



UNIVERSIDAD TÉCNICA DEL NORTE
FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS
CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES

TEMA

**SISTEMA DE CONTROL DEL PARQUEO TARIFADO (SISMERT) PARA EL
GOBIERNO AUTÓNOMO DESCENTRALIZADO DE LA CIUDAD DE IBARRA**

Autor: Luis Rodolfo Guamán Guamán

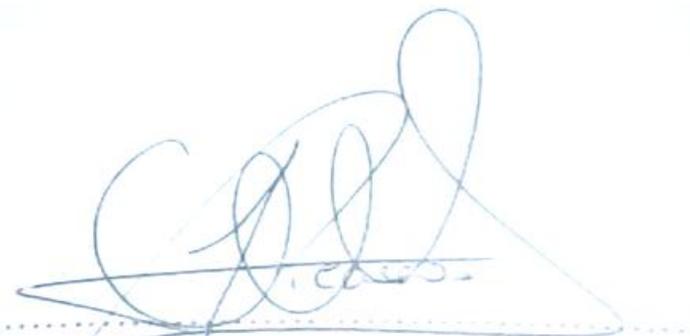
Director: Ing. Irving Reascos

Ibarra – Ecuador

2013

CERTIFICACIÓN

Certifico que la Tesis “SISTEMA DE CONTROL DEL PARQUEO TARIFADO (SISMERT) PARA EL GOBIERNO AUTÓNOMO DESCENTRALIZADO DE LA CIUDAD DE IBARRA” ha sido realizada en su totalidad por el señor: Luis Rodolfo Guamán Guamán portador de la cédula de identidad número: 100324984-2.



Ing. Irving Reascos
Director de la Tesis

CERTIFICACIÓN

Ibarra 21 de Enero del 2013

Señores
UNIVERSIDAD TÉCNICA DEL NORTE
Presente

De mis consideraciones.-

Siendo auspiciantes del proyecto de tesis del Señor LUIS RODOLFO GUAMÁN GUAMÁN con CI: 100324984-2 quien desarrolló su trabajo con el tema "SISTEMA DE CONTROL DEL PARQUEO TARIFADO (SISMERT) PARA EL GOBIERNO AUTÓNOMO DESCENTRALIZADO DE LA CIUDAD DE IBARRA", me es grato informar que se han superado con satisfacción las pruebas técnicas y la revisión de cumplimiento de los requerimientos funcionales, por lo que se recibe el proyecto como culminado y realizado por parte del señor LUIS RODOLFO GUAMÁN GUAMÁN.

El señor LUIS RODOLFO GUAMÁN GUAMÁN puede hacer uso de este documento para los fines pertinentes en la Universidad Técnica del Norte.

Atentamente,


Ing. Paul Barahona
DIRECTOR TIC



UNIVERSIDAD TÉCNICA DEL NORTE



CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE INVESTIGACIÓN

A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

Yo, LUIS ROLDOLFO GUAMÁN GUAMÁN, con cedula de identidad Nro. 100324984-2, manifiesto mi voluntad de ceder a la Universidad Técnica del Norte los derechos patrimoniales consagrados en la ley de propiedad intelectual del Ecuador, artículo 4, 5 y 6, en calidad de autor del trabajo de grado denominado: **“SISTEMA DE CONTROL DEL PARQUEO TARIFADO (SISMERT) PARA EL GOBIERNO AUTÓNOMO DESCENTRALIZADO DE LA CIUDAD DE IBARRA”**, que ha sido desarrollada para optar por el título de Ingeniería en Sistemas Computacionales, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En mi condición de autor me reservo los derechos morales de la obra antes mencionada, aclarando que el trabajo aquí descrito es de mi autoría y que no ha sido previamente presentado para ningún grado o calificación profesional.

En concordancia suscribo este documento en el momento que hago entrega del trabajo final en formato impreso y digital a la biblioteca de la Universidad Técnica del Norte

Firma

Nombre: LUIS RODOLFO GUAMÁN GUAMÁN

Cédula: 100324984-2

Ibarra a los 4 días del mes de marzo del 2013



UNIVERSIDAD TÉCNICA DEL NORTE

BIBLIOTECA UNIVERSITARIA

AUTORIZACIÓN DE USO Y PUBLICACIÓN A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

1. IDENTIFICACIÓN DE LA OBRA

La UNIVERSIDAD TÉCNICA DEL NORTE dentro del proyecto Repositorio Digital institucional determina la necesidad de disponer los textos completos de forma digital con la finalidad de apoyar los procesos de investigación, docencia y extensión de la universidad.

Por medio del presente documento dejo sentada mi voluntad de participar en este proyecto, para lo cual ponemos a disposición la siguiente investigación:

DATOS DE CONTACTO	
CEDULA DE IDENTIDAD	1003249842
APELLIDOS Y NOMBRES	LUIS RODOLFO GUAMÁN GUAMÁN
DIRECCIÓN	Ibarra-Av. El Retorno y Nazacoto Puento
EMAIL	gluis4229@hotmail.com
TELÉFONO FIJO	
TELÉFONO MOVIL	0994366732

DATOS DE LA OBRA	
TITULO	“SISTEMA DE CONTROL DEL PARQUEO TARIFADO (SISMERT) PARA EL GOBIERNO AUTÓNOMO DESCENTRALIZADO DE LA CIUDAD DE IBARRA”
AUTOR	LUIS RODOLFO GUAMÁN GUAMÁN
FECHA	04 DE MARZO DEL 2013
PROGRAMA	PREGRADO
TITULO POR EL QUE	INGENIERÍA EN SISTEMAS COMPUTACIONALES
DIRECTOR	ING. IRVING REASCOS

2. AUTORIZACIÓN DE USO A FAVOR DE LA UNIVERSIDAD

Yo, LUIS RODOLFO GUAMÁN GUAMÁN, con cedula de identidad Nro. 100324984-2, en calidad de autor y titular de los derechos patrimoniales de la obra o trabajo de grado descrito anteriormente, hago entrega del ejemplar respectivo en forma digital y autorizo a la Universidad Técnica del Norte, la publicación de la obra en el Repositorio Digital Institucional y el uso del archivo digital en la biblioteca de la universidad con fines académicos, para ampliar la disponibilidad del material y como apoyo a la educación, investigación y extensión, en concordancia con la Ley de Educación Superior Artículo 143.


.....

Firma

Nombre: LUIS RODOLFO GUAMÁN GUAMÁN

Cédula: 100324984-2

Ibarra a los 4 días del mes de marzo del 2013

DEDICATORIA

A DIOS, por darme vida y hacer realidad un sueño para seguir creciendo, y la oportunidad de compartir con mi familia.

A mis Padres Luis y María por traerme a este mundo y llenarme de mucho amor, porque en cada momento han estado junto a mí brindándome de su apoyo y comprensión, que ha sido incondicional, y sé que seguiré contando con el apoyo de ellos en cada una de las metas que deseo alcanzar en mi vida.

A mis Hermanos y Hermanas que han estado junto a mí regalándome muchos momentos felices en mi vida, y motivándome para alcanzar esta meta.

Esta Meta no hubiese sido posible cumplirla sin el apoyo de cada uno de ustedes, gracias por todo y principalmente por formar parte de mi vida.

Luis LG...

AGRADECIMIENTOS

A la Universidad Técnica del Norte, a la Facultad de Ingeniería en Ciencias Aplicadas por haberme acogido en sus aulas y tener la oportunidad de formarme como profesional al brindarme conocimientos que me servirán para ejercer mi profesión.

Al GAD-I por darme la oportunidad de desarrollar mi proyecto de Tesis y conjuntamente por que en esta institución adquiriré experiencia profesional junto a los compañeros que me brindaron su apoyo para lograr finalizar el Proyecto de Tesis.

Al Ing. Irving Reascos por su apoyo incondicional porque ha guiado la realización del proyecto con su experiencia profesional y como persona admiro mucho su constancia y los valores que le caracterizan.

Al Ing. Rodrigo Naranjo que estuvo guiando con sus conocimientos la realización del proyecto, durante la etapa inicial.

A mi familia por que día a día me dieron sus mejores deseos para culminar mi proyecto.

Un sincero agradecimiento a Leidy por formar parte de mi vida y brindarme de los mejores consejos para seguir luchando y alcanzar mi meta.

Tabla de Contenidos

CERTIFICACIÓN	ii
CERTIFICACIÓN	iii
CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE INVESTIGACIÓN	iv
AUTORIZACIÓN DE USO Y PUBLICACIÓN A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE	v
DEDICATORIA	vii
AGRADECIMIENTOS	viii
Tabla de Contenidos.....	ix
Índice de Tablas.....	xiv
Índice de Figuras	xvi
Resumen	xix
Summary	xx
Introducción.....	1
1. Antecedentes.....	1
2. GAD-I.....	1
3. SISMERT.....	1
4. Base Legal de la Institución.....	2
❖ Misión.....	2
❖ Visión.....	2
❖ Objetivos Generales.....	3
❖ Objetivos Específicos.....	3
5. Organigrama Estructural por Procesos.....	4
6. Área SISMERT.....	6
7. Problema.....	7
8. Objetivos.....	8
❖ Objetivo General.....	8
❖ Objetivos Específicos.....	8
9. Justificación.....	8
10. Alcance.....	9
1. Capítulo I.....	10
Marco Teórico.....	10

1.1. Herramientas de Desarrollo	10
1.1.1. Base de Datos.....	11
❖ PostgreSQL.....	11
1.1.2. Servidores de Aplicaciones.....	13
❖ HTTP Apache.....	13
❖ Tomcat.....	14
1.1.3. Lenguajes de Programación.....	15
❖ CSS.....	15
❖ PHP.....	16
❖ JAVASCRIPT.....	17
1.1.4. Frameworks.....	18
❖ Symfony 1.16.....	19
❖ Doctrine.....	20
❖ Ext JS 4.....	21
❖ Sencha Touch 2.....	25
1.1.5. Reportes.....	28
❖ JasperReports.....	28
❖ iReport.....	29
1.1.6. Web Services.....	29
❖ Definición.....	29
❖ XML.....	30
❖ WSDL.....	31
❖ SOAP.....	32
➤ Librería NUSOAP.....	33
1.2. Análisis de Formas de Pagos Electrónicos	35
1.2.1. Protocolos de Seguridad en Transacciones Electrónicas.....	36
1.2.2. Proveedores de Transacciones Electrónicas.....	37
1.3. Sistema De Control De Versiones (VCS).....	39
1.3.1. Definición.....	40
1.3.2. Sistemas de control de versiones locales.....	40
1.3.3. Sistemas de control de versiones centralizados.....	41
1.3.4. Sistemas de control de versiones distribuidos.....	42
1.3.5. Reseña histórica de Git.....	43
1.3.6. Fundamentos de Git.....	43
1.3.7. Integridad de GIT.....	45
1.3.8. Los Tres Estados de GIT.....	45
1.4. Core GAD-I.....	47
1.5. Metodología de Desarrollo GAD-I.....	48
1.5.1. Definición.....	48
1.5.2. Objetivo.....	48
1.5.3. Alcance.....	49
1.5.4. Definiciones Y Abreviaturas.....	49
1.5.5. Documentos De Referencia.....	50
1.5.6. Políticas.....	50
1.5.7. Diagrama De Flujo.....	53
1.5.8. Descripción del Procedimiento.....	54

2. Capítulo II.....	57
Procesos Sismert	57
2.1. Definición de Proceso.	57
2.1.1. Importancia de los procesos	57
2.2. Procedimiento De Venta De Tickets Y Emisión De Notificaciones.....	58
2.2.1. Objetivo.....	58
2.2.2. Alcance.....	58
2.2.3. Roles y Responsabilidades	59
2.2.4. Términos y Definiciones	59
2.2.5. Diagrama De Flujo (Ver figura 2.1).....	60
2.2.6. Descripción Del Procedimiento.....	62
2.2.7. Documentos De Referencia.....	64
2.3. Procedimiento De Planificar Las Zonas De Estacionamiento Regulado.....	64
2.3.1. Objetivo.....	64
2.3.2. Alcance.....	64
2.3.3. Roles y Responsabilidades	65
2.3.4. Términos Y Definiciones	65
2.3.5. Procedimiento	66
2.3.6. Descripción Del Procedimiento.....	67
2.3.7. Documentos De Referencia.....	68
2.4. Procedimiento De Venta De Tickets Electrónicos A Través De Un Sms.....	68
2.4.1. Objetivo.....	68
2.4.2. Alcance.....	69
2.4.3. Roles y Responsabilidades	69
2.4.4. Términos y Definiciones	70
3. Capítulo III	72
Diseño y Desarrollo de Software.	72
3.1. Roles.	73
3.1.1. Integrantes del Equipo.....	75
3.2. Recepción de Solicitud para Desarrollo de Software	75
3.3. Asignación y Planificación del Proyecto Tecnológico	75
3.4. Elaboración y Corrección del Proyecto Tecnológico	76
3.5. Obtención y Documentación de Requisitos.....	76
3.5.1. Historias de Usuario	77
3.5.2. Pila de Producto	90
3.5.3. Metáfora.	91
3.6. Diseño del Sistema.....	91
3.6.1. Arquitectura del Sistema.	92
3.6.2. Arquitectura Funcional.....	94
3.7. Pilas de Iteración	95
3.7.1. Pila de Iteración I	95

3.7.2.	Pila de Iteración II.....	95
3.7.3.	Pila de Iteración III.....	95
3.7.4.	Pila de Iteración IV	96
3.7.5.	Pila de Iteración V.....	96
3.7.6.	Pila de Iteración VI	96
3.7.7.	Pila de Iteración VII.....	97
3.7.8.	Pila de Iteración VIII.....	97
3.7.9.	Pila de Iteración IX	97
3.7.10.	Pila de Iteración X.....	97
3.8.	Descripción de Iteraciones.....	98
3.8.1.	Iteración I	98
3.8.2.	Iteración II.....	102
3.8.3.	Iteración III.....	104
3.8.4.	Iteración IV	106
3.8.5.	Iteración V.....	109
3.8.6.	Iteración VI	110
3.8.7.	Iteración VII.....	114
3.8.8.	Iteración VIII.....	115
3.8.9.	Iteración IX	119
3.8.10.	Iteración X.....	120
3.9.	Desarrollo de Historias de Usuario por Iteración.	121
3.9.1.	Iteración I (Módulo de Registro y Compras de Saldo).....	121
➤	Historia 2: Repositorio de Datos SISMERT	121
➤	Historia 3: Seguridad y Control de Acceso.....	123
➤	Historia 1: Requerimientos para uso del Parqueadero Sismert	126
➤	Historia 4: Contenido del SMS para uso del Parqueadero	130
➤	Historia 5: Registro de Vehículos.....	131
3.9.2.	Iteración II (Módulo de Registro y Compras de Saldo)	134
➤	Historia 5: Registro de vehículos.....	134
➤	Historia 7: Compras de Saldo, Información Registros	135
3.9.3.	Iteración III (Módulo Inspección Zona Parqueo).....	137
➤	Historia 8: Cambiar Lugar de Estacionamiento.....	137
➤	Historia 9: Notificaciones Inspector y Ciudadano	140
3.9.4.	Iteración IV (Módulo Inspección Zona Parqueo).....	141
➤	Historia 6: Establecer Identificación de Sitios para el Parqueadero	141
3.9.5.	Iteración V (Módulo de Registros y Compras de Saldo).....	145
3.9.6.	Iteración VI (Módulo Control Parqueo vía Aplicación Móvil)	147
➤	Historia 10: Plantear nueva solución de notificación Hacia el inspector	147
➤	Historia 11: Creación Aplicación Móvil y Seguridad.....	148
➤	Historia 12: Consulta Tiempo Parqueo y Notificaciones.....	150
➤	Historia 13: Registro y Compras de Saldo en Aplicación Móvil.....	152
3.9.7.	Iteración VII (Módulo de Registro y Compras de Saldo).....	154
➤	Historia 14: Recaudación de Compras de Saldo SISMERT.	154
3.9.8.	Iteración VIII (Módulo Web Ciudadano).....	155
➤	Historia 15: Servicios en la web para el ciudadano	155
➤	Historia 16: Recargas SISMERT en Línea	159
3.9.9.	Iteración IX (Módulo de Registro y Compras de Saldo).....	160

➤	Historia 7: Compras de Saldo, Información Registros	161
➤	Historia 14: Recaudación de Compras de Saldo SISMERT	162
3.9.10.	Iteración X (Módulo de Registro y Compras de Saldo)	163
➤	Historia 9: Notificaciones Inspector y Ciudadano	164
3.10.	Subir cambios de versiones al repositorio GIT	165
4.	Capítulo IV.....	168
	Conclusiones y Recomendaciones	168
4.1.	Análisis Costo Beneficio.....	168
4.2.	Impactos	170
4.3.	Conclusiones	171
4.4.	Recomendaciones	172
	Glosario de Términos	174
	Bibliografía.....	175
	ANEXOS	179
	Anexo 1: Ordenanza de organización y establecimiento del SISMERT de Ibarra.....	179
	Anexo 2: Acta de reunión para obtención de requerimientos para el desarrollo del sistema.	179
	Anexo 3: Proyecto Tecnológico.....	179
	Anexo 4: Acta de revisión y corrección del Proyecto Tecnológico.....	179
	Anexo 5: Actas de reunión de obtención de requisitos y revisión del sistema.	179
	Anexo 6: Manual de Usuario.....	179
	Anexo 7: Manual Técnico.....	179
	Anexo 8: Artefactos de la Metodología de desarrollo GAD-I.	179
	Anexo 9: Acta de entrega de Proyecto o Módulos del Sistema.....	179

Índice de Tablas

TABLA 1. 1: CARACTERÍSTICAS COMPARATIVAS DE NUSOP Y WSDL-GENERATOR.....	35
TABLA 1. 2: COMPARACIÓN PROTOCOLOS DE SEGURIDAD SSL Y SET	37
TABLA 1. 3: DEFINICIONES Y ABREVIATURAS	49
TABLA 1. 4: DOCUMENTOS INTERNOS	50
TABLA 1. 5: DOCUMENTOS EXTERNOS.....	50
TABLA 1. 6: DESCRIPCIÓN DEL PROCEDIMIENTO.....	56
TABLA 2. 1: ROLES Y RESPONSABILIDADES VENTA TICKETS, EMISIÓN NOTIFICACIONES	59
TABLA 2. 2: TÉRMINOS Y DEFINICIONES VENTA TICKETS, EMISIÓN NOTIFICACIONES	60
TABLA 2. 3: DESCRIPCIÓN DEL PROCEDIMIENTO VENTA TICKETS, EMISIÓN NOTIFICACIONES.....	63
TABLA 2. 4: DOCUMENTOS DE REFERENCIA INTERNOS-EXTERNOS VENTA TICKETS, EMISIÓN NOTIFICACIONES	64
TABLA 2. 5: ROLES Y RESPONSABILIDADES ZONAS PARQUEO	65
TABLA 2. 6: TÉRMINOS Y DEFINICIONES ZONAS PARQUEO	65
TABLA 2. 7: DESCRIPCIÓN DEL PROCEDIMIENTO ZONAS PARQUEO.....	68
TABLA 2. 8: DOCUMENTOS REFERENCIA INTERNOS-EXTERNOS (ZONAS PARQUEO).....	68
TABLA 2. 9: ROLES Y RESPONSABILIDADES VENTA TICKETS ELECTRÓNICOS	69
TABLA 2. 10: TÉRMINOS Y DEFINICIONES TICKETS ELECTRÓNICO.....	70
TABLA 2. 11: GESTIÓN DE ESTACIONAMIENTO TARIFADO.	71
TABLA 3. 1: ROLES DEL SISTEMA.....	74
TABLA 3. 2: INTEGRANTES DEL EQUIPO.	75
TABLA 3. 3: HISTORIA DE USUARIO 1.....	77
TABLA 3. 4: HISTORIA DE USUARIO 2.....	78
TABLA 3. 5: HISTORIA DE USUARIO 3.....	79
TABLA 3. 6: HISTORIA DE USUARIO 4.....	80
TABLA 3. 7: HISTORIA DE USUARIO 5.....	80
TABLA 3. 8: HISTORIA DE USUARIO 6.....	81
TABLA 3. 9: HISTORIA DE USUARIO 7.....	82
TABLA 3. 10: HISTORIA DE USUARIO 8.....	83
TABLA 3. 11: HISTORIA DE USUARIO 9.....	83
TABLA 3. 12: HISTORIA DE USUARIO 10.....	84
TABLA 3. 13: HISTORIA DE USUARIO 11.....	85
TABLA 3. 14: HISTORIA DE USUARIO 12.....	86
TABLA 3. 15: HISTORIA DE USUARIO 13.....	87
TABLA 3. 16: HISTORIA DE USUARIO 14.....	88
TABLA 3. 17: HISTORIA DE USUARIO 15.....	89
TABLA 3. 18: HISTORIA DE USUARIO 16.....	90
TABLA 3. 19: PILA DE PRODUCTO SCOPT. FUENTE: PROPIA.	91
TABLA 3. 20: PILA ITERACIÓN 1	95
TABLA 3. 21: PILA ITERACIÓN 2.....	95
TABLA 3. 22: PILA ITERACIÓN 3.....	96
TABLA 3. 23: PILA ITERACIÓN 4.....	96
TABLA 3. 24: PILA ITERACIÓN 5.....	96

TABLA 3. 25: PILA ITERACIÓN 6.....	96
TABLA 3. 26: PILA ITERACIÓN 7.....	97
TABLA 3. 27: PILA ITERACIÓN 8.....	97
TABLA 3. 28: PILA ITERACIÓN 9.....	97
TABLA 3. 29: PILA ITERACIÓN 10.....	97
TABLA 3. 30: ITERACIÓN 1-HISTORIA DE USUARIO 1.....	100
TABLA 3. 31: ITERACIÓN 1-HISTORIA DE USUARIO 2.....	100
TABLA 3. 32: ITERACIÓN 1-HISTORIA DE USUARIO 3.....	101
TABLA 3. 33: ITERACIÓN 1-HISTORIA DE USUARIO 4.....	101
TABLA 3. 34: ITERACIÓN 1-HISTORIA DE USUARIO 5.....	102
TABLA 3. 35: ITERACIÓN 2-HISTORIA DE USUARIO 5.....	103
TABLA 3. 36: ITERACIÓN 2-HISTORIA DE USUARIO 7.....	104
TABLA 3. 37: ITERACIÓN 3-HISTORIA DE USUARIO 8.....	105
TABLA 3. 38: ITERACIÓN 3-HISTORIA DE USUARIO 9.....	106
TABLA 3. 39: ITERACIÓN 4-HISTORIA DE USUARIO 6.....	109
TABLA 3. 40: ITERACIÓN 5-HISTORIA DE USUARIO 5.....	110
TABLA 3. 41: ITERACIÓN 6-HISTORIA DE USUARIO 10.....	110
TABLA 3. 42: ITERACIÓN 6-HISTORIA DE USUARIO 11.....	111
TABLA 3. 43: ITERACIÓN 6-HISTORIA DE USUARIO 12.....	112
TABLA 3. 44: ITERACIÓN 6-HISTORIA DE USUARIO 13.....	114
TABLA 3. 45: ITERACIÓN 7-HISTORIA DE USUARIO 14.....	115
TABLA 3. 46: ITERACIÓN 8-HISTORIA DE USUARIO 15.....	118
TABLA 3. 47: ITERACIÓN 8-HISTORIA DE USUARIO 16.....	118
TABLA 3. 48: ITERACIÓN 9-HISTORIA DE USUARIO 7.....	119
TABLA 3. 49: ITERACIÓN 9-HISTORIA DE USUARIO 14.....	119
TABLA 3. 50: ITERACIÓN 10-HISTORIA DE USUARIO 9.....	120
TABLA 3. 51: MODELO RELACIONAL DE CONTROL DE ACCESO AL SISTEMA	124
TABLA 3. 52: ESTRUCTURA DEL CONTROLADOR DE UN ITEM MENU (EXTJS)	128
TABLA 3. 53: CONTENIDO SMS PARA INICIAR SERVICIO DE ESTACIONAMIENTO	131
TABLA 3. 54: MODELO Y STORE PARA REGISTRO DE PLACAS VEHICULARES (EXTJS).	132
TABLA 3. 55: CONTENIDO SMS AL CAMBIAR EL SITIO DE PARQUEO.	138
TABLA 3. 56: CONTENIDO SMS NOTIFICACIÓN CIUDADANO E INSPECTOR.....	138
TABLA 3. 57: CÓDIGO DE IDENTIFICACIÓN DE UN SITIO DE ESTACIONAMIENTO.....	142
TABLA 3. 58: AGRUPACIÓN DE SITIOS DE INSPECCIÓN (I) EN ZONAS.	143
TABLA 3. 59: MODELO RELACIONAL AGRUPACION SITIOS (I) POR ZONAS.....	143
TABLA 3. 60: SERVICIOS SMS INTERPRETADOS POR EL SISTEMA.....	164

Índice de Figuras

FIGURA 1: ORGANIGRAMA ESTRUCTURAL POR PROCESOS DEL GAD-I.....	5
FIGURA 1. 1: ARQUITECTURA FUNCIONAL Y HERRAMIENTAS DE DESARROLLO.....	10
FIGURA 1. 2: ARQUITECTURA FUNCIONAL POSTGRESQL.....	12
FIGURA 1. 3: ARQUITECTURA FUNCIONAL SERVIDOR WEB APACHE.....	14
FIGURA 1. 4: ARQUITECTURA FUNCIONAL APACHE TOMCAT.....	15
FIGURA 1. 5: ARQUITECTURA FUNCIONAL CSS.....	16
FIGURA 1. 6: ARQUITECTURA FUNCIONAL PHP.....	17
FIGURA 1. 7: ARQUITECTURA FUNCIONAL JAVASCRIPT.....	18
FIGURA 1. 8: ARQUITECTURA FUNCIONAL SYMFONY.....	20
FIGURA 1. 9: ARQUITECTURA FUNCIONAL DOCTRINE.....	21
FIGURA 1. 10: ARQUITECTURA MVC DE UNA APLICACIÓN EXT JS.....	23
FIGURA 1. 11: ESTRUCTURA DE DIRECTORIOS DE UNA APLICACIÓN MVC.....	24
FIGURA 1. 12: ESTÁNDARES WEB SENCH TOUCH.....	25
FIGURA 1. 13: CARACTERÍSTICAS SENCH TOUCH DISPOSITIVO TÁCTIL Y MÓVIL.....	26
FIGURA 1. 14: ARQUITECTURA MVC SENCH TOUCH.....	27
FIGURA 1. 15: EMPAQUETADO NATIVO DE UNA APLICACIÓN TOUCH.....	27
FIGURA 1. 16: FUNCIONALIDAD DE JASPERREPORTS.....	28
FIGURA 1. 17: FUNCIONALIDAD DE IREPORT PARA GENERAR UN REPORTE.....	29
FIGURA 1. 18: ESTRUCTURA DE SOAP.....	33
FIGURA 1. 19: ARQUITECTURA FUNCIONAL INTERDIN.....	38
FIGURA 1. 20: ARQUITECTURA FUNCIONAL DE PAYPAL.....	39
FIGURA 1. 21: DIAGRAMA DE CONTROL DE VERSIONES LOCAL.....	40
FIGURA 1. 22: DIAGRAMA DE CONTROL DE VERSIONES CENTRALIZADO.....	41
FIGURA 1. 23: DIAGRAMA DE CONTROL DE VERSIONES DISTRIBUIDO.....	42
FIGURA 1. 24: SISTEMA QUE TIENDEN A ALMACENAR DATOS DE CAMBIOS EN CADA ARCHIVO.....	44
FIGURA 1. 25: INFORMACIÓN DE GIT COMO INSTANTÁNEAS DEL PROYECTO A LO LARGO DEL TIEMPO.....	44
FIGURA 1. 26: DIRECTORIO DE TRABAJO, ÁREA DE PREPARACIÓN, Y DIRECTORIO DE GIT.....	46
FIGURA 1. 27: ESTRUCTURA CORE GAD-I.....	47
FIGURA 1. 28: DIAGRAMA DE FLUJO DESARROLLO DE SOFTWARE.....	53
FIGURA 2. 1: VENTA DE TICKETS Y EMISIÓN NOTIFICACIONES.....	61
FIGURA 2. 2: DESCRIPCIÓN PROCEDIMIENTO ZONAS PARQUEO.....	66
FIGURA 3. 1: ARQUITECTURA DEL SISTEMA.....	93
FIGURA 3. 2: ARQUITECTURA FUNCIONAL DEL SISTEMA.....	94
FIGURA 3. 3: MODELO RELACIÓN REGISTROS Y CONTENIDO SMS PARA PARQUEO.....	122
FIGURA 3. 4: INTEGRACIÓN DE ESTRUCTURA DE DATOS EN ESQUEMA SISMERT.....	122
FIGURA 3. 5: ESQUEMA DE DATOS DE ADMINISTRACIÓN DE SISTEMAS GAD-I.....	123
FIGURA 3. 6: MENUS PARA INTERFAZ DE CONTROL DE LA APLICACIÓN.....	124
FIGURA 3. 7: ESTRUCTURA DEL MÓDULO DE ACCESO SFGUARDAUTH DEL CORE (GAD-I).....	125
FIGURA 3. 8: INTERFAZ DE ACCESO A LA APLICACIÓN INTEGRAL (CORE).....	125
FIGURA 3. 9: INTERFAZ DE ESCRITORIO DE LAS APLICACIONES INTEGRADAS EN EL CORE.....	126
FIGURA 3. 10: ESQUEMA INICIAL AL CREAR LA APLICACIÓN EN EL CORE.....	127
FIGURA 3. 11: ESTRUCTURA MVC DE LA APLICACIÓN SISMERT EN EXTJS.....	127

