



REPÚBLICA DEL ECUADOR  
**Escuela Politécnica Nacional**  
" E SCIENTIA HOMINIS SALUS "

La versión digital de esta tesis está protegida por la Ley de Derechos de Autor del Ecuador.

Los derechos de autor han sido entregados a la "ESCUELA POLITÉCNICA NACIONAL" bajo el libre consentimiento del (los) autor(es).

Al consultar esta tesis deberá acatar con las disposiciones de la Ley y las siguientes condiciones de uso:

- Cualquier uso que haga de estos documentos o imágenes deben ser sólo para efectos de investigación o estudio académico, y usted no puede ponerlos a disposición de otra persona.
- Usted deberá reconocer el derecho del autor a ser identificado y citado como el autor de esta tesis.
- No se podrá obtener ningún beneficio comercial y las obras derivadas tienen que estar bajo los mismos términos de licencia que el trabajo original.

El Libre Acceso a la información, promueve el reconocimiento de la originalidad de las ideas de los demás, respetando las normas de presentación y de citación de autores con el fin de no incurrir en actos ilegítimos de copiar y hacer pasar como propias las creaciones de terceras personas.

***Respeto hacia sí mismo y hacia los demás.***

# **ESCUELA POLITÉCNICA NACIONAL**

## **FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA**

**DISEÑO DE LA RED INALÁMBRICA DEL CENTRO COMERCIAL  
ESPIRAL Y DESARROLLO DE UNA APLICACIÓN EN JAVA PARA  
UNA TABLET QUE PERMITA LA LOCALIZACIÓN DE OFERTAS Y  
PRODUCTOS DENTRO DEL MISMO CON LA IMPLEMENTACIÓN  
DE UN PROTOTIPO DE PRUEBA**

**PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN  
ELECTRÓNICA Y REDES DE INFORMACIÓN**

**DARWIN ANTONIO MENA ALVARADO**  
darwinbrin@hotmail.com

**ROBERTO RAMIRO JACHO BRAVO**  
robert\_j68@hotmail.com

**DIRECTORA: MSc. SORAYA LUCÍA SINCHE MAITA**  
soraya.sinche@epn.edu.ec

**Quito, Septiembre 2014**

## **DECLARACIÓN**

Nosotros, Darwin Mena - Roberto Jacho, declaramos bajo juramento que el trabajo aquí descrito es de nuestra autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que hemos consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedemos nuestros derechos de propiedad intelectual correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

Darwin Mena

Roberto Jacho

## **CERTIFICACIÓN**

Certifico que el presente trabajo fue desarrollado por Darwin Mena y Roberto Jacho, bajo mi supervisión.

MSc. Soraya Sinche  
DIRECTORA DEL PROYECTO



## **AGRADECIMIENTO**

No hay mayor satisfacción que concluir con tus objetivos a corto y largo plazo a pesar que existan adversidades en el camino. Agradezco a mi mamá, quien solamente con sus palabras, su aliento, su amor puro y sincero, sabe levantarme hasta en los momentos más difíciles, gracias por estar siempre ahí. A mi papá, mi hermano, tíos, primos y amigos por su constante apoyo y soporte.

Roberto

## **AGRADECIMIENTO**

Agradezco principalmente a Dios por permitirme cumplir uno de mis anhelos y llenarme de fortaleza en los momentos más difíciles.

A mis padres, luz y sendero de mi vida, por enseñarme que todo es posible y por su apoyo incondicional sobre todas las cosas, pues con su humildad y sencillez me han dado todo.

A Michelle, infinitas gracias mi amor por tú apoyo incondicional, por estar conmigo en las buenas y malas; motivo, luz y sueño para seguir adelante.

Un agradecimiento muy especial a nuestra tutora MSc. Soraya Sinche por su gentil y valiosa colaboración en el desarrollo del presente proyecto.

Darwin

## **DEDICATORIA**

*Dedico este trabajo a mis padres, Ana y Ramiro con mucho cariño y amor, quienes son los pilares fundamentales en mi vida. Sin su ayuda y soporte no hubiese podido conseguir culminar con este proyecto.*

*A toda mi familia que ha estado aportando con palabras de aliento en los buenos y malos momentos durante toda mi vida.*

Roberto

## **DEDICATORIA**

*Dedicado a mis padres el pilar fundamental de mi vida, pues me han enseñado  
que con esfuerzo y dedicación si se puede salir adelante.*

*A mi sobrino Martin quien lleno de magia e inocencia ilumina nuestro hogar  
día con día.*

*A mi pedacito de cielo en la tierra el amor de mi vida Michelle, reflejo de luz  
quien ilumina mi vida y mi corazón  
Te Amo con toda mi alma, mi amor.*

Darwin

## CONTENIDO

### CAPÍTULO 1

<b>1. FUNDAMENTOS TEÓRICOS.....</b>	<b>1</b>
1.1 INTRODUCCIÓN .....	1
1.2 REDES INALÁMBRICAS 802.11 .....	1
1.2.1 COMPONENTES DE UNA RED IEEE 802.11 .....	2
1.2.2 MODOS DE OPERACIÓN.....	3
1.2.3 SERVICIOS DE IEEE 802.11 .....	4
1.2.4 ARQUITECTURA IEEE 802.11 .....	5
1.2.4.1 Capa Física .....	5
1.2.4.2 Capa Enlace de Datos .....	8
1.2.5 SEGURIDAD EN REDES IEEE 802.11 .....	8
1.2.5.1 Privacidad Equivalente a Cableado (WEP) .....	10
1.2.5.2 Acceso Inalámbrico Protegido (WPA).....	12
1.2.5.3 Acceso Inalámbrico Protegido-2 (WPA2).....	12
1.2.6 VENTAJAS Y DESVENTAJAS GENERALES DE IEEE 802.11 FRENTE A UNA RED CABLEADA .....	13
1.2.7 CONSIDERACIONES PARA DISEÑAR UNA RED INALÁMBRICA IEEE 802.11 .....	14
1.3 LENGUAJE DE PROGRAMACIÓN JAVA.....	16
1.3.1 CARACTERÍSTICAS DEL LENGUAJE JAVA.....	16
1.3.1.1 Independiente de la Plataforma .....	16
1.3.1.2 Distribuido .....	17
1.3.1.3 Orientado a Objetos .....	17
1.3.2 ARQUITECTURA DE JAVA.....	17
1.3.2.1 Máquina Virtual de Java.....	18
1.3.2.2 Interfaz de Programación de Aplicaciones (API) .....	19
1.3.3 PLATAFORMA JAVA .....	19
1.4 SISTEMA OPERATIVO ANDROID .....	21
1.4.1 ARQUITECTURA ANDROID .....	21
1.4.2 VERSIONES DISPONIBLES DE ANDROID.....	27
1.5 HERRAMIENTAS PARA EL DESARROLLO .....	29
1.6 EMULADOR DE TABLET.....	31
1.7 FUNCIONALIDAD DE UNA TABLET .....	32

### CAPÍTULO 2

<b>2. DISEÑO DE LA RED INALÁMBRICA DEL CENTRO COMERCIAL     ESPIRAL.....</b>	<b>34</b>
2.1 SITUACIÓN ACTUAL DEL CENTRO COMERCIAL ESPIRAL .....	34
2.1.1 ANTECEDENTES .....	34
2.1.2 NECESIDADES DEL CLIENTE .....	40
2.2 CLASIFICACIÓN DE LOS USUARIOS .....	47
2.2.1 USUARIOS FIJOS.....	47
2.2.2 VISITANTES.....	47

2.3 ANÁLISIS DE TRÁFICO.....	49
2.3.1 OBTENCIÓN DE DATOS PARA CONOCER LAS HORAS PICO.....	50
2.3.2 CÁLCULO DE LA CAPACIDAD CONSUMIDA POR CADA APLICACIÓN PARA EL ENLACE DE INTERNET.....	51
2.3.2.1 Video <i>Streaming</i> y Youtube .....	51
2.3.2.2 Redes Sociales .....	52
2.3.2.3 Descargas .....	53
2.3.2.4 Correo Electrónico.....	53
2.3.2.5 Navegación Web.....	54
2.3.2.6 Aplicaciones del Centro Comercial .....	54
2.4 SITE SURVEY.....	56
2.4.1 SITE SURVEY PASIVO.....	57
2.4.2 INTERFERENCIA.....	64
2.5 CARACTERÍSTICAS DE LOS EQUIPOS QUE FORMAN PARTE DE LA RED INALÁMBRICA .....	66
2.6. SURVEY ACTIVO .....	68
2.7 CABLEADO ESTRUCTURADO .....	79
2.8 DIRECCIONAMIENTO .....	85
2.8.1 VLANS A CREARSE .....	85
2.8.2 DIRECCIONAMIENTO IP.....	86
2.9 ESPECIFICACIONES DE LOS EQUIPOS .....	88
2.9.1 ELECCIÓN DEL PUNTO DE ACCESO .....	88
2.9.2 ELECCIÓN DEL SWITCH DE ACCESO.....	91
2.9.3 CONTROLADOR DE RED INALÁMBRICA .....	93
2.9.4 DIMENSIONAMIENTO DEL SERVIDOR DE BASE DE DATOS.....	96

## CAPÍTULO 3

<b>3. DESARROLLO DE LA APLICACIÓN .....</b>	<b>99</b>
3.1 REQUERIMIENTOS DE LA APLICACIÓN.....	99
3.2 DIAGRAMAS UML .....	99
3.2.1 DIAGRAMA DE CASOS DE USO.....	100
3.2.2 DIAGRAMA DE CLASES .....	100
3.2.3 DIAGRAMA DE ACTIVIDADES.....	102
3.3 ESTRUCTURA DE LAS CARPETAS .....	103
3.4 JSON ( <i>JAVA SCRIPT OBJECT NOTATION</i> ).....	104
3.4.1 NOTACIÓN JSON FORMA OBJECT/CLASE .....	106
3.5 MANEJO DE DATOS EN ANDROID.....	106
3.5.1 SERVIDOR XAMPP .....	106
3.5.3 MANEJO DE DATOS - SERVICIOS WEB .....	107
3.5.3.1 Archivos PHP .....	107
3.5.3.2 Fichero “ <i>DB_CONFIG.PHP</i> ” .....	108
3.5.3.3 Fichero “ <i>DB_CONNECT.PHP</i> ”.....	109
3.5.3.4 Fichero “ <i>GET_LOCAL_TIPO.PHP</i> ”.....	111
3.5.3.5 Fichero “ <i>GET_LOCAL.PHP</i> ” .....	113
3.5.3.6 Fichero “ <i>GET_PRODUCTO_TIPO.PHP</i> ”.....	114
3.5.3.7 Fichero “ <i>GET_OFERTA_TIPO.PHP</i> ” .....	114

3.5.3.8 Fichero "GET_PRODUCTO_DETALLES.PHP" .....	115
3.5.3.9 Fichero "GET_OFERTA_DETALLES.PHP" .....	115
3.5.3.10 Fichero "GET_LAPTOPS.PHP" .....	115
3.5.3.11 Fichero "GET_TABLETS.PHP" .....	117
3.5.3.12 Fichero "GET_IMPRESORAS.PHP" .....	117
3.5.3.13 Fichero "GET_OTROSTECNOLOGIA.PHP" .....	117
3.5.3.14 Fichero "GET_CLIENTES.PHP" .....	118
3.5.3.15 Fichero "UPDATE_CLIENTES.PHP" .....	119
<b>3.6 ACTIVIDADES IMPLEMENTADAS EN LA APLICACIÓN .....</b>	<b>122</b>
3.6.1 ACTIVIDADES ANDROID .....	122
3.6.2 CLASE PRINCIPAL .....	123
3.6.3 INTERFAZ "ACTIVITY_MAIN.XML" .....	127
3.6.4 CLASE "JSON_PARSER" .....	128
3.6.5 CLASE "LOCALES ACTIVITY" .....	131
3.6.6 CLASE "BÚSQUEDA PRODUCTOS ACTIVITY" .....	138
3.6.7 CLASE "OFERTAS ACTIVITY" .....	138
3.6.8 CLASE "LOCAL DETALLES ACTIVITY" .....	139
3.6.9 CLASE "OFERTAS LOCAL ACTIVITY" .....	149
3.6.10 CLASE "PRODUCTO DETALLES ACTIVITY" .....	153
3.6.11 CLASE "OFERTA DETALLES ACTIVITY" .....	157
3.6.12 CLASE SUBSUELO uno ACTIVITY .....	158
3.6.13 CLASES QUE MANEJAN EL MAPA DEL SITIO DEL CENTRO COMERCIAL .....	161
3.6.14 CLASE "BUZON ACTIVITY" .....	166
3.6.15 CLASE "MAPA ACTIVITY" .....	167

## **CAPÍTULO 4**

### **4. IMPLEMENTACIÓN DEL PROTOTIPO DE PRUEBA Y PRESENTACIÓN**

<b>DEL COSTO REFERENCIAL DE LA SOLUCIÓN .....</b>	<b>172</b>
4.1 PRESENTACIÓN GENERAL DEL PROTOTIPO DE PRUEBA .....	172
4.2 ANÁLISIS DE RESULTADOS DE LA APLICACIÓN .....	173
DESARROLLADA .....	173
4.3 COSTO DE LA RED PASIVA.....	179
4.4 COSTO DE LA RED ACTIVA.....	180
4.4.1 ALTERNATIVA HP .....	180
4.4.2 ALTERNATIVA CISCO.....	181
4.4.3 SELECCIÓN DE LOS EQUIPOS DE CONECTIVIDAD .....	182
4.4.4 SELECCIÓN DEL SERVIDOR PARA LA BASE DE DATOS .....	182
4.5 COSTOS DE ENLACE DE INTERNET .....	182
4.6 COSTO DE LA APLICACIÓN.....	183
4.7 COSTO TOTAL DEL PROYECTO .....	184

## **CAPÍTULO 5**

<b>5.CONCLUSIONES Y RECOMENDACIONES .....</b>	<b>185</b>
---	------------

5.1 CONCLUSIONES.....	185
5.2 RECOMENDACIONES .....	187
<b>REFERENCIAS .....</b>	<b>189</b>
<b>ANEXOS</b>	
<b>ANEXO A: MANUAL DE USUARIO .....</b>	<b>A-1</b>
<b>ANEXO B: CÓDIGO .....</b>	<b>B-1</b>
<b>ANEXO C: GRÁFICAS DE TRÁFICO OBTENIDO EN 20 LOCALES COMERCIALES .....</b>	<b>C-1</b>
<b>ANEXO D: MODELO DE ENCUESTAS REALIZADAS EN EL CENTRO COMERCIAL ESPIRAL .....</b>	<b>D-1</b>



## ÍNDICE DE FIGURAS

### CAPÍTULO 1

<b>Figura 1.1:</b> Topologías Lógicas Inalámbricas .....	3
<b>Figura 1.2:</b> Modelo de Capas IEEE 802.11 .....	6
<b>Figura 1.3:</b> Funcionamiento del protocolo de seguridad WEP.....	11
<b>Figura 1.4:</b> Virtualización de Java .....	17
<b>Figura 1.5:</b> Esquema de la Arquitectura Java.....	18
<b>Figura 1.6:</b> Máquina Virtual de Java.....	19
<b>Figura 1.7:</b> Arquitectura Android en Capas .....	22
<b>Figura 1.8:</b> Componentes básicos de una Aplicación.....	25
<b>Figura 1.9:</b> <i>SDK Manager</i> .....	30
<b>Figura 1.10:</b> Emulador para Android 2.2 .....	32

### CAPÍTULO 2

<b>Figura 2.1:</b> Esquema de la red Inalámbrica para el Centro Comercial El Espiral	41
<b>Figura 2.2:</b> Resultados de la Pregunta 1 – Encuesta Visitantes.....	43
<b>Figura 2.3:</b> Resultados de la Pregunta 2 – Encuesta Visitantes.....	43
<b>Figura 2.4:</b> Resultados de la Pregunta 3 – Encuesta Visitantes.....	43
<b>Figura 2.5:</b> Resultados de la Pregunta 4 – Encuesta Visitantes.....	43
<b>Figura 2.6:</b> Resultados de la Pregunta 5 – Encuesta Visitantes.....	43
<b>Figura 2.7:</b> Resultados de la Pregunta 6 – Encuesta Visitantes.....	43
<b>Figura 2.8:</b> Resultados de la Pregunta 7 – Encuesta Visitantes.....	44
<b>Figura 2.9:</b> Resultados de la Pregunta 8 – Encuesta Visitantes.....	44
<b>Figura 2.10:</b> Resultados de la Pregunta 9 – Encuesta Visitantes.....	44
<b>Figura 2.11:</b> Resultados de la Pregunta 10 – Encuesta Visitantes.....	44
<b>Figura 2.12:</b> Resultados de la Pregunta 11 – Encuesta Visitantes.....	44
<b>Figura 2.13:</b> Resultados de la Pregunta 1 – Encuesta Locales.....	46
<b>Figura 2.14:</b> Resultados de la Pregunta 2 – Encuesta Locales.....	46
<b>Figura 2.15:</b> Resultados de la Pregunta 3 – Encuesta Locales.....	46
<b>Figura 2.16:</b> Resultados de la Pregunta 4 – Encuesta Locales.....	46
<b>Figura 2.17:</b> Resultados de la Pregunta 5- Encuesta Locales.....	47
<b>Figura 2.18:</b> Horas de mayor utilización de la red del local con picos máximos..	50
<b>Figura 2.19:</b> Resultados del tamaño de un Video.....	51
<b>Figura 2.20:</b> Resultados del tiempo de carga de Video.....	51

<b>Figura 2.21:</b> Tamaño de una página promedio al abrir facebook. ....	52
<b>Figura 2.22:</b> Tiempo de carga de una página al abrir facebook. ....	53
<b>Figura 2.23:</b> Tráfico generado en el servidor de base de datos Mysql.....	54
<b>Figura 2.24:</b> Severidad de la interferencia en el canal 11 .....	65
<b>Figura 2.25:</b> Interferencia canal 11 .....	65
<b>Figura 2.26:</b> Survey Activo Subsuelo 2, AP1 .....	69
<b>Figura 2.27:</b> <i>Survey</i> Activo Subsuelo 2, AP2.....	70
<b>Figura 2.28:</b> <i>Survey</i> Activo Subsuelo 2, AP3.....	70
<b>Figura 2.29:</b> <i>Survey</i> Activo Subsuelo 2, AP4.....	71
<b>Figura 2.30:</b> Survey Activo subsuelo uno, AP2 .....	72
<b>Figura 2.31:</b> Survey Activo subsuelo uno, AP-tres añadido .....	73
<b>Figura 2.32:</b> Distribución de APs en el nivel cero .....	74
<b>Figura 2.33:</b> Distribución de APs en el nivel cero .....	75
<b>Figura 2.34:</b> Distribución de APs en el Subsuelo 2 .....	76
<b>Figura 2.35:</b> Distribución de APs en el Subsuelo 1 .....	76
<b>Figura 2.36:</b> Distribución de APs en una planta Tipo .....	77
<b>Figura 2.37:</b> Planeamiento de frecuencias de la Red Inalámbrica .....	78
<b>Figura 2.38:</b> Distribución del cableado horizontal para la planta Subsuelo Dos..	79
<b>Figura 2.39:</b> Distribución del cableado horizontal para la planta Subsuelo Uno..	80
<b>Figura 2.40:</b> Distribución del cableado horizontal para la planta 0,2,4,6 .....	80
<b>Figura 2.41:</b> Distribución del cableado horizontal para la planta 1,3,5 .....	81
<b>Figura 2.42:</b> Distribución del cableado estructurado para la interconexión de los Access Point hacia el Cuarto de Telecomunicaciones.....	82
<b>Figura 2.43:</b> <i>Rack</i> de comunicaciones .....	85
<b>Figura 2.44:</b> Distribución Lógica de la red .....	86
<b>CAPÍTULO 3</b>	
<b>Figura 3.1:</b> Diagrama de Casos de Uso de la Aplicación .....	100
<b>Figura 3.2:</b> Diagrama de Clases de la Aplicación.....	101
<b>Figura 3.3:</b> Diagrama de Actividades de la aplicación.....	102
<b>Figura 3.4:</b> Estructura de Carpetas de un proyecto <i>Android</i> .....	103
<b>Figura 3.6:</b> Directorio de ubicación de los ficheros xml .....	107
<b>Figura 3.7:</b> Diagrama Base de Datos implementada en <i>MySQL</i> .....	108
<b>Figura 3.8:</b> Ciclo de vida de una Actividad .....	123

<b>Figura 3.9:</b> Modo gráfico de la interfaz <code>activity_main.xml</code> .....	128
<b>Figura 3.10:</b> Modo gráfico de la interfaz <code>localestipo.xml</code> .....	138
<b>Figura 3.11:</b> Modo gráfico de la intez <code>productos.xml</code> .....	139
<b>Figura 3.12:</b> Modo gráfico de la interfaz <code>ofertas.xml</code> .....	139
<b>Figura 3.13:</b> Interfaz - Productos de un local de tecnología por categoría. ....	143
<b>Figura 3.14:</b> Interfaz-Local de otra categoría para este caso del Tipo="Joyeria"144	
<b>Figura 3.15:</b> Modo gráfico de la interfaz <code>local.xml</code> .....	148
<b>Figura 3.16:</b> interfaz de un local y sus ofertas .....	152
<b>Figura 3.17:</b> Modo gráfico de la interfaz <code>ofertaslocal.xml</code> .....	153
<b>Figura 3.18:</b> Interfaz producto detalles muestra como ejemplo una tv. ....	155
<b>Figura 3.19:</b> Modo gráfico de la interfaz <code>productinfo.xml</code> .....	157
<b>Figura 3.20:</b> Interfaz oferta detalles muestra como ejemplo una laptop. ....	158
<b>Figura 3.21:</b> Modo gráfico de la interfaz <code>ofertinfo.xml</code> .....	159
<b>Figura 3.22:</b> Modo gráfico de la interfaz <code>subsuelodos.xml</code> .....	161
<b>Figura 3.23:</b> Modo gráfico de la interfaz <code>subsuelouno.xml</code> .....	162
<b>Figura 3.24:</b> Modo gráfico de la interfaz <code>plantabaja.xml</code> .....	163
<b>Figura 3.25:</b> Modo gráfico de la interfaz <code>niveluno.xml</code> .....	163
<b>Figura 3.26:</b> Modo gráfico de la interfaz <code>niveldos.xml</code> .....	164
<b>Figura 3.27:</b> Modo gráfico de la interfaz <code>niveltres.xml</code> .....	164
<b>Figura 3.28:</b> Modo gráfico de la interfaz <code>nivelcuatro.xml</code> .....	165
<b>Figura 3.29:</b> Modo gráfico de la interfaz <code>nivelcinco.xml</code> .....	165
<b>Figura 3.30:</b> Modo gráfico de la interfaz <code>nivelseis.xml</code> .....	165
<b>Figura 3.31:</b> API key de la aplicación "CenEspiral" .....	167
<b>Figura 3.32:</b> Interfaz Ubicación del Centro Comercial Espiral .....	171
<b>CAPÍTULO 4</b>	
<b>Figura 4.1:</b> Esquema del prototipo de prueba .....	173
<b>ANEXO A</b>	
<b>Figura A.1:</b> Menú Configuración en <i>Android</i> .....	A-1
<b>Figura A.2:</b> Instalador de la aplicación .....	A-2
<b>Figura A.3:</b> Instalador " <i>CenEspiral.apk</i> " .....	A-2
<b>Figura A.4:</b> Pantalla Principal .....	A-3
<b>Figura A.5:</b> Interfaz Búsqueda Local .....	A-4
<b>Figura A.6:</b> Interfaz Búsqueda Productos .....	A-5

<b>Figura A.7:</b> Interfaz Búsqueda Ofertas.....	A-5
<b>Figura A.8:</b> Interfaz Galería de Locales.....	A-6
<b>Figura A.9:</b> Modelo Interfaz Niveles .....	A-7
<b>Figura A.10:</b> Interfaz local tipo en general.....	A-8
<b>Figura A.11:</b> Interfaz local tipo “TECNOLOGIA” .....	A-8
<b>Figura A.12:</b> Información Ofertas .....	A-10
<b>Figura A.13:</b> Información Producto.....	A-11
<b>Figura A.14:</b> Buzón de sugerencias .....	A-11
<b>Figura A.15:</b> Ubicación del Centro Comercial Espiral .....	A-12
<b>Figura A.16:</b> Ubicación del Usuario.....	A-13

## **ANEXO C**

<b>Figura C.1:</b> Gráfico del tráfico del Local 22 Novicompu .....	C-1
<b>Figura C.2:</b> Gráfico del tráfico del Local 24 Powernet .....	C-1
<b>Figura C.3:</b> Gráfico del tráfico del Local 34 Compukit .....	C-2
<b>Figura C.4:</b> Gráfico del tráfico del Local 33 Universal Technology .....	C-2
<b>Figura C.5:</b> Gráfico del tráfico del Local 41 Konami Games.....	C-2
<b>Figura C.6:</b> Gráfico del tráfico del Local 47 RCM Sistemas.....	C-3
<b>Figura C.7:</b> Gráfico del tráfico del Local 50 Casa Bolivia .....	C-3
<b>Figura C.8:</b> Gráfico del tráfico del Local 67 Personalízate .....	C-3
<b>Figura C.9:</b> Gráfico del tráfico del Local 75 Movie Light .....	C-4
<b>Figura C.10:</b> Gráfico del tráfico del Local 80 Audiocenter .....	C-4
<b>Figura C.11:</b> Gráfico del tráfico del Local 81 Levelup.....	C-4
<b>Figura C.12:</b> Gráfico del tráfico del Local 94 Mágico Bazar.....	C-5
<b>Figura C.13:</b> Gráfico del tráfico del Local 97 Hellen Novedades .....	C-5
<b>Figura C.14:</b> Gráfico del tráfico del Local 98 Cell Mep .....	C-5
<b>Figura C.15:</b> Gráfico del tráfico del Local 102 Fachas.....	C-6
<b>Figura C.16:</b> Gráfico del tráfico del Local 105 Ecuaventura .....	C-6
<b>Figura C.17:</b> Gráfico del tráfico del Local 124 Capcom .....	C-6
<b>Figura C.18:</b> Gráfico del tráfico del Local 134 EC Robotics .....	C-7
<b>Figura C.19:</b> Gráfico del tráfico del Local 136 Soltec .....	C-7
<b>Figura C.20:</b> Gráfico del tráfico del Local 140 Ecuavisión .....	C-7

## ÍNDICE DE TABLAS

### CAPÍTULO 1

<b>Tabla 1.1:</b> Comparación de Estándares de IEEE 802.11 [5] .....	7
<b>Tabla 1.2:</b> Funcionalidad de una <i>tablet</i> .....	33

### CAPÍTULO 2

<b>Tabla 2.1:</b> Locales y computadores en el Centro Comercial El Espiral .....	34
<b>Tabla 2.2:</b> Número de usuarios fijos y visitantes por cada nivel .....	49
<b>Tabla 2.3:</b> Resumen de tráfico para visitantes del centro comercial Espiral.....	55
<b>Tabla 2.4:</b> Resumen de tráfico para empleados y dueños de los locales .....	56
<b>Tabla 2.5:</b> Resultados obtenidos del <i>Site Survey</i> Pasivo en el subsuelo 2.....	59
<b>Tabla 2.6:</b> Resultados obtenidos del <i>Site Survey</i> Pasivo en el subsuelo 1.....	59
<b>Tabla 2.8:</b> Resultados obtenidos del <i>Site Survey</i> Pasivo en el piso1 .....	61
<b>Tabla 2.10:</b> Resultados obtenidos del <i>Site Survey</i> Pasivo en el piso 3 .....	62
<b>Tabla 2.11:</b> Resultados obtenidos del <i>Site Survey</i> Pasivo en el piso .....	63
<b>Tabla 2.12:</b> Resultados obtenidos del <i>Site Survey</i> Pasivo en el piso 5 .....	63
<b>Tabla 2.13:</b> Resultados obtenidos del <i>Site Survey</i> Pasivo en el piso 6 .....	64
<b>Tabla 2.14:</b> Comparación de estándares inalámbricos.....	66
<b>Tabla 2.15:</b> Distribución de Puntos de Acceso Inalámbrico por piso .....	68
<b>Tabla 2.16:</b> Resumen del cableado horizontal por piso .....	81
<b>Tabla 2.17:</b> Resumen del cableado vertical por piso .....	82
<b>Tabla 2.18:</b> Resumen del cableado estructurado del Centro Comercial.....	83
<b>Tabla 2.19:</b> Resumen de los elementos de área de trabajo .....	83
<b>Tabla 2.20:</b> Cálculo de unidades para dimensionar el <i>Rack</i> de equipos .....	84
<b>Tabla 2.21:</b> Altura Normalizada del <i>Rack</i> de equipos .....	84
<b>Tabla 2.22:</b> Resumen del direccionamiento IP para cada VLAN .....	88
<b>Tabla 2.23:</b> Requerimientos mínimos de los Puntos de Acceso .....	90
<b>Tabla 2.24:</b> Comparación de tres marcas de AP's.....	91
<b>Tabla 2.25:</b> Características mínimas del <i>Switch</i> de acceso.....	92
<b>Tabla 2.26:</b> Comparación de dos marcas de <i>Switches</i> .....	93
<b>Tabla 2.27:</b> Requisitos mínimos que debe cumplir el <i>LAN Controller</i> a escoger..	94
<b>Tabla 2.28:</b> Comparación de dos marcas de <i>Wireless LAN Controllers</i> .....	95
<b>Tabla 2.29:</b> Dimensionamiento del CPU .....	97
<b>Tabla 2.30:</b> Dimensionamiento de la memoria RAM.....	98

<b>Tabla 2.31:</b> Dimensionamiento del Disco Duro .....	98
<b>CAPÍTULO 4</b>	
<b>Tabla 4.1:</b> Muestra las Actividades que realiza la actividad principal “MAIN” ....	174
<b>Tabla 4.2:</b> Muestra las Actividades que realiza la actividad LOCALES .....	174
<b>Tabla 4.3:</b> Muestra las Actividades que realiza la actividad OFERTAS.....	175
<b>Tabla 4.4:</b> Muestra las Actividades que realiza la actividad PRODUCTOS.....	175
<b>Tabla 4.5:</b> Muestra las Actividades que realiza la actividad LOCAL DETALLES .....	176
<b>Tabla 4.6:</b> Muestra las Actividades que realiza la actividad OFERTAS LOCAL	176
<b>Tabla 4.7:</b> Muestra las Actividades que realiza la actividad PRODUCTO DETALLES .....	177
<b>Tabla 4.8:</b> Muestra las Actividades que realiza la actividad OFERTA DETALLES .....	177
<b>Tabla 4.9:</b> Muestra las Actividades que realiza la actividad de acuerdo al piso	178
<b>Tabla 4.10:</b> Muestra los resultados de las diferentes actividades del Mapadel Sitio por pisos .....	178
<b>Tabla 4.11:</b> Muestra las Actividades que realiza la actividad BUZÓN DE SUGERENCIAS .....	179
<b>Tabla 4.12:</b> Costo de la red pasiva .....	179
<b>Tabla 4.13:</b> Costo de la instalación y certificación de puntos .....	180
<b>Tabla 4.14:</b> Costo de la alternativa HP .....	181
<b>Tabla 4.15:</b> Costo alternativa Cisco .....	181
<b>Tabla 4.16:</b> Servidores para la Base de Datos .....	182
<b>Tabla 4.17:</b> Costo del enlace a Internet.....	183
<b>Tabla 4.18:</b> Costo de implementación de la aplicación.....	184
<b>Tabla 4.19:</b> Costo total del proyecto .....	184

## ÍNDICE DE CÓDIGOS

### CAPÍTULO 3

<b>Código 3.1:</b> Variables de conexión a la base de datos .....	109
<b>Código 3.2:</b> Funciones de conexión y desconexión .....	109
<b>Código 3.3:</b> Función <i>Connect</i> .....	110
<b>Código 3.4:</b> Función <i>Close</i> .....	110
<b>Código 3.5:</b> Consulta a la base de datos .....	111
<b>Código 3.7:</b> Respuesta de GET_LOCAL_TIPO.PHP .....	113
<b>Código 3.8:</b> Consulta empleada en el fichero GET_LOCAL.PHP .....	113
<b>Código 3.9:</b> Consulta en el fichero GET_PRODUCTOS_TIPO.PHP .....	114
<b>Código 3.10:</b> Consulta en el fichero GET_OFERTA_TIPO.PHP .....	114
<b>Código 3.11:</b> Consulta en el fichero “GET_PRODUCTO_DETALLES.PHP” .....	115
<b>Código 3.12:</b> Consulta en el fichero “GET_PRODUCTO_DETALLES.PHP” .....	116
<b>Código 3.13:</b> Consulta de fichero GET_LAPTOPS.PHP .....	116
<b>Código 3.14:</b> Consulta de fichero GET_TABLETS.PHP .....	117
<b>Código 3.15:</b> Consulta en el fichero GET_IMPRESORAS.PHP .....	118
<b>Código 3.16:</b> Consulta en el fichero GET_OTROS_TECNOLOGIA.PHP .....	119
<b>Código 3.17:</b> Consulta en el fichero GET_CLIENTES.PHP .....	120
<b>Código 3.18:</b> Fichero UPDATE_CLIENTES.PHP.....	120
<b>Código 3.19:</b> Consulta en el fichero UPDATE_CLIENTES.PHP .....	121
<b>Código 3.20:</b> Muestra si el cliente fue ingresado.....	121
<b>Código 3.21:</b> Función <i>OnCreate</i> .....	125
<b>Código 3.23:</b> Llamada al evento <i>click</i> .....	126
<b>Código 3.24:</b> <i>RelativeLayout</i> .....	127
<b>Código 3.25:</b> Atributos de un <i>ImageView</i> .....	127
<b>Código 3.26:</b> Librerías en <i>JSONPARSER</i> .....	129
<b>Código 3.27:</b> Variables en la clase “ <i>JSONPARSER</i> ” .....	129
<b>Código 3.29:</b> Búfer de datos.....	131
<b>Código 3.30:</b> Clase <i>Locales Activity</i> .....	132
<b>Código 3.31:</b> Método <i>OnCreate</i> .....	133
<b>Código 3.32:</b> Adaptación de la lista.....	134
<b>Código 3.33:</b> Evento <i>Tocar</i> de cada botón.....	135
<b>Código 3.34:</b> Datos de Locales por tipo .....	136

<b>Código 3.35:</b> Adaptación de datos en la lista .....	137
<b>Código 3.36:</b> Obtención del id de un local.....	140
<b>Código 3.37:</b> Comparación en locales del tipo tecnología .....	142
<b>Código 3.38:</b> Comparación para el caso Joyería.....	143
<b>Código 3.38:</b> Ejecución de la función LoadAllProducts .....	147
<b>Código 3.40:</b> Evento Cambiar de Actividad.....	147
<b>Código 3.41:</b> Ejemplo de <i>ImageButton</i> .....	148
<b>Código 3.42:</b> Función <i>onCreate</i> de la clase “Ofertas Local”.....	149
<b>Código 3.45:</b> Características cargadas del producto seleccionado.....	156
<b>Código 3.47:</b> Creación del <i>ArrayList</i> “localList” .....	159
<b>Código 3.48:</b> Creación del objeto <i>JSON GET</i> .....	160
<b>Código 3.49:</b> Evento <i>Touch</i> para obtener <i>GetLocalDetails</i> .....	160
<b>Código 3.50:</b> Parámetros de la clase “ <i>BuzonActivity</i> ” .....	166
<b>Código 3.51:</b> Parámetros ingresados en el elemento <i>application</i> .....	167
<b>Código 3.52:</b> Parámetros ingresados en el elemento <i>manifest</i> .....	168
<b>Código 3.53:</b> Declaración de variables.....	168
<b>Código 3.54:</b> Función <i>OnCreate</i> .....	169
<b>Código 3.55:</b> Funciones “Ir A Espiral”, “Mi posición”, “Marcador” .....	170

## ÍNDICE DE ECUACIONES

### CAPÍTULO 2

<b>Ecuación 2.1:</b> Muestreo población infinita.....	42
<b>Ecuación 2.2:</b> Muestreo población finita .....	45



## RESUMEN

El presente proyecto de titulación describe el diseño de la red inalámbrica del Centro Comercial Espiral y el desarrollo de una aplicación que puede ser desplegada en una *Tablet* con el Sistema Operativo Android. El programa permite la ubicación de locales dentro del centro comercial, conocer cuáles son los productos y las ofertas que ofrece cada almacén dentro del mismo.

En el Capítulo 1 se presenta los fundamentos teóricos de una red 802.11, así como las consideraciones necesarias para el diseño de una red inalámbrica. Se muestra la arquitectura del sistema operativo Android, se realiza un breve análisis de las versiones que existen hasta la actualidad, así como de las herramientas necesarias para desarrollar la aplicación en el lenguaje de programación Java.

En el Capítulo 2 se incluye el diseño de la red inalámbrica para el centro comercial, se inicia con el análisis de la situación actual del Centro Comercial Espiral, especificando los usuarios que utilizarán la red y del tráfico que fluirá por la misma. Luego se muestran los resultados que se obtuvieron después de realizar el *Site Survey* Pasivo y Activo. Se presenta el diseño del sistema de cableado estructurado de los puntos de acceso, el direccionamiento IP que tendrá la red inalámbrica y el requerimiento de los equipos con su respectiva comparación en el mercado

El el Capítulo 3 contiene el desarrollo de la aplicación en el lenguaje Java para Android, teniendo en cuenta los diagramas UML, la base de datos y la comunicación con la misma mediante el manejo de ficheros php en notación JSON, adecuados para asegurar el rendimiento y correcto funcionamiento del programa. Por lo cual se detalla cada una de clases con las respectivas funciones principales implementadas

En el Capítulo 4 se presenta el prototipo de prueba del proyecto, así como el análisis de los resultados de la aplicación. También se muestra el costo que tendría la implementación tanto de la red inalámbrica, como de la aplicación;

presentando dos alternativas para la compra de equipos para la red activa y pasiva.

En el Capítulo 5 se presentan las conclusiones y recomendaciones resultantes de este proyecto de titulación.

En los anexos se incluye un manual de instalación y utilización del programa en una tablet con Android; el código fuente de la aplicación, las figuras obtenidas al realizar el análisis de tráfico en el centro comercial, y las encuestas realizadas a los clientes y propietarios de los locales del mismo.

## PRESENTACIÓN

Actualmente existe un enorme crecimiento de usuarios que utilizan dispositivos móviles y terminales inteligentes, debido a la necesidad de interactuar con servicios en cualquier momento y en cualquier lugar, por lo tanto surge la obligación de crear aplicaciones para este tipo de terminales.

Al ingresar a un centro comercial, la forma de localizar cada uno de los almacenes es a través de un mapa, publicado en las entradas del establecimiento. Sin embargo si el cliente desea saber qué productos y ofertas dispone cada local debe acercarse al mismo. En la actualidad la tendencia al uso de dispositivos móviles va en aumento tanto es así que la demanda en el mercado de *tablets* y teléfonos inteligentes a crecido considerablemente. Es por ello que el desarrollo de la presente aplicación permitirá a los visitantes del Centro Comercial el informarse de manera adecuada.

Se presenta al usuario una aplicación que permite obtener información acerca de los locales del Centro Comercial Espiral, los productos y ofertas que cada uno de ellos ofrece, y la ubicación de los mismos, con una interfaz amigable para el usuario y cliente de la entidad.

Adicional, el usuario tendrá acceso a la red inalámbrica del establecimiento, pudiendo obtener los datos para cargar la aplicación, inclusive navegar en la Internet.

# CAPÍTULO 1

# 1. FUNDAMENTOS TEÓRICOS

## 1.1 INTRODUCCIÓN

En el capítulo presente se detallan los conceptos necesarios para realizar todo el proyecto. Los fundamentos de la arquitectura IEEE 802.11, los puntos que se debe tomar en cuenta para un óptimo diseño de una red inalámbrica bajo éste estándar, y las herramientas para facilitar la recolección de parámetros como la interferencia, el tráfico en la red.

Se resume el lenguaje de programación Java, las herramientas necesarias para el desarrollo de una aplicación bajo *Eclipse* para Android y el Sistema Operativo Android.

## 1.2 REDES INALÁMBRICAS 802.11

Una red inalámbrica de área local (WLAN) permite la comunicación entre terminales y servidores, en el área ocupada por una empresa o institución localmente.

La facilidad que brinda una red inalámbrica es permitir la utilización de dispositivos móviles de todo tamaño y característica, manejando un mismo protocolo, y/o estándar de comunicación entre las estaciones; es decir que todos los dispositivos hablen el mismo idioma.

El estándar IEEE 802.11 surge como una solución para una WLAN, a nivel de capa física y de subcapa MAC, transmitiendo a través de ondas electromagnéticas; propuesta por el Instituto de Ingenieros Eléctricos y Electrónicos (IEEE), siendo la más acogida a nivel mundial, brindando la interoperabilidad entre dispositivos con IEEE 802.11 que cuenten con la certificación Wi-Fi (*Wireless Fidelity*).

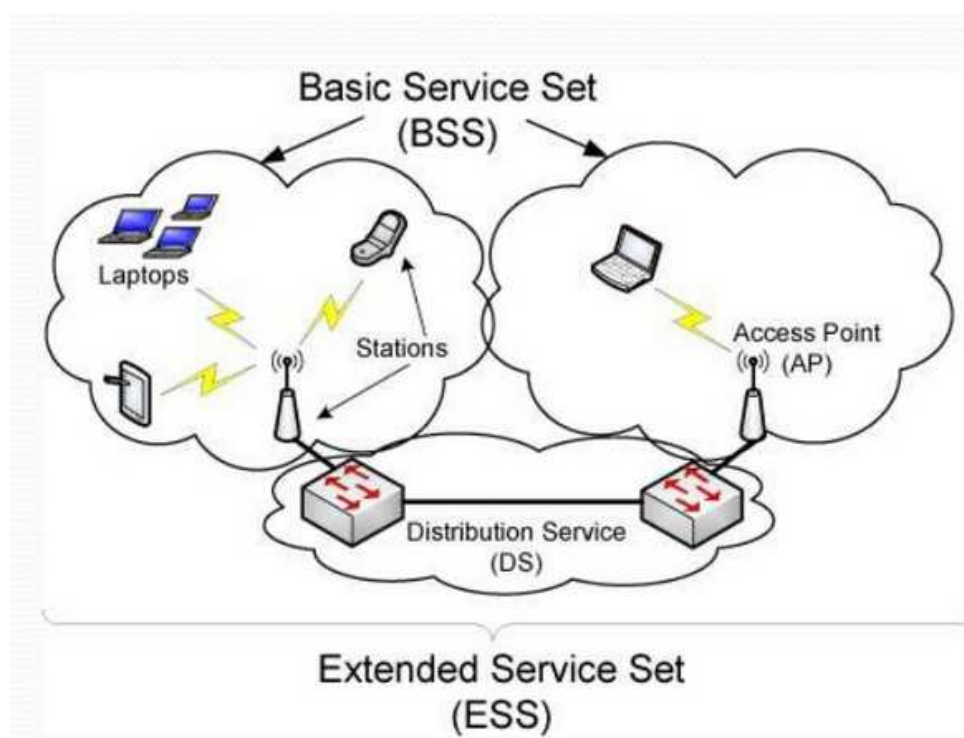
### 1.2.1 COMPONENTES DE UNA RED IEEE 802.11 [1]

El protocolo IEEE 802.11 es una tecnología que permite proporcionar conectividad entre estaciones inalámbricas e infraestructuras de redes cableadas.

Se puede mencionar como elementos a los siguientes:

- **Estación Inalámbrica:** Dispositivo que cumpla con el estándar IEEE 802.11, para relacionarse con la subcapa MAC y la capa Física y poder transmitir a través del medio inalámbrico, en diferentes frecuencias dependiendo del estándar utilizado.
- **Punto de Acceso (AP):** Dispositivo que puede actuar como un punto central de la red inalámbrica, permite la comunicación entre estaciones.
- **Puente (Bridge):** Dispositivo que permite la comunicación entre redes, es decir interconecta segmentos de red haciendo la transferencia de datos de una red hacia otra.
- **Conjunto de Servicio Básico (BSS):** Conjunto de estaciones que coordinan su acceso al medio; tiene un identificador de conjunto de servicio (SSID), el área en la cual trabaja es conocida como área de servicios básicos (BSA).
- **Conjunto de Servicio Básico independiente (IBSS):** Se define como el conjunto de estaciones que se comunican entre sí sin necesidad de un punto de acceso en común.
- **Conjunto de servicio Extendido (ESS):** Es una asociación de dos o más BSS, conectados por un sistema de distribución común que tendrán un único identificador de conjunto de servicio (SSID). Esto permite la creación de una red inalámbrica de tamaño y complejidad arbitrarios.

- **Sistema de distribución (DS):** El Sistema de Distribución, permite la interconexión inalámbrica de puntos de acceso en una red, puede ser ampliado mediante múltiples puntos de acceso sin la necesidad de un cable troncal que los conecte.



**Figura 1.1:** Topologías Lógicas Inalámbricas [2]

### 1.2.2 MODOS DE OPERACIÓN

Se puede mencionar dos tipos de operación de una red IEEE 802.11: Ad-hoc e infraestructura.

- **Ad-hoc:** En este modo las estaciones constituyen una IBSS, es decir que en su funcionamiento no dependen de un punto de acceso inalámbrico.
- **Infraestructura:** En este modo la BSS debe disponer de un punto de acceso inalámbrico mediante el cual se conectarán todas las estaciones; es decir, el cliente utiliza el AP para acceder a los recursos de la red cableada.

Tanto en el modo de infraestructura como en el modo de ad-hoc el SSID permite identificar el nombre de la red inalámbrica.

### 1.2.3 SERVICIOS DE IEEE 802.11

Los servicios de IEEE 802.11 son:

- **Autenticación:** Servicio de estación mediante el cual se establece la identidad de cada una de ellas para realizar la asociación y poder transmitir datos. En éste se realiza un intercambio de identidades con un AP o con otra estación, se ofrecen dos tipos de servicios de autenticación:
  - **Autenticación Abierta** (*Open System Authentication*), en donde cualquiera que solicite autenticarse será aceptado;
  - **Autenticación de llave compartida**, en donde para poder autenticarse en la red la estación debe conocer la una clave.
- **Desautenticación:** Servicio con el cual se termina la relación de autenticación, es decir deja de pertenecer a la red a la cual la estación estuvo asociada.
- **Confidencialidad de los Datos:** Servicio que permite mantener la privacidad necesaria entre quienes comparten los datos mediante la red, para lo cual se utilizan mecanismos de encriptación.
- **Entrega de MSDU:** Propiamente se refiere a la entrega y envío de datos entre estaciones mediante la red.
- **Asociación:** Servicio mediante el cual se establece una asociación entre el AP y la estación, con esto, el DS (Sistema de Distribución) puede determinar a qué AP se encuentra asociada cierta estación; tomando en cuenta que ésta puede asociarse a un AP a la vez.



- **Reasociación:** Servicio del *DS* que permite la transición entre BSS dentro de una ESS cuando la estación se encuentra en movimiento. Lo que permite que una estación se asocie a un AP de acuerdo al nivel de señal que reciba de cada uno de los APs.
- **Desasociación:** Servicio del *DS* que es invocado cuando se desea terminar una asociación, puede ser que el servicio sea invocado por la estación o por el AP; la desasociación es una notificación, no un pedido, lo que quiere decir que no puede ser rechazado por ninguna de la partes. Todos los datos de movilidad almacenados son removidos.
- **Distribución:** Servicio del *DS* que permite el envío de datos a través del sistema de distribución entre APs para que llegue a su correcto destinatario.
- **Integración:** Servicio que permite la comunicación entre el *DS* (Sistema de Distribución) y una red que no sea 802.11. Los detalles de la función de distribución dependen del sistema de distribución, es decir no está indicado dentro del estándar.

#### 1.2.4 ARQUITECTURA IEEE 802.11

El estándar IEEE 802.11 cubre especificaciones para la capa física y para la subcapa MAC del modelo de referencia ISO/OSI.

##### 1.2.4.1 Capa Física [4]

Es el medio por el cual se enviará la información, en ésta se definen las características eléctricas, mecánicas y funcionales del canal de comunicación; como se detalla en la Figura 1.2.

Se divide en una Subcapa Dependiente del medio (PMD) y una Subcapa de convergencia (PLCP), las subcapas a su vez son administradas por la entidad de

gestión de la capa física (PLME) que proporciona las interfaces de gestión de la capa.

Capa de Enlace de Datos	Subcapa de enlace de datos (LLC)		Entidad de Administración de estaciones
	Subcapa de acceso al medio (MAC)	Entidad de Administración de la subcapa MAC	
Capa Física	Procedimiento de Convergencia (PLCP)	Entidad de Administración de la Capa Física (PLME)	
	Dependiente del medio físico (PMD)		

**Figura 1.2:** Modelo de Capas IEEE 802.11 [3]

Se puede definir tres tecnologías dentro de la capa física de IEEE 802.11 para difundir la señal: Espectro Ensanchado por Secuencia Directa (DSSS), Espectro Ensanchado por Salto de frecuencia (FHSS) e Infrarrojo.

- **Espectro Ensanchado por Secuencia Directa (DSSS):** Opera en el rango que va desde 2.4 GHz hasta los 2.483 GHz, teniendo total 83 MHz de ancho de banda. Éste se divide en 14 canales de 5 MHz de AB por canal, para prevenir interferencias entre canal los canales adyacentes deben estar separados al menos 22 MHz, por lo que existen 3 canales sin sobrelaparse o interferir entre sí. Utiliza la modulación DBPSK (*Differential Binary Phase Shift Keying*) alcanzando una velocidad de 1 Mbps, y la modulación DQPSK (*Differential Quadrature Phase Shift Keying*) con 2 Mbps de velocidad.
- **Espectro Ensanchado por Salto de Frecuencia (FHSS):** Transmite parte de la información en una determinada potencia y en un intervalo de tiempo menor a 400ms; pasado este tiempo se cambia la frecuencia de emisión y se sigue transmitiendo pero a otra frecuencia. Se describe mediante la

modulación en frecuencia FSK (*Frequency Shift Keying*) trabajando a una velocidad de 1 Mbps ampliable hasta 2 Mbps.

- **Infrarrojo:** No ha sido demasiado utilizada en redes inalámbricas, ocupa frecuencias justo por debajo del rango de la luz visible, los infrarrojos no pueden pasar a través de objetos opacos, aunque si se refleja en ciertas superficies. Se especifica dos modulaciones para esta tecnología: 16 ppm y 4 ppm, a velocidades de 1 y 2 Mbps respectivamente.

Hasta el momento se tienen seis etapas por las que ha pasado el estándar 802.11, las cuales se detallan en la tabla 1.1, donde se observan características como tipo de modulación, frecuencia a la que trabaja, velocidad de transmisión, entre otros.

Parámetro	802.11a	802.11b	802.11g	802.11n	802.11ac	802.11ad
<b>Frecuencia (GHz)</b>	5.15 a 5.35 5.475 a 5.725 5.725 a 5.85	2.4 a 2.4835	2.4 a 2.4835	2.4 a 2.4835 5.15 a 5.850	5	2.4 5 60
<b>Modulación</b>	OFDM	DSSS CCK	OFDM DSSS CCK	OFDM MIMO	OFDM MIMO	OFDM BEAMFORMIG
<b>Ancho de banda de canal (MHz)</b>	20	22	22	20 y 40	20, 40, 80, 160	Hasta 2160
<b>Tasa de transmisión (Mbps)</b>	6, 9, 12, 18, 24, 36, 54	1, 2, 5.5, 11	1, 2, 5.5, 6, 9, 11, 12, 18, 22, 24, 33, 36, 48, 54	Hasta 300	Hasta 6 Gbps	Hasta 7 Gbps
<b>MIMO</b>	1	1	1	4x4	8x8	-
<b>Portadoras por canal</b>	48 de datos, 4 pilotos	1	48 de datos, 4 pilotos	52 de datos, 4 pilotos a 20 MHz 108 de datos, 6 pilotos a 40 MHz	52 de datos, 4 pilotos a 20 MHz 108 de datos, 6 pilotos a 40 MHz 234 de datos, 8 pilotos a 80 MHz 468 de datos, 16 pilotos a 160 MHz	-

**Tabla 1.1:** Comparación de Estándares de IEEE 802.11 [5]

#### 1.2.4.2 Capa Enlace de Datos [4]

La capa de enlace de datos se divide en dos subcapas la Subcapa MAC y la LLC (Subcapa de Control de Enlace Lógico).

Subcapa de Control de Enlace Lógico (LLC), es la encargada del control de flujo y de errores. La subcapa MAC se divide, a su vez, en otras dos subcapas:

- **Subcapa MAC (Control de Acceso al Medio)** es responsable del mecanismo de acceso y la fragmentación de los paquetes. Realiza controles de acceso al medio de transmisión a través de CSMA/CA (*Carrier Sense Multiple Access with Collision Avoidance*).
- **Entidad de administración de la subcapa MAC (MAC Management)** Se encarga de administrar las actividades de *Roaming*<sup>1</sup> dentro del ESS (Conjunto de Servicios Extendidos), la energía, y los procesos de asociación y disociación durante la registración.

#### 1.2.5 SEGURIDAD EN REDES IEEE 802.11

Al hablar sobre la seguridad en una red, se debe tomar en cuenta que el hecho de utilizar un medio inalámbrico (radiofrecuencia) el cual es de fácil acceso, hace que la información transmitida a través de ella puede ser más vulnerable que en una red cableada. Para asegurar la información, un sistema debe proveer: autenticación al tratar de asociarse a la red, confidencialidad de los datos, integridad de la información, y disponibilidad.

- **Autenticación:** Al momento de realizar la autenticación del usuario, se asegura que en realidad se trata de la persona es quien dice ser; y de esta manera obtener el acceso a la red, y a la información que ésta proporciona.

---

<sup>1</sup> *Roaming*: Es la capacidad que tiene una estación para poder moverse de un área de cobertura a otra sin perder conectividad con la red.

Para esto se puede utilizar una lista de nombres de usuario y contraseña, que pueden estar almacenados localmente en el *Access Point*, o en una base de datos externa como un servidor *Radius*.

El estándar 802.1x define el uso del Protocolo de Autenticación Extensible (EAP), el cual puede ser usado sobre redes ethernet, 802.11, *Token Ring*, y FDDI (*Fiber Distributed Data Interface*).

El trabajo de EAP es el siguiente: El cliente o el dispositivo envía un mensaje de inicio al Punto de Acceso, quien le pide identificarse y permite solo el envío de mensajes de autenticación; el cliente envía su identidad que es reenviada al servidor de autenticación para comprobar los datos, al ser correctos el servidor envía un mensaje de aceptación al *Access Point* y éste la reenvía al cliente para comenzar la transmisión de información.

Para la autenticación se utiliza mecanismos *Hash* (criptografía), como SHA y MD5, para el intercambio de llaves; en el ambiente WLAN son usados EAP, EAP-TTLS (EAP Tunelizado).

- **Confidencialidad:** Se refiere a mantener los datos en un alto grado de privacidad, es decir que la información llegue y sea entendida nada más por el usuario autorizado a recibirla. Para esto se utilizan varios mecanismos como lo son Filtrado MAC, Filtrado SSID, WEP (*Wired Equivalence Protocol*), WPA (*WiFi Protected Access*), el estándar 802.11i<sup>2</sup>, 802.11w<sup>3</sup>, así como VLANs, firewalls, ACLs, métodos que no son propios de una red inalámbrica, pero se pueden aplicar en función de obtener un sistema de seguridad más robusto.
- **Integridad:** Se refiere a mantener la información intacta y entregarla al usuario sin ningún cambio no autorizado. Un mecanismo para mantener la integridad de los datos es el código de redundancia cíclica (CRC), aunque

---

<sup>2</sup> 802.11i: Estándar dirigido a la seguridad en una red inalámbrica abarca protocolos de autenticación y codificación.

<sup>3</sup> 802.11w: Estándar dirigido a mejorar la Capa de Control de Acceso al Medio para aumentar la seguridad de los protocolos de autenticación y codificación

acompañado con WEP puede resultar inseguro, ya que puede ser cambiado sin necesidad de conocer la llave. Para los protocolos WPA y WPA2 se incluye un mensaje de código de autenticación más seguro y un contador de tramas, con lo cual se puede descartar un ataque de repetición (*replay attack*).

- **Disponibilidad:** Asegurar que la red va a estar disponible todo el tiempo, y en caso de recibir un ataque, pueda recuperarse lo más rápido posible. Para mantener la disponibilidad de la misma se pueden utilizar mecanismos de redundancia tomando en cuenta la misma seguridad para todo el sistema.

Para asegurar los parámetros mencionados anteriormente, se puede utilizar protocolos de seguridad creados en función de la necesidad de proteger la información de terceras personas, los más utilizados en una red IEEE 802.11 son: WEP (*Wired Equivalent Privacy*), WPA (*WiFi Protected Access*) y WPA2.

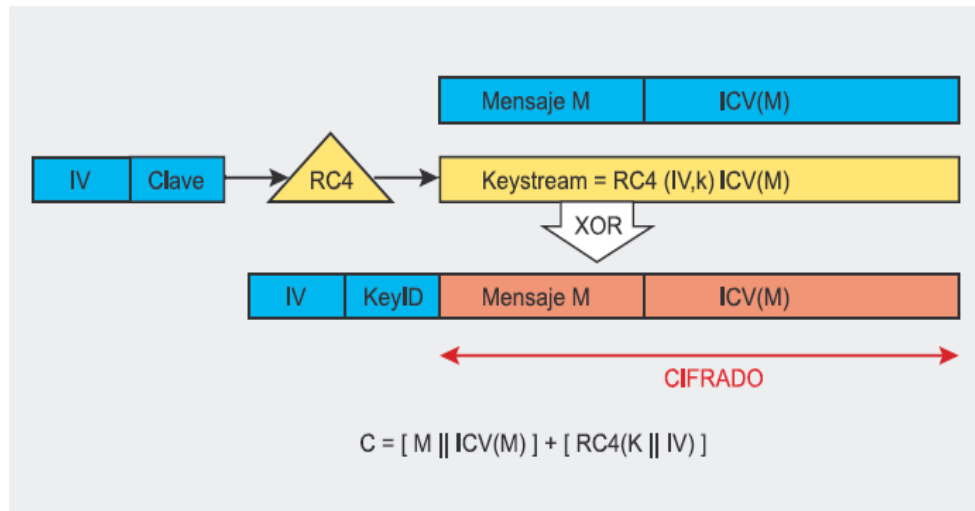
#### 1.2.5.1 Privacidad Equivalente a Cableado (WEP) [6]

Sistema de encriptación estándar soportado por la mayoría de las soluciones inalámbricas, incluido en los estándares 802.11a, b y g utiliza una misma clave simétrica y estática compartida entre las estaciones de trabajo y en el punto de acceso. Los datos son encriptados con el algoritmo de cifrado RC4<sup>4</sup> de 64 o 128 bits que corresponden a 24 bits del vector de inicialización (IV) más 40 o 104 bits de la clave secreta; éstos 40 bits son distribuidos manualmente, mientras que el vector de inicialización es generado dinámicamente para cada trama transmitida.

El proceso de encriptación WEP es el siguiente:

---

<sup>4</sup> RC4: Algoritmo de cifrado simétrico que genera una clave de flujo, que es pasada por la función XOR para producir el código cifrado. Para el descifrado se utiliza la misma función de cifrado.



**Figura 1.3:** Funcionamiento del protocolo de seguridad WEP [6]

- La clave compartida manualmente de 40 o 104 bits se concatena con el vector de inicialización (IV) de 24 bits formando una cadena de 64 o 128 bits.
- Se calcula el CRC-32 de los datos que se van a transmitir, valor que se incluye dentro de la trama IEEE 802.11.
- A través del algoritmo RC4 se encriptan la cadena que contiene la clave compartida y el IV, de lo cual resulta el *Keystream*.
- Se aplica la función XOR entre el *Keystream* y los datos dando como resultado el texto encriptado.
- A este resultado se añade el vector de inicialización en texto plano dentro de la trama 802.11.

En la actualidad no es recomendable el uso de este protocolo debido a las debilidades que presenta, tales como tener una longitud fija y dependiendo del fabricante un valor fijo del vector de inicialización, además no dispone de un mecanismo de protección contra mensajes repetidos.

### 1.2.5.2 Acceso Inalámbrico Protegido (WPA) [6]

Protocolo desarrollado como borrador del estándar 802.11i en función de omitir las debilidades de WEP, trabaja a nivel MAC, su uso es recomendado para oficinas pequeñas y usuarios domésticos. Las claves son distribuidas dinámicamente a través de un servidor RADIUS.

Puede trabajar con dos modos de autenticación: WPA-PSK (Llave Pre-compartida), donde se comparte manualmente las claves y WPA-EAP (Protocolo de Autenticación Extensible), con el uso del estándar 802.1x<sup>5</sup> y de un servidor RADIUS.

WPA utiliza TKIP<sup>6</sup> (*Temporal Key Integrity Protocol*) para la generación temporal de claves y MIC (Chequeo de Integridad de Mensaje) para revisar la integridad de los datos.

### 1.2.5.3 Acceso Inalámbrico Protegido-2 (WPA2) [6]

Es la versión completa del estándar 802.11i, fue emitida por la alianza WiFi, compatible con WEP y WPA, existen dos tipos del protocolo WPA2: Personal y Empresarial.

WPA2-Personal, específicamente diseñado para uso doméstico, encripta los datos con AES y utiliza una contraseña para establecer el acceso a la red. Por otro lado, WPA2-Empresarial diseñado para redes empresariales, cifra los datos con AES, verifica la integridad de los usuarios con el protocolo EAP (*Extensible Authentication Protocol*), que proporciona un mecanismo estándar para aceptar los métodos de autenticación de acuerdo a la variedad de escenarios y tipos de dispositivos

WPA2 emplea el protocolo de cifrado y autenticación CCMP, (*Cipher Block Chaining/Message Authentication Code Protocol*) basado en el algoritmo de encriptación de bloques AES, además utiliza el método de autenticación de

---

<sup>5</sup> 802.1x: Norma IEEE para el control de acceso a la red basada en puertos.

<sup>6</sup> TKIP: Protocolo de encriptación que trabaja con claves temporales generadas con el algoritmo RC4 combinadas con la dirección MAC del cliente y un vector de inicialización.



mensajes CBC-MAC (*Cipher Block Chaining*) para el chequeo de integridad de mensaje (MIC), usa una clave única pero con diferentes vectores de inicialización, el vector es incrementado en cada fragmento del paquete.

Las principales ventajas que presenta WPA2 son:

- Alto nivel de seguridad
- Cifrado más seguro
- Autenticación en la red inalámbrica transparente

### **1.2.6 VENTAJAS Y DESVENTAJAS GENERALES DE IEEE 802.11 FRENTE A UNA RED CABLEADA**

Al poder acceder a la red mediante un medio inalámbrico, el uso de dispositivos móviles con certificado WiFi va en aumento facilitando la obtención de información de una red cableada, así como servicios de correo electrónico, bases de datos, navegación por la Internet, e incluso el acceso a una red corporativa, ya sea local, o mediante VPNs<sup>7</sup>.

Como ventajas generales de una red inalámbrica frente a una red cableada se tienen:

- Se reducen costos de instalación, puede ser fácilmente adaptable y brindar escalabilidad sin la necesidad de requerir el cambio del cableado.
- Define un estándar que permite que el usuario móvil mantenga la conexión, entre cambios de APs (*Roaming*).

Por otro lado se tienen algunos aspectos a tomar en cuenta al escoger la implementación de una red inalámbrica:

---

<sup>7</sup> VPN: Consiste en generar una red privada a través de una red pública, generando políticas de seguridad y gestión de la red privada

- La red inalámbrica es más vulnerable a recibir ataques, debido a que cualquiera puede tener acceso al medio de transmisión (radiofrecuencia).

### 1.2.7 CONSIDERACIONES PARA DISEÑAR UNA RED INALÁMBRICA IEEE 802.11 [7]

Existen ciertos elementos que se debe tomar en cuenta al momento de estudiar los requerimientos para el diseño de la red inalámbrica, tales como:

- **Número máximo de usuarios:** Este parámetro, permite conocer la cantidad de usuarios que estarán conectados en la red para que ésta mantenga un desempeño óptimo, sin que ninguno de ellos pierda conectividad; mientras mayor sea la cantidad de usuarios por cada punto de acceso, disminuye el rendimiento de la red.
- **Tipos de Aplicaciones:** Definir las aplicaciones y servicios que van a correr en la misma; como por ejemplo abrir una página *web*, conexión a una base de datos, o con un servidor de archivos, transferir datos multimedia, aplicaciones que requieran una transmisión en tiempo real.
- **Velocidad de Transmisión:** Velocidad a la cual será transmitida la información desde el punto de acceso a la estación móvil, puede disminuir en función de las características propias del medio (en este caso el aire) y de obstáculos que pueden haber entre la conexión. Las velocidades dadas en cada uno de los estándares son teóricas, por lo que es imposible alcanzarla al momento de una conexión real entre los dispositivos.
- **Frecuencia de operación:** La frecuencia de operación definida para el estándar 802.11 está en los 2.4 GHz y en los 5 GHz; es necesario tomar en cuenta las regulaciones que existan en el país para el uso de estas frecuencias, aunque son frecuencias de libre acceso, al momento de

utilizarlas con fines comerciales, la regularización con respecto a cada país es muy importante.

- **Interferencia:** Dentro de la frecuencia en la que trabaja la red inalámbrica diseñada, existirán otras redes operando en la misma frecuencia, por lo que es necesario realizar un análisis de canales utilizados por cada una de las redes vecinas las cuales pueden ocasionar interferencia.
- **Área de Cobertura:** Es necesario analizar si el ambiente donde funciona la red es abierto o cerrado, si es en el interior de un edificio, oficina, casa (*indoor*), en este caso la señal se atenúa más fácilmente, por lo que puede ser necesario el uso de más puntos de acceso; o si es diseñada para un patio, un parque, o un enlace entre puntos externos (*outdoor*) en donde la señal tendrá mayor alcance, llegando hasta el máximo valor que defina el estándar a escogerse. Este factor es dependiente de la frecuencia de operación, por lo que si la red trabaja a 2.4 GHz el alcance es mayor alcance que al escoger la banda de los 5 GHz.
- **Material con el que están contruidos los edificios:** La propagación de la señal a través de muros o paredes depende del material de construcción, existen materiales que reflejan a las ondas (madera, tabla roca), por lo que facilitan la transmisión de las mismas; así como materiales que absorben y atenúan la señal (hierro, acero, cemento).
- **Conexión de la WLAN con una red cableada:** Tomar en cuenta la forma en la que se conectan los dispositivos de acceso a la red, y a una fuente de energía, ya sea mediante una instalación eléctrica exclusiva para los mismos, o mediante la característica que ofrece el estándar IEEE 802.3af (*Power over Ethernet*), la cual nos brinda energía mediante la conexión Ethernet hacia la red cableada.
- **Disponibilidad de productos en el mercado:** Una vez planteados todos los requerimientos para el diseño, y tomando las características necesarias

para un óptimo rendimiento de la red, se escogen los dispositivos que soporten dichos requerimientos.

### **1.3 LENGUAJE DE PROGRAMACIÓN JAVA**

Java es un lenguaje de alto nivel concurrente, orientado a objetos y basado en clases que marca dos elementos principales: un compilador y un intérprete.

El compilador genera los llamados *bytecodes* o códigos de bytes, los mismos que son almacenados en un fichero *.class* para luego ejecutarse por el intérprete o también denominado máquina virtual de java (*JVM*).

Los códigos de bytes son un conjunto de instrucciones correspondientes a un lenguaje de máquina, lo que garantiza que no dependerán de ningún procesador en específico sino de la máquina virtual directamente, permitiendo que cualquier programa desarrollado en este lenguaje corran en cualquier plataforma.

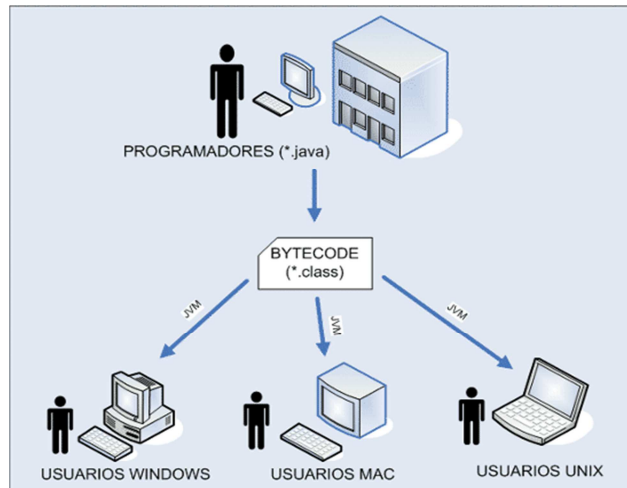
#### **1.3.1 CARACTERÍSTICAS DEL LENGUAJE JAVA**

##### **1.3.1.1 Independiente de la Plataforma [8]**

Una de las principales características de Java es su independencia del tipo de plataforma es decir no dependerá de la arquitectura como tal en la que se esté ejecutando el programa.

*Javac* o el denominado ("*java compiler*") es el compilador en java encargado de transformar el código que tiene una extensión *.java* en códigos de bytes y almacenarlo en ficheros *.class* los mismos que serán ejecutados en cualquier Sistema Operativo o dispositivo por medio de la "*java virtual machine*" o máquina virtual de java.

La figura 1.4 muestra la virtualización de Java.



**Figura 1.4:** Virtualización de Java [8]

### 1.3.1.2 Distribuido

Java es un lenguaje altamente distribuido por su versatilidad en manejo de diferentes contenidos y aplicaciones, desde la creación de páginas *Web* así como de programas convencionales, pero principalmente posee un conjunto de clases para trabajar sobre su contenido en la red denominados “sockets”, los mismos que generan un punto de comunicación entre procesos para emitir y recibir información.

### 1.3.1.3 Orientado a Objetos

Se refiere a la utilización de objetos para la programación. Un objeto es la instanciación de una clase para tener acceso a los métodos (operaciones que pueden realizarse sobre el objeto) y atributos (características que tiene el objeto) que pueden ser reutilizados en distintos casos.

## 1.3.2 ARQUITECTURA DE JAVA [9]

En la figura 1.5, se observa que la plataforma Java tiene tres bloques esenciales que son la API Java SE, el JRE (*Java Runtime Environment*) y el conjunto de Desarrollo Java (*JDK*).

El JRE contiene las librerías, la máquina virtual y demás componentes para la ejecución de las aplicaciones. Mientras que el JDK contiene al JRE y proporciona al usuario las herramientas necesarias para compilar y ejecutar los programas.

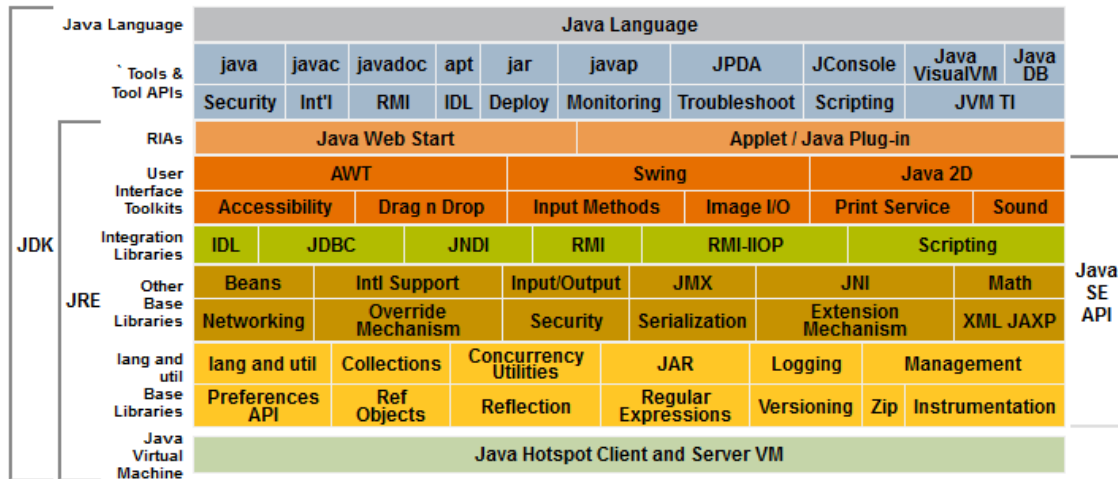


Figura 1.5: Esquema de la Arquitectura Java [9]

### 1.3.2.1 Máquina Virtual de Java [10]

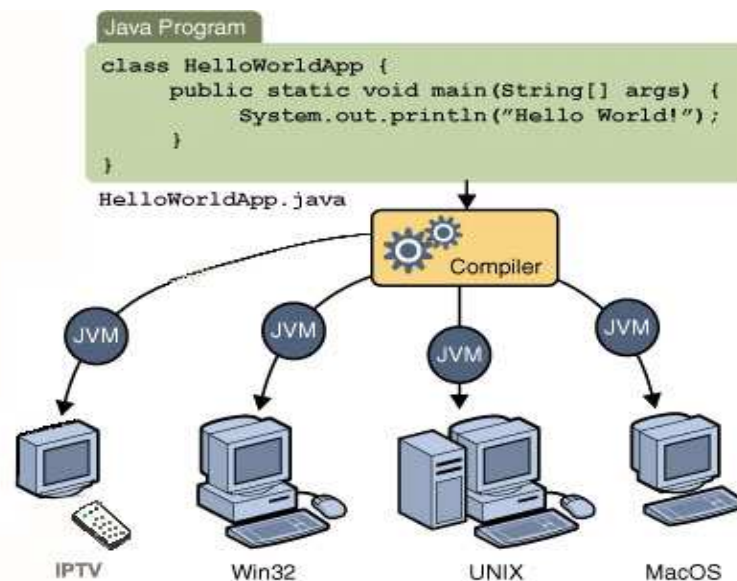
La Máquina Virtual de Java es prácticamente el emulador de un procesador a nivel de software encargada del procesamiento de los archivos `.class` los mismos que contienen `bytecodes` y su traducción a lenguaje de máquina es decir que es el componente dentro de la arquitectura que lo hace independiente de la plataforma en la que se esté implementando (figura 1.6), en tal virtud existe una máquina virtual específica para cada dispositivo.

Actualmente otro de los servicios que brinda la JVM es proporcionar un compilador JIT (*just in time*) el mismo que compila a petición de la aplicación ciertos archivos necesarios mejorando así su rendimiento y eficacia, así como también el denominado *Garbage Colector* o recolector de basura que nos permite recuperar recursos de memoria que no se esté utilizando.

### 1.3.2.2 Interfaz de Programación de Aplicaciones (API)

Una API dentro de Java es un conjunto de paquetes organizados en temas que contienen una gran cantidad de clases relacionadas al desarrollo de software en general.

Las diferentes ediciones de la plataforma Java proporcionan su propia API.



**Figura 1.6:** Máquina Virtual de Java [10]

### 1.3.3 PLATAFORMA JAVA [10]

La plataforma Java como tal presenta varias versiones clasificadas principalmente de acuerdo al hardware en el cual se va ejecutar y las necesidades de los usuarios.

- **Java Edición Empresarial (Java EE)**

Esta plataforma desarrolla aplicaciones en arquitectura distribuida por lo tanto se enfoca a servicios *web*, servicios de nombres, XML, autenticación así como también el manejo de transacciones y concurrencia; como APIs propias dentro de esta plataforma se tiene: *Enterprise JavaBeans*, *servlets*, *portlets*, *JavaServer Pages*, entre otras.

- **Plataforma Java Edición Estándar (Java SE)**

Esta plataforma básicamente permite el desarrollo de aplicaciones en servidores y todo tipo de ordenadores personales, como su objetivo es el desarrollo de aplicaciones para el usuario final, las APIs dentro de esta plataforma son multimedia, redes de comunicación, interfaz gráfica de usuario etc.

- **Plataforma Java Edición Micro (Java ME)**

Esta Plataforma permite el desarrollo de aplicaciones en dispositivos móviles como son teléfonos celulares asistentes personales digitales (PDAs), entre otros.

Básicamente con este tipo de plataforma se tendrá un entorno limitado en memoria, visualización, procesamiento, etc.

- ***Virtual Kilobyte Machine (KVM)***

Corresponde a la máquina virtual más pequeña desarrollada por Sun, está escrita en lenguaje C con aproximadamente 240000 líneas de código y diseñada para tener una carga de memoria entre 40 KB y 80 KB y sobre todo alta portabilidad.

- ***Compact Virtual Machine (CVM)***

Esta máquina virtual ya está diseñada para dispositivos con un poco más de procesamiento, donde básicamente ya tenemos mejoras en cuanto a memoria, recursos y sincronismo, así como también el soporte a hilos nativos e invocación a métodos remotos.

- **Configuración de dispositivos limitados de conexión (CLDC)**

Está orientado a dispositivos que ya pueden establecer conexiones aun cuando la capacidad gráfica y de memoria es limitada como por ejemplo teléfonos, PDAs, para este tipo de dispositivos aporta con las siguientes funcionalidades:



- Soporte de acceso a redes
- Soporte de E/S básica
- Un subconjunto de lenguaje Java y KVM o *kilobyte virtual machine*.

## 1.4 SISTEMA OPERATIVO ANDROID

Android contrariamente a lo que se tenía en cuenta fue desarrollada en sus inicios por la compañía Android Inc. en Palo Alto California, luego Google compró sus derechos en Julio del 2005; pero a partir del año 2008 Google lanza el Android SDK 0.9 beta.

