



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.

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA

**DESARROLLO DE UN MAPA INTERACTIVO PARA LA CIUDAD DE
QUITO QUE PERMITA VISUALIZAR LOS PARÁMETROS DE QoS
(CALIDAD DE SERVICIO) PARA LA SUPERTEL
(SUPERINTENDENCIA DE TELECOMUNICACIONES)**

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

**ARÉVALO COSTALES DIEGO PAUL
bendermetal888@gmail.com**

**GUZMÁN CABASCANGO DAVID ALFONSO
dguzman@supertel.gob.ec**

**DIRECTOR: ING. XAVIER CALDERÓN HINOJOSA, MSc.
xavier.calderon@epn.edu.ec**

**CODIRECTOR: ING. TARQUINO SÁNCHEZ, MBA.
tarquino.sanchez@epn.edu.ec**

Quito, Octubre 2013

DECLARACIÓN

Nosotros, Diego Paul Arévalo Costales y David Alfonso Guzmán Cabascango 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.

Diego Paul Arévalo Costales

David Alfonso Guzmán Cabascango

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por Diego Paul Arévalo Costales y David Alfonso Guzmán Cabascango, bajo mi supervisión.

ING. XAVIER CALDERÓN, MSc.
DIRECTOR DEL PROYECTO

ING. TARQUINO SÁNCHEZ, MBA
CODIRECTOR DEL PROYECTO

AGRADECIMIENTO

A Dios por darme la fuerza para luchar día a día por mis objetivos y metas.

A mi familia, María Elena, Paul, Tere, Teodoro, Teresa Etelvina y Ángel Alberto. Por estar siempre a mi lado, por todo el amor que he recibido de ustedes, por todos sus consejos, por haberme brindado apoyo incondicional desde el momento en que vine a este mundo.

Agradecimiento especial al Ing. Xavier Calderón e Ing. Tarquino Sánchez, por su constante predisposición para ayudar, su paciencia y consejos a lo largo del desarrollo del presente proyecto.

A David, por su dedicación y por haber compartido sus conocimientos.

A Alejandro, mi gran amigo, por siempre estar conmigo en los momentos buenos, pero también en los tiempos difíciles, y haberme ayudado a superarlos.

Diego Paul Arévalo Costales

Gracias mamá, no sé qué sería de mi sin ti. Mi reloj despertador, Luis Felipe. Vilma, por enseñarme a estudiar. Santiago por mostrarme el significado de perseverar. Gracias papá, por mostrarme lo duro de la vida. Miguel por enseñarme a trabajar.

Pablo, Mauricio, Alex, Ernesto, José Luis, Jaime y Ailed, por enseñarme el valor de la amistad. Álvaro, Xavier y Tarquino, por la oportunidad. Paúl, por la comprensión y esfuerzo.

Gracias a todas las personas que han colaborado con sus experiencias, consejos, guía, observaciones y exhortaciones.

David Alfonso Guzmán Cabascango

DEDICATORIA

A mi madre María Elena, siempre has estado conmigo brindándome tu amor y comprensión. Me has enseñado a ser fuerte y luchar en la vida. Me has brindado tu apoyo incondicional en todo momento. Has dedicado tu vida entera por hacerme feliz. Te amo mamá.

A mis queridos tíos Paul y Teresa, por todo el esfuerzo y cariño que me han brindado a lo largo de mi vida, siempre les estaré eternamente agradecido. Gracias por ayudarme a cumplir esta meta tan importante para mí.

Diego Paul Arévalo Costales

A los que han estado para mí desde el día uno.

Bernardina mi luz, guía, espíritu, coraje, fuerzas, esperanzas y enseñanzas.

Vilma, Santiago y Luis amigos, compañía y sobre todo hermanos.

Miguel, el papá que Dios me dio.

Manuel, el padre que la vida me regaló.

De todos he aprendido.

David Alfonso Guzmán Cabascango

CONTENIDO

CONTENIDO	V
ÍNDICE DE TABLAS	IX
ÍNDICE DE FIGURAS	X
ÍNDICE DE ESPACIOS DE CÓDIGO	XIII
RESUMEN	XV
PRESENTACIÓN	XVII
1. MARCO TEÓRICO	1
1.1 Introducción	1
1.2 Definiciones	3
1.2.1 Software Libre y Licenciamiento	3
1.2.2 Herramientas de desarrollo	3
1.2.2.1 Lenguaje de Programación	4
1.2.2.2 Framework	6
1.2.2.3 Interfaz de Programación de Aplicaciones (API)	7
1.2.2.4 Control de Versiones	7
1.2.2.5 Patrones de Diseño	7
1.2.2.6 Sistema Operativo	7
1.2.2.7 Base de Datos	8
1.2.2.8 Servidor Web	9
1.2.3 Sistema de Información geográfica (SIG)	10
1.2.3.1 Servidor de mapas	13
1.2.4 Aplicación web	13
1.2.4.1 Web 2.0 y las Aplicaciones Web Colaborativas	15
1.3 Sistemas móviles celulares	17
1.3.1 Evolución de los sistemas celulares	18
1.3.2 Sistema Global Para Las Comunicaciones Móviles (GSM)	18
1.3.2.1 Arquitectura de la Red GSM	18
1.3.3 Acceso Múltiple por División de Código (CDMA)	21
1.3.3.1 Características Generales	22
1.3.3.2 Arquitectura	23
1.4 Servicio móvil avanzado en Ecuador	25
1.4.1 Operadora	25
1.4.2 Calidad de Servicio (QoS)	26
1.4.3 Calidad de la Experiencia (QoE)	28
1.4.3.1 Gestión de la Calidad de la Experiencia	29
1.4.4 Superintendencia de telecomunicaciones SUPERTEL	30
1.4.5 Parámetros de QoS	33
1.4.5.1 Determinación de las Zonas de Medición en Ciudades	33
1.4.5.2 Zona de Cobertura	35
1.4.5.3 Porcentaje de Llamadas Establecidas	37
1.4.5.4 Tiempo de Establecimiento de Llamadas	38
1.4.5.5 Porcentaje de Llamadas Caídas	38
1.4.5.6 Porcentaje de Mensajes Cortos con Éxito	41
1.4.5.7 Tiempo Promedio de Entrega de Mensajes Cortos	42
1.4.5.8 Calidad de Conversación (MoS)	43
1.4.6 Proceso para el control de la calidad en el SMA	45
1.4.6.1 Fase 1, Mediciones de Calidad del Servicio	45
1.4.6.2 Fase 2, Comunicación de los Problemas de Calidad del Servicio	48
1.4.6.3 Fase 3, Determinación de Plazos para Solucionar los Problemas de QoS	49
1.4.6.4 Fase 4 y 5, Verificación de la Solución a los Problemas de QoS	50

2.	ANÁLISIS DE REQUERIMIENTOS EN SOFTWARE Y HARDWARE	51
2.1	Introducción	51
2.2	Situación actual.....	51
2.2.1	Recopilación de Datos	51
2.2.2	Sectorización de la ciudad de Quito	52
2.2.3	Sistema de Información geográfica (SIG).....	53
2.3	Sistema de Registro de Mediciones Telefonía Móvil (GTM)	56
2.3.1	Usuarios.....	56
2.3.2	Manejo de Reclamos e Información	56
2.4	Análisis de requerimientos.....	58
2.4.1	Datos espaciales.....	58
2.4.2	Columnas espaciales.....	61
2.4.2.1	Definición	61
2.4.3	Sistema de Referencia Espacial.....	62
2.4.3.1	Identificador de Sistema de Referencia.....	65
2.4.3.2	Sistema de Referencia WGS84 - EPSG 4326	66
2.4.4	Índices Espaciales	66
2.4.5	Funciones Espaciales	69
2.4.6	Operaciones entre Objetos Espaciales	69
2.4.7	Análisis Comparativo de Bases de Datos con soporte para Datos Espaciales	70
2.5	Servicios de Información Geográfica.....	72
2.5.1	Arquitectura de servicios	72
2.6	Catálogo de servicios OGC	73
2.7	Web Map Service.....	74
2.7.1	GetMap	77
2.7.2	GetCapabilities.....	79
2.7.3	GetFeatureInfo.....	80
2.8	Análisis de requerimientos para servidores de mapas.....	83
2.9	Dimensionamiento del/los servidores	86
2.9.1	Determinación del Ambiente de Desarrollo	87
2.9.2	Requerimientos en software y hardware	88
2.9.2.1	Servidor de Base de Datos	89
2.9.2.2	Servidor de Mapas.....	90
2.9.2.3	Servidor Web	90
3.	DISEÑO DEL PROTOTIPO.....	92
3.1	Introducción	92
3.2	Diseño del Prototipo.....	95
3.2.1	Marco de Desarrollo.....	95
3.2.2	Modelo de Dominio	97
3.2.3	Modelos de Casos de Uso.....	98
3.2.3.1	Diagrama de Casos de Uso.....	98
3.2.3.2	Casos de Uso	100
3.2.3.2.1	CdU:01 Navegar Sobre El Mapa	100
3.2.3.2.2	CdU:02 Buscar Localidad	100
3.2.3.2.3	CdU:03 Generar Capas	101
3.2.3.2.4	CdU:04 Visualizar Mapa Base.....	102
3.2.3.2.5	CdU:05 Visualizar Zonas De Medición	102
3.2.3.2.6	CdU:06 Visualizar Mapa Ciudad De Quito Sectorizado	103
3.2.3.2.7	CdU:07 Consultar Parámetros De QoS	103
3.2.3.2.8	CdU:08 Visualizar Parámetros De QoS.....	104
3.2.3.2.9	CdU:09 Realizar Reclamos CIR-SUPTEL	104
3.2.3.2.10	CdU:10 Observar Reclamos Realizados	105
3.2.3.2.11	CdU:11 Visualizar Rutas Drive-Test.....	106
3.2.3.2.12	CdU:12 Visualizar Localización De Radio Bases	109
3.2.3.2.13	CdU:13 Visualizar Descripción de Parámetros de QoS	110
3.2.3.2.14	CdU:14 Visualizar Leyendas	111

3.2.3.2.15	CdU:15 Autenticación dentro del Sistema	112
3.2.3.2.16	CdU:16 Generar Sectorización	112
3.2.3.2.17	CdU:17 Configurar Visualización de Capas	113
3.2.3.2.18	CdU:18 Configurar Visualización de Zonas	114
3.2.3.2.19	CdU:19 Configurar Información Descriptiva	116
3.2.3.2.20	CdU:20 Generar Reportes	117
3.2.3.3	Especificación Complementaria	120
3.2.3.4	Funcionalidad	120
3.2.3.5	Registro y Gestión de Errores	121
3.2.3.6	Seguridad	122
3.2.3.7	Facilidad de Uso	123
3.2.3.8	Rendimiento	123
3.2.3.9	Soporte	124
3.2.3.10	Restricciones de Implementación	125
3.2.3.11	Componentes Adquiridos	125
3.2.3.12	Cuestiones Legales	125
3.2.3.13	Componentes de Libre Distribución	125
3.2.4	Modelo de Diseño Dinámico	125
3.2.4.1	Diagramas de Secuencia	127
3.2.5	Modelo de Diseño Estático	129
3.2.5.1	Diagrama de Clases	129
3.2.5.2	Diagramas Entidad Relación	131
3.2.5.2.1	Desarrollo de Vistas	134
3.2.5.2.2	Módulo de Sectorización	135
3.2.5.2.3	Módulo de Calidad de la Experiencia	138
3.2.5.3	Arquitectura del Prototipo	139
3.2.5.4	Visión de la Aplicación	139
3.2.5.5	Detalle de la Implementación	140
4.	IMPLEMENTACIÓN Y PRUEBAS DEL PROTOTIPO	141
4.1	Introducción	141
4.2	Implementación de la infraestructura	143
4.2.1	Implementación del Servidor Web Apache v2.4	143
4.2.2	Implementación del Servidor Mapas MS4W	145
4.2.3	Implementación del Servidor de Base de Datos Oracle 10g	151
4.2.4	Implementación del servidor de aplicaciones Oracle Web Logic 12c	152
4.3	Implementación de la Base de Datos	156
4.3.1	Creación de Tablas	156
4.3.2	Creación de Vistas	157
4.3.3	Creación deL Procedimiento Almacenado SECTORIZAR	159
4.3.3.1	Función Sdo_geom.sdo_area ()	159
4.3.3.2	Función Sdo_aggr_union()	159
4.3.4	Creación de Funciones	161
4.3.4.1	Función FUNC_PARAM_FROM_ZONA	161
4.4	Migración de Datos	161
4.4.1	Migración Esquema GTM	161
4.4.2	Migración DE Datos Geográficos	162
4.5	Integración de Base de Datos Oracle Oracle WebLogic Sever 12c	172
4.6	Implementación de la Aplicación Web	174
4.6.1	Implementación DE LA Capa de Persistencia	176
4.6.1.1	Estructura del Proyecto JPA	176
4.6.1.1.1	Unidad de Persistencia	176
4.6.1.1.2	Entidades	177
4.6.1.1.3	Contexto de Persistencia	177
4.6.1.1.4	Administrador de Entidades	179
4.6.2	Implementación Capa de Negocio	179
4.6.2.1	Estructura del proyecto EJB	180

4.6.2.1.1	Definición de Objetos de Sesión.....	181
4.6.2.1.2	Implementación de Objetos de Sesión	182
4.6.3	Implementación de Servicios Web	183
4.6.3.1	Estructura del Proyecto EJB	183
4.6.3.2	Java API for XML Web Services – API Java para Servicios Web XML (JAX-WS) ...	184
4.6.3.3	Archivo WSDL (Lenguaje de Descripción de Servicios Web)	184
4.6.4	Implementación Capa de Presentación.....	185
4.6.4.1	Estructura del Proyecto Web JEE	187
4.6.4.1.1	Archivo de configuración	187
4.6.4.1.2	Directorio de Depuración	191
4.6.4.1.3	Controladores	191
4.6.4.1.4	Vistas	198
4.6.4.1.5	Mapas	200
4.6.4.1.6	Reportes	204
4.6.5	Implementación de Niveles de Seguridad	207
4.6.5.1	Servidor Single Sign On.....	207
4.7	Entorno de Desarrollo Colaborativo.....	210
4.7.1	Repositorio	211
4.7.1.1	Cliente	211
4.7.2	Maven	211
4.8	Implementación del Módulo de Administrador	212
4.8.1	Módulo Login	212
4.8.2	Sectorización	212
4.9	Pruebas al Sistema prototipo calidadSMA	215
4.9.1	Pruebas de Funcionalidad - USUARIOS	215
4.9.1.1	Mapa de la Ciudad de Quito Sectorizado (Capa Base).....	215
4.9.1.2	Visualización de Parámetros de QoS, Leyendas e Información	216
4.9.1.3	Visualización de Mediciones de QoE	216
4.9.1.4	Visualización de Rutas y Radio Bases	217
4.9.1.5	Envío de Reclamos hacia el CIR-SUPERTEL.....	218
4.9.1.6	Mediciones de la Percepción del Servicio QoE	219
4.9.2	Pruebas comparación ejecución.....	219
4.9.2.1	Pruebas con Servicios Web.....	220
4.9.2.2	Prueba con Configuración de Recursos Locales	221
4.10	Optimización y Mejoras.....	222
4.10.1	Configuración Servidor de Caché de Mapas	223
4.11	Análisis de Costos	226
5.	CONCLUSIONES Y RECOMENDACIONES	230
5.1	Conclusiones	230
5.2	Recomendaciones	233
	REFERENCIAS BIBLIOGRÁFICAS.....	235
	REFERENCIAS BIBLIOGRÁFICAS WEB	237
	ANEXOS	
	ANEXO A: Introducción a JPA	
	ANEXO B: The JavaEE 6 Tutorial	
	ANEXO C: SOA Using Java Web Services	
	ANEXO D: Spring Security 3	
	ANEXO E: The JavaEE 5 Tutorial	
	ANEXO F: Java Platform, Enterprise Edition V5 Specification	
	ANEXO G: Version Control with Subversion	
	ANEXO H: EJB 3 in Action	
	ANEXO I: Instalación del Servidor MapProxy	
	ANEXO J: Instalación del Servidor Oracle	

ÍNDICE DE TABLAS

CAPÍTULO 1

Tabla 1.1	Equipo terminal de la red GSM	20
Tabla 1.2	Parámetros de QoS – SUPERTEL	33
Tabla 1.3	Estimación duración de cada llamada de prueba	33
Tabla 1.4	Parámetro Zona de Cobertura	36
Tabla 1.5	Valores de nivel de señal, parámetro cobertura	36
Tabla 1.6	Parámetro porcentaje de llamadas establecidas	37
Tabla 1.7	Parámetro tiempo de establecimiento de llamada	39
Tabla 1.8	Parámetro porcentaje de llamadas caídas	39
Tabla 1.9	Porcentaje de mensajes cortos con éxito	41
Tabla 1.10	Parámetro tiempo promedio de entrega de mensajes cortos	43
Tabla 1.11	Parámetro MOS	43
Tabla 1.12	Escala de valores que determina la calidad de conversación	44
Tabla 1.13	Tamaño muestras para mediciones MOS, en función de la población	44
Tabla 1.14	Equipos SUPERTEL para mediciones de campo	47
Tabla 1.15	Calendario para modificación de mapas publicados por las operadoras	49

CAPÍTULO 2

Tabla 2.1	Análisis Comparativo SIGs	54
Tabla 2.2	Credenciales para acceder a la base de datos del GTM	56
Tabla 2.3	Análisis Comparativo <i>shapefile</i> - Bases de Datos Espaciales	60
Tabla 2.4	Tipos de columnas espaciales	62
Tabla 2.5	Sistemas de referencia espacial	66
Tabla 2.6	Parámetros SRID 4326	67
Tabla 2.7	Funciones Espaciales	69
Tabla 2.8	Operaciones entre Objetos Espaciales	70
Tabla 2.9	Análisis Comparativo Bases De Datos Con Soporte Para Datos Espaciales	71
Tabla 2.10	Parámetros operación GetMap	77
Tabla 2.11	Parámetros de una operación GetCapabilities	80
Tabla 2.12	Parámetros operación <i>GetfeatureInfo</i>	81
Tabla 2.13	Caracteres especiales reservados – WMS	82
Tabla 2.14	Petición GetMap, usando métodos GET Y POST	83
Tabla 2.15	Análisis Comparativo Servidores de Mapas	84
Tabla 2.16	Requerimientos mínimos Sistema Operativo Debian Squeeze	88
Tabla 2.17	Requerimientos mínimos para operación VMware PLayer V9.0	89
Tabla 2.18	Requerimientos mínimos para operación Oracle EE 10g	89
Tabla 2.19	Requerimientos mínimos para operación SQL Developer 10g	90
Tabla 2.20	Requerimientos mínimos para operación Servidor de Mapas	90
Tabla 2.21	Requerimientos mínimos de hardware para ambiente de desarrollo	91

CAPÍTULO 3

Tabla 3.1	Marco de Desarrollo de Artefactos UP. c-comenzar; r-refinar	96
Tabla 3.2	CdU:01 Navegar Sobre El Mapa	100
Tabla 3.3	CdU:02 Buscar Localidad	101
Tabla 3.4	CdU:03 Generar Capas	102
Tabla 3.5	CdU:04 Visualizar Mapa Base	103
Tabla 3.6	CdU:05 Visualizar Zonas De Medición	104
Tabla 3.7	Visualizar Mapa Ciudad De Quito Sectorizado	104
Tabla 3.8	CdU:07 Consultar Parámetros De QoS	106
Tabla 3.9	CdU:08 Visualizar Parámetros De QoS	107
Tabla 3.10	CdU:09 Realizar Reclamos CIR-SUPERTEL	107
Tabla 3.11	CdU:10 Observar Reclamos Realizados	108
Tabla 3.12	CdU:11 Visualizar Rutas Drive-Test	110
Tabla 3.13	CdU:12 Visualizar Localización De Radio Bases	110

Tabla 3.14 CdU:13 Visualizar Descripción De Parámetros De QoS	112
Tabla 3.15 CdU:14 Visualizar Leyendas	113
Tabla 3.16 CdU:15 Autenticación Dentro Del Sistema	114
Tabla 3.17 CdU:16 Generar Sectorización	115
Tabla 3.18 CdU:17 Configurar Visualización De Capas	116
Tabla 3.19 CdU:18 Configurar Visualización De Zonas	117
Tabla 3.20 CdU:19 Configurar Información Descriptiva	118
Tabla 3.21 CdU:20 Generar Reportes	119
Tabla 3.22 Tabla de Descripción de Parámetros de Seguridad Consideradoel Prototipo	122

CAPÍTULO 4

Tabla 4.1 Parámetros Generales MapFile	148
Tabla 4.2 Parámetros de configuración capa dentro MapServer	150
Tabla 4.3 Parámetros de Conexión Servidor de Base de Datos	152
Tabla 4.4 Costos directos del proyecto	227
Tabla 4.5 Costos Indirectos del Proyecto	228
Tabla 4.6 Resumen análisis de costos del proyecto	229

ÍNDICE DE FIGURAS

CAPÍTULO 1

Figura 1.1 Proceso de Análisis de Herramientas para el Desarrollo del Sistema Informático	2
Figura 1.2 Tipos de Lenguajes de Programación	5
Figura 1.3 Esquema de Análisis del Sistema Operativo	8
Figura 1.4 Comparación entre Bases de Datos y Almacenamiento de Ficheros	8
Figura 1.5 Interacción Navegador - Servidor Web	9
Figura 1.6 Estructura Común de una URL	9
Figura 1.7 Imagen Satelital de la Ciudad de Quito	10
Figura 1.8 Conjunto de Datos Vectoriales que Definen la Zonificación de la Ciudad de Quito	11
Figura 1.9 Mapa de Quito Sectorizado	12
Figura 1.10 Servidor de Mapas - Internet Map Server (IMS)	13
Figura 1.11 Funcionamiento Aplicación Web	14
Figura 1.12 Representación de Sitio Web 1.0	15
Figura 1.13 Representación de Sitio Web 2.0	16
Figura 1.14 Celda	17
Figura 1.15 Características de la Evolución de los Sistemas Celulares	19
Figura 1.16 Subsistemas de la Red GSM	20
Figura 1.17 Componentes de la Red GSM	21
Figura 1.18 Comparación de Espectro de Señales	22
Figura 1.19 Arquitectura Básica CDMA	24
Figura 1.20 Esquema de la Relación entre Satisfacción del Cliente, y QoS	27
Figura 1.21 Fases de la Utilización de los Servicios desde el Punto de Vista del Cliente	27
Figura 1.22 Gestión de la Calidad de la Experiencia	30
Figura 1.23 Recorrido Ideal para Determinar una Zona de Medición	35
Figura 1.24 Esquema de una Radiobase Tipo A	40
Figura 1.25 Esquema de una Radiobase Tipo B	40
Figura 1.26 Esquema de una Radiobase Tipo C	40
Figura 1.27 Diagrama del Proceso para el Control de la Calidad del SMA	46

CAPÍTULO 2

Figura 2.1 Archivos que Conforman un <i>Shapefile</i>	53
Figura 2.2 Añadiendo una Capa Vectorial en Quantum GIS	55
Figura 2.3 Zonas de la Ciudad de Quito - Quantum GIS	55
Figura 2.4 Formulario para Reclamos o Solicitud de Información - CIR	57
Figura 2.5 Principio de Interoperabilidad	59

Figura 2.6 Jerarquía de Datos Espaciales OGC	61
Figura 2.7 Geoide de la Tierra	62
Figura 2.8 Intersección Teórica del Geoide y Elipsoide	63
Figura 2.9 Representación Teórica de un Punto Fundamental	63
Figura 2.10 Representación <i>Datum</i> Teórico	64
Figura 2.11 Sistema de Referencia WGS84	65
Figura 2.12 Bounding-box Contendiendo a una Figura	67
Figura 2.13 Análisis Jerárquico utilizando R-tree.....	68
Figura 2.14 Arquitectura Lógica de 4 niveles y su Interacción con Clientes Livianos y Pesados ...	73
Figura 2.15 Funcionamiento WMS.....	75
Figura 2.16 Arquitectura Servicios Web - OGC	76
Figura 2.17 Ejemplo de Respuesta GetCapabilities - formato XML	79
Figura 2.18 Imagen Sectorización Quito formato PNG.....	85
Figura 2.19 Imagen Sectorización Quito formato GIF.....	86
Figura 2.20 Ambiente de Desarrollo	87

CAPÍTULO 3

Figura 3.1 Proceso Unificado de Desarrollo	92
Figura 3.2 Desarrollo de Software Iterativo.....	93
Figura 3.3 Fases de UP	94
Figura 3.4 Disciplinas de UP Utilizadas	94
Figura 3.5 Disciplinas de UP.....	95
Figura 3.6 Modelo de Dominio Diseño del Prototipo.....	97
Figura 3.7 Descripción Actores Sistema Prototipo.....	98
Figura 3.8 Diagrama de Casos de Uso	99
Figura 3.9 Detalle de la Especificación Complementaria	120
Figura 3.10 Registro de Acciones en la Aplicación.....	121
Figura 3.11 Descripción de las Facilidades de Uso	123
Figura 3.12 Funcionalidad de Scripts a Nivel de Cliente	124
Figura 3.13 Soporte del Prototipo Diseñado	124
Figura 3.14 Detalle de Componentes de Libre Distribución	126
Figura 3.15 Diagrama de Secuencia del Sistema para la Inicialización de la Aplicación.....	127
Figura 3.16 Diagrama de Secuencia del Sistema para la Visualización de Parámetros de QoS .	128
Figura 3.17 Visualización Detalle Parámetros de QoS e Ingreso Reclamos.....	129
Figura 3.18 Clases del Sistema, referencia el Modelo Estático I.....	130
Figura 3.19 Clases del Sistema, referencia el Modelo Estático II.....	131
Figura 3.20 Controladores Web I	132
Figura 3.21 Controladores Web II	132
Figura 3.22 Envío de Reclamos CIR.....	133
Figura 3.23 Tabla QOS_ZONAS_QUITO	134
Figura 3.24 Diagrama Vista QOS_CONECEL	134
Figura 3.25 Diagrama Vista QOS_OTECCEL	135
Figura 3.26 Diagrama Vista QOS_TELECSA	135
Figura 3.27 Modelo relacional Sectorización	136
Figura 3.28 Diagrama de Flujo Procedimiento Almacenado SP_SECTORIZACIÓN	137
Figura 3.29 Tabla QOS_BUFFER.....	138
Figura 3.30 Estructura de la Tabla QOE	138
Figura 3.31 Visión de la Arquitectura del Prototipo.....	139
Figura 3.32 Detalle de la Implementación.....	140

CAPÍTULO 4

Figura 4.1 Proceso de Implementación Sistema Prototipo calidadSMA.	142
Figura 4.2 Estructura de la Aplicación Web calidadSMA.....	144
Figura 4.3 Servidor MS4W iniciado.....	145
Figura 4.4 Estructura Básica Archivo MapFile	146
Figura 4.5 Cabecera Configuración MapFile	147
Figura 4.6 Generación Capa Sectorización Ciudad de Quito	149

Figura 4.7 Base de Datos Oracle 10g Iniciada	151
Figura 4.8 Inicio de Instalación Oracle WebLogic 12c	152
Figura 4.9 Parámetros de Instalación Típica de Oracle WebLogic Server 12c	153
Figura 4.10 Fin de Instalación Oracle WebLogic 12c	153
Figura 4.11 Pantalla de Inicio de Configuración Oracle WebLogic 12c	154
Figura 4.12 Creación de Dominio para Oracle WebLogic 12c	154
Figura 4.13 Configuración de Oracle WebLogic 12c, para brindar Soporte para JAX-WS	155
Figura 4.14 Definición de Credenciales de Acceso al Servidor Oracle WebLogic 12c	155
Figura 4.15 Parámetros de Configuración Oracle WebLogic 12c	156
Figura 4.16 Tolerancia en Objetos Espaciales	160
Figura 4.17 Base de Datos Esquema GTM – SUPERTEL	162
Figura 4.18 Añadir Capa Vectorial en Quantum GIS	163
Figura 4.19 Selección del Archivo “.shp”	163
Figura 4.20 Herramienta <i>shp2sdo</i> en ejecución	164
Figura 4.21 Leer Archivo Origen <i>shp2sdo</i>	165
Figura 4.22 Definición de la Tabla que Contendrá los Datos Espaciales - <i>shp2sdo</i>	165
Figura 4.23 Definición de Nombres de Columnas - <i>shp2sdo</i>	166
Figura 4.24 Especificación del Valor de Tolerancia - <i>shp2sdo</i>	166
Figura 4.25 Elección de la Versión de BDD Oracle a exportar - <i>shp2sdo</i>	167
Figura 4.26 Especificación del Valor de Precisión - <i>shp2sdo</i>	167
Figura 4.27 Archivos Generados por el Comando <i>shp2sdo</i>	168
Figura 4.28 Archivo QOS_ZONAS_QUITO.ctl	169
Figura 4.29 QOS_ZONAS_QUITO.dat	169
Figura 4.30 Ejecución del Comando <i>sqlldr</i>	170
Figura 4.31 Fin del Proceso de Carga del Archivo <i>sqlldr</i>	170
Figura 4.32 Tabla QOS_ZONAS_QUITO con Datos Cargados	171
Figura 4.33 Creación del índice espacial tabla QOS_ZONAS_QUITO	171
Figura 4.34 Proceso de Migración de Archivos .shp a Oracle 10g	172
Figura 4.35 Configuración de Origen de Datos Oracle WebLogic 12c	173
Figura 4.36 Selección de Controlador para la Conexión de Origen de Datos	173
Figura 4.37 Destino de Configuración de Orígenes de Datos	174
Figura 4.38 Estructura del Proyecto de Persistencia a la Base de Datos	177
Figura 4.39 Representación de Clases Entidades	178
Figura 4.40 Detalle de Configuración de Entidades	178
Figura 4.41 Ejemplo de Métodos de la Interfaz EntityManager	179
Figura 4.42 Funcionalidad de Clases EJB	180
Figura 4.43 Paquetes Contenedores de Clases EJB	180
Figura 4.44 Definición de Interfaces EJB	181
Figura 4.45 Ejemplo de Objetos de Sesión EJB	181
Figura 4.46 Funcionamiento de Entornos JAX-WS	183
Figura 4.47 Paquete Contenedor de Servicios Web	183
Figura 4.48 Ejemplo de Archivo WSDL Publicado	185
Figura 4.49 Especificación de Clase en Lenguaje de Marcado	186
Figura 4.50 Proyecto Web Java JEE	187
Figura 4.51 Estructura del Proyecto Web Java JEE	187
Figura 4.52 Detalle de Configuración del Archivo pom.xml	188
Figura 4.53 Detalle de la Sección de Configuración de Dependencias	188
Figura 4.54 Demostración de Almacenamiento del Repositorio Local	189
Figura 4.55 Detalle de la Sección Construcción del Archivo de Configuración	189
Figura 4.56 Detalle de la Configuración de la Sección Repositorios	190
Figura 4.57 Detalle de la Estructura de Código Fuente de la Aplicación Web	190
Figura 4.58 Detalle del Directorio de Depuración	191
Figura 4.59 Detalle del Paquete Contenedor de las Clases Controlador	191
Figura 4.60 Detalle de Funcionalidades Implementadas en las Clases Controlador	192
Figura 4.61 Ejemplo de Implementación de Clase Controlador	192
Figura 4.62 Visualización de Archivos de Configuración del Proyecto	193
Figura 4.63 Detalle de Configuración del Archivo action-servlet.xml	194
Figura 4.64 Detalle de Configuración de Clases de Spring Framework	194
Figura 4.65 Detalle de Configuración de Clases Personalizadas	195

Figura 4.66 Detalle de Funcionamiento de Métodos Publicados en el Servidor	195
Figura 4.67 Detalle de Funcionalidad de Servlets	196
Figura 4.68 Detalle de Solicitudes HTTP enviadas al Servidor	197
Figura 4.69 Detalle de Parámetros Enviados en las Solicitudes HTTP	197
Figura 4.70 Detalle de Respuesta a la Solicitud HTTP	197
Figura 4.71 Panel Contenedor Principal	200
Figura 4.72 Análisis de una Petición WMS	201
Figura 4.73 Panel Principal Contenedor del Mapa	202
Figura 4.74 SRIDs Utilizados en la Aplicación calidadSMA	203
Figura 4.75 Funcionamiento JasperReports	206
Figura 4.76 Reporte de Reclamos	208
Figura 4.77 Mecanismo de Autenticación SSO	209
Figura 4.78 Interfaz del Servidor de Versionamiento SVN	210
Figura 4.79 Descarga del Proyecto en un Cliente	211
Figura 4.80 Ejecución de Comandos Maven para Construcción del Proyecto	211
Figura 4.81 Generación de Grilla para Sectorización	212
Figura 4.82 Grilla de 250m de Lado	213
Figura 4.83 Generación Grilla 100m de Lado	213
Figura 4.84 Medición de Grilla de Área Igual 10000m ²	214
Figura 4.85 Generación Archivo shape con Grilla Ciudad de Quito	214
Figura 4.86 Pantalla Inicio Aplicación calidadSMA	215
Figura 4.87 Visualización Parámetro Cobertura - Movistar	216
Figura 4.88 Visualización de Mediciones de QoE	217
Figura 4.89 Visualización de Radio Bases Instaladas	217
Figura 4.90 Determinación de Parámetros de QoS en un Punto de Quito	218
Figura 4.91 Formulario para Registro de Reclamos	218
Figura 4.92 Formulario de Recepción de Mediciones de QoE	219
Figura 4.93 Latencia de Ejecución de Procedimientos Almacenados utilizando Servicios Web ..	220
Figura 4.94 Respuesta de Ejecución del Procedimiento Almacenado PARAM_FROM_ZONA ...	221
Figura 4.95 Latencia de Ejecución de SP utilizando Recursos Locales	221
Figura 4.96 Respuesta de Ejecución del Procedimiento Almacenado PARAM_FROM_ZONA ...	222
Figura 4.97 Configuración General Servidor Caché de Mapas	223
Figura 4.98 Definición Capa para Servidor de Caché de Mapas	223
Figura 4.99 Configuración Caché Mapproxy	224
Figura 4.100 Configuración Origen de Datos MapProxy	224
Figura 4.101 Configuración Almacenamiento de Caché MapProxy	225
Figura 4.102 Caché Generado en el Servidor MapProxy	226

ÍNDICE DE ESPACIOS DE CÓDIGO

CAPÍTULO 4

Espacio de Código 4.1 Script Generación Tabla Sectores Quito.	157
Espacio de Código 4.2 Creación Índice Espacial.	157
Espacio de Código 4.3 Creación de Claves Foráneas.	157
Espacio de Código 4.4 Creación Vista de Parámetros de QoS operadora Conecel.	158
Espacio de Código 4.5 Procedimiento Almacenado para Sectorización.	160
Espacio de Código 4.6 Función Parámetros de QoS de una Zona.	161
Espacio de Código 4.7 Detalle de Archivo de Configuración persistence.xml.	177
Espacio de Código 4.8 Configuración de Clase Administrador de Entidades.	179
Espacio de Código 4.9 Ejemplo de Implementación de Interfaz EJB.	182
Espacio de Código 4.10 Ejemplo de Clase Representando un Servicio Web.	184
Espacio de Código 4.11 Definición de documentos HTML Contenedores.	198
Espacio de Código 4.12 Definición de Bloque de Código JavaScript.	198
Espacio de Código 4.13 Definición de Variables y Constantes.	199
Espacio de Código 4.14 Panel principal, código JavaScript.	199
Espacio de Código 4.15 Inicialización de controles ExtJS.	200
Espacio de Código 4.16 Definición capa WMS.	201

Espacio de Código 4.17 Definición del Panel Contenedor del mapa.	202
Espacio de Código 4.18 Inclusión de la librería Proj4js en el contexto de la aplicación.	203
Espacio de Código 4.19 Transformación de proyecciones entre SRID 4326 y 900913.	204
Espacio de Código 4.20 Entidad Reclamo.java.	205
Espacio de Código 4.21 Cargar archivo jasper en entorno de desarrollo.	206
Espacio de Código 4.22 Obtener datos para el reporte.	207
Espacio de Código 4.23 Visualizar reporte formato PDF.	207
Espacio de Código 4.24 Capa con origen de datos del servidor de cache de mapas MapProxy.	225

RESUMEN

Con un total de 16.963.111 líneas activas del servicio móvil avanzado (telefonía móvil) en las tres diferentes operadoras existentes (CONECEL S.A.- Claro, TELECSA S.A.- CNT y OTECEL S.A.- Movistar) hasta Mayo de 2013, se hace imprescindible la aplicación de rigurosos métodos que aseguren el control de los servicios de telecomunicación y la calidad del servicio que las mencionadas operadoras actualmente brindan a sus usuarios terminales.

En el capítulo uno se describe el detalle del procedimiento de control, el cual debe cumplir varias fases, primero recolección de datos (mediciones de campo), seguido del procesamiento y almacenamiento de la información recopilada (almacenamiento en bases de datos, reportes, informes técnicos, etc.), finalmente una fase de difusión a los usuarios de la información recopilada y procesada.

En el capítulo dos se ha estudiado varias opciones en lo referente a desarrollo de software estructurado, interoperable y reusable; además de la manera en que se puede hacer interactuar sistemas que ofrecen servicios de mapas y geo-localización con las diferentes infraestructuras desarrolladas e implementadas. Además, se realizará una investigación del tipo de dato espacial, debido a que difiere mucho de los tipos de datos tradicionales como enteros, decimales, y alfanuméricos.

En el capítulo tres se realizará el planteamiento de una solución corporativa para el desarrollo de aplicaciones y sistemas web estándar; así también, una manera interactiva en la que el usuario de servicios de telecomunicaciones se haga participe de la información técnica que se manipula en temas de comunicaciones móviles. Los diseños presentados irán acorde a la estandarización que maneja la SUPERTEL. Además, la arquitectura propuesta garantizará alta disponibilidad, seguridad, interoperabilidad (debido a la variedad de orígenes de datos) y buen rendimiento en condiciones de alta demanda por parte de los usuarios.

Tomando como base las recomendaciones de la *Open Geospatial Consortium* (OGC) para todo lo referente a sistemas de información geográfica y los estándares *Java Enterprise Edition* (JEE) para el desarrollo de aplicaciones web,

se presentará una solución que satisfaga las necesidades descritas en el análisis de requerimientos.

El capítulo cuarto, describirá el proceso de transformación de información geográfica en datos espaciales para su posterior almacenamiento en una base de datos espacial (caso específico motor de base de datos Oracle 11g *spatial*), para poder asociar mediante modelos relacionales los valores de las mediciones con las localizaciones geográficas. Consecuentemente, la información geográfica almacenada en una base de datos será publicada en una aplicación web que permita mostrar esta información sobre un mapa de la ciudad de Quito, utilizando los paradigmas de solapamiento de capas. Finalmente, se realizarán pruebas de funcionalidad de la aplicación desarrollada.

En el capítulo cinco se desarrollarán las conclusiones y recomendaciones del presente proyecto.

PRESENTACIÓN

Una de las tecnologías inalámbricas que mayor desarrollo y auge ha tenido en los últimos años ha sido la telefonía celular, la misma que inicialmente fue concebida estrictamente para transmisión y recepción de voz. Actualmente, la tecnología celular es capaz de brindar una mayor gama de servicios tales como, texto, datos, audio y video con definidas limitaciones.

Actualmente, los usuarios de telefonía móvil tienen diferentes percepciones de la calidad de servicio en redes móviles debido a la insuficiente cantidad de información acerca del servicio que están contratando. El diseño de un sistema que muestre información acerca de parámetros de calidad del servicio móvil avanzado se basa en el estudio de parámetros de calidad de servicio, análisis de sistemas de información geográfica y diseño de aplicaciones empresariales bajo estándares abiertos y software libre.

La definición de los parámetros de calidad de servicio han sido previamente estudiados de las definiciones estipuladas en los contratos de concesión negociados entre el gobierno y las operadoras; así, el presente documento describe un detalle acerca de la definición de los parámetros de QoS y las fuentes de información que representan los parámetros de calidad de servicio controlados por la SUPERTEL.

El análisis de sistemas de información geográfica es necesario para encontrar la manera de representar conceptos técnicos como porcentaje de SMS's exitosos, porcentaje de llamadas establecidas y cobertura, sobre un mapa. La publicación de parámetros de calidad de servicio sobre un mapa interactivo es una manera de permitir al usuario navegar a través de la información que la SUPERTEL recolecta y procesa.

Finalmente, el diseño e implementación de un aplicativo empresarial basado en estándares abiertos permite que el producto pueda interactuar en ambientes de infraestructuras heterogéneas.

Los Autores

1. MARCO TEÓRICO

1.1 INTRODUCCIÓN

La constante investigación reflejada en los notables avances tecnológicos han permitido que las comunicaciones se tornen un sector importante en el desenvolvimiento de la cotidianidad en ámbitos políticos, económicos, financieros sociales y culturales. Una de las tecnologías inalámbricas que mayor desarrollo y auge ha tenido en los últimos años ha sido la telefonía celular, la misma que inicialmente fue concebida estrictamente para transmisión y recepción de voz, actualmente la tecnología celular es capaz de brindar una mayor gama de servicios tales como: texto, datos, audio y video con definidas limitaciones; mensajes de texto cortos, correo electrónico, conectividad con la red internet y con esto acceso a redes sociales entre otros productos. La convergencia de estos servicios a través de tecnologías celulares móviles se ha definido como el servicio móvil avanzado; mismo que es cada vez más atractivo para el mercado ecuatoriano, debido a la influencia de referentes internacionales.

En el presente capítulo se especificarán las definiciones principales inherentes al desarrollo del proyecto, entre los cuales se puede encontrar conceptos referentes al desarrollo de software: aplicación web, aplicación web colaborativa, software libre, lenguaje de programación, patrones de diseño; así como nociones referentes a las tecnologías de sistemas móviles celulares, el sistema móvil avanzado definido en el país y sus parámetros de calidad de servicio.

Se revisará la situación actual en lo referente al organismo de control técnico de las telecomunicaciones en el Ecuador, las definiciones que este ente ha efectuado con las operadoras en lo referente a parámetros de calidad de servicio y su proceso de medición.

Así también, se definirán nociones de las herramientas e infraestructura digital¹ que se utilizarán en el desarrollo y estructuración de la aplicación, para mostrar parámetros de calidad de servicio móvil avanzado en la ciudad de Quito.

¹ *Framework* o infraestructura digital, conjunto de módulos de software estructurado, que permiten la reutilización de código fuente en el desarrollo de aplicaciones.

Se ha planteado un análisis por capas partiendo del nivel inferior en el diseño del sistema. Inicialmente se definirá la plataforma (Sistema Operativo) que albergará ambiente de desarrollo y pruebas. Luego se seleccionará la tecnología que se usará para el desarrollo de la aplicación, así como el lenguaje de programación inherente a la tecnología utilizada. Se revisará los motores de bases de datos que pueden ser utilizadas para los propósitos de almacenar información geográfica, así como su compatibilidad e interoperabilidad, en soluciones libres y licenciadas.

También se revisarán los conceptos de arquitectura de sistemas de información geográfica, para así poder describir la estructura de comunicaciones que se utilizará en el diseño de la aplicación.

Finalmente, se definirán los servidores de mapas existentes, los servicios que ofrecen y cuáles de éstos serán de utilidad para el desarrollo e implementación del servicio de visualización de parámetros de calidad de servicio en el servicio móvil avanzado. La Figura 1.1 esquematiza el proceso de análisis planteado de herramientas a seguir.

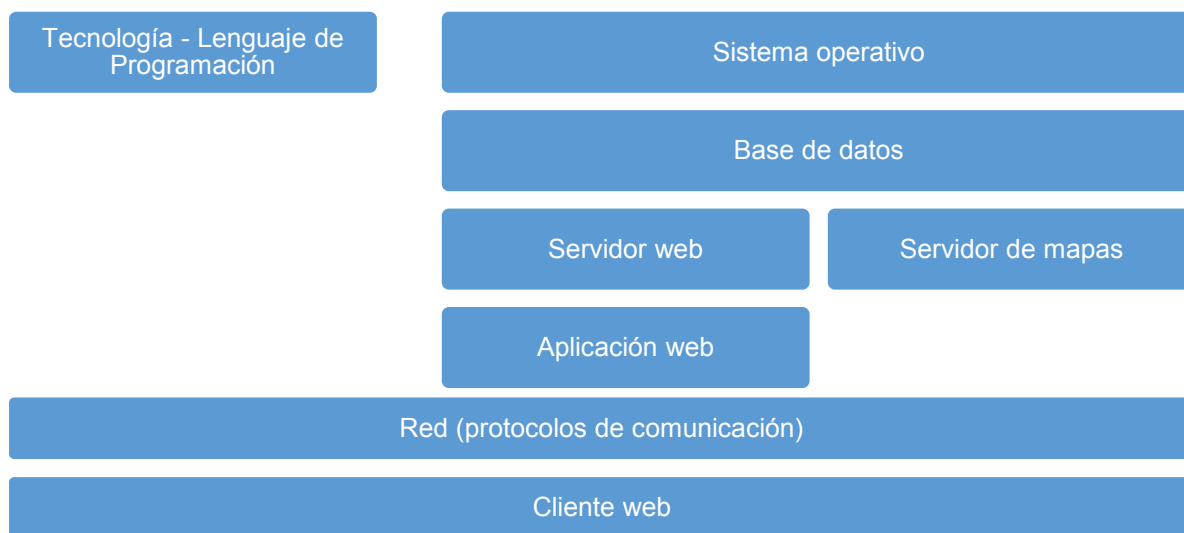


Figura 1.1 Proceso de Análisis de Herramientas para el Desarrollo del Sistema Informático

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

1.2 DEFINICIONES

1.2.1 SOFTWARE LIBRE Y LICENCIAMIENTO

El código (programas), es la tecnología mediante la cual el usuario, interactúa o da instrucciones al hardware de un computador y garantiza un correcto funcionamiento del mismo. El código es el encargado de dirigir a los ordenadores. Bajo esta premisa, surgen inquietudes como: ¿quién debe controlar el código?, ¿quién debe tener la libertad de utilizar el código?, ¿quién puede manejar o modificar dicho código?.

De este hecho nace el concepto de software libre, mantener libres los códigos. Un aspecto que se debe destacar de código libre es que el mismo está orientado a tener el control sobre los codificadores y que su trabajo sea transparente para todos; es decir, libertad en el manejo y modificación de dichos controles para adaptarlos a requerimientos específicos o para mejorar su funcionamiento. El término libre en este contexto no se relaciona con el hecho de que un desarrollador no deba recibir una retribución económica por su aporte, solo está enfocado a la libertad de tomar el control sobre dicho código.[11]. El software libre es un tema de la libertad para los usuarios de ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software.

Un programa, para que se pueda considerar libre debe ser publicado con una licencia de software libre. Generalmente se utiliza la Licencia Pública General del proyecto GNU (GPL/GNU, *General Public License* – Licencia Pública General), pero eventualmente también se utilizan otras licencias de software libre. Para el software de GNU únicamente se utiliza otras licencias, solo si son compatibles con la licencia GPL de GNU. La documentación del software libre es documentación libre, publicada como GNU-FDL (*Free Documentation Licence* – Licencia de Documentación Libre), para que se pueda redistribuir y mejorar al igual que el software al cual describe.[11]

1.2.2 HERRAMIENTAS DE DESARROLLO

Las herramientas de desarrollo permiten realizar aplicativos, programas, rutinas, utilitarios y sistemas para que el hardware funcione y pueda producir resultados. La oferta actual pone a disposición herramientas de programación en el mercado

tanto para desarrolladores experimentados como para los que no lo son. Las herramientas de programación más comunes del mercado cuentan hoy en día con programas de depuración (también conocidos como *debugger*), los cuales son utilitarios que hacen posible la ejecución de código fuente para la realización de pruebas. Las herramientas de programación también pueden integrar servidores de aplicación, componentes de desarrollo y librerías.

1.2.2.1 Lenguaje de Programación

Dentro de un computador el hardware necesita de un conjunto de instrucciones para su funcionamiento y uso, este conjunto de instrucciones deberá ser escrito en un lenguaje de programación. Este lenguaje es un medio a través del cual es posible crear un conjunto de instrucciones válidas a ser ejecutadas por un computador, para esto un lenguaje de programación debe constar de un conjunto finito de símbolos, constituyendo estos símbolos el léxico del lenguaje. También debe determinar un conjunto finito de reglas propias del lenguaje, este conjunto de reglas constituyen la sintaxis de un lenguaje de programación. Además, el lenguaje define una semántica² para cada posible instrucción dada a la máquina. En la Figura 1.2 se esquematiza la organización de los diferentes tipos de lenguaje de programación.

Cabe acotar las diferencias entre lenguaje compilador y lenguaje intérprete. En el lenguaje compilador primero se realiza un análisis del léxico, la sintaxis y finalmente la semántica del programa fuente. Luego se traduce las instrucciones a un código intermedio, se optimiza este código para finalmente generar código en lenguaje de máquina; una vez que se tiene las instrucciones en lenguaje de máquina se procede a la ejecución del programa. Mientras que en el lenguaje intérprete, el código fuente es traducido a lenguaje de máquina sentencia por sentencia y se va ejecutando directamente en la máquina, no existe proceso de generación de código intermedio ni optimización de código.

² Aspectos de la interpretación y correspondencia de signos como símbolos, palabras, expresiones y representaciones.

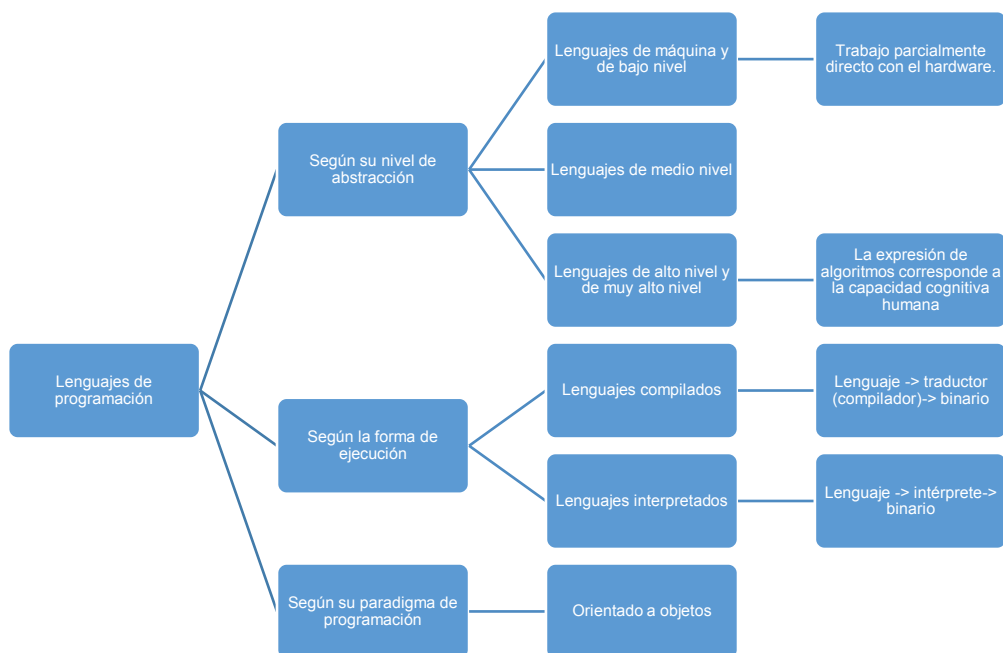


Figura 1.2 Tipos de Lenguajes de Programación

Fuente: Adaptado de: *Lenguajes de programación*. Recuperado de: <http://guimi.net>. (Diciembre 2011).

Dentro de los lenguajes de programación de mayor uso actualmente se tienen:

➤ **Java**

Desarrollado por la compañía *Sun Microsystems*, el lenguaje para la programación en Java, es un lenguaje orientado a objetos, multiplataforma, con licencia GNU GPL. La máquina virtual de Java (JVM *Java Virtual Machine* – Máquina Virtual Java) forma parte de la plataforma Java, está situada en un nivel superior al hardware del sistema sobre el que se pretende ejecutar la aplicación, y este actúa como un puente que entiende tanto el código intermedio, como el sistema sobre el que se pretende ejecutar.

El *bytecode* o código intermedio es generado por el compilador del lenguaje Java. Este código intermedio es interpretado por la máquina virtual de Java, ésta permite crear una máquina virtual de Java propiamente dicha común a cualquier equipo independiente del sistema operativo sobre el cual se instale, así cualquier aplicación que se programara en Java podría ser ejecutada en cualquier máquina

y/o dispositivo sin tener que compilar de nuevo el programa o volver a escribirlo para cada equipo.

➤ C#

C# es un lenguaje orientado a objetos creado por Microsoft dentro de su plataforma Microsoft .NET. C# combina los mejores elementos de múltiples lenguajes de amplia difusión como C++, Java, Visual Basic y Delphi. La idea principal detrás del lenguaje es combinar la potencia de lenguajes como C++ con la sencillez de lenguajes como Visual Basic, para que además la migración a este lenguaje por los programadores de C/C++/Java sea lo más simple posible.

➤ JavaScript

Lenguaje de programación basado en paradigmas como: programación basada en prototipos, sentencias imperativas y lenguaje interpretado (*scripting*). Actualmente existen dos tipos de *JavaScript*, por un lado está el que se ejecuta en el cliente, este es el *Javascript* propiamente dicho, aunque técnicamente se denomina *Navigator JavaScript*; y por otro lado está el *Javascript* que se ejecuta en el servidor, denominado *LiveWire Javascript*.

1.2.2.2 Framework

El término *framework* abarca varios ámbitos dentro de la programación, su alcance no solo se reduce al desarrollo de aplicaciones web. *Framework* hace referencia a una estructura que está compuesta por varios elementos personalizables y que a su vez se pueden intercambiar, para facilitar el proceso de desarrollo de una aplicación. Un *framework* es un patrón a seguir para el desarrollo de una aplicación, al ser considerado un patrón es modificable y configurable. A través del proceso de modificaciones del patrón original finalmente se termina por construir la aplicación en su totalidad. La utilización de un *framework* permite acelerar el proceso de desarrollo, ya que permite reutilizar código ya existente. Existe una gama muy amplia de *frameworks* para desarrollo de aplicaciones, orientados a la interfaz del usuario, orientados al control de eventos, orientados al manejo de clases, etc.[31]

1.2.2.3 Interfaz de Programación de Aplicaciones (API)

Son los métodos, funciones, clases, procedimientos, protocolos prescritos por el sistema operativo de un equipo o por una aplicación, a través del cual otra aplicación que está siendo desarrollada puede hacer peticiones. Un API proporciona acceso a funcionalidades implementadas en cierta aplicación. El objetivo de este concepto es mejorar la eficacia operativa, completando la aplicación existente con características comunes de gestión. A través de esta se pueden realizar integraciones avanzadas, las cuales pueden realizar tareas sofisticadas, tales como flujos de trabajo automatizado basado en políticas de negocio y otros criterios compartidos entre aplicaciones.[32]

1.2.2.4 Control de Versiones

El versionamiento o control de versiones de ciertos desarrollos de software se define como la gestión de los cambios que se realiza al código fuente o diseño del software o producto. Una versión hace referencia al estado en un momento dado en el que se encuentra cierto producto o desarrollo.

1.2.2.5 Patrones de Diseño

Los patrones de diseño son modelos de estructuras e implementaciones, los mismos que permiten dar soluciones a problemas de programación y estructuración a nivel de sentencias de código.³

1.2.2.6 Sistema Operativo

Un sistema operativo es la interfaz que existe entre el hardware y las aplicaciones o programas, esta interfaz puede ser analizada desde dos ópticas diferentes dependiendo de su función; puede ser considerado una máquina extendida o máquina virtual debido la abstracción de las operaciones básicas sobre los dispositivos electrónicos que conforman la plataforma. El sistema operativo puede ser visto como un gestor de recursos (Figura 1.3), debido a que es el encargado de administrar, repartir, ordenar y controlar los diferentes dispositivos de hardware.[8]

³ Patrones de diseño. <http://mit.ocw.universia.net/6.170/6.170/f01/pdf/lecture-12.pdf>. Septiembre 2011.

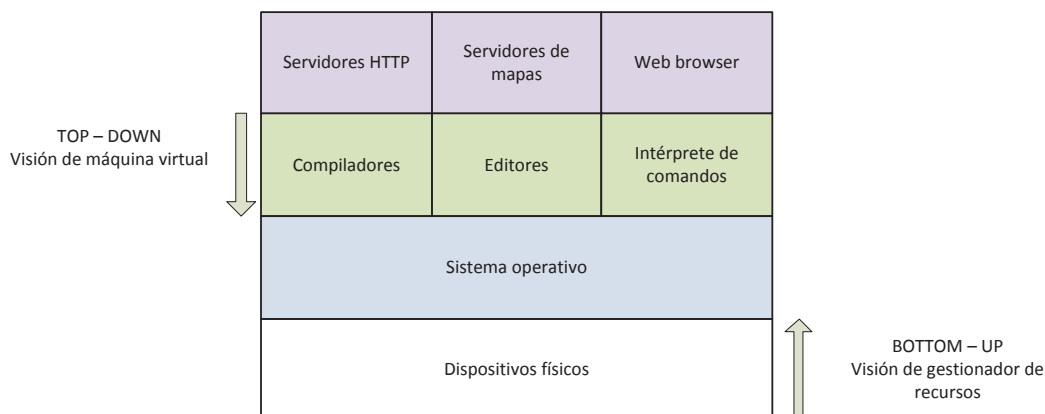


Figura 1.3 Esquema de Análisis del Sistema Operativo

Fuente: Adaptado de: Tanenbaum, Andrew. (2003). *Sistemas operativos modernos*. (pp. 99-112). México: Pearson Education.

1.2.2.7 Base de Datos

Las bases de datos son consideradas una representación informática o conjunto de estructuras de datos interrelacionadas, las mismas que hacen referencia a entidades o abstracciones de objetos reales. Estas estructuras de datos por lo general están compartidas por varios procesos de forma simultánea y son accedidas por distintos tipos de usuarios. La Figura 1.4 muestra ventajas que ofrece un almacenamiento de información utilizando una base de datos frente al uso de ficheros de almacenamiento aislados.[18]

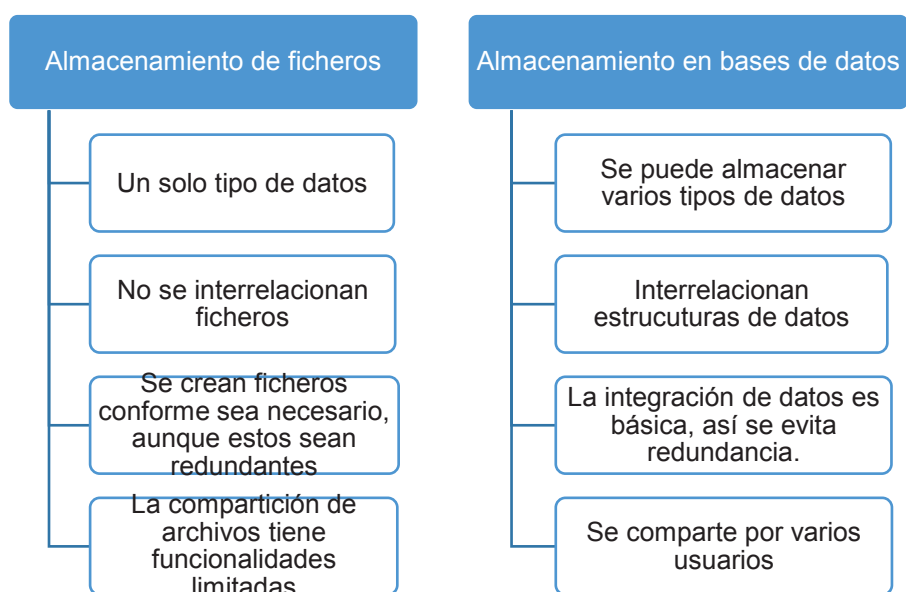


Figura 1.4 Comparación entre Bases de Datos y Almacenamiento de Ficheros

Fuente: Adaptado de: Camps Rafael, Casillas Luis, Costal Dolors, Gilbert Marc, Martín Carme & Pérez Oscar. (2005). *Bases de datos*. (pp. 7-13). España: Eureka Media.

1.2.2.8 Servidor Web

Un servidor web es una aplicación responsable de interpretar, enviar y recibir las peticiones HTTP que realizan los clientes de este servicio, navegadores web. Un navegador web es una aplicación cliente, capaz de enviar, recibir e interpretar mensajes HTTP los cuales pueden contener texto plano, páginas escritas en lenguaje de marcado, scripts, etc. El navegador, mediante el uso de una URL, solicita un recurso específico dentro de un servidor web específico.[10]

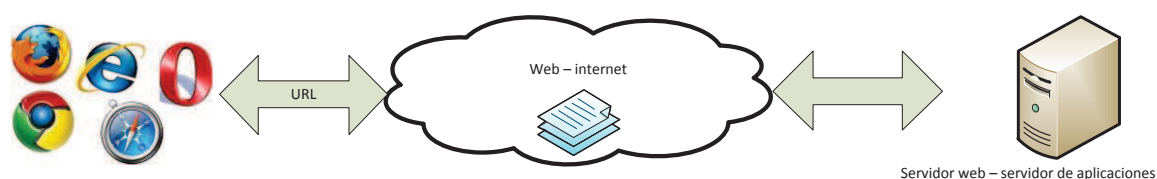


Figura 1.5 Interacción Navegador - Servidor Web

Fuente: Esquema de Autoría propia (Arévalo D., y Guzmán D.).

Comúnmente las peticiones y respuestas de estos servidores envían y reciben contenido estático (texto, texto HTML, imágenes, etc.) y contenido dinámico (scripts). La transmisión de información está a cargo del protocolo HTTP. Las solicitudes iniciales de la comunicación, contienen URL's las mismas que son direcciones únicas de los servidores; las direcciones URL contienen el nombre de dominio que indica la localización del ordenador en el internet que es el encargado de responder con la información solicitada; y la ruta indica la ubicación del archivo solicitado dentro del sistema de archivos del servidor. La Figura 1.6 muestra un ejemplo de cómo está estructurada una URL.

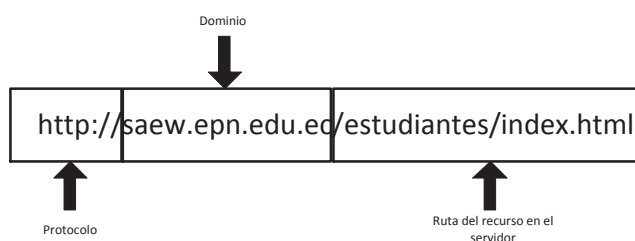


Figura 1.6 Estructura Común de una URL

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

1.2.3 SISTEMA DE INFORMACIÓN GEOGRÁFICA (SIG)

Un SIG en esencia permite visualizar, consultar y analizar datos de tipo espacial (medir distancias geográficas, determinar áreas, etc.). Los datos espaciales son información geográfica que define la ubicación de un ente; un ejemplo son las coordenadas geográficas (longitud y latitud).

Los datos geoespaciales pueden almacenarse de dos maneras diferentes, como datos de tipo ráster o datos de tipo vectorial.

➤ Datos ráster

Un dato ráster o simplemente ráster es un conjunto de celdas o píxeles las cuales tienen un número fijo de filas y columnas. Para el caso de las celdas, cada una tiene un valor numérico y un tamaño geográfico. Un ejemplo de un dato tipo ráster son las fotos aéreas o las imágenes satelitales, como la que se aprecia en la Figura 1.7.

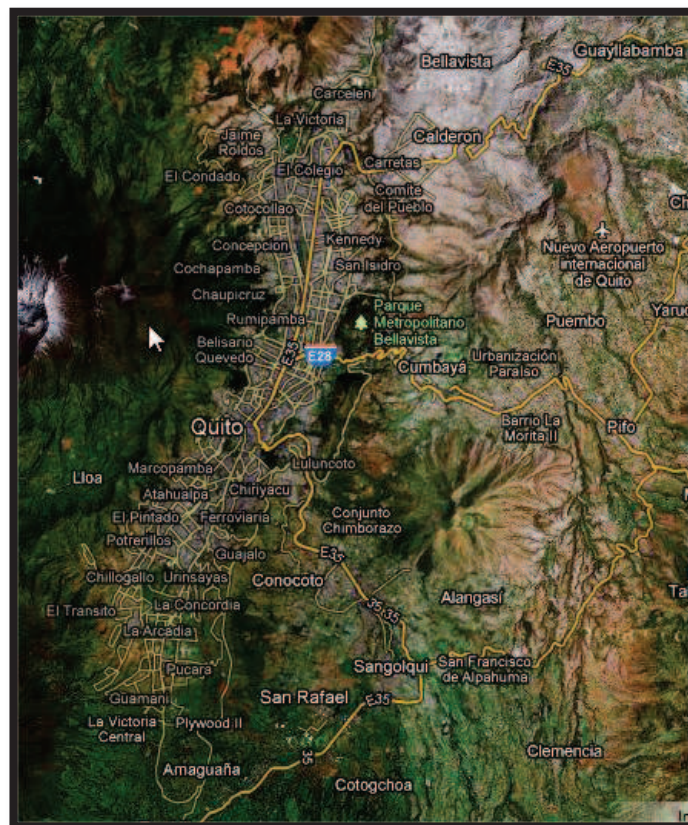


Figura 1.7 Imagen Satelital de la Ciudad de Quito

Fuente: Google (2012). Recuperado de : <https://maps.google.com.ec/>. (Diciembre 2012).

➤ Datos Vectoriales

Los datos vectoriales son una forma de definir la localización de un objeto en el espacio utilizando coordenadas, cada coordenada consta de un grupo de valores expresados en coordenadas esféricas o rectangulares dependiendo del estándar con el que se esté trabajando (x,y). Los tipos de datos vectoriales son:

- Punto, una coordenada (x,y).
- Línea, un par de coordenadas (x,y) conectadas entre sí.
- Segmento, varias líneas conectadas.
- Polígono, varias líneas unidas con la condición de que, el primer punto de la primera línea sea igual al último punto de la última.

Un ejemplo de datos vectoriales es mostrado en la Figura 1.8.

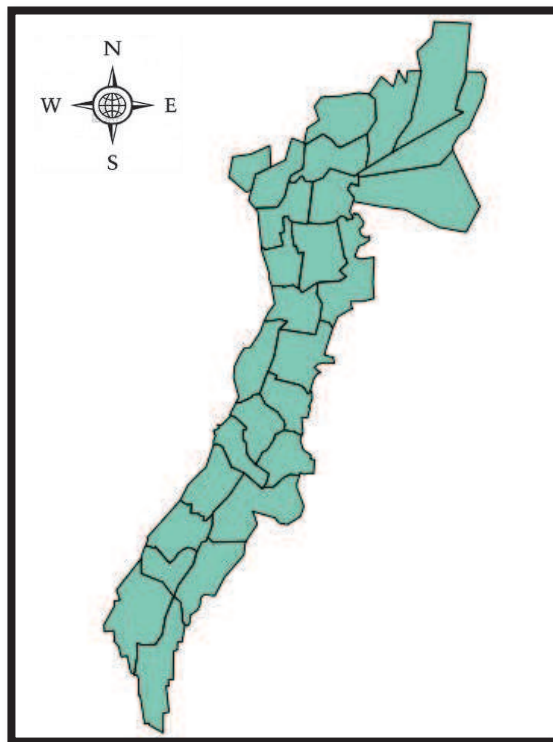


Figura 1.8 Conjunto de Datos Vectoriales que Definen la Zonificación de la Ciudad de Quito

Fuente: Adaptado de: *Anexo 5*, Contratos de Concesión proporcionados por la SUPERTEL. Enero 2012

Como se aprecia en la Figura 1.8, los datos vectoriales por si solos, no entregan información asimilable para un usuario común. Sin embargo, proyectando imágenes vectoriales (a manera de capas), sobre imágenes de tipo ráster, el usuario puede apreciar de manera clara lo que los datos vectoriales representan. Como lo que se observa en la Figura 1.9, por ejemplo.

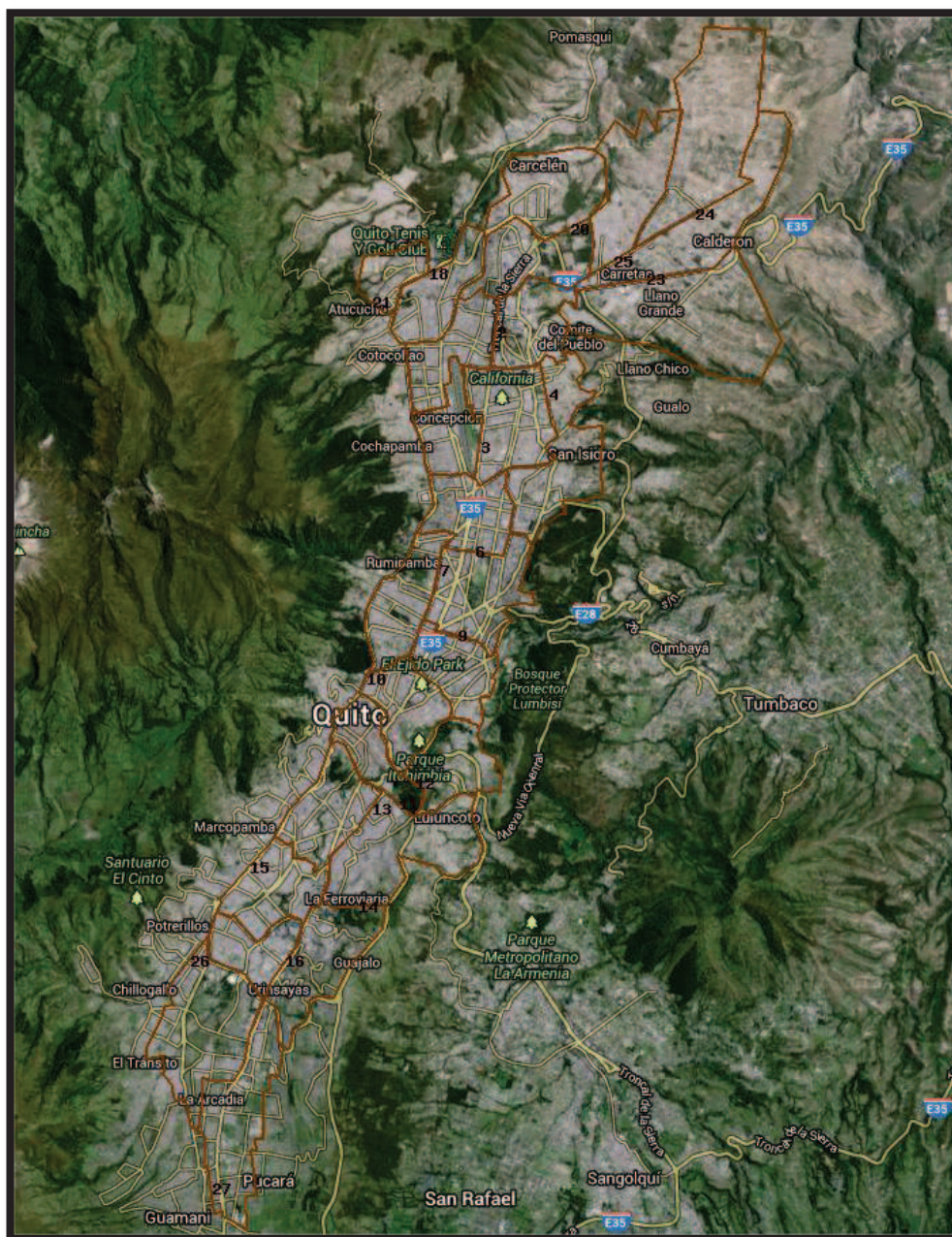


Figura 1.9 Mapa de Quito Sectorizado

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

1.2.3.1 Servidor de mapas

Un servidor de mapas es una plataforma que proporciona información geográfica de mapas, a través de transmisión de datos contenidos en protocolos estandarizados de comunicaciones, como es el caso de HTTP (Figura 1.10). Esta transmisión está basada en una arquitectura cliente servidor. [10].

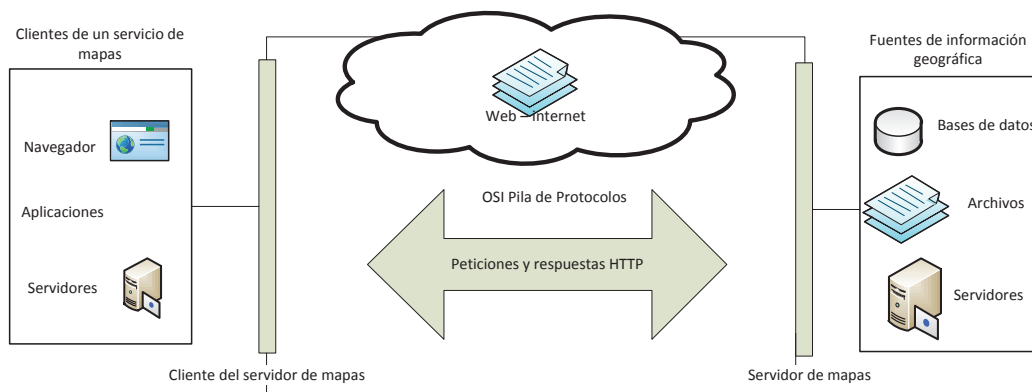


Figura 1.10 Servidor de Mapas - Internet Map Server (IMS)

Fuente: Adaptado de: *OpenGIS Web Map Service (WMS) Implementation Specification*. Versión Electrónica. Recuperado de: https://portal.opengeospatial.org/files/?artifact_id=1081&version. (Marzo 2012).

1.2.4 APLICACIÓN WEB

Una aplicación web tiene como principal característica la generación automática o bajo demanda de contenido; por ejemplo, si se habla de una tienda electrónica, en la cual constantemente se consultan, compran y venden productos, sería ineficiente que un administrador sea el encargado de actualizar los datos por cada transacción que realice la tienda, inclusive se podría tener problemas de inconsistencia de datos. Sin embargo, al manejar un esquema de aplicación web, la información se genera y actualiza dinámicamente a través de mensajes de petición y respuesta en cada transacción, con lo que se logra tener control sobre la presentación, transmisión y almacenamiento de la información. A manera de ejemplo, se esquematiza el funcionamiento de una aplicación web (caso de estudio: tienda electrónica) en la Figura 1.11 .

Las características relevantes de una aplicación web son:

- Creación y manejo de páginas personalizadas según el perfil que posea un usuario dentro de una organización (administrador, usuario común, visitante, etc.).
- Su arquitectura es de tipo cliente-servidor; en la cual un equipo, el servidor, espera solicitudes y envía respuestas, y los clientes son los que envían las solicitudes y reciben las respuestas.
- Una aplicación web también puede asumir el rol de medio de gestión de una organización; es decir, una aplicación web puede interactuar con diferentes equipos o programas informáticos dentro de una empresa, como pueden ser equipos de interconexión, bases de datos, etc. y administrarlos.

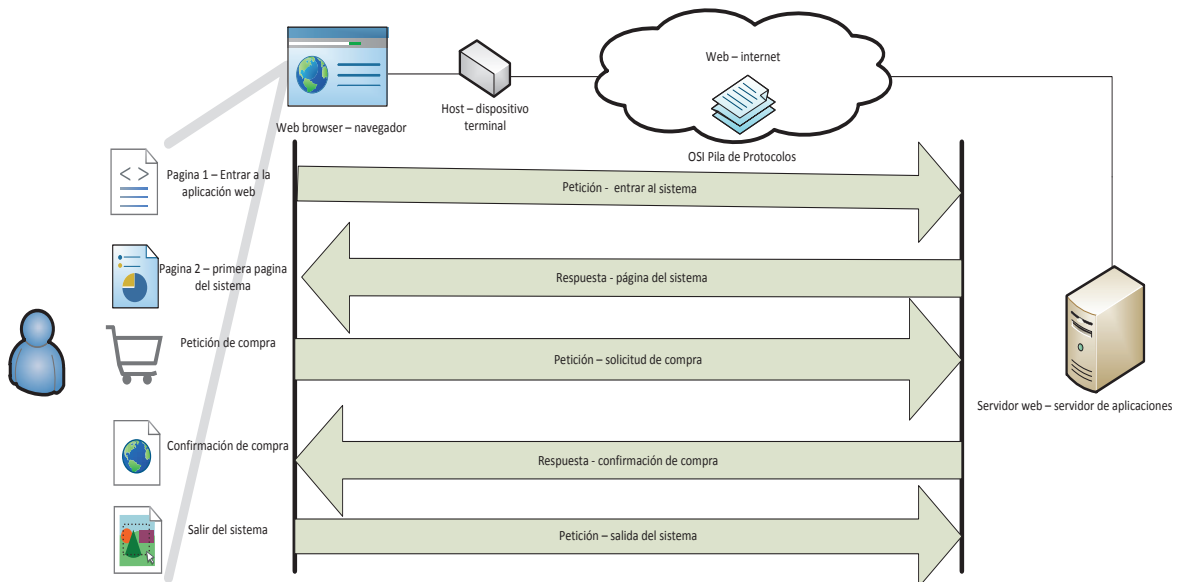


Figura 1.11 Funcionamiento Aplicación Web

Fuente: Esquema de Autoría propia (Arévalo D., y Guzmán D.).

En cuanto al diseño de una aplicación web, en la actualidad existen muchas herramientas de desarrollo, lenguajes, y entornos de desarrollo, etc. Que brindan soluciones cada vez más específicas a un problema en particular. Existen herramientas libres y comerciales.

Una aplicación web debe constituir un medio de fácil uso, fácil navegación dentro del sitio, agradable a la vista, siempre debe ofrecer algún valor agregado, ser

novedoso y enfocado a un t3pico espec3fico. Son algunos de los consejos que brindan los desarrolladores de aplicaciones web m3s experimentados.

Es fundamental aclarar que una aplicaci3n web, es un tipo de aplicaci3n cliente servidor, que requiere de un servidor web (de aplicaciones), para su funcionamiento. As3 tambi3n, distintas aplicaciones web pueden interactuar u ofrecer ciertas funcionalidades a otros sistemas a trav3s de la publicaci3n de servicios. Un servicio web constituye una manera de exponer funciones de aplicaciones usando la red mundial. Esto se logra utilizando protocolos abiertos, de manera que cualquier aplicaci3n con acceso a la Web que pueda interactuar con los servicios publicados. La ventaja de los servicios web es que permiten que aplicaciones escritas en diferentes lenguajes de programaci3n e implantadas en diferentes plataformas se comuniquen entre s3 en el amplio radio de acci3n de Internet.[10]

1.2.4.1 Web 2.0 y las Aplicaciones Web Colaborativas

Los conceptos de la web 1.0 surgieron debido a la masificaci3n del uso de la internet, los mismos que hacen referencia a como los usuarios acceden e interact3an con el contenido de la internet. As3 se define a la web 1.0, como la forma de acceder a la informaci3n en modo solo de lectura, esto quiere decir que el usuario no puede interactuar con el contenido de las p3ginas de determinado sitio web.