FIGURA 3. 12: ACCIÓN DE UN ÍTEM-MENU DESDE EL CONTROLADOR EXTJS.....	128
FIGURA 3. 13: INTERFAZ DE REGISTRO DE TELÉFONOS SISMERT.....	129
FIGURA 3. 14: VENTANA PARA BÚSQUEDA Y EDICIÓN DE UN REGISTRO (TELÉFONO)	129
FIGURA 3. 15: INTERFAZ DE EDICIÓN DE UN REGISTRO DE TELÉFONO SISMERT.....	129
FIGURA 3. 16: MODELO RELACIONAL CONTENIDO SMS - INICIAR SERVICIO ESTACIONAMIENTO.	131
FIGURA 3. 17: MODELO RELACIONAL DE REGISTRO DE VEHÍCULOS	132
FIGURA 3. 18: INTERFAZ DE REGISTRO DE PLACAS VEHICULARES.	133
FIGURA 3. 19: BÚSQUEDA DE UN TELÉFONO REGISTRADO.....	133
FIGURA 3. 20: VENTANA PRINCIPAL DE REGISTRO DE VEHÍCULOS.....	134
FIGURA 3. 21: VENTANA DE ADMINISTRACIÓN DE VEHÍCULOS REGISTRADOS EN UN TELÉFONO.	135
FIGURA 3. 22: INTERFAZ PARA RECARGAS DE SALDO SISMERT.....	135
FIGURA 3. 23: VENTANA FUNCIONAL DE COMPRAS DE SALDO SISMERT.....	136
FIGURA 3. 24: VENTANA CONSULTAS DE SALDO SISMERT	136
FIGURA 3. 25: VENTANA DE INFORMACIÓN Y BÚSQUEDA DE REGISTROS.	137
FIGURA 3. 26: INTERFAZ ADMINISTRACIÓN DE TELÉFONOS INSPECTORES.....	139
FIGURA 3. 27: VENTANA DE ASIGNACIÓN Y EDICIÓN DE UN NÚMERO DE TELÉFONO AL INSPECTOR.	139
FIGURA 3. 28: FORMULARIO DE ASIGNACIÓN Y EDICIÓN DE UN SITIO A UN INSPECTOR.....	140
FIGURA 3. 29: VENTANA FUNCIONAL DE ASIGNACIÓN Y EDICIÓN DE UN SITIO A UN INSPECTOR.	141
FIGURA 3. 30: MODELO RELACIONAL CÓDIGO SITIOS PARQUEO.	142
FIGURA 3. 31: FORMULARIO INICIA DE INGRESO Y EDICIÓN DE SITIOS DE PARQUEO.....	144
FIGURA 3. 32: VENTANA FUNCIONAL PARA INGRESO Y EDICIÓN DE SITIOS DE PARQUEO.....	144
FIGURA 3. 33: VENTANA DE ASIGNACIÓN DE SITIOS DE INSPECCIÓN (I) EN UNA ZONA.....	145
FIGURA 3. 34: VENTANA PARA EDICIÓN DE SITIOS (I) ASIGNADOS EN UNA ZONA.....	145
FIGURA 3. 35: MODELO DE DATOS DE UN VEHÍCULO A REGISTRARSE.	146
FIGURA 3. 36: VENTANA DE REVISIÓN E INGRESO DE VEHÍCULOS A SER REGISTRADOS.	147
FIGURA 3. 37: COMPONENTES DE UN FORMULARIO EN SENCHÁ TOUCH.....	148
FIGURA 3. 38: AUTENTIFICACIÓN PARA ACCESO A LA APLICACIÓN MÓVIL.	149
FIGURA 3. 39: VENTANA DE LISTA DE APLICACIONES EN LA APLICACIÓN MÓVIL	149
FIGURA 3. 40: ESTRUCTURA MVC DE LA APLICACIÓN MÓVIL SISMERT.....	150
FIGURA 3. 41: FORMULARIO DE CONSULTA DE DISPONIBILIDAD DE ESTACIONAMIENTO.	151
FIGURA 3. 42: FORMULARIO DE INGRESO DE NOTIFICACIONES.	151
FIGURA 3. 43: INTERFAZ FUNCIONAL CONSULTANDO DISPONIBILIDAD DE TIEMPO.	151
FIGURA 3. 44: FORMULARIO REGISTRO DE TELÉFONOS EN APLICACIÓN MÓVIL.....	152
FIGURA 3. 45: FORMULARIO DE REGISTRO DE VEHÍCULOS EN APLICACIÓN MÓVIL.	152
FIGURA 3. 46: FORMULARIO DE COMPRAS DE SALDO EN APLICACIÓN MÓVIL.	153
FIGURA 3. 47: FORMULARIO PARA INICIAR SERVICIO DE ESTACIONAMIENTO EN APLICACIÓN MÓVIL.....	153
FIGURA 3. 48: VENTANA DE INFORMACIÓN Y BÚSQUEDA DE RECAUDACIONES	155
FIGURA 3. 49: VENTANA DE RECAUDACIÓN DIARIA POR INSPECTOR.	155
FIGURA 3. 50: MODELO RELACIONAL ACCESO WEB CIUDADANO.....	156
FIGURA 3. 51: FORMULARIO DE ACCESO A LA APLICACIÓN WEB DEL CIUDADANO.	157
FIGURA 3. 52: VENTANA DE REGISTRO DE USUARIO CIUDADANO.....	157
FIGURA 3. 53: VENTANA RESTAURAR USUARIO CIUDADANO.	158
FIGURA 3. 54: VENTANA PRINCIPAL DE LA APLICACIÓN WEB CIUDADANO.....	158
FIGURA 3. 55: VENTANA PARA CONSULTA DE USO DEL PARQUEADERO.....	159
FIGURA 3. 56.: VENTANA DEL REPORTE EN PDF DE USO DEL PARQUEADERO	159
FIGURA 3. 57: PROTOTIPO DE COMPRAS DE SALDO EN LÍNEA.....	160
FIGURA 3. 58: ESTRUCTURA DE LA CLASE GENÉRICA IMiREPORT PARA GENERAR REPORTES.	161
FIGURA 3. 59: REPORTE ESTADÍSTICO DE USO DEL PARQUEADERO POR AÑO.....	161

FIGURA 3. 60: REPORTE DE REGISTROS DE TELÉFONOS SISMERT.....	162
FIGURA 3. 61: REPORTE RECAUDACIONES A INSPECTORES.	163
FIGURA 3. 62: VENTANA DE BÚSQUEDA DE RECAUDACIONES QUE GENERA REPORTES.....	163
FIGURA 3. 63: ESTRUCTURA FUNCIONAL DE ENVÍO Y RECEPCIÓN DE SMS.....	165
FIGURA 3. 64: COMANDO COMMIT PARA VERSIONAR EL PROYECTO.....	166
FIGURA 3. 65: COMANDO FETCH PARA OBTENER DATOS DEL REPOSITORIO REMOTO DEL GIT.....	166
FIGURA 3. 66: COMANDO MERGE PARA ENVIAR DATOS AL REPOSITORIO REMOTO DEL GIT.	167
FIGURA 3. 67: COMANDO PUSH PARA INGRESAR CAMBIOS EN EL REPOSITORIO DEL GIT.	167
FIGURA 4. 1: TIEMPO DE CONTROL DEL PARQUEADERO.....	170

Resumen

En el presente documento se detalla el proyecto que ha sido desarrollado para crear una nueva solución de uso del estacionamiento de las zonas reguladas del SISMERT mejorando el control del parqueo Tarifado de la ciudad de Ibarra.

La elaboración del Sistema de control de Parqueo Tarifado consta de cuatro capítulos en los que se detalla los procesos que se han realizado para la elaboración del sistema.

En el Capítulo uno se describe las herramientas de software que se utiliza para desarrollar la aplicación.

En el Capítulo dos se describe los procedimientos que se debe seguir para realizar el control del parqueadero tarifado.

A continuación en el Capítulo tres se desarrolla el sistema siguiendo la metodología GAD-I siendo una metodología de rápido desarrollo de proyectos tecnológicos.

Finalmente en el Capítulo cuatro se detalla las conclusiones y recomendaciones alcanzadas luego de haber completado el desarrollo del sistema.

Palabras Claves: Envío SMS, Aplicación Móvil, Servicios Web, SISMERT, Control de Parqueo Tarifado, ExtJS, Sencha Touch.

Summary

In this present document detailed the Project has been developed to create a new solution use of parking areas regulated SISMERT improving parking control tariffed of Ibarra`s city.

The elaboration of System parking control tariffed consists four chapters as detailed the processes that realized for system elaboration.

In Chapter one describes the software tools used to develop the application.

In Chapter two describes the procedures you must follow to perform the parking control tariffed.

Then In Chapter three developed the system following the GADI methodology being a methodology for rapid development of technology projects.

Finally in chapter four are detailing the findings and recommendations that have been reached after you have complete the development of the system.

Keywords: sending SMS, mobile application, web sevice, SISMERT, parking control tariffed, ExtJS, Sencha Touch.

Introducción.

En el presente capítulo se describe los antecedentes, Base Legal de la Institución y el Área en la que se desarrolla el Sistema de Control de Parqueo Tarifado. También se describe el problema que actualmente existe, la Solución que se pretende dar mediante los objetivos y alcance del Sistema.

1. Antecedentes.

A continuación se presenta los antecedentes de la Institución y Área Auspiciante para el Desarrollo del Sistema, la misma que va a servir para comprender el enfoque y el entorno donde se implantara el proyecto que se lo va a Desarrollar en la Dirección de Tecnologías y comunicación (TIC¹).

2. GAD-I

El Gobierno Autónomo Descentralizado de Ibarra(GAD-I²) es una Institución pública que tiene como misión “Planificar, regular, ejecutar y promover el desarrollo integral sostenible del cantón de Ibarra, a través de servicios de calidad eficientes y transparentes con la participación activa de la ciudadanía socialmente a fin de lograr un buen vivir”.

El GAD-I en su afán de mejorar la calidad de vida de los ciudadanos y administración de cada uno de sus procesos dentro y fuera de la institución, hace uso de los recursos de las nuevas tecnologías a través de gobierno electrónico brindando a los ciudadanos un conjunto de servicios inteligentes que mejoran la calidad de vida y aportan al desarrollo social, económico y cultural. **Fuente: GAD-I.**

3. SISMERT

El GAD-I a través de una Ordenanza crea el "Sistema Municipal de Estacionamiento Rotativo Tarifado" de la ciudad de Ibarra "SISMERT³", con el objetivo de apoyar el desarrollo territorial, la

¹TIC: Tecnologías de la información y Comunicaciones.

²GAD-I: Gobierno Autónomo Descentralizado de Ibarra.

³ SISMERT: Sistema Municipal de Estacionamiento Rotativo Tarifado

conectividad y movilidad dentro del área urbana de la ciudad, así como mejorar las condiciones de circulación peatonal, vehicular y ciclística.

El Sistema de Estacionamiento Rotativo brinda el servicio del parqueadero a la ciudadanía acogiendo a medios de transporte como:

- Vehículos Pesados
- Vehículos Livianos
- Motos

La tarifa que cada chofer debe cancelar es según el tiempo que hace uso del parqueadero, y su duración máxima es de dos horas; el servicio está disponible de Lunes a Sábado a partir de las 8 de la mañana hasta las 7 de la noche, excepto los días sábados ya que su horario de atención es de 8 de la mañana hasta las 2 de la tarde. Es así como la Municipalidad ha tratado de regularizar el tráfico vehicular de la ciudad de Ibarra.

La media hora de parqueadero es de 25 centavos y la hora 40 centavos, “lo más importante es decirle a la ciudadanía que si utiliza la tarjeta de parqueo no será multada, pero si no hay cumplimiento la multa por primera vez cuesta 3 dólares y si es una reincidencia costará hasta 10 dólares”.

Cabe señalar que anteriormente este servicio había sido controlado por la empresa privada y tuvo que ser suspendido por que su administración no estuvo acorde a los requerimientos que el ciudadano requería, dentro del sistema aun no se lograba regularizar cada uno de sus procesos de funcionalidad y control del mismo. Después de ser regularizados el GAD-I decidió implantarlo nuevamente y así ofrecer un servicio que ayude al descongestionamiento vehicular en el centro de la ciudad de Ibarra.

Fuente: Recolección de datos en la web y SISMERT.

4. Base Legal de la Institución.

Datos tomados de: **Reglamento Orgánico Funcional 2010** del GAD-I (Filosofía Organizacional).

❖ Misión.

El GAD-I planifica, regula, ejecuta y promueve el desarrollo integral sostenible del cantón, a través de servicios de calidad eficientes y transparentes con la participación activa de la ciudadanía socialmente responsable a fin de lograr el buen vivir.

❖ Visión.

Seremos un municipio líder en gestión con responsabilidad social, que garantice equidad, honestidad, trabajo y eficiencia por qué Ibarra se constituya en un cantón próspero, atractivo e incluyente, capital de los servicios y el conocimiento, referente del buen vivir en la región norte del Ecuador.

❖ **Objetivos Generales.**

El Plan Estratégico del GAD-I 2010-2014, por una Ibarra hacia el futuro, segura, productiva y social, determina cuatro temas estratégicos en base al análisis FODA⁴, para identificar los cinco objetivos generales.

1. **Identidad, Participación y Ciudadanía:** Construcción de nueva ciudadanía, multiétnica y pluricultural, con participación ciudadana, incrementando la seguridad, la protección del medio ambiente, la esperanza y la calidad de vida de los ciudadanos.
2. **Recuperación Económica, Desarrollo y Competitividad:** Impulsar el crecimiento de la economía, a partir del mejoramiento de la competitividad y productividad, facilitando la instalación de actividades productivas.
3. **Desarrollo Territorial:** Orientar el desarrollo físico y ambiental del municipio de forma que permita elevar la calidad de vida de sus habitantes, consolidar y desarrollar el espacio público de manera equitativa y mejorar la imagen urbana.
4. **Desarrollo Institucional:** Garantizar una administración municipal con responsabilidad social, bajo los principios de eficiencia, eficacia y transparencia, con procesos y procedimientos estandarizados.

❖ **Objetivos Específicos.**

1. Construir una ciudad incluyente, solidaria, con la participación de los ibarreños en todos los aspectos de la vida de la ciudad. Auspiciar la igualdad, la cohesión y la integración social y territorial, con respeto a los ciudadanos con capacidades especiales y de la tercera edad.
2. Aumentar la esperanza y la calidad de vida de la población, a través de un medio ambiente sano y sustentable, con acceso equitativo y seguro al agua, aire y suelo.
3. Promover los mecanismos necesarios para que Ibarra sea una ciudad segura, con sistemas adecuados de protección de la vida y bienes de las ciudadanas y los ciudadanos, prevención de riesgos y protección en caso de desastres, en base a mecanismos que incluyan la participación social.
4. Garantizar en coordinación con el Estado el acceso libre a una educación de calidad, en un ambiente seguro y confortable, con respeto a las culturas, las tradiciones y las etnias, que integre contenidos locales, de manera que se garanticen procesos identitarios a largo plazo, así como vincular la educación a los objetivos de desarrollo del cantón.

⁴ FODA: Fortalezas, Oportunidades, Amenazas, Debilidades

5. Consolidar estrategias que integren el deporte, el ejercicio y la recreación como partes fundamentales del mejoramiento de calidad de vida de los habitantes del cantón, fortaleciendo el uso de los espacios públicos y de encuentro común.
6. Promover el desarrollo de las culturas karanki, Imbayacuna, Cayambi, Natabuela, Awá, Afrochoteña y mestiza, así como de los aportes de los migrantes, a través de mecanismos de recuperación de memoria colectiva, difusión de los artes y saberes tradicionales, educación en contenidos propios de cada cultura, preservación de valores y potenciación de su desarrollo futuro, con respeto, equidad e integración.
7. Promover las artes y la creatividad, en un proceso que contemple la recuperación de memoria y la difusión cultural como mecanismos coadyuvantes a la creación de patrimonios vivos de la cultura de Ibarra y de nuevos imaginarios urbanos para la creación de una nueva ciudadanía en la que la cultura sea parte esencial del mejoramiento de la calidad de vida en el cantón.
8. Proyectar a la comunicación como herramienta transversal en los procesos participativos, sociales, de salud y medio ambiente, de cultura, educación, deportes y recreación, utilizando para el efecto todas las tecnologías para mantener informados a los ciudadanos.

5. Organigrama Estructural por Procesos.

Los datos presentados se han tomado de: **Resolución Administrativa Sobre Gestión Organizacional Por Procesos--Definitivo IMI 2010** (Capítulo II).

La estructura Administrativa de la Ilustre Municipalidad de Ibarra responde a las necesidades peculiares que debe satisfacer, la importancia de los servicios públicos a prestarse y con la finalidad de ejercer todas y cada una de las funciones que a ésta le competen para el mejor cumplimiento de sus fines, con apego a lo establecido en el Código Orgánico de Ordenamiento Territorial, Autonomía y Descentralización COOTAD.

La estructura organizacional se encuentra definida en áreas administrativas, teniendo en cuenta que las distintas dependencias constituyan un organismo racionalmente integrado, en función de la división del trabajo, y la gestión para ofertar una gama de productos a la colectividad; entendiéndose por productos los bienes y/o servicio similares o interrelacionados destinados a satisfacer la demanda de usuarios o clientes externos e internos.

Los clientes externos son los beneficiarios de los productos generados en el sistema de desarrollo económico cantonal; los clientes internos son los favorecidos con productos elaborados por los procesos institucionales. **Ver Figura 1.**

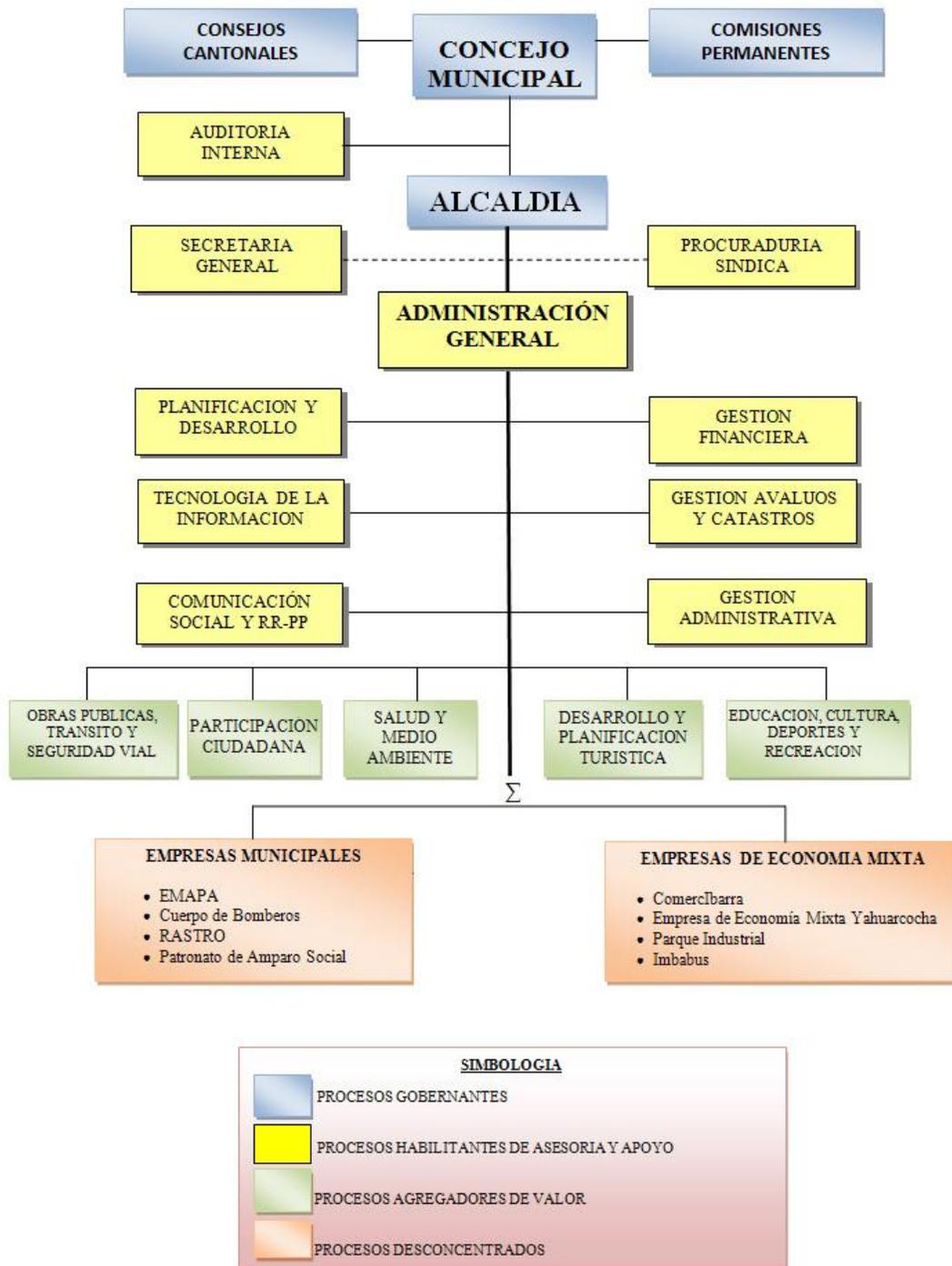


Figura 1: Organigrama estructural por procesos del GAD-I

Fuente: Resolución administrativa – Gestión Organizacional por Procesos - IMI 2010

6. Área SISMERT.

El Sistema Municipal de Estacionamiento Tarifado controla y regula el tráfico vehicular de las principales calles del centro de la ciudad de Ibarra, especialmente las calles céntricas en donde se parquean a doble fila.

Desde abril del 2010 el GAD-I asumió el control del parqueo tarifado en la ciudad, luego de la ampliación de las zonas de parqueo la recaudación subió de un 45 a 50%. Los lugares más conflictivos del parqueo tarifado, son las zonas aledañas al mercado Amazonas, la calle Simón Bolívar y la calle Olmedo, porque en estas vías hay mayor afluencia de vehículos que se estacionan a los costados lo que genera un inconveniente en el tráfico.

Los tipos de contravenciones para la ciudadanía al hacer mal uso de las plazas de estacionamiento, las mismas que constan en la ordenanza municipal son las siguientes:

- a) La permanencia de un vehículo en una plaza de estacionamiento dentro de la zona regulada del SISMERT, sin el documento que habilite la ocupación de la plaza regulada.
- b) La permanencia de un vehículo en la zona regulada del SISMERT, con el documento habilitante de ocupación de la vía pública alterado o que no haya sido emitido por el administrador del sistema.
- c) Colocación de forma incorrecta o poco visible, del documento habilitante para uso del SISMERT.
- d) La permanencia de un vehículo en la zona regulada, luego de haber transcurrido el tiempo acreditado por el documento habilitante del uso del SISMERT.
- e) La permanencia de un vehículo común en una zona regulada, luego de haber transcurrido el tiempo máximo permitido de 3 horas continuas a pesar de cubrir el tiempo excedido con un documento habilitante de uso del SISMERT.
- f) Uso indebido de las plazas de estacionamiento constitutivas del SISMERT.
- g) Tarjeta en Blanco.
- h) Error en marcación de la hora.

El uso de las tarjetas, para que las personas no sean multadas deben comprar con anticipación la tarjeta de parqueo en los 56 puntos de venta de Sismert o a los mismos inspectores.

La media hora de parqueo cuesta 25 centavos y una hora 40 centavos, “lo más importante es decirle a la ciudadanía que si utiliza la tarjeta de parqueo no será multada, pero si no hay cumplimiento la multa

por primera vez cuesta 5 dólares y si es una reincidencia costará hasta 10 dólares según la última reforma de la ordenanza Municipal". **Fuente: SISMERT del GAD-I y ordenanza ANEXO 1.**

7. Problema.

En la actualidad el sistema de parqueo tarifado SISMERT se encuentra en funcionamiento, y en un proceso de renovación tecnológica. El servicio de parqueo está ubicado en distintos puntos de la ciudad; estos puntos de parqueo pueden ser utilizados por los ciudadanos y demás personas de fuera de la ciudad. Actualmente existen alrededor de 1100 plazas de estacionamiento.

El parqueo tarifado actualmente se encuentra controlado por un grupo de personas (Inspectores) ubicadas en ciertos puntos donde se encuentran los sitios de parqueo, el personal tiene que movilizarse de un lugar a otro para registrar el uso del parqueadero por parte de cada usuario.

De acuerdo a este análisis se puede constatar que el desempeño no es muy eficiente, provocando mucha inconformidad a los usuarios, la misma que se origina cuando el usuario quiere hacer uso del servicio y no puede porque los puntos de parqueo no abastecen a toda la demanda de usuarios, otra inconformidad se debe a la falta del Inspectores encargados del registro de parqueo y en muchas ocasiones el usuario no puede hacer uso del servicio por estos inconvenientes señalados.

El sistema de parqueo no puede ser procesado con total normalidad debido a la falta de cultura proveniente de los ciudadanos porque en muchas ocasiones tratan de evitar pagar el costo del servicio cuando el personal encargado del SISMERT no se encuentra presente, y en muchas ocasiones se produce inconvenientes entre usuario e Inspector.

Al final de la jornada el personal entrega el registro de parqueo, pero este proceso no es tan transparente, por que los Inspectores en muchas ocasiones no llevan la información real de cada uno de los registros que se obtuvieron durante el día, esto se produce porque no se puede realizar un seguimiento a cada uno de los inspectores y constatar los registros que toma cada Inspector.

Estas situaciones han hecho que los procesos que se realizan dentro del Sistema de parqueo tarifado no sean tan eficientes y este servicio tenga un mal uso.

8. Objetivos.

❖ Objetivo General.

Desarrollar un sistema Informático para el parqueo tarifado(SISMERT) del Gobierno Autónomo de Ibarra que ayude a un mejor control de cobro del mismo, basado en el uso de herramientas de software libre, que agilizaran el proceso de este servicio, evitar pérdidas económicas y brindar un mejor servicio para su cobro.

❖ Objetivos Específicos.

1. Analizar las formas o maneras de pago Electrónico más seguras para el cliente.
2. Seleccionar y estudiar las herramientas para el proceso de transacciones electrónicas.
3. Analizar el modelo de envío de datos por parte del usuario a través de un dispositivo móvil.
4. Realizar la interpretación de los datos receptados del usuario para su posterior proceso de reenvío de confirmaciones.
5. Diseñar e implementar un modelo de factura para el usuario del parqueo tarifado que podrá acceder al final de cada mes.
6. Definir e implantar la arquitectura y funcionamiento de la solución planteada.
7. Capacitar a los usuarios finales del sistema de Parqueo tarifado.
8. Elaboración de la documentación para el buen uso y desenvolvimiento del sistema.

9. Justificación.

En la actualidad es importante hacer uso de las herramientas informáticas que ayuden a regularizar y automatizar los distintos procesos que se llevan a cabo dentro de las empresas o instituciones públicas y privadas, por tal razón se puede observar que hoy en día la ciudadanía de Ibarra que se moviliza a través de vehículos livianos, motos, y los propios vehículos de carga, hacen uso con frecuencia del sistema actual de parqueo tarifado del GAD-I, por lo que se puede determinar que necesita de un sistema informático que regule y controle el proceso de parqueo y sea esta una forma adicional del uso del estacionamiento, que en muchas ocasiones provoca congestiones vehiculares en las principales calles de la ciudad y disgusto por parte de cada usuario porque el servicio no es abastecido por los pocos inspectores que mantienen el control del sistema de parqueo tarifado.