Android se define como un conjunto de software desarrollado con los subsistemas necesarios para ofrecer una solución integral con un núcleo o kernel de Linux y basado la mayor parte en Java. Presenta las siguientes características:

- Aplicación *framework* que permite el reúso y reemplazo de aplicaciones.
- Soporte a tecnologías como Bluetooth, EDGE, 3G y WiFi.
- Soporte a hardware como: Cámara, GPS, acelerómetro.
- *Dalvik Virtual Machine* (DVM) máquina virtual de Java optimizada para dispositivos móviles.
- Integra un *browser* basado en código abierto.
- Base de datos *SQLite* para almacenamiento de datos.

### 1.4.1 ARQUITECTURA ANDROID [11]

La arquitectura de Android como se muestra en la figura 1.7, se puede dividir en cuatro capas de la siguiente manera:

- Aplicaciones.
- Estructura o armazón de aplicaciones.
- Librerías y ejecución.
- Núcleo o *kernel* de Linux.
- Núcleo o *kernel* de Linux.

En la base se tiene el núcleo o *kernel* de Linux versión 2.6 y 3.0; adaptado a las características necesarias para ser ejecutado sobre dispositivos móviles; éste actúa como una capa de relación entre el hardware y las demás capas de la arquitectura, donde encontramos los *drivers* de cada elemento que posee el teléfono relacionados con los dispositivos como audio, cámara, wifi, etc.

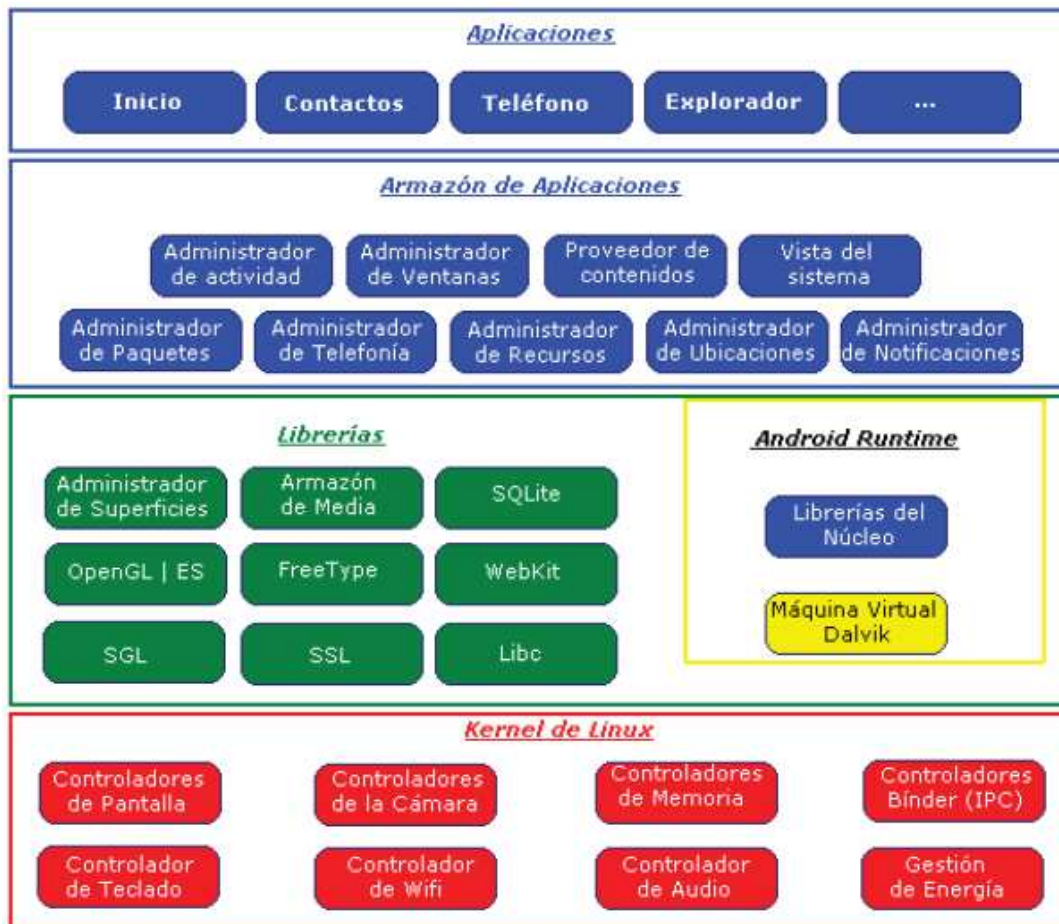


Figura 1.7: Arquitectura *Android* en Capas [11]

- **Librerías y ejecución:**

En esta capa, ubicada sobre el *kernel* la componen un conjunto de librerías nativas de Android, escritas en C y C++ y sirven para proporcionar funcionalidad a las aplicaciones que contienen tareas que se repiten con frecuencia. Entre las librerías que se encuentran en esta capa están:

- **Administrador de superficies:** Manejo de la pantalla y de las ventanas correspondientes a distintas aplicaciones activas.

- **Free Type:** Permite trabajar con distintos tipos de fuentes.
- **Libc:** Incluye librerías necesarias en el lenguaje C.
- **Media Framework:** Proporciona las herramientas necesarias para el manejo de multimedia.
- **SQLite:** Motor de base de datos, con la cual crea y gestiona datos de las aplicaciones.
- **OpenGL y SGL:** Mantienen la capacidad gráfica de Android, al manejar gráficos en 2D y 3D; pudiendo interactuar entre ambas características.
- **WebKit:** Suministra un motor para aplicaciones *web* y que utilizan un navegador, incluido por defecto en Android.
- **SSL:** Posibilita la utilización de comunicaciones seguras.

Junto a estas librerías se encuentra el entorno de ejecución, el cual incluye las librerías del núcleo, y la máquina Virtual Dalvik, que utilizando el mismo concepto de la máquina virtual de Java, y dado las limitaciones que se tiene en procesador y memoria RAM, genera ficheros Dalvik ejecutables (.dex). Cada aplicación corre en su propio proceso Linux con una instancia de la máquina virtual.

- **Estructura o armazón de aplicaciones [12]:**

Sobre las librerías se encuentra una estructura que brinda un contexto para desarrollar, este *framework* permite aprovechar un sistema de clases y servicios ya construidos, necesarios para ejecutar las aplicaciones, éstas son:

- El Administrador de Actividades (*Activity Manager*) que se encarga de administrar la pila de actividades de la aplicación y el ciclo de vida de la misma.
- Administrador de ventanas (*Windows Manager*) que organiza lo que se mostrará en la pantalla.
- Proveedor de contenido (*Content Provider*) que encapsula los datos que compartirán entre aplicaciones.
- Sistema de Vistas (*Views System*) que son elementos que construyen las interfaces de usuarios como botones, listas o del mismo navegador *web*.
- Administrador de paquetes (*Package Manager*) que permite obtener información de los paquetes instalados en el sistema operativo, también gestiona la instalación de nuevos paquetes, estos contienen el archivo *.apk* (instalador de aplicaciones) los archivos *.dex* que facilitan la descarga e instalación de aplicaciones.
- Administrador de telefonía (*Telephony Manager*) que cumple con administrar llamadas, mensajes de texto o multimedia.
- Administrador de Recursos (*Resource Manager*) que gestiona los elementos que forman parte de la aplicación y que están fuera del código.
- Administrador de Posición (*Location Manager*) que permite determinar la posición geográfica del dispositivo mediante GPS si lo posee u otras redes.

- Administrador de Notificaciones (*Notification Manager*) que engloba los servicios de notificación a los clientes mediante alertas en la barra de estado.

- **Aplicaciones**

La capa Aplicación interactúa directamente con los usuarios permitiéndoles el manejo de contactos, de navegadores o *browser*, juegos, etc.

Los bloques básicos de una aplicación son: actividades, *intents*, vistas, servicios, proveedor de contenido, receptores de difusión, manifiesto (figura 1.8).



**Figura 1.8:** Componentes básicos de una Aplicación [12]

- **Actividades**

Cualquier aplicación en Android está formada por una o varias pantallas, cada uno de estos elementos corresponden a una actividad. La función principal es la creación de la interfaz del usuario. Una actividad puede lanzar otras actividades para obtener una aplicación con varias interfaces, que faciliten la interacción entre los datos y el usuario.

- **Intents**

Son mensajes del sistema que corren en el dispositivo y notifican a las aplicaciones de los eventos que se están generando (llegada de un SMS, notificación de llamada, inserción de una tarjeta SD, actividades lanzadas desde el menú principal). Los componentes que se están generando pueden ser internos o externos de la aplicación.

- **Vistas**

Son todos los elementos que pertenecen a la interfaz de una aplicación, tales como un botón o una caja de texto, lo habitual es definir las vistas en un fichero XML, para que el objeto sea creado a través de este fichero. Esta definición es similar a la definición de una página *web* utilizando código HTML.

- **Servicios**

Un servicio es un proceso o tarea ejecutada en segundo plano que no posee una interfaz de usuario. Son de larga duración, por lo que correrán independientemente a cualquier actividad. Un ejemplo muy común es el reproductor de música, que correrá mientras el usuario realiza otra opción, como enviar un mensaje de texto.

- **Proveedores de Contenido (*Content Providers*)**

Gestiona un conjunto de datos que serán utilizados en una aplicación, los cuales pueden estar embebidos en la misma aplicación, puede ser llamado desde una base de datos SQLite, en la *web*, o en cualquier otro lugar de almacenamiento a donde se pueda acceder. Un ejemplo muy común de un proveedor de contenido, es la lista de contactos almacenada en el dispositivo.

- **Archivo de Configuraciones (*Android.Manifest*)**

El archivo de configuraciones es un componente XML en el que se describen los componentes de la aplicación, tales como las actividades,

intenciones, servicios, librerías, la versión mínima de Android para correr la aplicación, los permisos necesarios.

- **Receptores de Difusión (*Broadcast Receivers*)**

Son componentes que responden a avisos y anuncios de difusión. Un ejemplo es un evento de batería baja, un mensaje de texto recibido, llamada entrante. No presentan una interfaz de usuario, aunque algunas veces utilizan una barra de progreso para mostrar avances o íconos en la barra de estado.

#### 1.4.2 VERSIONES DISPONIBLES DE ANDROID [13]

Este sistema operativo, en conjunto con los teléfonos móviles y *tablets*, han evolucionado rápidamente, desde la 1.0 que fue la primera versión comercial, hasta la versión 4.4, recientemente salida al mercado.

- ***Cupcake*: Versión de Android 1.5**

Presenta transiciones en las pantallas animadas, *widgets*, posibilidad de grabar y reproducir videos, subir videos a Youtube, e imágenes a Picasa, teclado con predicción de texto.

- ***Donut*: Versión de Android 1.6**

Presenta mejoras en la búsqueda por voz, búsqueda mediante marcadores, historiales, contactos y páginas *web* desde la pantalla de inicio, mejora en el *Android Market*.

- ***Eclair*: Versión de Android 2.0/2.1**

Para esta versión se optimiza la velocidad de hardware, soporta pantallas de mayor tamaño y mejor resolución, mejora en el navegador y la compatibilidad con HTML5, dispone de Zoom Digital, teclado virtual mejorado.

- **Froyo: Versión de Android 2.2**

Presenta mejoras en la velocidad de las aplicaciones, incluye *hostpot Wifi*<sup>8</sup>, una cuenta de *Microsoft Exchange*<sup>9</sup>, marcación por voz, transmisión de contactos mediante *Bluetooth*, soporta la instalación de aplicaciones en la memoria expansible.

- **Ginger Bread: Versión de Android 2.3**

Se incluye el soporte para pantallas extra grandes y de alta resolución, nuevos efectos de audio como ecualización, virtualización de auriculares y refuerzo de los graves, soporte para pagos mediante NFC (*Near Field Communications*)<sup>10</sup>, administrador de descargas para poder descargar archivos grandes.

- **Honey Comb: Versión de Android 3.0/3.4**

En esta actualización se mejora el soporte para *tablets*, acceso a un escritorio 3D con *widgets* rediseñados, *videochat* mediante *Google Talk*, añade soporte para accesorios conectados a través de la conexión USB, los *widgets* pueden personalizarse de forma manual.

- **Ice Cream Sandwich: Versión de Android 4.0**

Esta versión unifica el uso de cualquier dispositivo, una *tablet*, celular, los *widgets*<sup>11</sup> son redimensionables, se puede bajar la barra de notificaciones con el dispositivo bloqueado, incluye reconocimiento facial, las carpetas son más fáciles de crear y administrar, contiene una interfaz limpia y moderna, el corrector de texto es mejorado, ofreciendo la opción a tocar una palabra y visibilizar una lista de las opciones de edición y palabras similares.

---

<sup>8</sup> *Hotspot Wifi*: servicio que permite a un dispositivo mantenerse conectado a la red inalámbrica cuando existe alta demanda de tráfico

<sup>9</sup> *Microsoft Exchange*: servicio de correo electrónico propietario de Microsoft.

<sup>10</sup> NFC2: servicio de control de acceso y fidelidad en la compra de productos y servicios en línea

<sup>11</sup> *Widgets*: pequeñas aplicaciones cuyo objetivo es presentar información, dentro de *Android* podemos citar por ejemplo *widgets* que muestran el clima.



- **Jelly Bean: Versión de Android 4.1/4.2/4.3**

Ofrece un ajuste automático de los *widgets* en el escritorio, dispone de un dictado por voz mejorado sin necesidad de una conexión a Internet, notificaciones mejoradas, soporte para pantallas inalámbricas, incluye la posibilidad de ir directamente a la cámara, cifrado de aplicaciones, *Google Chrome* se convierte en el navegador por defecto, dispone de una nueva función *Sound Reach* que permite saber que música se está escuchando.

- **Kit Kat : Versión de Android 4.4**

En esta versión de Android se reduce el consumo de batería al utilizar cualquier aplicación, permite la grabación de todo lo que está pasando en la pantalla, muestra el estado de la batería en porcentaje, reduce el consumo de memoria RAM, pudiendo ser utilizado con 512 MB, en la arquitectura, se vuelve no indispensable el uso de la máquina virtual de *Dalvik*, siendo sustituido por nuevas versiones, el manejo de correo dispone de carpetas anidadas, fotos de contactos y una mejor experiencia de navegación, permite el control de dispositivos como un televisor, a través del uso de rayos infrarrojos.

## 1.5 HERRAMIENTAS PARA EL DESARROLLO [13]

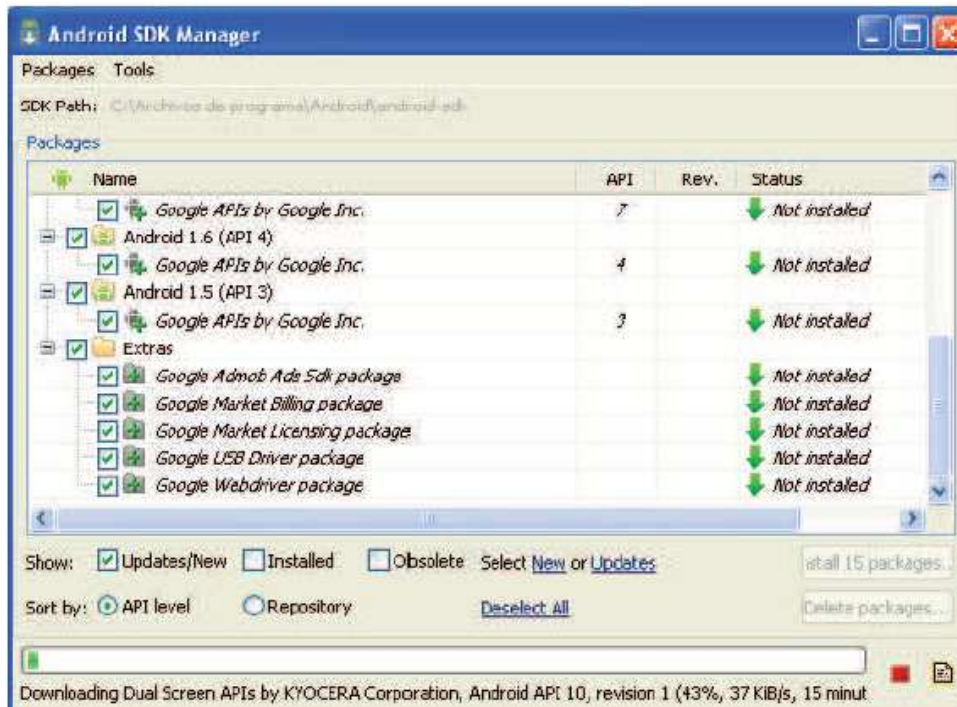
Para el diseño y la programación de la aplicación es necesario disponer de una versión de *Eclipse* para poder desarrollar el proyecto. Además se necesita el kit de desarrollo de software para Android SDK (*Software Development Kit*).

El mismo que se encuentra en varias versiones y para diferentes plataformas en la siguiente página *web*:

- “<http://developer.android.com/sdk/index.html>”

Una vez que se inicia con la instalación del mismo, el instalador comprueba si el equipo dispone del *Java SE Development Kit* (JDK), en caso de no tenerlo, devolverá un mensaje con un *link* desde donde se lo obtiene.

Cuando finalice la instalación, se ejecutará el *SDK manager*, en el que se deberán seleccionar todas las casillas deshabilitadas, para instalar todas las versiones de Android así como sus herramientas.



**Figura 1.9: SDK Manager [13]**

Cuando las herramientas hayan sido instaladas en su totalidad, se procede a abrir eclipse e instalar el ADT (*Android Developer Tools*) *Plugin* en el entorno de desarrollo mediante la utilización de las siguientes instrucciones:

- Clic en “*Help*” y seleccionar “*Install New Software*”.
- Seguir las instrucciones del *wizard* habilitado, presionar “*Add*” para añadir un Nuevo *plugin*.
- Buscar el *ADT Plugin* e incluir la siguiente dirección *URL*:
  - <https://ddl-ssl.google.com/android/eclipse/>

En esta página se encuentran los repositorios para la instalación del complemento necesario para iniciar con la programación.

- Una vez cargado el listado del software disponible, seleccionar la opción “*Developer Tools*” la cual contiene las herramientas de desarrollo necesarias.
- Se procede a aceptar el acuerdo de licencia y se procede con la descarga de las herramientas, una vez culminado el proceso, es necesario reiniciar eclipse para visualizar las nuevas opciones descargadas.

## **1.6 EMULADOR DE TABLET**

El emulador que proporciona el entorno de desarrollo Eclipse permite tener corriendo a la aplicación de manera independiente y de esta manera controlar los errores existentes en la aplicación; la figura 1.11 muestra la vista de un emulador para la versión 2.2 de Android.

Al momento de tener un proyecto listo para ejecutar, se lo corre como una Aplicación de Android, entonces se lanzará el emulador más apropiado.

Una vez iniciado el emulador, es recomendable no cerrarlo hasta terminar con las pruebas de la aplicación, ya que necesita muchos recursos de memoria y procesamiento para volver a iniciarse, lo cual disminuye la eficiencia del computador. Si se genera un nuevo ejecutable de la aplicación programada, éste se reinstalará automáticamente en el emulador.



**Figura 1.10:** Emulador para Android 2.2 [13]

## 1.7 FUNCIONALIDAD DE UNA TABLET

Actualmente las tabletas se han convertido en uno de los dispositivos más utilizados, principalmente por su versatilidad, tamaño y diseño a continuación la tabla 1.1 muestra la funcionalidad de las características de una *tablet*.

CARACTERÍSTICA	DESCRIPCIÓN
Herramientas para control de servicios	GPS, acelerómetro, brújula digital, estado de red, reproducción de contenido multimedia, notificaciones, animaciones, almacenamiento interno
Internet Móvil	Maneja web versión 2 para mayor interactividad multimedia, documentos, redes sociales, compatibilidad con plataformas Moodle y Joomla
Aplicaciones Apps	Nativas: Desarrolladas específicamente para una <i>Tablet</i> , permitiendo aprovechar al máximo sus características Híbridas: Emulan a páginas <i>web</i> pero adaptadas al tamaño del dispositivo
Realidad aumentada	Soporte de elementos de diseño en 3D y 2D
Sistema Operativo Android	El código de Android es abierto, permitiendo desarrollar gratuitamente aplicaciones
Multitarea	Función y gestión de varias aplicaciones a la vez dejándolas en modo suspensión, evitando el uso excesivo de memoria

**Tabla 1.2:** Funcionalidad de una *tablet*

# CAPÍTULO 2

## 2. DISEÑO DE LA RED INALÁMBRICA DEL CENTRO COMERCIAL ESPIRAL

### 2.1 SITUACIÓN ACTUAL DEL CENTRO COMERCIAL ESPIRAL

#### 2.1.1 ANTECEDENTES

El Espiral es el primer centro comercial construido en la ciudad de Quito en el año de 1980, se encuentra ubicado en el centro norte de la capital, en la calle Jorge Washington número N 20-80 y Av. Amazonas.

Tiene una forma de espiral, cuenta con 11 plantas distribuidas de la siguiente manera:

- Dos subsuelos destinados a zona de parqueaderos.
- Dos subsuelos y siete niveles (numerados de 0 a 6) en donde están distribuidos un total de 160 locales comerciales. Cada uno de ellos dispone de los servicios básicos de energía eléctrica, agua potable y teléfono.

Los locales indistintamente brindan servicios en las áreas de tecnología, joyería, bisutería, calzado, ropa, café y bar, peluquería, agencia de viajes, entre otros, los cuales se detallan en la tabla 2.1.

N°	Nombre Comercial	Descripción	Host
2	ESTUDIO CERO	Venta de CDs	1
3	MARCES PLACE CIA LTDA	Venta computadores	2
5	DIGITAL GAMES	Comercio venta de equipos electrónicos	1
7	PALOMINO	Venta zapatos ortopédicos	1
8	HOUSE OF STYLE	Cosmetología y tatuajes	1
9	LOCAL DESOCUPADO	_____	1

**Tabla 2.1:** Locales y computadores en el Centro Comercial El Espiral (1 de 6)

N°	Nombre Comercial	Descripción	Host
11	GARAJE	_____	—
12	NXA	Tecnología	2
15	PUNTO SISTEMA	Tecnología	2
16	LOCAL DESOCUPADO	_____	—
17	CIA. MARKET	Cuñas y producciones	2
18	SALA COMUNAL	_____	1
19	VIDEO MUSICAL	Venta de CDs	—
20	COMPUQUITO	Venta de suministros y juegos	1
21	POWER NET	Venta de suministros de computadoras	1
22	LABORATORIO NOVICOMPU	Servicio técnico	2
23	LOCAL DESOCUPADO	_____	—
24	POWER NET	Venta de suministros de computadoras	1
25	GIGASOFT	Venta de software	3
26	LABHU SHAH BOUTIQUE	Artesanía Indu	—
27	LABHU SHAH BOUTIQUE	Artesanía Indu	—
28	THE SERIES STORE	Venta de videos	1
29	D&S	Estudio jurídico	1
30	PELUQUERÍA ESPIRAL	Peluquería unisex	
31	AQUINO OPTICA	Óptica	2
32	MOVIE ZONE	Venta de películas y video juegos	1
33	UNIVERSAL TECHNOLOGY	Venta de equipos de almacenamiento magnético	1
34	COMPUKIT	Venta computadoras	2
35	SAGATUR	Agencia de viajes	3
36	DIGITAL GAMES	Juegos en red	1
37/38	SACA LA RESACA	Restaurante	—
39	BIPPACELL	Telefonía celular	1
40	CSV SUMINISTROS	Papelería suministros de oficina	2
41	KONAMI GAMES	venta de video juegos	1
42	MAREA NEGRA RESC.	Ropa	1
43	PELUQUERIA LIZ UNISEX	Peluquería	—
44	CABANA EXPRESHION	Distribución de tarjetas	1

**Tabla 2.1:** Locales y computadores en el Centro Comercial El Espiral (2 de 6)



N°	Nombre Comercial	Descripción	Host
45	ENIGMA TATOO ESTUDIO	Tatuajes cosméticos y artísticos	2
46	MEGA UNIFORMES	Comercialización de uniformes	3
47	RCM SISTEMAS	Diseño gráfico	2
48	MEGA UNIFORMES	Comercialización de uniformes	—
49	CASA BOLIVIA	Artículos promocionales	—
50	CASA BOLIVIA	Artículos promocionales	
51/52	SALA DE BELLEZA MAXBER	Sala de Belleza	1
53	PRONATU'S & SENSUAL SHOP	Centro Naturista	1
54/55 /56	PRO DJ	Venta de equipos de audio profesional	2
57	JN MODA Y MODA	Ropa	1
58	CORE COM	Cyber internet	1
59	COLLAGE SACOS	Venta de ropa	1
60	ARAMIS	Venta de artículos de cuero	—
61	JM ROPA PARA NIÑOS Y TARJETERÍA	Tarjetería y algo mas	—
62	ALMACENES CHIMBORAZO	Artículos de cuero	—
63	JOYERÍA CEDILLO	Joyería y afines	1
64	CAMISERÍA INGLESA	Ropa de vestir para caballeros	1
65	MAQUEÑO REPUBLIK	Venta de camisetas y accesorios	1
66	LA MINA	Venta de joyas de plata	1
67	PERSONALÍZATE	Diseños para estampados de camisetas	1
69	JOYERIA AMBAR	Joyería	1
70	BOUTIQUE MODELS	Venta de ropa para dama	2
71	BRUSEHNKO	Moda informal	—
72	JOYERIA GRIJALVA	Joyería	—
73	POLI'S	Boutique	1
74	POSDATA	Tarjetería	
75	MOVIE LIGHT	Venta de videos	1
76	LABHU SHAH BOUTIQUE	Artesanía indu	1
77	TIAN SPORT	Venta calzado	1

**Tabla 2.1:** Locales y computadores en el Centro Comercial El Espiral (3 de 6)

N°	Nombre Comercial	Descripción	Host
78	AMERICAN GAMES	Venta de juegos <i>Play Station</i>	1
79	LABHU SHA	Ropa Indú	—
80	AUDIO CENTER	Venta de CDs y Videos Originales	1
81	LEVEL UP	Venta de máquinas y accesorios de video juegos	1
82	ARENA ROPA INTIMA	Lencería	—
83	FABICUERO	Elaboración y venta artículos de cuero	1
84	ALMACENES DALI	Venta de ropa	—
85	C.C.S. S.A.	Telefonía celular	2
86	TAMASHAN	Bazar	1
87	ANALUISA JOYEROS PLATERIA	Taller de joyería y platería	—
88	MAQUEÑO	Bodega	2
89	START PLATERIA & SEMIJOYA	Venta de semijoya y platería	1
90	THE HOUSE ROCK	Artículos para el <i>rock</i>	1
91	BORDADOS SARAI	Bordados	1
92	NOVEDADES ROSE BAUTISTA	Ropa causal e informal	—
93	JOYERIA AMATISTA	Venta de joyas de oro	1
94	EL MAGICO BAZAR	Bazar	2
95	DE ONDAS	Venta prendas de vestir	—
96	PUNTO EXTREMO SKATE SHOP	Venta de zapatos y accesorios de <i>skateboard</i>	1
97	HELEN NOVEDADES	Venta de gafas	1
98	CELL MEP	Telefonía celular	1
99/100	LATIN TRAVEL	Correo	4
101	D. XAEL	Venta de prendas de vestir	—
102	FACHAS	Ropa casual	1
103	INTERNET	Internet	1
104	COMERCIALIZADORA ÉXITO	Ropa y productos publicitarios	1
105	ECUAVENTURA	Agencia de viajes , turismo recepción	2
106	DA TECHNOLOGY	Venta de celulares	1
107	LA TORRE	Ropa de caballero	1

**Tabla 2.1:** Locales y computadores en el Centro Comercial El Espiral (4 de 6)

N°	Nombre Comercial	Descripción	Host
108	TOPY SPORT	Venta de prendas de vestir	1
109	ENSUEÑOS KAREN	Bazar	—
110	ANIME	Venta de videos	1
111	DIGITAL PRODUCCIONES	Audio y tecnología	1
112	CONSUHID COURIER	Courier internacional	1
113	COBO IMPORTACIONES	Venta de impresoras y suministros de computación	1
114	MEGA POSTER	Impresiones	2
115	CITYCELL	Comercialización de celulares	1
116	LATIN TRAVEL	Courier	2
117	PERRO LOCO	Venta de juegos	1
118	PERSONALIZATE	Diseños para estampados de camisetas	2
119	TABOO	Venta de accesorios y bisutería	1
120	PUNTO DIGITAL	Tecnología	1
121	BILL	Juegos en red	2
122	NMA	Impresiones CDs camisetas	2
123	JOHANNA PAULINA GONZALES	Venta de ropa juvenil	2
124	CAPCOM	Venta de videojuegos	—
125	KAAS	Venta de <i>Blue Ray</i> - juegos pc	1
126	"ARAMIS" EL CUERO	Artículos de cuero	—
127	CAMARA ARTESANAL	Oficina de servicios	1
128	SP GLOBAL CENTER	Estética	2
129	PORTA	Reparación técnica de celulares	1
130	ALL TIME SPAI- SONS	Distribuidora de cosméticos	1
131	GAMA CENTER SOLUTIONS	Servicio técnico exprés de consolas de video juegos	1
132	PLAY IT LOUD JUEGOS	Juegos electrónicos	—
133	PLAY IT LOUD JUEGOS	Juegos electrónicos	—
134	EC ROBOTICS	Diseño electrónico, robótica móvil, venta de computadores	2
135	CYBERCOMP	Venta de computadoras y accesorios, servicio técnico	2

**Tabla 2.1:** Locales y computadores en el Centro Comercial El Espiral (5 de 6)

N°	Nombre Comercial	Descripción	Host
136	SOLTEC	Venta de impresoras y suministros de computación	3
137	FOLKLORE GIORGIO	Venta de videos	1
138	GRAFICAR PUBLICIDAD	Diseño gráfico y sistemas	4
139	KARLA TATOO	Perforaciones y tatuajes	2
140	ECUAVISION	Productora de televisión	3
141	ALTO TRÁFICO	Video juegos	1
142	ALTO TRÁFICO	Video juegos	1
143	EL MUNDO DEL ROCK	Venta de música	2
144	YANBAL	Venta de cosméticos	4
145	EXOTIC THAI	Ropa y bisutería	—
146	MULTI GAMES	Venta de repuesto y máquinas	3
147	YANATEX	Importación de telas	—
148	ADMINISTRACION		4
149	CORPORACIONES UNIDAS	Envío de encomiendas y giros	2
150	VIDEO ZOOM	Venta de películas	—
151	UNIVERSAL TECHNOLOGY	Oficinas y departamento técnico	—
152	EROVER PC	Venta de computadoras y accesorios, servicio técnico	7
153	COMPUTADORAS		
154			
155	MULTINATION CORP	Venta de impresoras y suministros de computación	1
156	VIZZON	Venta de Semijoyas	2
157	SEMIJOYAS	Venta de Semijoyas	1
158	BAILA BOYS	Juegos de <i>Play Station</i>	—
159	RING SIDE	Accesorios de lucha libre	—
160	MULTICOSAS	Mini Market	—
<b>LOCALES OCUPADOS</b>			<b>155</b>

**Tabla 2.1:** Locales y computadores en el Centro Comercial El Espiral [14]

Como se puede observar existen 155 locales ocupados, y en ellos se encuentran funcionando cada una de las tiendas.

### **2.1.2 NECESIDADES DEL CLIENTE**

Actualmente la tendencia de la utilización de dispositivos que se conecten a una red y dispongan de una salida hacia la Internet es una necesidad que tanto los propietarios, como los clientes y visitantes de cada local comercial. Los clientes buscan la forma óptima de encontrar información de lo que requieren, mientras que los propietarios de los locales están obligados a difundir sus productos y servicios, ganando así más clientes en sectores que antes no se podía llegar.

Al visitar el Centro Comercial Espiral se observó que no dispone de una red propia, ni de cableado estructurado; además la arquitectura propia del establecimiento, dificulta que ésta sea diseñada sin tener que realizar un cambio arquitectónico del mismo. Lo que incurriría a entrar en gastos altos para el centro comercial.

Además, tomando en cuenta que tanto clientes como propietarios disponen de dispositivos móviles inalámbricos, como laptops, *tablets*, celulares, e incluso computadores de escritorio, buscan tener acceso inalámbrico a una red de datos, es necesario diseñar una red inalámbrica.

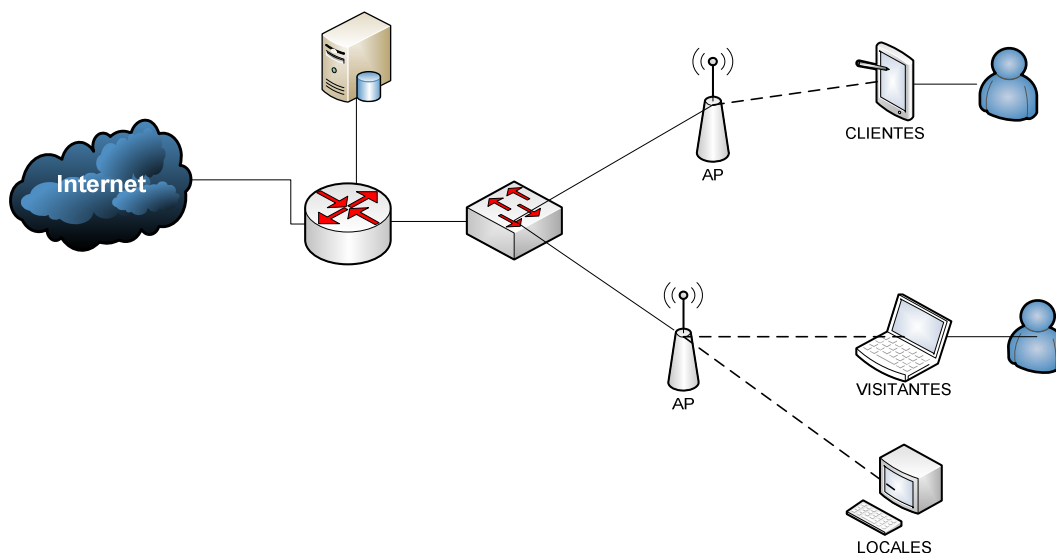
Con esto se logrará también la comunicación entre la aplicación y la base de datos que contendrá información de los locales.

Cada local comercial actualmente dispone de una conexión a internet a través, de conexiones DSL o con módems inalámbricos con tecnologías 3.5G y 4G; cuyo servicio es entregado por CNT, Movistar y Claro respectivamente.

Por lo que se plantea la solución que se presenta en la Figura 2.1.

El mecanismo para obtener las necesidades del cliente con respecto al uso de internet y de tener acceso a una aplicación de publicidad, fue utilizando fuentes de

información primarias; visitando el centro comercial y consiguiendo información directamente con los involucrados, a través de entrevistas y encuestas a un grupo de locales y a cierta cantidad de clientes y visitantes, determinando así ciertos parámetros necesarios para el diseño de la red inalámbrica.



**Figura 2.1:** Esquema de la red Inalámbrica para el Centro Comercial El Espiral

La encuesta se realizó con preguntas de opción múltiple, facilitando el proceso de análisis de la información recolectada. Ésta se realizó en el transcurso de una semana, durante el horario de atención (9:30 a 19:30).

Se realizó dos tipos de encuestas; una para los clientes y visitantes del centro comercial, y otra para los propietarios de cada uno de los locales.

Para determinar el número de encuestas para los clientes, se utilizó la fórmula de muestra aleatoria simple con población infinita<sup>12</sup>; debido a que los visitantes del establecimiento no son en un número exacto sino variable; como muestra la ecuación 2.1.

<sup>12</sup> Muestreo en donde cada unidad tiene la probabilidad equitativa de ser incluida en la muestra. Al ser de población infinita quiere decir que ésta no tiene un valor definido.

En donde:

- $n$  = Muestra (número de locales y visitantes a encuestar).
- $Z_{\alpha}$  = Grado de confiabilidad o nivel de confianza.
- $p$  = Probabilidad de ocurrencia (grado de validez de la respuesta).
- $q$  = Probabilidad de no-ocurrencia.
- $d$  = Precisión (depende del investigador, Costo y tiempo)

$$n = \frac{Z_{\alpha}^2 p q}{d^2}$$

**Ecuación 2.1:** Muestreo población infinita [15]

Al no disponer de un estudio anterior, se maximiza la probabilidad de ocurrencia  $p$  al 50%.

Los valores utilizados para las encuestas de los locales comerciales fueron los siguientes:

- $Z_{\alpha} = 1.960$ ; valor de  $Z$  más utilizado (nivel de confianza 95%)
- $p = 50\%$ ;
- $q = 50\%$ ;
- $E = 5\%$  para garantizar que los resultados de la encuesta sean válidos en el diseño.

Para este caso se tiene:

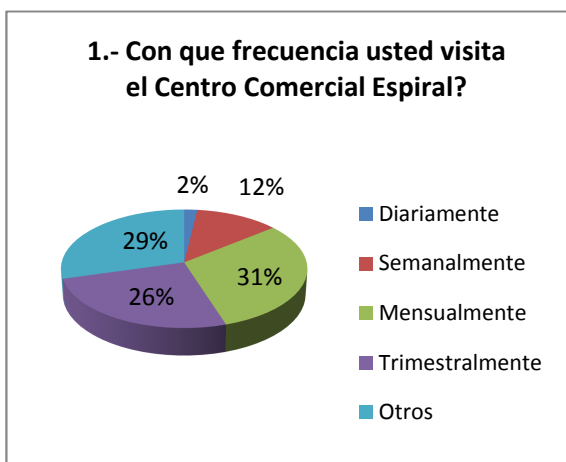
$$n = \frac{1.96 * 1.96 * 0.5 * 0.5}{0.05 * 0.05}$$

$$n = \frac{0.964}{0.0025}$$

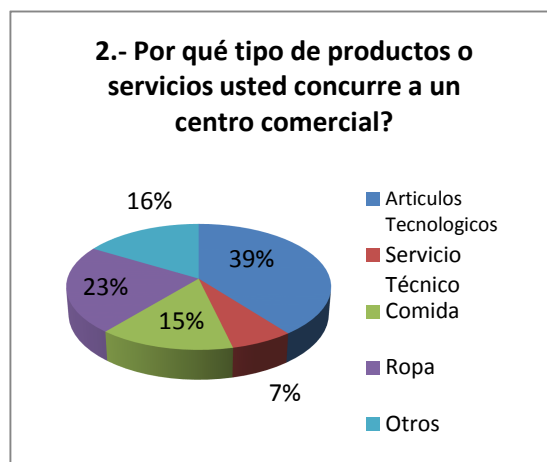
$$n = 384.16$$

Esto indica que el número de clientes a encuestar es de 385.

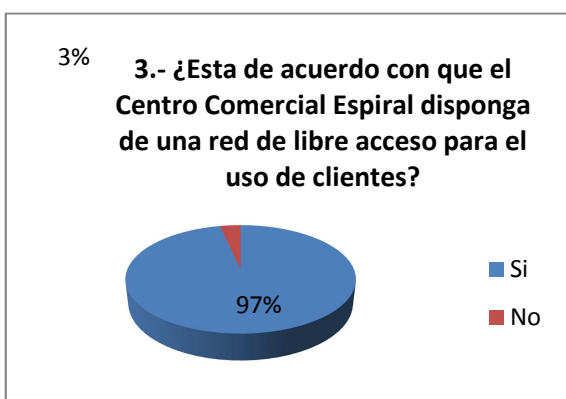
Los resultados obtenidos de las encuestas a los clientes del Centro Comercial se muestran en las figuras 2.2 a 2.12.



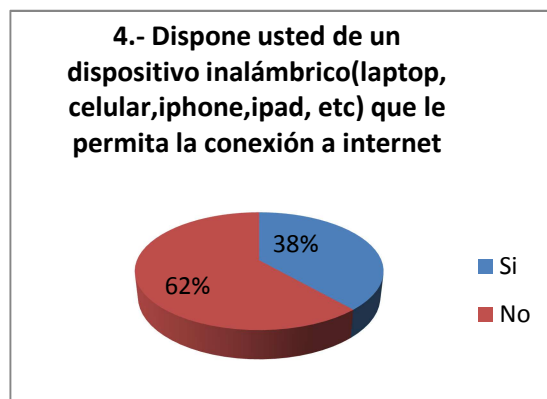
**Figura 2.2:** Resultados de la Pregunta 1 – Encuesta Visitantes



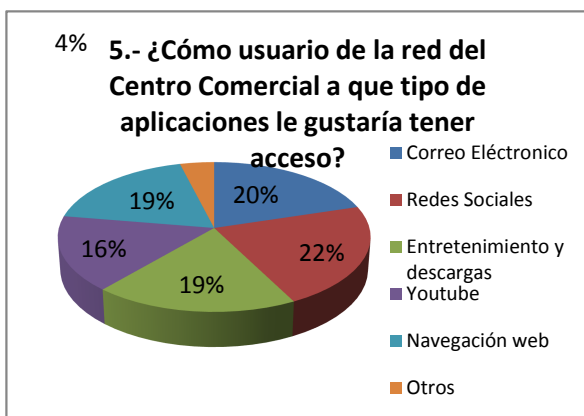
**Figura 2.3:** Resultados de la Pregunta 2 – Encuesta Visitantes



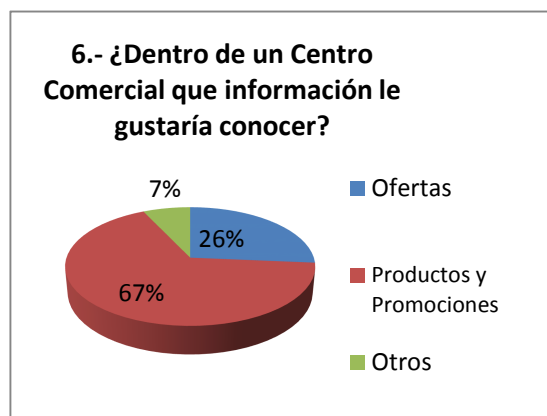
**Figura 2.4:** Resultados de la Pregunta 3 – Encuesta Visitantes



**Figura 2.5:** Resultados de la Pregunta 4 – Encuesta Visitantes

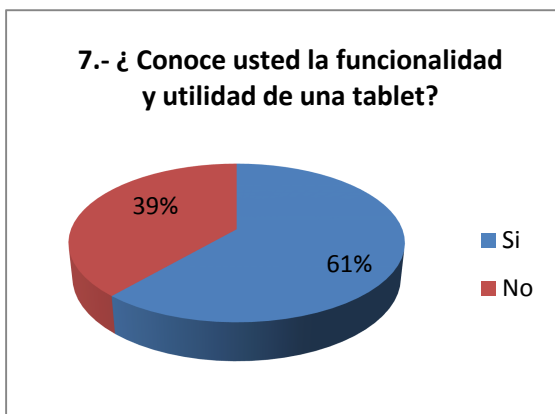


**Figura 2.6:** Resultados de la Pregunta 5 – Encuesta Visitantes

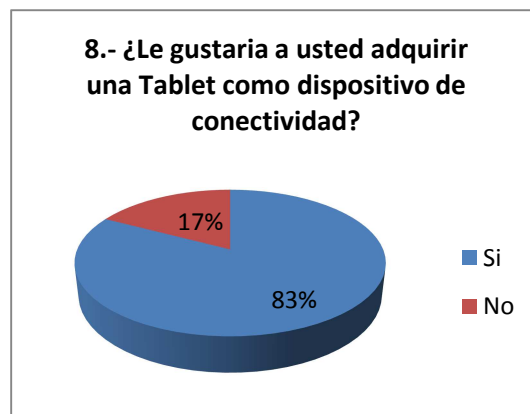


**Figura 2.7:** Resultados de la Pregunta 6 – Encuesta Visitantes

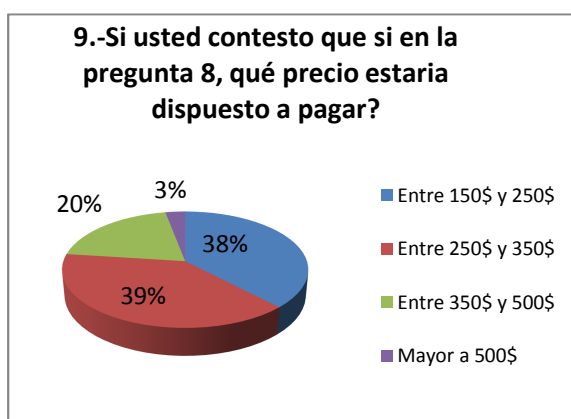




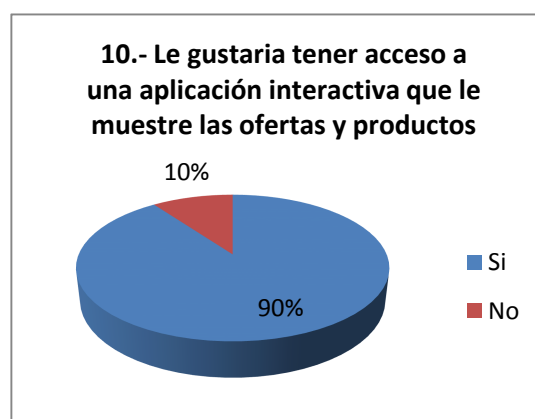
**Figura 2.8:** Resultados de la Pregunta 7 – Encuesta Visitantes



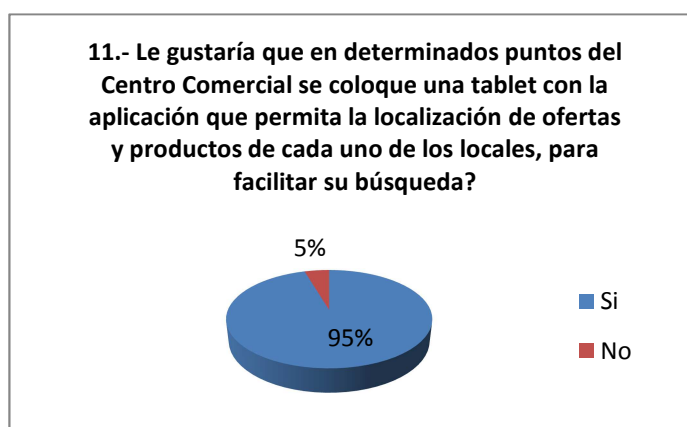
**Figura 2.9:** Resultados de la Pregunta 8 – Encuesta Visitantes



**Figura 2.10:** Resultados de la Pregunta 9 – Encuesta Visitantes



**Figura 2.11:** Resultados de la Pregunta 10 – Encuesta Visitantes



**Figura 2.12:** Resultados de la Pregunta 11 – Encuesta Visitantes

Para determinar el número de encuestados en los locales comerciales se utilizó el muestreo aleatorio simple de población finita que se indica en la ecuación 2.2.

$$n = \frac{Z_{\alpha}^2 N p q}{d^2 (N-1) + Z_{\alpha}^2 p q}$$

**Ecuación 2.2:** Muestreo población finita [15]

En donde:

- $n$  = Muestra (número de locales y visitantes a encuestar).
- $Z_{\alpha}$  = Grado de confiabilidad o nivel de confianza.
- $p$  = Probabilidad de ocurrencia (grado de validez de la respuesta).
- $q$  = Probabilidad de no-ocurrencia.
- $d$  = Precisión (depende del investigador, Costo y tiempo)
- $N$  = Universo (número total de locales y visitantes).

Los valores utilizados para las encuestas de los locales comerciales fueron los siguientes:

- $S = 2$ ; valor utilizado en poblaciones entre 10 y 10000.
- $N = 160$ ; número total de locales comerciales dentro del Espiral.
- $p = 50\%$ ; valor máximo utilizado al no tener estudios anteriores
- $q = 50\%$ ;
- $E = 5\%$  para garantizar que los resultados de la encuesta sean válidos en el diseño.

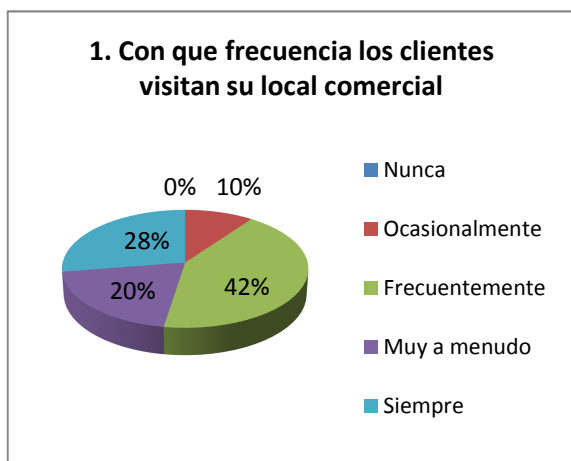
Para este caso se tiene:

$$n = \frac{1.96 * 1.96 * 160 * 0.5 * 0.5}{0.05 * 0.05 (160 - 1) + 1.96 * 1.96 * 0.5 * 0.5}$$

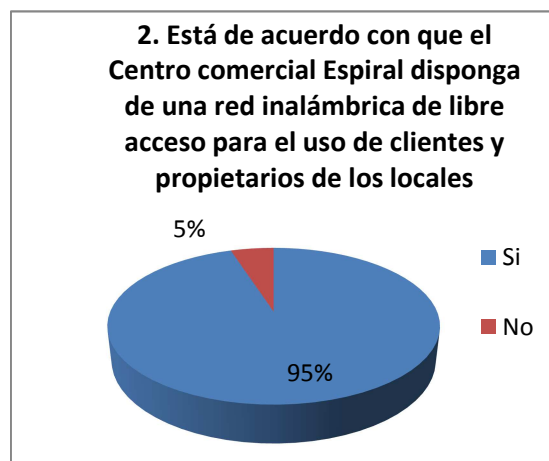
$$n = \frac{153.664}{1.3579} = 113,16$$

De donde se concluye que el número de encuestas a realizarse a los locales es de 114.

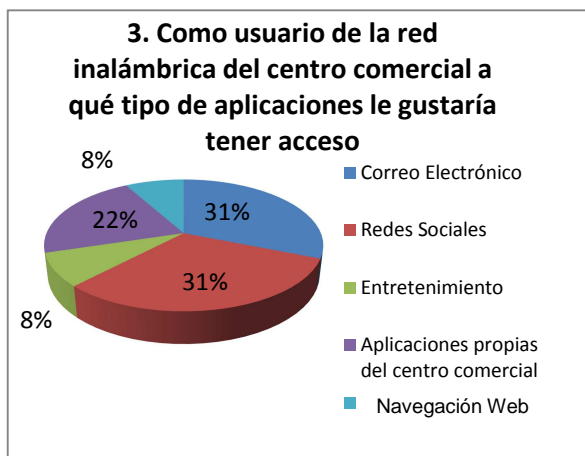
Los resultados obtenidos de en las encuestas a los locales del establecimiento se muestran en las figuras 2.13 a 2.17.



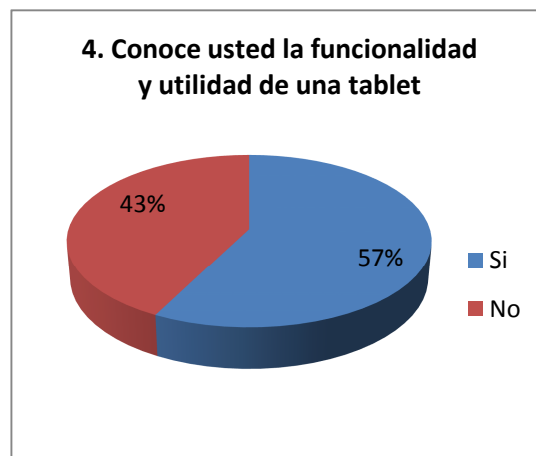
**Figura 2.13:** Resultados de la Pregunta 1 – Encuesta Locales



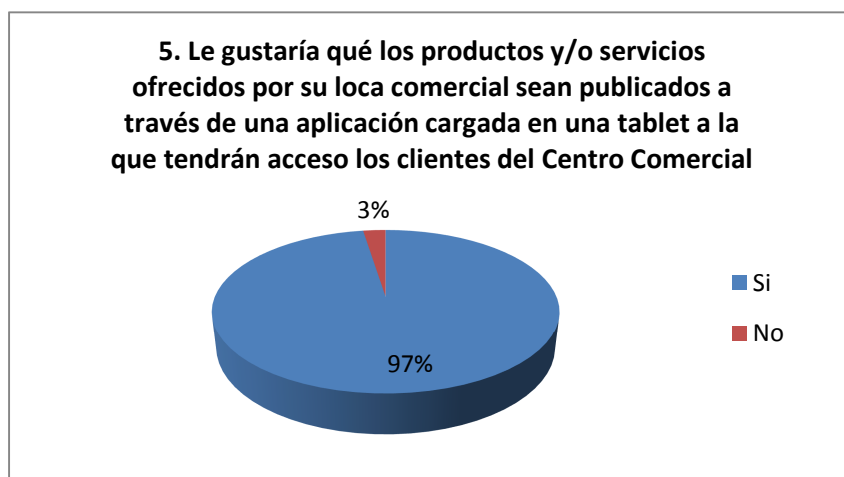
**Figura 2.14:** Resultados de la Pregunta 2 – Encuesta Locales



**Figura 2.15:** Resultados de la Pregunta 3 – Encuesta Locales



**Figura 2.16:** Resultados de la Pregunta 4 – Encuesta Locales



**Figura 2.17:** Resultados de la Pregunta 5- Encuesta Locales

## **2.2 CLASIFICACIÓN DE LOS USUARIOS**

### **2.2.1 USUARIOS FIJOS**

Como ya se definió anteriormente, el Centro Comercial Espiral dispone de 160 locales, teniendo como promedio dos computadores por cada uno de ellos; además existen tres personas encargadas de la administración del mismo, por lo que existirán alrededor de 320 usuarios permanentes de la red inalámbrica.

Dado que en las encuestas se obtuvo que el 95% de los dueños de los locales comerciales están de acuerdo con tener el servicio de Internet inalámbrico, se suma un total de 304 usuarios conectados permanentemente a la red.

La mayor concentración de locales comerciales se encuentra entre el subsuelo 2, subsuelo 1 y la planta baja (Nivel 0).

### **2.2.2 VISITANTES**

Son los usuarios que estarán conectados esporádicamente mientras visiten el centro comercial. Para saber el número de visitantes y compradores que acuden diariamente al centro comercial, se tomó en cuenta el número de ventas diarias promedio que realizan cada local, durante un mes; a su vez basados en datos proporcionados por el administrador del Espiral, propietarios y empleados de los

locales; y de la constatación de los datos mediante visitas realizadas diariamente al Centro Comercial.

En base a estas consideraciones se tiene que el promedio de visitantes y compradores diarios es de 1000 personas, considerando que la mayoría de usuarios son esporádicos. Además se conoce que la afluencia en las horas pico es del 33% del total de las visitas diarias<sup>13</sup>, por lo que se tendrá 300 clientes dentro del centro comercial a esta hora, siendo los niveles 0 y 1 los de mayor afluencia de visitantes, con el 15% del total en cada nivel. Para el resto de niveles se tiene que la afluencia está dividida en partes iguales, es decir 10% por cada nivel<sup>14</sup>.

Dado que en las encuestas se obtuvo que el 38% de visitantes disponen de un dispositivo móvil, se tendrá alrededor de 114 usuarios conectados en horas pico.

Como se observa en la tabla 2.2, se tomó en cuenta el número de usuarios conectados simultáneamente, de la siguiente manera:

- En el subsuelo dos existe la mayor cantidad de locales comerciales (36 locales), considerando un promedio de dos dispositivos por local se tiene un total de 72 usuarios fijos, de acuerdo con las encuestas solo el 95% está de acuerdo con la instalación de la red inalámbrica es decir se estiman 69 usuarios concurrentes.
- Para el caso de los visitantes son 114 usuarios conectados en horas pico que disponen de un dispositivo inalámbrico, y de acuerdo con el porcentaje de afluencia para este piso se obtienen 12 usuarios concurrentes. Finalmente entre usuarios de locales y visitantes se tiene un total de 81 usuarios concurrentes en el subsuelo 2.

---

<sup>13</sup> Dato suministrado por la administración del centro Comercial Espiral en Octubre 2012.

<sup>14</sup> Datos proporcionados por la administración del centro Comercial en octubre 2012, sustentados en función de la facturación de cada local, y de las visitas realizadas a lo largo de este proyecto

- En el subsuelo 1 existe un total de 42 usuarios, de los cuales 39 pueden estar conectados simultáneamente, mientras que de los 114 visitantes en horas pico, existirán 12 visitantes simultáneos en este nivel. Teniendo un total de 51 usuarios en este piso.
- En el piso 0 y 1, 29 usuarios simultáneos estarán conectados en la red, y de acuerdo al porcentaje de afluencia de visitantes, habrán 18 usuarios simultáneos. Dando un total de 47 usuarios por cada uno de estos niveles.
- En el resto de pisos (del piso 2 al 6) existirán 41 usuarios conectados simultáneamente por nivel, 29 usuarios de los locales y 12 visitantes.

Piso	Usuarios de los locales del CC Espiral		Visitantes del CC Espiral		Suma de Usuarios + Visitantes
	Usuarios por Piso (dos por cada local)	Usuarios conectados simultáneamente (95%U-Piso)	%Afluencia simultánea de visitantes por Piso	Visitantes por piso (114 x %Afluencia)	Total Usuarios Simultáneos
-2	72	69	10	12	81
-1	42	39	10	12	51
0	30	29	15	18	47
1	30	29	15	18	47
2	30	29	10	12	41
3	30	29	10	12	41
4	30	29	10	12	41
5	30	29	10	12	41
6	30	29	10	12	41

**Tabla 2.2:** Número de usuarios fijos y visitantes por cada nivel

## 2.3 ANÁLISIS DE TRÁFICO

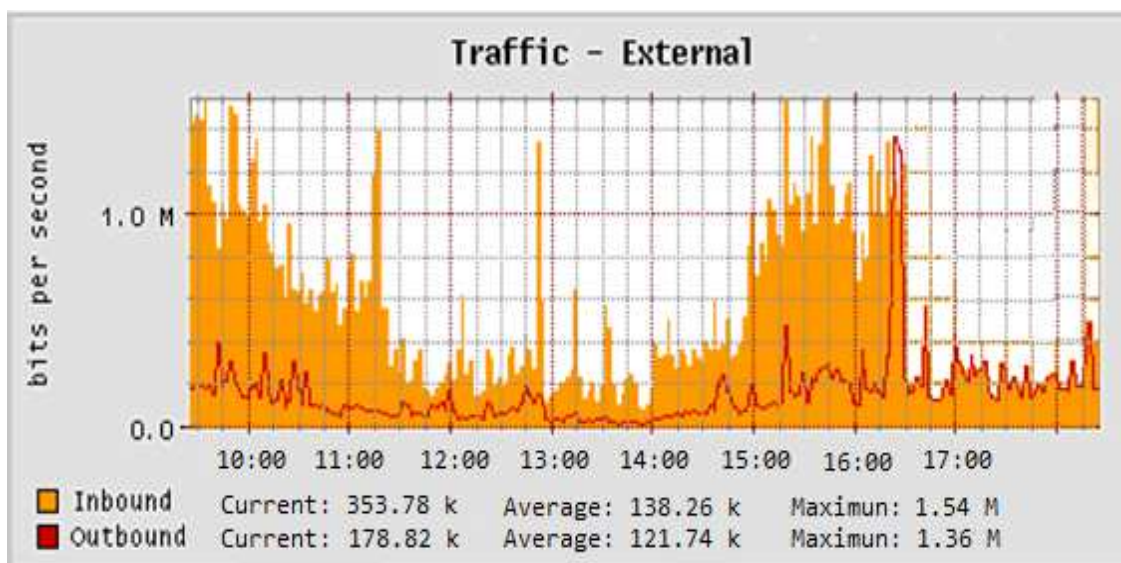
De acuerdo con las encuestas realizadas a los clientes del centro comercial se obtuvieron los siguientes resultados, los mismos que proporcionan los requerimientos de los usuarios dentro de la red inalámbrica.

### 2.3.1 OBTENCIÓN DE DATOS PARA CONOCER LAS HORAS PICO

Para poder realizar el cálculo del ancho de banda necesario para el diseño de la red, se recolectó información de varios equipos que disponen de una conexión a Internet a través de distintos proveedores, localizados dentro del Centro Comercial Espiral mediante el uso de un *router* inalámbrico temporal con un *firmware* DDWRT.

Este dispositivo fue colocado en 20 locales, tomando en cuenta el tipo de local<sup>15</sup>; obteniendo un total de 20 gráficas con el tráfico de red que atraviesa por cada uno de los mismos (Anexo C).

El local en el que se observa los picos más altos se muestra en la figura 2.18; de la cual se concluye que el horario de 9:30 a 10:30 y de 15:00 a 16:00 son las horas de mayor uso de la red, con picos de 1,54 Mbps.



**Figura 2.18:** Horas de mayor utilización de la red del local con picos máximos

<sup>15</sup> Se valoró la cantidad de locales que utilizaban la conexión a Internet para compra y venta de productos, envío de cotizaciones, documentos desde y hacia sucursales, horario de atención y uso general de Internet.

## 2.3.2 CÁLCULO DE LA CAPACIDAD CONSUMIDA POR CADA APLICACIÓN PARA EL ENLACE DE INTERNET

### 2.3.2.1 Video Streaming y Youtube

Al ser YouTube un sitio web de contenido multimedia tal como audio y video hace de este un sitio muy conocido y utilizado para un usuario estándar en el internet. Mediante la utilización de la herramienta online *Website Optimization* que permite analizar el tamaño y *Rapid Search Metric* para tiempo de carga de un sitio web, para el caso de youtube el análisis considera la calidad de imagen que interviene en el video.

Se considera un video promedio de 480 píxeles puesto que la mayoría de videos se presentan en esta calidad. A continuación se presentan los resultados obtenidos:

### Web Page Speed Report

<b>URL:</b>	https://www.youtube.com/watch?v=p3936fD3UP8
<b>Title:</b>	YouTube - Fan - Made - Frozen: Una Aventura Congelada - Trailer Final - Español Latino - HD
<b>Date:</b>	Report run on Tue May 13 00:07:43EDT2014

### Diagnosis

#### Global Statistics

Total HTTP Requests:	3
Total Size:	10382 bytes

**Figura 2.19:** Resultados del tamaño de un Video [16]

Estimated Load Time				
Modem	UMTS	DSL 768	T1	T3
2.69 s	18.44 s	1.34 s	1.03 s	0.34 s

**Figura 2.20:** Resultados del tiempo de carga de Video [17]



La figura 2.20 presenta los tiempos de descarga que presentaría el video de acuerdo a la conexión de internet empleada, en este caso se utiliza una conexión *DSL* (Línea de abonado digital) puesto que es la que poseen los locales actualmente, por lo tanto el tiempo de carga correspondiente tiene un valor de 1.34 segundos.

$$C_{Youtu} = \frac{10382 \text{ Bytes}}{1.34 \text{ s}} \times \frac{1 \text{ KB}}{1024 \text{ Bytes}} \times \frac{8 \text{ bits}}{1 \text{ Byte}} = 60,53 \text{ Kbps}$$

### 2.3.2.2 Redes Sociales

Actualmente las Redes Sociales se han convertido en una de las principales formas de comunicación. Se considera a Facebook como ejemplo para los análisis correspondientes al tráfico, puesto que es la red social más utilizada en nuestro país.

La figura 2.22 muestra los tiempos de descarga correspondientes al navegar en facebook, de acuerdo a la conexión de internet empleada, para nuestro caso se utiliza una conexión *DSL* puesto que es la que poseen los locales actualmente por lo tanto el tiempo de carga correspondiente es de 0,44 segundos.

## Web Page Speed Report

<b>URL:</b>	https://www.facebook.com/TeleamazonasEcuador?ref=ts
<b>Title:</b>	Redirecting...
<b>Date:</b>	Report run on Tue May 13 22:44:27EDT2014

## Diagnosis

### Global Statistics

Total HTTP Requests:	2
Total Size:	920 bytes

**Figura 2.21:** Tamaño de una página promedio al abrir facebook. [16]

Estimated Load Time				
Modem	UMTS	DSL 768	T1	T3
0.89 s	6.07 s	0.44 s	0.34 s	0.11 s

**Figura 2.22:** Tiempo de carga de una página al abrir facebook. [17]

$$C_{Fa} = \frac{920 \text{ Bytes}}{0,44 \text{ s}} \times \frac{1 \text{ KB}}{1024 \text{ Bytes}} \times \frac{8 \text{ bits}}{1 \text{ Byte}} = 1634 \text{ Kbps}$$

### 2.3.2.3 Descargas

Se conoce que dentro de cada local comercial, se realizan descargas de catálogos, manuales, información de productos, instaladores, entre otros; un usuario descarga archivos, con un tamaño promedio de 10 MB<sup>16</sup> en un tiempo efectivo de 8,92 minutos, el cual se obtuvo al realizar la descarga desde la administración del centro comercial.

De donde se obtiene:

$$C_{des} = \frac{10240 \text{ KBytes}}{1 \text{ descarga}} \times \frac{8 \text{ bits}}{1 \text{ Byte}} \times \frac{1 \text{ descarga}}{8,92 \text{ min}} \times \frac{1 \text{ min}}{60 \text{ s}} = 153,06 \text{ Kbps}$$

Por lo que se obtiene una tasa de 153 Kbps.

### 2.3.2.4 Correo Electrónico

Por tratarse de un Centro Comercial la información que se enviará en un correo electrónico son facturas, proformas, catálogos, información en texto plano, que alcanza en tamaño los 2 MB en promedio<sup>17</sup> se obtuvo un tiempo efectivo de 2,3 minutos, el cual se consiguió al realizar varias pruebas de carga.

$$C_{correo} = \frac{2048 \text{ KBytes}}{1 \text{ correo}} \times \frac{8 \text{ bits}}{1 \text{ Byte}} \times \frac{1 \text{ correo}}{2.3 \text{ min}} \times \frac{1 \text{ min}}{60 \text{ seg}} = 118,72 \text{ Kbps}$$

<sup>16</sup> Tamaño de archivo obtenido del promedio de descarga de archivos en la administración del Centro Comercial Espiral.

<sup>17</sup> Dato Obtenido al visitar y comprobar los locales. Tamaños de facturas escaneadas, archivos pdf, documentos de texto y hojas de cálculo.

### 2.3.2.5 Navegación Web

Se considera Navegación web a la búsqueda de información y navegación a páginas en internet, en tal virtud la (figura 2.18) anteriormente citada, muestra los valores de:

- Tráfico promedio de entrada: 138.26 Kbps
- Tráfico promedio de salida: 121.74 Kbps

Promediando los valores se obtiene 130 Kbps para navegación web. La gráfica se obtuvo del router inalámbrico que utiliza el firmware DD-WRT.<sup>18</sup>

### 2.3.2.6 Aplicaciones del Centro Comercial

La aplicación desarrollada para el Centro Comercial realiza consulta de información hacia la base de datos en Mysql. La figura 2.23 muestra los datos estadísticos del tráfico presente al realizar una consulta obtenidos al ingresar a la interfaz web “phpMyAdmin”.



**Figura 2.23:** Tráfico generado en el servidor de base de datos Mysql

$$C_{\text{APLICACIÓN}} = \frac{2,33 \text{ KB}}{1 \text{ consulta}} \times \frac{1 \text{ consulta}}{1,8 \text{ s}} \times \frac{8 \text{ bits}}{1 \text{ Byte}} = 10,35 \text{ Kbps}$$

<sup>18</sup> DD-WRT: Firmware no Oficial para routers inalámbricos especialmente de la marca Linksys, utilizado para administración y control de tráfico.

La tabla 2.3 muestra los cálculos de tráfico para los usuarios del centro comercial, tomando en cuenta la encuesta realizada; así como también el valor del tráfico por aplicación obtenido en el análisis anteriormente mencionado. Donde el porcentaje de utilización es el valor correspondiente al resultado de las encuestas (figura 2.6).

Aplicación	Usuarios	% de Utilización de aplicaciones <sup>19</sup>	Tráfico (Kbps)	Total (Kbps)
Correo Electrónico	114	20	118,72	2706,8
Redes Sociales	114	22	16,34	409,8
Entretenimiento y descargas	114	19	153,06	3315,27
Video Streaming- Youtube	114	16	60.53	1104,07
Navegación Web	114	19	130	2815,8

**Tabla 2.3:** Resumen de tráfico para visitantes del centro comercial Espiral

En la tabla 2.4 se describe los cálculos de tráfico para los propietarios de los locales comerciales como anteriormente se mencionó son 160 locales los mismos que tienen en promedio de 2 PCs cada uno y tomando en cuenta los resultados de las encuestas.

Una vez recolectada la información necesaria la capacidad del canal necesario en el caso más crítico es:

$$C_{AB} = 2706,8+409,8+3315,27+1104,07+2815,8+11777,024+1620,93+3918,34+3328+728,64$$

$$C_{AB} = 31724,674 \text{ Kbps}$$

$$C_{AB} = 30,98 \text{ Mbps}$$

<sup>19</sup> Datos obtenidos de la figura 2.6 (encuestas realizadas a los visitantes del Espiral).

Aplicación	Locales	Host	% de Utilización de aplicaciones <sup>20</sup>	Tráfico (Kbps)	Total (Kbps)
Correo Electrónico	160	2	31	118,72	11777,024
Redes Sociales	160	2	31	16,34	1620,93
Entretenimiento y descargas	160	2	8	153,06	3918,34
Navegación Web	160	2	8	130	3328,00
Aplicaciones del C.C <sup>21</sup>	160	2	22	10,35	728,64

**Tabla 2.4:** Resumen de tráfico para empleados y dueños de los locales

Del análisis de usuarios fijos y visitantes del ítem 2.2 se obtienen:

- 114 usuarios visitantes en horas pico
- 304 usuarios fijos

Dando un total de 418 usuarios ocupantes de la red, se obtiene que la capacidad por usuario es:

$$C_{\text{usuario}} = \frac{31724,674 \text{ Kbps}}{418 \text{ usuarios}}$$

$$C_{\text{usuario}} = 75,90 \text{ Kbps}$$

## 2.4 SITE SURVEY

Se define como el recurso que permite analizar el medio físico en donde intervendrá la red a diseñarse. Con esta herramienta se analiza los parámetros requeridos para el diseño de la red, así como el nivel de interferencia que existe actualmente en el área, descubriendo las redes que actualmente se encuentran operando, y parámetros como el SSID, canal de trabajo, dirección MAC del

<sup>20</sup> Datos obtenidos de la figura 2.15 (encuestas realizadas en los locales comerciales).

<sup>21</sup> Por aplicaciones del centro comercial se encuentra incluido la capacidad necesaria para una conexión óptima a una base de datos.

dispositivo, máxima señal emitida (dBm). Una vez obtenida esta información, se procede a escoger los mejores parámetros para que nuestra red tenga un correcto y efectivo funcionamiento. La herramienta específica a utilizarse para este análisis es *AIRMAGNET Survey*.

Para un correcto análisis se tomará en cuenta los siguientes pasos:

- Obtener un diagrama del centro comercial Espiral, donde se realizará el diseño.
- Realizar una inspección visual del área.
- Obtener información respecto a redes inalámbricas cercanas al área, reconociendo su SSID, velocidad de transmisión, intensidad de su señal y canal en el cual trabaja.
- Determinar la ubicación que tendrían los *Access Points*.
- Verificar que los *Access Points* tengan una correcta localización.

#### **2.4.1 SITE SURVEY PASIVO**

Permite encontrar redes inalámbricas dentro del estándar IEEE 802.11 (a, b, g), en los alrededores del área de influencia para nuestro diseño.

Los resultados obtenidos del programa AIRMAGNET se tabularon en función del estándar, la dirección *MAC*, el canal utilizado, el *SSID* de cada punto, y la potencia de los mismos, distribuidos por cada nivel dentro del centro comercial.

En la tabla 2.5 se detalla los resultados obtenidos para el subsuelo 2, en donde se observa que el canal más utilizado es el 11, con 8 puntos de acceso, y el AP con mayor potencia (-71 dBm) trabaja en canal 1.

En la tabla 2.6 se muestra el resultado del *Site Survey* para el subsuelo 1, en donde el canal más utilizado es el 11, con 13 puntos de acceso, mientras que el *AP* que emite la potencia más alta (-76 dBm) trabaja en el canal 5.

En la tabla 2.7 se detalla los puntos de acceso que reflejan un nivel de señal en el nivel cero, donde se observa que el canal más afluido es el 11, con 14 puntos de acceso trabajando en el mismo, incluido el *AP* con la potencia más alta (-76 dBm).

En la tabla 2.8 se muestra el resultado del *site survey* pasivo para el nivel uno, obteniendo que el canal más utilizado es el 11, con 15 puntos de acceso trabajando en este canal, incluyendo el de mayor potencia (-75 dBm).

En la tabla 2.9 se detalla el resultado del *site survey* pasivo para el nivel dos, siendo el canal 11 el más utilizado, con 12 puntos de acceso conectados a través del mismo, incluyendo el que registra la potencia más alta (-70 dBm).

En la tabla 2.10 se muestra el resultado del *site survey* pasivo realizado en el nivel tres, obteniendo que el canal más utilizado es el 11, con 11 puntos de acceso conectados a través del mismo, mientras que el *AP* con la potencia más alta (-69 dBm) trabaja en el canal 3.

En la tabla 2.11 se muestra el resultado del *site survey* pasivo realizado en el nivel cuatro, obteniendo que el canal de mayor uso es el 11, con 11 puntos de acceso trabajando en el mismo, incluyendo el que registra mayor potencia (-70 dBm).

En la tabla 2.12 se detalla el *site survey* pasivo para el nivel cinco, en donde se observa que el canal más utilizado es el 11, con 8 puntos de acceso trabajando en el mismo, mientras que el *AP* con la potencia más alta (-69 dBm) trabaja en el canal 6.

En la tabla 2.13 se muestra el resultado del *site survey* pasivo en el nivel seis, donde los canales más utilizados son el 6 y 11, con 6 puntos de acceso conectados en cada canal, el *AP* que registra la potencia más alta (-77 dBm) trabaja en el canal 6.

