar aun más las cosas, todos los directorios contienen dos directorios especiales:

- ✚ El directorio actual, representado por el punto
- ✚ El directorio padre representado por dos puntos (..)

Estando en el directorio `/home/Omar` se puede acceder a `Tesis.txt` con `/home/Fer/Tesis.txt` (path absoluto) o bien `../Fer/Tesis.txt` (path relativo). En `Fer` se podría acceder como `./Tesis.txt` o simplemente `Tesis.txt`

✦ *Sistema de Archivos.*

El sistema de archivo de Linux sigue todas las convenciones de Unix, lo cual significa que tiene una estructura determinada, compatible y homogénea con el resto de los sistemas Unix. Al contrario que en Windows o MS-DOS el sistema de archivos en cualquier sistema Unix no está ligado de una forma directa con la estructura del hardware, esto es, no depende de si un determinado ordenador tiene 1, 2 o 7 discos duros para crear las unidades `c:\`, `d:\` o `m:\`.

Todos los sistemas de archivos basados en Unix tienen un origen único la raíz o root representada por `/`. Bajo este directorio se encuentran todos los ficheros a los que puede acceder el sistema operativo.

Estos ficheros se organizan en distintos directorios cuya misión y nombre son estándar para todos los sistema Unix. En la tabla 4.1 se muestra la estructura del sistema de archivos, esta tabla hay que leerla con atención, pues es a partir de ella es que se recoge la información para WiNPS, y además resulta importante para otros conceptos que se mencionarán en los temas siguientes.

<code>/</code>	Raíz del sistema de archivos.
<code>/dev</code>	Contiene ficheros del sistema representando los dispositivos que estén físicamente instalados en el ordenador.
<code>/etc</code>	Este directorio está reservado para los ficheros de configuración del sistema. En este directorio no debe aparecer ningún fichero binario (programas). Bajo este deben parecer otros dos subdirectorios: <code>/etc/X11</code> Ficheros de configuración de X <i>Window</i> <code>/etc/skel</code> Ficheros de configuración básica que son copiados al directorio del usuario cuando se crea uno nuevo.
<code>/lib</code>	Contiene las librerías necesarias para que se ejecuten los programas que residen en <code>/bin</code> (no las librerías de los programas de los usuarios).
<code>/proc</code>	Contiene ficheros especiales que o bien reciben o envían información al <i>kernel</i> del sistema.
<code>/sbin</code>	Contiene programas que son únicamente accesibles al superusuario o <i>root</i> .
<code>/usr</code>	Este es uno de los directorios más importantes del sistema puesto que contiene los programas de uso común para todos los usuarios. Su estructura suele ser similar a la siguiente: <code>/usr/X11R6</code> Contiene los programas para ejecutar X <i>Window</i> . <code>/usr/bin</code> Programas de uso general, lo que incluye el compilador de C/C++. <code>/usr/doc</code> Documentación general del sistema. <code>/usr/etc</code> Ficheros de configuración generales. <code>/usr/include</code> Ficheros de cabecera de C/C++ (.h). <code>/usr/info</code> Ficheros de información de <i>GNU is Not Unix (GNU)</i> . <code>/usr/lib</code> Librerías generales de los programas. <code>/usr/man</code> Manuales accesibles con el comando <i>man</i> . <code>/usr/sbin</code> Programas de administración del sistema. <code>/usr/src</code> Código fuente de programas.
<code>/var</code>	Este directorio contiene información temporal de los programas

*Existen además de los anteriores otros directorios que se suelen localizar en el directorio `/usr`, como por ejemplo las carpetas de los programas que se instalen en el sistema.*

Tabla 4.1 Estructura básica del sistema de archivos en Unix.

### 4.3.2 Herramientas de monitoreo de una red Wi-Fi en Linux.

Este apartado surge de la necesidad de conocer las herramientas de monitoreo principales que se han desarrollado para este OS, el conocimiento del funcionamiento resultó importante durante el desarrollo de WiNPS pues una de las partes operativas de nuestro sistema es un módulo de monitoreo, con el cual se realizó el mapping del *workspace*.

Como ya vimos en la breve introducción de los parámetros que conforman la estructura esencial de Linux, estamos en la posibilidad de entender el funcionamiento de algunas de las herramientas que existen para monitorear el comportamiento de una terminal que forma parte de una red Wi-Fi.

Como se pudo dar cuenta, decimos “algunas herramientas” pues existe toda una gama de ellas, esto es debido que algunos de los fabricantes proporcionan herramientas GNU para mostrar el desempeño de su hardware. A continuación solo mencionaremos las herramientas genéricas que acompañan una instalación típica de Linux.

#### 4.3.2.1 Wireless Extensions.

Su creador<sup>9</sup> define a las *wireless extensions* como una *Application Programming Interface (API)* de WLAN para el sistema operativo Linux, la creación de estas extensiones de la API, el *Network Driver Interface Specification (NDIS)* viene a solventar un el problema de que no existía un modo de implantar una WLAN en dos PC con Linux.

A partir de esto, las *wireless extensions* se pueden definir como una API que permite manipular interfaces de red de tipo WLAN, dicha API está compuesta de una variedad de herramientas y archivos de configuración. Estas herramientas reciben el nombre de *Wireless Tools* y son usadas para cambiar la configuración de las interfaces de red WLAN en tiempo real, también nos permiten obtener la información de la configuración actual a modo de contar con estadísticas y herramientas de diagnostico de la misma red.

En la figura 4.11 se puede apreciar la implementación de las *wireless extensions*, esta implementación va desde lo que ve el usuario, hasta la parte física, a modo de que exista una interacción entre la red inalámbrica y el usuario.

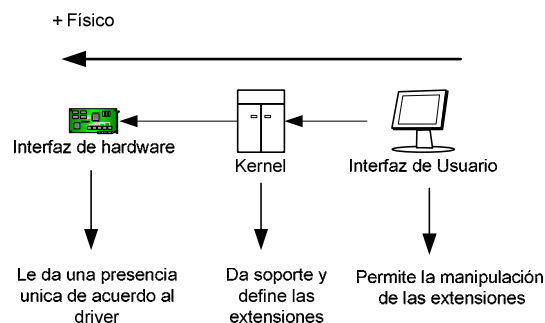


Figura 4.11 Partes en la que se implementan las wireless extensions.

<sup>9</sup> Jean Tourrilhes establece el concepto de *Wireless* Extensión a finales de 1996, y es el 23 de enero de 1997 cuando documenta esta nueva API.

Retomando lo anterior, las *wireless extensions* se implementan primeramente a nivel de usuario, esto es, la implementación se basa en el conjunto de herramientas o *wireless tools* a las que el usuario puede acceder para visualizar los datos de red, así como la manipulación de los mismos.

El segundo nivel es el correspondiente al *kernel* del OS que es en donde se escribe la definición de estas extensiones. Por último, la interfaz de hardware es en donde se efectúa el tercer nivel de implementación, el cual se realiza para cada tipo de driver con el que vincula a las extensiones hasta 2005 solo se soportan las tarjetas con el protocolo IEEE 802.11b.

#### ✚ *Wireless Tools.*

Las herramientas desarrolladas se encuentran en el path */proc* del sistema de archivos de Linux, el principal objetivo de estas herramientas es presentar información básica y además se pretende contar con una interfaz comprensible para el usuario.

Estas herramientas han sido muy importantes en el desarrollo de este trabajo de tesis, pues nos hemos apoyado en el funcionamiento de estas herramientas para hacer que funciones los módulos de WiNPS.

Ahora bien, las herramientas que encontramos en el sistema Linux son:

#### ✚ */proc/net/wireless*

Como bien se mencionaba, esta herramienta propiamente es un archivo que se encuentra en la entrada */proc/net/wireless*, como se recordara en */proc* es un pseudo sistema de archivos que proporciona información y estadísticas actuales del sistema, de acuerdo al tipo de interfaz de hardware, la información se ubicara en un nivel diferente.

*/proc/net/wireless* está diseñado para dar algunas estadísticas específicas de cada interfaz inalámbrica en el sistema<sup>10</sup>. Este path es básicamente un clon de */proc/net/dev* el cual da estadísticas de una interfaz ethernet.

La salida para la ejecución de esta entrada resulta:

```
Inter-| sta-| Quality      | Discarded packets      | Missed | WE  
face | tus | link level noise | nwid crypt frag retry misc | beacon | 16  
wlan0: 0000 100 228 0          0 0 0 0 22 7033 0
```

La información que se muestra estará habilitada de acuerdo al tipo de dispositivo; esta además permite al usuario tener una mejor idea acerca del sistema, es decir, un valor muy alto en *Discarded Nwid* indica que existe un problema en el ID de la red, o bien existe una red cercana. En el caso de *Quality Level* le puede proporcionar al usuario en donde existen áreas que presentan gran interferencia en la comunicación (*track shadow areas*).

---

<sup>10</sup> Hay que recordar que en cada sistema se puede tener más de una interfaz de cada tipo.

### **Parámetros.**

*Status*: El valor de este campo depende del dispositivo inalámbrico que este conectado.

*Quality Link*: Nos muestra la calidad de la recepción en el dispositivo.

*Quality Level*: Proporciona la información de la calidad en el receptor, esto lo realiza tomando en base un nivel de 255 puntos para un enlace al 100%.

*Quality Noise*: Muestra el nivel de silencio en el receptor, este no tiene que ver con la recepción de los paquetes.

*Discarded Nwid*: Proporciona el número de paquetes que han sido descartados por tener un identificador de red inválido.

*Discarded Crypt*: Este campo da el número de paquetes en los que fue imposible realice la descriptación.

*Discarded Misc*: Este es un campo libre, en la actualidad no tiene un uso.

### **iwconfig**

Esta herramienta es básicamente un clon de *ifconfig*, pero a diferencia de este, *iwconfig* está orientado a las interfaces inalámbricas. Esta herramienta ha sido diseñada para configurar todos los parámetros posibles del *driver* y del hardware, esto para lograr un modo de operación específico (por ejemplo la frecuencia).

*Iwconfig* también puede ser usado para desplegar estos parámetros, y además muestra las estadísticas provenientes de */proc/net/wireless*. Todos estos parámetros y estadísticas dependen del dispositivo, por lo que existirá el caso en que alguna de las estadísticas o parámetros no este disponible, o bien, el rango de operación no sea el mismo.

El modo de utilizar la herramienta es:

```
iwconfig [interface]
iwconfig interface [essid X]
[nwid N]
[mode M]
[freq F]
[channel C]
[sens S ]
[ap A ]
[nick NN ]
[rate R]
[rts RT]
[frag FT]
[txpower T]
[enc E]
[key K]
[power P]
[retry R]
[commit]
iwconfig --help
iwconfig --version
```

Por interfaz se entiende la asignación que Linux le hace al dispositivo de red, así, en algunos casos será eth0, eth1, wlan0 o wlan1. En el primer caso se dispondrán solo de las estadísticas del estado actual del *driver* y hardware. Como se ve, el segundo caso de uso es el más completo y se pueden apreciar los parámetros que se pueden manipular.

En el caso del tercer modo de uso, el comando proporciona ayuda acerca del uso, y finalmente en el cuarto caso se da la versión de la herramienta, actualmente la versión de las *wireless extensions* es XXXX.XX.

A continuación se presenta el resultado obtenido en la terminal en la que se desarrolla WiNPS.

```
[root@mobilelan net]# iwconfig wlan0
wlan0      IEEE 802.11g  ESSID:"mobilelan"  Nickname:"mobilelan"
          Mode:Managed  Frequency:2.437 GHz  Access Point: 00:12:17:AA:44:0C
          Bit Rate=54 Mb/s   Tx-Power:14 dBm
          RTS thr=2347 B   Fragment thr=2346 B
          Encryption key:F886-068C-7B40-6EE9-0BF1-329A-4D   Security
mode:restricted
          Power Management:off
          Link Quality:100/100  Signal level:-28 dBm  Noise level:-256 dBm
          Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
          Tx excessive retries:26  Invalid misc:9502  Missed beacon:0

[root@mobilelan net]#
```

### **Parámetros.**

*Essid.* El *Extended Service Set Identifier (ESSID)* o mejor conocido como nombre de la red, es usado para identificar las células que son partes de la misma red virtual, esto es muy diferente a la dirección MAC del AP o al *Network ID (NWID)* que definen a una célula específica.

*nwid/domain.* Con este se especifica el ID de la red, este parámetro es utilizado para diferenciar una red, pues las redes inalámbricas adyacentes comparten el mismo medio.

*freq/channel.* Este parámetro define la frecuencia de operación o el canal en el que el dispositivo va a trabajar.

*sens.* Con este parámetro se constituye el umbral de sensibilidad, este es el mínimo nivel de señal para que el hardware pueda considerar a un paquete como utilizable.

*mode.* Define el modo de operación del dispositivo, el cual depende de la topología de la red, así, podemos tener modo Ad-Hoc, Managed, Repeater, Monitor o Auto.

*ap.* Este parámetro fuerza a la tarjeta a registrar en un AP, mediante la dirección MAC de este; cuando la calidad de conexión es baja, la herramienta busca el AP con mejor calidad de señal, haciendo caso omiso a la especificación del usuario.

*nick[name].* Establece el nombre de la estación de trabajo.

*rate/bit[rate]*. Para tarjetas que soportan diferentes velocidades de transmisión de datos, esta opción permite definir en cual de ellas se estará trabajando.

*rts[\_threshold]*. RTS y CTS son elementos que se agregan a la transmisión para “escuchar” si el canal está limpio. Este parámetro configura el tamaño mínimo de un paquete que envía el RTS incrementa la eficiencia en el caso de que existiesen nodos ocultos o que hubiese una gran cantidad de nodos activos.

*frag[mentation\_threshold]*. La fragmentación permite seccionar un paquete IP en segmentos más pequeños que son transmitidos en el medio, a modo de incrementar la eficiencia. Este parámetro establece el tamaño máximo de dicha fragmentación.

*key/enc[ryption]*. Esta opción está dada para definir la llave de encriptación para una red segura.

*power*. Este parámetro es usado para manipular el esquema de administración de la potencia.

*txpower*. Este parámetro está habilitado solo para tarjetas que soportan múltiples potencias de transmisión, básicamente la idea de este es establecer una potencia de transmisión en dBm.

*retry*. Muchas de las tarjetas tienen retransmisiones de la MAC, y algunas permiten establecer el comportamiento de esta retransmisión, pues es mediante este parámetro que se puede definir estas opciones.

*commit*. En algunas tarjetas los cambios efectuados mediante las *wireless extensios* no aplican inmediatamente, y es por ello que este comando fuerza la tarjeta a aplicar cambios pendientes.

## iwspy

Esta herramienta proporciona estadísticas inalámbricas de nodos específicos, la creación de esta herramienta surge para dar soporte a Mobile IP. Mediante este comando, el usuario establecer una lista de direcciones del monitor en una interfaz de red inalámbrica, y además permite contar con la calidad de conexión para cada una de estas direcciones.

Esta información es igual a la disponible en el campo *Quality Link*, *Signal Level* y *Noise Level* de `/proc/net/wireless`. Esta información se actualiza cada vez que un nuevo paquete es recibido.

Vale la pena mencionar que esta herramienta es totalmente funcional para nodos que forman parte de la célula inalámbrica, por esta razón no se pueden monitorear APs que no están asociados a los nodos existentes.

Ahora bien, el modo de utilizar esta herramienta es:

```
iwspy interface
iwspy interface [+] DNSNAME | IPADDR | HWADDR [...]
iwspy interface off
iwspy interface setthr low high
iwspy interface getthr
```



### Parámetros.

*DNSNAME* | *IPADDR*. Define una dirección IP o en algunos casos el DSN, como el hardware trabaja con direcciones de hardware<sup>11</sup>, *iwspy* traduce estas direcciones a través de ARP.

*HWADDR*. Establece una dirección de hardware, el formato para esta dirección es *XX:XX:XX:XX*.

+. Mediante este parámetro se agrega un nuevo conjunto de direcciones en la lista actual, dicha lista es única para cada dispositivo.

*Off*. Borra la lista actual de direcciones y deshabilita la función de monitoreo.

*Setthr*. Define el umbral mínimo y máximo de la intensidad de la señal para la función de monitoreo.

*Getthr*. Recupera el umbral actual de la intensidad de la señal para la función de monitoreo.

### iwpriv

Esta herramienta es un complemento de *iwconfig*, a diferencia de este, *iwpriv* trabaja sobre parámetros específicos de cada driver. Sin argumentos, *iwpriv* enlista los comandos disponibles para cada interfaz de hardware y los parámetros que se requieren para trabajar.

Usando esta información, el usuario podría aplicarla solo para comandos específicos de la interfaz con la que trabaja, es por ello que la documentación de cada driver podría indicar como usar esta información.

Para nuestro driver de una tarjeta Linksys *Wireless G*, la información desplegada por el comando fue:

```
[root@mobilelan net]# iwpriv wlan0
wlan0   Available private ioctl :
        setwpa           (8BE1) : set   1 int   & get   0
        setkey           (8BE2) : set   1 int   & get   0
        associate       (8BE3) : set   1 int   & get   0
        disassociate    (8BE4) : set   1 int   & get   0
        drop_unencrypted (8BE5) : set   1 int   & get   0
        countermeasures (8BE6) : set   1 int   & get   0
        deauthenticate  (8BE7) : set   1 int   & get   0
        auth_alg        (8BE8) : set   1 int   & get   0
        ndis_reset      (8BF0) : set   0       & get   0
        power_profile   (8BF1) : set   1 int   & get   0
        network_type    (8BF2) : set   1 char  & get   0

[root@mobilelan net]#
```

---

<sup>11</sup> A estas direcciones comúnmente se les conoce como direcciones MAC



## iwlist

El uso que se le da a esta herramienta es para obtener información adicional de una interfaz de red inalámbrica que no aparece desplegada por *iwconfig* e *iwpriv*. El principal argumento de esta herramienta es usado para seleccionar una categoría de información.

*Iwlist* despliega en forma detallada toda la información relacionada a esta categoría, incluyendo también la desplegada por *iwconfig*.

La sintaxis para esta herramienta es:

```
iwlist interface scanning
iwlist interface frequency
iwlist interface rate
iwlist interface key
iwlist interface power
iwlist interface txpower
iwlist interface retry
iwlist interface event
iwlist --help
iwlist --version
```

### Parámetros

*scan[ning]*. Proporciona la lista de APs y células Ad-hoc en el rango de alcance de la tarjeta, además muestra una información completa sobre este (ESSID, Calidad, Frecuencia, Modo de Operación, etc.). Resulta obvio mencionar que la información dependiera de las limitaciones de la propia tarjeta.

*freq[uecy]/channel*. Obtiene una lista de las frecuencias disponibles en el dispositivo, así como los canales definidos para la misma.

*rate/bit[rate]*. Enlista las velocidades de transmisión soportadas por el dispositivo.

*key/enc[ryption]*. Devuelve una lista de los tamaños de las llaves de encriptación y despliega las llaves disponibles para el dispositivo.

*Power*. Enlista diversos atributos para la administración de la potencia así como los modos del dispositivo.

*Txpower*. Proporciona una lista de los valores definidos en el dispositivo para la potencia de transmisión.

*Retry*. Obtiene una lista de los límites de retransmisión y el tiempo de vida para esta.

*ap/accesspoint/peers*. Esta función es obsoleta, pues fue sustituida por *scanning*,

*event*. Enlista los eventos inalámbricos soportados por el dispositivo.

*--version*. Proporciona la versión de la herramienta.

Para el caso de la terminal en la que se desarrolla WinPS, el resultado de la ejecución de esta herramienta es:

```
[root@mobilelan net]# iwlist wlan0 scan
wlan0    Scan completed :
          Cell 01 - Address: 00:12:17:AA:44:0C
                    ESSID:"mobilelan"
                    Protocol:IEEE 802.11b
                    Mode:Managed
                    Frequency:2.437 GHz (Channel 6)
                    Quality:0/100  Signal level:-32 dBm  Noise level:-256 dBm
                    Encryption key:on
                    Bit Rate:1 Mb/s
                    Bit Rate:2 Mb/s
                    Bit Rate:5.5 Mb/s
                    Bit Rate:11 Mb/s
                    Bit Rate:18 Mb/s
                    Bit Rate:24 Mb/s
                    Bit Rate:36 Mb/s
                    Bit Rate:54 Mb/s
                    Extra:bcn_int=100
                    Extra:atim=0

          Cell 02 - Address: 00:12:17:18:F7:B8
                    ESSID:"mobilelan"
                    Protocol:IEEE 802.11b
                    Mode:Managed
                    Frequency:2.437 GHz (Channel 6)
                    Quality:0/100  Signal level:-74 dBm  Noise level:-256 dBm
                    Encryption key:on
                    Bit Rate:1 Mb/s
                    Bit Rate:2 Mb/s
                    Bit Rate:5.5 Mb/s
                    Bit Rate:11 Mb/s
                    Bit Rate:18 Mb/s
                    Bit Rate:24 Mb/s
                    Bit Rate:36 Mb/s
                    Bit Rate:54 Mb/s
                    Extra:bcn_int=100
                    Extra:atim=0

Extra:wpa_ie=dd160050f20101000050f20201000050f20201000050f201

[root@mobilelan net]#
```

Por otro lado, volviendo a la implementación de la *wireless extensions*, encontramos que las modificaciones en el *kernel* han sido incluidas en las últimas versiones de Linux, es decir, desde la versión 2.0.30 y 2.1.17 se han incluido dichas extensiones a modo de que el *kernel* actual las soporte.

Las *wireless tools* están deshabilitadas por defecto, para hacer uso de ellas, los usuarios necesitan la opción `CONFIG_NET_RADIO` en la configuración del *kernel*<sup>12</sup>, esta opción habilita la elección de las interfaces inalámbricas disponibles de una lista, previamente probada, para cada versión de Linux. Las herramientas son bastante sencillas y también tiene la facilidad de compilarse en cualquier distribución de Linux que tenga el soporte en el *kernel*.

---

<sup>12</sup> Este procedimiento se describe con más detalle en el apéndice B.

Asimismo, existe una particularidad muy importante de estas herramientas, esta consiste en las modificaciones de los drivers de una red inalámbrica, es decir, se requiere que exista una compatibilidad entre *driver - kernel*, para que exista un soporte sobre el cual se implementen las *wireless extensions*.

En el momento que comenzó a desarrollarse este trabajo de tesis, los únicos drivers compatibles, son los correspondientes a las tarjetas inalámbricas basadas en el protocolo 802.11b y algunas tarjetas 802.11g que solo estaban disponibles en Estados Unidos y España.

Ahora bien, como se recordará, en el capítulo anterior se había comentado que el entorno en el que se va a estar trabajando está basado en el IEEE 802.11g, por lo que en el dispositivo móvil no podríamos instalar una tarjeta del tipo b.

Esta es una de las *barreras técnicas* que mencionábamos en el capítulo tres, para solventarla, requerimos de un intérprete que nos permita conectar una tarjeta no compatible con el *kernel* original de Linux a modo de poder habilitar las *wireless extensions*. En el siguiente inciso la siguiente sección se describe el funcionamiento de este intérprete, y de cómo es posible que nos permita la conexión e interacción con la red.

#### 4.3.2.2 *KWiFiManager*.

El paquete *KWiFiManager* es un juego de herramientas que permiten administrar las tarjetas de red inalámbrica LAN (tarjeta PC, PCI o miniPCI) bajo el entorno de escritorio K<sup>13</sup>. Proporciona información sobre su conexión actual y le permite establecer hasta diez configuraciones independientes y utilizar cuatro configuraciones preconfiguradas por scripts específicos de una distribución.

Si se encuentra en un lugar en el que no hay ninguna red preconfigurada disponible, puede cambiar de forma dinámica a una red disponible sin realizar esfuerzos para configurarla. *KWiFiManager* soporta todas las tarjetas para redes inalámbricas LAN existentes que utilizan la interfaz de extensiones inalámbricas. Esto incluye todas las tarjetas inalámbricas LAN que funcione en el sistema operativo Linux.

Para la ejecución de esta herramienta puede ser a partir del comando *KWiFiManager* desde el shell de Linux, o bien a través del *Menú K*<sup>14</sup>, donde de forma predeterminada está localizada en el grupo Aplicaciones. Si *KWiFiManager* ya se está ejecutando pero está minimizado en la bandeja del sistema puede restaurarlo pulsando una vez en el icono de la bandeja del sistema.

Si existe más de una tarjeta inalámbrica LAN en su sistema, abra más de una instancia de *KWiFiManager*; cada instancia mostrará información sobre una tarjeta diferente de forma automática. Los elementos de la interfaz de la aplicación son:

---

<sup>13</sup> Esto se refiere a la interfaz gráfica basada en librerías Qt que sirve como un layer de comunicación entre el kernel de Linux y el usuario.

<sup>14</sup> EL menú K es el modo análogo al menú inicio de Windows, solo que este es usado para el escritorio KDE.

#### ✦ *Ventana principal*

La ventana principal de *KWiFiManager* está formada por cinco partes:

- ✦ *Visualización de la calidad de la señal.* Aquí puede ver la calidad y tipo de la conexión activa.
- ✦ *Velocidad de la conexión.* Un indicador de la velocidad actual de la conexión aparece en la parte derecha de la ventana principal, sobre la información de configuración. Si los ajustes de velocidad se establecen en AUTO, el valor cambiará una vez que la tarjeta ajuste la velocidad de la conexión de acuerdo a la calidad de la señal. La escala del gráfico de barras se ajustará automáticamente por encima de 108 MBit/s cuando la velocidad de conexión sobrepase los 11 MBit/s.
- ✦ *Configuración actual.* Aquí puede encontrar información acerca de la configuración de su tarjeta.
- ✦ *Información del Punto de Acceso.* La última línea de la ventana principal, muestra información acerca de su Punto de acceso.
- ✦ *Información sobre las redes disponibles.* El área inferior izquierda de la ventana principal contiene un botón llamado Buscar redes. Si pulsa este botón, *KWiFiManager* intentará recuperar una lista con todas las redes que estén en el rango de su tarjeta.

#### ✦ *Visualizador de estadísticas.*

Opcionalmente, seleccionando Estadísticas de conexión en el menú Archivo, una ventana separada aparece mostrando el gráfico de nivel de señal y nivel de ruido en rojo. La diferencia (SEÑAL menos RUIDO) es la calidad de la señal, la cual se muestra en la ventana principal.

Algunas tarjetas no tienen información significativa sobre el ruido. Si este es el caso de su tarjeta y le molesta la línea roja que es irrelevante, desactive el mostrar el nivel de ruido en la ventana de las estadísticas quitando la selección en *Preferencias->Mostrar gráfica de ruido* en la ventana de estadísticas en la ventana principal de *KWiFiManager*.

#### ✦ *Editor de configuración.*

Desde aquí se accede al módulo de configuración en el Centro de control de *Common Desktop Environment (KDE)*, esta es quizá la parte más práctica del paquete *KWiFiManager*. Aquí se pueden cambiar las configuraciones básicas de su tarjeta LAN inalámbrica. El módulo puede administrar hasta diez configuraciones independientes para la tarjeta.

#### 4.3.2.3 *KwaveControl.*

*KWaveControl* es una herramienta para dispositivos inalámbricos, esta utilidad está basada también en las *wireless extensions*. Cuando se ejecuta sobre KDE, aparece un pequeño panel en la bandeja de iconos de la barra de tareas, el cual muestra la calidad con la que se recibe actualmente la señal, así como la velocidad de transmisión.

Cuando se restaura dicho panel, la ventana principal proporciona la información del dispositivo, las propiedades que despliega las podemos ver en la tabla 4.2.

IP	DEFAULT GATEWAY	LINK QUALITY
SIGNAL LEVEL	NOISE LEVEL	BIT RATE
PROTOCOL	OPERATION MODE	SENSITIVITY
FREQUENCY	CHANNEL	TRANSMIT POWER
ESSID	ENCRYPTION	MAC ADDRESS OF THE ACCESS POINT
SENT AND RECEIVED BYTES	NICKNAME	RTS THRESHOLD
FRAGMENTATION THRESHOLD	POWER MANAGEMENT	TRAFFIC
INVALID RX'S AND TX'S		

*Tabla 4.2 Propiedades que pueden ser obtenidas mediante KWaveControl.<sup>15</sup>*

A partir de la versión 0.4 se han ido adaptando nuevas funcionalidades, en la figura 4.12 se puede ver el aspecto de la versión más actual de esta herramienta y como para cada nueva funcionalidad, se ha agregado una nueva pestaña a esta aplicación.



*Figura 4.12 GUI de KWaveControl.*

#### 4.3.2.4 WaveLANApplet.

Esta es una herramienta más básica que las dos anteriores, desde su concepción, esta herramienta fue diseñada como un accesorio informativo de la barra de tareas del escritorio de KDE y *GNU Network Object Model Environment (GNOME)*.

Se puede decir que *Wavelan-applet* muestra la intensidad de la señal en una tarjeta inalámbrica en seis pasos (uno de ellos para indicar la ausencia de dispositivo, otro cuando no existe señal y los cuatro últimos para indicar suficiente, regular, buena y excelente señal).

<sup>15</sup> Las propiedades marcadas en cursiva solo estarán disponibles si el dispositivo las soporta.

Actualmente se encuentra disponible la versión 0.8 que proporciona el porcentaje de la intensidad de señal; en la figura 4.13 se muestra la ejecución en tiempo real de la aplicación.



Figura 4.13 GUI de WaveLANApplet.

### 4.3.3 Programación Gráfica en Linux.

Al contrario que otros sistemas operativos, por defecto el trabajo con Linux no se realiza de una forma gráfica, sino introduciendo comandos de forma manual. Linux dispone de varios programas que se encargan de interpretar los comandos que introduce el usuario y realiza las acciones oportunas en respuesta.

Estos programas denominados *shell* son el modo típico de comunicación en todos los sistemas Unix incluido Linux. Para muchas personas el hecho de tener que introducir los comandos de forma manual les puede parecer intimidante y difícil.

No obstante, casi todas las distribuciones más recientes incluyen el sistema *X Window* (no *X Windows*), el cual es el encargado de controlar y manejar la interfaz de usuario. *X Window* es mucho más poderoso que otros entornos similares como *Microsoft Windows*, puesto que permite que el usuario tenga un control absoluto de la representación de los elementos gráficos.

En nuestro diagrama de bloques que se muestra en la figura 4.14 se puede ver como *X Window* se diseña como una capa interprete entre usuario y el sistema de archivos de Linux, permitiendo una mejor comunicación.

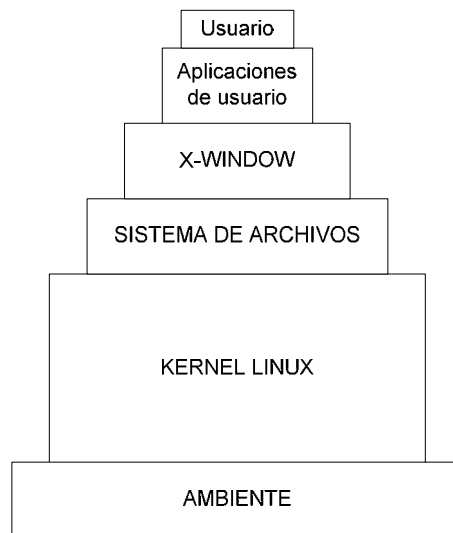


Figura 4.14 Relación de X Window con el OS Linux.

El sistema de ventanas *X Window* fue desarrollado a mediados de los años 80 en el MIT para dotar de una interfaz gráfica a los sistemas Unix. Este protocolo permite la interacción gráfica en red entre un usuario y una o más computadoras haciendo transparente la

red para éste. Generalmente se refiere a la versión 11 de este protocolo, X11, el que está en uso actualmente.

*X Window* es el encargado de visualizar la información gráfica y es totalmente independiente del sistema operativo. El sistema de ventanas X distribuye el procesamiento de aplicaciones especificando enlaces cliente-servidor.

Asimismo tenemos que *X Window* provee servicios para acceder a la pantalla, teclado y ratón, mientras que los clientes son las aplicaciones que utilizan estos recursos para interacción con el usuario. De este modo mientras el servidor se ejecuta de manera local, las aplicaciones pueden ejecutarse remotamente desde otras máquinas, proporcionando así el concepto de transparencia de red.

Con esta base, al momento de que se desea desarrollar programación visual hay que mencionar que en Linux también existen distintos Entornos de Desarrollo Integrado, como son *Glade*, *KDevelop* y algunos más.

Ahora bien, para estos entornos de desarrollo existen dos importantes bibliotecas en la que se basan para desarrollar GUI's, estas son GTK y Qt principalmente, en la actualidad existen más rutinas pero estas dos son las que prevalecen para el entorno GNOME y KDE.

#### 4.3.3.1 Qt.

Qt es una biblioteca multiplataforma para desarrollar interfaces gráficas de usuario. Fue creada por la compañía noruega Trolltech, en la figura 4.15 se puede ver el logotipo de Qt. Esta biblioteca es utilizada en KDE, un entorno de escritorio para sistemas como Linux o FreeBSD, entre otros. Utiliza el lenguaje de programación C++ de forma nativa y además existen *bindings*<sup>16</sup> para C, Python, Perl y Ruby entre muchos.

El API de la biblioteca cuenta con métodos para acceder a bases de datos mediante SQL, así como uso de XML y una multitud de otros para el manejo de ficheros, además de estructuras de datos tradicionales.

Inicialmente Qt apareció como biblioteca desarrollada por Trolltech (en aquel momento "Quasar Technologies") en 1992 siguiendo un desarrollo basado en el código abierto, pero no libre. Se usó activamente en el desarrollo del escritorio KDE (entre 1996 y 1998), con un notable éxito y rápida expansión. Esto fomentó el uso de Qt en programas cerrados para el escritorio, situación vista por el proyecto GNU como amenaza para el software libre.



*Figura 4.15 Logotipo de Qt.*

Para contrarrestar la situación se plantearon dos ambiciosas iniciativas, por un lado el equipo de GNU en 1997 inició el desarrollo del entorno de escritorio GNOME con GTK+

---

<sup>16</sup> Es una "ligadura" o referencia a otro símbolo más largo y complicado, y que se usa frecuentemente. Este otro símbolo puede ser un valor de cualquier tipo, numérico, de cadena, etc. o el nombre de una variable que contiene un valor o un conjunto de valores

para GNU/Linux. Por otro lado intentan hacer una biblioteca compatible con Qt pero totalmente libre, llamada Harmony.

En noviembre de 1998, anuncian el cambio de licencia de Qt que, a pesar de todo, no contaba con el beneplácito de la importante *Free Software Foundation*. El 4 de septiembre de 2000, Trolltech por fin ofrece la versión 2.2 con licencia *General Public License (GPL)*.

Qt cuenta actualmente con un sistema de doble licencia que merece una pequeña explicación. Qt cuenta con una licencia GPL para el desarrollo de software de código abierto (open source) y software libre y otra licencia de pago para el desarrollo de aplicaciones comerciales.

Actualmente se encuentra la versión 4 de la biblioteca, y además de las múltiples mejoras, ahora las librerías Qt son también liberadas bajo la GPL para Windows.

#### 4.3.3.2 GTK+.

Este es un grupo importante de bibliotecas o rutinas para desarrollar GUIs principalmente para los entornos gráficos GNOME, *XForms Common Environment (XFCE)* y *RISC OS on X (ROX)* de sistemas Linux. Es software libre (bajo la licencia LGPL<sup>17</sup>), multiplataforma y parte importante del proyecto GNU.