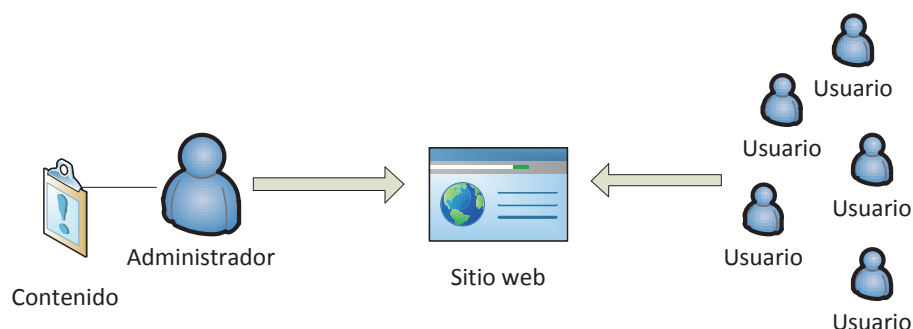


Figura 1.12 Representaci3n de Sitio Web 1.0

Fuente: Esquema de Autor3a propia (Ar3valo D., y Guzm3n D.).

Contrastando a este concepto nace el t3rmino web 2.0, el mismo que est3 relacionado a aplicaciones web que facilitan la interacci3n del usuario y el

contenido del sitio web; permitiendo a usuarios o comunidades de usuarios colaborar, compartir y generar contenido en sitios web.

Aplicación colaborativa o “*groupware*”, es un término que nace de trabajo en grupo, y hace referencia al desarrollo de tecnologías o aplicaciones que den soporte a la colaboración y comunicación de un grupo en el desempeño de cierta actividad (Figura 1.13). Un *groupware*, es la aplicación de la informática para dar soporte a trabajos cooperativos. Envuelve todo lo que hace referencia a la plataforma, software, hardware, servicios y soporte de procesamiento grupal.

Así se puede determinar que una aplicación web colaborativa es una aplicación que da soporte al trabajo cooperativo y comunicativo brindando su acceso a través de la web. Generalmente el trabajo en grupo incluye actividades como escribir, comunicarse oralmente, reunirse, compartir información y coordinar trabajo, este tipo de trabajo ocurre generalmente cuando las comunidades interactúan entre sí en tiempo real. Las reuniones a través de video conferencia pueden ser consideradas como un ejemplo de trabajo síncrono. Estos grupos pueden trabajar asincrónicamente como por ejemplo cuando se envía un mensaje de correo electrónico.[33]

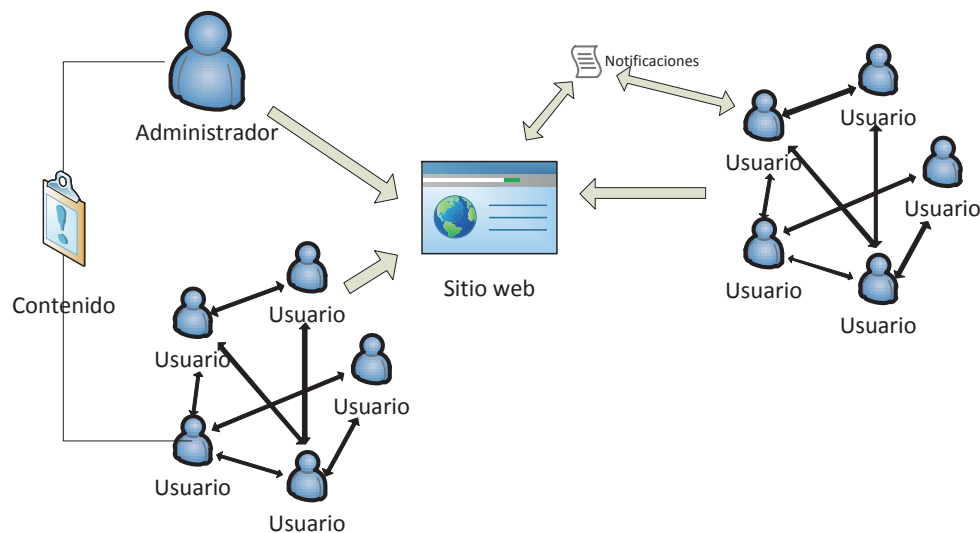


Figura 1.13 Representación de Sitio Web 2.0

Fuente: Esquema de Autoría propia (Arévalo D., y Guzmán D.).

1.3 SISTEMAS MÓVILES CELULARES

Un sistema celular se basa en brindar cobertura de servicio a un área a través de la utilización de estaciones base transmisoras-receptoras de ondas radioeléctricas. La zona donde se tiene cobertura y se brindará un determinado servicio es denominado como una celda o célula Figura 1.14.

En cada una de estas células se utiliza un determinado radiocanal⁴ de frecuencia, sub-banda de frecuencias dentro de la banda total que la operadora tendrá asignada. Debido a que el espectro electromagnético es un recurso limitado, los radiocanales también lo son, entonces se restringe el uso de un mismo radiocanal para estaciones bases diferentes, a menos de que estas estaciones base, estén lo suficientemente alejadas. De no cumplirse una distancia de reutilización⁵ se produce un fenómeno conocido como interferencia co-canal⁶.

La asignación de un canal de comunicación será dinámica entre los usuarios localizados en una determinada célula. Se define GOS (Grado de Servicio), como la probabilidad de encontrar un canal ocupado mientras se intenta establecer una comunicación.

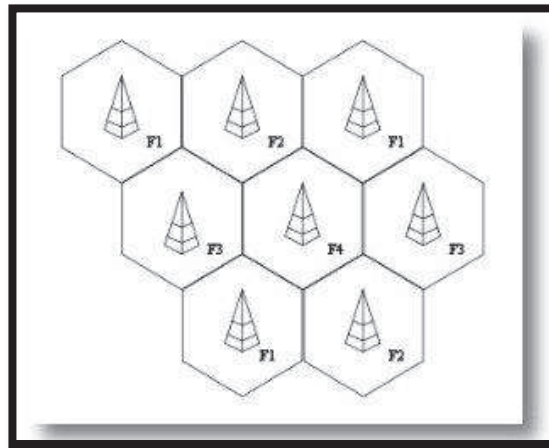


Figura 1.14 Celda

Fuente: *Geoposicionamiento GSM independiente de la red móvil (I): Introducción a la infraestructura de la red. GSM* Recuperado de: <http://www.kriptopolis.org/geoposicionamiento-gsm-1> (Agosto 2013).

⁴ Radiocanal: par de frecuencias utilizadas en una comunicación (enlace ascendente y enlace descendente).

⁵ Distancia de reutilización: mínima distancia entre dos células que compartan el mismo canal en la que la interferencia co-canal no afecta la comunicación.

⁶ Interferencia co-canal: Interferencia producida a una señal de determinada frecuencia por otra señal de la misma frecuencia, o en el mismo canal.

Para soportar una mayor densidad de usuarios en una celda, se puede dividir la célula añadiendo radiobases, teniendo así una mayor área de cobertura. Proceso conocido también como *Splitting*.

Un sistema móvil celular está constituido esencialmente por una central de conmutación, estaciones base y estaciones móviles. La central de conmutación controla y monitorea los radiocanales. Las estaciones base realizan funciones de transmisión y recepción directamente con las estaciones móviles. Las estaciones móviles son los equipos utilizados por los usuarios.

Una comunicación establecida no puede perderse debido al cambio de celda (Traspaso intercelular). El proceso que impide que una comunicación establecida, se corte debido a un traspaso de celda, se lo conoce como proceso de *HandOff* o *HandOver*.

1.3.1 EVOLUCIÓN DE LOS SISTEMAS CELULARES

La evolución de los sistemas celulares tiene su inicio en los años 80, con el transcurso del tiempo se han experimentado grandes avances en cuanto a tecnología y velocidades de transmisión, permitiendo mejoras en la calidad de los servicios prestados.

En la Figura 1.15 se detallan cada una de las generaciones a través de las cuales los sistemas celulares han venido evolucionando.

1.3.2 SISTEMA GLOBAL PARA LAS COMUNICACIONES MÓVILES (GSM)

GSM es un sistema de conmutación de circuitos, el cual soporta servicios de voz, datos, mensajes cortos, etc. Definiendo los mensajes de texto cortos como mensajes de hasta 160 caracteres y al servicio de datos con una velocidad de transferencia de 9.6 kbps.

1.3.2.1 Arquitectura de la Red GSM

En la arquitectura de la red GSM se identifica claramente que los componentes pueden ser agrupados en subsistemas, para los cuales se definen servicios, características y componentes. *Base Station Subsystem* – Subsistema de Estación Base (BSS), *Network Switching Subsystem* – Subsistema de Conmutación de Red (NSS), y *Operation & Service Subsystem* – Subsistema De

Operación y Servicio (OSS), son los tres subsistemas principales (esquematisados en la Figura 1.16).

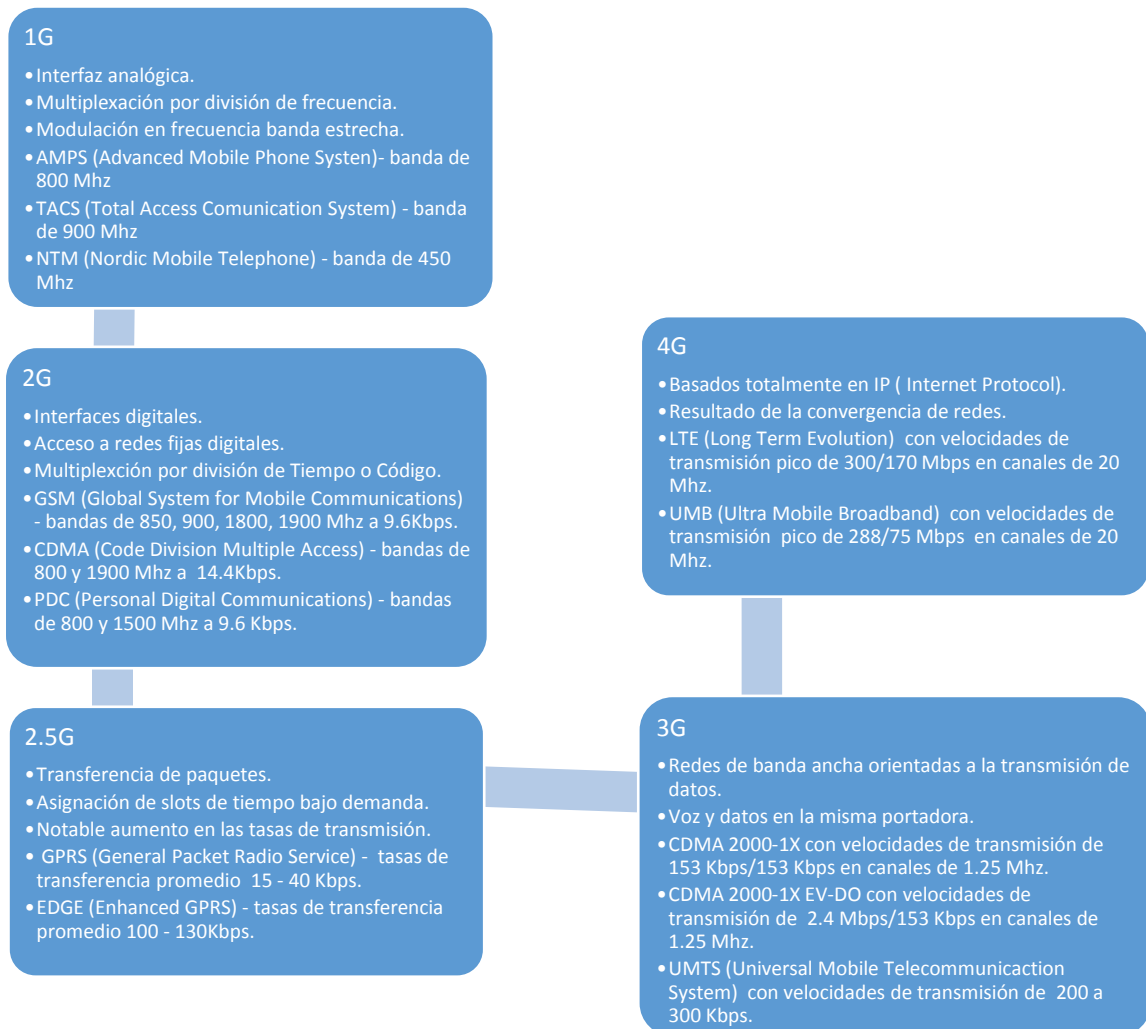


Figura 1.15 Características de la Evolución de los Sistemas Celulares

Fuente: Sinche, Soraya. *Apuntes Materia Comunicaciones Inalámbricas (2010)*.

Además, son parte de la arquitectura de una red GSM, el equipo terminal (conformado por la estación móvil y el módulo de identificación del abonado, conocido como tarjeta SIM), las interfaces y protocolos que se requieren para que cada uno de los componentes de la red interactúen. En la

Tabla 1.1 se describen los componentes mencionados.

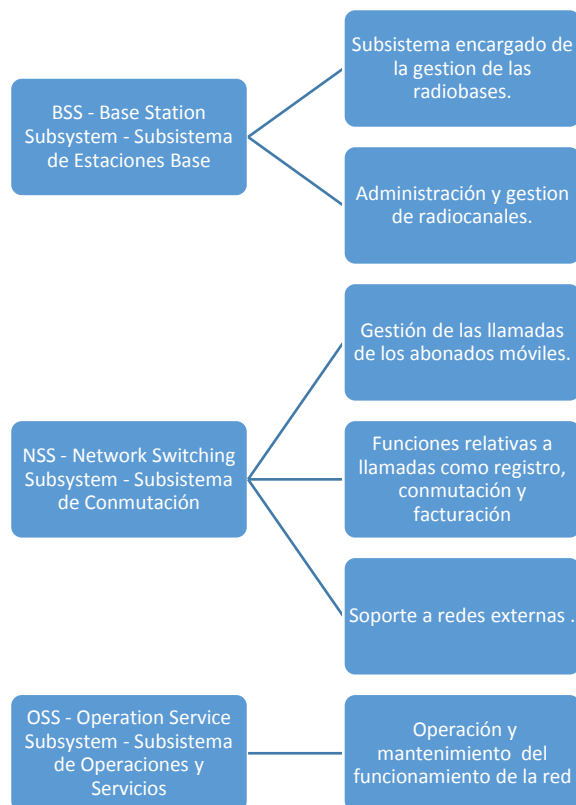


Figura 1.16 Subsistemas de la Red GSM

Fuente: Adaptado de: *Geoposicionamiento GSM independiente de la red móvil (I): Introducción a la infraestructura de la red GSM* Recuperado de: <http://www.kriptopolis.org/geoposicionamiento-gsm-1> (Agosto 2013).

**Tabla 1.1
Equipo Terminal de la Red GSM**

MS - Mobile Station - Estación Móvil	SIM - Subscriber Identity Module - Módulo de identidad del abonado
<ul style="list-style-type: none"> • Punto de entrada físico usado por el usuario GSM a la red móvil inalámbrica. • Gestiona la identidad del suscriptor a través del SIM. 	<ul style="list-style-type: none"> • Tarjeta asociada al abonado, identificándolo de manera única. • Identifica temporal e internacionalmente al abonado. • PIN & PUK, claves de desbloqueo. • Algoritmos de autenticación, generación de llaves y cifrado.

Fuente: Adaptado de: Arévalo Washima, Roberto Xavier. & Vizcaíno León, José Julio. *Estudio de tecnologías inalámbricas para zonas rurales del cantón Cuenca.* . (pp. 4, 5). (2007) Recuperado de: <http://dspace.ups.edu.ec/bitstream/123456789/181/2/Capitulo%201.pdf> (Marzo 2012).

En la Figura 1.17 se esquematizan los componentes de una red GSM y cómo interactúan entre sí.

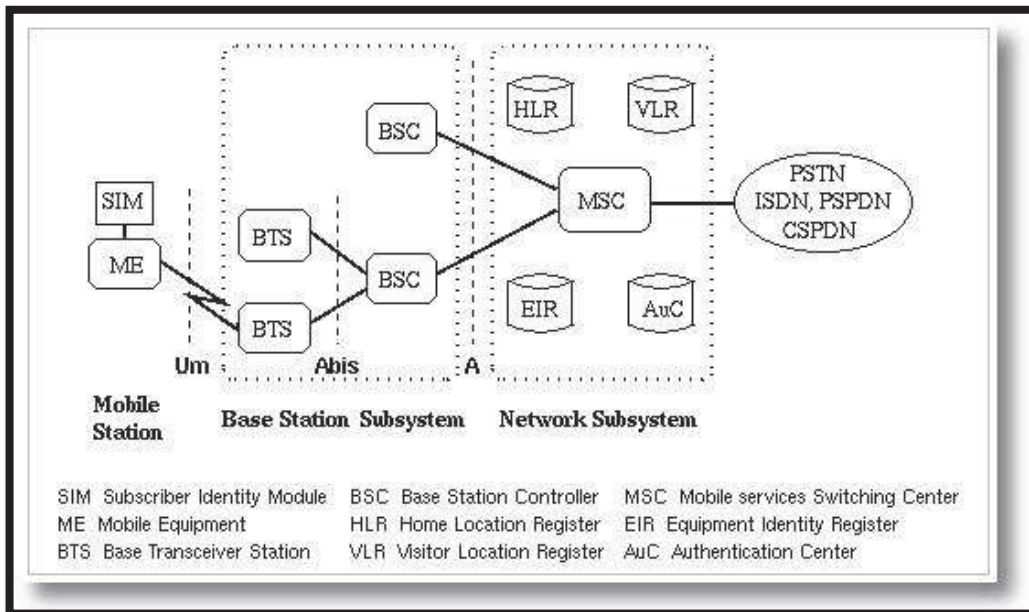


Figura 1.17 Componentes de la Red GSM

Fuente: *Geoposicionamiento GSM independiente de la red móvil (I): Introducción a la infraestructura de la red. GSM* Recuperado de: <http://www.kriptopolis.org/geoposicionamiento-gsm-1> (Agosto 2013).

1.3.3 ACCESO MÚLTIPLE POR DIVISIÓN DE CÓDIGO (CDMA)

En el año de 1995 la Asociación de Industrias de Telecomunicaciones (TIA) define el estándar *Interim Standard 95 (IS-95)*. En este documento se definen las bases para CDMA. Esta tecnología fue diseñada para mantener compatibilidad con AMPS. Inicialmente fue una tecnología desarrollada para aplicaciones militares, debido a su capacidad de mitigar los efectos de la interferencia y sus fortalezas en cuanto a seguridad, durante los procesos de transmisión.

CDMA basa su funcionamiento en la técnica de espectro extendido (*spread spectrum*). En esencia la técnica de espectro extendido utiliza un código, el cual es independiente de los datos que transmite la señal, al aplicar este código a la señal original se provoca un ensanchamiento de su espectro. En consecuencia, si una señal requiere en ancho de banda, un valor “n” como mínimo para su transmisión en banda base, luego de que la señal pasa por un proceso de ensanchamiento de espectro requerirá de un valor mucho mayor a “>>n”, como se esquematiza en la

Figura 1.18.

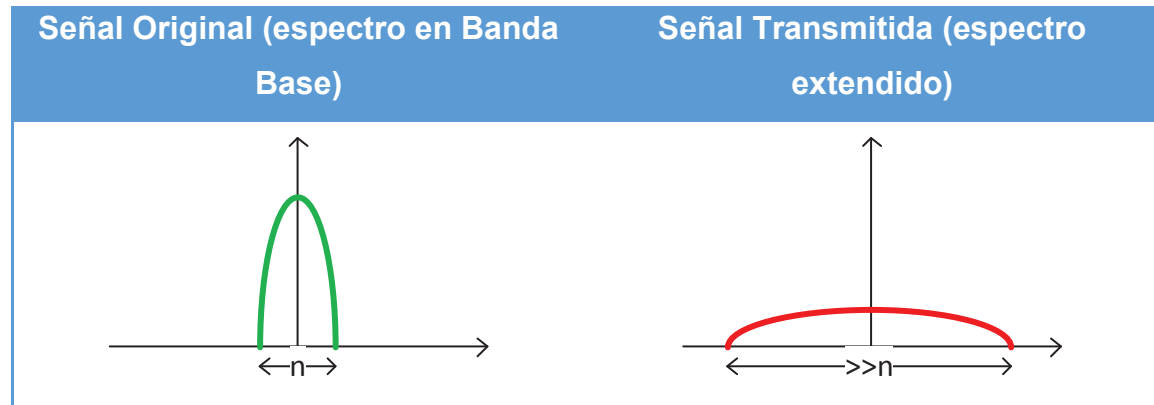


Figura 1.18 Comparación de Espectro de Señales

Fuente: Adaptado de: Alvarez, Robin. *Apuntes Materia Comunicaciones Inalámbricas (2010)*.

Es importante destacar que los códigos utilizados en CDMA, deben cumplir dos características fundamentales, deben ser códigos no-correlacionados⁷ u ortogonales⁸; y debe ser un código de alta velocidad.

Estos códigos son conocidos por todos los usuarios autorizados dentro de la red, gracias a este hecho CDMA ofrece un mayor grado de seguridad, evitando que escuchas no autorizados tengan acceso a la información transmitida.

1.3.3.1 Características Generales

La característica principal de un sistema CDMA es la utilización de todo el espectro de frecuencia durante todo el tiempo para su transmisión. Este hecho simplifica la planificación y diseño de la red; ya que no es necesario asignar diferentes valores de frecuencia para cada celda, como es el caso de FDMA, así tampoco es necesario establecer intervalos de uso de frecuencia, como es el caso de TDMA.

Cuando un usuario está utilizando el servicio de forma activa (se encuentra realizando una llamada, por ejemplo.), y se aleja de su estación base (hasta que el nivel de señal ha decrecido demasiado), se debe realizar un proceso que permita transferirlo hacia otra radio base (más cercana al usuario) que garantice la continuidad de su llamada, este proceso se conoce como proceso de *hand-off*.

⁷ La correlación mide el grado de similitud entre dos códigos.

⁸ Códigos que no interfieren unos con otros

Para el caso de CDMA, se conoce como *soft hand-off*, debido a que se comparte el mismo espectro en todas las celdas, el equipo móvil siempre recibe varias señales y simplemente escoge la mejor para continuar usándola. Por el contrario, cuando las celdas no operan a la misma frecuencia, el proceso de *hand-off* se vuelve más brusco ya que por un instante el usuario se desconecta de su señal original para conectarse a una nueva.

CDMA provee buena calidad en cuanto a transmisión de voz, además gracias a su proceso de codificación ayuda a prevenir fraudes. En CDMA todos los usuarios de la red comparten el mismo rango de frecuencia dentro del espectro radio eléctrico, pero las señales transmitidas se diferencian unas de otras porque están codificadas. A manera de analogía con la vida real, es como cuando varias personas se encuentran conversando utilizando idiomas diferentes dentro de una misma habitación.

En la red CDMA cada dispositivo móvil tiene asignado un código, mediante el cual ensancha su espectro. Además de los dispositivos móviles, las radio bases o estaciones base también conocen estos códigos.

Como se mencionó previamente los códigos deben ser ortogonales, ya que al realizar la correlación cruzada⁹ de un par de códigos diferentes el resultado será muy cercano a cero (si el resultado es igual a uno, los códigos son idénticos). De no ser así, el proceso de recuperación de la señal, en el lado del receptor se dificultaría, ya que al recibir varias señales provenientes de diferentes transmisores, no podría distinguir cual es la señal que le corresponde.

1.3.3.2 Arquitectura

La arquitectura de la red es representada a través de la Figura 1.19.

- a) Registro de Suscriptores Locales (HLR), es una base de datos donde se encuentra almacenada información sobre los usuarios de un sistema móvil celular.
- b) Registro de Suscriptores Visitantes (VLR) es una base de datos donde se encuentra almacenada información sobre los usuarios visitantes (usuarios

⁹ Análisis de la similitud de dos códigos (señales) en el dominio del tiempo.

que no han suscrito un contrato con la operadora) a un sistema móvil celular.

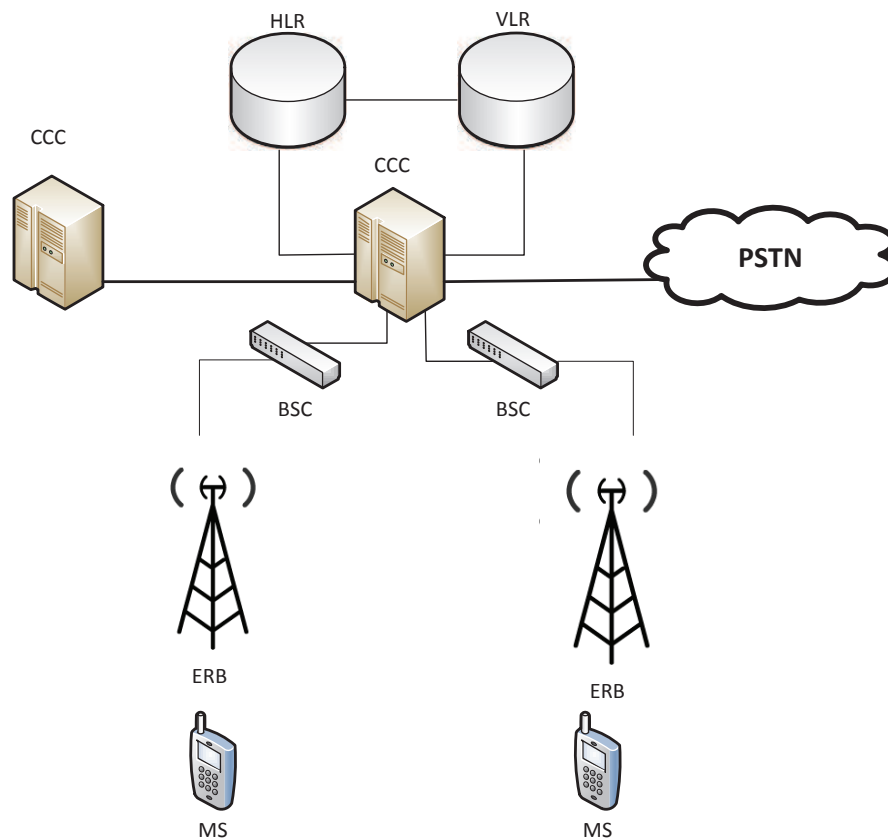


Figura 1.19 Arquitectura Básica CDMA

Fuente: CDMA: Arquitectura Recuperado de:

http://www.teleco.com.br/es/tutoriais/es_tutorialcdma/pagina_1.asp. (Enero 2013).

- c) Central de Conmutación y Control (CCC) es la central responsable por las funciones de conmutación y señalización para las estaciones móviles localizadas en un área geográfica designada como el área de la CCC.
- d) Controlador de Estación Base (BSC), su función es la de controlar a un grupo de ERBs. También suelen estar integradas dentro de la CCC.
- e) Estación Radio Base (ERB), brinda el servicio de comunicación con las estaciones móviles en una determinada área que constituye una célula.

- f) Estación Móvil, es el terminal utilizado por el suscriptor. La estación móvil es identificada por un MIN (Número de identificación de equipo móvil) y un número de serie electrónico (ESN). Cabe recalcar que estos números no son iguales, ya que el MIN es asignado por la empresa que brinda el servicio al usuario; mientras que el ESN es un identificador único, asignado por el fabricante del equipo.

Cada vez que se inicia un proceso de llamada, la ERB, verifica la validez de estos dos números y es así como se pueden prevenir fraudes y suplantación de identidades.

1.4 SERVICIO MÓVIL AVANZADO EN ECUADOR

El artículo 3 del Reglamento para la prestación del servicio móvil avanzado, Resolución No. 498-25-CONATEL-2002, define: “*Servicio Móvil Avanzado (SMA): es un servicio final de telecomunicaciones del servicio móvil terrestre¹⁰, que permite toda transmisión, emisión y recepción de signos, señales, escritos, imágenes, sonidos, voz, datos o información de cualquier naturaleza.*”

Los servicios finales de telecomunicaciones son servicios que proporcionan la capacidad completa para la comunicación entre usuarios. Para su operación estos servicios requieren mecanismos y elementos de conmutación.

1.4.1 OPERADORA

Operadora de red móvil u operadora, es la entidad que brinda el servicio móvil avanzado a los usuarios. La operadora, posee entera capacidad legal y tecnológica para la explotación de este servicio, dentro su zona de cobertura. La capacidad legal, hace referencia a que la operadora posee el permiso respectivo, obtenido mediante concesión (según reforma ley especial de telecomunicaciones Art. 4), para hacer uso de determinada(s) banda(s) del espectro radioeléctrico.

La Ley Especial de Telecomunicaciones en su Art. 2 define el espectro radioeléctrico como “*un recurso natural de propiedad exclusiva del Estado y como*

¹⁰Servicio móvil terrestre: Servicio móvil entre estaciones de base y estaciones móviles terrestres o entre estaciones móviles terrestres.

*tal constituye un bien de dominio público, inalienable e imprescriptible, cuya gestión, administración y control corresponde al Estado*¹¹.

La capacidad tecnológica de una operadora es toda la red de estaciones base propiedad de la operadora, mediante la cual garantiza el servicio a sus usuarios dentro de su área cobertura. Actualmente, en el Ecuador existen tres operadoras que brindan servicio de telefonía móvil celular y estas son: TELECSA (CNT-Alegro), OTECEL (Movistar) Y CONECEL (Claro).

1.4.2 CALIDAD DE SERVICIO (QOS)

El término calidad define la norma ISO 9000 como: *“grado en el que un conjunto de características (rasgos diferenciadores) inherentes (características permanentes) cumplen con los requisitos (necesidad o expectativa establecida, generalmente implícita u obligatoria)”*. Calidad de servicio (QoS) se define en la Recomendación UIT-T E.800 [1] como *“el efecto colectivo de rendimiento que determina el grado de satisfacción de un usuario del servicio.”* En general, la calidad de servicio se mide de una manera objetiva.

En las telecomunicaciones, calidad de servicio es por lo general una medida del rendimiento de los servicios prestados, no se incluye ningún mecanismo que contribuya a la mejora del rendimiento general del sistema y por lo tanto a mejorar la experiencia del usuario final. La totalidad de las características de un servicio que determinan el grado de satisfacción, las necesidades implícitas y explícitas de los usuarios hacia éste servicio se denominan calidad de servicio.

La calidad de servicio experimentada o percibida por el usuario nace de la diferencia entre lo que el cliente espera del servicio, sus expectativas y la experiencia real, es decir el servicio real que entrega el proveedor. La Figura 1.20 esquematiza la relación entre el cliente, el proveedor y la calidad del servicio.

Aspectos de la calidad de servicio durante el uso del servicio desde el punto de vista del cliente se esquematizan en la Figura 1.21.

¹¹ Ley Especial de telecomunicaciones reformada. 13 de Octubre 2011



Figura 1.20 Esquema de la Relación entre Satisfacción del Cliente, y QoS

Fuente: ETSI TS 102 250-1 V1.2.1. *Speech Processing, Transmission and Quality Aspects (STQ); QoS aspects for popular services in GSM and 3G networks; Part 1: Identification of Quality of Service criteria*

Recuperado de:

http://www.etsi.org/deliver/etsi_ts/102200_102299/10225001/01.02.01_60/ts_10225001v010201p.pdf
(Diciembre 2011).



Figura 1.21 Fases de la Utilización de los Servicios desde el Punto de Vista del Cliente

Fuente: ETSI TS 102 250-1 V1.2.1. *Speech Processing, Transmission and Quality Aspects (STQ); QoS aspects for popular services in GSM and 3G networks; Part 1: Identification of Quality of Service criteria.* (pp. 10).

Recuperado de:

http://www.etsi.org/deliver/etsi_ts/102200_102299/10225001/01.02.01_60/ts_10225001v010201p.pdf
(Diciembre 2011).

- Disponibilidad de la red: posibilidad de que el servicio sea ofrecido al usuario final.

- Accesibilidad de la red: probabilidad de que el usuario realice un registro exitoso dentro de la red.
- Accesibilidad del servicio: acceso garantizado e inmediato, a el/los servicios que el usuario desea utilizar.
- Sostenimiento del servicio: describe la terminación del servicio, de acuerdo o en contra de la voluntad del usuario.
- Integridad del servicio: describe la calidad del servicio durante su uso.

Es importante analizar la interacción entre todos los aspectos descritos, asumiendo que la red está disponible (existe cobertura), el equipo usuario intentará acceder a la red con el fin de registrarse. Una vez registrado en la red, el usuario intentará utilizar los servicios que la red ofrece (llamada de voz, envío de mensajes, etc.), cabe resaltar que si el usuario está registrado en la red no implica que los servicios requeridos estén disponibles.

Los indicadores de calidad de servicio no necesariamente son capaces de mostrar la experiencia real que está recibiendo un usuario, ya que los indicadores por sí mismos sólo son capaces de mostrar niveles de servicio de los diferentes factores que influyen en la experiencia del usuario y no la experiencia propiamente. De aquí la necesidad de investigar cual es la verdadera experiencia del cliente frente a un servicio.

1.4.3 CALIDAD DE LA EXPERIENCIA (QoE)

Además del término QoS, el término calidad de experiencia (QoE) se utiliza a menudo hoy en día con el fin de subrayar el carácter puramente subjetivo de evaluación de la calidad de las telecomunicaciones y su enfoque en la perspectiva del usuario del valor total del servicio prestado. QoE se define en la Recomendación UIT-T P.10/G.100 [I.2] en el Apéndice I como: *“la aceptabilidad global de una aplicación o servicio, tal como la percibe subjetivamente el usuario final. Incluye los efectos completos del sistema de extremo a extremo (cliente, el terminal, la red, la infraestructura de servicios, etc) y puede ser influenciada por las expectativas del usuario y el contexto.”*

Por lo tanto la calidad de la experiencia se mide subjetivamente por el usuario final y puede variar de un usuario a otro. Sin embargo, se estima a menudo usando mediciones objetivas. La inclusión del propio usuario a la calidad global de las telecomunicaciones, extiende la calidad de servicio objetiva a un concepto de calidad totalmente subjetivo de la experiencia. La calidad de la experiencia es diferente de un usuario a otro, ya que está influenciado por las experiencias y expectativas de los usuarios individuales de carácter personal.

La definición de calidad de servicio, incluye el grado de satisfacción de un usuario con un servicio. Por lo tanto, los aspectos no técnicos se incluyen, como por ejemplo, el entorno del usuario, sus expectativas, la naturaleza del contenido y su importancia. Pero la mayoría de los proveedores de servicios usaron el QoS sólo en relación a la interacción con el usuario del servicio actual con el fin de cotejar si las necesidades del usuario se han cumplido por la implementación del servicio de un proveedor (como es percibido por el usuario). Así que hubo un fuerte enfoque en el rendimiento real de la red y su influencia inmediata en los aspectos perceptibles de usuario mientras se omiten aspectos directamente subjetivos relacionados con el servicio.

La experiencia del cliente abarca todos los aspectos de las oferta de una empresa: la calidad del servicio al cliente, la publicidad, la presentación, las características del producto o servicio, la facilidad de uso y la confiabilidad. La experiencia del cliente es la respuesta interna y subjetiva de los clientes ante cualquier contacto directo o indirecto con una empresa. El contacto directo generalmente ocurre en el proceso de compra, uso y servicio, y suele ser iniciado por el cliente.

La satisfacción del cliente es un sentimiento personal de complacencia o decepción resultantes de la evaluación de los servicios prestados por una organización en relación a las expectativas.

1.4.3.1 Gestión de la Calidad de la Experiencia

El problema radica en que medir la satisfacción del cliente no está especificado. La satisfacción del cliente es esencialmente la culminación de una serie de experiencias del cliente; o podría decirse el resultado neto de las experiencias

buenas. Se produce cuando la diferencia entre las expectativas de los clientes y sus experiencias posteriores han sido cerradas.

Son los propios clientes, con sus experiencias previas (expectativas positivas o negativas), quienes deben ser monitoreados y sondeados, pero no se debe limitar a medir y distribuir esta información, el aspecto esencial es dar uso a la información recopilada y presentar soluciones a los problemas detectados. De aquí la necesidad de implementarse herramientas tecnológicas que recopilen analicen y distribuyan datos de gestión de la experiencia del cliente, así como el monitoreo de sus avances.

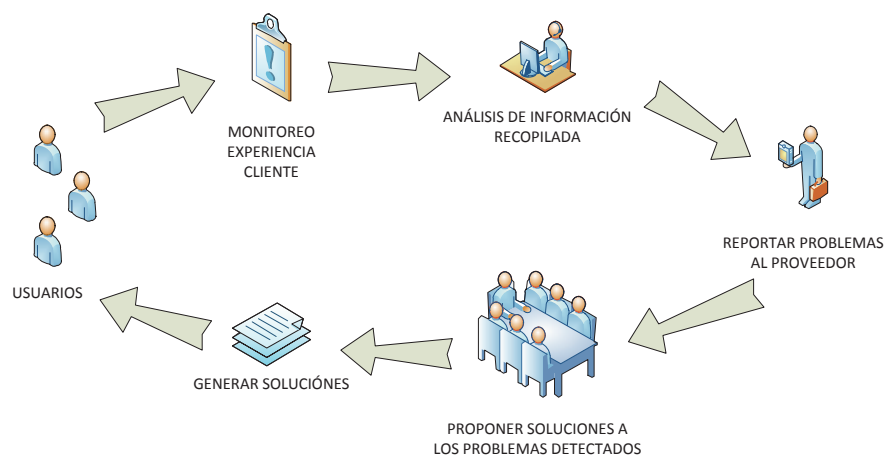


Figura 1.22 Gestión de la Calidad de la Experiencia

Fuente: Adaptado de; Meyer, Christopher., Schwager, Andre. *Understanding Customer Experience*. (Febrero 2007) (pp. 4) Recuperado de: <http://www.dlss.univr.it/documenti/Avviso/all/all845856.pdf> (Diciembre 2011).

Es importante destacar que en la actualidad todos los procesos de gestión de la calidad de experiencia ayudan a mejorar la percepción del cliente frente a los servicios o productos que consume.

1.4.4 SUPERINTENDENCIA DE TELECOMUNICACIONES SUPERTEL

La Constitución de la República del Ecuador, dispone:

- En el Artículo 66 se reconoce y garantiza a las personas: "25. El derecho a acceder a bienes y servicios públicos y privados de calidad, con eficiencia, eficacia y buen trato, así como a recibir información adecuada y veraz sobre su contenido y características".

- Artículo 213: "Las superintendencias son organismos técnicos de vigilancia, auditoría, intervención y control de las actividades económicas, sociales y ambientales, y de los servicios que prestan las entidades públicas y privadas, con el propósito de que estas actividades y servicios se sujeten al ordenamiento jurídico y atiendan al interés general. Las superintendencias actuarán de oficio o por requerimiento ciudadano. Las facultades específicas de las superintendencias y las áreas que requieran del control, auditoría y vigilancia de cada una de ellas se determinarán de acuerdo con la ley ..."
- Artículo 314: "El Estado será responsable de la provisión de los servicios públicos de agua potable y de riego, saneamiento, energía eléctrica, telecomunicaciones, vialidad, infraestructuras portuarias y aeroportuarias, y los demás que determine la ley. El Estado garantizará que los servicios públicos y su provisión respondan a los principios de obligatoriedad, generalidad, uniformidad, eficiencia, responsabilidad, universalidad, accesibilidad, regularidad, continuidad y calidad. El Estado dispondrá que los precios y tarifas de los servicios públicos sean equitativos, y establecerá su control y regulación".

En la Ley Especial de Telecomunicaciones en su artículo 34 se crea la Superintendencia de Telecomunicaciones y en su artículo 35 define sus funciones que son:

- a. Cumplir y hacer cumplir las resoluciones del Consejo Nacional de Telecomunicaciones (CONATEL).
- b. El control y monitoreo del espectro radioeléctrico.
- c. El control de los operadores que exploten servicios de telecomunicaciones.
- d. Supervisar el cumplimiento de los contratos de concesión para la explotación de los servicios de telecomunicaciones.
- e. Supervisar el cumplimiento de los contratos de concesión para la explotación de los servicios de telecomunicaciones.

- f. Controlar la correcta aplicación de los pliegos tarifarios aprobados por el CONATEL.
- g. Controlar que el mercado de las telecomunicaciones se desarrolle en un marco de libre competencia, con las excepciones señaladas en esta Ley.
- h. Juzgar a las personas naturales y jurídicas que incurran en las infracciones señaladas en esta Ley y aplicar las sanciones en los casos que correspondan.
- i. Las demás que le asigne la Ley y el Reglamento.

En base a estas funciones, la SUPERTEL, es el ente encargado del constante monitoreo y control de los servicios de telecomunicaciones prestados por las operadores a nivel nacional.

Para realizar este proceso de monitoreo en lo referente a servicio móvil avanzado y los parámetros de calidad de servicio la SUPERTEL implementó una metodología de tipo *drive test*. En esencia un *drive test*, son un conjunto de pruebas que se realizan en campo, mediante el uso de un vehículo, en el cual se incorporan equipos de medición, equipos de cómputo, GPS¹² y teléfonos celulares; con el objetivo de recabar información que permita determinar si los servicios que presta una operadora cumplen con ciertos parámetros técnicos o no.

Para poder determinar de una forma objetiva si los servicios prestados por las tres operadoras, a nivel nacional, son aceptables; la Secretaría Nacional de Telecomunicaciones (SENATEL) define una lista de parámetros de tipo técnicos y otros de apreciación por parte del usuario. Estos parámetros son establecidos en mutuo acuerdo, en los contratos de concesión firmados entre la operadora que va a brindar sus servicios y la SENATEL. La SUPERTEL es la encargada de supervisar el cumplimiento de estos parámetros y de imponer sanciones en caso de incumplimiento por parte de las operadoras.

¹² GPS, Global Positioning System, Sistema de Posicionamiento Global.

1.4.5 PARÁMETROS DE QoS

Los parámetros de calidad de servicio son valores que fijan las condiciones mínimas de calidad para la prestación de los servicios concesionados (Servicio Móvil Avanzado y Servicio telefónico de larga distancia internacional). Actualmente la SUPERTEL maneja siete parámetros técnicos de calidad de servicio los cuales se detallan en la Tabla 1.2.

Tabla 1.2
Parámetros de QoS – SUPERTEL

No	Parámetro de Calidad
1	Porcentaje de Llamadas Establecidas
2	Tiempo de Establecimiento de Llamada
3	Porcentaje de Llamadas Caídas
4	Zona de Cobertura
5	Calidad de Conversación
6	Porcentaje de Mensajes Cortos con Éxito
7	Tiempo Promedio de Entrega de Mensajes Cortos

Fuente: Adaptado de: *Anexo 5*, Contratos de concesión proporcionados por la SUPERTEL. Enero 2012.

1.4.5.1 Determinación de las Zonas de Medición en Ciudades

Para definir el tamaño del área de una zona de medición, interesa saber cuánto recorre un vehículo al realizar un total de 100 llamadas (estimación confiable del número de llamadas de prueba que deben realizarse por cada zona). En la Tabla 1.3 se detallan los intervalos de tiempo para cada fase de una llamada de prueba.

Tabla 1.3
Estimación de Duración de cada Llamada de Prueba

Detalle de Intervalo	Tiempo (s)
Tiempo de establecimiento (seg):	12
Duración de la llamada	45
Tiempo entre llamadas	15
Total	72

Fuente: Adaptado de: *Anexo 5*, Contratos de concesión proporcionados por la SUPERTEL. Enero 2012

El tiempo total de una llamada de prueba será de 72s, por lo tanto el tiempo que tarda el vehículo en recorrer una zona realizando las 100 llamadas de prueba es de: 7200 segundos \approx 2 horas (velocidad promedio de 40Km/h). Finalmente, se concluye que por cada zona de medición es necesario recorrer 80 Km de distancia para completar las 100 llamadas. Teóricamente existe 100 metros de distancia entre una cuadra y otra, con esta información se realiza pruebas con un área cuadrada de 1 Km². Si una zona ideal es un cuadrado de lado 1 kilómetro, se concluye que alcanzan 10 cuadras en cada lado.

Si se realiza un camino como el trazado en una zona cuadrada de 1Km de lado (1km²), se recorren horizontal y verticalmente una distancia total de 20Km, en total se recorren 40Km, con este dato se determina de manera proporcional que la distancia total es 4 veces mayor que la distancia recorrida al tener 1 Km² de área, por lo tanto el área de la zona deberá ser 4 veces mayor que 1Km². Esto es 4 Km². En la Figura 1.23, se esquematiza el recorrido del vehículo durante las mediciones dentro de una zona de 4km².

Además se deben tomar en cuenta las siguientes consideraciones:

- Determinar el tipo de población (Urbana / Rural), según la información de división Política Administrativa elaborada por el INEC.
- Una zona no puede estar compuesta al mismo tiempo por sitios calificados por el INEC como urbanos y rurales. Se debe cuidar que una zona no se sobreponga con otras zonas.
- El recorrido se lo hará sin repetir la misma trayectoria.
- Estas zonas deberán utilizarse para realizar todas las mediciones de calidad del servicio detalladas en este instructivo.

La tolerancia en el porcentaje de cobertura es 10% para zonas rurales y 5 % para zonas urbanas.

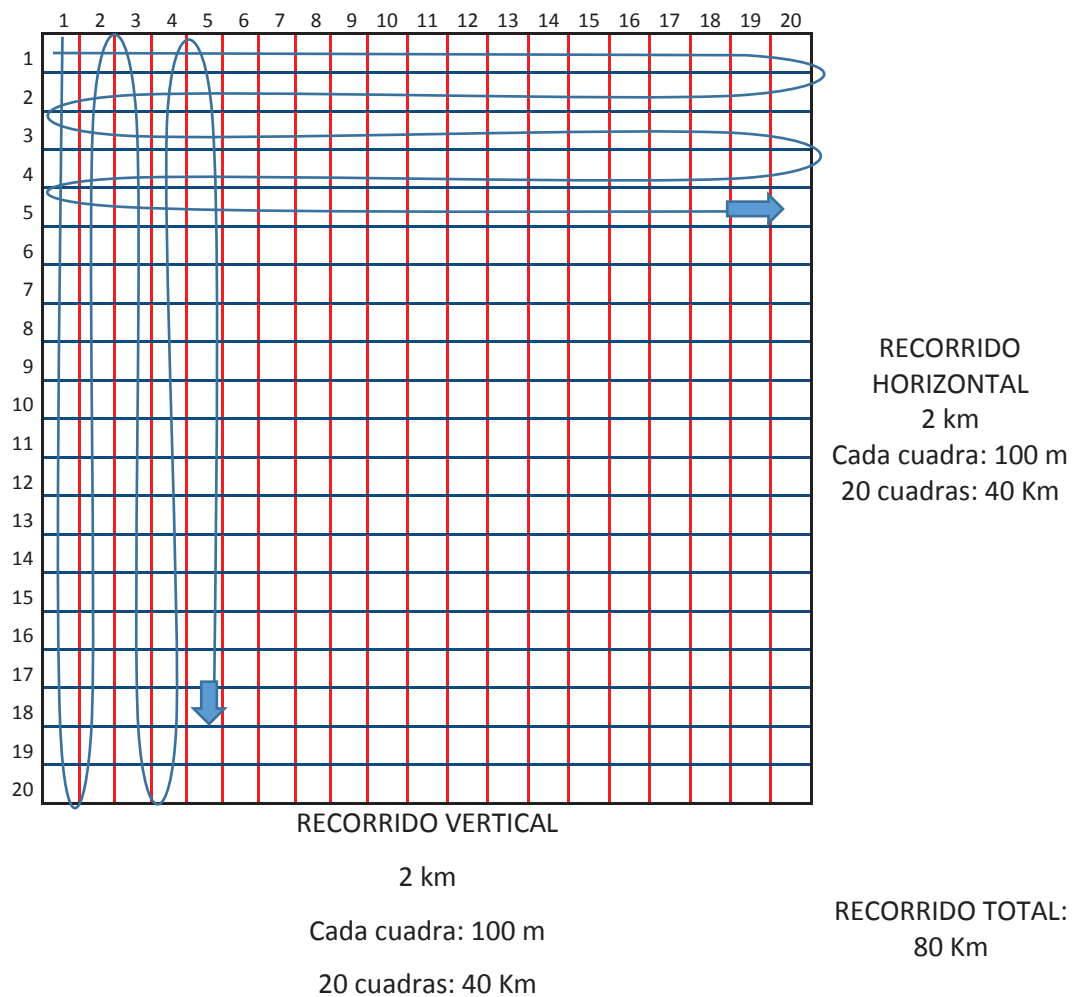


Figura 1.23 Recorrido Ideal para Determinar una Zona de Medición

Fuente: Adaptado de: Anexo 5, Contratos de concesión proporcionados por la SUPERTEL. Enero 2012

1.4.5.2 Zona de Cobertura

Es el área que la concesión ofrece a un usuario terminal dentro de la cual se deberá definir valores objetivos de nivel de señal que permitan la prestación del servicio concesionado. Los valores objetivos se detallan en la Tabla 1.4. Se medirá el parámetro nivel de señal para tecnología GSM y Ec/Io para CDMA/UMTS sobre el canal de control.

Las muestras válidas, se determinarán considerando los siguientes criterios:

- Se eliminan las muestras que tengan el valor “cero” o están “vacíos” en los parámetros de medición de cobertura (Rx Level o Ec/Io).

- Se eliminarán las muestras de frecuencias de canales de control que no correspondan a la banda de frecuencias concesionadas al operador.
- Se consideran como muestras válidas aquellas que tienen un valor de Rx Level/ o Ec/Io, asociado a una determinada coordenada.

Tabla 1.4
Parámetro Zona de Cobertura

Parámetro de Calidad	Valor Objetivo	Frecuencia de Medición	Tamaño de la Muestra
Zona de cobertura	Urbana: $\geq 95\%$ Rural o carreteras $\geq 90\%$	Trimestral	Las muestras válidas de nivel señal, se obtendrán sacando secuencialmente el valor promedio de las mediciones de cobertura que se registran cada 10 metros.

Fuente: Adaptado de: *Anexo 5*, Contratos de Concesión proporcionados por la SUPERTEL. Enero 2012

Tabla 1.5
Valores de Nivel de Señal, Parámetro Cobertura

Parámetro	GSM (nivel de señal)	CDMA/UMTS (Ec/Io)	Código de Colores
Urbano:	≥ -85 dBm.	≥ -14 dB	Verde
Zonas rurales	< -85 dBm y > -98 dBm.	< -14 dB y > -17 dB.	Amarillo
Sin servicio o mala cobertura	≤ -98 dBm.	≤ -17 dB.	Rojo

Fuente: Adaptado de: *Anexo 5*, Contratos de Concesión proporcionados por la SUPERTEL. Enero 2012

La Sociedad Concesionaria realizará las mediciones por medio de *drive test* conjuntamente con un delegado de la SUPERTEL, sobre la base de cobertura presentada y reportada a sus usuarios en su publicidad. Dentro de la zona de cobertura reportada por la Sociedad Concesionaria, la SUPERTEL establecerá las

rutas sobre las cuales se realizará la medición de posición, nivel de señal, y velocidad, preferentemente cada segundo. El *drive test* se realizará a unas velocidades máximas de 40 kilómetros por hora en zonas urbanas y 60 kilómetros por hora en carreteras y zonas rurales. El 90% de las muestras tomadas deberán estar dentro de los límites de velocidad establecidos.

1.4.5.3 Porcentaje de Llamadas Establecidas

El parámetro hace referencia al porcentaje de llamadas establecidas de manera exitosa respecto al número de intentos de llamadas en un período de medición. Se considera llamada establecida de manera exitosa cuando:

- El terminal llamado contesta.
- Si el terminal llamado está ocupado, fuera de aérea de cobertura, no contesta, con servicio restringido; la llamada se encamina al respectivo casillero de voz.
- El número marcado no existe y se recibe una notificación del particular.

Cuando debido a causas inherentes a la red (red fuera de servicio, congestionada¹³, etc.), la llamada se encamina a un casillero de voz o a un sistema de respuesta interactiva IVR; no se considera llamada establecida.

Tabla 1.6
Parámetro Porcentaje de Llamadas Establecidas

Parámetro de Calidad	Valor Objetivo	Frecuencia de Medición	Tamaño de la Muestra
Porcentaje de Llamadas Establecidas	>= 95%	Trimestral	Número mínimo de llamadas: 25 llamadas/km ²

Fuente: Adaptado de: *Anexo 5*, Contratos de Concesión proporcionados por la SUPERTEL. Enero 2012

Un aspecto importante es que las llamadas objeto de este análisis deberán originarse y terminar en la misma red de la sociedad concesionaria. Los

¹³ Cuando el número de usuarios dentro de la red supera la capacidad máxima de la misma.

terminales de destino serán números de prueba de OTECEL, CONECEL y TELECSA.

➤ **Método de Medición**

Llamadas de prueba ejecutadas a lo largo de un *drive test*. Las llamadas de prueba no deberán efectuarse en días atípicos (festividades, eventos especiales, etc.). Las pruebas se deben efectuar en escenarios totalmente controlados, de tal forma que se eliminan todas las posibilidades de que las llamadas no se establezcan por causas inherentes al usuario; es decir, se garantiza:

- Que el terminal se encuentra encendido.
- Dentro del área de cobertura de la operadora.
- Que se encuentra activado.
- Está habilitado de efectuar y recibir llamadas dentro de la red de la operadora.

El procesamiento de la medición debe realizarse, considerando que se obtendrá un valor del porcentaje de llamadas establecidas para cada zona de medición.

1.4.5.4 Tiempo de Establecimiento de Llamadas

Es el intervalo de tiempo medido en segundos que transcurre entre el instante en el que el usuario acciona el pulsador de envío de llamada luego de marcar el número seleccionado, y la recepción del tono de control de llamada. Su valor porcentual representa el porcentaje de llamadas establecidas dentro de un intervalo de tiempo definido ($\leq 12s$). Se detalla en la Tabla 1.7.

1.4.5.5 Porcentaje de Llamadas Caídas

El parámetro hace referencia al porcentaje de llamadas caídas con respecto al número total de llamadas establecidas, en un mes. Se considera como una llamada caída, cuando una llamada establecida no puede mantenerse por motivos atribuibles a la red de la sociedad concesionaria. Se detalla en la Tabla 1.8.

Tabla 1.7
Parámetro Tiempo de Establecimiento de Llamada

Parámetro de Calidad	Valor Objetivo	Frecuencia de Medición	Tamaño de la Muestra
Tiempo de Establecimiento de Llamada	$\geq 95\%$	Trimestral	Número mínimo de llamadas: 25 llamadas/km ²

Fuente: Adaptado de: *Anexo 5*, Contratos de Concesión proporcionados por la SUPERTEL. Enero 2012

Tabla 1.8
Parámetro Porcentaje de Llamadas Caídas

Parámetro de Calidad	Valor Objetivo	Frecuencia de Medición	Tamaño de la Muestra
Porcentaje de Llamadas Caídas	RBSs ¹⁴ tipo A $\leq 2\%$ RBSs tipo B $\leq 5\%$ RBSs tipo C $\leq 7\%$	Trimestral	Número mínimo de llamadas: 25 llamadas/km ² .

Fuente: Adaptado de: *Anexo 5*, Contratos de Concesión proporcionados por la SUPERTEL. Enero 2012

La clasificación A, B, C; hace referencia al tipo de radiobase, donde se realiza la medición.

- Tipo A (Figura 1.24) : radiobases adyacentes en todo su perímetro de área de servicio. Una radiobase adyacente, es aquella cuyos parámetros de operación garantizan la continuidad en la llamada.
- Tipo B (Figura 1.25): posee radiobases adyacentes, pero no cubren todo el perímetro del área de servicio.
- Tipo C (Figura 1.26): no posee radiobases adyacentes.

¹⁴ RBS acrónimo de Radio Base Station

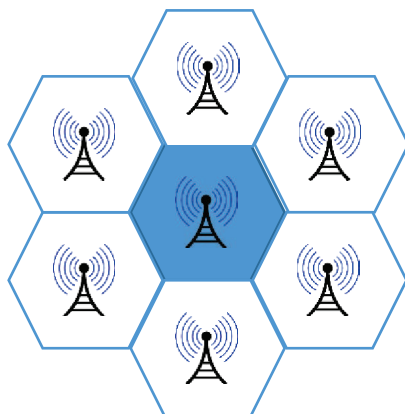


Figura 1.24 Esquema de una Radiobase Tipo A

Fuente: Adaptado de: Anexo 5, Contratos de Concesión proporcionados por la SUPERTEL. Enero 2012.

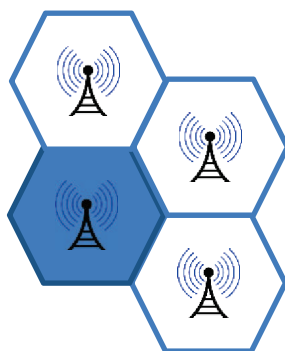


Figura 1.25 Esquema de una Radiobase Tipo B

Fuente: Esquema de Autoría Propia (Arévalo, D. y Guzmán, D.). Adaptado de: Anexo 5, Contratos de Concesión proporcionados por la SUPERTEL. Enero 2012.



Figura 1.26 Esquema de una Radiobase Tipo C

Fuente: Adaptado de: Anexo 5, Contratos de Concesión proporcionados por la SUPERTEL. Enero 2012.

➤ Método de medición

Para cada zona de medición las administraciones regionales deberán determinar si se trata de una zona servida por radiobases tipo A, B o C, a fin de obtener el

valor máximo en base al cual se evaluará este parámetro de calidad; en caso que una zona esté servida simultáneamente por dos tipos de radiobases, se escogerá la menos restrictiva en el parámetro de calidad.

Se deberá incluir un gráfico en el que se muestren los sitios donde se produjeron las llamadas caídas, en cada uno de ellos se deberá incluir el Identificador único de celdas (*Cell Id*). El porcentaje de llamadas caídas se obtendrá del total de llamadas caídas sobre el total de llamadas establecidas, para cada zona de medición.

1.4.5.6 Porcentaje de Mensajes Cortos con Éxito

Se define un servicio de mensajes cortos o *Short Message Service* (SMS), como un servicio mediante el cual los usuarios de los teléfonos móviles pueden intercambiar mensajes cortos de texto, con un tamaño máximo de 160 caracteres en GSM.¹⁵ El proceso de envío y recepción de mensajes se lo realiza a través de un centro de servicio de mensajes que en esencia es una base de datos que almacena los mensajes provenientes de los usuarios y los reenvía a sus destinatarios. Esta base datos además contiene información de origen, destinatario, fecha de recepción y envío.

El parámetro porcentaje de mensajes cortos con éxito, es el porcentaje del número de mensajes cortos recibidos exitosamente por el usuario destino, con respecto al número total de mensajes cortos enviados por el usuario origen dentro de la misma red del operador, en un período de un mes.

Tabla 1.9
Porcentaje de Mensajes Cortos con Éxito

Parámetro de Calidad	Valor Objetivo	Frecuencia de Medición	Tamaño de la Muestra
Porcentaje de mensajes cortos con éxito	>= 95 %	Trimestral	25 SMSs/ km ² .

Fuente: Adaptado de: *Anexo 5*, Contratos de Concesión proporcionados por la SUPERTEL. Enero 2012

¹⁵ Esta especificación técnica se define en el documento de la ETSI GSM 03.40

Las pruebas de calidad de SMSs no deberán efectuarse en días atípicos. Los terminales de destino serán terminales de prueba de la Superintendencia de Telecomunicaciones. Las pruebas de SMSs deben efectuarse en escenarios totalmente controlados, de tal forma que se desechen todas las posibilidades de que el SMS no llegue a su destinatario final por causas inherentes al usuario; es decir, se garantiza:

- Que los dos terminales se encuentran encendidos.
- Dentro del área de cobertura de la operadora.
- Que el terminal llamado se encuentra desocupado.
- Que los dos terminales se encuentran activados.
- Está habilitado para enviar y recibir SMSs.

Se pueden dar los siguientes eventos en las pruebas que se realicen:

- El sistema indica que el SMS se ha enviado, y el destinatario recibe el mensaje.
- El sistema indica que el SMS no se ha enviado, pero el destinatario sí recibe el mensaje.
- El sistema indica que el SMS se ha enviado, pero el destinatario no recibe el mensaje.
- El sistema indica que el SMS no se ha enviado, y el destinatario no recibe el mensaje.

1.4.5.7 Tiempo Promedio de Entrega de Mensajes Cortos

Es el tiempo promedio, medido en segundos, transcurrido entre el envío de un mensaje corto por parte del usuario origen y la recepción del mensaje por parte del usuario destino, en la misma red del operador. Se detalla en la Tabla 1.10.

Tabla 1.10
Parámetro Tiempo Promedio de Entrega de Mensajes Cortos

Parámetro de Calidad	Valor Objetivo	Frecuencia	Tamaño de
	Trimestral	de Medición	la Muestra
Tiempo promedio de entrega de mensajes cortos	<= 30 segundos	Trimestral	25 SMSs/ km ²

Fuente: Adaptado de: *Anexo 5*, Contratos de Concesión proporcionados por la SUPERTEL. Enero 2012

1.4.5.8 Calidad de Conversación (MoS)

Es una medida de la calidad extremo a extremo de la conversación de una llamada de servicio de voz dentro de la red de la sociedad concesionaria.

Tabla 1.11
Parámetro MOS

Parámetro de Calidad	Valor Objetivo	Frecuencia de Medición	Tamaño de la Muestra
Calidad de conversación	MOS >= 3	Semestral	Número mínimo de llamadas de prueba 25 llamadas/ km ²

Fuente: Adaptado de: *Anexo 5*, Contratos de Concesión proporcionados por la SUPERTEL. Enero 2012

➤ Método de medición

La medición del parámetro es de móvil (en movimiento) a móvil (en sitio fijo). El terminal en movimiento genera los patrones de voz que son enviados al terminal en sitio fijo. Un equipo móvil deberá estar ubicado en las instalaciones de cada una de las Unidades Administrativas Regionales o en un sitio adecuado para garantizar una buena recepción (sitio fijo). Además se debe considerar que este sitio debe cumplir con los niveles de señal establecidos en el parámetro de cobertura y debe estar libre de fuentes de ruido externo.

En el sitio fijo, se hace la comparación con el algoritmo correspondiente y encuentra el valor del MOS del punto en movimiento que generó el patrón de voz. El resultado de la medición es el valor del MOS del punto en movimiento con las

coordinadas de ubicación de éste punto. El evaluador deberá calificar la calidad de conversación de cada llamada de acuerdo a la escala de la Tabla 1.12.

Tabla 1.12
Escala de Valores que Determina la Calidad de Conversación

Grado	Calificación
Excelente	5
Muy bueno	4
Bueno	3
Regular	2
Malo	1

Fuente: Adaptado de: *Anexo 5*, Contratos de Concesión proporcionados por la SUPERTEL. Enero 2012

Para el tamaño de la muestra también se debe tomar en cuenta los valores de la Tabla 1.13.

Tabla 1.13
Tamaño Muestras para Mediciones MOS, en Función de la Población

Población por Ciudad	Error	Número Mínimo de Llamadas de Prueba
Mayor a 400.000	3%	1.067
50.000 a 400.000	4%	600
20.000 a 50.000	5%	384

Fuente: Adaptado de: *Anexo 5*, Contratos de Concesión proporcionados por la SUPERTEL. Enero 2012

En base a las facultades que le otorga la Constitución y la Ley a la Superintendencia de Telecomunicaciones, se ha determinado la necesidad de realizar mediciones de los mismos 7 parámetros de calidad establecidos en los Contratos de Concesión de las operadoras CONECEL S.A. y OTECEL S.A., que tienen relación con el desempeño de las redes instaladas, pero con una metodología de medición diferente, esto es utilizando equipos de *drive test* que permiten realizar la medición en forma no intrusiva, los cuales permiten determinar la calidad del servicio que percibe el usuario y en el mismo momento que se presentan los problemas de calidad del servicio, a fin de que se realicen los correctivos que sean necesarios para solucionarlos.

1.4.6 PROCESO PARA EL CONTROL DE LA CALIDAD EN EL SMA

El objetivo principal de realizar el control de calidad del servicio móvil avanzado, es detectar problemas de calidad del servicio que prestan las operadoras CONECEL S.A., OTECEL S.A. y TELECSA S.A.

El realizar las pruebas en campo permite obtener información que determina si los parámetros de calidad que constan en los Contratos de Concesión, reflejan la calidad que percibe el usuario; y realizar propuestas de modificación a los parámetros de calidad del servicio, definidos en los contratos de concesión de las tres operadoras.

Además, con el proceso de control de calidad se busca notificar a las operadoras los problemas de calidad de servicio detectados, a fin de que se presente una propuesta de solución, verificar las soluciones planteadas y de no existir una solución al problema, iniciar procesos de juzgamiento.

El proceso consta de 5 fases, como se detalla en la Figura 1.27, proceso diseñado por las administraciones técnicas de la SUPERTEL.

1.4.6.1 Fase 1, Mediciones de Calidad del Servicio

Como ya se mencionó anteriormente, la SUPERTEL realiza *drive test* para obtener las mediciones en campo de los parámetros de QoS.

Típicamente un sistema de *drive test* está compuesto por una plataforma para pruebas en campo, terminales móviles, GPS, y un computador. Una plataforma para pruebas de tipo *drive test*, es un equipo que permite tomar muestras de voz y datos en redes 2G, 2.5G & 3G realizando mediciones de calidad de servicio de voz como caída de llamadas y zonas con niveles de señal deficientes, así como *throughput* para sesiones de datos. Soporta múltiples tecnologías como AMPS, GSM/GPRS/EDGE, CDMA/CDMA2000 1X/1xEV-DO y UMTS WCDMA. Además soporta conexión con terminales móviles, e incluye software para post procesamiento y generación de reportes.

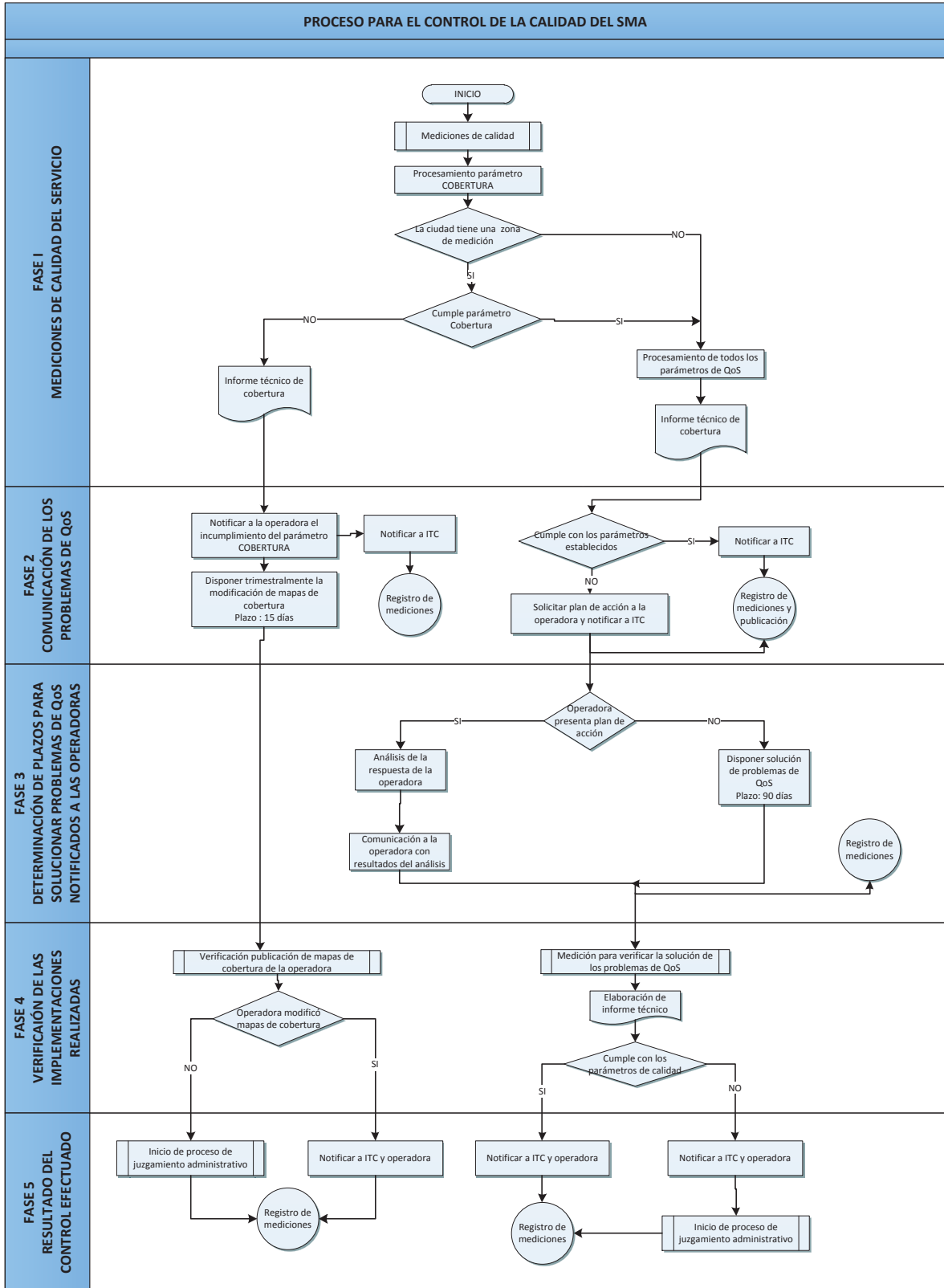


Figura 1.27 Diagrama del Proceso para el Control de la Calidad del SMA

Fuente: *Proceso para el control de la calidad del SMA.*, Contratos de Concesión proporcionados por la SUPERTEL. Enero 2012

La etapa de post procesamiento es la responsable de elaborar tablas, hojas de cálculo, y bases de datos, donde la información recopilada, ya filtrada y procesada; se presenta en un formato de fácil apreciación y comprensión.

Dentro de este esquema, actualmente la SUPERTEL cuenta con los equipos, detallados en la Tabla 1.14 , para realizar *drive test*.

Tabla 1.14
Equipos de la SUPERTEL para Mediciones de Campo

Plataforma para Pruebas	Terminales Móviles
INVEX 3G	LG CU500v
TEMS <i>investigation</i>	Nokia N95 Motorola V4000/V5000

Fuente: *Proceso para el control de la calidad del SMA.*, Contratos de Concesión proporcionados por la SUPERTEL. Enero 2012.

Existen dos métodos para realizar las mediciones de los parámetros durante el proceso de *drive test*.

- **Mediante escáner RF:** El cual, a través de la toma de muestras de niveles de señal permite obtener datos de cobertura y propagación de señales generadas por las RBS.
- **Mediante terminal móvil:** Permite realizar mediciones de dos maneras, la primera en modo *idle*, para determinar niveles de señal (y por ende cobertura). Y en modo *ONNET* o llamada, para determinar llamadas establecidas, llamadas caídas, tiempo de entrega de mensajes, etc.

El parámetro con el que se debe iniciar el proceso de medición, es el parámetro cobertura. Para este parámetro se realizará la medición utilizando el teléfono en modo *idle*, sin antena externa tanto para GSM, CDMA y UMTS. La medición se realizará tomando datos de los teléfonos. Se medirá el parámetro *Rx Level* para tecnología GSM y *Ec/Io* para CDMA/UMTS sobre el canal de control. Se debe realizar la medición en ciudades o localidades donde el operador informe que tiene cobertura.

La SUPERTEL tiene la facultad de disponer los ajustes a la información de cobertura publicada por las operadoras, por lo que se requiere que se verifique si se cumple con el valor mínimo establecido para el parámetro cobertura; al respecto se debe considerar lo siguiente:

- 1.- En caso que se cumpla con el parámetro cobertura y la operadora reporte que tiene cobertura, se debe realizar el procesamiento de los demás parámetros de calidad.
- 2.- Si la operadora no publica que tiene cobertura en una determinada localidad, aunque la administración regional ya haya realizado la medición de calidad del servicio, no se requiere evaluar el cumplimiento de los demás parámetros de calidad, por lo que no es necesario procesar dicha información.
- 3.- Si se verifica que en una determinada población no se cumple con el valor mínimo establecido para el parámetro cobertura, no es necesario realizar el procesamiento de las mediciones de los demás parámetros de calidad del servicio.
- 4.- En caso que en una o varias zonas no se cumpla con el parámetro cobertura y la operadora reporte que tiene cobertura, se debe notificar a la operadora a fin de que presente un plan de acción para mejorar la cobertura. En este caso, siempre se debe realizar el procesamiento de los demás parámetros de calidad.

Si no se soluciona el problema de cobertura, la disposición de modificación de los mapas de cobertura se lo realizará conforme a lo que se acuerde entre las operadoras y la SUPERTEL.

1.4.6.2 Fase 2, Comunicación de los Problemas de Calidad del Servicio

En primera instancia y de ser el caso se notifica a la operadora acerca del incumplimiento del parámetro cobertura, señalando que de acuerdo al contrato debe retirar de la página web que tiene cobertura en esa zona o localidad.

Se debe disponer a las operadoras la modificación de los mapas de cobertura según los plazos descritos en la Tabla 1.15:

Tabla 1.15
Calendario para Modificación de Mapas Publicados por las Operadoras

Trimestre	Plazo para Emitir Oficio a la Operadora	Plazo para que las Operadoras Modifiquen los Mapas de Cobertura
1ro	30 de marzo	15 de abril
2do	30 de junio	15 de julio
3ro	30 de septiembre	15 de octubre
4to	30 de diciembre	15 de enero

Fuente: *Proceso para el control de la calidad del SMA.*, Contratos de Concesión proporcionados por la SUPERTEL. Enero 2012.

Si existe incumplimiento en algún otro parámetro también se debe notificar a la operadora.

1.4.6.3 Fase 3, Determinación de Plazos para Solucionar los Problemas de QoS

La operadora luego de realizar las correcciones pertinentes, solicita realizar nuevas mediciones; para lo cual adjuntan el mapa de cobertura con las mediciones realizadas por la operadora en el que se observa que no tendrían problema de cobertura.

En el caso que la operadora admita que no cumple con el parámetro cobertura, cualquiera sea la causa, se debe proceder a modificar el mapa de cobertura o si la operadora indica que en un determinado plazo solucionará el problema. Se debe disponer una corrección en los mapas de cobertura publicados por la operadora.

También puede darse el caso que la operadora adjunta documentación con la que se verifica que no puede implementar nuevas instalaciones, para corregir el problema, debido a desacuerdos con el municipio o con la comunidad. De igual manera se debe corregir el mapa de cobertura, pero también se debe difundir y socializar con la comunidad la razón por la cual no se pudo solucionar el problema.

1.4.6.4 Fase 4 y 5, Verificación de la Solución a los Problemas de QoS

Si la operadora soluciona el o los inconvenientes detectados, se debe registrar la nueva medición, publicar los resultados obtenidos y notificar tanto a la operadora como a SUPERTEL, de no haber dado una solución, también se debe registrar la nueva medición y publicar el resultado, pero a diferencia del primer caso, la operadora está en la obligación de modificar sus mapas publicados; y si la operadora no acata la disposición de la SUPERTEL de modificar la cobertura publicada por la operadora cada administración regional deberá iniciar el proceso de juzgamiento administrativo por incumplimiento de contrato.

2. ANÁLISIS DE REQUERIMIENTOS EN SOFTWARE Y HARDWARE

2.1 INTRODUCCIÓN

En el presente capítulo se detallará como están almacenados los diferentes datos (sectores, parámetros de QoS, localización de Radio bases, etc.) que la SUPERTEL maneja con el objetivo de mantener un monitoreo de la calidad de servicio que las operadoras celulares ofrecen a los clientes en general. Así como la forma en que los usuarios pueden expresar sus dudas, inquietudes o reclamos referentes al servicio de telefonía móvil.

La información proporcionada por la SUPERTEL, ayudará a esquematizar los posibles caminos a seguir para el diseño de la aplicación web y la base de datos; así como las herramientas más adecuadas para su desarrollo. Finalmente se realizará el dimensionamiento del servidor/servidores, en los cuales la aplicación web y la base de datos se albergarán.

2.2 SITUACIÓN ACTUAL

2.2.1 RECOPIACIÓN DE DATOS

Toda red que brinda uno o varios servicios a los usuarios, luego de su implementación debe ser evaluada con el objetivo de que el servicio proporcionado cumpla con los parámetros de calidad que exija el usuario. Actualmente, la SUPERTEL realiza *drive-test*¹⁶ periódicamente (4 veces por año), en las diferentes ciudades beneficiarias del SMA. El primer paso a seguir en la realización del *drive test*, es definir el espacio geográfico donde se lo realizará: la ciudad, el sector o zona y finalmente la ruta a seguir. Los datos proporcionados por los equipos pasan a una fase de depuración y finalmente son almacenados en el sistema GTM¹⁷.

¹⁶ Es un proceso para recopilar información.

¹⁷ Sistema de Registro de Mediciones de Telefonía Móvil SUPERTEL.

2.2.2 SECTORIZACIÓN DE LA CIUDAD DE QUITO

Para el caso de la ciudad de Quito, se tienen 27 sectores definidos, cada sector tiene un área geográfica de aproximadamente 4Km². Las coordenadas (longitud y latitud) que definen un sector están almacenadas en un *shapefile* o archivo *shape*. Un *shapefile* almacena entidades geométricas con sus atributos. Es un formato de tipo propietario desarrollado por la *Environmental Systems Research Institute* (ESRI).

La función de almacenamiento de información de un *shapefile* soporta entidades geométricas de tipo punto, línea, y polígono. Las geometrías se almacenan como conjuntos de coordenadas vectoriales.

Un *shapefile* está compuesto por tres archivos fundamentales:

1. *Main file* o archivo principal: su extensión es “.shp”, contiene una cabecera con metadatos de las geometrías almacenadas, y un conjunto de registros donde se almacenan las geometrías.
2. *Index file* o archivo de índice: su extensión es “.shx”, dentro de este archivo se encuentra el índice de las geometrías contenidas en el archivo principal. Su función es la de optimizar en tiempo y uso de recursos la búsquedas de los registros almacenados dentro del *main file*.
3. *dBase*, su extensión es “.dbf”, dentro de este archivo se puede almacenar cualquier tipo de información adicional necesaria (medidas de distancias, áreas, etc.) y que tiene relación con cada uno de los objetos espaciales contenidos en el *main file*. A manera de ejemplo, dentro del *main file* se pueden almacenar todas las provincias del Ecuador como objetos espaciales de tipo polígono y dentro del archivo *dBase* se puede tener información de población, área, etc., de cada una de las provincias.

Es importante destacar que este archivo no contiene las coordenadas vectoriales que describen el/los objetos espaciales. Adicionalmente, también se tiene un archivo con extensión “.prj”, que es el encargado de almacenar el sistema de referencia utilizado por el archivo *shapefile*. El archivo “.prj” no es indispensable

ya que dicha información también está contenida en la cabecera del archivo *main file*. Para realizar cualquier operación (visualizar, editar, etc.) sobre un *shapefile*, los tres archivos descritos anteriormente deben estar dentro del mismo directorio, adicionalmente su nombre debe ser el mismo, se diferencian solo en la extensión del archivo.