Es así como las personas que utilizan vehículos como medio de transporte se beneficiaran al tener una opción más al momento de hacer uso del servicio, a través del uso de sus dispositivos móviles que permitan el envío de mensajes SMS para el uso del sistema de parqueo tarifado de la ciudad de Ibarra. Además el sistema de parqueo tarifado no cuenta con un portal web que facilite al usuario realizar recargas o compras electrónicas, para el posterior uso de este servicio, así como también la información de los lugares o puntos de ubicación de los sitios de estacionamiento, y el costo referencial dependiendo de el tiempo máximo de uso o una tasa mínima que el usuario debe pagar por el servicio. También es muy importante considerar que los usuarios tengan la posibilidad de ver todas las transacciones realizadas o solicitar la factura del uso de este servicio. Precisamente la obtención de la factura al final del mes sobre los gastos de todos los móviles es un servicio interesante sobre todo para las empresas y autónomos, ya que no tienen que acumular y guardar los resguardos para contabilizar los gastos de estacionamiento.

Para el Gobierno Autónomo Descentralizado de Ibarra es de gran importancia la implantación de este sistema, debido a que mejorara el control del sistema actual de parqueo tarifado, y el poco personal que labora para el control del mismo, tendrá la posibilidad de una mejor movilización hacia los distintos puntos de estacionamiento vehicular, teniendo un control sistematizado que ayude a evitar pérdidas económicas dentro de la institución, brindando así un mejor servicio de calidad para la ciudadanía.

10. Alcance.

El Sistema de control de parqueo tarifado tiene como funcionalidad la realización de cobros de la tarifa del parqueo que un usuario debe pagar cuando hace uso del mismo.

Este proceso se lo realizara a través de un dispositivo móvil vía SMS como también a través de un entorno web. El sistema será capaz de recibir los datos que un usuario envía desde el entorno web para realizar recargas o compras electrónicas, y vía SMS para realizar la compra del ticket electrónico que realiza el pago de la tarifa de parqueo.

También se encargara de realizar un proceso automático que consiste en el reenvío de notificaciones tanto a los usuarios como a los inspectores que realizan el control en los diferentes puntos de estacionamiento que existen.

1. Capítulo I

Marco Teórico

A continuación se describen algunos conceptos generales de cada una de las herramientas que luego serán necesarios conocerlos para el desarrollo del Aplicativo.

1.1. Herramientas de Desarrollo

Las herramientas de desarrollo son aquellos programas o frameworks que tengan cierta importancia en la elaboración de una aplicación, ayudando en el desempeño de su realización, tanto en la parte visual como en la funcional, para el buen desempeño de una aplicación.

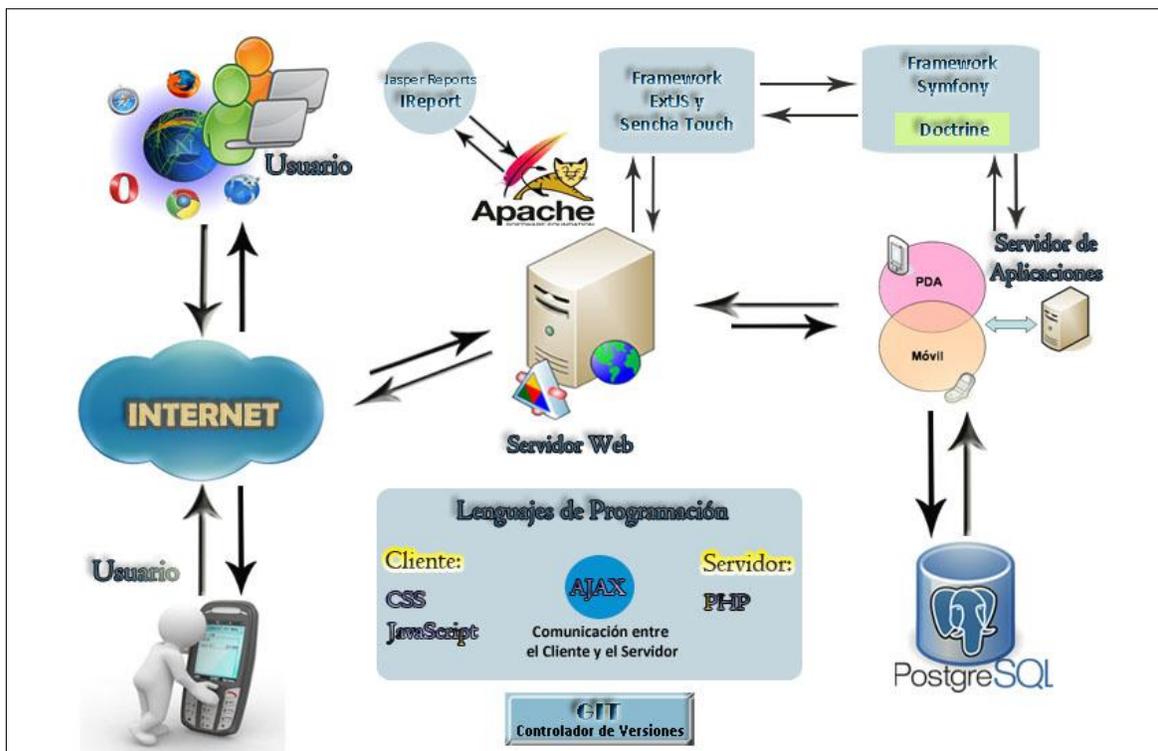


Figura 1. 1: Arquitectura Funcional y Herramientas de Desarrollo.
Fuente: Propia

En el gráfico anterior se presenta la funcionalidad que el sistema tendrá y las herramientas que intervienen en su arquitectura funcional.

El Sistema de Control de Parqueo Tarifado es una aplicación web que va a estar contenida por el servidor web Apache, como servidor de aplicaciones Apache-Tomcat para generar reportes desde IReport, el repositorio de datos alfanuméricos PostgreSQL. Para la creación de la interfaz de usuario se utiliza el framework de desarrollo web Ext JS y Sencha Touch, Framework Symfony que se encarga de interactuar con el servidor y su herramienta Doctrine que sirve para mapear los objetos relacionales de la Base de Datos. La versión de la aplicación es controlada por el repositorio de GIT.

1.1.1. Base de Datos

Las bases de datos son el método preferido para el almacenamiento estructurado de datos. Desde las grandes aplicaciones multiusuario, hasta los teléfonos móviles y las agendas electrónicas utilizan tecnología de bases de datos para asegurar la integridad de los datos y facilitar la labor tanto de usuarios como de los programadores que las desarrollaron. **[LIB1]**

❖ PostgreSQL⁵.

Es un sistema de gestión de base de datos objeto-relacional, distribuido bajo la licencia BSD⁶. Es el sistema de gestión de base de datos de código abierto más potente del mercado. PostgreSQL utiliza un modelo cliente servidor y usa multiprocesos para garantizar la estabilidad del sistema, un fallo en uno de los procesos no afectará el resto y el sistema continuara funcionando, utiliza el lenguaje SQL⁷ para consultas.

PostgreSQL fue el pionero en muchos de los conceptos existentes en el sistema objeto-relacional actual. Es un sistema objeto-relacional, ya que incluye características de orientación a objetos, como herencia, tipos de datos, funciones, restricciones, disparadores, reglas e integridad transaccional. PostgreSQL se caracteriza por ser un sistema estable, de alto rendimiento, gran flexibilidad ya que funciona en la mayoría de los sistemas operativos. **[WEB1]**

⁵ PostgreSQL: Gestor de Base de Datos.

⁶BSD: Distribución de Software Berkeley, es un sistema operativo derivado del sistema Unix.

⁷ SQL: lenguaje de consultas estructurado.

Características:

- Garantiza el concepto de transacción (ACID⁸).
- La implementación de SQL se adhiere al estándar ANSI-SQL 92/99.
- Tiene soporte completo para subqueries (incluye subselects en la cláusula FROM).
- Soporte para Replicación.
- Triggers, procedual lenguajes (PL/PgSQL,PL/R).
- Incluye soporte para el manejo de integridad de datos. Primary key, foreign key con acciones referenciales de cascada por update/delete, check constraints, unique constraint, not null constraint.
- Permite retornar resultados parciales de consultas con LIMIT/OFFSET.
- Proporciona acceso multi-usuario para potencialmente enormes cantidades de datos.
- Lenguaje de definición de datos DDL⁹

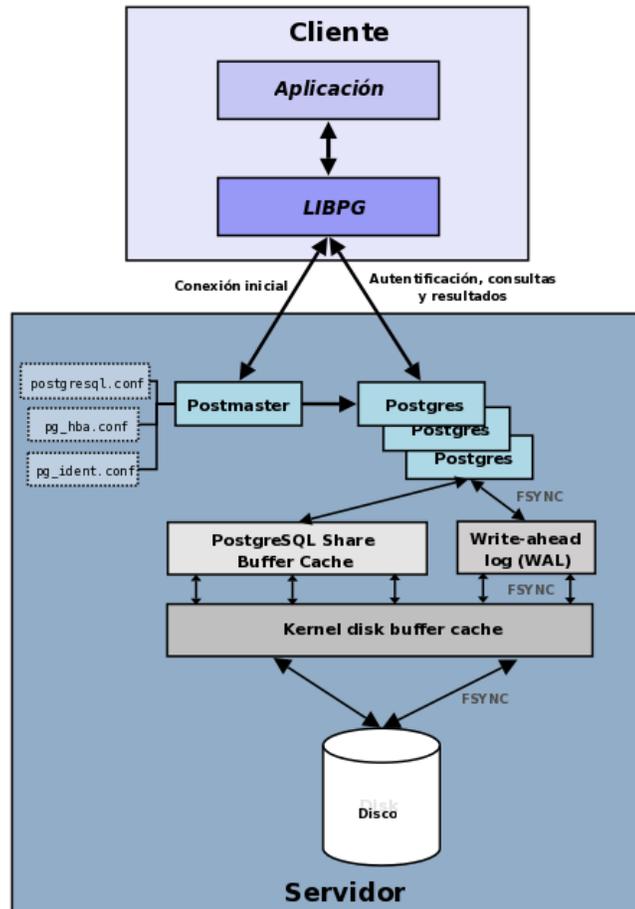


Figura 1. 2: Arquitectura funcional PostgreSQL.
Fuente : [WEB31]

⁸ ACID: Atomicidad, Consistencia, Aislamiento y Durabilidad.

⁹ DDL: Lenguaje de Definición de Datos

1.1.2. Servidores de Aplicaciones.

Un servidor de aplicaciones consiste en un contenedor que abarca la lógica de negocio de un sistema, y provee respuestas a las peticiones de distintos dispositivos que tienen acceso a ella. Trabaja con el modelo de cliente-servidor, donde el lado del cliente ejecuta requerimientos o peticiones de procesamiento y al otro lado, el servidor se encarga de procesar y responder dichas peticiones.

❖ HTTP Apache.

Es un servidor web HTTP ¹⁰de código abierto, para plataformas Unix (BSD, GNU/Linux, etc.), Microsoft Windows, Macintosh y otras, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual.

El servidor Apache se desarrolla dentro del proyecto HTTP Server (httpd) de la Apache Software Foundation [**WEB2**].

Características:

- Altamente configurables.
- Bases de datos de autenticación y negociado de contenido.
- Modular.
- Código abierto.
- Multi-plataforma.
- Extensible.
- Fácil conseguir ayuda/soporte.
- La licencia Apache es una descendiente de la licencias BSD, no es GPL. Esta licencia permite hacer lo que quiera con el código fuente siempre y cuando se reconozca su trabajo.

A continuación se muestra como el servidor web apache interactúa con el cliente mediante una petición y respuesta.

¹⁰ HTTP: Protocolo de Transferencia de Hipertexto

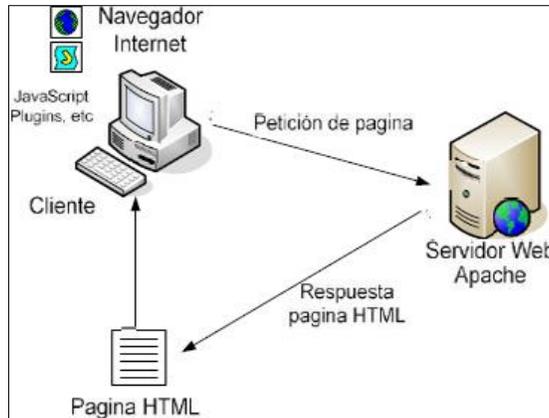


Figura 1. 3: Arquitectura Funcional Servidor web Apache.
Fuente: [WEB32]

❖ Tomcat.

Funciona como un contenedor de servlets desarrollado bajo el proyecto Jakarta en la Apache Software Foundation. Tomcat implementa las especificaciones de los servlets y de JavaServer Pages (JSP) de Sun Microsystems. Funciona en cualquier sistema operativo que disponga de la máquina virtual Java [WEB3].

Características:

Tomcat 7.x

- Implementado de Servlet¹¹ 3.0 JSP 2.2 y EL 2.2
- Mejoras para detectar y prevenir “fugas de memoria” en las aplicaciones web
- Limpieza interna de código
- Soporte para la inclusión de contenidos externos directamente en una aplicación web

En el siguiente grafico se puede observar la funcionalidad del servidor de aplicaciones tomcat que actúa como un contenedor de servlets para acceder a un servidor de base de datos mediante la petición o llamada de un cliente web.

¹¹ Servlet: La palabra servlet deriva de otra anterior, applet, que se refería a pequeños programas que se ejecutan en el contexto de un navegador web.

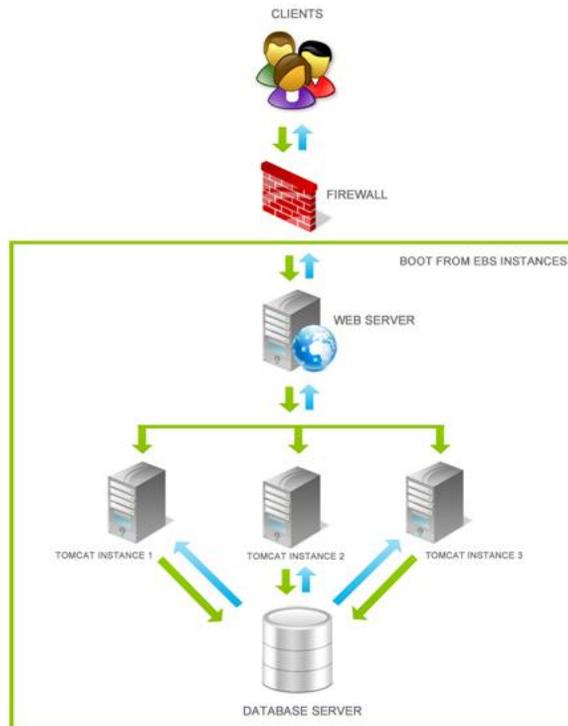


Figura 1. 4: Arquitectura funcional Apache Tomcat.
Fuente: Propia [WEB33]

1.1.3. Lenguajes de Programación.

Los lenguajes de programación permiten codificar o preparar los documentos necesarios para representar la funcionalidad de una aplicación que contiene una o más páginas web.

❖ CSS.

El nombre hojas de estilo en cascada viene del inglés Cascading Style Sheets, del que toma sus siglas. Es un lenguaje usado para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML¹² y XHTML. CSS es la mejor forma de separar los contenidos y su presentación y es imprescindible para crear páginas web complejas. La información de estilo puede ser adjuntada como un documento separado o en el mismo documento HTML.

El W3C (World Wide Web Consortium) es el encargado de formular la especificación de las hojas de estilo que servirán de estándar para navegadores [WEB4].

¹²HTML: Hypertext Markup Language.

Sintaxis:

CSS tiene una sintaxis muy sencilla, que usa unas cuantas palabras claves tomadas del inglés para especificar los nombres de sus selectores, propiedades y atributos.

Una hoja de estilos CSS consiste en una serie de reglas. Cada regla consiste en uno o más selectores y un bloque de estilos con los estilos a aplicar para los elementos del documento que cumplan con el selector que les precede. Cada bloque de estilos se define entre llaves, y está formado por una o varias declaraciones de estilo con el formato propiedad: valor; [WEB5].

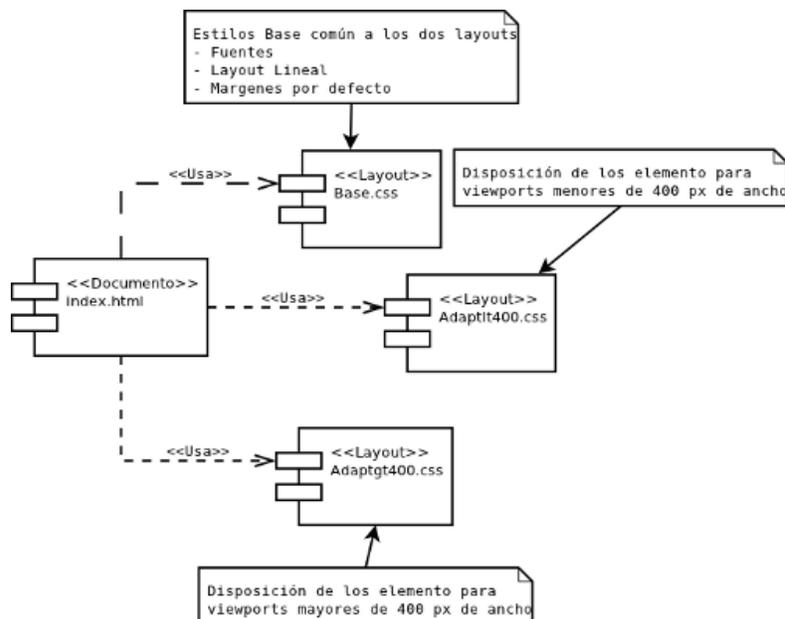


Figura 1. 5: Arquitectura Funcional CSS
Fuente:[WEB34]

❖ PHP.

PHP es un acrónimo recursivo que significa PHP Hypertext Pre-processor. Es un lenguaje de programación interpretado (Lenguaje de alto rendimiento), diseñado originalmente para la creación de páginas web dinámicas. Se usa principalmente para la interpretación del lado del servidor (server-side scripting) pero actualmente puede ser utilizado desde una interfaz de línea de comandos o en la creación de otros tipos de programas incluyendo aplicaciones con interfaz gráfica usando las bibliotecas Qt o GTK+ [WEB6].

Características:

- Publicado bajo la PHP License, la Free Software Foundation considera esta licencia como software libre.

- Puede ser desplegado en la mayoría de los servidores web.
- La curva de aprendizaje es muy corta.
- Puede ser ejecutado en la mayoría de los sistemas operativos, tales como Unix (y de ese tipo, como Linux o Mac OS X) y Microsoft Windows.
- Permite la conexión a diferentes tipos de servidores de bases de datos tales como MySQL, PostgreSQL, Oracle, ODBC, DB2, Microsoft SQL Server, Firebird y SQLite.
- El código fuente escrito en PHP es invisible al navegador web y al cliente ya que es el servidor el que se encarga de ejecutar el código y enviar su resultado HTML al navegador. Esto hace que la programación en PHP sea segura y confiable.
- Capacidad de expandir su potencial utilizando módulos (llamados ext's o extensiones).
- Posee una amplia documentación en su sitio web oficial.
- Permite aplicar técnicas de programación orientada a objetos.
- No requiere definición de tipos de variables aunque sus variables se pueden evaluar también por el tipo que estén manejando en tiempo de ejecución.
- Tiene manejo de excepciones (desde PHP5).
- El programador puede aplicar cualquier técnica de programación o de desarrollo que le permita escribir código ordenado, estructurado y manejable.

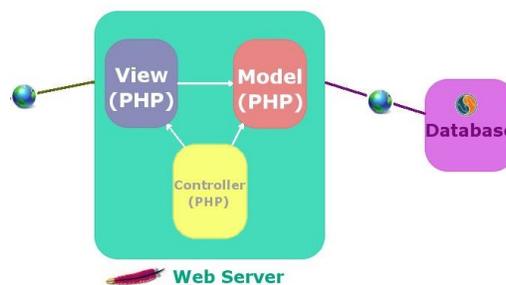


Figura 1. 6: Arquitectura funcional PHP.
Fuente: [WEB35]

❖ JAVASCRIPT.

Es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a prototipos, imperativo, débilmente tipado y dinámico, desarrollado para incrementar las funcionalidades del lenguaje HTML. Se utiliza principalmente en su forma del lado del cliente (client-side), implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas [WEB7].

Características:

- Se utiliza principalmente en el lado del cliente, implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas, en bases de datos locales al navegador.
- JavaScript se diseñó con una sintaxis similar al C, aunque adopta nombres y convenciones del lenguaje de programación Java. Sin embargo Java y JavaScript no están relacionados y tienen semánticas y propósitos diferentes.
- Todos los navegadores modernos interpretan el código JavaScript integrado en las páginas web. Para interactuar con una página web se provee al lenguaje JavaScript de una implementación del Document Object Model (DOM).
- No es necesario declarar los tipos de variables que van a utilizarse.
- Maneja objetos dentro de nuestra página Web y sobre ese objeto podemos definir diferentes eventos. Dichos objetos facilitan la programación de páginas interactivas [WEB8].
- Responde a eventos en tiempo real. Con esto podemos cambiar totalmente el aspecto de nuestra página al gusto del usuario, evitándonos tener en el servidor una página para cada gusto, hacer cálculos en base a variables cuyo valor es determinado por el usuario, etc.

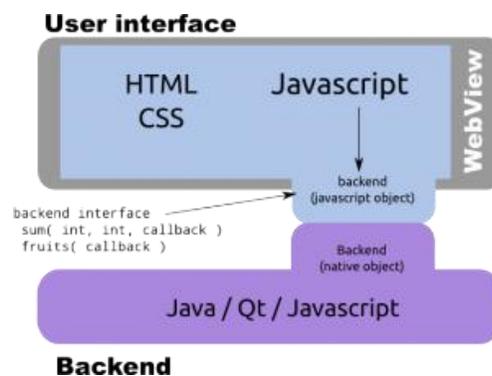


Figura 1. 7: Arquitectura funcional JavaScript.

Fuente: [WEB36]

1.1.4. Frameworks.

Un framework simplifica el desarrollo de una aplicación mediante la automatización de algunos de los patrones utilizados para resolver las tareas comunes. Además, un framework proporciona estructura al código fuente, forzando al desarrollador a crear código más legible y más fácil de mantener. Por último, un framework facilita la programación de aplicaciones, ya que encapsula operaciones complejas en instrucciones sencillas [LIB5].

Es una estructura conceptual y tecnológica de soporte definido, normalmente con artefactos o módulos de software concretos, con base a la cual otro proyecto de software puede ser más fácilmente organizado y desarrollado. Típicamente, puede incluir soporte de programas, bibliotecas, y un lenguaje interpretado, entre otras herramientas, para así ayudar a desarrollar y unir los diferentes componentes de un proyecto [WEB9].

❖ **Symfony 1.16.**

Symfony es un completo framework diseñado para optimizar, gracias a sus características, el desarrollo de las aplicaciones web. Separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja. Además, automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación [LIB5].

Características [WEB10]:

- Fácil de instalar y configurar en sistemas Windows, Mac y Linux.
- Compatible solamente con PHP 5, para asegurar el mayor rendimiento y acceso a las características más avanzadas de PHP.
- Basado en la premisa de “convenir en vez de configurar”, en la que el desarrollador solo debe configurar aquello que no es convencional.
- Preparado para aplicaciones empresariales, ya que se puede adaptar con facilidad a las políticas y arquitecturas propias de cada empresa u organización.
- Flexible hasta cualquier límite y extensible mediante un completo mecanismo de plugins.
- Publicado bajo licencia MIT de software libre y apoyado por una empresa comprometida con su desarrollo.
- Traducido a más de 40 idiomas y fácilmente traducible a cualquier otro idioma.
- El manejo de cache reduce el uso de banda ancha y la carga del servidor.
- La facilidad de soportar autenticación y credenciales facilita la creación de áreas restringidas y manejo de seguridad de los usuarios.
- El enrutamiento y las URLs inteligentes hacen amigable las direcciones de las páginas de la aplicación.
- Independiente del sistema gestor de bases de datos. Su capa de abstracción y el uso de Doctrine, permiten cambiar con facilidad de SGBD en cualquier fase del proyecto.

- Aunque utiliza MVC (Modelo Vista Controlador), tiene su propia forma de trabajo en este punto, con variantes del MVC clásico como la capa de abstracción de base de datos, el controlador frontal y las acciones.
- Sigue la mayoría de mejores prácticas y patrones de diseño para la web.
- Permite un mantenimiento muy sencillo.
- Está compuesto por una potente línea de comandos que facilitan generación de código, lo cual contribuye a ahorrar tiempo de trabajo.

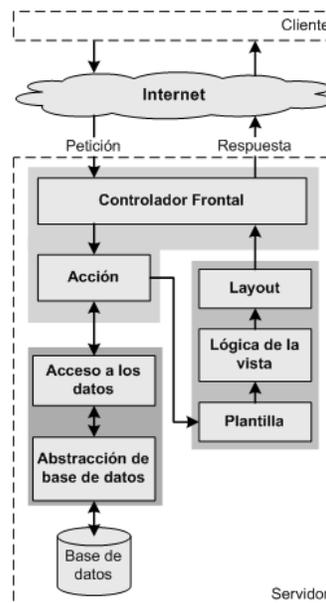


Figura 1. 8: Arquitectura funcional Symfony.
Fuente: [WEB 37]

❖ Doctrine.

Doctrine es un mapeador de objetos-relacional (ORM¹³) escrito en PHP que proporciona una capa de persistencia para objetos PHP. Es una capa de abstracción que se sitúa justo encima de un SGBD [WEB11].

Características:

- Se necesita un bajo nivel de configuración para empezar un proyecto.
- Puede generar clases a partir de una base de datos existente y después el programador puede especificar relaciones y añadir funcionalidad extra a las clases autogeneradas.

¹³ ORM: Object-Relational Mapping

- No es necesario generar o mantener complejos esquemas XML de base de datos como en otros frameworks.
- Otra característica importante de Doctrine es la posibilidad de escribir consultas de base de datos utilizando un dialecto de SQL denominado DQL (Doctrine Query Language) que está inspirado en Hibernate (Java).
- Soporte para datos jerárquicos.
- Soporte para hooks (métodos que pueden validar o modificar las escrituras y lecturas de la base de datos) y eventos para manejar la lógica de negocio relacionada.
- Herencia.
- Un framework de caché que utiliza diversos motores como memcached, SQLite o APC.
- Transacciones ACID.
- Diversos comportamientos del modelo (conjuntos anidados, internacionalización, log, índice de búsqueda);

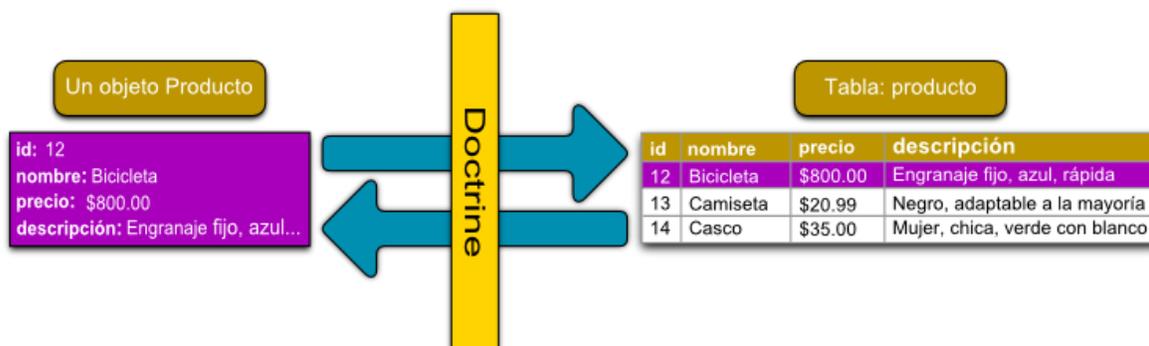


Figura 1. 9: Arquitectura funcional Doctrine.
Fuente: [WEB38]

❖ Ext JS 4.

Ext JS ayuda a mejorar la experiencia del usuario, proporciona componentes con funcionalidades avanzadas y de fácil implementación.