Inicialmente creado para construir el programa gráfico *GNU Image Manipulation Program (GIMP)*, GTK+ es la abreviatura de GIMP toolkit (conjunto de rutinas para GIMP) y es muy usada por los programadores de sistemas Linux junto con Qt del entorno KDE. En la figura 4.16 se aprecia el logotipo de esta biblioteca.



*Figura 4.16 Logotipo de GTK.*

GTK+ se ha diseñado para permitir programar con lenguajes como C, C++, Perl o Python. Actualmente su última versión es GTK+ 2, con una cantidad importante de mejoras respecto a la primera versión, aunque sin embargo, no es compatible con su primera versión. GTK+ se basa en tres bibliotecas del equipo de GTK+:

- ✚ *GLib* es una biblioteca de bajo nivel estructura básica de GTK+ y GNOME. Proporciona manejo de estructura de datos para C, portabilidad, interfaces para funcionalidades de tiempo de ejecución (runtime) como ciclos, hilos, carga dinámica o un sistema de objetos.
- ✚ *Pango* es una biblioteca para el diseño y renderizado de texto, hace hincapié especialmente en la internacionalización. Es el núcleo para manejar las fuentes y el texto de GTK+ 2.

---

<sup>17</sup> *Lesser General Public License (LGPL)* es una licencia que aplica a cualquier programa o trabajo que contenga una nota puesta por el propietario de los derechos del trabajo, estableciendo que puede ser distribuido bajo los términos de esta.



- ✚ ATK es una biblioteca para crear interfaces con características de una gran accesibilidad muy importante para personas discapacitadas o minusválidos. Pueden usarse útiles como lupas de aumento, lectores de pantalla, o entradas de datos alternativas al clásico teclado o ratón de ordenador.

#### 4.3.3.3 Borland Kylix, el entorno de desarrollo para WiNPS.

Para el desarrollo de WiNPS se eligió como entorno de desarrollo Kylix 3.0, esta decisión fue tomada considerando las bondades de este entorno de desarrollo. A continuación se mencionara un poco más sobre el mismo, esto a modo de que se entiendan las características más atractivas de Kylix.

Para comenzar encontramos que Borland Kylix permite la creación de una manera sencilla y rápida de sofisticadas aplicaciones de alto rendimiento en Linux. Básicamente, podríamos decir que es una versión de Delphi para Linux, con características similares a las que en su día tuvo Delphi 4 y algunos de los elementos de Delphi 5. En la figura 4.17 se ve el logotipo de Kylix 3.0.



*Figura 4.17 Logotipo de Kylix 3.0.*

No obstante, Kylix no es una simple conversión de Delphi para el sistema operativo GNU/Linux, además podemos decir que incorpora elementos que no se tenían contemplados en las versiones de Delphi. Tal es el caso de la librería de componentes multi-plataforma, conocida como *Cross Platform Component Library (CLX)*; el cual es un mecanismo de acceso a bases de datos alternativo al *Borland Database Engine (BDE)*, integración con Apache, entre otras más.

Dicha la librería de componentes para desarrollo multi-plataforma permite reutilizar, personalizar y extender componentes para la creación de aplicaciones gráficas y que acelera considerablemente el tiempo en que una aplicación sale al mercado, bien sea basada en la Web, en bases de datos o en entornos de escritorio.

#### ✚ *La interfaz de usuario.*

El entorno de desarrollo de Kylix es del tipo RAD y para usarlo se debe de trabajar con un entorno gráfico, como KDE o GNOME. El compilador puede ser utilizado directamente desde la línea de comandos de Linux pero, lógicamente, perderíamos la mayor parte de sus bondades.

Tanto Kylix como las aplicaciones que generemos con él pueden funcionar en los dos escritorios KDE y GNOME, no importa, por tanto, cual de los dos usemos. Lógicamente, también podemos generar aplicaciones no gráficas, que funcionarían directamente desde la línea de comandos del sistema.

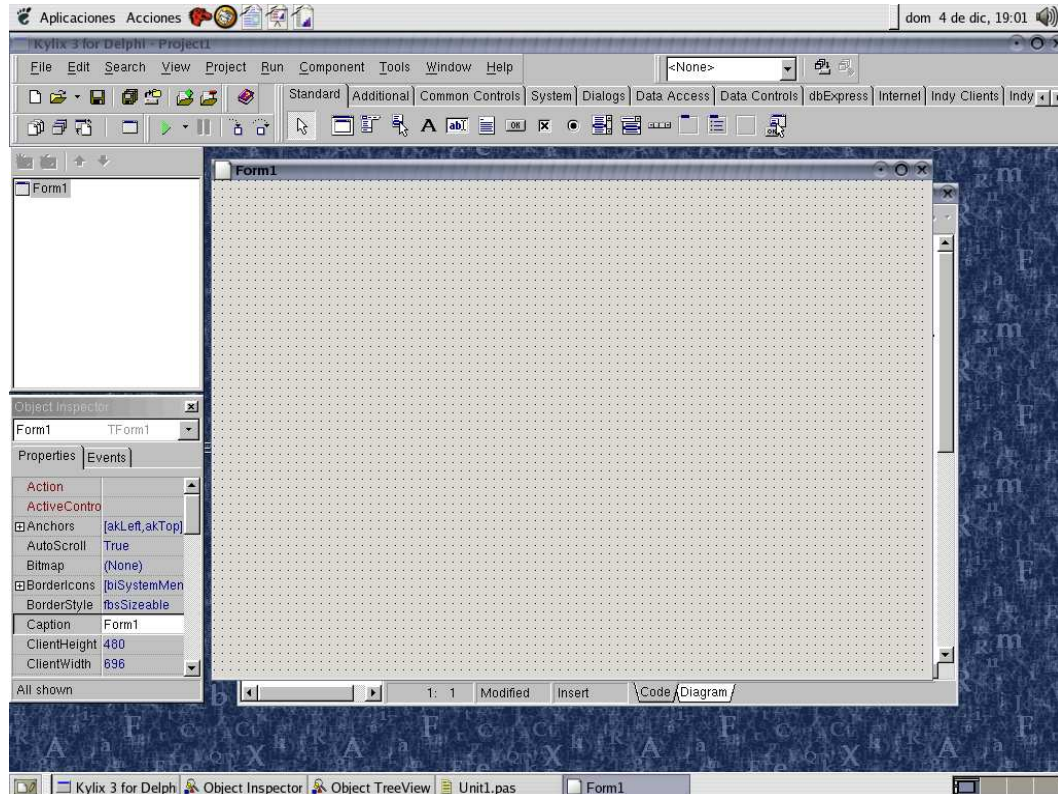


Figura 4.18 Interfaz de Usuario de Kylix 3.0.

En la figura 4.18 tenemos la interfaz de usuario, la cual es bastante parecida a la Delphi 5. En esta tenemos un menú de opciones, paletas de botones, la paleta de componentes, el Inspector de objetos (figura 4.19), el Editor de código y un formulario.

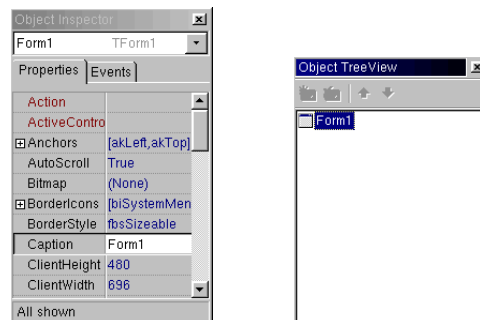


Figura 4.19 El inspector de objetos y Object Tree View.

La mayor parte de los objetos que encontramos en la paleta de componentes son idénticos o muy parecidos a los que existen en Delphi, lo cual allana el camino para cualquier usuario de esta conocida herramienta de Borland.

La metodología de trabajo es exactamente la misma que podemos encontrar en otros entornos RAD. Mediante operaciones de arrastrar y soltar gestionamos los componentes, personalizándolos en el Inspector de Objetos y asociando código a los eventos.

El Editor de código dispone de los mismos sistemas de asistencia a la edición, facilitando listas de métodos y propiedades o ventanas emergentes con información sobre parámetros. Las configuraciones de escritorio pueden almacenarse y recuperarse, co-

mo en Delphi 5. También tenemos a nuestra disposición la lista de tareas, así como todas las ventanas auxiliares para gestión de proyectos y depuración.

Lógicamente, todos los elementos de la interfaz de usuario de Kylix son configurables, las ventanas pueden adosarse, las paletas de componentes modificarse y cambiar de lugar, por ejemplo convirtiéndose en ventanas flotantes; el funcionamiento del editor puede personalizarse, etc.

⊕ *El compilador.*

Detrás de la interfaz de usuario que acabamos de describir, se encuentra el motor que hace funcionar a Kylix, el compilador. Concretamente se trata de la versión 14 del compilador de Object Pascal que Borland creó originalmente para Turbo Pascal, modificado para funcionar en Linux y generar tanto ejecutables como librerías o *shared object libraries*.

Este compilador sigue produciendo código x86, lo cual significa que Kylix no puede ser utilizado para desarrollar aplicaciones Linux para otras plataformas de procesador. El código generado es nativo y optimizado, lo cual significa que no es necesaria ninguna capa intermedia de emulación, como pudiera ser Wine<sup>18</sup>. Esto permite una distribución teóricamente fácil y sin necesidad de pagar derechos adicionales por el uso de librerías de soporte.

Una característica interesante del compilador de Kylix es la incorporación de BASM (*Built-in assembler*), un ensamblador incorporado que nos permite introducir en el código de los módulos instrucciones directamente en ensamblador. Se contempla el uso de las instrucciones de Pentium Pro, Pentium II, MMX, AMD 3DNow! y SSE, lo cual permite efectuar optimizaciones manuales allí donde se precise sin necesidad de recurrir a un ensamblador externo.

⊕ *La librería de componentes CLX.*

Aparte de la novedad de ser uno de los primeros entornos de desarrollo profesional y comercial para Linux, seguramente entre las características más importantes de Kylix se encuentra la aportación de una nueva librería de componentes.

Esta librería viene a ser equivalente a la *Visual Component Library (VCL)* que utilizan Delphi y C++ Builder en Windows, se llama CLX. Su propiedad más destacable radica en estar diseñada desde el principio como una solución multiplataforma.

Al iniciar el proyecto Kylix, Borland tenía como objetivo convertir su actual VCL para poder usarla en Linux. Este objetivo, no obstante, se presentó como un importante desafío por la fuerte dependencia que la VCL tenía respecto al API de Windows.

Para recrear todos los elementos gráficos, esenciales a la hora de diseñar interfaces de usuario, existían básicamente, dos opciones posibles, trabajar directamente con el gestor de ventanas o usar una librería de *widgets* o controles ya existente. Se tomó la segunda opción, iniciándose entonces una nueva decisión: ¿qué librería usar? Las dos más difundidas eran GTK y Qt, asociadas tradicionalmente a los escritorios GNOME y KDE, respectivamente.

---

<sup>18</sup> Emulador de la plataforma de Windows sobre Linux.

Tras un largo periodo durante el cual lo único que trascendió es que Borland no tomaba la decisión, llegó la noticia de que Qt, de la empresa noruega Trolltech, sería la base para la nueva CLX.

Esta librería, esencialmente de controles gráficos, está disponible en varios sistemas operativos, lo cual posibilita que la CLX pueda ser usada, sin cambio alguno, en Linux y Windows. Se alcanzaba así el objetivo de la multiplataforma sin necesidad de partir totalmente desde cero.

CLX, sin embargo, no es simplemente un envoltorio escrito en Object Pascal en torno a la librería Qt. Además de los elementos de interfaz de usuario, englobados en lo que se denomina VisualCLX, también se han recreado otros elementos como las clases de utilidad general, componentes para acceso a datos o que hacen posible la comunicación en redes Internet/intranet. Esas clases y componentes se agrupan en categorías llamadas BaseCLX, DataCLX y NetCLX.

Hay que destacar el hecho de que cualquier aplicación escrita con Kylix que tenga interfaz de usuario necesita las librerías de Qt para poder funcionar. Borland llegó en su momento a un acuerdo con Trolltech para que los usuarios de Kylix puedan redistribuir Qt, si es necesario, sin que por ello tengan que pagar licencia alguna.

#### 4.3.4 Características utilizadas en el sistema del cliente y el servidor.

En esta sección hablaremos del tema correspondiente a los elementos con los que contamos, o mejor dicho, las características de la infraestructura sobre las que se va a montar WiNPS, tanto en la parte del cliente, como en el servidor; con esto pretendemos tener un panorama completo y consistente del funcionamiento del motor de localización.

Dentro de las especificaciones que debe de cumplir el cliente, encontramos en primer lugar el OS, muchas fueron las discusiones sobre el tipo de plataforma sobre la que se debería montar la aplicación, se analizaron los pros y contras entre una y otra plataforma existente.

Inicialmente se había pensado en utilizar el OS de Windows, sin embargo, este presentaba muchos aspectos desfavorables, el principal era que no existen los suficientes permisos para acceder a la capa del hardware de la tarjeta inalámbrica. A primera vista se podría pensar que este no es un gran impedimento, sin embargo, el problema radica en que WiNPS quedaría limitado para un chip de tarjeta inalámbrica.

Así pues, y como lo hemos ido mencionado en múltiples secciones de este trabajo, decidimos que el sistema se desarrollaría un OS Linux, dicha elección se realizó principalmente por los siguientes puntos:

- ✦ *La versatilidad que presenta para el manejo de la capa de hardware.*
- ✦ *No se tienen que solicitar permisos especiales al fabricante del hardware,*
- ✦ *Y como consecuencia de ambos puntos, el programa no estaría supeditado a un solo fabricante de tarjetas inalámbricas.*
- ✦ *En Linux se pueden desarrollar aplicaciones bastante amigables, como es el caso de los desarrollos RAD realizados con Kylix.*

Con base en estos puntos y tomando en cuenta lo que mencionamos en la sección de las herramientas de monitoreo, podemos recapitular lo que mencionábamos en el capítulo anterior, es decir, las características que debe de tener el sistema cliente deben de estar apegadas a los siguientes puntos:

- ✦ *OS*: El cliente debe contar con un sistema operativo Linux.
- ✦ *Interfaz de red*: Se debe de contar con una tarjeta para conexión a la red inalámbrica basada en el protocolo 802.11 b o g, con la cual se leerán las variables físicas recogidas del *workspace*.
- ✦ *Wireless Tools*: La versión de Linux debe de tener instalada la última versión de las *wireless tools*.
- ✦ *Conexión Inalámbrica*: El sistema debe de estar asociado a una red que se base en el protocolo 802.11 b o g.
- ✦ *Entorno de desarrollo*: El sistema operativo debe soportar la instalación de Kylix 3.0.

Ahora bien, hablando específicamente de los elementos reales del sistema, se tiene que el sistema cliente sobre el que se desarrollo WiNPS fue una laptop con las siguientes características:

- ✦ *Características generales*: Laptop Dell modelo 700m, con procesador a 1.5 GHz y 512 MB en RAM.
- ✦ *OS*: La computadora posee un OS Windows XP y una la distribución Fedora de Linux en su versión 4.0.
- ✦ *Interfaz de Red*: Se cuenta con una tarjeta inalámbrica marca Linksys modelo WPC54G conectada al puerto PCMCIA de la computadora, esta tarjeta es compatible con los protocolos 802.11 b y g.
- ✦ *Wireless Tools*: Se tiene instalada la versión 2.50.
- ✦ *Entorno de desarrollo*: Se tiene instalada la versión 3.0 de Kylix.
- ✦ *Otras consideraciones*: Se instaló la versión 1.1 de Ndiswrapper para poder manipular la interfaz inalámbrica.

Por otro lado, en la parte del servidor, tenemos que mencionar que cuando el motor de búsqueda WiNPS estaba en la fase de desarrollo, una de las premisas fundamentales era el ambiente en donde iba a estar alojada tal aplicación.

Esto viene a razón de que ya existía un lugar físico sobre el que se debía trabajar, por lo que aplicación debería estar encamina hacia las características físicas del lugar, así como a las características de la infraestructura de la red.

El lugar donde se desarrolló el presente trabajo de tesis fue el *Laboratorio de Redes* que pertenece a la *División de Ingeniería de la Facultad de Ingeniería*.



En la figura 4.20 se muestra la distribución de la red con la que cuenta laboratorio.

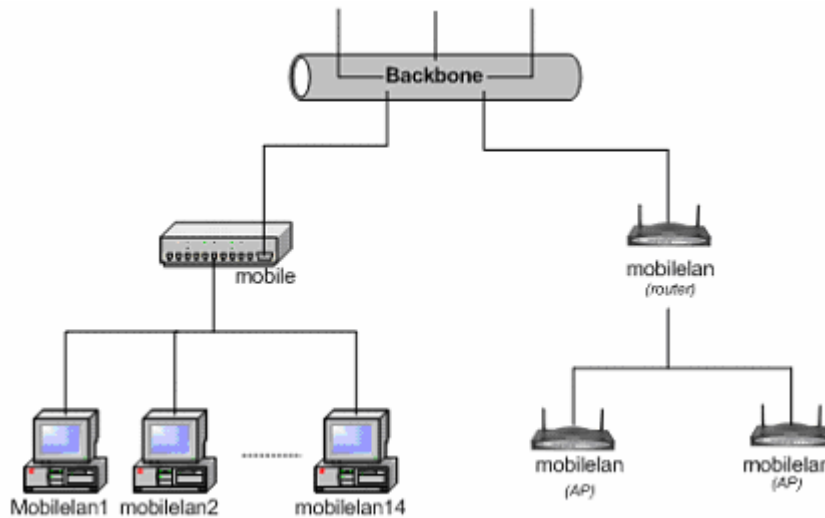


Figura 4.20 Infraestructura de la red sobre la que se monta WiNPS.

Se puede ver que se cuenta con una red del laboratorio cuenta con dos secciones:

- ✦ *Un segmento alámbrico.* Este está basado en IEEE 802.3 (Ethernet), dicho segmento cuenta con 14 máquinas.
- ✦ *Un segmento inalámbrico.* Este segmento está conformado por dos Access Points y un *router* que se basan en el protocolo 802.11g, en cada uno son aceptados hasta 20 máquinas que cuenten ya sea con el protocolo mencionado o con 802.11b. En este punto vale la pena precisar que el NWID de ambos segmentos fue definido con el nombre de *mobilelan*, esto a modo de unificar las ambos tipos de interfaces.

En este punto del presente trabajo, resulta importante menciona que *para el correcto funcionamiento de WiNPS solo se necesita estar dado de alta en la red, no es necesario que se tenga comunicación con el servidor.* Esto se debe a que WiNPS funciona sobre las intensidades de señal de los puntos de acceso disponible para el usuario.

Así pues, podemos decir que WiNPS es una aplicación que se instala en el cliente y que solo tendrá funcionamiento en el, desde su concepción y durante su desarrollo se trato de que la aplicación fuera lo más genérica posible. Es por ello que no se requiere alguna característica específica del sistema que funge como servidor, solo se requieren tres direcciones MAC de los APs.

#### 4.4 Documentación del software.

Con toda la teoría que se ha venido desarrollando desde el inicio de este escrito, hemos llegado al momento en el que describiremos la parte funcional del sistema; ahora bien, resulta necesario hacer algunas acotaciones que engloban el contenido de todos los temas y capítulos anteriores; así pues, tenemos lo siguiente:

- ✦ La aplicación WiNPS se basa en el sistema de localización denominado *mapping*.
- ✦ Para su funcionamiento se requiere un OS Linux, pues permite una manipulación más abierta sobre la capa de hardware.
- ✦ Se requiere de una interfaz de hardware compatible con el protocolo 802.11 b o g.
- ✦ Se requieren necesariamente de tres APs para el correcto funcionamiento, pues el algoritmo de aproximación realiza correcciones en base a la intensidad de señal recibida por cada AP.
- ✦ La intensidad de la señal no es constante en el mismo punto, por lo que la medición no es repetible cuando el usuario se sitúa en un punto, es por ello que se toma un tiempo de estabilización así como un algoritmo para tratar de homologar las mediciones en ese punto.
- ✦ WiNPS al basarse en *mapping*, no requiere de comunicación con el servidor, solo se requiere de las direcciones MAC de los APs para monitorear la intensidad de señal recibida en el sistema cliente.
- ✦ Como consecuencia del punto anterior, WiNPS es una aplicación cliente y solo se ejecuta desde el dispositivo móvil.
- ✦ Para el desarrollo de WiNPS se requirió del paquete de desarrollo Kylix, el cual se basa en Object Pascal.

##### 4.4.1 Diagramas de Flujo de Datos.

En el capítulo tres mencionábamos ya el flujo básico de información que sigue WiNPS, así pues, reproduciendo el flujo de información obtenemos la figura 4.21.

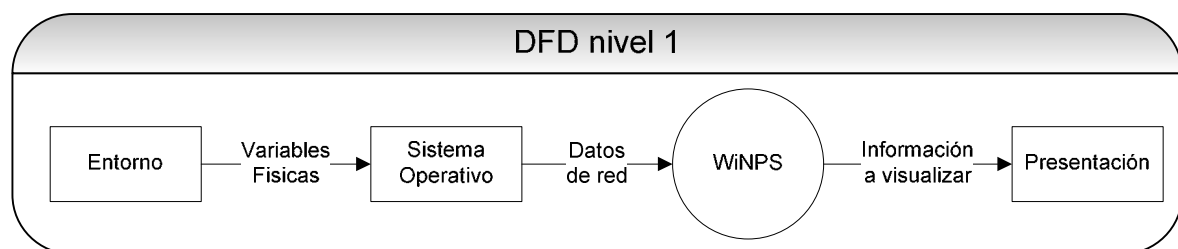


Figura 4.21 Diagrama de Flujo de Datos nivel 1 de WiNPS.

Esta es la esencia del funcionamiento de la aplicación, esto es, se recoge la información de las variables físicas del entorno, tales como intensidad de señal, calidad, ruido, etc.; posteriormente estas variables son procesadas por la capa NDIS del propio OS con lo que se les asigna un valor en específico legible para la aplicación.

Posteriormente estos datos de la red son los que recoge WiNPS para procesarlos y así, dependiendo de la función que se este solicitando, mostrará al usuario, mediante una GUI construida con Kylix, la información que el pidió.

Una ampliación de la parte de WiNPS en el flujo de información, es la que se muestra en la figura 4.22.

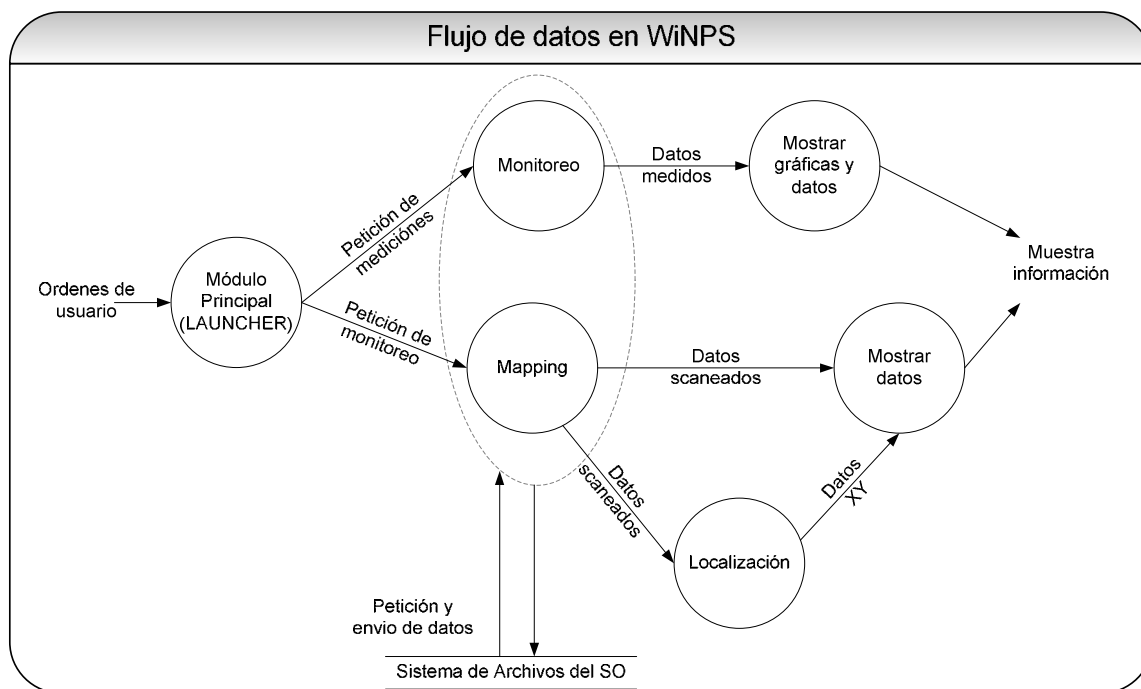


Figura 4.22 Diagrama de Flujo de Datos de WiNPS.

A nivel de la aplicación vemos que existe un módulo principal al que se le ha denominado como *launcher* o lanzador, a partir del cual, y en base la elección de la opción del usuario, se ejecutan los otros módulos.

Se podría hacer una expansión de cada uno de esos módulos, sin embargo, la descripción de éstos la dejaremos para los siguientes apartados, en los que, mediante el uso de diagramas de flujo, se podrá apreciar mejor su funcionamiento.

Así pues, como se puede deducir de las dos figuras anteriores, resulta obvio decir que WiNPs no es programa aislado, sino que está fuertemente vinculado con el sistema de archivos del OS, que es mediante el cual se obtiene la información del entorno.

Es por ello que incluimos la figura 4.23, en la cual presentamos el diagrama de flujo de datos de nivel dos de todo WiNPS, en este diagrama se incluyen los módulos con los que interactúa en el OS, es por ello que en la figura se puede ver con más detalle el flujo de información con el WiNPS trabaja.

De esta ilustración vale la pena mencionar que a nivel de OS, las variables físicas son interpretadas por las herramientas que habíamos mencionado (las *wireless tools* y las *NDIS tools*), la información se almacena en un apartado del sistema de archivos de linux, el cual específicamente está en la ruta */proc/net/*.



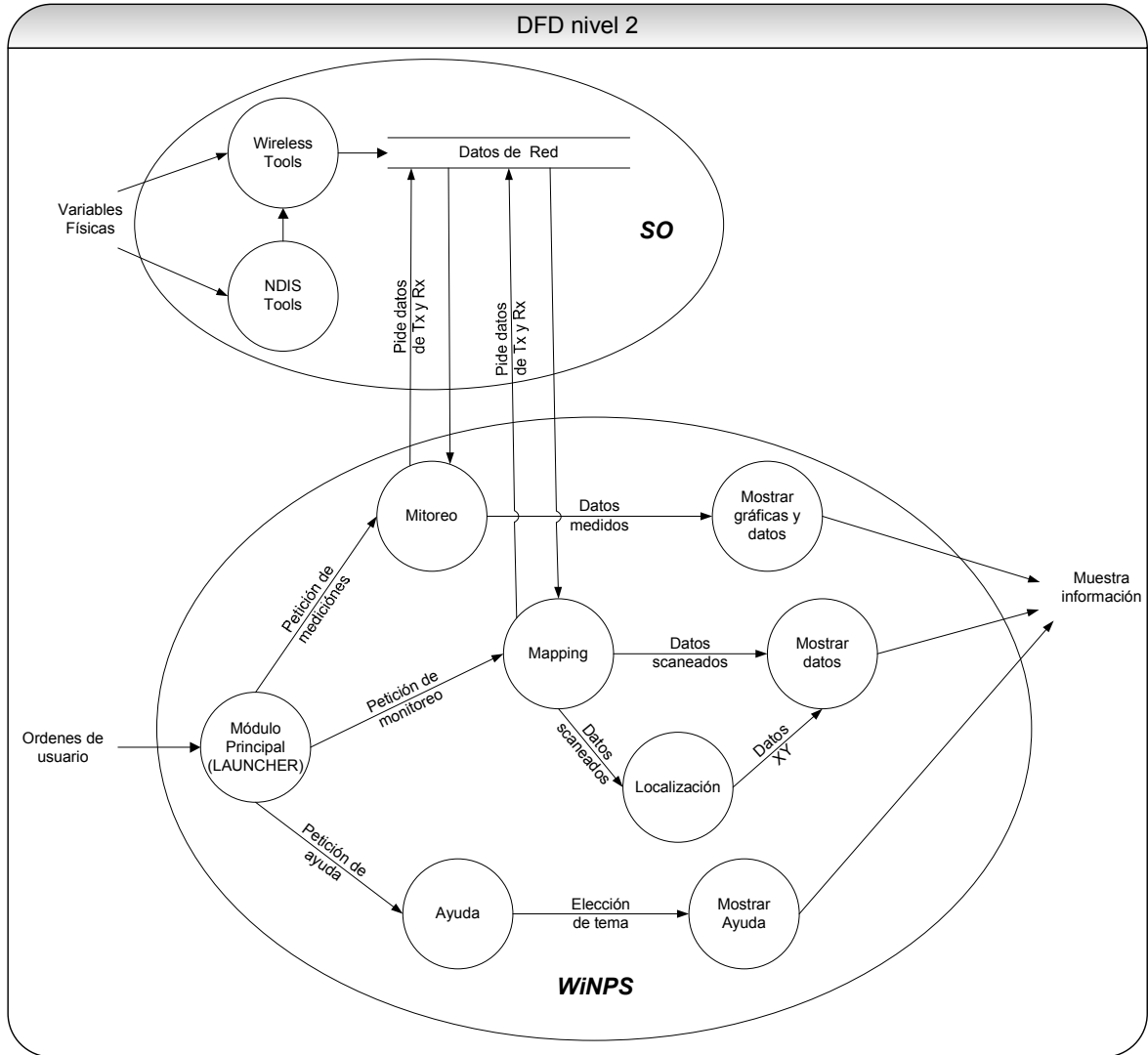


Figura 4.23 Diagrama de Flujo de Datos nivel 2 de WinPS.

Para capturar los datos de la red, se implementó la función *BuscarString*, esta es la función que utilizan los módulos de monitoreo y mediciones, pues forma el vínculo entre el sistema de archivos de Linux y WinPS. A partir de esta función se realiza la búsqueda en el sistema de archivos a modo de capturar la información de los datos de las variables físicas que son necesarias en función de la petición hecha por el usuario.

#### 4.4.2 Diseño de interfaz.

Uno de los aspectos sobresaliente de los entornos de programación RAD, es la facilidad con la que se puede crear la interfaz antes de implementar toda la programación que la respalda, además permite diseñar todo el sistema de una manera muy práctica y visual.

##### 4.4.2.1 Launcher.

Siguiendo la línea del flujo de datos mostrado en la figura 4.23, tenemos encontramos la primer interfaz, esta es la correspondiente al *launcher*, la figura 4.24, nos muestra la interfaz de este módulo.



Figura 4.24 Interfaz del Launcher.

Básicamente es una ventana en la que se encuentran una serie de menús los cuales van ligados a cada una de las aplicaciones que conforman WinNPS, a continuación se menciona una breve descripción de la asociación de cada menú indicado en la figura 4.25.

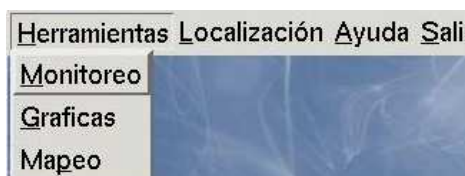


Figura 4.25 Menú que conforma al Launcher.

- ✦ En el menú *Herramientas* encontramos tres aplicaciones:
  - ✦ Módulo *Monitoreo*. El usuario puede realizar un escaneo progresivo de la red.
  - ✦ Módulo *Gráficas*. El usuario visualizará el desenvolvimiento de algunas variables físicas a lo largo del tiempo.
  - ✦ Módulo *Mapeo*. Aquí el usuario puede capturar las intensidades de señales en un punto mediante el algoritmo desarrollado para ese fin.
- ✦ El siguiente menú es *Localización*, que básicamente no despliega ningún submenú, solo manda llamar al motor de localización de WinNPS.

- ✦ El comando de *Ayuda*, como su nombre lo indica, menciona los aspectos más relevantes del funcionamiento de WiNPs, esto a modo de servir como referencia para el usuario.
- ✦ Y finalmente el comando *Terminar*, que finaliza WiNPS.

#### 4.4.2.2 Módulo Monitoreo.

El diseño para el módulo de *Monitoreo* es el que encontramos en la figura 4.26, en esta figura se puede apreciar que el módulo básicamente recoge la información del sistema de archivos e informa cada uno de los campos que se especifican en la figura 4.26, esta herramienta es muy similar a *KWaveControl* con la diferencia que el usuario realiza el muestreo de la información en un tiempo que el mismo especifica, asimismo, puede detener dicho muestreo en el momento que se desee.

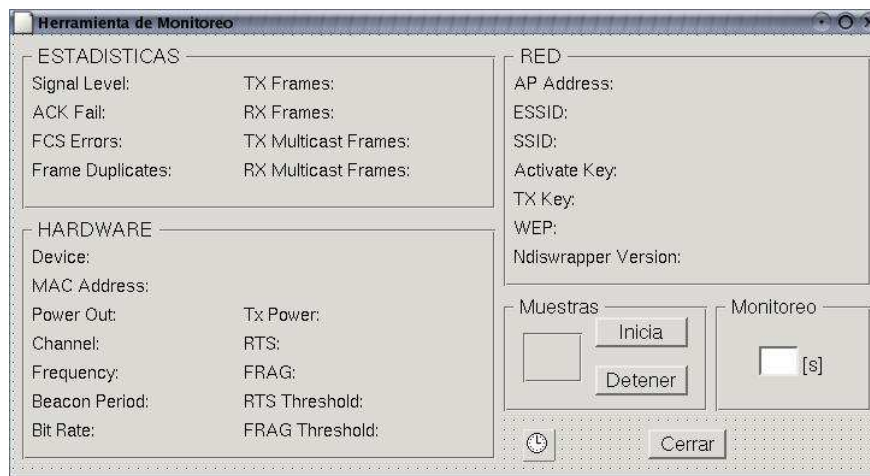


Figura 4.26 Interfaz del módulo Monitoreo.

Esta interfaz está compuesta por:

- ✦ *Etiquetas*. Están conformadas por 54, de las cuales 27 son informativas para conocer el concepto que se está capturando, las otras 27 es en donde el sistema informa los datos de la red obtenidos por la interfaz.
- ✦ *Marcos*. Esta compuesto por cinco elementos, cada marco agrupa un conjunto de variables físicas de acuerdo a la naturaleza de ellas.
- ✦ *Caja de Texto*. Se tendrá una sola caja de texto en la que el usuario podrá ingresar un tiempo de muestreo numérico, en caso de no ser válido, se mostrará un mensaje de error.
- ✦ *Botones*. Existen tres botones, el botón *Inicia* deberá comenzar a muestrear la información captada por la interfaz de hardware, para ello el usuario debió haber informado un tiempo de muestreo en la caja de texto, de no ser así, se desplegara un mensaje de error; botón *Detener/Continua*, este únicamente detiene el proceso de muestreo de información, y el botón *Cerrar*, es el encargado de terminar la aplicación y devolver el control al *Launcher*.

- ✦ **Timer.** Esta es una herramienta propia de Kylix, y es un contador auxiliar en el que se pueden ligar eventos.

#### 4.4.2.3 Módulo de Graficación.

En la figura 4.27, encontramos la imagen de la interfaz correspondiente al módulo de graficación. Básicamente se puede decir que esta es una extensión del módulo de monitoreo, con la diferencia de que aquí el tiempo de muestreo es fijo y las variables son representadas en una gráfica XY.