Nombre	Fecha de modifica...	Tipo	Tamaño
zonas.dbf	22/08/2012 1:47	Archivo DBF	1 KB
zonas.shp	22/08/2012 1:45	Archivo SHP	12 KB
zonas.shx	22/08/2012 1:46	Archivo SHX	1 KB

Figura 2.1 Archivos que Conforman un Shapefile

Fuente: Esquema de Autoría Propia (Arévalo, D. y Guzmán, D.).

Este archivo permite visualizar los 27 sectores que conforman la ciudad de Quito zonificada, se requiere de un paquete adicional, Sistema de Información Geográfica (SIG) para realizar esta tarea.

2.2.3 SISTEMA DE INFORMACIÓN GEOGRÁFICA (SIG)

En la actualidad existen varias opciones para aplicaciones SIG de escritorio tanto de tipo comercial (*Arcgis*, *Mapinfo*, etc.) como *open source* (*Gvsig*, *Quantum GIS*, etc.). Debido a que en el presente proyecto se prioriza el uso de herramientas libres, en la Tabla 2.1, se realizará un análisis comparativo de las dos principales opciones de tipo *open source* (*Gvsig* y *Quantum GIS*), comparando las características más relevantes y las que brinden facilidades para la realización del proyecto.

Las funciones de análisis vectorial son fundamentales para el desarrollo del presente proyecto, específicamente serán utilizadas en el análisis de la solución para la sectorización de la ciudad de Quito. Por tal motivo se utilizará el paquete SIG *QUANTUM GIS*, este software está publicado bajo Licencia Pública GPL.

Para visualizar archivos *shape*, dentro de la interfaz gráfica de Quantum GIS, se debe seleccionar la opción añadir una capa vectorial, como se observa en la Figura 2.2.

Se selecciona el archivo *zonas.shp*¹⁸ que contiene los datos geográficos de las zonas de Quito y a continuación se podrán visualizar los datos, Figura 2.3.

Tabla 2.1
Análisis Comparativo de SIGs

	gvSIG	QuantumGIS - QGIS
Sitio web	http://www.gvsig.com/	http://www.qgis.org/
Basada en estándares OGC	SI	SI
Interfaz gráfica de usuario	SI	SI
Compatibilidad S.O.	Windows, Linux.	Windows, Linux.
Bases de datos espaciales soportadas	Oracle Spatial, Postgis, HyperSQL Database Engine, MYSQL	Oracle Spatial, Postgis, MYSQL, SpatialLite
Soporte OGC - WMS	SI	SI
Soporte OGC - WFS	SI	SI
Soporte OGC - WCS	NO	SI
Soporte <i>Shapefiles</i>	SI	SI
Publicación de información geoespacial y metadatos a través de servicios web estándares OGC	Si, Mapserver. (solo en versión 1.1)	Si, Mapserver
Funciones para análisis vectorial	No posee módulo específico	45 funciones
Funciones para operaciones vectoriales fundamentales	SI	SI

Fuente: Tabla de Autoría Propia (Arévalo, D. y Guzmán, D.).

¹⁸ Archivo proporcionado por la SUPERTEL. Enero 2012

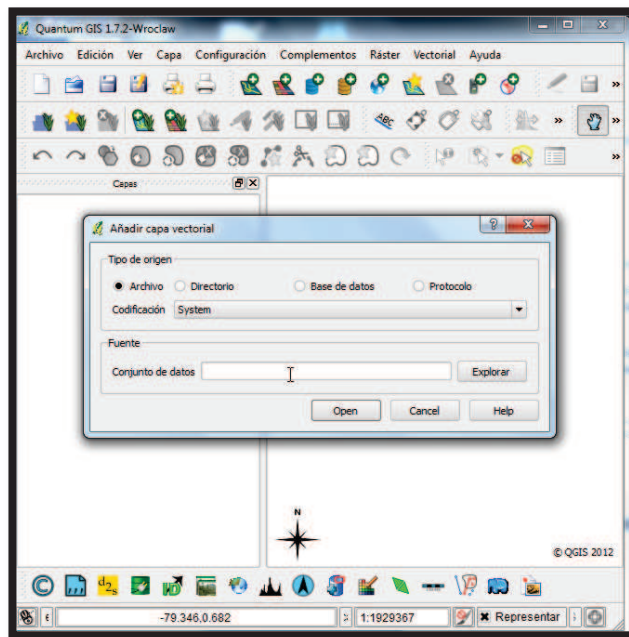


Figura 2.2 Añadiendo una Capa Vectorial en Quantum GIS
Fuente: Esquema de Autoría propia (Arévalo, D. y Guzmán, D.).

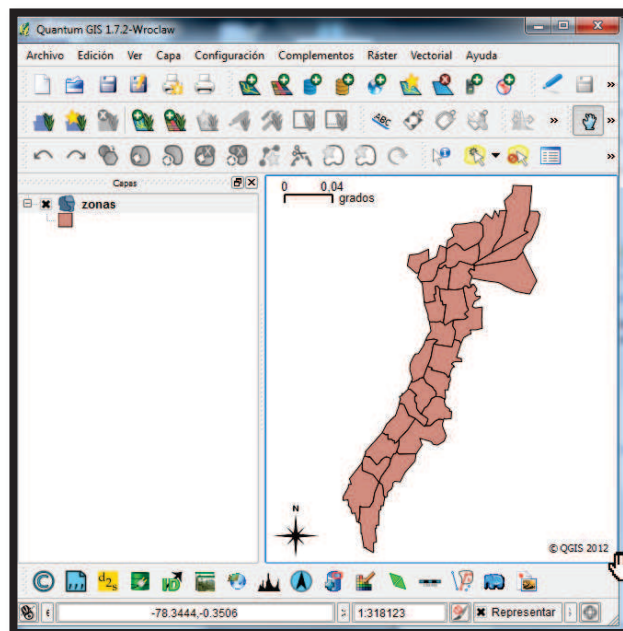


Figura 2.3 Zonas de la Ciudad de Quito - Quantum GIS
Fuente: Esquema de Autoría Propia (Arévalo, D. y Guzmán, D.).

2.3 SISTEMA DE REGISTRO DE MEDICIONES TELEFONÍA MÓVIL (GTM)

Este sistema almacena los datos depurados de las mediciones de parámetros de QoS para luego presentar reportes (a manera de vistas) donde se pueda observar un registro detallado de los parámetros de QoS obtenidos. Dentro de cada vista también consta información relacionada a las 5 fases que contempla el proceso de medición de parámetros, análisis de resultados, solución a implementarse (de ser necesario), nuevo proceso de medición (luego de implantarse la solución) y finalmente un reporte de si la solución arrojó resultados satisfactorios o no. La base de datos que almacena toda esta información utiliza un motor de base de datos Oracle versión 10g¹⁹.

2.3.1 USUARIOS

Para el desarrollo del presente proyecto, la SUPERTEL puso a disposición un usuario de tipo solo-lectura, que tenga acceso garantizado a la base de datos del GTM. Las credenciales para este usuario son las detalladas en la Tabla 2.2.

Tabla 2.2
Credenciales para Acceder a la Base de Datos del GTM

Usuario	Password	Permisos	Esquema
<i>consulta</i>	<i>consulta</i>	<i>Solo lectura</i>	<i>GTM</i>

Fuente: Tabla de Autoría Propia (Arévalo, D. y Guzmán, D.).

2.3.2 MANEJO DE RECLAMOS E INFORMACIÓN

La Superintendencia de Telecomunicaciones (SUPERTEL) entrega información y recepta reclamos por la calidad de los servicios telecomunicaciones, radiocomunicaciones, radiodifusión y televisión, entre otros, a través del Centro de Información y Reclamos (CIR). El usuario puede realizar sus quejas, reclamos o dudas de varias maneras como:

- Vía telefónica y de forma gratuita a través del número *1800 567567*

¹⁹ Información proporcionada Dirección de Tecnologías de la Información (DTI) – SUPERTEL. Enero 2012

- Llenando el formulario de reclamos e información que está disponible en la siguiente url :
http://www.supertel.gob.ec/index.php/component/option,com_facileforms/Itemid,317/



The image shows a web form titled 'Información Personal' with the following fields: Fecha (with a calendar icon), Nombres, Apellidos, Cédula de ciudadanía, Dirección, Ciudad, Provincia, Teléfono, Fax, E-mail, Servicio, and Operadora. Below the personal information section is a 'Mensaje' section with a text area and an 'Enviar' button.

Figura 2.4 Formulario para Reclamos o Solicitud de Información - CIR

Fuente: SUPERTEL. Versión Electrónica. Disponible en:

http://www.supertel.gob.ec/index.php/component/option,com_facileforms/Itemid,317/ (Diciembre 2012).

- El usuario también puede enviar un correo electrónico a cualquiera de estas direcciones: cir@supertel.gob.ec o info@supertel.gob.ec y manifestar alguna inquietud y queja.
- Finalmente, el usuario tiene la posibilidad de reportar una queja o duda personalmente en las oficinas del CIR ubicadas en: Av. Amazonas N4071 y Gaspar de Villarroel – Quito.

2.4 ANÁLISIS DE REQUERIMIENTOS

En la actualidad, el uso de la información geográfica se encuentra muy difundida y existe variedad de aplicaciones que hacen uso de la misma, con esta premisa surge la necesidad de estandarizar y definir normas para el uso y representación de la información de tipo geográfica.

Norma es un texto en el cual se acuerdan aspectos técnicos de un componente o servicio, y este es elaborado por un organismo de normalización por ejemplo la ISO. Un estándar es un documento no elaborado por un organismo de estandarización, pero se lo acepta y usa en consenso, por ejemplo los documentos que emite el *Open GeoSpatial Consortium*, OGC. El OGC, es un consorcio de más de 300 organizaciones sin fines de lucro cuyo objetivo es definir especificaciones de interoperabilidad para los SIG, por consenso.

La interoperabilidad se la puede definir como la capacidad que deben tener, dos o más sistemas, para intercambiar información de forma eficiente; además la información debe poder ser usada y entendida por cualquiera de los sistemas involucrados, como se ejemplifica en la Figura 2.5. Varios sistemas heterogéneos proveen información (Oracle, PostgrsSql, etc.), pero el servicio que consume estos datos no se preocupa por sus diferentes orígenes, el sistema en conjunto es interoperable.

En los SIG, la interoperabilidad es fundamental ya que se pueden presentar casos en los cuales la información geográfica provenga de diferentes organizaciones y maneje diferentes formatos; por lo tanto, deba ser procesada y unificada para poder ser difundida a una comunidad. Para facilitar la interoperabilidad de los SIG, surge el conjunto de normas ISO 19100, desarrollado por el comité técnico ISO/TC211.

2.4.1 DATOS ESPACIALES

Si bien los archivos *shapefile* son livianos y de fácil uso, no representan una manera óptima de almacenar información debido a que son archivos independientes, no pueden ser relacionados. Una manera más ordenada y eficiente de almacenar información de tipo geográfica son las bases de datos

espaciales. En la Tabla 2.3 se realiza un análisis comparativo de las ventajas que ofrecen las bases de datos espaciales frente a los archivos *shapefile*.

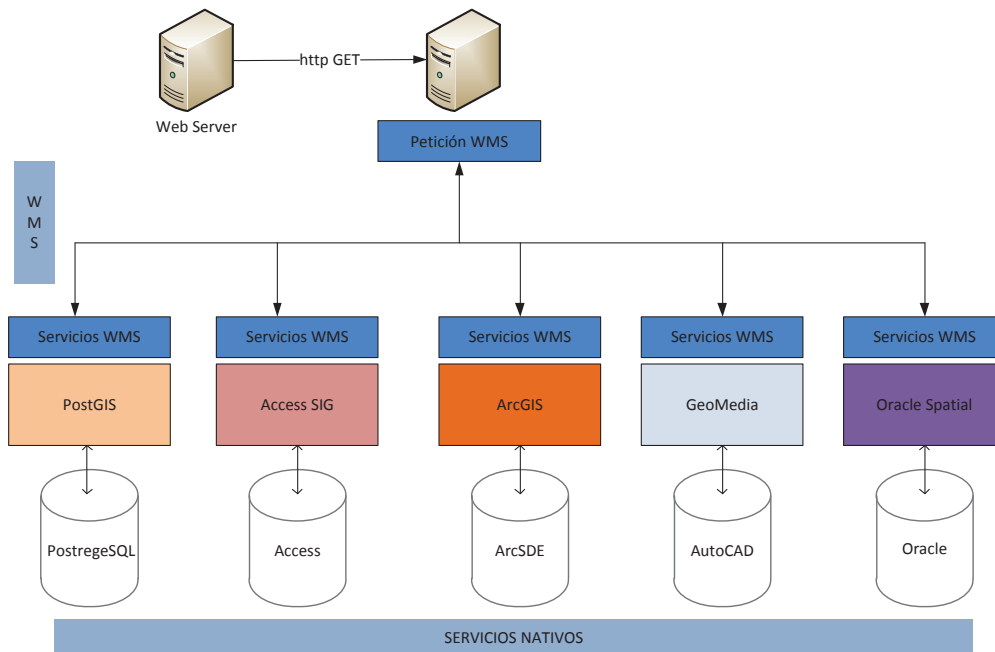


Figura 2.5 Principio de Interoperabilidad

Fuente: Adaptado de <http://www.prodevelop.es/es/tecs/geo/ide>. (Diciembre 2011).

A parte de las características mostradas en la Tabla 2.3, las bases de datos en general brindan otras funcionalidades importantes como seguridad (mediante usuarios y perfiles), respaldo de la información (*backups*), herramientas para minería de datos²⁰, generación de reportes; mismas características, que un archivo *shapefile* por sí solo no provee (requiere de software adicional).

A pesar de todas las ventajas que presentan las bases de datos espaciales frente a los archivos *shapefile*, se debe analizar la información a almacenar ya que si no se requiere mayor procesamiento de la información, los datos espaciales no se relacionan con otros datos y no existen gran volumen de información (no superior a 2GB²¹), se puede optar por el uso de *shapefiles*, debido a que resulta más económico y más sencillo de administrar.

²⁰ Se refiere a procesos a través de los cuales se analizan grandes volúmenes de información y se determina patrones, comportamientos, tendencias, etc.

²¹ Información proporcionada por la ESRI.

Actualmente la SUPERTEL posee la información de la sectorización de la ciudad de Quito en un archivo que contiene datos vectoriales - *zonas.shp*; no existe una base de datos donde se almacene esta información. Para este caso específico si se requiere migrar todos los datos contenidos en dicho archivo hacia un motor de base datos que soporte datos de tipo espacial; ya que la información, de las entidades geométricas, debe relacionarse con el registro de mediciones GTM que posee la SUPERTEL, para poder asociar determinados valores de mediciones para una zona específica. Esta relación permitirá presentar información útil y entendible al usuario final.

Tabla 2.3
Análisis Comparativo Shapefile - Bases de Datos Espacial

	<i>Shapefile</i>	BDD Espacial
Forma de almacenamiento	Conjunto de archivos	Sistema de base de datos relacional (RDBMS)
Base de Datos	Archivos <i>dBase</i>	Sistema de base de datos relacional (RDBMS)
Tipos de entidades soportadas	Puntos, Líneas, Polígonos	Puntos, Líneas, Polígonos, Funciones espaciales, Imágenes, representaciones cartográficas, etc.
Topología²²	-	Orientada a características
Integridad Entidad²³	NO	SI
Integridad Dominio²⁴	NO	SI
Capacidad máxima de almacenamiento	2 GB (por archivo)	Dependiente de las características de hardware, donde se aloje la BDD

Fuente: Adaptado de: *Oracle Spatial Developer's Guide*. Recuperado de: http://docs.oracle.com/cd/B28359_01/appdev.111/b28400/sdo_intro.htm. (Marzo 2012). & *ESRI Shapefile Technical Description*. Recuperado de: <http://www.esri.com/library/whitepapers/pdfs/shapefile.pdf>. (Marzo 2012).

²² La topología define cómo las entidades de puntos, líneas y polígonos comparten una geometría coincidente.

²³ Garantiza entidades únicas mediante el uso de claves primarias.

²⁴ Garantiza coherencia en los datos almacenados (tipos de datos definidos, claves foráneas, campos no nulos, etc.).

2.4.2 COLUMNAS ESPACIALES

Observando el mapa de una ciudad, se pueden distinguir: puntos, líneas o polígonos que definen la localización de carreteras (líneas), estaciones de combustible (puntos), divisiones regionales o políticas (polígonos), etc. Partiendo de este antecedente, un mapa se lo puede definir como una representación visual de una porción de territorio, proyectada sobre una imagen en dos dimensiones; que mantiene relación de posición y tamaño real.

Los datos que definen la posición de este territorio se los conoce como datos espaciales. Una base de datos maneja columnas de tipo texto, numérico, booleano, etc., cuando una base de datos brinda soporte para datos espaciales debe manejar columnas de tipo espaciales.

2.4.2.1 Definición

El OGC en su estándar *OpenGIS Implementation Specification for Geographic information - Simple feature access - Part 2:SQL option*²⁵ versión 1.2.1 publicado el 2010-08-04 define la siguiente jerarquía para los datos espaciales Figura 2.6.

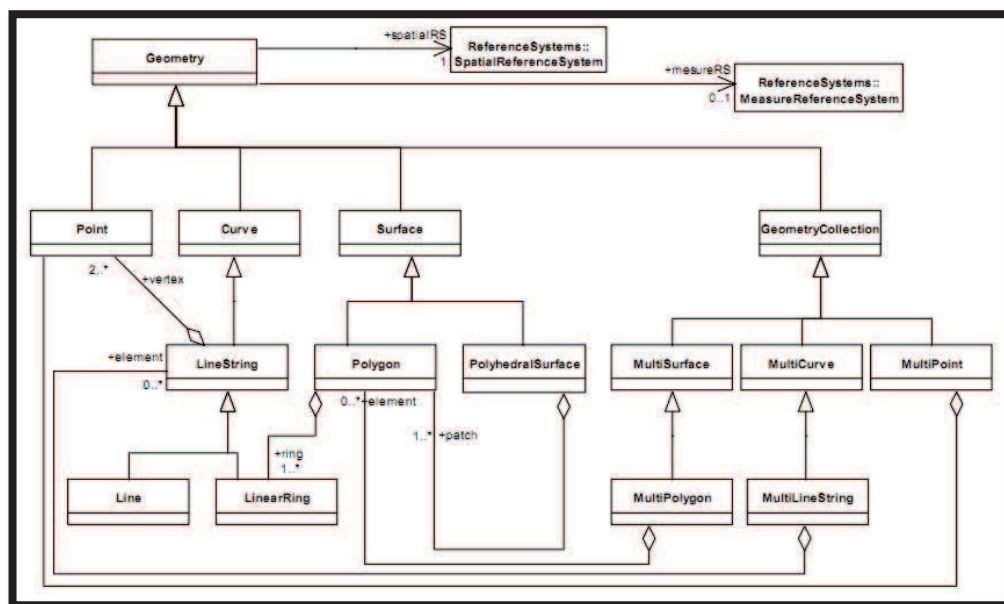


Figura 2.6 Jerarquía de Datos Espaciales OGC

Fuente: *OGC Simple Feature Access - Part 1: Common Architecture*. Versión Electrónica. Recuperado de: http://portal.opengeospatial.org/files/?artifact_id=25355%E2%80%8E. (Marzo 2012).

²⁵ Estándar OGC de tipo abierto disponible en: <http://www.opengeospatial.org/standards/sfs>. Marzo 2012.

En resumen una base de datos espacial debe brindar soporte para columnas de tipo espacial como se describe en la Tabla 2.4.

Tabla 2.4
Tipos de Columnas Espaciales

Columna Espacial	Descripción
Geometry	Súper clase de tipo abstracta.
Point	Una coordenada (x,y).
Line	Un par coordenadas (x,y).
Polygon	Un conjunto de coordenadas (x,y), primera y última coordenada iguales.
Geometrycollection	Un conjunto de geometrías distintas.
Multipoint	Un conjunto de puntos.
Multiline	un conjunto de líneas
Multipolygon	Un conjunto de polígonos

Fuente: Adaptado de: *OGC Simple Feature Access - Part 1: Common Architecture*. Versión Electrónica. Recuperado de: http://portal.opengeospatial.org/files/?artifact_id=25355%E2%80%8E. (Marzo 2012).

2.4.3 SISTEMA DE REFERENCIA ESPACIAL

Para poder ubicar con precisión un punto en la tierra debe conocerse la forma general de la tierra, estos modelos de forma de la tierra son llamados *datums*. El *datum* es un modelo de referencia; parte del análisis de la forma aproximada de la tierra, es decir la superficie teórica de la tierra. El geode es una figura en la cual todos los puntos que la conforman son perpendiculares al campo gravitacional.

Y la forma teórica de la tierra, representada por un elipsoide (definido por un radio mayor y uno menor). El inverso de la diferencia entre radio mayor y radio menor se conoce como la inversa de aplanamiento.[35]

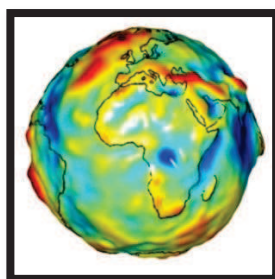


Figura 2.7 Geoide de la Tierra

Fuente: *Gravity Anomaly Maps and The Geoid*. Recuperado de: <http://earthobservatory.nasa.gov/Features/GRACE/page3.php>. (Diciembre 2012).

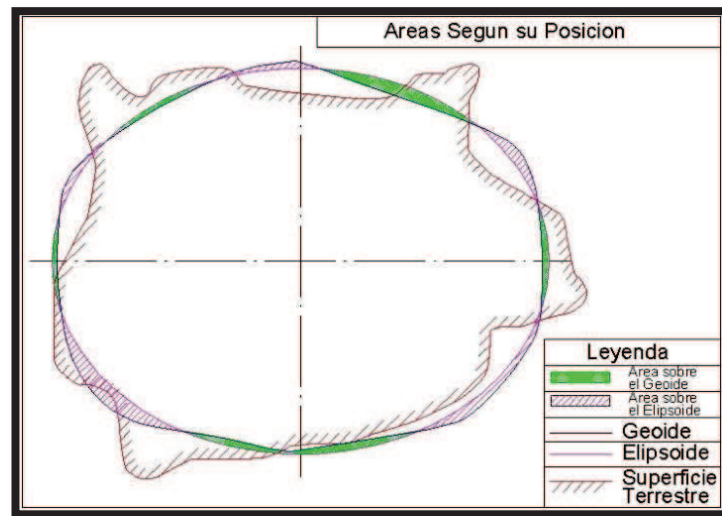


Figura 2.8 Intersección Teórica del Geoide y Elipsoide

Fuente: Fernández-Coppel, Ignacio. (2001). *Localizaciones Geográficas El Datum*. (pp. 8). Valladolid: Universidad de Valladolid. Recuperado de: <http://www.elgeomensor.cl/downloads/topografia%20y%20geodesia/index.php?file=datum.pdf>. (Enero 2012).

Los puntos donde intersecan el geoide y un elipsoide (a un determinado radio mayor y menor dado), se los conoce como puntos fundamentales. El *datum* es este punto fundamental. Un *datum* por lo tanto está compuesto de un punto fundamental y el elipsoide, que al intersecar con el geoide, lo genera. [35]

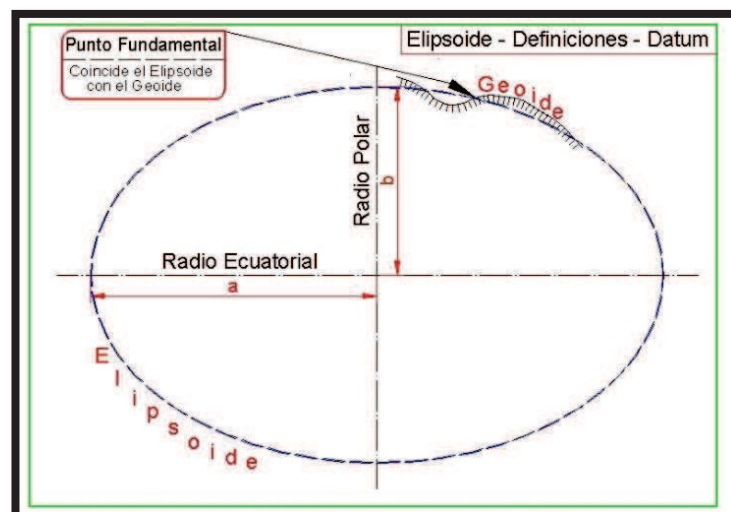


Figura 2.9 Representación Teórica de un Punto Fundamental

Fuente: Fernández-Coppel, Ignacio. (2001). *Localizaciones Geográficas El Datum*. (pp. 11). Valladolid: Universidad de Valladolid. Recuperado de: <http://www.elgeomensor.cl/downloads/topografia%20y%20geodesia/index.php?file=datum.pdf>. (Enero 2012)

Al trazar una tangente al elipsoide en el punto fundamental, y una perpendicular a esta tangente en el mismo punto. La perpendicular genera un ángulo ϕ con el plano ecuatorial. Finalmente, si se une este punto de intersección con el centro de la tierra, se genera otro ángulo λ (formado con el eje x de la tierra). Al quedar definidos este par de ángulos (ϕ , λ), también queda definido el sistema de referencia (Figura 2.10). [35]

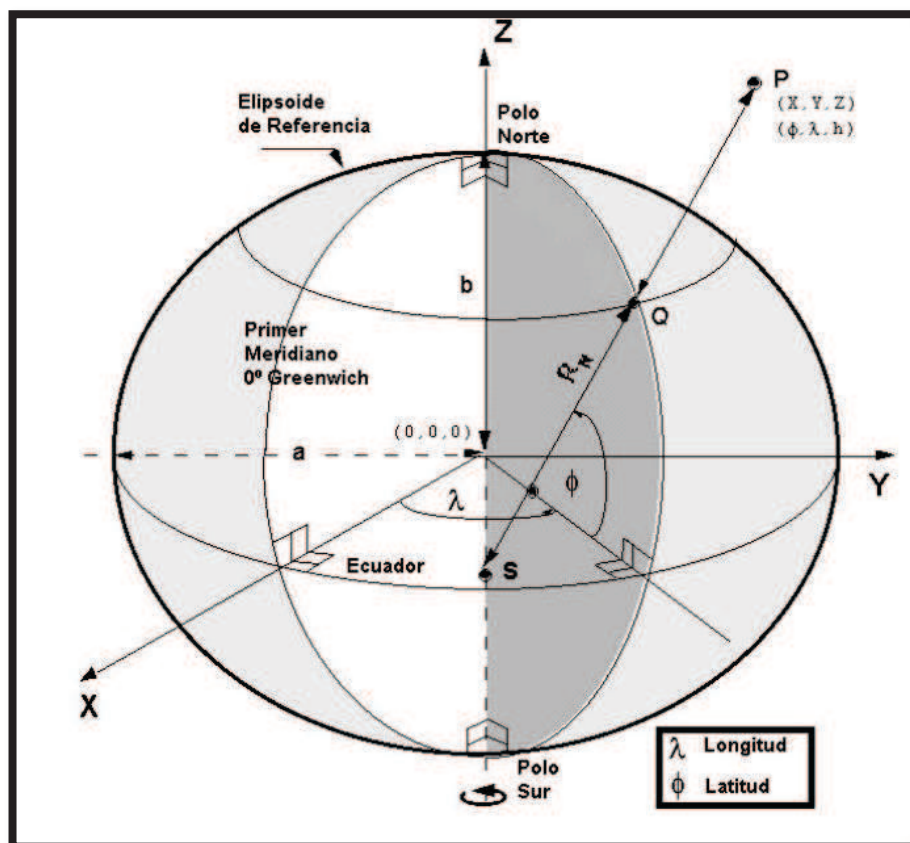


Figura 2.10 Representación Datum Teórico

Fuente: Fernández-Coppel, Ignacio. (2001). *Localizaciones Geográficas El Datum*. (pp. 13). Valladolid: Universidad de Valladolid. Recuperado de: <http://www.elgeomensor.cl/downloads/topografia%20y%20geodesia/index.php?file=datum.pdf>. (Enero 2012)

Debido a que el *datum* depende de un elipsoide específico, existen diferencias en la localización de un punto al usar *datums* distintos. Es decir un *datum* tiene su campo de aplicación específico. Sin embargo, dada la necesidad de un sistema de referencia de aplicación mundial; es decir, un *datum* que permita referenciar cualquier punto de la superficie terrestre, un *datum* universal, la *World Geodetic System* – Sistema Geodésico Mundial (WGS), en el año de 1984 y mediante el

uso de satélites define el elipsoide *World Geodetic System* (WGS84). Este elipsoide define un radio mayor de 6'378.137 m y radio menor de 6'356.752 m.²⁶

Cabe destacar que una de las características más relevantes de este *datum* (WGS84), es que puede considerarse de tipo geodésico o geocéntrico, ya que el punto origen coincide con el centro de gravedad de la tierra. En la Figura 2.11, se puede observar que el plano x (plano ecuatorial.), coincide con el plano meridional (meridiano de Greenwich) de la tierra, el plano Y.

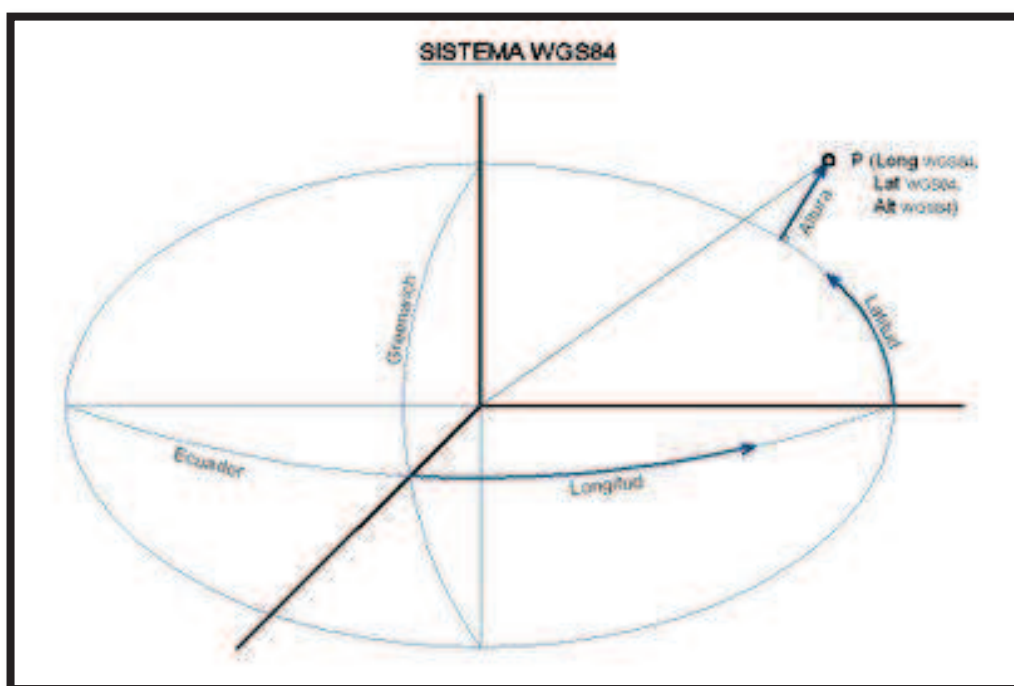


Figura 2.11 Sistema de Referencia WGS84

Fuente: *Sistemas Geodésicos de Referencia*. Recuperado de: <http://www.ign.es/ign/layoutIn/actividadesGeodesiaStmagd.do> (Enero 2012).

2.4.3.1 Identificador de Sistema de Referencia

Un objeto espacial debe estar asociado a un Identificador de Sistema de Referencia Espacial (SRID). El SRID determina el sistema de coordenadas usado para todos los objetos espaciales de una columna; además brinda un significado a los valores de las coordenadas que contiene un objeto espacial. El SRID consiste en un único número entero para un determinado sistema de referencia espacial

²⁶ Tomada de la base de datos proporcionada por la EPSG, disponible en <http://www.epsg.org/guides/docs/G7-1.zip>. Diciembre 2012

dentro de una base de datos, este número lo define la *European Petroleum Survey Group* (EPSG).

La EPSG tiene hasta la fecha Agosto 2012 definidos 4364 SRIDs diferentes. En la Tabla 2.5, se listan los más usados actualmente. De los cuales destaca 4326 ya que su *datum* es WGS84 que como se mencionó anteriormente tiene su campo de aplicación a nivel mundial.

Tabla 2.5
Sistemas de Referencia Espacial

Sistema de referencia espacial	SRID
OSGB 1936	27700
ETRS89	4258
TM75	29903
OSNI 1952	29901
IRENET 95	2157
WGS84	4326

Fuente: Adaptado de: *EPSG Geodetic Parameter Dataset*. Recuperado de: <http://www.epsg.org/geodetic.html> (Enero 2012).

Los datos proporcionados por la SUPERTEL tienen un SRID 4326, en la Tabla 2.6 se detallan parámetros del sistema de referencia espacial correspondiente.

2.4.3.2 Sistema de Referencia WGS84 - EPSG 4326

Es un sistema de referencia terrestre convencional, definido bajo las recomendaciones *International Earth Rotation Service-IERS*. Este sistema de referencia es de tipo geocéntrico (aunque también se lo considera geodésico) fijo con la tierra y orientado positivamente (derecha). En la Tabla 2.6 se detallan los parámetros del SRID 4326.

2.4.4 ÍNDICES ESPACIALES

Los índices permiten acceso rápido a los datos almacenados en una base de datos. Las estructuras de datos conocidas como árboles-b o *b-trees* mantienen los datos ordenados y permiten tener acceso de forma rápida. Los datos de tipo texto, números, fechas entre otros tipos, permiten usar *b-trees* ya que, los números, por ejemplo tienen un orden ya definido y los otros tipos permiten

realizar comparaciones como “mayor que” o “igual a”. Los datos de tipo espacial requieren un tratamiento diferente, no se pueden aplicar operaciones básicas “mayor que” o “menor que”, por lo que el indexado se realiza utilizando un cuadro o celda que delimite a una figura espacial conocido como *bounding-box*, Figura 2.12.

Tabla 2.6
Parámetros SRID 4326

Nombre	WGS84
Código	4326
Sistema de Coordenadas	Geodésica
<i>Datum</i>	WGS84
Elipsoide	WGS84
Radio Mayor	6378137
Inversa de aplanamiento ²⁷	298,257223563
Unidad de medida	Grados

Fuente: Adaptado de: OGC Simple Feature Access - Part 1: Common Architecture. Versión Electrónica. Recuperado de: http://portal.opengeospatial.org/files/?artifact_id=25355%E2%80%8E. (Marzo 2012).

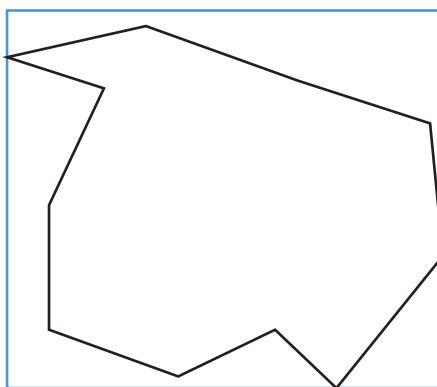


Figura 2.12 Bounding-box Conteniendo a una Figura

Fuente: Esquema de Autoría propia (Arévalo, D. y Guzmán, D.).

La idea de utilizar un *bounding-box* es poder realizar consultas de si un cuadro o celda está contenida en otra, con lo que se evita consultar si una figura espacial está dentro de otra (implica más tiempo de procesamiento). Actualmente se utilizan arboles-r o *r-trees* para el manejo de los índices espaciales, las celdas que

²⁷ Inverso del cociente entre la diferencia del semi eje mayor y menor de un elipsoide, sobre su semi eje mayor.

abarcan grandes áreas están en las jerarquías más altas. En la Figura 2.13, se observa que los objetos R1, R2 son los contenedores más grandes, por lo tanto son jerárquicamente el nivel más alto respecto de los demás objetos. La limitación de los índices espaciales es que no proveen resultados exactos, al tratarse de una comparación de áreas arrojan datos aproximados.

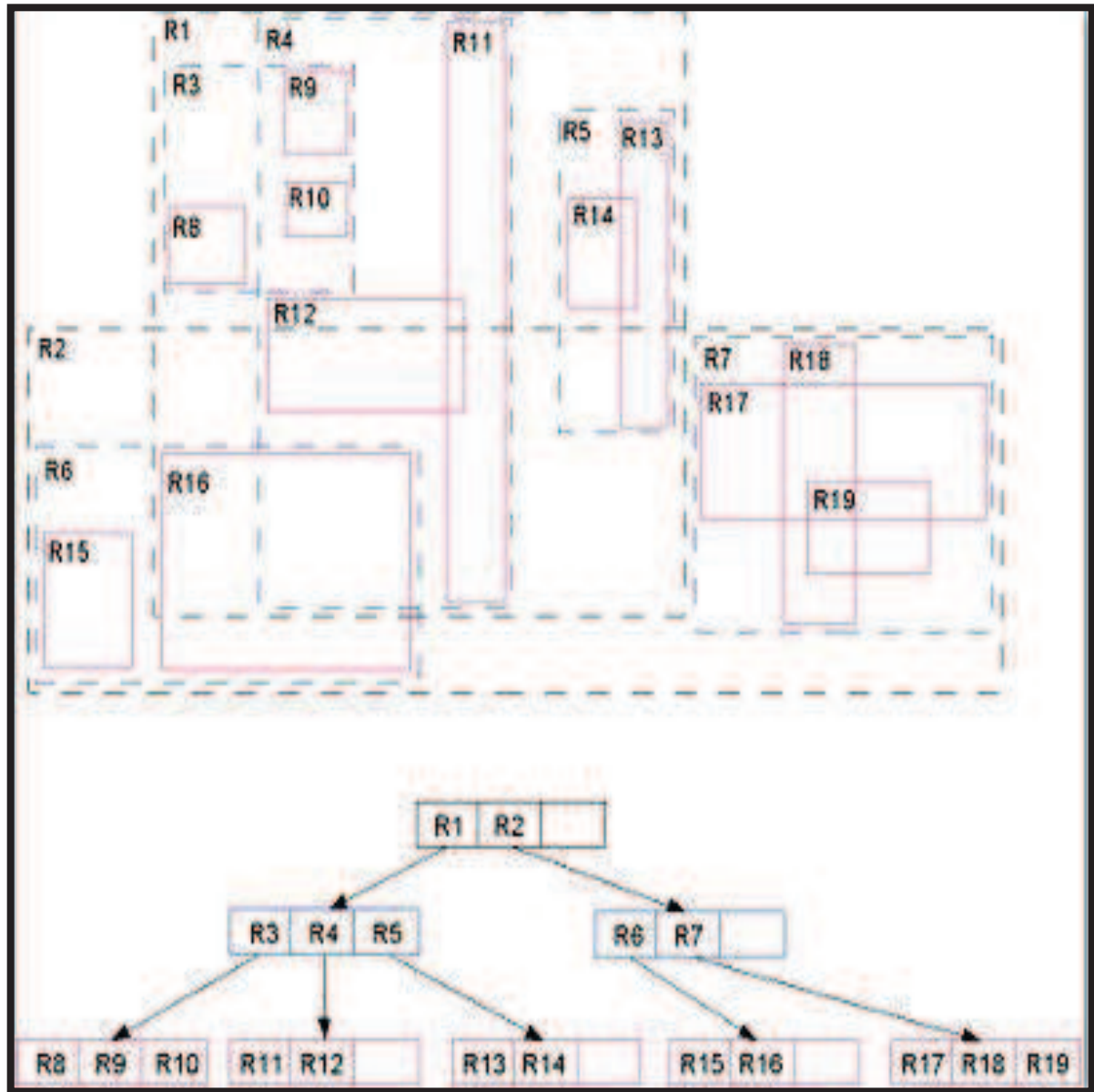


Figura 2.13 Análisis Jerárquico utilizando R-tree

Fuente: *R-trees: An Average Case Analysis*. Recuperado de:

https://www.google.com.ec/url?sa=t&rct=j&q=&esrc=s&source=web&cd=13&ved=0CD0QFjAC0Ao&url=http%3A%2F%2Fwww.cs.bu.edu%2Ffac%2Fgkollios%2Fada05%2FLectNotes%2Flect12-05.ppt&ei=E-8SUoWWF_GWyAHS4YGDw&usg=AFQjCNEz-kikyXKFueFZR4Wt8njvqeoK0g. (Marzo 2012).

2.4.5 FUNCIONES ESPACIALES

Una base de datos espacial debe proveer funciones específicas para manejo de datos espaciales que se presentan en la Tabla 2.7.

Tabla 2.7
Funciones Espaciales

Función	Descripción
Constructores	Crear geometrías a partir de datos tipo texto o binarios.
Serialización	De una geometría obtener un arreglo de caracteres que la defina (GML, KML, JSON, etc.).
Consultas con predicado	Comparar relaciones entre geometrías y devolver valores falso o verdadero.
Medición	Determinar áreas, distancias, etc.
Extracción	Extraer porciones de una geometría dada.
Funciones de agregado	Determinar estadísticas de conjunto de geometrías.
Buffers	Determinar áreas de influencia, alrededor de otras geometrías.

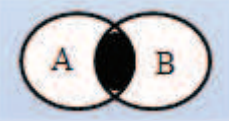
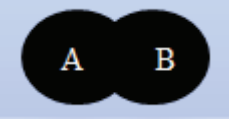
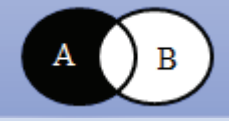

Fuente: Adaptado de: *EPSG Geodetic Parameter Dataset*. Recuperado de: <http://www.epsg.org/geodetic.html> (Enero 2012).

El resultado de cualquier función espacial derivado de dos objetos espaciales sólo es válido si dichos objetos tienen el mismo SRID basado en la misma unidad de medida, dato y proyección usados para determinar las coordenadas de los objetos. Las unidades de medida más comunes de un SRID son metros o metros cuadrados.

2.4.6 OPERACIONES ENTRE OBJETOS ESPACIALES

Conjunto de procedimientos que permiten realizar operaciones entre objetos espaciales. Su funcionamiento es similar a las operaciones entre conjuntos (*and*, *or*, *xor*, etc.), esto se debe a que, cuando se define un objeto espacial, dentro de una columna espacial. Ésta se almacena como un arreglo o *array* de longitudes y latitudes.

Tabla 2.8
Operaciones entre Objetos Espaciales

Operación	Descripción	
Intersección (A, B)	Retorna un objeto espacial, el cual topológicamente es la intersección de A y B.	
Unión (A, B)	Retorna un objeto espacial, el cual topológicamente es la unión de A y B.	
Diferencia (A, B)	Retorna un objeto espacial, el cual topológicamente es la diferencia de A y B.	
Diferencia simétrica (A, B)	Retorna un objeto espacial, el cual topológicamente es la diferencia simétrica de A y B.	

Fuente: Adaptado de: OGC Simple Feature Access - Part 1: Common Architecture. Versión Electrónica. Recuperado de: http://portal.opengeospatial.org/files/?artifact_id=25355%E2%80%8E. (Marzo 2012).

2.4.7 ANÁLISIS COMPARATIVO DE BASES DE DATOS CON SOPORTE PARA DATOS ESPACIALES

Como se analizó anteriormente, se debe almacenar toda la información geográfica dentro de una base de datos que brinde soporte para éste tipo de información. Se analizarán los motores de base de datos *Postgres 8.4* (de libre distribución) y Oracle 10g (no de libre distribución). Se escogió *Postgres 8.4*, ya que dentro las opciones libres, posee un módulo dedicado a brindar soporte para información geográfica (*PostGis*) y se ha convertido en un estándar para desarrollo de aplicaciones geo-referenciadas. La opción de Oracle 10g, se debe a que es el motor que actualmente maneja la SUPERTEL. En la Tabla 2.9, se comparan las características más relevantes de éstas dos opciones.

Tabla 2.9
Análisis Comparativo Bases de Datos con Soporte para Datos Espaciales

	ORACLE 10g	POSTGRES 8.4
Licenciamiento	Propietario.	FLOSS (Free and open source).
Sistema operativo	Windows, Linux.	Windows, Linux, Mac.
Módulo espacial	Oracle Spatial.	PostGis.
Funciones ETL (extraer, transformar, cargar) para datos espaciales	Si, shp2sdo (aplicación de consola).	Si, shp2pgsql (con interfaces gráficas desde la versión 1.5), QuantumGIS.
Controladores para aplicativos de tipo espacial	OGR/GDAL, SharpMap.net.	SharpMap.NET, JDBC postgis.jar, OGR/GDAL.
Mapeo Objeto- Relacional	Hibernate Spatial.	Hibernate Spatial.
Compatibilidad SIG de escritorio (Libres y comerciales)	GvSig, QuantumGIS, uDig, GeoRaptor, ESRI ArcGIS SDE, FME, Manifold, CadCorp, Autocad FDO y MapInfo.	OpenJump, QuantumGIS, GvSig, uDig, ESRI ArcGIS, ZigGIS, Manifold, FME, CadCorp, Autocad FDO y MapInfo.
Compatibilidad SIG WEB	Manifold, MapDotNet, Mapserver, GeoServer, MapGuide ArcGIS.	Manifold, MapDotNet, ArcGIS, Mapserver, GeoServer, FeatureServer, MapGuide Open Source.
Funciones espaciales	175 entre funciones, transformaciones, y herramientas de minería.	Más de 300 funciones y operaciones.
Indexado espacial	<i>R-trees</i> .	GIST (basado en <i>R-trees</i> , incluye mejores, evita indexar registros nulos, filtra la información a ser indexada).
Soporte para manejo de coordenadas geodésicas	Si, desde la versión 9i.	No.
Basado en estándares OGC 06-104r4²⁸	Si.	Si.

Fuente: Adaptado de: *Oracle Spatial Developer's Guide*. Recuperado de: http://docs.oracle.com/cd/B28359_01/appdev.111/b28400/sdo_intro.htm. (Marzo 2012). & *PostGis. PostGis Documentation*. Recuperado de: <http://postgis.net/documentation>. (Marzo 2012).

²⁸ Estándar que define requerimientos para almacenar información geográfica en una base datos relacional.

Se pueden observar muchas similitudes en la comparación realizada, pero el punto de discrepancia es: *Oracle Spatial 10g* brinda soporte para coordenadas geodésicas y permite realizar operaciones con este tipo de datos, sin necesidad de realizar transformaciones. Tomando en cuenta que los datos proporcionados (*shapefiles*) por la SUPERTEL utilizan un SRID EPSG 4326 con *datum* WGS84, la base de datos a utilizar va a ser *Oracle Spatial 10g*.²⁹

2.5 SERVICIOS DE INFORMACIÓN GEOGRÁFICA

En la norma ISO 19119 se define un entorno de trabajo para el desarrollo de aplicaciones; en el cual, los usuarios puedan acceder a información geográfica proveniente de diversas fuentes, mediante el uso de interfaces genéricas dentro de un ambiente tecnológico abierto a través de una interfaz, una aplicación, proporciona parte de su funcionalidad es decir brinda un servicio.

La norma ISO 19119 define a una interfaz, como un conjunto de operaciones que caracterizan una aplicación. Y una operación, es definida como la especificación de una transformación o consulta, que un objeto puede recibir para que se ejecute y devuelva un valor. Esta operación posee un nombre y una lista de parámetros.

La estandarización de las interfaces garantiza la interoperabilidad entre los diferentes SIG. [30]

2.5.1 ARQUITECTURA DE SERVICIOS

La arquitectura ISO 19190 está basada en un modelo multicapa. Se utiliza una arquitectura lógica como modelo de referencia que posee cuatro capas. Este modelo puede ser adaptado a diferentes arquitecturas físicas que pueden ir desde aplicaciones monolíticas hasta arquitecturas cliente/servidor. Una arquitectura lógica es un conjunto de servicios y sus correspondientes interfaces representadas en el sistema.

La arquitectura física es un conjunto de componentes e interfaces asociadas que implementan servicios. Dentro de las arquitecturas cliente servidor, la norma distingue dos tipos de clientes: clientes pesados (*thick*) y livianos (*thin*). Los

²⁹ Versión manejada en la actualidad por la SUPERTEL.

clientes pesados contendrán gran parte de la funcionalidad en el servicio de usuario. En cambio, los clientes livianos contendrán principalmente los servicios de presentación e interacción del usuario. [30]

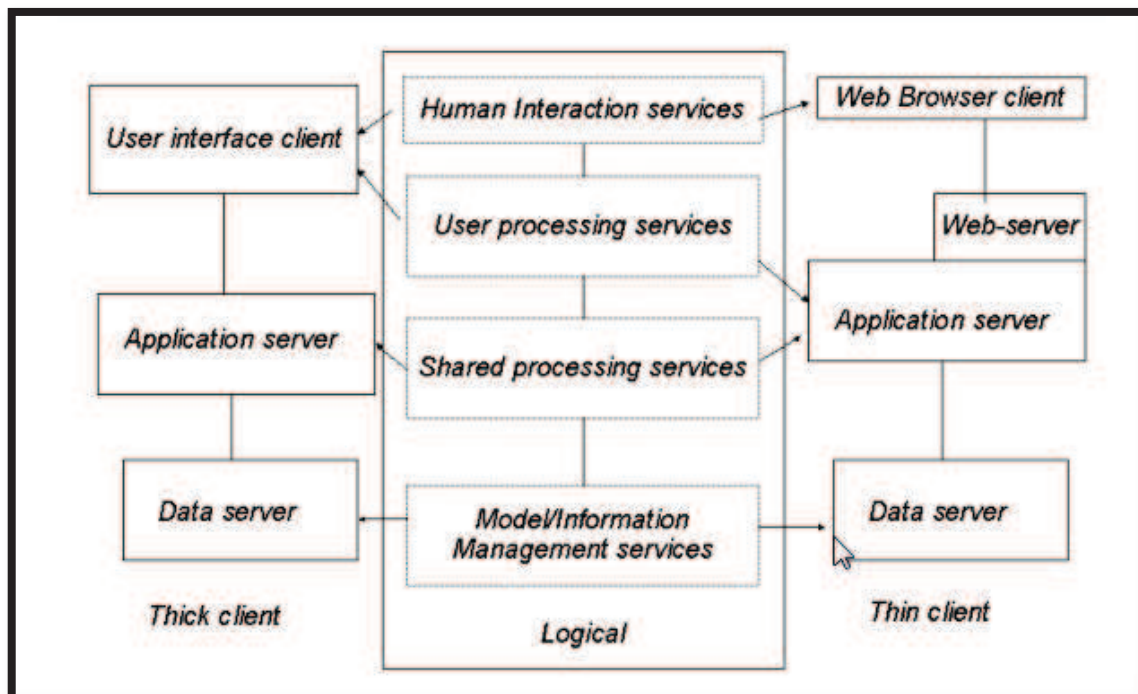


Figura 2.14 Arquitectura Lógica de 4 niveles y su Interacción con Clientes Livianos y Pesados

Fuente: *Introducción a la Normalización Información Geográfica Familia ISO 19100*. Versión Electrónica. Disponible en:

http://coello.ujaen.es/Asignaturas/pcartografica/Recursos/IntroduccionNormalizacion_IG_FamiliaISO_19100_rev1.pdf. (Marzo 2012).

Un navegador Web, es un ejemplo de un cliente liviano o *thin client*, corresponde a la interfaz de usuario del cliente, la cual, interactúa con un servidor web mediante el uso del protocolo HTTP y su contenido está representado como HTML y/o XML.

2.6 CATÁLOGO DE SERVICIOS OGC

EL OGC, en su documento OGC 07-006r1, con fecha 2007-02-23 define un catálogo de servicios. EL catálogo de servicios tiene la capacidad de publicar, acceder y buscar; metadatos³⁰ correspondientes a datos, servicios o fuentes de

³⁰ Información de los datos, comúnmente llamados datos de los datos.

información geográfica. Los metadatos representan características que pueden ser consultadas, procesadas y presentadas. Los Servicios de búsqueda de metadatos *Catalogue Service Web* son los que permiten el acceso a los catálogos de metadatos. Estos catálogos son uno de los tres servicios fundamentales que debe existir en una Infraestructura de Datos Espaciales: consulta, visualización, y descarga. Este servicio, al igual que los otros que se ofrecen deben cumplir con el requisito indispensable de ser interoperables.

El documento OGC 07-006r1 en su sección 7 define al catálogo general de modelos de interface (*General Catalogue Interface Model*) como un conjunto de interfaces abstractas de servicio que permiten el descubrimiento, acceso, mantenimiento y organización de los catálogos que contienen información geoespacial y sus fuentes. Estas interfaces están destinadas a permitir tanto a usuarios como aplicaciones encontrar información que se encuentre en diferentes ambientes distribuidos, incluyendo a la *World Wide Web* (WWW).

Para el caso de las aplicaciones se definen perfiles, estos perfiles se basan en el protocolo de enlace a utilizarse, por ejemplo HTTP.

2.7 WEB MAP SERVICE

El estándar *WMS* proporciona una interfaz HTTP para solicitar imágenes de mapas geo-referenciados, provenientes de bases de datos espaciales o archivos *shapefiles*. Una solicitud *WMS* define las capas geográficas y áreas de interés a ser procesadas. La respuesta a esta solicitud es una o un conjunto de imágenes (JPEG; PNG, etc.) de mapas geo-referenciados. [29]

Un *Web Service* es definido como un servicio disponible en el Internet, usa XML para el intercambio de mensajes. Su diseño debe soportar interoperabilidad, usa WSDL³¹ para describir la interfaz del servicio. Otros sistemas interactúan con el servicio web, en la forma prescrita en su descripción, y utilizando mensajes SOAP³².

³¹ Lenguaje de descripción de servicios web (WSDL).

³² Simple Object Access Protocol, protocolo basado en XML, que permite intercambio de información entre aplicaciones sobre HTTP.

Un *Web Map Service (WMS)* produce de forma dinámica mapas, los mapas se generan a partir de datos espaciales que tengan definido un sistema de referencia espacial. Los mapas generados son imágenes en formatos conocidos como PNG, GIF o JPEG. Se accede a los mapas generados a través de la red.

Un aspecto importante a destacar es que no se tiene un acceso a los datos sino a una representación geográfica de ellos. Los datos espaciales pueden provenir de distintos servidores. Un WMS permite la integración de datos ráster y vectoriales en una aplicación web.

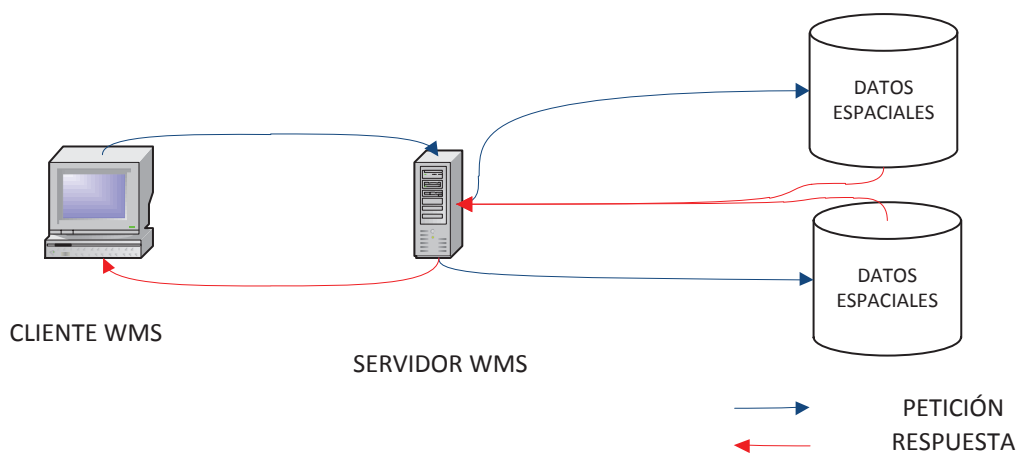


Figura 2.15 Funcionamiento WMS

Fuente: Adaptado de: *OpenGIS Web Map Service (WMS) Implementation Specification*. Versión Electrónica. Recuperado de: https://portal.opengeospatial.org/files/?artifact_id=1081&version. (Marzo 2012).

El estándar internacional para WMS es el ISO 19128: Interfaz de servidor web de mapas. El estándar originalmente definido por OGC especifica el comportamiento de un servicio que produce mapas geo-referenciados de manera dinámica. Especifica operaciones para:

- Devolver la descripción de los mapas ofrecidos por un servicio (metadatos).
- Devolver un mapa.
- Realizar consultas acerca de características específicas mostradas en un mapa.

En resumen, la especificación se encarga de estandarizar la forma en que los clientes consultan mapas geo-referenciados y la manera en que los servidores describen los metadatos de estos mapas. En la Figura 2.16

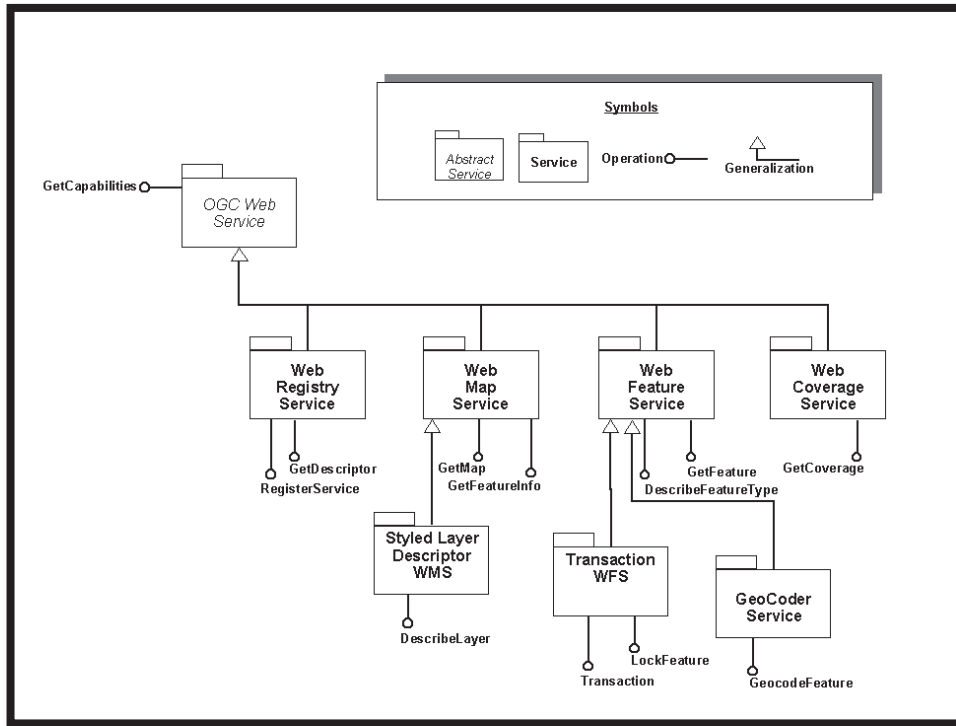


Figura 2.16 se describe la arquitectura propuesta por el OGC para los servicios de mapas.

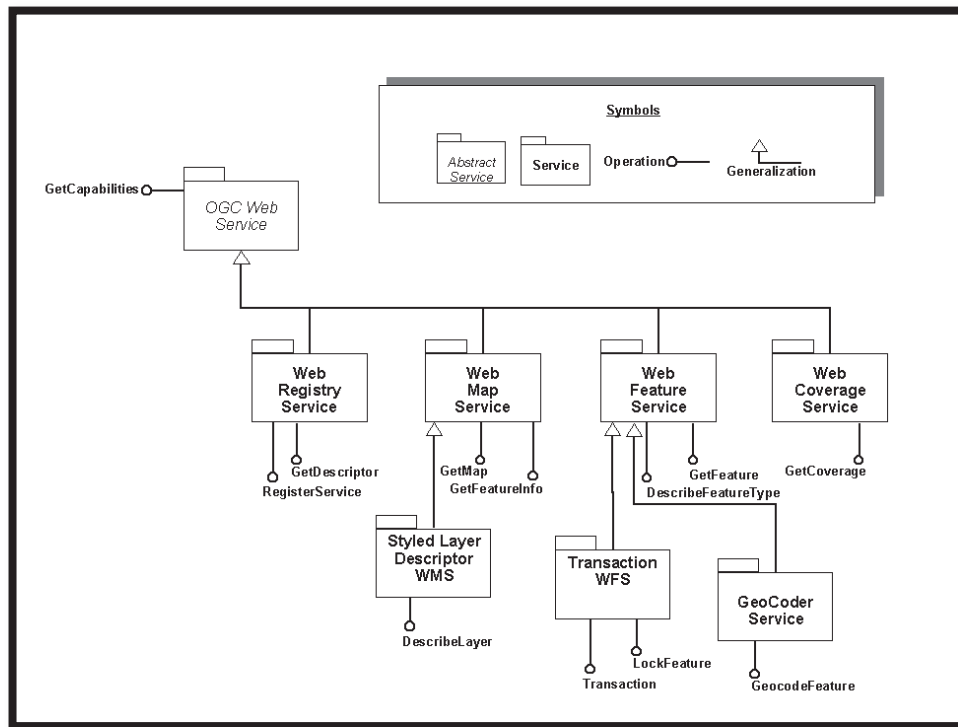


Figura 2.16 Arquitectura Servicios Web - OGC

Fuente: *Introducción a la Normalización Información Geográfica Familia ISO 19100*. Versión Electrónica.

Disponible en:

http://coello.ujaen.es/Asignaturas/pcartografica/Recursos/IntroduccionNormalizacion_IG_FamiliaISO_19100_rev1.pdf. (Marzo 2012).

La OGC define tres operaciones para WMS:

- **GetMap:** Obtiene una imagen de un mapa creado.
- **GetCapabilities:** informa sobre mapas que el servicio puede crear, sus características (metadatos del servicio como SRID, tipos de coordenadas manejados, etc.).
- **GetFeatureInfo:** informa sobre características de objetos en particular contenidos en un mapa.

2.7.1 GETMAP

Cuando un cliente realiza una petición *GetMap*, está en capacidad de definir la información como: las capas a mostrar en el mapa (superposición de capas), los

posibles estilos para representar dichas capas, la porción de la tierra cartografiada, el sistema de referencia de coordenadas a usar, el formato de salida, el tamaño de la imagen de salida, el color y la transparencia de fondo. En la Tabla 2.10, se describe la lista de parámetros definibles dentro de una petición *GetMap*:

Tabla 2.10
Parámetros de Operación *GetMap*

Parámetro Solicitado	Obligatorio (M) / Opcional (O)	Descripción
<i>VERSION=versión</i>	M	Versión de la petición.
<i>REQUEST=GetMap</i>	M	Nombre de la petición.
<i>LAYER=layer_list</i>	M	Lista de capa(s) de los mapas separados por comas.
<i>STYLES=style_list</i>	M	Lista de estilo(s) de representación, por capa solicitada, separados por coma.
<i>CRS=namespace:identifier</i>	M	Sistema de referencia de coordenadas.
<i>BBOX=minx, miny, maxx, maxy</i>	M	Esquinas del cuadro delimitador, medido en CRS.
<i>WIDTH=output_width</i>	M	Ancho del mapa, en pixeles.

Tabla 2.11
Parámetros de Operación *GetMap* (continuación)

Parámetro Solicitado	Obligatorio (M) / Opcional (O)	Descripción
-----------------------------	---------------------------------------	--------------------

<i>HEIGHT=output_height</i>	M	Altura del mapa, en pixeles.
<i>FORMAT=output_format</i>	M	Formato de salida del mapa.
<i>TRANSPARENT=TRUE FALSE</i>	O	Transparencia de fondo del mapa, por defecto FALSE.
<i>BGCOLOR=color_value</i>	O	Valor del color (en formato hexadecimal) para el fondo.
<i>EXCEPTIONS=exception_format</i>	O	Formato para reportar excepciones, por defecto XML.
<i>TIME=time</i>	O	Valor de tiempo deseado para la capa.
<i>ELEVATION=elevation</i>	M	Elevación deseada para la capa.
<i>Other sample dimensions(s)</i>	M	Valor de otras dimensiones como apropiadas.

Fuente: Adaptado de: *OpenGIS Web Map Service (WMS) Implementation Specification*. Versión Electrónica. Recuperado de: https://portal.opengeospatial.org/files/?artifact_id=1081&version. (Marzo 2012).

Cuando se producen varias capas de características similares (igual tamaño, mismo sistema de referencia de coordenadas), se pueden superponer para producir una composición de capas. EL uso de formatos de imagen como PNG o GIF, que soportan transparencia, permite que las todas capas superpuestas puedan ser observadas.

2.7.2 GETCAPABILITIES

Cuando un cliente realiza una petición de tipo *GetCapabilities*, obtiene como respuesta un archivo XML, que contiene los metadatos del servicio ofrecido conforme al formato definido en el archivo XSD³³ del servicio. El primer bloque de metadatos, llamado `<Service>`, provee información general acerca del servicio:

- Nombre, título, URL del servicio (obligatorio).
- Lista de palabras clave, información del contacto, etc. (opcional)

```

WMS_GetCapabilities.xml

<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="WMS_GetCapabilities_with_XSLT.xsl"?>
<WMT_MS_Capabilities version="1.1.1">
<Service>
<Name>OGC:WMS</Name>
<Title>LUBW (BW) Kartendienst GDI_SCHUTZGEBIET_NATUR</Title>
<Abstract>LUBW GDI_SCHUTZGEBIET_NATUR Web Map Service</Abstract>

```

Figura 2.17 Ejemplo de Respuesta *GetCapabilities* - formato XML

Fuente: Esquema de Autoría propia (Arévalo, D. y Guzmán, D.).

Luego se tiene un bloque donde se describen la lista de operaciones soportadas por el servidor, los formatos de salida para cada una de esas operaciones y la URL para cada operación. En la Tabla 2.12 se describen los parámetros que conforman una solicitud de tipo *GetCapabilities*.

Tabla 2.12
Parámetros de una Operación *GetCapabilities*

³³ XML Schema (XSD): define la correcta estructura de los elementos del documento XML

Parámetro Solicitado	Obligatorio (M) / Opcional (O)	Descripción
VERSION=versión	O	Versión de la petición.
SERVICE=WMS	M	Tipo de servicio.
REQUEST=GetCapabilities	M	Nombre de la petición.
FORMAT=MIME_type	O	Formato de salida de los metadatos del servicio.
UPDATESEQUENCE=string	O	Números de secuencia o cadenas de caracteres para control de caché.

Fuente: Adaptado de: *OpenGIS Web Map Service (WMS) Implementation Specification*. Versión Electrónica. Recuperado de: https://portal.opengeospatial.org/files/?artifact_id=1081&version. (Marzo 2012).

2.7.3 GETFEATUREINFO

Invocando una petición de tipo *GetFeatureInfo* el cliente indica el mapa que desea consultar, además de una localización (desplazamiento en el eje transversal y longitudinal que inicia en la esquina superior izquierda del mapa) de la cual desea recabar información. Para poder ejecutar esta petición sobre una capa su atributo consultable debe estar definido o haber sido heredado como “TRUE” o verdadero. Los parámetros que se describen en la Tabla 2.13 conforman una petición *GetFeatureInfo*.

Todas estas operaciones pueden ser invocadas usando un navegador web enviando peticiones en las URLs³⁴. El contenido de la URL dependerá del tipo de operación a realizarse. Al definir la URL se debe tomar en cuenta los caracteres especiales reservados (Tabla 2.14) para el uso exclusivo de WMS.

Tabla 2.13
Parámetros de Operación *GetfeatureInfo*

³⁴ Siglas de Uniform Resource Locator (URL).

Parámetro Solicitado	Obligatorio (M) / Opcional (O)	Descripción
VERSION=1.3.0	M	Versión de la petición.
REQUEST=GetFeatureInfo	M	Nombre de la petición.
Map request part	M	Copia parcial de los parámetros del mapa solicitado, del cual se requiere información.
QUERY_LAYERS=layer-list	M	Lista de capa(s) requeridas, separadas por coma.
INFO_FORMAT=output_format	M	Formato de salida para información de características.
FEATURE_COUNT=number	O	Número de características sobre las cuales se va a recibir información, por defecto 1.
I=pixel_column	M	Coordenada en I de la característica solicitada del mapa.
J=pixel_column	M	Coordenada en J de la característica solicitada del mapa.
EXCEPTIONS=exceptions-format	O	Formato en que las excepciones van a ser reportadas, XML por defecto.

Fuente: Adaptado de: *OpenGIS Web Map Service (WMS) Implementation Specification*. Versión Electrónica. Recuperado de: https://portal.opengeospatial.org/files/?artifact_id=1081&version. (Marzo 2012).

Tabla 2.14
Caracteres Especiales Reservados - WMS

Caracter	Uso Reservado
?	Separador que indica el inicio de la cadena de consulta.
&	Separador entre parámetros de la cadena de consulta.
=	Separador entre el nombre y valor de un parámetro.
'	Separador utilizado para diferenciar valores dentro de una lista, los cuales conforman un solo valor para un parámetro.
+	Representa un caracter espacial.

Fuente: Adaptado de: *OpenGIS Web Map Service (WMS) Implementation Specification*. Versión Electrónica. Recuperado de: https://portal.opengeospatial.org/files/?artifact_id=1081&version. (Marzo 2012).

La OGC define también la implementación de WMS sobre una plataforma de aplicaciones distribuidas que comprende hosts de Internet que soportan *Hypertext Transfer Protocol* - Protocolo para Transferencia de Hipertexto (HTTP). HTTP soporta dos métodos para realizar peticiones GET y POST, la diferencia radica en que usando el método GET, los parámetros enviados hacia el servidor viajan contenidos en la URL (visibles al cliente) mientras que, usando el método POST, los parámetros viajan “ocultos” al cliente.

La especificación WMS OGC 03-109r1 define que al usar una petición HTTP GET, los parámetros del WMS a enviarse deben ir codificados usando “*Keywords/Value Pairs (KVP)*”; y separados mediante el símbolo “&” (símbolo que se encuentra en la tabla 2.13).

Actualmente se usan dos versiones WMS, la versión WMS 1.1.1 y la versión 1.3.0. La diferencia entre estas dos especificaciones, está en el sistema de referencia a utilizar. WMS 1.1.1 utiliza *Spatial Reference System (SRS)*, en tanto que WMS 1.3.0 utiliza *Coordinate Reference System (CRS)*. Al utilizar SRS, la referencia a los objetos espaciales se la hace a través de índices: utilizando CRS a través de coordenadas. En la Tabla 2.15, se ejemplifica una petición *GetMap* para WMS 1.1.1 usando método GET y método POST.

Tabla 2.15
Petición GetMap Usando Métodos GET Y POST

Tipo de Petición	Detalle
Método GET	http://www.airsip.org/wms/process.cgi?REQUEST=GetMap&FORMAT=image/gif&WIDTH=640&HEIGHT=480&LAYERS=temperature&SRS=EPSG:4326&BBOX=110.,40.,-80.,30&VERSION=1.1.1
Método POST	<pre><?xml version="1.0" encoding="UTF-8"?> <ogc:GetMap xmlns:ogc="http://www.opengis.net/ows" xmlns:gml="http://www.opengis.net/gml" version="1.1.1" service="WMS"> <StyledLayerDescriptor version="1.0.0"> <NamedLayer> <Name>states</Name> <NamedStyle> <Name>population</Name> </NamedStyle> </NamedLayer> </StyledLayerDescriptor> <BoundingBox srsName="http://www.opengis.net/gml/srs/epsg.xml#4326"> <gml:coord> <gml:X>-130.0</gml:X> <gml:Y>24.0</gml:Y> </gml:coord> <gml:coord> <gml:X>-66.0</gml:X> <gml:Y>50.0</gml:Y> </gml:coord> </BoundingBox> <Output><Format>image/png</Format> <Size> <Width>600</Width> <Height>320</Height> </Size> </Output> <Exceptions>application/vnd.ogc.se+xml</Exceptions> </ogc:GetMap></pre>

Fuente: Tabla de Autoría propia (Arévalo, D. y Guzmán, D.).

2.8 ANÁLISIS DE REQUERIMIENTOS PARA SERVIDORES DE MAPAS

Dentro de las dos opciones libres que lideran en el desarrollo de aplicaciones geo-referenciadas destacan *Geoserver* y *Mapserver*. En la Tabla 2.16 se comparan ciertas características (las más relevantes para el desarrollo del presente proyecto) para poder determinar el servidor de mapas más idóneo.

El servidor de mapas a utilizar será *Mapserver* (MS4W versión para Windows), debido al soporte que brinda *Mapserver* para consultas utilizando sentencias SQL, aspecto fundamental debido a que se utiliza un modelo de datos relacional para almacenar los datos de sectorización de la ciudad de Quito y parámetros de QoS.

Tabla 2.16
Análisis Comparativo de Servidores de Mapas

Parámetro	Geoserver	Mapserver
Tecnología	J2EE ³⁵	CGI ³⁶
Administración	WEB	Archivo de configuración <i>mapfile</i> (archivo texto plano).
Soporte WMS	SI	SI
Soporte Oracle 10g – Oracle Spatial	SI	SI (Con librería extra GDAL)
Consultas a bases de datos	Filtros OGC	Sentencias básicas SQL

Fuente: Adaptado de: *Open Source Geospatial Foundation. MapServer 6.2.0 Documentation*. Recuperado de: <http://mapserver.org/documentation.html>. (Septiembre de 2013). & *GeoServer Documentation*. Recuperado de: <http://docs.geoserver.org/>. (Septiembre 2013).

El servidor de mapas responderá a las peticiones entregando imágenes de los mapas o figuras solicitadas, es importante realizar un análisis del formato de las imágenes. Dentro de las opciones que MS4W ofrece para el manejo de imágenes, tenemos los formatos *Joint Photographic Experts Group (JPEG)*, *Portable Network Graphics (PNG)* y *Graphics Interchange Format (GIF)*.

Como requerimiento fundamental para el formato a utilizar, tenemos el concepto de transparencia, debido a que el mapa de la ciudad de Quito junto con las sectorizaciones y parámetros de QoS estarán sobrepuestos unos con otros. Por tanto, el formato de imagen escogido deberá soportar transparencia. Partiendo de este requerimiento, el formato JPEG queda descartado, ya que no soporta transparencia. PNG y GIF brindan soporte para transparencia mediante el uso del canal alfa.

La diferencia es que PNG es un formato que maneja modos de color RGB con una combinación de colores máxima de 16.7 millones de colores. En cambio GIF es un formato en el cual la combinación de colores máxima es 256 colores. PNG

³⁵ Plataforma Java para desarrollo de aplicaciones web.

³⁶ Common Gateway Interface (Interfaz de entrada común), tecnología pionera en el desarrollo de aplicaciones web.

es un formato que ofrece mejor calidad frente a GIF, sin embargo es importante analizar el tipo de imagen que se va a manejar. Por ejemplo, si fueran fotografías de alta calidad, existiría diferencia pero debido a que son imágenes de polígonos, punto, líneas o mapas, no existiría diferencia en cuanto a calidad de imagen. A continuación se realizará una comparación en cuanto al tamaño de la imagen generada, a través de una petición de tipo *GetMap*.

La Figura 2.18 y la Figura 2.19 muestran la sectorización de la ciudad de Quito. No se observa diferencias en la imágenes generadas, sin embargo, mientras el tamaño de la imagen generada en formato PNG es de 19.9 KB, la imagen en formato GIF es de tamaño 10.7KB. Diferencia significativa y motivo por el cual el formato de imagen será GIF.

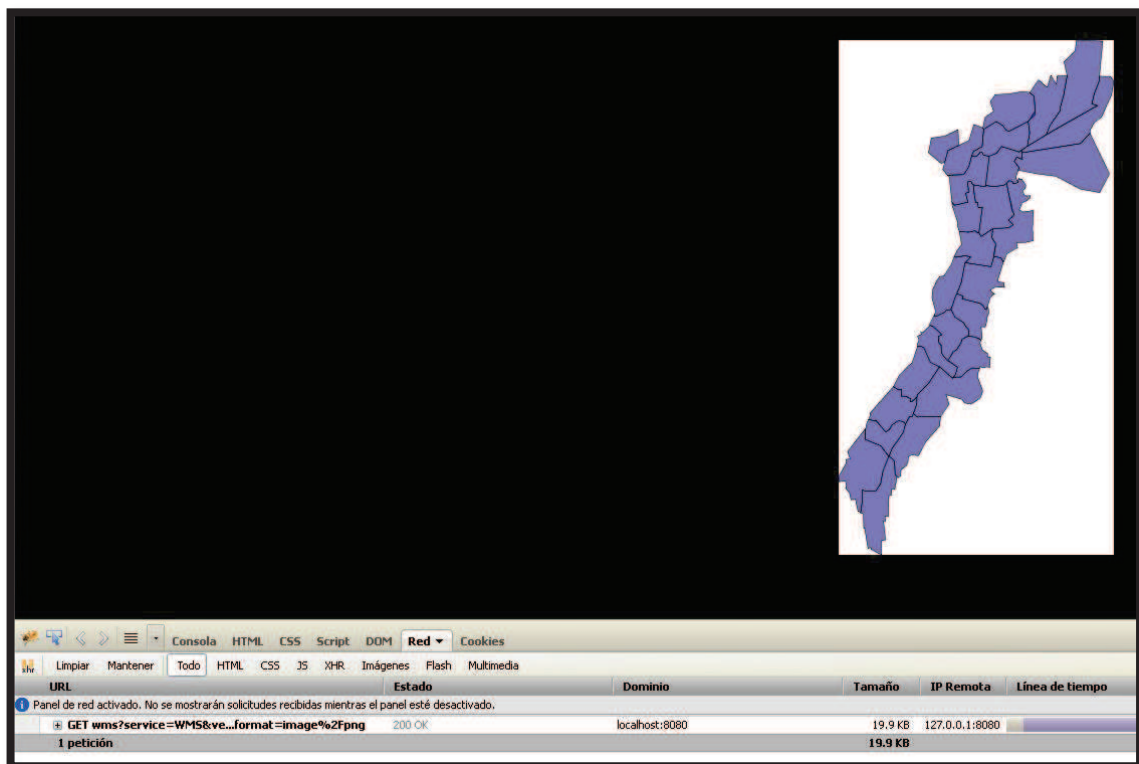


Figura 2.18 Imagen Sectorización Quito formato PNG

Fuente: Esquema de Autoría propia (Arévalo, D. y Guzmán, D.).