Es una biblioteca de JavaScript para el desarrollo de aplicaciones web interactivas usando tecnologías como AJAX, DHTML¹⁴ y DOM¹⁵. Fue desarrollada por Sencha [WEB12].

¹⁴ DHTML: Dynamic HTML Usado para crear efectos de páginas web Dinámicas.

¹⁵ DOM: Document Object Model.

Características:

ExtJS posee características muy importantes para el desarrollo de aplicaciones web, porque se puede logra crear páginas web dinámicas e interactivas con la finalidad de ofrecer una interfaz amigable al usuario.

- Dispone de un conjunto de componentes (widgets) para incluir dentro de una aplicación web, como:
 - ✓ Cuadros y áreas de texto.
 - ✓ Campos para fechas.
 - ✓ Campos numéricos.
 - ✓ Combos.
 - ✓ Radiobuttons y checkboxes.
 - ✓ Editor HTML.
 - ✓ Árbol de datos.
 - ✓ Pestañas.
 - ✓ Barra de herramientas.
 - ✓ Menús al estilo de Windows.
 - ✓ Paneles divisibles en secciones.
 - ✓ Sliders.
- Elementos de datos (con modos de sólo lectura, datos ordenables, columnas que se pueden bloquear y arrastrar, etc.).
- Varios de sus componentes están capacitados para comunicarse con el servidor usando AJAX.
- También contiene numerosas funcionalidades que permiten añadir interactividad a las páginas HTML, como:
 - ✓ Cuadros de diálogo.
 - ✓ Quicktips para mostrar mensajes de validación e información sobre campos individuales.

Arquitectura de ExtJS 4

- ✓ Modelo
- ✓ Vista
- ✓ Controlador
- ✓ Estructuración de carpetas

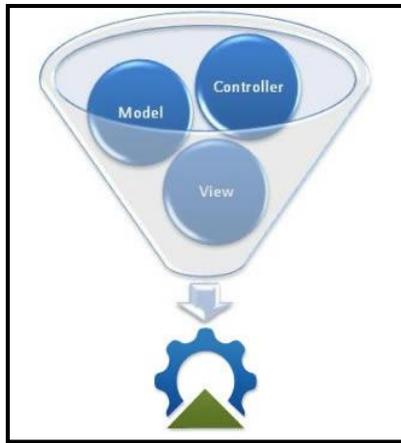


Figura 1. 10: Arquitectura MVC de una Aplicación Ext JS
Fuente: [WEB13].

- ✓ **Modelo** es una colección de campos y datos. Un Modelo sabe como persistir sus valores el mismo a través del paquete de datos y puede ser relacionado con otros Modelos por medio de asociaciones. Los Modelos son la evolución de lo que en Ext JS 3 eran los Records, y normalmente son usados en conjunto con los Stores para desplegar datos en componentes como los grids o combos.
- ✓ Una **Vista** es cualquier tipo de componente visual, pueden ser grids, arboles o paneles.
- ✓ **Controlador** es el código que hace que tu aplicación trabaje, es el código que lleva la lógica para desplegar las vistas o instanciar Modelos por ejemplo [WEB22].

Control de Eventos

- ✓ Eventos en ExtJS
- ✓ Clase Observable
- ✓ Eventos personalizados
- ✓ Controladores de Eventos

Modelo de componentes y Contenedores

- ✓ Component Manager
- ✓ Ciclo de vida
- ✓ Configuración de componentes
- ✓ Layout Managers
- ✓ Viewport
- ✓ Panel-TabPanel

Formularios

- ✓ FormPanel
- ✓ Campos y componentes de captura
- ✓ Layouts y componentes

Herencia y Jerarquía

- ✓ Jerarquía de componentes
- ✓ Sobrecargar y preconfiguración de una clase

Ext.data y Componentes

- ✓ Conexiones
- ✓ Stores
- ✓ Modelos

Estructura de Archivos ExtJS

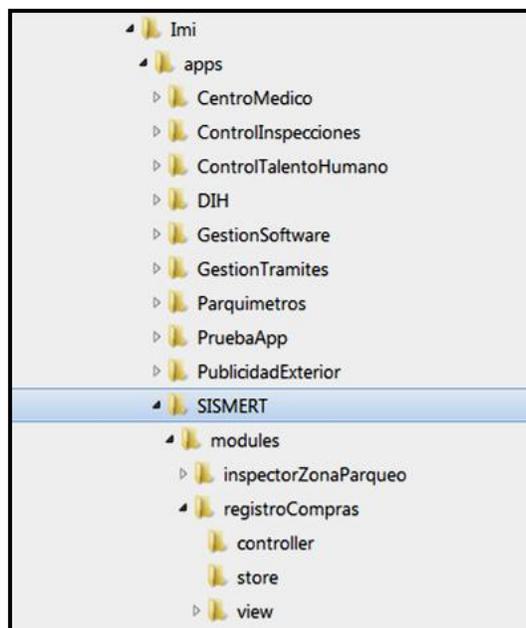


Figura 1. 11: Estructura de Directorios de una Aplicación MVC.
Fuente: Propia

Las aplicaciones ExtJS siguen una estructura de directorio unificado que es el mismo para cada aplicación. Todas las clases se colocan en la carpeta app, que contiene a su vez sub-carpetas para generar el espacio de nombres de modelos, vistas, controladores y almacenes (stores).

❖ Sencha Touch 2

Sencha Touch es un framework que permite crear aplicaciones web para dispositivos táctiles y móviles, utiliza los estándares web de HTML5, CSS3 y Javascript para desarrollar aplicaciones web para dispositivos móviles con la apariencia de ser nativos de los sistemas IOS, Android y Blackberry. Sencha Touch ha sido creada por la empresa Sencha (los mismos creadores del framework Ext JS 4) el mismo que mejora su rendimiento al estar construido sobre la misma arquitectura de ExtJS 4 y sistema de clases, aprovechando la robusta selección de características de ExtJS [WEB14][WEB15].

Estándares Web de Sencha Touch

Es el primer framework de aplicaciones web completo y específicamente desarrollado para aprovechar la mayor potencia y flexibilidad de HTML5, CSS3 y JavaScript.

Sencha Touch hace uso de los nuevos elementos disponibles en la especificación de los estándares web para reproducir vídeo, audio, y además guardar datos en línea. Incorpora una extensa capa de estilo CSS3 que aporta entre otras mejoras de un aspecto nativo a las interfaces desarrolladas con él. Con el uso de estos estándares web Sencha toma ventaja sobre la aceleración de hardware para liberar el acceso hacia la aplicación sin importar que navegador o que dispositivo móvil se esté usando.

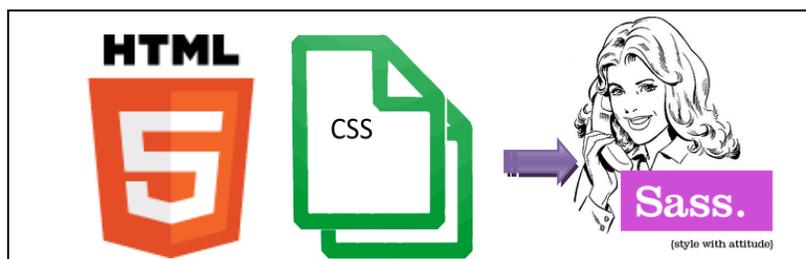


Figura 1. 12: Estándares web Sencha Touch.
Fuente: Propia Edición Gráfica

Se puede optimizar el peso de la librería en entornos productivos de forma trivial llegando a pesar menos de 120Kb en versiones comprimidas con Gzip y minificadas.

Implementa temas basados en imágenes utilizando SASS¹⁶ que es una extensión de CSS, para generar temas de las aplicaciones web para dispositivos móviles; donde los módulos de CSS3 aún no están implementados o su ejecución es muy lenta. Abriendo la puerta para conseguir que las aplicaciones hechas en Sencha puedan ser ejecutadas en Windows Phone y permitiendo de manera fácil y rápida modificar ciertas características como color, bordes y degradados de una aplicación.

¹⁶ SASS: style with attitude, es una extensión de CSS.

Características:

- a) **Multiplataforma:** La misma aplicación que se desarrolla en Sencha Touch va a funcionar en muchos teléfonos móviles inteligentes y tabletas (mediante la configuración de los perfiles de los dispositivos), acortando la curva de aprendizaje porque ya no se tiene que desarrollar aplicaciones individuales para cada sistema operativo de cada dispositivo móvil, sino una sola, ahorrando tiempo y dinero en su desarrollo.
- b) **Dispositivos Táctiles:** Esta diseñado principalmente para dispositivos táctiles (aunque también existen estándares para funcionar con el mouse y el teclado), por ello proporciona una variedad de eventos táctiles como:
- **Toque:** Tocar con un solo dedo en la pantalla.
 - **Doble toque:** Dos toques rápidos en la pantalla.
 - **Deslizar:** Al mover un dedo por la pantalla de izquierda a derecha o de arriba hacia abajo.
 - **Apretar o extender:** Cuando toca la pantalla con dos dedos acercándolos y alejándolos.
 - **Girar:** Cuando coloca dos dedos en la pantalla y los gira hacia la derecha o hacia la izquierda.

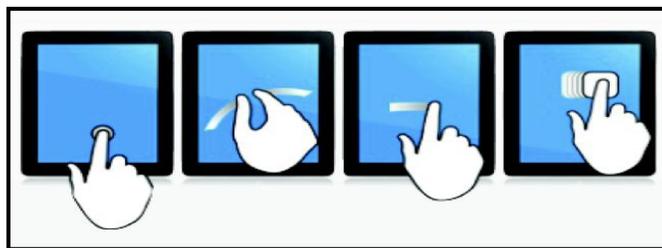


Figura 1. 13: Características Sencha Touch Dispositivo Táctil y Móvil.
Fuente: [WEB14]

c) Componentes de trabajo

Sencha Touch principalmente trabaja con Javascript, y entre los componentes de más utilidad se puede señalar los que tenemos a continuación:

Paneles, Toolbars, TabPanels, Listas, Listas Anidadas, Carouseles, Botones, Scrolls, Campos de Texto, Campo de Contraseña, Iconos, DatePicker, y Diversidad de Efectos [WEB16].

Integración de datos

Al igual que el framework de ExtJS, Sencha Touch implementa el patrón de diseño MVC en el lado del cliente y nos ofrece una API rica y eficaz para manejar flujos de datos desde una gran variedad de fuentes. Podemos acceder y enviar datos directamente a través de AJAX, JSONP o YQL.

Además añadir los datos obtenidos a elementos específicos de cada una de las vistas en el lado del cliente, y utilizar los datos sin conexión gracias a que se puede crear almacenes locales. Ver Figura 6

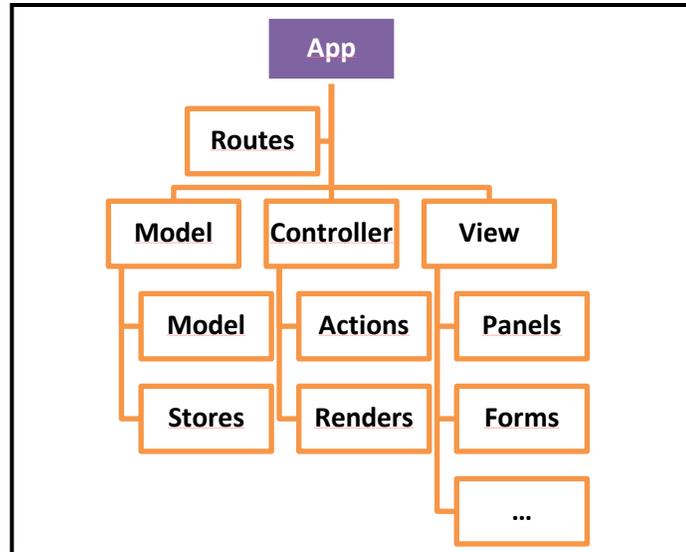


Figura 1. 14: Arquitectura MVC Sencha Touch.

Fuente: Propia

Empaquetado Nativo

Sencha Touch tiene su propio Empaquetado Nativo para convertir la aplicación a IOS o Android, a través de un simple comando, por lo que pero no es necesario escribir un propio wrapper o usar terceras aplicaciones o soluciones para generar una aplicación nativa.

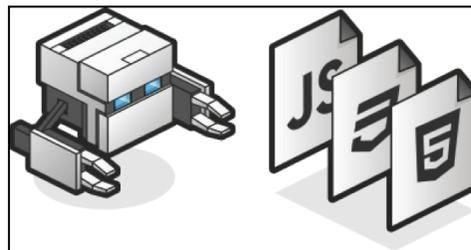


Figura 1. 15: Empaquetado Nativo de una Aplicación Touch.

Fuente: [WEB14]

Además Sencha Touch interactúa perfectamente con PhoneGap¹⁷ Build que puede ser usado para distribuir las aplicaciones a través de un mecanismo que empotra a la aplicación en una Shell nativa para acceder a elementos del dispositivo de la forma más sencilla posible también, gracias a las características de PhoneGap se puede hacer uso de la API nativa del dispositivo para acceder a la lista de contactos, la cámara, el GPS¹⁸, batería, etc directamente desde JavaScript.

¹⁷PhoneGap: Usado para distribuir aplicaciones en la App Store de Sencha Touch o en Android Marketplace.

¹⁸ GPS: Global Positioning System: Sistema de posicionamiento global

1.1.5. Reportes

❖ JasperReports

JasperReports es la mejor herramienta de código libre en Java para generar reportes. Puede entregar ricas presentaciones o diseños en la pantalla, para la impresora o para archivos en formato PDF, HTML, RTF, XLS, CSV y XML. Está completamente escrita en Java y se puede utilizar en una gran variedad de aplicaciones de Java, incluyendo J2EE o aplicaciones Web, para generar contenido dinámico.

JasperReports trabaja en forma similar a un compilador y a un intérprete. El usuario diseña el reporte codificándolo en XML de acuerdo a las etiquetas y atributos definidos en un archivo llamado `jasperreports.dtd` (parte de JasperReports). Usando XML el usuario define completamente el reporte, describiendo donde colocar texto, imágenes, líneas, rectángulos, cómo adquirir los datos, como realizar ciertos cálculos para mostrar totales, etc. **Fuente:** [WEB17].

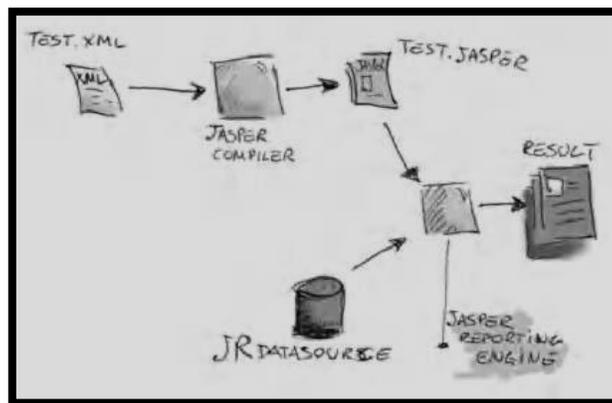


Figura 1. 16: Funcionalidad de JasperReports.
Fuente: [WEB17].

De la figura anterior se puede describir como JasperReports realiza el proceso de creación de un reporte.

El archivo fuente XML debe ser compilado para obtener un reporte real. La versión compilada del fuente es nombrada "archivo jasper" (este termina con `.jasper`). Un Archivo jasper es el compilado de un código fuente. Además del archivo jasper, se necesita datos para producir un reporte. En algunos casos se necesita generar un reporte que no muestre datos dinámicos, solo texto estático, pero esto puede simplificarse a un reporte que tiene solamente un registro vacío. Para proporcionar estos registros al "jasper engine" se necesita usar una interfaz especial específica llamada `JRDataSource`.

Una fuente de datos + un Archivo jasper = un “archivo print”. Un “archivo print” puede exportarse en muchos formatos como PDF, HTML, RTF, XML, XLS, CVS, etc.

❖ iReport

iReport es un diseñador visual de código libre para JasperReports escrito en Java. Es un programa que ayuda a los usuarios y desarrolladores que usan la librería JasperReports para diseñar reportes visualmente. A través de una interfaz rica y simple de usar, iReport provee las funciones más importantes para crear reportes en poco tiempo.

iReport provee a los usuarios de JasperReports una interfaz visual para construir reportes, generar archivos “jasper” y “print” de prueba. iReport nació como una herramienta de desarrollo, pero puede utilizarse como una herramienta de oficina para adquirir datos almacenados en una base de datos, sin pasar a través de alguna otra aplicación. **Fuente:** [WEB17].

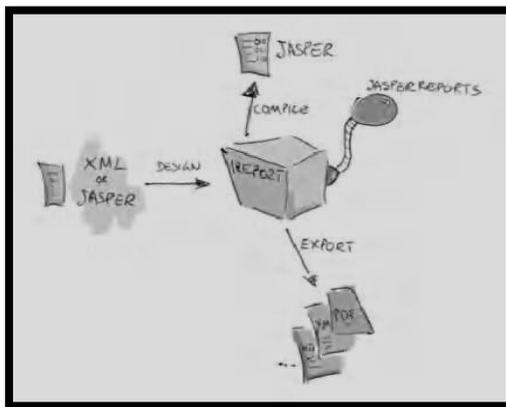


Figura 1. 17: Funcionalidad de iReport para generar un Reporte.

Fuente: [WEB17].

iReport puede leer y modificar ambos tipos de archivo, XML y jasper. A través de JasperReports, es capaz de compilar XML a archivos jasper y “ejecutar reportes” para llenarlos usando varios tipos de fuentes de datos (JRDataSource) y exportar el resultado a PDF, HTML, XLS, CSV.

1.1.6. Web Services

❖ Definición

Un servicio web (en inglés, Web services) es una tecnología que utiliza un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones.

Distintas aplicaciones de software desarrolladas en lenguajes de programación diferentes, y ejecutadas sobre cualquier plataforma, pueden utilizar los servicios web para intercambiar datos en redes de ordenadores como Internet. La interoperabilidad se consigue mediante la adopción de estándares abiertos. Las organizaciones OASIS¹⁹ y W3C son los comités responsables de la arquitectura y reglamentación de los servicios Web. Para mejorar la interoperabilidad entre distintas implementaciones de servicios Web se creó el organismo WS-I, encargado de desarrollar diversos perfiles para definir de manera más exhaustiva estos estándares [**WEB18**].

Los servicios que se pueda consumir u ofrecer no entregarán información sensible a los clientes, básicamente deberán entregarán resultados a una consulta hecha por un cliente el cuál envía parámetros y variables a funciones implementadas por el servidor, sin tener la necesidad de conocer ha detalle cómo están implementadas las mencionadas funciones.

En esta comunicación e intercambio de datos, se utilizan los denominados Web Services Protocol Stack definidos como el conjunto de servicios y protocolos de los Web Services.

Para realizar el Servicio web se utilizara XML (Extensible Markup Language), SOAP (Simple Object Access Protocol), y WSDL (Web Services Description Language), estos son los estándares para el protocolo de comunicación del Web-Services.

❖ XML

Es un lenguaje de marcas desarrollado por el World Wide Web Consortium (W3C). Deriva del lenguaje SGML y permite definir la gramática de lenguajes específicos (de la misma manera que HTML es a su vez un lenguaje definido por SGML) para estructurar documentos grandes. A diferencia de otros lenguajes, XML da soporte a bases de datos, siendo útil cuando varias aplicaciones se deben comunicar entre sí o integrar información. (Bases de datos Silberschatz).

XML no ha nacido sólo para su aplicación en Internet, sino que se propone como un estándar para el intercambio de información estructurada entre diferentes plataformas. Se puede usar en bases de datos, editores de texto, hojas de cálculo y casi cualquier cosa imaginable.

XML es una tecnología sencilla que tiene a su alrededor otras que la complementan y la hacen mucho más grande y con unas posibilidades mucho mayores. Tiene un papel muy importante en la actualidad ya que permite la compatibilidad entre sistemas para compartir la información de una manera segura, fiable y fácil [**WEB19**].

¹⁹ OASIS: Acrónimo de Organization for the Advancement of Structured Information Standards

Ventajas del XML

- ✓ Es extensible: Después de diseñado y puesto en producción, es posible extender XML con la adición de nuevas etiquetas, de modo que se pueda continuar utilizando sin complicación alguna.
- ✓ El analizador es un componente estándar, no es necesario crear un analizador específico para cada versión de lenguaje XML. Esto posibilita el empleo de cualquiera de los analizadores disponibles. De esta manera se evitan bugs y se acelera el desarrollo de aplicaciones.
- ✓ Si un tercero decide usar un documento creado en XML, es sencillo entender su estructura y procesarla. Mejora la compatibilidad entre aplicaciones. Podemos comunicar aplicaciones de distintas plataformas, sin que importe el origen de los datos, es decir, podríamos tener una aplicación en Linux con una base de datos Postgres y comunicarla con otra aplicación en Windows y Base de Datos MS-SQL Server.
- ✓ Transformamos datos en información, pues se le añade un significado concreto y los asociamos a un contexto, con lo cual tenemos flexibilidad para estructurar documentos.

Partes de un documento XML

Un documento XML está formado por el prólogo y por el cuerpo del documento así como texto de etiquetas que contiene una gran variedad de efectos positivos o negativos en la referencia opcional a la que se refiere el documento, hay que tener mucho cuidado de esa parte de la gramática léxica para que se componga de manera uniforme.

❖ WSDL

Las siglas de Web Services Description Language, un formato XML que se utiliza para describir servicios Web. La versión 1.0 fue la primera recomendación por parte del W3C y la versión 1.1 no alcanzó nunca tal estatus. La versión 2.0 se convirtió en la recomendación actual por parte de dicha entidad.

WSDL describe la interfaz pública a los servicios Web. Está basado en XML y describe la forma de comunicación, es decir, los requisitos del protocolo y los formatos de los mensajes necesarios para interactuar con los servicios listados en su catálogo. Las operaciones y mensajes que soporta se describen en abstracto y se ligan después al protocolo concreto de red y al formato del mensaje.

Así, WSDL se usa a menudo en combinación con SOAP y XML Schema. Un programa cliente que se conecta a un servicio web puede leer el WSDL para determinar qué funciones están disponibles en el servidor. Los tipos de datos especiales se incluyen en el archivo WSDL en forma de XML Schema. El cliente puede usar SOAP para hacer la llamada a una de las funciones listadas en el WSDL.

El WSDL nos permite tener una descripción de un servicio web. Especifica la interfaz abstracta a través de la cual un cliente puede acceder al servicio y los detalles de cómo se debe utilizar [WEB20].

Estructura del WSDL

La estructura del WSDL tiene los siguientes elementos:

✓ **Tipos de Datos**

<types>: Esta sección define los tipos de datos usados en los mensajes. Se utilizan los tipos definidos en la especificación de esquemas XML.

✓ **Mensajes**

<message>: Aquí definimos los elementos de mensaje. Cada mensaje puede consistir en una serie de partes lógicas. Las partes pueden ser de cualquiera de los tipos definidos en la sección anterior.

✓ **Tipos de Puerto**

<portType>: Con este apartado definimos las operaciones permitidas y los mensajes intercambiados en el Servicio.

✓ **Bindings**

<binding>: Especificamos los protocolos de comunicación usados.

✓ **Servicios**

<service>: Conjunto de puertos y dirección de los mismos. Esta parte final hace referencia a lo aportado por las secciones anteriores.

Con estos elementos no se puede saber qué hace un servicio pero sí disponemos de la información necesaria para interactuar con él (funciones, mensajes de entrada/salida, protocolos).

❖ **SOAP**

Es un protocolo estándar que define cómo dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos XML. Este protocolo deriva de un protocolo creado por David Winer en 1998, llamado XML-RPC²⁰. SOAP fue creado por Microsoft, IBM y otros y está actualmente bajo el auspicio de la W3C. Es uno de los protocolos utilizados en los servicios Web [WEB21].

²⁰ XML-RPC: protocolo antecesor de SOAP.

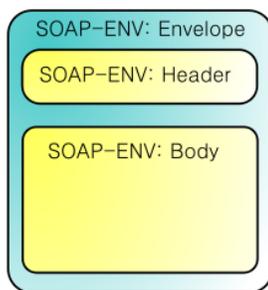


Figura 1. 18: Estructura de SOAP
Fuente: [WEB21].

Características

SOAP puede formar la capa base de una "pila de protocolo de web service", ofreciendo un framework de mensajería básica en la cual los web services se puedan construir. Este protocolo basado en XML consiste de tres partes: un sobre (envelope), el cual define qué hay en el mensaje y cómo procesarlo; un conjunto de reglas de codificación para expresar instancias de tipos de datos (header).

Una conversión para representar llamadas a procedimientos y respuestas (body). El protocolo SOAP tiene tres características principales:

- 1) Extensibilidad (seguridad y WS-routing son extensiones aplicadas en el desarrollo).
- 2) Neutralidad (SOAP puede ser utilizado sobre cualquier protocolo de transporte como HTTP, SMTP²¹, TCP²²).
- 3) Independencia (SOAP permite cualquier modelo de programación).

La arquitectura SOAP consiste de muchas capas de especificación: para el formato del mensaje, MEP²³(Message Exchange Patterns), subyacentes enlaces de protocolo de transporte, modelo de procesamiento de mensajes, y extensibilidad del protocolo. SOAP es el sucesor de XML-RPC, a pesar de que toma el transporte y la neutralidad de la interacción y el envelope / header / body.

➤ Librería NUSOAP

NuSOAP es un kit de herramientas (ToolKit) para desarrollar Web Services bajo el lenguaje PHP. Está compuesto por una serie de clases que nos hacen mucho más fácil el desarrollo de un Web Services.

²¹ SMTP: (Simple Mail Transfer Protocol), para el correo electrónico.

²² TCP: Protocolo de Control de Transmisión.

²³ MEP: (Message Exchange Patterns), Modelo de procesamiento de mensajes de SOAP.

Provee soporte para el desarrollo de clientes (aquellos que consumen los Servicios Web) y de servidores (aquellos que los proveen). NuSOAP está basado en SOAP 1.1, WSDL 1.1 y HTTP 1.0/1.1 [WEB25].

Características

- ✓ Permite a los desarrolladores crear y consumir servicios web SOAP.
- ✓ No requiere ninguna extensión especial de PHP.
- ✓ Soporta gran parte del SOAP 1.1, y puede generar WSDL 1.1
- ✓ También se lo utiliza en la serialización.

La implementación del proceso de transacciones electrónicas se lo realiza con servicios web para el envío y recepción de la información desde la aplicación hacia el proveedor que facilita la realización de pagos electrónicos.

Las herramientas seleccionadas para la creación de los servicios web es el lenguaje de programación PHP y la librería NUSOAP que nos permite crear y consumir servicios web.

A continuación se muestra las características de la librería **NUSOAP** que es la herramienta seleccionada para crear y consumir servicios web para el desarrollo del presente proyecto. Se selecciono la librería **Php-WSDL-Creator**, que también permite interactuar con los servicios web, con el objetivo de realizar una comparativa entre las dos herramientas.

Características NUSOAP	Características Php-WSDL-Creator
Permite desarrollar Web Services bajo el lenguaje PHP y generar WSDL.	Se puede utilizar para generar el Lenguaje de definición de un servicio Web (WSDL).
Compuesto por una serie de clases que permite el fácil desarrollo de Web Services.	Las funciones de un servicio web se las importa desde una clase php.
Provee soporte para el desarrollo de clientes (aquellos que consumen los Web Services) y de servidores (aquellos que los proveen).	Analiza el código de una determinada clase de archivos de script PHP y genera una definición WSDL de la lista de las funciones de la clase.
Permite crear y consumir Servicios Web basados en SOAP 1.1, WSDL 1.1 y HTTP 1.0/1.1.	Ciertas funciones pueden ser omitidas a no ser incluidas en la definición WSDL basado en el tipo de acceso a las funciones.

Para su uso no se requiere ninguna extensión especial de PHP.	El WSDL generado se puede guardar en un archivo determinado, devuelve una cadena o sirve para descargar.
Se basa en el uso de XML. Clase Nativa de SOAP.	Software libre LGPL.
Herramienta de software libre	Para su funcionalidad es necesario que los métodos de la clase php estén bien documentados.
La instalación es bastante sencilla	WSDL Creator no puede manejar una clase sin métodos.
Es uno de los que están en una fase de desarrollo mucho más avanzada.	Si un archivo no contiene una clase no generará un error.
No necesita módulos adicionales y se lo utiliza en la serialización..	