Se podría pensar que este módulo es muy similar al *KWiFiManger* de KDE, la diferencia es que aquí se puede elegir no solo la variable de intensidad de señal, sino que también se pueden visualizar dos variables de transmisión y recepción de paquetes de información.

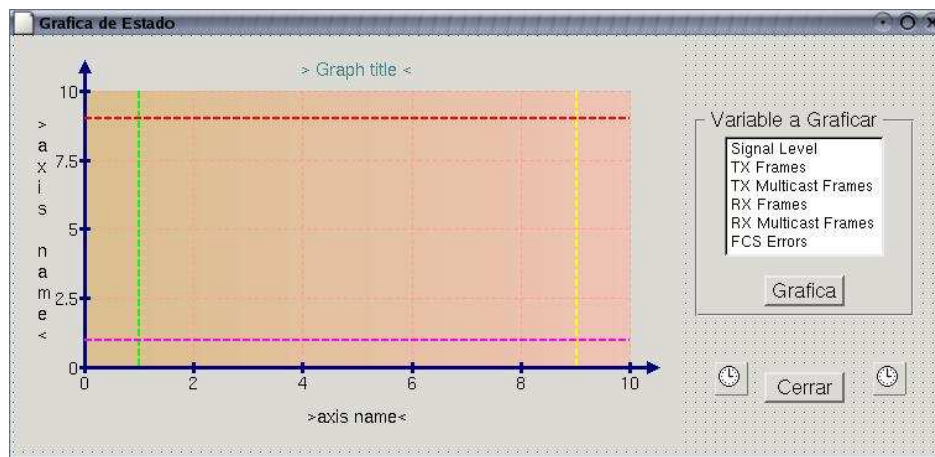


Figura 4.27 Interfaz del módulo de graficación.

Esta interfaz está compuesta por:

- ✦ **Área de Graficación.** Esta es la zona en la que se graficarán los elementos que el usuario elija para la visualización, en el área se informará el tipo de variable, y el número de muestra que se va visualizando.
- ✦ **Marcos.** La interfaz tiene un único marco, en este se agrupan las variables a graficar elementos, así como el botón de comando que ejecuta la acción.
- ✦ **Listbox.** Se tendrá un solo elemento, aquí estarán contenidas las diferentes variables que pueden ser graficadas, cuando se cambie de una a otra variable, el proceso de graficación se detendrá.
- ✦ **Botones.** Existen dos botones, el botón *Grafica* deberá comenzar a muestrear la información captada por la interfaz de hardware en un tiempo determinado, acto seguido deberá comenzar a plasmar la información en la zona de graficación. y el botón *Cerrar*, es el encargado de terminar la aplicación y devolver el control al *Launcher*.
- ✦ **Timer.** Como bien se mencionaba en la interfaz anterior, este *Timer* servirá como un contador auxiliar para el muestreo de información.

#### 4.4.2.4 Módulo de Mapeo.

La interfaz del módulo de mapeo es la que encontramos en la figura 4.28, este módulo, al igual que los otros que hemos ido mostrando, realiza un proceso de muestreo, en que se toman 100 muestras en un intervalo de 20 segundos, posteriormente se realizará el proceso de homologación, para capturar en un archivo de texto la medición promedio en un punto determinado físicamente por el usuario.

A lo largo del tiempo en el que dura el proceso de muestreo, se verá reflejada en la interfaz, y de manera simultánea, la intensidad de señal de los tres APs que conforman la red así como su dirección MAC que le corresponde.

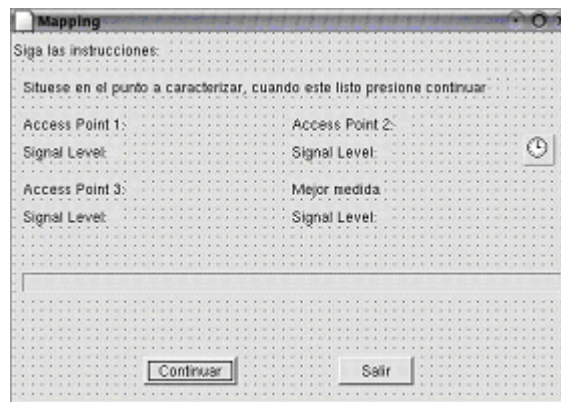


Figura 4.28 Interfaz del módulo de Mapeo.

Esta interfaz está compuesta por:

- ✦ *Etiquetas.* Compuesta por 19 etiquetas, de las cuales dos son de carácter de instrucciones, cada una de ellas informará el estatus del proceso, de las 16 restantes, ocho serán informativas del nombre del AP y la intensidad de señal se estará informando en las ocho restantes.
- ✦ *ProgressBar.* Este es un elemento propio de Kylix y se tendrá un solo elemento, aquí se visualizara de manera gráfica el estado del proceso que se está ejecutando, esto se realiza en una barra que se irá iluminando de acuerdo al avance en el proceso en ejecución.
- ✦ *Botones.* Existen dos botones, el botón *Iniciar/Continuar* este *item* es con el que el usuario podrá desencadenar todo el proceso de *mapping*, además permitirá controlar el avance en cada etapa, cuando todo el proceso halla terminado, saldrá un mensaje informativo que indicara la que la ejecución fue completada. el otro botón es *Salir*, que es el encargado de terminar la aplicación y devolver el control al *Launcher*.
- ✦ *Timer.* Como bien se mencionaba en la interfaz anterior, este *Timer* servirá como un contador auxiliar para el muestreo de información.



#### 4.4.2.5 Módulo de Localización.

Este módulo es lo que llamamos el motor de búsqueda de WinNPS, la interfaz es la que se muestra en las figuras 4.29 y 4.30.



Figura 4.29 Interfaz del módulo de Localización.

La figura 4.29 corresponde a la interfaz en la que se pregunta donde se encuentra el mapa que corresponde al lugar en donde se va a realizar la localización del móvil. La composición de esa interfaz es:

- ✦ *Etiquetas.* Este es un solo elemento y tiene una finalidad indicativa que induce al usuario a introducir la ruta donde se encuentra el mapa.
- ✦ *Caja de Texto.* Este es un elemento en el que el usuario informara la ruta en la que se encuentra el mapa que corresponde al *workspace* en el que se encuentra inmerso el móvil.
- ✦ *Botones.* Solo existe el botón *Continuar* básicamente este sirve para cargar el mapa en la interfaz de la figura 4.30.



Figura 4.30 Interfaz del workspace del módulo de Localización.

En la figura 4.30 tenemos el *workspace* del motor de localización de WinPS, como se puede ver no está precargado ningún mapa, con lo que se optimiza el rendimiento de la aplicación así como su portabilidad. Básicamente está formado de:

- ✦ *Botones*. Solo existe el botón *Localiza*, básicamente este sirve para cargar el algoritmo diseñado para la búsqueda del móvil en la red.
- ✦ *Shape*. Este es un solo elemento y tiene la finalidad de representar al móvil en cuestión, el ancho del círculo está dado a escala, y presupone el rango de exactitud del algoritmo, esto es, el móvil se encontrara en los límites de esa circunferencia.
- ✦ *Timer*. Como bien se mencionaba en la interfaz anterior, este *Timer* servirá como un contador auxiliar para el muestreo de información.

#### 4.4.2.6 Módulo de Ayuda.

En la figura 4.31 encontramos la interfaz del módulo de ayuda, de manera general en esta interfaz se carga un archivo de ayuda en formato HTML que contiene la información general de WinPS.

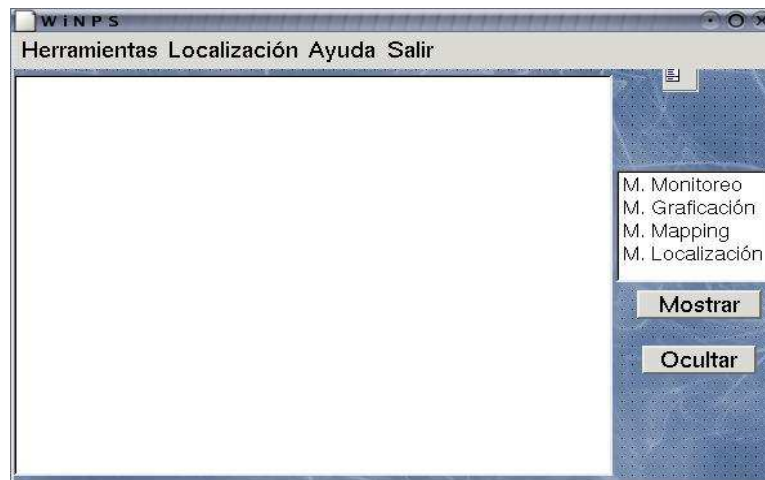


Figura 4.31 Interfaz del módulo de Ayuda.

Como se puede ver, la interfaz es bastante básica, pues solo carga la información de los diversos archivos de ayuda, para ponerlo en el área de la trabajo de la ventana del *launcher*. Otro de los elementos que se tiene es el botón *Ocultar*, que únicamente oculta la ayuda y limpia la ventana del *Launcher*, devolviéndole el control de la aplicación nuevamente.

#### 4.4.3 Diagramas de casos de uso.

Como bien se mencionaba en la figura 4.23, WiNPS se compone de cuatro módulos, estos son *Launcher*, *Monitoreo*, *Mediciones* o *Mapping* y *Ayuda*; aquí salta a la vista que en ningún momento mencionábamos los módulos de localización y el de graficación, esto se debe a la dependencia funcional que existe entre módulos.

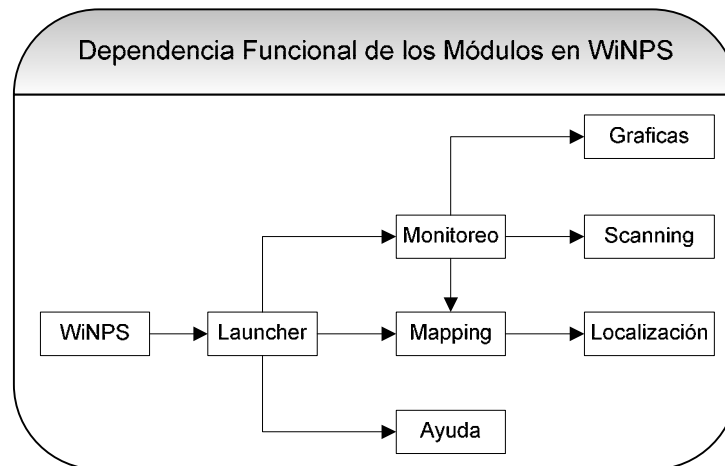


Figura 4.32 Dependencia funcional de los módulos de WiNPS.

Así pues, y de acuerdo a la figura 4.32 podemos ver cuando se desarrollo el módulo de *mapping*, este tomo elementos del módulo de monitoreo, es por ello que su funcionamiento sea similar; lo mismo ocurre con los módulos de gráficas y *scanning*, estos toman la funcionalidad del módulo de monitoreo. Por otro lado, cuando se llevo a cabo el diseño del sistema, se definió al el motor de localización como un caso particular del Módulo de *Mapping*.

Esta dependencia funcional sirvió en gran medida para la reutilización de código, que como bien se sabe, reduce los tiempos de procesador y como consecuencia hace más eficiente a la aplicación.

Asimismo, se encontrará que muchas de las descripciones, módulos y caminos que se encuentran contenidos en los casos de uso, son repetitivos. Así pues, comenzaremos por presentar la figura 4.33, la cual nos muestra el diagrama principal de casos de uso.



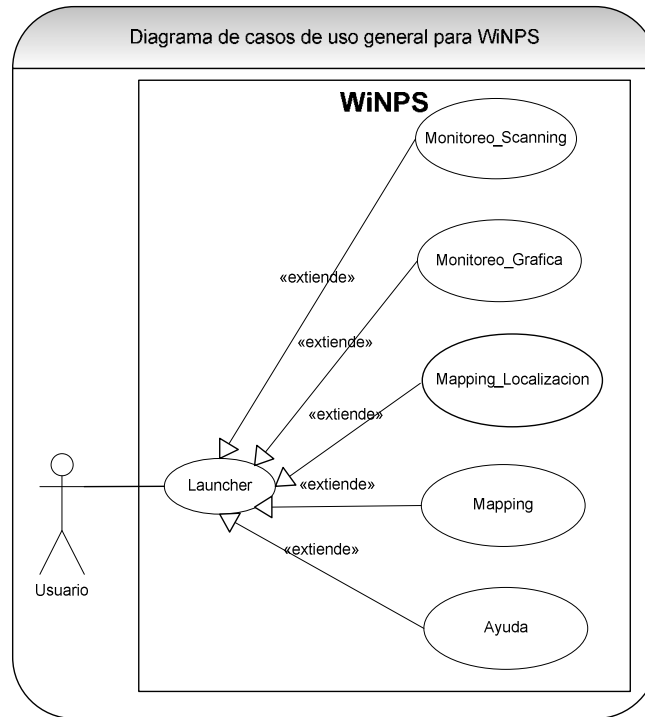


Figura 4.33 Diagrama general de Casos de Uso para WinPS.

En el diagrama 4.33 se ven representados cada uno de los módulos y el actor principal (el usuario). De acuerdo a la definición de caso de uso, cada módulo nos representa un caso de uso diferente pues cada uno de ellos tiene una interacción diferente entre el usuario y el sistema. Con base en ello, iremos haciendo mención a cada caso de uso, tal es el caso de la figura 4.34.

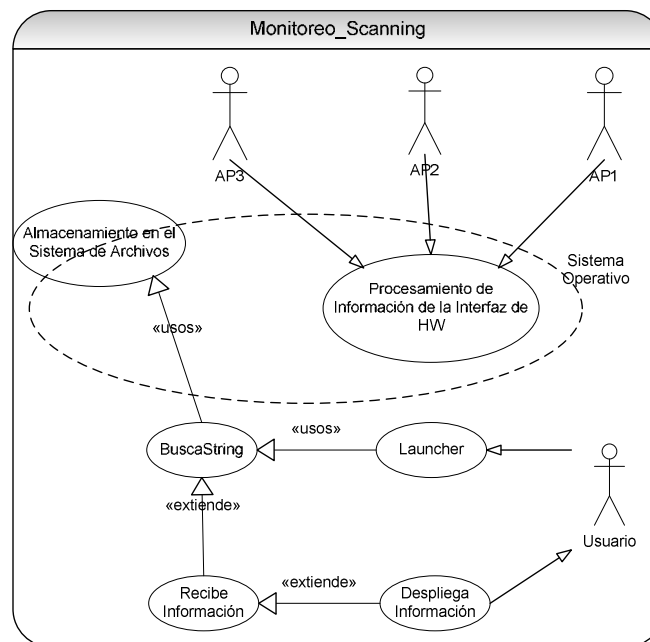


Figura 4.34 Diagrama de Casos de Uso para Monitoreo\_Scanning.

En este escenario tenemos que intervienen cuatro actores, uno de ellos es propiamente el usuario, y los otros tres son los APs. Esto se decidió, porque al igual que el usuario, los APs son necesarios para que se ejecute WiNPS, esto ya se mencionaba en el capítulo anterior cuando se mencionaba que WiNPS no es un sistema desarrollado de manera aislada.

De la figura 4.34 también resulta importante mencionar que en dicho diagrama se incluyeron más casos de uso, estos son:

- ✦ Los casos de uso que corresponden a la parte del OS, esto se debe a que es ahí donde se procesa la información y se almacena en el sistema de archivos de Linux<sup>19</sup>.
- ✦ *BuscarString*. Este es un caso que hace referencia al elemento principal que controla WiNPS y que es la dependencia común entre los módulos de monitoreo, *scanning*, *mapping*, localización y gráficas.
- ✦ *Recepcion de Información*. Este caso también resulta reutilizable para los demás módulos al momento de recibir la información de la función *BuscarString*.
- ✦ *Despliegue de Información*. Para este caso, el uso será muy similar entre módulos, aunque en cada uno llegará a tener algunas variantes, pues el modo en que informa los datos procesados resultará diferente de acuerdo a cada módulo que se este ejecutando.

Luego entonces, el seguimiento para cada uso de uso seria:

Caso de Uso:	<b>BuscarString</b>
Precondiciones:	El usuario debe de informar el parámetro de acuerdo a la opción con la que se va a trabajar.
Postcondiciones:	La función devuelve un valor específico en función a la precondición informada.
Excepciones:	Ninguna
Flujo:	<ol style="list-style-type: none"> <li>1. Convierte el parámetro introducido por el usuario en un tipo legible para la función.</li> <li>2. Se accede al sistema de archivos del OS.</li> <li>3. Se almacena un segmento del archivo en la memoria.</li> <li>4. Se procede a la búsqueda de la cadena en función al parámetro establecido.             <ol style="list-style-type: none"> <li>4.1 Encuentra la información                 <ol style="list-style-type: none"> <li>4.1.1 Se realizan procesos auxiliares tales como conversión de unidades, separación de datos de texto, entre otros.</li> <li>4.1.2 Se regresa el dato procesado y finaliza la aplicación.</li> </ol> </li> <li>4.2 No encuentra la información                 <ol style="list-style-type: none"> <li>4.2.1 Se verifica que no sea fin de archivo                     <ol style="list-style-type: none"> <li>4.2.1.1 Si es fin de archivo                         <ol style="list-style-type: none"> <li>4.2.1.1.1 Devuelve mensaje de no encontrado y termina la aplicación</li> </ol> </li> <li>4.2.1.2 No es fin de archivo                         <ol style="list-style-type: none"> <li>4.2.1.2.1 Almacena otro segmento de memoria y vuelve al punto 4.</li> </ol> </li> </ol> </li> </ol> </li> </ol> </li></ol>

*Tabla 4.3 Caso de uso BuscarString.*

---

<sup>19</sup> Para el alcance de este trabajo, no tocaremos estos casos de uso para su descripción, solo se incluirán de manera ilustrativa.

Caso de Uso:	<b>Recibe Información</b>
Precondiciones:	Se debe de contar con los parámetros provenientes de <i>BuscarString</i>
Postcondiciones:	Se arreglan los parámetros recibidos en el formato numérico que se requiera.
Excepciones:	Cuando <i>BuscarString</i> no informa nada se devuelve un parámetro nulo y no se procesa dato alguno.
Flujo:	
	<ol style="list-style-type: none"> <li>1. Recibe la información de <i>BuscarString</i> en formato String</li> <li>2. Se retoman los parámetros que se introducen en <i>BuscarString</i> para definir que variable es la que se está leyendo (signal level, TX frames, RX Frames, etc.)</li> <li>3. De acuerdo a la variable a la que se este haciendo referencia, la información se transformará en <i>Integer</i> o <i>Float</i>.</li> <li>4. Se almacena en una variable lógica que corresponde la información que se está muestreando.</li> <li>5. Se termina el caso de uso</li> </ol>

*Tabla 4.4 Caso de uso de Buscar Información.*

Caso de Uso:	<b>Despliega Información</b>
Precondiciones:	Se debe de contar con la variable que corresponde a la magnitud física que se está muestreando.
Postcondiciones:	Ninguna
Excepciones:	Ninguna
Flujo:	
	<ol style="list-style-type: none"> <li>1. Toma el valor de la variable lógica que corresponde a la variable física.</li> <li>2. Informa el valor a las etiquetas de la interfaz.</li> <li>3. Kylix interpreta la información en base a sus librerías CLX y despliega en pantalla el formulario así como los demás elementos que lo acompañen, tales como áreas de gráfico, etiquetas, marcos, etc.</li> <li>4. Se termina el caso de uso.</li> </ol>

*Tabla 4.5 Caso de uso de Despliega Información.*

Para este módulo, el proceso indicado en el diagrama 4.34 se repetirá  $n$  veces hasta que el usuario decide detener la aplicación, pues esta herramienta de WiNPS muestrea la información de la red en un tiempo establecido.

De nueva cuenta hay que mencionar que en algunos, por no decir todos, reciclan estos tres casos de uso, con el fin de reutilizar los mismos procedimientos para evitar una gran codificación.

Para el módulo *Monitoreo\_Grafica*, tenemos la representación que se ve en la figura 4.35, en este diagrama nuevamente se está haciendo uso de las tres funciones que mencionábamos en el módulo de *Monitoreo\_Scanning*.

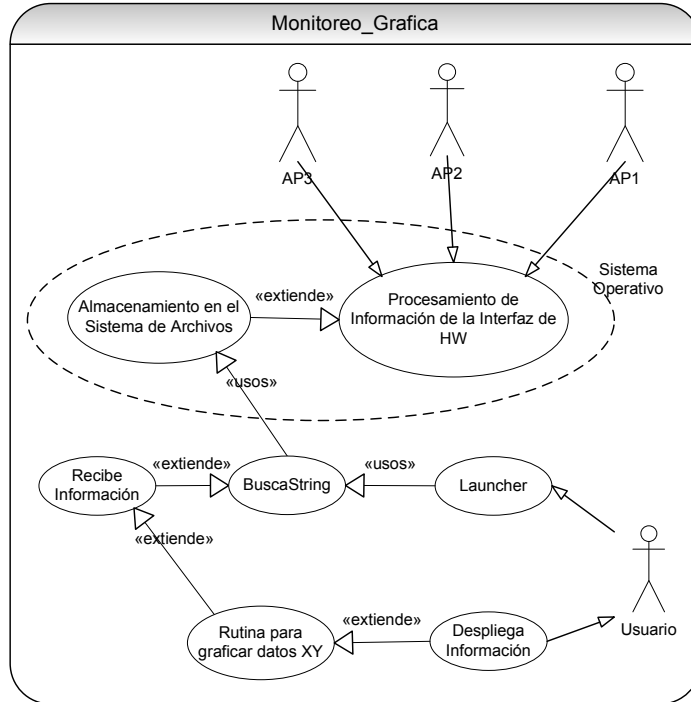


Figura 4.35 Diagrama de Caso de Uso para *Monitoreo\_Grafica*.

Este caso de uso es muy similar a su predecesor, solo se le ha añadido el caso de uso que corresponde a la rutina para graficar datos XY que funciona como se muestra en la Tabla 4.6.

Caso de Uso:	<b>Graficar Datos XY</b>
Precondiciones:	Se debe de contar con la variable que corresponde a la magnitud física que se está muestreando.
Postcondiciones:	Ninguna
Excepciones:	Ninguna
Flujo:	<ol style="list-style-type: none"> <li>1. Toma el valor de la variable lógica del programa que corresponde a la variable física que el usuario eligió.</li> <li>2. Se retoma el parámetro asociado a esa variable para saber a que medición corresponde (Signal level, TX frames, RX Frames, etc.) para informar las cabeceras y títulos.</li> <li>3. La información se transforma en tipo <i>Float</i>; en caso de venir vacío, se informa un cero.</li> <li>4. Kylix interpreta la información y delega el control a la librería <i>Qth2DGraph</i> que interpreta la información para desplegarla en el área de graficación.</li> <li>5. Se termina el caso de uso.</li> </ol>

Tabla 4.6 Caso de uso de *Graficar Datos XY*.

Caeríamos en la redundancia al nuevamente describir los casos de uso *BuscarString*, *Recibe información* y *Despliega información*, es por ello que no se mencionaran estos casos, pues en caso de tener alguna duda, se puede siempre recurrir a la tablas 4.5, 4.6 y 4.7.

El diagrama que sigue ahora es el de *Mapping*, este lo podemos apreciar en la figura 4.36, una vez mas, aquí se ve que la estructura se sigue conservando, de hecho también presenta el mismo vínculo hacia el uso de *BuscarString*.

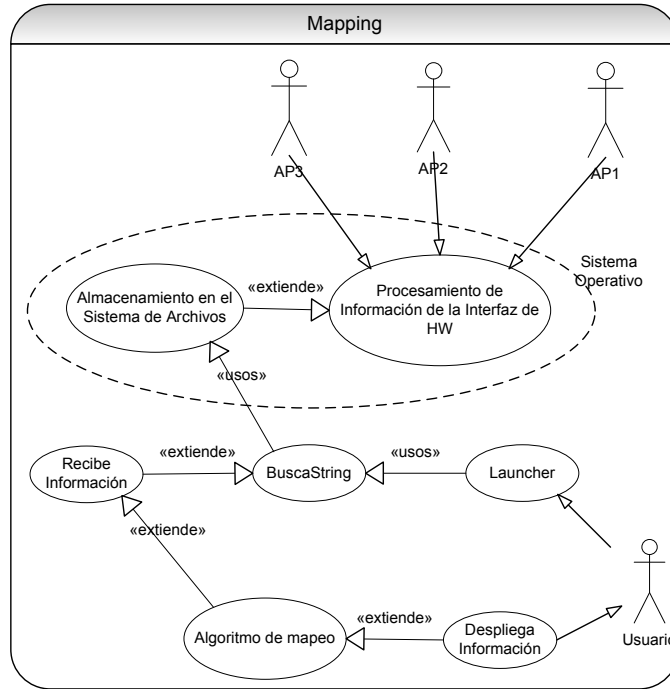


Figura 4.36 Diagrama de Casos de Uso para Mapping.

A modo de dejar completada la descripción de este caso de uso, vemos que el único cambio es el caso de uso del algoritmo de mapeo, es por ello que incluimos en la tabla 4.7 el flujo del nuevo caso de uso del módulo, hay que acotar también que este procedimiento de mapping se realiza simultáneamente para cada AP.

Caso de Uso:	<b>Mapping</b>
Precondiciones:	Se debe de contar con la variable de intensidad de señal para cada AP que conforma el worksapce.
Postcondiciones:	Ninguna
Excepciones:	Ninguna
Flujo:	<ol style="list-style-type: none"> <li>1. Toma el valor de la variable lógica del programa que corresponde a la variable física de intensidad de señal en un instante y para cada AP.</li> <li>2. La información se transforma en tipo <i>Float</i>; en caso de venir vacío, se informa un cero.</li> <li>3. Para cada AP se verifica que el dato está informado, entonces: <ol style="list-style-type: none"> <li>3.1 El dato viene vacío. <ol style="list-style-type: none"> <li>3.1.1 Se informa un cero a un acumulador de la sumatoria.</li> </ol> </li> <li>3.2 No está vacío. <ol style="list-style-type: none"> <li>3.2.1 Se convierte a <i>Float</i> el dato y se le adiciona al acumulador de la sumatoria, también al acumulador de número de muestras se le agrega uno.</li> </ol> </li> </ol> </li> <li>4. Se verifica que el contador del número de lecturas del sistema de archivos sea igual a 20 <ol style="list-style-type: none"> <li>4.1 Si es diferente <ol style="list-style-type: none"> <li>4.1.1 Se incrementa en uno el contador de muestras, se realiza un nuevo muestreo para cada AP y el control se pasa a <i>BuscarString</i> para procesar la nueva muestra y pasar nuevamente al paso 1.</li> </ol> </li> </ol> </li> </ol> <p>Continúa...</p>

Caso de Uso:	<b>Mapping (continuación)</b>
Precondiciones:	Se debe de contar con la variable de intensidad de señal para cada AP que conforma el worksapce.
Postcondiciones:	Ninguna
Excepciones:	Ninguna
Flujo:	
4.2	Es igual
4.2.1	Se verifica que el acumulador de número de muestras sea diferente de cero.
4.2.1.1	Si es nulo
4.2.1.1.1	La lectura promedio será cero
4.2.1.2	No es nulo
4.2.1.2.1	La lectura promedio será la división del acumulador de la sumatoria y el acumulador de número de muestras.
4.2.1.3	Se guarda el valor promedio de la intensidad de señal en cada AP dentro de un archivo de texto.
4.2.1.4	Se muestra un mensaje de fin de proceso.
5.	Se termina el caso de uso.

Tabla 4.7 Caso de uso de Mapping.

En la figura 4.37 presentamos el siguiente diagrama de caso de uso, el cual corresponde al sistema de localización. Se puede ver que nuevamente se retoma la estructura de sus predecesores, con la diferencia que el caso de uso del algoritmo puede utilizar nuevamente *BuscarString*.

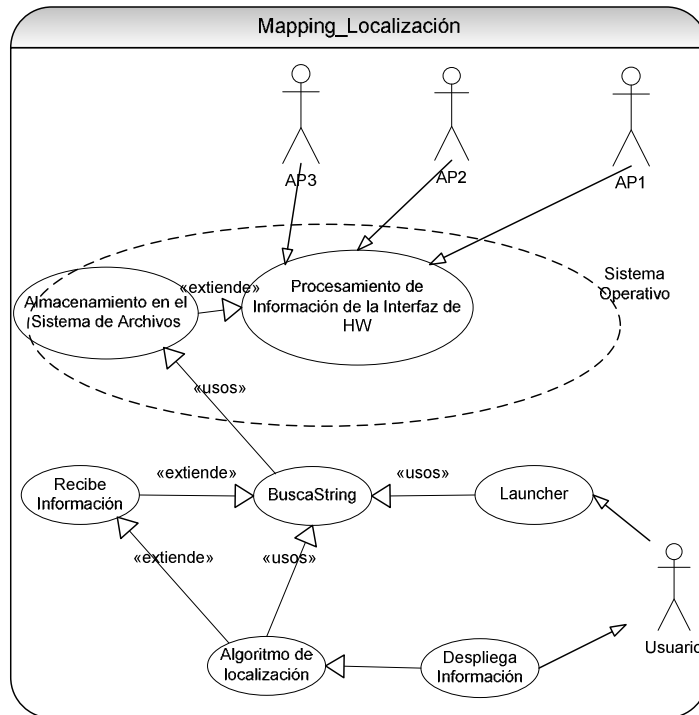


Figura 4.37 Diagrama de Casos de Uso para Mapping\_Localización.

En la aplicación del algoritmo de localización, al igual que en proceso de mapping, el procedimiento se realiza de manera simultánea para cada AP, para obtener las intensidades de señal que capta el móvil en un punto en el espacio.

Así pues, el caso de uso para el algoritmo de localización será el que se muestra en la tabla 4.8:

Caso de Uso:	Localización
Precondiciones:	Se debe de contar con la variable de intensidad de señal para cada AP que conforma el <i>workspace</i> .
Postcondiciones:	Ninguna
Excepciones:	Ninguna
Flujo:	
	<ol style="list-style-type: none"> <li>1. Toma el valor de la variable lógica del programa que corresponde a la variable física de intensidad de señal en un instante y para cada AP.</li> <li>2. La información se transforma en tipo <i>Float</i>; en caso de venir vacío, se informa un cero.</li> <li>3. Para cada AP se verifica que el dato está informado, entonces:             <ol style="list-style-type: none"> <li>3.1 Si el dato viene vacío.                 <ol style="list-style-type: none"> <li>3.1.1 Se informa un cero al acumulador de la sumatoria.</li> </ol> </li> <li>3.2 No está vacío.                 <ol style="list-style-type: none"> <li>3.2.1 Se convierte a <i>Float</i> el dato y se le adiciona al acumulador de la sumatoria, también al acumulador de número de muestras se le agrega uno.</li> </ol> </li> </ol> </li> <li>4. Se verifica que el contador del número de lecturas del sistema de archivos sea igual a 20             <ol style="list-style-type: none"> <li>4.1 Si es diferente                 <p style="padding-left: 40px;">Se incrementa en uno el contador de muestras, se realiza un nuevo muestreo para cada AP y el control se pasa a <i>BuscarString</i> para procesar la nueva muestra y pasar nuevamente al paso 1.</p> <ol style="list-style-type: none"> <li>4.1.1                     <ol style="list-style-type: none"> <li>4.1.1.1 La lectura promedio será cero.</li> </ol> </li> </ol> </li> <li>4.2 Si es igual                 <ol style="list-style-type: none"> <li>4.2.1 Se verifica que el acumulador de número de muestras sea diferente de cero.                     <ol style="list-style-type: none"> <li>4.2.1.1 Si es nulo                             <ol style="list-style-type: none"> <li>4.2.1.1.1 La lectura promedio será cero.</li> </ol> </li> <li>4.2.1.2 Si no es nulo                             <ol style="list-style-type: none"> <li>4.2.1.2.1 La lectura promedio será la división del acumulador de la sumatoria y el acumulador de número de muestras.</li> </ol> </li> </ol> </li> </ol> </li> </ol> <p>Se asocia la intensidad de señal del AP1 a las coordenadas XY correspondientes con lo que descartan datos de AP2 y AP3 que no coincidan. En caso de que no coincidan se elimina el dato y el control se pasa a <i>BuscarString</i> para procesar la nueva muestra y pasar nuevamente al paso 1.</p> <p>Se asocia la intensidad de señal del AP2 a las coordenadas XY que quedaron de la selección del paso 5 con lo que descartan datos de AP3 que no coincidan. En caso de que no coincidan se elimina el dato y el control se pasa a <i>BuscarString</i> para procesar la nueva muestra y pasar nuevamente al paso 1.</p> </li> <li>7. Con la coordenada XY obtenida en 6 se realiza la conversión a unidades en pantalla.</li> <li>8. Se verifica que la coordenada coincida con la limitación espacial del <i>workspace</i>, a modo de garantizar una medida confiable.             <ol style="list-style-type: none"> <li>8.1 Si es diferente.                 <ol style="list-style-type: none"> <li>8.1.1 Se elimina el dato y el control se pasa a <i>BuscarString</i> para procesar la nueva muestra y pasar nuevamente al paso 1.</li> </ol> </li> <li>8.2 Si es igual.                 <ol style="list-style-type: none"> <li>8.2.1 Se dibuja en pantalla el móvil en la posición correspondiente al lugar calculado.</li> </ol> </li> </ol> </li> <li>9. Se termina el caso de uso.</li> </ol>

*Tabla 4.8 Caso de uso de Mapping\_Localización.*

Para finalizar esta sección de casos de uso, en la figura 4.38 se muestra el que corresponde al módulo de ayuda.

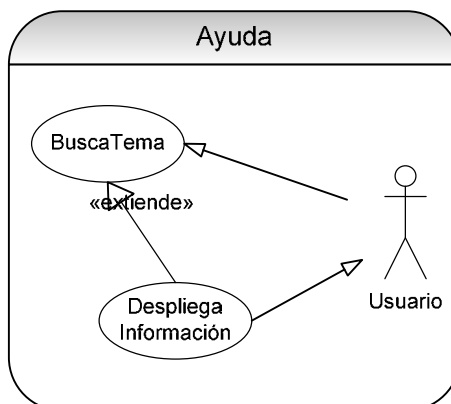


Figura 4.38 Diagrama de Casos de Uso para Ayuda.

A diferencia de los casos de uso anteriores, en esta ocasión, el caso de uso Despliega información es diferente al de la tabla 4.7, además se añade un nuevo caso, este es el de buscar tema; a continuación se da la descripción de cada uno de ellos.

Caso de Uso:	<b>Buscar Tema</b>
Precondiciones:	Ninguna
Postcondiciones:	Ninguna
Excepciones:	Ninguna
Flujo:	<ol style="list-style-type: none"> <li>1. Kylix interpreta carga la información del formulario en base a una librería CLX que permite la vinculación con archivos de texto.</li> <li>2. Se carga el archivo de texto que contiene el índice de ayuda para WinPS.</li> <li>3. El usuario elige un tema y se informa el archivo texto que contiene la información.</li> <li>4. Se termina el caso de uso.</li> </ol>

Tabla 4.9 Caso de uso de Buscar Tema.