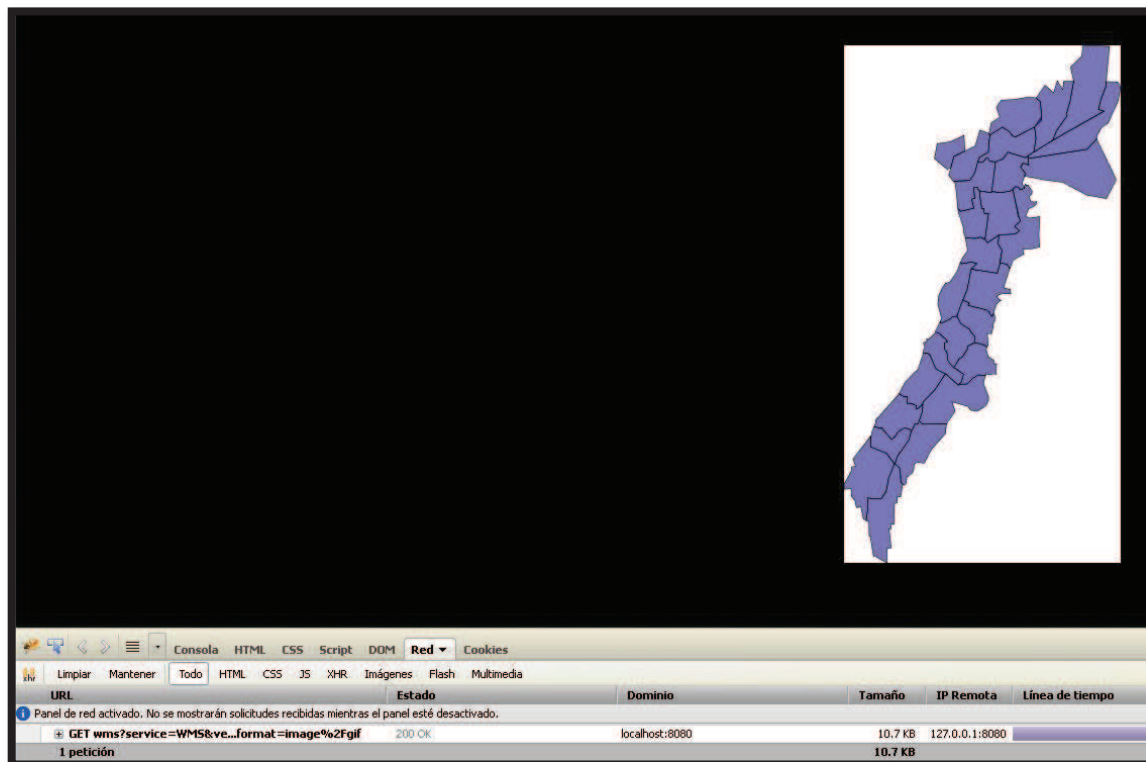


Figura 2.19 Imagen Sectorización Quito formato GIF

Fuente: Esquema de Autoría propia (Arévalo, D. y Guzmán, D.).

2.9 DIMENSIONAMIENTO DEL/Los SERVIDORES

Luego del análisis previo realizado en el presente capítulo sobre las tecnologías que brindan soporte para los requerimientos de la aplicación a desarrollarse, además de la situación actual de la información y procesos a automatizarse³⁷, dentro del sistema de monitoreo y control que la SUPERTEL mantiene sobre las operadoras del SMA, es fundamental determinar la infraestructura que permita el desarrollo de la aplicación, así como su alojamiento, utilización, y administración.

Un reto inicial dentro del desarrollo del presente proyecto, es emular (en parte), la infraestructura que tiene la SUPERTEL para el manejo y utilización de la información técnica (recopilada y procesada), de las mediciones de parámetros de QoS realizadas.

³⁷ Específicamente en lo concerniente a: Divulgación de la información a la ciudadanía en general de los parámetros de QoS medidos y recepción de reclamos/quejas por parte de los usuarios del SMA.

2.9.1 DETERMINACIÓN DEL AMBIENTE DE DESARROLLO

La infraestructura será de tipo distribuida, es decir, existirán varios servidores, alojados en diferentes equipos anfitrión (host) sean estos físicos o virtuales. Cada servidor proveerá funcionalidades específicas a la aplicación.

Los servidores necesarios son:

1. Servidor de base de datos (para el alojamiento de los datos espaciales).
2. Servidor web (contenedor de la aplicación web a desarrollarse).
3. Servidor de mapas (generar imágenes de sectores, zonas, etc. a partir de los datos espaciales).
4. Servidor de versionamiento (almacenamiento del código fuente).

El ambiente de desarrollo propuesto se detalla en la Figura 2.20.

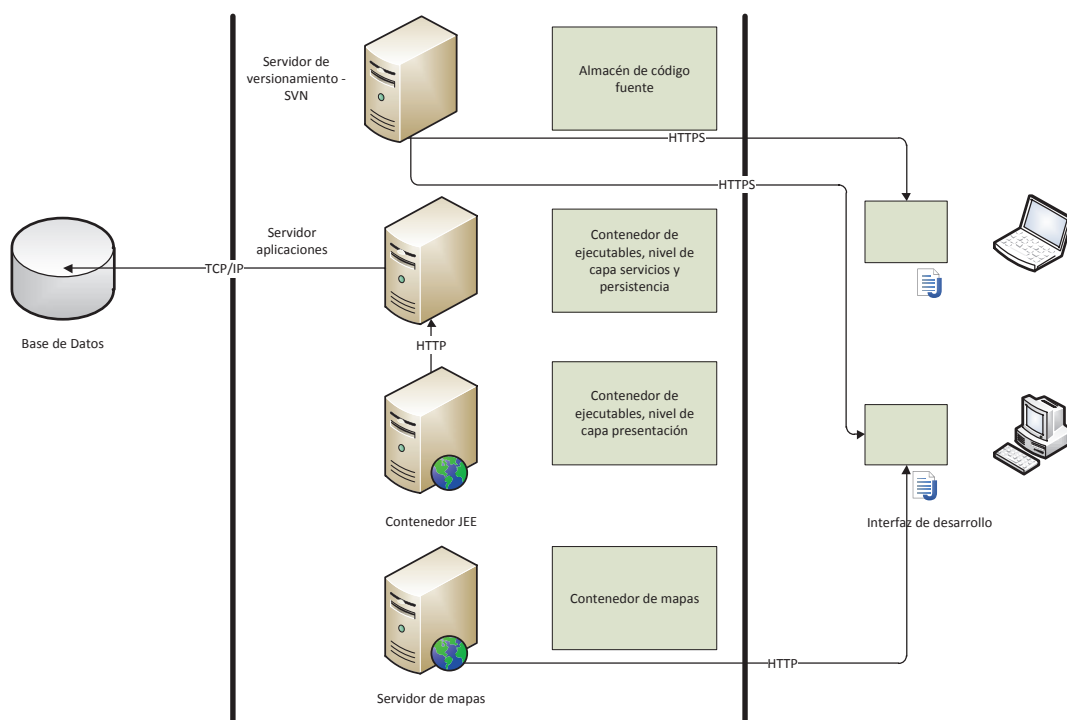


Figura 2.20 Ambiente de Desarrollo

Fuente: Esquema de Autoría propia (Arévalo, D. y Guzmán, D.).

2.9.2 REQUERIMIENTOS EN SOFTWARE Y HARDWARE

Se ha optado por virtualizar³⁸ los servidores detallados en la Figura 2.20 debido a que con la virtualización se optimizan los recursos en hardware, además de garantizar flexibilidad a la hora de reasignar recursos a cada servidor. Por lo tanto se tendrá un equipo host (servidor físico) donde se alojen tres servidores virtuales. El sistema operativo escogido para el equipo anfitrión es Debian *Squeeze*, este es un sistema operativo liviano y tiene compatibilidad garantizada con el resto de software a utilizarse.

Tabla 2.17
Requerimientos Mínimos del Sistema Operativo Debian *Squeeze*

Tipo de instalación	RAM ³⁹ (mínimo) MB	RAM (recomendado) MB	Espacio en Disco Duro GB
Sin ambiente gráfico	64	256	1 Gigabyte
Con ambiente gráfico	128	512	5 Gigabyte

Fuente: Adaptado de: *Debian Documentation*. Recuperado de: <http://www.debian.org/doc/>. (Septiembre de 2013).

Como software para virtualización, la solución a implementarse es *VMware Player versión 8.0*. *VMware Player* es la opción gratuita dentro del conjunto de software de virtualización a pequeña escala que ofrece *VMware*. *VMware Player* permite la creación, edición y ejecución de máquinas virtuales, además que garantiza compatibilidad absoluta con el sistema operativo anfitrión. En la Tabla 2.17, podemos visualizar los requerimientos mínimos de operación del paquete de software, cabe recalcar que estos requerimientos son para la instalación y ejecución de *VMware Player*, no tienen relación alguna con los requerimientos de las máquinas virtuales a instalarse.

³⁸ Dentro de un equipo físico (host o anfitrión), alojar varios equipos (virtuales) que van a utilizar y compartir el hardware del equipo host.

³⁹ Random-access memory (RAM), módulos de almacenamiento temporal que utiliza un computador para la carga y ejecución de procesos de su S.O. y otros programas.

Tabla 2.18
Requerimientos Mínimos para la Operación de VMware Player V9.0

CPU (mínimo) GHz	CPU (recomendado) GHz	RAM (mínimo) GB	RAM (recomendado) GB	Espacio en Disco Duro MB
1	2	1	2	150

Fuente: Adaptado de: *VMWare FAQs*. Recuperado de: <http://www.vmware.com/products/player/faqs.html>. (Enero 2012).

2.9.2.1 Servidor de Base de Datos

El sistema operativo de este servidor virtual será *Debian Squeeze* versión con interfaz gráfica. El motor de base de datos será *Oracle 10g Enterprise Edition*, teniendo los requerimientos mínimos detallados en la Tabla 2.18.

Tabla 2.19
Requerimientos mínimos para operación Oracle EE 10g

CPU (mínimo) GHz	CPU (recomendado) GHz	RAM (mínimo) GB	RAM (recomendado) GB	Espacio en Disco Duro GB
-	-	0.5	Mayor 0.5	1.5 - 3

Fuente: Adaptado de: *Oracle Developer's Guide*. Recuperado de: http://docs.oracle.com/cd/B28359_01/appdev.111/b28400/sdo_intro.htm. (Marzo 2012).

Adicionalmente, se requiere mínimo 1 GB en swap⁴⁰ (característica específica para Linux). Oracle no especifica una velocidad para el procesador ni número de núcleos necesarios, debido a que este parámetro será analizado en función de la cantidad de registros que va a manejar la base de datos. Razón por la cual este requerimiento se lo determinará cuando la base de datos este diseñada y los datos cargados en la base.

Además, se requiere de un paquete que permita de manera sencilla la administración de la base de datos (de preferencia con interfaz gráfica). Como solución se ha optado por *SQL Developer* versión 3.2, herramienta gratuita recomendada y distribuida por Oracle. Sus requerimientos de funcionamiento, sobre una plataforma Linux se detallan en la Tabla 2.20.

⁴⁰ Espacio en disco duro utilizado como memoria de intercambio.

Tabla 2.20
Requerimientos Mínimos para la Operación SQL Developer 10g

CPU (mínimo) GHz	CPU (recomendado) GHz	RAM (mínimo) GB	RAM (recomendado) GB	Espacio en Disco Duro MB
2	2	1	1	110

Fuente: Adaptado de: *SQL Developer Documentation*. Recuperado de: <http://www.oracle.com/technetwork/developer-tools/sql-developer/documentation/index.html>. (Marzo 2012).

Como un requerimiento adicional es necesario disponer del paquete JDK 1.6.11 o posterior para Linux.

2.9.2.2 Servidor de Mapas

Este servidor virtual, contará con un sistema operativo *Windows XP* y el paquete *Mapserver for Windows (MS4W)*. No se definen requerimientos de instalación para este tipo de servidores ya que son bastante básicos. Es necesario también que este equipo cuente con una instalación de QGIS versión 1.8, como SIG de escritorio. La Tabla 2.20, describe los requerimientos mínimos para el funcionamiento del servidor de mapas seleccionado.

Tabla 2.21
Requerimientos mínimos para operación Servidor de Mapas

CPU (mínimo) GHz	CPU (recomendado) GHz	RAM (mínimo) GB	RAM (recomendado) GB	Espacio en Disco Duro MB
1	1	0.5	1	200

Fuente: Adaptado de: *Open Source Geospatial Foundation. MapServer 6.2.0 Documentation*. Recuperado de: <http://mapserver.org/documentation.html>. (Septiembre de 2013).

2.9.2.3 Servidor Web

El servidor web a instalarse es *Apache-Tomcat* versión 7.1. De igual manera que en los anteriores casos los requerimientos son básicos para su instalación.

En base a los requerimientos detallados en las Tabla 2.17 a Tabla 2.21 se procede a levantar la línea base de requerimientos de hardware para el desarrollo del prototipo, como se aprecia Tabla 2.22.

Tabla 2.22
Requerimientos Mínimos de Hardware para el Ambiente de Desarrollo.

Requerimiento SOFTWARE	CPU GHz	RAM MB	Disco Duro MB
<i>Debian Squeeze Host</i>	2	512	5000
<i>Debian Squeeze Virtual</i>	1	512	5000
<i>Oracle 10g SQL Developer 10g</i>	-	1024	3000
<i>Servidor de Mapas</i>	1	1024	110
<i>VMware Player</i>	1	1024	200
<i>VMware Player</i>	2	1024	150
Estimación MÍNIMA	4	4096	100000

Fuente: Tabla de Autoría propia (Arévalo, D. y Guzmán, D.).

3. DISEÑO DEL PROTOTIPO

3.1 INTRODUCCIÓN

Después de haber analizado la problemática de la situación actual de la SUPERTEL, se tiene en el presente capítulo la presentación del prototipo de software diseñado bajo los procesos estandarizados en la entidad auspiciante. Con la recopilación y análisis de requerimientos, realizados en el capítulo 2, se presentará los diferentes documentos realizados acerca del modelo de software desarrollado. Esto implica, detallar el marco de trabajo utilizado para estructurar, planificar y controlar el proceso de desarrollo en este sistema de información geográfica.

En el análisis y diseño se ha puesto énfasis en la solución conceptual que satisface los requisitos detallados, basados en el proceso unificado. Apoyados en un análisis y diseño orientado a objetos.

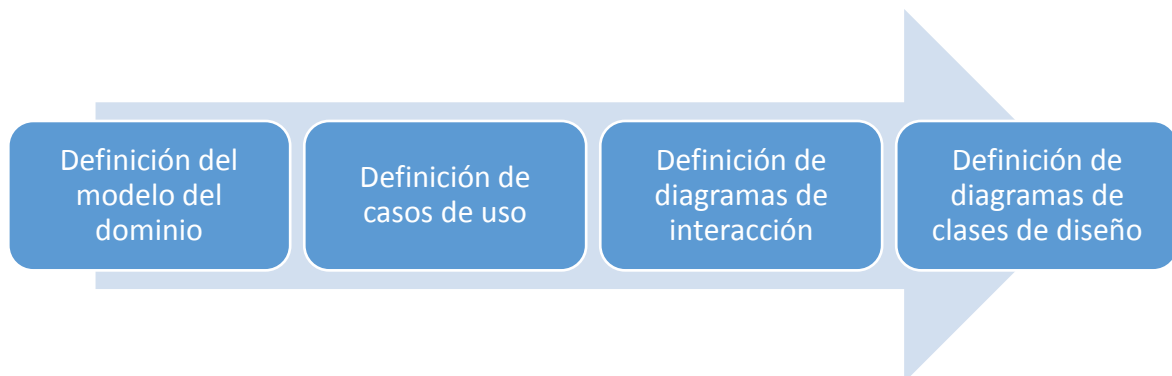


Figura 3.1 Proceso Unificado de Desarrollo

Fuente: Adaptado de: Rumbaugh, J., Jacobson, I., & Booch, G. (2000). *EL Lenguaje Unificado de Modelado. Manual de Referencia.* (pp. 21-34). Madrid: Pearson Education.

El proceso unificado de desarrollo de software describe un enfoque para la construcción, desarrollo y mantenimiento del software desarrollado. El proceso unificado fomenta las buenas prácticas del desarrollo de software, entre las cuales se destaca el desarrollo iterativo, mismo que hace referencia al diseño de software basado en pequeñas partes o iteraciones; teniendo así partes de un todo, las mismas que pueden llevar a cabo pruebas, integraciones, y ejecuciones.

Cada iteración incluye su propia fase de análisis de requisitos, diseño, implementación y pruebas.

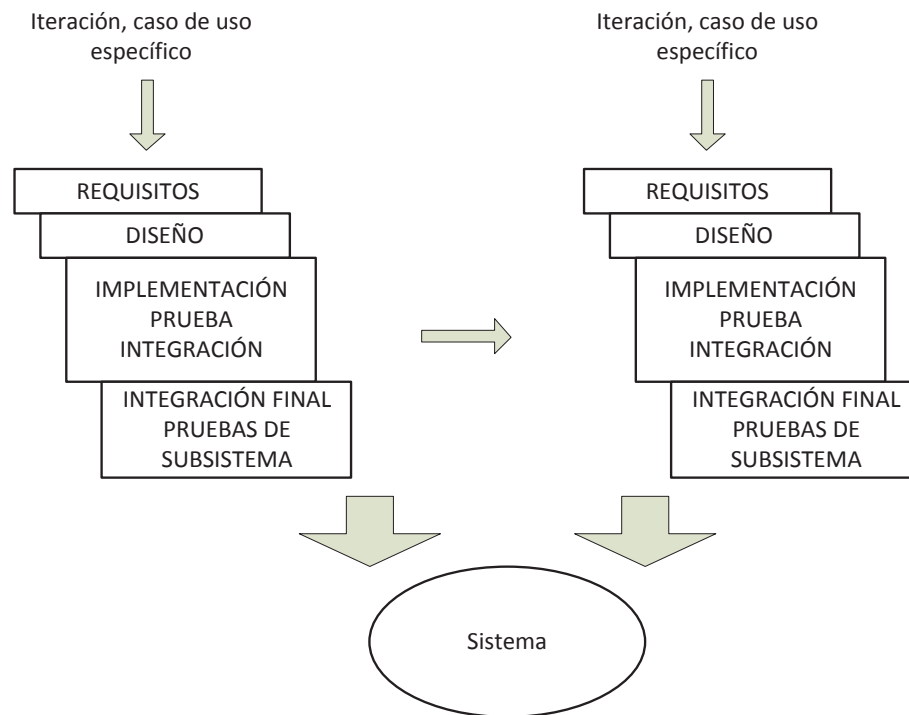


Figura 3.2 Desarrollo de Software Iterativo

Fuente: Adaptado de: Rumbaugh, J., Jacobson, I., & Booch, G. (2000). *EL Lenguaje Unificado de Modelado. Manual de Referencia.* (pp. 21-34). Madrid: Pearson Education.

Basado en el lenguaje UML (*Unified Modeling Language* – Lenguaje Unificado de Modelado), lenguaje gráfico que permite visualizar, detallar, construir y documentar un sistema ofreciendo un modelo estándar, se describirán aspectos conceptuales como procesos de negocios y operaciones del sistema, esquemas de bases de datos; así también, diagramas de clases y software reutilizable. [3]

Se ha tomado como referencia las definiciones de desarrollo del proceso conocido como UP (*Unified Process* – Proceso Unificado). Si bien la programación extrema XP (*Extreme Programming* – Programación Extrema) no hace énfasis en las fases que define UP en el desarrollo de software, tales como escritura de casos de uso y documentos de requisitos. Así también, el UP recomienda dedicar tiempo al diseño de modelos visuales antes de la programación; el caso contrario es recomendado en la programación extrema. Sin

embargo, para la descripción del análisis, diseño e implementación del prototipo se ha tomado modelos del Proceso Unificado debido a su facilidad de comprensión, y estandarización en los flujos de trabajo.

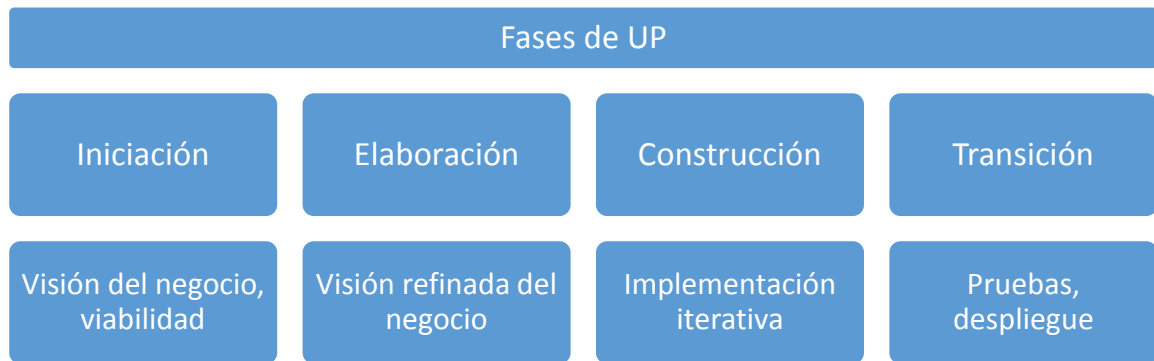


Figura 3.3 Fases de UP

Fuente: Adaptado de: Rumbaugh, J., Jacobson, I., & Booch, G. (2000). *EL Lenguaje Unificado de Modelado. Manual de Referencia*. (pp. 21-34). Madrid: Pearson Education.

El Proceso Unificado especifica actividades de trabajo, como la descripción de casos de uso, en disciplinas. Una disciplina es un conjunto de actividades en un área determinada, como las realizadas en el análisis de requerimientos. En el UP, un artefacto define cualquier resultado del trabajo como por ejemplo: código, gráficos, esquemas, diagramas, etc. Para el desarrollo del presente proyecto y descripción de este capítulo, se ha hecho énfasis en los artefactos de las siguientes disciplinas (Figura 3.4):



Figura 3.4 Disciplinas de UP Utilizadas

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

Teniéndose sin embargo otras disciplinas definidas en UP. El proceso UP, no obliga a desarrollar todas las disciplinas especificadas debido a que este proceso es adaptable a la realidad de cada proyecto. Por lo que se ha elegido ciertos artefactos del UP para la descripción del capítulo, describiéndose éstos en un documento conocido como Marco de Desarrollo.

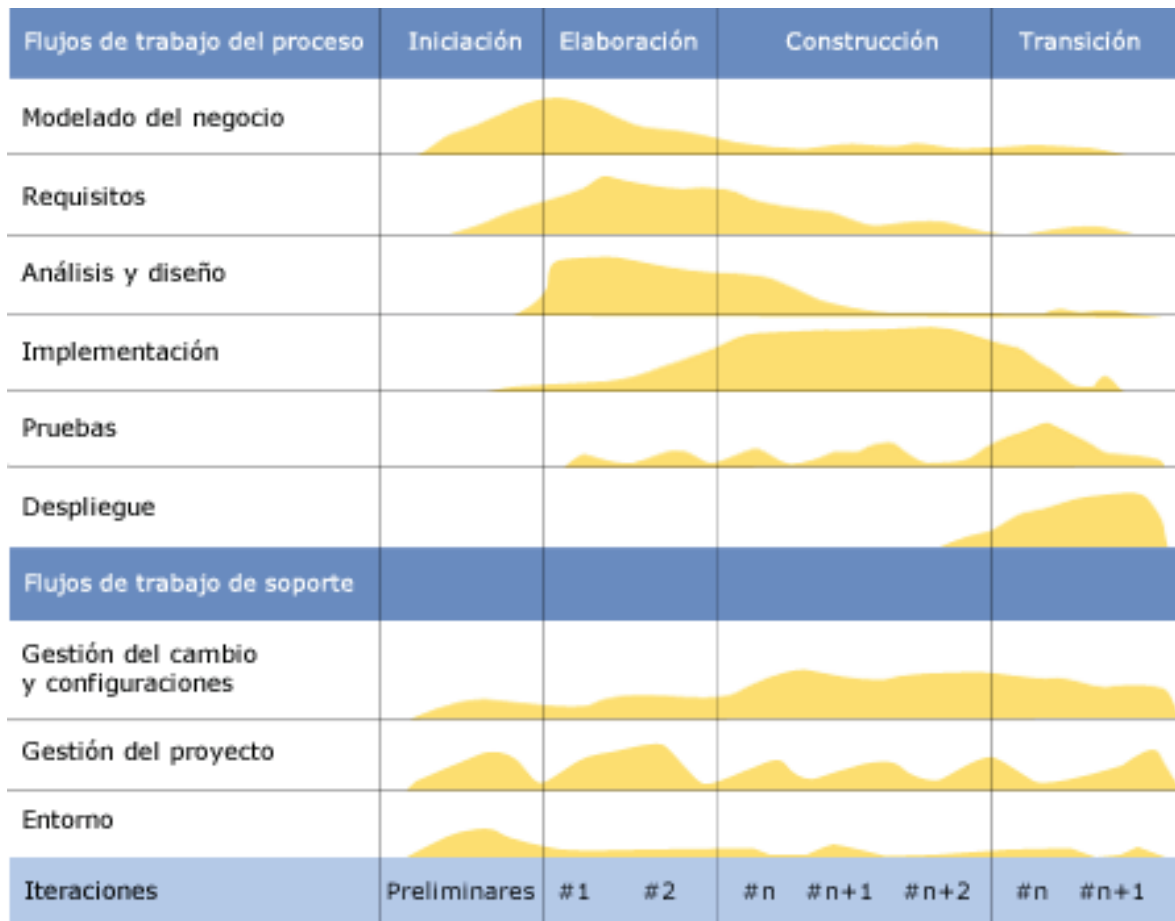


Figura 3.5 Disciplinas de UP

Fuente: Adaptado de, Proceso único de desarrollo de software. Recuperado de: <http://ingjesussoft.blogspot.com/2012/09/rup-proceso-unico-de-desarrollo-de.html>. (Noviembre 2012).

3.2 DISEÑO DEL PROTOTIPO

3.2.1 MARCO DE DESARROLLO

El marco de desarrollo es una descripción de los pasos UP y los artefactos adaptados para este proyecto. Definiendo:

- *Modelo de dominio.*- representación visual que muestra clases conceptuales dentro de un entorno de negocio.

- *Modelo de casos de uso.*- descripción de los requisitos funcionales y no funcionales relacionados, que describe a los actores utilizando un sistema para satisfacer cierto propósito.
- *Especificación complementaria.*- describe otros requisitos.
- *Modelo de diseño.*- modelamiento de objetos y su interacción.
- *Documento de arquitectura de SW.*- representación de estructuras de capas del software.
- *Modelo de datos.*- representación de objetos como tablas.

Tabla 3.1

Marco de Desarrollo de Artefactos UP. c-comenzar; r-refinar

Disciplina	Artefacto	Inicio	Elabo	Constr	Trans
		(I) I1	(E). E1..En	(C). C1..Cn	(T). T1..Tn
Modelado del Negocio	Modelo del Dominio		C		
	Modelo de Casos de Uso	C	R		
Requisitos	Especificación Complementaria	C	R		
	Modelo de Diseño		C	r	
	Documento de Arquitectura de SW		C		
Diseño	Modelo de Datos		C	r	

Fuente: Fuente: Adaptado de: Rumbaugh, J., Jacobson, I., & Booch, G. (2000). *EL Lenguaje Unificado de Modelado. Manual de Referencia.* (pp. 21-34). Madrid: Pearson Education.

3.2.2 MODELO DE DOMINIO

Un modelo del dominio es una representación de las clases conceptuales del mundo real, no de componentes software. No se trata de un conjunto de diagramas que describen clases software.[3]

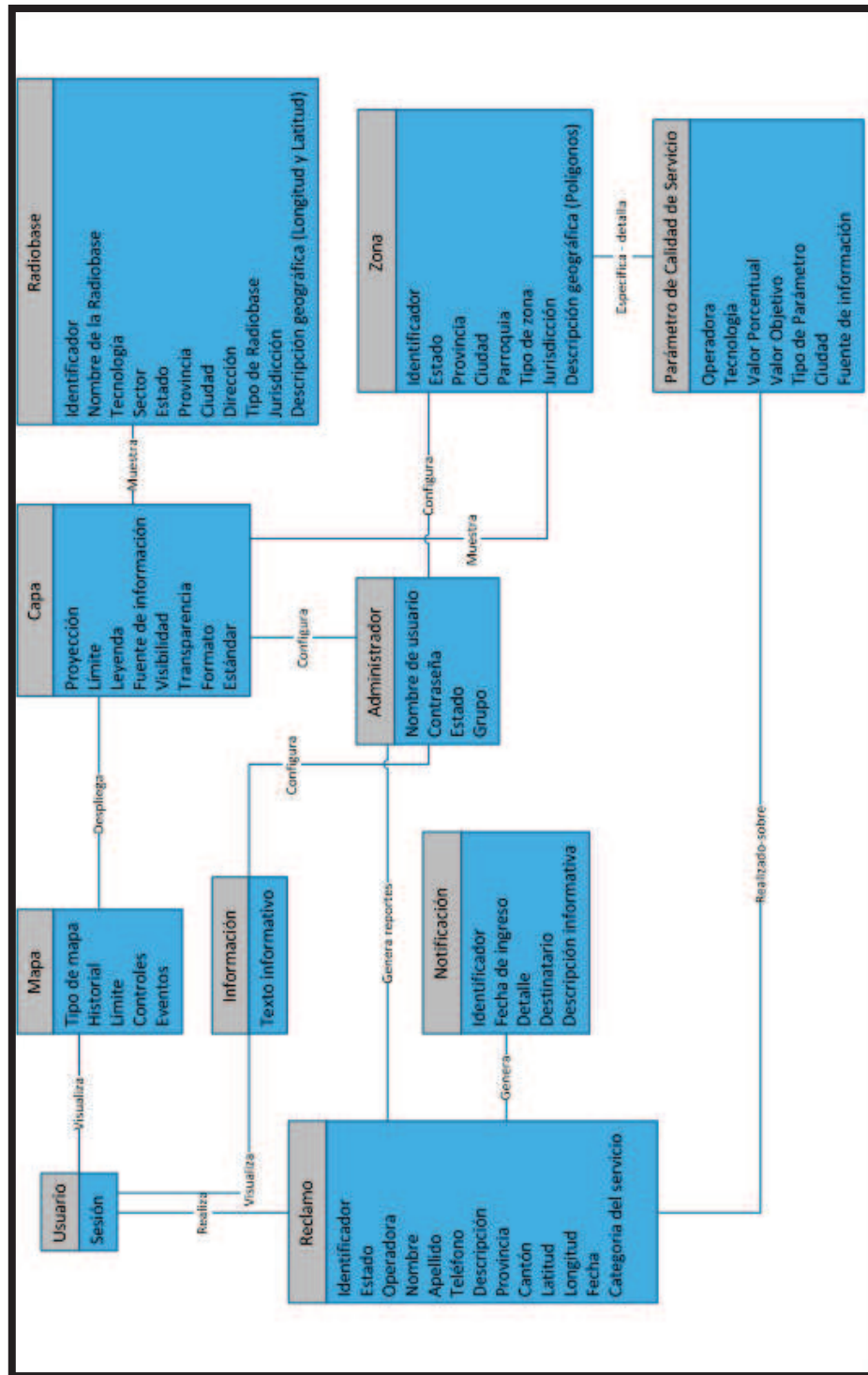


Figura 3.6 Modelo de Dominio Diseño del Prototipo

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

3.2.3 MODELOS DE CASOS DE USO

3.2.3.1 Diagrama de Casos de Uso

En las figuras 3.7 y 3.8 se describen los actores junto con sus casos de uso para el sistema prototipo, donde se han identificado diferentes actores en la interacción del sistema prototipo diseñado.

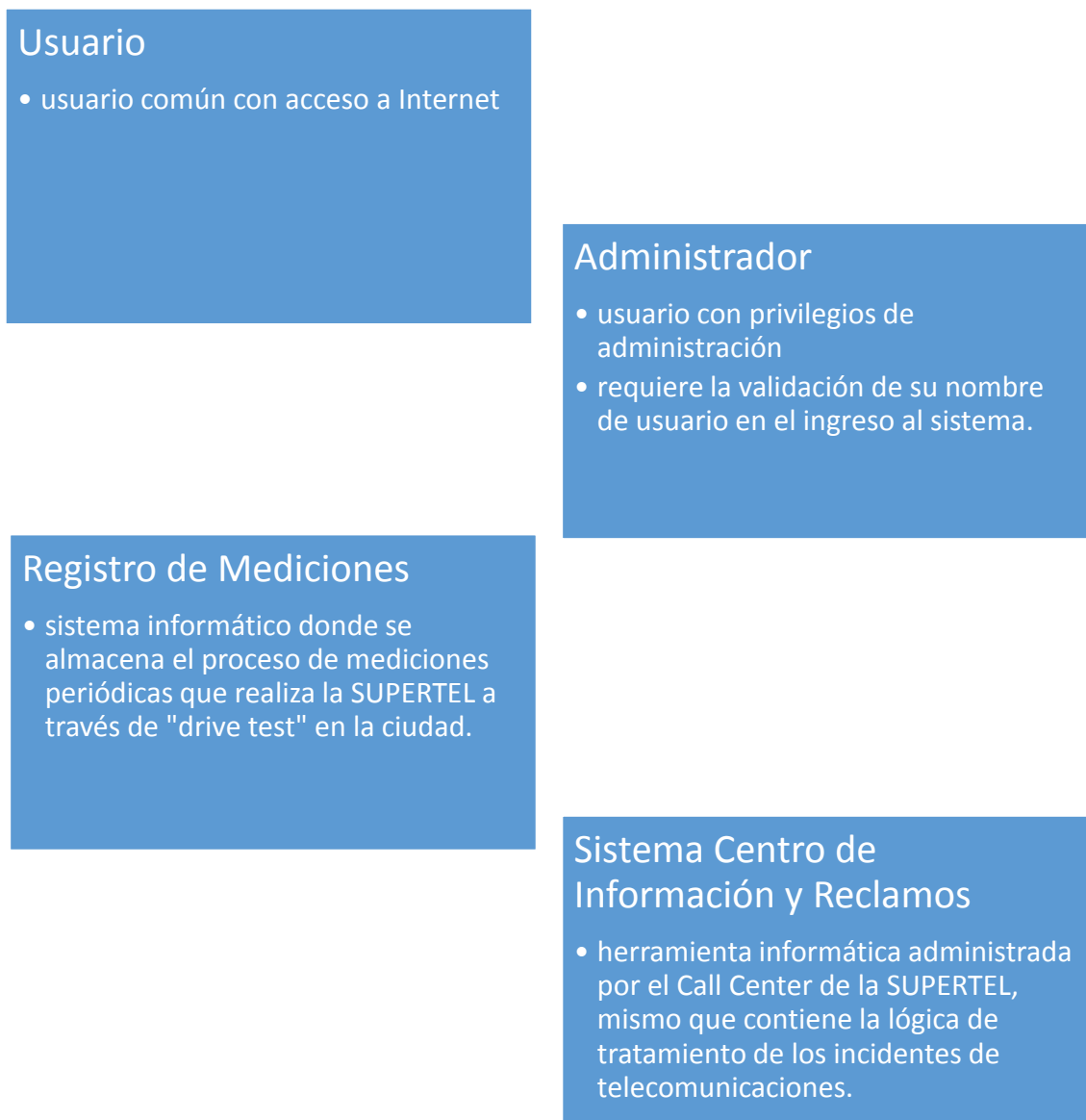


Figura 3.7 Descripción Actores Sistema Prototipo

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

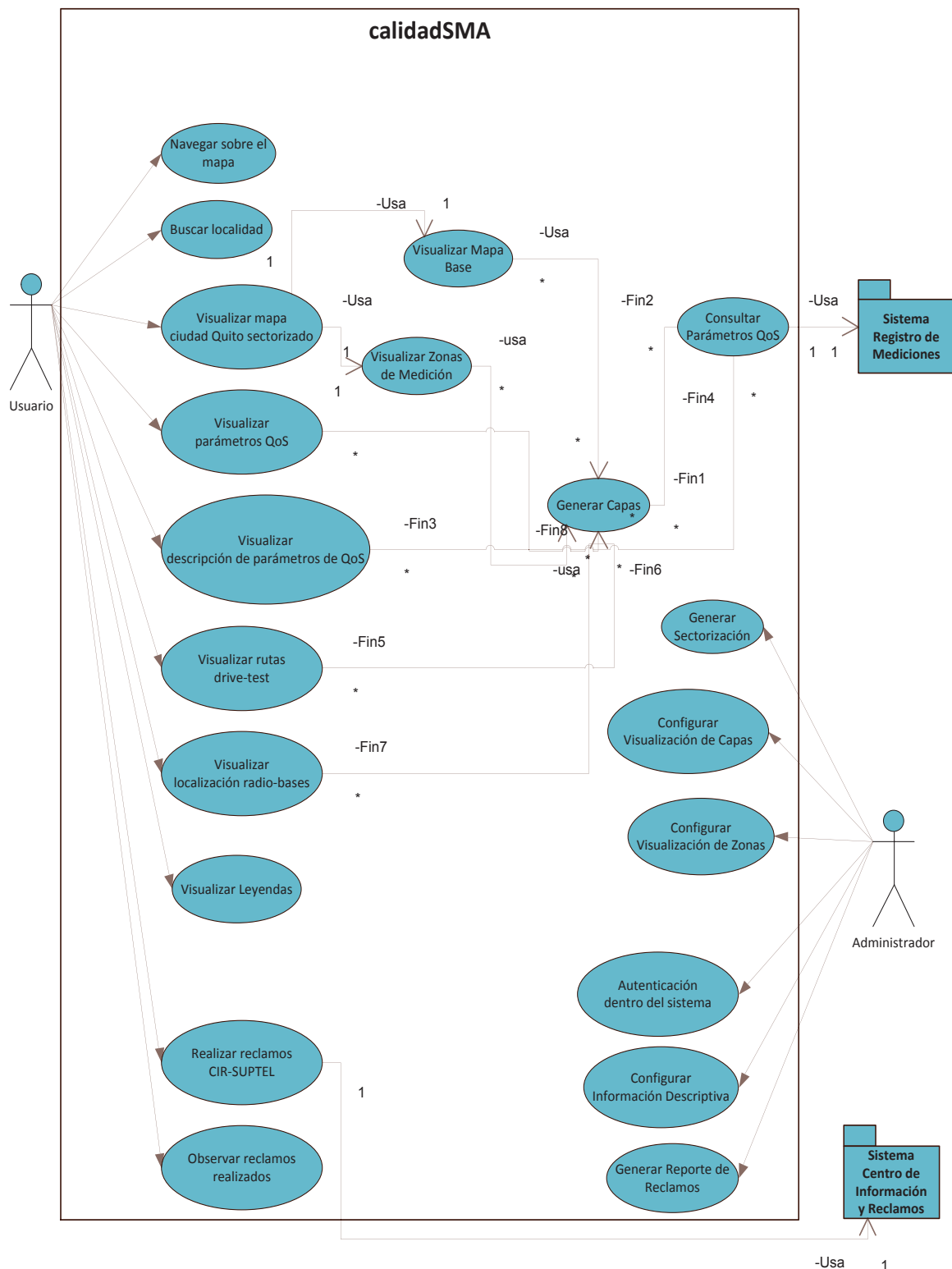


Figura 3.8 Diagrama de Casos de Uso

Fuente: Esquema de Autoría propia (Arévalo D., y Guzmán D.).

3.2.3.2 Casos de Uso

A continuación se describen cada uno de los casos de uso determinados para la aplicación.

3.2.3.2.1 CdU:01 Navegar Sobre El Mapa

El usuario que accede a la aplicación web puede recorrer todo el mapa de la ciudad de Quito (¡**Error! La autoreferencia al marcador no es válida.**).

Tabla 3.2
CdU:01 Navegar Sobre El Mapa

CdU:01
<p><i>Actor/Actores:</i> Usuario</p> <p><i>Nombre:</i> Navegar Sobre el Mapa</p>
<p><i>Objetivo:</i> Navegar sobre el mapa de la ciudad de Quito mostrado en la aplicación.</p>
<p><i>Pre-condición:</i> Mapa base cargado en la aplicación.</p>
<p><i>Flujo:</i> 1.- El usuario accede a la aplicación web a través de la respectiva url. 2.- El usuario se desplaza a través del mapa mostrado en pantalla. 3.- El usuario puede ajustar niveles de zoom, escoger tipo de mapa, etc.</p>
<p><i>Post-condición:</i> No-existe.</p>
<p><i>Excepciones:</i> No-existe.</p>

Fuente: Tabla de Autoría propia (Arévalo D., y Guzmán D.).

3.2.3.2.2 CdU:02 Buscar Localidad

El usuario que accede a la aplicación, puede ingresar el nombre de una dirección, localidad o calle, y la aplicación ubicara la posición seleccionada en el mapa expuesto (Tabla 3.3).

Tabla 3.3
CdU:02 Buscar Localidad

CdU:02
<p><i>Actor/Actores:</i> Usuario</p> <p><i>Nombre:</i> Buscar localidad</p>
<p><i>Objetivo:</i> Encontrar una posición específica (localidad, calle, etc.) dentro del mapa de la ciudad de Quito.</p>
<p><i>Pre-condición:</i> Mapa base cargado en la aplicación.</p>
<p><i>Flujo:</i> 1.- El usuario se sitúa en la barra para búsqueda de direcciones. 2.- El usuario ingresa el nombre de la localidad, calle o dirección que desea ubicar. 3.- El sistema procesa la petición y presenta al usuario las posibles respuestas. 4.- El usuario escoge la respuesta que satisface su requerimiento. 5.- El sistema ubica en el mapa la localización de su respuesta.</p>
<p><i>Post-condición:</i> La posición solicitada por el usuario dentro del mapa.</p>
<p><i>Excepciones:</i> 3' El sistema no puede ubicar lo dirección solicitada. + mensaje de error "Imposible ubicar dirección".</p>

Fuente: Tabla de Autoría propia (Arévalo D., y Guzmán D.).

3.2.3.2.3 CdU:03 Generar Capas

El sistema generará capas de tipo vectoriales o ráster (dependiendo del caso), las capas se generarán a partir de peticiones al servidor de mapas MS4W. Las capas se generarán bajo demanda, es decir cuando el usuario las solicite, a excepción de la capa base del mapa de la ciudad de Quito (de tipo ráster) y la capa que contiene las zonas de Quito (de tipo vectorial). Se detalla en la Tabla 3.4.

Tabla 3.4
CdU:03 Generar Capas

CdU:03
<p><i>Actor/Actores:</i> Sistema prototipo</p> <p><i>Nombre:</i> Generar capas</p>
<p><i>Objetivo:</i> Generar capas para mostrarlas y presentarlas a los usuarios a través de la interfaz gráfica de la aplicación.</p>
<p><i>Pre-condición:</i> No existe.</p>
<p><i>Flujo:</i> 1.- El sistema realiza una petición vía HTTP-GET a un servidor de mapas (si se va a generar una capa de tipo vectorial al servidor MS4W, si es el caso de una capa ráster a un servidor externo de <i>GoogleMaps</i>). 2.- El servidor responde con una imagen de tipo vectorial o ráster.</p>
<p><i>Post-condición:</i> Imagen vectorial o ráster que representa una capa.</p>
<p><i>Excepciones:</i> 2' El servidor no envía ninguna respuesta + mensaje en pantalla "fuera de servicio".</p>

Fuente: Tabla de Autoría Propia (Arévalo D., y Guzmán D.).

3.2.3.2.4 CdU:04 Visualizar Mapa Base

El usuario que accede a la aplicación web, inicialmente observará la capa base, la cual estará compuesta por una imagen ráster de la ciudad de Quito (podrá ser una imagen de tipo mapa, foto satelital, híbrida). Se detalla en la Tabla 3.5.

3.2.3.2.5 CdU:05 Visualizar Zonas De Medición

El usuario que accede a la aplicación web, observará sobreponiéndose a la capa base, la capa vectorial de los sectores pertenecientes a la ciudad de Quito (Tabla 3.6).

Tabla 3.5
CdU:04 Visualizar Mapa Base

<p>CdU:04</p> <p><i>Actor/Actores:</i> Usuario</p> <p><i>Nombre:</i> Visualizar Mapa Base</p>
<p><i>Objetivo:</i> Observar en la interfaz gráfica de la aplicación web una imagen de tipo ráster del mapa de la ciudad Quito.</p>
<p><i>Pre-condiciones:</i> Caso de uso CdU:03. Generar capas.</p>
<p><i>Descripción:</i> 1.- El usuario accede a la aplicación web. 2.- El sistema genera la capa base (imagen ráster del mapa de la ciudad de Quito). 3.- El sistema carga la capa base en la aplicación.</p>
<p><i>Post-condiciones:</i> Imagen ráster del mapa de la ciudad de Quito cargado en la aplicación.</p>
<p><i>Excepciones:</i> Mismas de CdU:03. Generar capas.</p>

Fuente: Tabla de Autoría Propia (Arévalo D., y Guzmán D.).

3.2.3.2.6 CdU:06 Visualizar Mapa Ciudad De Quito Sectorizado

3.2.3.2.7 *El usuario que accede a la aplicación web, inicialmente observará las 2 capas sobrepuestas: La capa base de la ciudad de Quito y la capa vectorial de los sectores pertenecientes a la ciudad de Quito. Estas dos capas, estarán siempre presentes y visibles. No existirá ningún control que permita deshabilitarlas (CdU:09 Realizar Reclamos CIR-SUPTEL*

El usuario, si tiene alguna inconformidad, queja, duda, etc., acerca del SMA en la ciudad de Quito. Puede enviarla directamente al CIR, a través de la aplicación. Este módulo tiene interacción directa con el Sistema Centro de Información y Reclamos. Se utilizará un servicio web para realizar el proceso de envío de quejas o inquietudes al CIR (Fuente: Tabla de Autoría Propia (Arévalo D., y Guzmán D.).

Tabla 3.10).

Tabla 3.7).

3.2.3.2.8 CdU:07 Consultar Parámetros De Qos

El sistema consulta el valor para un determinado parámetro de QoS, enviando como parámetros, la operadora y la zona de donde desea recibir la información de la medición. Se detalla en la Tabla 3.8.

Tabla 3.6
CdU:05 Visualizar Zonas De Medición

CdU:05
<p><i>Actor/Actores:</i> Usuario</p> <p><i>Nombre:</i> Visualizar Zonas de Medición</p>
<p><i>Objetivo:</i> Observar en la interfaz gráfica de la aplicación web los sectores definidos por la SUPERTEL.</p>
<p><i>Pre-condición:</i> Caso de uso CdU:03 Generar capas.</p>
<p><i>Flujo:</i> 1.- El usuario accede a la aplicación web. 2.- El sistema genera la capa vectorial de las zonas de Quito. 3.- El sistema carga la capa vectorial en la aplicación.</p>
<p><i>Post-condiciones:</i> Imagen vectorial de las zonas de la ciudad de Quito definidas por la SUPERTEL.</p>
<p><i>Excepciones:</i> Mismas de CdU:03. Generar capas.</p>

Fuente: Tabla de Autoría Propia (Arévalo D., y Guzmán D.).

3.2.3.2.9 CdU:08 Visualizar Parámetros De QoS

El usuario que accede a la aplicación web, dispondrá de un menú desplegable, donde podrá elegir, el parámetro que desea visualizar y de que operadora. El parámetro elegido se mostrará como un área de un determinado color, dentro de cada una de las zonas. El color se determinará en función de si cumple o no con el valor mínimo establecido para dicho parámetro. Se detalla en la Tabla 3.9.

3.2.3.2.10 CdU:09 Realizar Reclamos CIR-SUPTEL

El usuario, si tiene alguna inconformidad, queja, duda, etc., acerca del SMA en la ciudad de Quito. Puede enviarla directamente al CIR, a través de la aplicación. Este módulo tiene interacción directa con el Sistema Centro de Información y Reclamos. Se utilizará un servicio web para realizar el proceso de envío de quejas o inquietudes al CIR (Fuente: Tabla de Autoría Propia (Arévalo D., y Guzmán D.).

Tabla 3.10).

Tabla 3.7
Visualizar Mapa Ciudad De Quito Sectorizado

CdU:06
<p><i>Actor/Actores:</i> Usuario</p> <p><i>Nombre:</i> Visualizar Mapa Ciudad de Quito Sectorizado</p>
<p><i>Objetivo:</i> Observar en la interfaz gráfica de la aplicación web una representación del mapa de Quito con todos los sectores definidos por la SUPERTEL.</p>
<p><i>Pre-condición:</i> Casos de uso cdU3, cdU4 y cdU5.</p>
<p><i>Flujo:</i> 1.- El usuario accede a la aplicación web a través de la respectiva url. 2.- Se observará la capa base (una imagen ráster de la ciudad de Quito, podrá ser de tipo mapa, foto satelital o híbrida). Y sobreponiéndose a esta capa, una imagen vectorial de los</p>

sectores pertenecientes a la ciudad de Quito.
<i>Post-condición:</i> Las dos capas bases en la pantalla principal de la aplicación.
<i>Excepciones:</i> Mismas de CdU:03. Generar capas.

Fuente: Tabla de Autoría Propia (Arévalo D., y Guzmán D.).

3.2.3.2.11 CdU:10 Observar Reclamos Realizados

El usuario, podrá agregar una capa en donde se consten, a manera de marcas, todos los reclamos realizados por los usuarios (los generados utilizando la aplicación calidadSMA). De esta manera el usuario podrá apreciar de forma visual en que sectores de la ciudad existe mayor grado de inconformidad con el servicio prestado por las operadoras del SMA (Fuente: Tabla de Autoría Propia (Arévalo D., y Guzmán D.)).

Tabla 3.11).

Tabla 3.8
CdU:07 Consultar Parámetros De QoS

CdU:07
<i>Actor/Actores:</i> Sistema prototipo
<i>Nombre:</i> Consultar Parámetros de QoS
<i>Objetivo:</i> Consultar los valores medidos de parámetros de QoS almacenados en el Sistema de Registro de Mediciones de la SUPERTEL.
<i>Pre-condición:</i> Operadora y parámetro seleccionado por parte del usuario. Caso de uso CdU:03 Generar Capas.

<p><i>Flujo:</i></p> <ol style="list-style-type: none"> 1.- El sistema recibe la petición del módulo Generar Capas. 2.- El sistema envía la petición hacia el Sistema de Registro de Mediciones de la SUPERTEL. 3.- El Sistema de Registro de Mediciones responde con el valor numérico correspondiente a la medición del parámetro. 4.- El sistema devuelve el valor numérico al módulo Generar Capas.
<p><i>Post-condición:</i></p> <p>Valor numérico que representa valor medido para un parámetro de QoS de una operadora.</p>
<p><i>Excepciones:</i></p> <ol style="list-style-type: none"> 3' El Sistema de Registro de Mediciones no tiene registro alguno para la solicitud realizada. <p>+ Se devuelve valor nulo (no cero).</p>

Fuente: Tabla de Autoría Propia (Arévalo D., y Guzmán D.).

3.2.3.2.12 CdU:11 Visualizar Rutas Drive-Test

El usuario podrá agregar una capa en donde se visualicen todos los recorridos realizados por el personal de la SUPERTEL durante los *drive-test*. Estos datos no estarán almacenados dentro de la base de datos. Provenirán directamente de los archivos facilitados por la SUPERTEL (Tabla 3.12).

Tabla 3.9
CdU:08 Visualizar Parámetros De QoS

CdU:08
<p><i>Actor/Actores:</i></p> <p>Usuario</p> <p><i>Nombre:</i></p> <p>Visualizar Parámetros de QoS</p>
<p><i>Objetivo:</i></p> <p>Apreciar si se están cumpliendo o no con los parámetros de QoS, del SMA, dentro de un determinado sector de la ciudad de Quito.</p>
<p><i>Pre-condición:</i></p> <p>Los casos de uso CdU06.</p>

<p><i>Flujo:</i></p> <ol style="list-style-type: none"> 1.- El usuario selecciona del menú desplegable la operadora. 2.- Del menú desplegado, el usuario escoge un parámetro a visualizar. 3.- Se realiza una solicitud al módulo Generar Capas. 4.- Éste a su vez solicita el valor del parámetro utilizando el CdU:07 Consultar parámetros de QoS. 5.- El sistema devuelve una capa con la representación del parámetro solicitado. 6.- Mediante el uso de colores, el usuario, fácilmente podrá distinguir si, en el/los sectores de su interés, se cumple o no con el mínimo valor establecido para el parámetro de QoS seleccionado.
<p><i>Post-condición:</i></p> <p>Las dos capas bases en la pantalla principal de la aplicación.</p> <p>Áreas coloreadas representando los valores medidos y procesados para un parámetro de QoS de una operadora en particular.</p>
<p><i>Excepciones:</i></p> <p>2' El usuario escoge más de un parámetro de QoS. + Se superpone el último parámetro de QoS escogido. Mismos de caso de uso CdU:03.</p>

Fuente: Tabla de Autoría Propia (Arévalo D., y Guzmán D.).

Tabla 3.10
CdU:09 Realizar Reclamos CIR-SUPERTEL

CdU:09
<p><i>Actor/Actores:</i></p> <p>Usuario</p> <p><i>Nombre:</i></p> <p>Realizar Reclamos CIR-SUPERTEL</p>
<p><i>Objetivo:</i></p> <p>Enviar una queja o inquietud, relacionado exclusivamente con el SMA, al CIR, utilizando el formulario establecido por la SUPTEL para dicho proceso.</p>
<p><i>Pre-condición:</i></p> <p>Las dos capas bases en la pantalla principal de la aplicación, junto con áreas coloreadas, representando los valores medidos</p>

<p>y procesados para un parámetro de QoS de una operadora en particular.</p> <p>La necesidad del usuario de manifestar su discrepancia con los resultados mostrados, en la aplicación, de los parámetros de QoS medidos.</p> <p>La necesidad del usuario de manifestar una queja frente al servicio que recibe por parte de una operadora.</p>
<p><i>Flujo:</i></p> <ol style="list-style-type: none"> 1.- El usuario realiza un evento <i>click</i> en un punto, donde desee reportar su queja o discrepancia. 2.- Inicia el servicio web para comunicarse con el CIR. 3.- Se despliega un formulario en pantalla para la recepción del reclamo. 4.- El usuario llena el formulario. 5.- El usuario envía sus datos junto con su reclamo al CIR. 6.- El reclamo queda registrado en una base de datos.
<p><i>Post-condición:</i></p> <p>Reclamo registrado en la base de datos del CIR.</p> <p>Reclamo registrado en la base de datos que maneja la aplicación.</p>
<p><i>Excepciones:</i></p> <ol style="list-style-type: none"> 1' El usuario realiza un evento <i>click</i> fuera del área de cobertura. <p>+ No se inicia el proceso.</p>

Fuente: Tabla de Autoría Propia (Arévalo D., y Guzmán D.).

Tabla 3.11
CdU:10 Observar Reclamos Realizados

CdU:10
<p><i>Actor/Actores:</i></p> <p>Usuario</p> <p><i>Nombre:</i></p> <p>Observar Reclamos Realizados</p>
<p><i>Objetivo:</i></p> <p>Ver los reclamos realizados por los usuarios y su ubicación geográfica.</p>
<p><i>Pre-condición:</i></p> <p>Las dos capas bases en la pantalla principal de la aplicación.</p> <p>Reclamos registrados en la base de datos que maneja la aplicación.</p>

<p><i>Flujo:</i></p> <p>1.- El usuario selecciona del menú desplegable la operadora. 2.- Del menú desplegado, el usuario escoge la opción para visualizar reclamos. 3.- Los reclamos se muestran como marcas/puntos a lo largo de todas las zonas de la ciudad de Quito (capa vectorial).</p>
<p><i>Post-condición:</i></p> <p>Las dos capas bases en la pantalla principal de la aplicación, junto con marcas/puntos, que representan los reclamos enviados al CIR para una operadora determinada.</p>
<p><i>Excepciones:</i></p> <p>2' El usuario escoge un parámetro de QoS y la capa de reclamos + Se muestran las dos capas sobrepuestas. 3' El servidor de mapas local no está activo. + Mensaje en pantalla "fuera de servicio". 3' El servidor de mapas de <i>Google Maps</i> no está disponible. + Mensaje en pantalla de "fuera de servicio".</p>

Fuente: Tabla de Autoría Propia (Arévalo D., y Guzmán D.).

3.2.3.2.13 CdU:12 Visualizar Localización De Radio Bases

El usuario dispondrá de un menú, del cual luego de escoger la operadora, podrá agregar una capa en donde se visualice la localización geográfica de todas las radio bases de las operadoras. El símbolo que representará dentro del mapa a la radio base, corresponderá al logo de su respectiva operadora. Estos datos estarán almacenados dentro de la base de datos. Se detalla en la Tabla 3.13.

Tabla 3.12
CdU:11 Visualizar Rutas Drive-Test

<p>CdU:11</p> <p><i>Actor/Actores:</i> Usuario</p> <p><i>Nombre:</i> Visualizar Rutas Drive-Test</p>
<p><i>Objetivo:</i> Observar en el mapa de la ciudad de Quito las rutas seguidas durante los drive-test realizados por la SUPERTEL.</p>
<p><i>Pre-condición:</i> Las dos capas bases en la pantalla principal de la aplicación. Archivo que contiene la capa de rutas en el directorio de la</p>

aplicación.
<p><i>Flujo:</i></p> <p>1.- El usuario selecciona del menú desplegable la opción ver rutas de <i>drive test</i>.</p> <p>2.- Se muestran las rutas seguidas a lo largo de todas las zonas de la ciudad de Quito (capa vectorial).</p>
<p><i>Post-condición:</i></p> <p>Las dos capas bases en la pantalla principal de la aplicación, junto el trazo de la ruta que se siguió durante el <i>drive test</i>.</p>
<p><i>Excepciones:</i></p> <p>3' El servidor de mapas local no está activo. + Mensaje en pantalla "fuera de servicio".</p> <p>3' El servidor de mapas de <i>Google Maps</i> no está disponible. + Mensaje en pantalla de "fuera de servicio".</p>

Fuente: Tabla de Autoría Propia (Arévalo D., y Guzmán D.).

3.2.3.2.14 CdU:13 Visualizar Descripción de Parámetros de QoS

El usuario dispondrá de un panel, en donde se resuma una breve descripción de que representa cada uno de los parámetros de QoS mostrados en la aplicación. El panel tendrá la opción de ocultarse cuando el usuario así lo requiera. La información aquí presentada, provendrá de los anexos definidos en los contratos suscritos entre la SENATEL y cada una de las operadoras. [17]

Tabla 3.13
CdU:12 Visualizar Localización de Radio Bases

CdU:12
<p><i>Actor/Actores:</i></p> <p>Usuario</p> <p><i>Nombre:</i></p> <p>Visualizar Localización de Radio Bases</p>
<p><i>Objetivo:</i></p> <p>Observar en el mapa de la ciudad de Quito la ubicación geográfica de las radio bases de cada operadora.</p>
<p><i>Pre-condición:</i></p> <p>Las dos capas bases en la pantalla principal de la aplicación.</p>

La base de datos con la localización geográfica de todas las radio bases registradas por la SUPERTEL.
<p><i>Flujo:</i></p> <p>1.- El usuario selecciona del menú desplegable la operadora. 2.- El usuario selecciona del menú desplegable la opción ver radio bases. 3.- Las radio bases se muestran como marcas/puntos a lo largo de todas las zonas de la ciudad de Quito (capa vectorial).</p>
<p><i>Post-condición:</i></p> <p>Las dos capas bases en la pantalla principal de la aplicación, junto con marcas/puntos, que representan las radio bases de una operadora determinada.</p>
<p><i>Excepciones:</i></p> <p>2' El usuario escoge un parámetro de QoS y la capa radio bases. + Se muestran las dos capas sobrepuestas. 3' El servidor de mapas local no está activo. + Mensaje en pantalla "fuera de servicio". 3' El servidor de mapas de <i>Google Maps</i> no está disponible. + Mensaje en pantalla de "fuera de servicio".</p>

Fuente: Tabla de Autoría Propia (Arévalo D., y Guzmán D.).

3.2.3.2.15 CdU:14 Visualizar Leyendas

El usuario dispondrá de un panel en donde se visualice la descripción de la leyenda usada en la/las capas que se están mostrando en pantalla. Se detalla en la Tabla 3.15.

Tabla 3.14
CdU:13 Visualizar Descripción de Parámetros de QoS

CdU:13
<p><i>Actor/Actores:</i> Usuario</p>
<p><i>Nombre:</i> Visualizar Descripción de Parámetros de QoS</p>
<p><i>Objetivo:</i> Entender que representa cada uno de los parámetros de QoS.</p>
<p><i>Pre-condición:</i> No existen.</p>

<p><i>Flujo:</i> 1.- El usuario accede a la aplicación web a través de la respectiva url. 2.- En la pantalla principal, dentro de un menú, se muestra un breve resumen de que representan cada uno de los parámetros de QoS.</p>
<p><i>Post-condición:</i> No existen.</p>
<p><i>Excepciones:</i> No existen.</p>

Fuente: Tabla de Autoría Propia (Arévalo D., y Guzmán D.).

3.2.3.2.16 CdU:15 Autenticación dentro del Sistema

El administrador ingresará a una pantalla de login donde presentará sus credenciales (nombre y contraseña). El sistema validará las credenciales presentadas, y garantizará o no el acceso al módulo de administración. Se detalla en la Tabla 3.16.

3.2.3.2.17 CdU:16 Generar Sectorización

El administrador definirá un valor en Km² para los sectores que desee generar, ejecutará una función que generará los sectores en forma de datos vectoriales. Además la función deberá almacenar los datos generados en una tabla de la base de datos. La capa, de haber sido generada de manera correcta deberá mostrarse en pantalla, sobreponiéndose a un mapa base de la ciudad de Quito (Tabla 3.17).

Tabla 3.15
CdU:14 Visualizar Leyendas

<p>CdU:14</p> <p><i>Actor/Actores:</i> Usuario</p> <p><i>Nombre:</i> Visualizar Leyendas</p>
<p><i>Objetivo:</i> Entender que representa cada uno de los íconos, colores, formas, etc. dentro de cada capa generada.</p>

<i>Pre-condiciones:</i> Alguna capa mostrada en pantalla.
<i>Flujo:</i> 1.- El usuario accede a la aplicación. 2.- Al sistema genera las capas base. 3.- En un panel se muestra la leyenda para la capa actual. 4.- Si el usuario solicita otra capa, el sistema actualiza automáticamente la leyenda para la capa solicitada.
<i>Post-condiciones:</i> Panel que contiene la leyenda de la capa que se está mostrando en pantalla.
<i>Excepciones:</i> <i>No existen.</i>

Fuente: Tabla de Autoría Propia (Arévalo D., y Guzmán D.).

3.2.3.2.18 CdU:17 Configurar Visualización de Capas

El administrador definirá qué capas se van a visualizar dentro de la aplicación. Este parámetro dependerá de la columna estado definido para esta zona en la base de datos.⁴¹ Si la información que contiene la capa está desactualizada u obsoleta no debe mostrarse (Tabla 3.18). Como por ejemplo cuando el *Proceso para el control de la calidad en el SMA* se encuentra en fase 3.

Tabla 3.16
CdU:15 Autenticación dentro del Sistema

CdU:15
<i>Actor/Actores:</i> Administrador
<i>Nombre:</i> Autenticación dentro del Sistema.
<i>Objetivo:</i> Autenticarse como administrador dentro del sistema.

⁴¹ Este campo junto con la descripción de toda la tabla que contendrá las zonas se detallará en el punto 3.2.1. Modelo de diseño estático.

<p><i>Pre-condición:</i> Credenciales registradas en el servidor de autenticación SUPERTEL.</p>
<p><i>Flujo:</i> 1.- El administrador ingresa a una pantalla de <i>login</i>. 2.- EL administrador presenta sus credenciales (nombre y contraseña). 3.- El sistema valida las credenciales presentadas. 4.- Se garantiza o no el acceso al módulo de administración.</p>
<p><i>Post-condición:</i> Módulo de administración de la aplicación.</p>
<p><i>Excepciones:</i> 3' El administrador ingresa credenciales incorrectas. + Se muestra mensaje de error. + No se permite el acceso al módulo.</p>

Fuente: Tabla de Autoría Propia (Arévalo D., y Guzmán D.).

3.2.3.2.19 CdU:18 Configurar Visualización de Zonas

El administrador, definirá qué zonas se van a visualizar dentro de la aplicación. Este parámetro dependerá de la columna estado definido para esta zona en la base de datos. La columna GD_ESTADO hace referencia al estado (activo e inactivo), de la zona. Se detalla en la Tabla 3.19.

Tabla 3.17
CdU:16 Generar Sectorización

<p>CdU:16</p>
<p><i>Actor/Actores:</i> Administrador</p>
<p><i>Nombre:</i> Sectorización</p>
<p><i>Objetivo:</i> Generar una capa vectorial con sectores que tengan una</p>

determinada área.
<i>Pre-condición:</i> Módulo de administración de la aplicación.
<i>Flujo:</i> 1.- El administrador, definirá un valor en Km ² para los sectores que desee generar. 2.- Ejecutará la función que generará los sectores en forma de datos vectoriales (parámetro de entrada el área en Km ²). 3.- La función genera los sectores. 4.- La función almacena los datos generados en una tabla de la base de datos. 5.- La capa deberá mostrarse en pantalla, sobreponiéndose a un mapa base de la ciudad de Quito.
<i>Post-condición:</i> Mapa de la ciudad de Quito, que contenga los sectores generados. Polígonos que representan sectores, con un área específica y registrados dentro de la base de datos de la aplicación.
<i>Excepciones:</i> 3' Parámetro área ingresado mayor al área de cobertura definido. + Se muestra mensaje de error, especificando el valor máximo de área.

Fuente: Tabla de Autoría propia (Arévalo D., y Guzmán D.).

Tabla 3.18
CdU:17 Configurar Visualización de Capas

CdU:17
<i>Actor/Actores:</i> Administrador
<i>Nombre:</i> Configurar Visualización de Capas
<i>Objetivo:</i>

Determinar si una capa debe o no presentarse a los usuarios. ⁴²
<i>Pre-condición:</i> Módulo de administración de la aplicación.
<i>Flujo:</i> 1.- El administrador consulta con las tablas que contengan capas con información geográfica, la columna ESTADO. 2.- El administrador selecciona las capas con campo ESTADO igual a activo. 3.- El administrador pone a disposición de los usuarios las capas seleccionadas.
<i>Post-condición:</i> Capas en estado activo para ser mostradas a los usuarios.
<i>Excepciones:</i> No existen.

Fuente: Tabla de Autoría Propia (Arévalo D., y Guzmán D.).

3.2.3.2.20 CdU:19 Configurar Información Descriptiva

El administrador definirá el tipo de información descriptiva a mostrar a los usuarios y como mostrarla para que sea lo más asimilable y entendible para el usuario. La información descriptiva debe ser entendible por cualquier tipo de usuario (Tabla 3.20).

3.2.3.2.21 CdU:20 Generar Reportes

El administrador generará reportes cuando se detecten valores de parámetros de QoS bajo los umbrales establecidos por la SUPERTEL. Los reportes se presentarán en tablas (una por cada operadora), donde se muestre la zona y el/los parámetros que no están de acuerdo al contrato. Se detalla en la Tabla 3.21.

Tabla 3.19 **CdU:18 Configurar Visualización de Zonas**

⁴² Hecho que puede suscitarse por información desactualizada registrada en el sistema GTM.

CdU:18
<p><i>Actor/Actores:</i> Administrador</p> <p><i>Nombre:</i> Configurar Visualización de Zonas</p>
<p><i>Objetivo:</i> Determinar las zonas a mostrarse dentro de la aplicación.</p>
<p><i>Pre-condición:</i> Módulo de administración de la aplicación.</p>
<p><i>Flujo:</i> 1.- El administrador consulta con la tabla QoS_ZONAS_QUITO, la columna GD_ESTADO. 2.- El administrador selecciona las zonas con campo GD_ESTADO igual a activo. 3.- El administrador pone a disposición de los usuarios las zonas seleccionadas.</p>
<p><i>Post-condición:</i> Zonas de la ciudad de Quito en estado activo para ser mostradas a los usuarios.</p>
<p><i>Excepciones:</i> No existen.</p>

Fuente: Tabla de Autoría Propia (Arévalo D., y Guzmán D.).

Tabla 3.20
CdU:19 Configurar Información Descriptiva

CdU:19
<p><i>Actor/Actores:</i> Administrador</p> <p><i>Nombre:</i> Configurar información descriptiva.</p>
<p><i>Objetivo:</i> Definir la información descriptiva que se mostrará al usuario.</p>
<p><i>Pre-condición:</i></p>

Módulo de administración de la aplicación.
<p><i>Flujo:</i></p> <p>1.- El administrador analiza la información contenida en el anexo 5 de los contratos con las operadoras.</p> <p>2.- El administrador determina la información descriptiva a mostrar a los usuarios.</p>
<p><i>Post-condición:</i></p> <p>Leyendas e información de los parámetros de QoS que ayuden a entender a los usuarios, el proceso de monitoreo y control de los parámetros de QoS que realiza la SUPERTEL.</p>
<p><i>Excepciones:</i></p> <p>No existen.</p>

Fuente: Tabla de Autoría Propia (Arévalo D., y Guzmán D.).

Tabla 3.21
CdU:20 Generar Reportes

CdU:20
<p><i>Actor/Actores:</i></p> <p>Administrador</p>
<p><i>Nombre:</i></p> <p>Generar Reportes</p>
<p><i>Objetivo:</i></p> <p>Generar reportes por incumplimiento de parámetros de QoS.</p>
<p><i>Pre-condición:</i></p>

Módulo de administración de la aplicación.
<p><i>Flujo:</i></p> <ol style="list-style-type: none"> 1.- El administrador selecciona una operadora 2.- El administrador monitorea los parámetros de QoS a través de la aplicación. 3.- El administrador monitorea los reclamos y quejas por parte de los usuarios. 4.- El administrador genera un reporte de número de quejas por zona. 5.- El administrador genera un reporte de incumplimiento de parámetros registrando el valor medido frente al valor esperado en cada zona y por cada parámetro de QoS.
<p><i>Post-condición:</i></p> <p>Dos reportes por cada operadora, uno con la información de quejas por parte de los usuarios. Y otro donde se visualice el incumplimiento de parámetros en cada zona.</p>
<p><i>Excepciones:</i></p> <p>No existen.</p>

Fuente: Tabla de Autoría Propia (Arévalo D., y Guzmán D.).

3.2.3.3 Especificación Complementaria

Esta sección recoge los requerimientos específicos que no pudieron ser detallados en el análisis de casos de uso descrito anteriormente, en la Figura 3.9 se detallan todos los parámetros contemplados dentro de las especificaciones complementarias.

Especificación complementaria	Funcionalidad
	Registro y gestión de errores
	Seguridad
	Facilidad de uso
	Fiabilidad
	Rendimiento
	Soporte
	Restricciones de implementación
	Componentes adquiridos
	Componentes de libre distribución
	Cuestiones legales

Figura 3.9 Detalle de la Especificación Complementaria

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

3.2.3.4 Funcionalidad

- Aplicación web colaborativa.- Los reclamos de un usuario se reflejarán en tiempo real para los demás usuarios que ingresen al sistema.
- Mapa interactivo.- El mapa principal presentado como un visor trabaja con colores temáticos, haciendo la visualización de cumplimiento de valores contractuales de parámetros de QoS, de fácil comprensión.
- Notificaciones.- Un usuario que realiza un reclamo es inmediatamente notificado a través de correos electrónicos sobre el seguimiento que se está brindando a su requerimiento.
- Estadísticas.- A manera de reportes se presentará la base de registro de reclamos que realizan los usuarios de telefonía móvil avanzada.
- Versatilidad del visor en tomar diferentes fuentes de información.- El visor puede desplegar información geográfica almacenada en bases de datos, de servidores de mapas externos, como también de archivos

especializados en representar información geográfica como por ejemplo archivos de extensión KML⁴³ (*Keyhole Markup Language* – Lenguaje de Marcado *Keyhole*).

- Cantidad de usuarios.- 16'963.111 de usuarios de telefonía celular en Ecuador se espera que conozcan de la aplicación.

3.2.3.5 Registro y Gestión de Errores

El registro y gestión de errores está configurado a nivel de contenedor de la aplicación.



Figura 3.10 Registro de Acciones en la Aplicación

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

3.2.3.6 Seguridad

Para detallar la configuración de la seguridad en el prototipo se ha diferenciado dos módulos de la aplicación, el uno la interfaz del usuario común y el segundo la interfaz de un usuario administrador.

⁴³ Lenguaje de marcado fundamentado en XML para representar datos geográficos en tres dimensiones.

Tabla 3.22

Tabla de Descripción de Parámetros de Seguridad Considerados en el Prototipo

	Módulo de Administración	Módulo del Usuario Común
Confidencialidad	Debido a la naturaleza de la información no se implementará este concepto.	Debido a la naturaleza de la información no se implementa este concepto.
Autenticación	Los usuarios de este grupo requieren ingresar al sistema proporcionando credenciales.	No aplica.
Integridad	Se asegura a través de protocolos de transporte confiables, y validación de campos que la información del cliente llegue al servidor en su totalidad	Se asegura a través de protocolos de transporte confiables, y validación de campos que la información del cliente llegue al servidor en su totalidad
Disponibilidad	Requerimiento configurado a nivel de infraestructura.	Requerimiento configurado a nivel de infraestructura.
No rechazo	La aplicación no niega el acceso de clientes a los módulos programados a nivel de servidor.	La aplicación no niega el acceso de clientes a los módulos programados a nivel de servidor.

Fuente: Tabla de Autoría Propia (Arévalo D., y Guzmán D.).

3.2.3.7 Facilidad de Uso

Existen aspectos fundamentales, dentro del proceso de construcción de aplicaciones web, que favorecen en la comprensión y uso de las aplicaciones. Para el caso específico del presente proyecto, se describen en la Figura 3.11, todos los aspectos que se deben implementar para garantizar facilidad de uso por parte de los usuarios.

Interfaz web	Deberá ser soportada por los navegadores de mayor penetración en el mercado actual. Mozilla Firefox, Internet Explorer & Google Chrome
Conceptos técnicos de telefonía móvil	Serán abordados a través representaciones gráficas de fácil comprensión.
Navegación	Se utilizarán paneles de navegación; así como, control de eventos del ratón sobre el visor web.
Optimización	Se utilizará un módulo de almacenamiento caché a nivel de servidor de mapas para mejorar el rendimiento en procesamiento de mapas visualizados

Figura 3.11 Descripción de las Facilidades de Uso

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

3.2.3.8 Rendimiento

Se ha utilizado una arquitectura cliente servidor basada en controles JavaScript, donde los mismos son manejados desde el navegador del cliente. Pudiendo estos controles en el cliente realizar peticiones de requerimientos a servidores o bases de datos en la web o en la intranet.

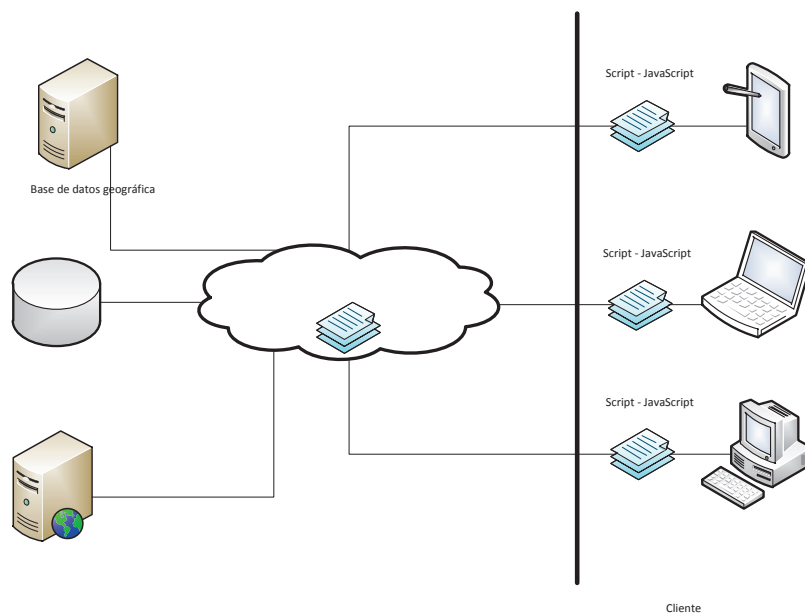


Figura 3.12 Funcionalidad de Scripts a Nivel de Cliente

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

3.2.3.9 Soporte

En la Figura 3.13, se describen los métodos de configuración de cada uno de los componentes que interactúan dentro de la aplicación a desarrollarse.

Configuración	Servidor de Mapas	La configuración estará basada en archivos de texto plano de extensión .map
	Servidor Caché de Mapas	La configuración estará basada en archivos de texto plano de extensión .yaml
	Contenedor de Aplicación	La configuración estará basada en archivos de texto plano.
	Aplicación	Cadenas de conexión y texto de etiquetas configurados en archivos de propiedades.

Figura 3.13 Soporte del Prototipo Diseñado

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

3.2.3.10 Restricciones de Implementación

La configuración de los servidores de mapas es independiente de la aplicación web, siendo la aplicación web solo un cliente del servicio WMS (*Web Map Service* – Servicio Web de Mapas). Siendo de ésta manera la administración de mapas independiente del prototipo desarrollado.

3.2.3.11 Componentes Adquiridos

Por estándar de desarrollo de la entidad auspiciante se ha trabajado con una base de datos propietaria ORACLE 11g con soporte para funcionalidades del tipo espacial (ORACLE SPATIAL 11g).

3.2.3.12 Cuestiones Legales

- Registro de Mediciones.- El sistema de registro de parámetros de calidad de servicio móvil avanzado es de propiedad intelectual de la SUPERTEL. Mismos datos que se utilizan para mostrar los mapas descriptivos de calidad de servicio.
- Uso de capas de *Google Maps*.- La licencia de publicación de mapas de *Google* a través del *framework OpenLayers* requiere un pago adicional.⁴⁴

3.2.3.13 Componentes de Libre Distribución

Dentro de los componentes necesarios para el desarrollo del proyecto se tienen: Lenguaje de programación, IDEs, *frameworks*, APIs, servidores, contenedores, etc. La Figura 3.14 detalla requerimientos junto con los componentes específicos.

3.2.4 MODELO DE DISEÑO DINÁMICO

Este modelo de diseño despliega la información de interacción de componentes que conforman el desarrollo de la aplicación. Todas las interacciones se diagraman en las figuras 3.15, Figura 3.16 y 3.17Figura 3.17.