Tabla 1. 1: Características comparativas de NUSOP y WSDL-GENERATOR
Fuente: Propia

Como se puede observar en la comparación anterior la librería NUSOAP está en una fase de desarrollo más avanzada que la librería **wsdl-creator**, además de poder acceder a la clase nativa para la creación de un servicio web.

1.2. Análisis de Formas de Pagos Electrónicos

Un sistema de pago electrónico realiza la transferencia del dinero entre el comprador y vendedor en una compra-venta electrónica. Una Transacción electrónica es un contrato realizado mediante medios electrónicos a través de la Red.

Los sistemas de pagos en línea consisten en una serie de transacciones al final de las cuales se puede realizar pagos mediante el uso de una tarjeta de crédito que es puesta en circulación por una entidad financiera. El análisis de las formas de pago que se plantearon es a través de los proveedores de servicio los mismos que realizan el proceso de transacciones electrónicas.

1.2.1. Protocolos de Seguridad en Transacciones Electrónicas

Los protocolos de seguridad permiten que la información enviada en una transacción electrónica viaje de forma segura, utilizando métodos de encriptación para que la información no pueda ser visible a terceras personas. Con la implementación de seguridad a través de estos protocolos el usuario puede estar tranquilo porque su información está segura al realizar un proceso de transacción electrónica.

➤ **SSL (Secure Sockets Layer)**

SSL proporciona autenticación y privacidad de la información entre extremos sobre Internet mediante el uso de criptografía. Habitualmente, sólo el servidor es autenticado (es decir, se garantiza su identidad) mientras que el cliente se mantiene sin autenticar.

Para pagar, el usuario debe rellenar un formulario con sus datos personales (tanto para el caso del envío de los bienes comprados, como para comprobar la veracidad de la información de pago), y los datos correspondientes a su tarjeta de crédito (número, fecha de caducidad, titular) [WEB29].

➤ **SET (Secure Electronic Transaction)**

Es un protocolo estándar para proporcionar seguridad a una transacción con tarjeta de crédito en redes de computadoras inseguras, en especial Internet.

SET surge de una solicitud de estándar de seguridad por VISA y MasterCard, utiliza técnicas criptográficas tales como certificados digitales y criptografía de clave pública para permitir a las entidades llevar a cabo una autenticación entre sí y además intercambiar información de manera segura [WEB28].

Comparación de Protocolos de Seguridad

	SSL	SET
Confidencialidad	X	X
Integridad	X	X
Autentifica los titulares de las tarjetas de crédito	X	X

	SSL	SET
Autentifica los comerciantes	X	X
Autentifica los bancos		X

Tabla 1. 2: Comparación Protocolos de Seguridad SSL y SET
Fuente: Propia

Desventajas de usar protocolo de seguridad SET:

- SET aun no termina de implantarse en el mercado como SSL.
- El despliegue de la información es muy lento y exige software especial, tanto para el comprador como para el comerciante.
- Aunque varios productos cumplan con el estándar SET, esto no significa necesariamente que sean compatibles.
- La implementación de un servicio para transacciones electrónicas es más costoso realizarlo con SET que con SSL.

Como se puede observar en la tabla comparativa de los protocolos de seguridad SET es la mejor opción para salvaguardar la información en la red, pero su implementación es muy costosa al requerir software especial, mientras que con SSL su implementación sería menos costosa, y al estar más tiempo en el mercado nos da la seguridad de que el proceso de una transacción se cumpla por completo.

1.2.2. Proveedores de Transacciones Electrónicas

A continuación se mencionan algunos proveedores que ofertan este servicio, describiendo las características de seguridad que tiene cada uno, y la forma en la que opera su servicio.

➤ **INTERDIN**

INTERDIN, es la operadora líder de tarjetas de crédito en el Ecuador, que conjuntamente con grandes del mercado ecuatoriano como son: Diners Club y Banco del Pichincha, aplica su filosofía de mercader, vender y servir de manera única, atendiendo a los tarjeta-habientes Diners Club, Visa Banco del Pichincha y Mastercard Banco del Pichincha.

Es un Portal de Servicios que ha incorporado la tecnología más avanzada para asegurar que la información que se suministra y se acceda esté debidamente protegida. También cuenta con el apoyo de Diners Club INTERDIN que presentan una plataforma segura, con distintos niveles de control para que los datos financieros estén seguros. Estas medidas permiten verificar correctamente la identidad

cuando se utiliza un servicio en línea, y protege la información mientras la misma viaja por Internet desde el computador hasta el servidor.

El Portal de Servicios transaccional utiliza el Protocolo de Seguridad "SSL" (Secure Socket Layer), el cual permite que la información viaje encriptada, por medio de códigos de claves secretas. SSL es un protocolo desarrollado por Netscape para transmitir documentos privados vía Internet. SSL trabaja usando una llave privada para codificar los datos que son transferidos sobre la conexión SSL, evitándose de esta manera que terceras personas vean este contenido o estos datos, capturen o repitan la información. Algunos Sitios en la Web usan este protocolo para obtener información confidencial del usuario, como lo son números y datos de tarjetas de crédito. Las páginas web que requieren una conexión SSL comienzan con https: en vez de http:.

También cuenta con "Barreras de Protección" o "Firewalls" para prevenir el acceso de alguna persona sin autorización al servidor. Todos los mensajes que entran o salen del servidor pasan por un "Firewall", el cual examina cada mensaje y bloquea aquellos que no cumplen con el criterio de seguridad específico. [WEB27]

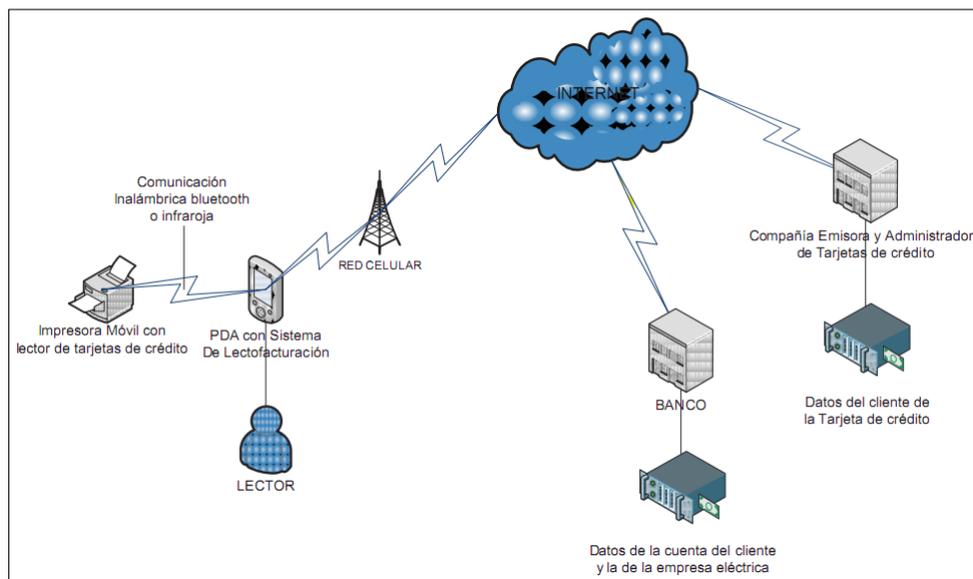


Figura 1. 19: Arquitectura funcional Interdín.

Fuente: Tesis-Solución móvil para facturación y recaudación en un ambiente distribuido.

➤ PAYPAL

Está Basado en un sistema de cuentas de correo electrónico, en el que el usuario se registra previamente. Paypal permite enviar pagos de forma segura a otra persona o empresa que disponga de una dirección de correo electrónico, utilizando para ello su tarjeta de crédito o cuenta bancaria. La red

de Paypal se apoya en la infraestructura financiera existente de cuentas bancarias y tarjetas de crédito, y su proveedor para el procesamiento de pagos es el banco estadounidense Wells Fargo.

El éxito de este servicio se debe a que es totalmente gratuito para sus usuarios, fácil y cómodo, permite el anonimato en las transacciones y es el sistema de pago por Internet más seguro. De hecho, Paypal ha impulsado las garantías de seguridad, a sabiendas de que una de las mayores barreras para el uso de la red en temas de pagos es el fraude cometido con el uso de tarjetas. Así pues, por ejemplo, Paypal se hace responsable de los costes ocasionados por el fraude con las tarjetas.

Combina un sistema de detección de clase mundial contra el fraude con un monitoreo de la cuenta las 24 horas del día para mantener a salvo de gente que intente hackear a las compañías para robar la información personal o financiera.

Como PayPal no intercambia ninguna información confidencial del usuario con las compañías donde se compre, también está a salvo de las amenazas internas dentro de una compañía. PayPal promueve la seguridad al comprar incluso en los sitios Web más pequeños del mundo [WEB30].

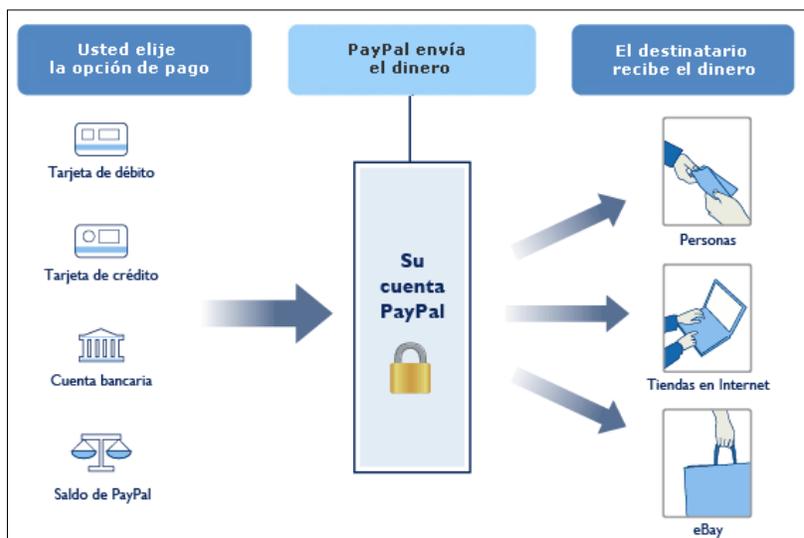


Figura 1. 20: Arquitectura funcional de Paypal.
Fuente: Paypal

1.3. Sistema De Control De Versiones (VCS)

La información fue tomado desde la página web del GIT (Sistema de control de versiones) **WEB23**.

1.3.1. Definición

El control de versiones es un sistema que registra los cambios realizados sobre un archivo o conjunto de archivos a lo largo del tiempo, de modo que puedas recuperar versiones específicas.

Un sistema de control de versiones (Version Control System o VCS en inglés) es una elección muy sabia. Te permite revertir archivos a un estado anterior, revertir el proyecto entero a un estado anterior, comparar cambios a lo largo del tiempo, ver quién modificó por última vez algo que puede estar causando un problema, quién introdujo un error y cuándo, y mucho más. Usar un VCS también significa generalmente que si fastidias o pierdes archivos, puedes recuperarlos fácilmente. Además, obtienes todos estos beneficios a un coste muy bajo [WEB23].

1.3.2. Sistemas de control de versiones locales

El método de control de versiones usado por mucha gente es copiar los archivos a otro directorio (quizás indicando la fecha y hora en que lo hicieron, si son avisados). Este enfoque es muy común porque es muy simple, pero también tremendamente propenso a errores. Es fácil olvidar en qué directorio te encuentras, y guardar accidentalmente en el archivo equivocado o sobrescribir archivos que no querías.

Para hacer frente a este problema, los programadores desarrollaron hace tiempo VCSs locales que contenían una simple base de datos en la que se llevaba registro de todos los cambios realizados sobre los archivos.

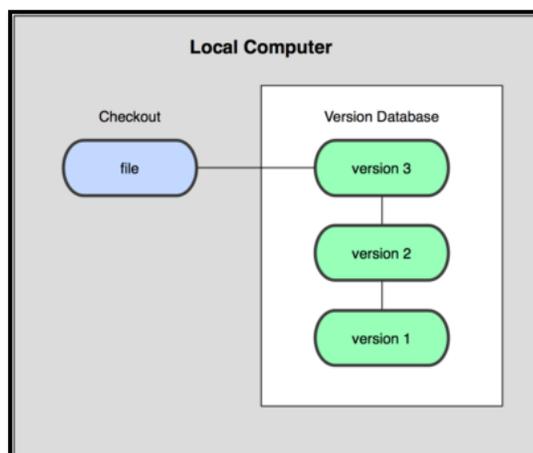


Figura 1. 21: Diagrama de control de versiones local.

Fuente: [WEB23].

Una de las herramientas de control de versiones más popular fue un sistema llamado rcs, que todavía podemos encontrar en muchos de los ordenadores actuales. Hasta el famoso sistema operativo Mac OS X incluye el comando rcs cuando se instala las herramientas de desarrollo. Esta herramienta funciona básicamente guardando conjuntos de parches (es decir, las diferencias entre archivos) de una versión a otra en un formato especial en disco; puede entonces recrear cómo era un archivo en cualquier momento sumando los distintos parches.

1.3.3. Sistemas de control de versiones centralizados

El siguiente gran problema que se encuentra la gente es que necesitan colaborar con desarrolladores en otros sistemas. Para solventar este problema, se desarrollaron los sistemas de control de versiones centralizados (Centralized Version Control Systems o CVCSs). Estos sistemas, como CVS, Subversion, y Perforce, tienen un único servidor que contiene todos los archivos versionados, y varios clientes que descargan los archivos de ese lugar central. Durante muchos años, éste ha sido el estándar para el control de versiones. Ver Figura 11.

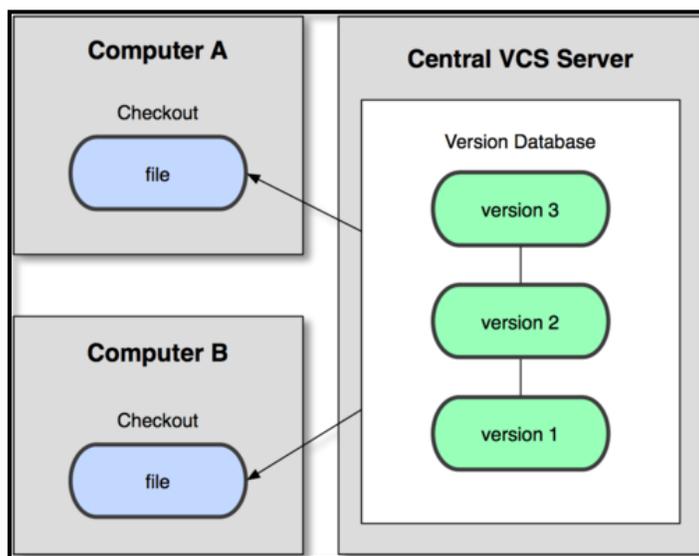


Figura 1. 22: Diagrama de control de versiones centralizado.

Fuente: [WEB23].

Esta configuración ofrece muchas ventajas, especialmente frente a VCSs locales. Por ejemplo, todo el mundo sabe hasta cierto punto en qué está trabajando el resto de gente en el proyecto. Los administradores tienen control detallado de qué puede hacer cada uno; y es mucho más fácil administrar un CVCS que tener que lidiar con bases de datos locales en cada cliente.

Sin embargo, esta configuración también tiene serias desventajas. La más obvia es el punto único de fallo que representa el servidor centralizado. Si ese servidor se cae durante una hora, entonces durante esa hora nadie puede colaborar o guardar cambios versionados de aquello en que están trabajando. Si el disco duro en el que se encuentra la base de datos central se corrompe, y no se han llevado copias de seguridad adecuadamente, pierdes absolutamente todo, toda la historia del proyecto salvo aquellas instantáneas que la gente pueda tener en sus máquinas locales. Los VCSs locales sufren de este mismo problema, cuando tienes toda la historia del proyecto en un único lugar.

1.3.4. Sistemas de control de versiones distribuidos

En un DVCS (como Git, Mercurial, Bazaar o Darcs), los clientes no sólo descargan la última instantánea de los archivos: replican completamente el repositorio. Así, si un servidor muere, y estos sistemas estaban colaborando a través de él, cualquiera de los repositorios de los clientes puede copiarse en el servidor para restaurarlo. Cada vez que se descarga una instantánea, en realidad se hace una copia de seguridad completa de todos los datos. Ver Figura 12.

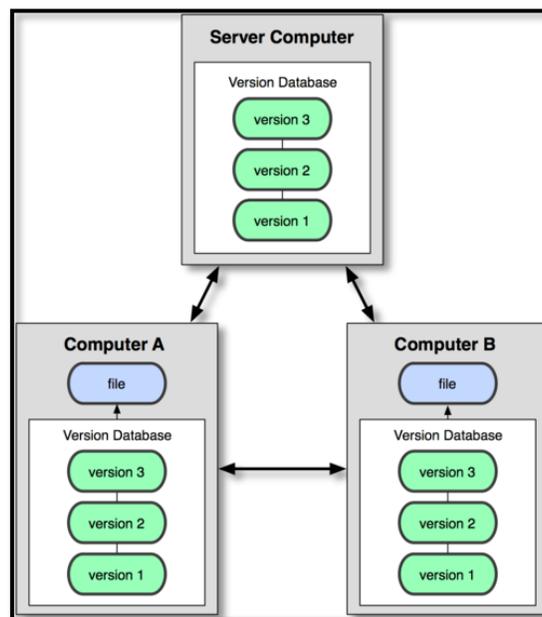


Figura 1. 23: Diagrama de control de versiones distribuido.

Fuente: [WEB23].

Es más, muchos de estos sistemas se las arreglan bastante bien teniendo varios repositorios con los que trabajar, por lo que se puede colaborar con distintos grupos de gente de maneras distintas

simultáneamente dentro del mismo proyecto. Esto permite establecer varios tipos de flujos de trabajo que no son posibles en sistemas centralizados, como pueden ser los modelos jerárquicos.

Para el versionamiento del sistema se tomo como herramienta de a apoyo el controlador de versiones GIT.

1.3.5. Reseña histórica de Git

Git comenzó con un poco de destrucción creativa y encendida polémica. El núcleo de Linux es un proyecto de software de código abierto con un alcance bastante grande.

Durante la mayor parte del mantenimiento del núcleo de Linux (1991-2002), los cambios en el software se pasaron en forma de parches y archivos. En 2002, el proyecto del núcleo de Linux empezó a usar un DVCS propietario llamado BitKeeper.

En 2005, la relación entre la comunidad que desarrollaba el núcleo de Linux y la compañía que desarrollaba BitKeeper se vino abajo, y la herramienta dejó de ser ofrecida gratuitamente.

Esto impulsó a la comunidad de desarrollo de Linux (y en particular a Linus Torvalds, el creador de Linux) a desarrollar su propia herramienta basada en algunas de las lecciones que aprendieron durante el uso de BitKeeper. Algunos de los objetivos del nuevo sistema fueron los siguientes:

- Velocidad
- Diseño sencillo
- Fuerte apoyo al desarrollo no lineal (miles de ramas paralelas)
- Completamente distribuido
- Capaz de manejar grandes proyectos como el núcleo de Linux de manera eficiente (velocidad y tamaño de los datos)

Desde su nacimiento en 2005, Git ha evolucionado para ser fácil de usar y aún así conservar estas cualidades iniciales. Es tremendamente rápido, muy eficiente con grandes proyectos, y tiene un increíble sistema de ramificación (branching) para desarrollo no lineal.

1.3.6. Fundamentos de Git

La principal diferencia entre Git y cualquier otro VCS (Subversion y compañía incluidos) es cómo Git modela sus datos. Conceptualmente, la mayoría de los demás sistemas almacenan la información como

una lista de cambios en los archivos. Estos sistemas (CVS, Subversion, Perforce, Bazaar, etc.) modelan la información que almacenan como un conjunto de archivos y las modificaciones hechas sobre cada uno de ellos a lo largo del tiempo, como se muestra en la siguiente figura [WEB24].

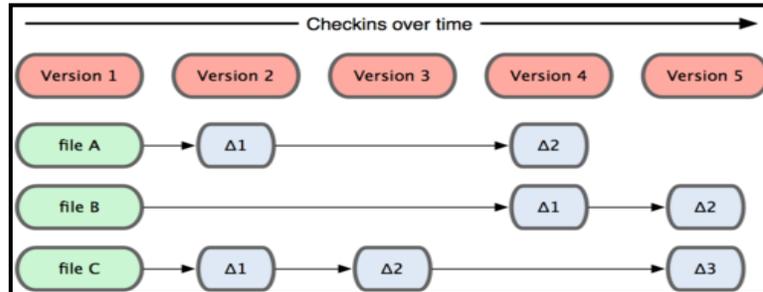


Figura 1. 24: Sistema que tienden a almacenar datos de cambios en cada archivo.

Fuente: [WEB24].

Git no modela ni almacena sus datos de este modo. En cambio, Git modela sus datos más como un conjunto de instantáneas de un mini sistema de archivos.

Cada vez que se confirma un cambio, o guarda el estado de un proyecto en Git, él básicamente hace una foto del aspecto de todos los archivos en ese momento, y guarda una referencia a esa instantánea. Para ser eficiente, si los archivos no se han modificado, Git no almacena el archivo de nuevo, sólo un enlace al archivo anterior idéntico que ya tiene almacenado. Git modela sus datos de la siguiente manera.

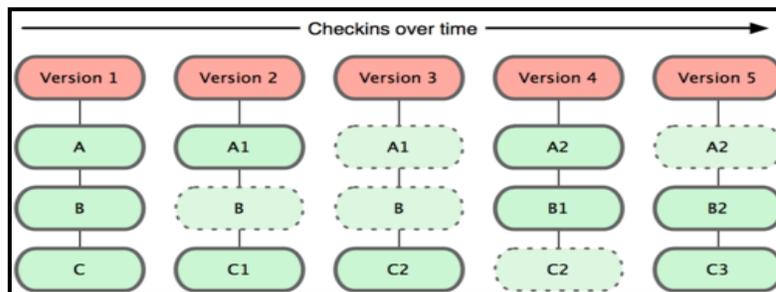


Figura 1. 25: Información de GIT como instantáneas del proyecto a lo largo del tiempo.

Fuente: [WEB24].

Esta es una distinción importante entre Git y prácticamente todos los demás VCSs. Hace que Git reconsidere casi todos los aspectos del control de versiones que muchos de los demás sistemas copiaron de la generación anterior. Git es más como un mini sistema de archivos con algunas herramientas tremendamente potentes construidas sobre él, más que como un VCS.

Operación local en GIT

La mayoría de las operaciones en Git sólo necesitan archivos y recursos locales para operar, por lo general no se necesita información de ningún otro ordenador de la red.

Como se tiene toda la historia del proyecto ahí mismo, en un disco local, la mayoría de las operaciones parecen prácticamente inmediatas.

1.3.7. Integridad de GIT

Todo en Git es verificado mediante una suma de comprobación antes de ser almacenado, y es identificado a partir de ese momento mediante dicha suma (checksum en inglés). Esto significa que es imposible cambiar los contenidos de cualquier archivo o directorio sin que Git lo sepa. Esta funcionalidad está integrada en Git al más bajo nivel y es parte integral de su filosofía.

No se puede perder información durante su transmisión o sufrir corrupción de archivos sin que Git sea capaz de detectarlo.

El mecanismo que usa Git para generar esta suma de comprobación se conoce como hash SHA-1. Se trata de una cadena de 40 caracteres hexadecimales (0-9 y a-f), y se calcula en base a los contenidos del archivo o estructura de directorios en Git. Un hash SHA-1 se lo representa de la siguiente manera.

`24b9da6552252987aa493b52f8696cd6d3b00373`

Git generalmente sólo añade información

Cuando Git realiza acciones en, casi todas ellas sólo añaden información a la base de datos de Git. Es muy difícil conseguir que el sistema haga algo que no se pueda deshacer, o que de algún modo borre información. Como en cualquier VCS, puedes perder o estropear cambios que no has confirmado todavía; pero después de confirmar una instantánea en Git, es muy difícil de perder, especialmente si envías (push) tu base de datos a otro repositorio con regularidad.

1.3.8. Los Tres Estados de GIT

Git tiene tres estados principales en los que se pueden encontrar tus archivos: confirmado (committed), modificado (modified), y preparado (staged).

- ✓ **Confirmado.-** Significa que los datos están almacenados de manera segura en tu base de datos local.
- ✓ **Modificado.-** Significa que has modificado el archivo pero todavía no lo has confirmado a tu base de datos.
- ✓ **Preparado.-** Significa que has marcado un archivo modificado en su versión actual para que vaya en tu próxima confirmación.

Esto nos lleva a las tres secciones principales de un proyecto de Git: el directorio de Git (Git directory), el directorio de trabajo (working directory), y el área de preparación (staging area). Ver Figura 15.

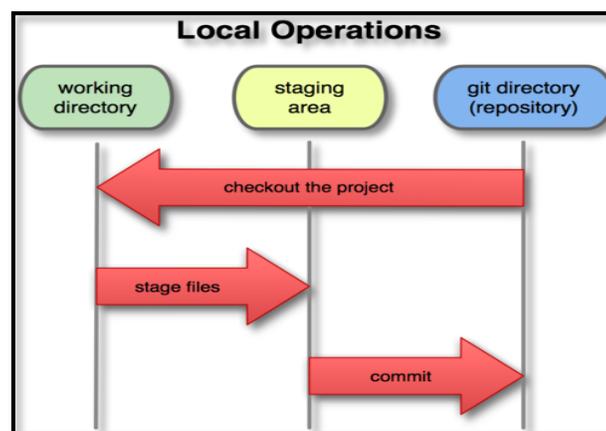


Figura 1. 26: Directorio de trabajo, área de preparación, y directorio de Git.

Fuente: [WEB24].

El directorio de Git es donde Git almacena los metadatos y la base de datos de objetos para un proyecto. Es la parte más importante de Git, y es lo que se copia cuando se clona un repositorio desde otro ordenador.

El directorio de trabajo es una copia de una versión del proyecto. Estos archivos se sacan de la base de datos comprimida en el directorio de Git, y se colocan en disco para que se lo pueda usar o modificar.

El área de preparación es un sencillo archivo, generalmente contenido en tu directorio de Git, que almacena información acerca de lo que va a ir en la próxima confirmación. A veces se denomina el índice, pero se está convirtiendo en estándar el referirse a ello como el área de preparación.

El flujo de trabajo básico en Git lo realiza de la siguiente manera:

1. Modificas una serie de archivos en el directorio de trabajo.
2. Preparas los archivos, añadiendo instantáneas de ellos al área de preparación.

3. Confirmar los cambios, lo que toma los archivos tal y como están en el área de preparación, y almacena esa instantánea de manera permanente en el directorio de Git.

1.4. Core GAD-I

El Core ²⁴ es una aplicación integral que alberga distintos proyectos, con la finalidad de facilitar la administración de los sistemas, también permite manejar las aplicaciones en una sola estructura de proyectos.

La definición de clases que se encuentran en una aplicación integrada hereda o se extiende desde las clases creadas en el Core.

Cuando el usuario accede a la aplicación puede visualizar una interfaz de escritorio con los sistemas a los que tiene acceso.

A continuación se muestra la estructura que maneja el Core.

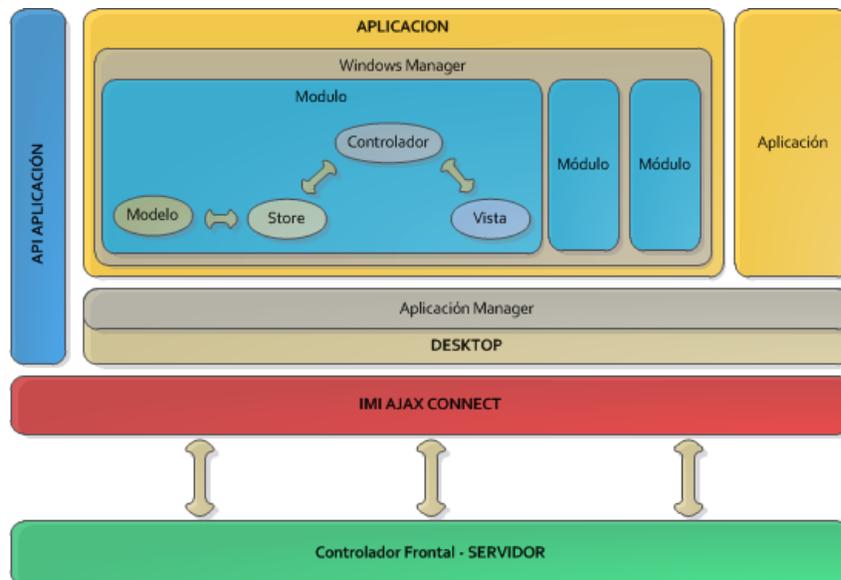


Figura 1. 27: Estructura Core GAD-I.
Fuente: Core GAD-I

²⁴ Core: Aplicación Integral (Reúne varias aplicaciones) del GAD-I

El controlador Frontal es la parte del servidor que interactúa con el API de la Aplicación mediante envío y recepción de peticiones realizadas con AJAX.