Caso de Uso:	<b>Despliega Información</b>
Precondiciones:	Se debe de informar el archivo que el usuario eligió en <i>Buscar Tema</i> .
Postcondiciones:	Ninguna.
Excepciones:	Ninguna.
Flujo:	<ol style="list-style-type: none"> <li>1. Se toma el archivo que el usuario eligió en <i>Buscar Tema</i>.</li> <li>2. Kylix interpreta la información en base a sus librerías CLX y despliega el contenido de los archivos de ayuda en el área designada para tal evento.</li> <li>3. El usuario puede desplazarse de acuerdo al tema desde una lista de temas.</li> <li>4. Se termina el caso de uso.</li> </ol>

Tabla 4.10 Caso de uso de Despliega Información para la Ayuda.



#### 4.4.4 Programación de la aplicación.

Esta es la parte medular del presente trabajo de tesis, y aunque la funcionalidad ya se ha ido explicando con los diagramas de caso de uso, los diagramas de flujo de datos, así como algunos otros esquemas preliminares, la esencia de WiNPS resulta bastante clara:

WiNPS toma una muestra, en un instante  $t$ , de la información captada en el sistema de archivos de Linux, verificará si cumple con las expectativas y en caso de serlo, las procesará y presentará en una GUI de acuerdo a las necesidades del usuario.

En el caso del motor de localización esta acción de muestro se realizará de manera simultánea para cada AP que conforman el workspace, aplicando un algoritmo de eliminación sucesiva, con lo que se garantizará que los datos de intensidad de señal corresponda a una región específica de dicho workspace.

Reiterativamente hemos comentado que el presente trabajo no pretendía que contuviese solo el sistema de localización, sino que se proponía que fuese un sistema completo de monitoreo para una red Wi-Fi y Ad-Hoc; es por ello que el propio sistema necesariamente debe tener diversos módulos, en los que se contengan utilerías para la caracterización de la propia red Wi-Fi.

Ahora bien, de acuerdo a la figura 4.30 que mostraba la dependencia funcional, encontramos que WiNPS se compone de diversos, módulos de los cuales ya mencionábamos los casos de uso para cada uno de ellos. Así pues, consideramos que es el momento de presentar los diagramas de flujo de cada módulo, sin olvidar que estos contienen la información más cercana a lo que sería la programación de WiNPS<sup>20</sup>.

##### 4.4.4.1 Módulo Launcher.

En primer lugar tenemos al módulo *launcher*, que como ya hemos mencionado, sirve únicamente para hacer llamados a los otros módulos, es por ello que el diagrama presentado en la figura 4.39 resulta tener una naturaleza básica.

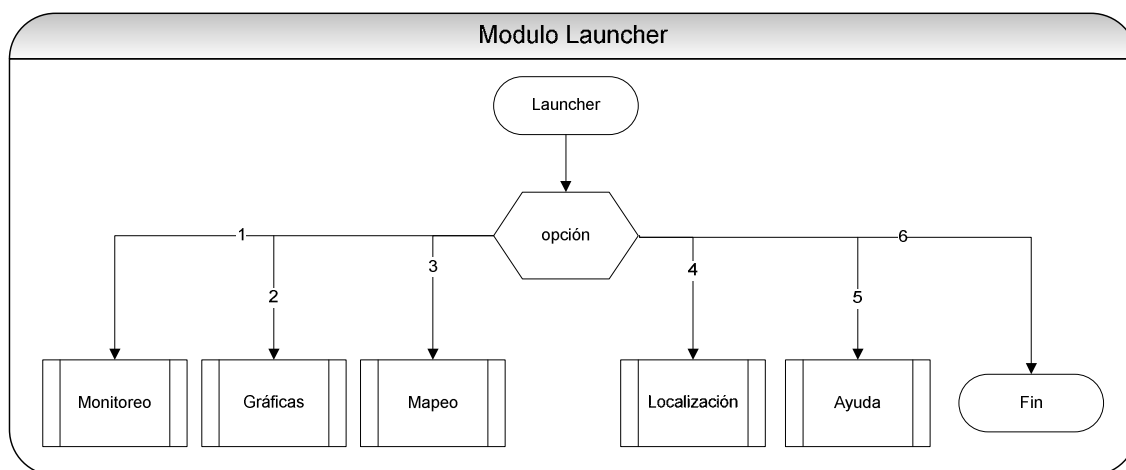


Figura 4.39 Diagrama de Flujo para el módulo Launcher.

<sup>20</sup> En este trabajo de tesis no se incluirá el código fuente utilizado, pues a nuestra consideración, la aplicación puede ser desarrollada a partir de cualquiera de los diagramas.

#### 4.4.4.2 Función BuscarString.

Como se recordará, en la sección de casos de uso hacíamos mención sobre una función que era constante en cada uno de los módulos, y esta hacía referencia a la extracción y procesamiento de la información proveniente del OS. Por esta razón decidimos que, antes de continuar con la explicación de los demás módulos, sería conveniente definir el funcionamiento de dicha función.

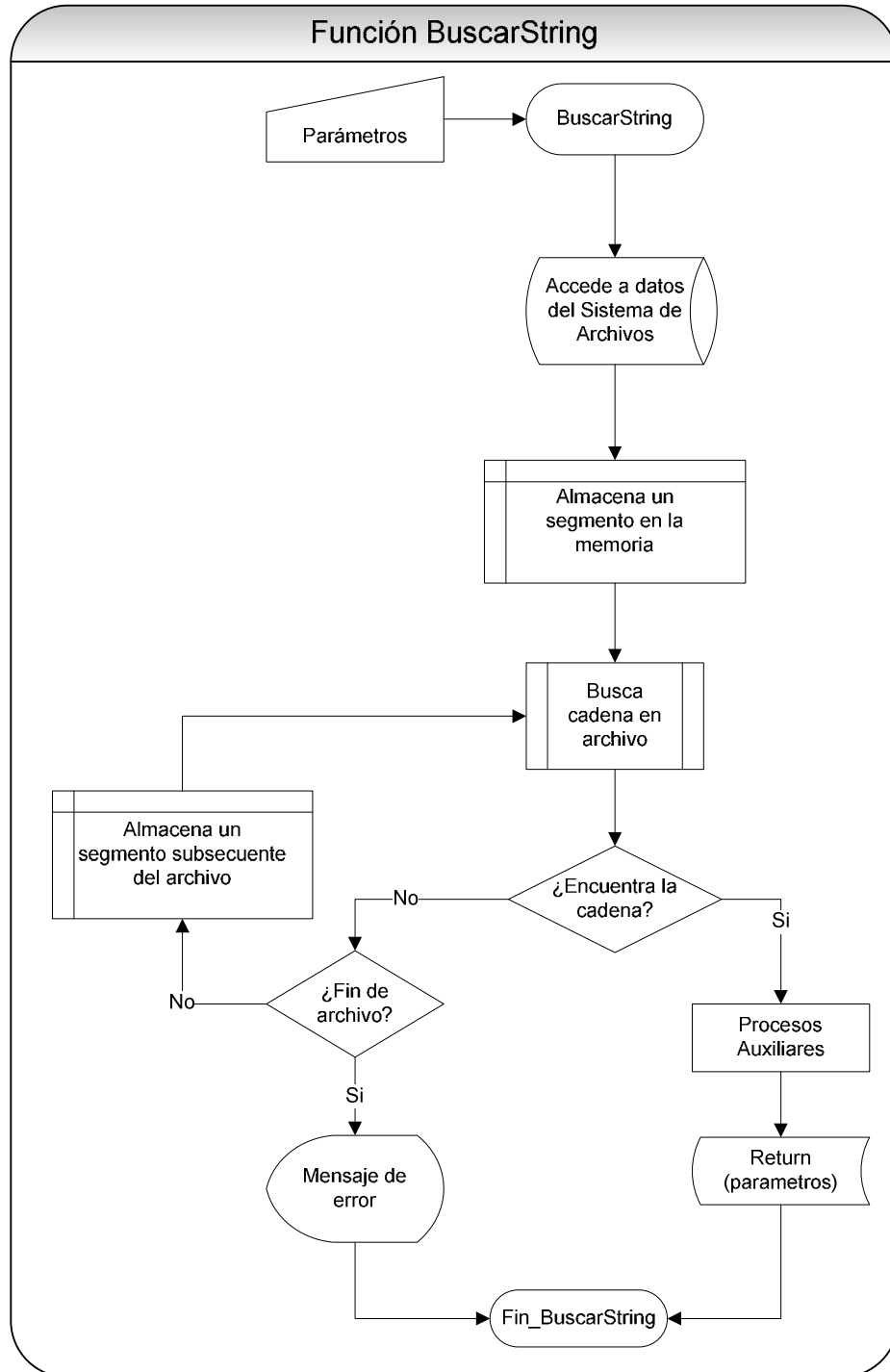


Figura 4.40 Diagrama de Flujo de la función BuscarString.

Así pues, hemos incluido en la figura 4.40 el diagrama de flujo de *BuscarString*; en primer lugar, tenemos que la función *BuscarString* debe recibir un parámetro de control, mediante el cual procesará la información de acuerdo al tipo de módulo que se trate, es decir, la función se da a la tarea de buscar un parámetro específico (intensidad de señal, paquetes recibidos, transmitidos, ruido, etc.) en el sistema de archivos.

Una vez que se ha identificado la variable a capturar, procede a acceder al sistema de archivos que guarda la información de la NDIS, tal y como lo hacen las herramientas *iwnconfig*, o *KWiFiManager*. Como probablemente la información sea bastante grande, entonces almacena en memoria un segmento del archivo y realiza la búsqueda del parámetro especificado.

En caso de no encontrar el parámetro, se tomará otro segmento de archivo y así hasta que lo encuentre, en caso de que no encuentre la información, se devolverá una cadena con un mensaje de error.

En el caso de que encuentre el dato, se realizarán una serie de procesos auxiliares que puede arrojar un valor numérico, un valor de tipo cadena o bien normalizar la intensidad de señal, esto a fin de devolver la información que se requiere en el módulo en cuestión.

Como se ve, básicamente esta función sirve para la búsqueda de parámetros, y mediante ella, WiNPS puede obtener la información del medio y de los tres APs que se encuentran en el *workspace*.

#### 4.4.4.3 Módulo Monitoreo.

Corresponde el momento de describir a la herramienta de monitoreo cuyo diagrama lo encontramos en la figura 4.41, aquí se puede ver que la herramienta realizará una verificación de que el intervalo ingresado por el usuario, sea un dato válido, con ello, se realizará un muestreo cada  $t$  segundos, indicados en la caja de texto, dicho muestreo se llevará a cabo hasta que el usuario detenga la ejecución con el botón correspondiente.

Como consecuencia de ello, en el diagrama se cuenta con la definición de *flg\_stop*, la cual permite para el muestreo que se encuentra en el ciclo. Ahora bien, dentro del ciclo, primero se definen los parámetros que vamos a obtener, para este módulo se requerirán de los elementos que se encuentran en */proc/net*<sup>21</sup>; algunas de las variables que se captan son Nivel de señal, Frames transmitidos, Frames recibidos, Dispositivo, Dirección MAC, Velocidad, Canal, Potencia de salida, Potencia de transmisión, entre otras.

Posteriormente se realiza la búsqueda de los parámetros en el sistema de archivos del OS, los valores contenidos para esos elementos físicos son almacenados en variables, esto sirve para que después sean representado en la GUI de la herramienta de monitoreo.

Para finalizar, se verifica que el usuario no haya detenido la aplicación, en el caso afirmativo, entonces la bandera *flg\_stop* se vuelve verdadera con lo que se detendrá la ejecución de esta aplicación.

---

<sup>21</sup>Se verá en el capítulo siguiente que para esta herramienta no estarán disponibles algunos datos, esto se debe a que el NDIS del OS solo puede captar ciertos elementos de acuerdo al modelo de tarjeta que se utilice.

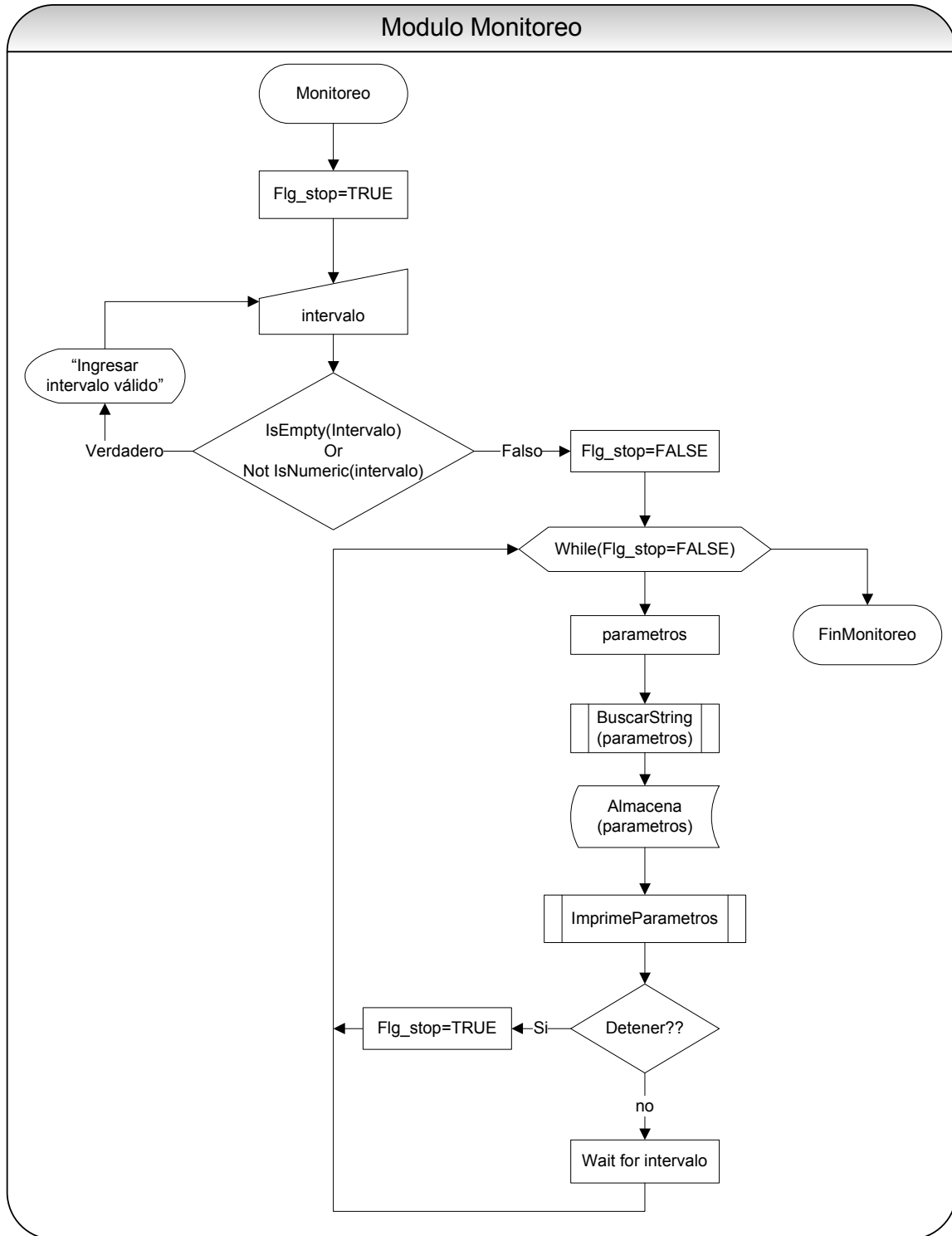


Figura 4.41 Diagrama de Flujo del módulo de monitoreo.

#### 4.4.4.4 Módulo de Graficación.

Ahora presentamos el módulo de graficación, en la figura 4.42 encontramos el diagrama de flujo de esta herramienta, lo que salta a la vista es que este módulo es básicamente una particularidad de la herramienta de monitoreo. La única diferencia subyace en que aquí no se especifica un intervalo de muestreo, además solo se muestrea una variable a la vez.

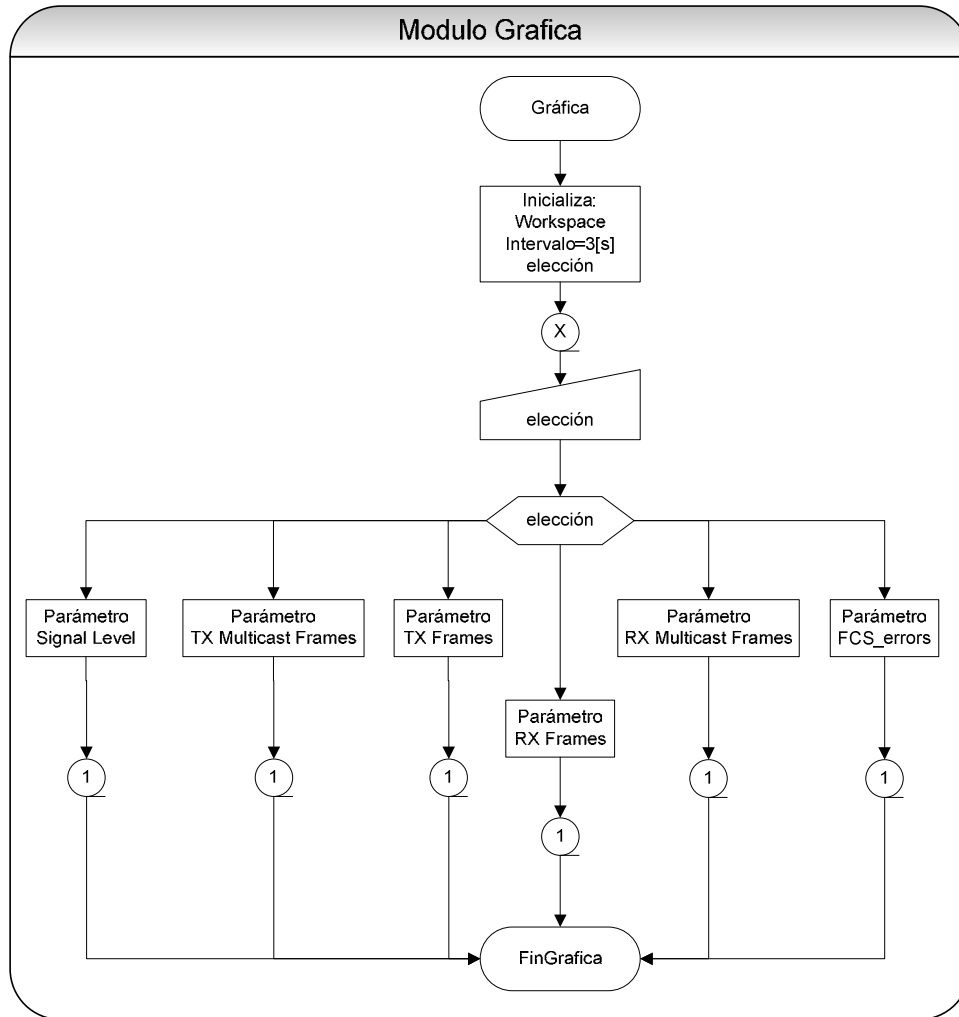


Figura 4.42 Diagrama de Flujo del módulo de graficación.

Ahora bien, revisando este diagrama corroboramos lo anterior, es decir, vemos que, en primer instancia se inicializan las variables del *workspace*, inicializa el intervalo para que se muestre información cada tres segundos; posteriormente, el usuario es el que realiza la selección de la variable a graficar.

Dependiendo de cada elección, se activa un parámetro que corresponde a cada variable física; así pues, esta es la diferencia substancial, pues el procedimiento de búsqueda y obtención de los valores numéricos es igual al del módulo de monitoreo, y por consecuencia, será el mismo procedimiento para cada parámetro. Es por ello que en diagrama se puede hacer referencia hacia el submódulo 1, cuyo diagrama de flujo está representado en la figura 4.43.

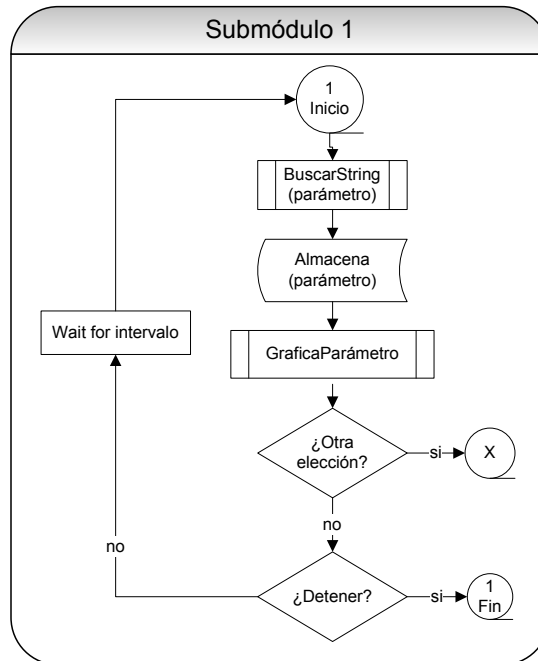


Figura 4.43 Diagrama de Flujo del submódulo 1 para el módulo de graficación.

Una observación hacia ambos diagramas es la referencia que hacen hacia una posición X en el proceso, esto se debe a que si el usuario decide realizar otra elección, se pasa el control hacia el paso anterior a la asignación de la variable; lo mismo ocurre con el caso de que exista una elección que no este contemplada, pues se le da la opción al usuario a elegir otra.

Para finalizar mencionaremos que una de la bondades de esta herramienta es no existen valores que se estén actualizando, esto es, en la GUI se autoajusta la escala de los ejes de acuerdo a la magnitud que se va midiendo cada tres segundos.

#### 4.4.4.5 Módulo de Mapping.

A diferencia de los dos módulos anteriores, esta herramienta de mapeo realiza un proceso de muestreo de la intensidad de señal que se recibe de cada AP, esto lo realiza de manera simultánea y basándose en la información que arroja el comando *iwlist <interfaz> scanning*. El diagrama de la figura 4.44 muestra el flujo de esta herramienta de mapeo, en primer lugar se crean los archivos de datos así como el archivo que contendrá las intensidades de señal.

El paso siguiente es inicializar las variables que nos servirán para parametrizar cada AP, es decir, se definen Dato (es el valor de intensidad de señal obtenido directamente del sistema de archivos del OS), *Sum*, *elem*, *prom*, *i* y *n* son las variables que nos permitirán obtener el promedio de las intensidades de señal, en tanto que MAC nos definirá sobre que AP estamos trabajando. También se definirá el intervalo de estabilización, por el cual el sistema estará esperando a que finalice para continuar con el proceso<sup>22</sup>.

<sup>22</sup> El intervalo de estabilización es un valor experimental que se obtuvo de diversas mediciones, con la consideración de este intervalo se reducen notablemente las variaciones de las mediciones de intensidad de señal.

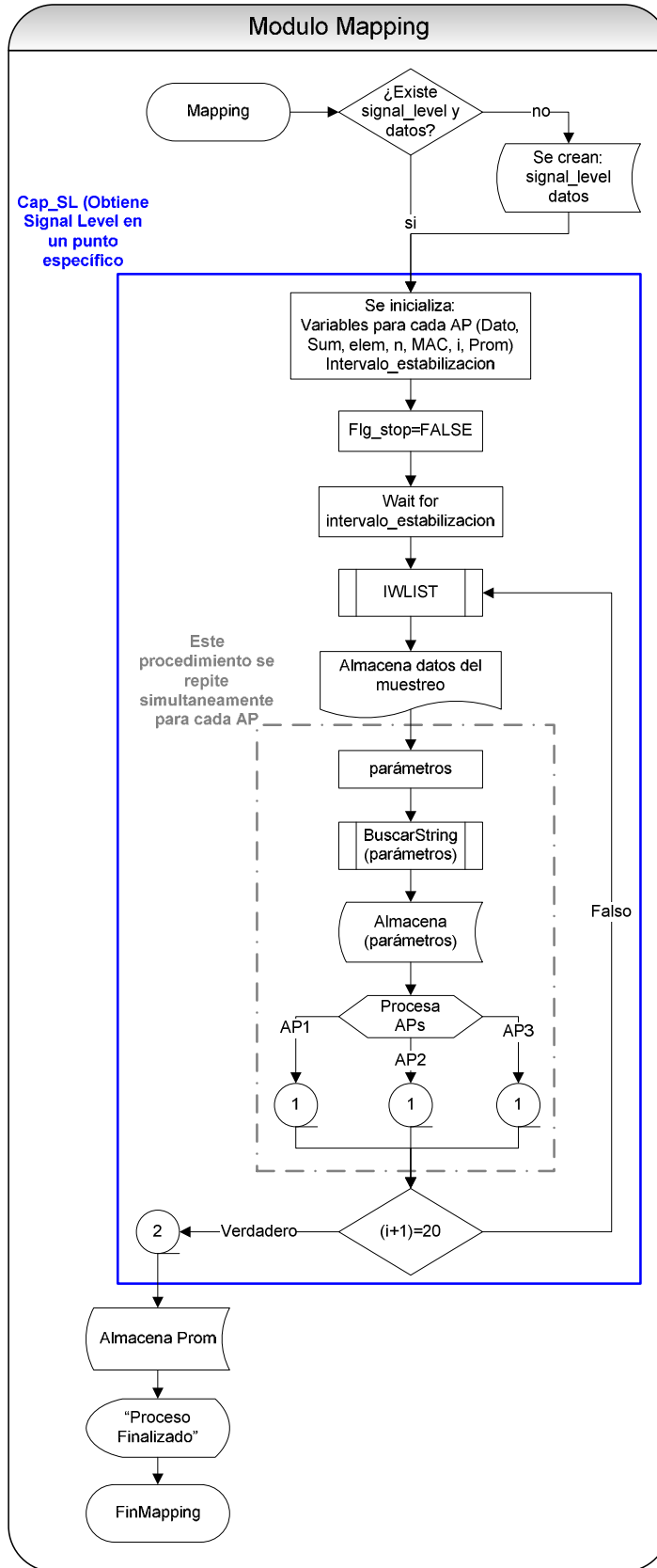


Figura 4.44 Diagrama de Flujo del módulo de mapeo.

Posterior a la espera del intervalo de estabilización, tenemos la ejecución del comando *iwlist* por medio del cual obtendremos la información pertinente, en tiempo real y simultánea de cada AP.

Una vez obtenida la información de *iwlist*, se almacenará en el archivo de datos, posteriormente, y con la ayuda de la función *BuscarString* se obtendrán los valores de los parámetros necesarios para el procesamiento del mapeo.

Ahora bien, con esa información, para cada AP se calculará la sumatoria de las intensidades de señal y el número de mediciones que hayan sido distintas de cero, esta sumatoria se calcula como lo indica el submódulo 1 de la figura 4.45.

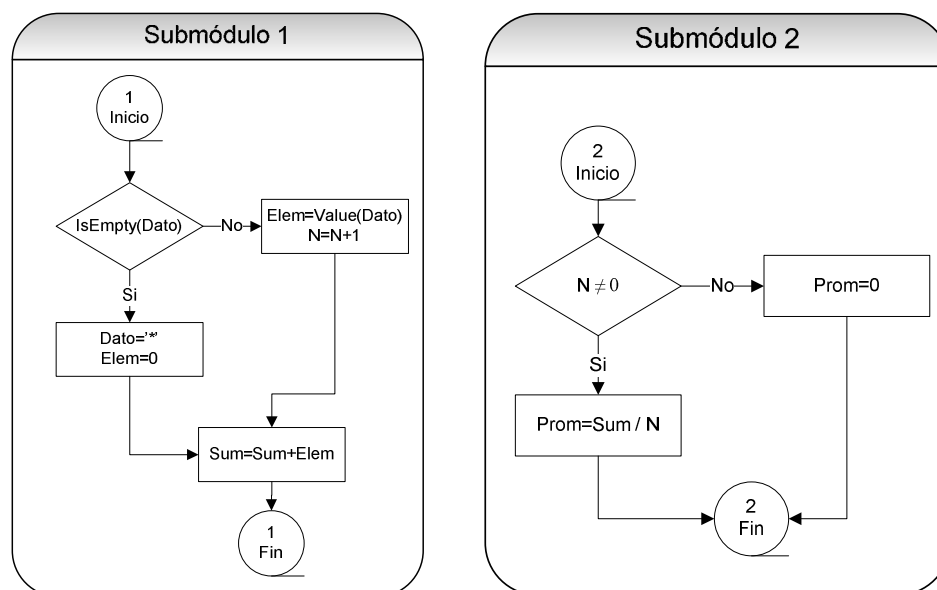


Figura 4.45 Diagramas de Flujo de los submódulos 1 y 2 de la herramienta de mapeo.

Como se puede ver en el diagrama 4.44 el proceso tomara 20 muestras<sup>23</sup> del comando *iwlist*, una vez que ha finalizado el proceso, se procede a obtener el promedio, esto se hace conforma a lo que se describe en el submódulo 2 de la figura 4.45. Cuando se cuenta con el promedio, se almacenan estos valores en su archivo correspondiente, también se almacenarán los valores de dato.

Cuando el proceso ha terminado, saldrá un mensaje en pantalla que el proceso ha finalizado y le pedirá al usuario que se situé en otro punto para tomar una nueva medición.

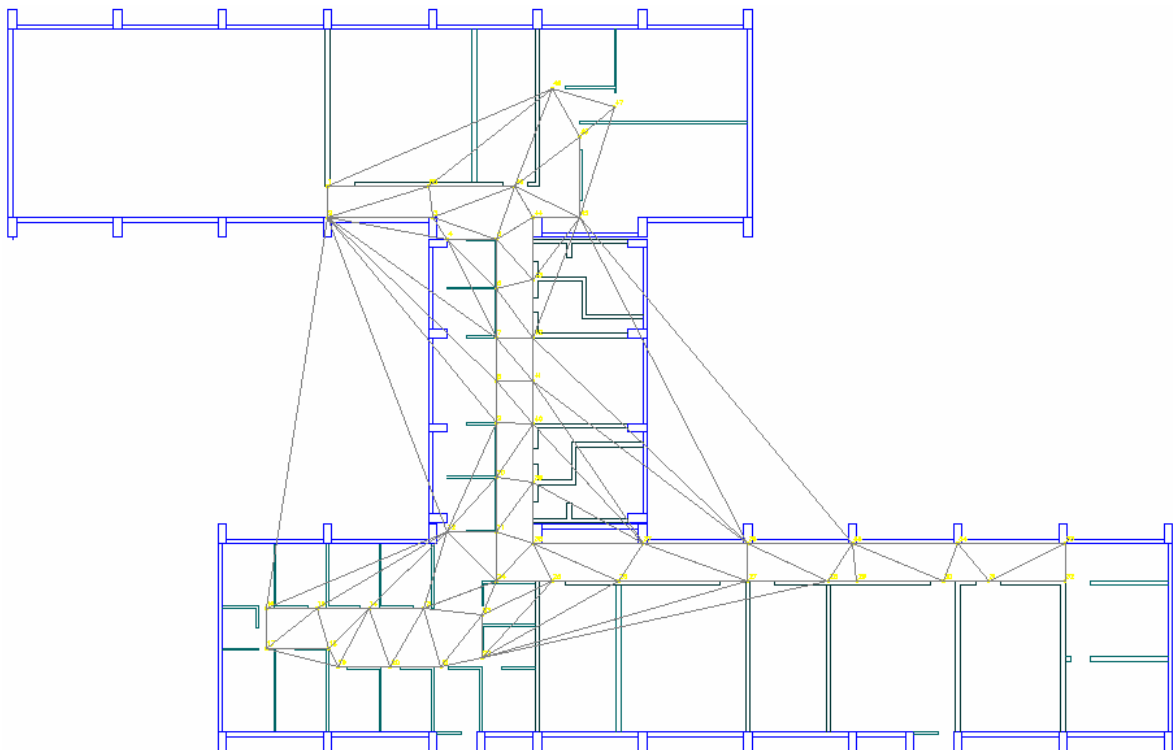
Ahora bien resulta importante mencionar que el proceso de *mapping* consiste en obtener un mapa representativo del *workspace*, esto tiene una gran utilidad para conocer las regiones existentes, así como poder situar a una medición dentro del mismo *workspace*, esto mediante la interpolación y el cruce de los diversos mapas para cada AP.

<sup>23</sup> En la práctica se ha visto que 20 mediciones son suficientes para obtener un valor de intensidad de señal representativa, aunque, esto no quiere decir que más muestras nos acercarán más a un valor, sin embargo, esta acción retrasaría el tiempo de ejecución de la herramienta de mapeo.



Es por ello que primero se deberán definir los puntos sobre los cuales vamos a realizar el proceso de mapeo, en nuestro caso, los puntos que elegimos son los que mostramos en la figura 4.46, ahí los puntos se encuentran distribuidos por todo el pasillo, estos puntos se encuentran sobre puertas, esquinas, y travesaños del *Departamento de Telecomunicaciones* de la DIE.

Mediante un proceso de triangulación<sup>24</sup>, se obtienen una serie de estas figuras, con el fin de tener valores intermedios entre cada punto, pues la separación entre cada uno de los puntos que definimos llega a ser de hasta 3 metros, con esta triangulación podemos tener un mapeo hasta con variaciones de centímetros; es por ello que la figura 4.46 tiene una serie de triángulos superpuestos a la imagen de la planta del edificio.



*Figura 4.46 Triangulación del Departamento de Telecomunicaciones de la DIE.*

Con base en esta triangulación, se pueden obtener distintos mapas para cada AP que nos muestran la distribución de la intensidad de señal para cada AP; a modo de ejemplo, presentamos la figura 4.47 que nos muestra la distribución de la señal en el pasillo, esta distribución corresponde a las intensidades de señal que capta el usuario del AP2<sup>25</sup>.

---

<sup>24</sup> Por cuestiones del alcance de este trabajo de tesis, no describiremos el proceso de triangulación, sin embargo, si se quiere ahondar más sobre este tema, se pueden consultar libros que hablen sobre altimetría y curvas de nivel.

<sup>25</sup> Esta representación se realiza con la herramienta de mapeo de WiNPS y AutoCAD, en el capítulo V se hablará más al respecto de esta distribución.

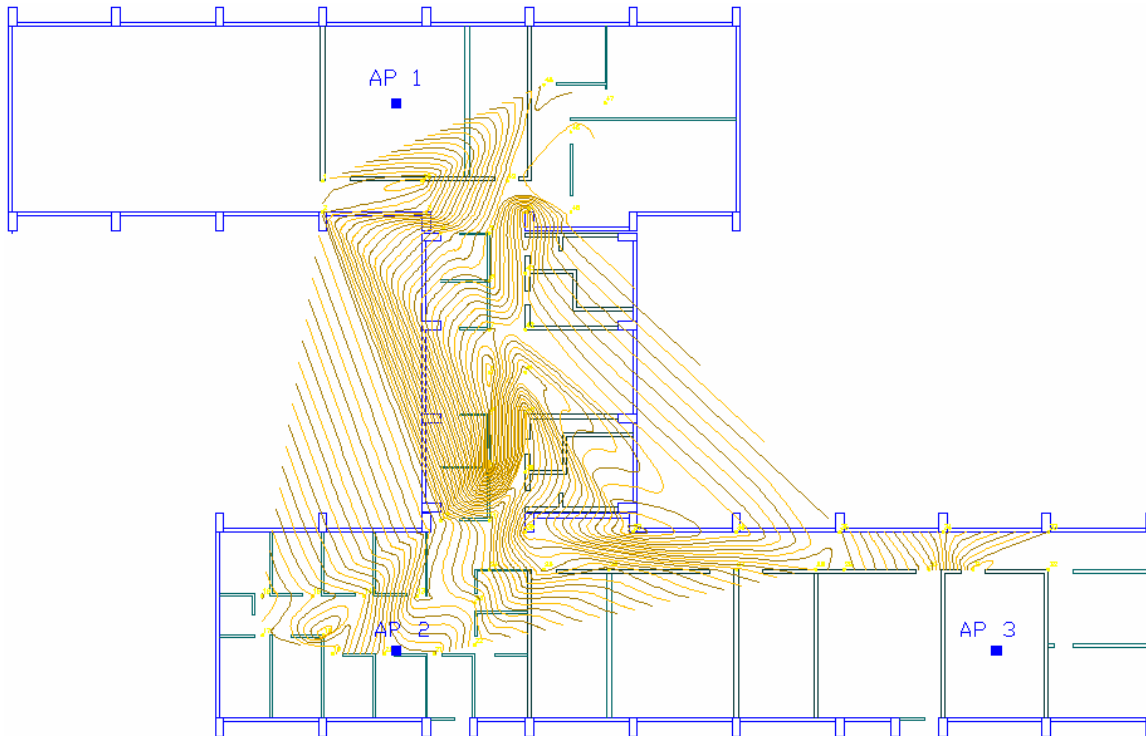


Figura 4.47 Mapping para AP2.