Medio	Dirección MAC	Canal	SSID	Potencia(dBm)
802.11g	00:14:D1:60:67:C5	1	DIGITAL GAMES	-76
802.11g	00:21:63:E0:1E:C7	1	CNT	-73
802.11g	00:22:75:06:39:AC	1	CallCell	-71
802.11g	00:E0:4D:D5:AB:48	1	Byron Romero Pena	-75
802.11b	90:84:0D:D6:7E:99	2	Apple-Sagatur	-75
802.11g	F4:EC:38:F9:F2:62	2	Maqueno Republik	-76
802.11g	B0:48:7A:C2:9C:24	5	Cyber	-76
802.11g	C8:3A:35:00:0A:08	6	Laboratorio NOVICOMPU	-96
802.11g	00:26:B6:6E:B2:A6	11	NOVICOMPU	-81
802.11g	00:26:B6:87:8D:4A	11	GameCenter	-83
802.11g	00:E0:4D:9E:46:A0	11	Casa Bolivia	-83
802.11g	88:9F:FA:07:EF:BC	11	Claro_Camacho	-79
802.11g	D8:5D:4C:FF:A4:66	11	Antigua Damasco	-77
802.11g	E8:39:DF:23:05:5A	11	VACA RUBI	-76
802.11g	E8:39:DF:23:09:76	11	ECUAVENTURA	-83
802.11g	E8:39:DF:23:1D:FE	11	Marea Negra	-64

**Tabla 2.5:** Resultados obtenidos del *Site Survey* Pasivo en el subsuelo 2

Medio	Dirección MAC	Canal	SSID	Potencia (dBm)
802.11g	00:14:D1:60:67:C5	1	DIGITAL GAMES	-84
802.11g	00:21:63:E0:1E:C7	1	CNT	-86
802.11g	00:22:75:06:39:AC	1	CallCell	-79
802.11g	00:E0:4D:D5:AB:48	1	Byron Romero Pena	-90
802.11b	90:84:0D:D6:7E:99	2	Apple-Sagatur	-86
802.11g	F4:EC:38:F9:F2:62	2	Maqueno Republik	-91
802.11g	B0:48:7A:C2:9C:24	5	Cyber	-76
802.11b	00:22:75:E3:C5:10	11	Belkin_e3c510	-84
802.11g	00:26:B6:6E:85:FE	11	GONZALO AYALA	-83
802.11g	00:26:B6:6E:B2:A6	11	NOVICOMPU	-87
802.11g	00:26:B6:7B:0F:38	11	Wifi AntiHacking	-89
802.11g	00:26:B6:87:8F:EA	11	Marcia	-88
802.11g	00:E0:4D:9E:46:A0	11	Casa Bolivia	-79
802.11g	88:9F:FA:07:EF:BC	11	Claro_Camacho	-89
802.11g	D8:5D:4C:FF:A4:66	11	Antigua Damasco	-89
802.11g	E8:39:DF:23:05:5A	11	VACA RUBI	-85
802.11g	E8:39:DF:23:05:FA	11	Karen Chavez	-86
802.11g	E8:39:DF:23:09:76	11	ECUAVENTURA	-92
802.11g	E8:39:DF:23:1D:FE	11	Marea Negra	-82

**Tabla 2.6:** Resultados obtenidos del *Site Survey* Pasivo en el subsuelo 1



Medio	Dirección MAC	Canal	SSID	Potencia(dBm)
802.11g	00:14:D1:60:67:C5	1	DIGITAL GAMES	-91
802.11g	00:21:63:E0:1E:C7	1	CNT	-92
802.11g	00:22:75:06:39:AC	1	CallCell	-86
802.11b	90:84:0D:D6:7E:99	2	Apple-Sagatur	-92
802.11g	B0:48:7A:C2:9C:24	5	Cyber	-85
802.11b	00:1E:E5:B5:FE:62	6	Callcell1	-88
802.11g	00:50:18:5C:66:CE	6	GP	-88
802.11g	EC:55:F9:5D:0C:4F	10	CLARO_APLUS	-87
802.11b	00:22:75:E3:C5:10	11	Belkin_e3c510	-76
802.11g	00:26:B6:6E:85:FE	11	GONZALO AYALA	-88
802.11g	00:26:B6:6E:B2:A6	11	NOVICOMPU	-94
802.11g	00:26:B6:7B:0F:38	11	Wifi AntiHacking	-84
802.11g	00:26:B6:87:8F:EA	11	Marcia	-81
802.11g	00:E0:4D:9E:46:A0	11	Casa Bolivia	-83
802.11g	88:9F:FA:07:EF:BC	11	Claro_Camacho	-95
802.11b	C8:D5:FE:03:21:E4	11	DIEGO S.T.	-88
802.11g	E8:39:DF:23:09:76	11	ECUAVENTURA	-89
802.11g	D8:5D:4C:FF:A4:66	11	Antigua Damasco	-92
802.11g	E8:39:DF:23:05:5A	11	VACA RUBI	-92
802.11g	E8:39:DF:23:05:FA	11	Karen Chavez	-90
802.11g	E8:39:DF:23:07:E6	11	Alexandra	-88
802.11g	E8:39:DF:23:1D:FE	11	Marea Negra	-95
802.11b	C8:D5:FE:03:21:E4	11	DIEGO S.T.	-81
802.11g	00:26:B6:83:20:1E	11	RedCarla	-87
802.11g	00:E0:4D:9E:46:A0	11	Casa Bolivia	-79

**Tabla 2.7:** Resultados obtenidos del *Site Survey* Pasivo en el piso 0

Medio	Dirección MAC	Canal	SSID	Potencia(dBm)
802.11g	00:21:63:E0:1E:C7	1	CNT	-95
802.11g	00:22:75:06:39:AC	1	CallCell	-90
802.11b	90:84:0D:D6:7E:99	2	Apple-Sagatur	-96
802.11g	B0:48:7A:C2:9C:24	5	Cyber	-91
802.11g	00:15:E9:13:82:08	6	Andinatel JC	-91
802.11b	00:1E:E5:B5:FE:62	6	Callcell1	-90
802.11g	00:E0:4D:9E:30:68	6	Diegob	-89
802.11g	C8:3A:35:5C:B3:C8	6	Nexxt	-89
802.11g	00:50:18:5C:66:CE	6	GP	-84
802.11g	00:27:22:80:43:FB	8	TRANSCON	-88
802.11g	EC:55:F9:5D:0C:4F	10	CLARO_APLUS	-83
802.11g	00:1A:73:CC:A3:9D	11	Claro_Alvarez	-89
802.11g	00:21:63:DE:06:0F	11	C.N.T	-90
802.11b	00:22:75:E3:C5:10	11	Belkin_e3c510	-82
802.11g	00:26:B6:6E:85:FE	11	GONZALO AYALA	-92
802.11g	00:26:B6:6E:B2:A6	11	NOVICOMPU	-97
802.11g	00:26:B6:7B:0F:38	11	Wifi AntiHacking	-76
802.11g	00:26:B6:83:20:1E	11	RedCarla	-87
802.11g	00:26:B6:87:8F:EA	11	Marcia	-72
802.11g	00:E0:4D:9E:46:A0	11	Casa Bolivia	-88
802.11g	18:F4:6A:08:BF:1A	11	Claro_bei su	-91
802.11b	C8:D5:FE:03:21:E4	11	DIEGO S.T.	-81
802.11g	D8:5D:4C:FF:A4:66	11	Antigua Damasco	-96
802.11g	E8:39:DF:23:05:FA	11	Karen Chavez	-95
802.11g	E8:39:DF:23:07:E6	11	Alexandra	-75
802.11g	E8:39:DF:23:09:76	11	ECUAVENTURA	-83

**Tabla 2.8:** Resultados obtenidos del *Site Survey* Pasivo en el piso1

Medio	Dirección MAC	Canal	SSID	Potencia(dBm)
802.11g	00:22:75:06:39:AC	1	CallCell	-96
802.11g	00:E0:4D:D5:AB:48	1	Byron Romero Pena	-91
802.11g	00:27:22:1E:69:CF	3	Unknown	-88
802.11g	B0:48:7A:C2:9C:24	5	Cyber	-94
802.11g	00:15:E9:13:82:08	6	Andinatel JC	-80
802.11g	00:1C:F0:99:E4:BB	6	Matriz	-94
802.11b	00:1E:E5:B5:FE:62	6	Callcell1	-95
802.11g	00:E0:4D:9E:30:68	6	Diegob	-81
802.11g	C8:3A:35:5C:B3:C8	6	Nexxt	-82
802.11g	00:50:18:5C:66:CE	6	GP	-77
802.11g	00:27:22:80:43:FB	8	TRANSCON	-81
802.11g	EC:55:F9:5D:0C:4F	10	CLARO_APLUS	-88
802.11g	00:21:63:DE:06:0F	11	C.N.T	-81
802.11b	00:22:75:E3:C5:10	11	Belkin_e3c510	-89
802.11g	00:26:B6:83:20:1E	11	RedCarla	-81
802.11g	00:26:B6:87:8F:EA	11	Marcia	-82
802.11g	00:E0:4D:9E:46:A0	11	Casa Bolivia	-93
802.11g	18:F4:6A:08:BF:1A	11	Claro_bei su	-85
802.11g	E8:39:DF:23:09:76	11	ECUAVENTURA	-88
802.11b	C8:D5:FE:03:21:E4	11	DIEGO S.T.	-77
802.11g	E8:39:DF:23:07:E6	11	Alexandra	-70
802.11g	E8:39:DF:23:1C:F2	11	Robotica EPN.	-96

**Tabla 2.9:** Resultados obtenidos del *Site Survey* Pasivo en el piso 2

Medio	Dirección MAC	Canal	SSID	Potencia(dBm)
802.11g	00:27:22:1E:69:CF	3	Unknown	-83
802.11g	00:15:E9:13:82:08	6	Andinatel JC	-69
802.11g	00:1C:F0:99:E4:BB	6	Matriz	-85
802.11g	00:22:75:4B:0A:2B	6	SOLTEC ESPIRAL	-91
802.11g	00:E0:4D:9E:30:68	6	Diegob	-74
802.11g	C8:3A:35:5C:B3:C8	6	Nexxt	-77
802.11g	00:50:18:5C:66:CE	6	GP	-82
802.11g	00:27:22:80:43:FB	8	TRANSCON	-74
802.11g	EC:55:F9:5D:0C:4F	10	CLARO_APLUS	-94
802.11g	00:1A:73:CC:A3:9D	11	Claro_Alvarez	-75
802.11g	00:21:63:DE:06:0F	11	C.N.T	-89
802.11b	00:22:75:E3:C5:10	11	Belkin_e3c510	-95
802.11g	00:26:B6:7B:0F:38	11	Wifi AntiHacking	-89
802.11g	00:26:B6:83:20:1E	11	RedCarla	-76
802.11g	00:26:B6:87:8F:EA	11	Marcia	-90
802.11g	18:F4:6A:08:BF:1A	11	Claro_bei su	-79
802.11b	C8:D5:FE:03:21:E4	11	DIEGO S.T.	-84
802.11g	E8:39:DF:23:09:76	11	ECUAVENTURA	-92
802.11g	E8:39:DF:23:1C:F2	11	Robotica EPN.	-88

**Tabla 2.10:** Resultados obtenidos del *Site Survey* Pasivo en el piso 3

Medio	Dirección MAC	Canal	SSID	Potencia(dBm)
802.11g	00:27:22:1E:69:CF	3	Unknown	-75
802.11g	00:15:E9:13:82:08	6	Andinatel JC	-77
802.11g	00:1C:F0:99:E4:BB	6	Matriz	-78
802.11g	00:22:75:4B:0A:2B	6	SOLTEC ESPIRAL	-86
802.11g	00:E0:4D:9E:30:68	6	Diegob	-85
802.11g	C8:3A:35:5C:B3:C8	6	Nexxt	-82
802.11g	00:50:18:5C:66:CE	6	GP	-89
802.11g	00:27:22:80:43:FB	8	TRANSCON	-83
802.11g	88:9F:FA:4A:9D:B1	10	CLARO_LOMBEIDA	-96
802.11g	00:1A:73:CC:A3:9D	11	Claro_Alvarez	-83
802.11g	00:21:63:DE:06:0F	11	C.N.T	-97
802.11g	00:26:B6:7B:0F:38	11	Wifi AntiHacking	-95
802.11g	00:26:B6:83:20:1E	11	RedCarla	-87
802.11g	00:26:B6:87:8F:EA	11	Marcia	-97
802.11g	18:F4:6A:08:BF:1A	11	Claro_bei su	-70
802.11b	C8:D5:FE:03:21:E4	11	DIEGO S.T.	-90
802.11g	E8:39:DF:23:04:3E	11	Juan	-88
802.11g	E8:39:DF:23:07:E6	11	Alexandra	-87
802.11g	E8:39:DF:23:09:76	11	ECUAVENTURA	-97
802.11g	E8:39:DF:23:1C:F2	11	Robotica EPN.	-82

**Tabla 2.11:** Resultados obtenidos del *Site Survey* Pasivo en el piso 4

Medio	Dirección MAC	Canal	SSID	Potencia (dBm)
802.11g	00:27:22:1E:69:CF	3	Unknown	-81
802.11g	00:15:E9:13:82:08	6	Andinatel JC	-84
802.11g	00:1C:F0:99:E4:BB	6	Matriz	-69
802.11g	00:22:75:4B:0A:2B	6	SOLTEC ESPIRAL	-77
802.11g	00:E0:4D:9E:30:68	6	Diegob	-89
802.11g	C8:3A:35:5C:B3:C8	6	Nexxt	-90
802.11g	00:50:18:5C:66:CE	6	GP	-97
802.11g	00:27:22:80:43:FB	8	TRANSCON	-90
802.11g	EC:55:F9:5D:0C:4F	10	CLARO_APLUS	-83
802.11g	00:1A:73:CC:A3:9D	11	Claro_Alvarez	-89
802.11g	00:26:B6:83:20:1E	11	RedCarla	-92
802.11g	00:E0:4D:9E:46:A0	11	Casa Bolivia	-83
802.11g	18:F4:6A:08:BF:1A	11	Claro_bei su	-80
802.11b	C8:D5:FE:03:21:E4	11	DIEGO S.T.	-97
802.11g	E8:39:DF:23:04:3E	11	Juan	-84
802.11g	E8:39:DF:23:07:E6	11	Alexandra	-96
802.11g	E8:39:DF:23:1C:F2	11	Robotica EPN.	-70

**Tabla 2.12:** Resultados obtenidos del *Site Survey* Pasivo en el piso 5

Medio	Dirección MAC	Canal	SSID	Potencia (dBm)
802.11g	00:26:B6:6E:43:20	1	JoseC	-89
802.11g	00:27:22:1E:69:CF	3	Unknown	-88
802.11g	00:15:E9:13:82:08	6	Andinatel JC	-93
802.11g	00:1C:F0:99:E4:BB	6	Matriz	-77
802.11g	00:22:75:4B:0A:2B	6	SOLTEC ESPIRAL	-89
802.11g	00:E0:4D:9E:30:68	6	Diegob	-97
802.11g	C8:3A:35:00:11:00	6	Camara PI	-89
802.11g	EC:55:F9:5D:0C:4F	10	CLARO_APLUS	-90
802.11g	00:21:63:DE:3B:04	11	Fantasias	-81
802.11g	00:26:B6:6E:10:3C	11	Virus	-85
802.11g	18:F4:6A:08:BF:1A	11	Claro_bei su	-93
802.11g	E8:39:DF:23:04:3E	11	Juan	-89
802.11g	E8:39:DF:23:1C:F2	11	Robotica EPN.	-80

**Tabla 2.13:** Resultados obtenidos del *Site Survey* Pasivo en el piso 6

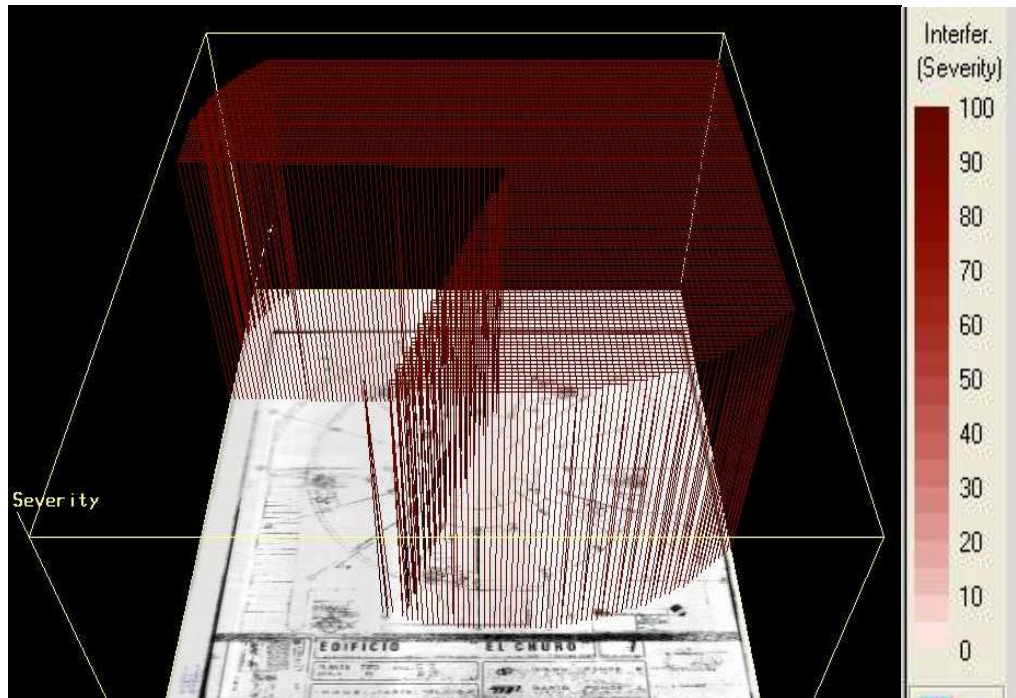
#### 2.4.2 INTERFERENCIA

Cuando existen redes inalámbricas dentro de la misma área geográfica, si queremos evitar interferencia, es necesario utilizar canales que no estén saturados, siempre y cuando éstos no se sobrelapen.

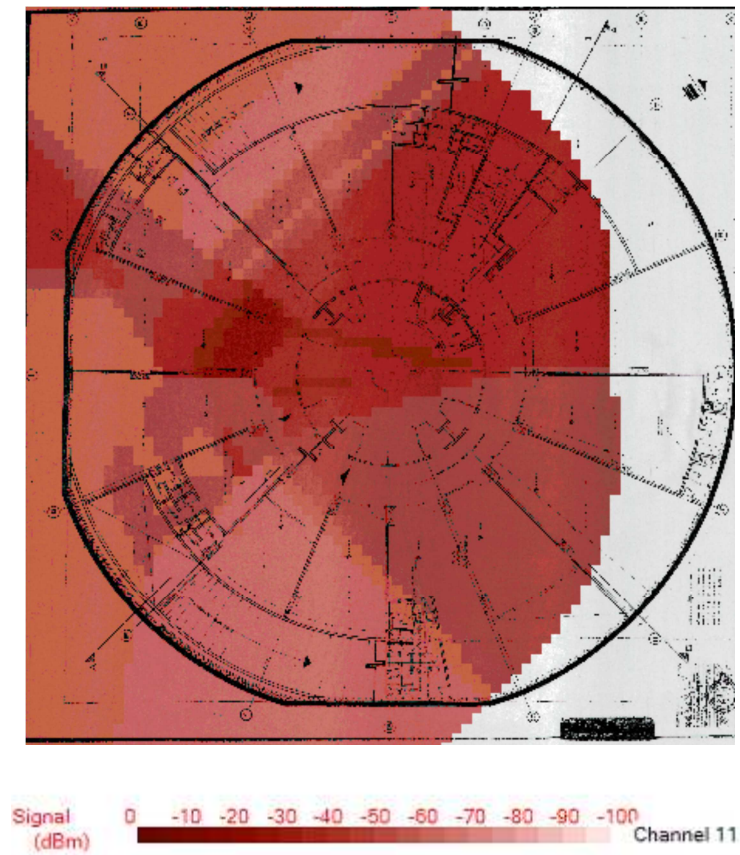
La figura 2.24 muestra la severidad de la interferencia en escala de 0 a 100, el nivel de señal de interferencia se muestra en la figura 2.25

Del resultado obtenido al realizar el *site survey* pasivo en todo el centro comercial, podemos concluir que el canal con mayor interferencia es el 11.

Por lo cual para el diseño se utilizará los canales 1 y 6 para evitar que los *Access Points* que se instalarán en un futuro se interfieran entre sí.



**Figura 2.24:** Severidad de la interferencia en el canal 11



**Figura 2.25:** Interferencia canal 11



## 2.5 CARACTERÍSTICAS DE LOS EQUIPOS QUE FORMAN PARTE DE LA RED INALÁMBRICA

Una vez analizado el tipo de usuarios y la capacidad promedio que cada uno necesita, se procede a calcular el número de *Access Points* necesarios para el diseño:

Se conoce que por recomendación CISCO, a un AP pueden conectarse entre 25 y 30 usuarios concurrentemente<sup>22</sup>, por lo que tomaremos el valor de 25 usuarios. Por lo que la capacidad mínima que debe tener el AP a escoger es de:

$$C_{\text{mínima}} = 25 * 75,9 = 1897,5 \text{ Kbps}$$

La tabla 2.14 muestra una comparación de estándares inalámbricos de tal manera que permita escoger el mejor estándar para el diseño.

Características	802.11 a	802.11b	802.11g	802.11n
<b>Canales no solapados</b>	8 canales	3 canales	3 canales	8 canales
<b>Cobertura en interiores</b>	10-40 metros	Más de 50 metros	30-50 metros	40-70 metros
<b>Interferencia</b>	A 5 GHz menos saturación del espectro	Interferencias con equipos como microondas y teléfonos	A 2.4 GHz presentan problemas de saturación	A 2.4GHz problemas de interferencia. A 5 GHz menos saturación
<b>Compatibilidad</b>	Incompatible con 802.11b/g	802.11	802.11b	802.11 a/b/g
<b>Adecuación de antenas</b>	Uso exclusivo de antenas integradas por el fabricante	Compatibilidad con antenas a 2.4 GHz	Compatibilidad con antenas a 2.4GHz	MIMO Compatibilidad con antenas a 2.4 GHz y 5 GHz
<b>Usuarios Simultáneos</b>	64	32	50	100

**Tabla 2.14:** Comparación de estándares inalámbricos [7]

<sup>22</sup> Dato obtenido de Ficha técnica: Puntos de Acceso Inalámbrico CISCO serie 500- pág. 5

De acuerdo a los requisitos del diseño, el número de usuarios máximos en cada punto de acceso es de 25, y en base al tráfico calculado, si un usuario de la red se encuentra utilizándola al 100%, estará manejando una velocidad de 478,65 Kbps, si los 25 usuarios conectados al punto de acceso se encuentran utilizando la red de manera similar, la velocidad mínima que debe manejar el AP es de 11966,3 Kbps. Con lo que se descarta el uso de 802.11b, por disponer de un throughput de 5.5 Mbps.

Según lo mostrado en la tabla 2.13 y en la tabla 1.1, el estándar que cumple con el diseño de la red es IEEE 802.11g, por lo cual será el estándar utilizado en la misma.

La tabla 2.15 muestra el número de usuarios que se conectarán simultáneamente por piso y el número de puntos de acceso necesarios para garantizar el buen desempeño de la red, los usuarios simultáneos por piso son el resultado de la suma de los usuarios fijos y visitantes concurrentes que se muestran en la tabla 2.2 antes mencionada.

Para el subsuelo 2, al tener 82 posibles usuarios conectados simultáneamente, será necesario el uso de cuatro puntos de acceso para cumplir con lo requerido.

Para el subsuelo 1, se tiene 51 posibles usuarios conectados simultáneamente, por lo cual será necesario el uso de 3 APs para este nivel.

Para las plantas 0 y 1, se tendrá un total de 47 posibles usuarios conectados simultáneamente en la red, y será necesario 2 puntos de acceso por cada nivel para cumplir con lo requerido.

Para las plantas 2, 3, 4, 5 y 6, puede haber 41 usuarios simultáneos conectados en la red, por lo que será necesario el uso de dos APs por cada nivel para cumplir este requisito.



Piso	Usuarios simultáneos	Número de APs necesarios
-2	81	4
-1	51	3
0	47	2
1	47	2
2	41	2
3	41	2
4	41	2
5	41	2
6	41	2
<b>TOTAL</b>		20

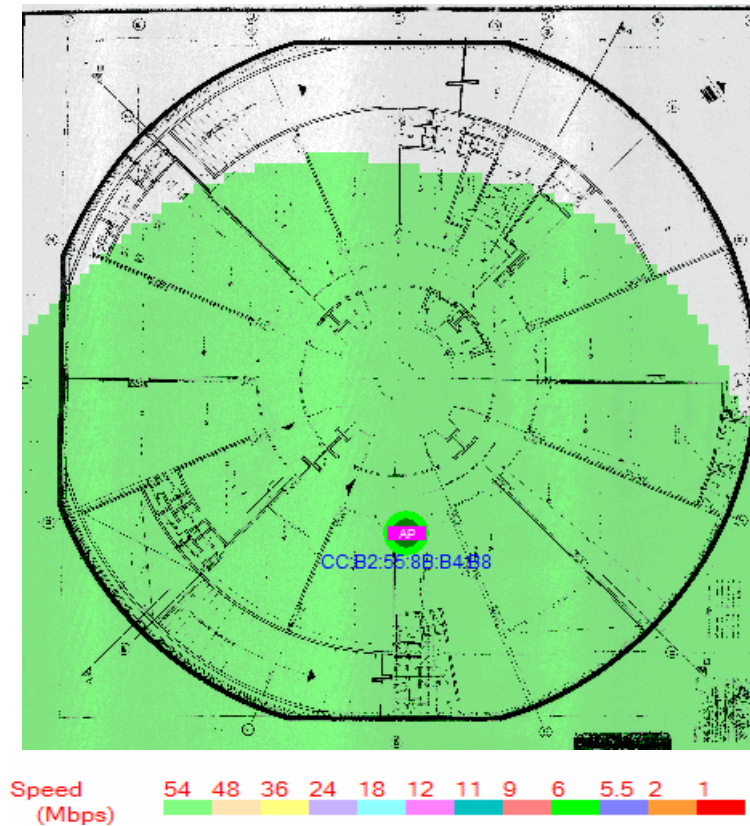
**Tabla 2.15:** Distribución de Puntos de Acceso Inalámbrico por piso

## 2.6. SURVEY ACTIVO

Mediante este procedimiento se observará si los puntos escogidos para la ubicación de los *Access Points* son los adecuados. Se procede a realizar las respectivas pruebas con el programa *Air Magnet Survey* (el mismo con el que se realizó el *Survey Pasivo*) colocando el número de APs que se detallaron anteriormente.

Para el análisis del nivel de señal y de velocidades máximas que ofrece un *Access Point* tipo se utilizó un AP *Dlink DAP-2360*, el cual cumple los requerimientos necesarios para nuestra red.

En la figura 2.26 se muestra el resultado del programa *Air Magnet Survey*, al colocar el primer AP en el subsuelo dos, al observar el nivel de señal y la velocidad que cubre el dispositivo en el piso, se utilizaría únicamente un solo punto de acceso, sin embargo al requerir mayor cantidad de usuarios es necesario colocar un segundo AP.



**Figura 2.26:** Survey Activo Subsuelo 2, AP1

En la figura 2.27, se observa la velocidad obtenida con el programa *Air Magnet Survey* al colocar el segundo AP en el subsuelo 2, en donde se puede concluir que cumple con los requisitos de señal y velocidad para este punto.

Además se observa que existe un espacio donde se atenúa la señal, por lo que en este lugar será ubicado el tercer punto.

En la figura 2.28 se observa el nivel de señal obtenida al colocar el tercer punto de acceso en el subsuelo dos, este punto cubre el espacio de sombra que se tenía en el segundo AP.

En la figura 2.29 se muestra el nivel de señal que emite el cuarto punto de acceso colocado en el subsuelo dos. Con este AP se completa los cuatro necesarios para este nivel, cubriendo así el requerimiento para el diseño de la red.

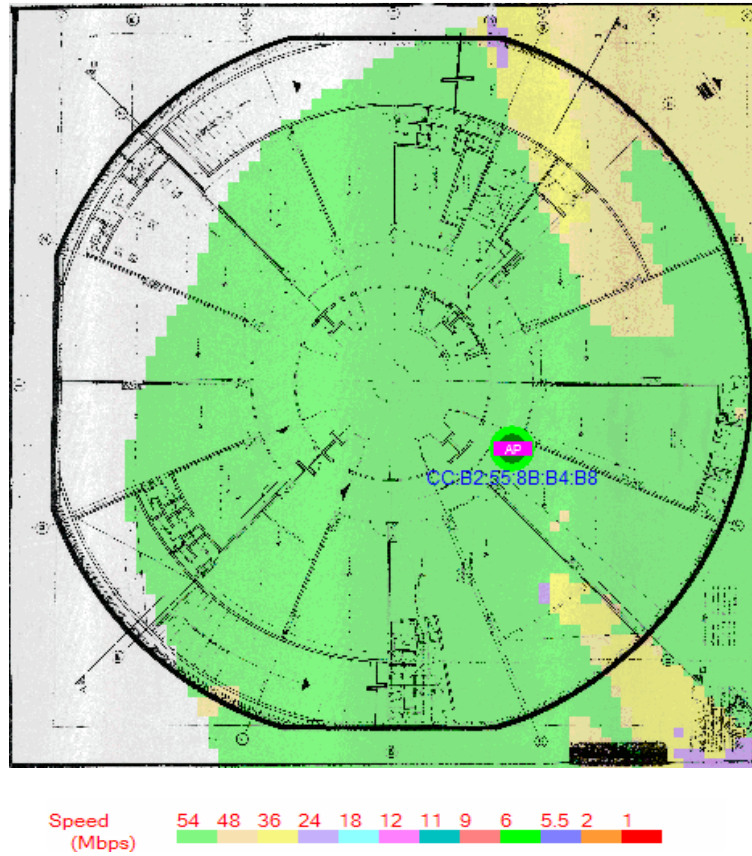


Figura 2.27: Survey Activo Subsuelo 2, AP2

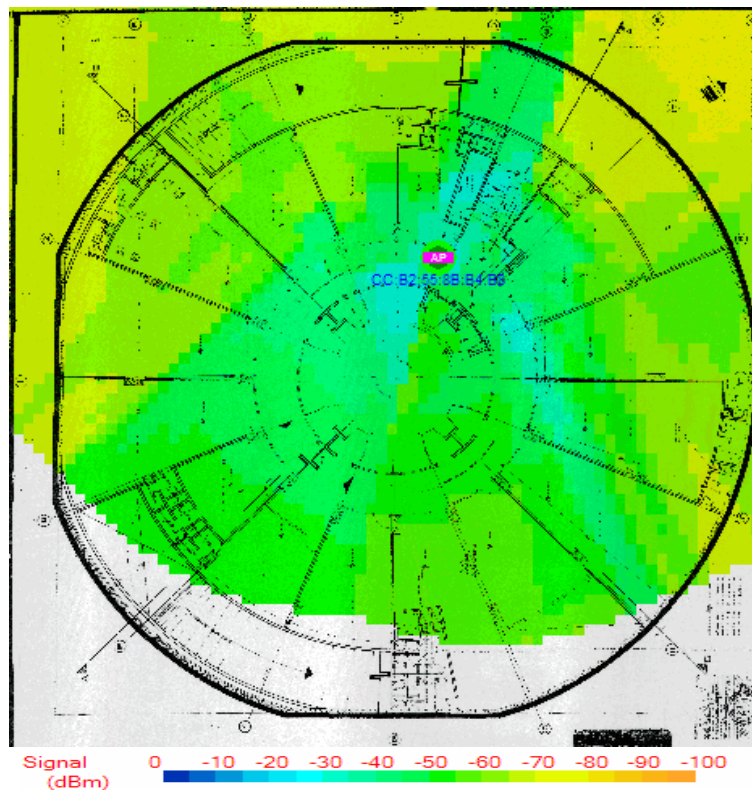
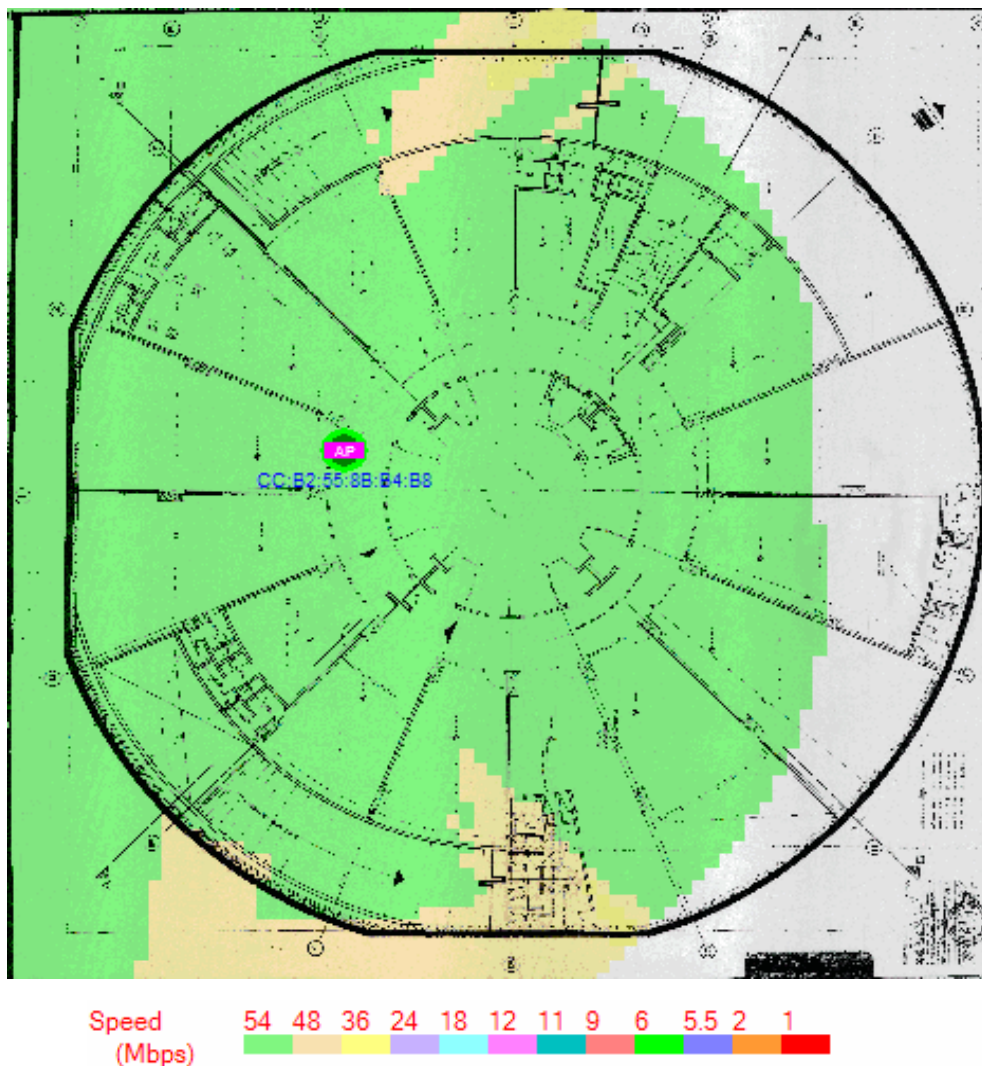


Figura 2.28: Survey Activo Subsuelo 2, AP3



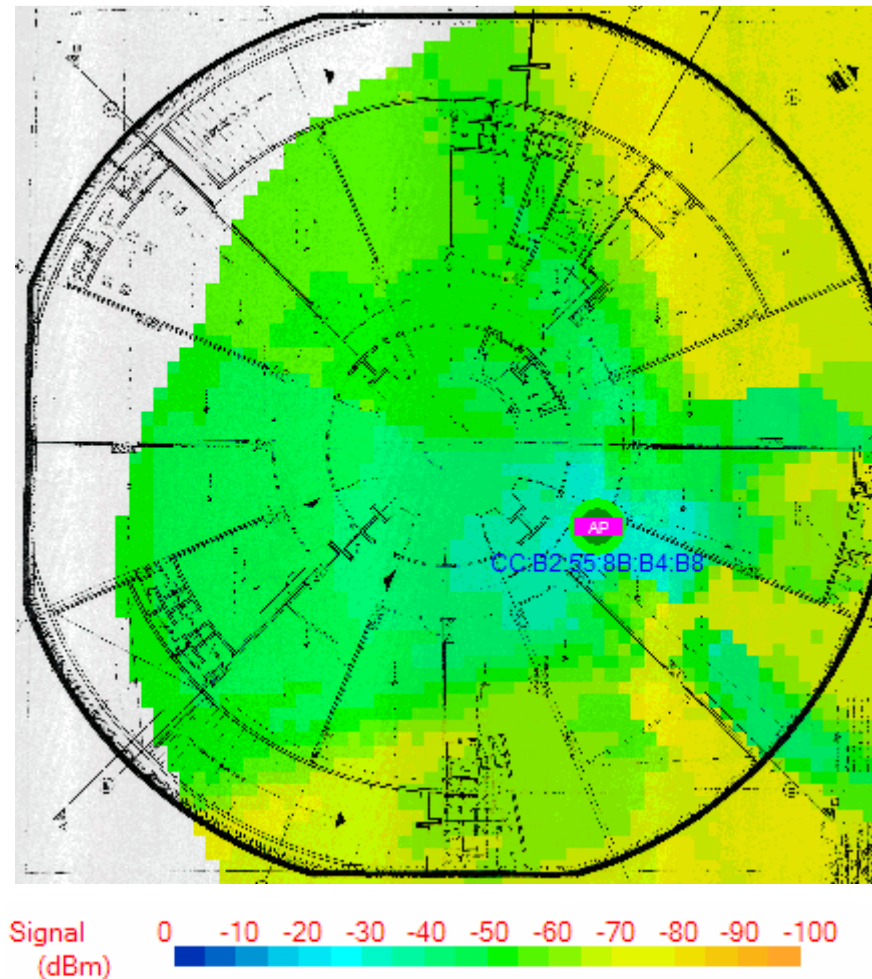
**Figura 2.29:** *Survey* Activo Subsuelo 2, AP4

En el subsuelo uno, fueron ubicados tres APs, sin embargo al momento de comparar con los requerimientos mínimos de velocidad en un punto no cumple; por lo cual se procede a redistribuir y añadir un AP más en este subsuelo.

Una vez realizado el cambio se vuelve a probar con la herramienta, y efectivamente se muestra que cumple con el requerimiento de señal y de número de usuarios máximos por cada AP. Por lo que en el subsuelo dos son necesarios cuatro APs para cumplir con los requerimientos de número de usuarios y de velocidades mínimas.



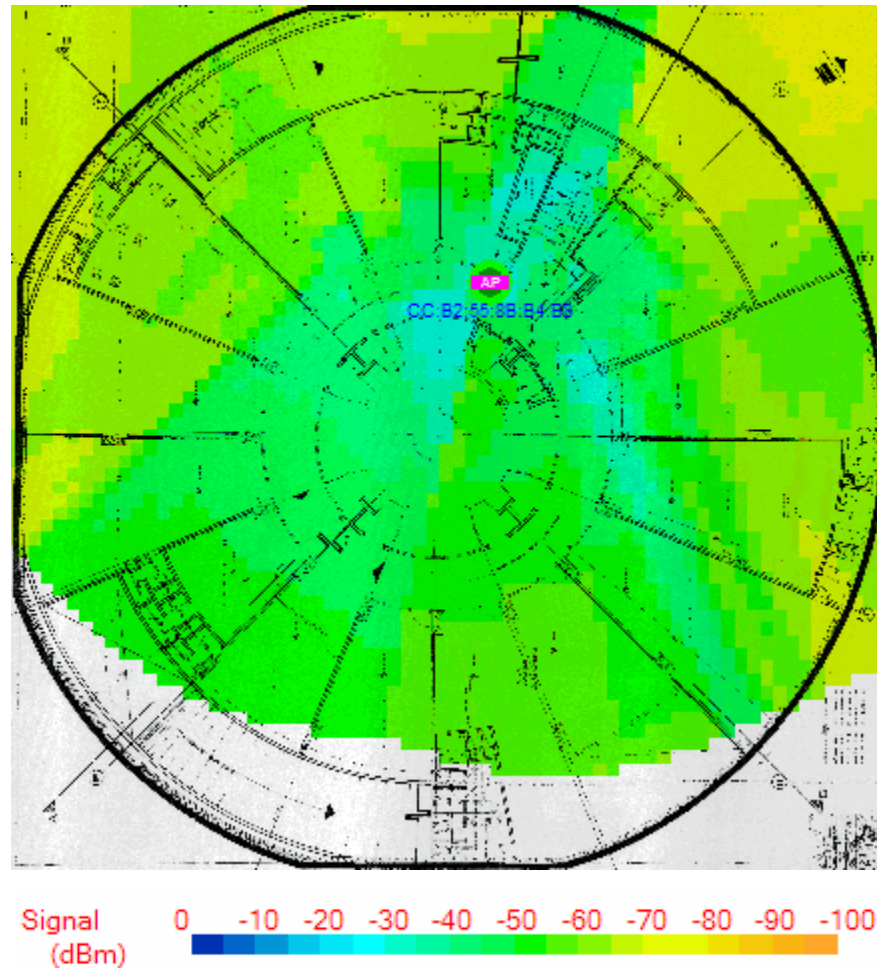
En el subsuelo uno, fueron ubicados tres APs, distribuyéndolos en base a los requerimientos, al revisar la figura 2.30 se observa que existe atenuación de la señal, por lo cual es necesario ubicar un cuarto punto de acceso en este piso.



**Figura 2.30:** Survey Activo subsuelo uno, AP 2

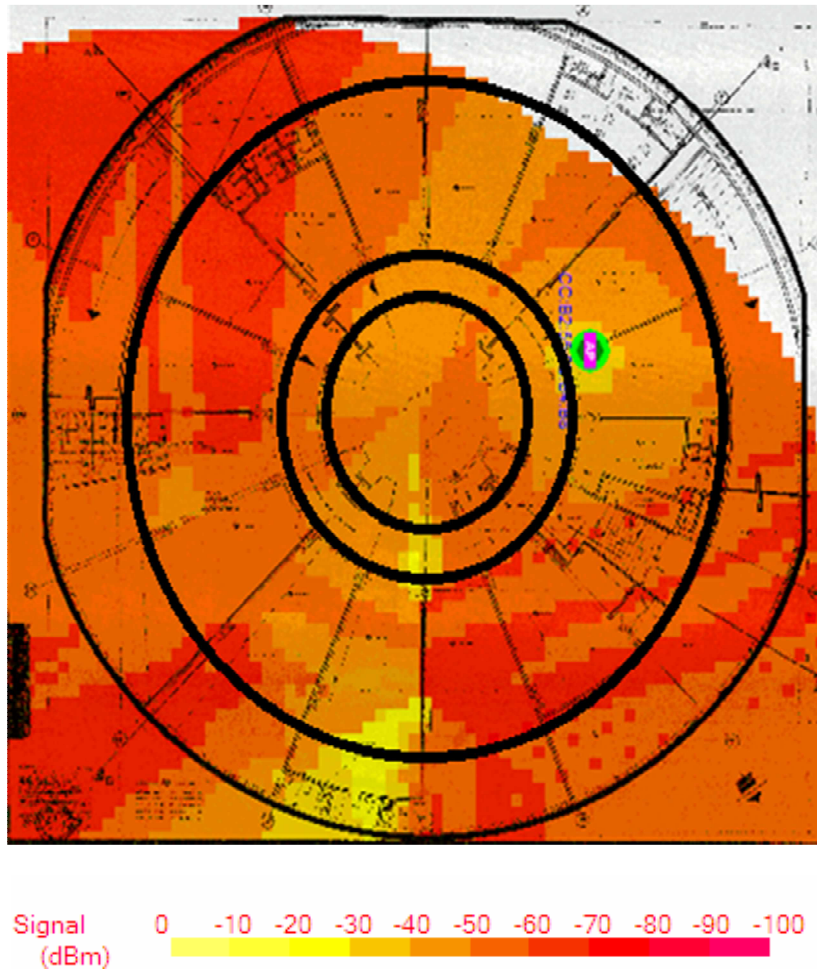
En la figura 2.31 se muestra el resultado del programa *Air Magnet Survey* al colocar el tercer punto de acceso en el subsuelo uno, como se observa, con este AP se elimina las atenuaciones que se tenía en el segundo equipo colocado.

En el nivel 0, 1, 2, 3, 4, 5 y 6, se colocaron dos *Access Points* por piso, al hacer las respectivas pruebas con la herramienta *Survey*, se comprobó que cumple con los requerimientos mínimos tanto de velocidad, como de número de usuarios conectados por cada piso.



**Figura 2.31:** *Survey* Activo subsuelo uno, AP-tres añadido

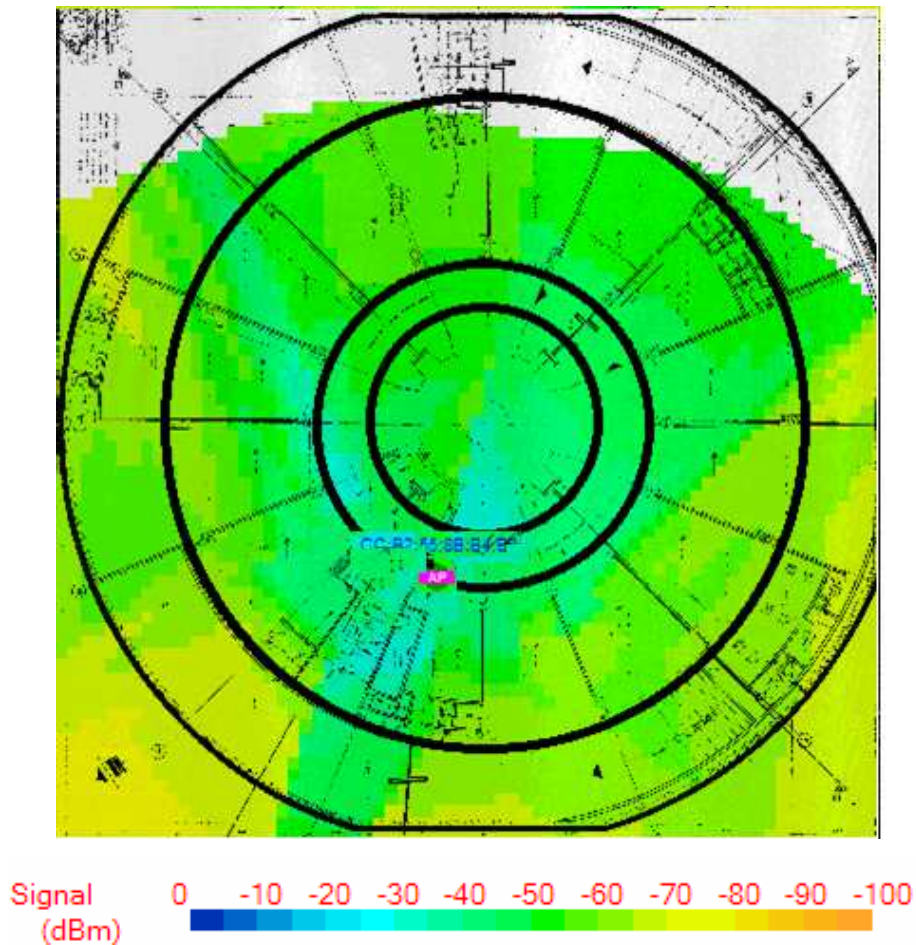
En la figura 2.32 se muestra el nivel de señal que refleja el primer *Access point* al realizar el *survey* activo en el nivel cero, donde se observa que la cobertura del mismo llega a todos los locales de este nivel, con una potencia mínima de -70 dBm, por lo que el AP será colocado en el lugar de prueba.



**Figura 2.32:** Distribución de APs en el nivel cero-uno

En la figura 2.33 se muestra el nivel de señal que refleja el segundo *Access point* colocado para realizar el *survey* activo, en la planta baja, como se observa en la misma, cumple con el nivel de señal y la velocidad requerida para el diseño de red, por lo cual con los dos APs colocados en el nivel cubren toda el área y cumple con el mínimo de usuarios en el mismo.

Para los siguientes niveles, (1, 2, 3, 4, 5 y 6), al ser idénticos al nivel cero, las figuras 2.31 y 2.32 describe los lugares donde se colocarán los puntos de acceso de cada piso, obteniendo como resultado el mismo nivel de potencia en cada uno de los APs colocados.

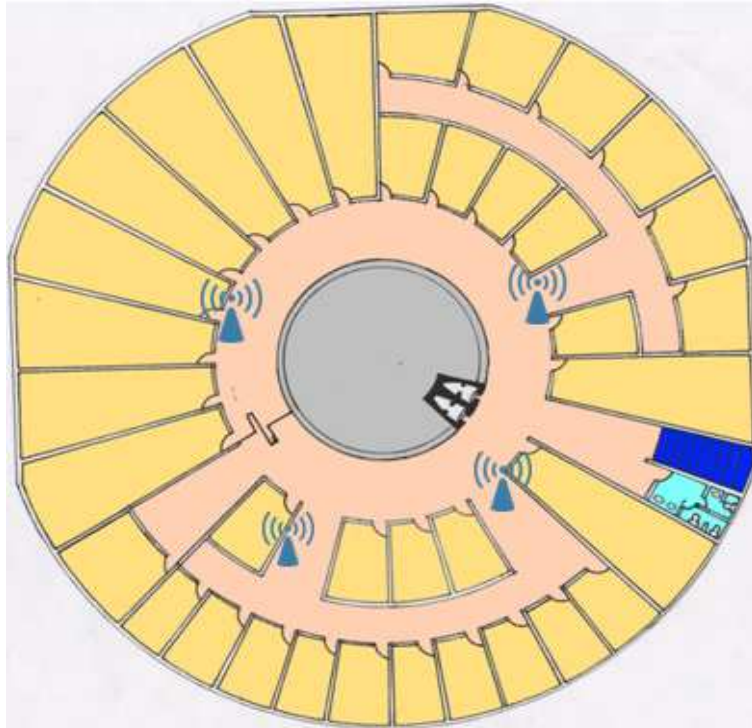


**Figura 2.33:** Distribución de APs en el nivel cero-dos

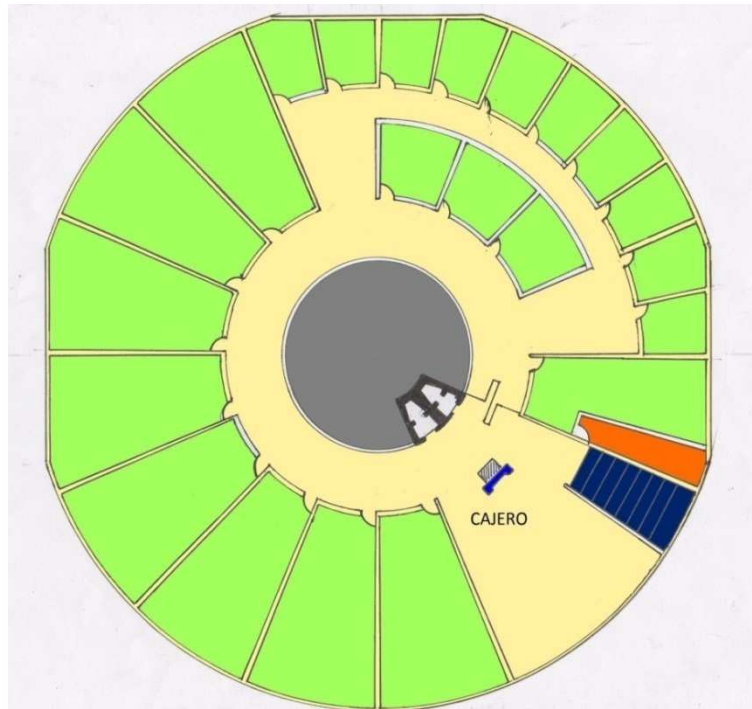
En la figura 2.34 se muestra esquemáticamente los lugares donde los puntos de acceso serán ubicados en función de lo obtenido al realizar el *survey* activo; para este caso, en el subsuelo dos, se colocarán cuatro APs, con los cuales se cumple con el requerimiento de cobertura, velocidad de transmisión y número de usuarios máximos conectados en cada *access point*.

En la figura 2.35 se muestra los lugares donde estarán ubicados los puntos de acceso en el subsuelo uno, de acuerdo a lo obtenido al realizar el *survey* activo. Para este caso serán colocados dos APs, con lo cual se cumple con los requerimientos de cobertura, velocidad de transmisión y número de usuarios máximos conectados en cada *access point* en este nivel.



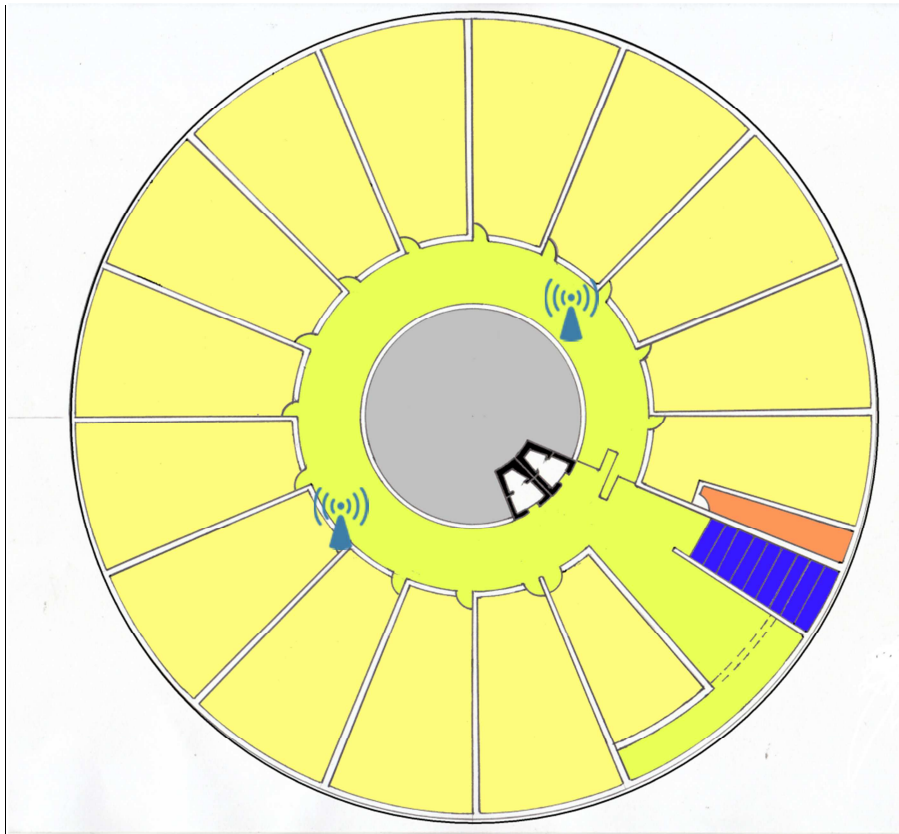


**Figura 2.34:** Distribución de APs en el Subsuelo 2. [18]



**Figura 2.35:** Distribución de APs en el Subsuelo 1 [18]

En la figura 2.36 se muestra la distribución de puntos de acceso para los niveles cero, uno, dos, tres, cuatro, cinco y seis, obtenidos al realizar el *survey* activo. Para estos casos se necesitan de dos APs por cada nivel para cumplir con los requerimientos de velocidad, cobertura, y usuarios máximos conectados en cada nivel.

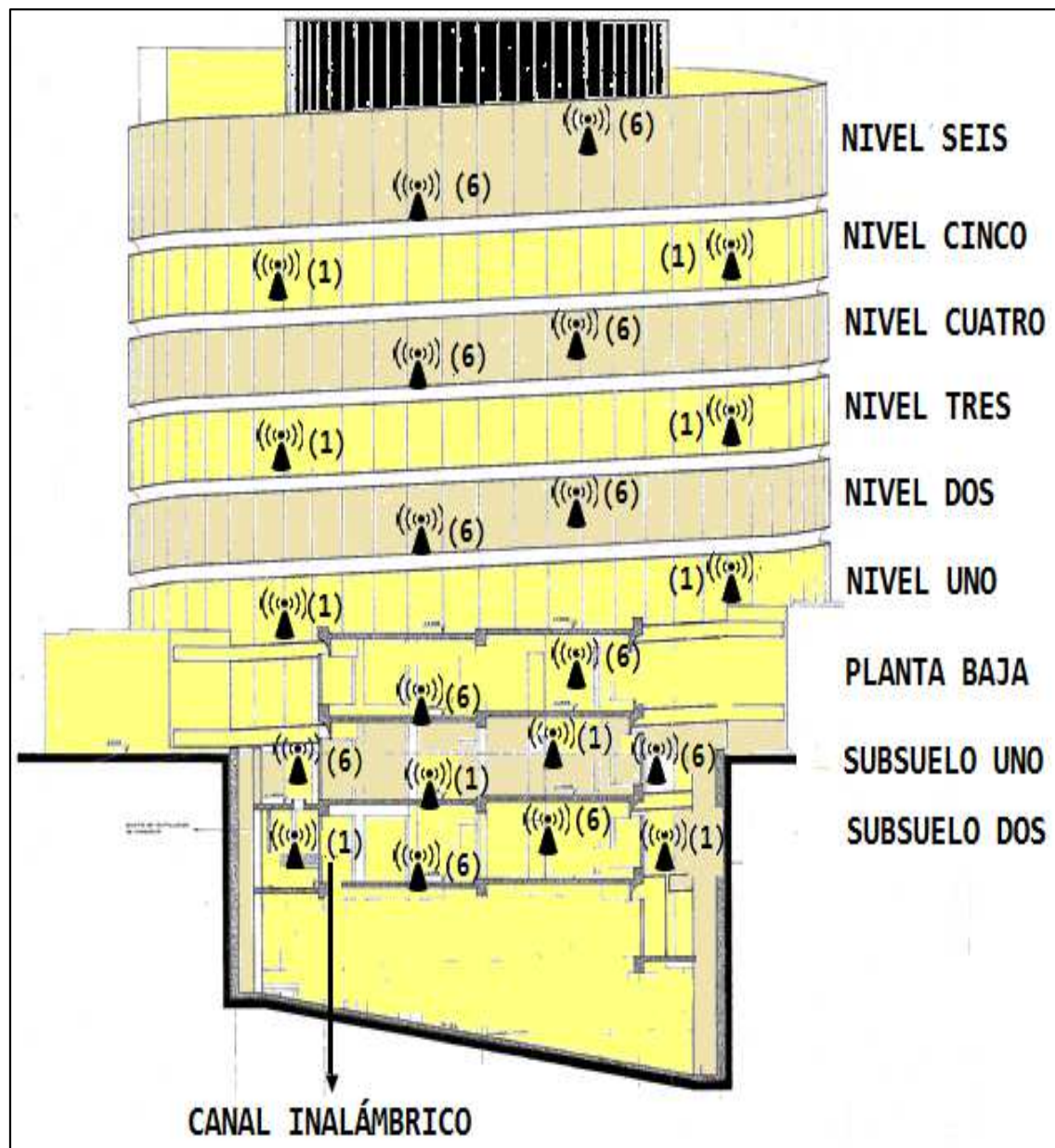


**Figura 2.36:** Distribución de APs en una planta Tipo [18]

La figura 2.37 muestra el planeamiento de frecuencias de cada uno de los puntos de acceso para garantizar que no exista interferencia entre los canales de operación.

De acuerdo al análisis y los resultados mostrados anteriormente se concluye que las características que cumplirán con los requerimientos y buen rendimiento de la red inalámbrica del centro comercial son los siguientes:

- Ambiente: Para interiores
- Número de usuarios: más de 429 simultáneos
- Velocidad de datos: 25,5 Mbps- 802.11g
- Rango de cobertura. 30 m
- Seguridad. WPA2-PSK



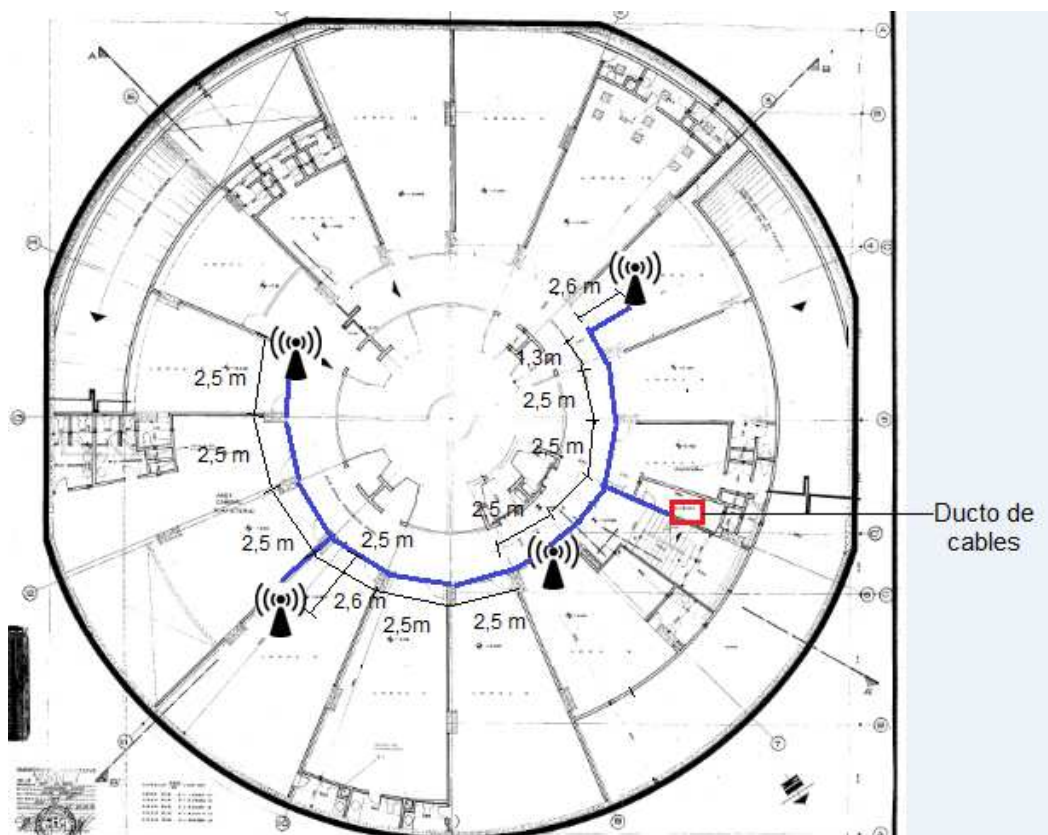
**Figura 2.37:** Planeamiento de frecuencias de la Red Inalámbrica

## 2.7 CABLEADO ESTRUCTURADO

Por las condiciones arquitectónicas del centro comercial el diseño del cableado estructurado para la presente red inalámbrica queda de la siguiente manera.

El cuarto de telecomunicaciones estará ubicado en la planta más alta junto a la administración, del análisis anteriormente realizado y en función del número de usuarios se colocarán veinte *Access Point* distribuidos de la siguiente manera: Dos *Access Point* por cada Piso, más dos APs adicionales en el Subsuelo 2, ya que en éste se encuentra el mayor número de usuarios y de locales.

Para llegar al cuarto de telecomunicaciones, cada AP estará conectado por medio de cable UTP cat 6A, el cual estará tendido a través de canaletas hasta el cuarto de servicios de cada piso, en donde existe un ducto de cables que recorren todos los pisos del centro comercial. Como se muestra a continuación en las figuras 2.38, 2.39, 2.40 y 2.41.



**Figura 2.38:** Distribución del cableado horizontal para la planta Subsuelo Dos



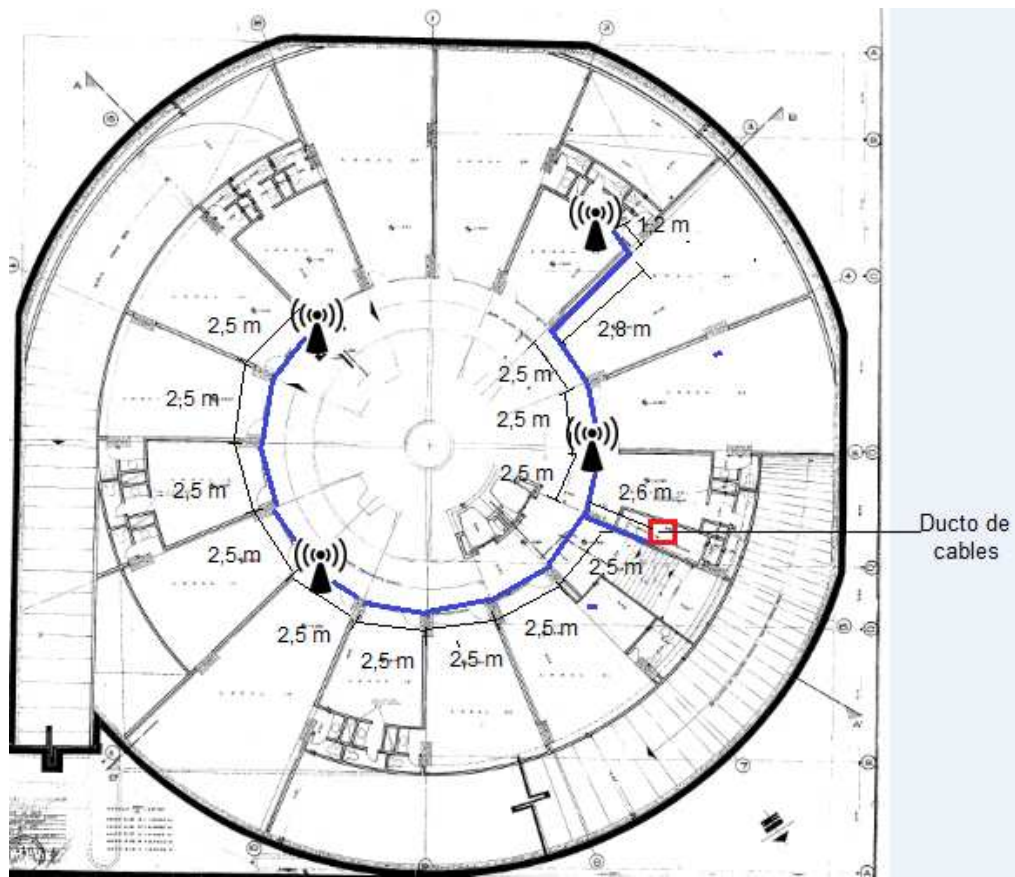


Figura 2.39: Distribución del cableado horizontal para la planta Subsuelo Uno

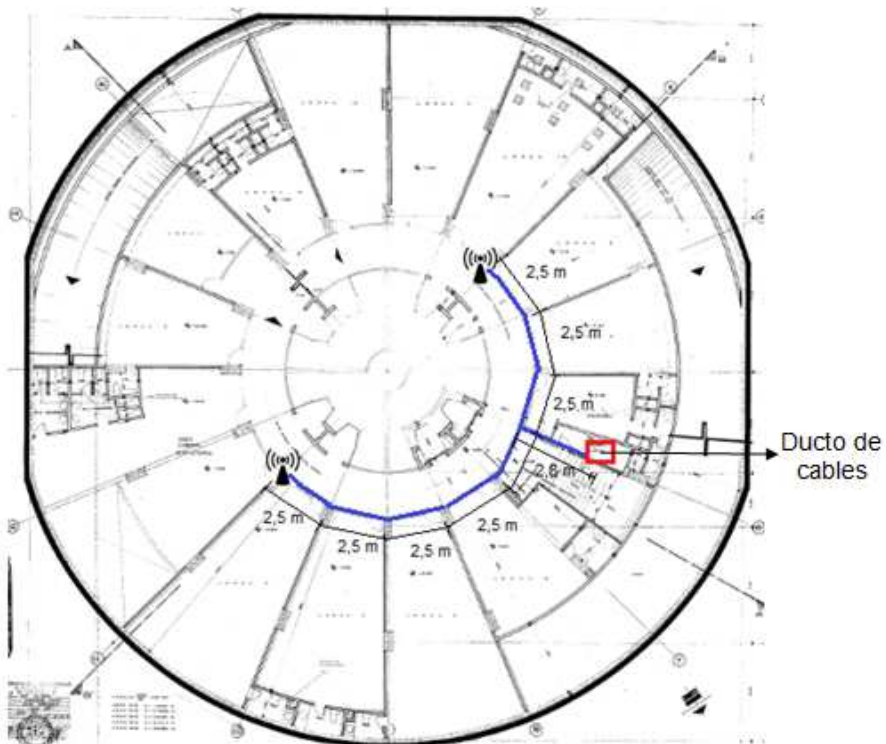
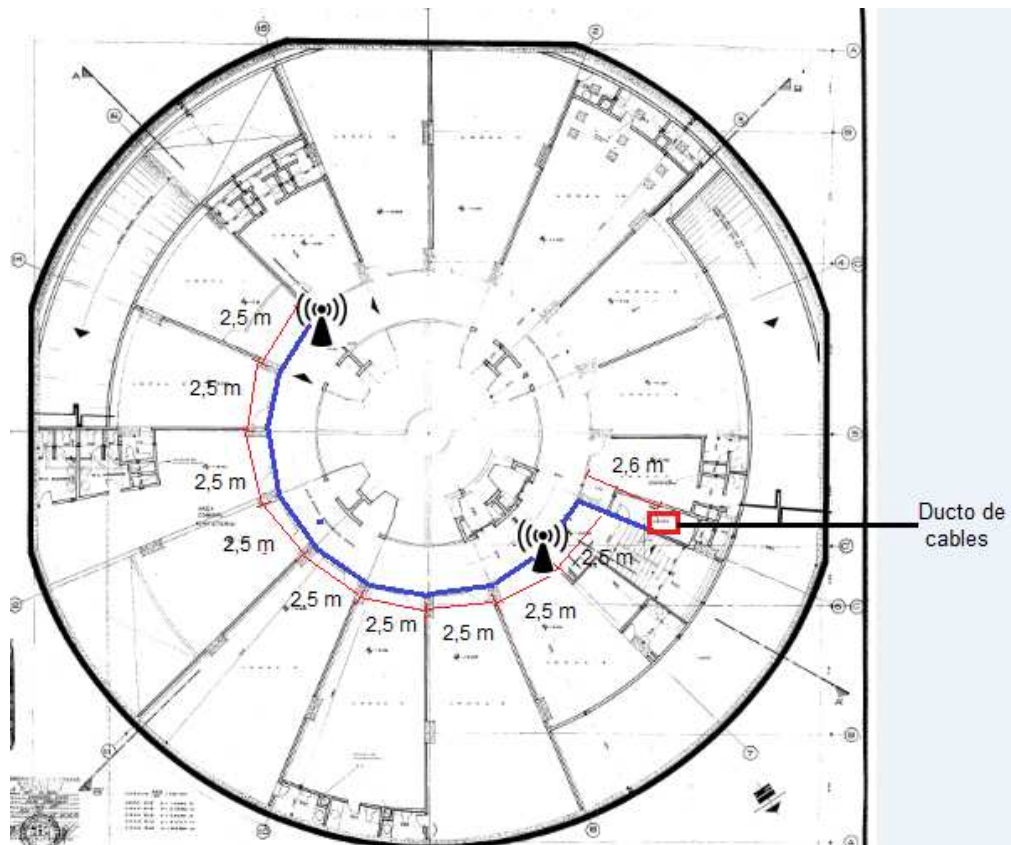


Figura 2.40: Distribución del cableado horizontal para las plantas 0,2,4,6



**Figura 2.41:** Distribución del cableado horizontal para las plantas 1,3,5

La tabla 2.16 muestra la cantidad de cable UTP cat 6A y canaletas necesarias por piso respectivamente, incluyendo dos metros de holgura en cable por cada piso.

PISO	CABLE UTP cat 6A[m]	CANALETAS	
		[m]	N°
<b>SUBSUELO DOS</b>	60,3	31,5	11
<b>SUBSUELO UNO</b>	61,4	36,6	13
<b>PLANTA BAJA</b>	27,2	22,6	8
<b>NIVEL UNO</b>	32,2	25,1	9
<b>NIVEL DOS</b>	27,2	22,6	8
<b>NIVEL TRES</b>	32,2	25,1	9
<b>NIVEL CUATRO</b>	27,2	22,6	8
<b>NIVEL CINCO</b>	32,2	25,1	9
<b>NIVEL SEIS</b>	27,2	22,6	8
<b>TOTAL</b>	327,1	202,3	83

**Tabla 2.16:** Resumen del cableado horizontal por piso

La figura 2.42 muestra distribución del cableado vertical que permite la interconexión que permite la interconexión de cada piso hacia el cuarto de telecomunicaciones.



**Figura 2.42:** Distribución del cableado estructurado para la interconexión de los *Access Point* hacia el Cuarto de Telecomunicaciones

PISO	CABLE UTP cat 6A [m]
<b>SUBSUELO DOS</b>	88
<b>SUBSUELO UNO</b>	80
<b>PLANTA BAJA</b>	36
<b>NIVEL UNO</b>	32
<b>NIVEL DOS</b>	28
<b>NIVEL TRES</b>	24
<b>NIVEL CUATRO</b>	20
<b>NIVEL CINCO</b>	16
TOTAL	324

**Tabla 2.17:** Resumen del cableado vertical por piso

Es decir que sumando el cableado horizontal y el vertical, el total de cable UTP cat 6a es de 633 m.

La tabla 2.18 muestra el resumen de todos los elementos que intervienen en el cableado estructurado del Centro Comercial Espiral.

ELEMENTOS	CANTIDAD
<b>Canaleta plana 24x14 mm</b>	83
<b>Unión plana 24x14 mm</b>	90
<b>Ángulo interno 24x14 mm</b>	20
<b>T plana 24x14 mm</b>	10
<b>Cajetín Superficial 4x2</b>	20
<b>Face Plates</b>	20
<b>Jacks RJ45</b>	60
<b>Cable UTP cat 6A (m)</b>	633
<b>Rollos de cable UTP cat 6A</b>	2

**Tabla 2.18:** Resumen del cableado estructurado del Centro Comercial

La tabla 2.19 muestra los elementos del área de trabajo, de cada una de salidas a las que se conectan los *Access Point*.

ELEMENTOS	CANTIDAD
<b>Cajetin Superficial 4x2</b>	20
<b>Face Plates</b>	20
<b>Jacks RJ45</b>	60
<b>Patch Cord (1.5 m)</b>	20

**Tabla 2.19:** Resumen de los elementos de área de trabajo

La tabla 2.20 muestra los equipos de comunicaciones del *rack* y su correspondiente dimensionamiento de la siguiente manera:



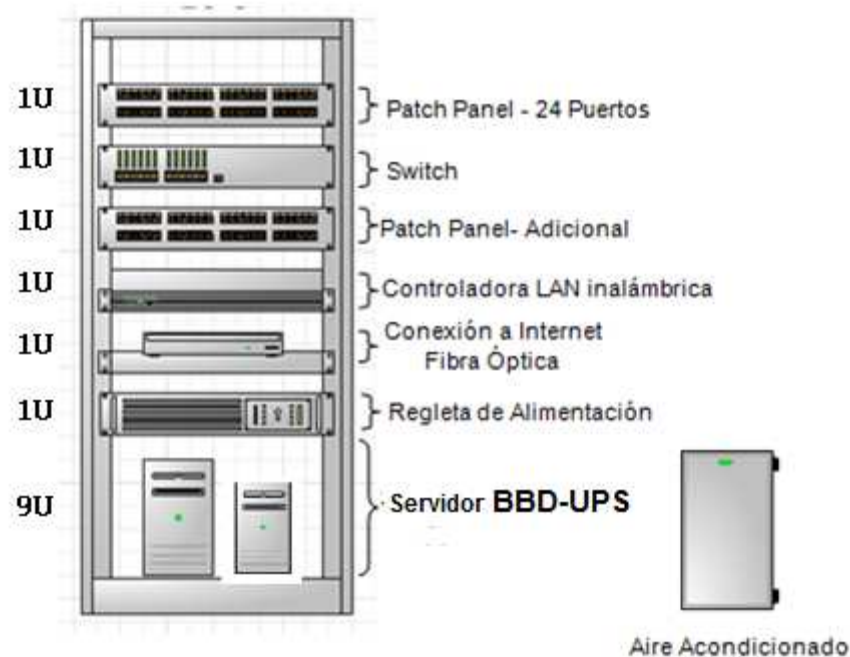
EQUIPO	CANTIDAD	UNIDAD DE RACK	DIMENSIÓN (cm)
Patch Panel	2	2U	8,88
Switch	1	1U	4,44
Controladora WLAN	1	1U	4,44
Regleta de Alimentación	1	1U	4,44
Equipo conexión Internet	1	1U	4,44
UPS-Servidor BBD	1	9U	39,60
		<b>TOTAL</b>	66,24

**Tabla 2.20:** Cálculo de unidades para dimensionar el *Rack* de equipos

Tipo	Altura estándar (cm)
<i>Rack de Piso</i>	80
	100
	120
	140
	160
	200
	220
<i>Rack de Pared</i>	30
	40
	45
	50
	55
	60

**Tabla 2.21:** Altura Normalizada del *Rack* de equipos [19]

Por lo tanto el *rack* de equipos para la red inalámbrica tendrá una altura estándar de 80 cm y su distribución se muestra en la figura 2.43.



**Figura 2.43:** Rack de comunicaciones

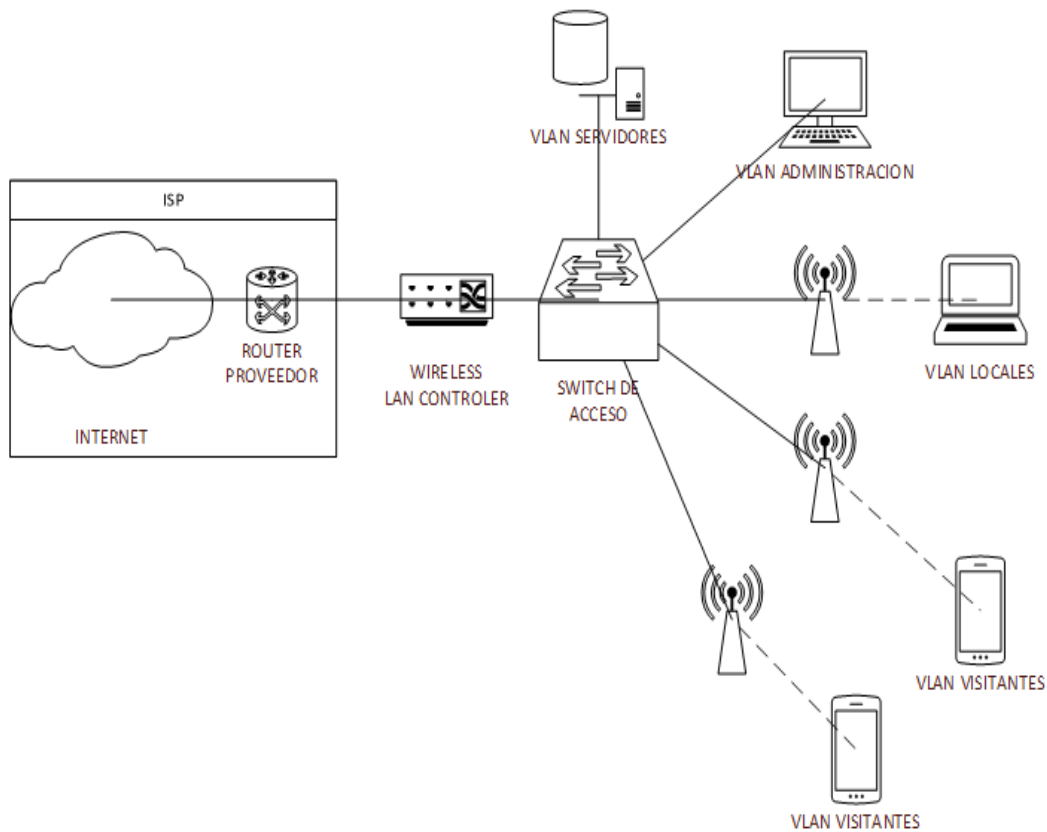
## 2.8 DIRECCIONAMIENTO

### 2.8.1 VLANS A CREARSE

Una vez obtenidos los tipos de usuarios que utilizarán la red, se crearán las siguientes VLANs:

- **VLAN Visitantes:** a esta podrán tener conexión todas las personas que visiten el Centro Comercial Espiral, ya sea para realizar una compra, averiguar precios o simplemente visitar el mismo.
- **VLAN locales:** a esta podrán acceder todos los empleados, dueños, y arrendatarios de los locales comerciales.
- **VLAN administración:** Esta red servirá para las personas encargadas de la administración y manejo de la red del centro comercial.

- **VLAN Servidores:** En esta red se encontrará el servidor de datos necesarios para el funcionamiento del programa que correrá en una *Tablet*. Además se podrá poner en un futuro otros servidores, como *Firewall*, *Radius*, entre otros.



**Figura 2.44:** Distribución Lógica de la red

### 2.8.2 DIRECCIONAMIENTO IP

Al conocer el número de VLANs, se procede a realizar el direccionamiento IP para cada una de ellas. El total de usuarios de la red entre visitantes y usuarios de los locales, se suma un total de 604 posibles usuarios conectados simultáneamente.

Se utilizará una dirección IPv4 de clase B, dentro del rango de direcciones privadas, para cumplir con el requerimiento del número de usuarios.