El API integra varias aplicaciones, estas aplicaciones están compuestas por módulos y su estructura es modelo, vista controlador permitiendo manejar aplicaciones independientes en la parte del cliente.

1.5. Metodología de Desarrollo GAD-I.

La metodología que se utiliza para la implementación de la presente aplicación ha sido desarrollada para los proyectos que se elaboran dentro de la dirección de TIC del GAD-I. La metodología se encuentra en un proceso de pruebas para ajustar al máximo el desarrollo rápido de un proyecto de software.

Es por esta razón que el sistema se lo ha desarrollado con esta metodología que pretende ser una metodología rápida y confiable para crear proyectos de software eicientes dentro de la dirección de TIC.

1.5.1. Definición

Es una metodología de rápido desarrollo ya que está constituida por la metodología XP y SCRUM, porque se tomaron los procesos más eficientes de cada metodología, con el fin de obtener proyectos más eficientes y en corto tiempo.

El enfoque de desarrollo propuesto constituye una configuración del proceso de la metodología del GAD-I de acuerdo a las características del proyecto, las actividades a realizar y los artefactos (entregables) que serán generados.

1.5.2. Objetivo

Definir y ejecutar las fases del ciclo de vida de desarrollo de software para atender a las áreas usuarias que solicitan requerimientos de soluciones informáticas acordes al plan estratégico TIC o a los requerimientos emergentes del GAD-I y satisfacer sus necesidades de automatización de procesos.

1.5.3. Alcance

Aplica a los requerimientos de desarrollo de software de todas las Direcciones del GAD-I.

1.5.4. Definiciones Y Abreviaturas

ABREVIATURAS		
Nº	TÉRMINO	DEFINICIÓN
1	TIC	Tecnología de la Información y Comunicación
2	GAD-I	Ilustre Municipalidad de Ibarra
3	POA	Plan Operativo Anual
4	BDD	Base de Datos
DEFINICIONES		
Nº	TÉRMINO	DEFINICIÓN
1	HISTORIA DE USUARIO	Documento en el cual se describen de forma corta la funcionalidad de un software, descrita desde la perspectiva del usuario o cliente.
2	ITERACIÓN	Es una repetición de trabajo de duración fija, en la cual se implementan las funcionalidades de un software.
3	MÉTODO X.Y.Z	<p>El método más común para numerar las versiones de un sistema, dependiendo de la importancia de los cambios es el número que se debe cambiar. Cada una de las cifras cambia de acuerdo a:</p> <ul style="list-style-type: none"> • X.Y.Z: Software Nuevo, Cambios muy drásticos en el desarrollo, Reescritura o la incompatibilidad con versiones anteriores. • X.Y.Z: Cuando hay modificaciones en el contenido o la funcionalidad, pero no lo suficientemente importantes como para decir que ya no es el mismo. • X.Y.Z: Cuando se hacen correcciones, añadir o eliminar comentarios, renombrar variables, pero no se ha añadido ni eliminado nada relevante. <p>NOTA: Siempre que se hace un cambio en una cifra de la izquierda, la cifra de la derecha se reinicia en 0.</p>
4	PILA DE ITERACIÓN	Lista de funcionalidades o requerimientos seleccionados para realizar durante una iteración.
5	PILA DE PRODUCTO	Es un inventario o una lista priorizada de requerimientos de usuario que deben incorporarse al producto software a través de las sucesivas iteraciones.

Tabla 1. 3: Definiciones y Abreviaturas

Fuente: Metodología de Desarrollo GAD-I

1.5.5. Documentos De Referencia

DOCUMENTOS INTERNOS	
Nº	Título del Documento
1	Ordenanza que Regula la Estructura y Gestión Organizacional por Procesos de la Ilustre Municipalidad de Ibarra
2	Manual de Normas y Estándares de Programación (http://wiki.imi.gob.ec intranet)

Tabla 1. 4: Documentos Internos

Fuente: Metodología de Desarrollo GAD-I

DOCUMENTOS EXTERNOS	
Nº	Título del Documento
1	Normas de control interno para las entidades, organismos del sector público y de las personas jurídicas de derecho privado que dispongan de recursos públicos.

Tabla 1. 5: Documentos Externos

Fuente: Metodología de Desarrollo GAD-I

1.5.6. Políticas

a) Para ejecutar una petición de desarrollo de sistema, esta deberá constar en el POA de la Dirección de TIC, durante el año se atenderán únicamente estas peticiones con excepción de los sistemas emergentes.

b) Para que un software nuevo se desarrolle; deben estar definidos los procesos y políticas a automatizar en el área usuaria solicitante. En caso de no ser así, la Dirección de TIC la apoyará de manera técnica; pero debe existir un compromiso por parte del solicitante para que antes de iniciar el desarrollo de la aplicación, estos queden establecidos.

c) En caso de no existir el software o hardware necesario para dar solución a los requerimientos, la Dirección de TIC planificará la adquisición del mismo para el año fiscal en curso o bien para el siguiente. En caso de ser de extrema urgencia, la adquisición será autorizada por la máxima autoridad previa justificación técnica documentada.

d) En la definición de los requerimientos deben participar los usuarios de todas las unidades involucradas en el nuevo sistema.

- e) Luego de cada reunión que se realice, se deberá llenar el Acta de Reunión para constancia de lo actuado y de los compromisos adquiridos.
- f) El tiempo de duración de una iteración para obtener un entregable será de 4 semanas.
- g) Los derechos de autor sobre un software y su documentación, creados o desarrollados por los funcionarios de la Dirección de TIC en el ejercicio de sus funciones, así como aquellos desarrollados en contratos con terceros, corresponderán solamente al GAD-I y se deberá analizar la factibilidad de realizar el trámite para registrarlos en el organismo competente.
- h) La Dirección de TIC será el encargado de custodiar el software desarrollado, así como de manejar un inventario de los mismos.
- i) Se deben usar los ambientes de desarrollo y producción de forma independiente el uno del otro. Se considerará implementar el ambiente de pruebas.
- j) Es responsabilidad de la Dirección de TIC elaborar el Manual de Usuario y Manual Técnico de todo el software que se desarrolle al interior del municipio con el objetivo de contar con información para el uso, instalación y configuración del mismo.
- k) La Dirección de TIC considerará el desarrollo de aplicaciones web y/o móviles que automaticen los procesos o trámites orientados al uso de instituciones y ciudadanos en general.
- l) A cada Analista de Sistemas se le designará como su responsabilidad del desarrollo de uno o varios sistemas según la naturaleza y relación con los sistemas a su cargo, sin embargo deberá trabajarse en equipo con los otros Analistas, para conocimiento de la estructura (archivos, programas, campos, rutinas, etc.).
- m) El Analista de Sistemas Informáticos deberá incluir el Manual de Usuario en la opción de ayuda de cada sistema para que esté siempre disponible.
- n) Durante el desarrollo del sistema, se deberán realizar reuniones semanales breves, entre el Responsable de Software y los Analistas, con el objetivo de determinar: los avances (que se hizo), los obstáculos(los problemas) y soluciones, y lo que se va a hacer.

o) Para el desarrollo, los Analistas deberán usar las normas y estándares de programación establecidas en la Dirección de TIC, las mismas que pueden encontrarse en el sitio intranet: <http://wiki.imi.gob.ec>, establecidas en el Manual de Normas y Estándares de Programación.

p) Para el versionado de software se usará el método X.Y.Z (MAYOR.Menor.micro). Para mayor detalle ver **Método X.Y.Z** en el punto “**3 DEFINICIONES Y ABREVIATURAS**”.

A continuación se puede visualizar el diagrama de flujo que presenta la Metodología GAD-I para el desarrollo de Proyectos de Software. **Figura 1.28**

1.5.7. Diagrama De Flujo

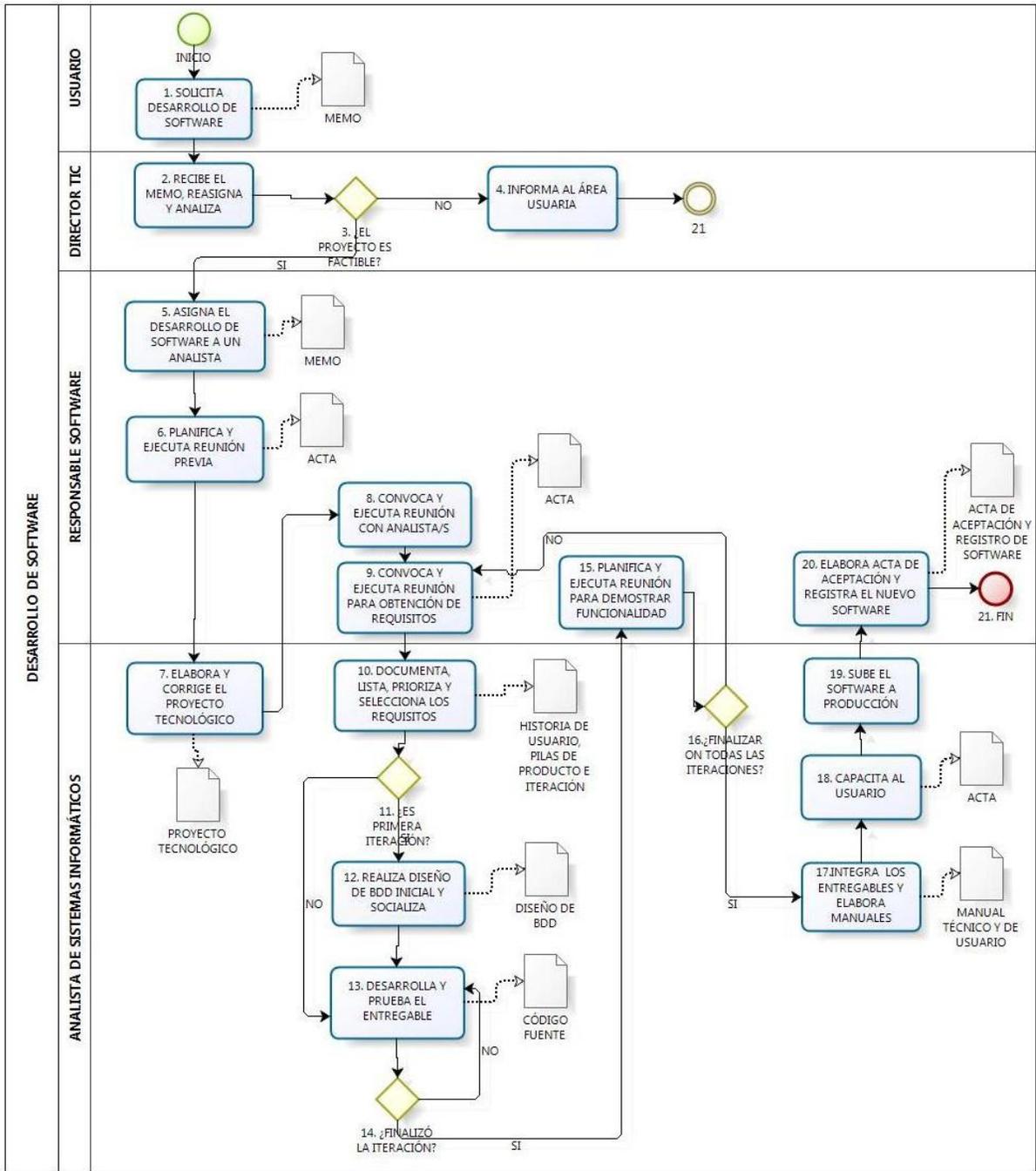


Figura 1. 28: Diagrama de Flujo Desarrollo de Software.

Fuente: Metodología GAD-I

1.5.8. Descripción del Procedimiento

N°	ACTIVIDAD	DESCRIPCIÓN	RESPONSABLE
1	SOLICITA DESARROLLO DE SOFTWARE	Solicita el desarrollo de software, para ello entrega un memorándum a nombre del Director del Área Usuaria solicitante. Lo dirige al Director de TIC.	Director Departamental
2	RECIBE EL MEMO, REASIGNA Y ANALIZA	Recibe el memorándum, reasigna al Responsable de Software y lo analizan conjuntamente. Determinan la factibilidad técnica, operativa y económica de su ejecución.	Director de TIC
3	¿EL PROYECTO ES FACTIBLE?	<ul style="list-style-type: none"> ▪ Si el proyecto no es factible: Va a la actividad N° 4. ▪ Si el proyecto es factible: Continúa en la actividad N° 5. 	Director de TIC
4	INFORMA AL ÁREA USUARIA	Informa al área usuaria solicitante mediante un memorándum los motivos por los cuales no es factible el desarrollo del software.	Director de TIC
5	ASIGNA EL DESARROLLO DE SOFTWARE A UN ANALISTA	Asigna el proyecto al Plan Operativo Anual de TIC y a su respectivo Analista de Sistemas de acuerdo al cronograma de actividades. En proyectos emergentes se asignará directamente al responsable. Comunica al analista a través de memorándum. Crea el proyecto en la herramienta de gestión de proyectos (en caso de que aplique).	Responsable de Software
6	PLANIFICA Y EJECUTA REUNIÓN PREVIA	Planifica y ejecuta una reunión al Analista y al área solicitante para establecer objetivos y alcance del nuevo sistema, registra el Acta en el módulo de Reuniones. Además, se solicita y recepta toda la información necesaria como Base Legal, Flujo del Proceso, Formatos y Tipos de Documentos de Entrada y Salida, entre otros.	Responsable de Software
7	ELABORA Y CORRIGE EL PROYECTO TECNOLÓGICO	Elabora el proyecto en base al <i>Formato de Proyecto Tecnológico (FOR-3.9.3.1)</i> establecido en el Procedimiento <i>Gestión de Proyectos Tecnológicos (PRO-3.9.3)</i> . Socializa el proyecto con el Responsable de Software, Director de TIC y demás Analistas. Realiza los cambios requeridos en caso de que estos existan.	Analista de Sistemas Informáticos

8	CONVOCA Y EJECUTA REUNIÓN CON ANALISTAS	Convoca a una reunión de inducción al proyecto a los Analistas que participan directa o indirectamente en él. Se registra el Acta de Reunión en el Módulo de Reuniones.	Responsable Software
9	CONVOCA Y EJECUTA REUNIÓN PARA OBTENCIÓN DE REQUISITOS	Convoca a una reunión al Área Usuaría y Analista/s para determinar los requisitos del software. Registra el Acta de Reunión en el Módulo de Reuniones. Para apoyo durante la obtención de requisitos, se puede elaborar una lista de preguntas. <i>NOTA: En caso de ser necesarias más reuniones, se realizarán con el objetivo de obtener la mayoría de los requisitos para poder hacer el diseño inicial de la BDD.</i>	Responsable Software
10	DOCUMENTA, LISTA, PRIORIZA Y SELECCIONA LOS REQUISITOS	Documenta, con la participación del usuario, los requisitos obtenidos durante la Reunión de obtención de Requisitos en el Formato de Historia de Usuario. <ul style="list-style-type: none"> ▪ Ver FOR-3.9.4.1 Elabora la pila del producto y la pila de iteración de acuerdo a las prioridades del sistema. <ul style="list-style-type: none"> ▪ Ver FOR-3.9.4.2 y FOR-3.9.4.3 	Analista de Sistemas Informáticos
11	¿ES PRIMERA ITERACIÓN?	Si es la primera iteración que se está realizando ir a la actividad N° 12. Caso contrario ir a la actividad N° 13.	Analista de Sistemas Informáticos
12	REALIZA DISEÑO INICIAL DE BDD Y SOCIALIZA	Realiza el diseño inicial de la base de datos analizando los requerimientos obtenidos, y la reutilización de tablas existentes en el ambiente de producción. Socializa con los analistas de sistemas. Realiza las correcciones necesarias al diseño.	Analista de Sistemas Informáticos
13	DESARROLLA Y PRUEBA EL ENTREGABLE	Diseña, desarrolla y prueba el entregable de acuerdo a los requisitos de la Pila de Iteración. Coordina constantemente reuniones con el Área Usuaría para aclarar dudas acerca de los requerimientos del software. <i>NOTA: Durante este paso, puede realizar correcciones al diseño de la base de datos.</i>	Analista de Sistemas Informáticos
14	¿FINALIZÓ LA ITERACIÓN?	Si aún no finaliza la iteración, va al paso N° 13. Si finalizó la iteración va al paso N° 15.	Analista de Sistemas Informáticos
15	PLANIFICA Y	Planifica una reunión con el Área Usuaría y	Responsable de

	EJECUTA REUNIÓN PARA DEMOSTRAR FUNCIONALIDAD	el Analista de Sistemas para demostrar la funcionalidad realizada durante la iteración. Se registra el Acta de Reunión en el Módulo de Reuniones.	Software
16	¿FINALIZARON LAS ITERACIONES?	Si no finalizaron las iteraciones, va al paso N° 9. Si finalizaron las iteraciones, va al paso N° 17. NOTA: <i>Las iteraciones finalizan cuando ya no existen historias de usuario por desarrollar.</i>	Responsable de Software
17	INTEGRA LOS ENTREGABLES Y ELABORA MANUALES	Integra los entregables realizados durante todo el proceso. Realiza ajustes y configuraciones necesarias. Elabora el Manual Técnico y de Usuario. ▪ Ver FOR-3.9.4.4 y FOR-3.9.4.5	Analista de Sistemas Informáticos
18	CAPACITA AL USUARIO	Convoca al Área Usuaría y la capacita en el uso del nuevo software. Se registra el Acta de Reunión en el Módulo de Reuniones.	Analista de Sistemas Informáticos
19	SUBE EL SOFTWARE A PRODUCCIÓN	Sube el software realizado al entorno de producción para su uso real por el área usuaria. Crea las tablas y realiza las configuraciones necesarias en el servidor.	Analista de Sistemas Informáticos
20	ELABORA ACTA DE ACEPTACIÓN Y REGISTRA EL NUEVO SOFTWARE	Elabora el Acta de Término de Proyectos o Requerimientos de acuerdo al formato. Entrega al usuario a través de memorándum para que la firme. ▪ Ver FOR-3.9.4.6 Registra el nuevo software en la Caracterización de Software. Ver FOR-3.9.4.8	Responsable de Software
21	FIN		

Tabla 1. 6: Descripción del Procedimiento

Fuente: Metodología de Desarrollo GAD-I

2. Capítulo II

Procesos Sismert

En el presente capítulo se explica la definición de un proceso y su posterior desarrollo.

2.1. Definición de Proceso.

Un proceso es un conjunto de actividades o eventos (coordinados u organizados) que se realizan o suceden (alternativa o simultáneamente) bajo ciertas circunstancias con un fin determinado. Este término tiene significados diferentes según la rama de la ciencia o la técnica en que se utilice.

Al proceso también se lo puede definir como la añadidura y valoración de documentación escrita en toda causa, que sirve para entender y esclarecer los hechos.

Hay aspectos que tienen en común todos los procesos. Estos son: Nombre, Entradas, Salidas, Actividades, Tareas, Encargado.

Los aspectos de un proceso son muy relevantes al momento de su creación, por lo que se lo puedo identificar fácilmente cuando existe un conjunto relativamente grande de procesos, los mismos que contienen un conjunto de actividades relacionadas y secuenciales que convierte unos factores iniciales (inputs) en bienes o servicios deseados (outputs), añadiendo un valor a los mismos.[WEB26]

2.1.1. Importancia de los procesos

Los procesos constituyen una parte importante de los costos de la organización, ya que permiten dar más efectividad en cada una de las actividades que realiza la Institución, mejorando el desempeño en el mercado y dentro de la misma Institución.

Se puede mencionar entre los más importantes los siguientes puntos en los que un proceso puede mejorar el rendimiento interno y externo de una Institución.

- Permite a la Organización o Institución centrarse en el cliente.

- Permite predecir y controlar el cambio.
- Aumentar la capacidad de la empresa para competir.
- Mantiene a la organización centrada en el proceso.
- Previene errores y da soluciones.
- Desarrolla un sistema de evaluación en las áreas de la empresa.

A continuación se describe los procesos que se consideran importantes para la realización de la aplicación, los mismos que ayudaran al desarrollo del Sistema de Control de Parqueo Tarifado SISMERT.

2.2. Procedimiento De Venta De Tickets Y Emisión De Notificaciones

2.2.1. Objetivo

Proveer y controlar el uso de tarjetas pre-pago a los usuarios del SISMERT con la finalidad evitar el abuso del uso de la vía pública, para de esta manera garantizar espacios de estacionamiento a la mayor cantidad de personas y reducir el congestionamiento vehicular en la zona central y comercial de la ciudad.

2.2.2. Alcance

De acuerdo a la demanda del año anterior, se prevé la compra de las especies valoradas, las mismas que las elabora el Instituto Geográfico Militar, estas especies son registradas por la Tesorería Municipal, el responsable del SISMERT solicita a Tesorería una cantidad suficiente para proveer a los Supervisores del SISMERT, quienes a su vez se encargan de repartir a los Inspectores a su cargo, al final del día los inspectores reportan los valores recaudados por venta de tickets a sus Supervisores.

Los Supervisores entregan los valores al Asistente Operativo del SISMERT quien elabora el cuadro diario para la supervisión del Responsable del SISMERT, el Asistente Operativo del SISMERT entrega los valores a Tesorería, se descarga la cantidad de tarjetas correspondientes al valor entregado y finalmente tesorería ingresa los valores a las arcas municipales.

2.2.3. Roles y Responsabilidades

ROLES Y RESPONSABILIDADES	
Rol	Responsabilidad
Supervisor de operaciones del SISMERT	Prever la cantidad de tickets que se venderán el año siguiente, Solicitar tickets a tesorería, organizar y supervisar la distribución de tickets.
Supervisor del SISMERT	Supervisar y controlar la venta de tickets a los inspectores del SISMERT, reportar la venta diaria al Asistente Operativo del SISMERT.
Inspectores del SISMERT	Vender tickets de estacionamiento, controlar el uso de tarjetas y elaborar las multas a los vehículos infractores conforme lo estipula la ordenanza, reportar la venta diaria a los Supervisores.
Asistente Operativo del SISMERT	Recaudar los valores diarios de los Supervisores, elaborar el cuadro diario y entregar el dinero a Tesorería
Tesorería	Proveer de tarjetas al SISMERT y recolectar los dineros obtenidos por la venta de tickets

Tabla 2. 1: Roles y Responsabilidades Venta Tickets, emisión Notificaciones

Fuente: Procesos TIC del GAD-I 2012

2.2.4. Términos y Definiciones

ABREVIATURAS		
Nº	TÉRMINO	DEFINICIÓN
1	GAD-I	

		Ilustre Municipalidad de Ibarra
2	SISMERT	Sistema Municipal de Estacionamiento Tarifado
DEFINICIONES		
N°	TÉRMINO	DEFINICIÓN
1	SISTEMA INFORMÁTICO	Es un conjunto de elementos que interactúan entre sí con el fin de apoyar las actividades de una empresa o negocio. Como elementos importantes está el equipo humano el cual está formado por las personas que utilizan el sistema y el hardware, necesario para que el sistema de información pueda operar. El sistema realiza cuatro actividades básicas: entrada, almacenamiento, procesamiento y salida de información.
2	SUPERVISOR	Es el coordinador y el encargado de controlar el trabajo de los inspectores de ruta.
3	INSPECTOR	Es el que realiza las inspecciones y controles de los estacionamientos tarifados.

Tabla 2. 2: Términos y Definiciones Venta Tickets, emisión Notificaciones

Fuente: Procesos TIC del GAD-I 2012

2.2.5. Diagrama De Flujo (Ver figura 2.1)

A continuación se muestra el flujo que se debe seguir para la realización y buen uso de este procedimiento.

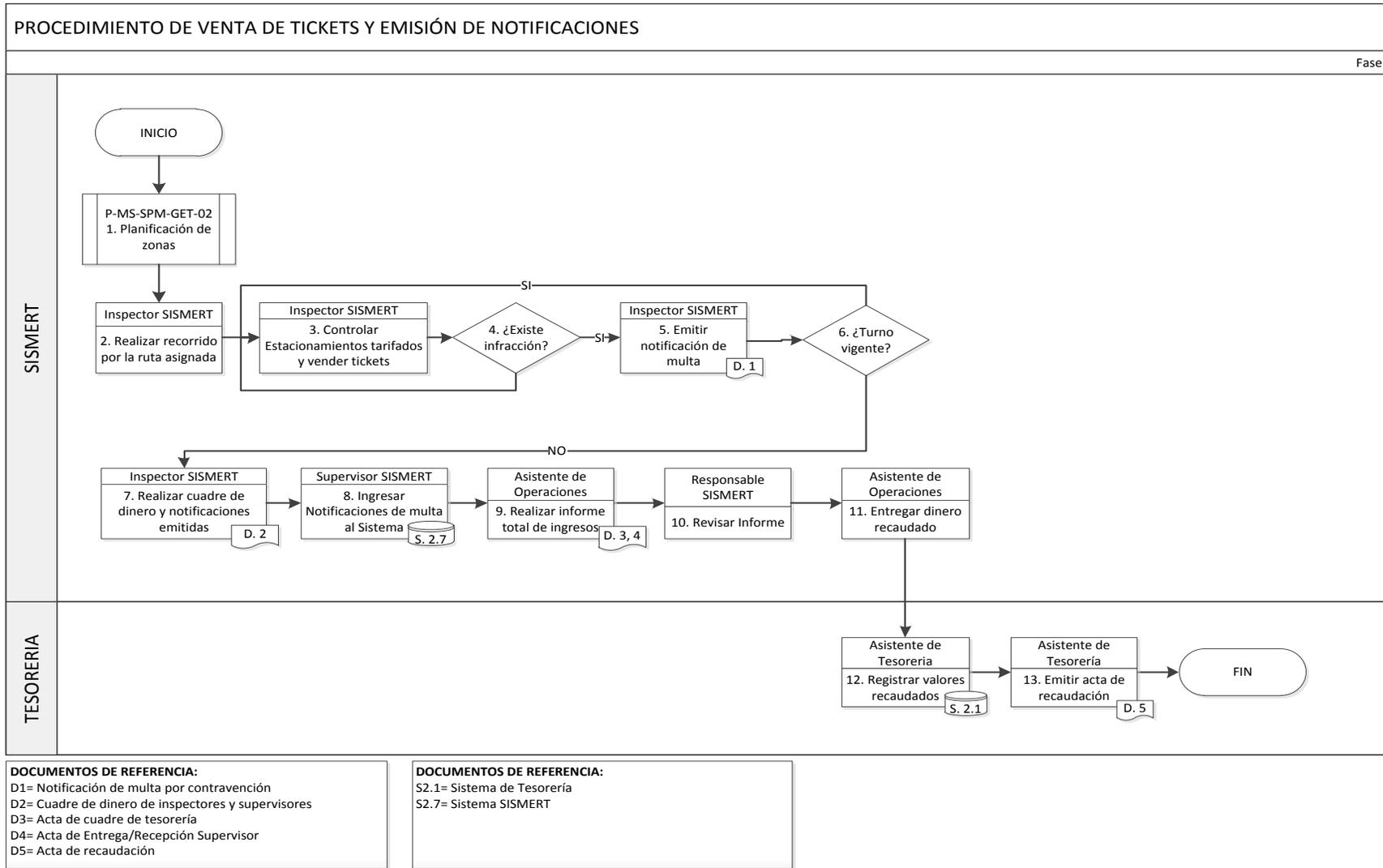


Figura 2. 1: Venta de Tickets y emisión Notificaciones.
 Fuente: Procesos TIC del GAD-I 2012

2.2.6. Descripción Del Procedimiento

N°	ACTIVIDAD	DESCRIPCIÓN	RESPONSABLE
1	PROCEDIMIENTO DE PLANIFICACIÓN DE ZONAS	Se realiza el procedimiento de Planificar zonas de estacionamiento tarifado. Planificación de zonas	Responsable SISMERT
2	REALIZAR RECORRIDO POR LA RUTA ASIGNADA	Realiza el recorrido por las rutas indicadas verificando si existe infracciones	Inspector del SISMERT
3	CONTROLAR ESTACIONAMIENTOS TARIFADOS Y VENDER TICKETS	Realiza el control de los estacionamientos verificando si existe infracciones, además vende tickets cuando el ciudadano lo solicita.	Inspector del SISMERT
4	¿EXISTE INFRACCIÓN?	<ul style="list-style-type: none"> • Si existe infracción: Ir actividad 5 • Si no existe infracción: Ir actividad 3 	Inspector del SISMERT
5	EMITIR NOTIFICACIÓN DE MULTA	Se emite la notificación de multa por la infracción identificada.	Inspector del SISMERT
6	¿TURNO VIGENTE?	<ul style="list-style-type: none"> • Si turno de trabajo está vigente: Ir actividad 3 • Si turno de trabajo no está vigente: Ir actividad 7 	Inspector del SISMERT
7	REALIZAR CUADRE DE DINERO Y NOTIFICACIONES EMITIDAS	Se realiza la entrega del dinero recaudado y se realiza el cuadro diario respectivo,	Inspector del SISMERT

		y se entrega las notificaciones emitidas	
8	INGRESAR NOTIFICACIONES DE MULTA AL SISTEMA	Las notificaciones emitidas son ingresadas al sistema de SISMERT	Supervisor del SISMERT
9	REALIZAR INFORME TOTAL DE INGRESOS	Realiza el informe global del dinero recaudado por cada inspector y envía al responsable del SISMERT para su aprobación.	Asistente de operaciones
10	REVISAR INFORME	Revisa y aprueba el informe.	Responsable de SISMERT
11	ENTREGAR DINERO RECAUDADO	Realiza la entrega del dinero recaudado a Tesorería y firma el acta de entrega/recepción	Responsable de SISMERT
12	REGISTRAR VALORES RECAUDADOS	Se registra los valores recaudados por SISMERT en el sistema de recaudación y tesorería.	Asistente de Tesorería
13	EMITIR ACTA DE RECAUDACIÓN	Se emite el acta de recaudación de entrega/recepción del dinero recaudado.	Asistente de Tesorería
14	FIN		

Tabla 2. 3: Descripción del Procedimiento Venta Tickets, emisión Notificaciones

Fuente: Procesos TIC del GAD-I 2012

2.2.7. Documentos De Referencia

DOCUMENTOS INTERNOS	
N°	Título del Documento
1	Ordenanza de creación, organización y establecimiento del Sistema Municipal de Estacionamiento Regulado para el Cantón Ibarra
DOCUMENTOS EXTERNOS	
N°	Título del Documento
1	Código orgánico de Organización Territorial, autonomía y descentralización COOTAD

Tabla 2. 4: Documentos de Referencia Internos-Externos Venta Tickets, emisión Notificaciones

Fuente: Procesos TIC del GAD-I 2012

2.3. Procedimiento De Planificar Las Zonas De Estacionamiento Regulado

2.3.1. Objetivo

Determinar las rutas que controlaran los Inspectores y supervisores del SISMERT en la zona regulada (zona azul) con la finalidad de optimizar el servicio y los tiempos de respuesta a los usuarios del SISMERT y optimizar los recursos humanos, materiales y económicos.