#### 4.4.4.6 Módulo de Localización.

Para finalizar, es momento de describir al motor de localización, este es el módulo principal de este trabajo de tesis, así pues, comenzaremos por mencionar que la funcionalidad de este módulo está compuesta de seis diagramas.

Básicamente su funcionamiento se basa en el módulo de mapeo, esto es, la obtención de la posición se basa en el mismo proceso de muestreo que se realiza en el módulo de *mapping*.

Comenzaremos pues con describir el diagrama de la figura 4.48, posteriormente iremos haciendo referencia a los demás procesos así como su fundamento; en el caso del algoritmo de localización, este se tratará en otro apartado, pues la larga descripción requiere de una explicación muy puntual para comprender su funcionamiento, además este es el eje sobre el que gira el presente trabajo.

Así pues, se definen las variables de las que se van a estar haciendo uso, estas son las que hacen referencia a las posiciones ( $S_x$  y  $S_y$ ), el número de muestras tomadas diferentes a cero ( $N$ ) y los valores mínimos para cada AP, estos valores serán diferentes para cada mapeo, esto es, los valores de  $sl1min=-82$ ,  $sl2min=-86$ ,  $sl3min=-68.55$  son únicamente aplicables para el departamento de telecomunicaciones.

Después de ese paso se procede a la extracción de la información de los archivos de coordenadas (*/root/data/xy*) y del mapa de intensidades de señal (*/root/data/mapa*), esta información se almacena en sus respectivos arreglos (coordenadas y mapa); posteriormente se definirá la constante de término, la cual tendrá será el equivalente al número de puntos mapeados.

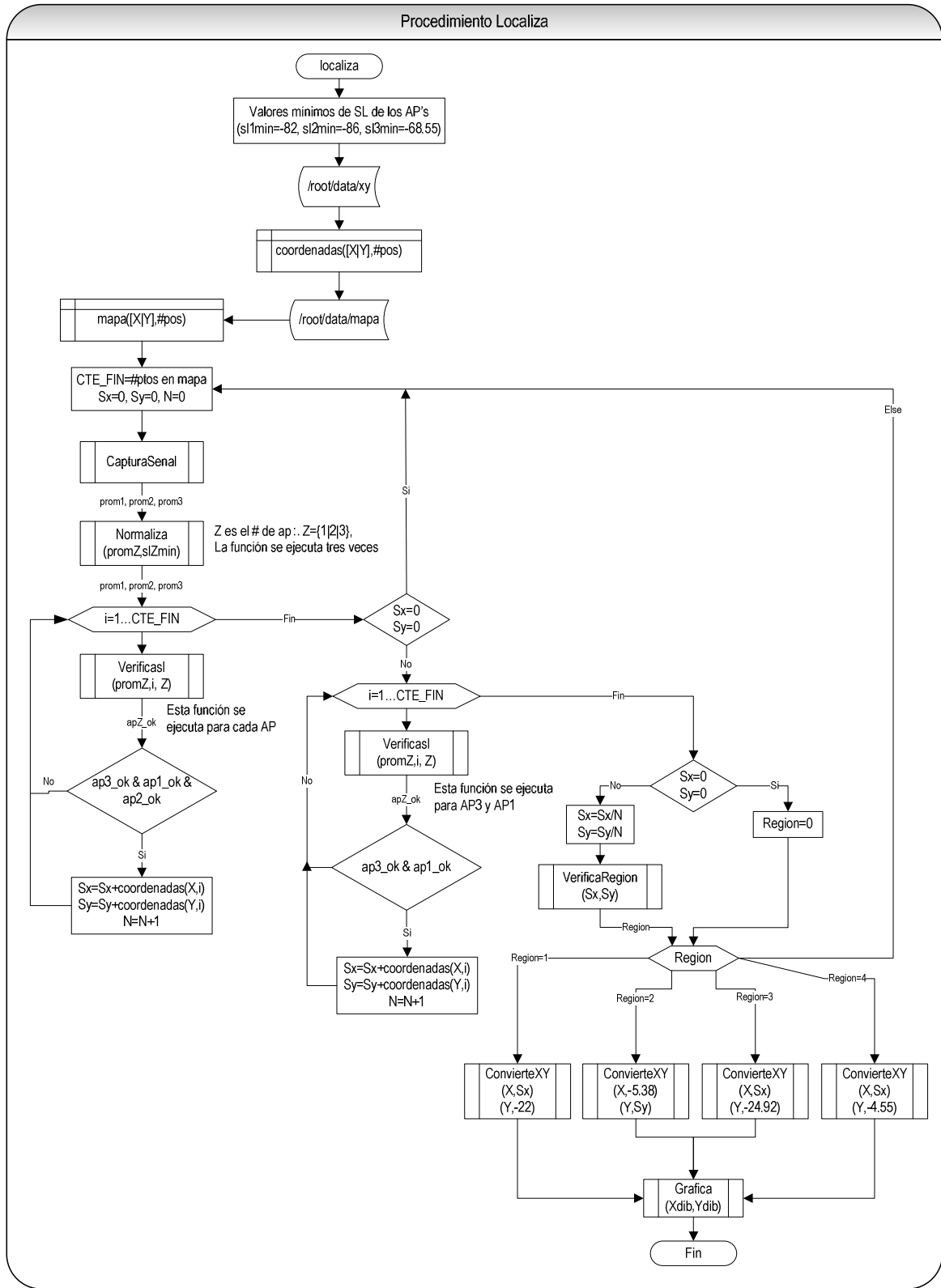


Figura 4.48 Diagrama de Flujo del módulo de localización.

Una vez que se tiene toda el área de variables y constantes, se procede a la captura de la intensidad de señal en el punto, tal y como lo muestra la figura 4.49; esto se realiza de manera similar al proceso de *mapping*, con la diferencia de que aquí nos devuelve un archivo, sino que se devuelven los valores de la intensidad de señal promedio para cada AP.

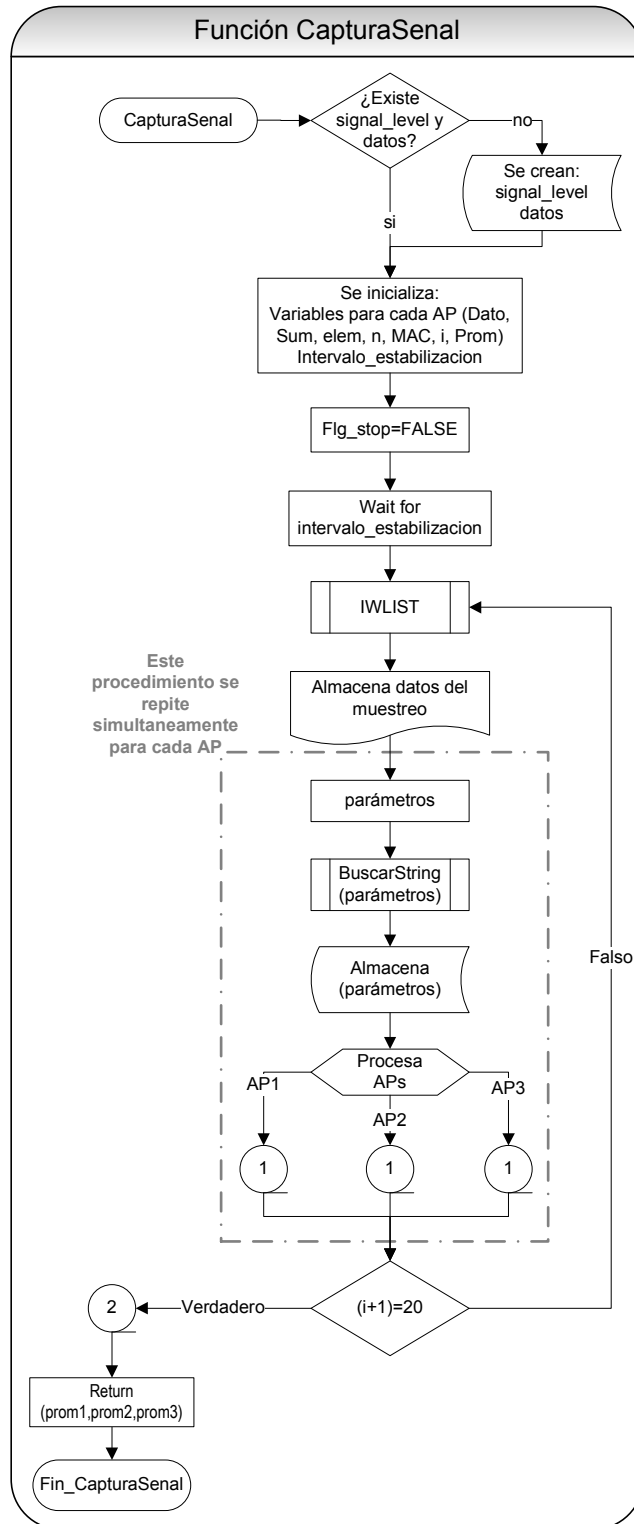


Figura 4.49 Diagrama de Flujo de la función de CapturaSenal.

Cuando se cuentan con los valores representativos de la intensidad de señal para cada AP, se normalizan estos valores a modo de eliminar los valores de intensidad nulos y ponerlos con relación al valor mínimo de intensidad de señal, este procedimiento se realiza de acuerdo a la figura 4.50.

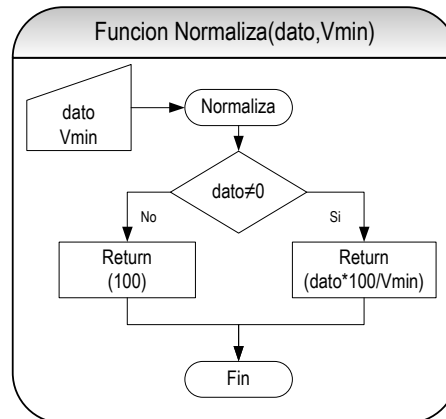


Figura 4.50 Diagrama de Flujo de la función de Normalización.

Con esto, el valor 100 representará una ausencia de señal, se eliminarán los signos, y una mejor calidad de señal presentará un valor muy cercano a cero, claro, sin llegar a serlo. La función de normalización se ejecutará para cada valor de intensidad de señal captado por su respectivo AP. Posteriormente se procede a la aplicación del *algoritmo de localización* diseñado por nosotros, el cual se explicará en el siguiente tema.

#### 4.4.4.7 El algoritmo de localización.

El desarrollo del algoritmo resultó un tanto complicado, pues básicamente la totalidad de este fue desarrollado por nosotros, esto se debe a que el tema de redes inalámbricas es relativamente reciente, y no existen muchos desarrollos documentados sobre sistemas de localización como en el caso de GPS u otro sistema de localización, es por ello que no existe en la actualidad un algoritmo público para la ubicación de un usuario en redes inalámbricas.

Por ese motivo, presentamos nuestro algoritmo de localización basado en el sistema de *mapping* para una red inalámbrica Wi-Fi, asimismo, hay que mencionar que existieron muchas versiones para este algoritmo, sin embargo, la que presentamos a continuación es la que satisfizo nuestras expectativas tanto en el desarrollo del software como en los resultados de la localización.

Ahora bien, ya en los casos de uso ya se vislumbraba la funcionalidad de este algoritmo, sin embargo, a continuación se dará otro enfoque, que quizás sea más descriptivo; para ello debemos de tomar en cuenta el diagrama principal de *módulo de localización* (figura 4.48), a partir de este, se obtiene la figura 4.51, que contiene el diagrama de flujo que corresponde al algoritmo de posicionamiento.

Luego entonces, comenzaremos por decir que para el funcionamiento del algoritmo, se requieren de tres valores de intensidad de señal representativos que el móvil capta de los tres APs con los que cuenta el área de trabajo.

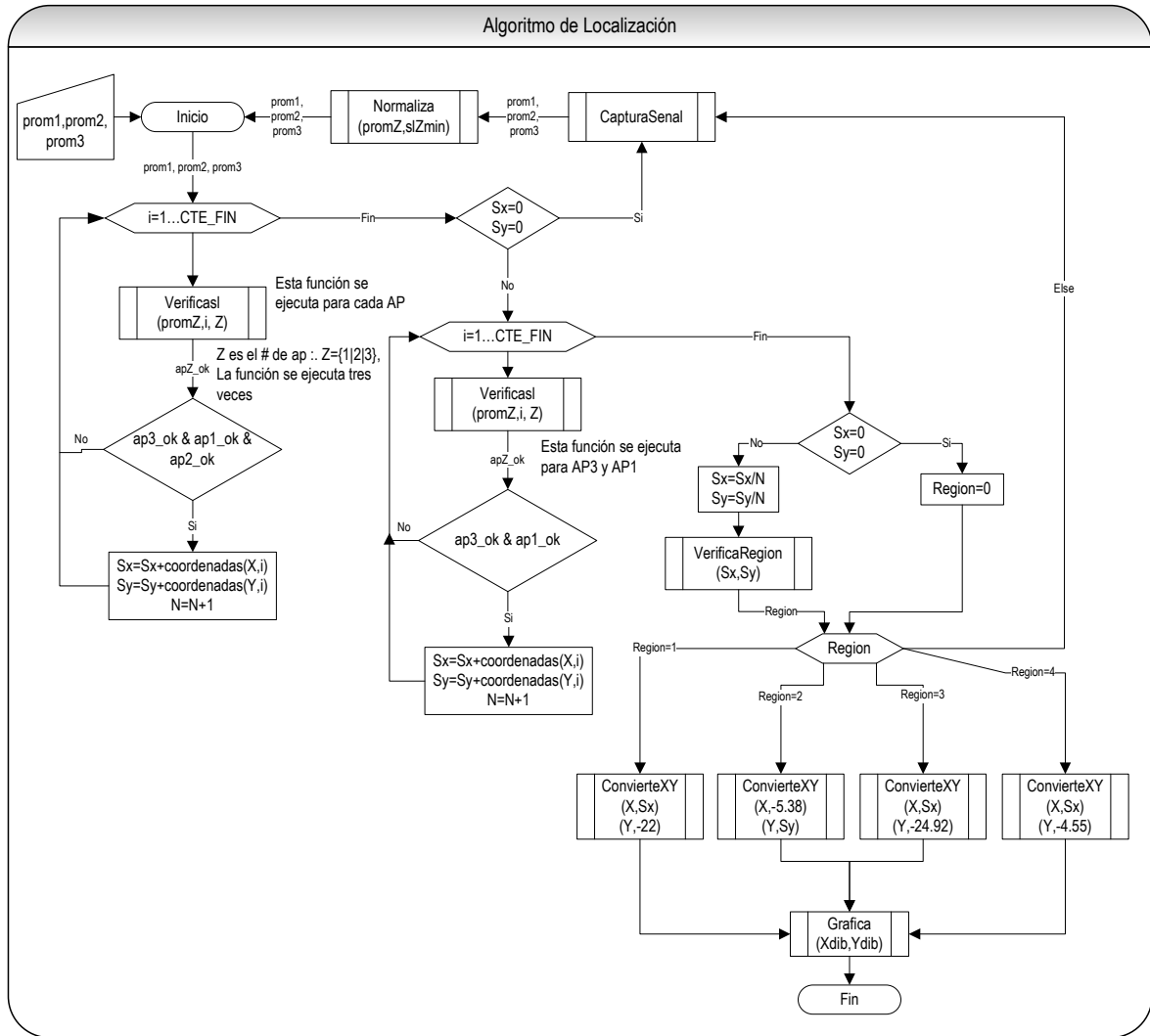


Figura 4.51 Diagrama de Flujo del algoritmo de localización.

1. Se posicionará el cursor en el primer registro del arreglo del mapa de intensidades de señal, este procedimiento se realizara hasta que se hallan terminado de leer todos los registros del arreglo.

1.1. Se verificará el valor de cada intensidad de señal conforme a lo descrito en el diagrama de la figura 4.52.

Básicamente lo que indica el diagrama es que se define un factor de tolerancia de 0.08, con este valor se define un rango compuestos de un  $L_{sup}$  y un  $L_{inf}$  con base en la medición de la intensidad de señal captada por el móvil.

Luego se verifica que el valor leído en el mapa de intensidad de señal se encuentre dentro del rango, de ser así, se devuelve una bandera en verdadero. Esta operación se realizará para la intensidad captada en el AP1, AP2 y AP3.

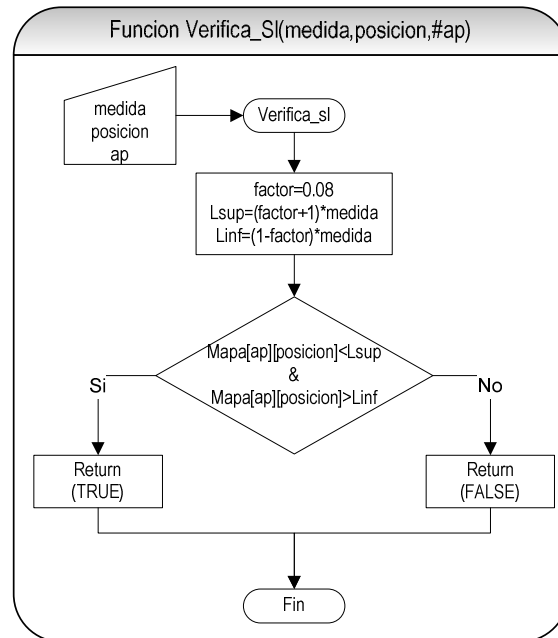


Figura 4.52 Diagrama de Flujo de la función de Verifica\_Sl

- 1.2. Posteriormente se pregunta si las tres intensidades del mapa se encuentran dentro del rango de la intensidad de señal del móvil.
  - 1.2.1. En caso de que el valor leído del mapa se encuentre en la región de tolerancia, se incrementa el valor del acumulador de valores de X y Y ( $S_x$  y  $S_y$ ) con el valor de la coordenada que le corresponde, también se añade 1 al acumulador de elementos ( $N$ ).
  - 1.2.2. En caso de no ser así, se procede a una nueva lectura del registro y pasamos al punto 1.1
2. Una vez que se terminaron de leer todos los registros, se procede a la verificación de que haya habido por lo menos algún punto que haya satisfecho la pertenencia al mapeo.
  - 2.1. En caso de no encontrarlo, se procede a otra lectura de la intensidad de señal captada en el móvil para cada AP.
  - 2.2. De no ser así, nuevamente se posicionará el cursor en el primer registro del arreglo del mapa de intensidades de señal, este procedimiento se realizará hasta que se hallan terminado de leer todos los registros del arreglo.
    - 2.2.1. A diferencia del punto 1.2 en donde se preguntaba que si todas las mediciones del mapeo cayeran dentro del rango de tolerancia, ahora solo se preguntará por las medidas del mapa de intensidades correspondientes a los AP3 y AP1, esto se debe a que estos siempre presentaron datos poco variables y con pocas pérdidas de señal.
      - 2.2.1.1. Si se encuentran en la región de tolerancia se incrementan los acumuladores de X, Y y N con sus respectivos valores.
      - 2.2.1.2. En caso contrario, se procede a una nueva lectura y se pasa al punto 2.2

3. Nuevamente, cuando se han terminado de leer todos los registros, se procede a la verificación de que haya habido por lo menos algún punto que haya satisfecho la pertenencia a la región.

3.1. Si no existió ningún punto que haya satisfecho la condición, entonces se inicializa la variable *región* en 0.

3.2. De lo contrario se procede al cálculo del promedio de las medidas  $S_x$  y  $S_y$ , se realiza de esta manera, pues como sabemos, una de las características geométricas del promedio es que el valor medio de coordenadas siempre estará dentro de la región de puntos, y tenderá a estar aún más cerca de donde exista una mayor concentración de mediciones similares, tal es el caso del ejemplo que se muestra en la figura 4.53.

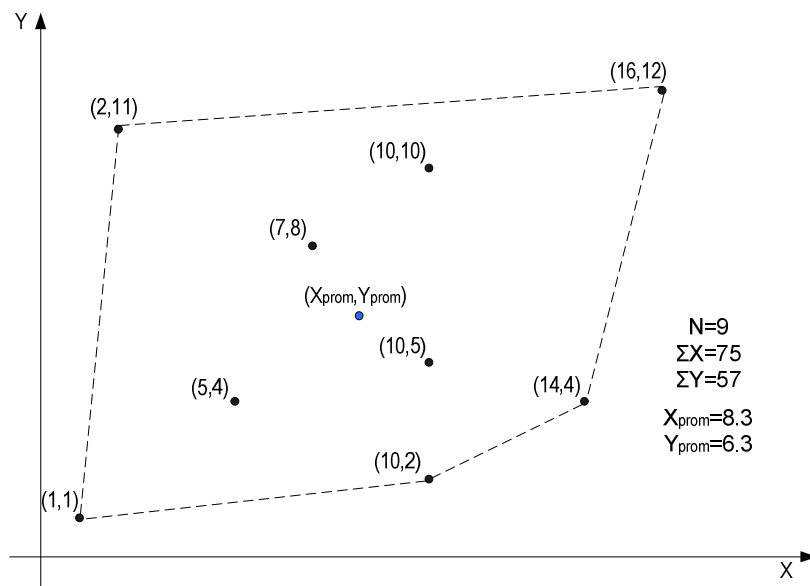


Figura 4.53 Ejemplo de la propiedad geométrica del valor promedio de coordenadas.

4. Como penúltimo paso tenemos la verificación de las coordenadas obtenidas en el promedio, la inclusión de este paso de verificación se debe a que existe la posibilidad de obtener un punto fuera del *workspace*, lo que indicaría que el usuario andaría “volando” fuera del edificio.

Para evitar esta situación, se decidió partir al *workspace* en regiones, para decidir cual era la mejor manera de distribuir las regiones, se concluyó limitar al área de localización a las áreas de tránsito de las personas, es decir, el móvil se ubicará dentro del pasillo del departamento de Telecomunicaciones.

La base de esta decisión subyace en la consecuencia que tiene la limitación del área de trabajo, es decir, se reduce el área de localización y se incrementa la precisión en la estimación del posicionamiento, y el mapeo reduce la distancia entre puntos, lo que coadyuva también a la localización del usuario.

Así pues, las regiones las hemos definido como se muestra en la figura 4.54.



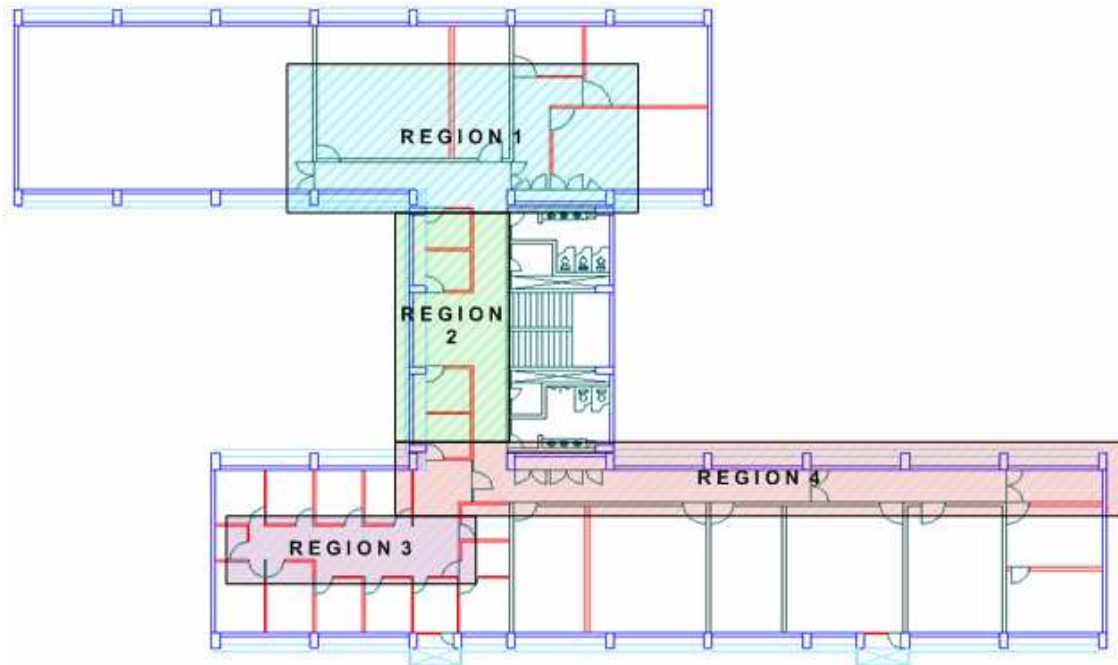


Figura 4.54 Regiones del departamento de telecomunicaciones de la DIE.

Ahora bien, la verificación de las coordenadas nos ayudara a averiguar si el usuario se encuentra en alguna de las regiones establecidas, para llevar a cabo esta acción se implementa la función *Verifica\_region*, cuyo diagrama lo podemos ver en la figura 4.55.

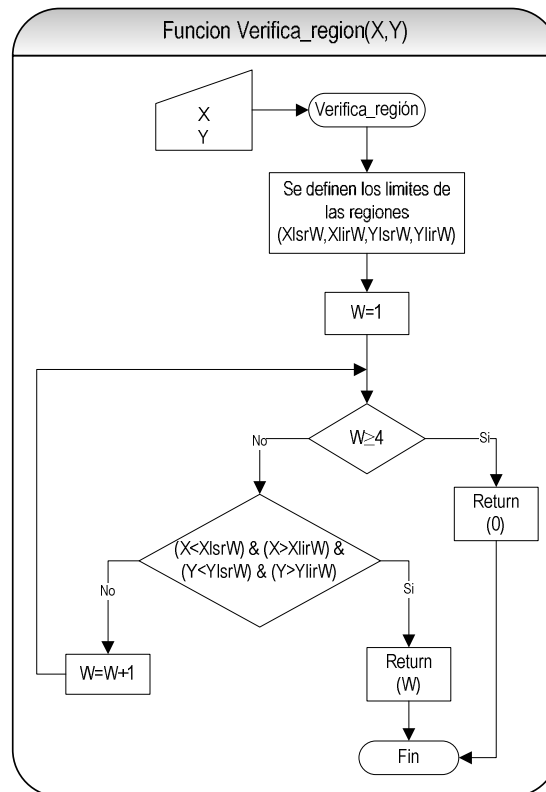


Figura 4.55 Diagrama del módulo *Verifica\_Region*.

Básicamente lo que se realiza en la función es definir los límites de las regiones con base en dos puntos XY contrapuestos, tal y como lo indica la figura 4.56; en dicha figura se aprecia que con solo conocer los valores de X y Y en dos puntos opuestos, uno inferior ( $X_{lirW}, Y_{lirW}$ ) y otro superior ( $X_{lsrW}, Y_{lsrW}$ ), es más que suficiente para determinar si un punto existe dentro de la región.

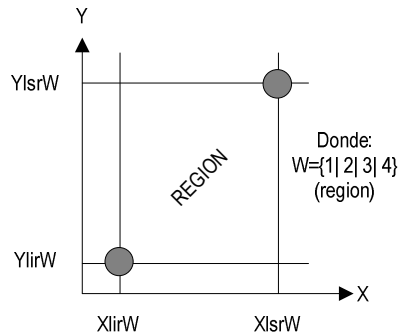


Figura 4.56 Ejemplo de cómo se definen los límites de las regiones.

Así, todo punto  $(X,Y)$  en una región  $W$  se puede definir como el que cumple con la condición  $X > X_{lirW} \text{ AND } X < X_{lsrW} \text{ AND } Y > Y_{lirW} \text{ AND } Y < Y_{lsrW}$ . En el diagrama 4.55 se ve también que si el punto se halla en la región, entonces se devuelve el número de la región que le corresponde, en caso de no ser así se verifica el espacio siguiente, si no se ubica en alguna de las cuatro regiones entonces se asigna un cero a la variable *región*.

- 4.1. Si el valor de la variable *región* es nulo, entonces no existió ningún punto que haya satisfecho la condición de pertenencia a una región, por lo que se vuelven a medir nuevamente las intensidades de señal que capta el usuario de los tres APs, posteriormente se vuelve a iniciar el algoritmo de localización.
- 4.2. En caso contrario, se procede a la conversión de las unidades de X y Y reales a unidades de dibujo, esta conversión se realiza de acuerdo a lo que se describe en la figura 4.57.

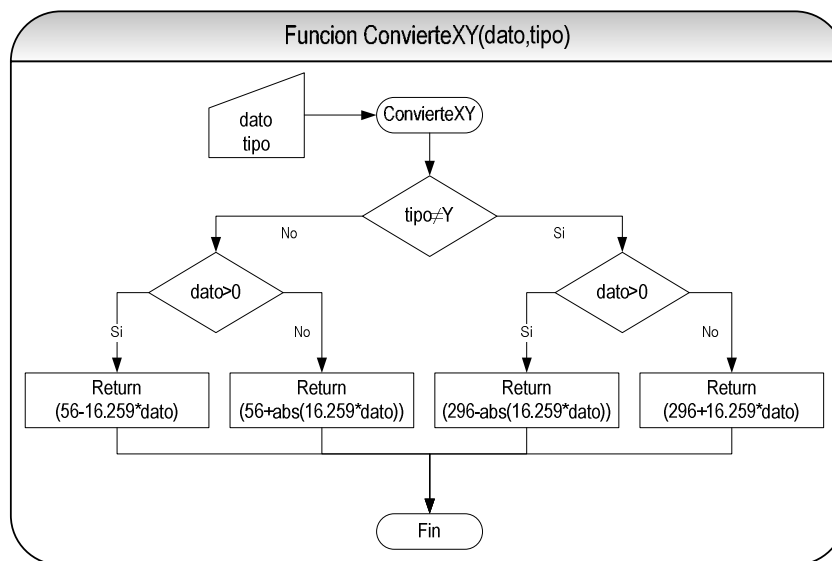
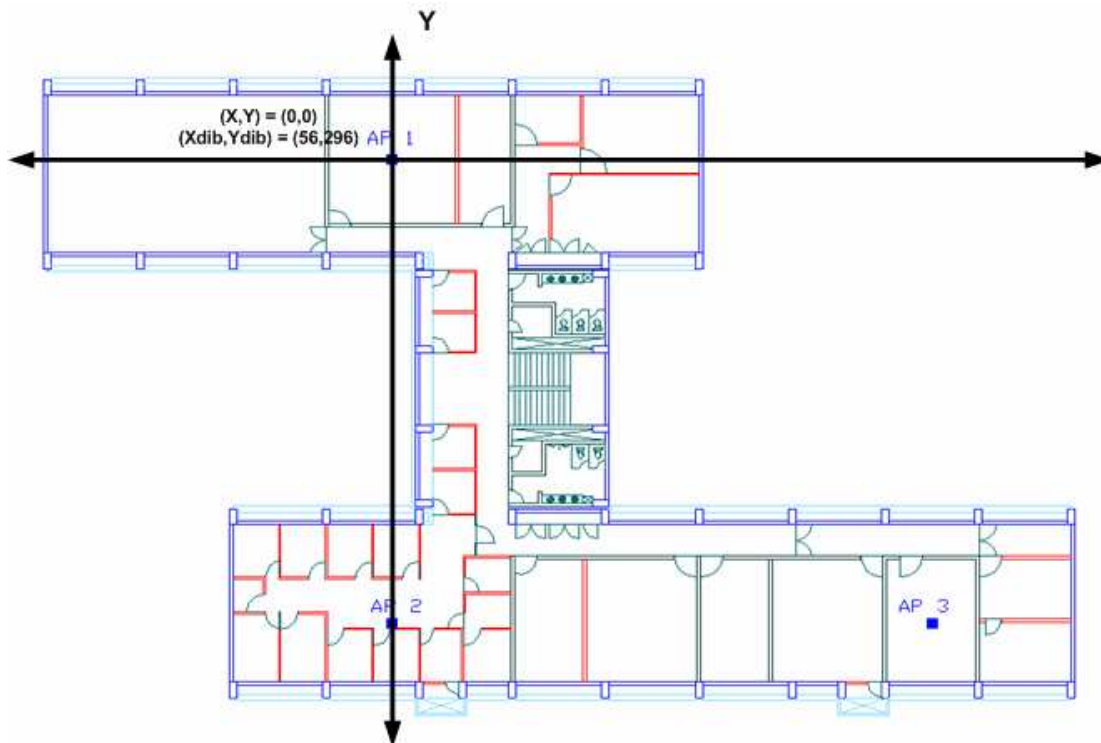


Figura 4.57 Diagrama de flujo de la función ConvierteXY.

Funcionalmente lo que se realiza es una conversión, basándose en el tipo de dato (X o Y) y la posición de este en el sistema cartesiano (esto se efectuó en función a los signos de las coordenadas), después se realiza un mapeo del sistema XY real a un sistema absoluto basado en unidades de dibujo representado por pixeles.

Para efectuar el mapeo de un sistema a otro, se estableció que el origen de nuestro sistema cartesiano estaría en la posición que ocupa el AP del laboratorio de redes, así la distribución (contenida en el archivo `/root/data/xy`) tendría incluida esta referencia cartesiana.



*Figura 4.58 Ubicación del sistema cartesiano.*

Posteriormente se llevo a cabo la equivalencia de unidades, como se puede ver en la figura 4.58, tenemos que al origen le corresponden las coordenadas (56,296) en unidades de dibujo<sup>26</sup>.

Los valores constantes que aparecen en los cálculos del diagrama 4.57 se calcularon con una simple regla de tres, lo ideal hubiese sido una sola conversión sin embargo, la peculiaridad de las coordenadas absolutas del área de dibujo nos obligó a hacer una distinción entre las regiones del sistema cartesiano.

5. Para finalizar el algoritmo de localización, una vez que son convertidas las coordenadas, se procede a la graficación del móvil, para esto, simplemente se manda llamar un comando predefinido de Kylix que nos posiciona la figura de acuerdo a las coordenadas en el dibujo

<sup>26</sup> El sistema coordinado del área de dibujo tiene su origen en la esquina superior izquierda del formulario, además, tienen la peculiaridad de que los valores de las ordenadas y abscisas nunca tendrán valores negativos.

#### 4.4.5 Pruebas preliminares del sistema.

A lo largo del tema que hace referencia a la programación de la aplicación, hemos visto diversas figuras que muestran capturas en pantalla del funcionamiento de WinNPS en tiempo real, en algunos casos hemos visto la teoría que propicia a que el sistema funcione de la manera descrita en los diagramas; con esto, de cierto modo queda comprobado de manera implícita el funcionamiento de la aplicación, sin embargo habría que formalizar un poco más este rubro.