⁴⁴ Condiciones de uso de servicio de Google Maps. http://maps.google.com/help/terms_maps.html. 14/11/12

Software Libre	Sistema operativo contenedor de la aplicación	Debian
	Lenguaje de programación	Java
	IDE de desarrollo	Eclipse
	Contenedor de la aplicación	Apache Tomcat
	Servidor de Mapas	MapServer
	Servidor Caché de Mapas	MapProxy
	API de desarrollo de controles de JavaScript para mapas	OpenLayers
	API de desarrollo de controles JavaScript	ExtJS
	Framework de desarrollo de modelo Vista-Controlador	SpringFramework
	Framework de desarrollo de persistencia	Hibernate
	Integrador de servicios web	JAX-WS

Figura 3.14 Detalle de Componentes de Libre Distribución

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

3.2.4.1 Diagramas de Secuencia

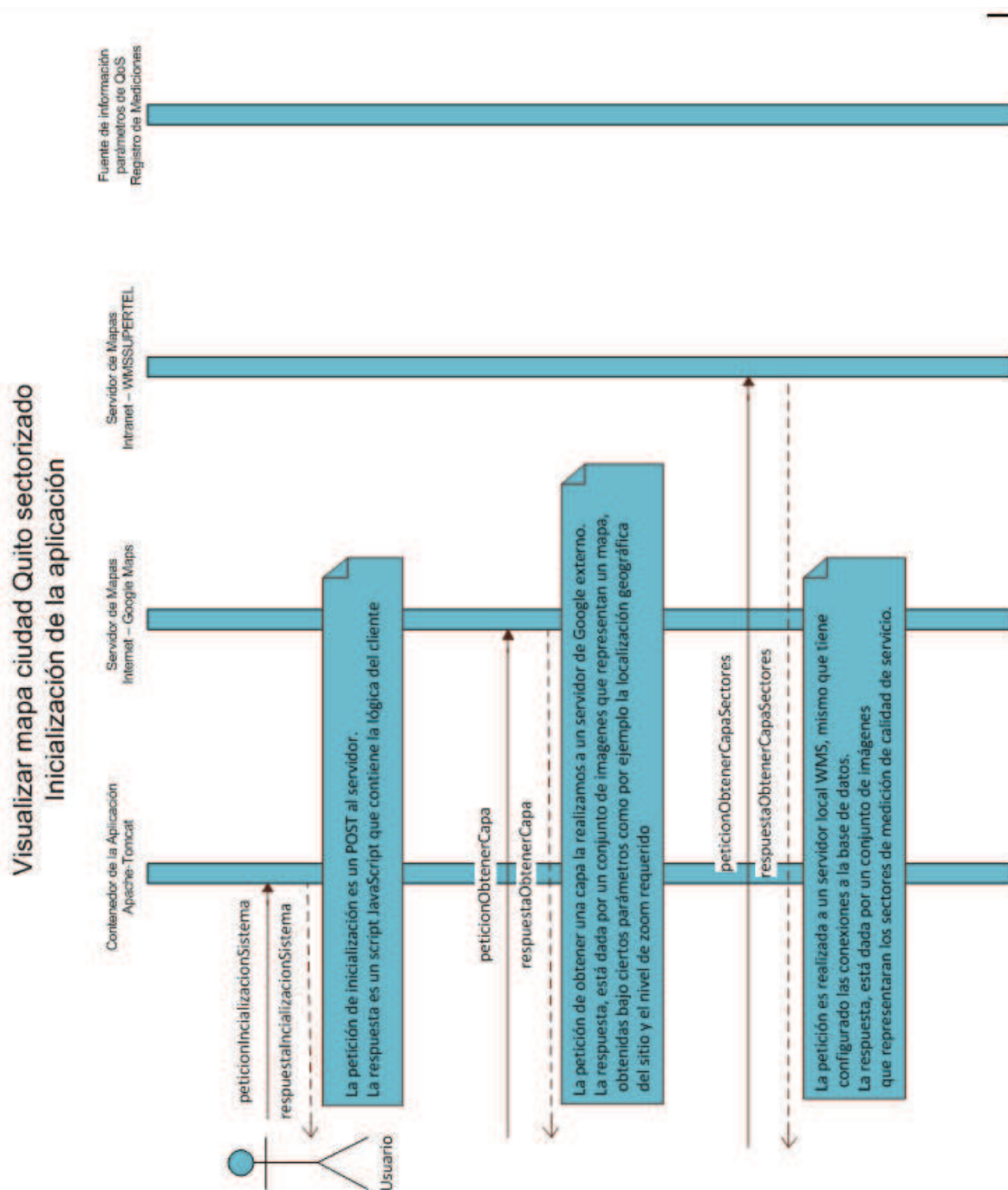


Figura 3.15 Diagrama de Secuencia del Sistema para la Inicialización de la Aplicación

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

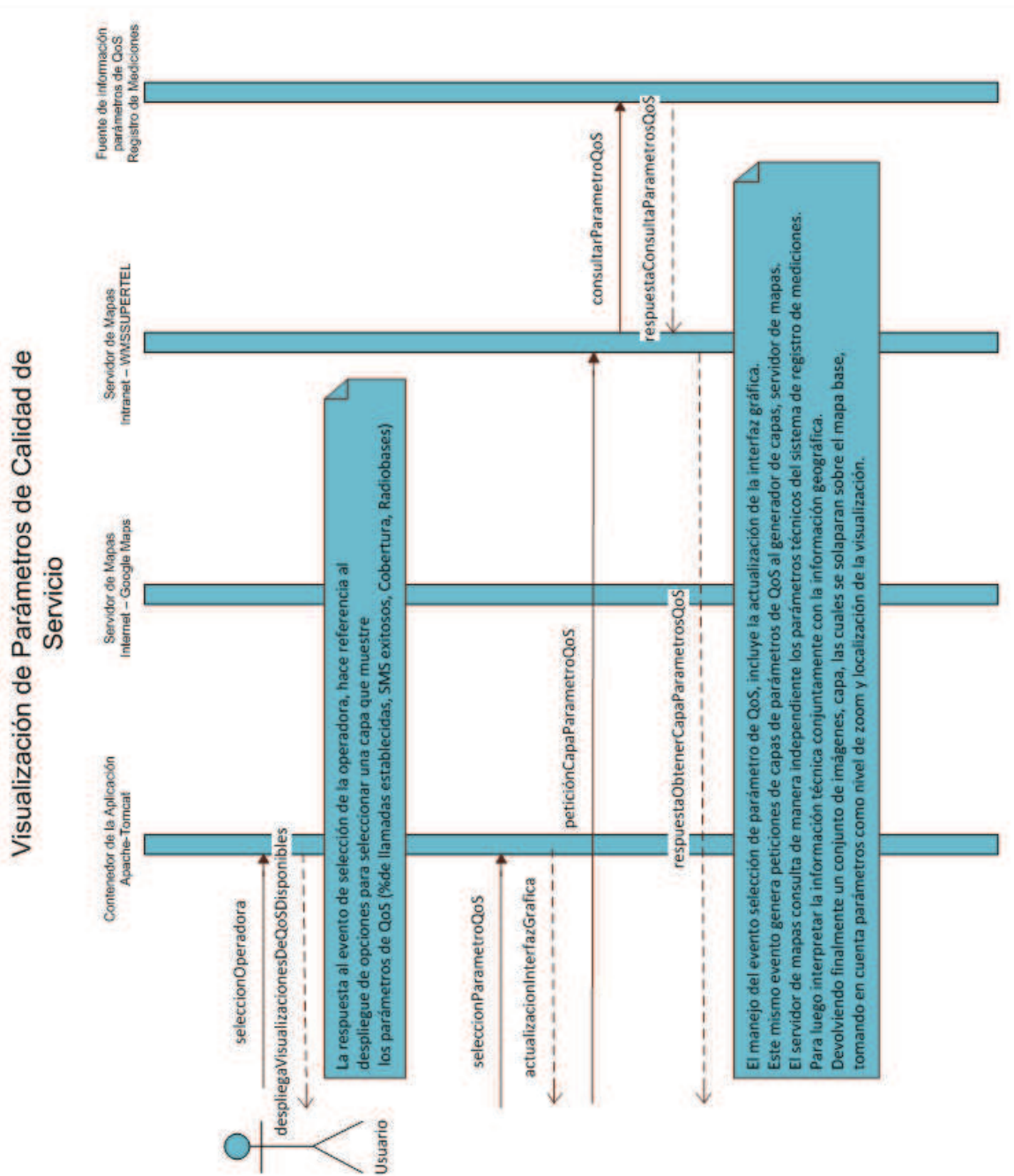


Figura 3.16 Diagrama de Secuencia del Sistema para la Visualización de Parámetros de QoS

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

Visualización de Detalle de Parámetros de Calidad de Servicio por Zona y Generación de Reclamos

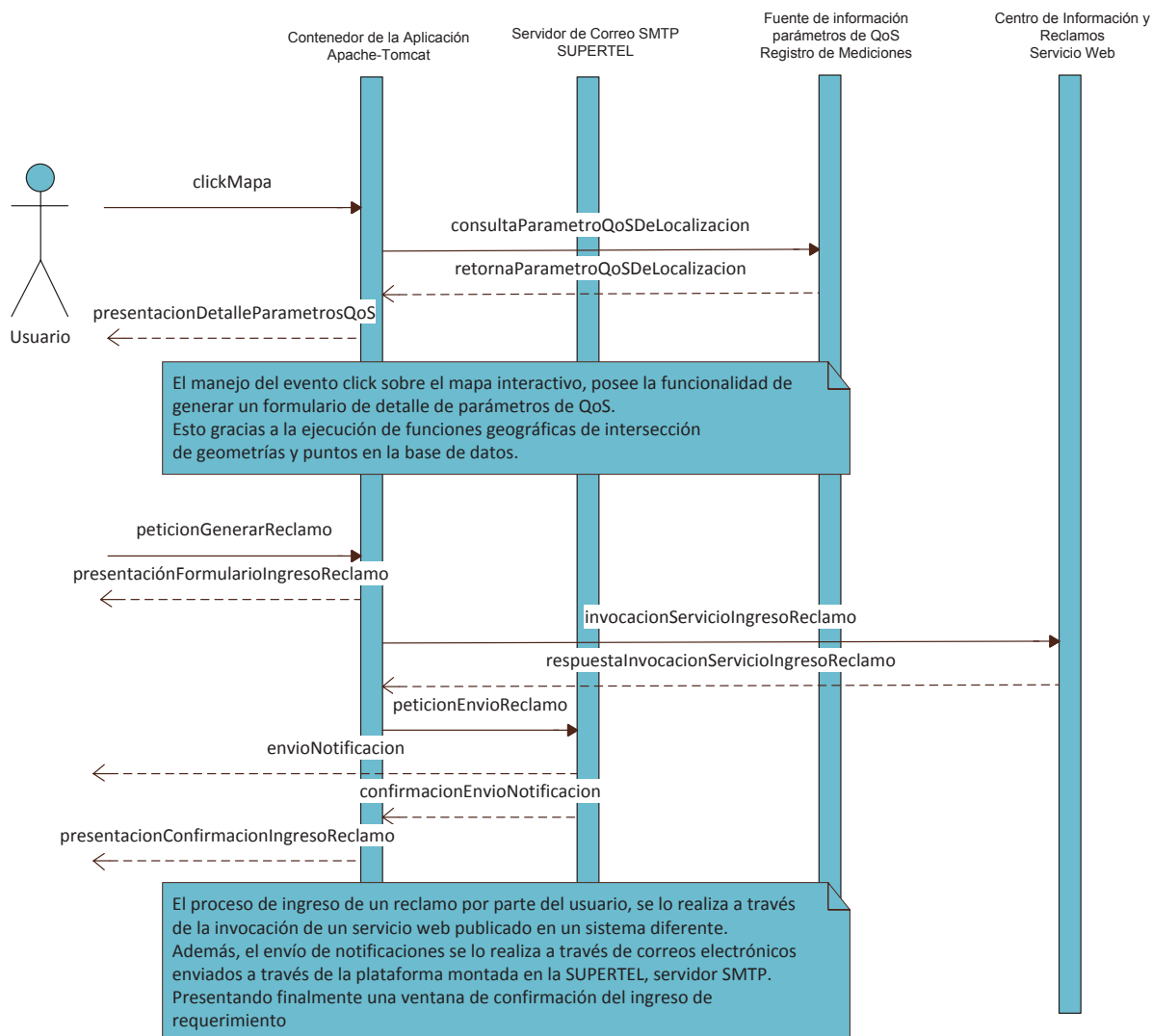


Figura 3.17 Visualización Detalle Parámetros de QoS e Ingreso Reclamos

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

3.2.5 MODELO DE DISEÑO ESTÁTICO

3.2.5.1 Diagrama de Clases

Debido a las necesidades de funcionalidad propias de la aplicación se ha considerado en el diseño, entidades que hacen referencia a tablas en la base de datos, estas entidades son representadas por las clases Reclamo, ParametroCalidad, Zona, Configuración. Las demás funcionalidades serán

desarrolladas mediante el uso clases propias de un *framework* o funcionalidades propias de las herramientas y servidores. La Figura 3.18 y Figura 3.19, describen nombres, atributos y métodos de cada una de las clases.

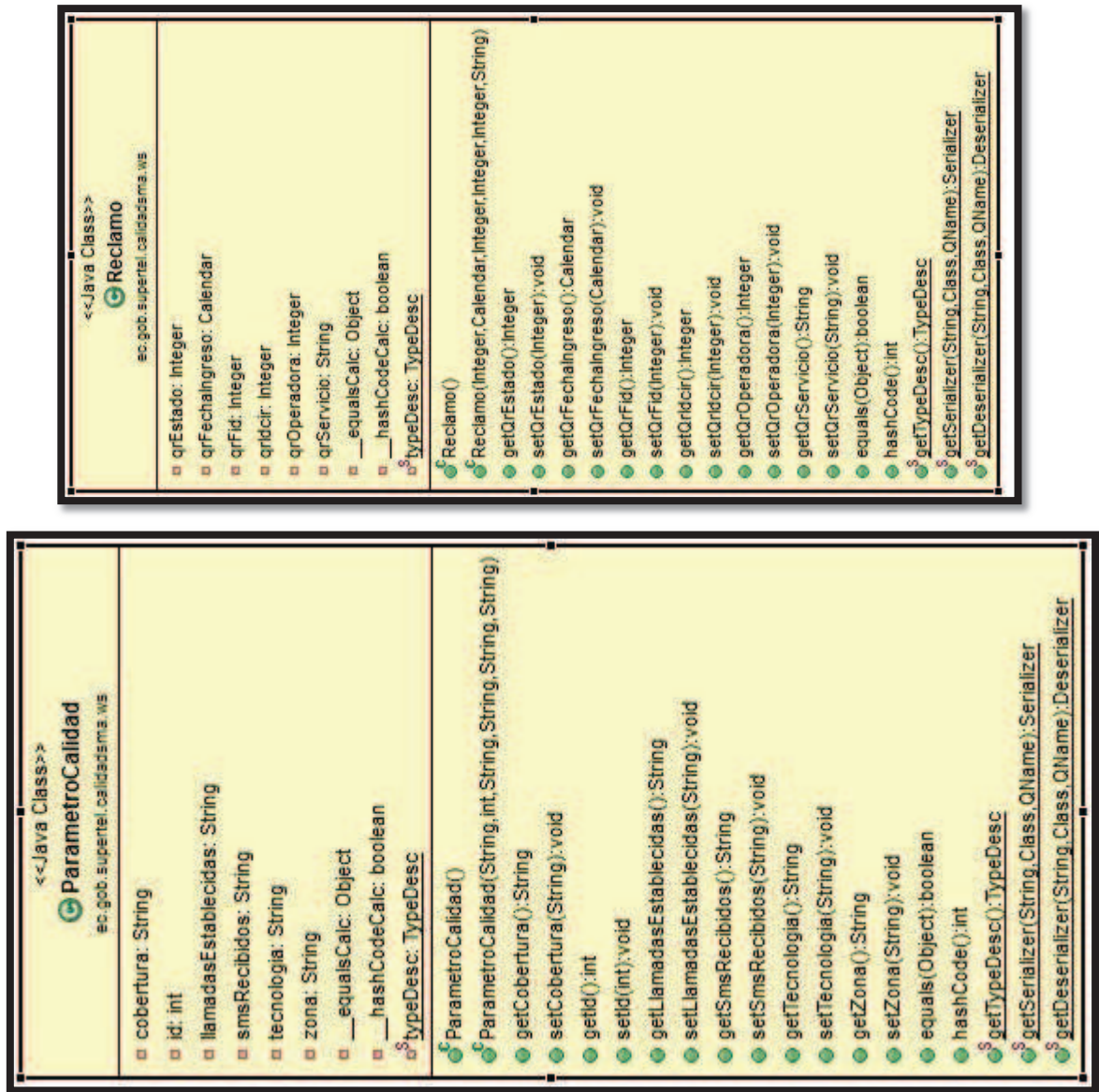


Figura 3.18 Clases del Sistema, referencia el Modelo Estático I

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

Basados en el patrón MVC, se definen los controladores para la aplicación web, esquematizados en las figuras 3.20 y 3.21.

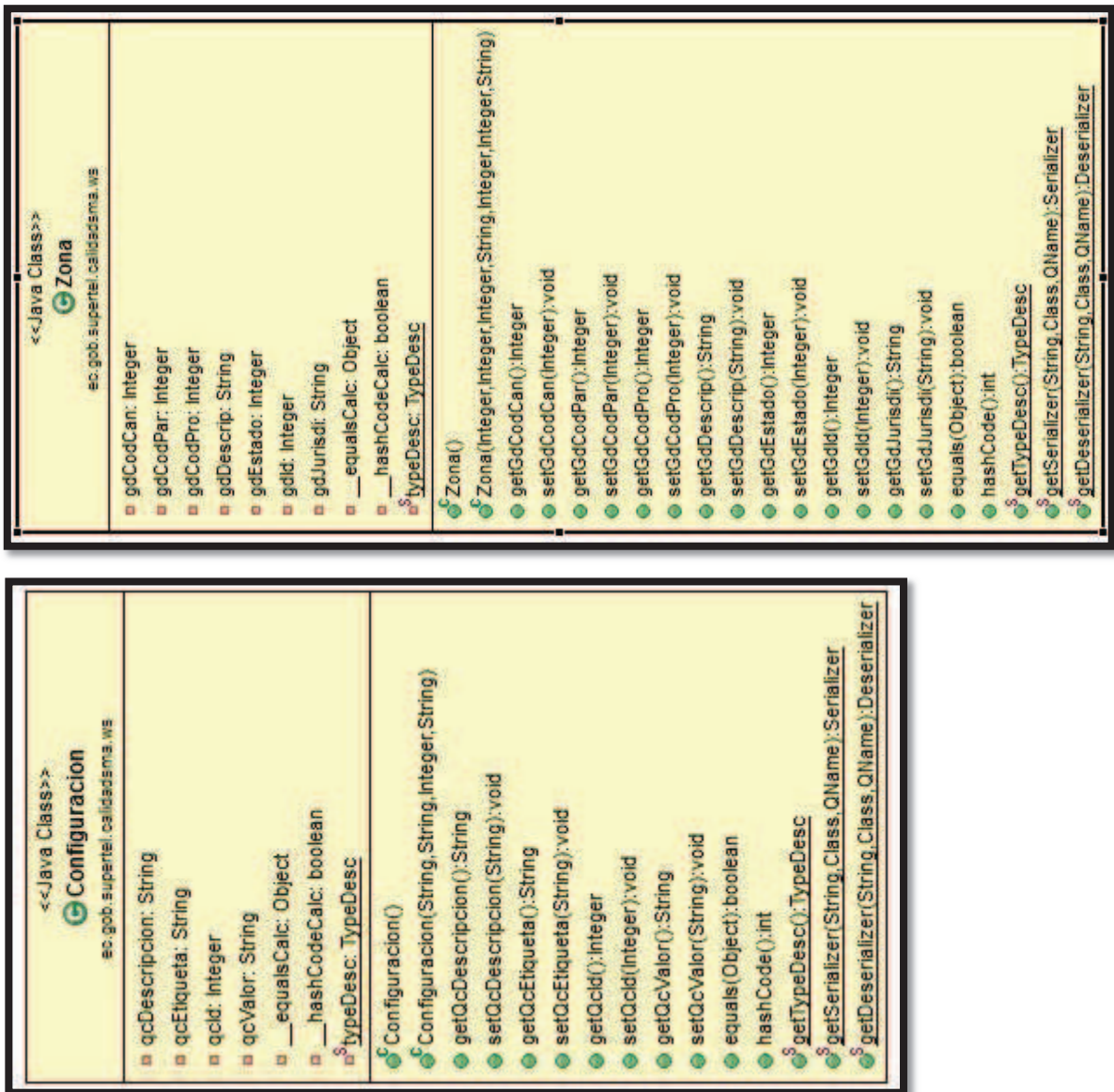


Figura 3.19 Clases del Sistema, referencia el Modelo Estático II

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

Para el proceso de envío de reclamos al CIR (consumiendo el servicio web proporcionado por la SUPERTEL), se definen las clases descritas en la Figura 3.22.

3.2.5.2 Diagramas Entidad Relación

El diagrama de la base de datos consta de algunos elementos, que comprenden tablas y vistas. A continuación se detalla todos los componentes de la base a

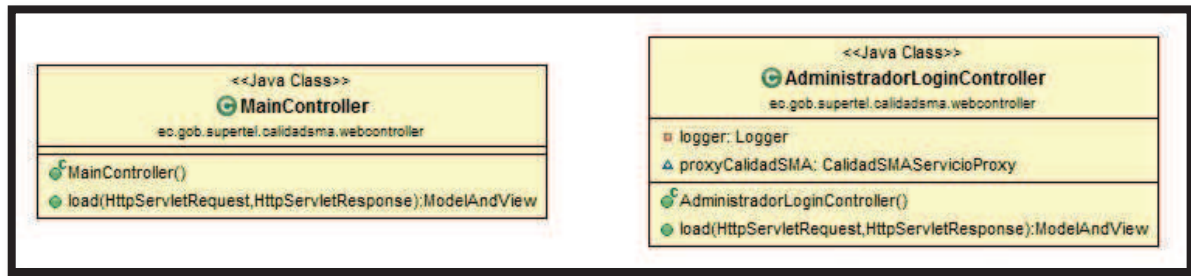


Figura 3.20 Controladores Web I

Fuente: Esquema de Autoría propia (Arévalo D., y Guzmán D.).

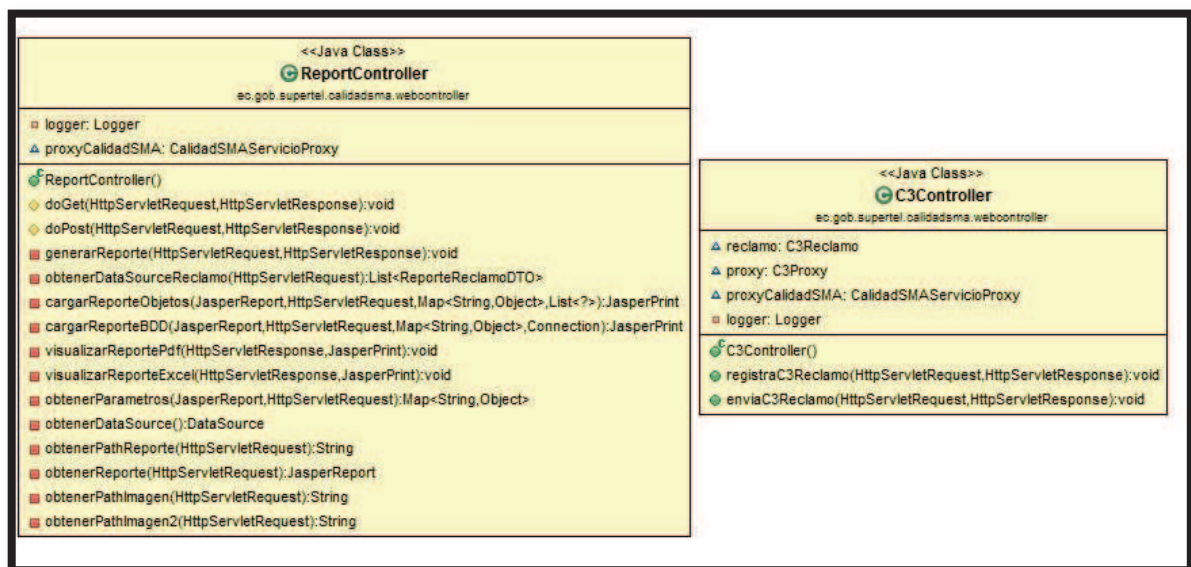


Figura 3.21 Controladores Web II

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

desarrollarse. La información proveniente de los archivos `zonas.shp`, provistos por la SUPERTEL, serán almacenados en un tabla llamada `QOS_ZONAS_QUITO`. La cual está estructurada como se muestra en la tabla de la Figura 3.23.

De la tabla, destaca la columna `GD_COORDENADAS`, la cual será el campo que contenga los registros de la información geográfica. Este campo será de tipo `MDSYS.SDO_GEOMETRY`. Este tipo de dato es el manejado por el motor de base de datos Oracle para almacenamiento de información geográfica. Esta columna deberá estar ligada a un índice espacial, el cual será denominado `QOS_ZONAS_IDX1`.

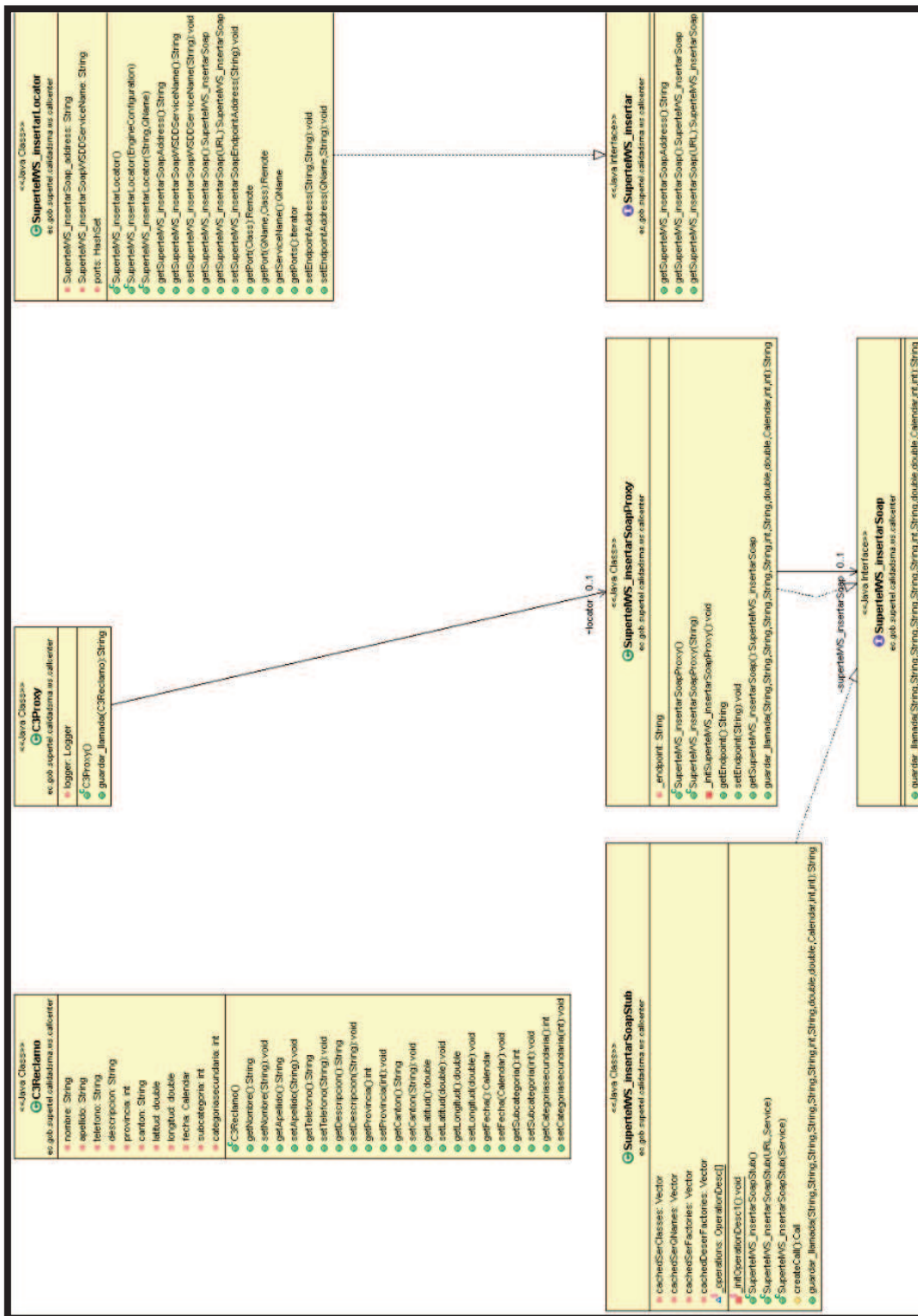


Figura 3.22 Envío de Reclamos CIR
Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

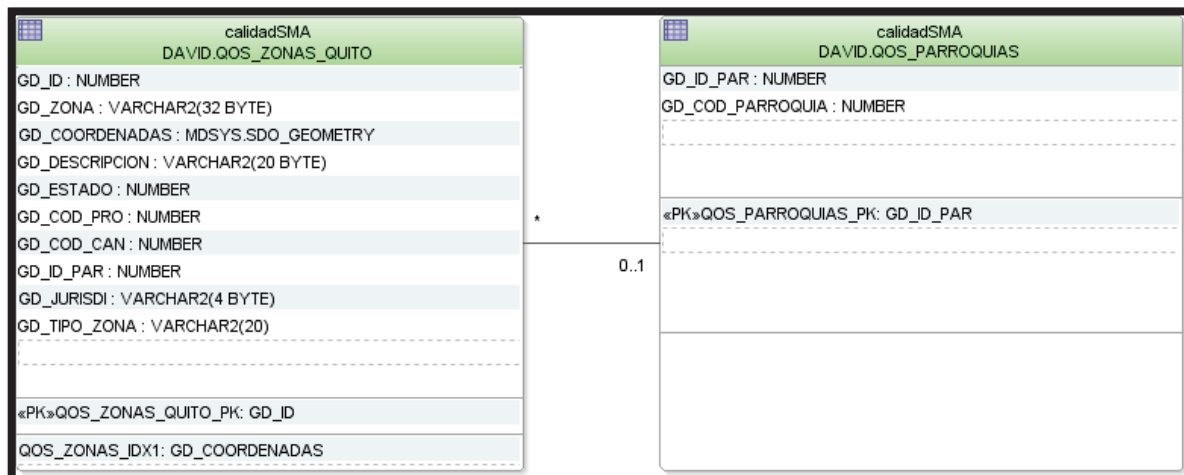


Figura 3.23 Tabla QOS_ZONAS_QUITO.

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

3.2.5.2.1 Desarrollo de Vistas

Para poder mostrar los parámetros de QoS correspondientes a cada una de las zonas de la ciudad de Quito, se van a crear tres vistas (1 por cada operadora). Estas vistas contendrán los datos de cada una de la zona, los cuales provienen de los registros de la tabla QOS_ZONAS_QUITO. Junto con los respectivos valores porcentuales (de acuerdo a cada zona) de cada uno de los parámetros de QoS, que se encuentran dentro del sistema GTM de la SUPERTEL en las vistas STM1MV, STM7MV Y STM9MV, descritas en el capítulo anterior. En la Figura 3.24, Figura 3.25 y Figura 3.26, se detallan las columnas de cada una de las vistas, así como las tablas o vistas de las cuales provendrán los datos.

➤ VISTA QOS_CONECEL



Figura 3.24 Diagrama Vista QOS_CONECEL

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

➤ VISTA QOS_OTECCEL

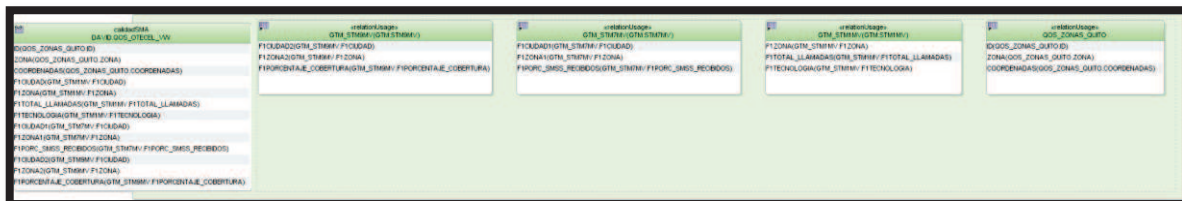


Figura 3.25 Diagrama Vista QOS_OTECCEL

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

➤ VISTA QOS_TELECSA

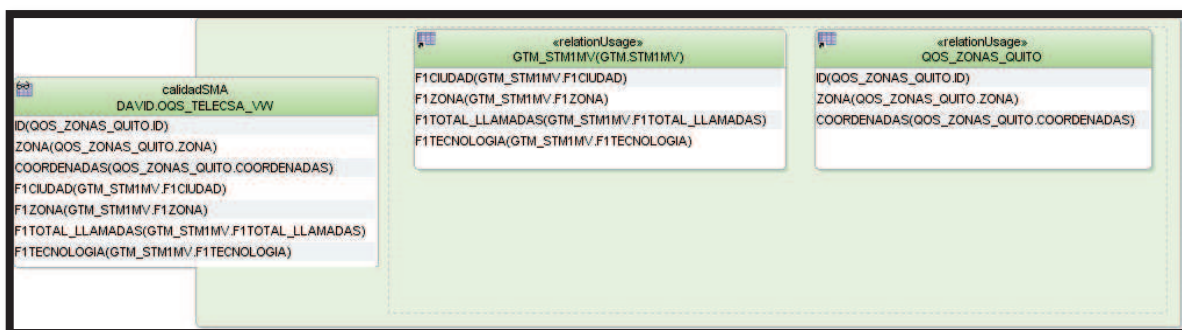


Figura 3.26 Diagrama Vista QOS_TELECSA

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

Es importante destacar que en la base de datos que maneja el sistema GTM se encuentran los datos de mediciones de todas las provincias y ciudades donde fueron realizados, por tanto en la fase de implementación se debe aplicar filtros por provincia y cantón o ciudad.

3.2.5.2.2 Módulo de Sectorización

El proceso de sectorización se lo lleva a cabo en el lado de la base de datos, debido a la potencialidad que ofrece el paquete *Oracle Spatial* en su versión 10g, para realizar operaciones entre objetos espaciales.

Basándonos en los estudios realizados por la SUPERTEL, en primera instancia se definirá una zona de cobertura del SMA, donde en teoría se debe contar con dicho servicio. La información de la zona de cobertura será almacenada en una tabla llamada QOS_COBERTURA_CIUDAD. Esta zona definirá el perímetro para la generación de las zonas. Dentro de este perímetro, y con ayuda del SIG de

escritorio Quantum GIS se diagramará un aproximado teórico de las cuadras que conforman la ciudad.

Para esto se asume que una cuadra es una figura geométrica cuadrada de lado 100 m y con un área equivalente a 10000 m². Una vez diagramadas las cuadras, se procederá a intersecarlas con la zona de cobertura, obteniendo como resultado el conjunto de cuadras de una ciudad que están contenidas dentro la zona de cobertura. Esta información será almacenada dentro de una tabla llamada QOS_CUADRAS_ZONA.

Para poder distinguir a qué provincia y cantón pertenecen cada una de las zonas. Se requieren dos tablas adicionales QOS_PROVINCIA Y QOS_CANTON. Estas tablas contendrán los códigos de cada provincia o cantón, asignados por el INEN, junto con su respectivo nombre. En la Figura 3.27, se describen las tablas y columnas a crearse, así como sus relaciones.

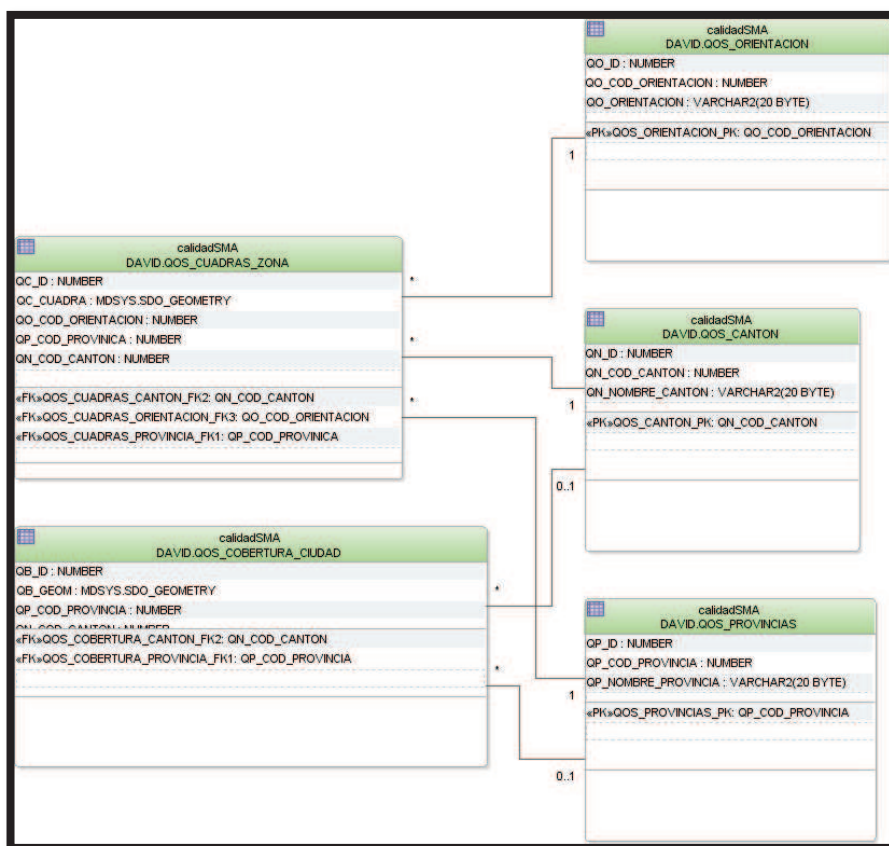


Figura 3.27 Modelo relacional Sectorización

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

Con los datos de las cuadras en la base de datos se procede al diseño de un procedimiento almacenado SP_SECTORIZACION que reciba como parámetros de entrada el área en Km² de la zona a generarse, y devuelva un número de zonas de área aproximada a la solicitada. La Figura 3.28 detalla el diagrama de flujo a seguir para el desarrollo del procedimiento almacenado SP_SECTORIZACION.

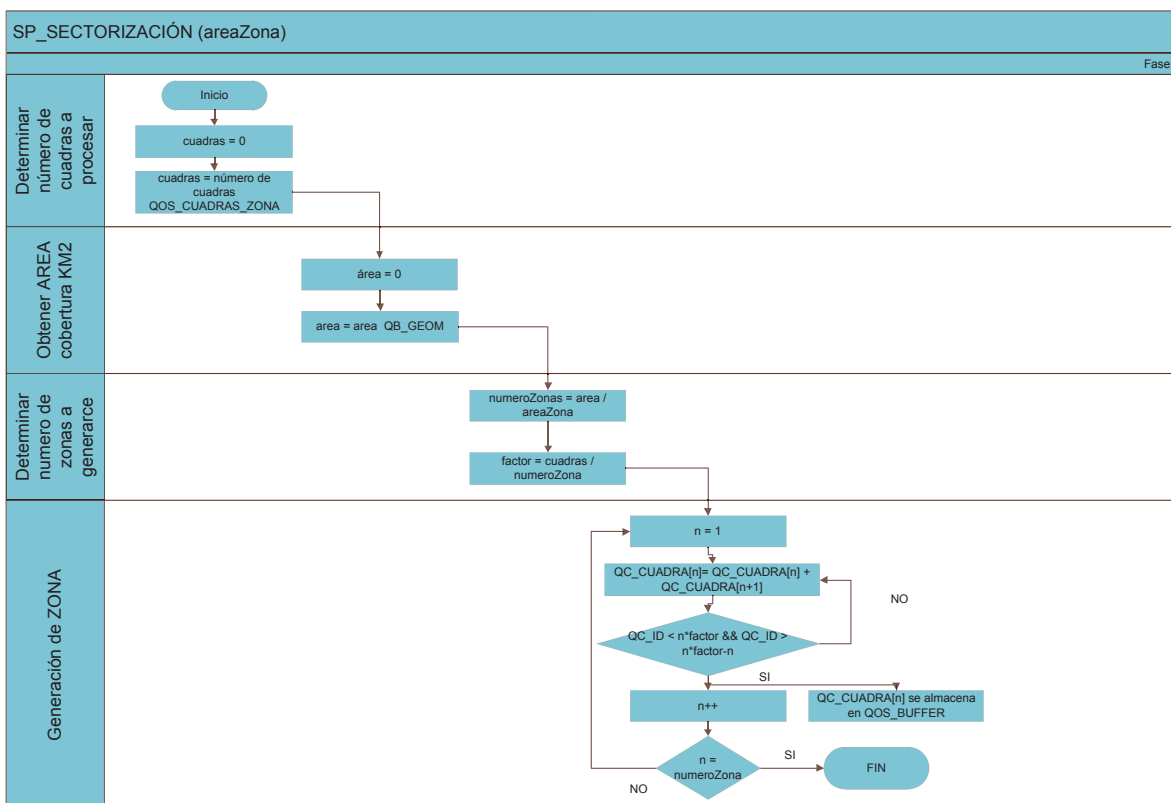


Figura 3.28 Diagrama de Flujo Procedimiento Almacenado SP_SECTORIZACIÓN

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

Cada nueva zona generada será almacenada en la tabla QOS_BUFFER, la cual se describe en la Figura 3.29.

En el campo QB_ZONA se irán agregando los registros de las zonas generadas, debido al tipo de columna espacial debe poseer un índice espacial llamado QB_ZONA.

calidadSMA DAVID.QOS_BUFFER	
QB_ZONA	MDSYS.SDO_GEOMETRY
ROADBUFFERNEW_IDX: QB_ZONA	

Figura 3.29 Tabla QOS_BUFFER

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

3.2.5.2.3 Módulo de Calidad de la Experiencia

El sistema dispondrá de una tabla llamada QOE, donde se almacenarán la calificación (en función de la percepción que el usuario tiene del servicio) otorgada por los usuarios para cada una de las operadoras. La tabla almacenará los datos de las calificaciones (valores numéricos en escala del 1 al 5), y la coordenada espacial correspondiente. Para esta columna debe existir el índice espacial correspondiente. La estructura de la tabla se muestra en la Figura 3.30.

QOE	
PK	<u>qe_id</u>
	qe_operadora qe_cobertura qe_sms qe_llamada qe_coordenada
11	

Figura 3.30 Estructura de la Tabla QOE

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

3.2.5.3 Arquitectura del Prototipo

La siguiente sección describe las interacciones a nivel de controles, servidores y clientes que realiza la aplicación web para mostrar el mapa interactivo con la descripción de los parámetros de calidad de servicio móvil avanzado.

3.2.5.4 Visión de la Aplicación

En la Figura 3.31 se esquematiza la visión de la aplicación donde se detalla la interconexión de todos los componentes, servidores y equipos.

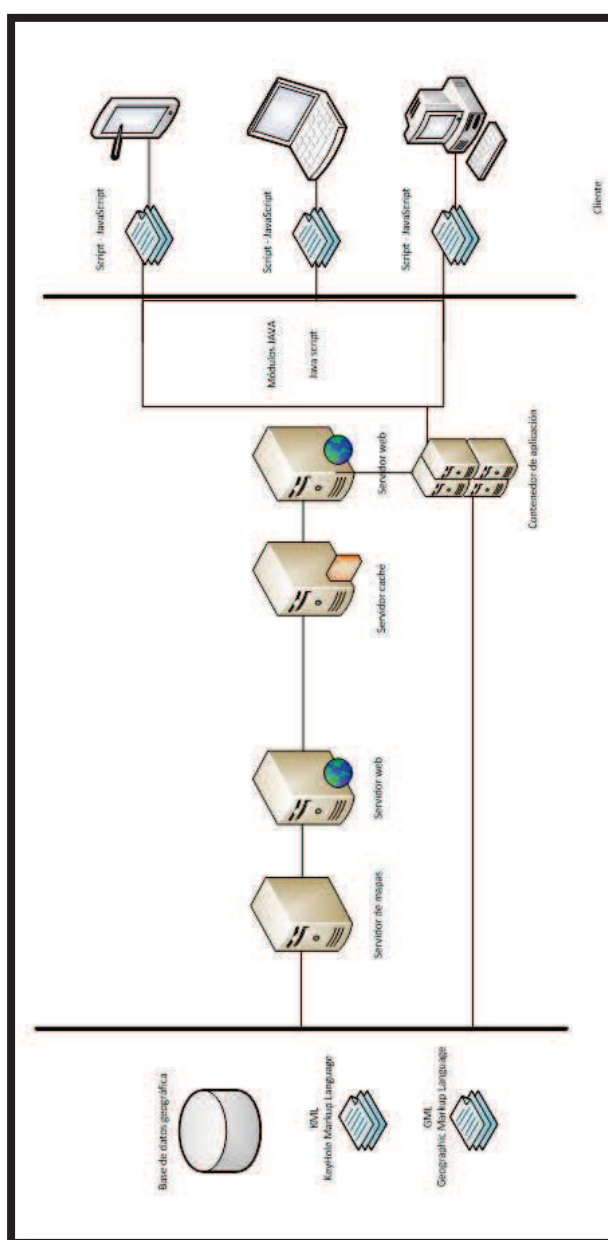


Figura 3.31 Visión de la Arquitectura del Prototipo

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

3.2.5.5 Detalle de la Implementación

En la Figura 3.32 se describen todos los componentes a implementarse, junto con las herramientas a utilizarse, los servidores que alojarán los diferentes servicios y como interactuarán entre todos los componentes, servidores y sistemas externos (GTM y CIR).

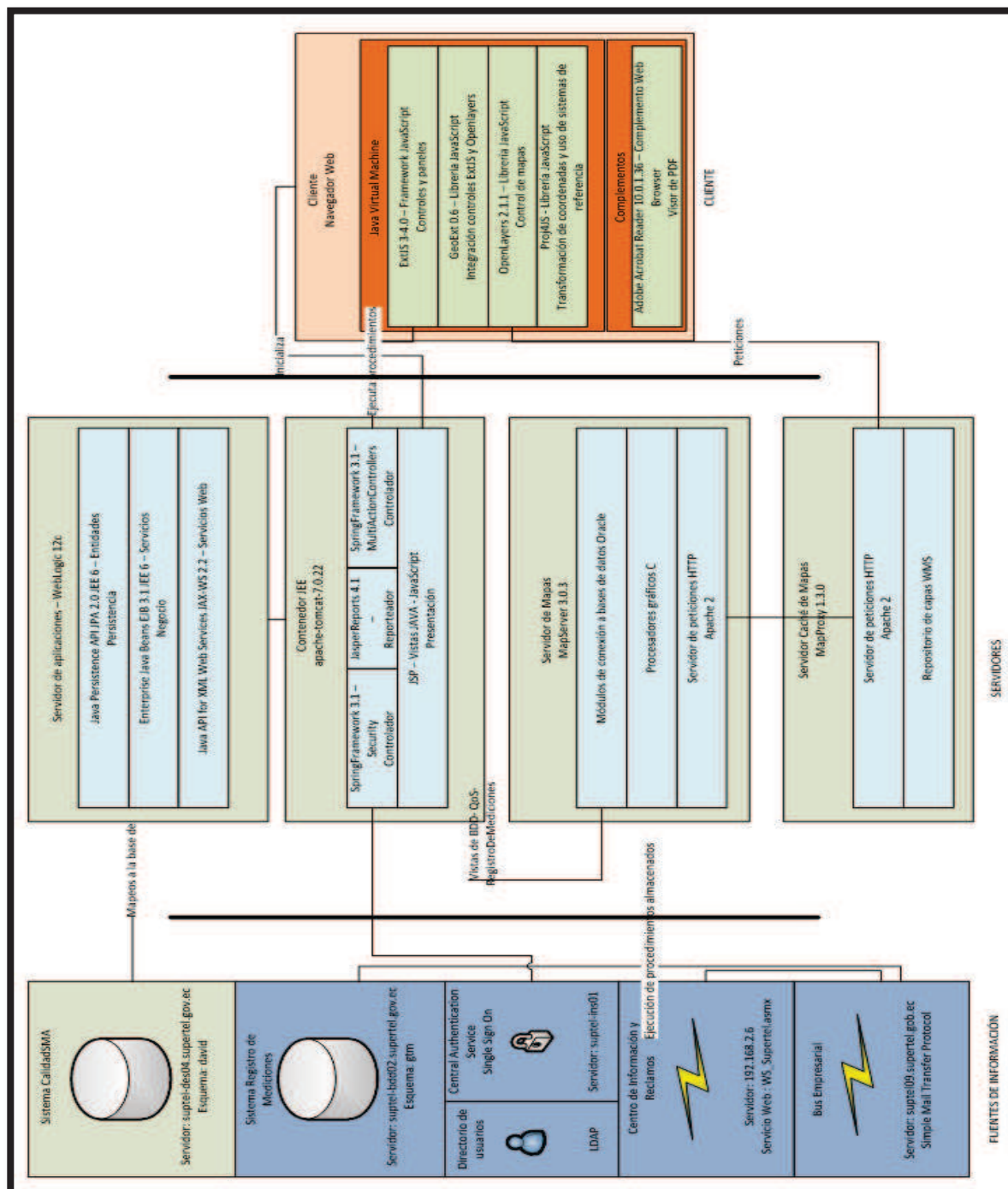


Figura 3.32 Detalle de la Implementación

Fuente: Esquema de Autoría propia (Arévalo D., y Guzmán D.).

4. IMPLEMENTACIÓN Y PRUEBAS DEL PROTOTIPO

4.1 INTRODUCCIÓN

En la fase de implementación y pruebas de la aplicación se persiguen tres objetivos principales. El primer objetivo será plasmar en una versión inicial (comúnmente llamada *beta*), todos los requerimientos definidos en el capítulo anterior para el sistema prototipo. A partir de esta versión, corregir o mejorar aspectos en los cuales se detecten falencias (previo a la realización de pruebas). Con las mejoras y correctivos del caso, se procederá a implementar la versión final del sistema prototipo.

Como se definió en el capítulo 3, el proceso unificado fomenta las buenas prácticas del desarrollo de software, entre las cuales se destaca el desarrollo iterativo. La fase de implementación se la realizará en varias partes o iteraciones; teniendo al final del proceso el sistema completo. Las iteraciones las definirán los casos de uso detallados en el capítulo 3. Cada iteración vendrá acompañada de sus respectivas pruebas de funcionamiento. En la Figura 4.1 se realizará una descripción del proceso de implementación del sistema prototipo.

Las condiciones preliminares especifican la implementación de todos los servidores involucrados, completada esta fase se procederá a crear la base de datos del sistema prototipo calidadSMA a partir del modelo definido en el capítulo anterior.

El proceso siguiente es el de migración de datos del sistema GTM a partir de los archivos de extensión “.dmp”, facilitados por la SUPERTEL, y toda la información geográfica de las zonas y radio bases a la base de datos implementada.

El primer módulo a implementar será el de usuario, el reto inicial será lograr diagramar las zonas de la ciudad de Quito a partir de la información contenida en la base de datos. Para lograr este objetivo se debe diseñar una versión inicial de la interfaz gráfica de la aplicación, con las vistas creadas y las zonas diagramadas. Luego se inicia con el despliegue de los datos de QoS en pantalla.

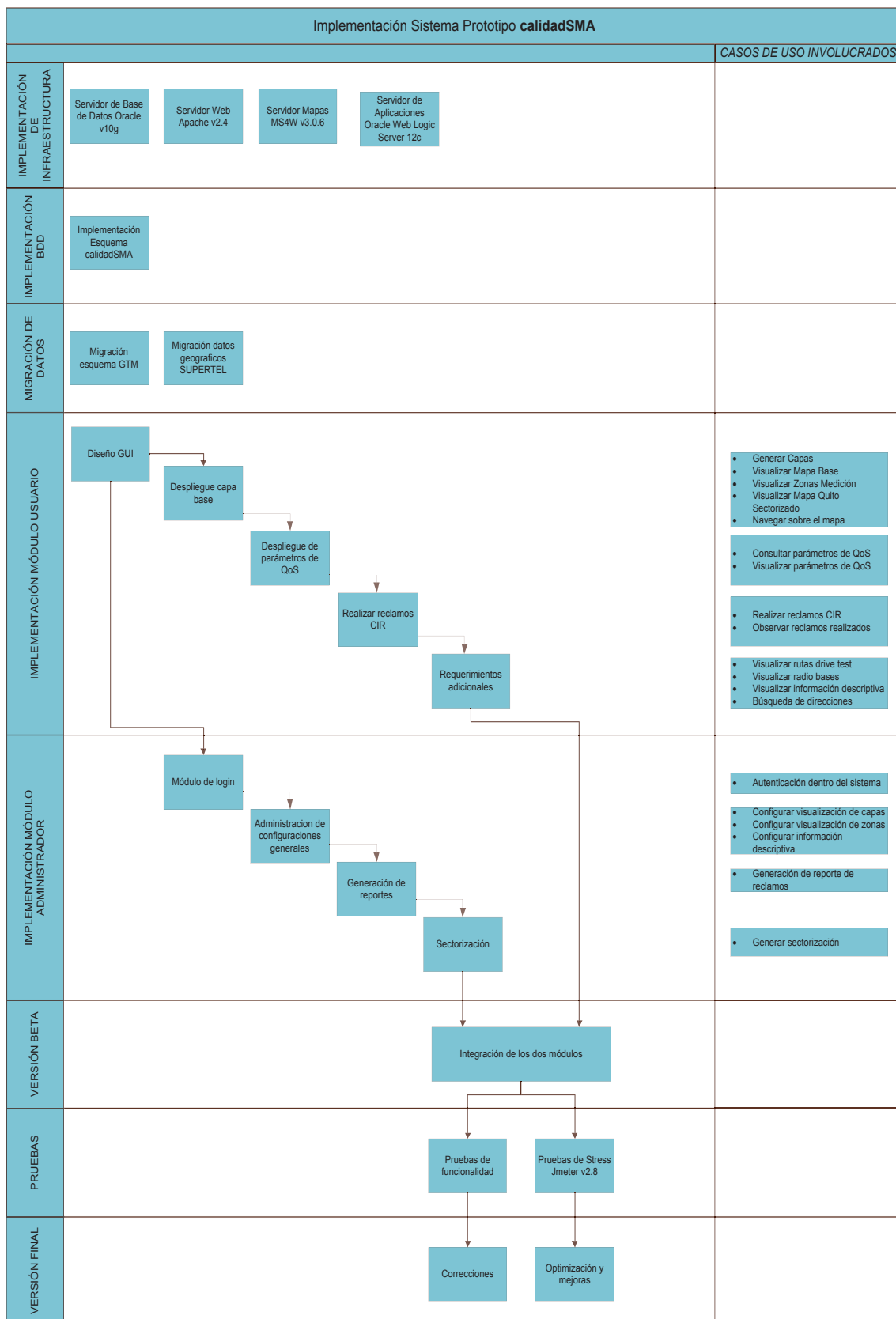


Figura 4.1 Proceso de Implementación Sistema Prototipo calidadSMA.

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

La siguiente iteración consistirá en implementar el servicio web para la comunicación con el CIR; y así, enviar los reclamos de los usuarios y registrarlos en la tabla respectiva.

Finalmente se terminarán de implementar todos los restantes requerimientos. Culminado (y operativo) todo este módulo se iniciará con el diseño del módulo de administración, posteriormente se debe comenzar con la implementación del proceso de *login*, administración de configuraciones generales del sistema y generación de reportes.

La iteración final será la implementación del módulo de sectorización. Una vez implementados los dos módulos surge la versión beta del sistema. La cual será sometida a dos tipos de pruebas: de funcionalidad y de *stress*. De los resultados que arrojen estas pruebas se podrá definir correcciones y mejoras para la versión final del sistema prototipo.

4.2 IMPLEMENTACIÓN DE LA INFRAESTRUCTURA

4.2.1 IMPLEMENTACIÓN DEL SERVIDOR WEB APACHE V2.4

La instalación de este servidor no implica mayores complicaciones. Se debe descomprimir el contenido del archivo en un directorio (para ambientes linux), o usar la versión con instalador para plataformas Windows.

Para que el servidor Apache funcione es necesario que se encuentre instalado el paquete JDK de Java y que exista la variable de entorno JAVA_HOME que direcciona al directorio de instalación del JDK, si no se encuentra la variable JAVA_HOME esta debe ser definida con el comando JAVA_HOME=[ruta jdk] en Linux, set JAVA_HOME=[ruta jdk] en caso de Windows.

Una vez instalado el servidor Apache, en la subcarpeta "bin" se encuentran dos scripts para arrancar y detener el servidor: "*startup.sh*" y "*shutdown.sh*" ("*startup.bat*" y "*shutdown.bat*" en Windows). Para comprobar el correcto funcionamiento del servidor, se puede acceder a la siguiente url : <http://localhost> y direccionándose al puerto 8080.

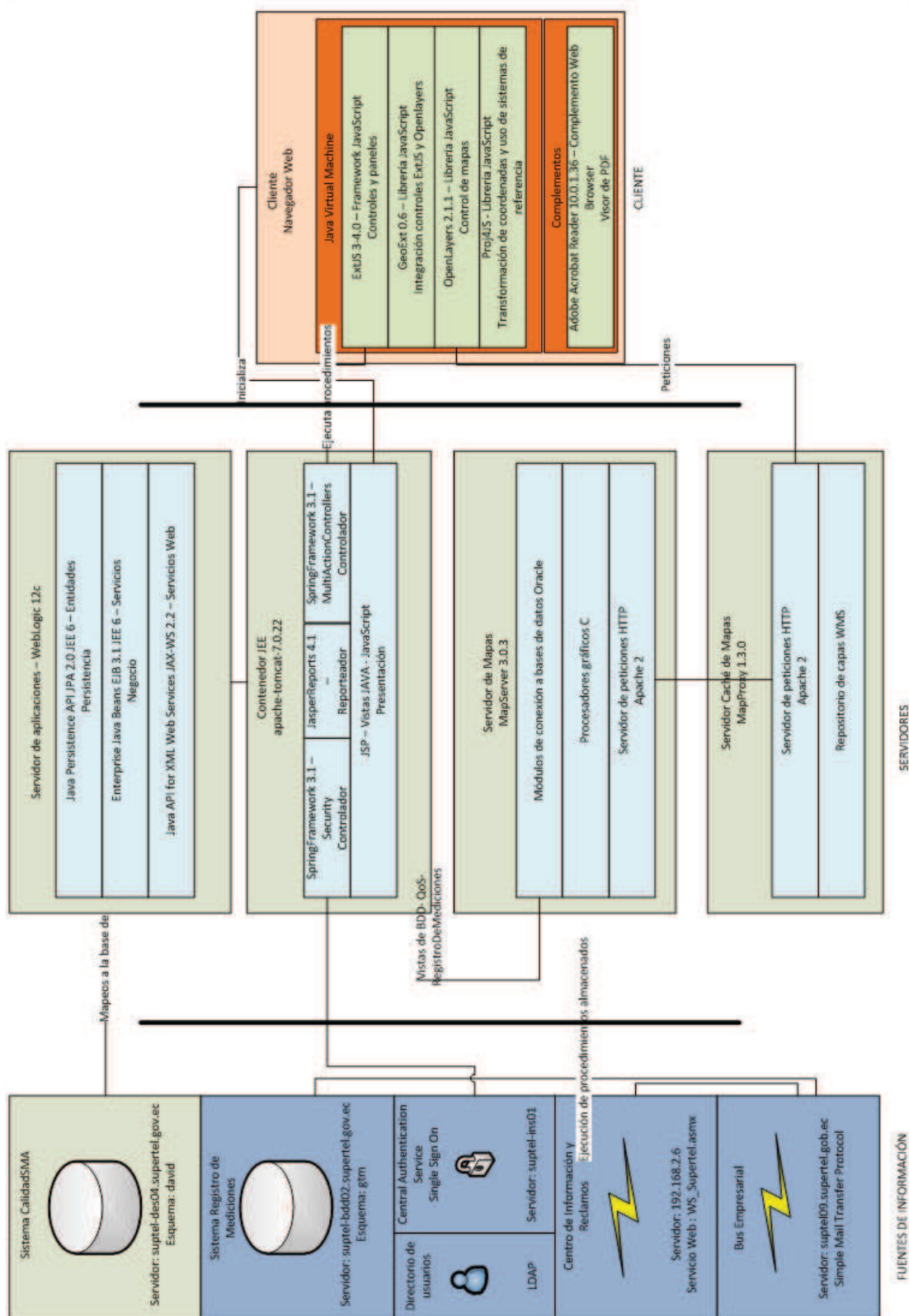


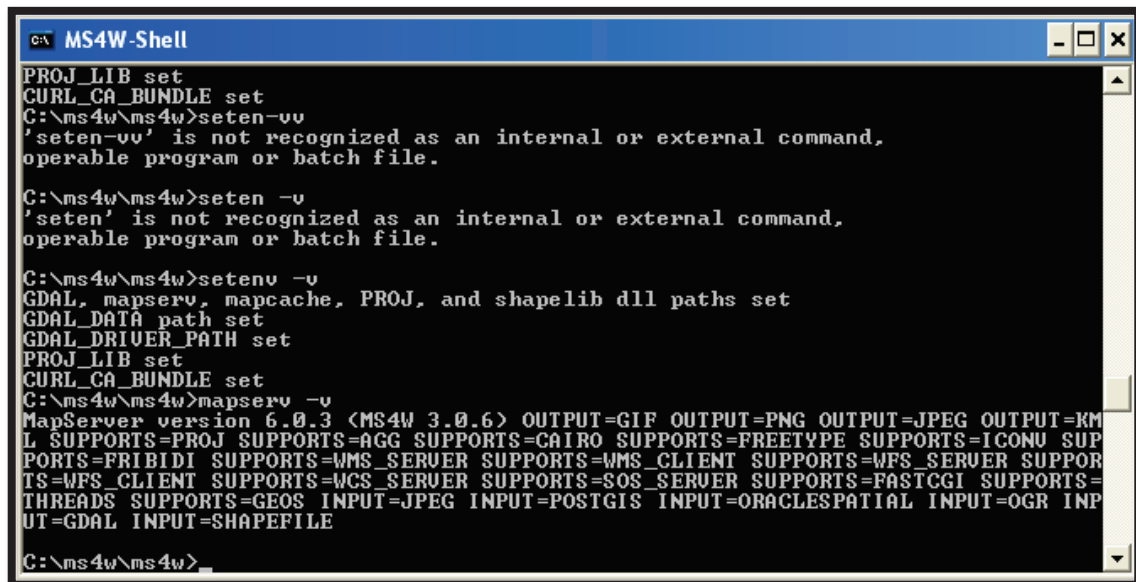
Figura 4.2 Estructura de la Aplicación Web calidadSMA.

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

4.2.2 IMPLEMENTACIÓN DEL SERVIDOR MAPAS MS4W

Para la instalación de este servidor se adjunta un manual en el anexo I. El servidor requiere para su funcionamiento del servidor web Apache, adicionalmente para el sistema prototipo calidadSMA, se debe considerar la inclusión de la librería “libmap.dll” para brindar soporte a los paquetes *Oracle Spatial*, esta librería debe ser ubicada dentro de la carpeta *cgi-bin* del servidor web Apache.

Ejecutando el comando “*mapserv -v*”, se puede observar (Figura 4.3) que el soporte para *Oracle Spatial* está activado.



```

C:\MS4W-Shell
PROJ_LIB set
CURL_CA_BUNDLE set
C:\ms4w\ms4w>seten-uv
'seten-uv' is not recognized as an internal or external command,
operable program or batch file.

C:\ms4w\ms4w>seten -u
'seten' is not recognized as an internal or external command,
operable program or batch file.

C:\ms4w\ms4w>setenv -u
GDAL, mapserv, mapcache, PROJ, and shapelib dll paths set
GDAL_DATA path set
GDAL_DRIVER_PATH set
PROJ_LIB set
CURL_CA_BUNDLE set
C:\ms4w\ms4w>mapserv -v
MapServer version 6.0.3 (MS4W 3.0.6) OUTPUT=GIF OUTPUT=PNG OUTPUT=JPEG OUTPUT=KM
L SUPPORTS=PROJ SUPPORTS=AGG SUPPORTS=CAIRO SUPPORTS=FREETYPE SUPPORTS=ICONV SUP
PORTS=FRIBIDI SUPPORTS=WMS_SERVER SUPPORTS=WMS_CLIENT SUPPORTS=WFS_SERVER SUPPOR
TS=WFS_CLIENT SUPPORTS=WCS_SERVER SUPPORTS=SOS_SERVER SUPPORTS=FASTCGI SUPPORTS=
THREADS SUPPORTS=GEOS INPUT=JPEG INPUT=POSTGIS INPUT=ORACLESPATIAL INPUT=OGR IMP
UT=GDAL INPUT=SHAPEFILE
C:\ms4w\ms4w>

```

Figura 4.3 Servidor MS4W iniciado

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

El servidor de mapas posee un archivo de configuración principal de extensión “.map”, este archivo es fundamental dentro del proceso de generación de capas. En la Figura 4.4 se detalla la estructura básica de un archivo “.map”.

```

MAP
NAME "sample"
STATUS ON
SIZE 600 400
SYMBOLSET "../etc/symbols.txt"
EXTENT -180 -90 180 90
UNITS DD
SHAPEPATH "../data"
IMAGECOLOR 255 255 255
FONTSET "../etc/fonts.txt"

#
# Start of web interface definition
#
WEB
    IMAGEPATH "/ms4w/tmp/ms_tmp/"
    IMAGEURL "/ms_tmp/"
END # WEB

#
# Start of layer definitions
#
LAYER
    NAME 'global-raster'
    TYPE RASTER
    STATUS DEFAULT
    DATA bluemarble.gif
END # LAYER
END # MAP

```

Figura 4.4 Estructura Básica Archivo MapFile⁴⁵

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

Se distingue una etiqueta global de tipo MAP, dentro de esta se encuentra un conjunto de definiciones para todas las capas a generarse, como son nombre, tamaño, *bounding box*, origen de datos, SRID, carpetas destino (donde se van a almacenar las imágenes generadas), y la otra etiqueta fundamental es LAYER, se podrán definir cuantas capas (*layers*), como sean necesarias. Dentro de cada LAYER se debe definir un nombre de capa, tipo de capa, transparente o no, color de fondo o línea, etc.

Para el sistema prototipo calidadSMA se define la estructura detallada en la Figura 4.5, de archivo *MapFile*.

⁴⁵ Fuente : <http://mapserver.org/introduction.html#introduction-to-the-mapfile>

```

MAP
NAME "MS"
STATUS ON
SIZE 800 400
IMAGECOLOR 255 255 255
#DAVIDGUZMAN CAMBIOS PARA TRANSPARENCIA EN LAYER
IMAGETYPE          png
OUTPUTFORMAT
:
NAME PNG
DRIVER "GD/PNG"
MIMETYPE "image/png"
IMAGEMODE RGB
EXTENSION "png"
END

#DAVIDGUZMAN
TRANSPARENT ON

EXTENT -87.1187 -7.3349 -71.2143 3.3655
UNITS meters
PROJECTION
:
"init=epsg:4326"
END #end projection

SYMBOLSET "C:/ms4w/apps/chameleon/etc/symbols.sym"
FONTSET "C:/ms4w/apps/chameleon/etc/fonts.txt"

SHAPEPATH          "c:/mapserver_data/coberturas"

```

Figura 4.5 Cabecera Configuración *MapFile*

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

Se puede distinguir la etiqueta que contiene a todo el archivo (MAP), en la cabecera del archivo se definen los parámetros generales para la generación de los mapas, estos parámetros se describen en la Tabla 4.1.

Tabla 4.1
Parámetros Generales MapFile

Campo	Parámetro	Descripción
Name	MS	Nombre del archivo.
Status	ON	Activa o desactiva todo el mapfile.
Size	800 400	Tamaño en pixeles de la imagen a generarse.
Imagecolor	255 255 255	Color de fondo en formato RGB.
Imagetype	png	Formato de salida de la imagen.
Transparent	on	Activa transparencia en las imágenes a generarse.
Extent	-87.1187 -7.3349 71.2143 3.3655	- <i>Bounding box</i> del mapa a generarse.
units	Meters	Unidad de las coordenadas del mapa.
projection	Init=epsg:4326	SRID del mapa.
Symbolset	C:/.../symbols.sym	Ubicación y nombre de archivo del <i>symbolset</i> a utilizar (<i>path</i> absoluto).
Fontset	C:/.../fonts.txt	Ubicación y nombre de archivo del tipografías a utilizar (<i>path</i> absoluto).
shapepath	C:/mapserver_data/coberturas	<i>Path</i> al directorio que contiene los archivos <i>shape</i> .

Fuente: Tabla de Autoría Propia (Arévalo D., y Guzmán D.).

Para la generación de una capa específica se tiene un ejemplo mostrado en la Figura 4.6.

```

LAYER
  NAME "SECTORES"
  CONNECTIONTYPE oraclespatial
  CONNECTION "david/david@orc1"
  DATA "COORDENADAS FROM QOS_ZONAS_QUITO USING SRID 4326"
  PROJECTION
    "init=epsg:4326"
  END #end projection
  METADATA
    "ows_title" "Sectores"
    "ows_abstract" ""
    "ows_keywordlist" ""
    "ows_extent" "-87.1187, -7.3349, -71.2143, 3.3655"
    "ows_metadatasurl_type" "TC211"
    "ows_dataurl_format" "text/html"
    "ows_dataurl_href" "http://www.yourorganization.org/geonetwork/srv/en/metadata.show?id="
    "ows_style_default_title" "default"
    "ows_format" "image/png"
    "ows_srs" "EPSG:4326"
    "ows_attribution_onlineresource" "http://www.yourorganization.org/"
    "ows_attribution_title" "Data from Your Organization"
    "ows_attribution_logourl_width" "20"
    "ows_attribution_logourl_height" "20"
    "ows_attribution_logourl_format" "image/jpg"
    "ows_attribution_logourl_href" "http://www.yourorganization.org/geonetwork/images/very_small_logo.jpg"
  END #end metadata
  TYPE LINE
  STATUS ON
  TRANSPARENCY 100
  CLASS
    NAME "SECTORES"
    SIZE 20
    TEXT ([zona])
    STYLE
      COLOR 128 64 0
      WIDTH 2
    END
    LABEL
      SIZE medium
      COLOR 30 0 0
      FONT arial
      POSITION cr
    END
  END #end class
END #end layer

```

Figura 4.6 Generación Capa Sectorización Ciudad de Quito

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

Esta capa genera la sectorización de la ciudad de Quito. Los parámetros que se describen en la Tabla 4.2 son los necesarios para la generación de una capa dentro de un servidor de mapas.

Tabla 4.2
Parámetros de Configuración Capa dentro de *MapServer*

Campo	Parámetro	Descripción
Name	SECTORES	Nombre de la capa.
Connnectiontype	oraclespatial	Motor de base de datos espacial.
Connection	“david/david@orcl”	Credenciales de conexión (base de datos).
Data	“COORDENADAS FROM QOS_ZONAS_QUITO USING SRID 4326”	Sentencia de consulta donde se especifica la columna con los datos espaciales y el SRID a utilizar.
Projection	Init=epsg:4326	SRID de la capa a generarse.
Type	LINE	Define como van a ser dibujadas las imágenes a generarse.
NAME	SECTORES	Clase sectores, dentro de esta se definen parámetros como:
1. Size	20	Tamaño de letra de etiqueta.
2. Text	([zona])	Etiqueta para generar leyendas.
3. Style	COLOR 120 64 0 WIDTH 2	Color de la imagen a generarse y grosor de línea de contorno.
4. Label	SIZE médium COLOR 30 0 0 FONT arial POSITION cr	Parámetros de formato de la leyenda. Tamaño, color de letra, tipo de letra, y posición de la etiqueta.

Fuente: Tabla de Autoría Propia (Arévalo D., y Guzmán D.).

4.2.3 IMPLEMENTACIÓN DEL SERVIDOR DE BASE DE DATOS ORACLE 10G

Finalizada la instalación se tienen dos usuarios: SYS y SYSTEM, ambos tienen roles de usuario normal y SYSDBA. Se inicia una instancia de la base de datos, este proceso solo puede ser llevado a cabo por el administrador de la base de datos, para lo cual el administrador inicia una terminal *sqlplus*⁴⁶ en modo SYSDBA (comando / AS SYSDBA) dentro de la terminal. El comando para iniciar la instancia de la base de datos es STARTUP.

```

Archivo Editar Ver Terminal Ayuda
root@qosdebian:/home/qosepn# su - oracle
ulimit: 46: Illegal option -u
$ sqlplus / AS SYSDBA

SQL*Plus: Release 10.2.0.1.0 - Production on Tue Apr 24 20:50:47 2012

Copyright (c) 1982, 2005, Oracle. All rights reserved.

Connected to an idle instance.

SQL> STARTUP
ORACLE instance started.

Total System Global Area 436207616 bytes
Fixed Size                1219856 bytes
Variable Size             100664048 bytes
Database Buffers         331350016 bytes
Redo Buffers              2973696 bytes
Database mounted.
Database opened.
SQL>

```

Figura 4.7 Base de Datos Oracle 10g Iniciada

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

Adicionalmente se crea el usuario DAVID, este usuario tendrá privilegios de administrador pero solo sobre el esquema definido para el sistema prototipo. Finalmente se inicia el demonio *listener*, el cual permite conectarse remotamente hacia la base de datos, para este fin se debe ejecutar el comando "*lsnrctl start*". Los parámetros de conexión hacia la base de datos se describen en la Tabla 4.3.

⁴⁶ Terminal que provee un conjunto de comandos básicos para administración de una base de datos Oracle.

Tabla 4.3
Parámetros de Conexión Servidor de Base de Datos

Parámetro	Valor
Nombre de host	192.88.255.131
Puerto	1522
SID ⁴⁷	orcl

Fuente: Tabla de Autoría Propia (Arévalo D., y Guzmán D.).

4.2.4 IMPLEMENTACIÓN DEL SERVIDOR DE APLICACIONES ORACLE WEB LOGIC 12C

El paquete de instalación a utilizar es el correspondiente al archivo: **oepe-indigo-installer-12.1.1.0.1.201203120349-12.1.1-win32**⁴⁸. El proceso de instalación no genera dificultad, el único aspecto a tener en cuenta es el tener previamente instalado el paquete JDK versión 1.6, componente que se encuentra disponible en la dirección web:

<http://www.oracle.com/technetwork/es/java/javase/downloads/index.html>

El proceso de instalación inicia con la pantalla mostrada en la Figura 4.8:

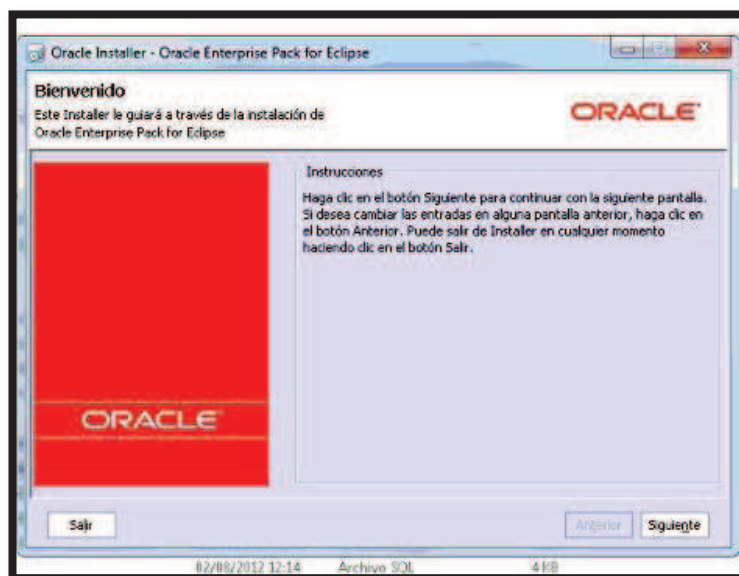


Figura 4.8 Inicio de Instalación Oracle WebLogic 12c

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

⁴⁷ Site Identifier, este identificador por defecto esta definido como orcl para versiones Enterprise Edition de Oracle DB y como XE para versiones Express Edition.

⁴⁸ Paquete disponible a través de la siguiente dirección web: <http://www.oracle.com/technetwork/middleware/weblogic/downloads/index.html>

Durante el proceso de instalación únicamente se deben escoger los valores por defecto. Finalmente, se desplegará una pantalla donde se indican los parámetros de instalación del servidor.

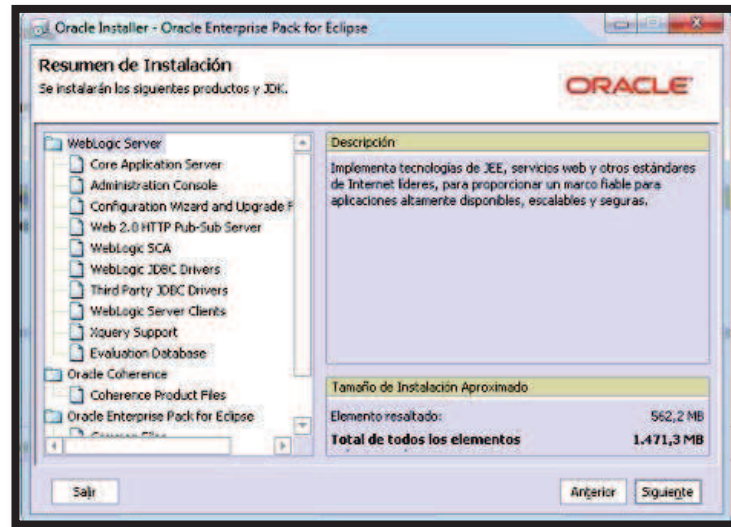


Figura 4.9 Parámetros de Instalación Típica de Oracle WebLogic Server 12c

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

Finalizado el proceso de instalación sin ningún inconveniente se debe desplegar la siguiente pantalla, donde se debe escoger la opción “Ejecutar *QuickStart*” para iniciar el proceso de configuración del servidor.

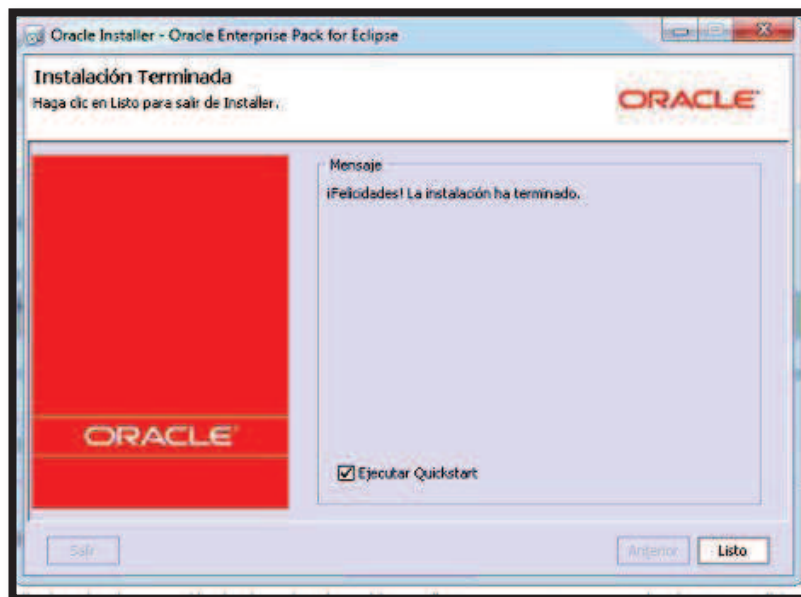


Figura 4.10 Fin de Instalación Oracle WebLogic 12c

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

En la pantalla de inicio se debe seleccionar la opción “*Getting Started with Weblogic*”.