2.3.2. Alcance

Para planificar las rutas se determina la demanda de estacionamientos en la zona regulada, el horario de trabajo, el número de inspectores disponibles, para luego proceder con la delimitación de las rutas de trabajo para los inspectores en dos turnos, el primer turno con todo el grupo de trabajo, y el segundo en horario de almuerzo y horas de menor demanda con la mitad del grupo de trabajo, se elaboran cuadros de trabajo asignados a los seis supervisores, quienes son los encargados de la recaudación diaria de valores y finaliza con la elaboración semanal de horarios de trabajo para inspectores y supervisores, asignando rutas diarias a cada inspector.

2.3.3. Roles y Responsabilidades

ROLES Y RESPONSABILIDADES	
Rol	Responsabilidad
Supervisor de operaciones del SISMERT	Prever la cantidad de tickets que se venderán el año siguiente, Solicitar tickets a tesorería, organizar y supervisar la distribución de tickets.
Supervisor del SISMERT	Supervisar y controlar la venta de tickets a los inspectores del SISMERT, reportar la venta diaria al Asistente Operativo del SISMERT.
Inspectores del SISMERT	Vender tickets de estacionamiento, controlar el uso de tarjetas y elaborar las multas a los vehículos infractores conforme lo estipula la ordenanza, reportar la venta diaria a los Supervisores.
Tesorería	Proveer de tarjetas al SISMERT y recolectar los dineros obtenidos por la venta de tickets

Tabla 2. 5: Roles y Responsabilidades Zonas Parqueo

Fuente: Procesos TIC del GAD-I 2012

2.3.4. Términos Y Definiciones

ABREVIATURAS		
N°	TÉRMINO	DEFINICIÓN
1	GAD-I	Ilustre Municipalidad de Ibarra
2	SISMERT	Sistema Municipal de Estacionamiento Tarifado
DEFINICIONES		
N°	TÉRMINO	DEFINICIÓN
1	SISTEMA INFORMÁTICO	Es un conjunto de elementos que interactúan entre sí con el fin de apoyar las actividades de una empresa o negocio. Como elementos importantes está el equipo humano el cual está formado por las personas que utilizan el sistema y el hardware, necesario para que el sistema de información pueda operar. El sistema realiza cuatro actividades básicas: entrada, almacenamiento, procesamiento y salida de información.
2	SUPERVISOR	Es el coordinador y el encargado de controlar el trabajo de los inspectores de ruta.
3	INSPECTOR	Es el que realiza las inspecciones y controles de los estacionamientos tarifados.

Tabla 2. 6: Términos y Definiciones Zonas Parqueo

Fuente: Procesos TIC del GAD-I 2012

2.3.5. Procedimiento

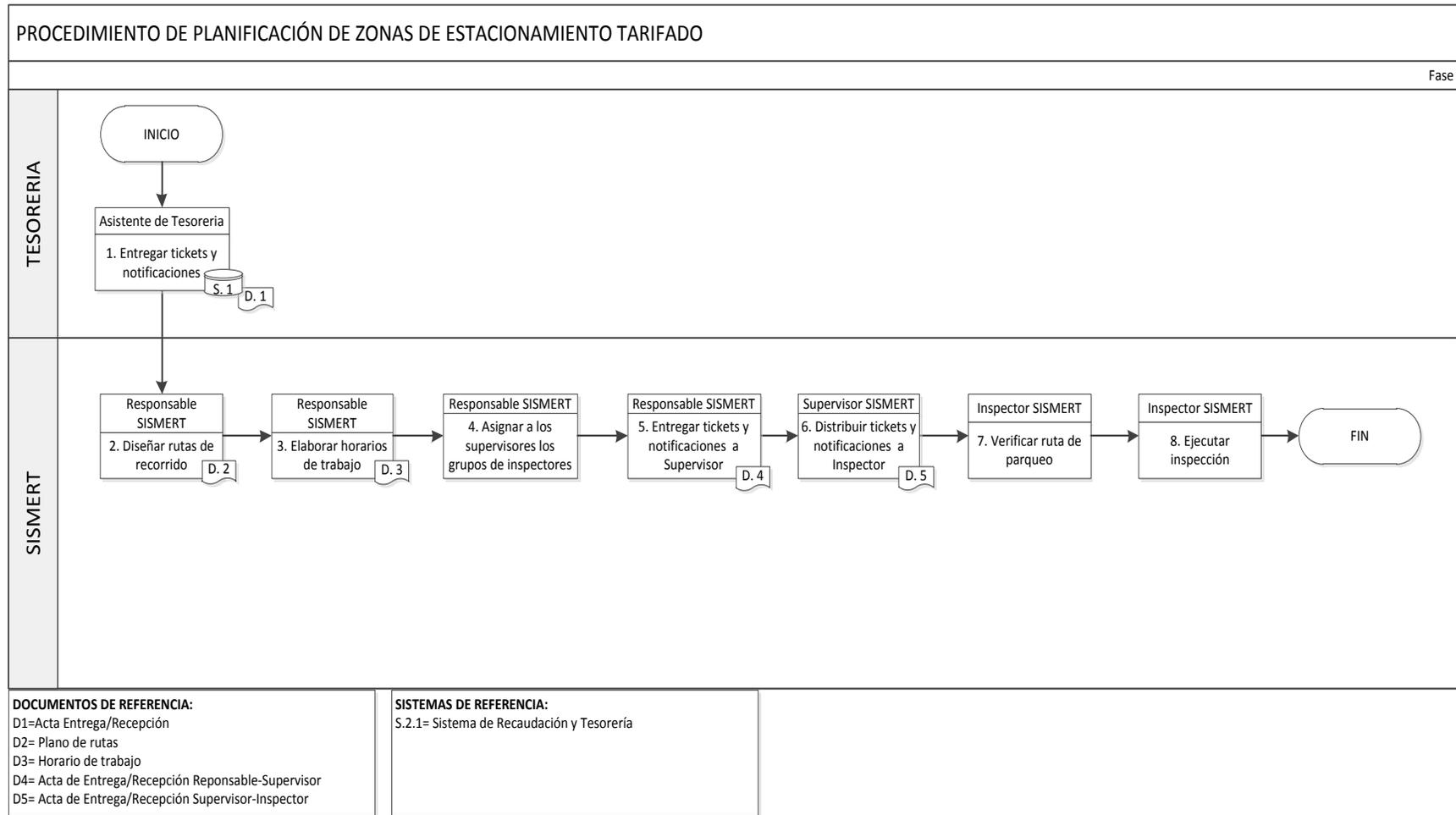


Figura 2. 2: Descripción Procedimiento Zonas Parqueo.
Fuente: Procesos TIC del GAD-I 2012

2.3.6. Descripción Del Procedimiento

N°	ACTIVIDAD	DESCRIPCIÓN	RESPONSABLE
1	ENTREGAR TICKETS Y NOTIFICACIONES	Tesorería entrega tickets y notificaciones al responsable de SISMERT y registra en el sistema de Recaudación y Tesorería	Asistente de Tesorería
2	DISEÑAR RUTAS DE RECORRIDO	Trazar las rutas en dos planos, las mismas que deberá recorrer cada inspector dependiendo del horario.	Responsable del SISMERT – Supervisores
3	ELABORAR HORARIOS DE TRABAJO	Asignar a cada supervisor una cantidad proporcionalmente igual de inspectores, para su control, dependiendo del número de supervisores e inspectores.	Responsable del SISMERT – Supervisores
4	ASIGNAR A LOS SUPERVISORES LOS GRUPOS DE INSPECTORES	Asignar un grupo de inspectores a cada supervisor para realizar los recorridos.	Responsable del SISMERT – Supervisores
5	ENTREGAR TICKETS Y NOTIFICACIONES A SUPERVISOR	Se realiza la distribución de tickets y notificaciones a los supervisores y se hace firmar en el acta de entrega/recepción.	Responsable del SISMERT – Supervisores
6	DISTRIBUIR TICKETS Y NOTIFICACIONES A INSPECTOR	Se realiza la distribución de tickets y notificaciones a los inspectores y se hace firmar en el acta de entrega/recepción.	Supervisores del SISMERT
7	VERIFICAR RUTA DE PARQUEO	El inspector realiza la verificación de la ruta de parqueo asignada para realizar la inspección.	Inspectores del SISMERT
8	EJECUTAR INSPECCIÓN	Aplicar lo que establece la ordenanza de creación del SISMERT,	Inspectores del SISMERT

		todo vehículo que se estacione en la zona azul más de 5 minutos, debe usar el documento habilitante (ticket), emitir la notificación de multa en case de haber cometido alguna infracción a la ordenanza.	
9	FIN		

Tabla 2. 7: Descripción del Procedimiento Zonas Parqueo

Fuente: Procesos TIC del GAD-I 2012

2.3.7. Documentos De Referencia

DOCUMENTOS INTERNOS	
N°	Título del Documento
1	Ordenanza de creación, organización y establecimiento del Sistema Municipal de Estacionamiento Regulado para el Cantón Ibarra
DOCUMENTOS EXTERNOS	
N°	Título del Documento
1	Código orgánico de Organización Territorial, autonomía y descentralización COOTAD

Tabla 2. 8: Documentos Referencia Internos-Externos (Zonas Parqueo)

Fuente: Procesos TIC del GAD-I 2012

2.4. Procedimiento De Venta De Tickets Electrónicos A Través De Un Sms

2.4.1. Objetivo

Proveer tickets electrónicos a los usuarios del SISMERT enviando un SMS para su compra, evitar la demora en realizar la compra de una tarjeta prepago cuando el inspector no se encuentra en el monto que el usuario desea usar una zona regulada de SISMERT.

Ayudando a evitar el congestionamiento vehicular en las principales calles del centro de la ciudad.

2.4.2. Alcance

De acuerdo a la demanda del uso del parqueadero SISMERT es indispensable ayudar a los inspectores a controlar el Sistema de estacionamiento con la implementación de una aplicación que pueda ser fácil de usar y agilizar el proceso de control del parqueo.

Las Compras de tarjetas electrónicas es a través de recargas de saldo en un número de teléfono registrado en el sistema, el mismo que se va debitando cada vez que el usuario envía un SMS para hacer uso del parqueadero, este mensaje es para la compra del ticket electrónico, remplazando la compra de una tarjeta prepago, el costo por el uso del parqueadero es el mismo que está establecido en la ordenanza Municipal.

El saldo se lo puede recargar acercándose al GAD-I y también lo podrán comprar a los mismos inspectores, si así se lo considere pertinente y seguro el proceso de esta transacción desde un inspector, para luego sus recaudaciones ser entregadas a tesorería.

2.4.3. Roles y Responsabilidades

ROLES Y RESPONSABILIDADES	
Rol	Responsabilidad
Inspectores del SISMERT	Realizar recargas de saldo en un número de teléfono registrado en el sistema, controlar el uso del parqueadero con la aplicación.
Tesorería	Recolectar los dineros obtenidos por las recargas de Saldo.
Ventanilla	Realizar recargas de saldo en un número de teléfono registrado en el sistema.

Tabla 2. 9: Roles y Responsabilidades Venta Tickets Electrónicos

Fuente: Procesos TIC del GAD-I 2012

2.4.4. Términos y Definiciones

ABREVIATURAS		
N°	TÉRMINO	DEFINICIÓN
1	GAD-I	Ilustre Municipalidad de Ibarra
2	SISMERT	Sistema Municipal de Estacionamiento Tarifado
DEFINICIONES		
N°	TÉRMINO	DEFINICIÓN
1	SISTEMA INFORMÁTICO	Es un conjunto de elementos que interactúan entre sí con el fin de apoyar las actividades de una empresa o negocio. Como elementos importantes está el equipo humano el cual está formado por las personas que utilizan el sistema y el hardware, necesario para que el sistema de información pueda operar. El sistema realiza cuatro actividades básicas: entrada, almacenamiento, procesamiento y salida de información.
2	SUPERVISOR	Es el coordinador y el encargado de controlar el trabajo de los inspectores de ruta.
3	INSPECTOR	Es el que realiza las inspecciones y controles de los estacionamientos tarifados.
4	VENTANILLA	Recaudador de las recargas de saldo SISMERT.

Tabla 2. 10: Términos y Definiciones Tickets Electrónico

Fuente: Procesos TIC del GAD-I 2012

La información obtenida en los procesos será útil para continuar con el siguiente capítulo correspondiente al desarrollo de la aplicación, tomando los procedimientos más relevantes de acuerdo a las características planteadas para el desarrollo del sistema de control de parqueo tarifado SISMERT.

La información presentada fue referenciada al proyecto de levantamientos de procesos del GAD-I, el cual consistió en elaborar y documentar cada uno de los procesos de esta Institución, y de este bloque de información se ha tomado los procesos y procedimientos que corresponde al área de SISMERT.

A continuación se muestra la relación que tiene con el Macro proceso de los Servicios Públicos que el GAD-I presta a la Ciudadanía como el uso del Parquero Tarifado.

PROYECTO: LEVANTAMIENTO DE PROCESOS GAD-I				
MACROPROCESO:	PRESTACIÓN DE SERVICIOS PÚBLICOS MUNICIPALES			
PROCESO:	GESTIÓN DE ESTACIONAMIENTO TARIFADO	TIPO:	Sustantivo	
PROPÓSITO:	Registrar y controlar el estacionamiento tarifado vehicular y ciclístico en el área urbana de la ciudad de Ibarra, para mejorar la circulación peatonal, vehicular y ciclística.			
PROVEEDORES	DISPARADORES	ACTIVIDADES	PRODUCTOS/SERVICIOS	CLIENTE
PI: Planeación Cantonal	Ordenanza de creación, organización y establecimiento del SISMERT	Planificar las zonas de estacionamiento regulado.	Estacionamientos rotativos	CE: Ciudadanía
	Solicitud de informes diarios	Asignación de grupos, supervisores e Inspectores y rutas de trabajo	Informe diario	
		Gestión de ventas de tickets		
		Control de estacionamientos tarifados		
		Gestión de parquímetros		
EJECUTORES	RESPONSABLE DEL PROCESO	ALCANCE DEL PROCESO	PUNTOS DE CONTROL	REQUERIMIENTOS DEL CLIENTE
Unidad Municipal del Sistema de estacionamiento Rotativo tarifado	Supervisor de Operaciones del SISMERT	Desde la señalización de zonas tarifadas, formación de grupos para el control y dando el servicio de estacionamiento tarifado a la ciudad.		Calidad del servicio

Tabla 2. 11: Gestión de Estacionamiento Tarifado.

Fuente: Proyecto Levantamiento de Procesos GAD-I 2012

3. Capítulo III

Diseño y Desarrollo de Software.

El presente capítulo se desarrolla a través de la metodología para desarrollo de Software establecida por la Dirección de TIC del GAD-I, que es una combinación de las metodologías ágiles XP y SCRUM. El proceso se lo realizara siguiendo el Diagrama de Flujo indicado en el Capítulo II, que inicia cuando el usuario solicita la elaboración de un proyecto, y pasa por la Dirección de TIC.

Una vez aprobado el proyecto se asigna a un analista de sistemas, quien se encarga de elaborar el Proyecto Tecnológico, este será expuesto ante el grupo de analistas de la dirección TIC, para continuar con la obtención de requisitos, y documentación de los mismos (Historias de Usuario, Pilas de Producto, Pilas de Iteración), y así proceder con el desarrollo de cada iteración.

Al final de cada iteración se planifica y ejecuta una reunión para demostrar su funcionalidad, una vez terminadas las iteraciones, se procede a elaborar el Manual Técnico y Manual de Usuario, a capacitar al usuario y finalmente subir el Software a Producción elaborando la respectiva Acta de Aceptación y registro de este nuevo Software.

Sistema De Control Del Parqueo Tarifado (SISMERT)

“SCOPT GAD-I”

Luis Rodolfo Guamán Guamán

Versión 1.0

3.1. Roles.

Se clasifica a todas las personas que intervienen o tienen interés en el desarrollo del proyecto tecnológico en: Usuario, Director TIC, Responsable de Software, Analista de Sistemas Informáticos, Programador.

A continuación se describen los roles que intervienen en la Aplicación:

Nombre	Descripción	Responsabilidad
Usuario (User)	Una persona o grupo de personas que tiene un sólido conocimiento del proceso del negocio actual y clarificará los problemas que el nuevo sistema ha de resolver.	<ul style="list-style-type: none">• Solicitar desarrollo de Software.• Definir la metáfora del sistema.• Definir el valor de negocio.• Ayudar a crear las historias de usuario.• Escribir o especificar las pruebas de aceptación.• Realizar conjuntamente con el equipo de desarrollo la planificación de las entregas.
Programador (Programmer)	Responsable del desarrollo del proyecto.	<ul style="list-style-type: none">• Estimar historias de usuario.• Realizar prototipos.• Escribir código.• Diseñar.• Participar en reuniones para

Nombre	Descripción	Responsabilidad
		la planificación y demostración de funcionalidad. <ul style="list-style-type: none"> • Escribir pruebas unitarias. • Integrar los Entregables y elaborar Manuales. • Capacitar al Usuario
Director TIC	Encargado de la gestión del proyecto internamente y hacia los clientes.	<ul style="list-style-type: none"> • Recibir Solicitud de Desarrollo de Software. • Analizar y Reasignar el Proyecto. • Informar al Área Usuaría.
Analista de Sistemas Informáticos.	Encargado del seguimiento y responsable de reunir métricas significativas para mostrar el estado del proyecto.	<ul style="list-style-type: none"> • Revisión de Estructura de Base de Datos. • Revisión de historias de usuario. • Revisión de Iteraciones completadas. • Comunicar el progreso del sistema.
Responsable de Software.	Encargado de Asignar un Proyecto a un Analista y realizar las pruebas de aceptación.	<ul style="list-style-type: none"> • Asignar el Desarrollo de Software a un Analista. • Convocar y ejecutar Reuniones. • Elaborar Memo y Acta de Reunión. • Realiza pruebas de aceptación, asegurando que el sistema funcione como se esperaba.

Tabla 3. 1: Roles del Sistema

Fuente: Propia

3.1.1. Integrantes del Equipo.

El cumplimiento de esta metodología se lleva a cabo por el grupo de personas que se involucran directamente en el desarrollo del sistema “SCOPT²⁵ GAD-I”. El equipo de desarrollo de software está conformado por las siguientes personas con su respectivo rol.

Nombre	Descripción	Rol XP
Ing. Irving Reascos.	Director de la Dirección de TIC.	Director TIC.
Lic. Sonia Bossano Subia.	Jefe del área de software de la dirección de TIC.	Responsable de Software.
Ing. Cristian Romero.	Analista de Sistemas de la Dirección de TIC.	Analista de Sistemas Informáticos.
Ing. Eduardo Navarrete.	Jefe del área de SISMERT	Usuario
Ciudadano	Usuario del parqueadero	Usuario
Luis Rodolfo Guamán Guamán	Tesista	Programador

Tabla 3. 2: Integrantes del Equipo.

Fuente: Propia.

3.2. Recepción de Solicitud para Desarrollo de Software

El jefe del área de SISMERT solicitó a la Dirección de TIC el desarrollo del software tecnológico para el control del parqueadero tarifado, con el fin de ofrecer un mejor servicio a los ciudadanos; presentando su respectivo memorándum.

El Director de TIC conjuntamente con el responsable de software ha comprobado la factibilidad técnica, operativa y económica para realizar el Proyecto Tecnológico “Sistema de Control de Parqueo Tarifado” del Área de SISMERT del Departamento de Tránsito del GAD-I.

3.3. Asignación y Planificación del Proyecto Tecnológico

El Proyecto Tecnológico “SISTEMA DE CONTROL DE PARQUEO TARIFADO” ha sido asignado al Plan Operativo Anual de la dirección de TIC del GAD-I.

²⁵ SCOPT: Sistema de Control de Parqueo tarifado

El mismo que también Ha sido asignado al Analista de Sistemas Cristian Romero, y será desarrollado por: Luis Guamán (Tesista Programador) con el respectivo memorándum de asignación de desarrollo de software.

Luego de haber asignado el Proyecto Tecnológico se realizo la reunión con el Analista de Sistemas, Programador y el Área Usuario de SISMERT, en la que se establecieron los objetivos y alcance del “Sistema De Control De Parqueo Tarifado”, para luego iniciar con su desarrollo. **VER ANEXO 2.**

3.4. Elaboración y Corrección del Proyecto Tecnológico

Luego de haber determinado los objetivos y alcance del Proyecto Tecnológico para el “Sistema De Control De Parqueo Tarifado”, se elabora el documento correspondiente a este proyecto con los principales puntos y requerimientos que el usuario ha establecido.

Además en el documento del proyecto se dispone de la información necesaria para elaborar la aplicación como es el caso de: software, hardware, personal y tiempo que se requiere para su desarrollo.

El documento fue revisado en la reunión donde estuvieron presentes el Responsable de Software, Director de TIC y Analistas de Sistemas, quienes propusieron expusieron sus inquietudes y recomendaciones para la corrección de este Documento de acuerdo a las funcionalidades que el sistema requiere, con la que quedo constatada en la respectiva acta de reunión. **VER ANEXO 3,4.**

3.5. Obtención y Documentación de Requisitos

La obtención de requisitos se obtiene en la reunión que se realizo siguiendo el proceso de desarrollo de software en la que estuvieron presentes el Analista de Sistemas de la Dirección de TIC, el Programador (Tesista) y el representante del Área Usuaría de SISMERT, donde se obtuvo los requisitos indispensables y necesarios para el desarrollo del “Sistema de Control de Parqueo Tarifado”, además se plantearon diferentes soluciones o pruebas de aceptación por parte del Analista de Sistemas y el Programador, para que la funcionalidad del sistema sea eficaz y de fácil uso.

Se debe señalar que las historias de usuario no solo reflejan beneficios para el usuario si no también que nos servirá para almacenar datos confiables, seguros y así el sistema podrá ofrecer una correcta funcionalidad; las historias pueden contener información como:

Seguridad, Almacenamiento y Disponibilidad de Datos, etc.

A continuación se detallan las historias de usuario recopiladas en las reuniones realizadas. En estas describen los requisitos que se diseñaran y programaran en las iteraciones propuestas del Sistema, además que servirán para describir la funcionalidad del mismo.

3.5.1. Historias de Usuario

HISTORIA DE USUARIO			
Fecha:	20 de Marzo 2012		
Proyecto:	Sistema de Control del parqueo Tarifado (SISMERT) para el Gobierno Autónomo Descentralizado de la ciudad de Ibarra		
Nombre de la historia:	Requerimientos para uso del Parqueadero Sismert		
ID de historia:	I	Iteración Asignada:	1
Dirección:	Transito y Transporte	Funcionario:	Eduardo Navarrete
Unidad:	SISMERT	Cargo:	Coordinador
Prioridad:	ALTA	Estimación:	130 horas
Analista Responsable:	Cristian Romero	Dependencia:	
Desarrollador:	Luis Guamán		
Descripción			
Como usuario (Sismert) quiero que los Ciudadanos se registren con su número de teléfono y cedula de identidad para poder identificarlos, además puedan modificar los datos del registro. También que pueda iniciar el servicio de parqueo con un SMS, puede ser a través de la contratación de una operadora móvil, de tal manera que el ciudadano pague la misma tarifa que con el Sistema actual.			
Pruebas de Aceptación			
<ul style="list-style-type: none"> ▪ Para el registro se creara un formulario donde el ciudadano ingrese su número de cedula y número de teléfono, especificando la operadora móvil que utiliza. ▪ El registro tendrá una ventana de edición de datos si el ciudadano ha ingresado su teléfono erróneo también podrá eliminar este registro. ▪ Para la recepción de sms no se contratará una operadora móvil, ya que el departamento de TIC cuenta con un sistema de envío y recepción de sms, el mismo que se lo utilizara para este fin. 			
Observaciones			
Solo los ciudadanos que consten en el sistema municipal pueden ser registrados, caso contrario deben acercarse al GAD-I para registrar sus datos.			

Tabla 3. 3: Historia de Usuario 1.

Fuente: Propia.

HISTORIA DE USUARIO			
Fecha:	26 de Marzo 2012		
Proyecto:	Sistema de Control del parqueo Tarifado (SISMERT) para el Gobierno Autónomo Descentralizado de la ciudad de Ibarra		
Nombre de la historia:	Repositorio de Datos SISMERT		
ID de historia:	II	Iteración Asignada:	1

HISTORIA DE USUARIO			
Dirección:	TIC	Funcionario:	Cristian Romero
Unidad:	Software	Cargo:	Analista
Prioridad:	ALTA	Estimación:	60 horas
Analista Responsable:	Cristian Romero	Dependencia:	
Desarrollador:	Luis Guamán		
Descripción			
<p>Como Analista de sistemas, quiero almacenar la información que generara el nuevo sistema para el control del parqueo tarifado, con la finalidad de que la información se encuentre segura y el sistema tenga una buena funcionalidad.</p> <p>También analizar e integrar los nuevos modelos de datos al esquema actual que se encuentra en la base de datos centralizada del GAD-I, verificando que el acceso e ingreso de datos sean correctos con los nuevos atributos que tendrá cada modelo relacional.</p>			
Pruebas de Aceptación			
<ul style="list-style-type: none"> ▪ Se utilizara el esquema actual de SISMERT para almacenar las tablas correspondientes al nuevo sistema. ▪ Comprobar funcionalidad a través del ingreso de datos. ▪ Utilización de datos existentes para integrarlos al nuevo sistema. ▪ El modelo de datos inicial se creara de acuerdo a la funcionalidad inicial que el usuario del Sismert ha solicitado (Sección de Registros). 			
Observaciones			

Tabla 3. 4: Historia de Usuario 2

Fuente: Propia.