Para tener un mejor desempeño de la red, se realizará el subneteo de la dirección clase B, de esta forma se reduce el tráfico de broadcast.

En la tabla 2.22 se muestra el direccionamiento IP para cada VLAN, incluyendo la dirección de subred, la máscara de subred, el número de host disponibles para usar, el rango de direcciones disponibles y la puerta de enlace de la red.

Para el caso de la VLAN Visitantes, se tiene 300 posibles usuarios conectados en la red, se utilizará la dirección 172.16.0.0 con máscara 255.255.254.0, con el cual se tiene 510 direcciones disponibles para *host*. La puerta de enlace será 172.16.0.1.

Para la VLAN locales, se necesita 304 direcciones, que serán asignadas a cada uno de los empleados y/o dueños de los locales comerciales. Luego de hacer el subneteo se obtiene la dirección 172.16.1.512 con máscara 255.255.254.0, teniendo 510 direcciones disponibles para *host*, la puerta de enlace será 172.16.1.513.

Para la VLAN de administración se necesita 24 direcciones, 19 para la administración de cada AP, una para el administrador del centro comercial, una para el administrador de la red, y se puede agregar dos para pruebas. Al realizar el subneteo se obtiene la dirección 172.16.4.0 con máscara 255.255.255.248, con 30 direcciones utilizables para *host*, y puerta de enlace 172.16.4.1

Para la red donde se ubicará el servidor es preciso al menos una dirección IP, al realizar el subneteo se obtiene la dirección 172.16.4.32 con máscara 255.255.255.252, tiene 6 direcciones IP para *host*, y puerta de enlace 172.16.4.33.

La administración del *DHCP* (Protocolo de Configuración dinámica de *host*) para cada VLAN, se realizará en el controlar de red LAN inalámbrico.

VLAN	Subred	Máscara de Subred	Direcciones para Hosts	Rango de direcciones válidas	Puerta de enlace
Visitantes	172.16.0.0	255.255.254.0 (/23)	510	172.16.0.1-172.16.1.254	172.16.0.1
Locales	172.16.1.512	255.255.254.0 (/23)	510	172.16.2.1-172.16.3.254	172.16.2.1
Administración	172.16.4.0	255.255.255.248 (/27)	30	172.16.4.1-172.16.4.30	172.16.4.1
Servidores	172.16.4.32	255.255.255.252 (/29)	6	172.16.4.33-172.16.4.38	172.16.4.33

**Tabla 2.22:** Resumen del direccionamiento IP para cada VLAN

## 2.9 ESPECIFICACIONES DE LOS EQUIPOS

### 2.9.1 ELECCIÓN DEL PUNTO DE ACCESO

Al momento de escoger los puntos de acceso para el diseño de la red inalámbrica, se comparará con tres modelos de marcas diferentes; para lo cual se tomará en cuenta las siguientes especificaciones mínimas necesarias para este diseño:

- **Equipo para ambiente enlaces *In-Door*:** Debido a que todos los equipos serán instalados en el interior del centro comercial, es necesario utilizar equipos que cumplan esta característica.
- **Trabajo en la tecnología 802.11 g:** Este requerimiento es necesario de acuerdo a nuestro diseño y análisis del punto 2.5.
- **Soporte de VLANs:** El punto de acceso debe manejar Redes de Área Local privadas virtuales, con el fin de solventar lo analizado en el punto 2.7.1.2.
- **Soporte al menos 25 usuarios:** El equipo debe cumplir con este ítem en función de lo mencionado anteriormente, en donde se consideró esta

cantidad de usuarios del equipo para el diseño de la red, aunque en las hojas de datos se tenga un número de usuarios superior a 500.




- **Certificación WiFi:** Para que cualquier dispositivo móvil que tenga esta certificación pueda conectarse a la misma sin ningún problema.
- **Puerto Fast Ethernet:** Debido a que en el caso más crítico, la velocidad que manejará será superior a 10 Mbps.
- **802.3af (Power over Ethernet):** Al momento de instalar los puntos de acceso, serán ubicados en lugares estratégicos, en donde se imposibilita la instalación de un punto eléctrico, y además por el diseño del centro comercial dificulta tener cerca un tomacorriente. Por lo que será necesario alimentar al dispositivo a través del mismo punto de red.
- **LWAPP (Light Weight Access Point Protocol):** Protocolo que permite la comunicación entre el punto de acceso y el controlador de LAN inalámbrico.
- **Antenas:** Se utilizará las antenas integradas en el punto de acceso, de tipo omnidireccionales, con una ganancia mínima de 2 dBi, la cual basta para cubrir el área asignada a los APs de cada piso, tal como se observó al realizar el *survey* activo, ya que para realizarlo se utilizó un *Access point* con este tipo de antena y ganancia. El uso de una antena sectorial es recomendable utilizar en ambientes *out door*, y si se requiere cubrir una área mayor a la actual.

La tabla 2.23 muestra las características mínimas de los puntos de acceso

Actualmente se puede escoger entre diversas marcas y modelos, variando su costo de acuerdo a varios aspectos, por lo que se debe considerar las características que se necesitan, y comparar el factor costo beneficio entre los modelos mostrados en la tabla 2.24.

PUNTO DE ACCESO		
ESPECIFICACIONES	CARACTERÍSTICAS MÍNIMAS	JUSTIFICACIÓN
Características Generales	<i>In door</i>	Los APs serán utilizado dentro del centro comercial
	Administrable	Configuración de parámetros de la red.
	Certificación <i>Wifi</i>	Compatibilidad con dispositivos <i>WiFi</i>
	Puerto <i>Fast Ethernet</i>	Conexión hacia el <i>Switch</i>
	25 usuarios simultáneos	Usuarios que estarán conectados en cada AP
	Antena omnidireccional integrada de 2 dBi	Obtener la señal y velocidad requerida.
Protocolos y Aplicaciones Soportadas	802.11g	Estándar utilizado para el diseño
	<i>Power over Ethernet</i> 802.3 af	Alimentación eléctrica del AP a través de la red.
	LWAPP	Protocolo de comunicación entre el AP y el WLC

**Tabla 2.23:** Requerimientos mínimos de los Puntos de Acceso

	<b>CISCO AIRONET 1130G [20]</b>	<b>DLINK DAP 2360 [21]</b>	<b>HP MSM430 [22]</b>
<b>Características</b>			
<b>Soporta 802.11 g</b>	Si	Si	Si
<b>Frecuencia de Operación</b>	2.4 Ghz	2.4 Ghz	2.4 Ghz
<b>Canales</b>	1 – 11 canales	1 – 11 canales	1-11 canales
<b>Puerto Ethernet</b>	10BaseT 100BaseTX	10/100/1000 Base TX	10/100/1000
<b>Antenas</b>	4 Antenas internas omnidireccionales	2 Antenas Externas Dipolo 5dBi	6 Antenas internas omnidireccionales
<b>PoE</b>	Si	Si	Si
<b>Soporte de Vlan</b>	16	32	80
<b>LWAPP</b>	Si	No	Si
<b>Múltiples SSID</b>	Si	Si	Si
<b>Seguridad</b>	WPA/WPA2/WEP 802.1x, WPS, ACL, Filtrado MAC	WPA/WPA2/WEP 802.1x, Filtrado MAC	WPA/WPA2/WE P 802.1x, Filtrado MAC,
<b>Administración</b>	HTTPS	HTTP, HTTPS, TELNET, SSH,	HTTPS, CLI
<b>Garantía</b>	<i>Lifetime</i>	<i>Lifetime</i>	<i>Lifetime</i>

**Tabla 2.24:** Comparación de tres marcas de AP's

### 2.9.2 ELECCIÓN DEL SWITCH DE ACCESO

Para elegir el mejor *switch* de acceso se tomará en cuenta que se requiere una velocidad total de backbone de 190 Mbps, ésta es la suma total de *Access Point* que estarán conectados al equipo, más la velocidad para acceder al servidor de datos que estará en el cuarto de telecomunicaciones.



Se han comparado dos marcas diferentes de modelos que cumplen con los parámetros para este diseño:

- Soporte de VLANs.
- Velocidad de *Backbone* de al menos 190 Mbps.
- Al menos 22 Puertos RJ45.
- *Switch* Administrables o semiadministrables.

La tabla 2.25 muestra las características mínimas necesarias del *Switch* de acceso para cumplir con los requerimientos de la red.

SWITCH DE ACCESO		
ESPECIFICACIONES	CARACTERÍSTICAS MÍNIMAS	JUSTIFICACIÓN
Características Generales	Puertos <i>Fast Ethernet PoE</i>	Para enlace y alimentación hacia los puntos de acceso inalámbrico
	Administrable	Necesidad de reconfiguración de puertos
	Capacidad de Actualización	Actualización de funcionalidades y sistema operativo
Protocolos y Aplicaciones Soportadas	SMTP V1, V2 y HTTP	Protocolos de gestión remota
	Soporte VLAN	Para facilidad de administración de la red
	<i>Ethernet, Fast Ethernet, Gigabit Ethernet</i>	Protocolos necesarios para la interconexión de datos
Alimentación	110 V, 50/60 Hz	Capacidad estandarizada para Ecuador

**Tabla 2.25:** Características mínimas del *Switch* de acceso

En la tabla 2.26 se dispone de una comparación de dos marcas reconocidas en el mercado para la elección del *Switch* que se utilizará para la red.

Características	HP 1910 24G [23]	CISCO CATALYST 2960 [24]
<b>Puertos</b>	24 Puertos autonegociables 10/100/1000 (10Base-T 100Base-TX 1000Base-T) PoE	24 puertos autonegociables 10/100/1000 (10 Base T 100Base-TX 1000Base-T) PoE
<b>Throughput</b>	41.7 Mbps	13.3 Mbps
<b>Backplane</b>	56 Gbps	16 Gbps
<b>VLANs</b>	Si	Si
<b>Enrutamiento</b>	Hasta 32 Rutas	Si
<b>Administración</b>	Web, CLI, SNMP, Ethernet MIB	Web, CLI, telnet, SNMP
<b>Estándares</b>	IEEE 802.1p, IEEE 802.1q, IEEE 802.3ab, IEEE 802.3af, IEEE 802.1d, IEEE 802.1x, IEEE 802.1w, IEEE 802.3	IEEE 802.1p, IEEE 802.1q, IEEE 802.1ab, IEEE 802.1af, IEEE 802.1d, IEEE 802.1x, IEEE 802.1w, IEEE 802.3ab, IEEE 802.1ab, IEEE 802.3ad
<b>Garantía</b>	3 años	2 años

**Tabla 2.26:** Comparación de dos marcas de *Switches*.

### 2.9.3 CONTROLADOR DE RED INALÁMBRICA

Debido al número de *Access Points* que se van a manejar, y para mantener una administración eficiente de la red inalámbrica y de todos sus recursos es necesario el uso de un *Wireless LAN Controller*.

Mediante este equipo se gestionarán los siguientes ítems:

- El acceso a cada una de las VLANs creadas, de acuerdo a los requerimientos de tráfico mencionados anteriormente, manteniendo como prioritario el tráfico de los locales del centro comercial.
- El DHCP de cada VLAN de la red diseñada.
- Se utilizará como cortafuegos para controlar el tráfico y filtrar el acceso a los servicios de red y los servicios de movilidad (*roaming*), así mismo el acceso a invitados, seguridad.

- El número máximo de usuarios que podrán conectarse en cada punto de acceso mientras cumpla con la capacidad requerida para cada usuario.
- La movilidad entre cada piso del Centro Comercial Espiral, asegurando que los invitados no pierdan la conexión con la red inalámbrica mientras se encuentren conectados, ya sea para usar la aplicación para la *tablet*, o cualquier otro servicio.

<b>CONTROLADOR LAN INALÁMBRICO</b>		
<b>ESPECIFICACIONES</b>	<b>CARACTERÍSTICAS MÍNIMAS</b>	<b>JUSTIFICACIÓN</b>
Características Generales	Manejo de 22 puntos de acceso	Número de APs que se utilizará en la red.
	Administrable	Configurar en el dispositivo sus servicios ofrecidos.
	Capacidad de Actualización	Actualización de funcionalidades y sistema operativo
Protocolos y Aplicaciones Soportadas	DHCP	Servicio que se manejará desde el <i>LAN controller</i>
	Soporte VLAN	Manejar usuarios distribuidos en la red.
	Un puerto <i>Fast Ethernet</i> ,	Protocolo necesario para la interconexión de datos en la red
	LWAPP	Protocolo de comunicación entre el <i>AP</i> y el <i>WLC</i>
Alimentación	110 V, 50/60 Hz	Capacidad estandarizada para Ecuador

**Tabla 2.27:** Requisitos mínimos que debe cumplir el *LAN Controller* a escoger

La tabla 2.28 muestra la comparación entre equipos de similares características y de dos marcas reconocidas como son HP y Cisco.

<b>Características</b>	<b>HP MSM 720 [25]</b>	<b>CISCO WLC 4402 [26]</b>
<b>Estándares que soporta</b>	802.11 a, 802.11b, 802.11g, 802.11n	802.11 a, 802.11b, 802.11g, 802.11n
<b>Puertos</b>	4 RJ-45 autosensing 10/100/1000, 10BaseT, 100BaseTX, 1000BaseT 2 RJ-45	2 RJ-45 <i>autosensing</i> 1000BaseX 1 RJ-45 Puerto de servicio 10/100 1 RJ-45 10/100/1000 Puerto de utilidad. 1 RS232 Puerto de Consola
<b>Número de APs</b>	De 10 a 40	12, 25 y 50
<b>Velocidad de Backplane</b>	12 Gbps	10 Gbps
<b>Soporte de VLANs</b>	Si	Si
<b>Firewall</b>	Si	Si
<b>Medio de Administración</b>	Web, CLI, SNMP	Web, CLI, SNMP

**Tabla 2.28:** Comparación de dos marcas de *Wireless LAN Controllers*

#### 2.9.4 DIMENSIONAMIENTO DEL SERVIDOR DE BASE DE DATOS

La tabla 2.29 muestra el dimensionamiento del procesador de la base de datos considerando los siguientes parámetros:

- **Porcentaje de utilización del CPU:** 12% (Dato tomado del dispositivo *gadget* de windows que permite medir el porcentaje de utilización al únicamente realizar una consulta hacia la base de datos).
- **Velocidad del CPU:** 2500[MHz] (Velocidad del procesador con el que se realizo el análisis).
- **Operaciones por segundo:** una consulta en 0,5 [s] (Dato obtenido en la interfaz de *MySQL*).
- **Peticiones por operación:** 2 debido a que en cada consulta recibe la petición y envía la respuesta.
- **Simultaneidad:** 418 usuarios concurrentes (Del análisis realizado en el literal 2.2).

La tabla 2.30 muestra el cálculo de la memoria RAM con los siguientes parámetros:

- **Número de conexiones:** 418 es decir una por cada usuario.
- **Servicio:** Uno porque es el acceso hacia la base de datos.
- **Solicitud:** Dos debido a que cada consulta necesita de una petición y envío de datos.
- **Tamaño de conexión:** 2,33 [KB] (Dato obtenido de la figura 2.23).

USO DE CPU						
<b>USO CPU</b> [MHz]	=	% USO CPU	x	# CPU	X	Vel. CPU [MHz]
		0,12		1		2500
		300 MHz				
<b>Operaciones por Segundo</b> [Op / (s x Usuario)]	=	Operaciones		÷		Segundo
		1				0,5
		2 [Op / (s x usuario)]				
<b>Peticiones por Operación</b> [Pet / Op]	=	2 [Pet/Op]				
<b>Peticiones por segundo</b> [Pet / s]	=	# Peticiones	÷	Tiempo pet. [s]	X	Simultaneidad
		1		0,5		418
		209 [Pet/s]				
<b>Consumo de CPU por Operación</b> [MHz x Op x s]	=	Uso CPU	X	Peticiones por operación	÷	Peticiones por segundo
		300		2		209
		2,87 [MHz x Op x s]				
<b>Consumo CPU por usuario</b> [MHz/usuario]	=	Operaciones por Segundo		X		Consumo de CPU por Operación
		2				2,87
		5,74 [MHz/usuario]				
<b>Umbral de utilización del CPU</b>	=	Usuarios concurrentes		X		Consumo CPU por usuario
		418				5,74
		2399,32 [MHz]				

**Tabla 2.29:** Dimensionamiento del CPU [27]

CÁLCULO DE MEMORIA RAM							
Tamaño RAM [KBytes]	=	#Conexiones	x	Servicio	x	Solicitud	Tamaño de conexión [KBytes]
		418		1		2	2,33
1947,88 [Kbytes]							

**Tabla 2.30:** Dimensionamiento de la memoria RAM [27]

La tabla 2.31 muestra el tamaño en el disco duro de la base de datos.

CÁLCULO DE DISCO DURO	
Espacio en Disco [GBytes]	=
Tamaño de la Base de Datos	
1,72 [GBytes]	

**Tabla 2.31:** Dimensionamiento del Disco Duro [27]

En la tabla 2.32 con las características mínimas del servidor y su comparación con dos marcas disponibles en el mercado.

CARACTERÍSTICAS	DELL POWEREDGE T110 [28]	HP PROLIANT ML115 G5 [29]
<b>Procesador:</b> 2500 [MHz]	Cumple Procesador: Intel Xeon 3.1 [GHz]	Cumple Procesador: AMD Opteron 3 [Ghz]
<b>Disco Duro:</b> 1 [TB]	Cumple Disco Duro: 2TB SATA	Cumple Disco Duro: 3TB SATA
<b>Memoria RAM:</b> 4 [GB]	Cumple Memoria RAM: 4 [GB]	Cumple Memoria RAM: 8 [GB]
<b>Puerto <i>Fast Ethernet</i></b>	Cumple	Cumple
<b>Garantía:</b> 1año	Cumple	Cumple

**Tabla 2.32:** Comparación características mínimas del servidor de BDD

# CAPÍTULO 3



## 3. DESARROLLO DE LA APLICACIÓN

### 3.1 REQUERIMIENTOS DE LA APLICACIÓN

El presente proyecto surge bajo la necesidad de proporcionar al usuario en general una aplicación que le permita, ubicar de manera fácil y completamente sencilla locales, ofertas y productos con sus respectivos precios y características que disponen los diferentes locales del Centro Comercial Espiral de nuestra ciudad, para lo cual se considera lo siguiente:

- La interfaz y usabilidad debe ser clara y sencilla de menús fáciles y entendibles para el usuario.
- La navegación entre las diferentes pantallas debe ser manejable de acuerdo con la información que necesita el usuario.
- Integración en el dispositivo, debe ser adecuada de acuerdo con las características que ofrece la *tablet* o dispositivo móvil Android.
- La aplicación debe ser adaptable a futuras modificaciones y ampliaciones.

La aplicación para el Centro Comercial Espiral está desarrollada bajo Eclipse el mismo que contendrá *SDK*, *plugins* y librerías necesarias para este tipo de aplicaciones.

### 3.2 DIAGRAMAS UML

Los Diagramas de Lenguaje Unificado de Modelado, están formados por diversos elementos gráficos para formar diagramas, con la finalidad de representar las diversas perspectivas de un sistema a través de un modelo puesto que son la representación simplificada de la realidad. Para este proyecto representan a cualquier cliente del Centro Comercial.

### 3.2.1 DIAGRAMA DE CASOS DE USO

El diagrama de la figura 3.1 representa todas las acciones que puede realizar el usuario al interactuar con la aplicación.

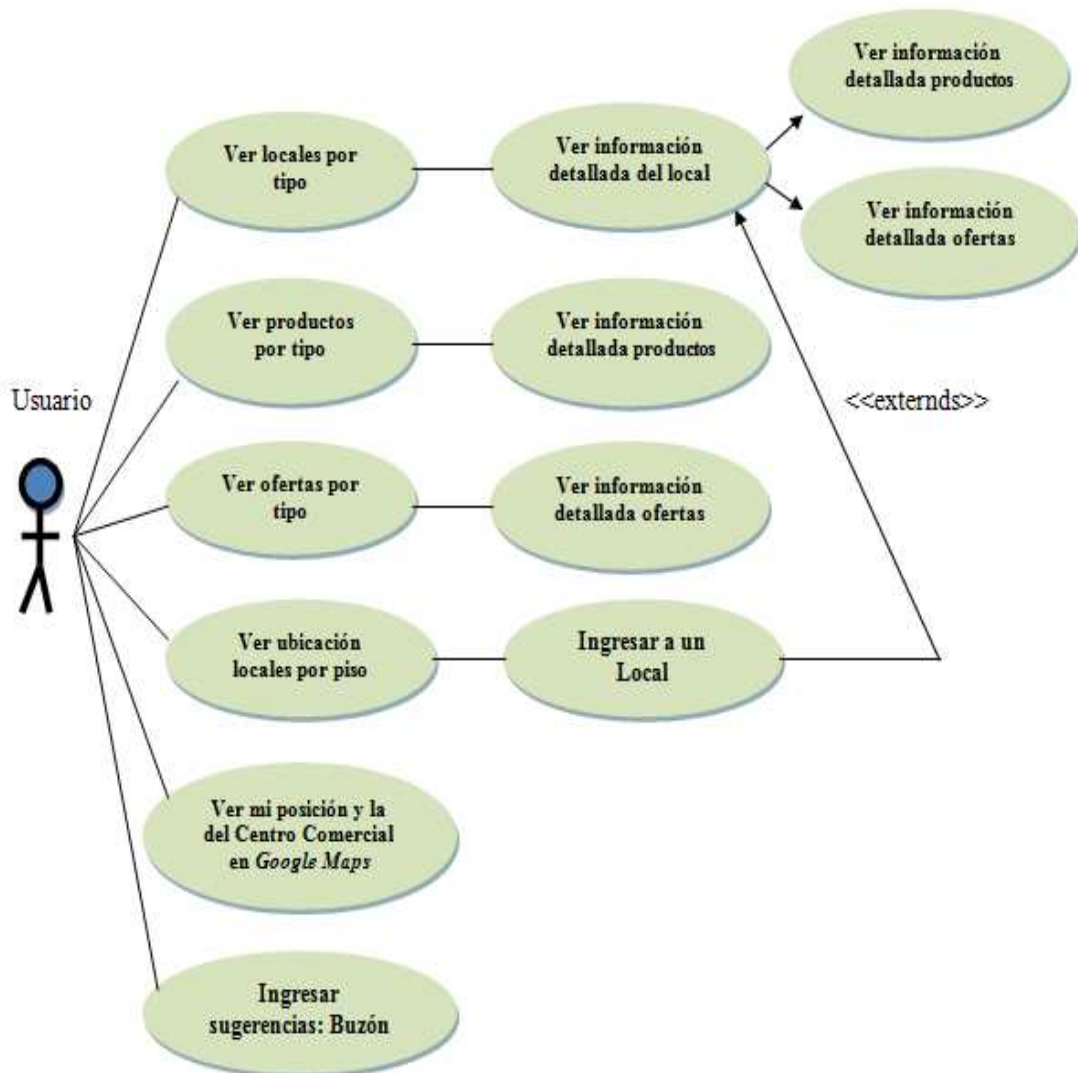
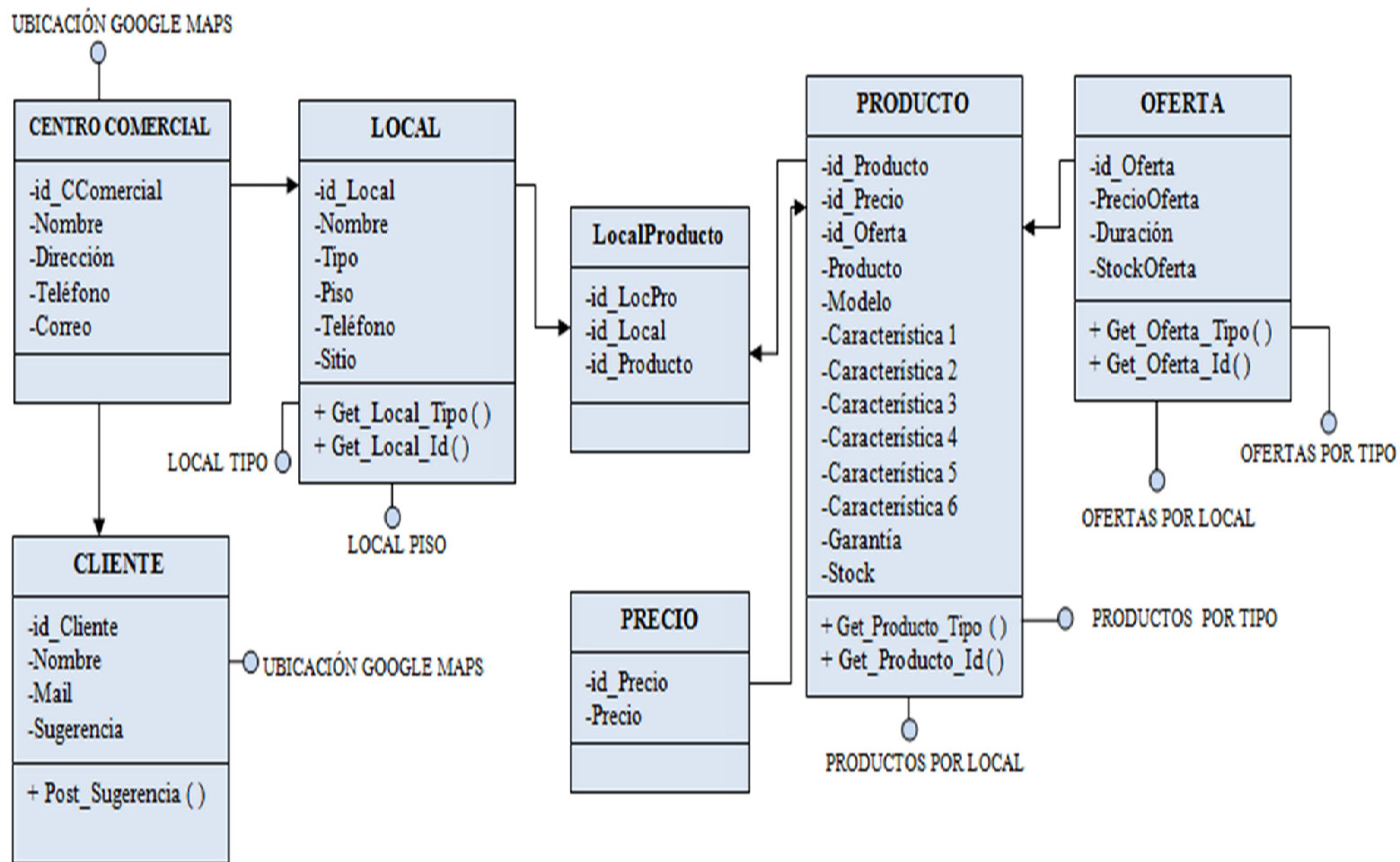


Figura 3.1: Diagrama de Casos de Uso de la Aplicación

### 3.2.2 DIAGRAMA DE CLASES

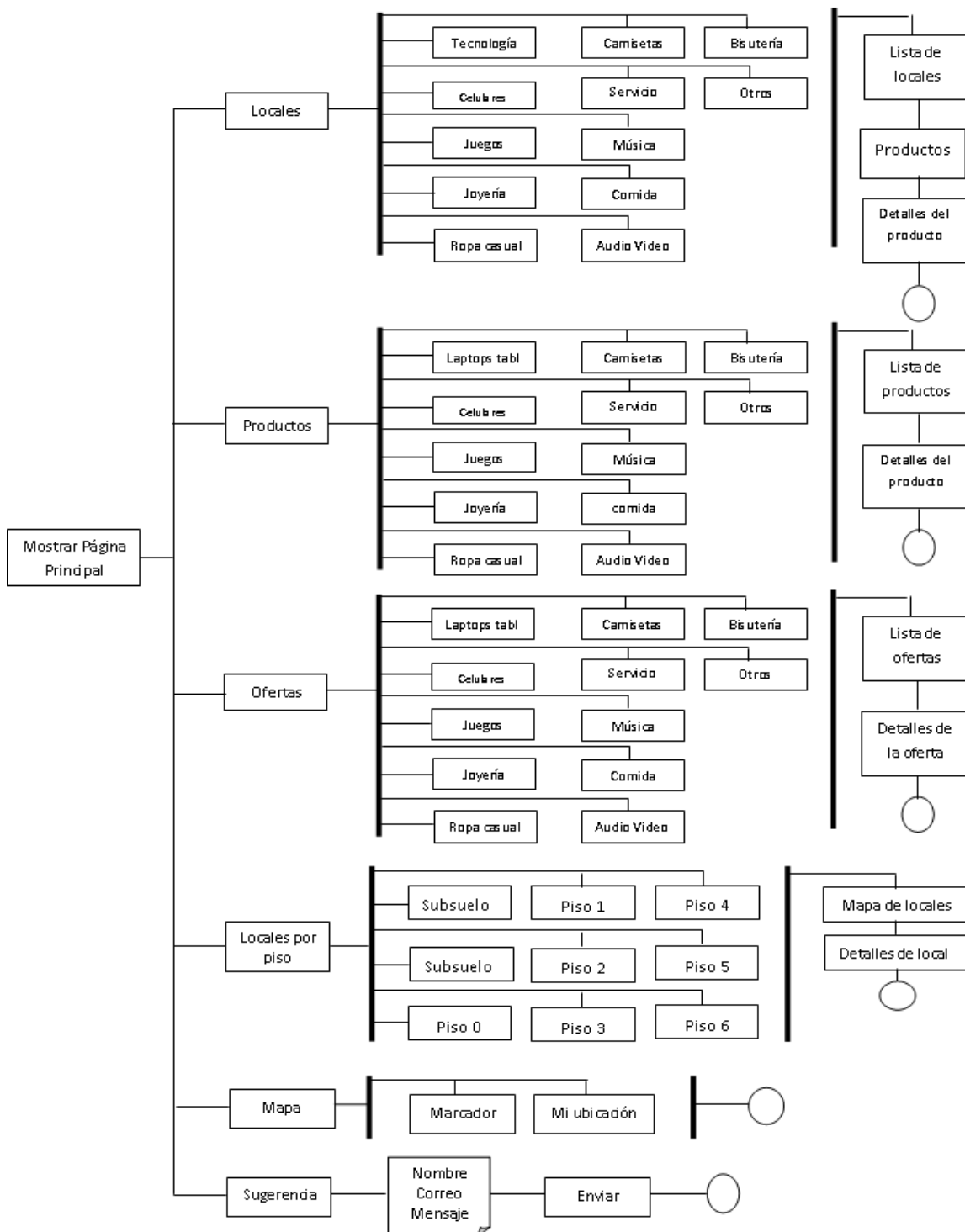
La figura 3.2 muestra el diagrama correspondiente a las clases que intervienen en la aplicación y su relación entre las funciones implementadas en cada una de las interfaces.



**Figura 3.2:** Diagrama de Clases de la Aplicación

### 3.2.3 DIAGRAMA DE ACTIVIDADES

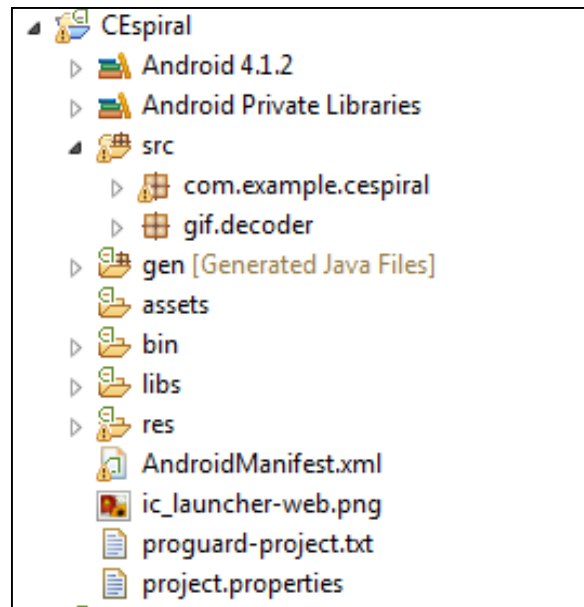
En la figura 3.3 se muestra el diagrama de actividades correspondiente a la aplicación.



**Figura 3.3:** Diagrama de Actividades de la aplicación

### 3.3 ESTRUCTURA DE LAS CARPETAS

Como se muestra en la figura 3.4 el sistema de directorio de un proyecto Android, los cuatros directorios más importantes son:



**Figura 3.4:** Estructura de Carpetas de un proyecto *Android*

- **SRC:** carpeta donde se incluyen las clases con el código fuente de la aplicación al igual que en cualquier otro proyecto Java. Este directorio se puede organizar en cualquier orden, y con cuantas subcarpetas como sean necesarias, con la peculiaridad de que, en un proyecto Android, las carpetas contenidas en *SRC* deben tener como mínimo dos niveles de profundidad. En este caso dentro de la carpeta *SRC* está la carpeta “*com.example.cespiral*”.
- **GEN:** carpeta generada y controlada automáticamente por el entorno desarrollo, no debe ser modificada puesto que se actualiza sola cada vez que se haga algún cambio dentro de la carpeta *RES*. Sirve, por lo tanto, como interfaz entre la carpeta *RES* y el código fuente contenido en la carpeta *SRC*, la interacción se permite gracias a la clase *R.Java*.  
Por ejemplo al acceder a un *textView*, para asignar un texto en específico, es decir de la siguiente manera:

```
“txtLaptop = (TextView) findViewById(R.id.txtLaptop);”
```

- **RES:** carpeta que contiene todo lo que no sea código Java, pero que se requiere que aparezca en la aplicación. Por defecto están creadas las carpetas *drawable* (para colocar imágenes), *layout* (para crear archivos XML que definen interfaces) y *values* (para colocar *Strings* y *arrays*).
- **Android Manifest.xml:** Se genera por defecto en un proyecto de *Android* y cumple el papel más importante dentro de la aplicación puesto que de él depende el correcto funcionamiento de la misma, contiene los permisos de los que hará uso la misma así como también la Actividad Principal y Actividades Secundarias, la explicación detallada del mismo se muestra más adelante.

### 3.4 JSON (*JAVA SCRIPT OBJECT NOTATION*) [30]

*JSON*, acrónimo de *JavaScript Object Notation*, es un formato ligero para el intercambio de datos, o a su vez un subconjunto de la notación literal de objetos de *JavaScript* que no requiere el uso de XML.

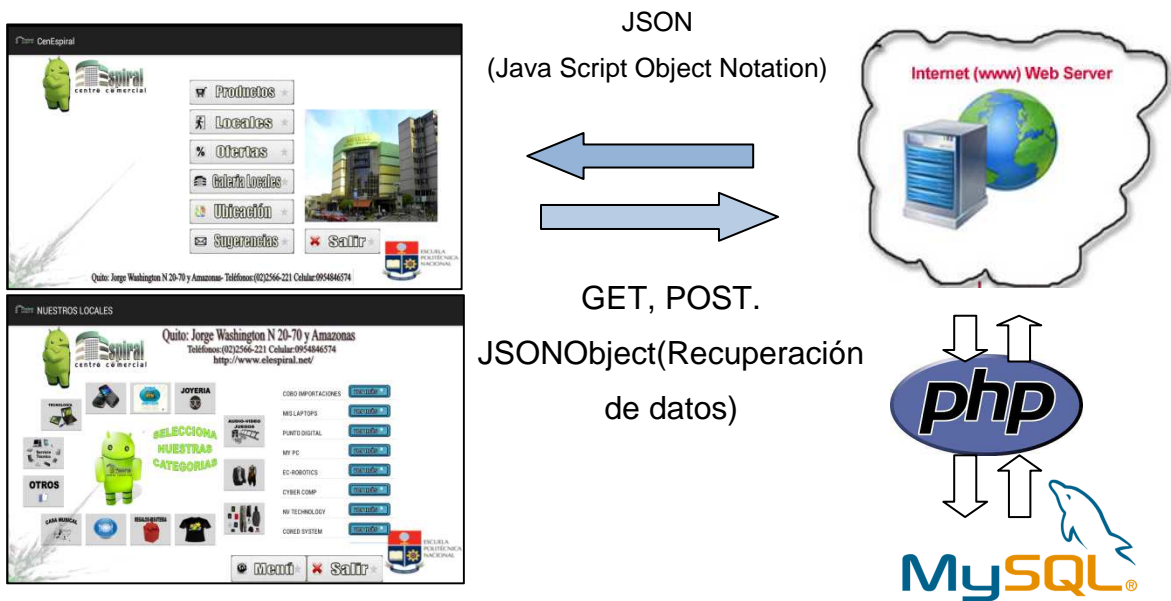
XML es una opción válida pero no es la más adecuada por ser demasiado pesada. La figura 3.5 muestra el esquema general de la aplicación y la utilización de la notación *JSON* para el manejo de datos.

Actualmente para el desarrollo de aplicaciones móviles y específicamente el manejo de datos requiere la optimización de recursos, por lo cual hace necesario la utilización de *JSON*.

Características Generales:

- Independiente de algún lenguaje específico.
- Basado en texto.
- De formato ligero.

- Fácil de “parsear”.<sup>23</sup>
- No define Funciones.
- No tiene estructuras invisibles.
- No tiene espacios de nombres.



**Figura 3.5:** Esquema General de la Aplicación.

Formas de representación JSON:

- *Object*: Es decir conjunto desordenado de pares nombre/valor.
- *Array*: Colección ordenada de valores.
- *Value*: Puede ser *string*, número, *booleano*, objeto o arreglo.
- *Number*: Valor numérico sin comillas.

La codificación es estrictamente UNICODE<sup>24</sup> y por defecto UTF-8(Formato de codificación de caracteres Unicode de 8-bits), pero también están permitidos UTF-16 y UTF-32.

<sup>23</sup>El parseo transforma una entrada de texto en una estructura de datos de manera que sea apropiada para ser procesada, mediante la identificación de símbolos para este caso comillas ""

<sup>24</sup>UNICODE es un estándar de codificación de caracteres diseñado para facilitar el tratamiento informático, transmisión y visualización de textos de múltiples lenguajes y disciplinas técnicas.

### 3.4.1 NOTACIÓN JSON FORMA *OBJECT/CLASE*

En esta notación se expresa un conjunto de propiedades cada una con su valor, expresada de la siguiente manera.

- Empieza con una llave de apertura {
- Termina con una llave de cierre }
- Sus propiedades se separan por comas ,
- Finalmente el Nombre y el valor separados por dos puntos “.”

*Ejemplo:*

```
{“Nombre1”:”Valor”,”Nombre2”:”Valor”}.
```

## 3.5 MANEJO DE DATOS EN ANDROID

### 3.5.1 SERVIDOR XAMPP

El servidor XAMPP es independiente de la plataforma, pero principalmente se trata de un software libre, que incluye una base de datos en MySQL, un servidor *web* Apache y los lenguajes intérpretes son PHP (Lenguaje de Hipertexto Pre procesado) y PERL (Lenguaje Práctico para la Extracción e Informe).

Por las ventajas anteriormente mencionadas se ha escogido a XAMPP como el servidor de base de datos y servicios *web*.

Tras la instalación del mismo, se genera en el directorio una serie de archivos con los que el servidor realiza sus operaciones.

La figura 3.6 muestra la ubicación `“C:\xampp\htdocs\xampp\phpfiles”` se encuentran los ficheros de extensión “.php” con los que se obtendrá la información necesaria de la base de datos del Centro Comercial. Estos archivos son cargados para el correcto funcionamiento de la aplicación.



Nombre	Fecha de modifica...	Tipo	Tamaño
android_connect	09/12/2012 17:23	Carpeta de archivos	
empleados	09/12/2012 17:23	Carpeta de archivos	
db_config	25/10/2012 15:47	Archivo PHP	1 KB
db_connect	24/10/2012 16:05	Archivo PHP	1 KB
delete_local	25/10/2012 10:54	Archivo PHP	2 KB
get_all_local	06/11/2012 14:21	Archivo PHP	2 KB
get_clientes	30/04/2013 12:02	Archivo PHP	2 KB
get_impresoras	28/05/2013 12:17	Archivo PHP	3 KB
get_laptops	28/05/2013 20:24	Archivo PHP	3 KB
get_local	19/02/2013 11:51	Archivo PHP	2 KB
get_local_piso	16/04/2013 14:43	Archivo PHP	2 KB
get_local_tipo	21/01/2013 14:46	Archivo PHP	2 KB
get_locales	16/02/2013 14:14	Archivo PHP	2 KB
get_oferta_detalle	15/04/2013 11:17	Archivo PHP	3 KB
get_oferta_tipo	16/04/2013 0:10	Archivo PHP	2 KB
get_ofertas	08/04/2013 12:05	Archivo PHP	2 KB

**Figura 3.6:** Directorio de ubicación de los ficheros xml

En la Figura 3.7 se presenta la estructura de la base de datos diseñada para la aplicación.

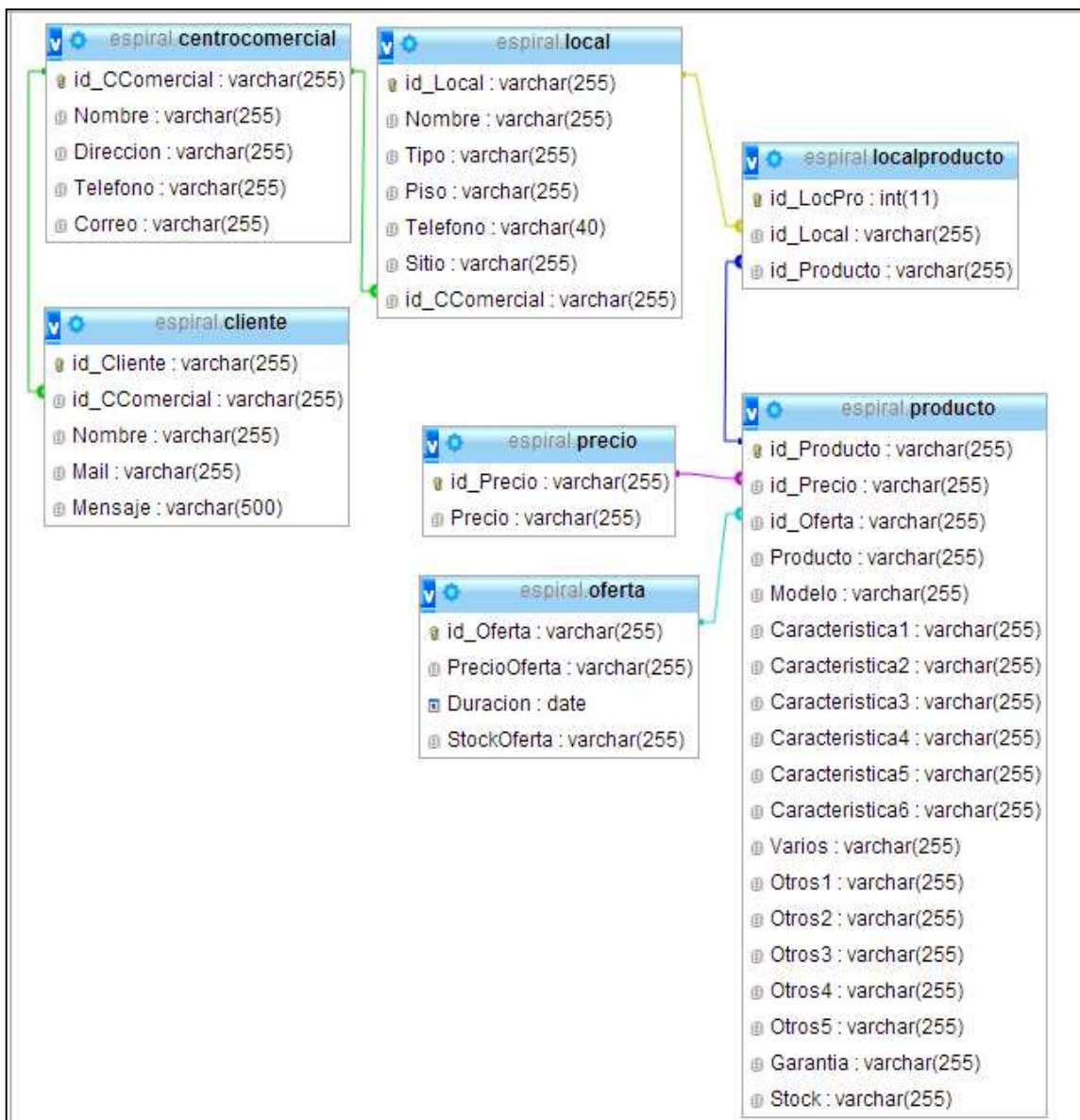
### 3.5.3 MANEJO DE DATOS - SERVICIOS WEB

La aplicación informativa del Centro Comercial Espiral, requiere del eficiente y correcto manejo del flujo de datos entre la base de datos y cada una de las actividades que forman parte de la misma.

El uso de servicios *web* garantiza la eficacia en cuanto a la respuesta de una petición.

#### 3.5.3.1 Archivos PHP

Al ser *PHP* (Lenguaje de Hipertexto Pre procesado) uno de los lenguajes de uso general, intérprete de servicios *web* también al lado del servidor, es invisible para el navegador ya que es el servidor el que se encarga de ejecutar el código y enviar su resultado HTML (Lenguaje de marcado hipertextual para el desarrollo de páginas *web*) al navegador. Una de sus características principales es su conectividad con el gestor de base de datos MySQL.



**Figura 3.7:** Diagrama Base de Datos implementada en *MySQL*

A continuación se describe cada uno de los ficheros de extensión XML implementados para la obtención de datos del gestor de base de datos MySQL descrito anteriormente.

### 3.5.3.2 Fichero “*DB\_CONFIG.PHP*”

EL fichero php “*db\_config.php*” es el primero implementado en los servicios para la conexión con la base de datos, se encuentra ubicado en la ruta: “*C:\xampp\htdocs\xampp\phpfiles*”; ubicación descrita anteriormente.

El fichero `db_config` contiene el siguiente código:

```

1  <?php
   * Todas las variables de conexión de la base de datos
2  define('DB_USER', "root");
3  define('DB_PASSWORD', "");
4  define('DB_DATABASE', "espiral");
5  define('DB_SERVER', "localhost");
6  ?>

```

### **Código 3.1:** Variables de conexión a la base de datos

Como se aprecia, en el fichero “`db_config.php`” se definen las variables necesarias para la conexión con la base de datos, es decir el usuario, contraseña, nombre y se especifica cuál es el servidor de la base de datos, como se mencionó, los ficheros deben implementarse en el lado del servidor y no del cliente.

#### **3.5.3.3 Fichero “`DB_CONNECT.PHP`”**

El fichero php “`db_connec.php`” establece la conexión con la base de datos, ubicado en la ruta: “`C:\xampp\htdocs\xampp\phpfiles`”

Se establecen dos funciones, una para establecer la conexión y otra para la desconexión con la base de datos:

```

// Constructor
1  function __construct() {
   // Conecta la base de datos
2  $this->connect();
3  }
// Destructor
4  function __destruct() {
   // Cierra la conexión
5  $this->close();

```

### **Código 3.2:** Funciones de conexión y desconexión

La función **`Connect()`** es la que realiza la conexión:

```

1    function connect() {
2    require_once __DIR__ . '/db_config.php';
3    $con = mysql_connect(DB_SERVER, DB_USER, DB_PASSWORD) or
      die(mysql_error());
4    $db = mysql_select_db(DB_DATABASE) or die(mysql_error()) or
      die(mysql_error());
5    return $con;
6    }

```

**Código 3.3:** Función *Connect*

**Fila 2:** Importa las variables de la base de datos. ***Require\_once*** permite la ubicación de un directorio en específico para este caso busca el fichero “db\_config.php” para importar sus variables.

**Fila 3:** Describe la conexión con la base de datos. La función ***mysql\_connect()*** de MySQL en PHP abre y realiza una conexión con el servidor de base de datos.

**Fila 4:** Selecciona la base de datos. ***Mysql\_select\_db()*** establece la base de datos activa actual en el servidor asociado con el identificador de enlace especificado, para este caso con la base de dato Espiral. ***Mysql\_error()*** devuelve el texto del mensaje de error de la operación MySQL anterior.

**Fila 5:** Retorna el cursor de la conexión.

La función ***Close()*** especifica el cierre de la conexión:

```

1    function close() {
2    mysql_close();
3    }

```

**Código 3.4:** Función *Close()*

**Fila 2: `mysql_close()`**, cierra la conexión no persistente al servidor de MySQL que está asociada con el identificador de enlace especificado, para este caso cierra la base de datos Espiral.

### 3.5.3.4 Fichero “`GET_LOCAL_TIPO.PHP`”

Este fichero muestra la lista de todos los locales por tipo o categoría contenidos en la base de datos del Centro Comercial, se encuentra ubicado en la ruta: “`C:\xampp\htdocs\xampp\phpfiles`”.

El código 3.5 muestra la consulta a la base de datos empleada.

```

1     $response = array();
2     require_once __DIR__ . '/db_connect.php';
3     $db = new DB_CONNECT();
4     if (isset($_GET["Tipo"])){
5         $Tipo = $_GET['Tipo'];
6         $result = mysql_query("SELECT *FROM local WHERE Tipo=$Tipo") or
7             die(mysql_error());
8         if (mysql_num_rows($result) > 0) {
9             $response["local"] = array();
10            while ($row = mysql_fetch_array($result)) {
11                $locales = array();
12                $locales["id_Local"] = $row["id_Local"];
13                $locales["Nombre"] = $row["Nombre"];
14                $locales["Tipo"] = $row["Tipo"];
15                $locales["Piso"] = $row["Piso"];
16                array_push($response["local"], $locales);
17            }
18        }
19    }

```

**Código 3.5:** Consulta a la base de datos

**Fila 1:** Arreglo o array donde se almacenan los locales.

**Fila 2:** Incluye el directorio de la clase conexión.

**Fila 3:** Genera un nuevo conector con la base de datos.

**Fila 4:** Chequeo de cada dato ingresado por teclado ingresando el Tipo.

**Fila 5 :** Declaración de la consulta empleada.

**Fila 6 – Fila 14:** Genera el arreglo de todos los locales existentes en la base de Datos.

- ***\$result:*** Variable donde se almacena el resultado de la consulta.
- ***isset:*** Es una función de paso de variables que realiza una evaluación de izquierda a derecha para comprobar si una variable está definida y no es *NULL* o nula, para este caso comprobará la variable “*Tipo*” de la tabla local, con su correspondiente valor ingresado por teclado.
- ***Mysql\_query():*** Envía una única consulta (múltiples consultas no están soportadas) a la base de datos actualmente activa en el servidor asociado con el identificador de enlace especificado, para este caso con la tabla “*Local*” de la Base de datos Espiral.
- ***GET:*** Se encuentra dentro de las variables predefinidas de PHP para pasar un arreglo de variables asociativo en este caso a parámetros URL entre la base de datos.
- ***\$response:*** Se encuentra dentro de las variables predefinidas de PHP para pasar el arreglo donde está el contenido del GET de todos los locales.
- ***mysql\_num\_rows:*** Obtiene el número de filas de un conjunto de resultados, en este caso el numero de filas de la consulta antes generada.
- ***mysql\_fetch\_array:*** Recupera el arreglo correspondiente a fila recuperada, es decir recupera cada uno de los campos de la consulta antes generada, por ejemplo: “*id\_Local*”= “22” etc.
- ***array\_push:*** Inserta el valor de un arreglo al final de otro, en este caso inserta en el arreglo local todo el arreglo generado de la consulta, es decir todos los locales por tipo encontrados.

Si el arreglo de respuesta fue exitoso, genera la representación en JSON (*JavaScript Object Notation* explicado anteriormente) del valor correspondiente caso contrario muestra el mensaje locales no encontrados.

```
1     $response["success"] = 1;
2     echo json_encode($response);
3     } else {
4     $response["success"] = 0;
5     $response["message"] = "Locales no encontrados";
6     echo json_encode($response);
     }
```

**Código 3.7:** Respuesta de GET\_LOCAL\_TIPO.PH

**Fila 1-6:** Muestra el mensaje JSON en caso de éxito o FALSO en caso de error

### 3.5.3.5 Fichero “GET\_LOCAL.PHP”

El fichero php “*get\_local.php*” muestra la lista de todos los locales por *id\_Local* contenidos dentro de la tabla local en base de datos del Centro Comercial, este fichero se encuentra ubicado en la ruta “*C:\xampp\htdocs\xampp\phpfiles*”.

Debido a que este fichero es muy similar al anterior y se repiten exactamente las mismas funciones únicamente se explicará la consulta empleada.

```
1     $result = mysql_query("SELECT *FROM local WHERE id_Local = $id_Local");
```

**Código 3.8:** Consulta empleada en el fichero GET\_LOCAL.PHP

**Fila 1:** Obtiene los locales ingresando por teclado el *id\_Local* correspondiente.

### 3.5.3.6 Fichero “GET\_PRODUCTO\_TIPO.PHP”

Muestra la lista de todos los productos por tipo que se obtiene como resultado a la consulta de base de datos del Centro Comercial.

La consulta empleada es la siguiente:

```

1      $result= mysql_query
2      ("SELECT A.id_Local,A.Nombre,B.id_LocPro,C.id_Producto,C.Producto,C.Modelo
3      from local A,localproducto B,producto C WHERE A.id_Local=B.id_Local AND
4      B.id_Producto=C.id_Producto AND A.Tipo=$Tipo") or die(mysql_error());

```

#### **Código 3.9:** Consulta en el fichero GET\_PRODUCTOS\_TIPO.PHP

**Fila 1-4:** Obtiene todos los productos por tipo ingresando el Tipo de la tabla *Local* Puesto que la relación en la base de datos entre las tablas producto y local es de varios a varios se requiere de la tabla intermedia denominada *local producto* para su interacción y por esto el uso de la consulta más elaborada.

### 3.5.3.7 Fichero “GET\_OFERTA\_TIPO.PHP”

El fichero php “*get\_oferta\_tipo.php*” muestra la lista de todas las ofertas por tipo como resultado a la consulta de base de datos del Centro Comercial.

La siguiente consulta muestra una relación entre cuatro tablas de la base de datos: *Producto*, *Modelo*, *PrecioOferta* y *Duración*.

```

1      $result = mysql_query
2      ("SELECT A.id_Local, A.Nombre, A.Piso, B.id_LocPro, C.id_Producto, C.Producto,
3      C.Modelo, D.PrecioOferta, D.Duracion from local A,localproducto B,producto
4      C,oferta D WHERE A.id_Local=B.id_Local AND B.id_Producto=C.id_Producto
5      AND C.id_Oferta=D.id_Oferta AND A.Tipo=$Tipo") or die(mysql_error());

```

#### **Código 3.10:** Consulta en el fichero GET\_OFERTA\_TIPO.PHP

**Fila 1-5:** Obtiene las ofertas por tipo ingresando el “Tipo” de la tabla local



### 3.5.3.8 Fichero “GET\_PRODUCTO\_DETALLES.PHP”

El fichero php “*get\_producto\_detalles.php*” muestra el detalle de un producto por *id\_LocPro*, que es la clave primaria de la tabla *localproducto* como resultado a la consulta de base de datos del Centro Comercial.

El código 3.11 muestra la consulta, mediante la relación entre cuatro tablas de la base de datos, donde el parámetro ingresado por teclado es *id\_LocPro*.

```
$result=mysql_query("SELECT A.id_Local,A.Nombre,A.Piso,A.Telefono,A.Sitio,C.id_Producto,C.Producto,C.Modelo,C.Disco,C.Procesador,C.Memoria,C.Camara,C.SistemaOperativo,C.Garantia,C.Stock,C.Pantalla,C.Varios,C.Otros1,C.Otros2,C.Otros3,C.Otros4,C.Otros5,D.Precio)
from local A,localproducto B,producto C,precio D
WHERE A.id_Local=B.id_Local
AND B.id_Producto=C.id_Producto
AND C.id_Precio=D.id_Precio
AND B.id_LocPro=$id_LocPro") or die(mysql_error());
```

**Código 3.11:** Consulta en el fichero “GET\_PRODUCTO\_DETALLES.PHP”

### 3.5.3.9 Fichero “GET\_OFERTA\_DETALLES.PHP”

El fichero php *get\_oferta\_detalles.php* muestra el detalle de una oferta por *id\_LocPro* donde éste es la clave primaria de la tabla *localproducto* como resultado a la consulta de base de datos del Centro Comercial.

El código 3.12 muestra la consulta entre cinco tablas de la base de datos y por eso el uso de una consulta más elaborada, donde el parámetro ingresado por teclado es *id\_LocPro*, clave primaria de la tabla *localproducto*.

### 3.5.3.10 Fichero “GET\_LAPTOPS.PHP”

El fichero php “*get\_laptops.php*” muestra las *laptops* que dispone cada local de la categoría tecnología realizando una consulta por *id\_Local* donde este es la clave primaria de la tabla local en base de datos del Centro Comercial.

```

$result = mysql_query
("SELECT A.id_Local,A.Nombre,A.Piso,A.Telefono,A.Sitio,C.id_Producto,C.Producto
,C.Modelo,C.Disco,C.Procesador,C.Memoria,C.Camara,C.SistemaOperativo,C.Gara
ntia,C.Stock,C.Pantalla,C.Varios,C.Otros1,C.Otros2,C.Otros3,C.Otros4,C.Otros5,D.
Precio,E.PrecioOferta,E.Duracion,E.StockOferta
from local A,localproducto B,producto C,precio D, oferta E
WHERE A.id_Local=B.id_Local
AND B.id_Producto=C.id_Producto
AND C.id_Precio=D.id_Precio
AND C.id_Oferta=E.id_Oferta
AND B.id_LocPro=$id_LocPro") or die(mysql_error());

```

**Código 3.12:** Consulta en el fichero “GET\_PRODUCTO\_DETALLES.PHP”

La consulta del código 3.13 muestra una relación entre cuatro tablas de la base de datos: *local*, *localproducto*, *producto* y *precio*, donde el parámetro ingresado por teclado es *id\_Local*, es decir obtiene la información correspondiente del local al que pertenece la *laptop*.

```

$result = mysql_query
("SELECT A.id_Local,A.Nombre,A.Piso,A.Telefono,A.Sitio,B.id_LocPro,C.id_Producto,C.Producto,C.Modelo,C.Disco,C.Procesador,C.Memoria,C.Camara,C.SistemaOperativo,C.Garantia,C.Stock,C.Pantalla,C.Varios,C.Otros1,C.Otros2,C.Otros3,C.Otros4,C.Otros5,D.Precio
from local A,localproducto B,producto C,precio D
WHERE A.id_Local=B.id_Local
AND B.id_Producto=C.id_Producto
AND C.id_Precio=D.id_Precio
AND C.Producto='LAPTOP'
AND B.id_Local=$id_Local") or die(mysql_error());

```

**Código 3.13:** Consulta de fichero GET\_LAPTOPS.PHP

### 3.5.3.11 Fichero “GET\_TABLETS.PHP”

Muestra las *tablets* que dispone cada local de la categoría tecnología realizando una consulta por *id\_Local*, clave primaria de la tabla local en base de datos del Centro Comercial.

La consulta muestra una relación entre cuatro tablas de la base de datos y por eso el uso de una consulta más elaborada, donde el parámetro ingresado por teclado es *id\_Local*, clave primaria de la tabla local.

```
$result = mysql_query
("SELECT A.id_Local,A.Nombre,A.Piso,A.Telefono,A.Sitio,B.id_LocPro,C.id_Producto,C.Producto,C.Modelo,C.Disco,C.Procesador,C.Memoria,C.Camara,C.SistemaOperativo,C.Garantia,C.Stock,C.Pantalla,C.Varios,C.Otros1,C.Otros2,C.Otros3,C.Otros4,C.Otros5,D.Precio
from local A,localproducto B,producto C,precio D
WHERE A.id_Local=B.id_Local
AND B.id_Producto=C.id_Producto
AND C.id_Precio=D.id_Precio
AND C.Producto='TABLET'
AND B.id_Local=$id_Local") or die(mysql_error());
```

**Código 3.14:** Consulta de fichero GET\_TABLETS.PHP

### 3.5.3.12 Fichero “GET\_IMPRESORAS.PHP”

El fichero php “get\_impresoras.php” muestra las impresoras que dispone cada local de la categoría tecnología realizando una consulta por *id\_Local*, clave primaria de la tabla local en base de datos del Centro Comercial.

La consulta del código 3.15 muestra una relación entre las tablas *local*, *localproducto*, *producto* y *precio*, donde el parámetro ingresado por teclado es *id\_Local*, clave primaria de la tabla *local*.

### 3.5.3.13 Fichero “GET\_OTROSTECNOLOGIA.PHP”

Muestra otros artículos tecnológicos en general que no sean laptops, impresoras y *tablets* que dispone cada local de la categoría tecnología realizando una consulta

por *id\_Local*, que es la clave primaria de la tabla *local* en la base de datos del Centro Comercial.

```

("SELECT A.id_Local,A.Nombre,A.Piso,A.Telefono,A.Sitio,B.id_LocPro,C.id_Producto,
C.Producto,C.Modelo,C.Disco,C.Procesador,C.Memoria,C.Camara,C.SistemaOperativo,
C.Garantia,C.Stock,C.Pantalla,C.Varios,C.Otros1,C.Otros2,C.Otros3,C.Otros4,
C.Otros5,D.Precio
from local A,localproducto B,producto C,precio D
WHERE A.id_Local=B.id_Local
AND B.id_Producto=C.id_Producto
AND C.id_Precio=D.id_Precio
AND C.Producto='TABLET'
AND B.id_Local=$id_Local") or die(mysql_error());

```

**Código 3.15:** Consulta en el fichero GET\_IMPRESORAS.PHP

La consulta del código 3.16 muestra una relación entre cuatro tablas de la base de datos, donde el parámetro ingresado por teclado es *id\_Local*, clave primaria de la tabla *local*. Pero especificando que no sean *tablets*, impresoras y laptops es decir muestra otro tipo de artículos tecnológicos en general para facilitar la actividad que más adelante se describe y sobre todo para facilitar al usuario la búsqueda en los locales de la categoría tecnología.

**3.5.3.14 Fichero “GET\_CLIENTES.PHP”**

El fichero php *“get\_clientes.php”* muestra la lista de todos los visitantes en general quienes hayan ingresado sus comentarios al buzón de sugerencias realiza la búsqueda por *id\_CComercial* contenidos dentro de la tabla cliente en base de datos del Centro Comercial.

```

$result = mysql_query
("SELECT A.id_Local,A.Nombre,A.Piso,A.Telefono,A.Sitio,B.id_LocPro,C.id_Producto,C.Producto,C.Modelo,C.Disco,C.Procesador,C.Memoria,C.Camara,C.SistemaOperativo,C.Garantia,C.Stock,C.Pantalla,C.Varios,C.Otros1,C.Otros2,C.Otros3,C.Otros4,C.Otros5,D.Precio
from local A,localproducto B,producto C,precio D
WHERE A.id_Local=B.id_Local
AND B.id_Producto=C.id_Producto
AND C.id_Precio=D.id_Precio
AND C.Producto<>'LAPTOP'
//Especifica que no sea laptop
AND C.Producto<>'TABLET'
//Especifica que no sea tablet
AND C.Producto<>'IMPRESORA'
//Especifica que no sea impresora
AND B.id_Local=$id_Local") or die(mysql_error());

```

**Código 3.16:** Consulta en el fichero GET\_OTROS\_TECNOLOGIA.PHP

La consulta a la base de datos utilizada se muestra en el código 3.17.

**Fila 1-2:** consulta que muestra todos los visitantes de la tabla clientes.

**Fila 3-16:** genera el arreglo correspondiente a cada atributo de las base de Datos.

### 3.5.3.15 Fichero “UPDATE\_CLIENTES.PHP”

El fichero php “update\_clientes.php” ingresa el comentario, nombre y mail del visitante en general quien haya ingresado al buzón de sugerencias el del Centro Comercial.

El código 3.18 muestra como se realiza el ingreso de la sugerencia.

**Fila 1:** comprueba si cada uno de los campos existen previo al ingreso de Datos.

**Fila 2-6:** método POST que realiza un ingreso de datos por teclado hacia la Base.

```

// Obtiene todos los clientes o visitantes de la tabla cliente por id_CComercial
1. $result = mysql_query
2. ("SELECT * from cliente WHERE id_CComercial=$id_CComercial") or
   die(mysql_error());
3. if (mysql_num_rows($result) > 0) {
4. $response["cliente"] = array();
5. while ($row = mysql_fetch_array($result)) {
6. $clientes = array();
7. $clientes["id_Cliente"] = $row["id_Cliente"];
8. $clientes["id_CComercial"] = $row["id_CComercial"];
9. $clientes["Nombre"] = $row["Nombre"];
10. $clientes["Mail"] = $row["Mail"];
11. $clientes["Mensaje"] = $row["Mensaje"];
12. array_push($response["cliente"], $clientes);
13. }

```

### Código 3.17: Consulta en el fichero GET\_CLIENTES.PHP

```

1   if(isset($_POST['id_Cliente'])    &&    isset($_POST['id_CComercial'])    &&
    isset($_POST['Nombre'])&&isset($_POST['Mail'])
    &&isset($_POST['Mensaje'])) {
2   $id_Cliente = $_POST['id_Cliente'];
3   $id_CComercial = $_POST['id_CComercial'];
4   $Nombre = $_POST['Nombre'];
5   $Mail = $_POST['Mail'];
6   $Mensaje = $_POST['Mensaje'];
   // Incluye la clase conexión
7   require_once __DIR__ . '/db_connect.php';
   // Conectando con la base
8   $db = new DB_CONNECT();

```

### Código 3.18: Fichero UPDATE\_CLIENTES.PHP

- **isset:** Es una función de paso de variables que realiza una evaluación de izquierda a derecha para comprobar si una variable está definida y no es *NULL* o nula, para este caso comprobará en los campos *id\_Cliente*, *id\_CComercial*, *nombre*, *mail* y *mensaje* de la tabla *cliente*.

- **POST:** Se encuentra dentro de las variables predefinidas de PHP para ingresar un arreglo de variables asociativo en nuestro caso entre la URL y la base de datos.

Para ingresar los datos antes mencionados la consulta empleada es la siguiente:

```
$result = mysql_query("INSERT INTO cliente (id_Cliente, id_CComercial,
Nombre,Mail,Mensaje)VALUES('$id_Cliente','$id_CComercial','$Nombre','$Mail','$Mensa
je')");
```

### Código 3.19: Consulta en el fichero UPDATE\_CLIENTES.PHP

- **\$result:** Variable donde se almacena el resultado de la consulta.
- **INSERT INTO:** Ingresa los datos correspondientes a la tabla Cliente.
- **Mysql\_query():**Envía una única consulta a la base de datos actualmente activa en el servidor asociado con el identificador de enlace especificado, para este caso con la tabla *Local* de la Base de datos *Espiral*.

```
Comprobamos si la fila fue insertada o no
1   if ($result) {
2   $response["success"] = 1;
3   $response["message"] = "Cliente ingresado";
4   echo json_encode($response);
5   } else {
6   }
7   } else {
8   $response["success"] = 0;
9   $response["message"] = "El campo requerido no existe";
10  echo json_encode($response);
11  }
```

### Código 3.20: Muestra si el cliente fue ingresado.

**Fila 1:** Comprueba si la fila fue insertada o no.

**Fila 2-4:** Genera el mensaje de Cliente ingresado.

**Fila 7-10:** En el caso contrario genera mensaje “El campo requerido no existe”.

- ***\$response***: Se encuentra dentro de las variables predefinidas de PHP para pasar el arreglo donde está el contenido del *POST* para este caso de todos los clientes.
- ***json\_encode***: Devuelve un *string* JSON codificado en caso de éxito o *FALSE* en caso de error.

## 3.6 ACTIVIDADES IMPLEMENTADAS EN LA APLICACIÓN

### 3.6.1 ACTIVIDADES ANDROID [31]

En *Android* una actividad es una clase extensión “.java” que se ubica en la carpeta contenedora del proyecto dentro SRC, *source* o código fuente explicado anteriormente, donde básicamente programamos nuestra aplicación.

Cada actividad controla los diferentes elementos generados previamente en los archivos XML y de esta manera se muestra en pantalla.

La figura 3.8 muestra métodos que se pueden sobrescribir si el caso así lo amerita, el método *OnCreate()* por ejemplo define todo lo que hace antes que la pantalla se muestre.

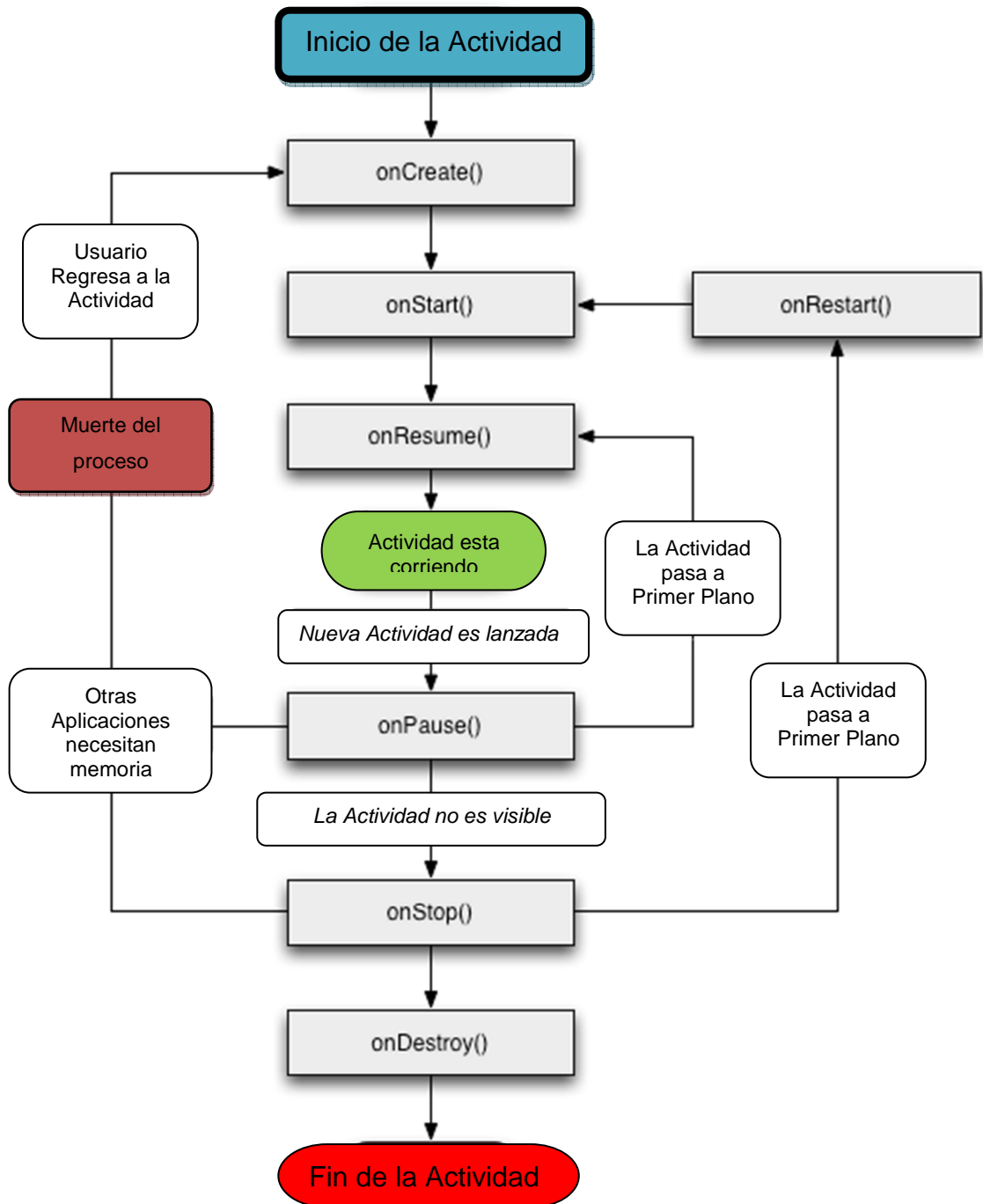
A continuación se presenta la explicación correspondiente a cada una de las clases empleadas en las diferentes actividades del proyecto.



### 3.6.2 CLASE PRINCIPAL

La clase “*MainActivity.java*” es la actividad principal de la aplicación, la primera en abrirse al momento de la ejecución en un dispositivo móvil Android (*Tablet*).

Esta Actividad direcciona a las diferentes Actividades del programa.



**Figura 3.8:** Ciclo de vida de una Actividad [31]

En específico, permite:

- Abrir la Actividad “Locales”, la misma que permite la búsqueda de locales por tipo o categorías.
- Abrir la Actividad “Productos”, que permite la búsqueda de productos por tipo.
- Abrir la Actividad “Ofertas”, con la que se realiza la búsqueda de ofertas por tipo.
- Abrir la Actividad “Galería de Locales”, que permite ubicar todos los locales por piso en el plano de acuerdo con su piso.
- Abrir la Actividad “Ubicación”, que muestra el mapa de google del sitio.
- Abrir la Actividad “Buzón de Sugerencias”, donde el usuario puede ingresar sus sugerencias respecto al Centro Comercial.

Las librerías que utiliza la Clase Main Activity.java son las siguientes:

- **import** android.app.Activity;
- **import** android.content.Intent;
- **import** android.os.Bundle;
- **import** android.view.View;
- **import** android.widget.ImageView;

La librería “*android.app.Activity*” permite extender a la Clase como Actividad, y de esta forma manejar los diferentes elementos contenidos del *layout* o diseño de la Aplicación es decir botones (*image Button*), cajas de texto (*text View*).

Por defecto está implementada la función *OnCreate()* la misma que maneja lo que se visualiza al momento de cargar la Actividad, esta función por defecto crea un

paquete de datos *Bundle*, donde se almacena los datos a mostrarse, a través de la librería “*android.os.Bundle*” previamente instanciada, como muestra el código 3.21.

```

1      @Override
2      public void onCreate(Bundle savedInstanceState) {
           // Función onCreate generada por defecto, donde se almacenan
           // las instancias de la Actividad Main.
3      super.onCreate(savedInstanceState);
           // Llama al layout que contiene el diseño de la Actividad llamado activity_main
4      setContentView(R.layout.activity_main);

```

**Código 3.21:** Función *OnCreate*

Para poder interactuar entre la carpeta SRC que contiene el código fuente, y la carpeta RES que contiene los *layouts*, es decir los archivos xml de la aplicación se utiliza la clase *R.java*, por tal motivo si se quiere obtener el *id* (identificador) de cualquier elemento se hace referencia al mismo utilizando la función *findViewById* como se observa en el código 3.22 contenido en la Clase *MainActivity.java*.

```

1      btnLocales= (ImageView) findViewById(R.id.btnLocales);
2      btnOfertas = (ImageView) findViewById(R.id.btnOfertas);
3      btnProductos=(ImageView) findViewById(R.id.btnProductos);
4      btnSitio=(ImageView) findViewById(R.id.btnMapa);
5      btnGaleria=(ImageView) findViewById(R.id.btnGaleria);
6      btnBuzon=(ImageView) findViewById(R.id.btnBuzon);

```

**Código 3.22:** Uso de la función *findViewById*

La clase “*Intent*” definida bajo la librería “*android.content.intent*”, es la que permitirá definir alguna acción a realizar sobre la actividad, controlando el flujo de datos.

En estas líneas de código se muestra como al llamar al evento *click* del *ImageView* (Vista de imagen), se controla cual será la actividad que se abrirá al dar *click* a la imagen.

```
// Crea el escucha del evento click de la imagen btnLocales
1. btnLocales.setOnClickListener(new View.OnClickListener() {
2.     @Override
3.     public void onClick(View view) {
4.         Intent i = new Intent(getApplicationContext(),LocalesActivity.class);
5.         startActivity(i);
6.     }
7. });
```

### Código 3.23: Llamada al evento *click*

**Fila 1:** Crea la escucha del evento click

**Fila 4:** Crea el intent que lanza el evento clic, para abrir la actividad locales

**Fila 5:** Comienza la actividad

- ***setOnClickListener()***: Implementa el método *click* de cualquier elemento del *layout* o diseñador de la aplicación, para este caso estoy implementando este en el *ImageView* llamado *btnLocales*.
- ***OnClickListener()***: Sobrescribe el método *onClick*(dar *click*), para ejecutar una acción específica, para este caso lanza al dar *click* lanza la Acitividad *LocalesActivity*.
- ***getApplicationContext()***: Permite obtener el contexto general de la Actividad.
- ***startActivity()***: Lanza la Actividad definida en la Clase *Intent*, por ejemplo para el presente caso cambia de la Actividad *Main* a la Actividad *Locales*.

### 3.6.3 INTERFAZ “ACTIVITY\_MAIN.XML”

El archivo *activity\_main.xml* contiene el diseño de la interfaz gráfica de la cual hace uso la Actividad *Main* antes mencionada.

Contiene los *ImageView* (vista de imágenes) que contiene las imágenes de los botones de esta actividad, adicional los logos e información del Centro Comercial. Todos estos contenidos están dentro de un *RelativeLayout*, como lo muestra el código 3.24.

```

1      <?xmlversion="1.0"encoding="utf-8"?>
2      <RelativeLayoutxmlns:android="http://schemas.android.com/apk/res/android"
3          android:layout_width="fill_parent"
4          android:layout_height="fill_parent"
5          android:background="@drawable/fondohorizontal"
6          android:orientation="horizontal"
7          android:gravity="center_horizontal">

```

**Código 3.24:** *RelativeLayout*

- ***RelativeLayout*:** Es una vista de grupo contenedora de cada uno de los elementos que forman la interfaz, empleándola se puede colocar cada elemento, basándolos en su posición relativa al padre y de esta manera manipular las distancias entre objetos como se indica a continuación en el código 3.25:

```

1      <ImageView
2          android:id="@+id/logo"//Establece el id del imageview
3          android:layout_width="wrap_content"
4          android:layout_height="wrap_content"
5          android:paddingLeft="60dp"
6          android:paddingTop="0dp"
7          android:src="@drawable/androides"//Carga la imagen androides
8          android:visibility="visible"/>

```

**Código 3.25:** Atributos de un *ImageView*

En el *ImageView* se especifica la posición hacia la derecha y con respecto al tope superior de la interfaz, al ser la carpeta *drawable* la contenedora de las imágenes por medio de *android:src="@drawable/androides"* donde se indica al visor de imagen que cargue la imagen *androides* (Imagen decorativa).