Figura 4.11 Pantalla de Inicio de Configuración Oracle *WebLogic* 12c

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

En primera instancia se procede a configurar el dominio para el servidor Oracle *WebLogic* 12c, para lo cual se debe crear un dominio nuevo de *WebLogic* 12c.

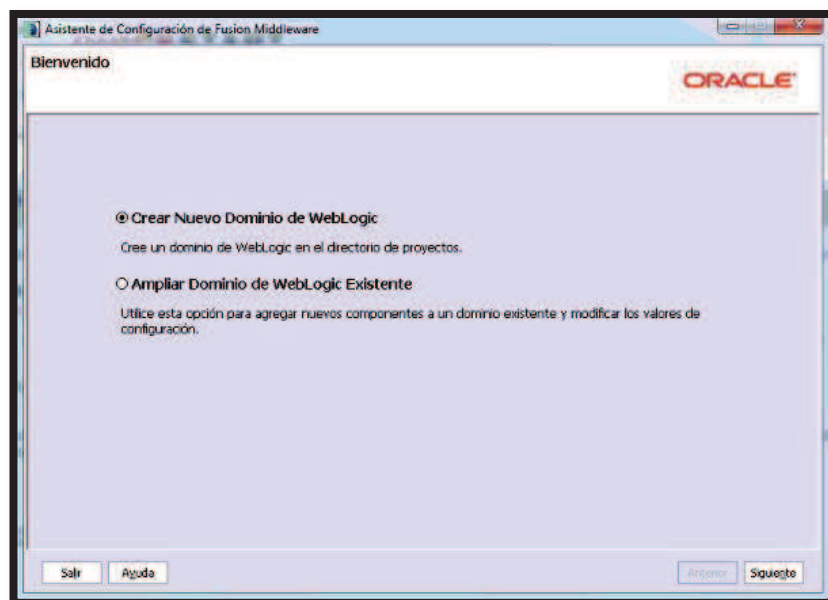


Figura 4.12 Creación de Dominio para Oracle *WebLogic* 12c

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

Es fundamental seleccionar la opción de soporte de productos para el servidor como se aprecia en la Figura 4.13. Para el caso de la aplicación calidadSMA el producto a soportar son servicios web desarrollados bajo el API *Java XML for Web Services* (JAX-WS), API que se encuentra integrada dentro de la plataforma JEE.

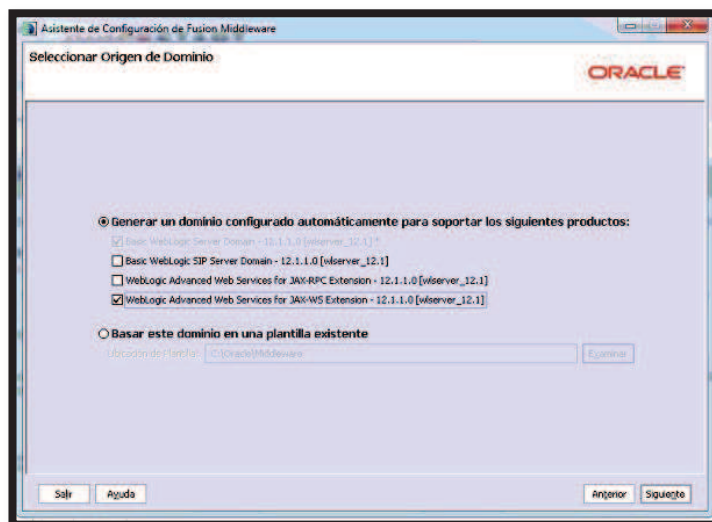


Figura 4.13 Configuración de Oracle WebLogic 12c, para brindar Soporte para JAX-WS

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

Finalmente, se debe proporcionar las credenciales de acceso al módulo de administración del servidor.

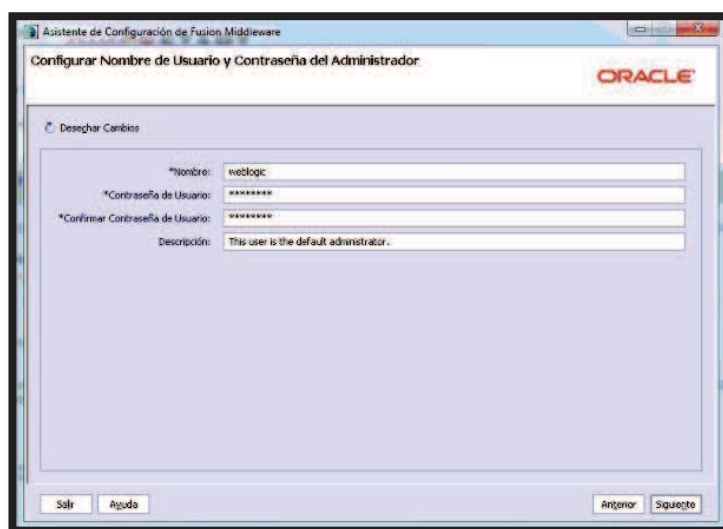


Figura 4.14 Definición de Credenciales de Acceso al Servidor Oracle WebLogic 12c

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

Previo al proceso de configuración del servidor se desplegará un resumen con todos los parámetros definidos posteriormente.

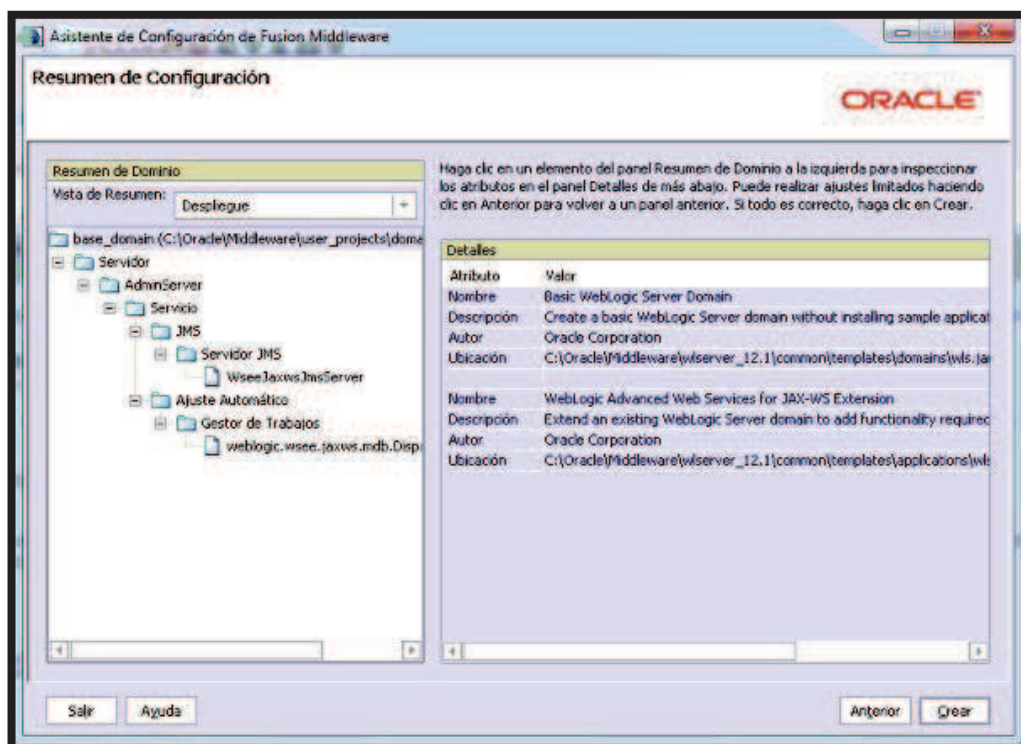


Figura 4.15 Parámetros de Configuración Oracle WebLogic 12c

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

Finalizada la configuración, se debe iniciar el servidor, luego acceder a la consola de administración a través de la siguiente url: <http://localhost:7001/console>, y proporcionar al credenciales configuradas previamente. La pantalla de administración debe desplegarse para poder continuar con la configuración de los orígenes de datos para el servidor instalado. Proceso que será descrito luego de la implementación de la base de datos.

4.3 IMPLEMENTACIÓN DE LA BASE DE DATOS

4.3.1 CREACIÓN DE TABLAS

Las tablas fueron creadas en base al diagrama presentado en el capítulo 3. A manera de ejemplo, a continuación se describe el proceso para la creación de la tabla QOS_ZONAS_QUITO. Con el comando CREATE TABLE (Espacio de Código 4.1) y definiendo cada una de las columnas se crea la tabla.

```
CREATE TABLE "DAVID"."QOS_ZONAS_QUITO"
( "GD_ID" NUMBER,
  "GD_ZONA" VARCHAR2(32 BYTE),
  "GD_COORDENADAS" "MDSYS"."SDO_GEOMETRY" ,
  "GD_DESCRIPCION" VARCHAR2(20 BYTE),
  "GD_ESTADO" NUMBER,
  "GD_COD_PRO" NUMBER,
  "GD_COD_CAN" NUMBER,
  "GD_ID_PAR" NUMBER,
  "GD_JURISDI" VARCHAR2(4 BYTE),
  "GD_TIPO_ZONA" VARCHAR2(20 BYTE)
) ;
```

Espacio de Código 4.1 Script Generación Tabla Sectores Quito

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

Luego se crean los índices espaciales (solo para las tablas que contengan información geográfica), utilizando el comando CREATE INDEX “nombre de tabla” ON “columna tipo espacial” *INDEXTYPE IS* “tipo de índice”.

```
CREATE INDEX "DAVID"."QOS_ZONAS_IDX1" ON "DAVID"."QOS_ZONAS_QUITO" ("GD_COORDENADAS")
INDEXTYPE IS "MDSYS"."SPATIAL_INDEX" ;
```

Espacio de Código 4.2 Creación Índice Espacial

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

Para finalmente crear las claves primarias (si son necesarias), y las claves foráneas (de ser necesarias).

```
ALTER TABLE "DAVID"."QOS_ZONAS_QUITO" ADD CONSTRAINT "QOS_ZONAS_QUITO_PK" PRIMARY KEY ("GD_ID") DISABLE;
ALTER TABLE "DAVID"."QOS_ZONAS_QUITO" MODIFY ("GD_ID" NOT NULL ENABLE);
```

Espacio de Código 4.3 Creación de Claves Foráneas

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

Proceso similar se debe seguir para la implementación de cada una de las tablas que constan en el esquema definido en el capítulo 3.

4.3.2 CREACIÓN DE VISTAS

En el Espacio de Código 4.4, se describe el proceso para la creación de las vistas, al ser un proceso similar para las tres operadoras solo se describe uno de los 3 *scripts* generados.


```

CREATE OR REPLACE FORCE VIEW "DAVID"."QOS_CONECEL" (
  "ID", "F1CIUDAD", "ZONA", "COORDENADAS", "F1PORC LLAMADAS ESTABL",
  "F1TECNOLOGIA", "F1PORC_SMSS_RECIBIDOS", "F1PORCENTAJE_COBERTURA") AS
(
SELECT qos zonas quito.id,
       gtm.stm1mv.flciudad@gtm,
       qos_zonas_quito.zona,
       qos_zonas_quito.coordenadas,
       nvl(to_number(gtm.stm1mv.f4porcentaje_de_llamadas, '999999.9999'),
          nvl(to_number(gtm.stm1mv.flporc_llamadas_establ, '999999.9999'), 0)),
       gtm.stm9mv.fltecnologia@gtm ,
       nvl(to_number(gtm.stm7mv.f4porc_smss_exitosos@gtm, '999999.9999') ,
          nvl(to_number(gtm.stm7mv.flporc_smss_recibidos@gtm, '999999.9999'), 0)),
       nvl(to_number(gtm.stm9mv.f4porcentaje_de_cobertura@gtm, '999999.9999') ,
          nvl(to_number(gtm.stm9mv.flporcentaje_cobertura@gtm, '999999.9999'),0))
FROM qos_zonas_quito, gtm.stm1mv@gtm , gtm.stm7mv@gtm, gtm.stm9mv@gtm
WHERE (gtm.stm1mv.flzona@gtm) = qos_zonas_quito.zona
AND gtm.stm1mv.flciudad@gtm = 'QUITO'
and gtm.stm1mv.floperadora@gtm = 'Conecel-Porta'
AND (gtm.stm7mv.flzona@gtm) = qos_zonas_quito.zona
AND gtm.stm7mv.flciudad@gtm = 'QUITO'
and gtm.stm7mv.floperadora@gtm = 'Conecel-Porta'
AND (gtm.stm9mv.flzona@gtm) = qos zonas quito.zona
AND gtm.stm9mv.flciudad@gtm = 'QUITO'
and gtm.stm9mv.floperadora@gtm = 'Conecel-Porta'
);

```

Espacio de Código 4.4 Creación Vista de Parámetros de QoS Operadora CONECEL

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

Los aspectos importantes a tomar en cuenta son:

- Los datos provenientes del sistema GTM son de tipo *varchar*, razón por la cual fue necesario realizar un proceso CAST⁴⁹, para obtener los datos de parámetros de QoS en un formato numérico. Se utilizó la función de Oracle con el siguiente formato: TO_NUMBER (cadena de caracteres, 'formato numérico de salida'). Como se observa en el Espacio de Código 4.4 el formato fue de seis cifras enteras con cuatro decimales.
- Como se mencionó en el capítulo uno, el proceso de medición de parámetros de QoS consta de cinco fases, y tanto en la primera como en la cuarta fase se realizan mediciones (la medición en fase cuatro, es para verificación si existe o no solución a los problemas detectados en fase uno). Por este motivo, es necesario escoger las mediciones de la fase con

⁴⁹ Proceso informático para transformar entre diferentes tipos de datos (por ejemplo de un dato tipo carácter a tipo numérico).

los datos más actuales. Es decir primero se deben escoger los datos en fase cuatro, si no existen se escogen los de fase uno. La función *Not Null Value* (NVL) de Oracle, facilita este proceso. La función *nvl* (parametro1, paramtero2), escoge por defecto parámetro 1, pero si este es nulo escoge el parámetro 2.

4.3.3 CREACIÓN DEL PROCEDIMIENTO ALMACENADO SECTORIZAR

Dentro del esquema MDSYS de la base de datos se encuentran alojadas las dos funciones a utilizar para la generación automática de los sectores. Para implementar este procedimiento, se utilizaron las siguientes funciones espaciales descritas a continuación.

4.3.3.1 Función Sdo_geom.sdo_area ()

Esta función devuelve el valor numérico del área de una geometría espacial, recibe como parámetros de entrada: la geometría espacial (columna espacial), un valor de tolerancia y la unidad de medida en la que devolverá el resultado. Su sintaxis se describe a continuación:

SDO_GEOM.SDO_AREA(geometría, tolerancia, unidad de medida).

Por ejemplo:

geometría =COORDENADAS (columna espacial de tabla QOS_ZONAS_QUITO).

tolerancia = 0.0005.

unidad de media = '*UNITS*=SQ_KM' (kilómetros cuadrados).

4.3.3.2 Función Sdo_aggr_union()

Esta función realiza la operación de unión u "OR" de un conjunto de geometrías espaciales, recibe como parámetros de entrada las geometrías espaciales a unir y un valor de tolerancia, este valor le permite a la función determinar cuándo dos geometrías espaciales están contiguas; es decir, en algún punto(s) de su perfil limítrofe son comunes. La tolerancia define cual será la tolerancia máxima para asumir que dos figuras están contiguas. Por ejemplo, dadas dos figuras A, B

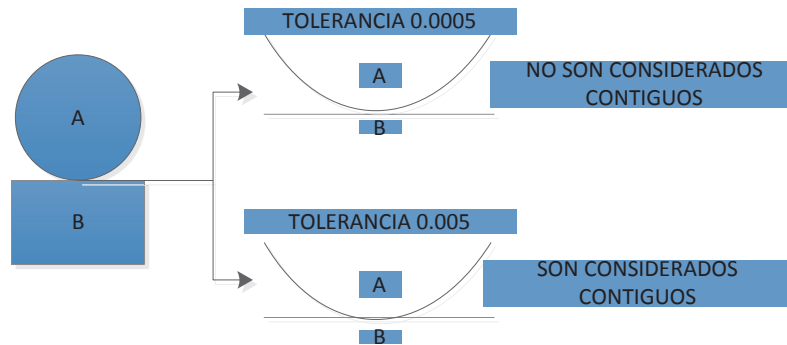


Figura 4.16 Tolerancia en Objetos Espaciales

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

Su sintaxis se describe a continuación:

SDO_AGGR_UNION(geometrías, tolerancia).

Por ejemplo:

geometría =COORDENADAS (columna espacial de tabla QOS_ZONAS_QUITO).

tolerancia = 0.0005.

El Espacio de Código 4.5, implementa el procedimiento SECTORIZAR utilizando las dos funciones anteriormente descritas.

```

create or replace
procedure sectorizar
is
manzanas number;
area number;
areaZonas number:= 4;
numeroZona number;
factor number;
contador number;
begin
execute immediate 'truncate table road_buffer1';
select count(*) into manzanas from road_buffer c;
select sum(sdo_geom.sdo_area ( c.geom, 0.0005 , 'unit=SQ_KM')) into area from road_buffer c ;
numeroZona := (area/areaZonas);
factor := (manzanas/numeroZona);

FOR contador IN 1..numeroZona
LOOP
INSERT INTO qos_buffer
SELECT
SDO_AGGR_UNION(
MDSYS.SDOAGGRTYPE(c.geom, 0.005))
FROM road_buffer c
where c.id < (contador*factor) and c.id > ((contador*factor)-factor);
END LOOP;
execute immediate 'commit';
end;

```

Espacio de Código 4.5 Procedimiento Almacenado para Sectorización

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

4.3.4 CREACIÓN DE FUNCIONES.

4.3.4.1 Función FUNC_PARAM_FROM_ZONA

La siguiente función `FUNC_PARAM_FROM_ZONA` recibe una coordenada geográfica (latitud y longitud) y devuelve la zona a la que pertenece esa coordenada, más los parámetros de QoS de esa zona (si están registrados en el sistema GTM).

```
CREATE OR REPLACE
FUNCTION "FUNC_PARAM_FROM_ZONA" (p_lat IN NUMBER,
p_lon IN NUMBER, tecnologia IN varchar2)
RETURN types.ref_cursor
AS
my_cursor types.ref_cursor;
BEGIN
OPEN my_cursor FOR
SELECT conecel.zona, conecel.flporc_llamadas_establ,
conecel.flporc_smss_recibidos, conecel.flporcentaje_cobertura, conecel.fltecnologia
FROM qos_otecel conecel
WHERE SDO_GEOM.SDO_INTERSECTION(conecel.coordenadas,
sdo_geometry(2001, 4326, SDO_POINT_TYPE(p_lat, p_lon, NULL), NULL, NULL), 0.005)
IS NOT NULL and conecel.fltecnologia = tecnologia;

RETURN my_cursor;
END;
```

Espacio de Código 4.6 Función Parámetros de QoS de una Zona

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

La función de Oracle Spatial que permite averiguar si un objeto espacial interseca con otro, es la función `SDO_GEOM.SDO_INTERSECTION` (objeto espacial 1, objeto espacial 2), cabe destacar que los dos objetos a comparar deben ser geometrías espaciales, por lo tanto la coordenada que se recibe como parámetro de entrada debe ser convertida a una geometría de tipo punto.

4.4 MIGRACIÓN DE DATOS

4.4.1 MIGRACIÓN ESQUEMA GTM

Para el proceso de migración de la base de datos GTM se ha utilizado la herramienta proporcionada por Oracle llamada "imp", esta herramienta se ejecuta a través de la consola "sqlplus" de Oracle. Aspectos importantes a tomar en cuenta antes de iniciar el proceso de migración son que este proceso debe realizarlo un usuario con privilegios de administrador, o el usuario "system" (de la base de datos) directamente autenticado como "sysdba". Y el otro aspecto es que no debe existir un esquema con el mismo nombre del que se va a importar.

Para este efecto, en una terminal se debe ejecutar el siguiente comando:

```
sqlplus / AS SYSDBA
```

Luego se ejecuta la sentencia “imp” con los parámetros que se detalla a continuación:

```
imp (usuario/contraseña)
```

Si el proceso se realizó con éxito, el esquema GTM debe estar disponible dentro de la base de datos local como se aprecia en la Figura 4.17.

F1REGIONAL	F1OPERADORA	F1PROVINCIA	F1CIUDAD	F1ZONA	F1TECNOLOGIA	F1FECHA_DE_MEDICION	
1	IRN	Conecel-Porta	IMBABURA	IBARRA	2	WCOMA	19/04/2011
2	IRN	Conecel-Porta	IMBABURA	IBARRA	4	WCOMA	18/04/2011
3	IRN	Conecel-Porta	IMBABURA	OTAVALO	OTAVALO	WCOMA	20/04/2011
4	IRN	Conecel-Porta	IMBABURA	COTACACHI	COTACACHI	WCOMA	20/04/2011
5	IRN	Conecel-Porta	IMBABURA	SAN PABLO	SAN PABLO	WCOMA	20/04/2011
6	IRC	Otecel-Movistar	GUAYAS	DATA DE PLAYAS	DATA DE PLAYAS	GSM	09/03/2011
7	IRC	Otecel-Movistar	GUAYAS	VALLE DE LA VIRGEN	VALLE DE LA VIRGEN	GSM	14/03/2011
8	IRC	Otecel-Movistar	GUAYAS	SAMBORONDON	SAMBORONDON	GSM	16/03/2011
9	IRC	Otecel-Movistar	SANTA ELENA	BAMBIL	BAMBIL	GSM	17/03/2011
10	IRC	Otecel-Movistar	EL ORD	EL RETIRO	EL RETIRO	GSM	23/03/2011
11	IRC	Otecel-Movistar	EL ORD	CHILLA	CHILLA	GSM	30/03/2011
12	DRG	Otecel-Movistar	GALAPAGOS	EL PROGRESO	EL PROGRESO	GSM	17/05/2011
13	DRG	Otecel-Movistar	GALAPAGOS	BELLAVISTA	BELLAVISTA	GSM	23/05/2011
14	DRG	Otecel-Movistar	GALAPAGOS	SANTA ROSA	SANTA ROSA	GSM	23/05/2011
15	DRG	Otecel-Movistar	GALAPAGOS	PUERTO AYORA	PUERTO AYORA	GSM	23/05/2011
16	DRG	Otecel-Movistar	GALAPAGOS	PUERTO BAQUERIZO MORENO	PUERTO BAQUERIZO MORENO	GSM	17/05/2011
17	IRN	Conecel-Porta	FRANCISCO DE ORELLANA	PUERTO FRANCISCO DE ORELLANA (COCA)	2	WCOMA	03/05/2011
18	IRN	Conecel-Porta	IMBO	TEMA	1	WCOMA	03/05/2011
19	IRC	Conecel-Porta	EL ORD	SANTA ROSA	ZONA 2	GSM	02/06/2011
20	IRC	Conecel-Porta	EL ORD	SANTA ROSA	ZONA 3	GSM	02/06/2011
21	IRC	Otecel-Movistar	GUAYAS	NARANJITO	2	GSM	30/06/2011
22	IRC	Otecel-Movistar	GUAYAS	CORONEL MARCELINO MARIJUEÑA	2	GSM	01/07/2011
23	IRS	Otecel-Movistar	CAÑAR	PARROQUIA INGAPIRCA	GSM	26/06/2011	
24	IRS	Otecel-Movistar	CAÑAR	BIBLIAN	PARROQUIA NAZON	GSM	27/06/2011
25	IRS	Otecel-Movistar	CAÑAR	AZOGUES	LUIS CORDEO	GSM	06/07/2011
26	IRC	Conecel-Porta	LOS RIOS	PIMCHA	PIMCHA	GSM	18/04/2011
27	IRC	Otecel-Movistar	LOS RIOS	PIMCHA	PIMCHA	GSM	18/04/2011
28	IRC	Conecel-Porta	LOS RIOS	LECHUGAL	LECHUGAL	GSM	19/04/2011
29	IRC	Otecel-Movistar	LOS RIOS	LECHUGAL	LECHUGAL	GSM	19/04/2011
30	IRC	Conecel-Porta	LOS RIOS	LA ESPERANZA	LA ESPERANZA	GSM	20/04/2011
31	IRC	Otecel-Movistar	LOS RIOS	LA ESPERANZA	LA ESPERANZA	GSM	20/04/2011

Figura 4.17 Base de Datos Esquema GTM – SUPERTEL

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

4.4.2 MIGRACIÓN DE DATOS GEOGRÁFICOS

Para poder visualizar las zonas a partir de su correspondiente archivo “.shp” se realiza el siguiente proceso:

Dentro del programa *Quantum GIS v1.8*:

- Se añade una capa ráster (Figura 4.18).
- En la ventana emergente, en conjunto de datos se escoge el archivo “.shp” que se desea desplegar (Figura 4.19).

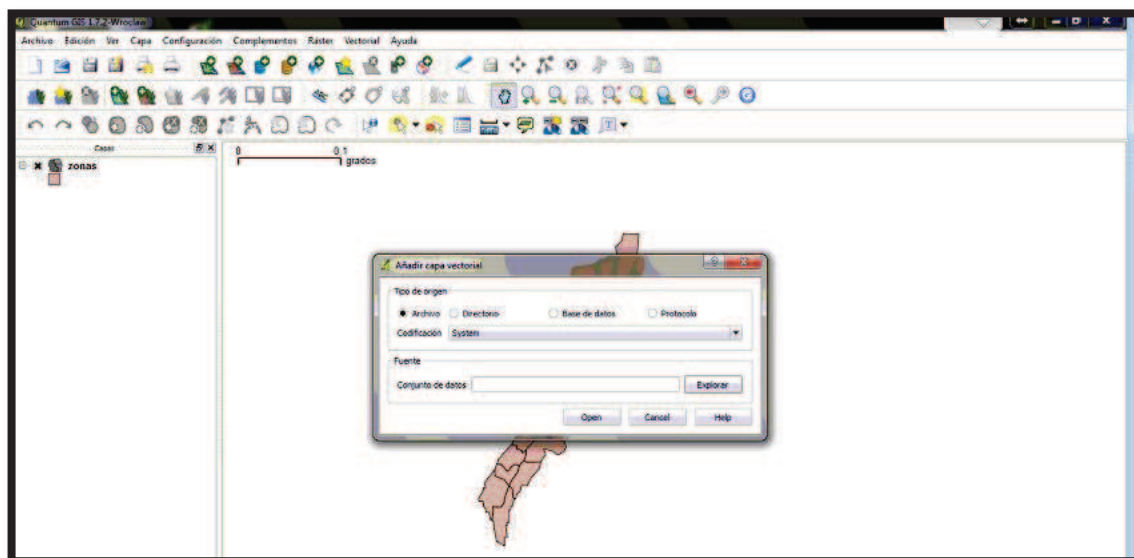


Figura 4.18 Añadir Capa Vectorial en Quantum GIS

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

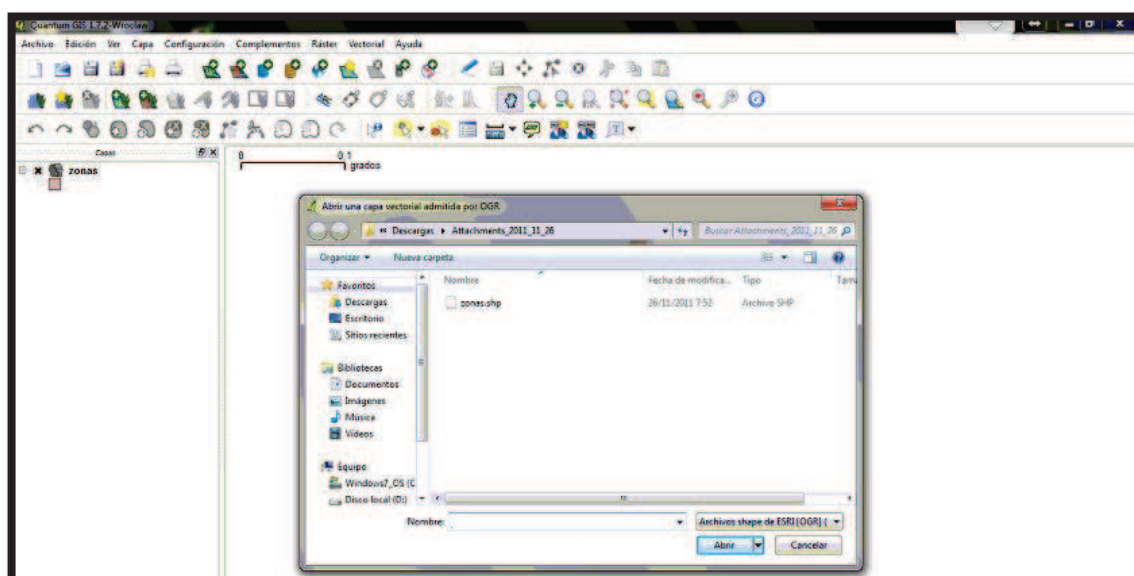


Figura 4.19 Selección del Archivo “.shp”

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

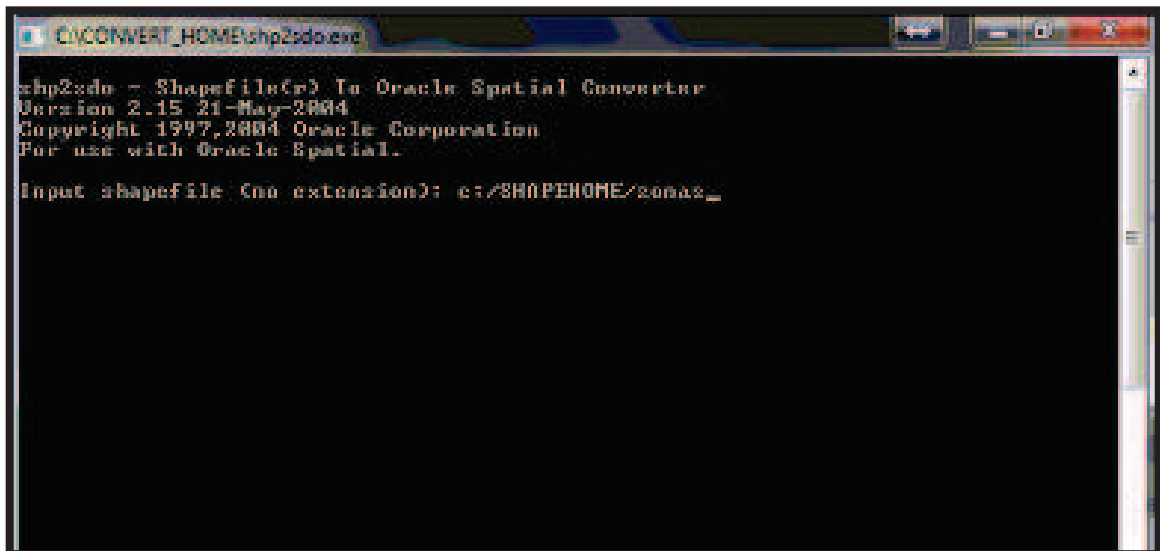
Una vez seleccionado el archivo, se desplegarán en pantalla las 26 zonas de Quito, cada zona con su respectiva información geográfica (latitud-longitud). El objetivo consiste en migrar todos los datos de las zonas a la base de datos donde contenga fundamentalmente: un identificador de cada zona, el nombre de la zona

y un conjunto de datos de tipo espacial en el que consten todos los puntos geográficos que conforman / delimitan la zona.

El paquete Oracle 10g proporciona una herramienta que permite migrar archivos de tipo *shape* hacia una base de datos del tipo espacial, esta herramienta es la *shp2sdo* (abreviatura de *shape to sdo*).

Una vez que se disponga del archivo “.shp”, el proceso para convertir los archivos es el siguiente:

- Crear dos carpetas la una que contenga el archivo *shape*, y una carpeta de destino para los archivos convertidos.
- Creadas las dos carpetas, ejecutar la herramienta *shp2sdo* desde una consola (windows) o terminal (linux); se desplegará una pantalla de inicio (Figura 4.20).
- En la línea *input shape file*, se debe especificar el *path* del directorio donde se encuentra el archivo a convertir.

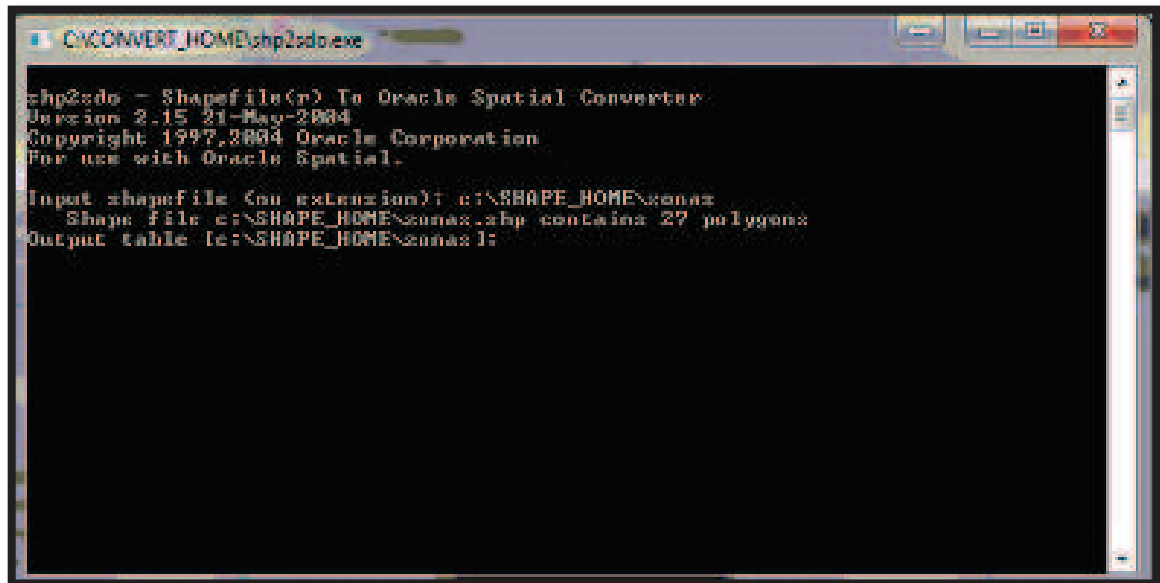


```
C:\CONVERT_HOME>shp2sdo.exe
shp2sdo - Shapefile(s) To Oracle Spatial Converter
Version 2.15 21-May-2004
Copyright 1997,2004 Oracle Corporation
For use with Oracle Spatial.
Input shapefile (no extension): c:/SHAPEHOME/zonas_
```

Figura 4.20 Herramienta *shp2sdo* en ejecución

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

- El comando solicitará el nombre de la tabla que se creará y el lugar donde se alojarán los datos.



```

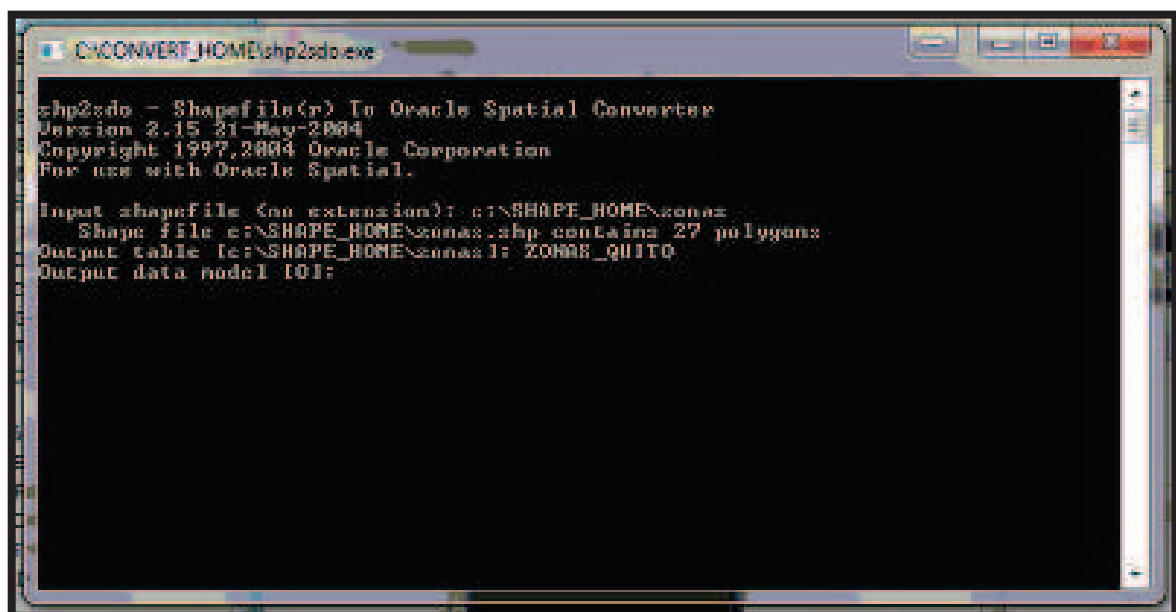
C:\CONVERT_HOME\shp2sdo.exe
shp2sdo - Shapefile(s) To Oracle Spatial Converter
Version 2.15 21-May-2004
Copyright 1997,2004 Oracle Corporation
For use with Oracle Spatial.
Input shapefile (no extension): c:\SHAPE_HOME\zonas
Shape file c:\SHAPE_HOME\zonas.shp contains 27 polygons
Output table (c:\SHAPE_HOME\zonas):

```

Figura 4.21 Leer Archivo Origen shp2sdo

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

- Se define el nombre de la tabla de la base de datos QOS_ZONAS_QUITO (nombre definido en el modelo de la base de datos).



```

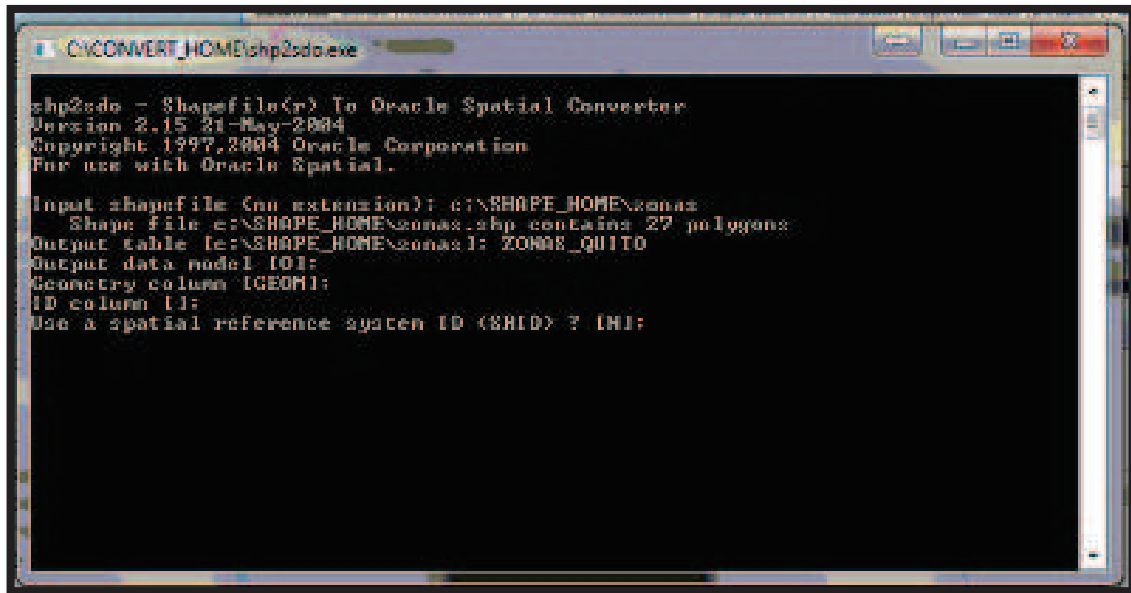
C:\CONVERT_HOME\shp2sdo.exe
shp2sdo - Shapefile(s) To Oracle Spatial Converter
Version 2.15 21-May-2004
Copyright 1997,2004 Oracle Corporation
For use with Oracle Spatial.
Input shapefile (no extension): c:\SHAPE_HOME\zonas
Shape file c:\SHAPE_HOME\zonas.shp contains 27 polygons
Output data model (Q): QOS_ZONAS_QUITO

```

Figura 4.22 Definición de la Tabla que Contendrá los Datos Espaciales - shp2sdo

Fuente: Esquema de Autoría propia (Arévalo D., y Guzmán D.).

- El comando pedirá especificar el modelo a usar, se usará el modelo relacional (es el modelo por defecto), además de definir un nombre para la columna que contendrá los datos espaciales (por defecto COORDENADA), y un nombre para la columna ID (por defecto ID).



```

C:\CONVERT_HOME\shp2sdo.exe
shp2sdo - Shapefile(s) To Oracle Spatial Converter
Version 2.15 31-May-2004
Copyright 1997,2004 Oracle Corporation
For use with Oracle Spatial.

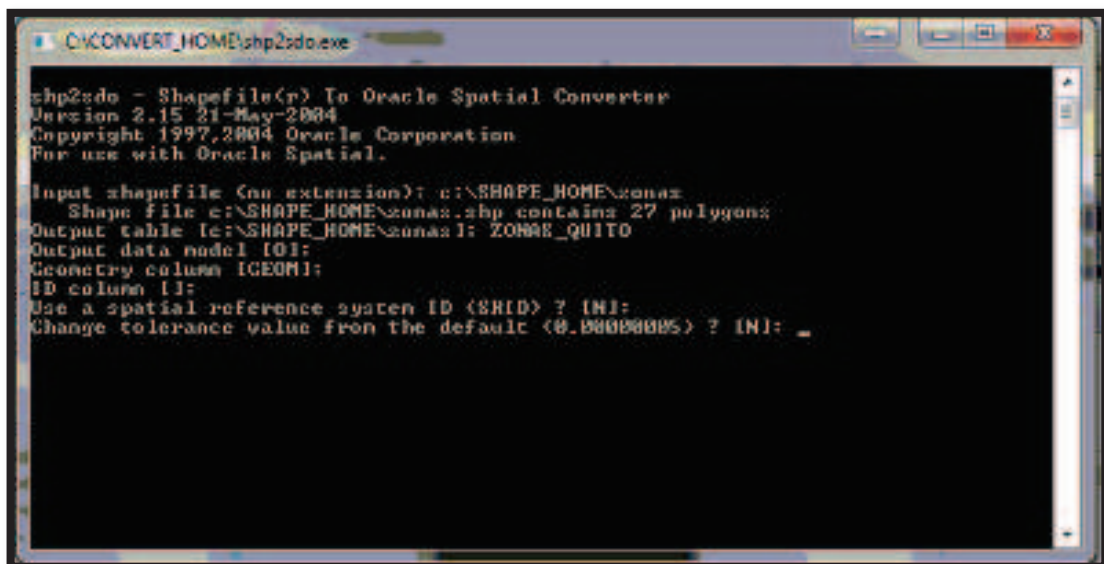
Input shapefile (no extension): c:\SHAPE_HOME\zonas
Shape file c:\SHAPE_HOME\zonas.shp contains 27 polygons
Output table (c:\SHAPE_HOME\zonas): ZONAS_QUITO
Output data model ID:
Geometry column (GEOM):
ID column (I):
Use a spatial reference system ID (SRID) ? (NI):

```

Figura 4.23 Definición de Nombres de Columnas - shp2sdo

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

- A continuación se define un valor de tolerancia igual a 0.00000005.



```

C:\CONVERT_HOME\shp2sdo.exe
shp2sdo - Shapefile(s) To Oracle Spatial Converter
Version 2.15 31-May-2004
Copyright 1997,2004 Oracle Corporation
For use with Oracle Spatial.

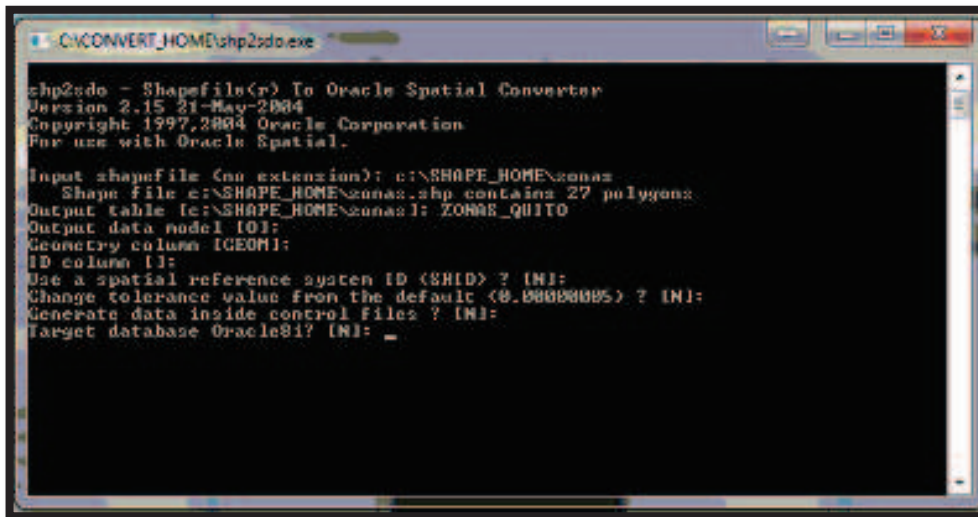
Input shapefile (no extension): c:\SHAPE_HOME\zonas
Shape file c:\SHAPE_HOME\zonas.shp contains 27 polygons
Output table (c:\SHAPE_HOME\zonas): ZONAS_QUITO
Output data model ID:
Geometry column (GEOM):
ID column (I):
Use a spatial reference system ID (SRID) ? (NI):
Change tolerance value from the default (0.00000005) ? (NI): _

```

Figura 4.24 Especificación del Valor de Tolerancia - shp2sdo

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

- La siguiente petición es por si desea exportar hacia un motor de base de datos Oracle 8i, como no es el caso (el sistema maneja versión 10g), simplemente se escoge el valor por defecto que es NO.



```

C:\CONVERT_HOME\shp2sdo.exe

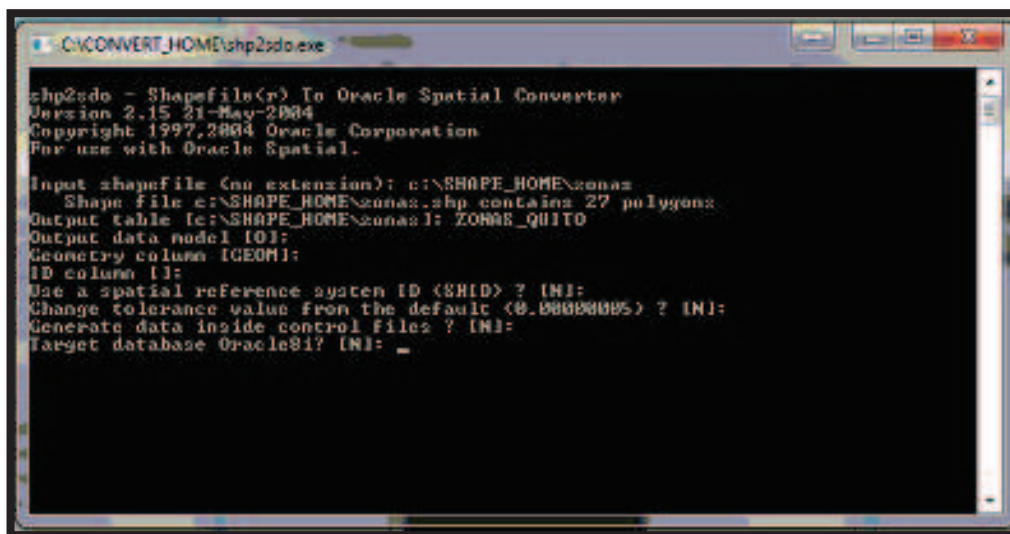
shp2sdo - Shapefile(s) To Oracle Spatial Converter
Version 2.15 21-May-2004
Copyright 1997,2004 Oracle Corporation
For use with Oracle Spatial.

Input shapefile (no extension): c:\SHAPE_HOME\zonas
  Shape file c:\SHAPE_HOME\zonas.shp contains 27 polygons
Output table (c:\SHAPE_HOME\zonas1): ZONAS_QUITO
Output data model (O):
Geometry column (GEOM):
ID column (I):
Use a spatial reference system ID (SRID) ? (N):
Change tolerance value from the default (0.00000005) ? (N):
Generate data inside control files ? (N):
Target database Oracle8i? (N): _
  
```

Figura 4.25 Elección de la Versión de BDD Oracle a exportar - shp2sdo

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

- Para esta clase de conversiones, la precisión es importante por la tanto se debe escoger de al menos seis dígitos (10 dígitos por defecto). Ya que las coordenadas definidas dentro del archivo “shp2sdo” contienen 6 cifras decimales.



```

C:\CONVERT_HOME\shp2sdo.exe

shp2sdo - Shapefile(s) To Oracle Spatial Converter
Version 2.15 21-May-2004
Copyright 1997,2004 Oracle Corporation
For use with Oracle Spatial.

Input shapefile (no extension): c:\SHAPE_HOME\zonas
  Shape file c:\SHAPE_HOME\zonas.shp contains 27 polygons
Output table (c:\SHAPE_HOME\zonas1): ZONAS_QUITO
Output data model (O):
Geometry column (GEOM):
ID column (I):
Use a spatial reference system ID (SRID) ? (N):
Change tolerance value from the default (0.00000005) ? (N):
Generate data inside control files ? (N):
Target database Oracle8i? (N): _
  
```

Figura 4.26 Especificación del Valor de Precisión - shp2sdo

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

- Finalmente, se podrá observar en la carpeta destino 3 archivos creados (Figura 4.27). Un archivo “.sql” mediante el cual contiene la definición de la tabla QOS_ZONAS_QUITO, un archivo “.ctl”, que se utiliza en conjunto con el comando “sqlldr” para cargar los datos que constan en el archivo “.dat” hacia la tabla creada.

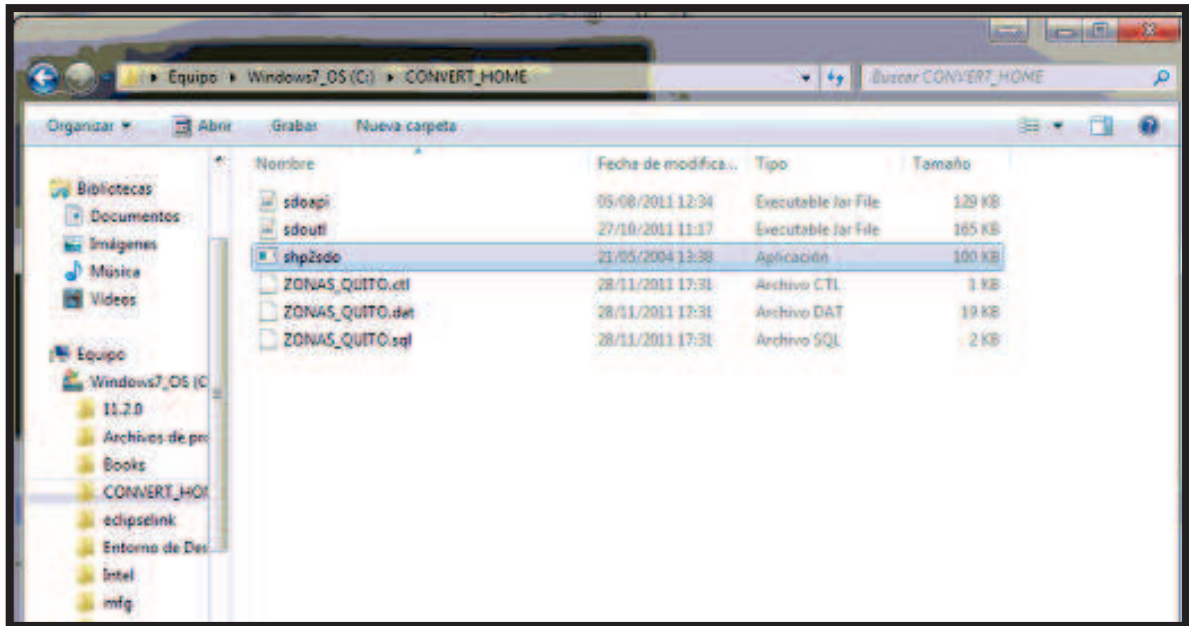
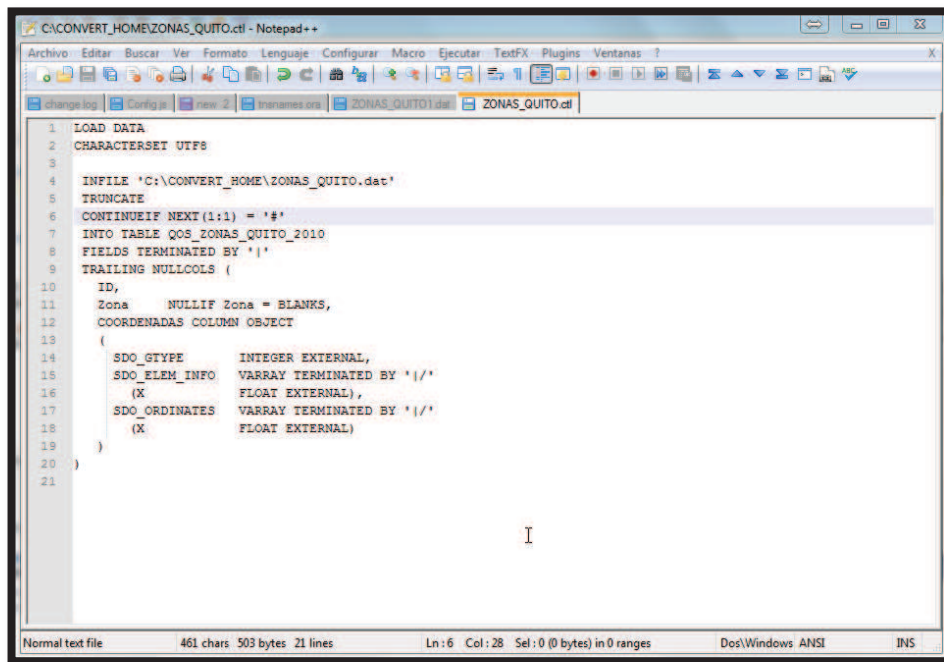


Figura 4.27 Archivos Generados por el Comando shp2sdo

Fuente: Esquema de Autoría propia (Arévalo D., y Guzmán D.).

- Se debe ejecutar las sentencias del archivo “QOS_ZONAS_QUITO.sql” (*script*), para crear la tabla QOS_ZONAS_QUITO.
- Una vez ejecutado el *script* la tabla estará creada, el proceso final es importar los datos con la ayuda del comando “SQLldr”, el archivo “QOS_ZONAS_QUITO.ctl” se encarga de cargar los datos del archivo “zonas_quito.dat” en la tabla creada anteriormente (QOS_ZONAS_QUITO) (Figura 4.28).
- El archivo “ZONAS_QUITO.dat” contiene toda la información de las zonas (id, nombre zona y las coordenadas que la delimitan).



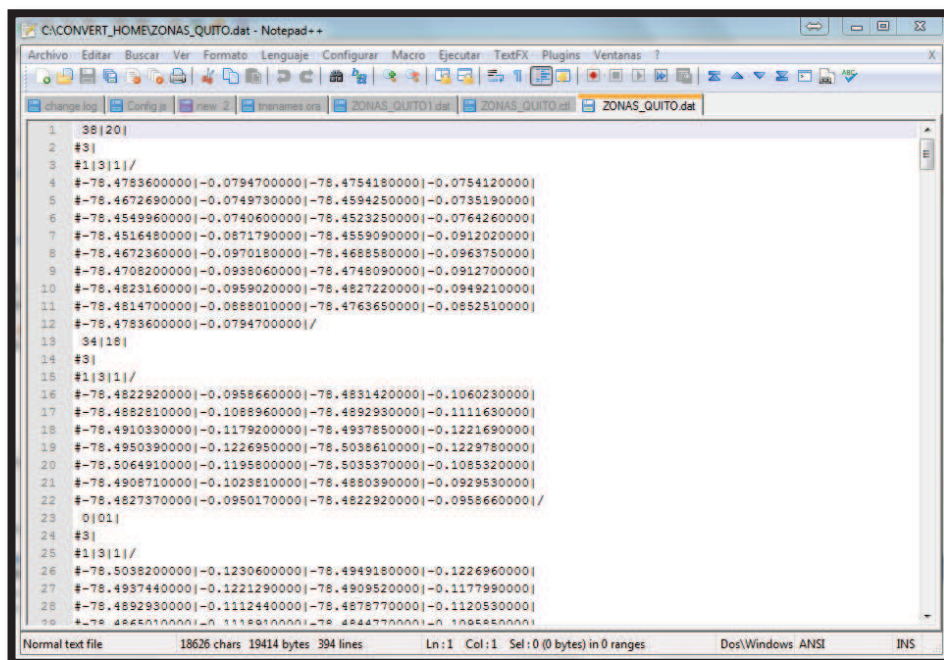
```

1 LOAD DATA
2 CHARACTERSET UTF8
3
4 INFILE 'C:\CONVERT_HOME\ZONAS_QUITO.dat'
5 TRUNCATE
6 CONTINUEIF NEXT (1:1) = '#'
7 INTO TABLE QOS_ZONAS_QUITO_2010
8 FIELDS TERMINATED BY '|'
9 TRAILING NULLCOLS (
10
11 ID,
12 Zona NULLIF Zona = BLANKS,
13 COORDENADAS COLUMN OBJECT
14 (
15 SDO_GTYPE INTEGER EXTERNAL,
16 SDO_ELEM_INFO VARRAY TERMINATED BY '||'
17 (X FLOAT EXTERNAL),
18 SDO_ORDINATES VARRAY TERMINATED BY '||'
19 (X FLOAT EXTERNAL)
20 )
21 )

```

Figura 4.28 Archivo QOS_ZONAS_QUITO.ctf

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).



```

1 #|20|
2 #3|
3 #1|3|1|/
4 #-78.478360000|-0.079470000|-78.475418000|-0.075412000|
5 #-78.467269000|-0.074973000|-78.459425000|-0.073519000|
6 #-78.454996000|-0.074060000|-78.452325000|-0.076426000|
7 #-78.451648000|-0.087179000|-78.455909000|-0.091202000|
8 #-78.467236000|-0.097018000|-78.468858000|-0.096375000|
9 #-78.470820000|-0.093806000|-78.474809000|-0.091270000|
10 #-78.482316000|-0.095902000|-78.482722000|-0.094921000|
11 #-78.481470000|-0.088801000|-78.476365000|-0.085251000|
12 #-78.478360000|-0.079470000|/
13 #4|18|
14 #3|
15 #1|3|1|/
16 #-78.482292000|-0.095866000|-78.483142000|-0.106023000|
17 #-78.488281000|-0.108936000|-78.489293000|-0.111163000|
18 #-78.491039000|-0.117920000|-78.493785000|-0.122169000|
19 #-78.495039000|-0.122695000|-78.503861000|-0.122978000|
20 #-78.506491000|-0.119580000|-78.503537000|-0.108532000|
21 #-78.490871000|-0.102381000|-78.488039000|-0.092953000|
22 #-78.482737000|-0.095017000|-78.482292000|-0.095866000|/
23 0|0|
24 #3|
25 #1|3|1|/
26 #-78.503820000|-0.123060000|-78.494918000|-0.122696000|
27 #-78.493744000|-0.122129000|-78.490952000|-0.117799000|
28 #-78.489293000|-0.111244000|-78.487877000|-0.112053000|
29 #-78.484510000|-0.111891000|-78.484477000|-0.108585000|

```

Figura 4.29 QOS_ZONAS_QUITO.dat

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

- El proceso inicia con la ejecución del comando “*sqlplus*” con las respectivas credenciales (usuario y contraseña de la base de datos).

- Dentro de la terminal, se ejecuta el comando “*sqlldr*” con los parámetros, usuario, contraseña e instancia como en el ejemplo:

sqlldr (usuario/contraseña@instacia base de datos) (path absoluto archivo de control) (path absoluto archivo de log)

```

C:\Windows\system32\CMD.exe - sqlplus david@orcl/david
COLUMN          INPUT          SAVE          WHENEVER SQLERROR
COMPUTE         LIST           SET           XQUERY
CONNECT        PASSWORD      SHOW

SQL> SHUTDOWN
ORA-01031: privilegios insuficientes
SQL> EXIT
Desconectado de Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options
C:\Users\practii_tecno>sqlplus david@orcl/david
SQL*Plus: Release 10.2.0.3.0 - Production on 08b Dic 3 15:58:28 2011
Copyright (c) 1982, 2006, Oracle. All Rights Reserved.

Conectado a:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options
SQL> host sqlldr david/david@orcl c:\CONVERT_HOME\ZONAS_QUITO.ct1 c:\CONVERT_HOME\log1

```

Figura 4.30 Ejecución del Comando *sqlldr*

Fuente: Esquema de Autoría propia (Arévalo D., y Guzmán D.).

Al finalizar la ejecución del comando, se observa el número de registros cargados en la tabla, además se puede revisar el archivo de log, para obtener más detalles del proceso de carga (Figura 4.31).

```

C:\Windows\system32\CMD.exe - sqlplus david@orcl/david
Conectado a:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options
SQL> host sqlldr david/david@orcl c:\CONVERT_HOME\ZONAS_QUITO.ct1 c:\CONVERT_HOME\log1
SQL*Loader: Release 10.2.0.3.0 - Production on 08b Dic 3 15:59:03 2011
Copyright (c) 1982, 2005, Oracle. All rights reserved.

Punto de confirmación alcanzado - recuento de registros lógicos 26
Punto de confirmación alcanzado - recuento de registros lógicos 27
SQL> _

```

Figura 4.31 Fin del Proceso de Carga del Archivo *sqlldr*

Fuente: Esquema de Autoría propia (Arévalo D., y Guzmán D.).

Si hubo éxito en la creación del índice espacial se debe obtener el mensaje “índice creado” (Figura 4.33).

En resumen, el proceso de migración de archivos *shape* hacia un modelo de datos relacional, se describe en la Figura 4.34.



Figura 4.34 Proceso de Migración de Archivos .shp a Oracle 10g

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

4.5 INTEGRACIÓN BASE DE DATOS ORACLE CON WEBLOGIC SEVER 12C

En la consola de administración de Oracle WebLogic 12c, se selecciona la opción “Servicios” y luego “Orígenes de Datos” (de tipo genérico). En la Figura 4.35 se detallan los nombres del nuevo origen de datos.

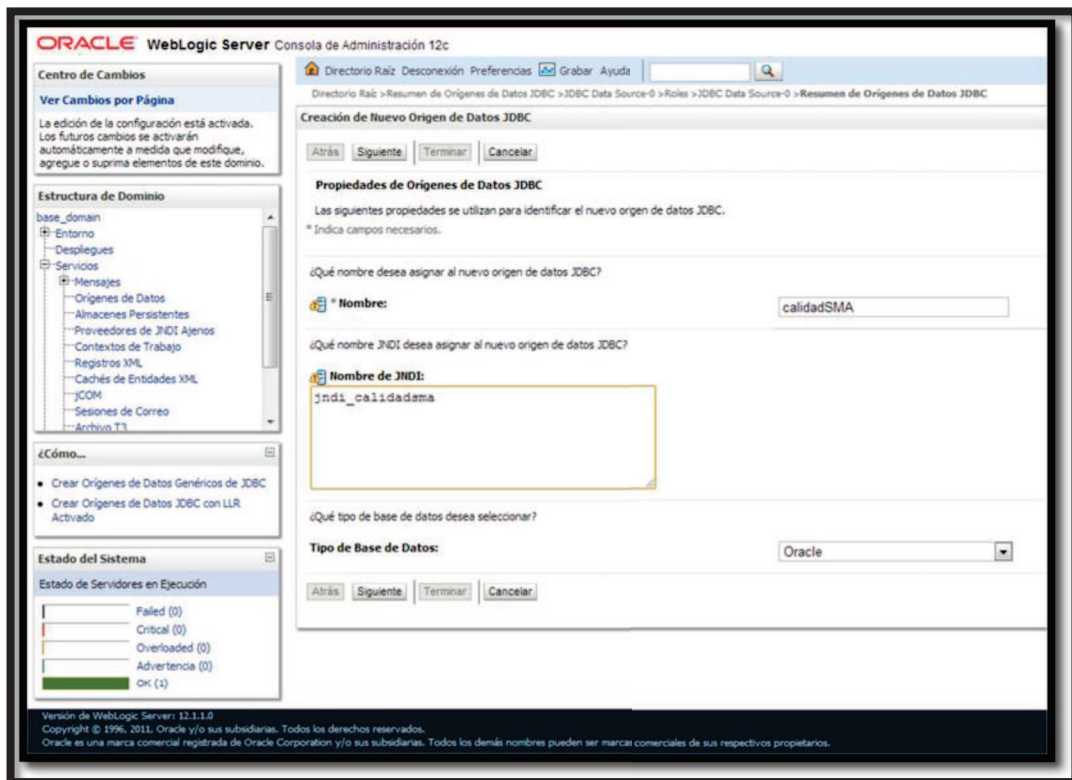


Figura 4.35 Configuración de Origen de Datos Oracle WebLogic 12c

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

Continuando con el proceso se debe definir el controlador para la conexión como se muestra en la Figura 4.36.

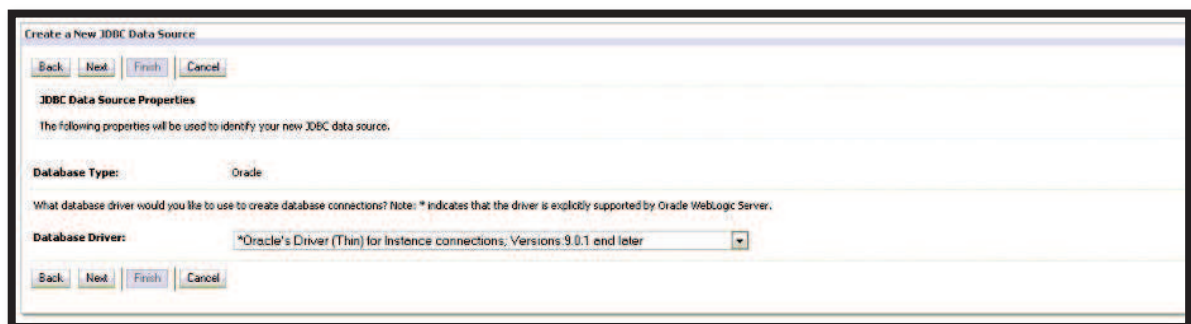


Figura 4.36 Selección de Controlador para la Conexión de Origen de Datos

Fuente: Esquema de Autoría propia (Arévalo D., y Guzmán D.).

También se debe proporcionar la URL de conexión, que para el caso de la aplicación será: `jdbc:oracle:thin:@//172.20.1.82:1521/orcl`. Finalmente, el destino de esta configuración será el servidor "AdminServer" configurado por defecto.

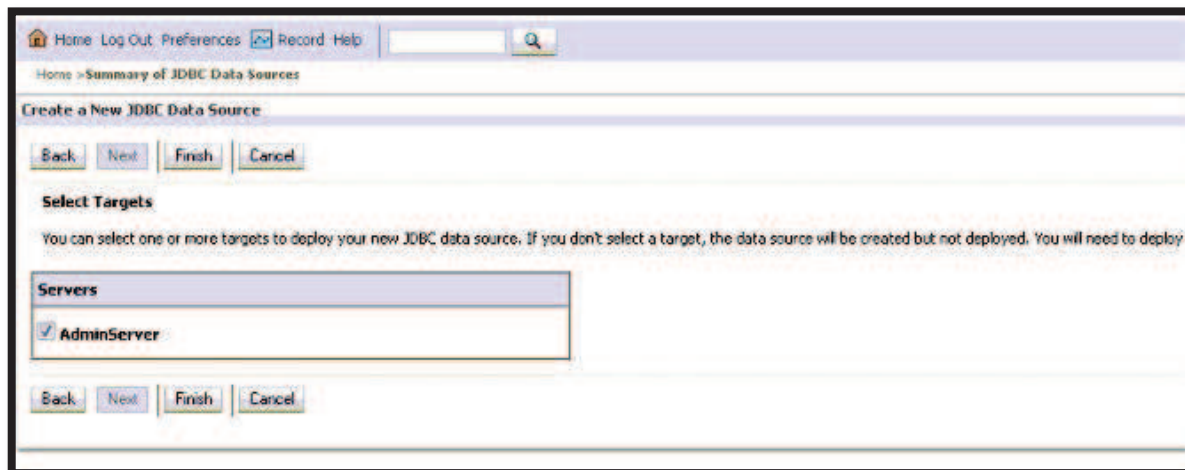


Figura 4.37 Destino de Configuración de Orígenes de Datos

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

4.6 IMPLEMENTACIÓN DE LA APLICACIÓN WEB

Basados en el estándar de desarrollo *Java Enterprise Edition 6* (JEE 6) se ha implementado la interoperabilidad de diferentes sistemas de ingreso y manipulación de información. En la Figura 4.2, se presenta la infraestructura modelada en la Superintendencia de Telecomunicaciones. La implementación está basada en un modelo de capas, donde se puede identificar la capa de persistencia, la capa de negocio y la capa de presentación.

La capa de persistencia está realizada bajo la ayuda de la implementación JPA 2.0 (*Java Persistence API* – API Java para Persistencia), este módulo de la aplicación está administrado por el servidor de aplicaciones WebLogic. El estándar EJB 3.1 (*Enterprise Java Beans* – Beans Java Empresariales) ayuda en el desarrollo de la capa de negocios. Finalmente, la capa de presentación está desarrollada con módulos del *framework* Spring 3.1. Se ha utilizado los módulos para publicación de métodos y funciones web a través de los conceptos de “*servlets*”, que básicamente constituye una manera de ejecutar código en el entorno del servidor.

Se ha desintegrado la aplicación en varios módulos, presentando además una capa de servicios web implementada a través de JAX-WS (*Java API for XML Web Services* – API Java para Servicios Web XML).

La capa de presentación se la ha representado a través de la programación de controladores del lado del servidor, publicados como “*servlets*” haciendo uso del *framework* Spring 3.1, y de vistas JSP (*Java Server Pages* – Páginas Java de Servidor), mismas que son contenedores de código JavaScript, hojas de estilos y recursos externos como librerías, imágenes y controles.

El concepto del *framework Spring Web MVC*, basado en un despachador a nivel de servidor que administra las peticiones y respuestas de los clientes web, nos permite extender esta funcionalidad a cualquier objeto previamente configurado (controladores, objetos que extienden la interfaz *MultiActionController*); así logrando publicar métodos y funciones que pueden ser ejecutados a través de llamadas HTTP del tipo *POST* y *GET*.

Las páginas JSP, para nuestros propósitos no se están usando para crear “*servlets*” y tampoco para construir código Java para el lado del servidor. Las páginas JSP en la aplicación implementada son simplemente contenedores de código JavaScript y etiquetas HTML.

Las librerías JavaScript que se utiliza para el manejo de mapas son *OpenLayers*, *GeoExt* y *Proj4JS* integrándolas con controles de escritorio ofrecidos por *ExtJS*. *OpenLayers* ofrece funcionalidades y controles que permiten hacer uso de servicios WMS (*Web Map Services* – Servicios Web de Mapas), permitiendo el manejo de objetos espaciales, tales como mapas, contenedores de mapas, capas, controles y eventos sobre los mapas, además de proyecciones.

GeoExt permite integrar controles *ExtJS*, como cajas de texto, botones y paneles, con funcionalidades del *framework OpenLayers*. Las librerías *Proj4JS*, nos permiten utilizar funcionalidades de conversión de coordenadas entre diferentes tipos de sistemas de referencia. Finalmente se utiliza controles estilizados *ExtJS*, mismos que nos permiten programar interfaces de usuario similares a las utilizadas en entornos de escritorio.

En la generación de reportes se ha utilizado librerías *JasperReports*, las cuales permiten generar archivos *PDF* tomando como fuente de información objetos Java. El *framework* hace uso de archivos de configuración de extensión “*jrxml*”,

contenedores del diseño, paneles, tablas, imágenes, grillas en lenguaje de marcado, que son interpretados por código fuente de las librerías *JasperReports*.

La seguridad de la aplicación ha sido implementada para el módulo de administración, integrando la plataforma web con un servidor SSO (*Single Sign-On* – Una Sola Entrada), para autorizar la autenticación del usuario. La fuente de información acerca de grupos, usuarios y roles de usuario se encuentra centralizada en un servidor LDAP (*Lightweight Directory Access Protocol* - Protocolo Ligero de Acceso a Directorios).

El servidor SSO no es parte de esta aplicación, es parte de la aplicación la integración del servicio haciendo uso del *framework Spring Web Security*.

4.6.1 IMPLEMENTACIÓN DE LA CAPA DE PERSISTENCIA

Se ha implementado la capa de persistencia en base a un proyecto JPA del IDE de programación Eclipse, este proyecto representa la solución al paradigma conocido como ORM (*Object-Relational Mapping* – Mapeo de Objetos a Bases de Datos Relacionales). Permite la interacción de la base de datos relacional a través de objetos Java conocidos como POJOs (*Plain Old Java Object* – Objeto Plano Java), a través de clases contenidas en el paquete “*javax.persistence*”.

4.6.1.1 Estructura del Proyecto JPA

El proyecto está constituido por entidades, unidad de persistencia, contexto de persistencia y un administrador de entidades (Figura 4.38).

4.6.1.1.1 Unidad de Persistencia

La unidad de persistencia, Espacio de Código 4.7, define el ámbito de trabajo de las entidades. Está determinado en el archivo de configuración *persistence.xml*, en el que se especifica el proveedor de conexiones, las fuentes de datos, los mapeos a la base de datos y la localización de las entidades.

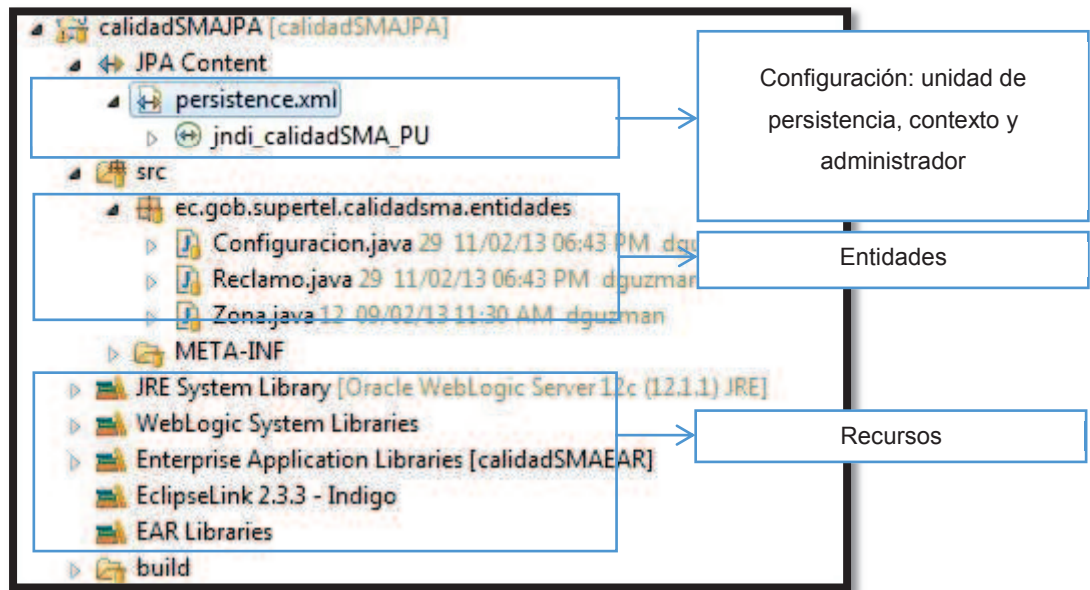
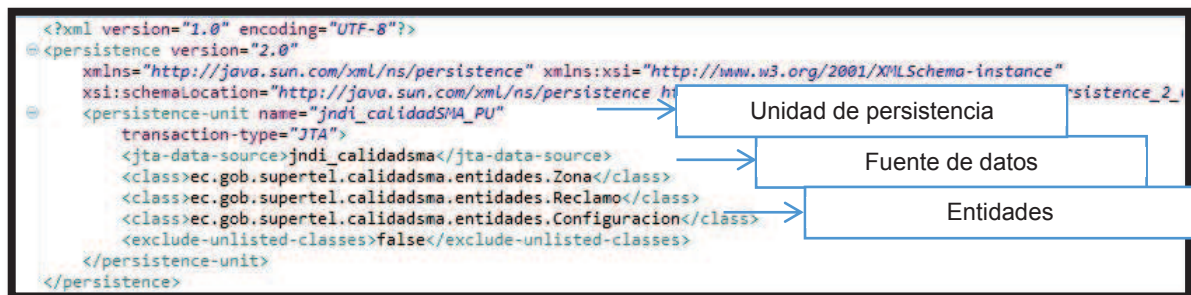


Figura 4.38 Estructura del Proyecto de Persistencia a la Base de Datos

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).



Espacio de Código 4.7 Detalle de Archivo de Configuración persistence.xml

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

4.6.1.1.2 Entidades

Objeto Java previamente configurado a través de anotaciones, que permiten la interacción con la base de datos relacional, mapeando tablas, secuencias, columnas, índices y vistas a clases Java (Figura 4.39).

4.6.1.1.3 Contexto de Persistencia

Es el conjunto de entidades que van a ser manipuladas por un administrador de entidades, *EntityManager*.