Ahora bien, por pruebas preliminares del sistema nosotros queremos hacer referencia al nivel más básico de pruebas, es decir, son pruebas unitarias que comprueban la ejecución de los componentes programados de manera individual, de modo de que estos simplemente sean “llamados” para su ejecución.

Con anterioridad ya se mostraba que el programa funciona, en el siguiente capítulo se probará la funcionalidad y otras cuestiones más, aquí solamente mostraremos que el sistema en sí, compiló de manera correcta y que se ejecuta correctamente el *launcher* y demás aplicaciones.

Así pues, tenemos en primer instancia encontramos a las figuras 4.59, la cual nos muestra en la figura a) el estado de la compilación, se ve que todas las unidades que conforman a WinNPS compilaron de manera satisfactoria, solo existe un *warning*, y este se debe a unas variables auxiliares que en su momento se utilizaron en las pruebas de funcionamiento; en la figura b) encontramos el *log* de eventos que nos indica que el módulo WinNPS se carga en memoria durante la ejecución de la aplicación.



Figura 4.59a Resultado de la compilación de WinNPS.

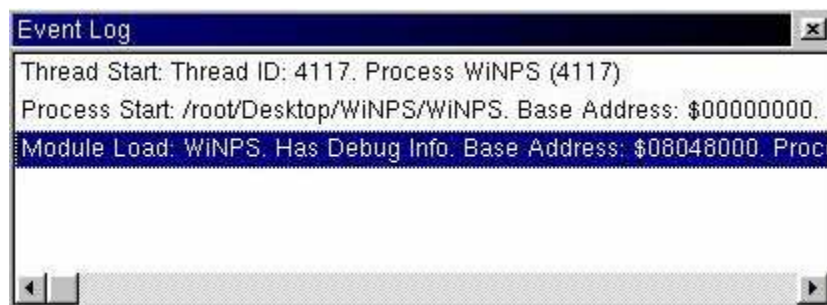


Figura 4.59b Carga de WinNPS en la memoria de la PC.

Al compilarse de manera satisfactoria, el archivo ejecutable de WinNPS se encontrará en la ruta donde se desarrolló la aplicación (*/root/Desktop/WINPS*), como prueba de ello tenemos la imagen de la figura 4.60, en la que se puede ver el archivo final sombreado.

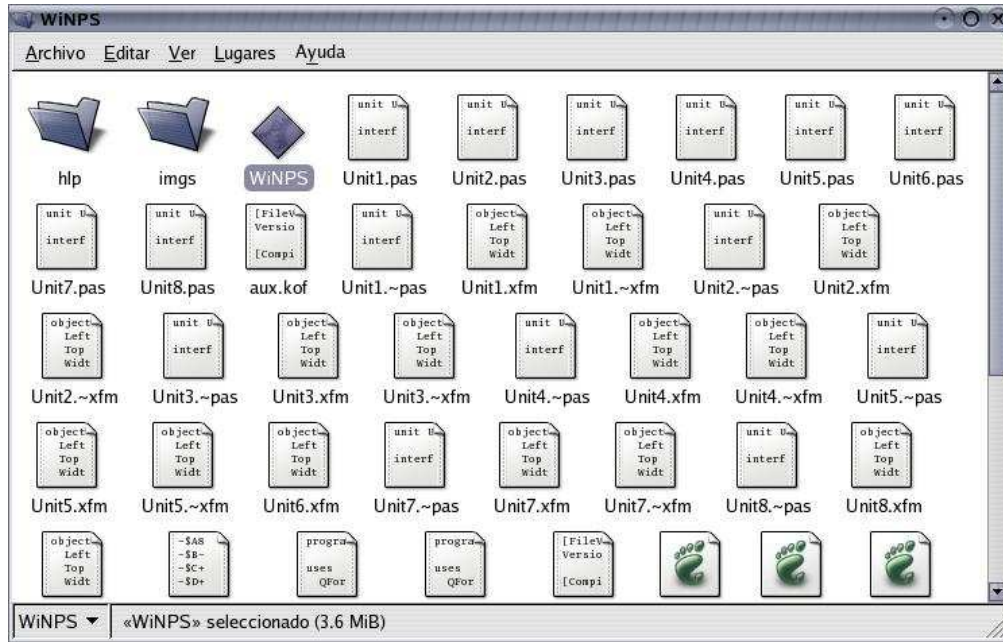


Figura 4.60 Archivo ejecutable de WinPS construido.

Para finalizar, mostraremos las figuras 4.61, 4.62, 4.63, 4.64 y 4.65 que muestran la evidencia de que WinPS funciona, por lo menos en el momento de vincular todas las herramientas que lo conforman; en el capítulo siguiente se realizarán otro tipo de pruebas que mostrará su funcionalidad. Luego entonces tenemos:

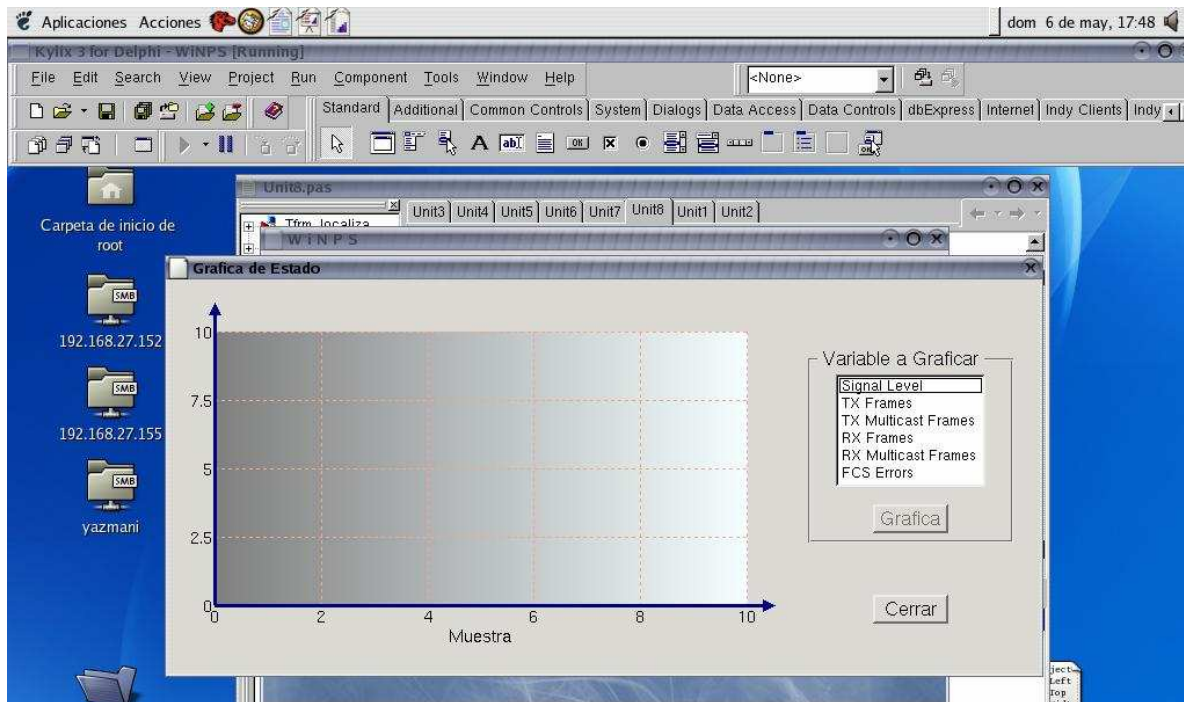
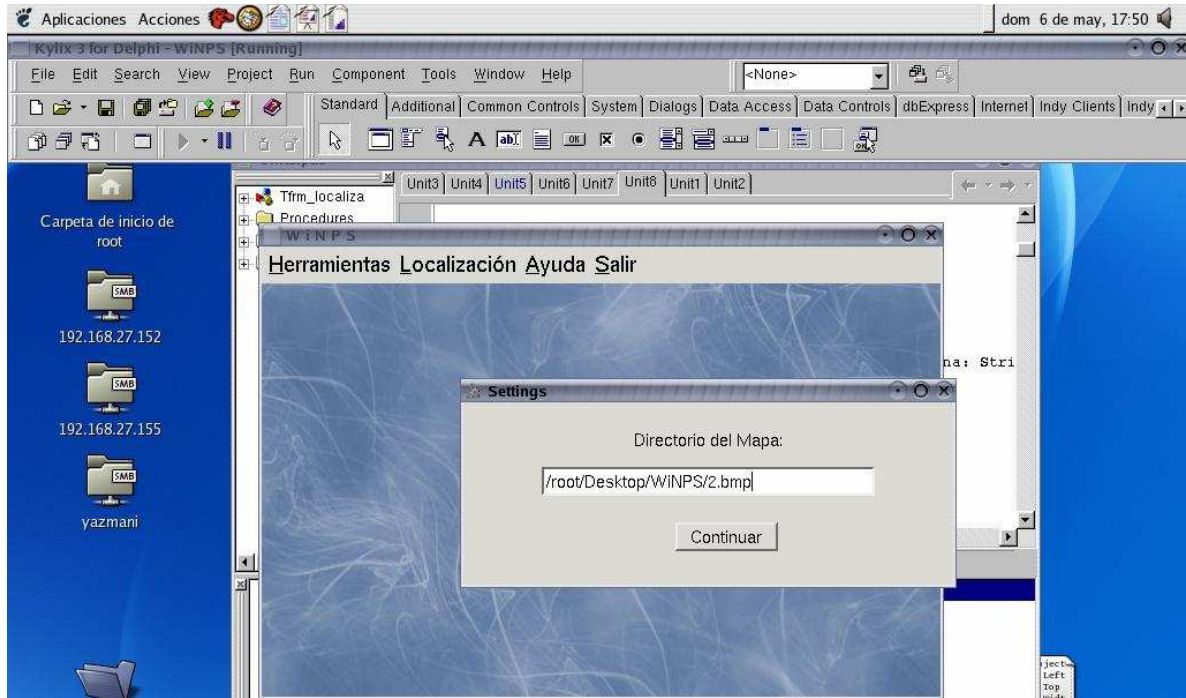
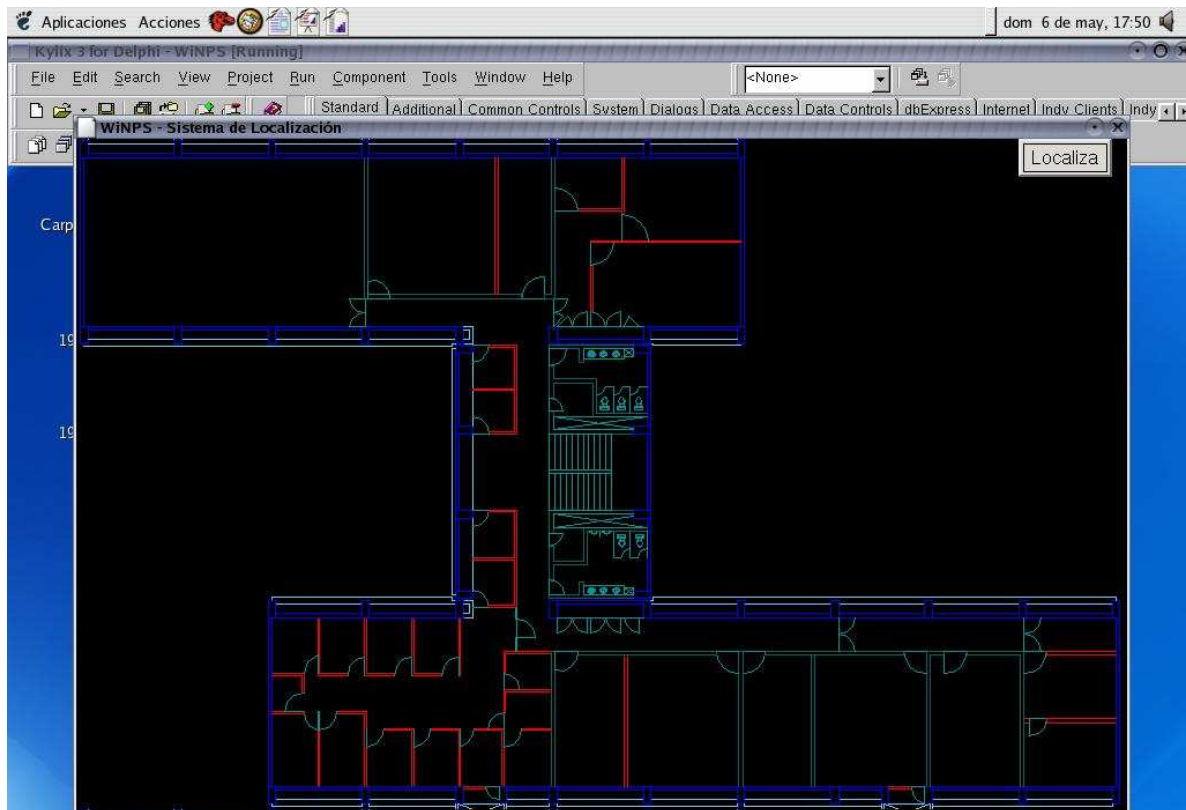


Figura 4.61 Carga de la aplicación de graficación.

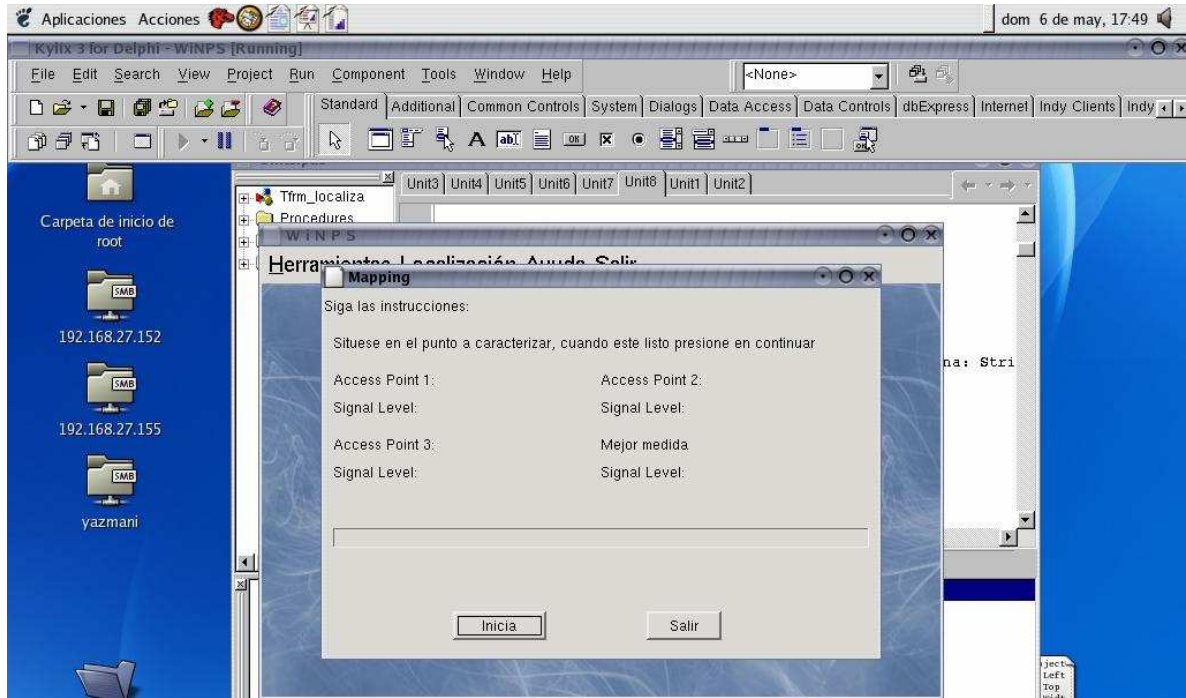




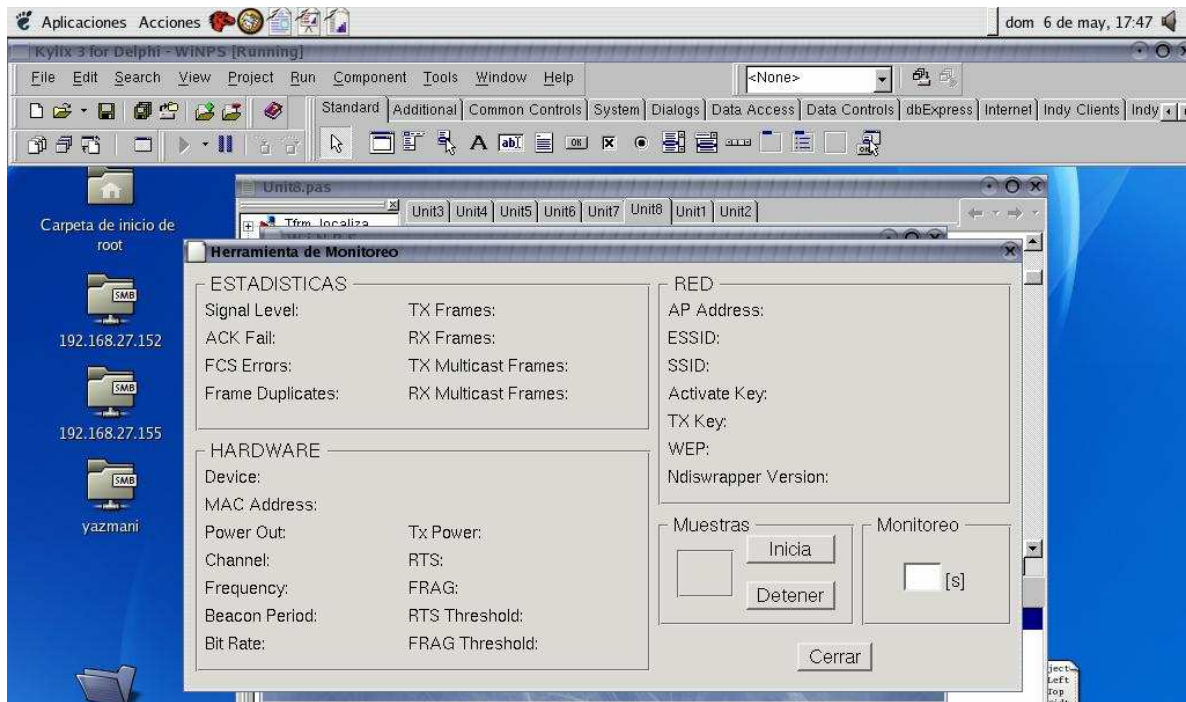
*Figura 4.62a Carga del mapa para la aplicación de localización.*



*Figura 4.62b Carga de la aplicación de localización.*



*Figura 4.63 Carga de la aplicación de mapeo.*



*Figura 4.64 Carga de la aplicación de monitoreo.*

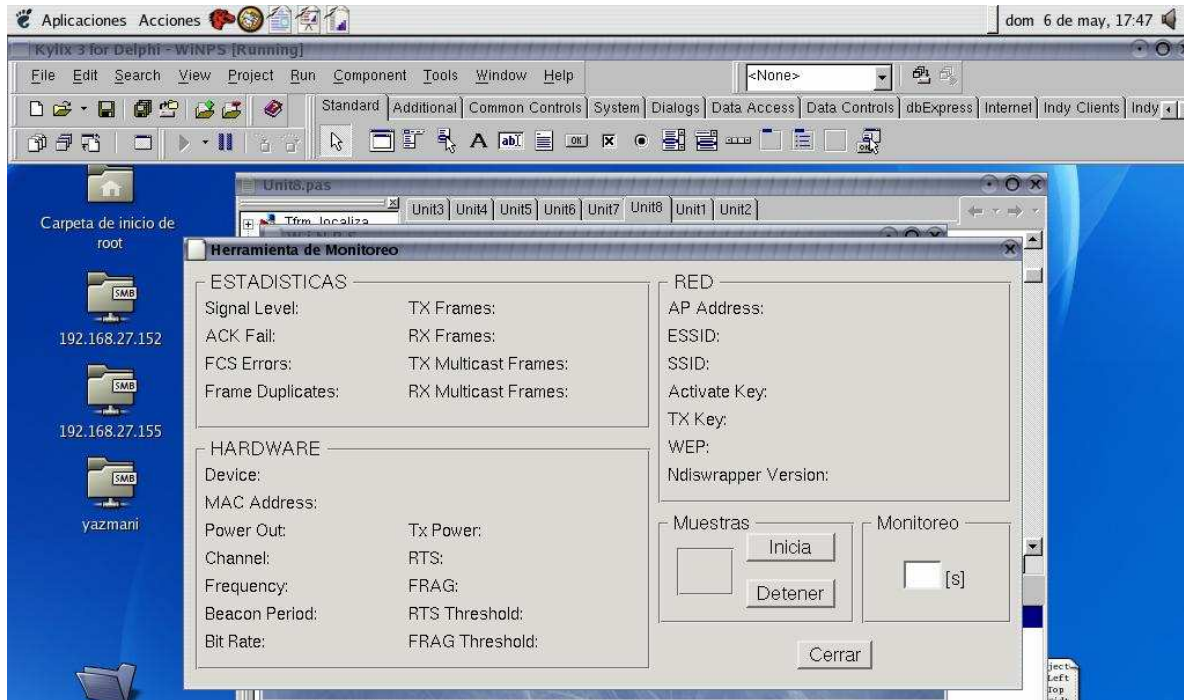


Figura 4.65 Carga de la aplicación de ayuda.

## 4.5 Consideraciones finales sobre el programa.

Para finalizar este capítulo, vamos a puntualizar los elementos más importantes que se deben de tomar en cuenta para la ejecución de la aplicación, también mencionaremos lo que puede realizar WiNPS. Así pues tenemos:

- ✦ WiNPS es un sistema que se instala en el cliente, y es independiente de la red, por lo que puede ser aplicado sobre cualquier red Wi-Fi.
- ✦ El sistema de localización está basado en *mapping*, el mapeo es diferente para cada *workspace* pues los obstáculos son diferentes para cada área de trabajo.
- ✦ Como el motor de localización se basa en *mapping*, WiNPS solo localizará a un móvil dentro del área del departamento de telecomunicaciones de la DIE.
- ✦ La localización se realiza solo dentro del pasillo del departamento de telecomunicaciones, no lo realiza dentro de laboratorios o cubículos de profesores. Se realiza de este modo por dos razones, la primera es que en la vida real, durante un recorrido a un centro comercial o museo, lo más importante es saber en donde te encuentras y donde están las salas, no en que parte de la sala te encuentras; la segunda es que con la reducción de espacios, el mapeo se hace más preciso y el algoritmo de localización arroja mejores resultados.
- ✦ No se toman en cuenta las localizaciones en redes Ad-hoc, pues estas siempre tendrán una ubicación relativa con respecto a otra terminal, lo que las hace más difíciles de rastrear en un área de trabajo amplia.



## CAPÍTULO V

### APLICACIÓN Y PRUEBAS FINALES

---

#### 5.1 Introducción.

Básicamente podemos definir a las pruebas como un proceso usado para identificar posibles fallas en la implementación, calidad o funcionalidad de un programa; se le puede considerar como una fase en el desarrollo de software que consiste en probar las aplicaciones construidas.

En un sistema de calidad de software (*Quality Assurance*), existen principalmente dos tipos de pruebas:

- ✦ *Pruebas unitarias*: Estas son comprobaciones que hacemos a las unidades lógicas de nuestro programa. Se verifica que una unidad funciona correctamente por sí misma, sin tener en cuenta las relaciones que pueda tener con otras partes del sistema.
- ✦ *Pruebas funcionales (o pruebas de sistema)*: Esta consiste en una comprobación global en la cual se hace énfasis en las colaboraciones entre unidades. Se prueba cada una de las opciones (o casos de uso) que ofrece el sistema, pudiendo ser procesos automáticos, acciones sobre la interfaz gráfica, etc.

En la definición que acabamos de dar, hemos hablado de unidades lógicas, aunque este concepto puede ser un poco ambiguo. Para ir entendiéndolo, diremos que las unidades lógicas de un programa son aquellas partes en que lo hemos dividido para entenderlo mejor. En nuestro caso estas unidades hacen referencia a los módulos que conforman a WiNPS.

Hacia el final del capítulo IV se hizo una referencia implícita a lo que son las pruebas unitarias, esto es, quedo demostrado que:

- ✦ *La compilación de WiNPS resultó exitosa.*
- ✦ *La construcción del ejecutable se realizó de manera correcta.*
- ✦ *Se mostró mediante capturas de pantalla, que cada uno de los módulos que conforman a WiNPS se "lanzan" sin generar algún error en el sistema.*
- ✦ *Como consecuencia del punto anterior, se comprobó también que cada unidad funciona correctamente sin tomar en cuenta las relaciones existentes entre si.*

Ahora bien, lo que resta son las pruebas funcionales del sistema, esto ya se vislumbraba cuando describíamos la programación de la aplicación, ahora bien, en el siguiente tema se comprobará que el sistema funciona correctamente tomando en cuenta las relaciones entre los módulos así como la interacción con el las variables físicas que se reciben del *workspace*.

## 5.2 Prueba del módulo Launcher.

### 5.2.1 Descripción.

Por la simplicidad de este módulo, para esta prueba se ejecutó la aplicación y se aguardó hasta que se cargó el módulo, posteriormente se procedió a la manipulación de los menús que se encuentran en el módulo

### 5.2.2 Resultados.

Básicamente esta unidad de WinNPS solo sirve para ejecutar todas las demás herramientas que la conforman. Ahora bien con base en el diseño de la interfaz para este que presentábamos en la figura 4.34, tenemos la figura 5.1, en la cual se ve la imagen que se obtiene durante el tiempo de ejecución del programa; como se puede ver, consiste de una ventana con un menú, donde este es el llamado *launcher*.



Figura 5.1 Imagen del Launcher durante la ejecución de WinNPS.

Para finalizar presentamos la tabla de resultados con la información más sobresaliente de la ejecución.

Unidad:	<b>Launcher</b>
Precondiciones:	Ninguna
Postcondiciones:	Ninguna
Excepciones:	Ninguna
<ol style="list-style-type: none"> <li>1. <b>¿Ocurrieron errores del sistema?</b> No</li> <li>2. <b>¿La unidad cumplió con la funcionalidad (explique)?</b> La unidad cumplió de manera satisfactoria con las especificaciones descritas en su diagrama de flujo, cada una de las aplicaciones se ejecutaron sin problema.</li> <li>3. <b>¿La unidad genera una terminación anormal (ABEND) si se le insertan datos erróneos o no hay red?</b> No aplica. Esta unidad no depende de datos externos, como consecuencia no depende de si hay o no conexión en la red</li> <li>4. <b>Comentario Final</b> El módulo se ejecutó de manera satisfactoria.</li> </ol>	

Tabla 5.1 Información general de la prueba para el Launcher.

## 5.3 Prueba del módulo graficación.

### 5.3.1 Descripción.

En el capítulo anterior ya se vio como opera internamente este módulo, ahora toca el turno de probar si realiza la funcionalidad para la que fue concebido. Así, para el caso de este módulo, se probará:

- ✦ Graficar cada una de las variables disponibles.
- ✦ ¿Qué ocurre cuando se selecciona otra variable y en ese momento la herramienta está graficando otra?

### 5.3.2 Resultados.

Con respecto a las dos preguntas que nos hacíamos en el punto anterior, podemos decir que, en efecto, realiza la graficación de cada una de las seis variables físicas expuestas en la lista. Para el segundo punto, vimos que la graficación se detiene en cuanto se pasa a otra variable, no genera ningún error.

En cuanto a la funcionalidad básica, la ventana se puede cerrar, mover, cambiar el tamaño, sin temor a que se genere un error. En la figura 5.2 mostramos la ejecución en tiempo real de la herramienta de graficación, en esa imagen se muestra la intensidad de señal, de la cual se han graficado 309 muestras hasta la captura de la imagen.

Esta herramienta nos puede recordar al *KWiFiManager*, solo que aquí podemos graficar otras variables a parte de la Intensidad de señal, asimismo, vemos que nuestra herramienta de graficación presenta una mayor sensibilidad a las variaciones de los factores físicos que se muestrean.

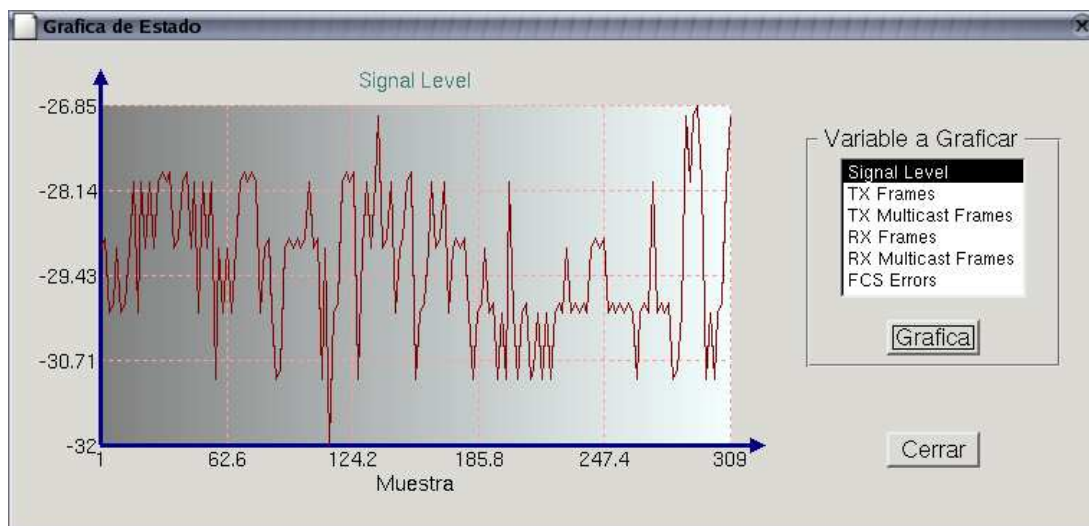


Figura 5.2 Imagen de la herramienta de Graficación durante la ejecución de WiNPS.

Para finalizar presentamos la tabla con los resultados con la información más sobresaliente de la ejecución.

Unidad:	Graficación
Precondiciones:	Se requiere tener activa la interfaz de red inalámbrica.
Postcondiciones:	Ninguna
Excepciones:	Si no existe la interfaz inalámbrica, se creará un mensaje de error.
<p><b>1. ¿Ocurrieron errores del sistema?</b> No</p> <p><b>2. ¿La unidad cumplió con la funcionalidad (explique)?</b> Sí, todas las variables se graficaron de manera correcta, la unidad cumplió de manera satisfactoria con las especificaciones descritas en su diagrama de flujo.</p> <p><b>3. ¿La unidad genera un ABEND si se le insertan datos erróneos o no hay red?</b> A esta unidad no se le puede ingresar un valor, por lo que ese caso no aplica, sin embargo, si no existe una interfaz de red, el módulo despliega en pantalla un mensaje de advertencia y le pide al usuario que verifique su conexión.</p> <p><b>4. Comentario Final</b> El módulo se ejecutó de manera satisfactoria, lo único que habría que preguntarse es si los resultados obtenidos son correctos, pues estos presentan muchas variaciones, a nuestra consideración estos se debe a que el diagrama e interfaz utilizada presentan una mayor sensibilidad.</p>	

*Tabla 5.2 Información general de la prueba para el módulo de graficación.*

## 5.4 Prueba del módulo de monitoreo.

### 5.4.1 Descripción.

Este módulo es el que nos sirvió en su momento para obtener el mapa de intensidad de señal del área de trabajo, también se vio como es posible que logre tales resultados. Así pues, toca el momento de verificar la funcionalidad de esta unidad, por lo se probará:

- ⊕ *Funcionalidad del módulo en base a lo descrito en el apartado de la programación.*
- ⊕ *¿Qué pasa si se inhabilitar la red antes y durante la ejecución del módulo?*
- ⊕ *¿Que ocurre cuando no se ingresa un valor válido en el tiempo de muestreo?*
- ⊕ *¿Qué sucede cuando no se ingresa un valor y se trata de ejecutar la aplicación?*
- ⊕ *Cuando se hace clic en detener y luego en continuar, ¿el número de muestras se reinicia?*

### 5.4.2 Resultados.

Con base en la cuestiones anteriores, en primer lugar encontramos que en efecto, la herramienta de monitoreo realiza el muestreo de la intensidad de señal muestreando la información cada  $x$  segundos.

En segundo lugar, si la red se inhabilita antes o después, se obtendrá en pantalla un mensaje de error que indica que se encontró un problema con la conexión, y se le pedirá al usuario compruebe el cable.

Para el caso en el que no se ingresa un valor válido o bien se trata de ejecutar la herramienta sin haber ingresado un valor de tiempo de muestreo, entonces saldrá un men-

saje en el que se le pedirá al usuario que ingrese un valor coherente. Y finalmente, cuando se detiene el muestreo, la aplicación comienza en la muestra que se quedó.

En la figura 5.3 se puede apreciar como la herramienta funciona, de acuerdo a la interfaz que mostrábamos en la figura 4.26, es aquí donde todas las etiquetas se llenan con la información de los datos de la red captados por NDIS del OS; en la figura se observa la doceava muestra tomada en intervalos de un segundo.

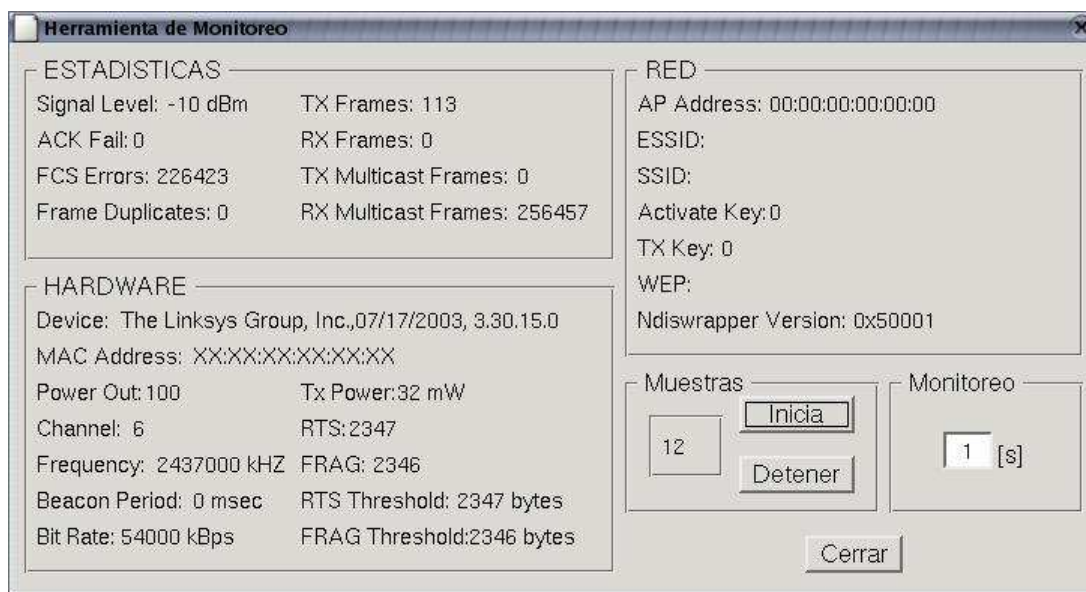


Figura 5.3 Imagen de la herramienta de Monitoreo durante la ejecución de WinNPS.