- **android:id:** Establece el id correspondiente a un elemento para la utilización del mismo en la clase java que maneja la Actividad. Para este caso el id del *ImageView* es *logo*.

En la figura 3.9 muestra lo descrito anteriormente en código, plasmado en forma gráfica.



**Figura 3.9:** Modo gráfico de la interfaz *activity\_main.xml*

### 3.6.4 CLASE “JSONPARSER”

Para la lectura de datos en formato JSON (Notación de Objetos en JavaScript) que generan los servicios *web* antes mencionados se utiliza la clase “*JSONParser.java*”.

El código 3.26 muestra las librerías que la clase utiliza.

```

import java.io.BufferedReader;
//Manejo del flujo de entrada y salida de datos
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.UnsupportedEncodingException;
import java.util.List;
import org.apache.http.HttpEntity;
//Manejo de datos con el servidor Apache
import org.apache.http.HttpResponse;
import org.apache.http.NameValuePair;
import org.apache.http.client.ClientProtocolException;
import org.apache.http.client.entity.UrlEncodedFormEntity;
import org.apache.http.client.methods.HttpGet;
import org.apache.http.client.methods.HttpPost;
import org.apache.http.client.utils.URLEncodedUtils;
import org.apache.http.impl.client.DefaultHttpClient;
import org.json.JSONException;
//Manejo de excepciones de la notación JSON
import org.json.JSONObject;

```

**Código 3.26:** Librerías en *JSONPARSER*

Las librerías importadas en la clase “*JSONParser.java*” manejan el flujo de entrada y salida de datos, así como también cada uno de los servicios *web* para la lectura de datos por parte del cliente a través del servidor apache, y finalmente el manejo de la notación JSON de los datos obtenidos en la URL. El código 3.27 muestra las variables declaradas:

```

Public class JSONParser {
    static InputStream is = null;
    static JSONObject jObj = null;
    static String json = "";

    // constructor
    public JSONParser() {
    }
}

```

**Código 3.27:** Variables en la clase “*JSONPARSER*”

- **InputStream:** Arreglo el cual lee los datos de los archivos del sistema a través de la red.
- **JSONObject:** Crea el objeto en formato JSON a ser procesado.

A continuación el código 3.28 el cual muestra la función que obtiene los datos en formato JSON de la URL.

```

1. public JSONObject makeHttpRequest(String url, String method,
2. List<NameValuePair> params) {
3. try {
4.     if(method == "POST"){
5.         DefaultHttpClient httpClient = newDefaultHttpClient();
6.         HttpPost httpPost = newHttpPost(url);
7.         httpPost.setEntity(new UriEncodedFormEntity(params));
8.         HttpResponse httpResponse = httpClient.execute(httpPost);
9.         HttpEntity httpEntity = httpResponse.getEntity();
10.        is = httpEntity.getContent();
11.    }elseif(method == "GET"){
12.        DefaultHttpClient httpClient = newDefaultHttpClient();
13.        String paramString = URLEncodedUtils.format(params, "utf-8");
14.        url += "?" + paramString;
15.        HttpGet httpGet = newHttpGet(url);
16.        HttpResponse httpResponse = httpClient.execute(httpGet);
17.        HttpEntity httpEntity = httpResponse.getEntity();
18.        is = httpEntity.getContent();
19.    }

```

**Código 3.28:** Obtención de datos en formato JSON

**Fila 1-2:** Declaración de la función JSONObject, cuyos parámetros son la “url”, el método y los parámetros necesarios de acuerdo a la información requerida.

**Fila 3-9:** Método POST para el envío de datos por defecto crea un *httpClient*; el envío es a través de la red con el arreglo “is”.

**Fila 10-17:** Método GET por defecto se define el *httpClient*, para la obtención de datos en la URL, los recibe a través de la red con el arreglo “is”.Especifica que los datos son utf-8 y para la obtención se formará una cadena tipo string con la URL +”Parámetros” que se requieren.



- **makeHttpRequest():** Función que devuelve un objeto JSON que pasa como parámetros de entrada el *string* de la URL completa del servicio *web*, el *string* del método pudiendo ser este un *GET* (para obtención de datos) o un *POST* (para el envío de datos) y finalmente el arreglo con cada uno de los parámetros correspondientes a los datos en la base.

El *Búfer* de datos que controla el flujo del mismo se muestra en el código 3.29

```

1. BufferedReader reader = newBufferedReader(new InputStreamReader(
2. is, "iso-8859-1"), 8);
3. StringBuilder sb = newStringBuilder();
4. String line = null;
5. while ((line = reader.readLine()) != null) {
6. sb.append(line + "\n");
7. }
8. is.close();
9. json = sb.toString();

```

**Código 3.29:** Búfer de datos

**Fila 1:** Crea el búfer para la lectura de datos.

**Fila 2:** Establece el formato UTF-8.

**Fila 3-6:** Genera el bufer correcto para cada línea de datos.

El formato de los datos a leer es UTF-8 (Formato de codificación de caracteres Unicode de 8-bits).

Las diferentes clases descritas más adelante hacen uso de esta clase para el envío de parámetros y obtención del objeto JSON que contiene los datos que cada una de las actividades mostrará en su interfaz correspondiente.

### 3.6.5 CLASE “LOCALES ACTIVITY”

La clase “*LocalesActivity.java*” esta actividad contiene un menú gráfico de las diferentes categorías a las que hemos dividido a los locales del Centro Comercial

y al seleccionar una de ellas realiza una búsqueda por tipo a la base de datos y muestra una lista con los resultados, lista que al seleccionar nos dirige a otra actividad con información más detallada de cada local.

El código 3.30 muestra los objetos y variables empleadas en esta clase:

```

publicclass LocalesActivity extends ListActivity implements OnTouchListener{
    // Declaración del dialogo de proceso
    private ProgressDialog pDialog;
    ImageButton ccespiral, laptop, celulares,juegos,joyeria,audio;
    String pid="";

    // Crea el objeto JSON
    JSONParser jParser = newJSONParser();
    //Arreglo donde se cargara la lista de locales
    ArrayList<HashMap<String, String>>localList;
    //URL donde se generan los datos JSON obtenidos desde la base
    privatestatic String url_all_products =
    "http://172.30.8.128/xampp/phpfiles/get_local_tipo.php";
    // Nombres de los Nodos JSON
    privatestaticfinal String TAG_SUCCESS = "success";
    privatestaticfinal String TAG_Locales = "local";
    privatestaticfinal String TAG_id_Local = "id_Local";
    privatestaticfinal String TAG_Nombre = "Nombre";

    // Arreglo JSON de locales
    JSONArray locales = null;

```

### Código 3.30: Clase *Locales Activity*

La declaración de variables a utilizar en la actividad es la siguiente:

- **private ProgressDialog:** Variable para generar un dialogo de proceso.
- **ImageButton:** Variable para el manejo de botones contenido en el XML.
- **JsonObject:** Variable para declaración del objeto JSON que contendrá los datos.
- **JSONArray:** Variable para la declaración de un arreglo de datos JSON.

- ***ArrayList<HashMap<String, String>>***: Declara una lista de arreglos y permite el mapeo o búsqueda de cada parámetro de la lista.

Se sobrescribe el método *onCreate*, que se cargará antes que se ejecute como lo muestra el código 3.31.

```

@Override
1. public void onCreate(Bundle savedInstanceState) {
2.     super.onCreate(savedInstanceState);
3.     setContentView(R.layout.locales);
4.     //Obtiene los ID de las imágenes
5.     laptop = (ImageButton) findViewById (R.id.imageButton1);
6.     celulares= (ImageButton) findViewById (R.id.imageButton2);
7.     juegos = (ImageButton) findViewById (R.id.imageButton3);
8.     joyeria= (ImageButton) findViewById (R.id.imageButton4);
9.     audio= (ImageButton) findViewById (R.id.imageButton5);
10.    //Añade el evento touch
11.    laptop.setOnClickListener(this);
12.    celulares.setOnClickListener(this);
13.    juegos.setOnClickListener(this);
14.    joyeria.setOnClickListener(this);
15.    audio.setOnClickListener(this);

```

**Código 3.31:** Método *OnCreate*

- ***findViewById():*** Método que permite la obtención de los identificadores correspondientes a cada uno de los elementos de la interfaz xml correspondiente.
- ***setOnClickListener():*** Método que añade el evento “tocar” a un elemento en específico.

El código 3.32 muestra el evento *click* adaptado a la lista de datos para de esta forma enviar el id del local a la siguiente actividad:

```

// Obtiene el listview
1. ListView lv = getListView();
//Al obtener un local lanza en la lista al dar clic se dirige a la actividad
  EditLocalActivity.
2. lv.setOnItemClickListener(new OnItemClickListener() {
3.   @Override
4.   publicvoid onItemClick(AdapterView<?> parent, View view,
5.   int position, long id) {
// Obtiene el valor pid de la lista
6.   String pid = ((TextView) view.findViewById(R.id.pid)).getText()
7.   .toString();
// Lanza la Actividad LocalDetallesActivity
8.   Intent in = newIntent(getApplicationContext(),
9.   LocalDetallesActivity.class);
// Envía el id_local a pid como información extra
10.  in.putExtra(TAG_id_Local, pid);
// En espera de respuesta de la actividad.
11.  startActivityForResult(in, 100);
12.  }

```

### Código 3.32: Adaptación de la lista

- ***getListView()***: Obtiene los elementos correspondientes de la lista .
- ***onItemClick()***: Este método adapta el arreglo de la lista, su posición y su id para que al momento de ejecutar el evento *click* obtener el *string* contenido en el *textView*(vista de texto), de cada elemento de la lista y a su vez pasar este como parámetro adicional a la siguiente actividad.
- ***startActivityForResult***: Lanza la siguiente actividad definida en la Clase *Intent*.

El código 3.33 presenta el evento “tocar” de cada imagen que maneja para realizar la búsqueda de locales por tipo.

```

@Override
public boolean onTouch(View v, MotionEvent m){
    switch(m.getAction())
    {
        case MotionEvent.ACTION_DOWN:
        {
            switch(v.getId()) {
                case R.id.imageButton1:
                    pid=""+"Tecnología"+"";
                    new LoadAllLocales().execute();
                    localList.clear();
                    break;
            }
        }
    }
}

```

**Código 3.33:** Evento *Tocar* de cada botón.

El presente código muestra el evento “*tocar*” de cada imagen contenida en la interfaz, la acción con la cual va a reaccionar será cuando el usuario haga un movimiento sobre la imagen muy similar al de dar un *click*.

```
Pid=""+"Tecnología"+""
```

La sentencia anterior ingresa el parámetro necesario para que la función *LoadAllLocales()* realice la búsqueda por tipo.

```
localList.clear();
```

Esta sentencia limpia la lista para que cada vez que se ejecute realice el proceso antes mencionado nuevamente y sobre todo no queden datos ya cargados anteriormente.

El código 3.34 muestra la función carga los datos correspondientes a los locales por tipo.

```

1. class LoadAllLocales extends AsyncTask<String, String, String> {
    //Se ejecuta el dialogo de Progreso
2. @Override
3. protectedvoid onPreExecute() {
4. super.onPreExecute();
5. pDialog = new ProgressDialog(LocalesActivity.this);
6. pDialog.setMessage("Cargando Locales espere porfavor...");
7. pDialog.setIndeterminate(false);
8. pDialog.setCancelable(false);
9. pDialog.show();
10. }
11. @Override
12. protected String doInBackground(String... args) {
    // Se crean los parámetros a pasar
13. List<NameValuePair> params = new ArrayList<NameValuePair>();
14. params.add(new BasicNameValuePair("Tipo", pid));
    // Se obtiene el string JSON de la URL
15. JSONObject json = JSONArray.makeHttpRequest(url_all_products, "GET",
16. params);
    // Chequea los logs de respuesta
17. Log.d("All Locales: ", json.toString());
18. try {
    // Chequea si fue exitoso
19. int success = json.getInt(TAG_SUCCESS);
20. if (success == 1) {
    // Obtiene el arreglo de locales
21. locales = json.getJSONArray(TAG_Locales);
    // Mapea cada elemento del arreglo
22. for (int i = 0; i < locales.length(); i++) {
23. JSONObject c = locales.getJSONObject(i);
    // Almacena cada ítem en l variable correspondiente
24. String id = c.getString(TAG_id_Local);
25. String name = c.getString(TAG_Nombre);
    // Crea un Nuevo HASHMAP
26. HashMap<String, String> map = new HashMap<String,
27. String>();
    // Añade a cada miembro de la lista su valor
28. map.put(TAG_id_Local, id);
29. map.put(TAG_Nombre, name);
    // Añade los datos a la lista de arreglo
30. localList.add(map);

```

**Código 3.34:** Datos de Locales por tipo

- **AsyncTask:** Clase auxiliar proporcionada por Android para ejecutar tareas en segundo plano, debido que al utilizar el hilo principal para ejecutar el proceso provocaría lentitud y posibles errores de ejecución.
- **HashMap<String, String>:** Esta función de Android permite mostrar elementos compuestos de una lista, donde en cada fila aparece más de un elemento.
- **Map:** Con este método se puede agregar elementos del *HashMap* a un *Arraylist* o lista de Arreglos.

Después de completar la tarea en segundo plano disminuye el diálogo de progreso como lo muestra el código 3.35.

```

1. @Override
2.   protected void onPostExecute(String file_url) {
3.       // Disminuye el progress dialog
4.       progressDialog.dismiss();
5.       // Actualiza el hilo en Segundo plano
6.       runOnUiThread(new Runnable() {
7.           @Override
8.           public void run() {
9.               /**Adapta los datos a la lista
10.              ListAdapter adapter = new SimpleAdapter(
11.                  LocalesActivity.this, localList,
12.                  R.layout.list_item, new String[] { TAG_id_Local, TAG_Nombre},
13.                  new int[] { R.id.pid, R.id.name });
14.               // Actualiza el listview
15.               listView.setAdapter(adapter);
16.           }
17.       });

```

**Código 3.35:** Adaptación de datos en la lista

“*Localestipo*” es la interfaz xml correspondiente a la actividad *LocalesActivity.java*.

En la figura 3.10 muestra su interfaz en modo gráfico.



**Figura 3.10:** Modo gráfico de la interfaz localestipo.xml

### 3.6.6 CLASE “BÚSQUEDA PRODUCTOS ACTIVITY”

La clase “*BusquedaProductosActivity.java*” contiene un menú gráfico de las diferentes categorías en las que se ha dividido a los locales del Centro Comercial y al seleccionar una de ellas realiza una búsqueda por tipo de los productos en la base de datos y muestra una lista con los resultados, lista que al seleccionar nos dirige a otra actividad con información más detallada de cada producto. “*Productos*” es la interfaz xml correspondiente a la actividad “*BusquedaProductoActivity.java*”. En la figura 3.11 se muestra su interfaz en modo gráfico.

### 3.6.7 CLASE “OFERTAS ACTIVITY”

La clase “*OfertasActivity.java*” es la actividad contiene un menú gráfico de las diferentes categorías en las que se ha dividido a los locales del Centro Comercial, al seleccionar una de ellas realiza una búsqueda por tipo de las ofertas en la base de datos y muestra una lista con los resultados, lista que al seleccionar nos dirige a otra actividad con información más detallada de cada oferta.





Figura 3.11: Modo gráfico de la interfaz productos.xml

“Ofertas” es la interfaz xml correspondiente a la actividad OfertasActivity.java. La figura 3.12 muestra su interfaz en modo gráfico.



Figura 3.12: Modo gráfico de la interfaz ofertas.xml

### 3.8.8 CLASE “LOCAL DETALLES ACTIVITY”

La clase “LocalDetallesActivity.java” se extiende a una *ListActivity* es decir una actividad de listas, esta clase recibe como parámetro el *id\_Local*, clave primaria de la tabla *local*, parámetro enviado por la clase *locales*, con este parámetro realiza la búsqueda de la información detallada de un local y de sus productos, contiene un botón que dirige a la actividad “OfertasLocalActivity.java” que muestra

las ofertas de un local en específico y a su vez al detalle de un producto mediante la Clase “*ProductoDetallesActivity*”.

Antes de cargar la interfaz es decir, la función *onCreate* se obtiene el parámetro enviado por la clase “*LocalesActivity.java*”. El siguiente código muestra como se obtiene el parámetro *id\_Local* para su posterior utilización en las respectivas funciones.

```

@Override
1. public void onCreate(Bundle savedInstanceState) {
2.     super.onCreate(savedInstanceState);
3.     setContentView(R.layout.local);
4.     Intent i = getIntent();
       // Obtiene el id_Local
5.     pid = i.getStringExtra(TAG_id_Local);
       // Ejecuta en Segundo plano la función Get Local
       Details
6.     new GetLocalDetails().execute();

```

**Código 3.36:** Obtención del id de un local

Los *intents* son el medio de activación de los componentes, contiene los datos que describen la operación que desarrollará el componente a quien va dirigido. Se declaran en el “*Android Manifest.xml*” descrito más adelante con la etiqueta <Intent>.

Pueden ser explícitos o implícitos. Los implícitos no especifican el componente al que va destinado, mientras que el explícito, sí. Según el componente, los “*intents*” se describen así:

- ***startActivity(Intent)***: Método que lanza una actividad.
- ***getIntent()***: Método que extrae la información de un intento.
- ***getListView()***: obtiene los elementos pertenecientes a una lista.

Esta clase contiene seis funciones para la obtención de la información *GetLocalDetails*, *LoadAllProductos*, *LoadAllLaptops*, *LoadAllTablets*, *LoadAllImpresoras*, *LoadAllImpresoras*.

- **Función *GetLocalDetails*** (**class** *GetLocalDetails* **extends** *AsyncTask<String, String, String>*), es una función que se ejecuta en segundo plano para evitar posibles errores al ejecutar la aplicación con el hilo principal que maneja la misma.

El *id\_Local* obtenido en esta función es el parámetro con el que realiza su procedimiento.

Se crea la lista que compara dos parámetros de cada elemento del arreglo.

```
List<NameValuePair> params = new ArrayList<NameValuePair>();
params.add(new BasicNameValuePair("id_Local", pid));
```

Para este caso se compara el *id\_Local = pid* (valor obtenido).

El objeto JSON que contiene la *URL* respectiva del servicio web requerido para la búsqueda, utiliza el URL siguiente:

```
url_local_detials = "http://172.30.8.128/xampp/phpfiles/get_local.php";
```

Es decir el servicio web que devuelve la información de un local por su *id\_Local*, explicado anteriormente.

```
JSONObject json = jsonParser.makeHttpRequest(url_local_detials, "GET",
params);
```

Como se observa, la sentencia anterior contiene el método GET que realiza la obtención de los datos, previo el paso de los parámetros respectivos, usando la clase "JSONParser" para la lectura de los datos.

Para el caso de los locales comerciales de tipo “Tecnología”, se han agregado 4 sub categorías para que al usuario se le facilite la búsqueda de productos.

Las categorías son las siguientes:

- *Laptops* (muestra todas las laptops del respectivo local).
- *Tablets* (muestra todas las tablets del respectivo local).
- *Impresoras* (muestra todas las impresoras disponibles del respectivo local).
- *Otros* (muestra los artículos tecnológicos en general).

Se obtiene el *string* del “Tipo”, sobre el cual realiza la comparación.

El código 3.37 muestra que si el atributo Tipo=“Tecnología”, los botones que realizan la búsqueda por categoría se vuelven visibles.

```
String tipo = product.getString(TAG_Tipo).toString();
if(tipo.equals("Tecnologia")){
    laptop.setVisibility(View.VISIBLE);
    tablet.setVisibility(View.VISIBLE);
    impresora.setVisibility(View.VISIBLE);
    otros.setVisibility(View.VISIBLE);
    txtLaptop.setVisibility(View.VISIBLE);
    txtTablet.setVisibility(View.VISIBLE);
    txtImpresora.setVisibility(View.VISIBLE);
    txtOtros.setVisibility(View.VISIBLE);
}
```

**Código 3.37:** Comparación en locales del tipo tecnología

En la figura 3.13 se muestra la interfaz de lo anteriormente explicado.



**Figura 3.13:** Interfaz - Productos de un local de tecnología por categoría.

Para el resto de categorías de los diferentes locales únicamente se muestra una imagen general de cada uno, se realiza la misma comparación Tipo="Categoría" (donde categoría puede ser: juegos, joyería, audio-video, ropa-gala, ropa-casual, camisetas, regalos, comida, música, servicio-técnico).

A continuación el código 3.38 muestra un ejemplo de esta comparación.

```

elseif (tipo.equals("Joyeria")){
new LoadAllProductos().execute();
prod.setVisibility(View.VISIBLE);
prod.setImageResource(R.drawable.joyeria);
}

```

**Código 3.38:** Comparación para el caso Joyeria

La figura 3.14 muestra la interfaz gráfica para un ejemplo de un local de joyas.

- **Función** *LoadAllProductos* (**class** LoadAllProductos **extends** AsyncTask<String, String, String>) al igual que la anterior se ejecuta en segundo plano.



**Figura 3.14:** Interfaz-Local de otra categoría para este caso del Tipo="Joyería"

El *id\_Local* obtenido anteriormente en esta función es el parámetro necesario para ejecutar la búsqueda de los productos en general de un local específico.

El objeto JSON es el siguiente:

```
JSONObject json = jParser.makeHttpRequest(url_all_productos, "GET",
params);
```

Los parámetros que necesita para ejecutar la búsqueda son:

- *Id\_Local* respectivo
- *url\_all\_productos* =  
[http://172.30.8.128/xampp/phpfiles/get\\_productos.php](http://172.30.8.128/xampp/phpfiles/get_productos.php)

Es decir el servicio *web* que devuelve la información de un local por su *id\_Local*, explicado anteriormente.

- **Función *LoadAllLaptops*** (**class** `LoadAllLaptops` **extends** `AsyncTask<String, String, String>`) es una función que se ejecuta en segundo plano. El *id\_Local* obtenido en esta función es el parámetro necesario para ejecutar la búsqueda de las laptops disponibles en un local específico.

El objeto JSON es el siguiente:

```
JSONObject json = jParser.makeHttpRequest(url_all_laptops, "GET",
                                           params);
```

Los parámetros que necesita para ejecutar la búsqueda son:

- *id\_Local* respectivo
- *url\_all\_laptops* =  
[http://172.30.8.128/xampp/phpfiles/get\\_laptops.php](http://172.30.8.128/xampp/phpfiles/get_laptops.php)

Es decir el servicio *web* que devuelve la información de un local por su *id\_Local*, explicado anteriormente.

- **Función *LoadAllTablets*** (**class** `LoadAllTablets` **extends** `AsyncTask<String, String, String>`) se ejecuta en segundo plano. El *id\_Local* obtenido en esta función es el parámetro necesario para ejecutar la búsqueda de las tablets disponibles en un local específico.

El objeto JSON es el siguiente:

```
JSONObject json = jParser.makeHttpRequest(url_all_tables, "GET",
                                           params);
```

Los parámetros que necesita para ejecutar la búsqueda son:

- *id\_Local* respectivo
- *url\_all\_tablets* =

[http://172.30.8.128/xampp/phpfiles/get\\_tablets.php](http://172.30.8.128/xampp/phpfiles/get_tablets.php)

Es decir el servicio *web* que devuelve la información de un local por su *id\_Local*, explicado anteriormente.

- **Función *LoadAllImpresoras*** (**class** *LoadAllImpresoras* **extends** *AsyncTask<String, String, String>*), mediante el *id\_Local* obtenido ejecuta la búsqueda de las impresoras disponibles en un local específico.

El objeto JSON es el siguiente:

```
JSONObject json = jParser.makeHttpRequest(url_all_impresoras, "GET",
params);
```

Los parámetros que necesita para ejecutar la búsqueda son:

- *id\_Local* respectivo
- *url\_all\_impresoras* = [http://172.30.8.128/xampp/phpfiles/get\\_impresoras.php](http://172.30.8.128/xampp/phpfiles/get_impresoras.php)

- **Función *LoadAllOtros*** (**class** *LoadAllImpresoras* **extends** *AsyncTask<String, String, String>*) de igual forma es una función que se ejecuta en segundo plano. El *id\_Local* obtenido anteriormente es el parámetro necesario para ejecutar la búsqueda de los artículos tecnológicos en general disponibles en un local específico.

El objeto JSON es el siguiente:

```
JSONObject json = jParser.makeHttpRequest(url_all_otros, "GET",
params);
```

Los parámetros que necesita para ejecutar la búsqueda son:

- *id\_Local* respectivo
- *url\_all\_otros*=[http://172.30.8.128/xampp/phpfiles/get\\_otros.php](http://172.30.8.128/xampp/phpfiles/get_otros.php)



- **Función LoadAllProducts** se ejecuta en el *if* respectivo de cada categoría como lo muestra el código 3.38.

```

elseif (tipo.equals("Regalos")){
new LoadAllProductos().execute();
prod.setVisibility(View.VISIBLE);
prod.setImageResource(R.drawable.regalos);
}

```

**Código 3.38:** Ejecución de la función LoadAllProducts

Se ejecutan en el evento *click* que maneja a cada *ImageButton* como se muestra en el código 3.39.

```

//Carga Solo laptops
publicvoid cargarLaptops(View v){
new LoadAllLaptops().execute();
localList.clear();
}

```

**Código 3.39:** Ejecución de la función LoadAllLaptops

El código 3.40 muestra el evento que al cambiar de actividad, envía el *id\_Local* a la siguiente actividad llamada *OfertasLocalActivity* que muestra las ofertas de cada local.

```

1. publicvoid cambiarActividad(View v){
   //Obtiene el contenido del cuadro de texto
2. num = txtLocal.getText().toString();
   //Crea un nuevo intent para llamar a la siguiente actividad
3. Intent i = newIntent(this, OfertasLocalActivity.class);
   //Guarda la cadena en el intent y se nombra como resultado
4. i.putExtra("resultado", num);
   //Ejecuta para que muestre la segunda actividad
5. startActivity(i);
}

```

**Código 3.40:** Evento Cambiar de Actividad

En la variable tipo *string* se obtiene el valor que se encuentra en el *textView*, se almacena en la cadena "resultado" y pasa a la siguiente actividad llamada "OfertasLocalActivity.class".

*Local* es la interfaz xml correspondiente a la actividad "LocalDetallesActivity.java" que contiene cada uno de los elementos de la misma.

La figura 3.15 muestra su interfaz en modo gráfico.



**Figura 3.15:** Modo gráfico de la interfaz local.xml

Por defecto los *ImageButtons* e *ImageView* no están visibles, como se muestra en el código 3.41.

```
<ImageButton
    android:id="@+id/prod1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginLeft="100dp"
    android:layout_marginTop="210dp"
    android:onClick="cargarLaptops"
    android:src="@drawable/fifa13"
    android:visibility="gone"
    android:background="@android:color/transparent"/>
```

**Código 3.41:** Ejemplo de *ImageButton*

- ***android:onClick***: Propiedad que controla directamente el evento *click* haciendo de esta manera más fácil su utilización y su llamado.
- ***android:visibility***: Propiedad que controla la visibilidad de un elemento si ésta toma el valor de “*gone*”, por defecto hace que el elemento no sea visible.
- ***android:background***: Propiedad que permite controlar el color fondo de los elementos, para este caso es transparente mostrando solamente la imagen.

### 3.6.9 CLASE “OFERTAS LOCAL ACTIVITY”

La clase “*OfertasLocalActivity.java*” se extiende a una *ListActivity* es decir una actividad de listas, esta clase recibe como parámetro el *id\_Local* clave primaria de la tabla local, parámetro enviado por la clase “*OfertasLocalActivity.java*”, con este parámetro realiza la búsqueda de la información detallada de un local y de sus ofertas.

El código 3.42 muestra en la función *onCreate*, que se ejecuta antes de cargar la interfaz, donde crea el paquete de datos *Bundle* llamado *local*, el mismo que permite la recuperación de datos enviado a la cadena de datos “*resultado*”.

```

@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.ofertaslocal);
    // Obteniendo el id_Local enviado por la clase
    LocalDetallesActivity.java
    txtLocal = (TextView) findViewById(R.id.inputLocal);
    Bundle local= getIntent().getExtras();
    num = local.getString("resultado");

```

**Código 3.42:** Función *onCreate* de la clase “Ofertas Local”

Los *intents* son el medio de activación de los componentes, contiene los datos que describen la operación que desarrollará el componente a quien va dirigido. Se declaran en el *Android Manifest.xml* descrito mas adelante con la etiqueta `<Intent>`.

Los *intents* se tratan de diferentes maneras:

- ***getIntent()***: Método que extrae la información de un intento y del elemento al que está asociado.
- ***getExtras()***: Método que obtiene la información extra contenido dentro de una cadena de caracteres.

Esta clase contiene tres funciones para la obtención de la información: ***GetLocalDetails, LoadAllProductos, LoadAllOfertas.***

- **Función *GetLocalDetails*** (**class** `GetLocalDetails` **extends** `AsyncTask<String, String, String>`), es una función que se ejecuta en segundo plano, el *id\_Local* obtenido anteriormente es el parámetro para que la función realice su procedimiento.

Se crea la lista que compara dos parámetros de cada elemento del arreglo.

```
List<NameValuePair> params = new ArrayList<NameValuePair>();
params.add(new BasicNameValuePair("id_Local", pid));
```

Para este caso se compara el *id\_Local = pid* (el valor obtenido).

Se genera el objeto JSON que contendrá el siguiente URL respectiva del servicio *web* requerido para la búsqueda.

```
url_local_detials = "http://172.30.8.128/xampp/phpfiles/get_local.php";
```

Es decir el servicio *web* que devuelve la información de un local por su *id\_Local*, explicado anteriormente.

```
JSONObject json = jsonParser.makeHttpRequest(url_local_detials, "GET",
params);
```

Como se observa contiene el método *GET* que realiza la obtención de los datos, previo el paso de los parámetros respectivos, utilizando la clase *JSONParser* para la lectura de datos.

- **Función *LoadAllProductos*** (**class** *LoadAllOfertas* **extends** *AsyncTask<String, String, String>*) se ejecuta en segundo plano, el *id\_Local* obtenido anteriormente en esta función es el parámetro necesario para ejecutar la búsqueda de las ofertas en general de un local específico.

El objeto JSON es el siguiente:

```
JSONObject json = jParser.makeHttpRequest(url_all_ofertas, "GET",
params);
```

Los parámetros que necesita para ejecutar la búsqueda son:

- *Id\_Local* respectivo
- *url\_all\_productos*  
[http://172.30.8.128/xampp/phpfiles/get\\_ofertas.php](http://172.30.8.128/xampp/phpfiles/get_ofertas.php)

Para las diferentes categorías de los diferentes locales se muestra una imagen general de cada uno de igual forma se realiza la comparación *Tipo="Categoria"* (donde categoría puede ser: juegos, joyería, audio-video, ropa-gala, ropa-casual, camisetas, regalos, comida, música, servicio-técnico).

- Función *LoadAllOfertas* se ejecuta en el *if* respectivo de cada categoría, así como se muestra en el código 3.43.

```

elseif (tipo.equals("Regalos")){
new LoadAllProductos().execute();
prod.setVisibility(View.VISIBLE);
prod.setImageResource(R.drawable.regalos);
}

```

**Código 3.43:** Función *LoadAllProductos*

En la figura 3.16 se observa un ejemplo de la interfaz gráfica en la categoría Ropa-Casual.



**Figura 3.16:** interfaz de un local y sus ofertas

“Ofertas local” es la interfaz xml correspondiente a la actividad *LocalDetallesActivity.java* que contiene cada uno de los elementos de la misma.

En la figura 3.17 se muestra su interfaz en modo gráfico (*Ofertaslocal*).



**Figura 3.17:** Modo gráfico de la interfaz ofertaslocal.xml

### 3.6.10 CLASE “PRODUCTO DETALLES ACTIVITY”

La clase “*ProductosDetallesActivity.java*” se extiende a una *Activity*, esta clase recibe como parámetro el *id\_LocPro*, clave primaria de la tabla *localproducto*, enviado por las clases *ProductoLocalActivity.java*, *ProductosActivity.java* puesto que ambas clases contienen la lista de los productos de un local y al dar *clic* en el ítem respectivo envían este parámetro, para que la clase muestre los detalles de un producto, es decir nombre del producto, características técnicas o características propias del mismo así como también su precio, stock del mismo.

Esta clase como se ha mencionado recibe como parámetro el *id\_LocPro*:

```
pid = i.getStringExtra(TAG_id_LocPro);
```

- **getStringExtra( ):** Método que recibe el valor string que se envió como información adicional, para este caso el valor del *id\_LocPro* de cada producto.

La siguiente función muestra la información del local a quien le pertenece el producto y además las especificaciones de cada producto.

```
<GetLocalDetails extends AsyncTask<String, String, String
```

En esta función se realiza dos comparaciones, la primera para mostrar el logo de cada local y la segunda para especificar los atributos de cada producto, por ejemplo, en el caso de un artículo de vestir tiene atributos tales como la talla, color, material, etc., pero en cambio para el caso de un artículo tecnológico tiene otros tipos de atributos como por ejemplo para el caso de una laptop tendremos atributos como disco, memoria RAM, pantalla, etc., realiza la comparación para que la aplicación pueda discriminar lo descrito.

Los parámetros que necesita esta función son:

- *Id\_LocPro* descrito anteriormente.
- La *url* correspondiente al servicio web respectivo.

```
url_product_details"http://192.168.0.104/xampp/phpfiles/get_producto_detalle.php";
```

Comparación para mostrar el logotipo por local:

```
String nom = product.getString(TAG_Nombre).toString();
```

Se extrae el string del producto que contiene su nombre en la variable "nom".

Se compara si el nombre corresponde a un local específico mostramos la imagen de su logotipo, para el ejemplo mostrado cargara la imagen del logotipo del local "NOVICOMPU", del centro comercial.

En el código 3.44 se muestra la comparación para mostrar los parámetros específicos correspondientes a cada producto.



```

if(nom.equals("NOVICOMPU")){
    ImageView logo = (ImageView) findViewById(R.id.logo);
    logo.setImageResource(R.drawable.novicompu);
}

```

### Código 3.44: Parámetros de cada producto

Se declara el *string* "IMPRESORA" y se lo compara con el valor extraído de la base de datos, para el ejemplo mostrará las características técnicas correspondientes a una impresora.

```

String pro = product.getString(TAG_Producto).toString();
String imp= "IMPRESORA";

```

Con el código 3.45 se carga cada uno de los parámetros correspondientes a una impresora y adicionalmente dos imágenes de la misma.

La figura 3.18 se muestra gráficamente lo descrito anteriormente



**Figura 3.18:** Interfaz producto detalles muestra como ejemplo una tv.

```

if(pro.equals(imp)){
    //Muestra el nombre del producto en el txtProducto
    txtProducto=(TextView)findViewById(R.id.inputProducto);
    txtProducto.setText(product.getString(TAG_Producto));
    //Muestra el precio del producto en el txt precio
    txtPrecio=(TextView)findViewById(R.id.inputPrecio);
    txtPrecio.setText(product.getString(TAG_Precio));
    //Muestra el modelo en el txtModelo
    txtModelo=(TextView)findViewById(R.id.inputModelo);
    txtModelo.setText(product.getString(TAG_Modelo));
    //Muestra la compatibilidad
    txtVarios1 = (TextView) findViewById(R.id.inputDis);
    txtVarios1.setText("Compatibilidad:");
    // Muestra es Sistema Operativo
    txtDisco=(TextView)findViewById(R.id.inputDisco);
    txtDisco.setText(product.getString(TAG_Sistema));
    //Muestra información de los cartuchos
    txtVarios2 = (TextView) findViewById(R.id.inputOtros1);
    txtVarios2.setText("Cartuchos:");
    //Muestra informacion adicional
    txtProcesador=(TextView)findViewById(R.id.inputProcesador);
    txtProcesador.setText(product.getString(TAG_Varios));
    //Muestra la imagen de la impresora
    ImageView prod1 = (ImageView) findViewById(R.id.prod1);
    prod1.setImageResource(R.drawable.impresora1);
    //Muestra la segunda imagen de la impresora
    ImageView prod2 = (ImageView) findViewById(R.id.prod2);
    prod2.setImageResource(R.drawable.impresora2);
}

```

**Código 3.45:** Características cargadas del producto seleccionado

“Productinfo” es la interfaz xml correspondiente a la actividad “ProductoDetallesActivity.java” que contiene cada uno de los elementos de la misma.

En la figura 3.19 se muestra su interfaz en modo gráfico.



**Figura 3.19:** Modo gráfico de la interfaz productinfo.xml

### 3.6.11 CLASE “OFERTA DETALLES ACTIVITY”

La clase “*OfertaDetallesActivity.java*” se extiende a una *Activity*, esta clase recibe como parámetro el *id\_LocPro*, clave primaria de la tabla *localproducto*, parámetro enviado por las clases *OfertasLocalActivity.java*, *OfertasActivity.java* puesto que ambas clases con tienen la lista de las ofertas de un local y al dar *clic* en el ítem respectivo envían este parámetro, para que de esta manera la clase muestre los detalles de la oferta, es decir nombre de la oferta, características técnicas o características propias de la misma así como también su precio, *stock*.

Esta clase es similar a la anterior por lo que no se describirá sus funciones pues es exactamente lo mismo cambiando únicamente la URL del servicio *web*, y aumentando el precio de la oferta y *stock* de la misma.

Los parámetros de entrada que utiliza son:

- *Id\_LocPro* del producto respectivo
- *url\_oferta\_detalle*= "http://192.168.0.104/xampp/phpfiles/get\_oferta\_detalle.php";

En la figura 3.20 se muestra la interfaz que resulta del código en funcionamiento.



**Figura 3.20:** Interfaz oferta detalles muestra como ejemplo una laptop.

*Ofertinfo* es la interfaz xml correspondiente a la actividad *OfertaDetallesActivity.java* que contiene cada uno de los elementos de la misma.

En la figura 3.21 se muestra la interfaz *ofertinfo.xml*.

### 3.6.12 CLASE SUBSUELO UNO ACTIVITY

La clase "SubsuelodosActivity.java" se extiende a una Actividad, esta clase contiene el mapa del subsuelo dos del Centro Comercial consta de 38 *ImageButtons* que contienen los marcadores con el número de cada local, al dar clic en uno de estos muestra el nombre del local respectivo y al seleccionar el botón "mas información" envía el *id\_Local* a la clase *Local Activity* explicada anteriormente.

Las variables y arreglos del siguiente código ya fueron explicados anteriormente, se crea la lista de arreglo que contendrá la lista de locales llamada "*localList*", se

declara el *string* que contendrá el URL del servicio *web* respectivo, los nodos necesarios para el manejo de la clase JSON (clase anteriormente explicada) y finalmente el objeto *JSONArray* llamado "locales", que contendrá la información requerida en este formato, tal como muestra el código 3.47.



**Figura 3.21:** Modo gráfico de la interfaz ofertinfo.xml

```

ArrayList<HashMap<String, String>>localList;
Private static String url_locales=
"http://192.168.0.104/xampp/phpfiles/get_local.php";
// Nombres de los Nodos JSON
privatstaticfinal String TAG_SUCCESS = "success";
privatstaticfinal String TAG_Local = "local";
privatstaticfinal String TAG_id_Local = "id_Local";
privatstaticfinal String TAG_Nombre = "Nombre";
// JSONArray llamado locales
JSONArray locales = null;

```

**Código 3.47:** Creación del *ArrayList* "localList"

Esta clase contiene la función *GetLocalDetails* (**class** *GetLocalDetails* **extends** *AsyncTask*<String, String, String>) que se ejecuta en segundo plano, recibe los parámetros necesarios anteriormente declarados, y obtiene la información necesaria de cada local.

```
// Compara los parámetros
List<NameValuePair> params = new ArrayList<NameValuePair>();
params.add(new BasicNameValuePair("id_Local", pid));
//Crea el objeto JSON con el respectivo http request en este caso GET para
obtener información
JSONObject json = jsonParser.makeHttpRequest(url_locales, "GET", params);
```

**Código 3.48:** Creación del objeto *JSON GET*

El código 3.48, compara el *id\_Local* con su respectivo valor para este caso llama al “*pid*”, este valor será enviado en el evento *touch* de cada marcador como se observa más adelante.

Ejecuta esta función en el respectivo evento *touch* de cada marcador, como se muestra en el código 3.49.

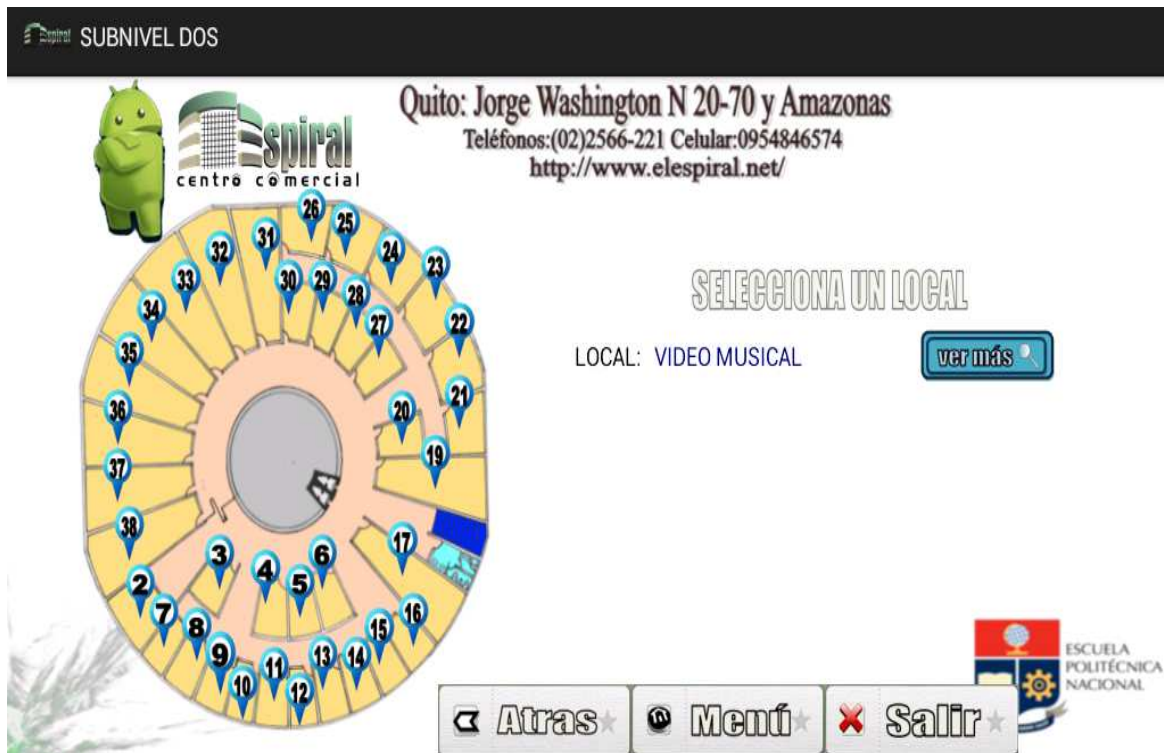
```
@Override
public boolean onTouch(View v, MotionEvent m){
    switch(m.getAction())
    {
        case MotionEvent.ACTION_DOWN:
        {
            switch(v.getId()) {
                case R.id.marcador2:
                    pid="\\"+"2"+"\\";
                    new GetLocalDetails().execute();
            }
        }
    }
}
```

**Código 3.49:** Evento *Touch* para obtener *GetLocalDetails*



*Subsuelo2* es la interfaz xml correspondiente a la actividad “*SubsuelodosActivity.java*” que contiene cada uno de los elementos de la misma.

La figura 3.22 muestra su interfaz en modo gráfico.



**Figura 3.22:** Modo gráfico de la interfaz subsuelodos.xml

### 3.8.13 CLASES QUE MANEJAN EL MAPA DEL SITIO DEL CENTRO COMERCIAL

Las clases que se describen a continuación manejan las mismas funciones de la clase *SubsuelodosActivity.java* anteriormente descrita, como única diferencia es el mapa del piso respectivo y los marcadores de que contienen el número de cada local.

En las figuras 3.23 a 3.30 se muestra la interfaz que mostrará las clases:

- *SubsuelounoActivity.java*

- *PlantabajaActivity.java*
- *NivelunoActivity.java*
- *NiveldosActivity.java*
- *NiveltresActivity.java*
- *NivelcuatroActivity.java*
- *NivelcincoActivity.java*
- *NivelseisActivity.java*



**Figura 3.23:** Modo gráfico de la interfaz subsuelouno.xml

Tanto en el subsuelo dos, como en el subsuelo uno, se encuentran la mayor cantidad de locales del centro comercial, para las siguientes plantas, se tendrá una distribución equitativa de locales por cada piso.



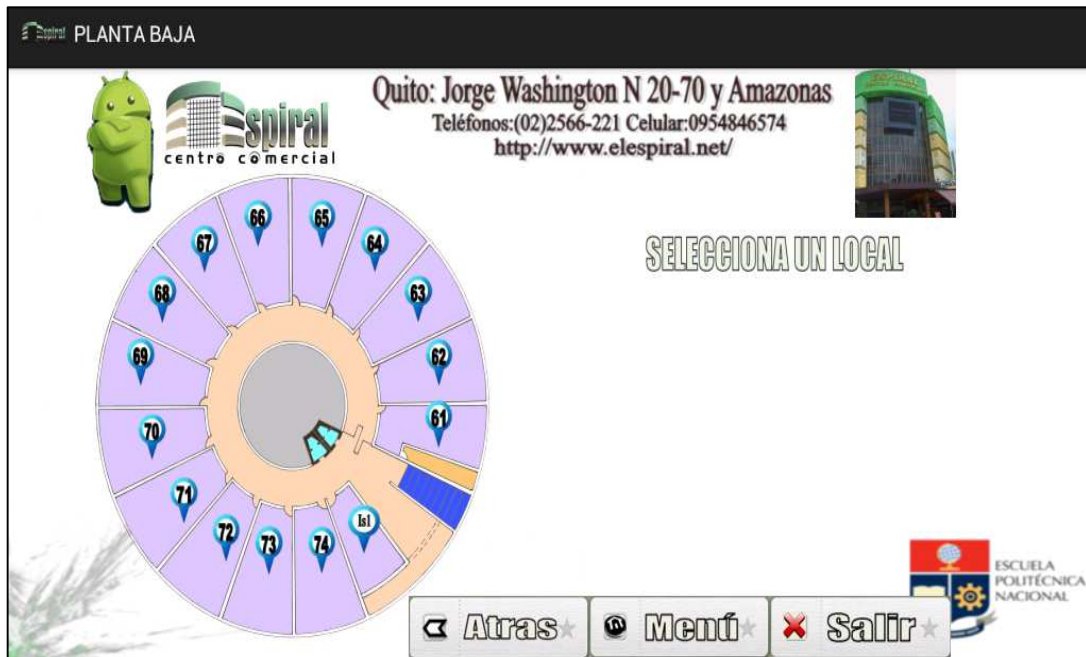


Figura 3.24: Modo gráfico de la interfaz plantabaja.xml

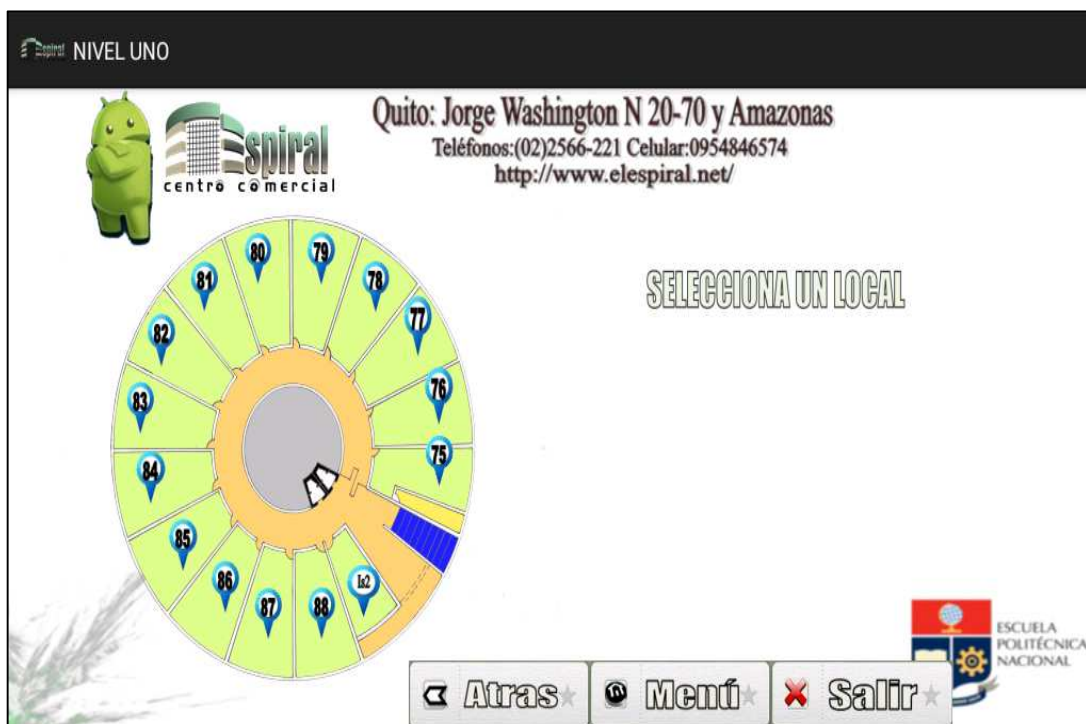


Figura 3.25: Modo gráfico de la interfaz niveluno.xml

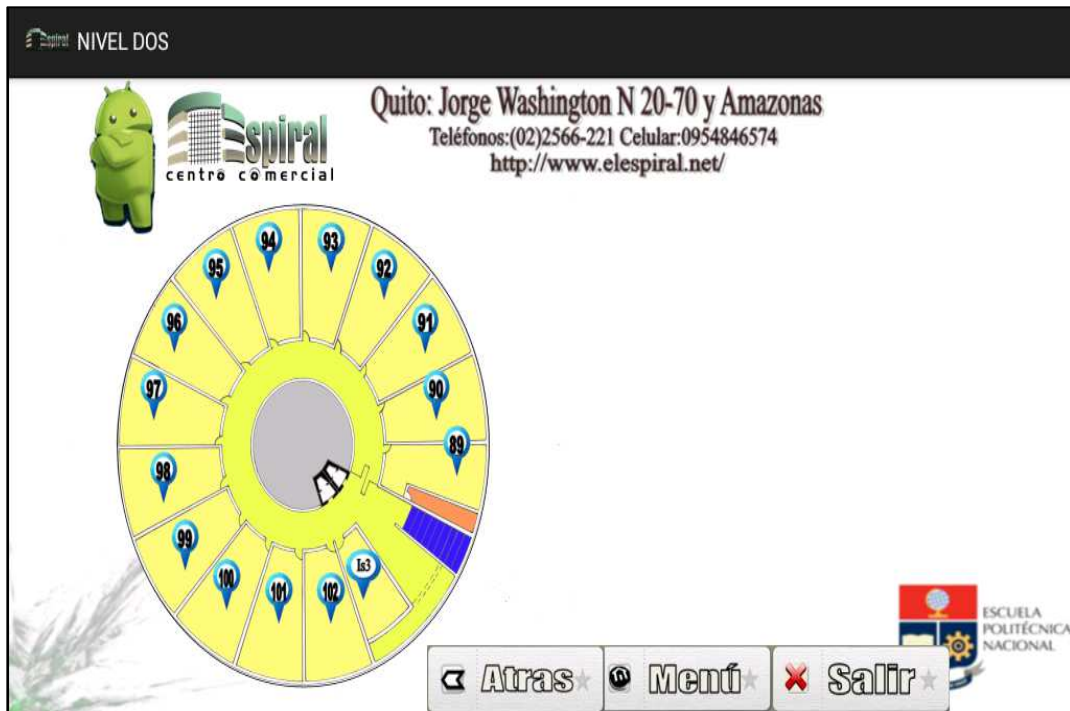


Figura 3.26: Modo gráfico de la interfaz niveldos.xml

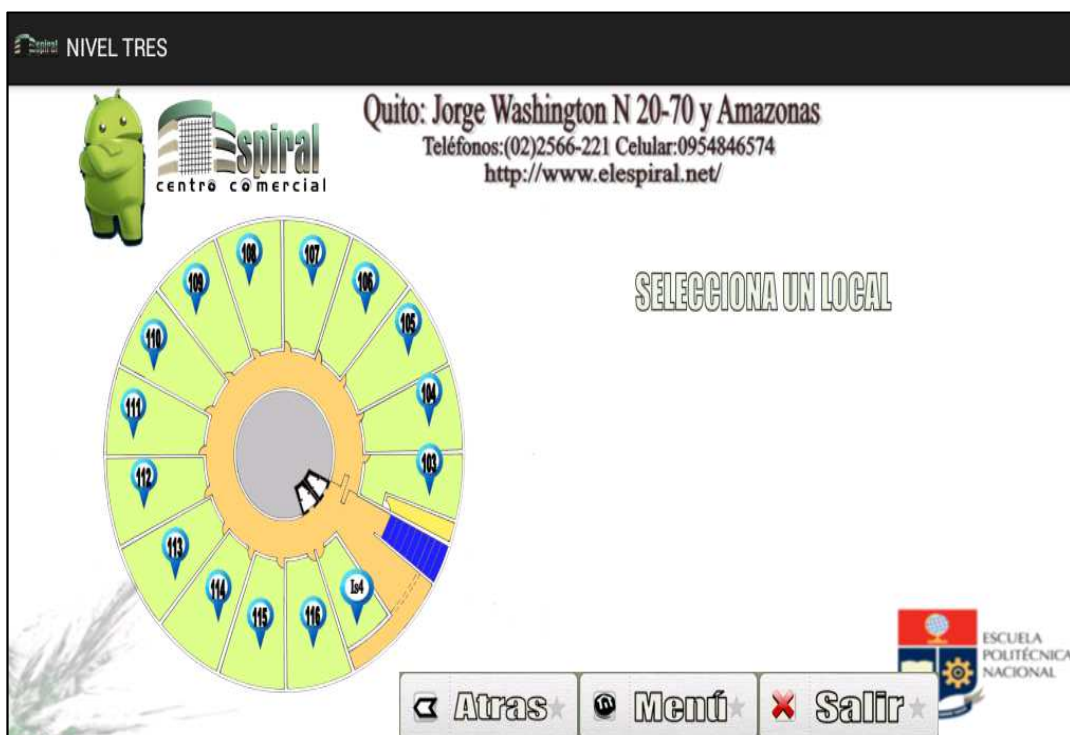


Figura 3.27: Modo gráfico de la interfaz niveltres.xml

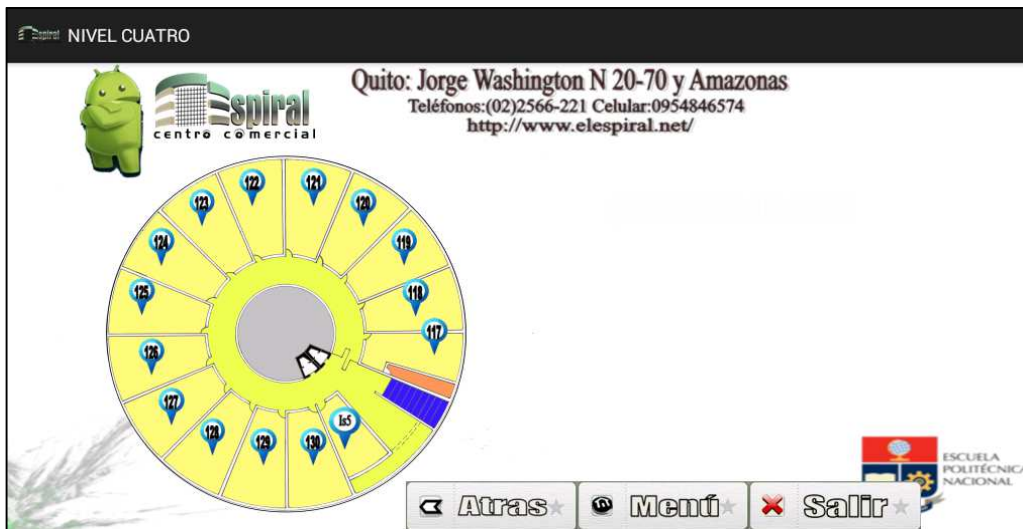


Figura 3.28: Modo gráfico de la interfaz nivelcuatro.xml

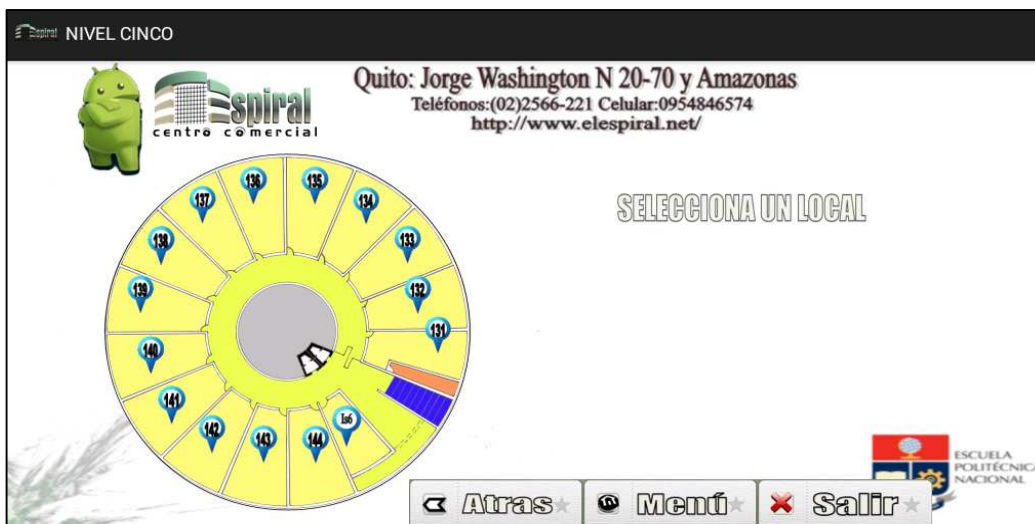


Figura 3.29: Modo gráfico de la interfaz nivelcinco.xml

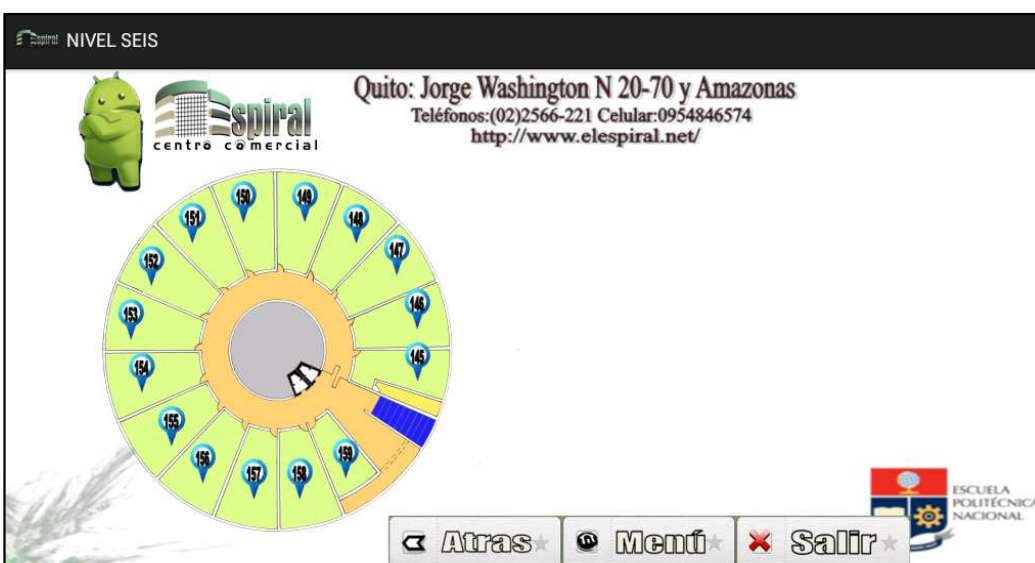


Figura 3.30: Modo gráfico de la interfaz nivelseis.xml

### 3.6.14 CLASE “BUZON ACTIVITY”

La clase “BuzonActivity.java” es la encargada del envío de las sugerencias de quienes visitan el centro comercial, para esto esta clase hace uso de los métodos GET Y POST, para la obtención y envío de datos respectivamente. En el código 3.50 se muestra la declaración de variables y parámetros necesarios.

```
// Declaramos en objeto JSON requerido
    JSONParser jParser = newJSONParser();
// Declaración del arreglo que maneja la lista de clientes
    ArrayList<HashMap<String, String>>clientList;
//URLs que contienen los servicios web
    privatestatic String url_clientes =
        "http://192.168.0.104/xampp/phpfiles/update_clientes.php";
    privatestatic String url_todosclientes =
        "http://192.168.0.104/xampp/phpfiles/get_clientes.php";
// Nombres de los nodos JSON
    privatestaticfinal String TAG_SUCCESS = "success";
    privatestaticfinal String TAG_Clientes = "cliente";
    privatestaticfinal String TAG_idCliente = "id_Cliente";
    privatestaticfinal String TAG_idCComercial = "id_CComercial";
    privatestaticfinal String TAG_Nombre = "Nombre";
    privatestaticfinal String TAG_Mail = "Mail";
    privatestaticfinal String TAG_Mensaje = "Mensaje";

// ArregloJSON de clientes
    JSONArray clientes = null;
```

**Código 3.50:** Parámetros de la clase “BuzonActivity”

Ésta clase contiene la función que realiza el envío del nombre, correo y la sugerencia respectiva de cada visitante. La función utilizada es la siguiente:

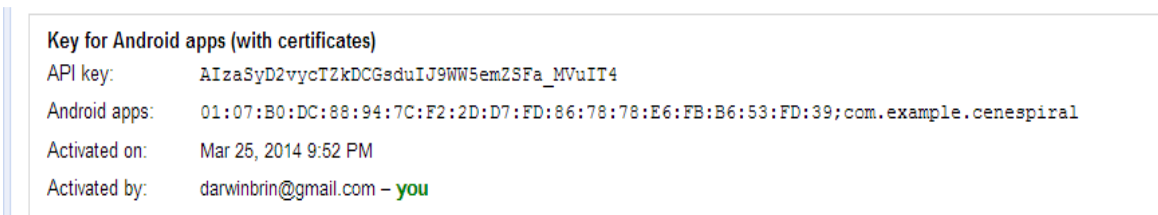
```
class SaveMensaje extends AsyncTask<String, String, String>
```

La función recibe los parámetros y variables declaradas anteriormente.

### 3.6.15 CLASE “MAPA ACTIVITY” [32]

La clase “*Mapa Activity*” muestra la ubicación en *google maps* del Centro Comercial Espiral y la posición exacta del usuario, usa el API de *Google Maps V2*, es decir la versión más reciente en el manejo de mapas y posicionamiento.

Es necesario aclarar que *Google Maps* no es un *software* libre por lo que está limitado a una serie de condiciones de servicio, se puede utilizar de forma gratuita siempre y cuando la aplicación no solicite más de 15000 codificaciones geográficas al día. Para poder utilizar el servicio es necesario registrar a la aplicación, luego del registro se proporciona una clave la misma que se debe incluir en el proyecto. La figura 3.31 muestra la clave generada para la aplicación “CenEspiral”.



**Figura 3.31:** API key de la aplicación “CenEspiral”.

En el fichero `AndroidManifest.xml`, dentro del elemento `<application>`, se añade el API key antes mencionado, como se muestra en el código 3.51.

```
<meta-data
    android:name="com.google.android.gms.version"
    android:value="@integer/google_play_services_version"/>
<meta-data
    android:name="com.google.android.maps.v2.API_KEY"
    android:value="
AIzaSyD2vycTZkDCGsduIJ9WW5emZSFa_MVuIT4_"/>
```

**Código 3.51:** Parámetros ingresados en el elemento `application`.



En el código 3.52 se define los permisos necesarios declarados dentro del elemento `<manifest>` para que la aplicación funcione correctamente.

```

<permission
    android:name="org.example.ejemplogooglemaps.permission.MAPS_RECEIVE"
    android:protectionLevel="signature"/>
<uses-permission
    android:name="org.example.ejemplogooglemaps.permission.MAPS_RECEIVE"/>
<uses-permission
    android:name="com.google.android.providers.gsf.permission.READ_GSERVICES"/>
<uses-permission android:name="android.permission.INTERNET"/>
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>

```

**Código 3.52:** Parámetros ingresados en el elemento *manifest*

Los permisos descritos permiten recibir mapas, acceso a internet, acceso al estado de red, almacenamiento externo y localización.

Las funciones implementadas en la clase “Mapa Activity” se describen a continuación:

- Declaración de variables: En el código 3.53 se muestra las variables para la clase “*MapaActivity*”, se definen la latitud y la longitud del centro comercial Espiral y la variable *GoogleMap*.

```

//Coordenadas latitud, longitud del Centro Comercial Espiral
private final LatLng ESP = new LatLng(-0.206599, -78.495901);
//Declaración de la variable GoogleMap
private GoogleMap mapa;

```

**Código 3.53:** Declaración de variable

- Función *Oncreate*: En el código 3.54 se muestra la función *OnCreate*.

```

@Override protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.mapasitio);
    //Declaración del fragment con la clase SupportMapFragment la misma que
    //contendra la vista del mapa

    mapa=((SupportMapFragment)getSupportFragmentManager().findFragmentById(R.id.ma
pes)).getMap();
    //Tipo de mapa
    mapa.setMapType(GoogleMap.MAP_TYPE_NORMAL);
    //Zoom
    mapa.moveCamera(CameraUpdateFactory.newLatLngZoom(ESP, 15));
    //Muestra mi localización
    mapa.setMyLocationEnabled(true);
    //Obtiene los controles de usuario en la interfaz del mapa
    //Habilita y deshabilita la brújula y el zoom del mapa
    mapa.getUiSettings().setZoomControlsEnabled(false);
    mapa.getUiSettings().setCompassEnabled(true);
    //Añade un marcador a la posición del centro comercial
    mapa.addMarker(new MarkerOptions()
        .position(ESP)
        .title("ESP")
        .snippet("Centro Comercial Espiral")
        .icon(BitmapDescriptorFactory.fromResource
(R.drawable.ic_launcher)).anchor(0.5f, 0.5f));
    //El marcador se maneja en el evento clic de mapa
    mapa.setOnMapClickListener(this);
}

```

### Código 3.54: Función *OnCreate*

- Funciones “Ir a Espiral”, “Mi posición”, “Marcador”

```

//Boton IR A ESPIRAL mueve la cámara a la posición del espiral
public void moveCamera(View view) {
    mapa.moveCamera(CameraUpdateFactory.newLatLng(ESP));
}

//Boton MI POSICIÓN mueve la cámara hacia la posición del usuario
public void animateCamera(View view) {
    if (mapa.getMyLocation() != null)
        mapa.animateCamera(CameraUpdateFactory.newLatLngZoom(
            new LatLng( mapa.getMyLocation().getLatitude(),
                mapa.getMyLocation().getLongitude()), 15));
}

```

### Código 3.55: Funciones “Ir A Espiral”, “Mi posición”, “Marcador”

```

//Boton MARCADOR añade un marcador a cualquier posición del mapa
public void addMarker(View view) {
    mapa.addMarker(new MarkerOptions().position(
        new LatLng(mapa.getCameraPosition().target.latitude,
            mapa.getCameraPosition().target.longitude));
    }
//Evento clic del mapa
@Override
public void onMapClick(LatLng puntoPulsado) {
    mapa.addMarker(new MarkerOptions().position(puntoPulsado).
        icon(BitmapDescriptorFactory
            .defaultMarker(BitmapDescriptorFactory.HUE_YELLOW)));
}

```

**Código 3.56:** Funciones “Ir A Espiral”, “Mi posición”, “Marcador” (Cont.)

**SupportMapFragment:** maneja el fragmento que contiene el mapa en la interfaz XML para la aplicación este fragmento se llama “mapes”.

**SetMapType:** Añade el tipo de mapa a utilizar por ejemplo en este caso mapa del tipo normal.

**MoveCamera:** Mueve la cámara hacia las coordenadas de latitud y longitud del Centro Comercial Espiral.

**SetMyLocation:** Obtiene la localización del dispositivo, si se indica la latitud y longitud se mueve la cámara a dicha posición.

```

mapa.animateCamera(CameraUpdateFactory.newLatLngZoom(new LatLng(
    mapa.getMyLocation().getLatitude(),mapa.getMyLocation().getLongitude()),
    15));

```

**SetZoomControlsEnabled:** Habilita o deshabilita el control del zoom.

**SetCompassEnabled:** Habilita o deshabilita la brújula que por lo general se muestran en los mapas de *google* para la orientación del usuario.

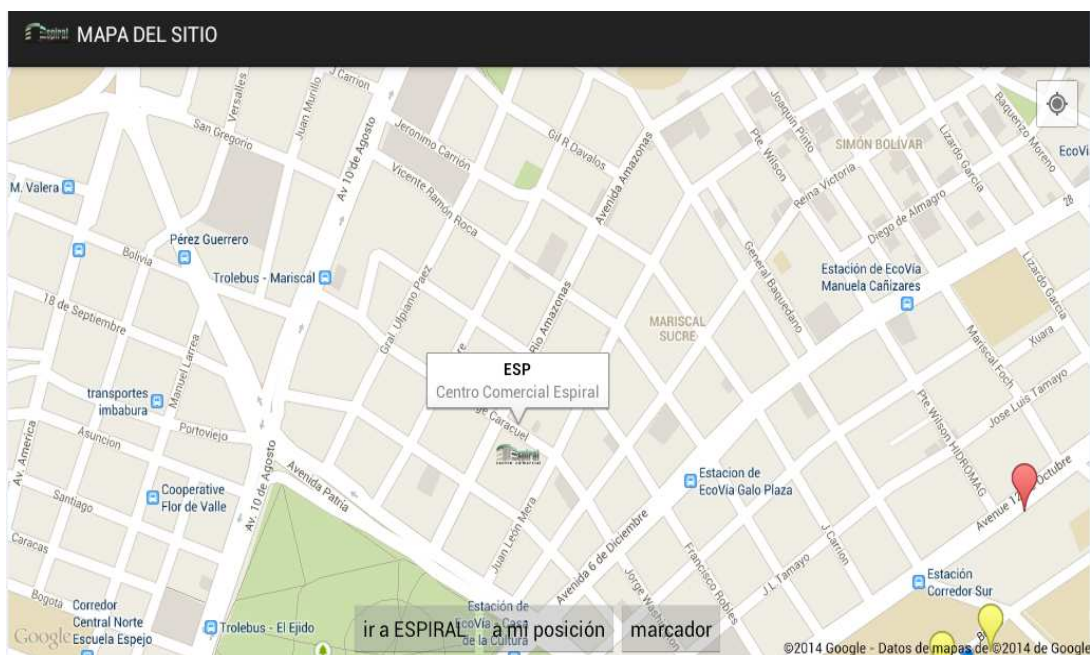


**MarkerOptions:** Contiene todas las opciones de marcador que dispone esta versión de *Google Maps* para *Android*, en este caso añade la etiqueta “ESP” y el nombre “CENTRO COMERCIAL ESPIRAL”.

De igual forma maneja la posición de un punto pulsado añade un marcador tipo de color amarillo, únicamente al dar clic en el botón “Marcador” y posteriormente en cualquier parte del mapa.

```
mapa.addMarker(new MarkerOptions().position(puntoPulsado).
    icon(BitmapDescriptorFactory
        .defaultMarker(BitmapDescriptorFactory.HUE_YELLOW));
```

La figura 3.32 muestra la interfaz de la Actividad del mapa anteriormente descrita.



**Figura 3.32:** Interfaz Ubicación del Centro Comercial Espiral

# CAPÍTULO 4

## 4. IMPLEMENTACIÓN DEL PROTOTIPO DE PRUEBA Y PRESENTACIÓN DEL COSTO REFERENCIAL DE LA SOLUCIÓN

En este capítulo se presentan los resultados que se obtuvo al armar el prototipo de prueba y se determinará el costo referencial de la solución, que incluye el diseño de la red inalámbrica, para lo cual se considera las características de los equipos, de acuerdo a los requerimientos obtenidos en el capítulo 2. También se incluye el costo referencial de la aplicación en base a las horas requeridas para el diseño de la misma.

Para la red activa, se consideraron equipos de varias marcas, comparando lo que cada uno de ellos ofrece y analizando en función del costo – beneficio.

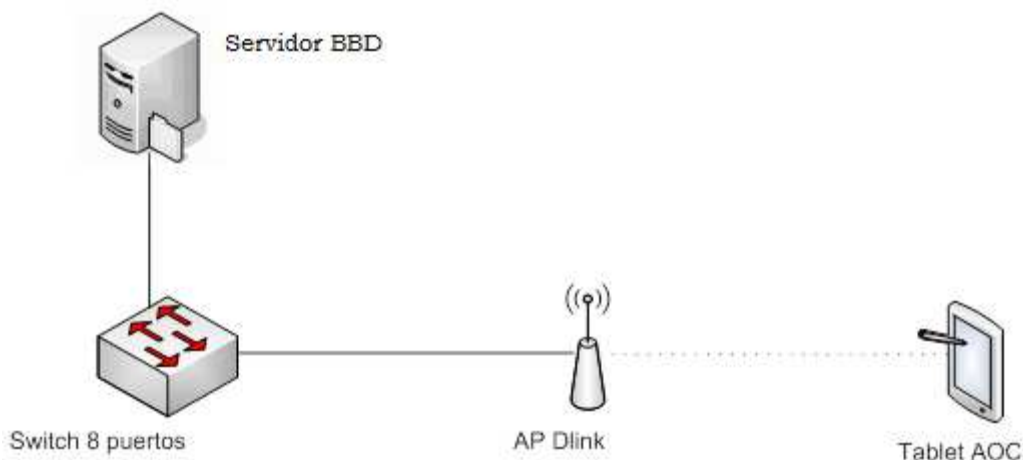
### 4.1 PRESENTACIÓN GENERAL DEL PROTOTIPO DE PRUEBA

Para realizar el prototipo se utilizaron equipos que cumplen con los requisitos mencionados en el capítulo dos, y para la prueba de la aplicación se usó una *tablet* AOC, con el sistema operativo Android 4.1, y un computador cargado con el sistema operativo Windows 7, en donde se encuentra el servidor XAMMP, que incluye MySQL; en donde se correrá la base de datos para el programa, y Apache; con los servicios web respectivos necesarios para la comunicación entre la base de datos y el programa instalado en la *tablet*.

Haciendo un listado de los equipos que se utilizó para el prototipo de prueba se tiene:

- *Un Access Point* Dlink DAP-2360.
- *Un Switch* de 8 puertos.
- Una *tablet* AOC de 7" con Sistema operativo Android 2.3
- Una laptop i5 con Sistema operativo Windows 7.

La figura 4.1 muestra el esquema del prototipo de prueba:



**Figura 4.1:** Esquema del prototipo de prueba

Al armar el prototipo no se tuvo ningún inconveniente; la presentación del programa tal como se requería, con respecto a la aplicación se desarrolló un total de 4 versiones para conseguir el resultado esperado. Los cambios realizados en la aplicación fueron en relación con su interfaz.

## 4.2 ANÁLISIS DE RESULTADOS DE LA APLICACIÓN DESARROLLADA

A continuación se realizarán las pruebas del funcionamiento de la aplicación CCEspiral.apk instalada en la *tablet*, sus resultados se presentan a continuación en las siguientes tablas.

- **Análisis de resultados de la actividad principal “MAIN ACTIVITY”**  
La tabla 4.1 muestra el funcionamiento de la actividad principal, la misma que ejecuta las diferentes actividades de la aplicación.
- **Análisis de resultados de la actividad “LOCALES”, “OFERTAS” y “PRODUCTOS”.**  
Las tablas 4.2, 4.3 Y 4.4 muestran la funcionalidad de las actividades “Locales”, “Ofertas” y “Productos”, que realizan una búsqueda por tipo o categoría.