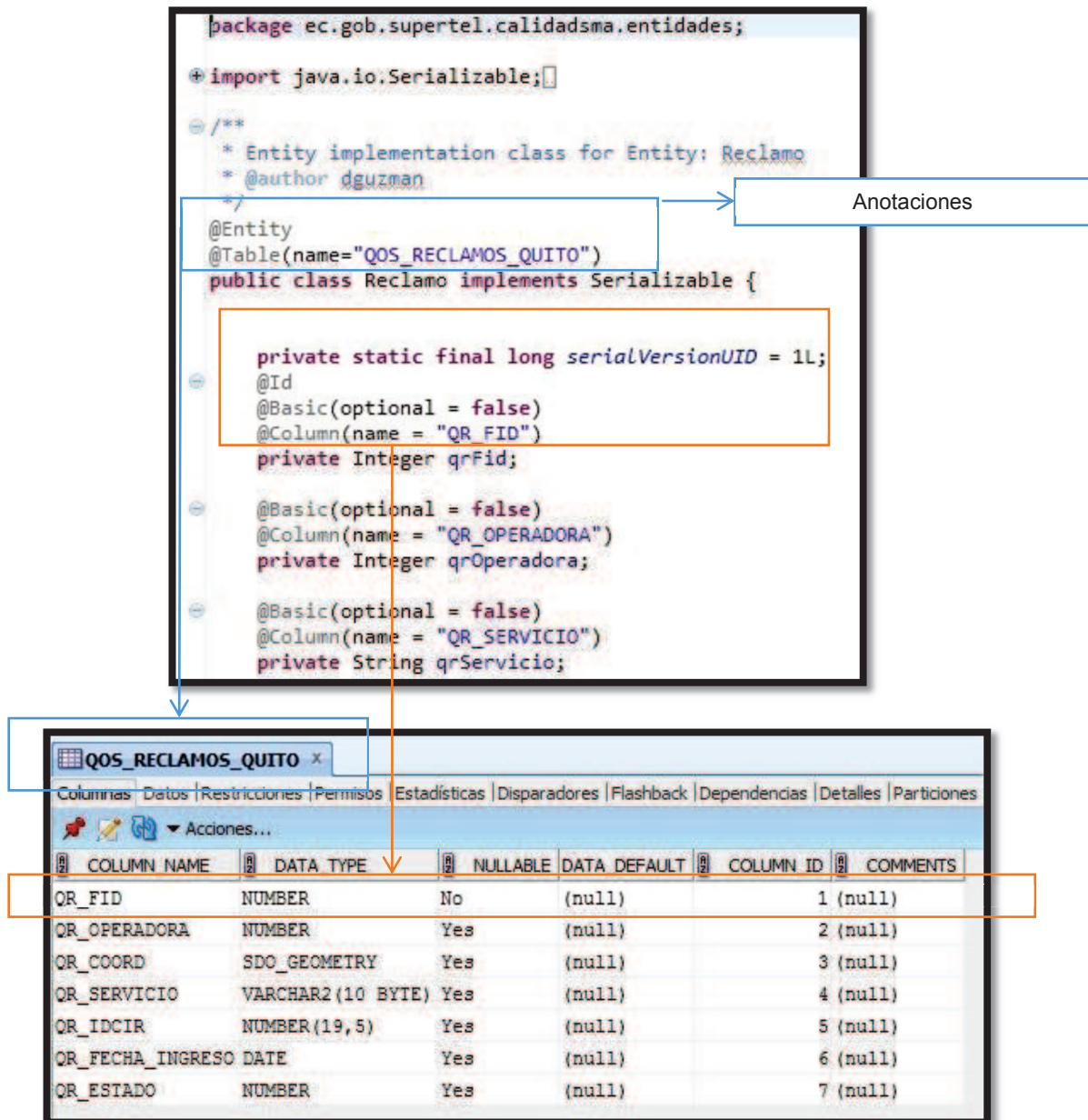


Figura 4.39 Representación de Clases Entidades

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

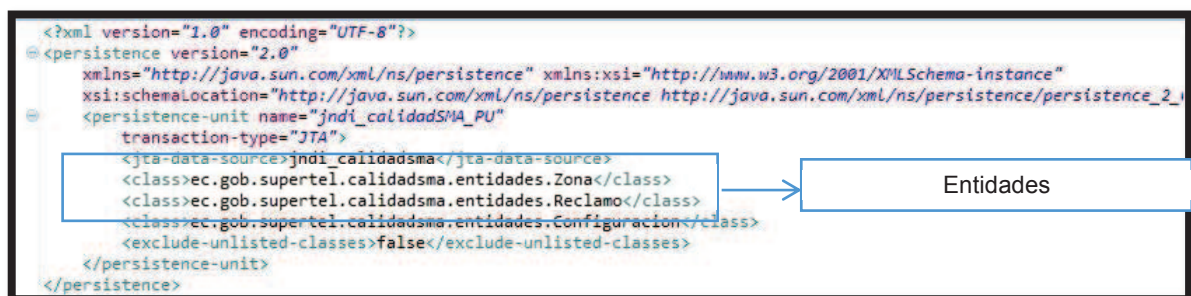
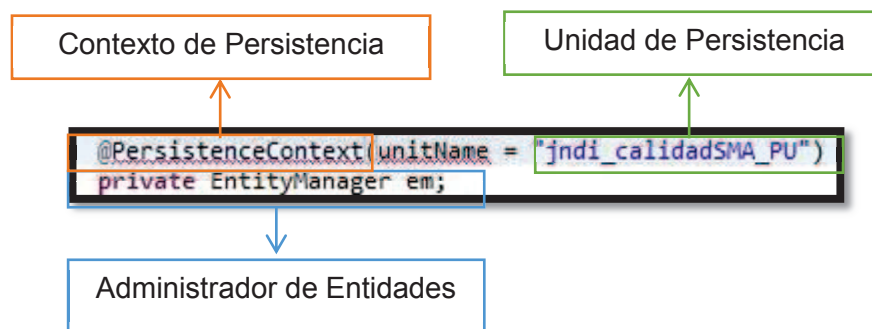


Figura 4.40 Detalle de Configuración de Entidades

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

4.6.1.1.4 Administrador de Entidades

Es el objeto encargado de almacenar y cargar las entidades del almacén; así también, de la utilización de funcionalidades como eliminación y creación de entidades, mismos cambios sobre el contexto de entidades se envía o replica a la base de datos. Este contenedor instancia los objetos necesarios para conectarse a la base de datos, a través de búsquedas JNDI⁵⁰ (*Java Naming and Directory Interface* - Interfaz de Programación de Aplicaciones (API) de Java para servicios de directorio) funcionalidad conocida como inyección de objetos.



Espacio de Código 4.8 Configuración de Clase Administrador de Entidades

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

La interfaz de administración, *EntityManager*, define los métodos que serán utilizados para interactuar con el contexto, *PersistenceContext*. El contexto como conjunto de entidades a ser administradas por la interfaz de administración está definido por la unidad de persistencia, *PersistenceUnit*; pudiendo la interfaz definir métodos como los detallados en la Figura 4.41.

```
em.persist(configuracion);      em.find(Configuracion.class, qcId);
```

Figura 4.41 Ejemplo de Métodos de la Interfaz *EntityManager*

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

4.6.2 IMPLEMENTACIÓN CAPA DE NEGOCIO

Haciendo hincapié en la definición de EJB's (*Enterprise Java Beans*) del tipo sesión, *Session Beans*, se utiliza la definición de EJB's del tipo entidad construidos en lo que se conoce como capa de persistencia. Esto quiere decir que

⁵⁰ Permite a los clientes descubrir y buscar objetos y datos a través de un nombre.

la especificación Java EJB incluye el detalle de objetos del tipo sesión (*Session Bean*) y objetos del tipo entidad (*Entity Bean*).

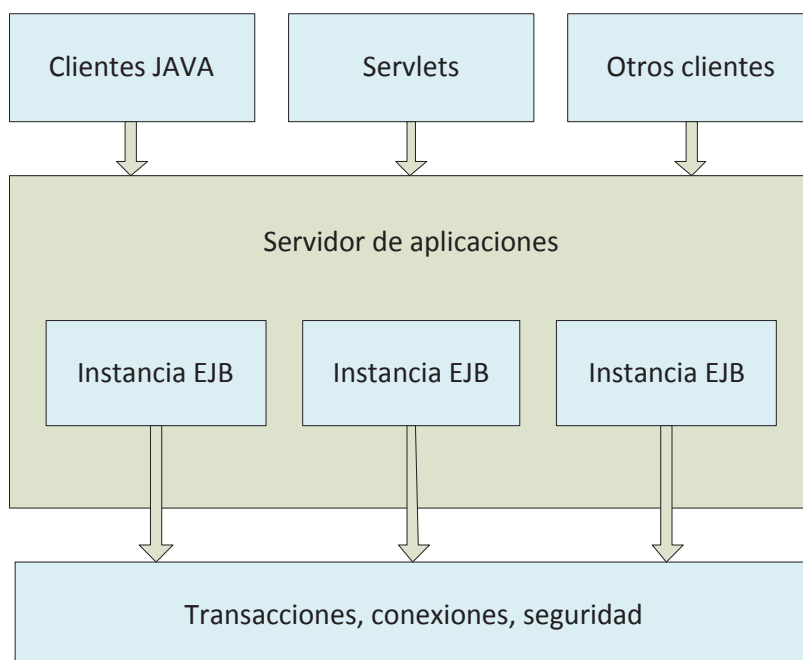


Figura 4.42 Funcionalidad de Clases EJB

Fuente: Esquema de Autoría propia (Arévalo D., y Guzmán D.).

4.6.2.1 Estructura del proyecto EJB

El proyecto está constituido por objetos EJB del tipo sesión con su respectiva definición en interfaces (Figura 4.43).

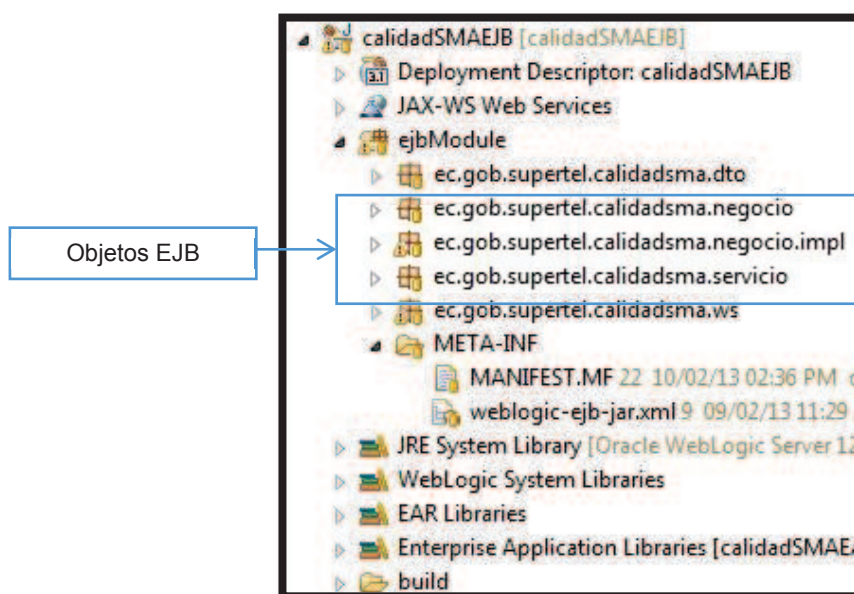


Figura 4.43 Paquetes Contenedores de Clases EJB

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

El acceso a la funcionalidad programada en un objeto EJB se lo realiza a través de definiciones de interfaces, como se esquematiza en la Figura 4.44.

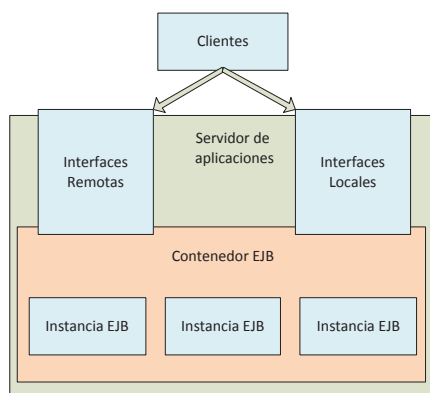


Figura 4.44 Definición de Interfaces EJB

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

4.6.2.1.1 Definición de Objetos de Sesión

Las interfaces están definidas en el paquete: `ec.gob.supertel.calidadsma.negocio`.

```

+ /**
package ec.gob.supertel.calidadsma.negocio;

+ import java.util.Calendar;

- /**
 * @ Interface que contiene los metodos que se utilizan en el negocio de Planificacion
 * @author dguzman
 * @version $Revision: 1.0 $
 * @date 27/11/2012
 */
@Local ← Definición de Interfaz Local
public interface ICalidadSMANegocio {

- /**
 * Metodo que busca una lista de zonas.
 * @return Lista de Zona
 */
List<Zona> buscarZonas(String ciudad); ← Definición métodos

- /**
 * Metodo que busca una zona por id
 * @return Lista de Zona
 */
Zona buscarZona(Integer gdId);

- /**
 * Metodo que actualiza el estado de la zona
 * @return Lista de Zona
 */
Boolean actualizarZona(Zona zona);

```

Figura 4.45 Ejemplo de Objetos de Sesión EJB

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

4.6.2.1.2 Implementación de Objetos de Sesión

Las implementaciones de las interfaces están definidas en el paquete: `ec.gob.supertel.calidadsma.negocio.impl`.

```

+ /**
package ec.gob.supertel.calidadsma.negocio.impl;

+ import java.sql.Date;

- /**
 * @author dguzman
 *
 */
@Stateless ← Definición de Objeto de Sesión del tipo Stateless
@TransactionManagement(TransactionManagementType.CONTAINER)
public class CalidadSMANegocio implements ICalidadSMANegocio {

    public static final Log LOG = new Log(CalidadSMANegocio.class);
    //unidad que utiliza la configuracion JTA de weblogic
    @PersistenceContext(unitName = "jndi_calidadSMA_PU")
    private EntityManager em;

    |
    //unidad utilizada con configuracion de recursos locales
    @PersistenceUnit(unitName = "calidadSMAJPALocalR")
    private EntityManagerFactory factory;

    -
    /*
     * (non-Javadoc)
     *
     * @see
     * ec.gob.supertel.calidadsma.negocio.ICalidadSMANegocio#buscarZonas(java
     * .lang.String)
     */
    @SuppressWarnings("unchecked")
    @Override
    public List<Zona> buscarZonas(String ciudad) {
        List<Zona> zonas = null;
        try {
            String select = "SELECT c FROM Zona c WHERE 1=1";

            if (ciudad != null && !ciudad.isEmpty()) {
                select += " AND upper(c.gdCodCan) like upper('%"
                    + ciudad.trim() + "%)";
            } else
                select += " ORDER BY c.gdId DESC";

            zonas = em.createQuery(select).getResultList();

        } catch (Exception e) {
            LOG.println("Error en la consulta buscarZonas"
                + e.getLocalizedMessage());
            return null;
        }
        return zonas;
    }
}

```

Implementación

Espacio de Código 4.9 Ejemplo de Implementación de Interfaz EJB

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

Se ha utilizado objetos de sesión del tipo sin estado, *Stateless*, debido a que este tipo de instancias no almacenan los estados de ejecución conforme se van instanciando y utilizando los métodos programados.

4.6.3 IMPLEMENTACIÓN DE SERVICIOS WEB

Para ofrecer interoperabilidad entre tecnologías se ha implementado servicios Web los cuales ofrecen una manera de comunicarse a través de mensajes escritos en lenguaje de marcado.

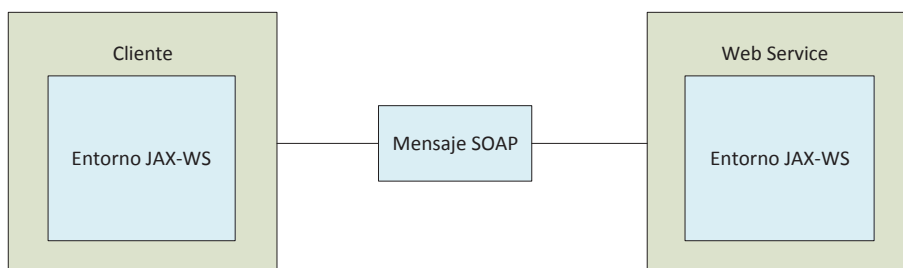


Figura 4.46 Funcionamiento de Entornos JAX-WS

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

4.6.3.1 Estructura del Proyecto EJB

Se puede identificar el paquete: `ec.gob.supertel.calidadsma.ws` (Figura 4.47). Paquete contenedor de las clases que se publicarán a manera de servicios web.

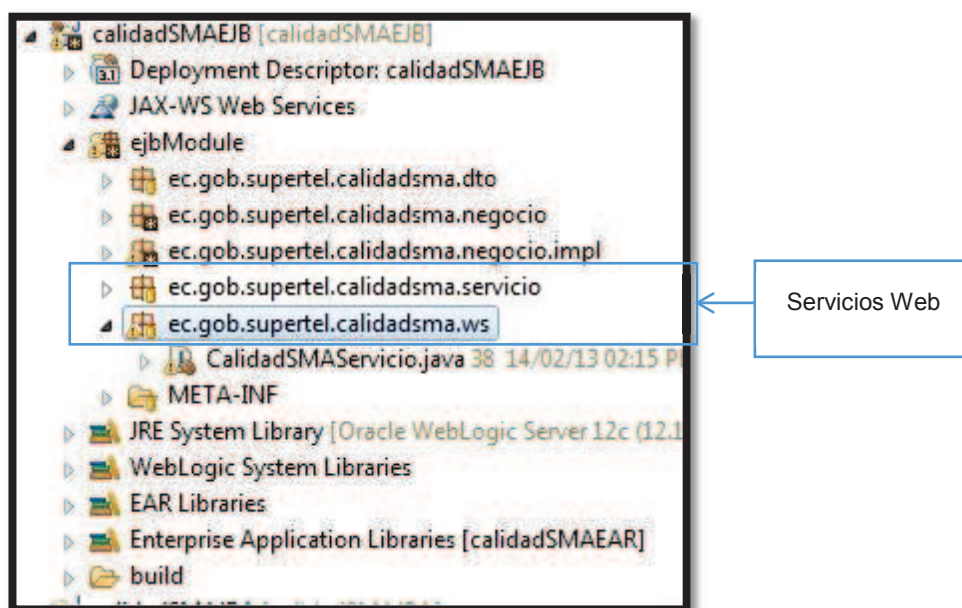


Figura 4.47 Paquete Contenedor de Servicios Web

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

4.6.3.2 Java API for XML Web Services – API Java para Servicios Web XML (JAX-WS)

El paquete que añade esta funcionalidad está definido en el espacio de nombres: `javax.jws`.

```

+ /**
package ec.gob.supertel.calidadsma.ws;

+ import java.util.ArrayList;

- /**
 * @author dguzman
 *
 */
+ @Stateless(mappedName = "calidadSMAService")
+ @WebService
public class CalidadSMAService implements ICalidadSMAService {

    public static final Log LOG = new Log(CalidadSMAService.class);
    @EJB
    public ICalidadSMANegocio calidadSMANegocio;

    /**
     * (non-Javadoc)
     *
     * @see
     * ec.gob.supertel.calidadsma.servicio.ICalidadSMAService#buscarZonas(java
     * .lang.String)
     */
    + @WebMethod
    public @WebResult(name = "zonas")
    List<Zona> buscarZonas(@WebParam(name = "ciudad") String ciudad) {
        List<Zona> listaRetorno = null;
        try {
            listaRetorno = calidadSMANegocio.buscarZonas(ciudad);
        } catch (Exception e) {
            LOG.println("Error @WebService CalidadSMAService @WebMethod buscarZonas/n"+e.getLocalizedMessage());
            return null;
        }
        return listaRetorno;
    }
}

```

The image shows a code editor with several callout boxes pointing to specific parts of the code:

- Definición de Servicio Web:** Points to the `@WebService` annotation.
- Definición de Método Web:** Points to the `@WebMethod` annotation.
- Definición de Parámetros Web:** Points to the `@WebParam` annotation.

Espacio de Código 4.10 Ejemplo de Clase Representando un Servicio Web

Fuente: Esquema de Autoría propia (Arévalo D., y Guzmán D.).

Como se observa en el Espacio de Código 4.10, haciendo uso de las notaciones estandarizadas para la especificación Java se puede dar la funcionalidad de publicación de métodos y parámetros a los objetos de sesión comunes.

4.6.3.3 Archivo WSDL (Lenguaje de Descripción de Servicios Web)

Archivo en lenguaje de marcado que especifica los métodos y modelos de objetos que serán publicados; así también, el modelo de mensaje que intercambiarán para el envío y recepción de objetos y parámetros que se procesarán en las respectivas clases.

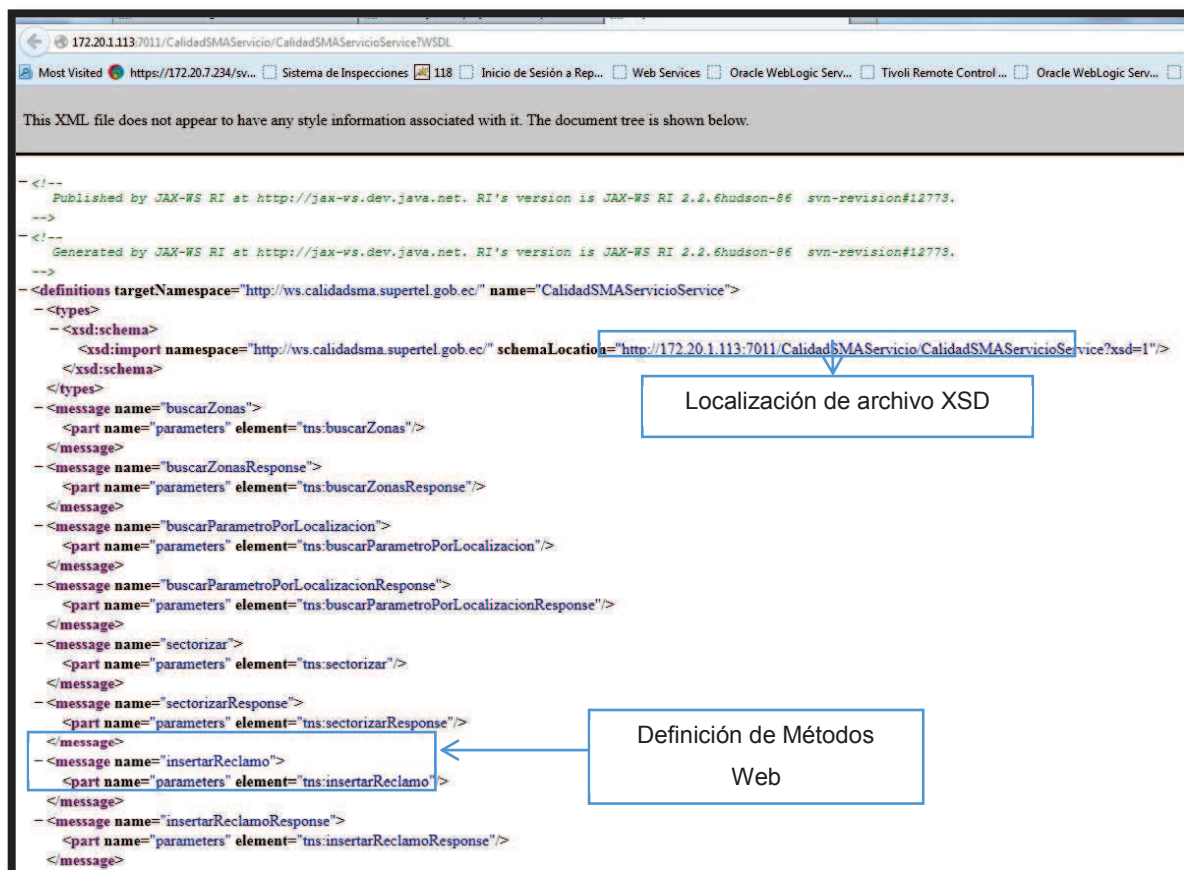


Figura 4.48 Ejemplo de Archivo WSDL Publicado

Fuente: Esquema de Autoría propia (Arévalo D., y Guzmán D.).

Para especificar el modelo de objetos que interactuarán en la ejecución de un servicio web se utiliza un archivo XSD⁵¹ (*XML Schema Definition* – Definición del Esquema XML). En la Figura 4.49, se ejemplifica la forma de definir clases utilizando lenguaje de marcado.

4.6.4 IMPLEMENTACIÓN CAPA DE PRESENTACIÓN

La capa de presentación para la implementación está totalmente desintegrada de la lógica de negocio, presentándose en un proyecto web JEE (*Java Enterprise Edition*) como se puede observar en la Figura 4.50.

⁵¹ Lenguaje utilizado para describir la estructura y las validaciones de los contenidos de los documentos XML.

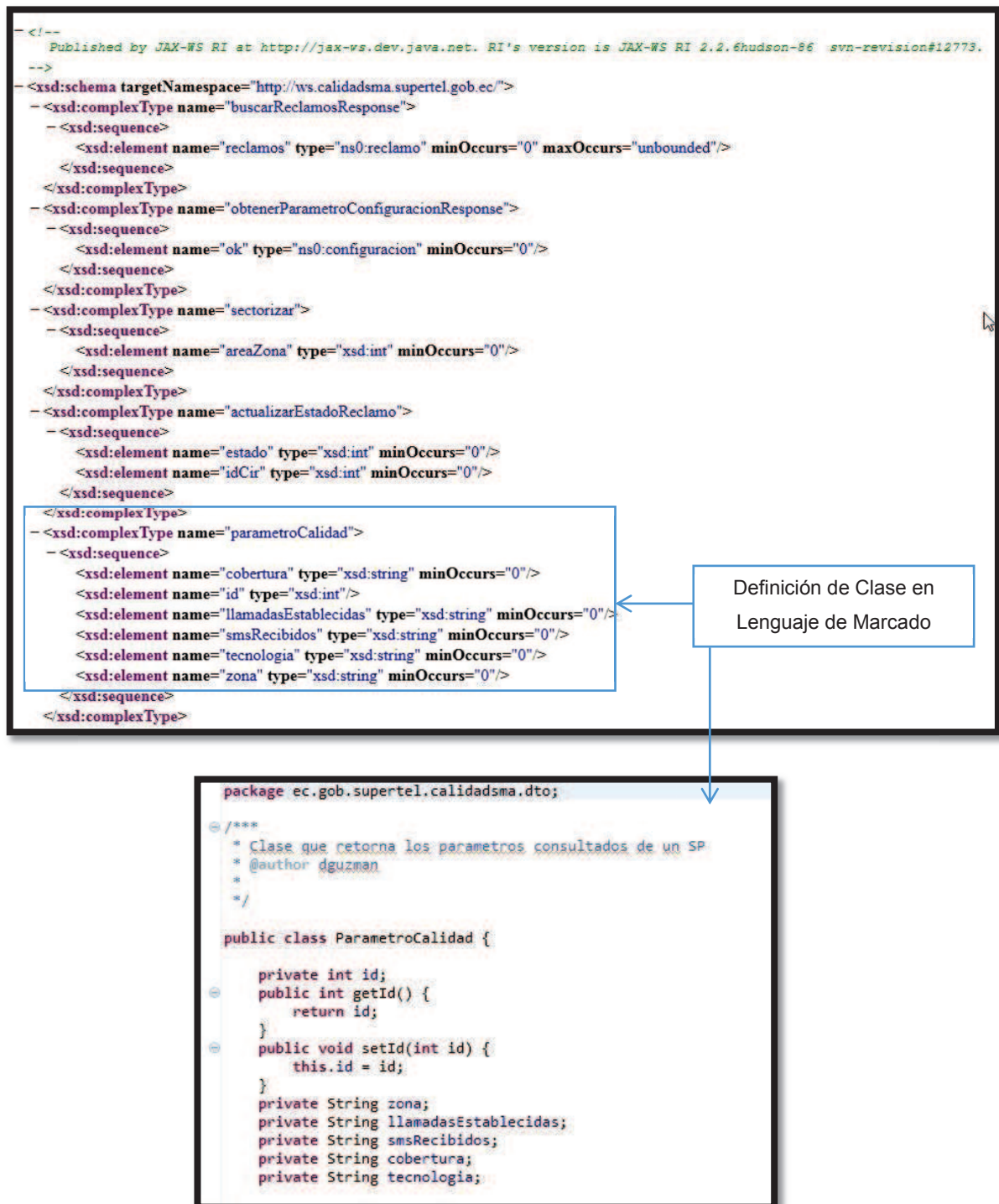


Figura 4.49 Especificación de Clase en Lenguaje de Marcado

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

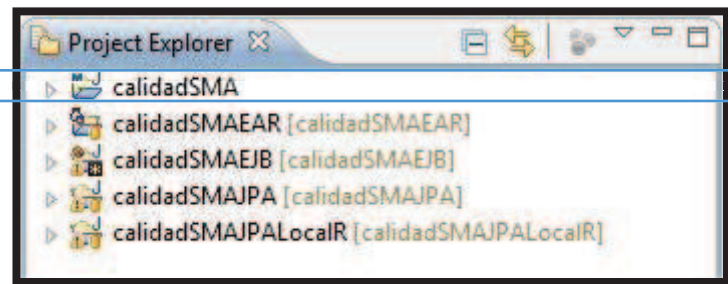


Figura 4.50 Proyecto Web Java JEE

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

Este proyecto está configurado como un proyecto *Maven*, esto debido a los beneficios en cuanto a administración de proyectos que ofrece la herramienta, de entre los cuales se puede recalcar administración de versiones, compilados, distribuciones y dependencias.

4.6.4.1 Estructura del Proyecto Web JEE

En la Figura 4.51 se puede observar la estructura de una aplicación web JEE gestionada por *Maven*. En donde se identifica la estructura del código fuente en el directorio “src”, un archivo de configuración del proyecto “pom.xml” y un directorio “target” donde se ejecutarán las compilaciones y ejecuciones de prueba.

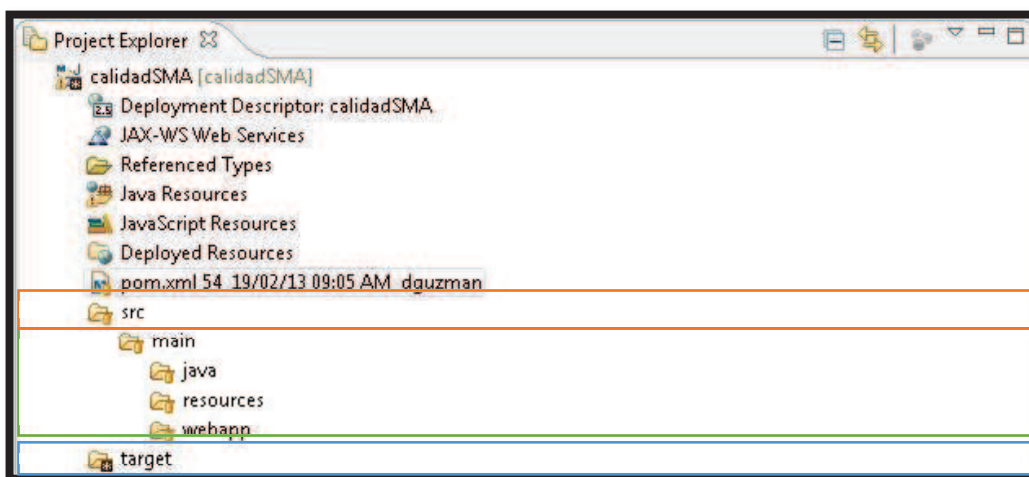


Figura 4.51 Estructura del Proyecto Web Java JEE

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

4.6.4.1.1 Archivo de configuración

El archivo de configuración contiene información acerca de la compilación que realizará el código fuente del proyecto, estructurándose a través de lenguaje de

marcado (XML) e identificando la configuración general del proyecto (versión, identificador de la aplicación, nombre de la aplicación, etc.), Las dependencias del proyecto, especificaciones de la construcción del proyecto, propiedades de la compilación y los repositorios de las dependencias del proyecto.

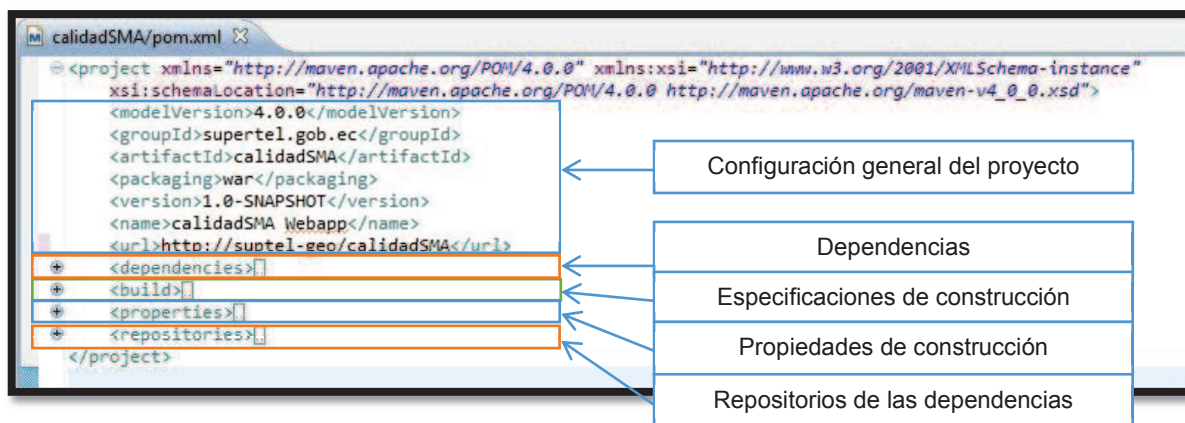


Figura 4.52 Detalle de Configuración del Archivo pom.xml

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

➤ Dependencias

Las dependencias, código fuente (archivos .jar), son administradas en un repositorio centralizado creado en los clientes mismo que lleva la estructura con la que se nombra la dependencia, mediante un identificador de grupo (*groupId*), identificador de artefacto (*artifactId*) y versión de la dependencia (*versionId*).



Figura 4.53 Detalle de la Sección de Configuración de Dependencias

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

Las dependencias especificadas en el archivo de configuración guardan una estructura de acuerdo a sus identificadores, almacenándose en el directorio “.m2” del disco local (Figura 4.54).

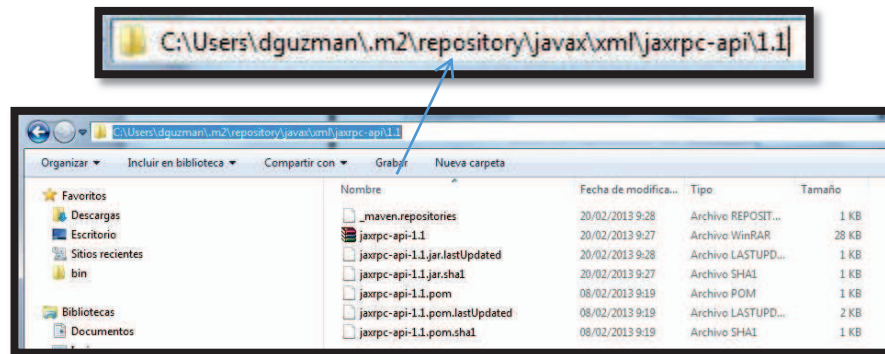


Figura 4.54 Demostración de Almacenamiento del Repositorio Local

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

➤ Construcción

Esta sección del archivo de configuración permite añadir complementos para la ejecución del proyecto como compiladores, contenedores y servidores de depuración.

```

<build>
  <plugins>
    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-compiler-plugin</artifactId>
      <version>2.0.2</version>
      <configuration>
        <source>1.5</source>
        <target>1.5</target>
      </configuration>
    </plugin>
    <plugin>
      <groupId>org.codehaus.mojo</groupId>
      <artifactId>tomcat-maven-plugin</artifactId>
      <configuration>
        <url>http://localhost:8080/manager</url>
        <server>mytomcat</server>
        <path>/calidadSMA</path>
      </configuration>
    </plugin>
    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-eclipse-plugin</artifactId>
      <inherited>true</inherited>
      <configuration>
        <classpathContainers>
          <classpathContainer>org.eclipse.jdt.launching.JRE_CONTAINER</classpathContainer>
          <classpathContainer>org.eclipse.jdt.USER_LIBRARY/TOMCAT_6.0.14_LIBRARY</classpathContainer>
        </classpathContainers>
      </configuration>
    </plugin>
  </plugins>
  <finalName>calidadSMA</finalName>
</build>

```

Figura 4.55 Detalle de la Sección Construcción del Archivo de Configuración

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

➤ Repositorios

Existen dos maneras de referenciar dependencias, la primera a un repositorio local (explicado en la sección anterior) y la segunda a repositorios en la web; ésta última ejemplificada en la Figura 4.56.

```

<repositories>
  <repository>
    <url>http://download.java.net/maven/1</url>
    <id>jsf12</id>
    <layout>legacy</layout>
    <name>Repository for library Library[jsf12]</name>
  </repository>
</repositories>

```

Figura 4.56 Detalle de la Configuración de la Sección Repositorios

Fuente: Esquema de Autoría propia (Arévalo D., y Guzmán D.).

El directorio “*src/main*” contiene todas las clases necesarias para la ejecución de la aplicación web. Mismo directorio de código fuente puede ser observado organizado a manera de paquetes en la carpeta “*Java Resources*”.

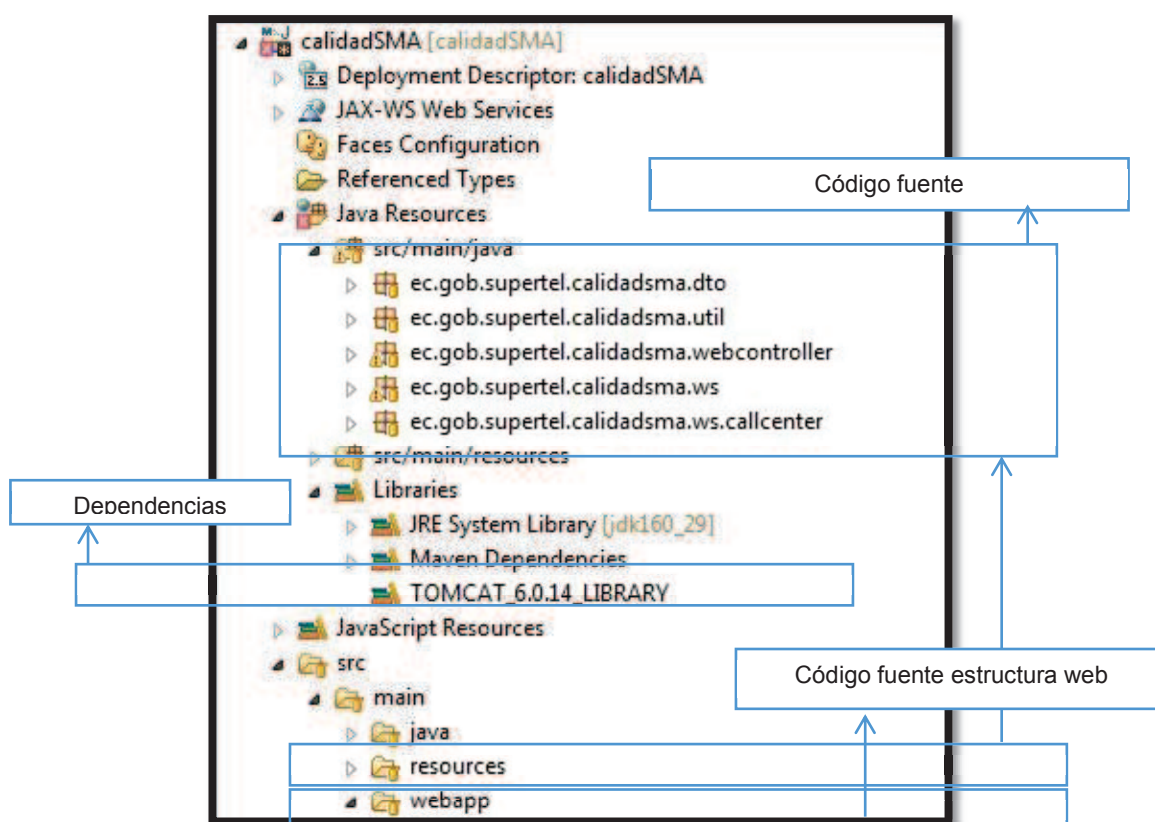


Figura 4.57 Detalle de la Estructura de Código Fuente de la Aplicación Web

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

4.6.4.1.2 Directorio de Depuración

Utilizada para generar la estructura de la aplicación en una ubicación donde se puede realizar pruebas y depuración de código.



Figura 4.58 Detalle del Directorio de Depuración

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

4.6.4.1.3 Controladores

Se ha implementado clases que gestionan las peticiones de los clientes a través del procesamiento de solicitudes del tipo HTTP/POST y respuestas a los clientes a través del protocolo de comunicación HTTP.

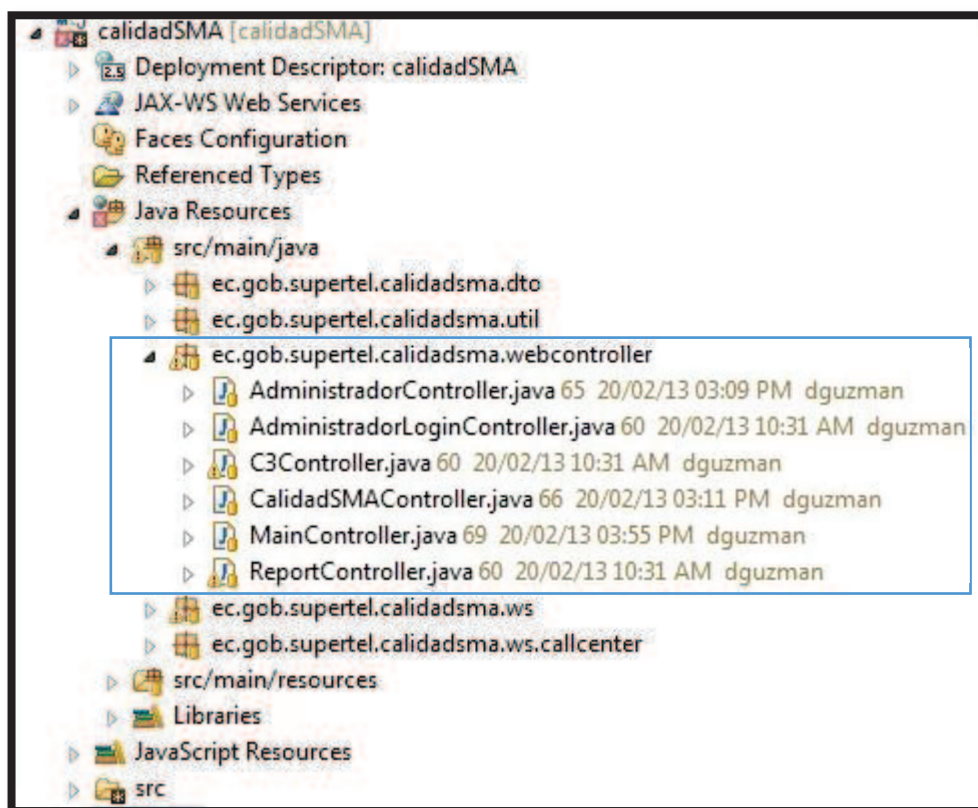


Figura 4.59 Detalle del Paquete Contenedor de las Clases Controlador

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

Como se puede observar, existen clases específicas para el manejo de las diferentes funcionalidades del sistema. En la Figura 4.60, se presenta un resumen de las funcionalidades implementadas en cada clase.



Figura 4.60 Detalle de Funcionalidades Implementadas en las Clases Controlador

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

Los controladores están implementados bajo los conceptos de Spring MVC, haciendo que cada uno de estos objetos programados herede a la clase “org.springframework.web.servlet.mvc.MultiActionController” que oculta los procesos de publicación de métodos en “*servlets*”.

```

package ec.gob.supertel.calidadsma.webcontroller;

import java.io.PrintWriter;

public class AdministradorController extends MultiActionController {
    private Logger logger = Logger.getLogger(this.getClass());
    CalidadSMAServicioProxy proxyCalidadSMA;

    /**
     * Método que ejecuta la funcionalidad de sectorización de la aplicación web
     * @param request
     *         Petición de cliente
     * @param response
     *         Respuesta al cliente
     * @throws Exception
     *         Excepciones
     */
    public void sectorizar(HttpServletRequest request,

```

Método a ser publicado

Figura 4.61 Ejemplo de Implementación de Clase Controlador

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

➤ Configuración de controladores

La configuración de los controladores se realiza a través de archivos especificados por el *framework* de desarrollo (Figura 4.62).

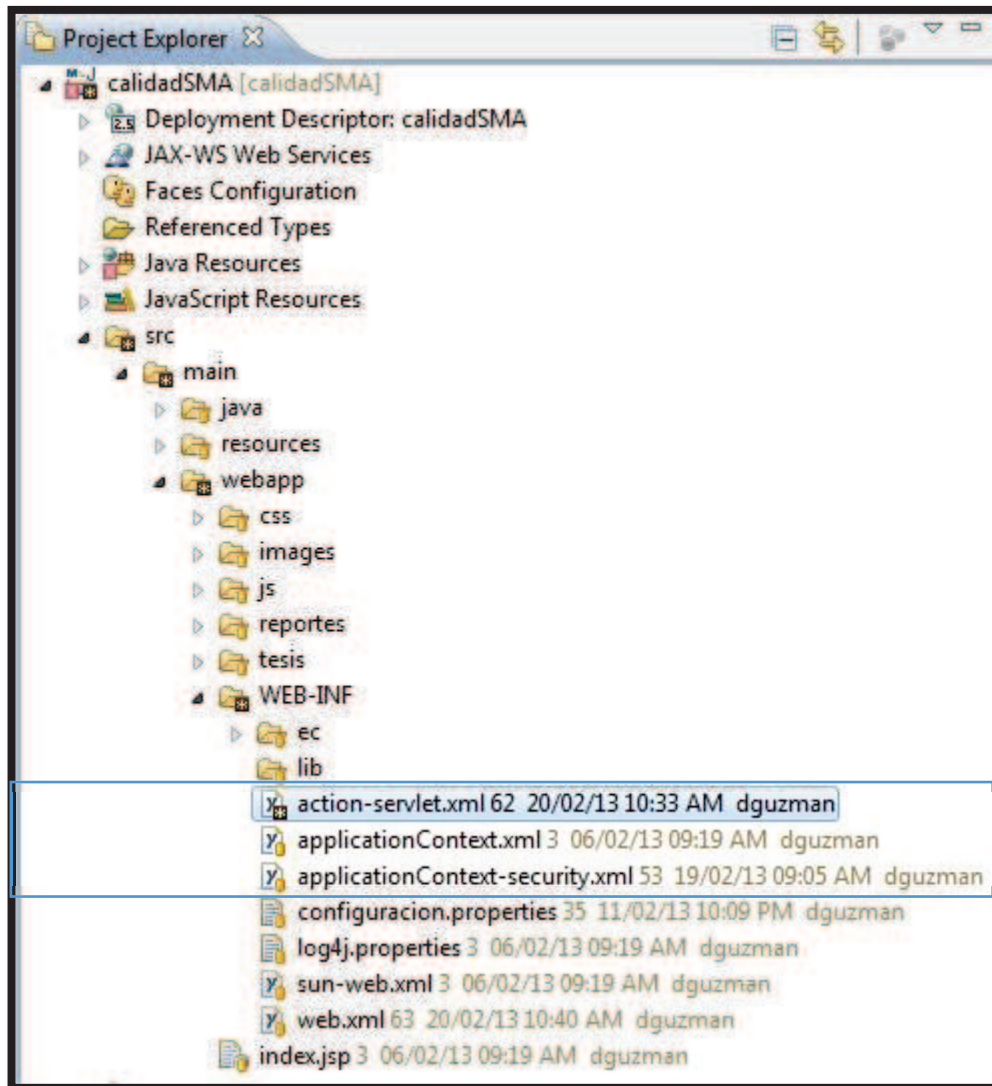


Figura 4.62 Visualización de Archivos de Configuración del Proyecto

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

En el archivo “action-servlet.xml” (Figura 4.63) se identifican etiquetas que hacen referencia a los objetos (*bean*) que interpretan los parámetros que se enviarán al servidor, objetos que resuelven las vistas (*views*) y objetos que mapearán las clases para que puedan publicar sus métodos; además de los paquetes en donde se encuentra el código fuente a ser publicado.

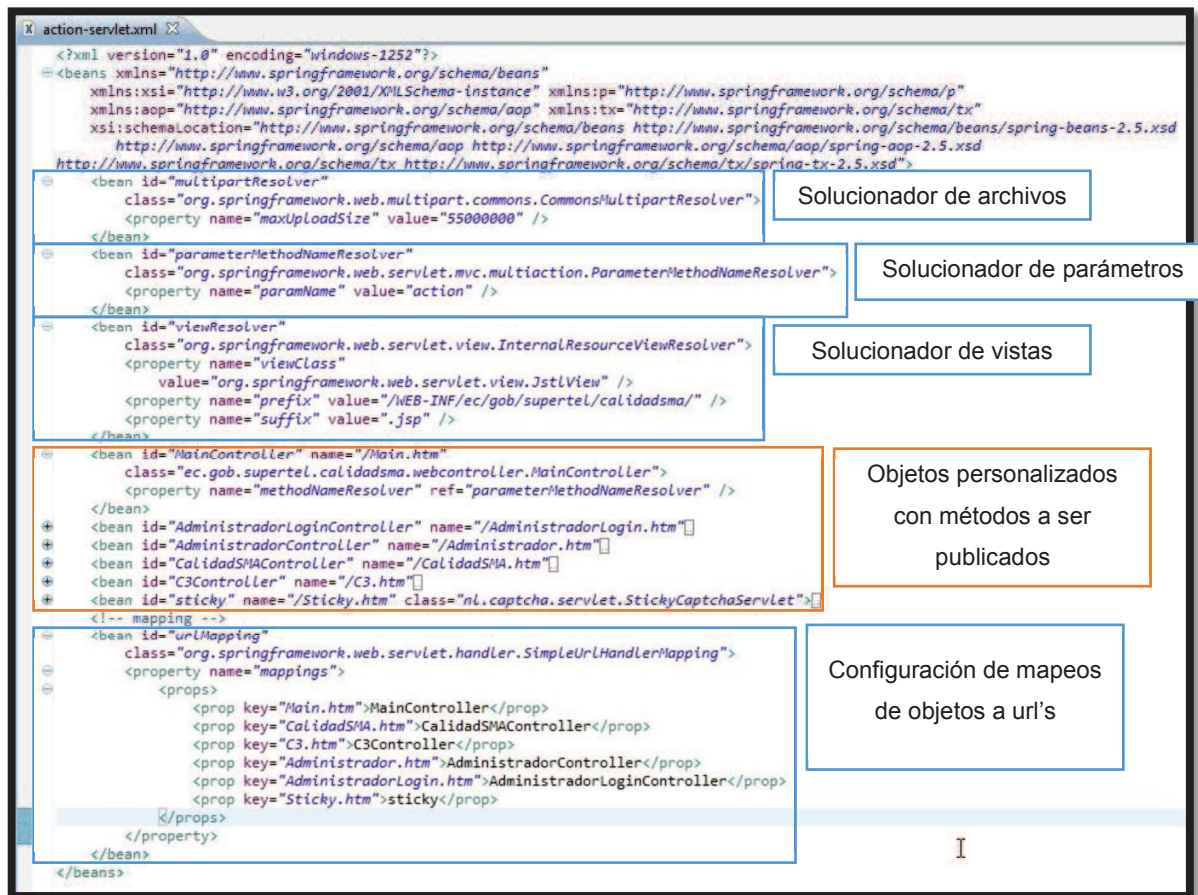


Figura 4.63 Detalle de Configuración del Archivo action-servlet.xml

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

➤ Clases de Spring Framework Web MVC

La clase que provee el servicio para resolver los parámetros que son enviados a través del protocolo HTTP es la clase:

org.springframework.web.servlet.mvc.multiaction.ParameterMethodNameResolver

A la cual se ha indicado que los nombres de los métodos a ser ejecutados en el servidor se especificarán con la palabra “action”.

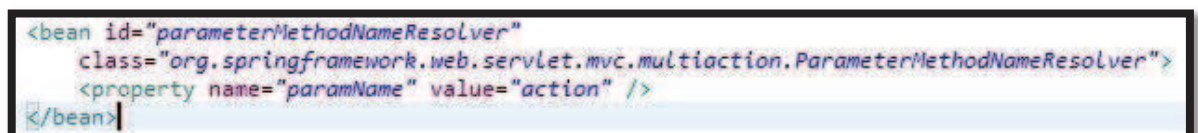


Figura 4.64 Detalle de Configuración de Clases de Spring Framework

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

➤ Clases personalizadas

Se instanciarán objetos del tipo controlador para publicar los métodos en “servlets”, las clases corresponden al espacio de nombres:

ec.gob.supertel.calidadsma.webcontroller

Los objetos que se instancien tendrán configurado previamente la URL del recurso, la localización de la clase y el proveedor de parámetros.

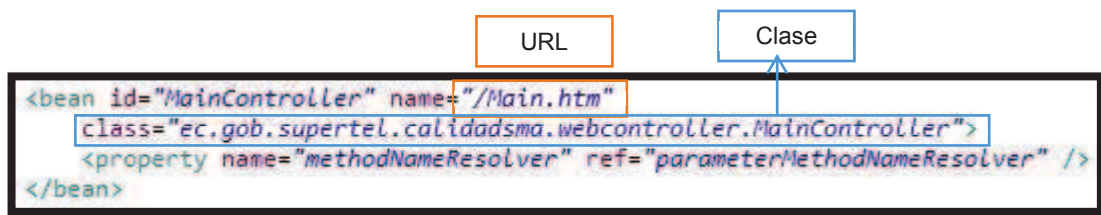


Figura 4.65 Detalle de Configuración de Clases Personalizadas

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

➤ Funcionamiento de controladores

Finalmente los controladores estarán publicados como recursos en el servidor; es de ésta manera que se pueden hacer llamadas como se presenta en la Figura 4.66.

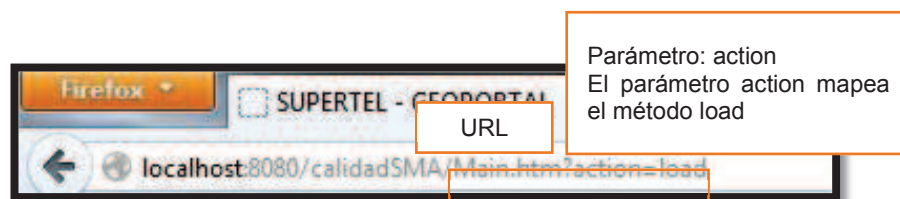


Figura 4.66 Detalle de Funcionamiento de Métodos Publicados en el Servidor

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

La URL “/Main.htm?action=load”, ejecuta el método “load” del “servlet” Main.htm que corresponde a la clase:

ec.gob.supertel.calidadsma.webcontroller.MainController

Esta clase contiene el método “load” que implementa la funcionalidad de devolver un objeto que representa la vista, interfaz, que será interpretada por el navegador. Esto se visualiza en la depuración del código que se presenta en la Figura 4.67.

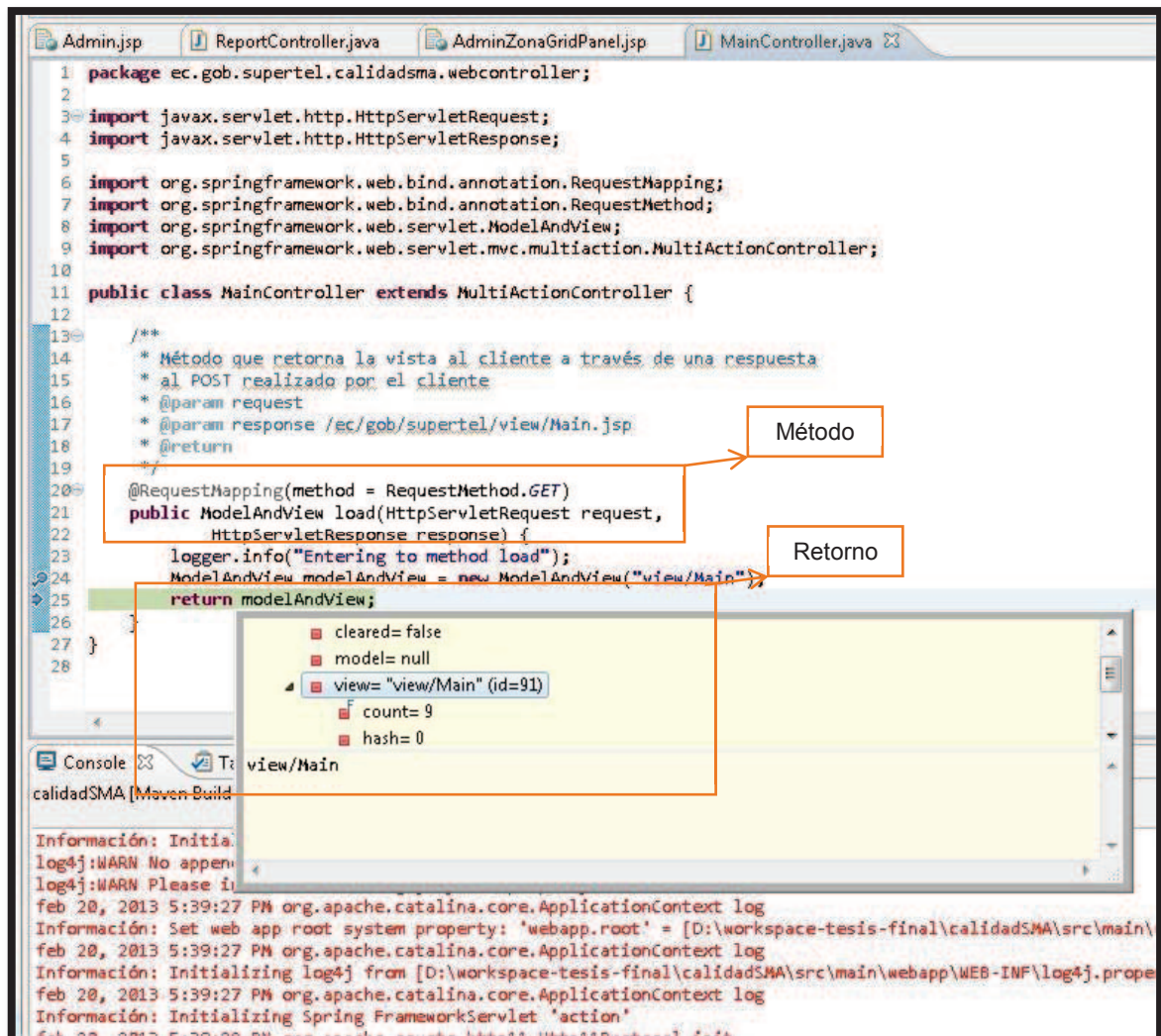


Figura 4.67 Detalle de Funcionalidad de Servlets

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

Esto gracias al envío y recepción de solicitudes HTTP del tipo GET, como se observa en la Figura 4.68 y Figura 4.69.

Teniendo como respuesta un archivo contenedor de código JavaScript (Figura 4.70) que contiene la funcionalidad de interacción del cliente a través del navegador con el servidor web.

URL	Status	Domain	Size	Remote IP	Timeline
GET Main.htm?action=load	200 OK	localhost:8080	59,3 KB	127.0.0.1:8080	38ms

Params	Headers	Response	HTML	Cookies
Response Headers				
Content-Language	en-US			
Content-Type	text/html; charset=UTF-8			
Date	Wed, 20 Feb 2013 22:28:24 GMT			
Server	Apache-Coyote/1.1			
Transfer-Encoding	chunked			
Request Headers				
Accept	text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8			
Accept-Encoding	gzip, deflate			
Accept-Language	en-US,en;q=0.8			
Connection	keep-alive			
Cookie	JSESSIONID=85ACAFF36E0445206C852F107A88038			
Host	localhost:8080			
User-Agent	Mozilla/5.0 (Windows NT 6.1; WOW64; rv:18.0) Gecko/20100101 Firefox/18.0			

Figura 4.68 Detalle de Solicitudes HTTP enviadas al Servidor

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

URL	Status	Domain	Size	Remote IP	Timeline
GET Main.htm?action=load	200 OK	localhost:8080	59,3 KB	127.0.0.1:8080	38ms

Params	Headers	Response	HTML	Cookies
action	load			

Figura 4.69 Detalle de Parámetros Enviados en las Solicitudes HTTP

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

URL	Status	Domain	Size	Remote IP	Timeline
GET Main.htm?action=load	200 OK	localhost:8080	59,3 KB	127.0.0.1:8080	38ms

Params	Headers	Response	HTML	Cookies
<pre><!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd" > <html> <head> <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"> <title>SUPERTEL - GEOPORTAL</title> <!-- Estilos --> <link rel="stylesheet" href="css/google.css" type="text/css"> <link rel="stylesheet" href="css/style.css" type="text/css"> <link rel="stylesheet" href="css/other.css" type="text/css"> <link rel="stylesheet" href="css/calidadSMA.css" type="text/css"> <link rel="stylesheet" type="text/css" href="js/ext-3.4.0/resources/css/ext-all.css" /> <!-- api Google --> <script src="http://maps.google.com/maps/api/js?v=3.7&sensor=false"></script> <!-- Busy indicator --> <script type="text/javascript" src="js/busy-indicator/cvi_busy_lib.js"></script> <!-- ExtJS 3.4.0 --> <script type="text/javascript" src="js/ext-3.4.0/adaptor/ext/ext-base.js"></script> <script type="text/javascript" src="js/ext-3.4.0/ext-all.js"></script> <!-- JS OpenLayers --> <script src="js/OpenLayers211/OpenLayers.js"></script> <!-- GeoExt --> <script type="text/javascript" src="js/geoext/GeoExt.js"></script> <!-- Profiles --></pre>				

Figura 4.70 Detalle de Respuesta a la Solicitud HTTP

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

4.6.4.1.4 Vistas

La capa de presentación está basada en bibliotecas JavaScript para desarrollo de aplicaciones web interactivas, estas librerías permiten interpretar código a nivel de cliente en los navegadores locales de cada usuario.

➤ ExtJS

Es una librería JavaScript que permite construir interfaces web enriquecidas con controles personalizables, extensibles y de fácil uso.

➤ Páginas Contenedoras

Los contenedores del código JavaScript son archivos de extensión “.jsp”, misma tecnología *Java Server Pages* para el proyecto no se está utilizando en su totalidad, debido a los requerimientos de funcionalidad. Así se tiene que estas páginas contienen etiquetas básicas HTML de definición de documentos (Espacio de Código 4.11).

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loos
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title><%=Constants.calidadSMAtitle%></title>
<!-- Estilos -->
<link rel="stylesheet" href="css/google.css" type="text/css">
```

Espacio de Código 4.11 Definición de Documentos HTML Contenedores

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

➤ Definición de Tipo de Texto Insertado

El código programado está contenido en etiquetas contenedoras de código JavaScript.

```
<script type="text/javascript">
//Constantes - URL de conexión a los servidores de mapas
var urlMapProxy = "<%=Constants.calidadSMAURLMapProxy%>";
var urlMapServer = "<%=Constants.calidadSMAURLMapServer%>";

//Variables para proyeccion de mapas y capas
var proj4326 = new OpenLayers.Projection("EPSG:4326");
var proj900913 = new OpenLayers.Projection("EPSG:900913");
var proj24877 = new OpenLayers.Projection("EPSG:24877");
```

Espacio de Código 4.12 Definición de Bloque de Código JavaScript

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

➤ Definición de Variables y Constantes

Las variables y constantes han sido definidas al inicio de cada bloque de código JavaScript.

```
//Constantes - URL de conexión a los servidores de mapas
var urlMapProxy = "<%=Constants.calidadSMAURLMapProxy%>";
var urlMapServer = "<%=Constants.calidadSMAURLMapServer%>";

//Variables para proyeccion de mapas y capas
var proj4326 = new OpenLayers.Projection("EPSG:4326");
var proj900913 = new OpenLayers.Projection("EPSG:900913");
var proj24877 = new OpenLayers.Projection("EPSG:24877");

//Variable mapa - objeto
var map, toolbarItems = [], contentPanel, xy;
//Contenedor de manejadores geograficos
var mapPanel;
```

Espacio de Código 4.13 Definición de Variables y Constantes

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

➤ Definición y Configuración de Controles *ExtJS*

El contenedor principal de la aplicación es un panel *ExtJS*, configurado con paneles de mapas, barras de herramientas, eventos y demás controles de interacción con el usuario.

```
//CONTENEDOR PRINCIPAL DE PANELES Y MAPAS
new Ext.Viewport({
  layout : "border",
  items : [
    {
      region : "center",
      id : "mappanel",
      title : "CALIDAD DE SERVICIO MÓVIL AVANZADO",
      xtype : "gx_mappanel",
      map : map,
      //layers: [gmap],
      extent: new OpenLayers.Bounds(-9580761.6242, -544824.2220, -8349908.8628, 156706.3181),
      split : true,
      tbar : toolbarItems,
      center: new OpenLayers.LonLat(-3.8, 173.28).transform(new OpenLayers.Projection("EPSG:4326"), map.getProjectionObject()),
      zoom : 7
      //items: [mapPanel]
    },
    {
      region : "west",
      width : '20%',
      split : true,
      items : [ {
        xtype : 'box',
        autoEl : {
          tag : 'img',
          src : 'images/dtil.jpg',
          height : '11%',
          width : '100%'
        }
      } ]
    }
  ]
});
```

Espacio de Código 4.14 Panel Principal, Código JavaScript

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

El control contenedor principal se lo puede apreciar en la Figura 4.71.

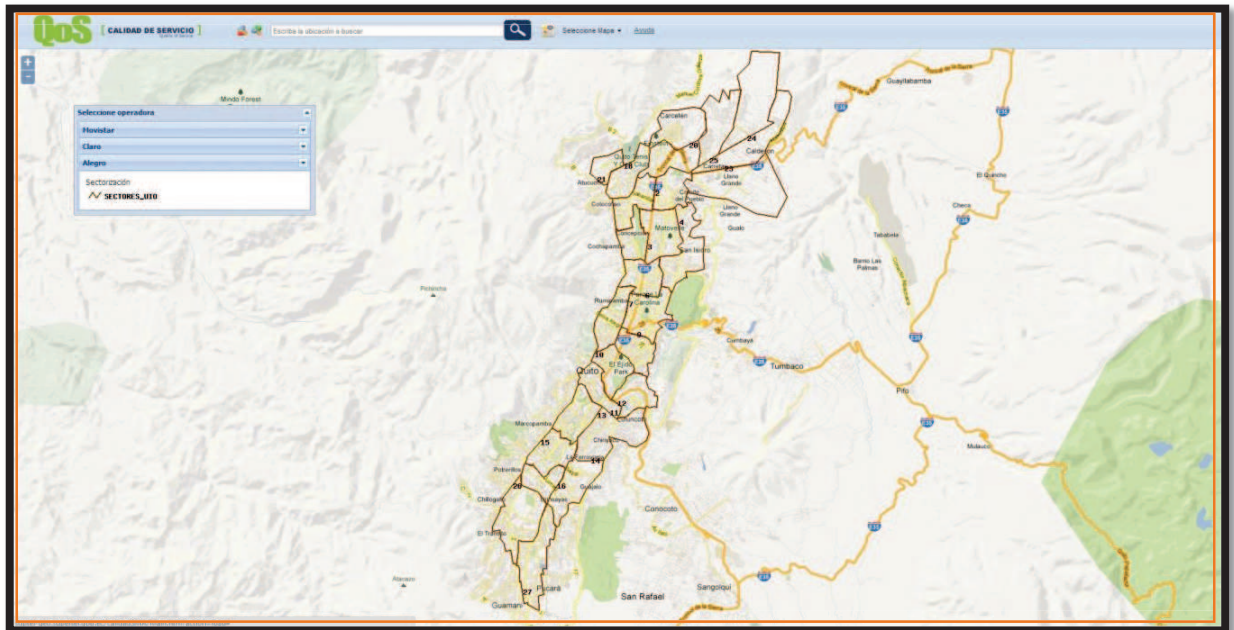


Figura 4.71 Panel Contenedor Principal

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

- Inicialización de Funcionalidad

Para inicializar la configuración de los controles *ExtJS* es necesario invocar las siguientes sentencias.

```

//***** INICIALIZACIÓN DE LOS CONTROLES *****
Ext.onReady(function() {
    if (window.XMLHttpRequest) { // code for IE7+, Firefox, Chrome, Opera, Safari
        xmlhttp = new XMLHttpRequest();
    } else { // code for IE6, IES
        xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");
    }
    //block_viewport();
    initialFunction();
});

```

Espacio de Código 4.15 Inicialización de Controles *ExtJS*

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

4.6.4.1.5 Mapas

- *OpenLayers*

Con la librería *OpenLayers*, se realizarán las peticiones WMS al servidor de mapas. Las imágenes serán proyectadas sobre el mapa base (*Google Maps*), para su visualización.

En primera instancia, se debe definir una nueva capa y su fuente de origen, en este caso WMS. Como parámetros se debe definir el nombre de la capa, nombre de *host* del servidor de mapas, SRID para su proyección (no necesariamente con el que está almacenado). Además del formato de imagen para mostrar, si tiene o no transparencia. También se puede definir si la capa se puede o no visualizar y el nivel de opacidad con el que se mostrará (Espacio de Código 4.16).

```
//CARGA DE LA CAPA PRINCIPAL DEL MAPA - DIBUJA LA SECTORIZACIÓN DE LA CIUDAD-QUITO
var layerSectoresUIO = new OpenLayers.Layer.WMS("Sectorización",
    urlMapProxy, {
        layers : [ 'qos_sectores_uio' ],
        EPSG : 900913,
        format : 'image/png',
        transparent : "true"
    }, {
        isBaseLayer : false,
        visibility : true,
        opacity : '1'
    });
```

Espacio de Código 4.16 Definición Capa WMS

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

El Espacio de Código 4.16 genera una petición WMS al servidor de mapas, si todos los parámetros son coherentes y la capa solicitada existe, el servidor de mapas devuelve varias imágenes (a manera de piezas de un todo, que es la capa completa).

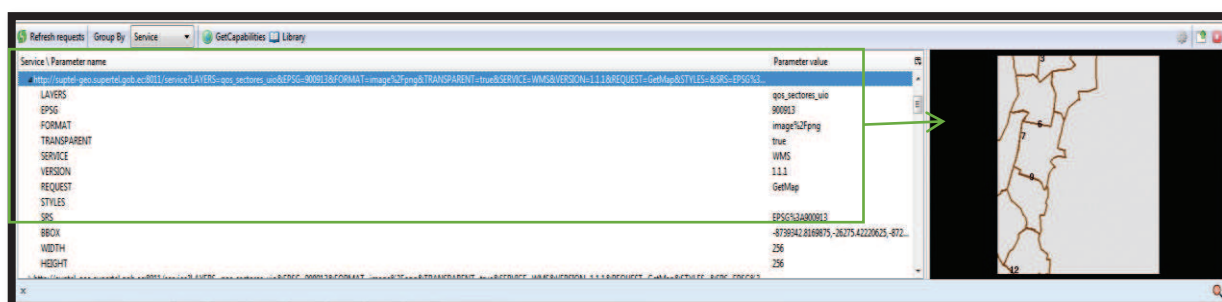


Figura 4.72 Análisis de una Petición WMS⁵²

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

⁵² Datos obtenidos gracias al complemento WMS Inspector, navegador Mozilla Firefox.

Como se aprecia en la Figura 4.72, todos esos parámetros son enviados al servidor y éste devuelve imágenes, que en el navegador se ordenan y generan la capa completa.

➤ *GeoExt*

Es una librería *JavaScript* que provee la funcionalidad de poder integrar controles del tipo escritorio *ExtJS* y controles de mapas *OpenLayers*, orquestando el manejo e interacción de la interfaz con el usuario final a través de la aplicación web enriquecida. *GeoExt* provee una suite de controles personalizables que hacen fácil la programación de controles de visualización, edición y presentación de datos geo-espaciales. Uno de los controles utilizados de esta librería es el control “*MapPanel*”, contenedor de la presentación, eventos y controles del mapa.

```

/**Panel principal contenedor del mapa**/
var mapPanel = new GeoExt.MapPanel({
    height : 400,
    width : 600,
    map : map,
    tbar : toolbarItems
});

```

Espacio de Código 4.17 Definición del Panel Contenedor del Mapa

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

En la interfaz se puede identificar el control como se presenta en la Figura 4.73.

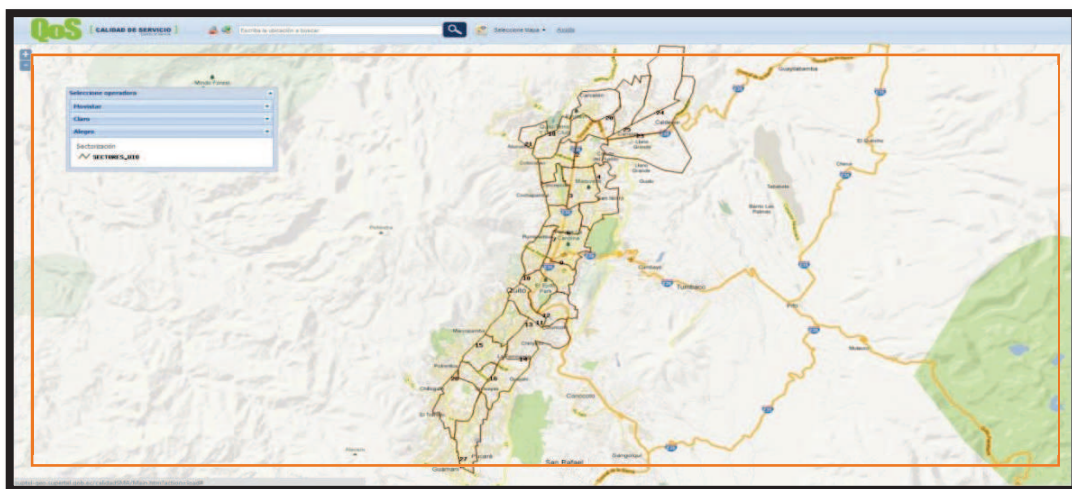


Figura 4.73 Panel Principal Contenedor del Mapa

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

➤ Proj4JS

Librería JavaScript utilizada para transformar coordenadas geográficas de un sistema de referencia hacia otro. También brinda soporte para transformaciones a nivel de *datum*. Su funcionalidad principal es la de permitir mostrar, a través del navegador web, mapas o capas que están almacenadas utilizando diferentes SRID dentro de una misma aplicación web. Para su utilización, primero incluir las librerías en el contexto de la aplicación.

```
<!-- Proj4js -->
<script type="text/javascript" src="js/proj4js/proj4js.js"></script>
<script type="text/javascript" src="js/proj4js/projCode/tmerc.js"></script>
<script type="text/javascript" src="js/proj4js/projCode/merc.js"></script>
<script type="text/javascript" src="js/proj4js/projCode/utm.js"></script>
<script type="text/javascript" src="js/proj4js/defs/EP5G900913.js"></script>
```

Espacio de Código 4.18 Inclusión de la Librería Proj4js en la Aplicación

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

Luego definir los SRIDs de proyección a utilizarse dentro de la aplicación, como se muestra en la Figura 4.74.

```
//Variables para proyeccion de mapas y capas
var proj4326 = new OpenLayers.Projection("EPSG:4326");
var proj900913 = new OpenLayers.Projection("EPSG:900913");
var proj24877 = new OpenLayers.Projection("EPSG:24877");
```

Figura 4.74 SRIDs Utilizados en la Aplicación calidadSMA

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

Después cada vez que se agregue una capa (sea cual fuere su origen y SRID), se debe definir el SRID en el cual se quiere proyectar la capa. Para este caso específico la capa *qos_sectores_quito* tiene un SRID 4326, sin embargo al tener como capa base los mapas proporcionados por *Google Maps* (SRID 900913), se debe realizar la transformación para que la imagen proyectada tenga coherencia respecto del mapa.

```
//Transformacion de la proyeccion
var coordenada = map.getLonLatFromPixel(
    e.xy).transform(
    new OpenLayers.Projection(
        'EPSG:900913'),
    new OpenLayers.Projection(
        'EPSG:4326'));
```

Espacio de Código 4.19 Transformación de proyecciones entre SRID 4326 y 900913

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

4.6.4.1.6 Reportes

Los reportes se han generado a través de una tabla *QOS_RECLAMOS_QUITO*, dentro del esquema de la aplicación. Esta tabla almacena únicamente los reclamos ingresados a través de la aplicación calidadSMA. Adicionalmente, en la capa de persistencia, se han generado la entidad “Reclamo.java” (Espacio de Código 4.20) para mapear la información de la tabla. Y de esta manera poder generar los reportes necesarios.

Como se aprecia en el Espacio de Código 4.20, es una clase con sus respectivos atributos, sin embargo las anotaciones que se encuentran en el contexto, indican que es una entidad que hace referencia a una tabla *QOS_RECLAMO_QUITO*, y los atributos son las columnas de la tabla.

➤ *JasperReports*

Para la generación de reportes, se dispone de la librería *JasperReport*, la cual trabaja en conjunto con la herramienta *iReport Designer*⁵³. En la herramienta *iReport*, se construye el reporte (formato, diseño, consultas a la base de datos, etc.). Con el reporte ya estructurado, se tienen dos archivos, el primero que tiene una extensión *.jrxml* y otro con una extensión *.jasper*. La extensión *.jrxml* hace referencia a un archivo *.xml* con etiquetas propias de la librería *JasperReports*.

⁵³ Disponible para descarga en <http://community.jaspersoft.com/project/ireport-designer>

```

package ec.gob.supertel.calidadsma.entidades;

import java.io.Serializable;

/**
 * Entity implementation class for Entity: Reclamo
 * @author dguzman
 */
@Entity
@Table(name="QOS_RECLAMOS_QUITO")
public class Reclamo implements Serializable {

    private static final long serialVersionUID = 1L;
    @Id
    @Basic(optional = false)
    @Column(name = "QR_FID")
    private Integer qrFid;

    @Basic(optional = false)
    @Column(name = "QR_OPERADORA")
    private Integer qrOperadora;

    @Basic(optional = false)
    @Column(name = "QR_SERVICIO")
    private String qrServicio;

    @Basic(optional = false)
    @Column(name = "QR_IDCIR")
    private Integer qrIdcir;

    @Basic(optional = false)
    @Column(name = "QR_FECHA_INGRESO")
    @Temporal(TemporalType.TIMESTAMP)
    private Calendar qrFechaIngreso;

    @Basic(optional = false)
    @Column(name = "QR_ESTADO")
    private Integer qrEstado;
}

```

Espacio de Código 4.20 Entidad Reclamo.java

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

Como se representa en la Figura 4.75, *JasperReports*, toma el archivo .jasper, lo rellena con los orígenes de datos definidos y finalmente exporta el reporte generado (formato PDF, HTML, XML, etc.).

Para el diseño del reporte se inicia definiendo un origen de datos, que puede ser una base de datos o un conjunto de entidades que hagan referencia a tablas de una base de datos. Además, se define el diseño del reporte (cabeceras, nombres, títulos, etc.), una vez culminado el diseño se compila el reporte, de donde se origina el archivo ".jasper", este archivo será el que se invoque desde la aplicación a través de un "servlet". Dentro del "servlet" inicialmente se ubica el archivo ".jasper" generado, luego se define la ruta al directorio y se lo almacena en el objeto *reporte* (clase *JasperReports*).

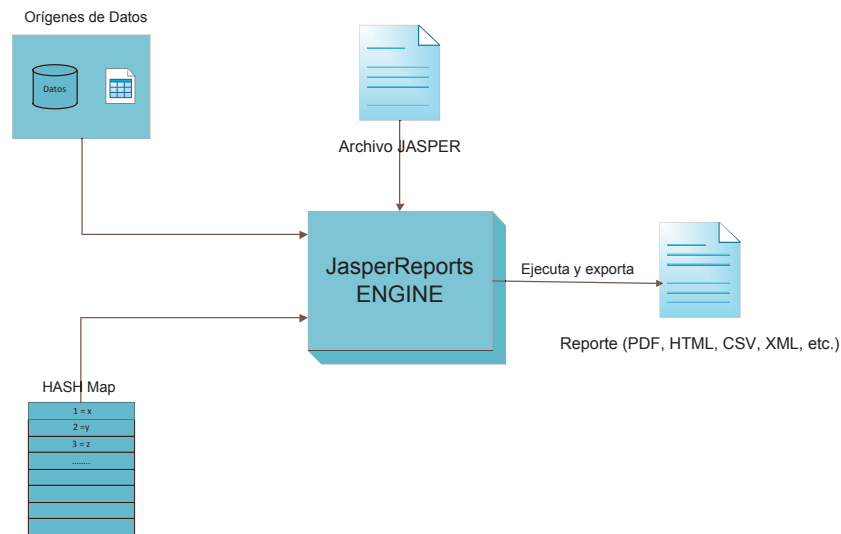


Figura 4.75 Funcionamiento *JasperReports*

Fuente: Adaptado de Jasper Reports for Confluenc. Recuperado de: <https://bobswift.atlassian.net/wiki/display/JSPR/Jasper+Reports+for+Confluence> (Marzo 2013).