Así pues, la tabla de resultados será el que se muestra a continuación:

Unidad:	<b>Monitoreo</b>
Precondiciones:	Se requiere tener activa la interfaz de red inalámbrica.
Postcondiciones:	Se crea un archivo con las muestras tomadas
Excepciones:	Si no existe la interfaz inalámbrica, se creará un mensaje de error. Si el dato ingresado no es válido, se obtendrá un error.
<ol style="list-style-type: none"> <li>1. <b>¿Ocurrieron errores del sistema?</b> No</li> <li>2. <b>¿La unidad cumplió con la funcionalidad (explique)?</b> Sí, todas las variables se muestrearon en el tiempo que el usuario indicó, algunas de ellas no mostraban ningún resultado, esto se deba que el sistema de archivos solo recibe la información que se capta directamente de la tarjeta, es por ello que la totalidad de la información dependerá de la compatibilidad de la tarjeta.</li> <li>3. <b>¿La unidad genera un ABEND si se le insertan datos erróneos o no hay red?</b> No se genera ningún ABEND, cuando se da cualquiera de los dos casos, el programa solo manda mensajes de error y no termina la ejecución de WinNPS.</li> <li>4. <b>Comentario Final</b> La ejecución del módulo se realiza de acuerdo a lo descrito en el diseño técnico y funcional de la herramienta.</li> </ol>	

Tabla 5.3 Información general de la prueba para el módulo de monitoreo.

## 5.5 Prueba del módulo de mapeo.

### 5.5.1 Descripción.

Como bien ya hemos mencionado en diversos apartados, esta herramienta realiza un proceso similar al módulo de monitoreo, con la diferencia de que aquí se realiza de manera simultánea, la pregunta obligada para las pruebas funcionales serían:

- ✦ Que ocurre si alguno de los AP está pagado, o bien no se capta señal de alguno de ellos.

### 5.5.2 Resultados.

Durante la concepción de esta herramienta se tomó en cuenta que existiría la posibilidad de que la terminal no reportara la intensidad de señal de un AP, para esos casos, a la variable se le asignaba un valor nulo, es por ello que el programa no crearía un ABEND si se llegará a dar ese caso.

En la figura 5.4 vemos una de las ventanas que presenta la herramienta durante la ejecución de WiNPS; como se puede ver, se capta la dirección MAC de cada AP así como la intensidad de señal que recibe de cada uno, además se presenta la mejor medida que se captó en el instante anterior, esto para tener una referencia y poder ver la evolución a corto plazo de la señal.

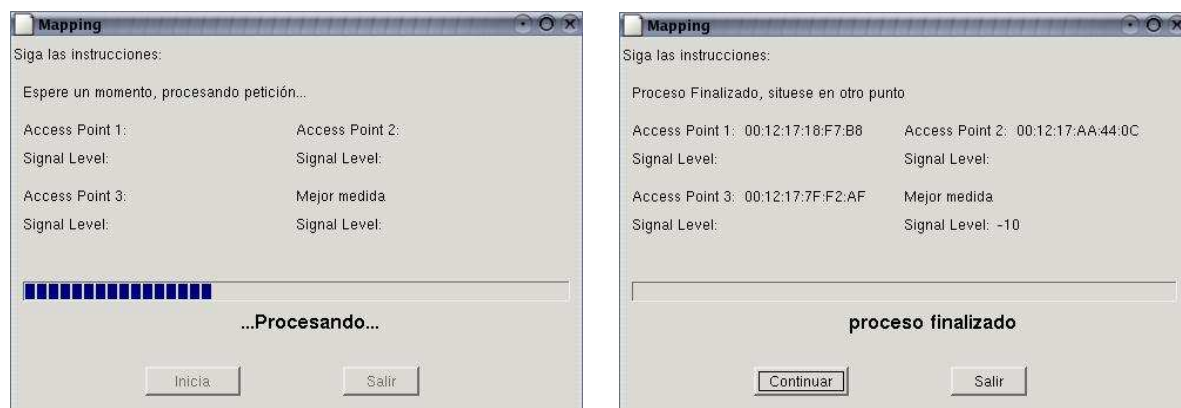


Figura 5.4 Imagen de la herramienta de Mapping durante la ejecución de WiNPS.

Ahora bien, la tabla de resultados queda de la siguiente manera:

Unidad:	<b>Mapping</b>
Precondiciones:	Se requiere tener activa la interfaz de red inalámbrica.
Postcondiciones:	Se crea un archivo con las muestras tomadas para cada AP.
Excepciones:	Ninguna
<ol style="list-style-type: none"> <li>1. <b>¿Ocurrieron errores del sistema?</b> No</li> <li>2. <b>¿La unidad cumplió con la funcionalidad (explique)?</b> Sí, la información de los tres AP fue captada correctamente, estos se corroboró con los archivos pertinentes en la dirección asignada.</li> <li>3. <b>¿La unidad genera un ABEND si se le insertan datos erróneos o no hay red?</b> No se genera ningún ABEND, pues el programa está preparado para pérdidas de señal.</li> <li>4. <b>Comentario Final</b> La ejecución del módulo se realiza de acuerdo a lo descrito en el diseño técnico y funcional de la herramienta.</li> </ol>	

Tabla 5.4 Información general de la prueba para el módulo de mapping.



## 5.6 Prueba del módulo de ayuda.

### 5.6.1 Descripción.

Al igual que el *launcher*, este módulo tiene un funcionamiento bastante básico e independiente de la red, así como de entradas explícitamente dispuestas por el usuario, con esto, la prueba funcional consistirá en que el módulo se ejecutará y deberá cargar los archivos de ayuda de acuerdo a la elección del usuario.

### 5.6.2 Resultados.

Se obtuvo la correcta carga de la GUI que conforma el módulo, tal y como lo muestra la evidencia de la figura 5.5; cuando el usuario elegía algún tópico, la información era desplegada en el área designada para ello.

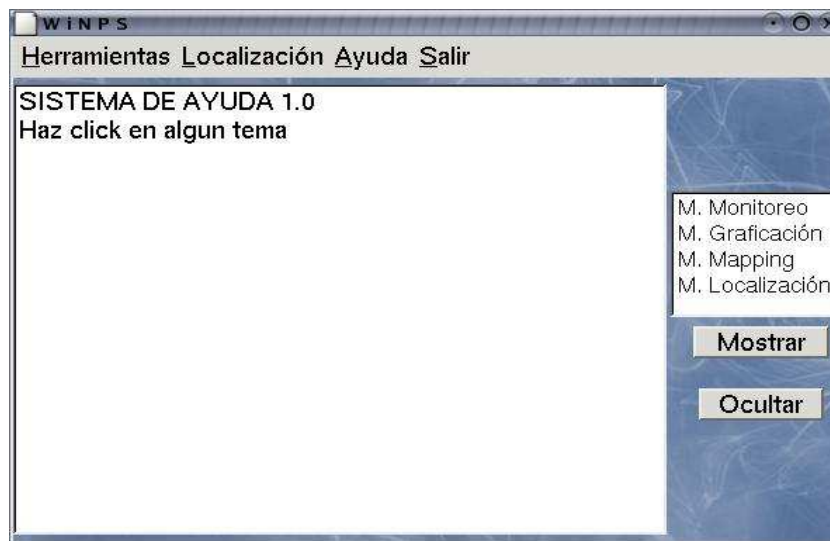


Figura 5.5 Imagen de la herramienta de Mapping durante la ejecución de WinPS.

De manera general, se puede decir que este módulo cumple con la funcionalidad para la que fue creado, a continuación se presenta la tabla con los resultados obtenidos.

Unidad:	<b>Ayuda</b>
Precondiciones:	Ninguna
Postcondiciones:	Ninguna
Excepciones:	Ninguna
1.	<b>¿Ocurrieron errores del sistema?</b> No
2.	<b>¿La unidad cumplió con la funcionalidad (explique)?</b> Sí, la funcionalidad es muy básica y solo consiste en desplegar textos de ayuda sobre la ventana del launcher.
3.	<b>¿La unidad genera un ABEND si se le insertan datos erróneos o no hay red?</b> No se genera ningún ABEND, pues el programa es independiente a estas dos condiciones.
4.	<b>Comentario Final</b> La ejecución del módulo se realiza de acuerdo a lo descrito en el diseño técnico y funcional de la herramienta.

Tabla 5.5 Información general de la prueba para el módulo de ayuda.

## 5.7 Prueba del módulo de localización.

### 5.7.1 Descripción.

La descripción de esta prueba final se realizó por dos razones, la primera es que es el tema medular del presente trabajo, y la segunda, por su complejidad en la funcionalidad, es necesario presentar el comportamiento que tuvo el sistema de localización al tratar de encontrar la posición en tres lugares distintos. Así pues, para probar la funcionalidad, se describirá de la manera más específica posible los resultados obtenidos para la localización de estos tres puntos.

### 5.7.2 Resultados.

En primera instancia, presentaremos el entorno necesario para la ejecución del motor de localización, esencialmente el algoritmo de localización requiere de dos archivos, uno que contiene la información de las coordenadas XY y otro que contiene la información de las intensidades de señal, en la 5.6 se muestran estos tres archivos.



Figura 5.6 Archivos base para la ejecución del sistema de posicionamiento.

Ahora bien, cuando se ejecuta el sistema, se le pide al usuario que ingrese la ruta en la que se encuentra el mapa, tal y como lo muestra la figura 5.7.

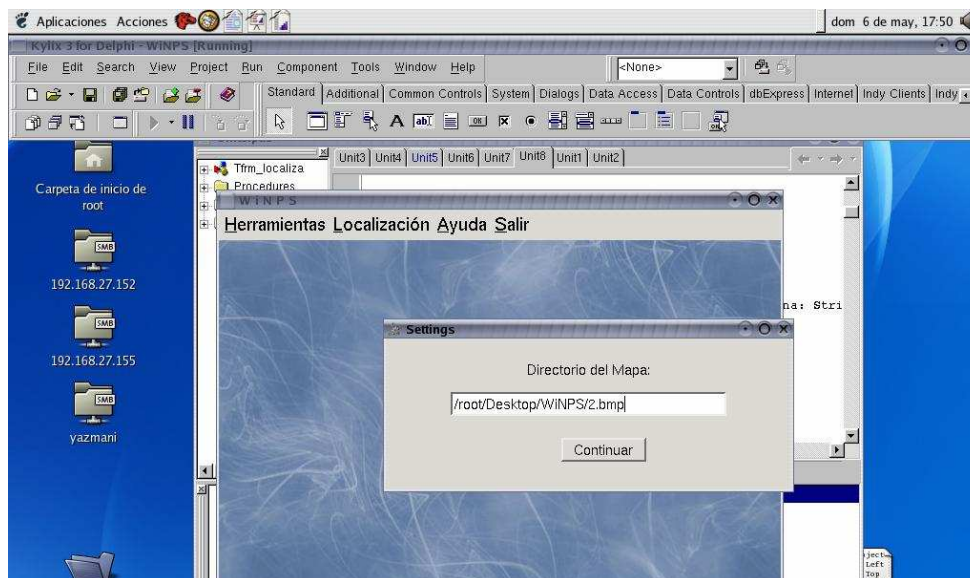
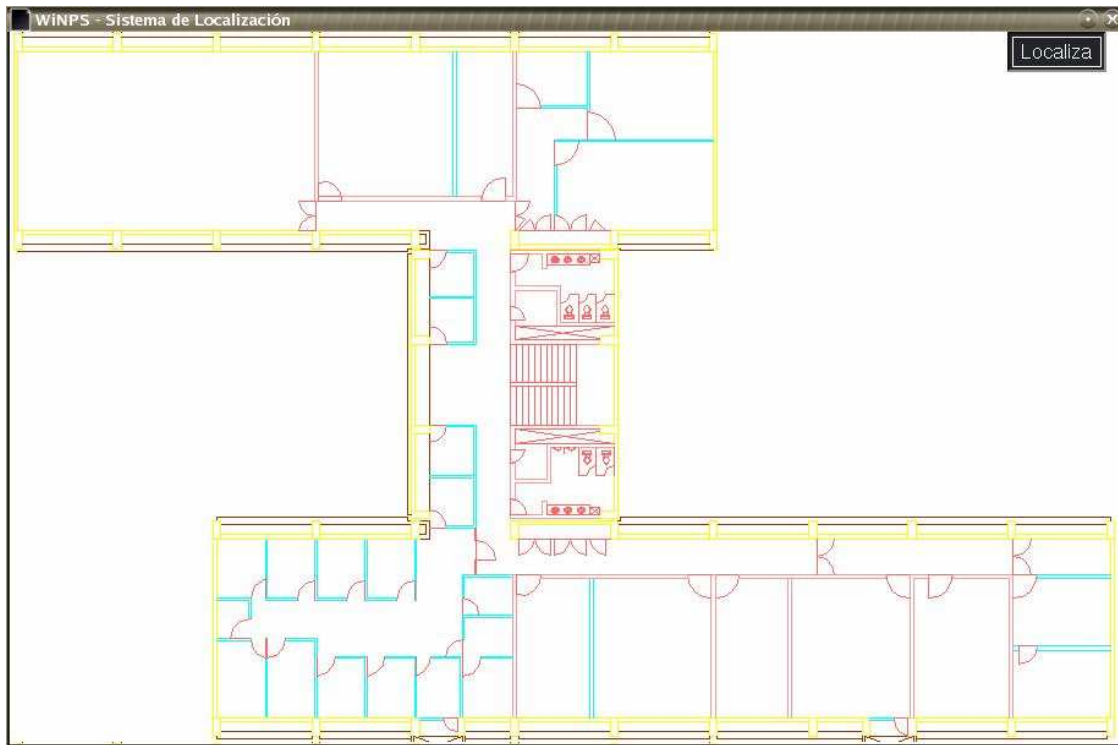


Figura 5.7 Petición del directorio en donde se aloja el mapa del workspace.



Una vez que se da la ubicación del mapa, el sistema lo cargará en la interfaz que se mostró en la figura 4.28, dando como resultado la creación de una GUI como la que se muestra en la figura 5.8<sup>27</sup>.



*Figura 5.8 GUI del sistema de localización en tiempo de ejecución.*

Cuando se terminó de cargar la interfaz, entonces se procede a la captura de la intensidad de señal que el usuario recibe desde el punto en el que se encuentra parado. Como nosotros estamos realizando la prueba sabemos que el usuario tiene las siguientes características:

- ✦ *Posición: (13.5,-21.5), el usuario se encuentra a la mitad del primer ventanal pasando el laboratorio de Comunicaciones Analógicas.*
- ✦ *El usuario está recibiendo una intensidad de señal de la siguiente manera:*

*AP1=-48.42[dbm]*

*AP2=-80.22[dbm] y*

*AP3=-56.10 [dbm]*

Con estos datos y basándonos en la información de los datos del mapa de señal, resulta obvio que los valores de la sumatoria de X y Y son distintos de cero, por lo que el sistema calculó que el móvil está en algún lado de la región 4, posteriormente con el valor promedio realiza un aproximación y obtiene la posición tal y como la muestra la figura 5.9, que como se puede apreciar, representa al móvil como un círculo, cuyo diámetro es de dos metros.<sup>28</sup>

---

<sup>27</sup> La imagen original presenta tonos muy oscuros y resulta difícil distinguir las divisiones, es por ello que la imagen de las figuras 5.8 y subsecuentes, serán de color claro.

<sup>28</sup> Esta es la región de tolerancia que se ha definido para la localización con WinPS.

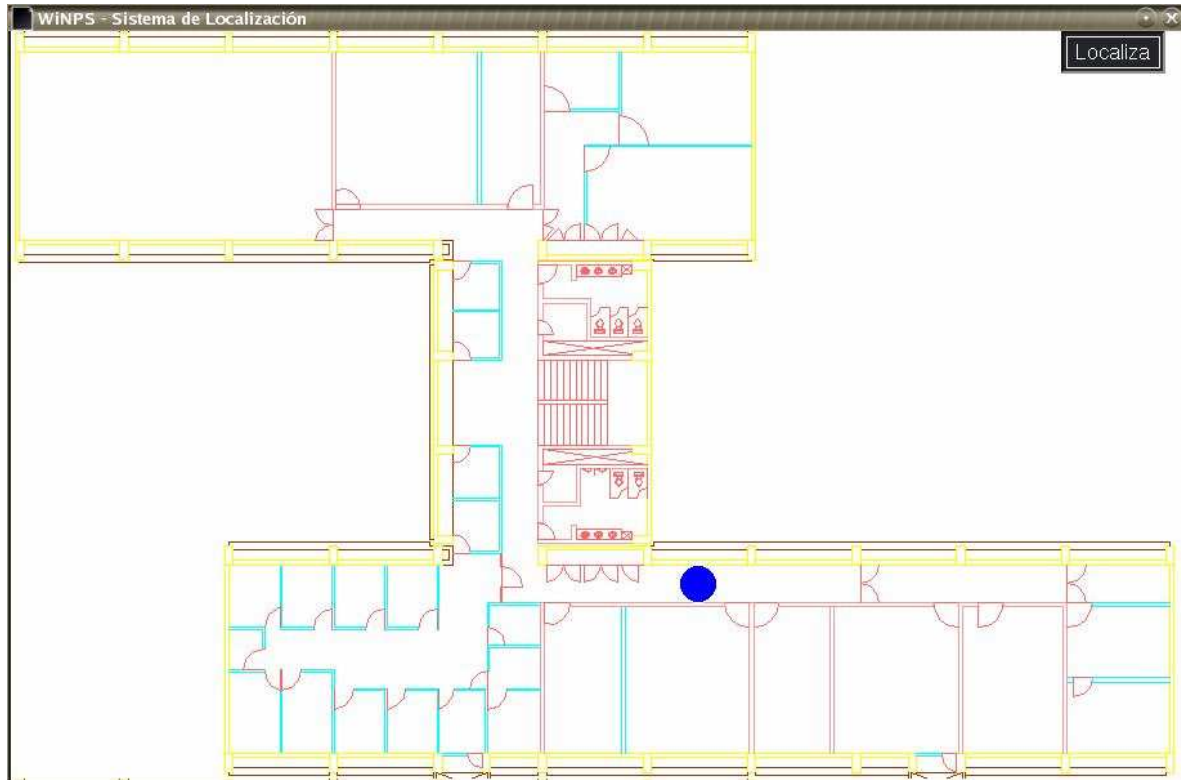


Figura 5.9 GUI del sistema de localización con el primer objeto hallado.

En ese caso el sistema calculó que el móvil debe de estar en la coordenadas **(13.98,21.71)**, esto presenta un **error de localización de 0.48 [m] para X y 0.21 [m] para Y.**

Ahora bien para el segundo caso, el usuario tendrá las siguientes condiciones:

- ⊕ *Posición: (-1.04, -25.14), el usuario se encuentra a la mitad del pasillo donde están los cubículos de los profesores, en frente del cubículo del Dr. Neri Vela.*
- ⊕ *El usuario está recibiendo una intensidad de señal de la siguiente manera:*

*AP1= -42.15 [dbm]*

*AP2= -70.37 [dbm] y*

*AP3= -44.10 [dbm]*

Nuevamente con estos valores, se obtiene una sumatoria de X y Y distintas de cero, cuando se realiza la comprobación de la región se encuentra que el punto cae en algún lado de la región 3; y cuando se obtiene la estimación con el valor medio, el sistema ubica al usuario en **(-0.93, -24.8)**, esto presenta un **error de 0.11 [m] para X y 0.34 [m] para Y.** En la figura 5.10, se puede ver la captura de pantalla cuando el sistema de localización ubica al usuario.

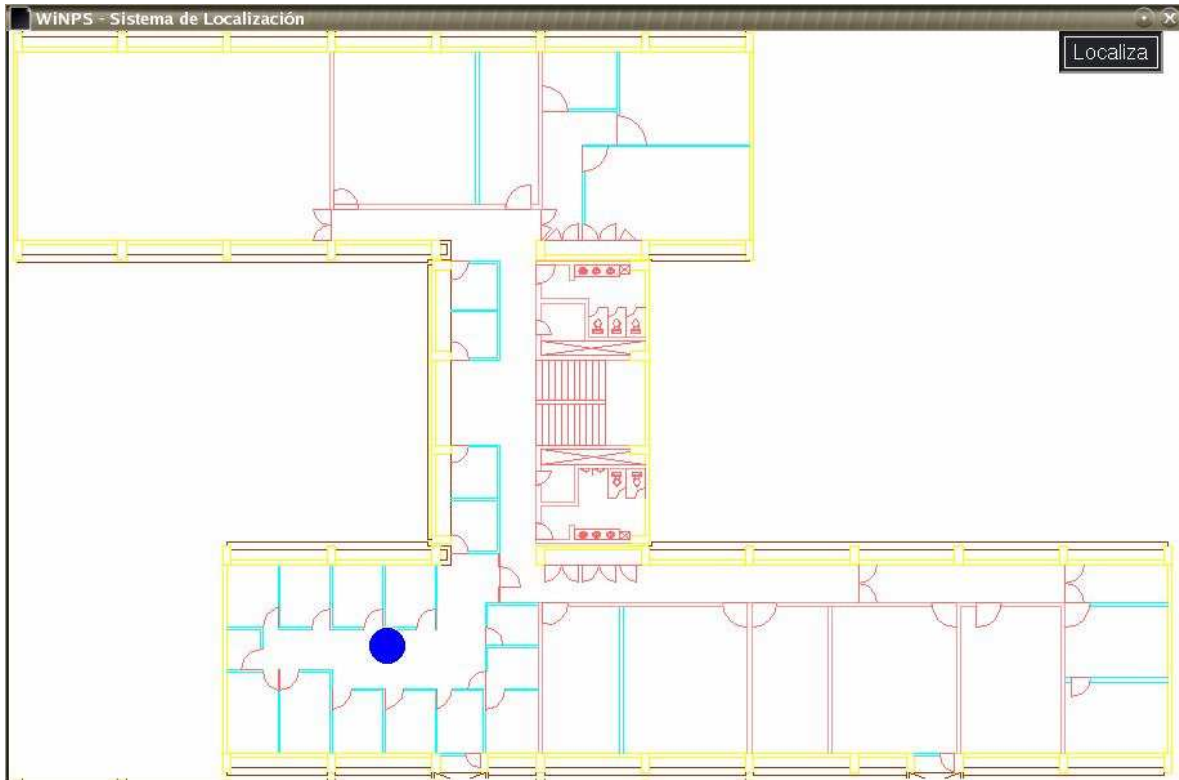


Figura 5.10 GUI del sistema de localización con el segundo objeto hallado.

Para finalizar la prueba funcional el sistema debe de localizar al usuario que cuenta con las siguientes condiciones:

- ✦ Posición: (3, -5), el usuario se encuentra en la columna que se localiza frente al cubículo de la profesora Margarita.
- ✦ El usuario está recibiendo una intensidad de señal de la siguiente manera:

$$AP1 = -57.75 \text{ [dbm]}$$

$$AP2 = -69.25 \text{ [dbm]} \text{ y}$$

$$AP3 = -60.10 \text{ [dbm]}$$

Con estos datos el sistema no encuentra ninguna coincidencia en el archivo del mapping, por lo que las sumatorias de X y Y son nulas, con esto, el sistema vuelve a tomar una nueva medición, obteniendo las siguientes intensidades:

$$AP1 = -67.20 \text{ [dbm]}$$

$$AP2 = -52.45 \text{ [dbm]} \text{ y}$$

$$AP3 = -53.50 \text{ [dbm]}$$

Con esta información, las sumatorias han dejado de ser nulas, sin embargo, al tratar de ubicar al punto promedio dentro de una región, es decir, el sistema devolvió una región nula, con esto, nuevamente se vuelven a tomar datos de intensidad de señal, quedando de la siguiente manera:

$$AP1 = -64.13 \text{ [dbm]}$$

AP2= -38.25 [dbm] y

AP3= -41.15 [dbm]

Ya con estos valores, las sumatorias resultan ser distintas de cero y se localiza al punto en la región 1, luego entonces con el valor promedio el sistema ubica al usuario en la posición (3.39, -4.67) con esto el **error en la medición sería de 0.39 [m] para X y 0.33 [m] en Y**. Así pues, en la figura 5.11 mostramos esta última figura de la localización del móvil.

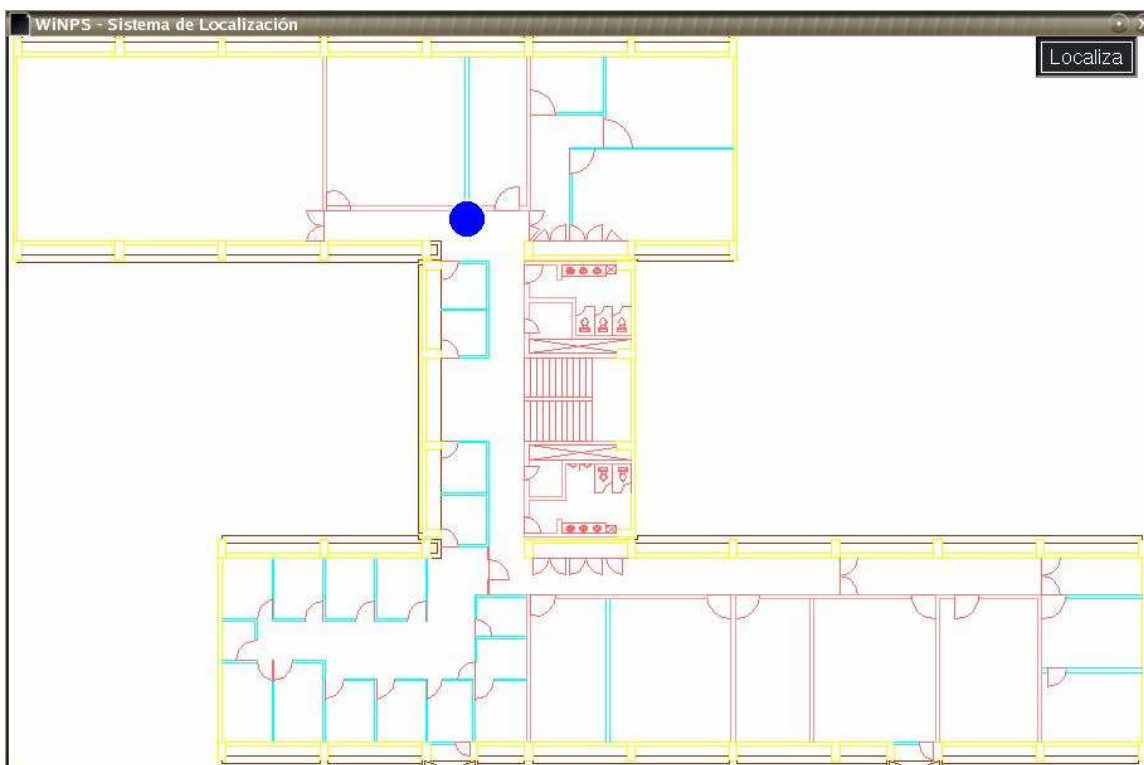


Figura 5.11 GUI del sistema de localización con el tercer objeto hallado

De manera general, se puede decir que este módulo cumple con la funcionalidad y su sistema de prevención de errores garantiza una medida confiable; a continuación se presenta la tabla de resultados.

Unidad:	Localización
Precondiciones:	Ninguna
Postcondiciones:	Ninguna
Excepciones:	Ninguna
<ol style="list-style-type: none"> <li>1. <b>¿Ocurrieron errores del sistema?</b> No</li> <li>2. <b>¿La unidad cumplió con la funcionalidad (explique)?</b> Sí, el proceso ya fue descrito mediante ejemplos en las líneas anteriores</li> <li>3. <b>¿La unidad genera un ABEND si se le insertan datos erróneos o no hay red?</b> No se genera ningún ABEND, pues el programa tiene su propia corrección de errores.</li> <li>4. <b>Comentario Final</b> La ejecución del módulo se realiza de acuerdo a lo descrito en el diseño técnico y funcional de la herramienta.</li> </ol>	

Tabla 5.6 Información general de la prueba para el módulo de localización

## CAPÍTULO VI CONCLUSIONES

---

El objetivo fundamental del presente trabajo, ha sido mostrar el proceso de construcción e implantación de un sistema de localización en redes inalámbricas híbridas Wi-Fi y Ad-hoc. A lo largo de los diversos capítulos, hemos mostrado las etapas que se requirieron para implementar nuestro sistema de localización.

En términos generales podemos decir que, con el desarrollo de este trabajo de tesis, se pudieron satisfacer todos aquellos objetivos generales que presentamos durante la introducción; asimismo, incrementamos nuestro conocimiento en algunas áreas de las telecomunicaciones y la computación. El poder plasmar nuestras ideas en este trabajo fue totalmente satisfactorio, asimismo, podemos concluir que este, fue un trabajo ampliamente sustentado pues logramos:

- ✦ *Conocer y compartir los principales detalles referentes a las diversas tecnologías de localización que se tienen en la actualidad.*
- ✦ *Investigar y aprender sobre los elementos principales que conforman al estándar IEEE 802.11.*
- ✦ *Implementar un sistema con ayuda de las herramientas desarrolladas de mapeo y monitoreo, para así dar lugar a lo que hemos definido como sistema de posicionamiento en una red inalámbrica (WiNPS - Wireless Network Position System).*
- ✦ *Hacer que WiNPS fuese un sistema independiente al servidor de la red, pues WiNPS funciona solo sobre las intensidades de señal captadas en la terminal, de los puntos de acceso.*

Ahora bien, en las siguientes líneas pondremos las conclusiones pertinentes a cada objetivo que plateamos al inicio de este trabajo.

### **6.1 Acerca del proceso de diseño e implementación de WiNPS.**

Hablando del proceso de diseño e implementación, podemos decir que este objetivo se cumplió de manera satisfactoria, pues pudimos aplicar nuestros conocimientos de diversas materias de ingeniería para el proceso de construcción de este sistema.

Es decir, a lo largo de las diversas materias que se imparten en la carrera, hemos visto que para desarrollar un sistema ya sea software o electrónico, son imprescindibles una serie de etapas, las cuales se han definido como Diseño, Construcción y Pruebas, así pues, a lo largo de los capítulos hemos reforzado esta manera de implementación, por lo que podemos concluir que dicho proceso tuvo la siguiente estructura:

- ✦ *Análisis.* Durante el capítulo 1 y 2 se estipularon las bases tecnológicas con las que contamos en la actualidad para el desarrollo de WiNPS, es decir, se analizaron las funcionalidades generales de los sistemas de localización, así como la teoría implicada en el protocolo IEEE 802.11 que es con el que se construyen las redes Wi-Fi.

- ✦ *Diseño.* En el capítulo número 3 se propuso la solución basada en el análisis, se puede decir que este capítulo fue una consolidación de toda la teoría de los capítulos anteriores; con ello se realizó un diseño para la construcción.
- ✦ *Construcción.* Esta etapa es la que se describe en el capítulo 4, pues, a lo largo de diagramas de casos de uso, diseños de interfaz, así como diagramas de flujo, se realizó la construcción de WiNPS.
- ✦ *Pruebas.* Las pruebas las comenzamos a vislumbrar hacia el final del capítulo 4, estas pruebas básicamente eran de funcionalidad, esto se realizó para conocer si cada módulo se ejecutaba y no existía algún error en el código.
- ✦ *Implantación y pruebas de integración.* Esta parte es la que vemos en el capítulo 5, lo que se hace aquí es probar cada módulo como un conjunto al que llamamos WiNPS, con esto se busca ver que el sistema sea implantado dentro de un *workspace*, el cual, es el área del departamento de telecomunicaciones de la DIE.

Otro de los puntos importantes al respecto a este objetivo, es lo que aparece dentro de la figura 6.1.

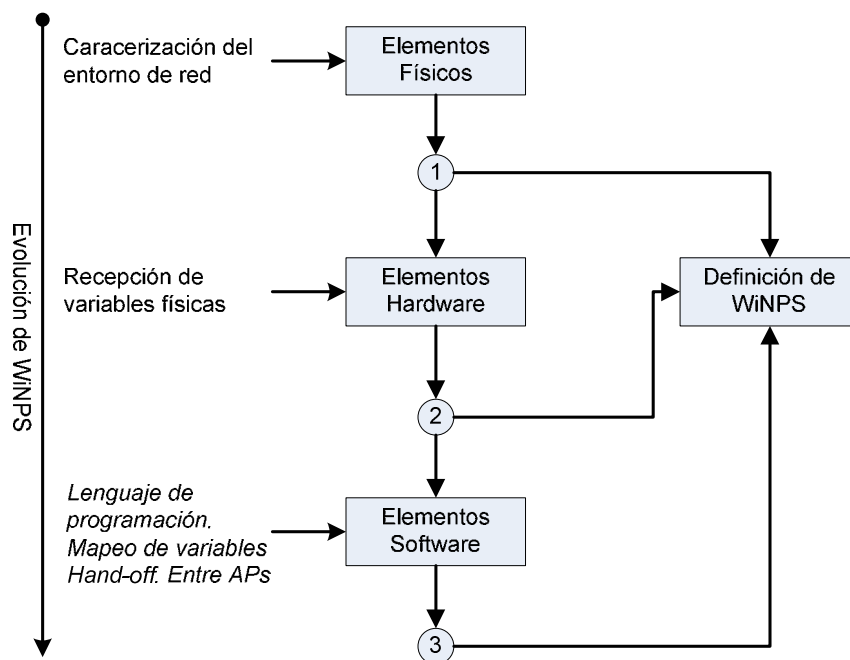


Figura 6.1 Evolución de WiNPS.

A partir de los elementos que aparecen, podemos realizar algunas conclusiones sobre las etapas de Análisis y Diseño de WiNPS.

- ✦ *Para diseñar e implementar un sistema en el que exista una interacción Usuario-Programa-Ambiente se deben de tomar en cuenta tres elementos, estos son los elementos físicos, los elementos de Hardware y los elementos de Software.*

- ✦ *WiNPS utilizó como elemento Físico a la intensidad de señal pues este es un factor que de manera genérica tienen todas las interfaces inalámbricas, además, al ser absoluta, podemos normalizarla.*
- ✦ *Como elemento de hardware elegimos una interfaz de red que fuera compatible con el protocolo 802.11g, pues actualmente es el estándar en Wi-Fi, y además es compatible con el protocolo 802.11b.*
- ✦ *Por elementos de Software tenemos el SO, podemos concluir que Linux presenta una gran facilidad para acceder a la capa de hardware, se puede decir que si se requieren hacer experimentos en lo que se requiera una interacción Usuario-Hardware, Linux sería una buena opción, pues no es necesario tener permisos especiales como los que se requerirían en Windows.*

Acerca de las pruebas, podemos concluir que WiNPS se ejecutó de manera correcta, y con base en las pruebas de funcionalidad que se realizaron para la implantación, queda comprobado que WiNPS **NO** solo es un programa de localización, sino es la unión de diversos módulos, mediante los cuales se puede conocer:

- ✦ *El comportamiento de las variables físicas y los factores de transmisión de la red, ya sea solo de manera numérica o gráfica.*
- ✦ *La caracterización de la intensidad de señal, a modo de conocer un mapa de intensidad de señal.*
- ✦ *Localización de un móvil dentro de una red Wi-Fi.*

## **6.2 Sobre nuestro algoritmo de localización basado en el mapeo.**

Cuando comenzábamos el desarrollo de nuestra propuesta de solución, nos dimos cuenta que hacían falta algunas cosas. Entre ellas, estaba la falta de un algoritmo de localización, en la actualidad existen sistemas comerciales que permiten ubicar una computadora en una red inalámbrica, sin embargo, los costos de estos programas llegan a ser de miles de dólares, es por ello que no existe públicamente un algoritmo de posicionamiento; solo existe el fundamento teórico de algunos métodos.