<b>ACTIVIDADES</b>	<b>OBSERVACIÓN</b>
Ejecuta la aplicación LOCALES	Correcto
Ejecuta la aplicación OFERTAS	Correcto
Ejecuta la aplicación PRODUCTOS	Correcto
Ejecuta la aplicación UBICACIÓN	Correcto
Ejecuta la aplicación MAPA DEL SITIO	Correcto
Ejecuta la aplicación BUZON DE SUGERENCIAS	Correcto

**Tabla 4.1:** Muestra las Actividades que realiza la actividad principal "MAIN"

<b>ACTIVIDADES</b>	<b>OBSERVACIÓN</b>
Búsqueda de locales TIPO = TECNOLOGIA	Correcto
Búsqueda de locales TIPO = CELULARES	Correcto
Búsqueda de locales TIPO = JUEGOS	Correcto
Búsqueda de locales TIPO = JOYERIA	Correcto
Búsqueda de locales TIPO = AUDIO-VIDEO	Correcto
Búsqueda de locales TIPO = ROPA-GALA	Correcto
Búsqueda de locales TIPO = ROPA-GENERAL	Correcto
Búsqueda de locales TIPO = CAMISETAS	Correcto
Búsqueda de locales TIPO = REGALOS	Correcto
Búsqueda de locales TIPO = COMIDA	Correcto
Búsqueda de locales TIPO = CASA MUSICAL	Correcto
Búsqueda de locales TIPO = OTROS	Correcto
Búsqueda de locales TIPO = SERVICIO TECNICO	Correcto

**Tabla 4.2:** Muestra las Actividades que realiza la actividad LOCALES

<b>ACTIVIDADES</b>	<b>OBSERVACIÓN</b>
Búsqueda de ofertas TIPO = TECNOLOGIA	Correcto
Búsqueda de ofertas TIPO = CELULARES	Correcto
Búsqueda de ofertas TIPO = JUEGOS	Correcto
Búsqueda de ofertas TIPO = JOYERIA	Correcto
Búsqueda de ofertas TIPO = AUDIO-VIDEO	Correcto
Búsqueda de ofertas TIPO = ROPA-GALA	Correcto
Búsqueda de ofertas TIPO = ROPA-GENERAL	Correcto
Búsqueda de ofertas TIPO = CAMISETAS	Correcto
Búsqueda de ofertas TIPO = REGALOS	Correcto
Búsqueda de ofertas TIPO = COMIDA	Correcto
Búsqueda de ofertas TIPO = CASA MUSICAL	Correcto
Búsqueda de ofertas TIPO = OTROS	Correcto
Búsqueda de ofertas TIPO = SERVICIO TECNICO	Correcto

**Tabla 4.3:** Muestra las Actividades que realiza la actividad OFERTAS

<b>ACTIVIDADES</b>	<b>OBSERVACIÓN</b>
Búsqueda de productos TIPO = TECNOLOGÍA	Correcto
Búsqueda de productos TIPO = CELULARES	Correcto
Búsqueda de productos TIPO = JUEGOS	Correcto
Búsqueda de productos TIPO = JOYERIA	Correcto
Búsqueda de productos TIPO = AUDIO-VIDEO	Correcto
Búsqueda de productos TIPO = ROPA-GALA	Correcto
Búsqueda de productos TIPO = ROPA-GENERAL	Correcto
Búsqueda de productos TIPO = CAMISETAS	Correcto
Búsqueda de productos TIPO = REGALOS	Correcto
Búsqueda de productos TIPO = COMIDA	Correcto
Búsqueda de productos TIPO = CASA MUSICAL	Correcto
Búsqueda de productos TIPO = OTROS	Correcto
Búsqueda de productos TIPO = SERVICIO TECNICO	Correcto

**Tabla 4.4:** Muestra las Actividades que realiza la actividad PRODUCTOS

- **Análisis de resultados de la actividad “LOCAL DETALLES”**

La tabla 4.5 muestra la funcionalidad de la actividad “Local Detalles”, la misma que en locales del tipo tecnología muestra cuatro subcategorías; “Laptops”, “Tables”, “Impresoras” y “Otros” para facilitar la búsqueda de estos productos en los locales de este tipo, además carga el logotipo del local correspondiente.

ACTIVIDADES	OBSERVACIÓN
Muestra 4 subcategorías para locales del TIPO=TECNOLOGIA	Correcto
Carga imagen general de las categorías de los diferentes Locales	Correcto
Carga del logotipo de cada local	Correcto
Recepción y envío de parámetros a la siguiente Actividad	Correcto
Carga lista de productos de cada local	Correcto

**Tabla 4.5:** Muestra las Actividades que realiza la actividad LOCAL DETALLES

- **Análisis de resultados de la actividad “OFERTAS LOCAL”**

La tabla 4.6 muestra la funcionalidad de la actividad “Ofertas Local”, que contiene la información de ofertas correspondiente a cada local.

ACTIVIDADES	OBSERVACIÓN
Carga información correspondiente a cada local	Correcto
Carga imagen general de las categorías de los diferentes Locales	Correcto
Carga del logotipo de cada local	Correcto
Recepción y envío de parámetros a la siguiente Actividad	Correcto
Carga lista de productos en oferta de cada local	Correcto

**Tabla 4.6:** Muestra las Actividades que realiza la actividad OFERTAS LOCAL

- **Análisis de resultados de la actividad “PRODUCTO DETALLES”**

La tabla 4.7 muestra la funcionalidad de la actividad “Producto Detalles”, la misma que contiene los detalles de cada producto.

ACTIVIDADES	OBSERVACIÓN
Carga información correspondiente al local perteneciente del producto	Correcto
Carga imagen del producto	Correcto
Carga de datos de las características del producto	Correcto
Visualización clara de textos e imágenes	Incorrecto-correcto

**Tabla 4.7:** Muestra las Actividades que realiza la actividad PRODUCTO DETALLES

- **Análisis de resultados de la actividad “Oferta Detalles”**

La tabla 4.8 muestra la funcionalidad de esta actividad que presenta los detalles de un producto en oferta.

ACTIVIDADES	OBSERVACIÓN
Carga información correspondiente al local al que pertenece la oferta	Correcto
Carga imagen del producto	Correcto
Carga de datos de las características del producto	Correcto
Visualización clara de textos e imágenes	Incorrecto-correcto
Visualización del Precio oferta	Correcto

**Tabla 4.8:** Muestra las Actividades que realiza la actividad OFERTA DETALLES

- **Análisis de resultados de la actividades correspondientes a cada piso**

Los diferentes niveles correspondientes a cada piso del centro comercial, se encuentran ubicados en la aplicación a manera de mapa del sitio, para acceder a través de las etiquetas numeradas a cada local, la tabla 4.9 muestra en funcionamiento de estas actividades.



ACTIVIDADES	OBSERVACIÓN
Muestra información clara de la ubicación de un local	Correcto
Cada marcador con el respectivo número del local es Legible	Correcto
Carga del nombre del local	Correcto
El botón “detalles” se ejecuta sin problemas	Correcto
Paso de parámetros a la siguiente actividad	Correcto

**Tabla 4.9:** Muestra las Actividades que realiza la actividad de acuerdo al piso

- **Análisis de resultados de las actividades que conforman el mapa del sitio del centro comercial**

Los resultados obtenidos en la prueba anterior se realizaron de igual manera para cada una de las actividades del mapa del sitio del Centro Comercial.

La tabla 4.10 muestra los resultados:

ACTIVIDADES	OBSERVACIÓN
SUBSUELO DOS ACTIVITY	Correcto
SUBSUELO UNO ACTIVITY	Correcto
PLANTA BAJA	Correcto
NIVEL UNO	Correcto
NIVEL DOS	Correcto
NIVEL TRES	Correcto
NIVEL CUATRO	Correcto
NIVEL CINCO	Correcto
NIVEL SEIS	Correcto

**Tabla 4.10:** Muestra los resultados de las diferentes actividades del Mapa del Sitio por pisos

- **Análisis de resultados de la actividad “Buzón de Sugerencias”**

ACTIVIDADES	OBSERVACIÓN
Carga los “ <i>EditText</i> ” para el ingreso de la información requerida	Correcto
Envía y Guarda las sugerencias en la base de datos	Correcto
Interfaz y manejo de botones entendible	Correcto

**Tabla 4.11:** Muestra las Actividades que realiza la actividad BUZÓN DE SUGERENCIAS

### 4.3 COSTO DE LA RED PASIVA

Para definir el número de elementos de la red pasiva se utilizó los planos del Centro Comercial Espiral, sabiendo que para tender el cable desde el Punto de Acceso hasta el Cuarto de telecomunicaciones se utilizará canaletas.

El costo de los elementos fueron proporcionados por la empresa *Redinco* en la marca *Panduit* y *Dexon*.

La tabla 4.12 presenta el costo de la red pasiva del Centro Comercial Espiral.

Elementos de Cableado estructurado			
Material	Cantidad	V. Unitario (\$)	V. Total (\$)
Rollo de cable UTP cat 6a	2	198,25	396,50
Canaleta 13x7 (3m largo)	83	1,50	124,50
FacePlates simples	22	1,55	34,10
Jackscat 6a	22	6,50	143,00
Patchcords 1.5m cat 6	22	5	110,00
Rack 6UR	1	70	70,00
Patch Panel cat6 24 puertos	1	24,90	24,90
Organizadores	1	13,50	13,50
Paneles de energía	1	53	53,00
		<b>Subtotal</b>	969,50
		<b>IVA 12%</b>	116,34
		<b>Total</b>	1085,84

**Tabla 4.12:** Costo de la red pasiva

En la tabla 4.13 se muestra el costo que tendrá la instalación y certificación de los puntos de red, en donde se colocarán los puntos de acceso. Valores que se obtuvieron de la empresa Solinper.

<b>Instalación y Certificación de Puntos</b>			
<b>Ítem</b>	<b>Cantidad</b>	<b>V. Unit</b>	<b>V. Total</b>
<b>Instalación de Puntos</b>	22	80,00	1760,00
<b>Pruebas de Certificación</b>	22	25,00	550,00
		<b>Subtotal</b>	2310,00
		<b>IVA 12%</b>	277,20
		<b>Total</b>	2587,20

**Tabla 4.13:** Costo de la instalación y certificación de puntos

#### **4.4 COSTO DE LA RED ACTIVA**

Para un óptimo funcionamiento de la red activa de Centro Comercial los equipos de conectividad deben cumplir con los requerimientos señalados en el capítulo 3. Además se ha considerado dos alternativas en marcas que cumplen con nuestros requerimientos y expectativas como son Cisco y HP, por lo que para la selección de los equipos se tomará en cuenta las características técnicas, confiabilidad y precios de los equipos.

En el capítulo 3 se analizó las características necesarias que cada marca debe cumplir, por lo que detallamos a continuación los costos de cada una de ellas.

##### **4.4.1 ALTERNATIVA HP**

A continuación la tabla 4.14 detalla los precios de los equipos HP que cumplen con los requerimientos para la implementación de la red inalámbrica.

<b>Equipo</b>	<b>Cantidad</b>	<b>Modelo</b>	<b>V. Unitario \$</b>	<b>V. Total \$</b>
<b><i>Acess Point</i></b>	20	HP MSM-430	565	11300,00
<b><i>Switch de distribución</i></b>	1	HP V1910 24G	900	900,00
<b><i>Wireless LAN Controller</i></b>	1	HP MSM 720 Access Controller	1754	1754,00
			<b>SUBTOTAL</b>	<b>15708,00</b>
			<b>IVA 12%</b>	<b>1884,96</b>
			<b>TOTAL</b>	<b>17592,96</b>

**Tabla 4.14:** Costo de la alternativa HP

#### 4.4.2 ALTERNATIVA CISCO

A continuación la tabla 4.15 detalla el precio que tendría los equipos de conectividad en la marca Cisco.

<b>Equipo</b>	<b>Cantidad</b>	<b>Modelo</b>	<b>V. Unitario \$</b>	<b>V. Total \$</b>
<b><i>Acess Point</i></b>	20	Cisco Aironet 1130	900	18000
<b><i>Switch de distribución</i></b>	1	CISCO Catalyst 2960	2500	2500
<b><i>Wireless LAN Controller</i></b>	1	CISCO WLC 4400	4300	4300
			<b>SUBTOTAL</b>	<b>24800,00</b>
			<b>IVA 12%</b>	<b>2976,00</b>
			<b>TOTAL</b>	<b>27776,00</b>

**Tabla 4.15:** Costo alternativa Cisco

#### 4.4.3 SELECCIÓN DE LOS EQUIPOS DE CONECTIVIDAD

Para la selección de los equipos de conectividad de las dos marcas propuestas se comparó las características de los equipos que cumplen una misma función, tal como se detallan en las tablas anteriores.

Como se puede observar todos los equipos de conectividad de las dos marcas, tanto HP como Cisco, cumplen con las especificaciones requeridas, por lo que para la selección de una de estas se tomó en cuenta las referencias en Costo total, y confiabilidad de los equipos.

Por lo que se decide utilizar los equipos HP, debido a la disponibilidad en el mercado local, y a precio de la solución.

#### 4.4.4 SELECCIÓN DEL SERVIDOR PARA LA BASE DE DATOS

En la tabla 4.16 se muestra la comparación de precios entre los dos servidores analizados en el capítulo 2.

<b>SERVIDOR PARA LA BASE DE DATOS</b>		
	<b>HP Proliant ML 115 G5</b>	<b>Dell Poweredge T110</b>
<b>Precio Sin IVA</b>	2235,00	1409,00
<b>IVA</b>	268,20	169,08
<b>Total</b>	2503,20	1578,08

**Tabla 4.16:** Servidores para la Base de Datos

Ambos servidores cumplen con los requerimientos mínimos para el servidor de la base de datos; por disponibilidad en el mercado, se elige el servidor HP Proliant ML 115 G5.

#### 4.5 COSTOS DE ENLACE DE INTERNET

La capacidad que requiere cubrir el Centro Comercial Espiral para la salida a Internet se calculó en el capítulo 2, donde se obtuvo que la capacidad en ancho de banda a contratar es de 31 Mbps.

Cabe destacar que el contrato con los proveedores de servicio incluyen los equipos de conexión con el mundo, y la instalación de los mismos en el lugar donde irán los equipos.

A continuación la tabla 4.17 muestra el detalle de costos que tendría el enlace a Internet por una capacidad de 31 Mbps, datos obtenidos a partir de cotizaciones a cada proveedor.

Proveedor	Capacidad	Costo mensual (Sin IVA)	Costo instalación (Sin IVA)
<b>New Access S.A.</b>	31 Mbps 1/1	\$ 1859,00	350
<b>Puntonet S.A.</b>	31 Mbps 1/1	\$ 1950,00	300

**Tabla 4.17:** Costo del enlace a Internet

Tomando en cuenta el servicio recibido por ambos proveedores, y el costo mensual que tendría la conexión a internet, se decide elegir la oferta de *New Access S.A.*

#### 4.6 COSTO DE LA APLICACIÓN

Para el cálculo del costo que tiene la implementación de la aplicación se tomó en cuenta las horas requeridas para el diseño, desarrollo y depuración de errores; no se incluyen costos de licencias ya que se utilizaron herramientas de desarrollo libres.

Los valores referenciales de la tabla 4.18 utilizados<sup>25</sup> fueron los siguientes:

- Salario para el personal de diseño es de \$ 6 por hora trabajada.
- Salario para el personal de desarrollo y depuración de la aplicación es de \$ 8 por cada hora laborada.

<sup>25</sup>Tomados de los salarios de los desarrolladores de la Empresa Interbases S.A. Empresa dedicada al desarrollo de Software

<b>Costo de implementación de la aplicación</b>			
<b>Actividad</b>	<b>Horas</b>	<b>Costo por Hora (\$)</b>	<b>Total (\$)</b>
<b>Diseño</b>	80	6	\$ 480
<b>Desarrollo</b>	100	8	\$ 800
<b>Depuración</b>	60	8	\$ 480
		<b>SUBTOTAL</b>	<b>\$ 1760,00</b>
		<b>IVA 12%</b>	211,20
		<b>TOTAL</b>	<b>\$ 1971,20</b>

**Tabla 4.18:** Costo de implementación de la aplicación

#### 4.7 COSTO TOTAL DEL PROYECTO

Una vez elegido los equipos a utilizar, el proveedor de servicios de Internet, y el Costo de la aplicación se procede a calcular qué valor tiene la implementación de la solución.

A continuación la tabla 4.19 muestra el costo total del proyecto:

<b>Item</b>	<b>Costo (\$)</b>
<b>Cableado Estructurado</b>	969,50
<b>Instalación y certificación de Puntos</b>	2310,00
<b>Equipos Activos de la red</b>	15708,00
<b>Servidor de Base de datos</b>	2235,00
<b>Enlace de Internet (Mensual)</b>	1859,00
<b>Instalación de Internet</b>	350,00
<b>Diseño, desarrollo y depuración de la aplicación</b>	1760,00
	<b>SUBTOTAL</b>
	<b>25.191,50</b>
	<b>IVA</b>
	<b>3022,98</b>
	<b>TOTAL</b>
	<b>28.214,48</b>

**Tabla 4.19:** Costo total del proyecto

# CAPÍTULO 5



## 5. CONCLUSIONES Y RECOMENDACIONES

### 5.1 CONCLUSIONES

- El manejo de información a través de dispositivos móviles se ha convertido en una tendencia tecnológica, por su facilidad de uso, su costo, su fácil conexión en cualquier establecimiento que disponga de una red *WiFi*; pudiendo así distribuir información de productos y ofertas de cada local del Centro Comercial Espiral.
- El diseño de la red inalámbrica en el Centro Comercial Espiral fue realizado en función de los análisis de requerimientos del mismo, teniendo como usuarios a visitantes y dueños de locales, incluyendo el dimensionamiento de los equipos escogidos para el diseño, asegurando el perfecto funcionamiento durante su horario habitual.
- Mediante el diseño de la red inalámbrica y de la aplicación para la localización de productos y ofertas se desea impulsar la compra y facilitar la localización de cualquier artículo ofrecido por los diferentes locales en el establecimiento.
- El flujo de tráfico que cursará por la red inalámbrica será variado, entre aplicaciones propias del centro comercial, así como navegación *web*, correo electrónico, youtube, redes sociales y *streaming*; el cual fue analizado para obtener un ancho de banda que asegure la disponibilidad de la red.
- El software *Air Magnet Survey* es una herramienta de gran utilidad ya que permite analizar el nivel de interferencia en todo el edificio, pudiendo escoger los mejores sitios para la colocación del punto de acceso en el diseño de la red inalámbrica, evitando al máximo la interferencia entre dispositivos cercanos e incluso los colocados para este diseño.

- El entorno de desarrollo integrado de java Eclipse, permite adaptar el SDK (kit de herramientas de desarrollo de *android*) a través del *plugin (Android Developed Tools)*, generando de esta manera el entorno más adecuado para el desarrollo de aplicaciones en android.
- El envío y recepción de información correspondiente a consultas hacia la base de datos, se facilita mediante el uso de servicios *web* y lectura de datos en formato *JSON*; permitiendo a los dispositivos móviles reducir notablemente sus tiempos de procesamiento así como también el uso de la red.
- Las bases de datos en MySQL son las más utilizadas en el desarrollo de aplicaciones móviles principalmente por la compatibilidad con los diferentes lenguajes de programación pero principalmente con el intérprete de servicios *web* PHP (Lenguaje de hipertexto preprocesado).
- El manejo de hilos en segundo plano en las diferentes actividades de la aplicación permite evitar posibles errores principalmente en el momento de la ejecución del programa.
- La version 2 de los mapas de google permiten mayor versatilidad para el manejo de geolocalización y posicionamiento en android debido a que incorpora nuevas librerías como *MapFragment* para la utilización de mapas vectoriales garantizando una mayor velocidad de carga.
- En el desarrollo de la aplicación se utilizó los *ImageButtons* para el manejo de imágenes puesto que con otros elementos existen errores de posición al momento de ejecutar la aplicación.

## 5.2 RECOMENDACIONES

- En caso de que el proyecto sea implementado, se debe capacitar al administrador de la red inalámbrica para que sea capaz de realizar cualquier mantenimiento y soporte a los usuarios de la misma.
- Realizar un estudio de factibilidad para el diseño de la red cableada del centro comercial Espiral, complementando la red inalámbrica, y evitando problemas de interferencia entre puntos de acceso propios del edificio y los pertenecientes a la red diseñada.
- Todos los equipos que se utilicen en la implementación de la red diseñada deben manejarse bajo la misma marca escogida; es decir, los puntos de acceso, *switches*, el controlador de red inalámbrico deben ser del mismo fabricante, evitando problemas de performance, o cuellos de botella entre el servidor de datos y las estaciones móviles conectadas.
- Este proyecto se puede complementar con la adición de un servidor web, para manejar la página del centro comercial espiral con mayor control e interacción entre compradores y vendedores, y un servidor de archivos, para el manejo de documentos de los dueños de los locales comerciales.
- Se debe establecer un proceso de conexión a la red inalámbrica, creando listas de acceso, reglas de control *web* en función del tipo de usuarios conectados en la red, definiendo el tiempo de conexión para cada usuario, y definiendo un servidor de respaldo.
- Crear manuales de configuración de cada elemento de la red inalámbrica, tales como del punto de acceso, del *switch*, del controlador; para no depender solamente del administrador de la red, en un caso de renuncia del mismo.

- Es necesario disponer de puntos de acceso de *Backup* configurados y listos para su conexión, para evitar caídas e intermitencias en todo el edificio del centro comercial.
- Se debe optimizar el código para dispositivos de menor tamaño, resolución y capacidad; es decir adecuar a la interfaz para garantizar el correcto funcionamiento de la aplicación.
- Se recomienda verificar que el *Android Manifest.xml* (archivo que maneja las interfaces de una aplicación en Android), contenga los permisos necesarios así como la declaración respectiva de cada actividad caso contrario ninguna aplicación en Android se ejecutará correctamente.
- Es necesario utilizar un dispositivo real para el caso de simular actividades que contengan mapas puesto que estos dependen directamente del *Google Play Services* para mostrarlos.
- Considerar que *google maps* permite utilizar sus mapas para aplicaciones siempre y cuando no sobrepasen de las 15000 visitas, caso contrario se deberá adquirir una licencia de utilización.

## REFERENCIAS

### Libros y Documentos

- [1] **ROLDAN, David.** Comunicaciones en Redes WLAN: EME-Madrid, 2007.
- [3] **IEEE-Standar.** 802.11 PHY Layers-Chapter:8. s.l. : IEE, 8.
- [4] **HARA, Bob.** IEEE 802.11 Handbook: A Designer' s Companion: IEEE Press, 2004
- [5] **Wireless LAN Medium Access Control(MAC) and Physical Layer specifications,2012.**
- [6] **LEHEMBRE, Guillaume.** *Seguridad Wifi-WEP, WPA, WPA2; 2008.*
- [7] **NEIL, Reid.** Manual de Redes Inalámbricas: Mc Graw Hill, México 2004
- [10] **MARTINEZ, Jorge.** Fundamentos de Programación en Java. Madrid : EME-Madrid, 2008.
- [11] **SMITH, Dave.** *Android Recipes A Problem Solution.* Estados Unidos : Apress, 2012.
- [14] **ESPIRAL, ADMINISTRACIÓN CENTRO COMERCIAL.** Locales del Centro Comercial. Quito : s.n., 2012.
- [18] **PONCE, Diego.** PLANOS CENTRO COMERCIAL ESPIRAL. Quito : s.n., 1980.
- [19] **GOMEZ, Joaquin.** Redes Locales: EDITEX-Madrid, 2006
- [20] **System, CISCO.** CISCO Aironet Data-Sheet. 2012.
- [21] **D-LINK.** DLINK DAP 2360 Data Sheet. 2012.
- [22] **HP MSM-802.11n Dual Radio Access Point Series Data Sheet.pdf.** 2011.
- [23] **HP v1910 24G Data Sheet.pdf.** 2012.
- [24] **CISCO Catalyst 2900 series.** CISCO Switch 2960 2900 series Data Sheet.pdf . 2012.
- [25] **HP MSM 720 Data Sheet.pdf.** 2012.
- [26] **CISCO Series.** CISCO WLC 4402 Lan Controller Data Sheet.pdf . 2012.
- [27] **AYALA, Christian.** Guía metodológica para el mejoramiento, gestión y seguridad en redes informáticas de centros educativos en base a soluciones abiertas, el modelo de gestión internet y RFC 2196.pdf, Quito-2012
- [28] **DELL POWEREDGE T110 SOFTSYSTEM Data Sheet. pdf .2012**

[29] **SERVIDORES HP PROLIANT**, Guia de portafolio.pdf. 2012

[30] **JENDROCK, Eric**. The Java EE 7 Tutorial. Oracle 2012

[31] **GARZARAN, Domingo**. *Aplicación para Android:Agenda Comercial Detalles de Clientes*. Madrid : etsinf, 2010

## Documentos y Páginas en línea

[2] **RUIZ, Larry**. Componentes y Topologías Inalámbricas.

**Disponible el:** 6 de Noviembre de 2013.

<http://www.slideshare.net/LarryRuiz/componenes-de-una-red-wireless>.

[8] **RODRIGUEZ, Alex**. Aprender Java desde Cero.

**Disponible el:** 13 de Noviembre de 2013.

[http://www.aprenderaprogramar.com/index.php?option=com\\_content&view=article&id=392:la-maquina-virtual-java-jvm-o-java-virtual-machine-  
compilador-e-interpretador-bytecode-cu00611b&catid=68:curso-aprender-  
programacion-java-desde-cero&Itemid=188](http://www.aprenderaprogramar.com/index.php?option=com_content&view=article&id=392:la-maquina-virtual-java-jvm-o-java-virtual-machine-compilador-e-interpretador-bytecode-cu00611b&catid=68:curso-aprender-programacion-java-desde-cero&Itemid=188).

[9] **ORACLE Java**. Java SE 6 Documentation.

**Disponible el:** 6 de Enero de 2012.

<http://docs.oracle.com/javase/6/docs/index.html>.

[12] **CATALAN, Adrian**. Desarrollo de Aplicaciones Moviles.pdf.

**Disponible el:** 7 de Noviembre de 2013

<http://www.maestrosdelweb.com/editorial/curso-android>.

[13] **BAEZ, Emanuel**. Introducción a Android.pdf.

**Disponible el:** 13 de Noviembre de 2012

<http://pendientedemigracion.ucm.es/info/tecnomovil/documentos/android.pdf>.

[15] **RODRIGUEZ, Ernesto**. Muestra y Muestreo.

**Disponible el:** 5 de Febrero de 2012.

[http://www.uaeh.edu.mx/docencia/P\\_Presentaciones/tizayuca/gestion\\_tecnologia/muestraMuestreo.pdf](http://www.uaeh.edu.mx/docencia/P_Presentaciones/tizayuca/gestion_tecnologia/muestraMuestreo.pdf)

[16] **KING, Andrew**. WebSiteOptimization. SITE.OPT.

**Disponible el:** 8 de Diciembre 2013

<http://www.websiteoptimization.com/services/analyze/>.

[17] **TOBER, Marcus.** Rapit Seachmetris.

**Disponible el:** 8 de Diciembre 2013

<http://rapid.searchmetrics.com/#3575>

[32] **TOMÁS, J; Pineda , M; Bataller, J.** Android Fundamentos-Seguridad y Posicionamiento

**Disponible el:** 3 de Mayo 2012

<http://www.androidcurso.com/index.php/recursos-didacticos/tutoriales-android/41-unidad-7-seguridad-y-posicionamiento/223-google-maps-api-v2>

# ANEXO A



## ANEXO A

### MANUAL DE USUARIO

“CenEspiral” es una aplicación en *Android* desarrollada para una *tablet* con la finalidad de mostrar: productos, ofertas, locales correspondientes al Centro Comercial Espiral, permitiendo al usuario en general una búsqueda mucho más fácil.

Su uso es intuitivo y la ventaja principal es la versatilidad con la que muestra la información requerida.

#### INSTALACIÓN:

- **Seleccionar la opción permitir la instalación de aplicaciones desconocidas en el menú configuración de la *tablet***



**Figura A.1:** Menú Configuración en *Android*

- **Copiar el instalador de la aplicación “CenEspiral.apk”, al directorio de archivos de la *tablet* o a la tarjeta de memoria de la misma**

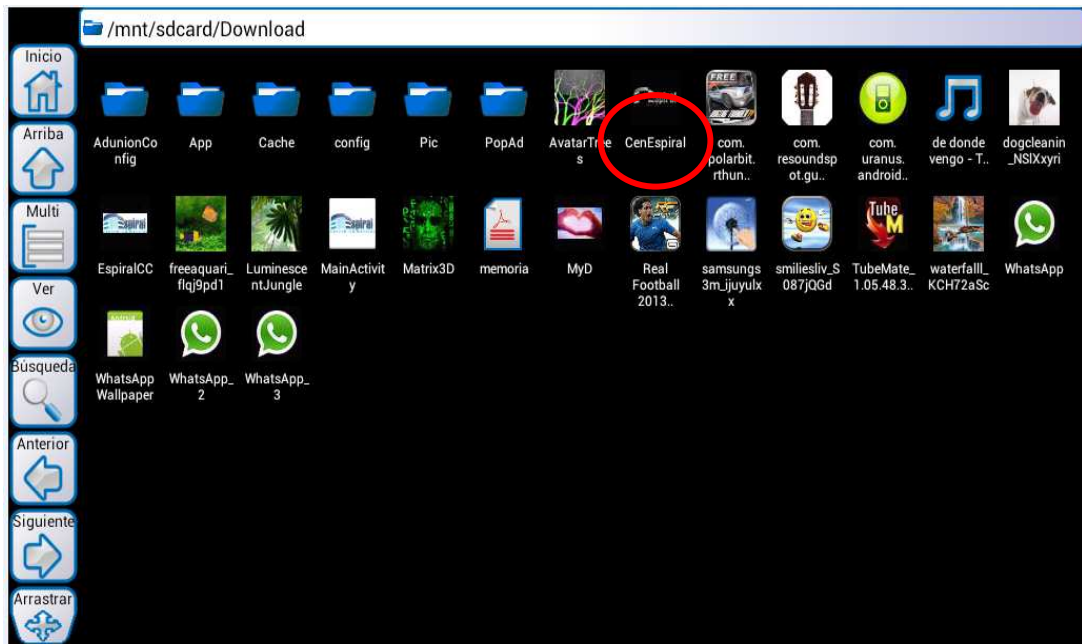


Figura A.2: Instalador de la aplicación

- Abrir el instalador “CenEspiral.apk” y seleccionar instalar

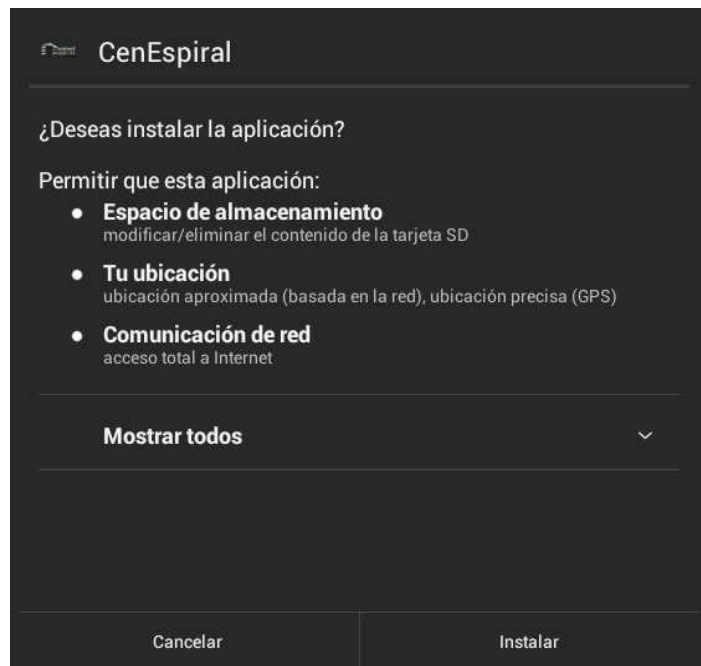


Figura A.3: Instalador “CenEspiral.apk”

## VISIÓN GENERAL Y FUNCIONALIDAD DE LA APLICACIÓN:

## 1. PANTALLA PRINCIPAL

La pantalla principal de la figura A.3 que se muestra a continuación desplegará las diferentes actividades a continuación su descripción:



**Figura A.4:** Pantalla Principal

-  **Locales** ★ Abre la interfaz Búsqueda de Locales
-  **Productos** ★ Abre la interfaz Búsqueda de Productos
-  **Ofertas** ★ Abre la interfaz Búsqueda de Ofertas
-  **Galería Locales** ★ Abre la interfaz Galería de Locales
-  **Ubicación** ★ Abre la interfaz Ubicación
-  **Sugerencias** ★ Abre la interfaz Buzón de Sugerencias
-  **Salir** ★ Cierra la aplicación

## 2. BÚSQUEDA LOCALES, PRODUCTOS Y OFERTAS

Las interfaces Búsqueda Locales, Productos y Ofertas como lo muestran las figuras A.5, A.6, A.7, muestran en lista la información correspondiente de acuerdo con los botones de la categoría seleccionada.

Muestra la lista de los locales de acuerdo con la categoría seleccionada



Lista de Locales




**Figura A.5:** Interfaz Búsqueda Locales



Figura A.6: Interfaz Búsqueda Productos



Figura A.7: Interfaz Búsqueda Ofertas

-  **Menú** ★ Regresa a la Interfaz Principal
-  **Salir** ★ Cierra toda la Aplicación
-  **ver más** Muestra información detallada de locales, productos y ofertas

### 3- GALERIA DE LOCALES

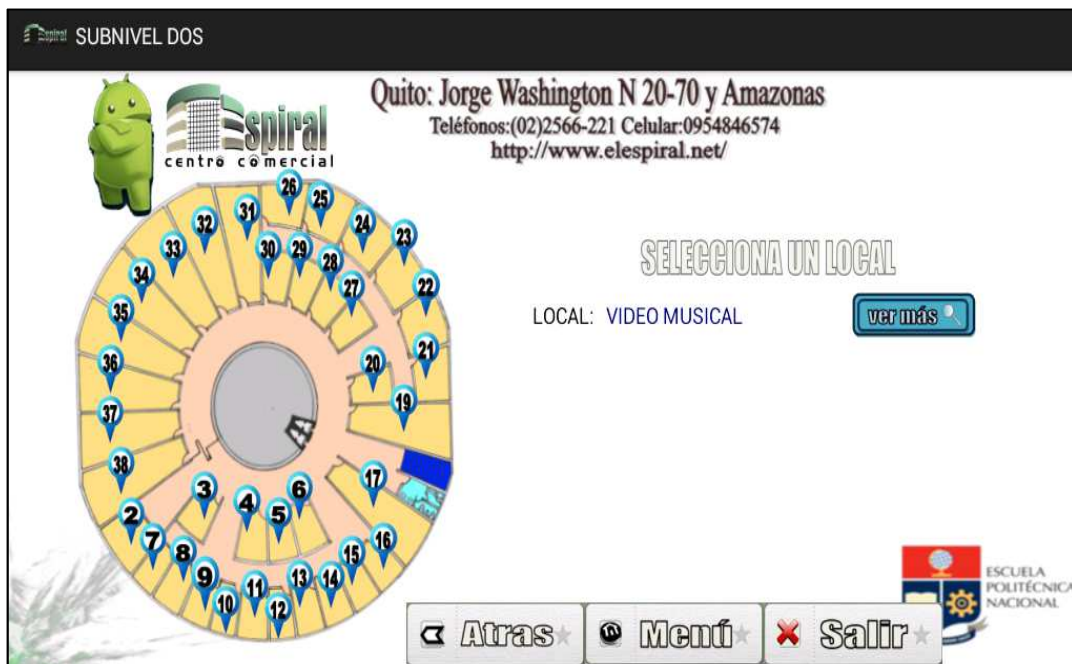


**Figura A.8:** Interfaz Galería de Locales




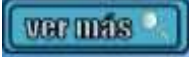
- SUBNIVEL DOS ★ Abre la interfaz Subnivel dos
- SUBNIVEL UNO ★ Abre la interfaz Subnivel Uno
- PLANTA BAJA ★ Abre la interfaz Planta Baja
- NIVEL UNO ★ Abre la interfaz Nivel Uno
- NIVEL DOS ★ Abre la interfaz Nivel Dos
- NIVEL TRES ★ Abre la interfaz Nivel Tres
- NIVEL CUATRO ★ Abre la interfaz Nivel Cuatro
- NIVEL CINCO ★ Abre la interfaz Nivel Cinco
- NIVEL SEIS ★ Abre la interfaz Nivel Seis



### 3.1 GALERIA DE LOCALES INTERFAZ NIVELES



**Figura A.9:** Modelo Interfaz Niveles

-  Etiqueta tipo de un local muestra información correspondiente al mismo.
-  **Menú** Regresa a la Interfaz Principal
-  **Salir** Cierra toda la Aplicación
-  **ver más** Muestra información detallada del local seleccionado.

### 3.2 INTERFAZ LOCAL







Figura A.10: Interfaz local tipo en general



Figura A.11: Interfaz local tipo "TECNOLOGIA"



-  Despliega la interfaz Ofertas correspondiente al local
-  Regresa a la pantalla principal
-  Regresa hacia la interfaz anterior
-  Cierra la aplicación

Para la interfaz correspondiente a un local del tipo Tecnología:

-  Muestra una lista de *laptops* que dispone el local
-  Muestra una lista de *tablets* que dispone el local
-  Muestra una lista de impresoras que dispone el local
-  Muestra una lista de otro tipo de artículos que dispone el local

#### 4. INFORMACIÓN OFERTAS

Al seleccionar el botón “Ver más” en la pantalla “Búsqueda Ofertas” se despliega la siguiente interfaz la cual contiene la información correspondiente a un producto en específico

**INFORMACION OFERTAS**

 **COBO IMPORTACIONES**

ENCUENTRANOS EN: NIVEL 3  
 LOCAL NÚMERO: 113  
 TELÉFONO: (02)2222 - 935

RECUERDA CONTACTANOS EN: email:coboimportaciones@elespiral.net

**PRODUCTO:** LAPTOP  
**MODELO:** HP Pavilion g4-1388la  
**PRECIO:** 433,98 \$  
**Disco:** 640 GB HDD  
**Procesador:** Intel Core i5-480M a 2,66GHz  
**Memoria:** DDR3 de 4GB RAM  
**Camara:** Web Cam 101  
**Sistema Operativo:** Windows 7 Home Basic  
**Pantalla:** Led HD 14 pulgadas  
**Tarjeta de red:** WLAN 802.11 a/b/g/n y Bluetooth  
**Garantía:** 12 MESES

**Precio Oferta:** 670 \$

**VALIDO HASTA:** 2013-02-13  
**APRESURATE:** 3 articulos disponibles



**Figura A.12:** Información Ofertas

## 5. INFORMACIÓN PRODUCTOS

Al seleccionar el botón “Ver más” en la pantalla “Búsqueda Productos” se despliega la siguiente interfaz la cual contiene la información correspondiente a un producto en específico. De igual forma se puede acceder al detalle de un producto desde el botón “Ver más” ubicado en la lista de productos de un local

**INFORMACION PRODUCTO**

 **UNITECH**

ENCUENTRANOS EN: PLANTA BAJA  
 LOCAL NÚMERO: 67  
 TELÉFONO: (02)2907 - 046




RECUERDA CONTACTANOS EN: <http://www.elespiral.net/unitech.html>

**PRODUCTO:** TABLET  
**MODELO:** TITAN 7010  
**PRECIO:** 144,90 \$  
**Capacidad:** 8 GB  
**Procesador:** Dual Core ARM Cotex  
**Memoria:** DDR3 1GB  
**Camara:** Frontal(0.3MP)-Posterior(2MP)  
**Sistema Operativo:** Android 4.0.3  
**Pantalla:** LCD tactil 7"  
**Tarjeta de red:** Wifi 802.11g  
**Stock:** 15 disponibles  
**Garantía:** 12 MESES



**Figura A.13:** Información Producto

-  Regresa a la pantalla principal
-  Regresa hacia la interfaz anterior
-  Cierra la aplicación

## 6. BUZÓN DE SUGERENCIAS

Permite el envío de sugerencias para el centro comercial hacia la base de datos



BUZON DE SUGERENCIAS


 Quito: Jorge Washington N 20-70 y Amazonas  
 Teléfonos: (02)2566-221 Celular: 0954846574  
<http://www.elespiral.net/>

NOMBRE:

E-mail:




Sugerencia:







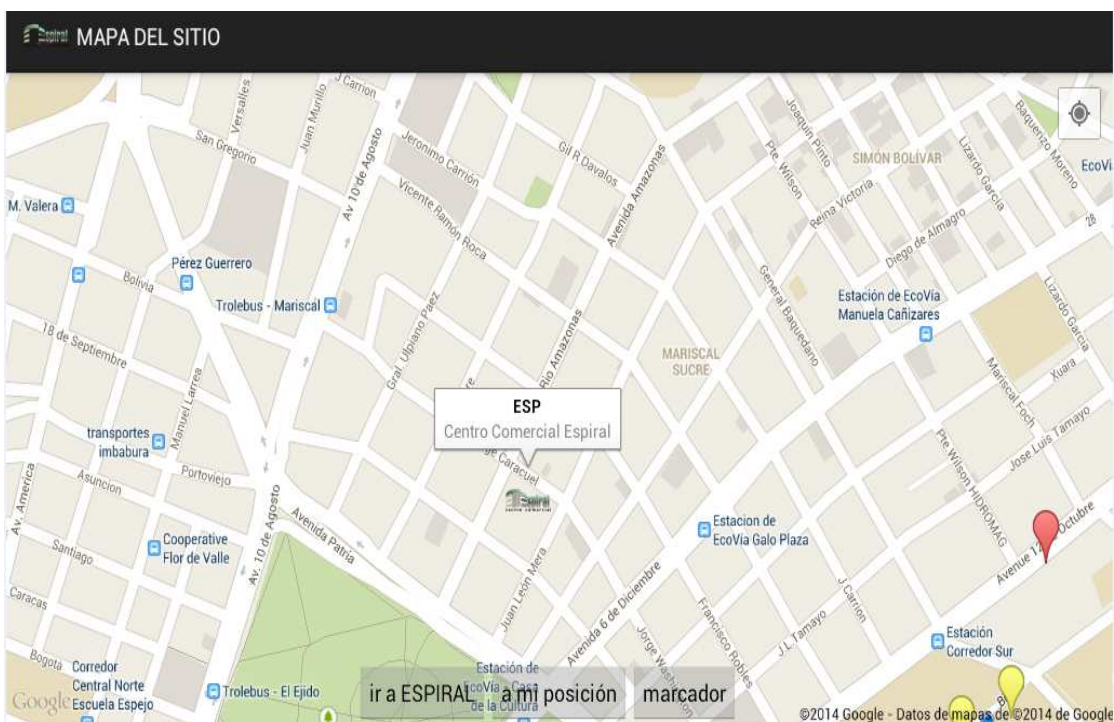

**Figura A.14:** Buzón de sugerencias

-  Regresa a la pantalla principal
-  Regresa hacia la interfaz anterior
-  Cierra la aplicación

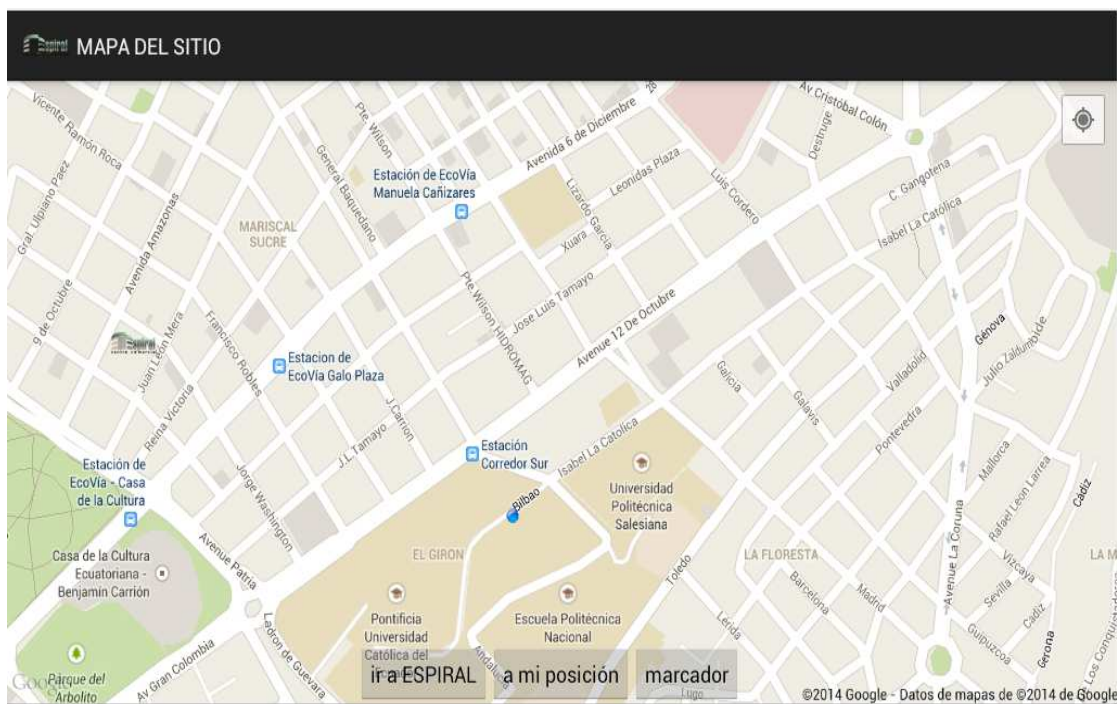
- 
 Envía la sugerencia

## 7. UBICACIÓN


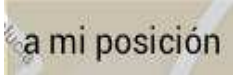
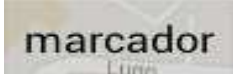
Muestra en el mapa de *google* la ubicación del Centro Comercial, y la ubicación del usuario en ese momento.



**Figura A.15:** Ubicación del Centro Comercial Espiral



**Figura A.16:** Ubicación del Usuario

-  Redirige la cámara del mapa hacia la posición del Centro Comercial
-  Redirige la cámara hacia la ubicación del usuario
-  Añade marcadores en puntos seleccionados del Mapa

# ANEXO B

# ANEXO B

## CÓDIGO

### B.1 CLASE PRINCIPAL

```
/**
 *          ESCUELA POLITÉCNICA NACIONAL
 *
 * CODIGO DE LA APLICACIÓN "CenEspiral"
 *
 * Autores: Darwin Mena Alvarado
 *          Roberto Jacho bravo
 *          Reserv@dos todos los derechos
 * */

package com.example.cenespiral;

import android.os.Bundle;
public class MainActivity extends Activity {
    ImageButton btnLocales;
    ImageButton btnOfertas;
    ImageButton btnProductos,btnSitio,btnGaleria,btnBuzon;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        // Función onCreate generada por defecto, donde se almacenan las
        // instancias de la Actividad Principal Main.
        super.onCreate(savedInstanceState);
        //Llama a la carpeta "Layout" que contiene la interfaz
        Activity_Main.xml a utilizarse
        setContentView(R.layout.activity_main);

        // Declaración de los Image Button a utilizarse
        btnLocales= (ImageButton) findViewById(R.id.btnLocales);
        btnOfertas = (ImageButton) findViewById(R.id.btnOfertas);
        btnProductos=(ImageButton) findViewById(R.id.btnProductos);
        btnSitio=(ImageButton) findViewById(R.id.btnMapa);
        btnGaleria=(ImageButton) findViewById(R.id.btnGaleria);
        btnBuzon=(ImageButton) findViewById(R.id.btnBuzon);
    }
}
```



```
// Evento clic del botón Locales
btnLocales.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        // Lanza la actividad BusquedaLocales
        Intent i = new Intent(getApplicationContext(),
            BusquedaLocalesActivity.class);
        startActivity(i);
        MainActivity.this.finish();
    }
});

// Evento clic del botón Productos
btnProductos.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View view) {
        // Lanza la actividad BusquedaProductos
        Intent i = new Intent(getApplicationContext(),
            BusquedaProductosActivity.class);
        startActivity(i);
        MainActivity.this.finish();
    }
});

// Evento clic del botón Ofertas
btnOfertas.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        // Lanza la actividad BusquedaOfertas
        Intent i = new Intent(getApplicationContext(),
            BusquedaOfertasActivity.class);
        startActivity(i);
        MainActivity.this.finish();
    }
});

// Evento clic del botón Galeria

btnGaleria.setOnClickListener(new View.OnClickListener() {

    @Override
```



```

    public void onClick(View view) {
        // Lanza la Actividad Local Sitio
        Intent i = new Intent(getApplicationContext(),
            LocalSitioActivity.class);
        startActivity(i);
        MainActivity.this.finish();
    }
});

// Evento clic del botón Buzon

btnBuzon.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View view) {
        // Lanza la Actividad Buzon
        Intent i = new Intent(getApplicationContext(), BuzonActivity.class);
        startActivity(i);
        MainActivity.this.finish();
    }
});

// Evento clic del botón Ubicación
btnSitio.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View view) {
        // Lanza la actividad Mapa
        Intent i = new Intent(getApplicationContext(), MapaActivity.class);
        startActivity(i);
        MainActivity.this.finish();
    }
});
}

//Boton Salir
public void ActividadSalir(View v){
    System.exit(0);

}

public void onFinish(){
    MainActivity.this.finish();
}
}

```

```
}
```

## B.2 CLASE BÚSQUEDA LOCALES

```
import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import org.apache.http.NameValuePair;
import org.apache.http.message.BasicNameValuePair;
import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;
import com.example.cenespiral.MainActivity;
import android.app.ListActivity;
import android.app.ProgressDialog;
import android.content.Intent;
import android.os.AsyncTask;
import android.os.Bundle;
import android.util.Log;
import android.view.MotionEvent;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.AdapterView;
import android.widget.ImageButton;
import android.widget.ListAdapter;
import android.widget.ListView;
import android.widget.SimpleAdapter;
import android.widget.TextView;
import android.widget.AdapterView.OnItemClickListener;

public class BusquedaLocalesActivity extends ListActivity implements OnClickListener {
    // Declaración del Progress Dialog a utilizarse
    private ProgressDialog pDialog;
    // Declaración de los ImageButton a utilizarse
    ImageButton ccespiral, tecnologia, celulares,juegos,joyeria,audio;
    //Variables tipo String
    String pid="";
    //Direccion IP del servicio web que contiene la información
    private static String ip = "192.168.0.105";

    // Creando el objeto JSON Parser
```

```

JSONParser jParser = new JSONParser();

ArrayList<HashMap<String, String>> localList;

// Url completa del servicio web que contiene la información
private static String url_todos_Locales =
"http://" + ip + "/xampp/phpfiles/get_local_tipo.php";

// Nombra los Nodos JSON empleados
private static final String TAG_SUCCESS = "success";
private static final String TAG_Locales = "local";
private static final String TAG_id_Local = "id_Local";
private static final String TAG_Nombre = "Nombre";

// Declaración del arreglo Locales
JSONArray locales = null;

@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.LocalEst);

    //Obtiene los ID de los ImageButton
    tecnologia = (ImageButton) findViewById (R.id.imageButton14);
    celulares= (ImageButton) findViewById (R.id.imageButton2);
    juegos = (ImageButton) findViewById (R.id.imageButton3);
    joyeria= (ImageButton) findViewById (R.id.imageButton4);
    audio= (ImageButton) findViewById (R.id.imageButton5);

    //Añade el evento onTouchListener
    tecnologia.setOnTouchListener(this);
    celulares.setOnTouchListener(this);
    juegos.setOnTouchListener(this);
    joyeria.setOnTouchListener(this);
    audio.setOnTouchListener(this);
    // Declaración del Hashmap o colección de objetos
    localList = new ArrayList<HashMap<String, String>>();

    // Declaración del ListView para manejo de la lista
    ListView lv = getListView();

```

```

// Evento clic de un local seleccionado
lv.setOnItemClickListener(new OnItemClickListener() {

    //Adapta la vista de la lista a la interfaz xml que contiene la
    lista donde se obtiene su posición y el ID
    @Override
    public void onItemClick(AdapterView<?> parent, View view,
        int position, long id) {
        // Obtiene el valor del Id del Local
        String pid = ((TextView) view.findViewById(R.id.pid)).getText()
            .toString();

        // Comienza la Actividad Local Detalles la cual realizará la
        consulta por ID
        Intent in = new
        Intent(getApplicationContext(),LocalDetallesActivity.class);

        // Envia del local a la actividad LocalDetallesActivity
        in.putExtra(TAG_id_Local, pid);

        // Comienza la nueva actividad y espera respuesta
        startActivityForResult(in, 100);
        //Cierra la actividad Actual antes de abrir la siguiente
        BusquedaLocalesActivity.this.finish();
    }
});

}

//Evento clic del Boton Menu
public void ActividadMenu(View v){
    //Crea un nuevo intent para llamar a la siguiente actividad
    Intent i = new Intent(this, MainActivity.class);
    //Ejecuta la actividad para que muestre la segunda actividad
    startActivity(i);
    BusquedaLocalesActivity.this.finish();
}

//Evento clic del botón Salir
public void ActividadSalir(View v){
    System.exit(0);//Cierra la Actividad
}

// Evento Touch
@Override

```

```

public boolean onTouch(View v, MotionEvent m){

    switch(m.getAction())
    {
        //Maneja el evento Down, es decir al seleccionar una imagen
        case MotionEvent.ACTION_DOWN:
        {
            //Envía los parámetros de búsqueda, es decir Tecnología, Celulares,
            Juegos, etc para mostrar los locales correspondientes
                switch(v.getId()) {
                    case R.id.imageButton14:
                        pid=""+"Tecnologia"+"";
                        //Ejecuta la Función LoadAllLocales
                        new LoadAllLocales().execute();
                        localList.clear();
                        break;

                    case R.id.imageButton2:
                        pid=""+"Celulares"+"";
                        new LoadAllLocales().execute();
                        localList.clear();
                        break;

                    case R.id.imageButton3:
                        pid=""+"Juegos"+"";
                        new LoadAllLocales().execute();
                        localList.clear();
                        break;

                    case R.id.imageButton4:
                        pid=""+"Joyeria"+"";
                        new LoadAllLocales().execute();
                        localList.clear();
                        break;

                    case R.id.imageButton5:
                        pid=""+"Audio-Video"+"";
                        new LoadAllLocales().execute();
                        localList.clear();
                        break;
                }
            break;
        }
    }
}

```

```

    }

    }

    return true;
}

// Espera el código de respuesta de la Actividad Busqueda Locales
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    // Tiempo de espera del código de respuesta
    if (resultCode == 100) {

        Intent intent = getIntent();
        finish();
        startActivity(intent);
    }
}

/**
 * Actividad en segundo plano
 * */
class LoadAllLocales extends AsyncTask<String, String, String> {

/**
 * Después de empezar el hilo en segundo plano muestra el progress Dialog */

    @Override
    protected void onPreExecute() {
        super.onPreExecute();
        progressDialog = new ProgressDialog(BusquedaLocalesActivity.this);
        progressDialog.setMessage("Cargando Locales espere porfavor...");
        progressDialog.setIndeterminate(false);
        progressDialog.setCancelable(false);
        progressDialog.show();
    }

/**
 * Obtiene los locales en segundo plano
 * */
    @Override

```

```

    protected String doInBackground(String... args) {
// Construcción de Parámetros
List<NameValuePair> params = new ArrayList<NameValuePair>();
params.add(new BasicNameValuePair("Tipo", pid));
// Obtiene el string JSON de la URL
JSONObject json = JSONObject.makeHttpRequest(url_todos_Locales, "GET",
        params);

// Chequea la respuesta JSON que contenga el string All Locales
Log.d("All Locales: ", json.toString());

try {
    // Chequea si la respuesta es correcta
    int success = json.getInt(TAG_SUCCESS);
    if (success == 1) {
        // Locales encontrados
        // Obtiene el arreglo de Locales
        locales = json.getJSONArray(TAG_Locales);

        // Revisa el bucle de todos los locales
        for (int i = 0; i < locales.length(); i++) {
            JSONObject c = locales.getJSONObject(i);

            // Almacena cada ítem en la variable correspondiente
            String id = c.getString(TAG_id_Local);
            String name = c.getString(TAG_Nombre);

            // Crea un nuevo Hashmap
            HashMap<String, String> map = new HashMap<String,
                String>();

            // Añade cada hijo al nodo Hashmap con su
            correspondiente valor
            map.put(TAG_id_Local, id);
            map.put(TAG_Nombre, name);

            // Finalmente añade los datos a la lista
            correspondiente
            localList.add(map);
        }
    }
}

```

```

    } catch (JSONException e) {
        e.printStackTrace();
    }

    return null;
}

/**
 * Después de completar la tarea en segundo plano disminuye el Progress
 * Dialog
 * **/
@Override
protected void onPostExecute(String file_url) {
    // Disminuye el Diálogo luego de obtener todos los locales
    progressDialog.dismiss();
    // Actualiza la interfaz UI con el hilo en segundo plano
    runOnUiThread(new Runnable() {
        @Override
        public void run() {
            /**
             * Actualiza los datos JSON en la lista "localList"
             * **/
            ListAdapter adapter = new SimpleAdapter(
                BusquedaLocalesActivity.this, localList,
                R.layout.list_item, new String[] { TAG_id_Local,
                    TAG_Nombre},
                new int[] { R.id.pid, R.id.name });
            // Adapta la vista de la Lista
            setListAdapter(adapter);
        }
    });
}
}

```

### B.3 CLASE BÚSQUEDA PRODUCTOS

```

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

import org.apache.http.NameValuePair;

```



```

import org.apache.http.message.BasicNameValuePair;
import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;
import android.app.ListActivity;
import android.app.ProgressDialog;
import android.content.Intent;
import android.os.AsyncTask;
import android.os.Bundle;
import android.util.Log;
import android.view.MotionEvent;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ImageButton;
import android.widget.ListAdapter;
import android.widget.ListView;
import android.widget.SimpleAdapter;
import android.widget.TextView;

public class BusquedaProductosActivity extends ListActivity implements OnClickListener {
    // Declaración del Progress Dialog a utilizarse
    private ProgressDialog pDialog;
    // Declaración de los ImageButton a utilizarse
    ImageButton ccespiral, laptop, celulares, juegos, joyeria, tecnologiaOtros;
    //Variables tipo String
    String pid="";
    //Dirección IP del servicio web que contiene la información
    private static String ip = "172.30.35.22";

    // Creando el objeto JSON Parser
    JSONParser jParser = new JSONParser();

    ArrayList<HashMap<String, String>> productList;

    // Url completa del servicio web que contiene la información

    private static String url_all_products =
        "http://" + ip + "/xampp/phpfiles/get_producto_tipo.php";
    private static String url_all_laptops =
        "http://" + ip + "/xampp/phpfiles/get_producto_tipolaptops.php";
    private static String url_all_tecnologiaOtros =

```

```

        "http://" + ip + "/xampp/phpfiles/get_producto_tipotecvarios.php";
// Nombres de los NODOS JSON
private static final String TAG_SUCCESS = "success";
private static final String TAG_productos = "productos";
private static final String TAG_Producto = "Producto";
private static final String TAG_Modelo = "Modelo";
private static final String TAG_id_LocPro = "id_LocPro";
// Declaración del arreglo productos
JSONArray productos = null;

@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.productos);
    //Obtiene los ID de los ImageButton
    laptop = (ImageButton) findViewById (R.id.imageButton1);
    celulares= (ImageButton) findViewById (R.id.imageButton2);
    juegos = (ImageButton) findViewById (R.id.imageButton3);
    joyeria= (ImageButton) findViewById (R.id.imageButton4);
    tecnologiaOtros=(ImageButton) findViewById (R.id.imageButton14);
    //Añade el evento onTouchListener
    laptop.setOnTouchListener(this);
    celulares.setOnTouchListener(this);
    juegos.setOnTouchListener(this);
    joyeria.setOnTouchListener(this);
    tecnologiaOtros.setOnTouchListener(this);
    // Declaración del Hashmap o colección de objetos
    productList = new ArrayList<HashMap<String, String>>();

    // Declaración del ListView para manejo de la lista
    ListView lv = getListView();

    // Evento clic del producto seleccionado
    lv.setOnItemClickListener(new OnItemClickListener() {
//Adapta la vista de la lista a la interfaz xml que contiene esta
    lista donde se obtiene su posición e ID
        @Override
        public void onItemClick(AdapterView<?> parent, View view,
            int position, long id) {
            // Obtiene el valor del Id del producto
            String pid = ((TextView) view.findViewById(R.id.pid)).getText()
                .toString();

```

```

// Comienza la Actividad Oferta Detalles la cual realizará la
// consulta por ID
Intent in = new Intent(getApplicationContext(),
    ProductoDetallesActivity.class);

// Evia el ID a la actividad Producto Detalles
in.putExtra(TAG_id_LocPro, pid);

// Comienza la Actividad esperando su respuesta en segundo
// plano
startActivityForResult(in, 100);
BusquedaProductosActivity.this.finish();
    }
});
}

public void ActividadMenu(View v){

    Intent i = new Intent(this, MainActivity.class);
    startActivity(i);
    BusquedaProductosActivity.this.finish();
}

public void ActividadSalir(View v){
    System.exit(0); //Cierra la actividad
}
//EVENTO TOUCH
@Override
public boolean onTouch(View v, MotionEvent m){

//Maneja el evento Down es decir al seleccionar una imagen
switch(m.getAction())
{
case MotionEvent.ACTION_DOWN:
{
//Envía los parámetros de búsqueda es decir Tecnología, Celulares,
//Juegos, etc para mostrar las ofertas correspondientes
switch(v.getId()) {
case R.id.imageButton1:
pid=""+"Tecnologia"+"";
new LoadAllLaptopsTabs().execute(); //Ejecuta la Función

```

*LoadALLapsTabs*

```

        productList.clear();
        break;

    case R.id.imageButton14:
        pid=""+"Tecnologia"+"";
        new LoadAllTecnologiaVarios().execute();//Ejecuta la Función
            LoadALLOfertasTecnologiaVarios
        productList.clear();
        break;

    case R.id.imageButton2:
        pid=""+"Celulares"+"";
        new LoadAllProducts().execute();//Ejecuta la Función
            LoadALLProductos
        productList.clear();
        break;

    case R.id.imageButton3:
        pid=""+"Juegos"+"";
        new LoadAllProducts().execute();
        productList.clear();
        break;

    case R.id.imageButton4:
        pid=""+"Joyeria"+"";
        new LoadAllProducts().execute();
        productList.clear();
        break;

    }
    break;
}

}

return true;
}

// Espera el código de respuesta de la Actividad Busqueda Locales
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent
data) {

```

```

    super.onActivityResult(requestCode, resultCode, data);
    // Tiempo de espera del código de respuesta
    if (resultCode == 100) {

        Intent intent = getIntent();
        finish();
        startActivity(intent);
    }

}

/**
 * Actividad en Segundo Plano Productos en General
 * */
class LoadAllProducts extends AsyncTask<String, String, String> {

    /**
     * Antes de empezar el hilo en segundo plano muestra el progress Dialog
     * */
    @Override
    protected void onPreExecute() {
        super.onPreExecute();
        progressDialog = new ProgressDialog(BusquedaProductosActivity.this);
        progressDialog.setMessage("Cargando Productos espere porfavor...");
        progressDialog.setIndeterminate(false);
        progressDialog.setCancelable(false);
        progressDialog.show();
    }

    /**
     * Obtiene todas las ofertas de la URL
     * */
    @Override
    protected String doInBackground(String... args) {
        // Construcción de Parámetros
        List<NameValuePair> params = new ArrayList<NameValuePair>();
        params.add(new BasicNameValuePair("Tipo", pid));
        // Obtiene el string JSON de la URL
        JSONObject json = jParser.makeHttpRequest(url_all_products, "GET",
            params);

        // Chequea la respuesta JSON que contenga el string ALL productos
        Log.d("All productos: ", json.toString());
    }
}

```

```

try {
    // Chequea si la respuesta es correcta
    int success = json.getInt(TAG_SUCCESS);

    if (success == 1) {
        // productos encontrados
        // Obtiene el arreglo de productos
        productos = json.getJSONArray(TAG_productos);

        // Revisa el bucle de todos los Productos Encontrados
        for (int i = 0; i < productos.length(); i++) {
            JSONObject c = productos.getJSONObject(i);

            // Almacena cada ítem en la variable correspondiente
            String id = c.getString(TAG_id_LocPro);
            String name = c.getString(TAG_Producto);
            String modelo = c.getString(TAG_Modelo);
            //Crea un nuevo Hashmap
            HashMap<String, String> map = new HashMap<String,
            String>();

            // Añade cada hijo al nodo Hashmap con su
            correspondiente valor
            map.put(TAG_id_LocPro, id);
            map.put(TAG_Producto, name);
            map.put(TAG_Modelo, modelo);
            // Finalmente añade los datos a la lista correspondiente
            productList.add(map);

        }

    }
} catch (JSONException e) {
    e.printStackTrace();
}

return null;
}

```

```
/**
```

```
* Después de completar la tarea en segundo plano disminuye el Progress Dialog
```

```
**/
```

```

@Override
    protected void onPostExecute(String file_url) {
        // Disminuye el Diálogo luego de obtener todos los locales
        pDialog.dismiss();
        // Actualiza la interfaz UI con el hilo en segundo plano
        runOnUiThread(new Runnable() {
            @Override
                public void run() {
                    /**
                     * Actualiza los datos JSON en la lista
                     * */
                    ListAdapter adapter = new SimpleAdapter(
                        BusquedaProductosActivity.this, productList,
                        R.layout.list_pro, new String[] { TAG_id_LocPro,
                            TAG_Producto, TAG_Modelo},
                        new int[] { R.id.pid, R.id.producto, R.id.modelo });
                    // Adapta la vista de la Lista
                    setListAdapter(adapter);
                }
            });
        }
    }

}

class LoadAllLaptopsTabs extends AsyncTask<String, String, String> {

    /**
     *Tarea en Segundo Plano para cargar productos de Laptops y Tablets
     * */
    @Override
    protected void onPreExecute() {
        super.onPreExecute();
        pDialog = new ProgressDialog(BusquedaProductosActivity.this);
        pDialog.setMessage("Cargando Productos espere porfavor...");
        pDialog.setIndeterminate(false);
        pDialog.setCancelable(false);
        pDialog.show();
    }

    /**
     * Obtiene todas las laptops y tablets de la categoria TECNOLOGÍA
     * */
}

```

```

@Override
    protected String doInBackground(String... args) {
        // Construcción de parámetros
        List<NameValuePair> params = new ArrayList<NameValuePair>();
        params.add(new BasicNameValuePair("Tipo", pid));

        // Obtiene el string JSON de la URL
        JSONObject json = jParser.makeHttpRequest(url_all_Laptabs, "GET",
            params);

        // Chequea la respuesta JSON que contenga el string ALL Productos
        Log.d("All productos: ", json.toString());

        try {
            // Chequea el resultado correcto
            int success = json.getInt(TAG_SUCCESS);

            if (success == 1) {
                // Productos encontrados
                // Obtiene el arreglo de productos
                productos = json.getJSONArray(TAG_productos);

                // Revisa el bucle de todos las Ofertas Encontradas
                for (int i = 0; i < productos.length(); i++) {
                    JSONObject c = productos.getJSONObject(i);

                    // Almacena cada ítem en la variable correspondiente
                    String id = c.getString(TAG_id_LocPro);
                    String name = c.getString(TAG_Producto);
                    String modelo = c.getString(TAG_Modelo);
                    // crea un nuevo hashmap
                    HashMap<String, String> map = new HashMap<String,
                    String>();

                    // Añade cada hijo al nodo Hashmap con su correspondiente valor
                    map.put(TAG_id_LocPro, id);
                    map.put(TAG_Producto, name);
                    map.put(TAG_Modelo, modelo);
                    //Finalmente añade los datos a la lista
                    correspondiente
                    productList.add(map);
                }
            }
        }
    }

```



```

    }
} catch (JSONException e) {
    e.printStackTrace();
}

return null;
}

/**
 * Después de completar la tarea en segundo plano disminuye el
 * Progress Dialog
 * **/
@Override
protected void onPostExecute(String file_url) {
    // Disminuye el Diálogo luego de obtener todos los locales
    pDialog.dismiss();
    // Actualiza la interfaz UI con el hilo en segundo plano
    runOnUiThread(new Runnable() {
        @Override
        public void run() {
            /**
             * Actualiza los datos JSON en la lista
             * **/
            ListAdapter adapter = new SimpleAdapter(
                BusquedaProductosActivity.this, productList,
                R.layout.list_pro, new String[] { TAG_id_LocPro,
                    TAG_Producto, TAG_Modelo},
                new int[] { R.id.pid, R.id.producto, R.id.modelo });
            // Adapta la vista de la lista
            setListAdapter(adapter);
        }
    });
}

}

/**
 * Tarea en segundo plano para obtener productos de otro tipo de
 * artículos tecnológicos
 * **/
class LoadAllTecnologiaVarios extends AsyncTask<String, String, String>

```

```

* Antes de empezar el hilo en segundo plano muestra el Progress
Dialog
* */
@Override
protected void onPreExecute() {
    super.onPreExecute();
    pDialog = new ProgressDialog(BusquedaProductosActivity.this);
    pDialog.setMessage("Cargando Locales espere porfavor...");
    pDialog.setIndeterminate(false);
    pDialog.setCancelable(false);
    pDialog.show();
}

/**
* Obtiene todas los artículos de la categoría TECNOLOGÍA excepto
LAPTOPS Y TABLETS
* */
@Override
protected String doInBackground(String... args) {
    // Construcción de parámetros
    List<NameValuePair> params = new ArrayList<NameValuePair>();
    params.add(new BasicNameValuePair("Tipo", pid));

    // Obtiene el string JSON de la URL
    JSONObject json =
    jParser.makeHttpRequest(url_all_tecnologiaOtros,
        "GET", params);

    // Chequea la respuesta JSON que contenga el string ALL Ofertas
    Log.d("All productos: ", json.toString());

    try {
        // Chequea el resultado correcto
        int success = json.getInt(TAG_SUCCESS);

        if (success == 1) {
            // Producto encontrado
            // Obtiene el arreglo de productos
            productos = json.getJSONArray(TAG_productos);

            // Revisa el bucle de todos los productos encontrados
            for (int i = 0; i < productos.length(); i++) {

```

```

        JSONObject c = productos.getJSONObject(i);

        // Almacena cada variable JSON en el ítem
        correspondiente
        String id = c.getString(TAG_id_LocPro);
        String name = c.getString(TAG_Producto);
        String modelo = c.getString(TAG_Modelo);
        // Crea un nuevo hashmap
        HashMap<String, String> map = new HashMap<String,
            String>();

        // Añade cada hijo al nodo Hashmap con su
        correspondiente valor
        map.put(TAG_id_LocPro, id);
        map.put(TAG_Producto, name);
        map.put(TAG_Modelo, modelo);
        //Finalmente añade los datos a la lista
        correspondiente
        productList.add(map);

    }

}

} catch (JSONException e) {
    e.printStackTrace();
}

return null;
}

/**
 * Después de completar la tarea en segundo plano disminuye el progress dialog
 */
@Override
protected void onPostExecute(String file_url) {
    // Disminuye el Dialogo luego de obtener todos los locales
    pDialog.dismiss();
    // Actualiza la interfaz UI con el hilo en segundo plano
    runOnUiThread(new Runnable() {
        @Override
        public void run() {
            /**
             * Actualiza los datos JSON en la lista

```



```

import android.widget.TextView;

public class BusquedaOfertasActivity extends ListActivity implements OnTouchListener {
    // Declaración del Progress Dialog a utilizarse
    private ProgressDialog pDialog;
    // Declaración de los ImageButton a utilizarse
    ImageButton ccespiral, laptop, celulares, juegos, joyeria, tecnologia;
    //Variables tipo String
    String pid="";
    //Dirección IP del servicio web que contiene la información
    private static String ip = "172.30.35.22";
    // Creando el objeto JSON Parser
    JSONParser jParser = new JSONParser();

    ArrayList<HashMap<String, String>> ofertList;

    // Url completa del servicio web que contiene la información
    private static String url_all_ofertas =
        "http://" + ip + "/xampp/phpfiles/get_oferta_tipo.php";
    private static String url_all_laptabs =
        "http://" + ip + "/xampp/phpfiles/get_oferta_tipolaptops.php";
    private static String url_all_tecnologiaOtros =
        "http://" + ip + "/xampp/phpfiles/get_oferta_tipotecvarios.php";
    // Nombres de los NODOS JSON
    private static final String TAG_SUCCES = "success";
    private static final String TAG_ofertas = "oferta";
    private static final String TAG_PrecioOferta = "Oferta";
    private static final String TAG_Producto = "Producto";
    private static final String TAG_Modelo = "Modelo";
    private static final String TAG_id_LocPro = "id_LocPro";
    // Declaración del arreglo locales
    JSONArray ofertas = null;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.ofertas);

        //Obtiene los ID de los ImageButton
        laptop = (ImageButton) findViewById (R.id.imageButton1);
        celulares= (ImageButton) findViewById (R.id.imageButton2);
        juegos = (ImageButton) findViewById (R.id.imageButton3);
        joyeria= (ImageButton) findViewById (R.id.imageButton4);
    }
}

```

```

tecnologia=(ImageButton) findViewById (R.id.imageButton14);
//Añade el evento onTouchListener
laptop.setOnTouchListener(this);
celulares.setOnTouchListener(this);
juegos.setOnTouchListener(this);
joyeria.setOnTouchListener(this);
tecnologia.setOnTouchListener(this);
// Declaración del Hashmap o colección de objetos
ofertList = new ArrayList<HashMap<String, String>>();

// Declaración del ListView para manejo de la lista
ListView lv = getListView();

// Evento clic del producto seleccionado
lv.setOnItemClickListener(new OnItemClickListener() {
//Adapta la vista de la lista a la interfaz xml que contiene esta lista
donde se obtiene su posición e ID
@Override
public void onItemClick(AdapterView<?> parent, View view,
int position, long id) {
// Obtiene el valor del Id del producto
String pid = ((TextView) view.findViewById(R.id.pid)).getText()
.toString();

// Comienza la Actividad Oferta Detalles la cual realizará la
consulta por ID
Intent in = new Intent(getApplicationContext(),
OfertaDetallesActivity.class);

// Envía el PID a la actividad Oferta Detalles
in.putExtra(TAG_id_LocPro, pid);

// Comienza la Actividad esperando su respuesta en segundo
plano
startActivityForResult(in, 100);
BusquedaOfertasActivity.this.finish();
}
});
}
public void ActividadMenu(View v){

```

```

Intent i = new Intent(this, MainActivity.class);
startActivity(i);
BusquedaOfertasActivity.this.finish();
}

public void ActividadAtras(View v){

    Intent i = new Intent(this, MainActivity.class);
    startActivity(i);
    BusquedaOfertasActivity.this.finish();
}
//Actividad Salir
public void ActividadSalir(View v){
    System.exit(0);
    BusquedaOfertasActivity.this.finish();
}
//Evento TOUCH
@Override
public boolean onTouch(View v, MotionEvent m){

    switch(m.getAction())
    {
        //Maneja el evento Down es decir al seleccionar una imagen
        case MotionEvent.ACTION_DOWN:
        {
            //Envía los parámetros de búsqueda es decir Tecnología, Celulares,
            //Juegos etc. para mostrar las ofertas correspondientes
            switch(v.getId()) {
                case R.id.imageButton1:
                    pid=""+"Tecnologia"+"";
                    new LoadAllofertasLapsTabs().execute();//Ejecuta la Función
                                                                LoadALLLapsTabs

                    ofertList.clear();
                    break;

                case R.id.imageButton14:
                    pid=""+"Tecnologia"+"";
                    new LoadAllofertasTecnologiaOtros().execute();//Ejecuta la
                                                                Función LoadAllofertasTecnologiaOtros

                    ofertList.clear();
                    break;
            }
        }
    }
}

```

```

        case R.id.imageButton2:
            pid="\\"+"Celulares"+"\\";
            new LoadAllofertas().execute();//Ejecuta la Función
                                     LoadAllofertas

            ofertList.clear();
            break;

        case R.id.imageButton3:
            pid="\\"+"Juegos"+"\\";
            new LoadAllofertas().execute();
            ofertList.clear();
            break;

        case R.id.imageButton4:
            pid="\\"+"Joyeria"+"\\";
            new LoadAllofertas().execute();
            ofertList.clear();
            break;

    }
    break;
}

return true;
}

// Espera el código de respuesta de la Actividad Búsqueda Locales
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent
data) {
    super.onActivityResult(requestCode, resultCode, data);
    // Tiempo de espera del código de respuesta
    if (resultCode == 100) {
        Intent intent = getIntent();
        finish();
        startActivity(intent);
    }
}

/**

```



```

* Actividad en Segundo Plano Ofertas en General
* */
class LoadAllOfertas extends AsyncTask<String, String, String> {

    /**
    * Antes de empezar el hilo en segundo plano muestra el progress Dialog*
    */
    @Override
    protected void onPreExecute() {
        super.onPreExecute();
        pDialog = new ProgressDialog(BusquedaOfertasActivity.this);
        pDialog.setMessage("Cargando Ofertas espere porfavor...");
        pDialog.setIndeterminate(false);
        pDialog.setCancelable(false);
        pDialog.show();
    }

    /**
    * Obtiene todas las ofertas de la URL
    * */
    @Override
    protected String doInBackground(String... args) {
        // Construcción de Parámetros
        List<NameValuePair> params = new ArrayList<NameValuePair>();
        params.add(new BasicNameValuePair("Tipo", pid));
        // Obtiene el string JSON de la URL
        JSONObject json = jParser.makeHttpRequest(url_all_ofertas, "GET",
        params);

        // Chequea la respuesta JSON que contenga el string ALL Ofertas
        Log.d("All ofertas: ", json.toString());

        try {
            // Chequea si la respuesta es correcta
            int success = json.getInt(TAG_SUCCESS);

            if (success == 1) {
                //Ofertas encontradas
                // Obtiene el arreglo de Ofertas
                ofertas = json.getJSONArray(TAG_ofertas);

                // Revisa el bucle de todos las Ofertas Encontradas
                for (int i = 0; i < ofertas.length(); i++) {