```
private JasperReport obtenerReporte(HttpServletRequest request) throws MalformedURLException, JRException{
    String nombre = request.getParameter("nombre");
    JasperReport reporte = null;
    try
    {
        String pathReporte = obtenerPathReporte(request)+ nombre + ".jasper";
        URL url = new URL(pathReporte);
        reporte = (JasperReport)JLoader.loadObject(url);
    }catch(Exception ex)
    {
        ex.printStackTrace();
    }
    return reporte;
}
```

Espacio de Código 4.21 Cargar Archivo *Jasper* en Entorno de Desarrollo

Fuente: Esquema de Autoría propia (Arévalo D., y Guzmán D.).

Luego se instancia un objeto "jasperprint" (clase *jasperPrint*), y con la utilización del método "*fillReport*" se almacena todos los datos a desplegarse dentro del reporte. Este método tiene como parámetros de entrada el archivo "*jasper*", los parámetros de consulta o discriminación de datos y finalmente la fuente u origen de datos.

```

private JasperPrint cargarReporteObjetos(JasperReport reporte, HttpServletRequest request, Map<String,
Object> parametros, List<?> dataSource) {
    try {
        JRDataSource reportDataSource = new JRBeanCollectionDataSource(dataSource);
        JasperPrint jasperPrint = null;
        jasperPrint = JasperFillManager.fillReport(reporte, parametros, reportDataSource);
        return jasperPrint;
    } catch (Exception e) {
        e.printStackTrace();
        return null;
    }
}

```

Espacio de Código 4.22 Obtener Datos para el Reporte

Fuente: Esquema de Autoría propia (Arévalo D., y Guzmán D.).

Finalmente, se define el tipo de contenido que arroja el “*servlet*” con esto el navegador sabe que lo que espera es un archivo tipo “.pdf”, y se coloca en el *stream* de salida el objeto “*jasperprint*” que fue llenado anteriormente.

```

private void visualizarReportePdf(HttpServletRequest response, JasperPrint jasperPrint) {
    try {
        response.setContentType("application/pdf");
        ServletOutputStream out = response.getOutputStream();
        JasperExportManager.exportReportToPdfStream(jasperPrint, out);
    } catch (Exception e) {
        e.printStackTrace();
    }
}

```

Espacio de Código 4.23 Visualizar Reporte en Formato PDF

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

Si el proceso culminó de forma correcta el navegador debe desplegar el reporte con datos, como se aprecia en la Figura 4.76.

4.6.5 IMPLEMENTACIÓN DE NIVELES DE SEGURIDAD

4.6.5.1 Servidor Single Sign On

La SUPERTEL dentro de su esquema de seguridades maneja los procesos de autenticación y acceso a aplicaciones a través de un mecanismo *Single Sign On* (SSO), mediante este mecanismo el usuario presenta sus credenciales una única vez y tiene acceso garantizado a varios sistemas o aplicaciones.

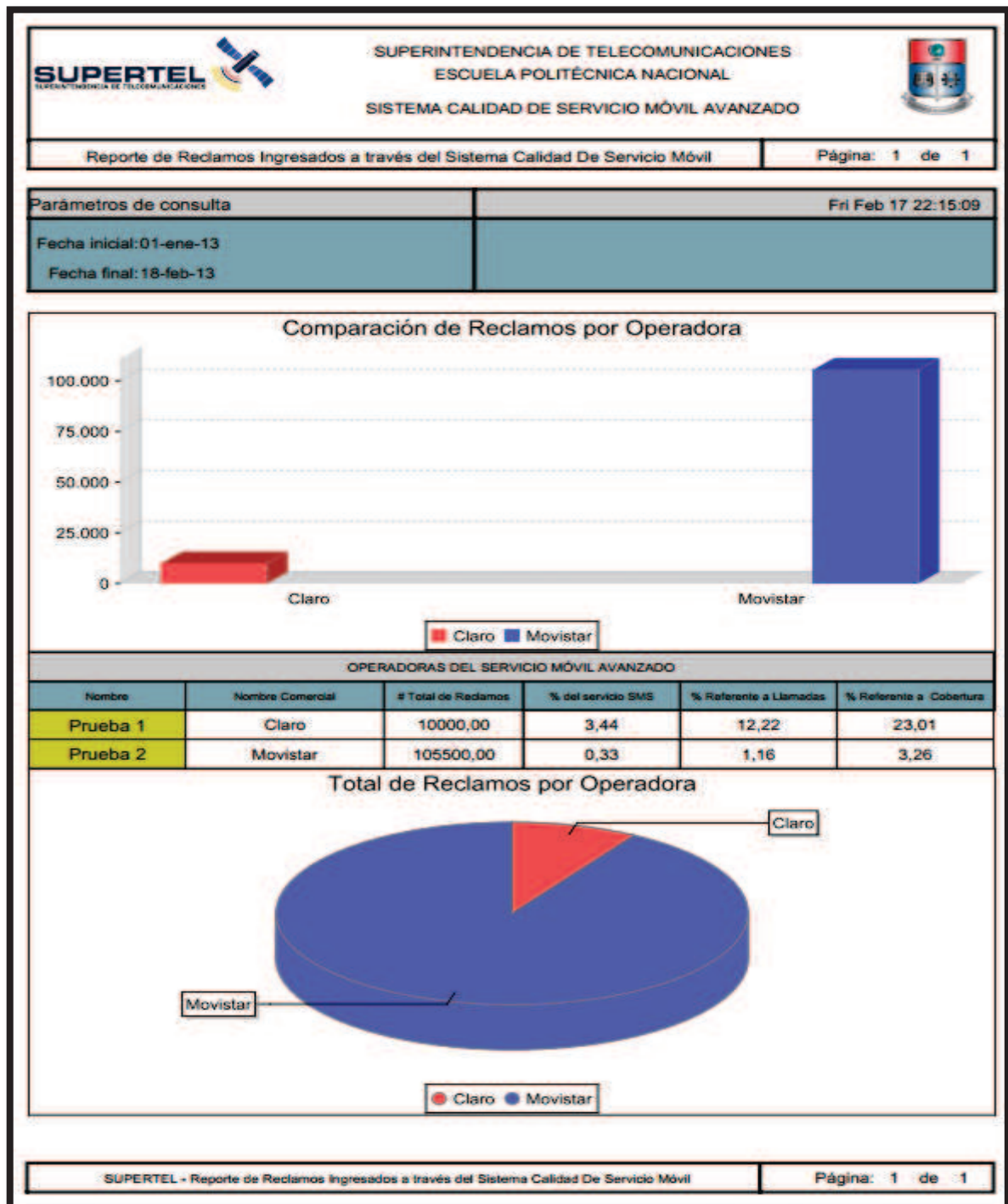


Figura 4.76 Reporte de Reclamos

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

Difiere de los procesos tradicionales de autenticación, en donde el usuario debe presentar sus credenciales cada vez que desea ingresar a una aplicación dentro de una organización.

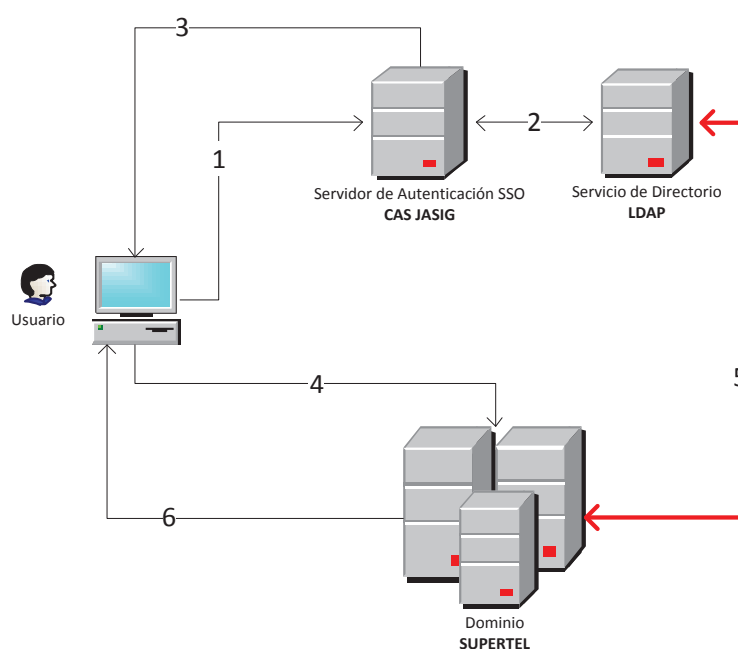


Figura 4.77 Mecanismo de Autenticación SSO

Fuente: Adaptado de About CAS. Recuperado de: <https://confluence.ucdavis.edu/confluence/display/IETP/About+CAS> (Marzo 2013).

SSO utiliza un Servicio Centralizado de Autenticación (CAS), al utilizar un servicio centralizado de autenticación, las credenciales (nombres de usuario, contraseñas, etc.) están almacenados en un directorio, de aquí la necesidad de la utilización de un servicio de directorio donde almacenar toda la información necesaria de los usuarios.

Para el caso específico de la SUPERTEL, se utiliza (Protocolo Ligero de Acceso a Directorios - *Lightweight Directory Access Protocol*) LDAP, y como servidor centralizado de autenticación CAS JASIG⁵⁴, es una aplicación web que requiere de un servidor web para su funcionamiento.

La Figura 4.77 esquematiza el mecanismo de autenticación con el servidor centralizado.

1) El usuario presenta sus credenciales al servidor de autenticación CAS JASIG.

⁵⁴ <http://www.jasig.org/cas>

- 2) El servidor CAS JASIG, verifica la autenticidad de las credenciales, consultando al servicio de directorio.
- 3) Si las credenciales son correctas, el servidor de autenticación devuelve un *token*⁵⁵ al usuario, este *token* es una *cookie*⁵⁶ que se almacena dentro del equipo del usuario.
- 4) Cuando el usuario desea ingresar a cualquier aplicación que pertenezca al dominio, envía su *token* a la aplicación o sistema.
- 5) La aplicación consulta la validez del *token* con el servicio de directorio.
- 6) Si el *token* es válido, la aplicación se muestra al usuario.

4.7 ENTORNO DE DESARROLLO COLABORATIVO

Se ha utilizado un ambiente de desarrollo en el que los participantes del proyecto pueden hacer uso de un solo repositorio de código compartido, mismo que permite la administración de versiones del producto final.

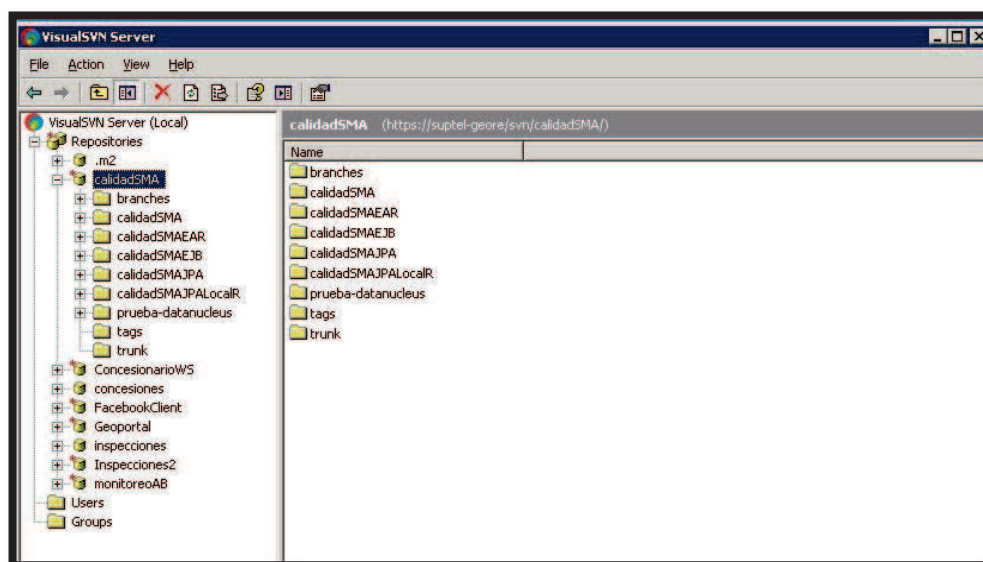


Figura 4.78 Interfaz del Servidor de Versionamiento SVN

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

⁵⁵ Componente informático que identifica un usuario en particular. Se caracteriza por no contener información de nombres ni contraseñas que puedan poner el riesgo la identidad del usuario.

⁵⁶ Es un archivo que envía un servidor a un usuario, es almacenado y contiene información del usuario (preferencias de navegación, identificador de usuario, etc.).

4.7.1 REPOSITORIO

Como se observa en la Figura 4.78, el servidor de versionamiento proporcionado por la SUPERTEL, mantiene los códigos de los proyectos desarrollados.

4.7.1.1 Cliente

Realizando conexiones a este servidor los clientes desarrolladores pueden tener acceso al código desarrollado.

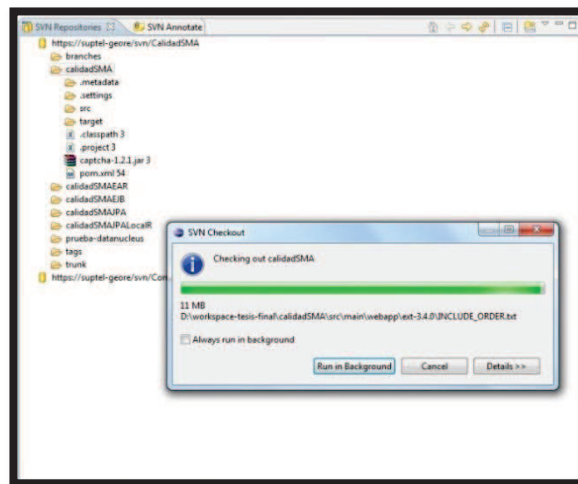


Figura 4.79 Descarga del Proyecto en un Cliente

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

4.7.2 MAVEN

La herramienta *Maven* es utilizada para la gestión y construcción del proyecto Java. La herramienta utiliza el archivo *POM* para realizar la construcción del proyecto.

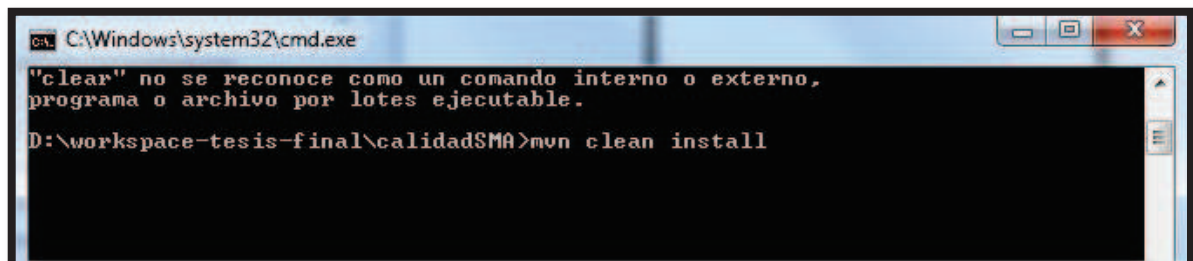


Figura 4.80 Ejecución de Comandos Maven para Construcción del Proyecto

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

4.8 IMPLEMENTACIÓN DEL MÓDULO DE ADMINISTRADOR

4.8.1 MÓDULO LOGIN

La SUPERTEL dentro de su esquema de seguridades, maneja los procesos de autenticación y acceso a aplicaciones, a través de un mecanismo *Single Sign On* (SSO), proceso que ya fue descrito en la sección 4.6.5.1.

4.8.2 SECTORIZACIÓN

Para implementar este módulo se parte de un análisis similar al realizado por la SUPERTEL, en el cual divide la ciudad de Quito en cuadras ideales de 100 m x 100 m (área de 10000 m²), con los datos del área de cobertura para el SMA y con el paquete QGIS v 1.8 se generará una matriz de cuadras ideales (cuadrados) para la zona de cobertura de la ciudad de Quito. El proceso se describe a continuación. Se importa el archivo “.shp” de zonas en el *workspace* de QGIS. Luego en herramientas vectoriales se escoge herramientas de investigación, cuadrícula vectorial.

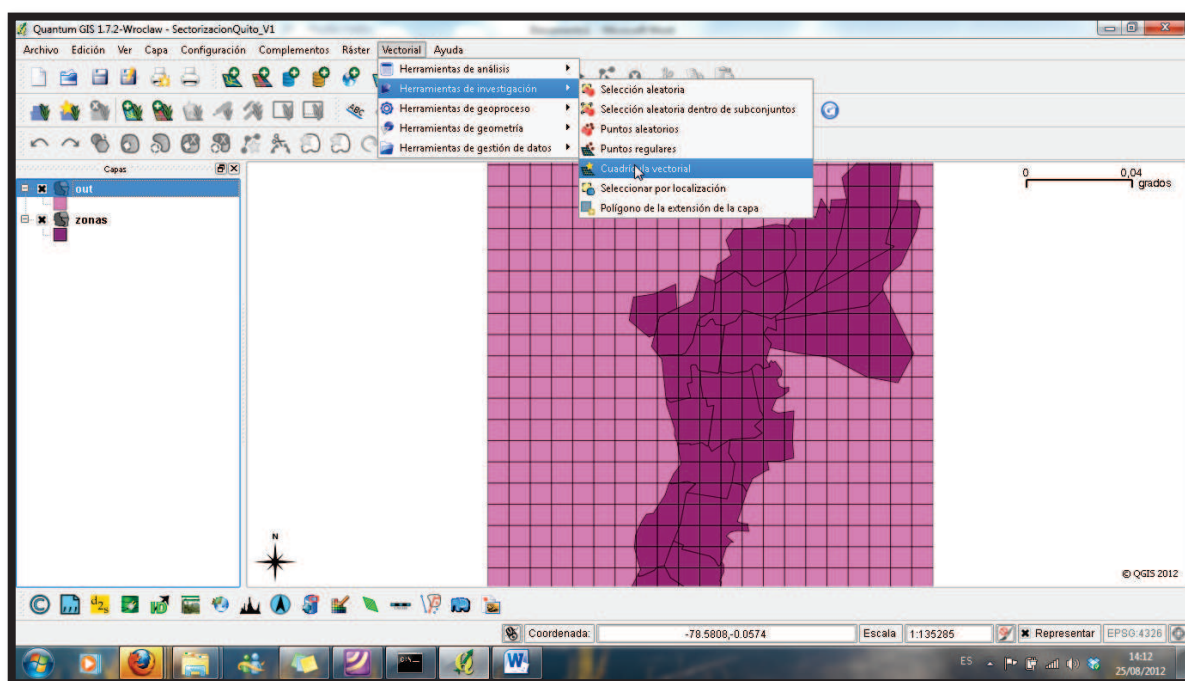


Figura 4.81 Generación de Grilla para Sectorización

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

En la ventana desplegada, se deben definir el valor por lado de cada cuadrícula (en radianes) y mantener la relación 1:1. El valor aproximado que genera cuadrículas de lado 250m es de 0.0045 por lado, estos valores fueron determinados de una manera empírica.

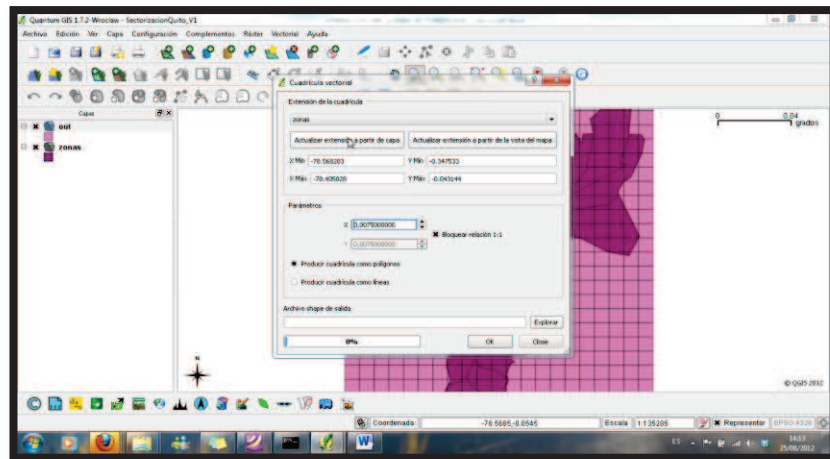


Figura 4.82 Grilla de 250m de Lado

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

Se genera otra capa vectorial y su correspondiente archivo *shape*, en la capa generada se observa datos importantes como el SRID y los valores que definen el *bounding box*, los cuales deben ser iguales al de la capa original. Para obtener las cuadrículas de área igual a 10000 m² (simulando cuadras de 100x100 ideales) se debe definir un valor de X igual a 0.0009.

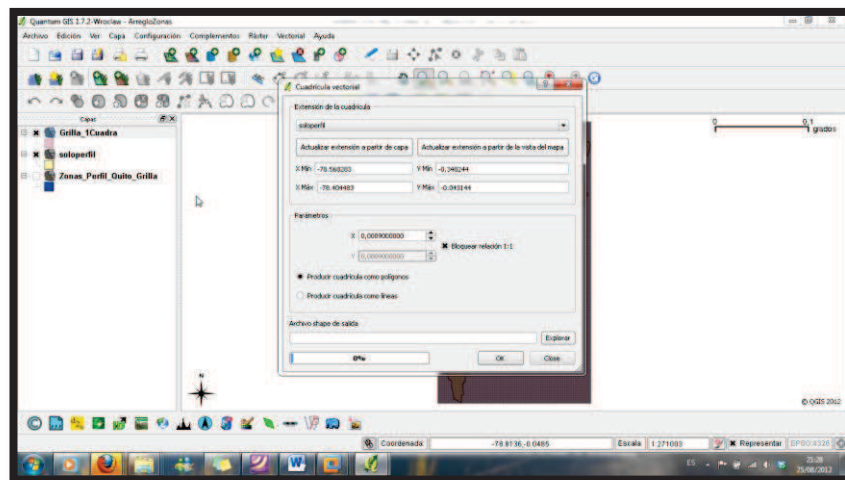


Figura 4.83 Generación Grilla 100m de Lado

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

La matriz generada posee cuadrículas de aproximadamente 10000 m^2 , como se observa en Figura 4.84.

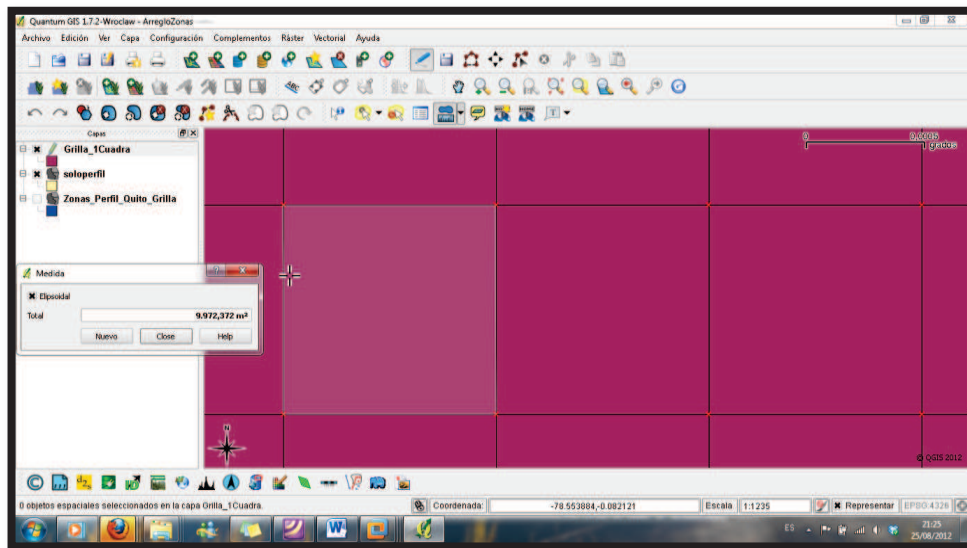


Figura 4.84 Medición de Grilla de Área Igual 10000 m^2

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

Con la matriz de nombre *Grilla_1Cuadra* cargada en el *workspace* se debe realizar una operación de intersección con la capa de cobertura de la ciudad de Quito, para obtener la matriz de cuadras dentro del área de cobertura definida por la SUPERTEL.

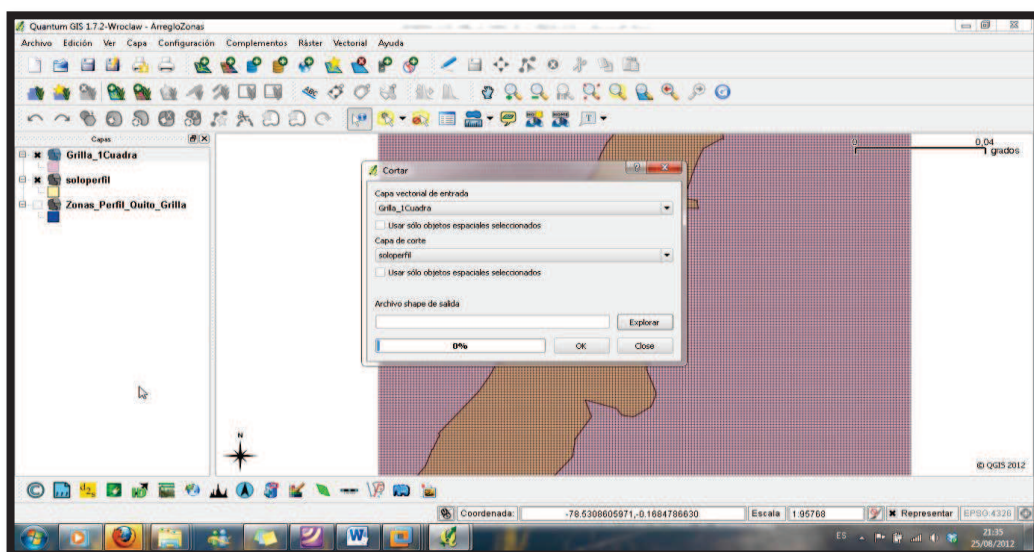


Figura 4.85 Generación Archivo *Shape* con Grilla Ciudad de Quito

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

Luego del proceso de intersección se genera otro archivo *shape*, el cual posee un total de 16661 registros (cuadras de 10000 m²). El proceso final consiste en cargar este archivo en el esquema de la base de datos del presente proyecto (Proceso similar al descrito en la sección 4.4.2. Migración de Datos Geográficos). Con los datos de la matriz en la base de datos y el procedimiento almacenado SECTORIZAR implementado, finalmente se define un formulario donde permita ejecutar el procedimiento SECTORIZAR con el parámetro ZONA definido por el usuario.

4.9 PRUEBAS AL SISTEMA PROTOTIPO CALIDADSMA

4.9.1 PRUEBAS DE FUNCIONALIDAD - USUARIOS

En base a los casos de uso descritos en el capítulo 3, se procede a comprobar la funcionalidad de la aplicación desarrollada.

4.9.1.1 Mapa de la Ciudad de Quito Sectorizado (Capa Base)

EL usuario ingresa a través de la url:

<http://suptel-geo.supertel.gob.ec/calidadSMA/>

Continuando con el flujo para el CDU, se despliega la interfaz de la aplicación donde se puede visualizar el mapa de la ciudad de Quito sectorizado.

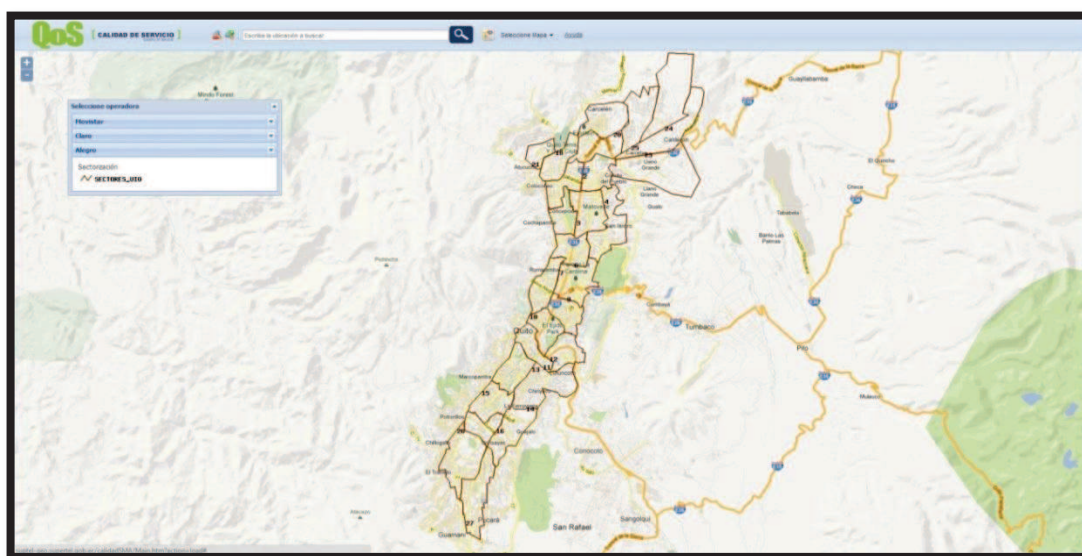


Figura 4.86 Pantalla Inicio Aplicación calidadSMA

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

4.9.1.2 Visualización de Parámetros de QoS, Leyendas e Información

Dentro de la aplicación se escoge una operadora y el respectivo parámetro a visualizar. La aplicación muestra en pantalla las zonas junto con sus respectivos valores porcentuales del parámetro seleccionado, así como la leyenda del gráfico generado y una breve explicación de los parámetros mostrados en pantalla.

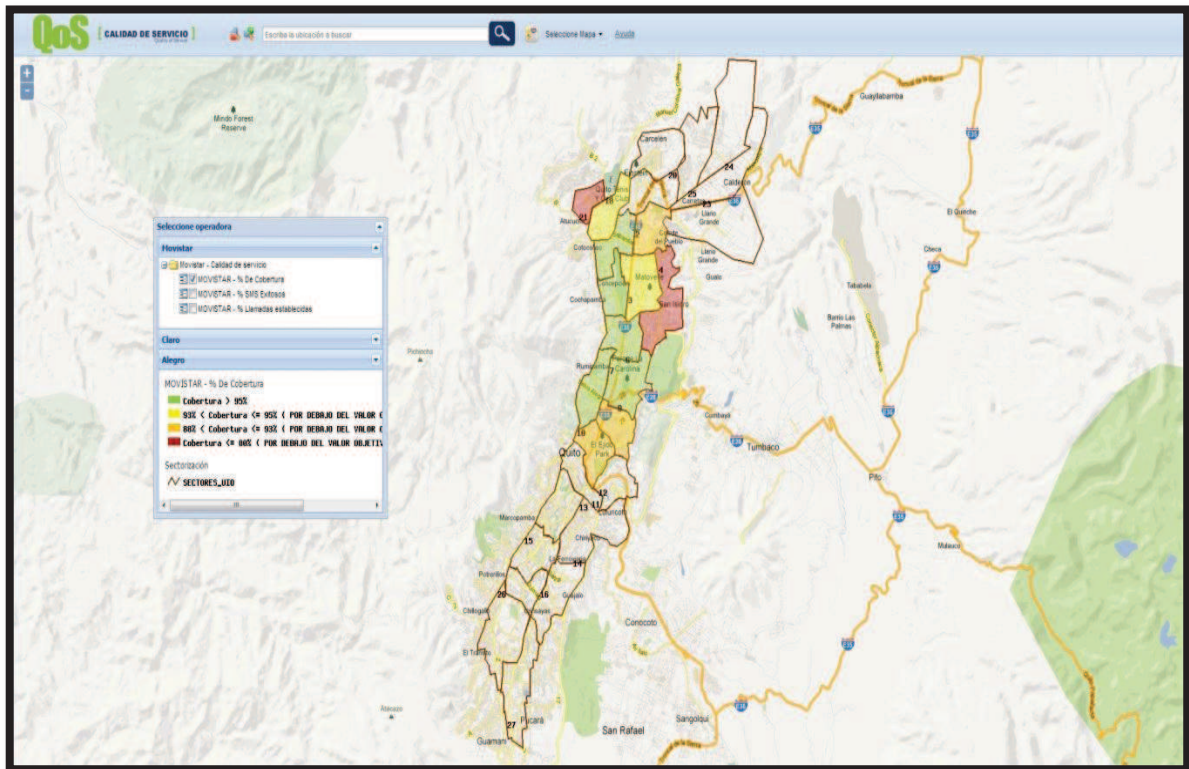


Figura 4.87 Visualización Parámetro Cobertura - Movistar

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

Se puede observar las zonas de Quito coloreadas en función del valor objetivo de medición reportado en el sistema GTM, además de la respectiva leyenda en la zona izquierda de la pantalla (donde se define qué color representa que valor o valores de medición).

4.9.1.3 Visualización de Mediciones de QoE

Dentro de la aplicación, se escoge una operadora y el parámetro QoE a visualizar. La aplicación muestra en pantalla todas las calificaciones otorgadas por parte de los usuarios a manera de puntos, así como la leyenda del gráfico generado y una breve explicación de los parámetros mostrados en pantalla. La Figura 4.88

muestra un ejemplo de cómo se muestran los resultados de las mediciones de QoE.

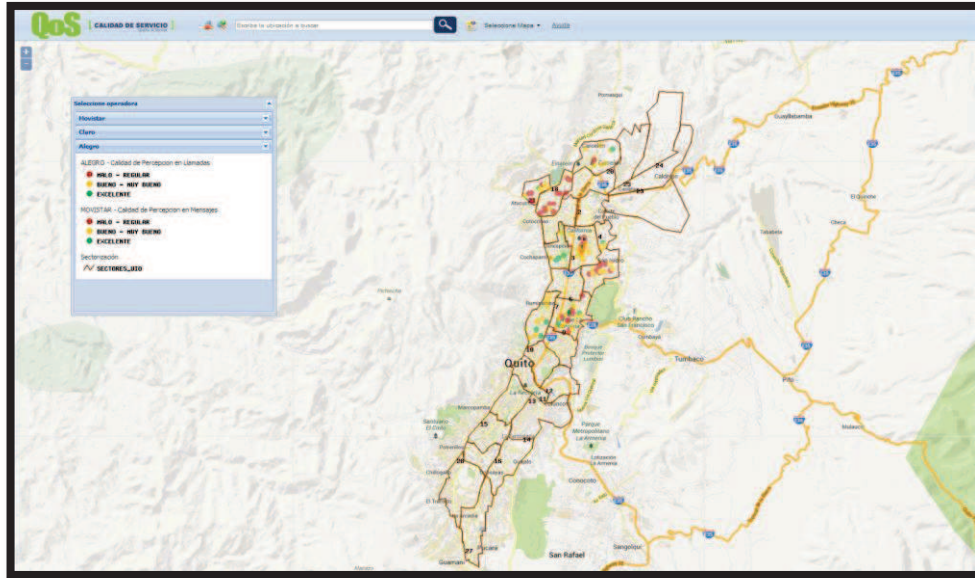


Figura 4.88 Visualización de Mediciones de QoE

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

4.9.1.4 Visualización de Rutas y Radio Bases

Dentro de la misma operadora, se selecciona el ítem radio bases instaladas y se comenzará la carga de la capa con la localización geográfica de cada una de las radio bases instaladas dentro de la ciudad de Quito.

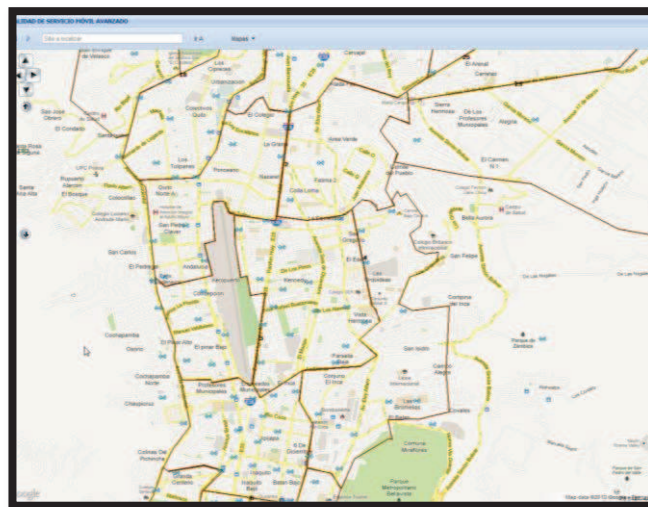


Figura 4.89 Visualización de Radio Bases Instaladas

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

4.9.1.5 Envío de Reclamos hacia el CIR-SUPERTEL

Al momento de realizar un evento *click* sobre cualquier punto del mapa sectorizado de la ciudad de Quito, se despliega una ventana con los datos de mediciones de todos los parámetros sobre ese punto.

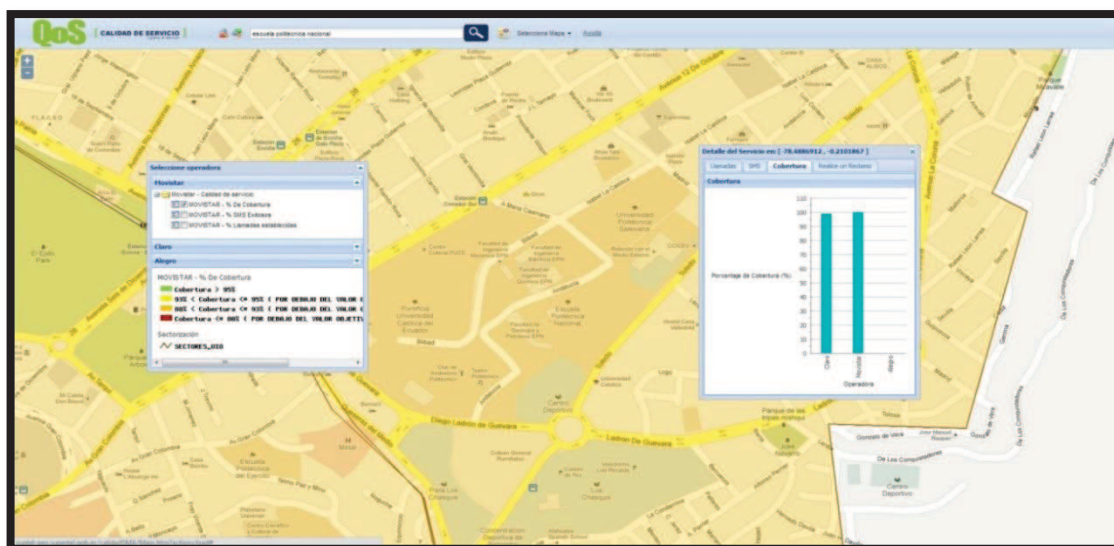


Figura 4.90 Determinación de Parámetros de QoS en un Punto de Quito

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

Y dependiendo si el usuario desea realizar un reclamo puede dar *click* sobre el botón superior que desplegará el siguiente formulario de reclamos.

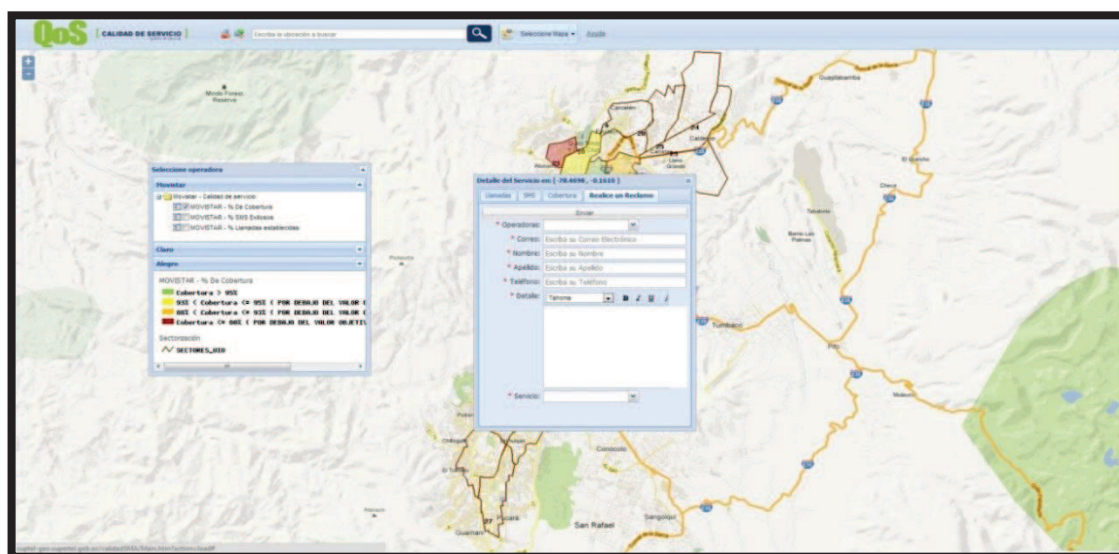


Figura 4.91 Formulario para Registro de Reclamos

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

Dentro de este formulario el usuario debe llenar todos los campos solicitados, para finalmente proceder a enviar el reclamo.

4.9.1.6 Mediciones de la Percepción del Servicio QoE

La Figura 4.92 muestra el formulario para recibir las mediciones de percepción del servicio por parte de los usuarios. Dentro del formulario, el usuario debe escoger la operadora objeto de su calificación, y mediante los controles de selección se pueden otorgar calificaciones del 1 (malo) hasta 5 (excelente).

Figura 4.92 Formulario de Recepción de Mediciones de QoE

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

4.9.2 PRUEBAS COMPARACIÓN EJECUCIÓN

Las pruebas de funcionalidad se la ha realizado a través de un ambiente cliente utilizando las siguientes características:

- Sistema operativo: Windows 7

- Navegador web: *Mozilla Firefox 19.0.2*
- Depurador de código: Complemento *Firebug 1.11.2* para *Mozilla Firefox*
- Además de un servidor, con las siguientes características:
- Contenedor JEE: *Apache Tomcat 7.0.22*
- Servidor de aplicaciones: *WebLogic 12c*

4.9.2.1 Pruebas con Servicios Web

La Figura 4.93 muestra el tiempo de ejecución de un procedimiento almacenado a través del consumo de servicios web publicados en el servidor de aplicaciones WebLogic 12c. Esto logrado a través de análisis y capturas de solicitudes del tipo HTTP GET teniendo como resultado un tiempo de ejecución y muestra de resultados de 935ms.

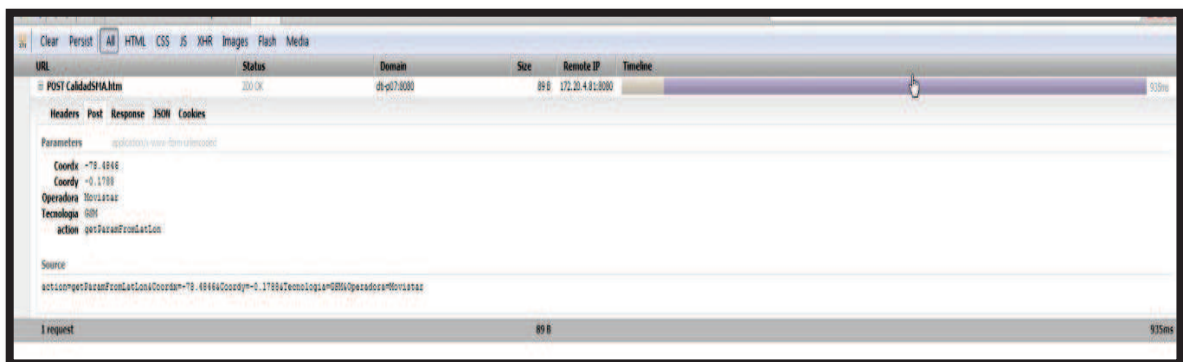


Figura 4.93 Latencia de Ejecución de Procedimientos Almacenados utilizando Servicios Web

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

La Figura 4.94 muestra los resultados obtenidos a través de parámetros contenidos en respuestas a las solicitudes HTTP GET.

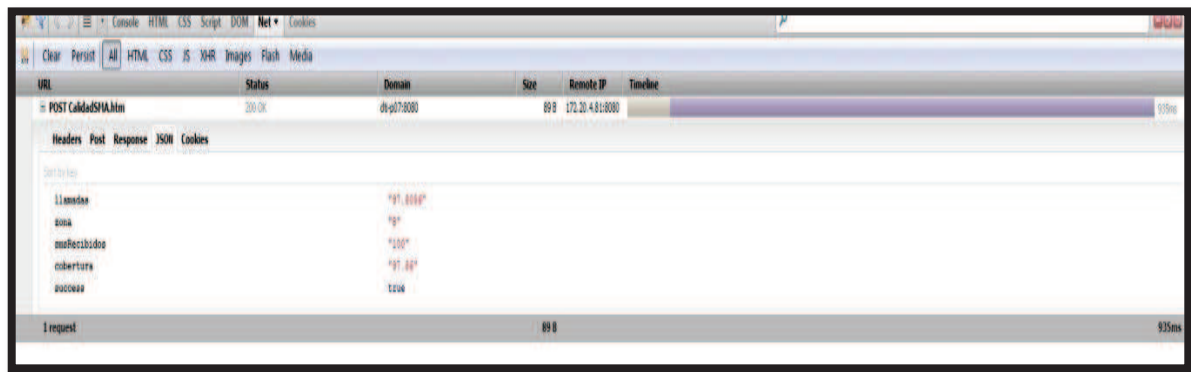


Figura 4.94 Respuesta de Ejecución del Procedimiento Almacenado PARAM_FROM_ZONA

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

4.9.2.2 Prueba con Configuración de Recursos Locales

En la versión inicial de la aplicación se implementó la ejecución de procedimientos almacenados a través de librerías propietarias de Oracle que permiten la ejecución de procedimientos almacenados y funciones del tipo geográfico en bases de datos Oracle, todo esto ejecutado desde un “*servlet*”, haciendo conexiones directas a la base de datos. Así, pudiendo comparar los tiempos de ejecución de un procedimiento almacenado en dos ambientes diferentes. Verificando finalmente que con librerías propietarias el retardo en la ejecución de funciones es de 2.13s.

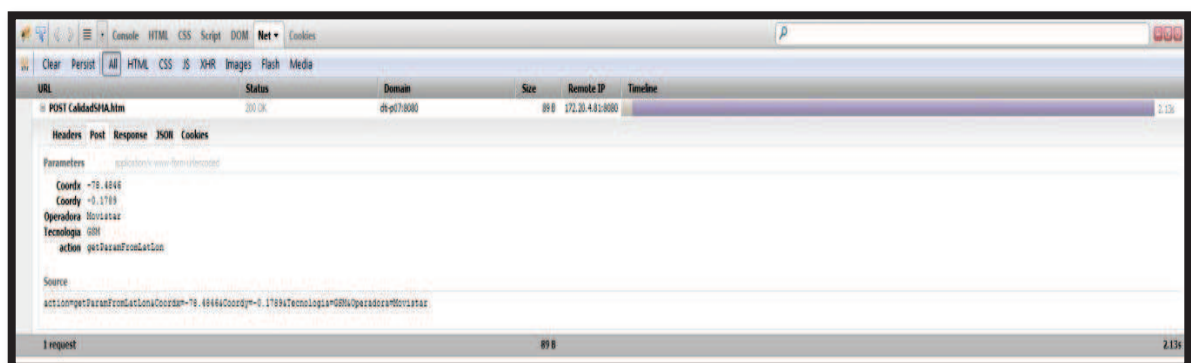


Figura 4.95 Latencia de Ejecución de SP utilizando Recursos Locales

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

En la Figura 4.96 se puede verificar que las respuestas son las mismas que se buscan en los dos ambientes de pruebas.

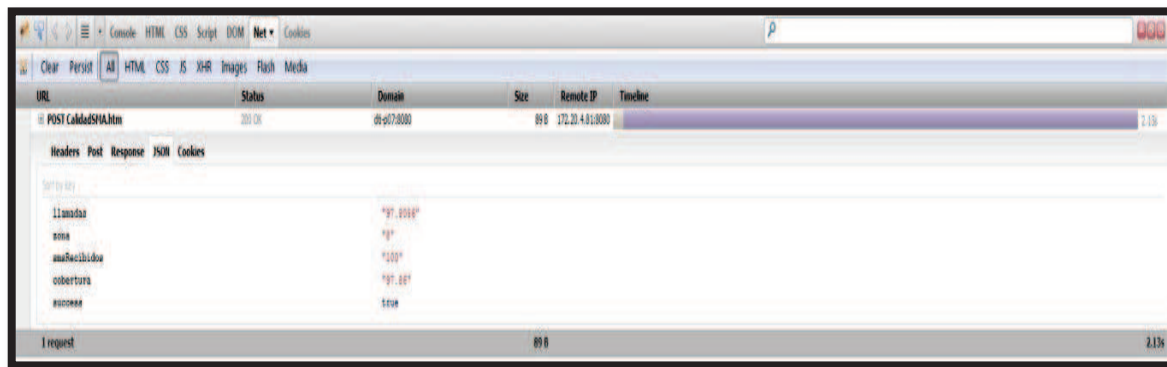


Figura 4.96 Respuesta de Ejecución del Procedimiento Almacenado PARAM_FROM_ZONA

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

4.10 OPTIMIZACIÓN Y MEJORAS

Esencialmente la aplicación desarrollada genera imágenes (mapas) a partir de información espacial (contenida en varios medios como, archivos *shape* y bases de datos). Tomando en cuenta, que la zona con el menor número de registros de coordenadas para su delimitación tiene un total de 1000 pares de coordenadas, procesar este número de datos, renderizarlos⁵⁷, para proyectarlos en la aplicación conlleva un tiempo de procesamiento alto; así también, tomando en cuenta que son 27 zonas el proceso tomará más tiempo aún.

Sin embargo, se puede apreciar que por ejemplo las zonas mostradas son imágenes que no van a variar o no están sujetas de mediciones o cálculos para su modificación (o al menos no en corto plazo), de igual manera los valores de los parámetros de QoS medidos tienen variaciones trimestrales o semestrales. Con este antecedente, y con el fin neto de optimizar tiempos de procesamiento para visualización de zonas y parámetros de QoS, se implementará un servidor caché de mapas.

Gracias al servidor de caché las imágenes de las zonas y parámetros solicitados por la aplicación web se almacenarán en una carpeta definida para este fin, de tal manera que si existe la imagen solicitada no sea necesario renderizarla, sino simplemente proyectar la imagen ya generada.

⁵⁷ Renderizar, se refiere a: generar imágenes basándose en modelos definidos dentro de scripts, archivos fuente, archivos de configuración, etc. Por ejemplo, para el caso específico de la presente aplicación, basado en el archivo *mapfile.map*.

4.10.1 CONFIGURACIÓN SERVIDOR DE CACHÉ DE MAPAS

La configuración del servidor de cache de mapas, reside en el archivo de configuración “*mapproxy.yam*”. La estructura básica de este archivo está conformada por un bloque de configuraciones generales, un bloque de las capas a generarse, un bloque que definirá la forma de almacenar en caché y finalmente un bloque fuente (*sources*), donde se definirá el origen de las imágenes. Para este caso particular serán el conjunto de peticiones (de capas) al servidor de mapas configurado previamente. Inicialmente, se define un *tag* en donde se especifican los SRIDs utilizados en las capas a almacenar, además se definen los formatos de imágenes que se van a utilizar.

```
wms:
  srs: ['EPSG:4326', 'EPSG:900913']
  image_formats: ['image/jpeg', 'image/png']
```

Figura 4.97 Configuración General Servidor Caché de Mapas

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

Para definir una capa primero se define el nombre de la capa (este nombre será el identificador para realizar las solicitudes vía HTTP), y se debe especificar el origen de datos (imágenes) para esta capa utilizando la etiqueta *sources*.

```
# Layer Sectores
- name: qos_sectores
  title: Sectorizacion Quito
  sources: [sectores_cache]
- name: qos_sectores_tesis
  title: Sectorizacion Quito Tesis
  sources: [sectorestesis_cache]
# - name: osm_full_example
#   title: Omniscale OSM WMS - osm.omniscale.net
#   sources: [osm_cache_full_example]
```

Figura 4.98 Definición Capa para Servidor de Caché de Mapas

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

Dentro del bloque que genera el caché están definidos los siguientes parámetros,

Sources (Orígenes de datos), el origen de datos para esta capa la(s) petición(es) al servidor de mapas para una capa específica.

```

caches:
  osm_cache:
    grids: [GLOBAL_MERCATOR, global_geodetic_sqrt2]
    sources: [osm_wms]
  sectores_cache:
    grids: [GLOBAL_MERCATOR, global_geodetic_sqrt2]
    sources: [qos_sectores_source]
  sectorestesis_cache:
    grids: [GLOBAL_MERCATOR, global_geodetic_sqrt2]
    sources: [qos_sectorestesis_source]

```

Figura 4.99 Configuración Caché Mapproxy

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

El bloque denominado *sources* obtiene las imágenes directamente del servidor de mapas. Por lo tanto se debe definir como interactuar con el servidor de mapas; es decir, el tipo de petición (WMS para el caso de la aplicación), la URL a la cual realizar la petición (la del servidor de mapas) y finalmente la capa que desea consultar. Como parámetros adicionales, se definen el origen de las leyendas de las imágenes de mapas.

```

# Source Sectores
qos_sectores_source:
  type: wms
  wms_opts:
    legendgraphic: true
    legendurl: http://192.168.255.137:8080/cgi-bin/WMSUPERTEL?EPSG=90091
  req:
    url: http://192.168.255.137:8080/cgi-bin/WMSUPERTEL?
    layers: SECTORES
    transparent: true

```

Figura 4.100 Configuración Origen de Datos MapProxy

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

Finalmente se debe especificar la ruta de la carpeta o directorio donde se alojarán todas las imágenes de los mapas.

```

# # cache options
cache:
  # where to store the cached images
  base_dir: './cache_data'
  # where to store lockfiles
  lock_dir: './cache_data/locks'
# # request x*y tiles in one step
# meta_size: [4, 4]
# # add a buffer on all sides (in pixel) when requesting
# # new images
# meta_buffer: 80

```

Figura 4.101 Configuración Almacenamiento de Caché MapProxy

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

Dentro de la aplicación web el cambio radica en re-direccionar las peticiones WMS de sectores y parámetros de QoS hacia el nuevo servidor cache de mapas.

```

//CARGA DE LA CAPA PRINCIPAL DEL MAPA - DIBUJA LA SECTORIZACIÓN DE LA CIUDAD
var layerSectores = new OpenLayers.Layer.WMS("Sectorización",
    "http://192.168.255.137:8011/service?",
    {layers: ['qos_sectores'],
      EPSG: 900913,
      format: 'image/png',
      transparent: "true"},
    {isBaseLayer: false,
      visibility: true,
      opacity: '1'}
    );

```

Espacio de Código 4.24 Capa con Origen de Datos del Servidor de Caché de Mapas MapProxy

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

Como se puede observar en el código que genera la sectorización, el proceso consiste en definir la nueva dirección del servidor que retornará las peticiones de imágenes de mapas. Es importante recalcar que continúa siendo una solicitud de tipo WMS. El hecho de que el servidor ahora sea de caché de mapas es irrelevante para la solicitud ya que éste servidor mantiene el estándar WMS-OGC. Cada vez que se realice una petición, esta invocará el servidor de caché. El servidor de caché busca dentro de la carpeta “*cache_data*” la capa solicitada.

Ahora el servidor de caché tiene dos opciones, si la capa existe toma la imagen y la devuelve a la aplicación para su proyección. Si no existe almacenado en caché la imagen solicitada, el servidor de caché realiza una petición al servidor de mapas, a su vez el servidor de mapas solicita los datos a la base de datos para renderizar la imagen solicitada y permitir que el servidor de caché almacene esta imagen para satisfacer futuras peticiones.

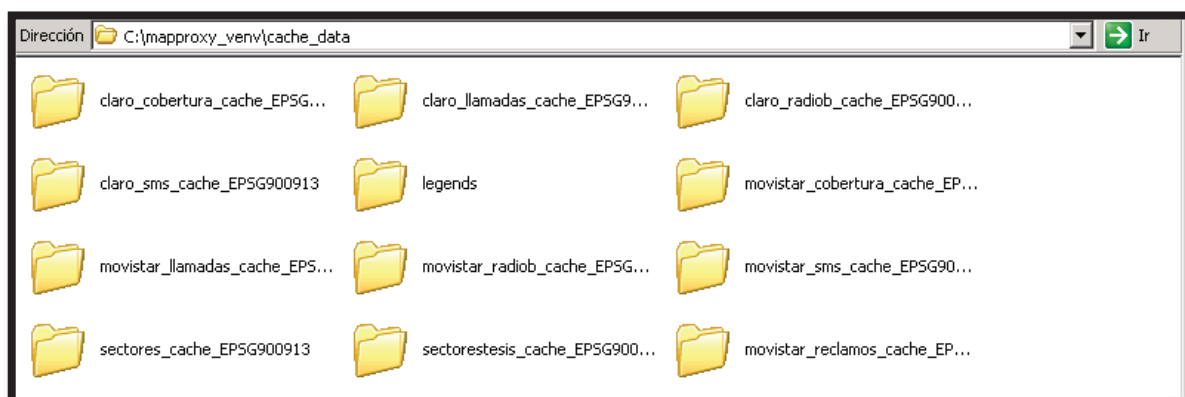


Figura 4.102 Caché Generado en el Servidor MapProxy

Fuente: Esquema de Autoría Propia (Arévalo D., y Guzmán D.).

Es importante recalcar que el servidor caché almacena imágenes inclusive por cada nivel(es) de zoom.

4.11 ANÁLISIS DE COSTOS

Existen varias formas de clasificar costos, en el desarrollo de este análisis se tomará en cuenta a la siguiente clasificación:

- **Costos Directos.**- Costos que se presentan específicamente en la elaboración de un producto o ejecución de un servicio.
- **Costos Indirectos.**- Costos pequeños presentados en la elaboración de un producto o ejecución de un servicio que no están relacionados directamente.

En la Tabla 4.4 y Tabla 4.5 se detallan los costos incurridos en el diseño, desarrollo e implementación del Proyecto “Desarrollo de un mapa interactivo para

la ciudad de Quito que permita visualizar los parámetros de QoS para la SUPERTEL”.

Tabla 4.4
Costos Directos del Proyecto

COSTOS DIRECTOS	
<u>Honorarios: Desarrollador Java Senior</u>	
Remuneración mensual promedio estimada:	USD 1.340,00
Número de Desarrolladores Java Senior:	2 personas
Tiempo estimado de trabajo - jornada completa:	4 meses
Total costo por honorarios a Desarrolladores Java Senior:	USD 10.720,00
<u>Asesoría Externa: Aplicaciones Web GEO referencias integradas con base de datos espaciales</u>	
Remuneración mensual promedio estimada:	USD 2.190,00
Numero de asesores externos:	1 persona
Tiempo estimado de asesoría:	6 horas
Total costo por Asesoría Externa (1 semana)	USD 410,62
<u>Honorarios: Investigación</u>	
Remuneración mensual promedio estimada:	USD 1340,00
Número de Analistas:	2 personas
Tiempo estimado de trabajo - jornada completa:	6 meses
Total costo por Investigaciones referentes a temas para desarrollo del proyecto:	USD 16.080,00

Fuente: Tabla de Autoría Propia (Arévalo D., y Guzmán D.).

Tabla 4.5
Costos Indirectos del Proyecto

COSTOS INDIRECTOS	
<u>Depreciaciones</u>	
2 Computadores para desarrollo de software	
Costo Inicial:	USD 1.200,00 cada PC
Valor residual estimado:	USD 400,00 cada PC
Tiempo vida útil:	3 años
Tiempo de depreciación:	2 años
Costo por depreciación de computadores:	USD 1.066,67
1 Servidor de prueba	
Costo Inicial:	USD 2.700,00 cada PC
Tiempo vida útil:	3 años
Tiempo de depreciación:	2 años
Costo por depreciación de servidor para pruebas	USD 1.800,00
Total costo por depreciaciones:	USD 2.866,67
<u>Otros Costos Indirectos</u>	
Estimado consumo energía eléctrica	USD 1.710,00
Estimado consumo internet	USD 540,00
Varios (copias, cables, movilización, etc.)	USD 600,00
Total costo por Otros Costos Indirectos	USD 2.850,00

Fuente: Tabla de Autoría Propia (Arévalo D., y Guzmán D.).

Tabla 4.6
Resumen Análisis de Costos del Proyecto

RESUMEN DE COSTOS			
		DETALLE	VALOR TOTAL
Costos Directos		Honorarios a Desarrolladores Java Senior:	USD 10.720,00
		Asesoría Externa	USD 410,62
		Investigaciones referentes a temas para desarrollo del proyecto:	USD 16.080,00
Costos Indirectos		Depreciaciones:	USD 2.866,67
		Otros Costos Indirectos	USD 2.850,00
	TOTAL COSTOS		USD 32.927,29

Fuente: Tabla de Autoría Propia (Arévalo D., y Guzmán D.).

Los salarios asignados en la Tabla 4.4 están basados en la tabla de escalas salariales proporcionada por el Servicio de Rentas Internas (Acuerdo Ministerial No. MRL-2010-00022 y MRL 2011-000020 DE NJS) del Ecuador. El costo del desarrollo del proyecto es de 32.927,29 USD durante un período de un año.

Contando con el apoyo de la entidad de control SUPTERTEL la ejecución y puesta en ambiente de producción del presente proyecto es totalmente viable. Tomando en cuenta que éste proyecto es pionero a nivel nacional como herramienta para que el usuario exprese su percepción respecto del servicio de telefonía celular móvil que recibe. Así como, para poder visualizar de forma asimilable los procesos de control que realiza la SUPERTEL.

Además la arquitectura propuesta es una base empresarial para futuros proyectos como Geoportales o SIGs en ambientes Web.

5. CONCLUSIONES Y RECOMENDACIONES

5.1 CONCLUSIONES

A través de implementaciones basadas en software de distribución libre y plataformas JAVA desarrolladas para ambientes empresariales como Edición Empresarial Java 6 - *Java Enterprise Edition 6* (JEE6) se ha logrado orquestar plataformas y servicios funcionando en ambientes de producción en la SUPERTEL, para tener un insumo robusto que permita la visualización de parámetros de calidad de servicio móvil avanzado en la ciudad de Quito; los cuales son obtenidos, procesados y almacenados por la SUPERTEL.

Haciendo uso de una metodología de desarrollo especificada por la entidad auspiciante, metodología de desarrollo RUP, se ha logrado tener el control de los planes de despliegue de software para ambientes de pruebas y producción; logrando publicar un servicio administrable, configurable y escalable gracias a la documentación, capacitación y configuración entregadas en un lenguaje común (UML). Esto gracias a la especificación y documentación de un marco de desarrollo, el cual determina el modelado del negocio, el análisis de requisitos y el diseño, esto expuesto en escritos como modelo del dominio, modelo de casos de uso, diagramas de secuencia, especificación complementaria, modelo de diseño, arquitectura y modelo de datos. Permitiendo a los recipientes del proyecto tomar control sobre la aplicación en cuanto a códigos fuente, configuración y administración.

En el estudio del desarrollo de aplicaciones web se pudo comparar arquitecturas de desarrollo, teniendo la limitante de uso de estándares e implementaciones convenidos en la entidad auspiciante. En lo referente a modelos de implementación se ha utilizado programación de paquetes siguiendo una estructura de capas, Modelo Vista Controlador – *Model View Controller* (MVC), esto seleccionado después del análisis de requerimientos, en los cuales se especifica una aplicación web orientada a la interacción de usuarios y mapas. Así también, se ha aplicado el estándar JEE6 que especifica la identificación de la capa de persistencia (API de Persistencia

JAVA - *Java Persistence API* - JPA), capa de servicios (Objetos Java Empresariales - *Enterprise Java Beans* - EJB), esto a causa de publicación de aplicaciones en servidores basados en estándares Java Empresariales.

La constante evolución tecnológica y el nacimiento de nuevas necesidades han influido en el creciente ofrecimiento de software de desarrollo, pudiendo éstos ser de naturaleza de fuente abierta (*open source*), sin licenciamiento, con licenciamiento parcial y software propietario. Mismo crecimiento de software libre de fuente abierta, que ha empezado a ser estandarizado, controlado y verificado, actualmente no consigue llenar las expectativas de desarrolladores en lo referente a soporte, interoperabilidad y calidad de los desarrollos; esto de la experiencia de trabajo con IDEs de desarrollo como Eclipse (Indigo, Helios y Juno), además de estándares de desarrollo como JEE5 y JEE6, y software *open source* como Hibernate, Spring Framework y JavaServer Faces. Especialmente para procesos específicos, como por ejemplo la ejecución de procedimientos almacenados en bases de datos Oracle desde una capa de persistencia existen varias soluciones de software propietario (DataNucleus implementación propietaria de JPA 2.1) y no una implementación de software libre (JPA 2.0 estándar JEE6).

Después de haber realizado el estudio del procesamiento de parámetros de calidad de servicio móvil avanzado en Ecuador, se escogió los parámetros que se considera de mayor facilidad de entendimiento para los usuarios del servicio móvil avanzado, los cuales son: cobertura, porcentaje de establecimiento de llamadas y porcentaje de mensajes de texto enviados con éxito.

En lo referente al cumplimiento de parámetros de QoS estipulados en anexos a los contratos de concesión se ha determinado que la operadora CONECEL – Claro está más apegada en el cumplimiento de los valores objetivos de parámetros de QoS.

Los datos de mediciones de QoE que posee tanto la SUPERTEL como cada una de la operadoras de telefonía móvil del Ecuador no reflejan la percepción del servicio por parte de los usuarios (valores subjetivos). Por este motivo se

puede aseverar que actualmente ningún involucrado se encuentra socializando mediciones de QoE referentes al servicio móvil avanzado en el Ecuador.

Es de importante relevancia la definición de un marco de desarrollo que especifique como se detallarán los requerimientos, para de ésta manera tener formalizada una metodología de desarrollo que determine el alcance de la funcionalidad a ser desarrollada.

El detalle de casos de uso, diagramas de secuencia y modelo de dominio, ayuda en la estructuración de paquetes y espacios de nombres, haciendo de esta manera el entendimiento del código fuente inherente a funcionalidades e implementaciones, para así dar al sistema continuidad y facilidad de mantenimiento y administración de un servicio en un ambiente de producción.

Gracias al análisis de librerías de libre distribución se logró seleccionar las herramientas y controles que mejor se acoplan al requerimiento inicial de desarrollo, estructurándose en un modelo de capas a nivel de cliente y servidor. Determinando la capa de presentación en base a librerías JavaScript de libre distribución como *OpenLayers*, *ExtJS*, *GeoExt* y *Proj4JS*; además de librerías Java como *JasperReports*. Éstas fueron seleccionadas debido a su fácil integración entre sí, extensibilidad, y cantidad de implementaciones de controles ofrecidas. Características que garantizaron un ambiente estándar de manipulación y navegación sobre datos, controles y eventos geo-espaciales.

Debido a la variedad de implementaciones de software libre, así como de servidores de aplicaciones de libre distribución, se ha escogido la última versión estable de la especificación estándar JEE6 para el desarrollo de una aplicación web empresarial en la capa de persistencia y servicios, garantizando de esta manera la interoperabilidad de las compilaciones en diferentes implementaciones de servidores JEE certificados.

Debido a la ausencia de una base de datos espacial que almacene la información de la sectorización de la ciudad de Quito, se ha evidenciado carencia de integridad en el dominio y entidad en las geometrías que describen la sectorización actual de la ciudad de Quito.

El modelamiento de objetos y tablas de bases de datos relacionales permiten la mejor administración y manipulación de la información, para el caso específico de este proyecto la implementación de una base de datos con información geográfica permite la generación de reportes con información geográfica, centralización de la información geográfica y desarrollo de procedimientos sobre la actualización de información geográfica contenida en una base de datos.

En ambientes de desarrollo de software grupales es necesario trabajar con un esquema de versionamiento y respaldo compatibles con las interfaces de desarrollo utilizadas, para mantener los códigos fuentes en versiones estables.

En la definición de un esquema de interacción con plataformas diferentes a las que se está desarrollando, es importante tomar en cuenta los estándares o especificaciones en los que las librerías están basadas como es el caso de la implementación JAX-WS para servicios web basada en el estándar JEE6.

Si bien la notación de los servicio web de mapas lleva explícitamente en su nombre el termino Servicios Web de Mapas – *Web Map Services (WMS)*, durante el desarrollo de la aplicación web se pudo verificar, que no siguen el estándar definido para los servicios web, esto quiere decir que carecen de la especificación de un Lenguaje de Descripción de Servicios Web - *Web Service Definition Language (WSDL)* y así también no utilizan un Protocolo Simple de Acceso a Objetos - *Simple Object Access Protocol (SOAP)*.

5.2 RECOMENDACIONES

1. Se recomienda responsabilizar de la administración y mantenimiento de la información geográfica a los expertos en parámetros técnicos y zonificación, a través del aprovisionamiento de un Sistema de Información Geográfica (SIG) de escritorio robusto.
2. Para evitar ataques de códigos maliciosos que ejecuten funciones *JavaScript* se recomienda la implementación de un control Prueba de *Turing* pública y automática para diferenciar máquinas y humanos -

Completely Automated Public Turing test to tell Computers and Humans Apart (CAPTCHA) en el formulario de ingreso de reclamos para diferenciar entre la manipulación de esta entrada de información por un humano o por una máquina.

3. La SUPERTEL debe implementar procesos para mediciones de QoE, estos procesos se deben fundamentar en la experiencia del cliente frente al servicio que recibe. Deben ser datos proporcionados directamente por parte de los usuarios. Un proceso similar debería ser ejecutado por parte de las operadores de telefonía móvil que actualmente brindan sus servicios dentro del territorio nacional.
4. La falta de uso de una topología para el proceso de sectorización de la ciudad de Quito realizado por la SUPERTEL ha provocado que lugares que pertenecen a la ciudad de Quito, y están dentro del área de cobertura garantizada por las operadoras no estén contempladas dentro de ningún sector de la ciudad, es por esto que se considera necesario la implementación de una topología de información geográfica para relacionar toda la información contenida en la base de datos geográfica, evitando inconsistencias a nivel de datos.
5. Debido a que la aplicación está orientada a la web, se considera atractivo la definición y presentación de un parámetro de calidad de servicio móvil que haga referencia a la transmisión de datos.
6. Se debe realizar el procesamiento y almacenamiento de mediciones y análisis de resultados del parámetro de calidad de servicio Medida de la Conversación - *Measurement of Speech* (MoS).

REFERENCIAS BIBLIOGRÁFICAS

1. ETSI EG 202 057-1 V1.3.1 (Julio de 2008) *Speech Processing, Transmission and Quality Aspects (STQ); User related QoS parameter definitions and measurements; Part 1: General.*
2. ETSI EG 202 057-2 V1.3.2 (Abril de 2011) *Speech and multimedia Transmission Quality (STQ) User related QoS parameter definitions and measurements; Part 2: Voice telephony, Group 3 fax, modem data services and SMS.*
3. ETSI TS 102 250-1 V1.2.1 (Marzo de 2007) *Speech Processing, Transmission and Quality Aspects (STQ); QoS aspects for popular services in GSM and 3G networks; Part 1: Identification of Quality of Service criteria.*
4. ETSI TS 102 250-2 V1.6.2 (Septiembre de 2008) *Speech Processing, Transmission and Quality Aspects (STQ); QoS aspects for popular services in GSM and 3G networks; Part 2: Definition of Quality of Service parameters and their computation.*
5. GSM 03.40. 96 (Julio de 1996) *Digital cellular telecommunications system (Phase 2+); Technical realization of the Short Message Service (SMS) Point-to-Point (PP).*
6. UIT-T P.800. SERIE P: (Agosto de 1996.) *CALIDAD DE TRANSMISIÓN TELEFÓNICA. Métodos de evaluación objetiva y subjetiva de la calidad Métodos de determinación subjetiva de la calidad de transmisión.*
7. Meyer, Christopher., Schwager, Andre. (Febrero, 2007) *Understanding Customer Experience.* Harvard Business Review OnPoint Versión Electrónica Disponible en:
<http://www.dlss.univr.it/documenti/Avviso/all/all845856.pdf>. Diciembre 2011

8. Tanenbaum, Andrew. (2003) *Sistemas operativos modernos*. Segunda edición. México. Pearson Education.
9. Ramos Martín, Alicia. y Ramos Martín, Jesús. *Aplicaciones web*. Versión *Electrónica*. Disponible en:
http://books.google.com.ec/books?id=LXs3YIMoeNgC&printsec=frontcover&hl=es&source=gbs_ge_summary_r&cad=0#v=onepage&q&f=false. Agosto de 2012.
10. Luján Mora, Sergio. (31 de Octubre de 2002) Programación de aplicaciones web: historia, principios básicos y clientes web. Versión Electrónica. Disponible en:
http://books.google.com.ec/books/about/Programaci%C3%B3n_de_aplicaciones_web.html?id=r9CqDYh2-loC&redir_esc=y. Agosto de 2012.
11. Stallman, Richard M. (2010) *Free Software, Free Society: Selected Essays of Richard M. Stallman* Segunda edición. Free Software Foundation.
12. Ballester, Eva Gómez y Barco, Patricio Martínez (Julio 2006) *Bases de datos 1*. Versión Electrónica. Disponible en: <http://www.dlsi.ua.es/asignaturas/bd>. Junio de 2012.
13. ETSI TS 102 250-2 V1.6.2 (2008-09). *Speech Processing, Transmission and Quality Aspects (STQ); QoS aspects for popular services in GSM and 3G networks; Part 2: Definition of Quality of Service parameters and their computation*
14. LEY ESPECIAL DE TELECOMUNICACIONES Y SU REFORMA. Ley No. 184
Registro Oficial No. 770 del 30 de agosto de 1995
15. REGLAMENTO GENERAL A LA LEY ESPECIAL DE
TELECOMUNICACIONES REFORMADA (Decreto No. 1790)

16. Rumbaugh, J., Jacobson, I., & Booch, G. (2000). EL Lenguaje Unificado de Modelado. *Manual de Referencia*. (pp. 21-34). Madrid: Pearson Education.
17. SUPERTEL. Anexo 5, Contratos de Concesión proporcionados por la SUPERTEL. Enero 2012.
18. Camps, P., Casillas, L., Costal, D., Gibert, M., Martin, C., Pérez, O. (2005) *Bases de datos*. Primera edición. España. Eureka Media.

REFERENCIAS BIBLIOGRÁFICAS WEB

19. IBM software group. *EJB 3.0 Overview 2007 IBM Corporation*. Versión Electrónica. Disponible en:
http://publib.boulder.ibm.com/infocenter/ieduasst/v1r1m0/index.jsp?topic=/com.ibm.iea.wasfpejb/wasfpejb/6.1/Overview/WASv61_EJB3FP_EJB3Overview/player.html. Diciembre de 2012.
20. Rod, Johnson., Juergen, Hoeller., Keith, Donald., & Colin, Sampaleanu. *Introduction to Spring Web MVC framework*. (2007). Versión Electrónica. Disponible en:
<http://static.springsource.org/spring/docs/3.0.x/reference/mvc.html>
 Diciembre 2012.
21. IBM software group. *Java Persistence API (JPA) Overview*. Versión Electrónica. Disponible en:
http://publib.boulder.ibm.com/infocenter/ieduasst/v1r1m0/index.jsp?topic=/com.ibm.iea.wasfpejb/wasfpejb/6.1/Overview/WASv61_EJB3FP_EJB3Overview/player.html. Diciembre 2012.
22. Oracle. *Oracle Spatial Converter (shp2sdo)*. (2012) Versión Electrónica. Disponible en: <http://www.oracle.com/technetwork/database/enterprise-edition/downloads/using-shp2sdo-134890.txt> . Marzo 2012.

23. Oracle. *Oracle Spatial Developer's Guide*. (2012) Versión Electrónica. Disponible en:
http://docs.oracle.com/cd/B28359_01/appdev.111/b28400/sdo_intro.htm.
Marzo 2012.
24. Open Source Geospatial Foundation. *MapServer 6.2.0 Documentation*. (2012) Versión Electrónica. Disponible en:
<http://mapserver.org/documentation.html>. Septiembre de 2013.
25. MapProxy Org. *MapProxy Documentation*. (2012) Versión Electrónica. Disponible en: <http://mapproxy.org/docs/1.5.0/>. Septiembre de 2013.
26. GeoExt Community. *GeoExt API Reference*. Versión Electrónica. Disponible en: <http://trac.osgeo.org/proj4js/wiki/UserGuide>. Septiembre de 2013.
27. ESRI Shapefile Technical Description. Versión Electrónica. Disponible en:
<http://www.esri.com/library/whitepapers/pdfs/shapefile.pdf>. Marzo 2012.
28. PostGis. *PostGis Documentation*. (2012) Versión Electrónica. Disponible en:
<http://postgis.net/documentation>. Marzo 2012.
29. Open Geospatial Consortium Inc. *OpenGIS® Implementation Standard for Geographic information - Simple feature access - Part 1: Common architecture*. (2011-05-28) Versión Electrónica. Disponible en:
http://portal.opengeospatial.org/files/?artifact_id=25355%E2%80%8E.
Marzo 2012.
30. Open GIS Consortium Inc. *Web Map Service Implementation Specification*. (2013-10-24) Versión Electrónica. Disponible en:
https://portal.opengeospatial.org/files/?artifact_id=1081&version. Marzo 2012.

31. *Introducción a la Normalización Información Geográfica Familia ISO 19100*. Versión Electrónica. Disponible en:
http://coello.ujaen.es/Asignaturas/pcartografica/Recursos/IntroduccionNormalizacion_IG_FamilialISO_19100_rev1.pdf. Marzo 2012.
32. *Qué es un framework web*. Versión Electrónica. Disponible en:
http://www.lsi.us.es/~javierj/investigacion_ficheros/Framework.pdf.
Diciembre 2011.
33. Bail, Sumanth. *Application Program Interface (API)*. Versión Electrónica. Disponible en:
<http://www.techrepublic.com/resource-library/whitepapers/application-program-interface-api-a-path-to-seamless-software-integration/>. Diciembre 2011.
34. J.L. Garrido, M. Gea, M. Noguera, M. González, J.A. Ibáñez. *Una Propuesta Arquitectónica para el Desarrollo de Aplicaciones Colaborativas*. Versión Electrónica. Disponible en: <http://www.aipo.es/articulos/3/24.pdf>. Diciembre 2011.
35. Fernández-Coppel, Ignacio. (2001). *Localizaciones Geográficas El Datum*. Valladolid: Universidad de Valladolid. Versión Electrónica. Disponible en:
<http://www.elgeomensor.cl/downloads/topografia%20y%20geodesia/index.php?file=datum.pdf>. Enero 2012.