Pues bien, al analizar el método de mapeo, proponemos un algoritmo original que no esta descrito en algún otra parte. Este método de mapeo es el que describe en la sección 4.4.4.7.

Asimismo resulta importante mencionar que nosotros no quisimos quedarnos solo con el software que realiza la localización, sino hicimos toda una suite de utilerías que permiten analizar el comportamiento de diversas variables, y que además, nos permiten caracterizar un entorno de trabajo.

Durante el capítulo 4, hemos visto en numerosas ocasiones el funcionamiento del algoritmo de localización, ahora bien a modo de conclusión, mostramos la precisión que tiene el sistema, ahora bien, en la tabla 6.1 se muestran las variaciones obtenidas por WiNPS cuando se probó con tres puntos.



<b>Posición Real</b>	<b>Posición Estimada con WiNPS</b>	<b>Variación en X</b>	<b>Variación en Y</b>
(13.5,-21.5)	(13.98,21.71)	0.48 [m]	0.21 [m]
(-1.04, -25.14)	(-0.93, -24.8)	0.11 [m]	0.34 [m]
(3, -5)	(3.39, -4.67)	0.39 [m]	0.33 [m]

*Tabla 6.1 Variaciones de Información general de la prueba para el módulo de mapping.*

Esta tabla, aunque no presenta muchos datos, nos permite comprobar que las variaciones no van más allá del medio metro, esto nos sorprendió, pues nosotros le habíamos dado hasta 2 [m] de tolerancia.

También podemos concluir que el sistema tiene un modo de autorrecuperación, es decir, si no encuentra la posición con base en las condiciones que tiene definidas, automáticamente el comienza a tomar otra medición y rastrear el nuevo punto, esto se pudo comprobar en la fase de pruebas, pues al no encontrar un valor satisfactorio, el procedimiento lo realizo tres veces.

Podemos decir que el algoritmo de localización funciona, pues hace las discriminaciones de las regiones, realiza el cálculo de la intensidad de señal promedio y con ello establece un punto en el espacio.

### **6.3 Mejoras que se le pueden hacer a WiNPS.**

Para finalizar este trabajo de tesis, mencionaremos algunas mejoras que se le pueden realizar a WiNPS, así pues, tenemos:

- ✦ *En el caso del sistema de graficación, se pueden realizar muestreos por tiempos, así podremos observar el comportamiento en función del tiempo y no en función de muestras.*
- ✦ *Para el módulo de monitoreo, podemos generar un archivos de salida con las variables que el usuario decida.*
- ✦ *Continuando con el sistema de monitoreo, se podría establecer que el usuario escogiese el AP a monitorear.*
- ✦ *Para el módulo de ayuda, esta se podría hacer de manera más dinámica, actualmente son archivos de texto con información muy básica.*
- ✦ *En el módulo de localización, la ubicación se podría realizar en tiempo real, a modo de que cuando el usuario avance, el punto de localización también se mueva; para lograr esto se debe ver como se solventaría el problema de estabilización de la intensidad de señal.*

### **6.4 Sobre la caracterización de la red.**

En el capítulo IV realizamos algunas acotaciones que vale la pena incluir en estas conclusiones, pues dichas acotaciones corroboran lo que se dice en la teoría; es decir, observábamos la relación existente entre el medio en el que está inmersa la red, y la variable física de intensidad de señal.



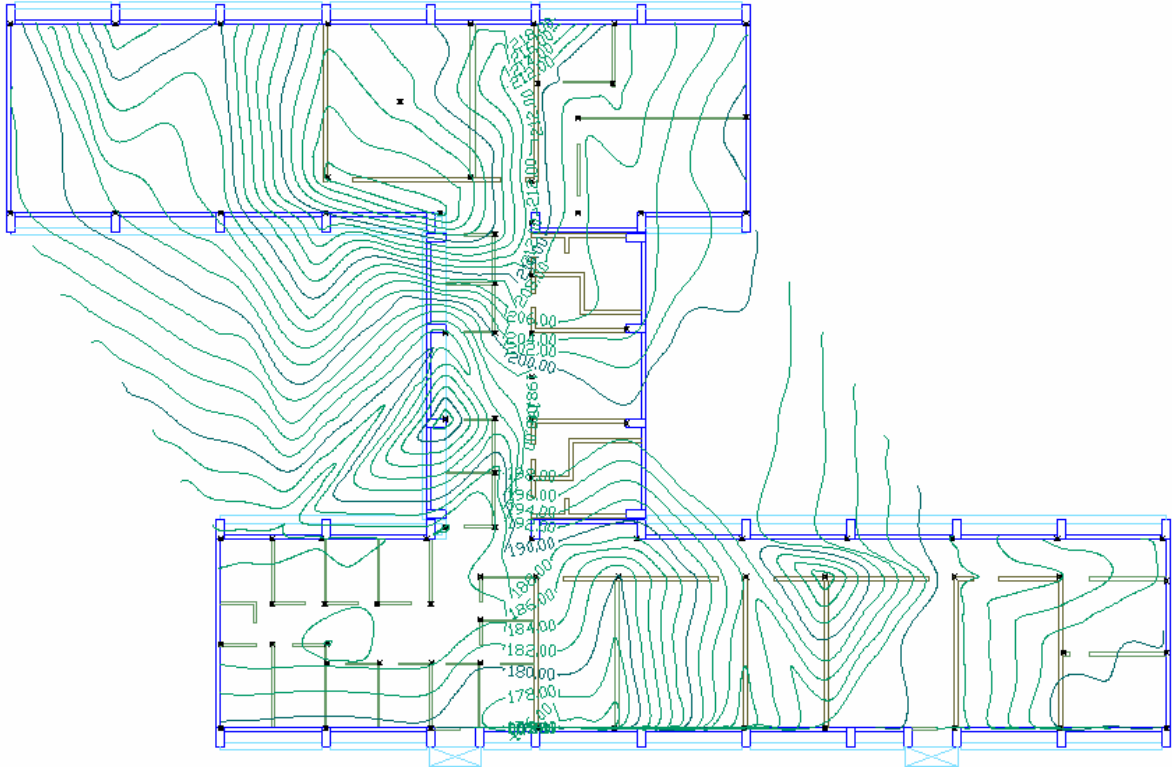


Figura 6.2 Distribución de la Intensidad de Señal normalizada.

Así pues, retomando nuevamente la figura 6.2 de la distribución de intensidades de señal, podemos decir que para el *workspace* del departamento de Telecomunicaciones de la DIE tenemos:

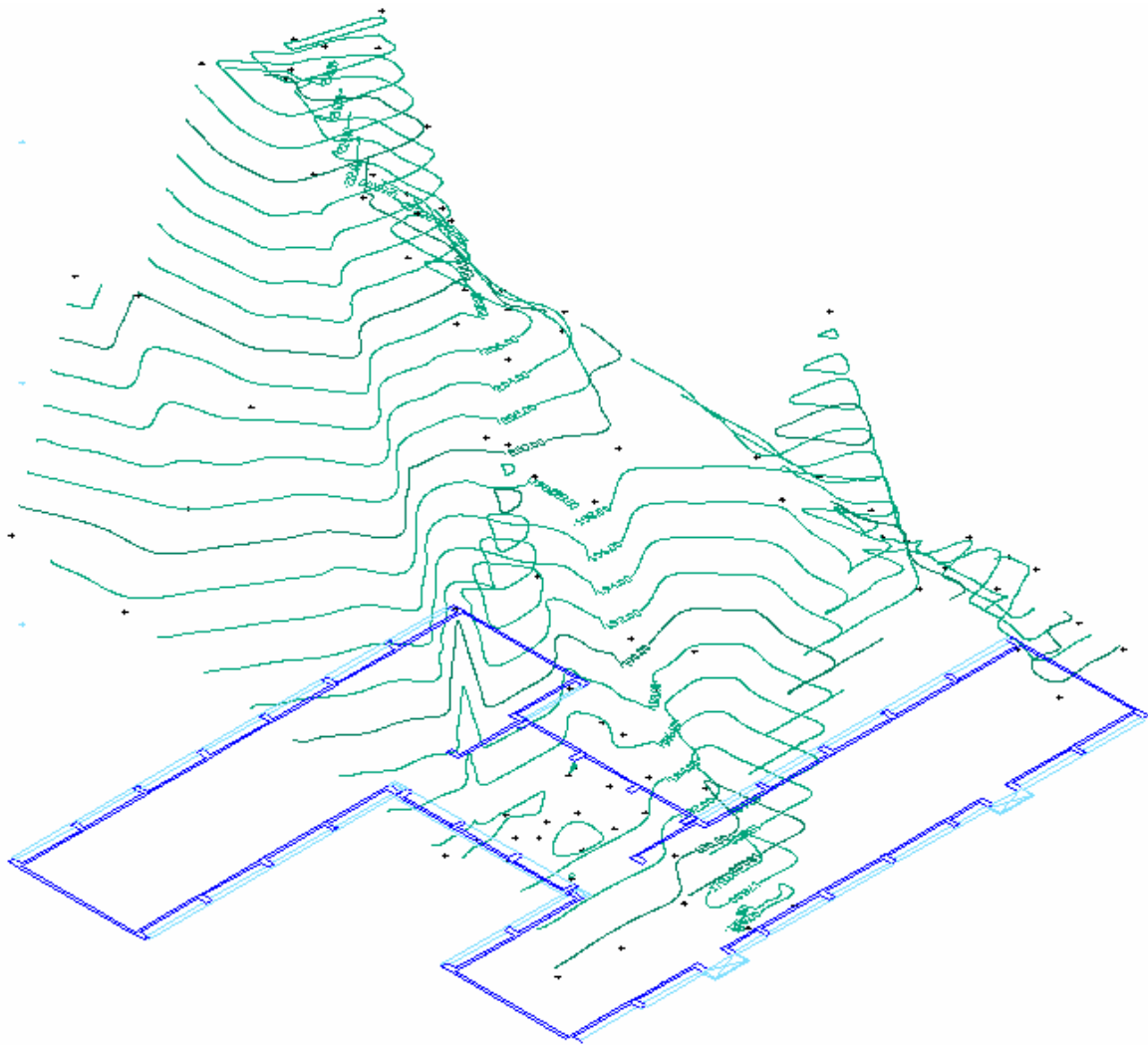
- ⊕ La señal se atenúa en la parte centro-derecha, esto se debe a que en esa parte se encuentran los sanitarios y ahí existen muros de concreto de hasta 15 [cm] de grosor.
- ⊕ En la parte inferior-derecha, podemos ver que la señal crece, esto se debe a los rebotes de la señal en las paredes de concreto, es por ello que el pasillo de esta zona presenta una mayor intensidad de señal.
- ⊕ Se puede ver que en el extremo derecho inferior más lejano las líneas son más dispersas, por lo que la intensidad de señal se va perdiendo, este efecto se debe a la atenuación y la gran distancia existente.
- ⊕ En la zona inferior izquierda, la señal se ha perdido, esto se debe a que se forma una caja con la estructura de concreto y además esta caja tendría, por así llamarlo, un recubrimiento de muros falsos y puertas, esto hace que la atenuación crezca.
- ⊕ Se ve que la intensidad de señal crece en la zona central izquierda, esto se debe a que el AP técnicamente tiene línea de vista con toda esa zona, además ayuda de que en esa parte existan amplios ventanales, así como el rebote de la señal en el pasillo con muros de concreto y paredes falsas.

De igual manera con estos puntos pudimos comprobar:

- ⊕ La atenuación de la señal.

- ⊕ *Existe una mayor intensidad de señal cuando existe línea de vista entre AP y terminal.*
- ⊕ *El "rebote" o efecto de multitrayectoria de la señal coadyuva a tener una mejor señal en lugares angostos.*
- ⊕ *Pudimos ver que la herramienta de monitoreo nos arroja resultados fieles y totalmente apegados a la teoría.*

Asimismo, nos resulta importante mostrar una distribución tridimensional de las intensidades de señal, mostrada la figura 6.3. Básicamente lo que podemos ver aquí es que en donde existen picos, existirá una mayor intensidad de señal, también se puede observar que los puntos más alejados, representados por valles, la intensidad se vuelve casi nula.



*Figura 6.3 Distribución Tridimensional de la Intensidad de Señal normalizada.*

## APÉNDICES

### A. NDISWRAPPER.

Esta no es propiamente una herramienta de monitoreo, podemos decir que es un medio con el cual podemos activar las llamadas *wireless extensions* del OS, hacemos mención de esta herramienta por el hecho de que nos fue de gran utilidad al momento de llevar a cabo el desarrollo de WinNPS.

Así pues, en primera instancia hay que mencionar que para todo OS, llámese Windows o Linux, tenga una completa funcionalidad entre los periféricos, es necesario contar con los drivers de cada uno de ellos. Básicamente un driver es un conjunto de archivos que proporcionan las especificaciones del funcionamiento del dispositivo.

Ahora bien, Linux al ser un sistema de distribución libre, algunas compañías no liberan las especificaciones de su hardware o bien no proveen a Linux drivers para estos dispositivos. Bueno, pues esa es la razón por la que surge NDIS Wrapper, es una herramienta que surge a partir de un proyecto que implementa una parte del kernel de la API de Windows en la API NDIS dentro del kernel de Linux.

El funcionamiento de esta herramienta se basa en el concepto de *driver wrapper*, este es un software que funciona como un adaptador entre el OS y un *driver* que no ha sido diseñado para el propio OS. Así, como su nombre lo indica, la herramienta no hace más que "envolver" la información de los drivers funcionales en Windows, a modo de que puedan ser funcionales totalmente sobre el kernel del propio Linux.

La explicación de cómo funciona esta herramienta sale de los alcances del presente trabajo de tesis, sin embargo, podemos decir que el proceso de "envoltura" no es más que una emulación binaria de la información recibida a nivel de la capa de hardware del propio dispositivo.

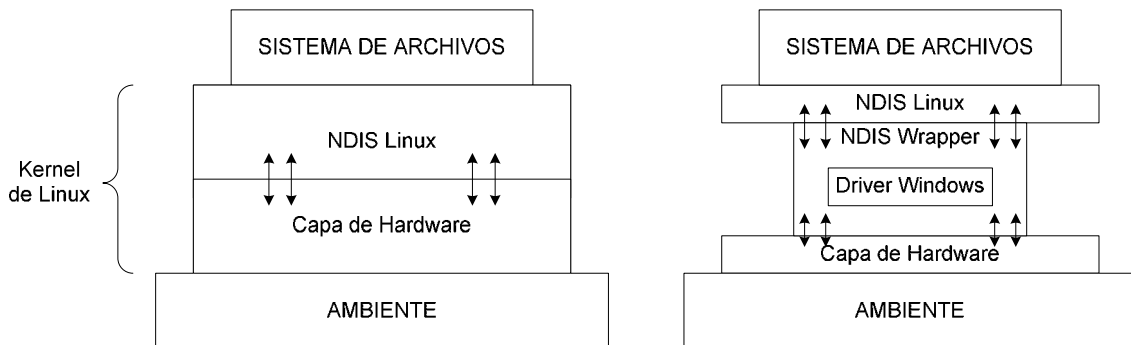


Figura A.1 Funcionamiento de NDIS Wrapper

La figura resulta bastante representativa de lo que se realiza con la herramienta, con el proceso de emulación se mapea la información recibida por la emulación del proceso de la API de Windows para que esta sea recibida por la NDIS de Linux a modo de que el kernel piense que es un proceso nativo de Linux.

## **B. Instalación de la tarjeta PCMCIA Linksys WPC54G version 1.3.**

Características del Sistema: Dell Inspiron 700m

Sistema Operativo: Fedora Core (kernel 2.6.9-1.667)

### *1. Actualización del sistema para que cuente con un stack de 16Kb*

Muchos de los drivers de Windows requieren tamaños de pila mayores o iguales a 12 Kb. El kernel por defecto de Fedora tiene un tamaño de 4 Kb, por lo que necesita ser actualizado. Sin embargo, se comprobó que esta operación no era necesaria para esta versión de la tarjeta, aunque el sistema se volvía inestable en ciertos momentos.

Ahora bien, existen muchas maneras de actualizar el sistema, la opción más sencilla es bajar la actualización apropiada del kernel. Para esto se debe de conocer la arquitectura y la versión del kernel, con los siguientes comandos:

```
[root@mobilelan ~]# uname -r
```

En nuestro caso se obtuvo lo siguiente  
2.6.9-1.667

```
[root@mobilelan ~]# rpm -q --qf '%{ARCH}\n' kernel
```

la ejecución arrojo o siguiente:  
i686

Con estos datos del sistema se tiene que bajar el RPM adecuado:

Para i586

<http://www.linuxant.com/driverloader/wlan/full/downloads-fc3-kernel-i586.php>

Para i686

<http://www.linuxant.com/driverloader/wlan/full/downloads-fc3-kernel-i686.php>

Una vez bajado el archivo, se procede a instalar los archivos, con la instrucción `rpm -Uvh nom_arch`; para nuestro caso, se obtuvo lo siguiente:

```
[root@mobilelan ~]# rpm -Uvh kernel-2.6.9-1.667.stk16.i686.rpm
Preparing...          ##### [100%]
 1:kernel             ##### [100%]
```

### *2. Se procede a la instalación de NDISWRAPPER*

```
[root@mobilelan ~]# tar xvzf ndiswrapper-1.1.tar.gz
[root@mobilelan ~]# cd ndiswrapper-1.1
[root@mobilelan ndiswrapper-1.1]# make
[root@mobilelan ndiswrapper-1.1]# make install
```

3. Se procede a la configuración de NDISWRAPPER, instalando el archivo de información del driver para Windows de la tarjeta Linksys (lsbcmnds.inf) el cual viene en el CD de instalación de la tarjeta.

```
[root@mobilelan ndiswrapper-1.1]# ndiswrapper
Usage: ndiswrapper OPTION

Manage ndis drivers for ndiswrapper.
-i inffile          Install driver described by 'inffile'
-d devid driver    Use installed 'driver' for 'devid'
-e driver          Remove 'driver'
-l                List installed drivers
-m                Write configuration for modprobe
-hotplug          (Re)Generate hotplug information
[root@mobilelan ~]# ndiswrapper -e lsbcmnds
[root@mobilelan ~]# ndiswrapper -i wpc54/lsbcmnds.inf
Installing lsbcmnds
[root@mobilelan ~]# ndiswrapper -l
Installed ndis drivers:
lsbcmnds          driver present, hardware present
```

4. Se actualizan los servicios de red modificando los parámetros y se da de alta la interfaz de red.

```
[root@mobilelan network-scripts]# cd /etc/init.d
[root@mobilelan init.d]# gedit pcmcia
////Aquí la línea deberá quedar
PCMCIA
# chkconfig: 2345 09 96

[root@mobilelan init.d]# gedit network
////Aquí la línea deberá quedar
NETWORK
# chkconfig: 2345 22 96

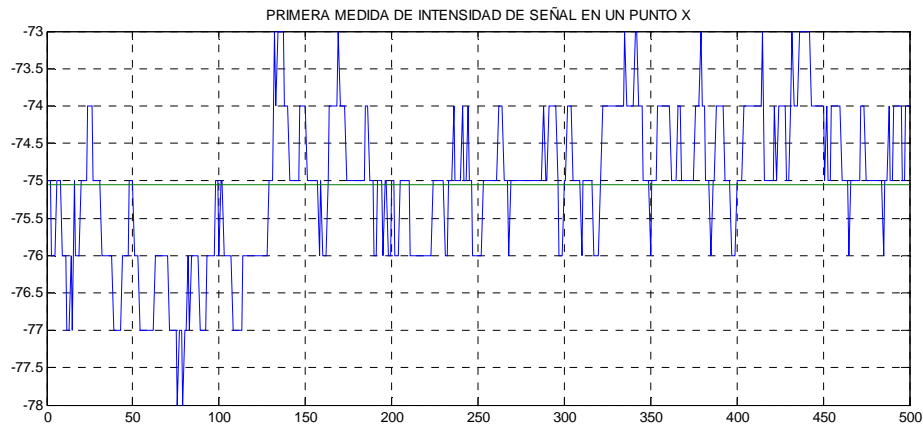
[root@mobilelan init.d]# gedit nfslock
////Aquí la línea deberá quedar
NFSLOCK
# chkconfig: 345 22 86

[root@mobilelan init.d]# cardctl status
Socket 0:
  3.3V CardBus card
  function 0: [ready]
Socket 1:
  no card
[root@mobilelan init.d]# modprobe ndiswrapper
[root@mobilelan init.d]# modprobe -l ndiswrapper
/lib/modules/2.6.9-1.667.stkl6/misc/ndiswrapper.ko
[root@mobilelan init.d]# ndiswrapper -m
Adding "alias wlan0 ndiswrapper" to /etc/modprobe.conf
[root@mobilelan init.d]#
```

### **C. El intervalo de estabilización.**

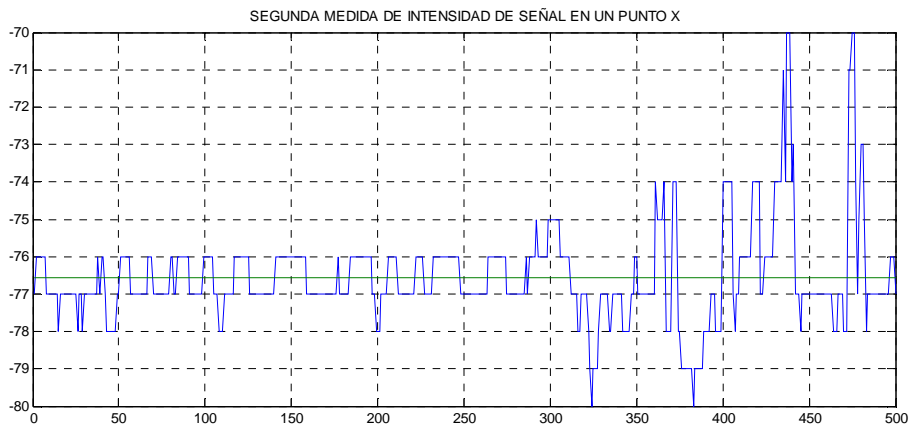
Cuando comenzábamos con el desarrollo del proceso de mapeo, llegamos a pensar que la intensidad de señal era algo constante, o por lo menos bastante estable, sin embargo, esta premisa fue del todo acertada.

En la figura C.1 encontramos una gráfica de intensidad de señal de 500 muestras tomadas en el mismo punto, como se puede ver, no existe un valor estable que sea definitivamente característico para el comportamiento de la intensidad de señal.



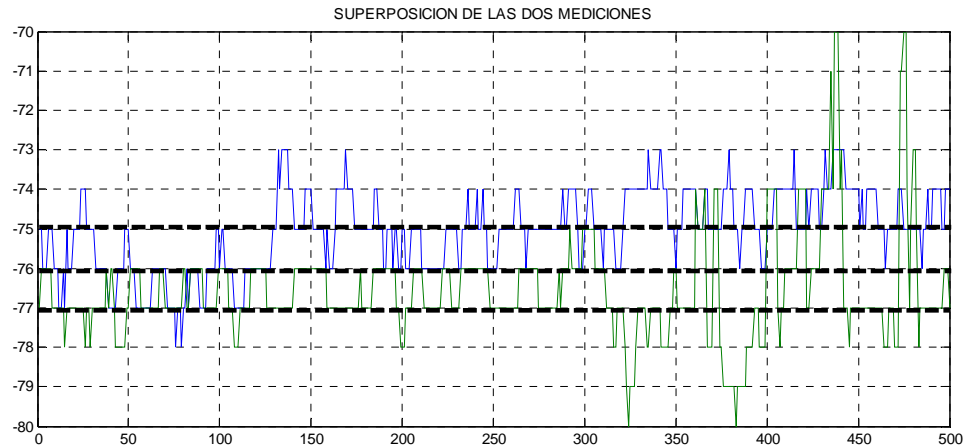
*Figura C.1 Medición de la intensidad de señal en un mismo punto (DIA 1).*

Sin embargo, en ese mismo punto al día siguiente hicimos el ejercicio, y como era de esperarse, la señal salió todavía más diferente a la de la figura 6.1; esto se puede corroborar en la figura C.2.



*Figura C.2 Medición de la intensidad de señal en un mismo punto (DIA 2).*

De las dos figuras ponemos observar algo curioso, al principio de los muestreos, los valores presentan gran variación posteriormente comienzan a estabilizarse. Si ahora superponemos ambas figuras, obtenemos el diagrama C.3.



*Figura C.3 Superposición de las dos mediciones.*

De esta figura podemos notar que la mayoría de los puntos caen en un rango, este rango es el mismo sin importar cual de las señales se trate, con esto se puede aproximar un valor característico de la medición.

Ahora bien, con todo esto podemos concluir dos cosas:

- ⊕ *La intensidad de señal puede estabilizarse si se le da un tiempo de tolerancia antes de comenzar las mediciones.*
- ⊕ *Se puede obtener un valor característico de intensidad de señal, el cual sea representativo del comportamiento de dicha variable física.*

Estas dos conclusiones las aplicamos al momento de desarrollar las herramientas de mapeo y localización, pues primero esperamos un tiempo  $n$ , lo que nos garantiza que la señal tendrá un menor número de variaciones, asimismo, se puede ver que con una función promedio es más que suficiente para obtener un valor característico, quizás no es el mejor, pero si es el más fácil de implementar.

## **ACRÓNIMOS Y ABREVIATURAS**

---

ABEND: Abnormal ENDing.

ACK: Acknowledgement.

AES: Advanced Encryption Standard.

A-FLT: Advanced Forward Link Trilateration.

AOA: Angle of Arrival.

API: Application Programming Interface.

BDE: Borland Database Engine.

BLN: Bluetooth Location Network.

BSA: Basic Service Area.

BSS: Basic Service Set.

BTS: Base Transceiver Station.

CA: Collision Avoidance.

CCK: Complementary Code Keying.

CDMA: Code Division Multiple Access.

CLX: Cross Platform Component Library.

CRC: Cyclic Redundancy Check.

CSMA/CA: Carrier Sense Multiple Access/Collision Avoidance.

CSMA/CD: Carrier Sense Multiple Access with Collision Detection.

CTS: Clear To Send.

DCF: Distributed Coordination Function.

DHSS: Direct Sequence Spread Spectrum.

DIFS: DCF Interframe Spacing.

DIFS: Distributed Inter Frame Space.

DLB: Geometrical Distance from LMU to BTS.

DMB: Geometrical Distance from MS to BTS.



DOA: Direction of Arrival.

DS: Distribution System.

DSSS: Direct-Sequence Spread Spectrum.

DTBS: Distributed Time Bounded Services.

E-OTD: Enhanced Observed Time Difference.

E.P.E: Ekahau Positioning Engine.

E911: Wireless Enhanced 911.

ESS: Extended Service Set.

ESSID: Extended Service Set Identifier.

ETSI: European Telecommunications Standards Institute.

FCC: Federal Communications Commission.

FDM: Frequency Division Multiplexing.

FHSS: Frequency Hopped Spread Spectrum.

GDOP: Geometric Dilution of Precision.

GIMP: GNU Image Manipulation Program.

GNOME: GNU Network Object Model Environment.

GNU: GNU is Not Unix.

GPL: General Public License.

GPRS: General Packet Radio Service.

GPS: Global Positioning System.

GSM: Global System for Mobile Communications.

GTD: Geometric Time Difference.

GUI: Graphic User Interface.

HDTV: High Definition Television.

HR-DSSS: High Rate DSSS.

IAPP: Inter-Access Point Protocol.

IBSS: Independent Basic Service Set.

IP: Internet Protocol.

IR: Infrared.

IrDA: Infrared Data Association.

ISM: Industry, Science and Technology.

KDE: K Desktop Environment o Common Desktop Environment.

LAC: Location Area Code.

LAI: Location Area Identity.

LGPL: Lesser General Public License.

LMU: Location Measurement Unit.

LOS: Line Of Sight.

LOT: Observed Time at the LMU.

MAC: Media Access Control.

MCC: Mobile Country Code.

MF: Huella Multirtrayecto.

MNC: Mobile Network Code.

MOT: Observed Time at the MS.

MPS: Mobile Positioning System.

MS: Mobile Station.

NDIS: Network Driver Interface Specification.

NIC: Network Interface Card.

NWID: Network ID.

OFDM: Orthogonal Frequency Division Multiplexing.

OTD: Observed Time Difference.

OS: Operative System.

PAN: Personal Area Network.

PC: Personal Computer.

PCF: Position Computation Function.

PCF: Puntual Coordination Function.

PDA: Personal Digital Assistant.

PDE: Position Determining Entity.  
PDT: Portable Data Terminal.  
PHY: Physical Layer.  
PLCP: Physical Layer Convergence Procedure.  
PLMN: Public Land Mobile Network.  
PMD: Physical Medium Dependent.  
PRC: Pseudo Random Code.  
PSK: Phase Shift Keying.  
QAM: Quadrature Amplitud Modulation.  
QoS: Quality of Service.  
RAD: Rapid Application Development.  
RF: Radio Frequency.  
ROX: RISC OS on X.  
RSL: Received Signal Level.  
RTD: Real Time Difference.  
RTS: Request To Send.  
SA: Selective Availability.  
SSID: Service Set Identifier.  
TA: Timing Advance.  
TCP/IP: Transmission Control Protocol/Internet Protocol.  
TDMA: Time Division Multiple Access.  
TDOA: Time Diference of Arrival.  
TOA: Time of Arrival.  
TIM: Traffic Indication Message.  
UMTS: Universal Mobile Telecommunications System.  
UNII: Unlicensed National Information Infrastructure.  
USB: Universal Serial Bus.  
UWB: Ultra Wide Band.

VCL: Visual Component Library.

VoIP: Voice over Internet Protocol.

WB: Wireless Bridge.

WEP: Wired Equivalent Privacy.

Wi-Fi: Wireless Fidelity.

WLAN: Wireless Local Area Network.

WiNPS: Wireless Network Position System.

WPAN: Wireless Personal Area Network.

WWAN: Wireless Wide Area Network.

XFCE: XForms Common Environment.

## **REFERENCIAS Y BIBLIOGRAFÍA**

---

**Estudio y análisis del desempeño de redes WLAN IEEE 802.11**

*Cervantes Luis, Meneses Sigrid*  
UNAM, 2005

**WI-FI, Análisis, diagnóstico y políticas públicas**

*Redondo Juan, Pérez Jorge, Ruano Vanesa, et Al.*  
Grupo de análisis y prospectiva del sector de las telecomunicaciones.

**Arquitectura con redes Wi-Fi**

*Cimadevila López Enrique*  
Dealer World, Gíreles Mundi

**Wireless Community Networks**

*Jain Saurabh; Agrawal Dharm*

**Wireless LAN Networking**

*White paper*  
U.S. Robotics

**Redes inalámbricas y Móviles avanzadas. IEEE 802.11 LAN Inalámbricas**

*Gómez Castellanos Javier*  
2003

**Investigation of the IEEE 802.11 medium access control sublayer funtions**

*Crow B.P, Widjaja I, Kim J.G, et Al.*  
INFOCOM

**Cordless Personal Communications**

*WalterH. W. Tuttlebee*  
Digital Cordless Technology, IEEE

**IEEE 802.1 system design**

*Prasad, N.R.*  
Pesonal Wireless Communications, IEEE

**Protocols and architectures for wireless networks**

*Karl Holger, Willig Andreas*  
Wiley, EUA, 2005

**The handbook of ad hoc wireless networks**

*Mohammad, Ilyas*  
Florida Atlantic University Boca Ratón, EUA, 2003

**Wireless LAN Systems**

*A. Santamaría, F.J. López-Hernández*  
Artech House, 1994

**Wireless Information Networks**

*Pahlavan Kaveh, Levesque Allen*  
Wiley series in telecommunications and signal processing, EUA, 1995

**Wireless networks**

*P. Nicopolitidis, M.S. Obaidat et Al.*  
John Wiley and Sons, England, 2003

**Redes de comunicación**

*León-García Alberto, Widjaja Indra*  
Mc Graw Hill, España, 2002

**Redes inalámbricas**

*Alejandro Reguarti Ferri, Carmen Pérez Berenguer*  
México, 2004

**Wi Fi**

*Caballar José A.*  
Alfaomega, México, 2005

**Wireless Extensions for Linux**

*Tourrilhes Jean*  
Linux wireless LAN Howto, Francia, 1997

**Wireless Location Technologies**

[http://www.unstrung.com/document.asp?doc\\_id=15069&page\\_number=2](http://www.unstrung.com/document.asp?doc_id=15069&page_number=2)

**Introduction to mobile and wireless computing**

<http://www.eecs.uc.edu/~dpa/courses/winter2003/locationproj.htm>

**Radio localization of mobile terminals in wireless networks**

<http://www.elet.polimi.it/dsp/tlc/position.htm>

**Mobile Positioning & Location-based services**

<http://www.surujdutta.com/technology.htm>

**Positioning a mobile station**

<http://www.willassen.no/msl/node6.html>

### **E-OTD & E-OTD-C**

<http://emc2.ucd.ie/deliverables/e=mc2.2.1.1.2002.doc>

### **EOTD, AOA, TOA & GPS**

<http://www.tid.es/premios/evolucion.pdf>

<http://marconi.ltt.dii.unisi.it/~giambene/ist2001.pdf>

<http://www.idi.ntnu.no/grupper/su/ordypningsprosjekt-2003/ordypning2003-walle.pdf>

### **OTD**

<http://www.ftw.at/ftw/events/telekommunikationsforum/WS2001/WS01docs/15>

### **GSM**

[http://www.portalgsm.com/documentacion\\_extendida/98\\_0\\_17\\_0\\_C](http://www.portalgsm.com/documentacion_extendida/98_0_17_0_C)

<http://www.wmlclub.com/articulos/fundamentosgsm.htm>

### **Sistemas móviles**

[http://www.une.edu.ve/~iramirez/te1/sistemas\\_moviles.htm](http://www.une.edu.ve/~iramirez/te1/sistemas_moviles.htm)

### **Glosario de tecnologías inalámbricas**

<http://www.voice-technologies.com/vt-voicetech-mai.html>

### **Redes de datos**

<http://www.airtel.es/Vodafone/Campus/Campus/0,1463,2949,00.html>

### **Wireless Tools for Linux**

[http://www.hpl.hp.com/personal/Jean\\_Tourrilhes/Linux/Tools.html](http://www.hpl.hp.com/personal/Jean_Tourrilhes/Linux/Tools.html)

### **Wireless LAN resources for Linux**

[http://www.hpl.hp.com/personal/Jean\\_Tourrilhes/Linux/Wireless.html](http://www.hpl.hp.com/personal/Jean_Tourrilhes/Linux/Wireless.html)

### **iwlist, ifconfig, iwconfig**

*Man pages for Fedora Core 3*

### **Aprendiendo Delphi 3 en 14 días.**

*Dan Osier, Steve Grobman y Steve Batson,*

*Prentice Hall Hispanoamericana*

*México 1997.*

### **Componentes y Manuales de Kylix y Delphi.**

<http://www.casadelphi.com/>

### **Sobre las características principales de Kylix**

<http://www.drbob42.com/kylix/>

<http://www.borland.com>

### **NDISWrapper**

<http://ndiswrapper.sourceforge.net/joomla/>

<http://es.wikipedia.org/wiki/NdisWrapper>

Instalación de una la interfaz de red con NDISWrapper

<http://sourceforge.net/projects/ndiswrapper/>

<http://www.ubuntu-es.org/node/2895>

<http://www.mrandersonmd.com/14/03/2007/instalando-el-adaptador-linksys-wpc54gs-en-ubuntu/>