```

```

JSONObject c = ofertas.getJSONObject(i);

// Almacena cada ítem en la variable correspondiente
String id = c.getString(TAG_id_LocPro);
String name = c.getString(TAG_Producto);
String modelo = c.getString(TAG_Modelo);
String precio = c.getString(TAG_PrecioOferta);
// Crea un nuevo Hashmap (Colección de Objetos)
HashMap<String, String> map = new HashMap<String,
String>();

// Añade cada hijo al nodo Hashmap con su
correspondiente valor
map.put(TAG_id_LocPro, id);
map.put(TAG_Producto, name);
map.put(TAG_Modelo, modelo);
map.put(TAG_PrecioOferta, precio);
// Finalmente añade los datos a la lista correspondiente
ofertList.add(map);

}

}
//Captura la excepción
} catch (JSONException e) {
    e.printStackTrace();
}

return null;
}

/**
 * Después de completar la tarea en segundo plano disminuye el Progress Dialog
 * **/
@Override
protected void onPostExecute(String file_url) {
    // Disminuye el Diálogo luego de obtener todos los locales
    pDialog.dismiss();
    // Actualiza la interfaz UI con el hilo en segundo plano
    runOnUiThread(new Runnable() {
        @Override
        public void run() {
            /**

```

```

        * Actualiza los datos JSON en la lista
        * */
ListAdapter adapter = new SimpleAdapter(
    BusquedaOfertasActivity.this, ofertList,
    R.layout.list_ofert, new String[] { TAG_id_LocPro,
        TAG_Producto, TAG_Modelo, TAG_PrecioOferta},
    new int[] { R.id.pid,
        R.id.producto, R.id.modelo, R.id.precio});
// Adapta la vista de la Lista
setListAdapter(adapter);
    }
});

}

}

/**
 * Tarea en Segundo Plano para cargar Ofertas de Laptops y Tablets
 * */
class LoadAllOfertasLapsTabs extends AsyncTask<String, String, String> {

/**
 * Antes de empezar el hilo en segundo plano muestra el progress Dialog* */
    @Override
    protected void onPreExecute() {
        super.onPreExecute();
        pDialog = new ProgressDialog(BusquedaOfertasActivity.this);
        pDialog.setMessage("Cargando Ofertas espere porfavor...");
        pDialog.setIndeterminate(false);
        pDialog.setCancelable(false);
        pDialog.show();
    }

/**
 * Obtiene las Ofertas de la URL
 * */
    @Override
    protected String doInBackground(String... args) {
        // Construcción de parámetros
        List<NameValuePair> params = new ArrayList<NameValuePair>();
        params.add(new BasicNameValuePair("Tipo", pid));
    }
}

```

```

// Obtiene el string JSON de la URL
JSONObject json = jParser.makeHttpRequest(url_all_Laptops, "GET",
params);

// Chequea la respuesta JSON que contenga el string ALL Ofertas
Log.d("All ofertas: ", json.toString());
try {
    // Chequea el resultado correcto
    int success = json.getInt(TAG_SUCCESS);

    if (success == 1) {
        // ofertas encontradas
        // Obtiene el arreglo de las ofertas
        ofertas = json.getJSONArray(TAG_ofertas);

        // Revisa el bucle de todos las Ofertas Encontradas
        for (int i = 0; i < ofertas.length(); i++) {
            JSONObject c = ofertas.getJSONObject(i);

            // Almacena cada ítem en la variable correspondiente
            String id = c.getString(TAG_id_LocPro);
            String name = c.getString(TAG_Producto);
            String modelo = c.getString(TAG_Modelo);
            String precio = c.getString(TAG_PrecioOferta);
            // Crea un Nuevo Hashmap o conjunto de objetos
            HashMap<String, String> map = new HashMap<String,
String>();

            // Añade cada hijo al nodo Hashmap con su
            correspondiente valor
            map.put(TAG_id_LocPro, id);
            map.put(TAG_Producto, name);
            map.put(TAG_Modelo, modelo);
            map.put(TAG_PrecioOferta, precio);
            //Finalmente añade los datos a la lista correspondiente
            ofertList.add(map);

        }

    }
} catch (JSONException e) {
    e.printStackTrace();
}

```

```

        return null;
    }

    /**
     * Después de completar la tarea en segundo plano disminuye el Progress Dialog
     * **/
    @Override
    protected void onPostExecute(String file_url) {
        // Disminuye el Dialogo luego de obtener todos los locales
        progressDialog.dismiss();
        // Actualiza la interfaz UI con el hilo en segundo plano
        runOnUiThread(new Runnable() {
            @Override
            public void run() {
                /**
                 * Actualiza los datos JSON en la lista
                 * **/
                ListAdapter adapter = new SimpleAdapter(
                    BusquedaOfertasActivity.this, ofertList,
                    R.layout.list_ofert, new String[] { TAG_id_LocPro,
                        TAG_Producto, TAG_Modelo, TAG_PrecioOferta},
                    new int[] { R.id.pid,
                        R.id.producto, R.id.modelo, R.id.precio});
                // Adapta la vista de la lista
                setListAdapter(adapter);
            }
        });
    }

}

/**
 * Tarea en segundo plano para obtener Ofertas de otro tipo de artículos
 * tecnológicos
 * **/
class LoadAllOfertasTecnologiaOtros extends AsyncTask<String, String, String> {

    /**
     * Antes de empezar el hilo en segundo plano muestra el progress Dialog
     * **/

```

```

@Override
protected void onPreExecute() {
    super.onPreExecute();
    progressDialog = new ProgressDialog(BusquedaOfertasActivity.this);
    progressDialog.setMessage("Cargando Ofertas espere porfavor...");
    progressDialog.setIndeterminate(false);
    progressDialog.setCancelable(false);
    progressDialog.show();
}

/**
 * Obtiene las Ofertas de la URL
 * */
@Override
protected String doInBackground(String... args) {
    // Construcción de parámetros
    List<NameValuePair> params = new ArrayList<NameValuePair>();
    params.add(new BasicNameValuePair("Tipo", pid));
    // Obtiene el string JSON de la URL
    JSONObject json = jParser.makeHttpRequest(url_all_tecnologiaOtros,
    "GET", params);

    // Chequea la respuesta JSON que contenga el string ALL Ofertas
    Log.d("All ofertas: ", json.toString());

    try {
        // Chequea el resultado correcto
        int success = json.getInt(TAG_SUCCESS);

        if (success == 1) {
            // ofertas encontradas
            // Obtiene el arreglo de Ofertas
            ofertas = json.getJSONArray(TAG_ofertas);

            // Revisa el bucle de todos las Ofertas Encontradas
            for (int i = 0; i < ofertas.length(); i++) {
                JSONObject c = ofertas.getJSONObject(i);

                // Almacena cada variable JSON en el ítem correspondiente
                String id = c.getString(TAG_id_LocPro);
                String name = c.getString(TAG_Producto);
                String modelo = c.getString(TAG_Modelo);
                String precio = c.getString(TAG_PrecioOferta);
            }
        }
    }
}

```

```

        // Crea un nuevo Hashmap-conjunto de objetos
        HashMap<String, String> map = new HashMap<String,
        String>();

        // Añade cada hijo al nodo Hashmap con su
        correspondiente valor
        map.put(TAG_id_LocPro, id);
        map.put(TAG_Producto, name);
        map.put(TAG_Modelo, modelo);
        map.put(TAG_PrecioOferta, precio);
        //Finalmente añade los datos a la lista correspondiente
        ofertList.add(map);

    }

}
} //captura la excepción
catch (JSONException e) {
    e.printStackTrace();
}

return null;
}

/**
 * Después de completar la tarea en segundo plano disminuye el progress dialog
 * **/
@Override
    protected void onPostExecute(String file_url) {
        // Disminuye el Diálogo luego de obtener todos los locales
        pDialog.dismiss();
        // Actualiza la interfaz UI con el hilo en segundo plano
        runOnUiThread(new Runnable() {
            @Override
                public void run() {
                    /**
                     * Actualiza los datos JSON en la lista
                     */
                    ListAdapter adapter = new SimpleAdapter(
                        BusquedaOfertasActivity.this, ofertList,
                        R.layout.list_ofert, new String[] { TAG_id_LocPro,
                            TAG_Producto, TAG_Modelo, TAG_PrecioOferta},
                        new int[] { R.id.pid,

```





```

TextView txtTelefono;
TextView txtSitio,txtLaptop,txtTablet,txtImpresora,txtOtros;
//Declaración de los imageview a utilizarse
ImageButton btnOfertas;
ImageButton laptop,tablet,impresora,otros,prod;
//Dirección IP del servidor que contiene los servicios web a utilizarse
private static String ip = "172.30.35.22";
String pid,num;

// Diálogo de progreso
private ProgressDialog pDialog;

// Declaración de la clase JSONParser
JSONParser jsonParser = new JSONParser();
// Declaración de los objetos JSON empleados
JSONParser jParser = new JSONParser();
JSONParser jlapParser = new JSONParser();
JSONParser jtabParser = new JSONParser();
//Declaración del Hashmap o conjunto de objetos
ArrayList<HashMap<String, String>> localList;
ArrayList<HashMap<String, String>> laptopList;
// Url completa del servicio web que contiene la información
private static final String url_local_detials =
"http://"+ip+"/xampp/phpfiles/get_local.php";
private static final String url_all_productos =
"http://"+ip+"/xampp/phpfiles/get_productos.php";
private static final String url_all_laptops =
"http://"+ip+"/xampp/phpfiles/get_laptops.php";
private static final String url_all_tablets =
"http://"+ip+"/xampp/phpfiles/get_tablets.php";
private static final String url_all_impresoras=
"http://"+ip+"/xampp/phpfiles/get_impresoras.php";
private static final String url_all_otros =
"http://"+ip+"/xampp/phpfiles/get_otrostecnologia.php";

// nodos JSON
private static final String TAG_SUCCESS = "success";
private static final String TAG_LOCAL= "local";
private static final String TAG_Nombre = "Nombre";
private static final String TAG_id_Local = "id_Local";
private static final String TAG_Producto = "Producto";
private static final String TAG_productos = "productos";

```

```

private static final String TAG_Tipo = "Tipo";
private static final String TAG_Piso = "Piso";
private static final String TAG_Sitio = "Sitio";
private static final String TAG_Telefono = "Telefono";
private static final String TAG_id_LocPro = "id_LocPro";
private static final String TAG_Modelo = "Modelo";
// Declaración de los arreglos empleados
JSONArray productos = null;
JSONArray laptops = null;
JSONArray tablets = null;
JSONArray impresoras = null;
JSONArray otrostec = null;

@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.Local);

    // Llama a la clase INTENT
    Intent i = getIntent();
    // Obtiene el ID del local enviado en la Actividad anterior
    pid = i.getStringExtra(TAG_id_Local);
    // Ejecuta la actividad en segundo plano GetLocalDetails
    new GetLocalDetails().execute();
    // Declaración del Hashmap o colección de objetos
    localList = new ArrayList<HashMap<String, String>>();
    //Obtiene los ID de los textview empleados
    txtLaptop = (TextView) findViewById(R.id.txtLaptop);
    txtTablet = (TextView) findViewById(R.id.txtTablet);
    txtImpresora= (TextView) findViewById(R.id.txtImpresora);
    txtOtros= (TextView) findViewById(R.id.txtVarios);
    //Obtiene los ID de los imagebutton empleados
    laptop = (ImageButton) findViewById (R.id.lap);
    tablet = (ImageButton) findViewById (R.id.tab);
    impresora = (ImageButton) findViewById (R.id.imp);
    otros = (ImageButton) findViewById (R.id.otros);
    prod=(ImageButton)findViewById(R.id.prod1);
    // Declaración del listview para manejo de la lista
    ListView lv = getListView();

    // Evento clic del producto seleccionado
    lv.setOnItemClickListener(new OnItemClickListener() {
//Adapta la vista de la lista a la interfaz xml que contiene esta lista

```

```

donde se obtiene su posición y ID
@Override
    public void onItemClick(AdapterView<?> parent, View view,
        int position, long id) {
        // Obtiene el valor del Id del producto y lo convierte a string
        String pid = ((TextView) view.findViewById(R.id.pid)).getText()
            .toString();
//Comienza la Actividad Producto Detalles la cual realizará la consulta por ID
        Intent in = new Intent(getApplicationContext(),
            ProductoDetallesLocalActivity.class);
        //Envía el ID del producto a la actividad Producto Detalles
        in.putExtra(TAG_id_LocPro, pid);

        // Inicia la actividad actual en espera de respuesta
        startActivityForResult(in, 100);
        LocalDetallesActivity.this.finish();

    }
});

}

public void cambiarActividad(View v){
    num = txtLocal.getText().toString(); //obtiene el contenido del cuadro
        de texto

    Intent i = new Intent(this, OfertasLocalActivity.class); //Crea un
        nuevo intent para llamar a la siguiente actividad
    i.putExtra("resultado", num); //Guarda la cadena en el intent con el
        nombre de resultado
    startActivity(i); //Ejecuta la actividad para que muestre la segunda
        actividad
    LocalDetallesActivity.this.finish();
}

public void ActividadAtras(View v){

    Intent i = new Intent(this, BusquedaLocalesActivity.class);
    startActivity(i);
    LocalDetallesActivity.this.finish();
}

public void ActividadMenu(View v){

```

```

        Intent i = new Intent(this, MainActivity.class);
        startActivity(i);
        LocalDetallesActivity.this.finish();
    }

    public void ActividadSalir(View v){
        System.exit(0);
    }

    //Carga Solo laptops
    public void cargarLaptops(View v){
        new LoadAllLaptops().execute();
        localList.clear();
    }

    //Carga solo tablets
    public void cargarTablets(View v){
        new LoadAllTablets().execute();
        localList.clear();
    }

    //Carga solo impresoras
    public void cargarImpresoras(View v){
        new LoadAllImpresoras().execute();
        localList.clear();
    }

    //Carga Otros artículos de tecnología
    public void cargarOtros(View v){
        new LoadAllOtros().execute();
        localList.clear();
    }

    /**
     * Tarea en segundo plano obtiene los detalles de cada local
     * */
    class GetLocalDetails extends AsyncTask<String, String, String> {

        /**
         * Antes de empezar el hilo en segundo plano muestra el progress Dialog
         * */

```

```

@Override
protected void onPreExecute() {
    super.onPreExecute();
    progressDialog = new ProgressDialog(LocalDetallesActivity.this);
    progressDialog.setMessage("Cargando Locales Espere...");
    progressDialog.setIndeterminate(false);
    progressDialog.setCancelable(true);
    progressDialog.show();
}

/**
 * Obtiene los detalles de los Locales de la URL
 * */
protected String doInBackground(String... params) {

    // Actualiza el UI del hilo en segundo plano
    runOnUiThread(new Runnable() {
        public void run() {

            int success;
            try {
                // Construcción de Parámetros
                List<NameValuePair> params = new
                ArrayList<NameValuePair>();
                params.add(new BasicNameValuePair("id_Local", pid));

                //Obtiene el detalle de los locales utilizando una
                petición GET de la clase JSON
                JSONObject json = jsonParser.makeHttpRequest(
                    url_Local_details, "GET", params);

                // Chequea la respuesta JSON
                Log.d("Single Product Details", json.toString());

                // Chequea el resultado correcto
                success = json.getInt(TAG_SUCCESS);
                if (success == 1) {
                    // En el caso de ser exitoso
                    JSONArray productObj = json
                        .getJSONArray(TAG_LOCAL); // Arreglo JSON

                    // Obtiene el primer objeto del arreglo JSON
                    JSONObject product = productObj.getJSONObject(0);
                }
            } catch (JSONException e) {
                e.printStackTrace();
            }
        }
    });
}

```

```
//Almacena el tipo y nombre en las variables string
String tipo =
product.getString(TAG_Tipo).toString();
String nom =
product.getString(TAG_Nombre).toString();

//CARGA LOS LOGOS DE CADA LOCAL
//SUBSUELO DOS
if(nom.equals("ESTUDIO CERO")){

    ImageView logo = (ImageView)
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local2);

}
else if(nom.equals("MARCESPLACE")){

    ImageView logo = (ImageView)
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local3);

}
else if(nom.equals("PUBLIPROMOCIONES")){

    ImageView logo = (ImageView)
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local4);

}
else if(nom.equals("DIGITAL GAMES")){

    ImageView logo = (ImageView)
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local536);

}
else if(nom.equals("RING SITE")){

    ImageView logo = (ImageView)
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local6);

}
else if(nom.equals("PALOMINO CENTRO ORTOPEDICO")){

    ImageView logo = (ImageView)
        findViewById(R.id.Logo);
```

```
        logo.setImageResource(R.drawable.Local7);
    }
    else if(nom.equals("HOUSE OF STYLE")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local8);
    }
    else if(nom.equals("REMI CENTER")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local9);
    }
    else if(nom.equals("MULTICELL")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local10);
    }
    else if(nom.equals("MIS LAPTOPS")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local12);
    }
    else if(nom.equals("ARTESANIAS TROYA")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local1314);
    }
    else if(nom.equals("CORED SYSTEMS")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local15);
    }
    else if(nom.equals("PUNTO SISTEMAS")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local16);
    }
}
```

```
}  
else if(nom.equals("SIA MARKET")){  
  
    ImageView logo = (ImageView)  
        findViewById(R.id.Logo);  
    logo.setImageResource(R.drawable.Local17);  
}  
else if(nom.equals("VIDEO MUSICAL")){  
  
    ImageView logo = (ImageView)  
        findViewById(R.id.Logo);  
    logo.setImageResource(R.drawable.Local19);  
}  
else if(nom.equals("MUNDO SYST")){  
  
    ImageView logo = (ImageView)  
        findViewById(R.id.Logo);  
    logo.setImageResource(R.drawable.Local20);  
}  
else if(nom.equals("POWER NET")){  
  
    ImageView logo = (ImageView)  
        findViewById(R.id.Logo);  
    logo.setImageResource(R.drawable.Local2124);  
}  
else if(nom.equals("NOVICOMPU")){  
  
    ImageView logo = (ImageView)  
        findViewById(R.id.Logo);  
    logo.setImageResource(R.drawable.novicompu);  
}  
else if(nom.equals("COMPU KIT")){  
  
    ImageView logo = (ImageView)  
        findViewById(R.id.Logo);  
    logo.setImageResource(R.drawable.Local23);  
}  
else if(nom.equals("GIGA SOFT")){  
  
    ImageView logo = (ImageView)  
        findViewById(R.id.Logo);  
    logo.setImageResource(R.drawable.Local25);  
}
```



```
else if(nom.equals("LABUTH SHAH")){

    ImageView logo = (ImageView)
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local2627);
}
else if(nom.equals("DHE SERIES STORE")){

    ImageView logo = (ImageView)
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local28);
}
else if(nom.equals("D&S ABOGADOS")){

    ImageView logo = (ImageView)
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local29);
}
else if(nom.equals("Peluqueria MAXBER")){

    ImageView logo = (ImageView)
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local30);
}
else if(nom.equals("AQUINO OPTICA")){

    ImageView logo = (ImageView)
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local31);
}
else if(nom.equals("MOVIEZONE")){

    ImageView logo = (ImageView)
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local32);
}
else if(nom.equals("UNIVERSAL TECHNOLOGY")){

    ImageView logo = (ImageView)
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local33);
}
else if(nom.equals("COMPUKIT")){
```

```
        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local34);
    }
    else if(nom.equals("SAGA TURS")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local35);
    }
    else if(nom.equals("CEVICHERIA SACA LA RESACA")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local3738);
    }
    //SUSBSUELO UNO
    else if(nom.equals("CALL CELL")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local3940);
    }
    else if(nom.equals("KONAMI GAMES")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local41);
    }
    else if(nom.equals("MAREA NEGRA")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local42);
    }
    else if(nom.equals("PELUQUERIA LIZ")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local43);
    }
    else if(nom.equals("CABANA IMPORT")){
```

```
ImageView logo = (ImageView)
    findViewById(R.id.Logo);
    logo.setImageResource(R.drawable.Local44);
}
else if(nom.equals("HOUSE OF STYLE")){

ImageView logo = (ImageView)
    findViewById(R.id.Logo);
    logo.setImageResource(R.drawable.Local45);
}
else if(nom.equals("MEGA UNIFORMES")){

ImageView logo = (ImageView)
    findViewById(R.id.Logo);
    logo.setImageResource(R.drawable.Local4648);
}
else if(nom.equals("RCM SISTEMAS")){

ImageView logo = (ImageView)
    findViewById(R.id.Logo);
    logo.setImageResource(R.drawable.Local47);
}
else if(nom.equals("CASA BOLIVIA")){

ImageView logo = (ImageView)
    findViewById(R.id.Logo);
    logo.setImageResource(R.drawable.Local4950);
}
else if(nom.equals("Peluqueria MAXBER")){

ImageView logo = (ImageView)
    findViewById(R.id.Logo);
    logo.setImageResource(R.drawable.Local5152);
}
else if(nom.equals("PRONATUS")){

ImageView logo = (ImageView)
    findViewById(R.id.Logo);
    logo.setImageResource(R.drawable.Local53);
}
else if(nom.equals("PRO DJ")){
```

```
        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local545556);
    }
    else if(nom.equals("JN COLLECTION")){
        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local57);
    }
    else if(nom.equals("CORE.COM")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local58);
    }
    else if(nom.equals("SACOS")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local59);
    }
    else if(nom.equals("CUERO ARAMIS")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local60);
    }
    //PLANTA BAJA
    else if(nom.equals("JM ROPA")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local6162);
    }
    else if(nom.equals("JOYERIA CEDILLO")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local63);
    }
    else if(nom.equals("CAMISERIA INGLESA")){

        ImageView logo = (ImageView)
```

```
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local64);
    }
    else if(nom.equals("MAQUENO REPUBLIC")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
            logo.setImageResource(R.drawable.Local6569);
    }
    else if(nom.equals("JOYERIA LA MINA")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
            logo.setImageResource(R.drawable.Local66);
    }
    else if(nom.equals("UNITECH")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
            logo.setImageResource(R.drawable.Local67);
    }
    else if(nom.equals("MUNDO DVD")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
            logo.setImageResource(R.drawable.Local68);
    }
    else if(nom.equals("CAPCOM")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
            logo.setImageResource(R.drawable.Local70);
    }
    else if(nom.equals("BRUSEHNKO")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
            logo.setImageResource(R.drawable.Local71);
    }
    else if(nom.equals("JOYERIA GRIJALBA")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
```

```
        logo.setImageResource(R.drawable.Local72);
    }
    else if(nom.equals("OXEN JEANS")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local73);
    }
    else if(nom.equals("POSDATA")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local74);
    }
    else if(nom.equals("PUNTO DIGITAL")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.is1);
    }
    //NIVEL UNO

    else if(nom.equals("MOVIE LIGHT")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local75);
    }
    else if(nom.equals("LABUTH SHAH 3")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local7679);
    }
    else if(nom.equals("TIAN SPORT")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local77);
    }
    else if(nom.equals("AMERICAN GAMES")){

        ImageView logo = (ImageView)
```

```
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local78);
    }
    else if(nom.equals("AUDIO CENTER")){

        ImageView logo = (ImageView)
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local80);
    }
    else if(nom.equals("LEVEL UP")){

        ImageView logo = (ImageView)
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local81);
    }
    else if(nom.equals("ALMACEN ARENA")){

        ImageView logo = (ImageView)
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local82);
    }
    else if(nom.equals("FABRICUERO")){

        ImageView logo = (ImageView)
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local83);
    }
    else if(nom.equals("ALMACENES DALI")){

        ImageView logo = (ImageView)
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local84);
    }
    else if(nom.equals("CELLCOM")){

        ImageView logo = (ImageView)
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local85);
    }
    else if(nom.equals("TAMASHAN")){

        ImageView logo = (ImageView)
        findViewById(R.id.Logo);
```

```

        logo.setImageResource(R.drawable.Local86);
    }
    else if(nom.equals("ANALUISA JOYEROS")){

        ImageView logo = (ImageView)
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local87);
    }
    else if(nom.equals("ISLA PANYOCHU")){

        ImageView logo = (ImageView)
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.is2);
    }

//NIVEL DOS
    else if(nom.equals("STAR PLATERIA")){

        ImageView logo = (ImageView)
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local89);
    }
    else if(nom.equals("THE HOUSE ROCK")){

        ImageView logo = (ImageView)
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local90);
    }
    else if(nom.equals("TECNOCABLE")){

        ImageView logo = (ImageView)
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local91);
    }
    else if(nom.equals("NOVEDADES ROSE")){

        ImageView logo = (ImageView)
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local92);
    }
    else if(nom.equals("JOYERIA AMATISTA")){

        ImageView logo = (ImageView)

```



```
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local93);
    }
    else if(nom.equals("EL MAGICO BAZAR")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
            logo.setImageResource(R.drawable.Local94);
        }
    else if(nom.equals("DE ONDAS")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
            logo.setImageResource(R.drawable.Local95);
        }
    else if(nom.equals("PUNTO EXTREMO")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
            logo.setImageResource(R.drawable.Local96);
        }
    else if(nom.equals("HELEN NOVEDADES")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
            logo.setImageResource(R.drawable.Local97);
        }
    else if(nom.equals("CELL MEP")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
            logo.setImageResource(R.drawable.Local98);
        }
    else if(nom.equals("LATIN TRAVEL")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
            logo.setImageResource(R.drawable.Local99100116);
        }
    else if(nom.equals("D KAREN")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
```

```
        logo.setImageResource(R.drawable.Local101);
    }
    else if(nom.equals("FACHA SPORT")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local102);
    }
    else if(nom.equals("BISUTERIA")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.is3);
    }

    //NIVEL TRES
    else if(nom.equals("NICELL")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local103);
    }
    else if(nom.equals("COMERSUR")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local104);
    }
    else if(nom.equals("ECUAVENTURA")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local105);
    }
    else if(nom.equals("TECNOLOGY CELL")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local106);
    }
    else if(nom.equals("LA TORRE")){

        ImageView logo = (ImageView)
```

```
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local107);
    }
    else if(nom.equals("TOPY SPORT")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local108);
    }
    else if(nom.equals("ENSUENOS KAREN")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local109);
    }
    else if(nom.equals("ANIME MANIA")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local110);
    }
    else if(nom.equals("DIGITAL PRODUCTION")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local111);
    }
    else if(nom.equals("CONSUHID COURIER")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local112);
    }
    else if(nom.equals("COBO IMPORTACIONES")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local113);
    }
    else if(nom.equals("PUBLIPROMOCIONES")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
```

```

        logo.setImageResource(R.drawable.Local114);
    }
    else if(nom.equals("CITY CELL")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local115);
    }
    else if(nom.equals("LATIN TRAVEL")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local116);
    }
    else if(nom.equals("ISLA PINGUINO")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.is4);
    }

    //NIVEL CUATRO
    else if(nom.equals("PERRO LOCO")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local117);
    }
    else if(nom.equals("KMIZZ")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local119);
    }
    else if(nom.equals("PUNTO DIGITAL")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local120);
    }
    else if(nom.equals("MY PC")){

        ImageView logo = (ImageView)

```

```
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local121);
    }
    else if(nom.equals("PUBLIPROMOCIONES")){

        ImageView logo = (ImageView)
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local122);
    }
    else if(nom.equals("REPARACIONES DIEGO")){

        ImageView logo = (ImageView)
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local123);
    }
    else if(nom.equals("NEW CAPCOM")){

        ImageView logo = (ImageView)
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local124);
    }
    else if(nom.equals("KAAS JUEGOS")){

        ImageView logo = (ImageView)
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local125);
    }
    else if(nom.equals("CUERO ARAMIS")){

        ImageView logo = (ImageView)
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local126);
    }
    else if(nom.equals("CAMARA ARTESANAL")){

        ImageView logo = (ImageView)
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local127);
    }
    else if(nom.equals("MARTIN CELL")){

        ImageView logo = (ImageView)
        findViewById(R.id.Logo);
```

```

        logo.setImageResource(R.drawable.Local128);
    }
else if(nom.equals("INGENIERIA CELULAR")){

    ImageView logo = (ImageView)
    findViewById(R.id.Logo);
    logo.setImageResource(R.drawable.Local129);
}
else if(nom.equals("CALZADOS EMY")){

    ImageView logo = (ImageView)
    findViewById(R.id.Logo);
    logo.setImageResource(R.drawable.Local130);
}
else if(nom.equals("KMIZZ")){

    ImageView logo = (ImageView)
    findViewById(R.id.Logo);
    logo.setImageResource(R.drawable.is5);
}

//NIVEL CINCO
else if(nom.equals("GAME CENTER")){

    ImageView logo = (ImageView)
    findViewById(R.id.Logo);
    logo.setImageResource(R.drawable.Local131);
}
else if(nom.equals("PLAY IT LOUD")){

    ImageView logo = (ImageView)
    findViewById(R.id.Logo);
    logo.setImageResource(R.drawable.Local132133);
}
else if(nom.equals("EC-ROBOTICS")){

    ImageView logo = (ImageView)
    findViewById(R.id.Logo);
    logo.setImageResource(R.drawable.Local134);
}
else if(nom.equals("CYBER COMP")){

    ImageView logo = (ImageView)

```

```
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local135);
    }
    else if(nom.equals("TFD")){

        ImageView logo = (ImageView)
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local136);
    }
    else if(nom.equals("FOLKLORE GREGORIO")){

        ImageView logo = (ImageView)
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local137);
    }
    else if(nom.equals("GRAFICAR PUBLICIDAD")){

        ImageView logo = (ImageView)
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local138);
    }
    else if(nom.equals("SMILE BOUTIQ")){

        ImageView logo = (ImageView)
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local139);
    }
    else if(nom.equals("ECUAVISION")){

        ImageView logo = (ImageView)
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local140);
    }
    else if(nom.equals("ALTO TRAFICO")){

        ImageView logo = (ImageView)
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local141142);
    }
    else if(nom.equals("SPAI SONS")){

        ImageView logo = (ImageView)
        findViewById(R.id.Logo);
```

```
        logo.setImageResource(R.drawable.Local143);
    }
    else if(nom.equals("YANBAL")){

        ImageView logo = (ImageView)
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local144);
    }
    else if(nom.equals("CONFITERIA LA NENA")){

        ImageView logo = (ImageView)
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.is6);
    }

    //NIVEL SEIS

    else if(nom.equals("EXOTIC THAI")){

        ImageView logo = (ImageView)
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local145);
    }
    else if(nom.equals("YNV TECHNOLOGY")){

        ImageView logo = (ImageView)
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local146);
    }
    else if(nom.equals("LAS UNIAS A SOPLETE")){

        ImageView logo = (ImageView)
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local147);
    }
    else if(nom.equals("ADMINISTRACION")){

        ImageView logo = (ImageView)
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local148);
    }
    else if(nom.equals("CORPORACIONES UNIDAS")){
```



```
        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local149);
    }
    else if(nom.equals("CAFU CIGARRILLOS")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local150);
    }
    else if(nom.equals("CELL PLANET")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local151);
    }
    else if(nom.equals("EROVER PC")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local152153154);
    }
    else if(nom.equals("TV.COM")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local155);
    }
    else if(nom.equals("VIZZON")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local156);
    }
    else if(nom.equals("ALHER TRAVEL")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local157);
    }

    if(tipo.equals("Tecnologia")){
```

```
laptop.setVisibility(View.VISIBLE);
tablet.setVisibility(View.VISIBLE);
impresora.setVisibility(View.VISIBLE);
otros.setVisibility(View.VISIBLE);

txtLaptop.setVisibility(View.VISIBLE);
txtTablet.setVisibility(View.VISIBLE);
txtImpresora.setVisibility(View.VISIBLE);
txtOtros.setVisibility(View.VISIBLE);
}
else if (tipo.equals("Juegos")){
    new LoadAllProductos().execute();
    prod.setVisibility(View.VISIBLE);
    prod.setImageResource(R.drawable.fifa13);
}
else if (tipo.equals("Joyeria")){
    new LoadAllProductos().execute();
    prod.setVisibility(View.VISIBLE);
    prod.setImageResource(R.drawable.joyas);
}
else if (tipo.equals("Audio-Video")){
    new LoadAllProductos().execute();
    prod.setVisibility(View.VISIBLE);
    prod.setImageResource(R.drawable.audio);
}
else if (tipo.equals("Ropa-Gala")){
    new LoadAllProductos().execute();
    prod.setVisibility(View.VISIBLE);
    prod.setImageResource(R.drawable.gala);
}
else if (tipo.equals("Ropa-Casual")){
    new LoadAllProductos().execute();
    prod.setVisibility(View.VISIBLE);
    prod.setImageResource(R.drawable.casual);
}
else if (tipo.equals("Camisetas")){
    new LoadAllProductos().execute();
    prod.setVisibility(View.VISIBLE);
    prod.setImageResource(R.drawable.camiseta);
}
else if (tipo.equals("Regalos")){
    new LoadAllProductos().execute();
    prod.setVisibility(View.VISIBLE);
```

```

        prod.setImageResource(R.drawable.regalos);
    }
    else if (tipo.equals("Comida")){
        new LoadAllProductos().execute();
        prod.setVisibility(View.VISIBLE);
        prod.setImageResource(R.drawable.comida);
    }
    else if (tipo.equals("Musica")){
        new LoadAllProductos().execute();
        prod.setVisibility(View.VISIBLE);
        prod.setImageResource(R.drawable.musica);
    }
    else if (tipo.equals("Servicio-Tecnico")){
        new LoadAllProductos().execute();
        prod.setVisibility(View.VISIBLE);
        prod.setImageResource(R.drawable.tecnico);
    }
}

// Muestra en la interfaz los datos correspondientes al piso, número de local, correo y
teléfono

        //Obtiene los identificadores de los textview utilizados
        txtPiso = (TextView) findViewById(R.id.inputPiso);
        txtLocal = (TextView) findViewById(R.id.inputLocal);
        txtSitio = (TextView) findViewById(R.id.inputSitio);
        txtTelefono = (TextView) findViewById(R.id.inputTelefono);

// Muestra los datos del local en el textview correspondiente

        txtPiso.setText(product.getString(TAG_Piso));
        txtLocal.setText(product.getString(TAG_id_Local));
        txtSitio.setText(product.getString(TAG_Sitio));
        txtTelefono.setText(product.getString(TAG_Telefono));
    }else{
    }
} catch (JSONException e) {
    e.printStackTrace();
}
}
});

return null;
}

```

```

    /**
    * Después de completar la tarea en segundo plano disminuye el progress dialog
    * **/
    protected void onPostExecute(String file_url) {
        progressDialog.dismiss();
    }
}

// Espera el código de respuesta de la Actividad Local Detalles
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);

    if (resultCode == 100) {

        Intent intent = getIntent();
        finish();
        startActivity(intent);
    }
}

/**
* Actividad en Segundo Plano que muestra los productos pertenecientes al Local
* */
class LoadAllProductos extends AsyncTask<String, String, String> {

    /**
    * Obtiene los productos del URL de acuerdo al ID del local
    * */
    protected String doInBackground(String... args) {
        // Construcción de parámetros
        List<NameValuePair> params = new ArrayList<NameValuePair>();
        params.add(new BasicNameValuePair("id_Local", pid));

        // Obtiene el string JSON de la URL
        JSONObject json = jParser.makeHttpRequest(url_all_productos, "GET",
            params);

        // Chequea la respuesta JSON correspondiente a ALL productos
        Log.d("All productos: ", json.toString());
    }
}

```

```

try {
    // Chequea el resultado correcto
    int success = json.getInt(TAG_SUCCESS);

    if (success == 1) {
        // producto encontrado
        // Obtiene el arreglo de productos
        productos = json.getJSONArray(TAG_productos);

        // Revisa el bucle de todos los Productos Encontrados
        for (int i = 0; i < productos.length(); i++) {
            JSONObject c = productos.getJSONObject(i);

            // Almacena cada item en la variable correspondiente
            String id = c.getString(TAG_id_LocPro);
            String name = c.getString(TAG_Producto);
            String modelo = c.getString(TAG_Modelo);
            //Crea un nuevo Hashmap
            HashMap<String, String> map = new HashMap<String,
            String>();
            // Añade cada hijo al nodo Hashmap con su correspondiente valor
            map.put(TAG_id_LocPro, id);
            map.put(TAG_Producto, name);
            map.put(TAG_Modelo, modelo);

            // Finalmente añade los datos a la lista correspondiente
            localList.add(map);

        }

    }
} catch (JSONException e) {
    e.printStackTrace();
}

return null;
}

/**
 * Actualiza y muestra la lista en la interfaz
 * **/
protected void onPostExecute(String file_url) {

```

```

runOnUiThread(new Runnable() {
    public void run() {
        /**
         * Actualiza los datos JSON en la lista correspondiente
         * */
        ListAdapter adapter = new SimpleAdapter(
            LocalDetallesActivity.this, locallist,
            R.layout.list_pro, new String[] { TAG_id_LocPro,
                TAG_Producto, TAG_Modelo},
            new int[] { R.id.pid, R.id.producto, R.id.modelo});
        // Adapta la vista de la lista
        setListAdapter(adapter);
    }
});

}

}

/**
 * Tarea en segundo plano que muestra solo laptops si el local seleccionado es del tipo
 * tecnología
 * */
class LoadAllLaptops extends AsyncTask<String, String, String> {

    /**
     * Obtiene los productos==laptops de la URL correspondiente
     * */
    protected String doInBackground(String... args) {
        // Construcción de parámetros
        List<NameValuePair> params = new ArrayList<NameValuePair>();
        params.add(new BasicNameValuePair("id_Local", pid));

        // Obtiene el string JSON de la URL
        JSONObject json = jParser.makeHttpRequest(url_all_laptops, "GET",
            params);

        // Chequea la respuesta JSON correspondiente a ALL Laptops
        Log.d("All laptops: ", json.toString());
    }
}

```

```

try {
    // Chequea el resultado exitoso
    int success = json.getInt(TAG_SUCCESS);

    if (success == 1) {
        // laptops encontradas
        // obtiene el arreglo de las laptops
        laptops = json.getJSONArray(TAG_productos);

        // Revisa el bucle de todos las laptops encontradas
        for (int i = 0; i < laptops.length(); i++) {
            JSONObject c = laptops.getJSONObject(i);

            // Almacena cada item en la variable correspondiente
            String id = c.getString(TAG_id_LocPro);
            String name = c.getString(TAG_Producto);
            String modelo = c.getString(TAG_Modelo);
            //Crea un nuevo Hashmap
            HashMap<String, String> map = new HashMap<String,
            String>();

            // Añade cada hijo al nodo Hashmap con su correspondiente valor
            map.put(TAG_id_LocPro, id);
            map.put(TAG_Producto, name);
            map.put(TAG_Modelo, modelo);

            // Finalmente añade los datos a la lista correspondiente
            localList.add(map);

        }

    }
} catch (JSONException e) {
    e.printStackTrace();
}

return null;
}

/**
 * Adapta a la interfaz la lista correspondiente
 * **/
protected void onPostExecute(String file_url) {

```

```

runOnUiThread(new Runnable() {
    public void run() {
        /**
         * Actualiza los datos JSON en el Listview
         * */
        ListAdapter adapter = new SimpleAdapter(
            LocalDetallesActivity.this, locallist,
            R.layout.list_pro, new String[] { TAG_id_LocPro,
                TAG_Producto, TAG_Modelo},
            new int[] { R.id.pid, R.id.producto, R.id.modelo});
        // Adapta los datos a la lista
        setListAdapter(adapter);
    }
});

}

}

/**
 * Tarea en segundo plano que muestra solo tablets si el local seleccionado es del tipo
 * tecnología
 * */
class LoadAllTablets extends AsyncTask<String, String, String> {

    /**
     * Obtiene los productos==tablets de la URL
     * */
    protected String doInBackground(String... args) {
        // Construcción de parámetros
        List<NameValuePair> params = new ArrayList<NameValuePair>();
        params.add(new BasicNameValuePair("id_Local", pid));

        //Obtiene el string JSON de la URL
        JSONObject json = jParser.makeHttpRequest(url_all_tablets, "GET",
            params);

        // Chequea la respuesta JSON correspondiente a ALL tablets
        Log.d("All tablets: ", json.toString());
    }
}

```



```

try {
    // Chequea el resultado exitoso
    int success = json.getInt(TAG_SUCCESS);

    if (success == 1) {
        // tablets encontradas
        // Obtiene el arreglo JSON de tablets
        tablets = json.getJSONArray(TAG_productos);

        // Revisa el bucle de todas las laptops encontradas
        for (int i = 0; i < tablets.length(); i++) {
            JSONObject c = tablets.getJSONObject(i);

            // Almacena cada ítem en la variable correspondiente
            String id = c.getString(TAG_id_LocPro);
            String name = c.getString(TAG_Producto);
            String modelo = c.getString(TAG_Modelo);
            //Crea un nuevo hashmap
            HashMap<String, String> map = new HashMap<String,
            String>();

            // Añade cada hijo al nodo Hashmap con su correspondiente valor
            map.put(TAG_id_LocPro, id);
            map.put(TAG_Producto, name);
            map.put(TAG_Modelo, modelo);

            // Finalmente añade los datos a la lista correspondiente
            localList.add(map);

        }

    }
} catch (JSONException e) {
    e.printStackTrace();
}

return null;
}

/**
 * Adapta la lista correspondiente
 * **/

```

```

protected void onPostExecute(String file_url) {

    runOnUiThread(new Runnable() {
        public void run() {
            /**
             * Actualiza los datos JSON en el listview
             * */
            ListAdapter adapter = new SimpleAdapter(
                LocalDetallesActivity.this, locallist,
                R.layout.list_pro, new String[] { TAG_id_LocPro,
                    TAG_Producto, TAG_Modelo},
                new int[] { R.id.pid, R.id.producto, R.id.modelo});
            // Adapta los datos a la lista
            setListAdapter(adapter);
        }
    });

}

}

/**
 * Tarea en segundo plano que muestra solo impresoras si el local seleccionado es del tipo
 * tecnología
 * */
class LoadAllImpresoras extends AsyncTask<String, String, String> {

    /**
     * Obtiene los productos==tablets de la URL
     * */
    protected String doInBackground(String... args) {
        // Construcción de parámetros
        List<NameValuePair> params = new ArrayList<NameValuePair>();
        params.add(new BasicNameValuePair("id_Local", pid));

        //Obtiene el string JSON de la URL
        JSONObject json = jParser.makeHttpRequest(url_all_impresoras, "GET",
            params);

        // Chequea la respuesta JSON correspondiente a ALL tablets
        Log.d("All impresoras: ", json.toString());

        try {

```

```

// Chequea el resultado exitoso
    int success = json.getInt(TAG_SUCCESS);

    if (success == 1) {
        // Impresoras encontradas
        // Obtiene el arreglo JSON de tablets
        impresoras = json.getJSONArray(TAG_productos);

        // Revisa el bucle de todas las laptops encontradas
        for (int i = 0; i < impresoras.length(); i++) {
            JSONObject c = impresoras.getJSONObject(i);

            // Almacena cada ítem en la variable correspondiente
            String id = c.getString(TAG_id_LocPro);
            String name = c.getString(TAG_Producto);
            String modelo = c.getString(TAG_Modelo);
            // Crea un nuevo hashmap
            HashMap<String, String> map = new HashMap<String,
            String>();

            // Añade cada hijo al nodo Hashmap con su correspondiente valor
            map.put(TAG_id_LocPro, id);
            map.put(TAG_Producto, name);
            map.put(TAG_Modelo, modelo);

            // Finalmente añade los datos a la lista correspondiente
            localList.add(map);

        }

    }

} catch (JSONException e) {
    e.printStackTrace();
}

return null;
}

/**
 * Adapta la lista correspondiente
 * **/
protected void onPostExecute(String file_url) {

```

```

runOnUiThread(new Runnable() {
    public void run() {
        /**
         * Actualiza los datos JSON en el listview
         * */
        ListAdapter adapter = new SimpleAdapter(
            LocalDetallesActivity.this, locallist,
            R.layout.list_pro, new String[] { TAG_id_LocPro,
                TAG_Producto, TAG_Modelo},
            new int[] { R.id.pid, R.id.producto, R.id.modelo});
        // Adapta los datos a la lista
        setListAdapter(adapter);
    }
});

}

}

```

```

/* * Tarea en segundo plano que muestra otro tipo de artículos que no sean
laptop, tablets o impresoras si el local seleccionado es del tipo tecnología
* */

```

```

class LoadAllOtros extends AsyncTask<String, String, String> {

    /**
     * Obtiene los productos==otros de la URL
     * */
    protected String doInBackground(String... args) {
        // Construcción de parámetros
        List<NameValuePair> params = new ArrayList<NameValuePair>();
        params.add(new BasicNameValuePair("id_Local", pid));

        //Obtiene el string JSON de la URL
        JSONObject json = jParser.makeHttpRequest(url_all_otros, "GET",
            params);

        // Chequea la respuesta JSON correspondiente a ALL otros
        Log.d("All otros: ", json.toString());

        try {
            // Chequea el resultado correcto

```

```

int success = json.getInt(TAG_SUCCESS);

if (success == 1) {
    // producto encontrado
    // Obtiene el arreglo JSON de otros productos
    otrostec = json.getJSONArray(TAG_productos);

    // Revisa el bucle de todas las laptops encontradas
    for (int i = 0; i < otrostec.length(); i++) {
        JSONObject c = otrostec.getJSONObject(i);

        // Almacena cada ítem en la variable correspondiente
        String id = c.getString(TAG_id_LocPro);
        String name = c.getString(TAG_Producto);
        String modelo = c.getString(TAG_Modelo);
        // Crea un nuevo hashmap
        HashMap<String, String> map = new HashMap<String,
        String>();

        // Añade cada hijo al nodo Hashmap con su
        correspondiente valor
        map.put(TAG_id_LocPro, id);
        map.put(TAG_Producto, name);
        map.put(TAG_Modelo, modelo);

        // Añade los datos a la lista correspondiente
        locallist.add(map);

    }

}

} catch (JSONException e) {
    e.printStackTrace();
}

return null;
}

/**
 * Adapta la lista correspondiente
 * **/
protected void onPostExecute(String file_url) {

```



```

public class ProductoDetallesActivity extends Activity{
    //Declaración de los textview a utilizarse
    TextView
    txtSitio,txtTelefono,txtNum,txtPiso,txtProducto,txtPrecio,txtModelo,txtProcesador,txtDis
    co,txtMemoria,txtCamara,txtSistema,txtGarantia,txtStock,txtPantalla,txtVarios;
    EditText txtejemplo;
    TextView
    txtVarios1,txtVarios2,txtVarios3,txtVarios4,txtVarios5,txtVarios6,txtVarios7,txtAnuncio;
    //Declaración de los imageview a utilizarse
    ImageView logo,prod1,prod2,logoC;
    String pid="",id;

    //Dirección IP del servidor que contiene los servicios web a utilizarse
    private static String ip = "172.30.35.22";

    // Diálogo de progreso
    private ProgressDialog pDialog;

    // Declaración de la clase JSONParser
    JSONParser jsonParser = new JSONParser();

    // Url completa del servicio web que contiene la información
    private static final String url_product_detials =
    "http://"+ip+"/xampp/phpfiles/get_producto_detalles.php";

    // nodos JSON
    private static final String TAG_SUCCESS = "success";
    private static final String TAG_PRODUCTO= "producto";
    private static final String TAG_Nombre = "Nombre";
    private static final String TAG_Piso = "Piso";
    private static final String TAG_Telefono = "Telefono";
    private static final String TAG_Sitio = "Sitio";
    private static final String TAG_Producto = "Producto";
    private static final String TAG_Numero = "id_Local";
    private static final String TAG_Precio = "Precio";
    private static final String TAG_Modelo = "Modelo";
    private static final String TAG_Disco = "Disco";
    private static final String TAG_Procesador = "Procesador";
    private static final String TAG_Memoria = "Memoria";
    private static final String TAG_Camara = "Camara";
    private static final String TAG_Sistema = "SistemaOperativo";
    private static final String TAG_Garantia = "Garantia";
    private static final String TAG_Stock = "Stock";

```

```

private static final String TAG_Pantalla = "Pantalla";
private static final String TAG_Varios = "Varios";
private static final String TAG_Otros1 = "Otros1";
private static final String TAG_Otros2 = "Otros2";
private static final String TAG_Otros3 = "Otros3";
private static final String TAG_Otros4 = "Otros4";
private static final String TAG_Otros5 = "Otros5";
private static final String TAG_id_LocPro = "id_LocPro";

@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.producinfo);
    logo = (ImageView)findViewById(R.id.Logo);

    // Llama a la clase INTENT
    Intent i = getIntent();

    // Obtiene el ID del local enviado en la Actividad anterior
    pid = i.getStringExtra(TAG_id_LocPro);

    // Ejecuta la actividad en segundo plano GetProductoDetails
    new GetProductoDetails().execute();

    txtAnuncio=(TextView)findViewById(R.id.inputAnuncio);
    Animation trasladar = AnimationUtils.LoadAnimation(this,
        R.anim.trasladar);
    txtAnuncio.setAnimation(trasladar);
}

public void ActividadAtras(View v){

    Intent i = new Intent(this, BusquedaProductosActivity.class);
    startActivity(i);
    ProductoDetallesActivity.this.finish();
}

public void ActividadMenu(View v){

    Intent i = new Intent(this, MainActivity.class);
    startActivity(i);
    ProductoDetallesActivity.this.finish();
}

```



```

public void ActividadSalir(View v){
    System.exit(0);
}

/**
 * Tarea en segundo plano obtiene los detalles de cada producto
 * */
class GetProductoDetails extends AsyncTask<String, String, String> {

    /**
     * Antes de empezar el hilo en segundo plano muestra el progress Dialog
     * */
    @Override
    protected void onPreExecute() {
        super.onPreExecute();
        pDialog = new ProgressDialog(ProductoDetallesActivity.this);
        pDialog.setMessage("Cargando Información Espere...");
        pDialog.setIndeterminate(false);
        pDialog.setCancelable(true);
        pDialog.show();
    }

    /**
     * Obtiene los detalles de los productos de la URL
     * */
    protected String doInBackground(String... params) {

        // Actualiza el UI del hilo en segundo plano
        runOnUiThread(new Runnable() {
            public void run() {
                int success;
                try {
                    // Construcción de Parámetros
                    List<NameValuePair> params = new
                    ArrayList<NameValuePair>();
                    params.add(new BasicNameValuePair("id_LocPro", pid));

                    //Obtiene el detalle de los productos utilizando una
                    petición GET de la clase JSON
                    JSONObject json = jsonParser.makeHttpRequest(
                        url_product_detials, "GET", params);
                }
            }
        });
    }
}

```

```

// Chequea la respuesta JSON
Log.d("Single Product Details", json.toString());

// Chequea el resultado correcto
success = json.getInt(TAG_SUCCESS);
if (success == 1) {
    // En el caso de ser exitoso
    JSONArray productObj = json
        .getJSONArray(TAG_PRODUCTO); // Arreglo JSON
    // Obtiene el primer objeto del arreglo JSON
    JSONObject product = productObj.getJSONObject(0);

    //Almacena el tipo y nombre en las variables string
    String pro =
        product.getString(TAG_Producto).toString();
    String imp= "IMPRESORA";
    String lap= "LAPTOP";
    String nom =
        product.getString(TAG_Nombre).toString();

// Muestra en la interfaz los datos correspondientes al piso, número de local, correo y
teléfono

    //Obtiene los identificadores de los textview utilizados
    txtPiso = (TextView) findViewById(R.id.inputPiso);
    txtNum = (TextView) findViewById(R.id.inputLocal);
    txtSitio = (TextView) findViewById(R.id.inputSitio);
    txtTelefono=(TextView)findViewById(R.id.inputTelefono);

// Muestra los datos del local en el textview correspondiente
    txtPiso.setText(product.getString(TAG_Piso));
    txtNum.setText(product.getString(TAG_Numero));
    txtSitio.setText(product.getString(TAG_Sitio));
    txtTelefono.setText(product.getString(TAG_Telefono));

    //CARGA LOS LOGOS DE CADA LOCAL
    //SUBSUELO DOS
    if(nom.equals("ESTUDIO CERO")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
            logo.setImageResource(R.drawable.Local2);
    }
}

```

```
}  
else if(nom.equals("MARCESPLACE")){  
  
    ImageView logo = (ImageView)  
        findViewById(R.id.Logo);  
    logo.setImageResource(R.drawable.Local3);  
}  
else if(nom.equals("PUBLIPROMOCIONES")){  
  
    ImageView logo = (ImageView)  
        findViewById(R.id.Logo);  
    logo.setImageResource(R.drawable.Local4);  
}  
else if(nom.equals("DIGITAL GAMES")){  
  
    ImageView logo = (ImageView)  
        findViewById(R.id.Logo);  
    logo.setImageResource(R.drawable.Local536);  
}  
else if(nom.equals("RING SITE")){  
  
    ImageView logo = (ImageView)  
        findViewById(R.id.Logo);  
    logo.setImageResource(R.drawable.Local6);  
}  
else if(nom.equals("PALOMINO CENTRO ORTOPEIDICO")){  
  
    ImageView logo = (ImageView)  
        findViewById(R.id.Logo);  
    logo.setImageResource(R.drawable.Local7);  
}  
else if(nom.equals("HOUSE OF STYLE")){  
  
    ImageView logo = (ImageView)  
        findViewById(R.id.Logo);  
    logo.setImageResource(R.drawable.Local8);  
}  
else if(nom.equals("REMI CENTER")){  
  
    ImageView logo = (ImageView)  
        findViewById(R.id.Logo);  
    logo.setImageResource(R.drawable.Local9);  
}
```

```
else if(nom.equals("MULTICELL")){

    ImageView logo = (ImageView)
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local10);
}
else if(nom.equals("MIS LAPTOPS")){

    ImageView logo = (ImageView)
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local12);
}
else if(nom.equals("ARTESANIAS TROYA")){

    ImageView logo = (ImageView)
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local1314);
}
else if(nom.equals("CORED SYSTEMS")){

    ImageView logo = (ImageView)
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local15);
}
else if(nom.equals("PUNTO SISTEMAS")){

    ImageView logo = (ImageView)
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local16);
}
else if(nom.equals("SIA MARKET")){

    ImageView logo = (ImageView)
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local17);
}
else if(nom.equals("VIDEO MUSICAL")){

    ImageView logo = (ImageView)
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local19);
}
else if(nom.equals("MUNDO SYST")){
```

```
ImageView logo = (ImageView)
    findViewById(R.id.Logo);
    logo.setImageResource(R.drawable.Local20);
}
else if(nom.equals("POWER NET")){

    ImageView logo = (ImageView)
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local2124);
    }
else if(nom.equals("NOVICOMPU")){

    ImageView logo = (ImageView)
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.novicompu);
    }
else if(nom.equals("COMPU KIT")){

    ImageView logo = (ImageView)
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local23);
    }
else if(nom.equals("GIGA SOFT")){

    ImageView logo = (ImageView)
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local25);
    }
else if(nom.equals("LABUTH SHAH")){

    ImageView logo = (ImageView)
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local2627);
    }
else if(nom.equals("DHE SERIES STORE")){

    ImageView logo = (ImageView)
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local28);
    }
else if(nom.equals("D&S ABOGADOS")){
```



```
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local3738);
    }
    //SUSBSUELO UNO
    else if(nom.equals("CALL CELL")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
            logo.setImageResource(R.drawable.Local3940);
    }
    else if(nom.equals("KONAMI GAMES")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
            logo.setImageResource(R.drawable.Local41);
    }
    else if(nom.equals("MAREA NEGRA")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
            logo.setImageResource(R.drawable.Local42);
    }
    else if(nom.equals("PELUQUERIA LIZ")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
            logo.setImageResource(R.drawable.Local43);
    }
    else if(nom.equals("CABANA IMPORT")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
            logo.setImageResource(R.drawable.Local44);
    }
    else if(nom.equals("HOUSE OF STYLE")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
            logo.setImageResource(R.drawable.Local45);
    }
    else if(nom.equals("MEGA UNIFORMES")){

        ImageView logo = (ImageView)
```

```
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local4648);
    }
    else if(nom.equals("RCM SISTEMAS")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
            logo.setImageResource(R.drawable.Local47);
    }
    else if(nom.equals("CASA BOLIVIA")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
            logo.setImageResource(R.drawable.Local4950);
    }
    else if(nom.equals("Peluqueria MAXBER")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
            logo.setImageResource(R.drawable.Local5152);
    }
    else if(nom.equals("PRONATUS")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
            logo.setImageResource(R.drawable.Local53);
    }
    else if(nom.equals("PRO DJ")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
            logo.setImageResource(R.drawable.Local545556);
    }
    else if(nom.equals("JN COLLECTION")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
            logo.setImageResource(R.drawable.Local57);
    }
    else if(nom.equals("CORE.COM")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
```



```

        logo.setImageResource(R.drawable.Local58);
    }
    else if(nom.equals("SACOS")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local59);
    }
    else if(nom.equals("CUERO ARAMIS")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local60);
    }
    //PLANTA BAJA
    else if(nom.equals("JM ROPA")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local6162);
    }
    else if(nom.equals("JOYERIA CEDILLO")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local63);
    }
    else if(nom.equals("CAMISERIA INGLESA")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local64);
    }
    else if(nom.equals("MAQUENO REPUBLIC")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local6569);
    }
    else if(nom.equals("JOYERIA LA MINA")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);

```

```
        logo.setImageResource(R.drawable.Local66);
    }
    else if(nom.equals("UNITECH")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local67);
    }
    else if(nom.equals("MUNDO DVD")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local68);
    }
    else if(nom.equals("CAPCOM")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local70);
    }
    else if(nom.equals("BRUSEHNKO")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local71);
    }
    else if(nom.equals("JOYERIA GRIJALBA")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local72);
    }
    else if(nom.equals("OXEN JEANS")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local73);
    }
    else if(nom.equals("POSDATA")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local74);
    }
}
```

```
    }  
    else if(nom.equals("PUNTO DIGITAL")){  
        ImageView logo = (ImageView)  
            findViewById(R.id.Logo);  
        logo.setImageResource(R.drawable.is1);  
    }  
    //NIVEL UNO  
  
    else if(nom.equals("MOVIE LIGHT")){  
        ImageView logo = (ImageView)  
            findViewById(R.id.Logo);  
        logo.setImageResource(R.drawable.Local75);  
    }  
    else if(nom.equals("LABUTH SHAH 3")){  
        ImageView logo = (ImageView)  
            findViewById(R.id.Logo);  
        logo.setImageResource(R.drawable.Local7679);  
    }  
    else if(nom.equals("TIAN SPORT")){  
        ImageView logo = (ImageView)  
            findViewById(R.id.Logo);  
        logo.setImageResource(R.drawable.Local77);  
    }  
    else if(nom.equals("AMERICAN GAMES")){  
        ImageView logo = (ImageView)  
            findViewById(R.id.Logo);  
        logo.setImageResource(R.drawable.Local78);  
    }  
    else if(nom.equals("AUDIO CENTER")){  
        ImageView logo = (ImageView)  
            findViewById(R.id.Logo);  
        logo.setImageResource(R.drawable.Local80);  
    }  
    else if(nom.equals("LEVEL UP")){  
        ImageView logo = (ImageView)  
            findViewById(R.id.Logo);
```

```
        logo.setImageResource(R.drawable.Local81);
    }
    else if(nom.equals("ALMACEN ARENA")){

        ImageView logo = (ImageView)
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local82);
    }
    else if(nom.equals("FABRICUERO")){

        ImageView logo = (ImageView)
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local83);
    }
    else if(nom.equals("ALMACENES DALI")){

        ImageView logo = (ImageView)
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local84);
    }
    else if(nom.equals("CELLCOM")){

        ImageView logo = (ImageView)
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local85);
    }
    else if(nom.equals("TAMASHAN")){

        ImageView logo = (ImageView)
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local86);
    }
    else if(nom.equals("ANALUISA JOYEROS")){

        ImageView logo = (ImageView)
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local87);
    }
    else if(nom.equals("ISLA PANYOCHU")){

        ImageView logo = (ImageView)
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.is2);
```

```
}

//NIVEL DOS
else if(nom.equals("STAR PLATERIA")){

    ImageView logo = (ImageView)
        findViewById(R.id.Logo);
    logo.setImageResource(R.drawable.Local89);
}
else if(nom.equals("THE HOUSE ROCK")){

    ImageView logo = (ImageView)
        findViewById(R.id.Logo);
    logo.setImageResource(R.drawable.Local90);
}
else if(nom.equals("TECNOCABLE")){

    ImageView logo = (ImageView)
        findViewById(R.id.Logo);
    logo.setImageResource(R.drawable.Local91);
}
else if(nom.equals("NOVEDADES ROSE")){

    ImageView logo = (ImageView)
        findViewById(R.id.Logo);
    logo.setImageResource(R.drawable.Local92);
}
else if(nom.equals("JOYERIA AMATISTA")){

    ImageView logo = (ImageView)
        findViewById(R.id.Logo);
    logo.setImageResource(R.drawable.Local93);
}
else if(nom.equals("EL MAGICO BAZAR")){

    ImageView logo = (ImageView)
        findViewById(R.id.Logo);
    logo.setImageResource(R.drawable.Local94);
}
else if(nom.equals("DE ONDAS")){

    ImageView logo = (ImageView)
        findViewById(R.id.Logo);
```

```
        logo.setImageResource(R.drawable.Local95);
    }
    else if(nom.equals("PUNTO EXTREMO")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local96);
    }
    else if(nom.equals("HELEN NOVEDADES")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local97);
    }
    else if(nom.equals("CELL MEP")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local98);
    }
    else if(nom.equals("LATIN TRAVEL")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local99100116);
    }
    else if(nom.equals("D KAREN")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local101);
    }
    else if(nom.equals("FACHA SPORT")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local102);
    }
    else if(nom.equals("BISUTERIA")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.is3);
    }
}
```

```
}

//NIVEL TRES
else if(nom.equals("NICELL")){

    ImageView logo = (ImageView)
        findViewById(R.id.Logo);
    logo.setImageResource(R.drawable.Local103);
}
else if(nom.equals("COMERSUR")){

    ImageView logo = (ImageView)
        findViewById(R.id.Logo);
    logo.setImageResource(R.drawable.Local104);
}
else if(nom.equals("ECUAVENTURA")){

    ImageView logo = (ImageView)
        findViewById(R.id.Logo);
    logo.setImageResource(R.drawable.Local105);
}
else if(nom.equals("TECNOLOGY CELL")){

    ImageView logo = (ImageView)
        findViewById(R.id.Logo);
    logo.setImageResource(R.drawable.Local106);
}
else if(nom.equals("LA TORRE")){

    ImageView logo = (ImageView)
        findViewById(R.id.Logo);
    logo.setImageResource(R.drawable.Local107);
}
else if(nom.equals("TOPY SPORT")){

    ImageView logo = (ImageView)
        findViewById(R.id.Logo);
    logo.setImageResource(R.drawable.Local108);
}
else if(nom.equals("ENSUENOS KAREN")){

    ImageView logo = (ImageView)
        findViewById(R.id.Logo);
```

```
        logo.setImageResource(R.drawable.Local109);
    }
    else if(nom.equals("ANIME MANIA")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local110);
    }
    else if(nom.equals("DIGITAL PRODUCTION")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local111);
    }
    else if(nom.equals("CONSUHID COURIER")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local112);
    }
    else if(nom.equals("COBO IMPORTACIONES")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local113);
    }
    else if(nom.equals("PUBLIPROMOCIONES")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local114);
    }
    else if(nom.equals("CITY CELL")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local115);
    }
    else if(nom.equals("LATIN TRAVEL")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local116);
    }
}
```



```
}  
else if(nom.equals("ISLA PINGUINO")){  
  
    ImageView logo = (ImageView)  
        findViewById(R.id.Logo);  
    logo.setImageResource(R.drawable.is4);  
}  
  
//NIVEL CUATRO  
else if(nom.equals("PERRO LOCO")){  
  
    ImageView logo = (ImageView)  
        findViewById(R.id.Logo);  
    logo.setImageResource(R.drawable.Local117);  
}  
else if(nom.equals("KMIZZ")){  
  
    ImageView logo = (ImageView)  
        findViewById(R.id.Logo);  
    logo.setImageResource(R.drawable.Local119);  
}  
else if(nom.equals("PUNTO DIGITAL")){  
  
    ImageView logo = (ImageView)  
        findViewById(R.id.Logo);  
    logo.setImageResource(R.drawable.Local120);  
}  
else if(nom.equals("MY PC")){  
  
    ImageView logo = (ImageView)  
        findViewById(R.id.Logo);  
    logo.setImageResource(R.drawable.Local121);  
}  
else if(nom.equals("PUBLIPROMOCIONES")){  
  
    ImageView logo = (ImageView)  
        findViewById(R.id.Logo);  
    logo.setImageResource(R.drawable.Local122);  
}  
else if(nom.equals("REPARACIONES DIEGO")){  
  
    ImageView logo = (ImageView)  
        findViewById(R.id.Logo);
```

```
        logo.setImageResource(R.drawable.Local123);
    }
    else if(nom.equals("NEW CAPCOM")){

        ImageView logo = (ImageView)
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local124);
    }
    else if(nom.equals("KAAS JUEGOS")){

        ImageView logo = (ImageView)
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local125);
    }
    else if(nom.equals("CUERO ARAMIS")){

        ImageView logo = (ImageView)
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local126);
    }
    else if(nom.equals("CAMARA ARTESANAL")){

        ImageView logo = (ImageView)
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local127);
    }
    else if(nom.equals("MARTIN CELL")){

        ImageView logo = (ImageView)
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local128);
    }
    else if(nom.equals("INGENIERIA CELULAR")){

        ImageView logo = (ImageView)
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local129);
    }
    else if(nom.equals("CALZADOS EMY")){

        ImageView logo = (ImageView)
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local130);
    }
}
```

```

    }
else if(nom.equals("KMIZZ")){

    ImageView logo = (ImageView)
        findViewById(R.id.Logo);
    logo.setImageResource(R.drawable.is5);
}

//NIVEL CINCO
else if(nom.equals("GAME CENTER")){

    ImageView logo = (ImageView)
        findViewById(R.id.Logo);
    logo.setImageResource(R.drawable.Local131);
}
else if(nom.equals("PLAY IT LOUD")){

    ImageView logo = (ImageView)
        findViewById(R.id.Logo);
    logo.setImageResource(R.drawable.Local132133);
}
else if(nom.equals("EC-ROBOTICS")){

    ImageView logo = (ImageView)
        findViewById(R.id.Logo);
    logo.setImageResource(R.drawable.Local134);
}
else if(nom.equals("CYBER COMP")){

    ImageView logo = (ImageView)
        findViewById(R.id.Logo);
    logo.setImageResource(R.drawable.Local135);
}
else if(nom.equals("TFD")){

    ImageView logo = (ImageView)
        findViewById(R.id.Logo);
    logo.setImageResource(R.drawable.Local136);
}
else if(nom.equals("FOLKLORE GREGORIO")){

    ImageView logo = (ImageView)
        findViewById(R.id.Logo);

```

```
        logo.setImageResource(R.drawable.Local137);
    }
else if(nom.equals("GRAFICAR PUBLICIDAD")){

        ImageView logo = (ImageView)
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local138);
    }
else if(nom.equals("SMILE BOUTIQ")){

        ImageView logo = (ImageView)
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local139);
    }
else if(nom.equals("ECUAVISION")){

        ImageView logo = (ImageView)
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local140);
    }
else if(nom.equals("ALTO TRAFICO")){

        ImageView logo = (ImageView)
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local141142);
    }
else if(nom.equals("SPAI SONS")){

        ImageView logo = (ImageView)
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local143);
    }
else if(nom.equals("YANBAL")){

        ImageView logo = (ImageView)
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local144);
    }
else if(nom.equals("CONFITERIA LA NENA")){

        ImageView logo = (ImageView)
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.is6);
    }
```

```
}

//NIVEL SEIS

else if(nom.equals("EXOTIC THAI")){

    ImageView logo = (ImageView)
        findViewById(R.id.Logo);
    logo.setImageResource(R.drawable.Local145);
}

else if(nom.equals("YNV TECHNOLOGY")){

    ImageView logo = (ImageView)
        findViewById(R.id.Logo);
    logo.setImageResource(R.drawable.Local146);
}

else if(nom.equals("LAS UNIAS A SOPLETE")){

    ImageView logo = (ImageView)
        findViewById(R.id.Logo);
    logo.setImageResource(R.drawable.Local147);
}

else if(nom.equals("ADMINISTRACION")){

    ImageView logo = (ImageView)
        findViewById(R.id.Logo);
    logo.setImageResource(R.drawable.Local148);
}

else if(nom.equals("CORPORACIONES UNIDAS")){

    ImageView logo = (ImageView)
        findViewById(R.id.Logo);
    logo.setImageResource(R.drawable.Local149);
}

else if(nom.equals("CAFU CIGARRILLOS")){

    ImageView logo = (ImageView)
        findViewById(R.id.Logo);
    logo.setImageResource(R.drawable.Local150);
}

else if(nom.equals("CELL PLANET")){

    ImageView logo = (ImageView)
```

```

        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local151);
    }
    else if(nom.equals("EROVER PC")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
            logo.setImageResource(R.drawable.Local152153154);
        }
    else if(nom.equals("TV.COM")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
            logo.setImageResource(R.drawable.Local155);
        }
    else if(nom.equals("VIZZON")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
            logo.setImageResource(R.drawable.Local156);
        }
    else if(nom.equals("ALHER TRAVEL")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
            logo.setImageResource(R.drawable.Local157);
        }
    }

//Detalles para el caso de impresoras
if(pro.equals(imp)){

txtProducto=(TextView)findViewById(R.id.inputProducto);
txtProducto.setText(product.getString(TAG_Producto));
txtPrecio=(TextView)findViewById(R.id.inputPrecio);
txtPrecio.setText(product.getString(TAG_Precio));
txtModelo=(TextView)findViewById(R.id.inputModelo);
txtModelo.setText(product.getString(TAG_Modelo));
txtVarios1 = (TextView) findViewById(R.id.inputDis);
txtVarios1.setText("Compatibilidad:");
txtDisco=(TextView)findViewById(R.id.inputDisco);
txtDisco.setText(product.getString(TAG_Sistema));
txtVarios2 = (TextView) findViewById(R.id.inputOtros1);
txtVarios2.setText("Cartuchos:");

```

```

txtProcesador=(TextView)findViewById(R.id.inputProcesador);
txtProcesador.setText(product.getString(TAG_Varios));
txtVarios3 = (TextView) findViewById(R.id.inputOtros1);
txtVarios3.setText("Velocidad de Impresión:");
txtProcesador=(TextView)findViewById(R.id.inputMemoria);
txtProcesador.setText(product.getString(TAG_Otros1));
txtVarios4 = (TextView) findViewById(R.id.inputOtros2);
txtVarios4.setText("Escaner:");
txtProcesador=(TextView)findViewById(R.id.inputCamara);
txtProcesador.setText(product.getString(TAG_Otros2));
txtVarios5 = (TextView) findViewById(R.id.inputOtros3);
txtVarios5.setText("Bluethoot:");
txtProcesador=(TextView)findViewById(R.id.inputSistema);
txtProcesador.setText(product.getString(TAG_Otros3));
txtVarios6 = (TextView) findViewById(R.id.inputOtros4);
txtVarios6.setText("Impresión CD:");
txtProcesador=(TextView)findViewById(R.id.inputPantalla);
txtProcesador.setText(product.getString(TAG_Otros4));
txtVarios7 = (TextView) findViewById(R.id.inputOtros5);
txtVarios7.setText("Otros:");
txtProcesador=(TextView)findViewById(R.id.inputOtros);
txtProcesador.setText(product.getString(TAG_Otros5));
txtStock=(TextView)findViewById(R.id.inputStock);
txtStock.setText(product.getString(TAG_Stock));
txtGarantia=(TextView)findViewById(R.id.inputGarantia);
txtGarantia.setText(product.getString(TAG_Garantia));
ImageView prod1 = (ImageView) findViewById(R.id.prod1);
prod1.setImageResource(R.drawable.impresora1);
ImageView prod2 = (ImageView) findViewById(R.id.prod2);
prod2.setImageResource(R.drawable.impresora2);
}
//Detalles para el caso de laptops
else if(pro.equals(lap)){
txtProducto=(TextView)findViewById(R.id.inputProducto);
txtProducto.setText(product.getString(TAG_Producto));
txtPrecio=(TextView)findViewById(R.id.inputPrecio);
txtPrecio.setText(product.getString(TAG_Precio));
txtModelo=(TextView)findViewById(R.id.inputModelo);
txtModelo.setText(product.getString(TAG_Modelo));
txtDisco=(TextView)findViewById(R.id.inputDisco);
txtDisco.setText(product.getString(TAG_Disco));
txtProcesador=(TextView)findViewById(R.id.inputProcesador);
txtProcesador.setText(product.getString(TAG_Procesador));

```

```

txtMemoria=(TextView)findViewById(R.id.inputMemoria);
txtMemoria.setText(product.getString(TAG_Memoria));
txtCamara=(TextView)findViewById(R.id.inputCamara);
txtCamara.setText(product.getString(TAG_Camara));
txtPantalla=(TextView)findViewById(R.id.inputPantalla);
txtPantalla.setText(product.getString(TAG_Pantalla));
txtSistema=(TextView)findViewById(R.id.inputSistema);
txtSistema.setText(product.getString(TAG_Sistema));
txtVarios=(TextView)findViewById(R.id.inputOtros);
txtVarios.setText(product.getString(TAG_Varios));
txtStock=(TextView)findViewById(R.id.inputStock);
txtStock.setText(product.getString(TAG_Stock));
txtGarantia=(TextView)findViewById(R.id.inputGarantia);
txtGarantia.setText(product.getString(TAG_Garantia));
ImageView prod1 = (ImageView) findViewById(R.id.prod1);
prod1.setImageResource(R.drawable.Laptop1);
ImageView prod2 = (ImageView) findViewById(R.id.prod2);
prod2.setImageResource(R.drawable.Laptop2);
}
}else{
}
} catch (JSONException e) {
e.printStackTrace();
}
}
});
return null;
}
/**
 * Después de completar la tarea en segundo plano disminuye el dialogo de progreso
 * **/
protected void onPostExecute(String file_url) {
pDialog.dismiss();
}
}
}

```

## B.8 CLASE OFERTA DETALLES

```

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

```



```

import org.apache.http.NameValuePair;
import org.apache.http.message.BasicNameValuePair;
import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;
import android.view.View;
import android.view.animation.Animation;
import android.view.animation.AnimationUtils;
import android.app.Activity;
import android.app.ProgressDialog;
import android.content.Intent;
import android.os.AsyncTask;
import android.os.Bundle;
import android.util.Log;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.TextView;

public class OfertaDetallesActivity extends Activity{
//Declaración de los textview a utilizarse
TextView
txtSitio,txtTelefono,txtNum,txtPiso,txtProducto,txtPrecio,txtModelo,txtProcesador,txtDis
co,txtMemoria,txtCamara,txtSistema,txtGarantia,txtStock,txtPantalla,txtVarios;
EditText txtejemplo;
TextView
txtVarios1,txtVarios2,txtVarios3,txtVarios4,txtVarios5,txtVarios6,txtVarios7,txtAnuncio,
txtPrecioOferta,txtDuracion;
//Declaración de los imageview a utilizarse
ImageView logo,prod1,prod2,logoC;
Button btnSave;
Button btnDelete;
String pid="";
//Dirección IP del servidor que contiene los servicios web a utilizarse
private static String ip = "172.30.35.22";
// Diálogo de progreso
private ProgressDialog pDialog;
// Declaración de la clase JSONParser
JSONParser jsonParser = new JSONParser();
// Url completa del servicio web que contiene la información
private static final String url_oferta_detalles =
"http://" + ip + "/xampp/phpfiles/get_oferta_detalles.php";

```

```

// nodos JSON
    private static final String TAG_SUCCESS = "success";
    private static final String TAG_OFERTA= "oferta";
    private static final String TAG_Nombre = "Nombre";
    private static final String TAG_Piso = "Piso";
    private static final String TAG_Telefono = "Telefono";
    private static final String TAG_Sitio = "Sitio";
    private static final String TAG_Producto = "Producto";
    private static final String TAG_Numero = "id_Local";
    private static final String TAG_Precio = "Precio";
    private static final String TAG_Modelo = "Modelo";
    private static final String TAG_Disco = "Disco";
    private static final String TAG_Procesador = "Procesador";
    private static final String TAG_Memoria = "Memoria";
    private static final String TAG_Camara = "Camara";
    private static final String TAG_Sistema = "SistemaOperativo";
    private static final String TAG_Garantia = "Garantia";
    private static final String TAG_StockOferta = "StockOferta";
    private static final String TAG_Pantalla = "Pantalla";
    private static final String TAG_Varios = "Varios";
    private static final String TAG_Otros1 = "Otros1";
    private static final String TAG_Otros2 = "Otros2";
    private static final String TAG_Otros3 = "Otros3";
    private static final String TAG_Otros4 = "Otros4";
    private static final String TAG_Otros5 = "Otros5";
    private static final String TAG_id_LocPro = "id_LocPro";
    private static final String TAG_PrecioOferta = "PrecioOferta";
    private static final String TAG_Duracion = "Duracion";

@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.ofertinfo);
    logo = (ImageView)findViewById(R.id.Logo);
    // Llama a la clase INTENT
    Intent i = getIntent();
    // Obtiene el ID del local enviado en la Actividad anterior
    pid = i.getStringExtra(TAG_id_LocPro);
    // Ejecuta la actividad en segundo plano GetProductoDetails
    new GetOfertasDetails().execute();
    //ANIMACIONES
    txtAnuncio=(TextView)findViewById(R.id.inputAnuncio);
    Animation trasladar = AnimationUtils.LoadAnimation(this,

```

```

        R.anim.trasladar);
        txtAnuncio.setAnimation(trasladar);
    }

    public void ActividadAtras(View v){

Intent i = new Intent(this, BusquedaOfertasActivity.class);
startActivity(i);
OfertaDetallesActivity.this.finish();
    }

    public void ActividadMenu(View v){
Intent i = new Intent(this, MainActivity.class);
startActivity(i);
OfertaDetallesActivity.this.finish();
    }

    public void ActividadSalir(View v){
        System.exit(0);
    }

    /**
     * Tarea en segundo plano obtiene los detalles de cada producto
     * */
    class GetOfertasDetails extends AsyncTask<String, String, String> {

        /**
         * Antes de empezar el hilo en segundo plano muestra el progress Dialog
         * */
        @Override
        protected void onPreExecute() {
            super.onPreExecute();
            pDialog = new ProgressDialog(OfertaDetallesActivity.this);
            pDialog.setMessage("Cargando Información Espere...");
            pDialog.setIndeterminate(false);
            pDialog.setCancelable(true);
            pDialog.show();
        }

        /**
         * Obtiene los detalles de las ofertas de la URL
         * */
        protected String doInBackground(String... params) {

```

```

// Actualiza el UI del hilo en segundo plano
runOnUiThread(new Runnable() {
    public void run() {

        int success;
        try {
            // Construcción de Parámetros
            List<NameValuePair> params = new ArrayList<NameValuePair>();
            params.add(new BasicNameValuePair("id_LocPro", pid));
//Obtiene el detalle de los productos utilizando una petición GET
de la clase JSON
            JSONObject json = jsonParser.makeHttpRequest(
                url_oferta_detalles, "GET", params);
            // Chequea la respuesta JSON
            Log.d("Single Product Details", json.toString());
            // Chequea el resultado correcto
            success = json.getInt(TAG_SUCCESS);
            if (success == 1) {
                // En el caso de ser exitoso
                JSONArray productObj = json
                    .getJSONArray(TAG_OFERTA); // Arreglo JSON

                // Obtiene el primer objeto del arreglo JSON
                JSONObject product = productObj.getJSONObject(0);
//Almacena los datos correspondientes producto y su nombre
en las variables tipo string pro y nom
                String pro = product.getString(TAG_Producto).toString();
                String nom = product.getString(TAG_Nombre).toString();
                String imp= "IMPRESORA";
                String lap= "LAPTOP";

//Obtiene los identificadores de los textview utilizados
                txtPiso = (TextView) findViewById(R.id.inputPiso);
                txtNum = (TextView) findViewById(R.id.inputLocal);
                txtSitio = (TextView) findViewById(R.id.inputSitio);
                txtTelefono=(TextView)findViewById(R.id.inputTelefono);
//Muestra los datos de piso,sitio,teléfono,número de local
en los text view
                txtPiso.setText(product.getString(TAG_Piso));
                txtNum.setText(product.getString(TAG_Numero));
                txtSitio.setText(product.getString(TAG_Sitio));
                txtTelefono.setText(product.getString(TAG_Telefono));

```

```
//CARGA LOS LOGOS DE CADA LOCAL
//SUBSUELO DOS
if(nom.equals("ESTUDIO CERO")){

    ImageView logo = (ImageView)
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local2);

}
else if(nom.equals("MARCESPLACE")){

    ImageView logo = (ImageView)
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local3);
}
else if(nom.equals("PUBLIPROMOCIONES")){

    ImageView logo = (ImageView)
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local4);
}
else if(nom.equals("DIGITAL GAMES")){

    ImageView logo = (ImageView)
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local536);
}
else if(nom.equals("RING SITE")){

    ImageView logo = (ImageView)
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local6);
}
else if(nom.equals("PALOMINO CENTRO ORTOPEDICO")){

    ImageView logo = (ImageView)
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local7);
}
else if(nom.equals("HOUSE OF STYLE")){
```

```
ImageView logo = (ImageView)
    findViewById(R.id.Logo);
    logo.setImageResource(R.drawable.Local8);
}
else if(nom.equals("REMI CENTER")){

ImageView logo = (ImageView)
    findViewById(R.id.Logo);
    logo.setImageResource(R.drawable.Local9);
}
else if(nom.equals("MULTICELL")){

ImageView logo = (ImageView)
    findViewById(R.id.Logo);
    logo.setImageResource(R.drawable.Local10);
}
else if(nom.equals("MIS LAPTOPS")){

ImageView logo = (ImageView)
    findViewById(R.id.Logo);
    logo.setImageResource(R.drawable.Local12);
}
else if(nom.equals("ARTESANIAS TROYA")){

ImageView logo = (ImageView)
    findViewById(R.id.Logo);
    logo.setImageResource(R.drawable.Local1314);
}
else if(nom.equals("CORED SYSTEMS")){

ImageView logo = (ImageView)
    findViewById(R.id.Logo);
    logo.setImageResource(R.drawable.Local15);
}
else if(nom.equals("PUNTO SISTEMAS")){

ImageView logo = (ImageView)
    findViewById(R.id.Logo);
    logo.setImageResource(R.drawable.Local16);
}
else if(nom.equals("SIA MARKET")){

ImageView logo = (ImageView)
```

```
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local17);
    }
    else if(nom.equals("VIDEO MUSICAL")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
            logo.setImageResource(R.drawable.Local19);
    }
    else if(nom.equals("MUNDO SYST")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
            logo.setImageResource(R.drawable.Local20);
    }
    else if(nom.equals("POWER NET")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
            logo.setImageResource(R.drawable.Local2124);
    }
    else if(nom.equals("NOVICOMPU")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
            logo.setImageResource(R.drawable.novicompu);
    }
    else if(nom.equals("COMPU KIT")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
            logo.setImageResource(R.drawable.Local23);
    }
    else if(nom.equals("GIGA SOFT")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
            logo.setImageResource(R.drawable.Local25);
    }
    else if(nom.equals("LABUTH SHAH")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
```

```
        logo.setImageResource(R.drawable.Local2627);
    }
    else if(nom.equals("DHE SERIES STORE")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local28);
    }
    else if(nom.equals("D&S ABOGADOS")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local29);
    }
    else if(nom.equals("Peluqueria MAXBER")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local30);
    }
    else if(nom.equals("AQUINO OPTICA")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local31);
    }
    else if(nom.equals("MOVIEZONE")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local32);
    }
    else if(nom.equals("UNIVERSAL TECHNOLOGY")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local33);
    }
    else if(nom.equals("COMPUKIT")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local34);
    }
}
```



```
}  
else if(nom.equals("SAGA TURS")){  
  
    ImageView logo = (ImageView)  
        findViewById(R.id.Logo);  
    logo.setImageResource(R.drawable.Local35);  
}  
else if(nom.equals("CEVICHERIA SACA LA RESACA")){  
  
    ImageView logo = (ImageView)  
        findViewById(R.id.Logo);  
    logo.setImageResource(R.drawable.Local3738);  
}  
  
//SUSBSUELO UNO  
else if(nom.equals("CALL CELL")){  
  
    ImageView logo = (ImageView)  
        findViewById(R.id.Logo);  
    logo.setImageResource(R.drawable.Local3940);  
}  
else if(nom.equals("KONAMI GAMES")){  
  
    ImageView logo = (ImageView)  
        findViewById(R.id.Logo);  
    logo.setImageResource(R.drawable.Local41);  
}  
else if(nom.equals("MAREA NEGRA")){  
  
    ImageView logo = (ImageView)  
        findViewById(R.id.Logo);  
    logo.setImageResource(R.drawable.Local42);  
}  
else if(nom.equals("PELUQUERIA LIZ")){  
  
    ImageView logo = (ImageView)  
        findViewById(R.id.Logo);  
    logo.setImageResource(R.drawable.Local43);  
}  
else if(nom.equals("CABANA IMPORT")){  
  
    ImageView logo = (ImageView)  
        findViewById(R.id.Logo);  
    logo.setImageResource(R.drawable.Local44);  
}
```

```
}  
else if(nom.equals("HOUSE OF STYLE")){  
  
    ImageView logo = (ImageView)  
        findViewById(R.id.Logo);  
    logo.setImageResource(R.drawable.Local45);  
}  
else if(nom.equals("MEGA UNIFORMES")){  
  
    ImageView logo = (ImageView)  
        findViewById(R.id.Logo);  
    logo.setImageResource(R.drawable.Local4648);  
}  
else if(nom.equals("RCM SISTEMAS")){  
  
    ImageView logo = (ImageView)  
        findViewById(R.id.Logo);  
    logo.setImageResource(R.drawable.Local47);  
}  
else if(nom.equals("CASA BOLIVIA")){  
  
    ImageView logo = (ImageView)  
        findViewById(R.id.Logo);  
    logo.setImageResource(R.drawable.Local4950);  
}  
else if(nom.equals("Peluqueria MAXBER")){  
  
    ImageView logo = (ImageView)  
        findViewById(R.id.Logo);  
    logo.setImageResource(R.drawable.Local5152);  
}  
else if(nom.equals("PRONATUS")){  
  
    ImageView logo = (ImageView)  
        findViewById(R.id.Logo);  
    logo.setImageResource(R.drawable.Local53);  
}  
else if(nom.equals("PRO DJ")){  
  
    ImageView logo = (ImageView)  
        findViewById(R.id.Logo);  
    logo.setImageResource(R.drawable.Local545556);  
}
```

```
else if(nom.equals("JN COLLECTION")){

    ImageView logo = (ImageView)
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local57);
    }
else if(nom.equals("CORE.COM")){

    ImageView logo = (ImageView)
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local58);
    }
else if(nom.equals("SACOS")){

    ImageView logo = (ImageView)
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local59);
    }
else if(nom.equals("CUERO ARAMIS")){

    ImageView logo = (ImageView)
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local60);
    }
//PLANTA BAJA
else if(nom.equals("JM ROPA")){

    ImageView logo = (ImageView)
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local6162);
    }
else if(nom.equals("JOYERIA CEDILLO")){

    ImageView logo = (ImageView)
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local63);
    }
else if(nom.equals("CAMISERIA INGLESA")){

    ImageView logo = (ImageView)
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local64);
    }
}
```

```
else if(nom.equals("MAQUENO REPUBLIC")){

    ImageView logo = (ImageView)
        findViewById(R.id.Logo);
    logo.setImageResource(R.drawable.Local6569);
}
else if(nom.equals("JOYERIA LA MINA")){

    ImageView logo = (ImageView)
        findViewById(R.id.Logo);
    logo.setImageResource(R.drawable.Local66);
}
else if(nom.equals("UNITECH")){

    ImageView logo = (ImageView)
        findViewById(R.id.Logo);
    logo.setImageResource(R.drawable.Local67);
}
else if(nom.equals("MUNDO DVD")){

    ImageView logo = (ImageView)
        findViewById(R.id.Logo);
    logo.setImageResource(R.drawable.Local68);
}
else if(nom.equals("CAPCOM")){

    ImageView logo = (ImageView)
        findViewById(R.id.Logo);
    logo.setImageResource(R.drawable.Local70);
}
else if(nom.equals("BRUSEHNKO")){

    ImageView logo = (ImageView)
        findViewById(R.id.Logo);
    logo.setImageResource(R.drawable.Local71);
}
else if(nom.equals("JOYERIA GRIJALBA")){

    ImageView logo = (ImageView)
        findViewById(R.id.Logo);
    logo.setImageResource(R.drawable.Local72);
}
else if(nom.equals("OXEN JEANS")){
```

```
        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local73);
    }
    else if(nom.equals("POSDATA")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local74);
    }
    else if(nom.equals("PUNTO DIGITAL")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.is1);
    }
    //NIVEL UNO

    else if(nom.equals("MOVIE LIGHT")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local75);
    }
    else if(nom.equals("LABUTH SHAH 3")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local7679);
    }
    else if(nom.equals("TIAN SPORT")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local77);
    }
    else if(nom.equals("AMERICAN GAMES")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local78);
    }
}
```

```
else if(nom.equals("AUDIO CENTER")){

    ImageView logo = (ImageView)
    findViewById(R.id.Logo);
    logo.setImageResource(R.drawable.Local80);
}
else if(nom.equals("LEVEL UP")){
    ImageView logo = (ImageView)
    findViewById(R.id.Logo);
    logo.setImageResource(R.drawable.Local81);
}
else if(nom.equals("ALMACEN ARENA")){

    ImageView logo = (ImageView)
    findViewById(R.id.Logo);
    logo.setImageResource(R.drawable.Local82);
}
else if(nom.equals("FABRICUERO")){

    ImageView logo = (ImageView)
    findViewById(R.id.Logo);
    logo.setImageResource(R.drawable.Local83);
}
else if(nom.equals("ALMACENES DALI")){

    ImageView logo = (ImageView)
    findViewById(R.id.Logo);
    logo.setImageResource(R.drawable.Local84);
}
else if(nom.equals("CELLCOM")){

    ImageView logo = (ImageView)
    findViewById(R.id.Logo);
    logo.setImageResource(R.drawable.Local85);
}
else if(nom.equals("TAMASHAN")){

    ImageView logo = (ImageView)
    findViewById(R.id.Logo);
    logo.setImageResource(R.drawable.Local86);
}
else if(nom.equals("ANALUISA JOYEROS")){
```

```
        ImageView logo = (ImageView)
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local87);
    }
else if(nom.equals("ISLA PANYOCHU")){

    ImageView logo = (ImageView)
    findViewById(R.id.Logo);
    logo.setImageResource(R.drawable.is2);
}

//NIVEL DOS
else if(nom.equals("STAR PLATERIA")){

    ImageView logo = (ImageView)
    findViewById(R.id.Logo);
    logo.setImageResource(R.drawable.Local89);
}
else if(nom.equals("THE HOUSE ROCK")){

    ImageView logo = (ImageView)
    findViewById(R.id.Logo);
    logo.setImageResource(R.drawable.Local90);
}
else if(nom.equals("TECNOCABLE")){

    ImageView logo = (ImageView)
    findViewById(R.id.Logo);
    logo.setImageResource(R.drawable.Local91);
}
else if(nom.equals("NOVEDADES ROSE")){

    ImageView logo = (ImageView)
    findViewById(R.id.Logo);
    logo.setImageResource(R.drawable.Local92);
}
else if(nom.equals("JOYERIA AMATISTA")){

    ImageView logo = (ImageView)
    findViewById(R.id.Logo);
    logo.setImageResource(R.drawable.Local93);
}
else if(nom.equals("EL MAGICO BAZAR")){
```

```
        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local94);
    }
    else if(nom.equals("DE ONDAS")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local95);
    }
    else if(nom.equals("PUNTO EXTREMO")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local96);
    }
    else if(nom.equals("HELEN NOVEDADES")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local97);
    }
    else if(nom.equals("CELL MEP")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local98);
    }
    else if(nom.equals("LATIN TRAVEL")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local99100116);
    }
    else if(nom.equals("D KAREN")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local101);
    }
    else if(nom.equals("FACHA SPORT")){
```



```
        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local102);
    }
    else if(nom.equals("BISUTERIA")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.is3);
    }

    //NIVEL TRES
    else if(nom.equals("NICELL")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local103);
    }
    else if(nom.equals("COMERSUR")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local104);
    }
    else if(nom.equals("ECUAVENTURA")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local105);
    }
    else if(nom.equals("TECNOLOGY CELL")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local106);
    }
    else if(nom.equals("LA TORRE")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local107);
    }
    else if(nom.equals("TOPY SPORT")){
```

```
        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local108);
    }
    else if(nom.equals("ENSUENOS KAREN")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local109);
    }
    else if(nom.equals("ANIME MANIA")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local110);
    }
    else if(nom.equals("DIGITAL PRODUCTION")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local111);
    }
    else if(nom.equals("CONSUHID COURIER")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local112);
    }
    else if(nom.equals("COBO IMPORTACIONES")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local113);
    }
    else if(nom.equals("PUBLIPROMOCIONES")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local114);
    }
    else if(nom.equals("CITY CELL")){
```

```
        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local115);
    }
    else if(nom.equals("LATIN TRAVEL")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local116);
    }
    else if(nom.equals("ISLA PINGUINO")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.is4);
    }

    //NIVEL CUATRO
    else if(nom.equals("PERRO LOCO")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local117);
    }
    else if(nom.equals("KMIZZ")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local119);
    }
    else if(nom.equals("PUNTO DIGITAL")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local120);
    }
    else if(nom.equals("MY PC")){

        ImageView logo = (ImageView)
            findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local121);
    }
    else if(nom.equals("PUBLIPROMOCIONES")){
```

```
        ImageView logo = (ImageView)
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local122);
    }
    else if(nom.equals("REPARACIONES DIEGO")){

        ImageView logo = (ImageView)
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local123);
    }
    else if(nom.equals("NEW CAPCOM")){

        ImageView logo = (ImageView)
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local124);
    }
    else if(nom.equals("KAAS JUEGOS")){

        ImageView logo = (ImageView)
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local125);
    }
    else if(nom.equals("CUERO ARAMIS")){

        ImageView logo = (ImageView)
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local126);
    }
    else if(nom.equals("CAMARA ARTESANAL")){

        ImageView logo = (ImageView)
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local127);
    }
    else if(nom.equals("MARTIN CELL")){

        ImageView logo = (ImageView)
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local128);
    }
    else if(nom.equals("INGENIERIA CELULAR")){
```

```
        ImageView logo = (ImageView)
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local129);
    }
else if(nom.equals("CALZADOS EMY")){

        ImageView logo = (ImageView)
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local130);
    }
else if(nom.equals("KMIZZ")){

        ImageView logo = (ImageView)
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.is5);
    }

//NIVEL CINCO
else if(nom.equals("GAME CENTER")){

        ImageView logo = (ImageView)
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local131);
    }
else if(nom.equals("PLAY IT LOUD")){

        ImageView logo = (ImageView)
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local132133);
    }
else if(nom.equals("EC-ROBOTICS")){

        ImageView logo = (ImageView)
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local134);
    }
else if(nom.equals("CYBER COMP")){

        ImageView logo = (ImageView)
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local135);
    }
else if(nom.equals("TFD")){
```

```
        ImageView logo = (ImageView)
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local136);
    }
    else if(nom.equals("FOLKLORE GREGORIO")){

        ImageView logo = (ImageView)
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local137);
    }
    else if(nom.equals("GRAFICAR PUBLICIDAD")){

        ImageView logo = (ImageView)
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local138);
    }
    else if(nom.equals("SMILE BOUTIQ")){

        ImageView logo = (ImageView)
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local139);
    }
    else if(nom.equals("ECUAVISION")){

        ImageView logo = (ImageView)
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local140);
    }
    else if(nom.equals("ALTO TRAFICO")){

        ImageView logo = (ImageView)
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local141142);
    }
    else if(nom.equals("SPAI SONS")){

        ImageView logo = (ImageView)
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local143);
    }
    else if(nom.equals("YANBAL")){
```

```
        ImageView logo = (ImageView)
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local144);
    }
else if(nom.equals("CONFITERIA LA NENA")){

    ImageView logo = (ImageView)
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.is6);
    }

//NIVEL SEIS

else if(nom.equals("EXOTIC THAI")){

    ImageView logo = (ImageView)
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local145);
    }
else if(nom.equals("YNV TECHNOLOGY")){

    ImageView logo = (ImageView)
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local146);
    }
else if(nom.equals("LAS UNIAS A SOPLETE")){

    ImageView logo = (ImageView)
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local147);
    }
else if(nom.equals("ADMINISTRACION")){

    ImageView logo = (ImageView)
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local148);
    }
else if(nom.equals("CORPORACIONES UNIDAS")){

    ImageView logo = (ImageView)
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local149);
    }
}
```

```

else if(nom.equals("CAFU CIGARRILLOS")){

    ImageView logo = (ImageView)
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local150);
    }
else if(nom.equals("CELL PLANET")){
    ImageView logo = (ImageView)
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local151);
    }
else if(nom.equals("EROVER PC")){

    ImageView logo = (ImageView)
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local152153154);
    }
else if(nom.equals("TV.COM")){

    ImageView logo = (ImageView)
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local155);
    }
else if(nom.equals("VIZZON")){

    ImageView logo = (ImageView)
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local156);
    }
else if(nom.equals("ALHER TRAVEL")){

    ImageView logo = (ImageView)
        findViewById(R.id.Logo);
        logo.setImageResource(R.drawable.Local157);
    }

//Muestra los detalles de impresoras
if(pro.equals(imp)){

txtProducto=(TextView)findViewById(R.id.inputProducto);
        txtProducto.setText(product.getString(TAG_Producto));

```



```
txtPrecio=(TextView)findViewById(R.id.inputPrecio);
txtPrecio.setText(product.getString(TAG_Precio));

txtModelo=(TextView)findViewById(R.id.inputModelo);
txtModelo.setText(product.getString(TAG_Modelo));

txtVarios1 = (TextView) findViewById(R.id.inputDis);
txtVarios1.setText("Compatibilidad:");
txtDisco=(TextView)findViewById(R.id.inputDisco);
txtDisco.setText(product.getString(TAG_Sistema));

txtVarios2 = (TextView)
findViewById(R.id.inputOtros1);
txtVarios2.setText("Cartuchos:");

txtProcesador=(TextView)findViewById(R.id.inputProcesador);
txtProcesador.setText(product.getString(TAG_Varios));

txtVarios3 = (TextView)
findViewById(R.id.inputOtros1);
txtVarios3.setText("Velocidad de Impresión:");

txtProcesador=(TextView)findViewById(R.id.inputMemoria);
txtProcesador.setText(product.getString(TAG_Otros1));

txtVarios4 = (TextView)
findViewById(R.id.inputOtros2);
txtVarios4.setText("Escaner:");

txtProcesador=(TextView)findViewById(R.id.inputCamara);
txtProcesador.setText(product.getString(TAG_Otros2));

txtVarios5 = (TextView)
findViewById(R.id.inputOtros3);
txtVarios5.setText("Bluetooth:");

txtProcesador=(TextView)findViewById(R.id.inputSistema);
txtProcesador.setText(product.getString(TAG_Otros3));

txtVarios6 = (TextView)
findViewById(R.id.inputOtros4);
txtVarios6.setText("Impresión CD:");
```

```

txtProcesador=(TextView)findViewById(R.id.inputPantalla);
txtProcesador.setText(product.getString(TAG_Otros4));

txtVarios7 = (TextView)
findViewById(R.id.inputOtros5);
txtVarios7.setText("Otros:");

txtProcesador=(TextView)findViewById(R.id.inputOtros);
txtProcesador.setText(product.getString(TAG_Otros5));

txtStock=(TextView)findViewById(R.id.inputStockOferta);
txtStock.setText(product.getString(TAG_StockOferta));

txtGarantia=(TextView)findViewById(R.id.inputGarantia);
txtGarantia.setText(product.getString(TAG_Garantia));

ImageView prod1 = (ImageView)
findViewById(R.id.prod1);
prod1.setImageResource(R.drawable.impresora1);

ImageView prod2 = (ImageView) findViewById(R.id.prod2);
prod2.setImageResource(R.drawable.impresora2);

}

//muestra los detalles de laptops
else if(pro.equals(lap)){

txtProducto=(TextView)findViewById(R.id.inputProducto);
txtProducto.setText(product.getString(TAG_Producto));

txtPrecio=(TextView)findViewById(R.id.inputPrecio);
txtPrecio.setText(product.getString(TAG_Precio));

txtModelo=(TextView)findViewById(R.id.inputModelo);
txtModelo.setText(product.getString(TAG_Modelo));

txtDisco=(TextView)findViewById(R.id.inputDisco);
txtDisco.setText(product.getString(TAG_Disco));

```

```
txtProcesador=(TextView)findViewById(R.id.inputProcesador);

txtProcesador.setText(product.getString(TAG_Procesador));

        txtMemoria=(TextView)findViewById(R.id.inputMemoria);
        txtMemoria.setText(product.getString(TAG_Memoria));
txtCamara=(TextView)findViewById(R.id.inputCamara);
txtCamara.setText(product.getString(TAG_Camara));

txtPantalla=(TextView)findViewById(R.id.inputPantalla);
txtPantalla.setText(product.getString(TAG_Pantalla));

txtSistema=(TextView)findViewById(R.id.inputSistema);
txtSistema.setText(product.getString(TAG_Sistema));

txtVarios=(TextView)findViewById(R.id.inputOtros);
txtVarios.setText(product.getString(TAG_Varios));

txtGarantia=(TextView)findViewById(R.id.inputGarantia);
txtGarantia.setText(product.getString(TAG_Garantia));

txtStock=(TextView)findViewById(R.id.inputStockOferta);
txtStock.setText(product.getString(TAG_StockOferta));

txtPrecioOferta=(TextView)findViewById(R.id.inputPrecioOferta);
txtPrecioOferta.setText(product.getString(TAG_PrecioOferta));
txtDuracion=(TextView)findViewById(R.id.inputDuracion);
txtDuracion.setText(product.getString(TAG_Duracion));

ImageView prod1 = (ImageView) findViewById(R.id.prod1);
        prod1.setImageResource(R.drawable.Laptop1);

        ImageView prod2 = (ImageView) findViewById(R.id.prod2);
        prod2.setImageResource(R.drawable.Laptop2);
    }

    }else{

    }

} catch (JSONException e) {
    e.printStackTrace();
}
```

```

        }
    }
});

    return null;
}

/**
 * Despues de completar la tarea en segundo plano disminuye el progress dialog
 * **/
protected void onPostExecute(String file_url) {

    pDialog.dismiss();
}
}
}
}

```

## B.9 CLASE JSON PARSER

```

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.UnsupportedEncodingException;
import java.util.List;

import org.apache.http.HttpEntity;
import org.apache.http.HttpResponse;
import org.apache.http.NameValuePair;
import org.apache.http.client.ClientProtocolException;
import org.apache.http.client.entity.UrlEncodedFormEntity;
import org.apache.http.client.methods.HttpGet;
import org.apache.http.client.methods.HttpPost;
import org.apache.http.client.utils.URLEncodedUtils;
import org.apache.http.impl.client.DefaultHttpClient;
import org.json.JSONException;
import org.json.JSONObject;

import android.util.Log;

public class JSONParser {
    static InputStream is = null;
    static JSONObject jsonObj = null;
    static String json = "";

    // constructor
    public JSONParser() {

    }

    // function get json from url
    // by making HTTP POST or GET mehtod
    public JSONObject makeHttpRequest(String url, String method,
        List<NameValuePair> params) {

        // Making HTTP request

```

```

try {
    // check for request method
    if(method == "POST"){
        // request method is POST
        // defaultHttpClient
        DefaultHttpClient httpClient = new DefaultHttpClient();
        HttpPost httpPost = new HttpPost(url);
        httpPost.setEntity(new UrlEncodedFormEntity(params));

        HttpResponse httpResponse = httpClient.execute(httpPost);
        HttpEntity httpEntity = httpResponse.getEntity();
        is = httpEntity.getContent();

    }else if(method == "GET"){
        // request method is GET
        DefaultHttpClient httpClient = new DefaultHttpClient();
        String paramString = URLEncodedUtils.format(params, "utf-8");
        url += "?" + paramString;
        HttpGet httpGet = new HttpGet(url);

        HttpResponse httpResponse = httpClient.execute(httpGet);
        HttpEntity httpEntity = httpResponse.getEntity();
        is = httpEntity.getContent();
    }

} catch (UnsupportedEncodingException e) {
    e.printStackTrace();
} catch (ClientProtocolException e) {
    e.printStackTrace();
} catch (IOException e) {
    e.printStackTrace();
}

try {
    BufferedReader reader = new BufferedReader(new InputStreamReader(
        is, "iso-8859-1"), 8);
    StringBuilder sb = new StringBuilder();
    String line = null;
    while ((line = reader.readLine()) != null) {
        sb.append(line + "\n");
    }
    is.close();
    json = sb.toString();
} catch (Exception e) {
    Log.e("Buffer Error", "Error converting result " + e.toString());
}

// try parse the string to a JSON object
try {
    jsonObj = new JSONObject(json);
} catch (JSONException e) {
    Log.e("JSON Parser", "Error parsing data " + e.toString());
}

// return JSON String
return jsonObj;
}
}

```

## B.10 CLASE BUZON ACTIVITY

```

public class BuzonActivity extends Activity {
    // Dialogo de progreso
    private ProgressDialog pDialog;
    //Declaración de imageView, textview y edittext utilizados
    ImageView enviar;
    TextView num;
    EditText Nombre,Mail,Mensaje;
    //Variables tipo string utilizadas
    String nombre,mail,mensaje;
    String CComercial= "";
    String idCliente="";
    //Dirección IP del servidor que contiene los servicios web a utilizarse
    private static String ip = "172.30.35.22";
    // Crea el objeto JSON Parser
    JSONParser jParser = new JSONParser();

    ArrayList<HashMap<String, String>> clientList;
    // Url completa del servicio web el que enviara los datos hacia la base
    private static String url_clientes =
"http://" + ip + "/xampp/phpfiles/update_clientes.php";
    private static String url_todosclientes =
"http://" + ip + "/xampp/phpfiles/get_clientes.php";
    // Nombra los nodos JSON
    private static final String TAG_SUCCESS = "success";
    private static final String TAG_Clientes = "cliente";
    private static final String TAG_idCliente = "id_Cliente";
    private static final String TAG_idCComercial = "id_CComercial";
    private static final String TAG_Nombre = "Nombre";
    private static final String TAG_Mail = "Mail";
    private static final String TAG_Mensaje = "Mensaje";

    // Declaración del arreglo JSON
    JSONArray clientes = null;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.buzon);

        //Obtiene los ID de EditText y ImageView
        Nombre = (EditText) findViewById (R.id.inputNombre);
        Mail = (EditText) findViewById (R.id.inputMail);
        Mensaje = (EditText) findViewById (R.id.inputSugerencia);
        enviar=(ImageView)findViewById(R.id.btnEnviar);

        // Declaración del hashmap o conjunto de objetos
        clientList = new ArrayList<HashMap<String, String>>();

        //Ejecuta la actividad LoadALClientes
        new LoadAllClientes().execute();

        //Boton enviar
        enviar.setOnClickListener(new View.OnClickListener() {

            @Override
            public void onClick(View arg0) {

                new SaveDatos().execute();
                BuzonActivity.this.finish();
            }
        });
    }
}

```

```

    });
}
}
//Boton Atras
public void ActividadAtras(View v){

    Intent i = new Intent(this, MainActivity.class);//Creamos un nuevo intent para
llamar a la siguiente actividad
    startActivity(i);//Ejecutamos la actividad para que muestre la segunda actividad
    BuzonActivity.this.finish();
}
//Boton Menu
public void ActividadMenu(View v){

    Intent i = new Intent(this, MainActivity.class);//Creamos un nuevo intent para
llamar a la siguiente actividad
    startActivity(i);//Ejecutamos la actividad para que muestre la segunda actividad
    BuzonActivity.this.finish();
}
//Boton Salir
public void ActividadSalir(View v){
    System.exit(0);
}
}
/**
 * Tarea en segundo plano que guarda la sugerencia del cliente
 * */
class SaveDatos extends AsyncTask<String, String, String> {

    /**
     * Antes de empezar el hilo en segundo plano muestra el diálogo
     * de progreso
     * */
    @Override
    protected void onPreExecute() {
        super.onPreExecute();
        pDialog = new ProgressDialog(BuzonActivity.this);
        pDialog.setMessage("Enviando ...");
        pDialog.setIndeterminate(false);
        pDialog.setCancelable(true);
        pDialog.show();
    }

    /**
     * Guarda la sugerencia
     * */
    protected String doInBackground(String... args) {

        // Obtiene los datos ingresados en los edit text
        String nombre = Nombre.getText().toString();
        String mail = Mail.getText().toString();
        String mensaje = Mensaje.getText().toString();

        CComercial="CESP";
        //Obtiene el numero de clientes de la lista
        int n = clientList.size();
        int t = n;
        //Aumenta en uno para que el numero obtenido sea el ID del
cliente al ingresar a la base
        int r = t +1;
        String num = "" + r;
        String res = (num);
        idCliente=res;
        // Construcción de los parámetros a enviar

```

```

        List<NameValuePair> params = new ArrayList<NameValuePair>();
        params.add(new BasicNameValuePair(TAG_idCliente,
idCliente.toString()));
        params.add(new BasicNameValuePair(TAG_idCComercial,
CComercial.toString()));
        params.add(new BasicNameValuePair(TAG_Nombre,nombre));
        params.add(new BasicNameValuePair(TAG_Mail,mail));
        params.add(new BasicNameValuePair(TAG_Mensaje,mensaje));

        // Envia los datos mediante el metodo POST a la url
        // correspondiente y a la base de datos
        JSONObject json = jParser.makeHttpRequest(url_clientes,
            "POST", params);

        // Chequea el envio exitoso
        try {
            int success = json.getInt(TAG_SUCCESS);

            if (success == 1) {

                Intent i = getIntent();
                // Envia el resultado con codigo 100 para
                // notificar que el producto fue enviado
                setResult(100, i);
                finish();
            } else {

            }

        } catch (JSONException e) {
            e.printStackTrace();
        }

        return null;
    }

    /**
     * Despues de completar la tarea disminuye el dialogo de
     * progreso
     * **/
    protected void onPostExecute(String file_url) {
        // dismiss the dialog once product uupdated
        pDialog.dismiss();
    }

    /**
     * Tarea en segundo plano para obtener la lista de usuarios que
     * ingresaron su sugerencia
     * **/
    class LoadAllClientes extends AsyncTask<String, String, String> {

        /**
         * Opteniendo todos los clientes de la url
         * **/
        @Override
        protected String doInBackground(String... args) {
            // Contrucción de parámetros
            CComercial="\""+CESP+"\"";
            List<NameValuePair> params = new ArrayList<NameValuePair>();
            params.add(new BasicNameValuePair("id_CComercial", CComercial));

            // Obtiene el string JSON de la url

```



```

JSONObject json = jParser.makeHttpRequest(url_todosclientes,
"GET", params);

// Chequea la respuesta JSON correspondiente a All Clientes
Log.d("All Clientes: ", json.toString());

try {
    // Chequea el resultado exitoso
    int success = json.getInt(TAG_SUCCESS);

    if (success == 1) {
        // Clientes encontrados
        // Obtiene el arreglo de clientes
        clientes = json.getJSONArray(TAG_Clientes);

        //Mapea todos los clientes
        for (int i = 0; i < clientes.length(); i++) {
            JSONObject c = clientes.getJSONObject(i);

            // Almacena cada item JSON en la variable
            // correspondiente
            String id = c.getString(TAG_idCliente);
            String name = c.getString(TAG_Nombre);

            // Crea un nuevo hashmap
            HashMap<String, String> map = new HashMap<String,
            String>();

            // Añade cada hijo al nodo Hashmap con su
            // correspondiente valor
            map.put(TAG_idCliente, id);
            map.put(TAG_Nombre, name);

            // Añade los datos a la lista correspondiente
            clientList.add(map);

        }
    }

} catch (JSONException e) {
    e.printStackTrace();
}

return null;
}

```

## B.11 CLASE MAPA ACTIVITY

```

import com.google.android.gms.maps.CameraUpdateFactory;
import com.google.android.gms.maps.GoogleMap;
import com.google.android.gms.maps.GoogleMap.OnMapClickListener;
import com.google.android.gms.maps.model.BitmapDescriptorFactory;
import com.google.android.gms.maps.model.LatLng;
import com.google.android.gms.maps.model.MarkerOptions;
import com.google.android.gms.maps.SupportMapFragment;
import android.os.Bundle;

```

```

import android.support.v4.app.FragmentActivity;
import android.view.View;

public class MapaActivity extends FragmentActivity implements OnMapClickListener{
    //Coordenadas latitud, longitud del Centro Comercial Espiral
    private final LatLng ESP = new LatLng(-0.206599, -78.495901);
    //Declaración de la variable GoogleMap
    private GoogleMap mapa;

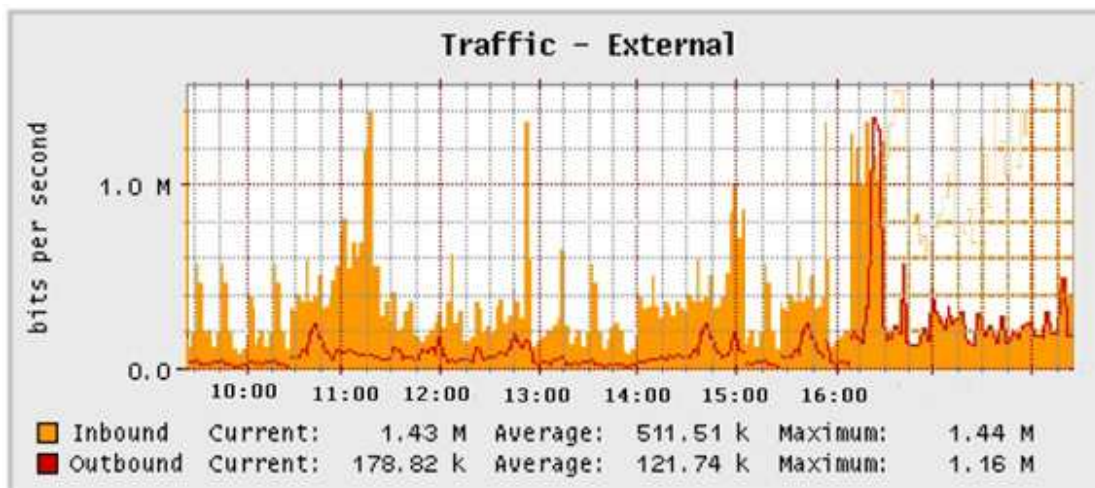
    @Override protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.mapasitio);
        //Declaración del fragment con la clase SupportMapFragment la
        //misma que contendrá la vista del mapa
        mapa =
        ((SupportMapFragment) getSupportFragmentManager().findFragmentById(R.id.mapes)).getMap();
        //Tipo de mapa
        mapa.setMapType(GoogleMap.MAP_TYPE_NORMAL);
        //Zoom
        mapa.moveCamera(CameraUpdateFactory.newLatLngZoom(ESP, 15));
        //Muestra mi localización
        mapa.setMyLocationEnabled(true);
        //Obtiene los controles de usuario en la interfaz del mapa
        //Habilita y deshabilita la brújula y el zoom del mapa
        mapa.getUiSettings().setZoomControlsEnabled(false);
        mapa.getUiSettings().setCompassEnabled(true);
        //Añade un marcador a la posición del centro comercial
        mapa.addMarker(new MarkerOptions()
            .position(ESP)
            .title("ESP")
            .snippet("Centro Comercial Espiral")
            .icon(BitmapDescriptorFactory.fromResource(R.drawable.ic_launcher))
            .anchor(0.5f, 0.5f));
        //El marcador se maneja en el evento clic de mapa
        mapa.setOnMapClickListener(this);
    }
    //Botón IR A ESPIRAL mueve la cámara a la posición del espiral
    public void moveCamera(View view) {
        mapa.moveCamera(CameraUpdateFactory.newLatLng(ESP));
    }
    //Botón MI POSICIÓN mueve la cámara hacia la posición del
    // usuario
    public void animateCamera(View view) {
        if (mapa.getMyLocation() != null)
            mapa.animateCamera(CameraUpdateFactory.newLatLngZoom(
                new LatLng( mapa.getMyLocation().getLatitude(),
                    mapa.getMyLocation().getLongitude()), 15));
    }
    //Botón MARCADOR añade un marcador a cualquier posición del mapa
    public void addMarker(View view) {
        mapa.addMarker(new MarkerOptions().position(
            new LatLng(mapa.getCameraPosition().target.latitude,
                mapa.getCameraPosition().target.longitude)));
    }
    //Evento clic del mapa
    @Override
    public void onMapClick(LatLng puntoPulsado) {
        mapa.addMarker(new MarkerOptions().position(puntoPulsado).
            icon(BitmapDescriptorFactory
                .defaultMarker(BitmapDescriptorFactory.HUE_YELLOW)));
    }
}

```

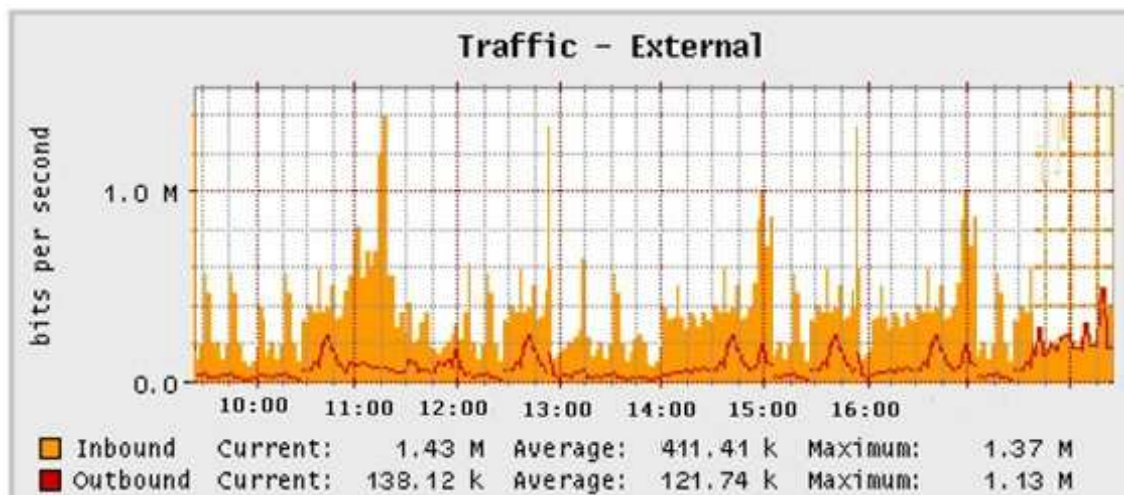
# ANEXO C

## ANEXO C

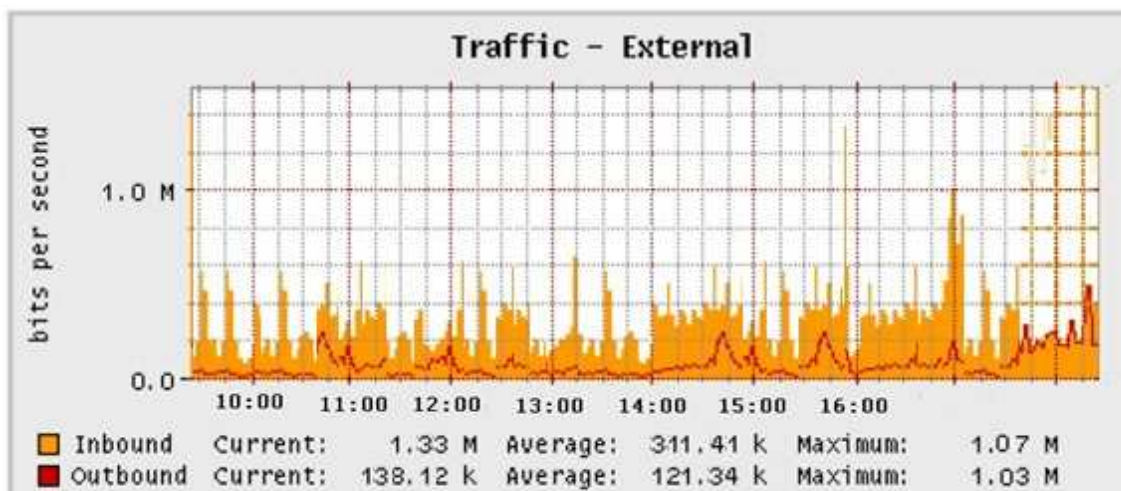
## GRÁFICAS DE TRÁFICO OBTENIDO EN 20 LOCALES COMERCIALES



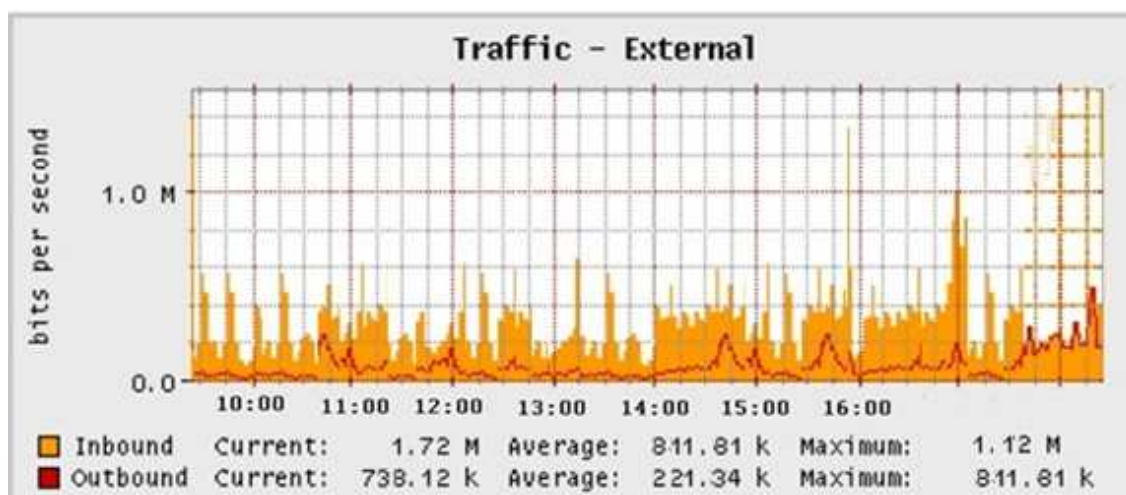
**Figura C.1:** Gráfico del tráfico del Local 22 Novicompu



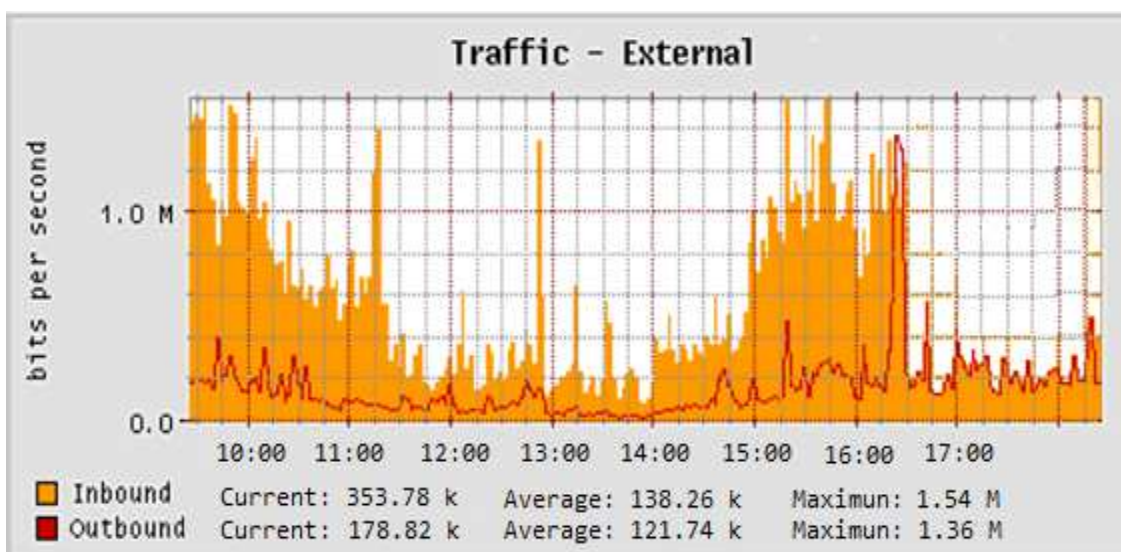
**Figura C.2:** Gráfico del tráfico del Local 24 Powernet



**Figura C.3:** Gráfico del tráfico del Local 34 Compukit

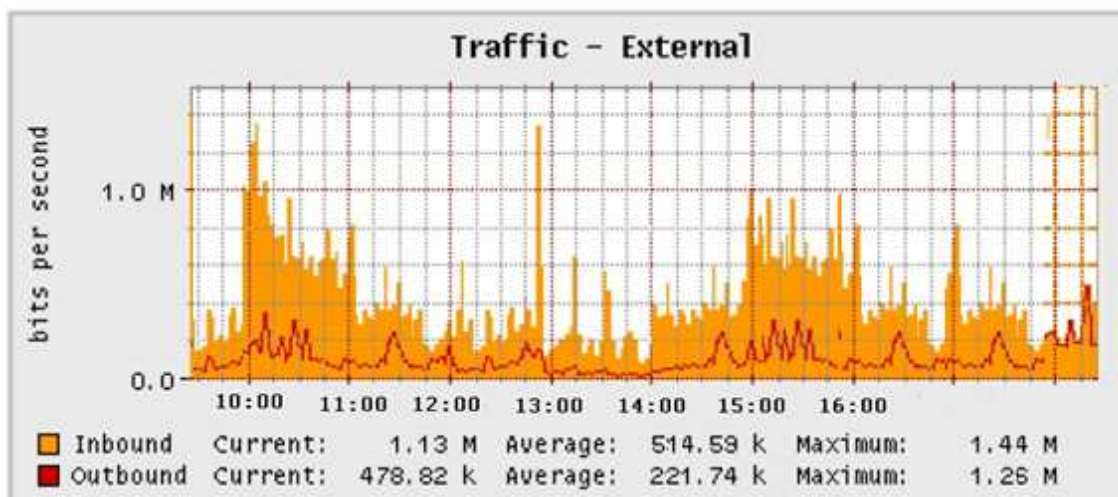


**Figura C.4:** Gráfico del tráfico del Local 33 Universal Technology

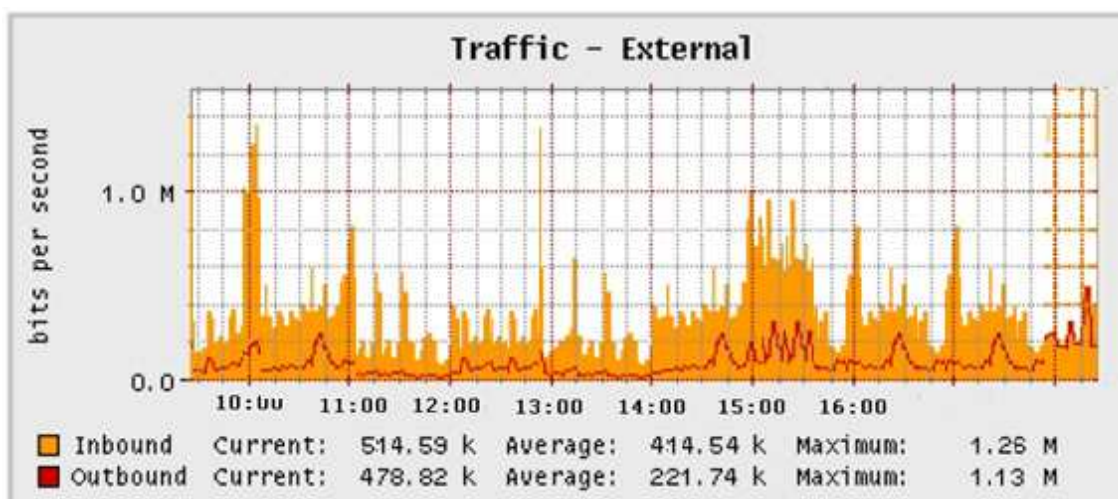


**Figura C.5:** Gráfico del tráfico del Local 41 Konami Games

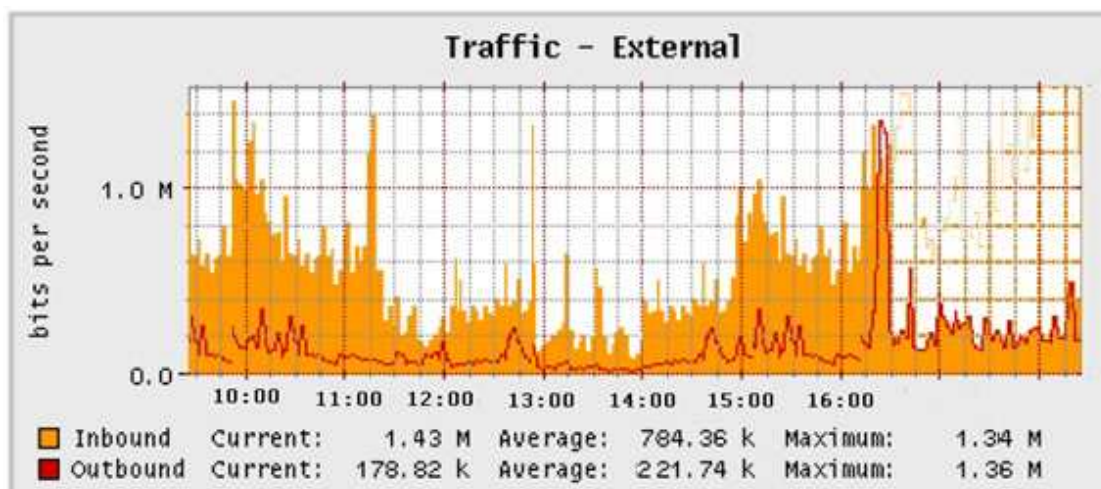




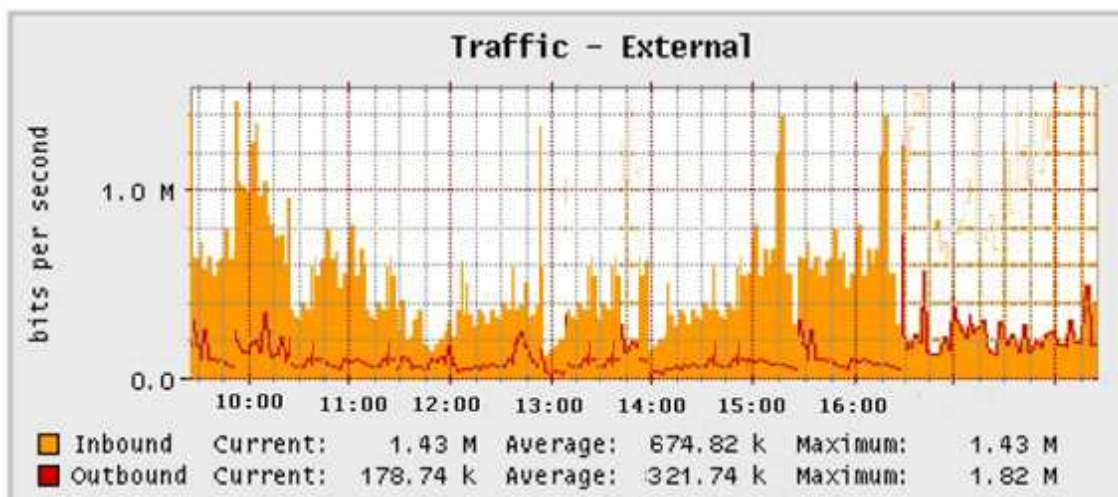
**Figura C.6:** Gráfico del tráfico del Local 47 RCM Sistemas



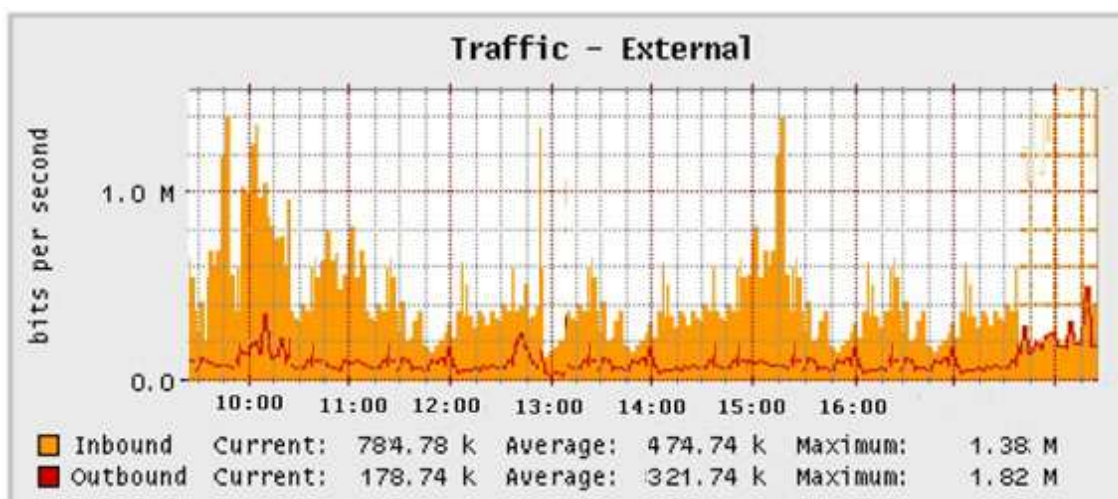
**Figura C.7:** Gráfico del tráfico del Local 50 Casa Bolivia



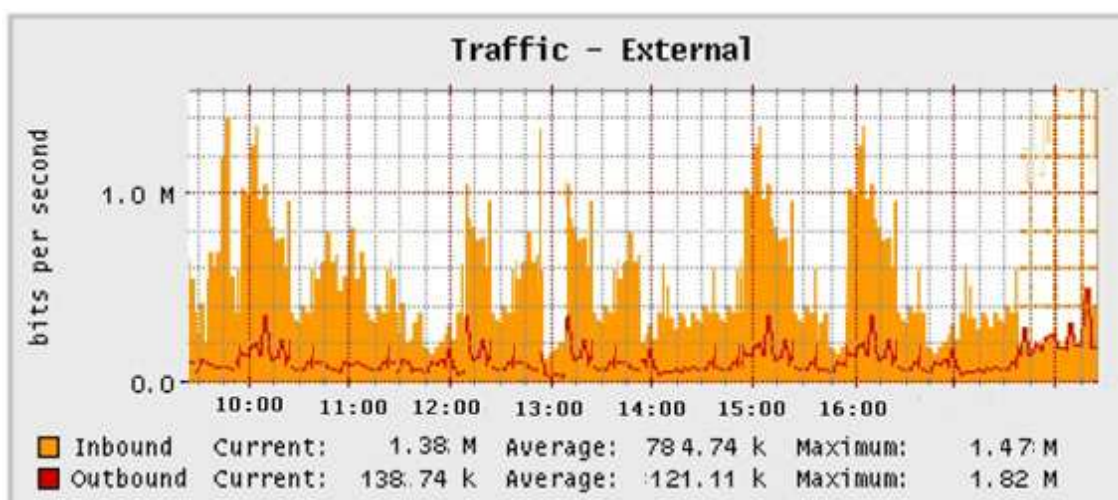
**Figura C.8:** Gráfico del tráfico del Local 67 Personalízate



**Figura C.9:** Gráfico del tráfico del Local 75 Movie Light

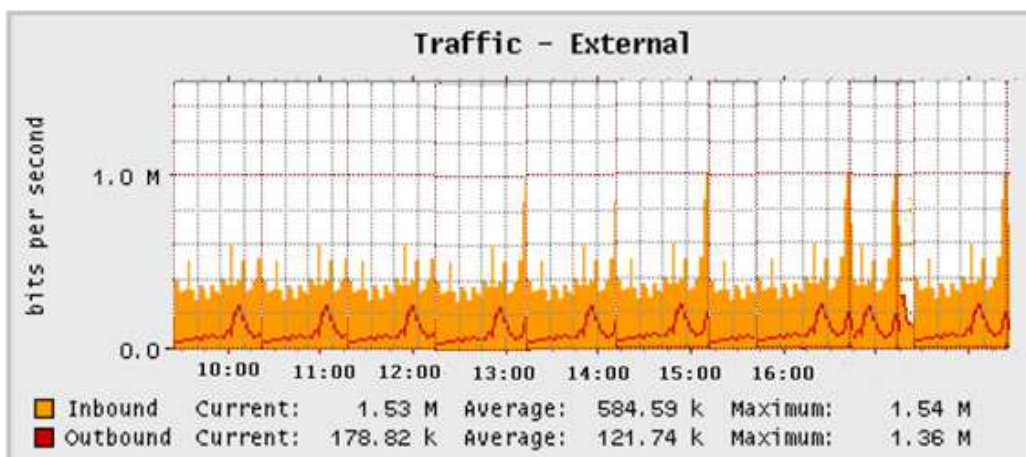


**Figura C.10:** Gráfico del tráfico del Local 80 Audiocenter

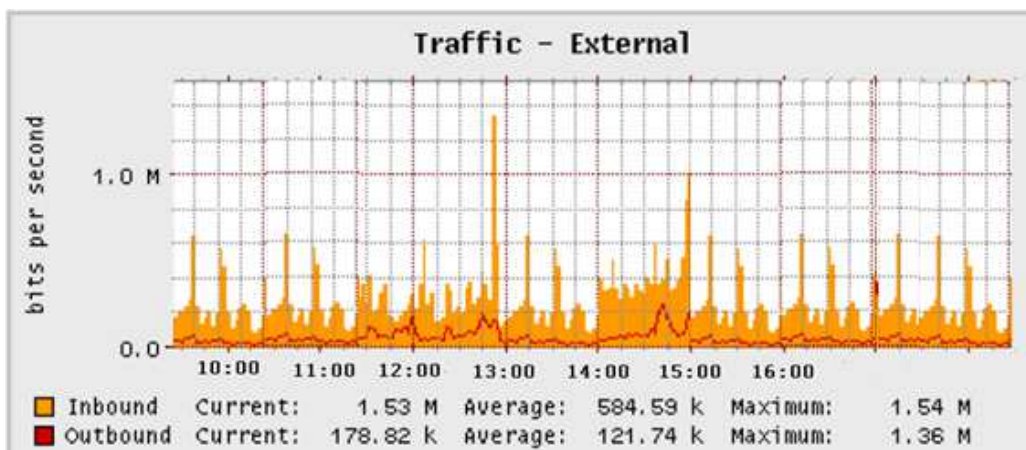


**Figura C.11:** Gráfico del tráfico del Local 81 Levelup

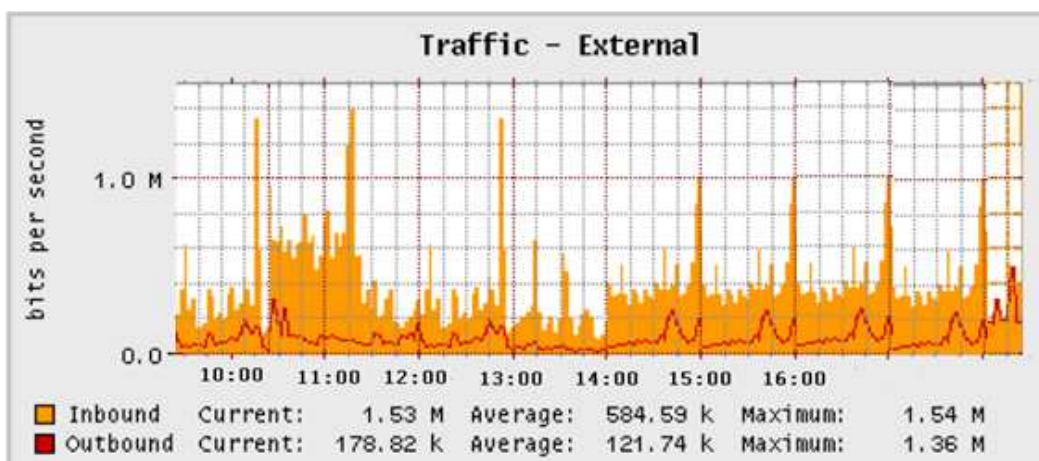




**Figura C.12:** Gráfico del tráfico del Local 94 Mágico Bazar

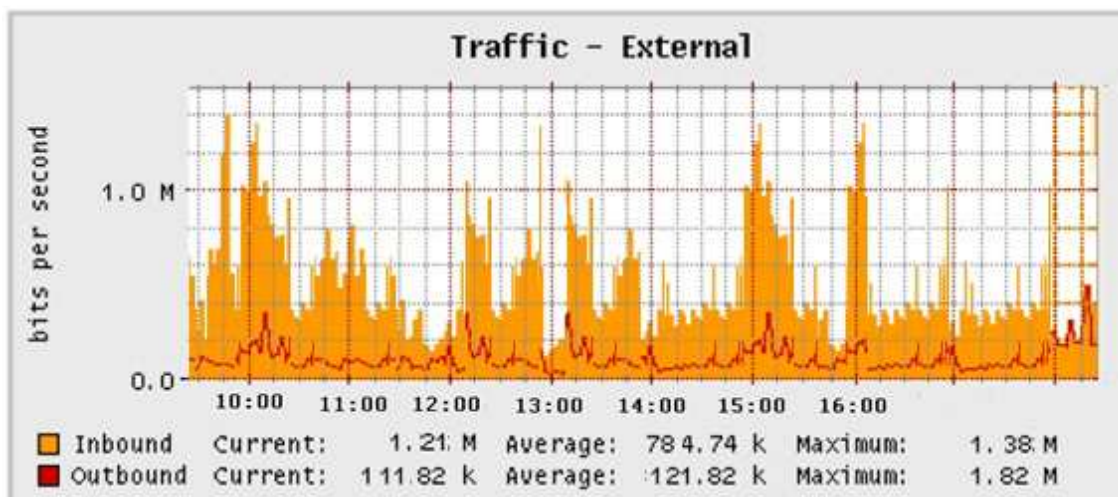


**Figura C.13:** Gráfico del tráfico del Local 97 Hellen Novedades

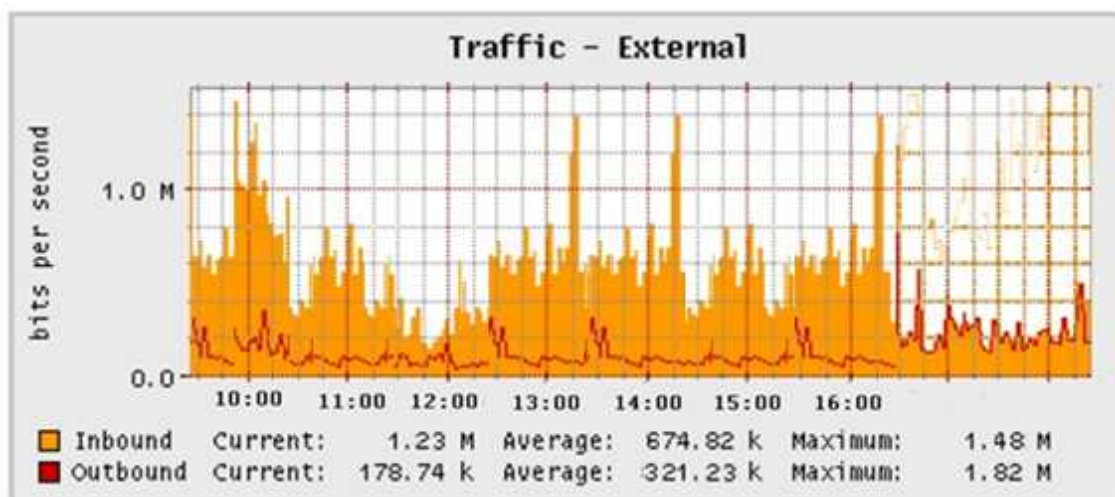


**Figura C.14:** Gráfico del tráfico del Local 98 Cell Mep

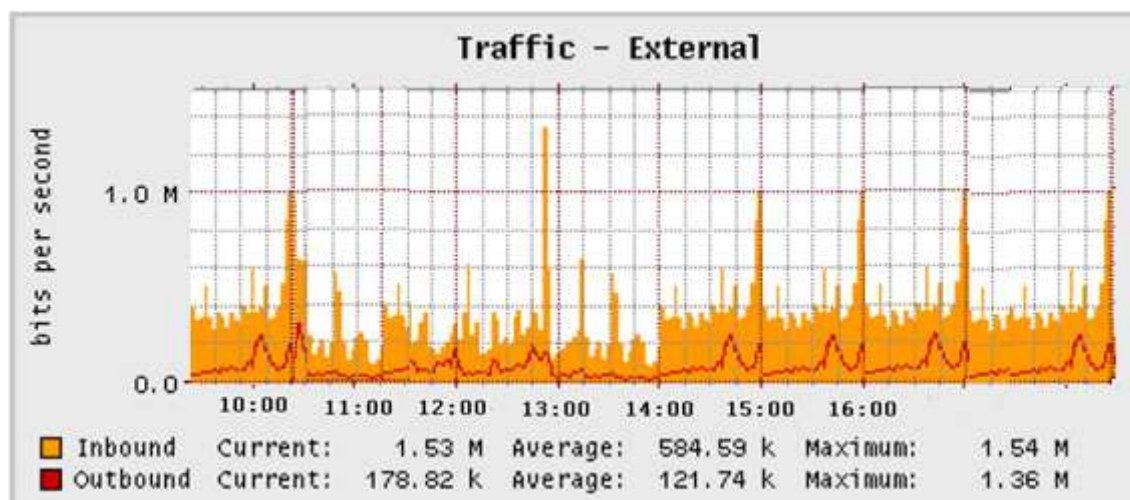




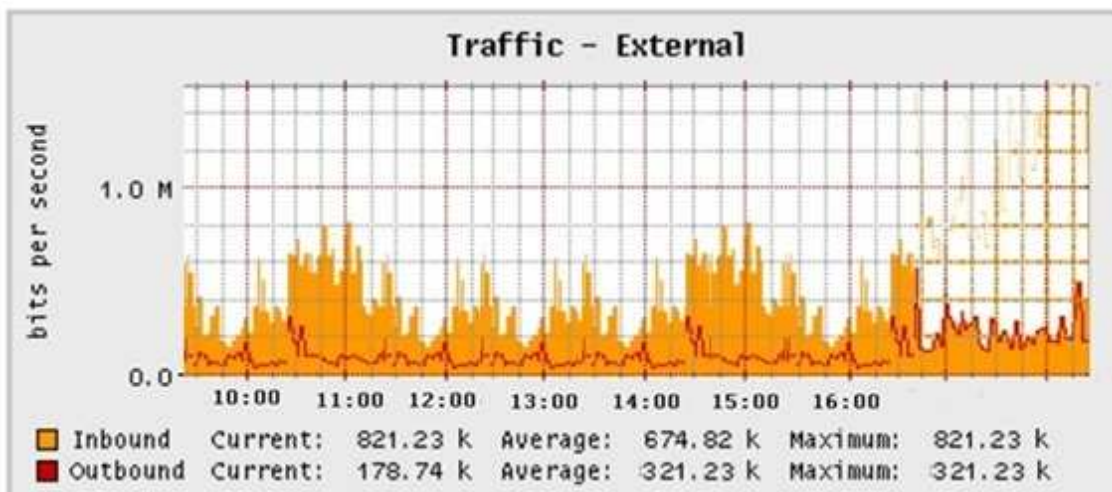
**Figura C.15:** Gráfico del tráfico del Local 102 Fachas



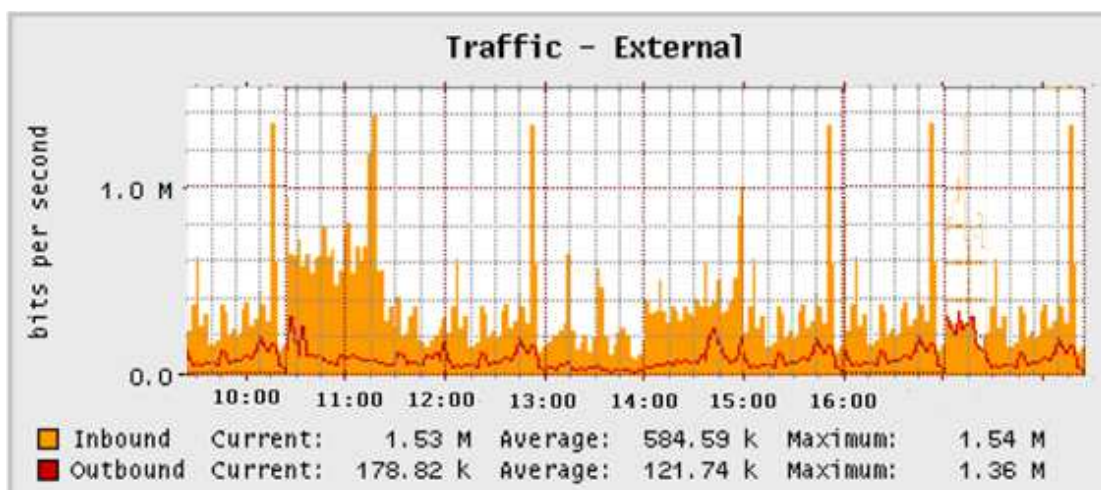
**Figura C.16:** Gráfico del tráfico del Local 105 Ecuaventura



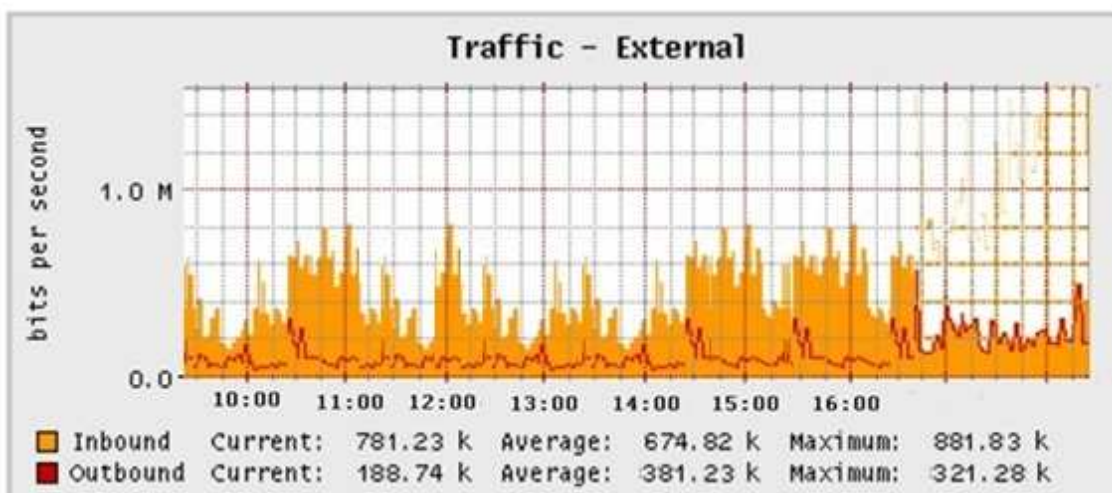
**Figura C.17:** Gráfico del tráfico del Local 124 Capcom



**Figura C.18:** Gráfico del tráfico del Local 134 EC Robotics



**Figura C.19:** Gráfico del tráfico del Local 136 Soltec



**Figura C.20:** Gráfico del tráfico del Local 140 Ecuavisión

# ANEXO D

## ANEXO D

# MODELO DE ENCUESTAS REALIZADAS EN EL CENTRO COMERCIAL ESPIRAL

### D1. ENCUESTAS PARA VISITANTES



ESCUELA POLITÉCNICA NACIONAL  
FACULTAD DE INGENIERIA ELÉCTRICA Y ELECTRÓNICA  
INGENIERIA EN ELECTRÓNICA Y REDES DE INFORMACIÓN



**INSTRUCCIONES:** La presente encuesta es anónima tiene como objetivo el recolectar la información necesaria previo a un proyecto de investigación en este establecimiento. Conteste con una **X** en cada uno de los recuadros en blanco correspondientes a cada una de las preguntas.

1. **¿Con qué frecuencia usted visita el Centro Comercial Espiral?**  
 Diariamente  Semanalmente  Mensualmente  Trimestralmente   
 Otros
2. **¿Por qué tipo de productos o servicios usted concurre a un centro comercial?**  
 Artículos tecnológicos  Servicio Técnico  Comida  Ropa   
 Otros
3. **¿Está de acuerdo con que el Centro Comercial Espiral disponga de una red inalámbrica de libre acceso para el uso de sus clientes?**  
 Sí  No
4. **¿Dispone de un dispositivo inalámbrico (laptop, celular, iphone, ipad, etc) que le permita la conexión a internet?**  
 Sí  No
5. **¿Cómo usuario de la red inalámbrica del centro comercial a qué tipo de aplicaciones le gustaría tener acceso?**  
 Correo Electrónico   
 Redes Sociales   
 Entretenimiento (Juegos)   
 Navegación web   
 Otros

6. **Dentro de un Centro Comercial que información le gustaría conocer?**  
 Ofertas   
 Productos y Promociones   
 Otros
7. **Conoce usted la funcionalidad y utilidad de una TABLET?**  
 Sí  No
8. **Le gustaría a usted adquirir una TABLET como dispositivo de conectividad?**  
 Sí  No
9. **Si usted contestó que si en la pregunta anterior, qué precio estaría dispuesto a pagar?**  
 Entre 150 y 250       Entre 250 y 350       Entre 350 y 500   
 Mayor a 500
10. **Le gustaría tener acceso a una aplicación interactiva que le muestre las ofertas y productos de los locales en el Establecimiento Comercial cargada directamente en su Tablet?**  
 Sí  No
11. **Le gustaría que en determinados puntos del Centro Comercial se coloque una tablet con la aplicación que permita la localización de ofertas y productos de cada uno de los locales, para facilitar su búsqueda?**  
 Sí  No

## D.2 ENCUESTAS PARA PROPIETARIOS DE LOCALES DEL CENTRO COMERCIAL



**ESCUELA POLITÉCNICA NACIONAL**  
**FACULTAD DE INGENIERIA ELÉCTRICA Y ELECTRÓNICA**  
**INGENIERIA EN ELECTRÓNICA Y REDES DE INFORMACIÓN**



**INSTRUCCIONES:** La presente encuesta es anónima tiene como objetivo el recolectar la información necesaria previo a un proyecto de investigación en este establecimiento. Conteste con una **X** en cada uno de los recuadros en blanco correspondientes a cada una de las preguntas.

1. **Con qué frecuencia los clientes visitan su local Comercial**  
 Nunca.....   
 Ocasionalmente.....   
 Frecuentemente.....   
 Muy a menudo.....   
 Siempre.....

2. **Está de acuerdo con que el Centro Comercial Espiral disponga de una red inalámbrica de libre acceso para el uso de clientes y propietarios de los locales.**  
Si..........  
No.......
3. **Como usuario de la red inalámbrica del Centro Comercial a qué tipo de aplicaciones le gustaría tener acceso.**  
Correo Electrónico..........  
Redes Sociales..........  
Entretenimiento (Juegos).......  
Aplicaciones propias de Centro Comercial.....   
Navegación Web..........
4. **Conoce usted la funcionalidad y utilidad de una TABLET.**  
Si..........  
No.......
5. **Le gustaría que los productos y/o servicios ofrecidos por su local sean publicitados a través de una aplicación cargada en una TABLET a la que tendrán acceso los clientes del Centro Comercial.**  
Si..........  
No.......