HISTORIA DE USUARIO			
Fecha:	06 de Abril 2012		
Proyecto:	Sistema de Control del parqueo Tarifado (SISMERT) para el Gobierno Autónomo Descentralizado de la ciudad de Ibarra		
Nombre de la historia:	Seguridad y control de acceso		
ID de historia:	III	Iteración Asignada:	1
Dirección:	Transito y Transporte	Funcionario:	Eduardo Navarrete
Unidad:	SISMERT	Cargo:	Coordinador
Prioridad:	ALTA	Estimación:	12 horas
Analista Responsable:	Cristian Romero	Dependencia:	
Desarrollador:	Luis Guamán		
Descripción			
<p>Como usuario del sistema, quiero autenticarme ingresando el nombre de usuario y contraseña, de tal manera que el acceso sea seguro y confiable para que otros usuarios no puedan acceder al sistema.</p>			

HISTORIA DE USUARIO
Al sistema solo pueden ingresar los usuarios que tengan permiso de acceso y si no es usuario informarle que no tiene permiso para ingresar a este sistema.
Pruebas de Aceptación
<ul style="list-style-type: none"> ▪ Se utilizara la base de datos centralizada de usuarios del GAD-I. ▪ Cada usuario tendrá permisos de acceso hacia los menús e ítems que contenga la aplicación para la administración que se genera en la parte interna del sistema. ▪ El sistema estará publicada en la aplicación Integral del GAD-I que posee el modulo de seguridad para controlar el acceso de usuarios.
Observaciones

Tabla 3. 5: Historia de Usuario 3.

Fuente: Propia.

HISTORIA DE USUARIO			
Fecha:	10 de Abril 2012		
Proyecto:	Sistema de Control del parqueo Tarifado (SISMERT) para el Gobierno Autónomo Descentralizado de la ciudad de Ibarra		
Nombre de la historia:	Contenido del SMS para uso del Parqueadero		
ID de historia:	IV	Iteración Asignada:	1
Dirección:	Transito y Transporte	Funcionario:	Eduardo Navarrete
Unidad:	SISMERT	Cargo:	Coordinador
Prioridad:	MEDIA	Estimación:	30 horas
Analista Responsable:	Cristian Romero	Dependencia:	
Desarrollador:	Luis Guamán		
Descripción			
<p>Como usuario (Sismert) quiero que el ciudadano escriba un SMS para la solicitud de estacionamiento en un sitio del SISMERT. El SMS debe estar formado con el siguiente contenido: su placa vehicular, las calles donde se va a parquear, y el tiempo que va a permanecer estacionado.</p> <p>El usuario (Ciudadano) no debe demorarse en escribir el contenido del SMS y debe identificar el lugar donde se estaciona y el tiempo que permanecerá estacionado a través de un código que se le asigne al sitio de parqueo y al tiempo de uso.</p>			
Pruebas de Aceptación			
<ul style="list-style-type: none"> ▪ Se asignara una letra del alfabeto a cada vehículo, y asi el ciudadano podrá enviar en su SMS con su placa vehicular o la letra asignada, reduciendo el tiempo de escritura del contenido del SMS. ▪ Para el tiempo de estacionamiento se creara una numeración que identificara a cada tiempo establecido en área de SIMERT. 			

HISTORIA DE USUARIO
<ul style="list-style-type: none"> ▪ El ciudadano debe incluir la letra P en su SMS para indicar que se estacionó y poder hacer huso del sistema. ▪ Para el sitio de parqueo se realizará un análisis para obtener su código de identificación. ▪ El usuario enviara el siguiente SMS que contendrá: <ul style="list-style-type: none"> ✓ P A 1 o P AA0000 1(Sin Sitio de Parqueo)
Observaciones

Tabla 3. 6: Historia de Usuario 4.

Fuente: Propia.

HISTORIA DE USUARIO			
Fecha:	16 de Abril 2012		
Proyecto:	Sistema de Control del parqueo Tarifado (SISMERT) para el Gobierno Autónomo Descentralizado de la ciudad de Ibarra		
Nombre de la historia:	Registro de vehículos		
ID de historia:	V	Iteración Asignada:	1,2,5
Dirección:	Transito y Transporte	Funcionario:	Eduardo Navarrete
Unidad:	SISMERT	Cargo:	Coordinador
Prioridad:	ALTA	Estimación:	90 horas
Analista Responsable:	Cristian Romero	Dependencia:	
Desarrollador:	Luis Guamán		
Descripción			
<p>Como usuario (Sismert) quiero que un ciudadano registre su vehículo, este debe ser asignado a un número de teléfono previamente registrado, y también tenga la opción de eliminar un registro de un vehículo cuando el ciudadano necesite darle de baja.</p> <p>Analizar la manera de reutilizar los datos de los vehículos que se encuentran registrados en las certificaciones de matriculación vehicular del SISMERT.</p>			
Pruebas de Aceptación			
<ul style="list-style-type: none"> ▪ Se almacenara en la Base de Datos a todos los vehículos que pasen por la certificación del Sismert (tomar información vehicular para reutilizar en este sistema). ▪ Se creara un formulario en el que primero ingrese un número de teléfono valido, y seguidamente se mostrara el formulario para que ingrese su vehículo a registrar. ▪ Se comprobara si el vehículo ingresado consta entre los vehículos almacenados en el Sistema. ▪ Se asignara su respectiva identificación de la placa vehicular con una letra del alfabeto, para el envío del SMS (PCC5705 A). ▪ Se creara un reporte de los vehículos asignados a un usuario, con la opción de eliminar el registro de un vehículo. 			
Observaciones			

Tabla 3. 7: Historia de Usuario 5.

Fuente: Propia.

HISTORIA DE USUARIO			
Fecha:	16 de Abril 2012		
Proyecto:	Sistema de Control del parqueo Tarifado (SISMERT) para el Gobierno Autónomo Descentralizado de la ciudad de Ibarra		
Nombre de la historia:	Establecer Identificación de Sitios para el Parqueadero		
ID de historia:	VI	Iteración Asignada:	4
Dirección:	Transito y Transporte	Funcionario:	Eduardo Navarrete
Unidad:	SISMERT	Cargo:	Coordinador
Prioridad:	ALTA	Estimación:	112 horas
Analista Responsable:	Cristian Romero	Dependencia:	
Desarrollador:	Luis Guamán		
Descripción			
<p>Como usuario (Sismert) quiero que los ciudadanos identifiquen los sitios donde se encuentran ubicados los puntos del parqueadero del Sismert por su nombre de calle, para que este lo envíen en su SMS cuando inicien el servicio de estacionamiento.</p>			
Pruebas de Aceptación			
<ul style="list-style-type: none"> ▪ Primero se identificaran las calles que están dentro de las zonas de Sismert. ▪ Se formara un sitio del parqueo con la calle principal y transversal. ▪ El Sitio de parqueadero tendrá un código de identificación con el que el ciudadano enviara en su sms cuando haga uso del parqueadero. ▪ Se creara una zona albergando algunos sitios de inspección para organizar por grupos los lugares o sitios de estacionamiento. ▪ El formato del código del sitio será de la siguiente manera Simón Bolívar y Pedro de Moncayo -> SBPM. ▪ El usuario iniciará el servicio de estacionamiento enviando el siguiente SMS que contendrá: ✓ P A SBPM 1 o P AA0000 SBPM 1(Identificando el Sitio de Parqueo) 			
Observaciones			

Tabla 3. 8: Historia de Usuario 6.

Fuente: Propia.

HISTORIA DE USUARIO			
Fecha:	1 de Mayo 2012		
Proyecto:	Sistema de Control del parqueo Tarifado (SISMERT) para el Gobierno Autónomo Descentralizado de la ciudad de Ibarra		
Nombre de la historia:	Compras de Saldo, Información Registros		
ID de historia:	VII	Iteración Asignada:	2,9
Dirección:	Transito y Transporte	Funcionario:	Eduardo Navarrete
Unidad:	SISMERT	Cargo:	Coordinador
Prioridad:	ALTA	Estimación:	65 horas

HISTORIA DE USUARIO			
Analista Responsable:	Cristian Romero	Dependencia:	
Desarrollador:	Luis Guamán		
Descripción			
<p>Como usuario (Sismert) que los Ciudadanos puedan realizar recargas electrónicas con la que podrán comprar minutos para usar el parqueadero. Las compras de saldo serán solo para los teléfonos que estén registrados. Además visualizar información de los registros de teléfonos, y visualizar gráficamente como utiliza el parqueadero.</p>			
Pruebas de Aceptación			
<p>Se creara en el sistema la opción de comprar el saldo de las siguiente manera:</p> <ul style="list-style-type: none"> ▪ Comprar Saldo Sismert esta opción funcionara solo en intranet. ▪ Posibilidad de realizar pagos en línea, el ciudadano podrá comprar saldo a través de un sitio web. <p>Para la visualización de la información se tendrá:</p> <ul style="list-style-type: none"> ▪ Una ventana que muestra los registros de todos los teléfonos. ▪ Informe estadístico del uso del parqueadero por año. 			
Observaciones			

Tabla 3. 9: Historia de Usuario 7.

Fuente: Propia.

HISTORIA DE USUARIO			
Fecha:	1 de Mayo 2012		
Proyecto:	Sistema de Control del parqueo Tarifado (SISMERT) para el Gobierno Autónomo Descentralizado de la ciudad de Ibarra		
Nombre de la historia:	Cambiar Lugar de Estacionamiento		
ID de historia:	VIII	Iteración Asignada:	3
Dirección:	Transito y Transporte	Funcionario:	Eduardo Navarrete
Unidad:	SISMERT	Cargo:	Coordinador
Prioridad:	MEDIA	Estimación:	38 horas
Analista Responsable:	Cristian Romero	Dependencia:	
Desarrollador:	Luis Guamán		
Descripción			
<p>Como usuario (Sismert) quiero que los Ciudadanos tengan la opción de cambiarse de un sitio a otro cuando hacen uso del parqueadero y este cambio sea informado a los inspectores más cercanos.</p> <p>Los Inspectores deben tener un número de teléfono al que se les comunice el cambio e inicio de parqueo con un SMS, debe haber una administración de asignación y edición del número de teléfono de cada Inspector.</p>			

HISTORIA DE USUARIO
Pruebas de Aceptación
<ul style="list-style-type: none"> ▪ Se creara la opción en la que el ciudadano pueda enviar un sms indicando el lugar a donde se cambiara de sitio parqueo. ▪ El sistema enviara un sms al inspector indicándole la placa vehicular del ciudadano, el lugar en el que se estaciono y el tiempo disponible. ▪ Crear interfaz grafica para asignación y edición de los números de teléfonos de cada inspector.
Observaciones

Tabla 3. 10: Historia de Usuario 8.

Fuente: Propia.

HISTORIA DE USUARIO			
Fecha:	22 de Mayo 2012		
Proyecto:	Sistema de Control del parqueo Tarifado (SISMERT) para el Gobierno Autónomo Descentralizado de la ciudad de Ibarra		
Nombre de la historia:	Notificaciones Inspector y Ciudadano		
ID de historia:	IX	Iteración Asignada:	3,10
Dirección:	Transito y Transporte	Funcionario:	Eduardo Navarrete
Unidad:	SISMERT	Cargo:	Coordinador
Prioridad:	ALTA	Estimación:	70 horas
Analista Responsable:	Cristian Romero	Dependencia:	
Desarrollador:	Luis Guamán		
Descripción			
<p>Como usuario (Sismert) quiero que los ciudadanos e inspectores puedan ser notificados al momento de hacer uso del parqueadero, esta notificación tiene que ser vía SMS al Ciudadano se le debe avisar el tiempo de inicio y fin del servicio, y al Inspector la misma información incluido la placa del vehículo que esté usando el servicio. Además debe haber una interfaz grafica para la asignación de sitios de Inspección.</p>			
Pruebas de Aceptación			
<ul style="list-style-type: none"> ▪ Hacia el ciudadano se le enviara una notificación indicándole si se pudo o no hacer el uso del estacionamiento. ▪ Se enviara un sms al inspector cuando el ciudadano haga uso del parqueadero indicándole la hora de inicio y fin del servicio con la respectiva placa vehicular del ciudadano. ▪ Para informarle al inspector se creará una interfaz gráfica que permita administrar el lugar de inspección que se la asigne a un Inspector. 			
Observaciones			

Tabla 3. 11: Historia de Usuario 9.

Fuente: Propia.

HISTORIA DE USUARIO			
Fecha:	03 de Septiembre del 2012		
Proyecto:	Sistema de Control del parqueo Tarifado (SISMERT) para el Gobierno Autónomo Descentralizado de la ciudad de Ibarra		
Nombre de la historia:	Plantear nueva solución de notificación Hacia el inspector		
ID de historia:	X	Iteración Asignada:	6
Dirección:	TIC	Funcionario:	Sonia Bossano
Unidad:	Software	Cargo:	Responsable Software
Prioridad:	ALTA	Estimación:	24 horas
Analista Responsable:	Cristian Romero	Dependencia:	
Desarrollador:	Luis Guamán		
Descripción			
<p>Como Responsable de Software se necesita que se analice una nueva solución de notificación hacia los inspectores, porque la solución planteada no es factible económicamente al tener un costo muy elevado por el envío de SMS de notificaciones al Inspector. La solución más factible puede ser a través de una aplicación móvil para que el inspector controle el uso del parqueadero.</p>			
Pruebas de Aceptación			
<ul style="list-style-type: none"> ▪ Se analizara la posibilidad de crear una aplicación móvil para el inspector. ▪ Se verificara si esta aplicación planteada es posible desarrollarla con el framework ExtJS para su solución. ▪ Establecer los servicios que va a ofrecer la aplicación móvil. ▪ Analizar y crear una nueva aplicación para un dispositivo móvil, integrada al sistema actual del control del parqueo tarifado. 			
Observaciones			
<p>Se ha podido constatar que la notificación hacia los inspectores no es factible económicamente, por que el envío de SMS debería ir hacia todos los inspectores, resultando asi un costo demasiado alto para esta solución, por eso es recomendable plantear y analizar una nueva solución.</p>			

Tabla 3. 12: Historia de Usuario 10.

Fuente: Propia

HISTORIA DE USUARIO			
Fecha:	11 de Septiembre del 2012		
Proyecto:	Sistema de Control del parqueo Tarifado (SISMERT) para el Gobierno Autónomo Descentralizado de la ciudad de Ibarra		
Nombre de la historia:	Creación Aplicación Móvil y Seguridad.		
ID de historia:	XI	Iteración Asignada:	6
Dirección:	Transito y Transporte	Funcionario:	Eduardo Navarrete
Unidad:	SISMERT	Cargo:	Coordinador
Prioridad:	ALTA	Estimación:	80 horas
Analista Responsable:	Cristian Romero	Dependencia:	
Desarrollador:	Luis Guamán		
Descripción			
<p>Como usuario (SISMERT) se requiere crear una aplicación móvil para el control del tiempo de estacionamiento y además implementar la seguridad para que solo los inspectores puedan acceder a esta aplicación. Esta aplicación deber tener una interfaz amigable para que los inspectores no tengan problemas al momento de acceder y hacer uso de sus servicios.</p>			
Pruebas de Aceptación			
<ul style="list-style-type: none"> ▪ Se creara una aplicación Touch, ya que se podrá visualizar de mejor manera los formularios que tendrá la aplicación, facilitando su uso. ▪ Para el control de la seguridad la aplicación se acoplara con la base de datos centralizada del municipio. ▪ Se reutilizara la lógica de acceso de la aplicación integral del GAD-I (Core). ▪ El acceso será por cada inspector y cada uno tendrá su cuenta de usuario. 			
Observaciones			

Tabla 3. 13: Historia de Usuario 11.

Fuente: Propia

HISTORIA DE USUARIO			
Fecha:	11 de Septiembre del 2012		
Proyecto:	Sistema de Control del parqueo Tarifado (SISMERT) para el Gobierno Autónomo Descentralizado de la ciudad de Ibarra		
Nombre de la historia:	Consulta Tiempo Parqueo y Notificaciones en Aplicación Móvil.		
ID de historia:	XII	Iteración Asignada:	6
Dirección:	Transito y Transporte	Funcionario:	Eduardo Navarrete
Unidad:	SISMERT	Cargo:	Coordinador
Prioridad:	ALTA	Estimación:	70 horas
Analista Responsable:	Cristian Romero	Dependencia:	

HISTORIA DE USUARIO	
Desarrollador:	Luis Guamán
Descripción	
<p>Como usuario (SISMERT) se requiere que los inspectores tengan acceso a la verificación del tiempo disponible de un vehículo ingresando su placa y comprueben el tiempo que les resta para ocupar el servicio, caso contrario tengan la posibilidad de emitir una notificación con los datos principales como placa vehicular, tipo de infracción y hora en la que se cometio la infracción.</p>	
Pruebas de Aceptación	
<ul style="list-style-type: none"> ▪ Se creara la opción de ingresar una placa vehicular, y a continuación se podrá visualizar el tiempo que un vehículo tiene disponible para estacionarse. ▪ Si el tiempo se termina se mostrara un mensaje indicando que el vehículo ya no dispone del tiempo para usar el estacionamiento. ▪ También se creara la opción de ingresar una notificación en el caso de que un vehículo haya infringido el uso de una zona regulada del SISMERT. 	
Observaciones	

Tabla 3. 14: Historia de Usuario 12.

Fuente: Propia

HISTORIA DE USUARIO			
Fecha:	11 de Septiembre del 2012		
Proyecto:	Sistema de Control del parqueo Tarifado (SISMERT) para el Gobierno Autónomo Descentralizado de la ciudad de Ibarra		
Nombre de la historia:	Registro y Compras de Saldo en Aplicación Móvil.		
ID de historia:	XIII	Iteración Asignada:	6
Dirección:	Transito y Transporte	Funcionario:	Eduardo Navarrete
Unidad:	SISMERT	Cargo:	Coordinador
Prioridad:	ALTA	Estimación:	118 horas
Analista Responsable:	Cristian Romero	Dependencia:	
Desarrollador:	Luis Guamán		
Descripción			
<p>Como usuario (SISMERT) se requiere que la aplicación tenga acceso al registro de un teléfono y registro de vehículos como en la aplicación web interna, el inspector pueda realizar recargas a un número de teléfono ingresando el numero y el monto de la recarga. También que el inspector pueda iniciar el servicio de estacionamiento de un Ciudadano cuando este lo solicite.</p>			
Pruebas de Aceptación			

HISTORIA DE USUARIO
<ul style="list-style-type: none"> ▪ Se creara un formulario similar al registro de la aplicación web elaborado anteriormente, con esto el inspector podrá ingresar nuevos teléfonos de los ciudadanos para el uso del parqueadero. ▪ Se creara el formulario para registro de vehículos con los datos requeridos en la aplicación web interna de forma que el registro sea similar. ▪ Crear un formulario que permita ingresar el monto de la recarga y el número de teléfono al que se acreditar el saldo. ▪ Adicionalmente se creara la opción de iniciar el servicio de parqueo ingresando el número de teléfono y la placa del vehículo que se va a estacionar. ▪ Cada formulario tendrá su propio control y validaciones respectivamente.
Observaciones
Las opciones creadas son exclusivamente para el inspector.

Tabla 3. 15: Historia de Usuario 13.

Fuente: Propia

HISTORIA DE USUARIO			
Fecha:	29 de Octubre del 2012		
Proyecto:	Sistema de Control del parqueo Tarifado (SISMERT) para el Gobierno Autónomo Descentralizado de la ciudad de Ibarra		
Nombre de la historia:	Recaudación de Compras de Saldo SISMERT		
ID de historia:	XIV	Iteración Asignada:	7,9
Dirección:	Transito y Transporte	Funcionario:	Eduardo Navarrete
Unidad:	SISMERT	Cargo:	Coordinador
Prioridad:	ALTA	Estimación:	66 horas diarias
Analista Responsable:	Cristian Romero	Dependencia:	
Desarrollador:	Luis Guamán		
Descripción			
<p>Como usuario (SISMERT) quiero poder recaudar el dinero a cada inspector de las recargas realizadas en el día de todas las transacciones generadas por la aplicación móvil, también quiero verificar cuánto dinero se ha recaudado en el día por el monto de recargas (1, 3, 5 y 10 dólares).</p> <p>Además necesito generar reportes de estas transacciones por responsable y fecha o simplemente de todas las transacciones sin filtrarlos.</p>			
Pruebas de Aceptación			
<ul style="list-style-type: none"> ▪ Se creara una ventana donde se mostrara los registros de las recaudaciones por fecha, responsable. ▪ Se creara un formulario en el que se ingrese la cedula del responsable que ha realizado la recarga, y se obtendrá el total del dinero recaudado en el día. 			

<ul style="list-style-type: none"> ▪ Se creara un reporte de todos los inspectores con el dinero recaudado en el día. ▪ Además se tendrá varios reportes que informen de las recaudaciones que se han realizado en las recargas del SISMERT con o sin filtros.
Observaciones

Tabla 3. 16: Historia de Usuario 14.

Fuente: Propia

HISTORIA DE USUARIO			
Fecha:	06 de Noviembre del 2012		
Proyecto:	Sistema de Control del parqueo Tarifado (SISMERT) para el Gobierno Autónomo Descentralizado de la ciudad de Ibarra		
Nombre de la historia:	Servicios en la web para el Ciudadano		
ID de historia:	XV	Iteración Asignada:	8
Dirección:	Transito y Transporte	Funcionario:	Eduardo Navarrete
Unidad:	SISMERT	Cargo:	Coordinador
Prioridad:	ALTA	Estimación:	150 horas
Analista Responsable:	Cristian Romero	Dependencia:	
Desarrollador:	Luis Guamán		
Descripción			
<p>Como usuario (SISMERT) se requiere que el ciudadano tenga acceso a los servicios desde la web sin necesidad de acercarse al GAD-I (Desde cualquier lugar que tenga acceso a internet). El ciudadano tendrá la opción de registrar su número de teléfono y vehículo como en la aplicación de intranet, cada usuario debe tener su propia cuenta de acceso para su seguridad por que estos datos deben ser visibles solo para el usuario que ha ingresado.</p> <p>Además se debe contar con la opción de que el ciudadano acceda al reporte del uso de estacionamiento por mes, también pueda visualizar información de las calles donde se encuentran los puntos de parqueo del SISMERT y acceda a la información de cómo debe usar el sistema y que servicios brinda.</p>			
Pruebas de Aceptación			
<ul style="list-style-type: none"> ▪ Se creará una nueva aplicación reutilizando la aplicación de intranet con la finalidad de mantener la seguridad y acceso a la aplicación. ▪ También se creará un formulario de registro y así el usuario tenga permiso de acceder a esta aplicación. ▪ Si ha olvidado su contraseña, el sistema le proporcionará la opción de restaurar su clave de acceso al sistema, mediante un formulario que le pedirá datos que ingreso en su registro. ▪ Se creará un módulo de seguridad integrada a la aplicación interna, donde se le pedirá al usuario que ingrese el número de cédula y contraseña para acceder al sistema. ▪ Se creará los respectivos formularios para el registro de teléfonos y vehículos similar a la interfaz de la aplicación interna ya creada. 			

HISTORIA DE USUARIO
<ul style="list-style-type: none"> ▪ El ciudadano visualizará una ventana para acceder al reporte de uso de estacionamiento por mes donde debe ingresar el número de teléfono y el mes del que desea obtener los datos. ▪ Adicionalmente el usuario podrá descargar un documento PDF donde se le muestra como se debe usar el sistema y las opciones que tiene. ▪ También se le mostrará al usuario una ventana con las calles que el SISMERT tiene el servicio de estacionamiento.
Observaciones
Los Ciudadanos podrán ingresar a esta aplicación solo si están registrados como ciudadanos en el Municipio, caso contrario deberán acercarse a la Institución para solicitar su registro.

Tabla 3. 17: Historia de Usuario 15.

Fuente: Propia

HISTORIA DE USUARIO			
Fecha:	04 de Diciembre del 2012		
Proyecto:	Sistema de Control del parqueo Tarifado (SISMERT) para el Gobierno Autónomo Descentralizado de la ciudad de Ibarra		
Nombre de la historia:	Recargas SISMERT en Línea		
ID de historia:	XVI	Iteración Asignada:	8
Dirección:	TIC	Funcionario:	Personal TIC
Unidad:	Software	Cargo:	Analistas de Sistemas
Prioridad:	BAJA	Estimación:	18 horas
Analista Responsable:	Cristian Romero	Dependencia:	
Desarrollador:	Luis Guamán		
Descripción			
<p>Como usuario de TIC se necesita implementar el servicio de compras en línea, por tal razón se necesita crear la interfaz gráfica para realizar el proceso de compras en línea de las recargas de Saldo SISMERT, ya que posteriormente se lo implementará con los demás procesos de transacciones electrónicas que brindará el GAD-I hacia los ciudadanos. Este proceso de compras será realizado por el personal de TIC reutilizando la GUI creada en esta historia.</p>			
Pruebas de Aceptación			
<ul style="list-style-type: none"> ▪ Se creara un formulario donde el ciudadano ingrese su cedula de identidad, monto de la recarga, y el tipo de tarjeta con la que va a comprar. ▪ El formulario quedara listo para el envío de datos cuando se implemente las seguridades respectivas, y el convenio con la empresa que presta el servicio de realizar transacciones electrónicas. 			

HISTORIA DE USUARIO
Observaciones
El proceso de transacciones electrónicas se lo implementara posteriormente, cuando se haya establecido las seguridades de acceso, se debe tomar mucha precaución por el motivo de estar en la web, la misma que puede estar amenazada por hackers.
Con esta seguridad se protegerá el acceso indebido a la página web que brindara el servicio de transacciones electrónicas.
Las seguridades a implementar lo desarrollara el personal de la dirección de TIC por lo que se pretende crear un balcón de servicios web para la ciudadanía. El proceso de Transacciones de pagos en línea se lo implementara con la adquisición del firewall para controlar la seguridad de este servicio.

Tabla 3. 18: Historia de Usuario 16.

Fuente: Propia

3.5.2. Pila de Producto

La pila de producto recopila todas las historias de usuario que se han recopilado en la obtención de requerimientos para el desarrollo del sistema.

Las historias se encuentran sin clasificarlas por la iteración a la que le corresponde, ayuda a establecer la funcionalidad que ve a tener el sistema con los compromisos que se ha adquirido en las historias de usuario.

A continuación se detalla la pila de producto que se ha formado para elaborar el Sistema de Control de Parqueo Tarifado.

PILA DE PRODUCTO		
ID	Nombre Historia de Usuario	Prioridad
1	Requerimientos para uso del Parqueadero Sismert.	ALTA
2	Repositorio de Datos SISMERT.	ALTA
3	Seguridad y Control de Acceso.	ALTA
4	Contenido del SMS para uso del Parqueadero.	MEDIA
5	Registro de vehículos.	ALTA
6	Establecer Identificación de Sitios para el Parqueadero.	MEDIA
7	Compras de Saldo, Información Registros.	ALTA
8	Cambiar Lugar de Estacionamiento.	MEDIA
9	Notificaciones Inspector y Ciudadano.	ALTA
10	Plantear nueva solución de notificación Hacia el inspector.	ALTA

PILA DE PRODUCTO		
ID	Nombre Historia de Usuario	Prioridad
11	Creación Aplicación Móvil y Seguridad.	ALTA
12	Consulta Tiempo Parqueo y Notificaciones.	ALTA
13	Registro y Compras de Saldo en Aplicación Móvil.	ALTA
14	Recaudación de Compras de Saldo SISMERT	ALTA
15	Servicios en la web para el Ciudadano	ALTA
16	Recargas SISMERT en Línea	BAJA

Tabla 3. 19: Pila de Producto SCOPT.

Fuente: Propia.

Más adelante se agrupa las historias de usuario asignándoles a una iteración dependiendo de las tareas que se debe realizar, formando la aplicación mediante la elaboración de Iteraciones.

3.5.3. Metáfora.

Tras haber obtenido los requisitos reflejados en los documentos de historias de usuario, conjuntamente con el personal de la Dirección de TIC y el Área Usuaría de SISMERT, se ha podido determinar la metáfora del sistema que se describe a continuación:

El sistema que se va desarrollar es una aplicación WEB, con la que se podrá hacer uso del parqueadero de SISMERT a través de un registro de teléfonos y vehículos de los ciudadanos, este servicio se complementara con el envío de SMS y notificaciones desde y hacia el ciudadano, el mismo que se lo usara mediante compras de saldo acumulativos en un número de teléfono previamente registrado.

Desde la aplicación web también se podrá consultar el saldo disponible y hacer compras de saldo en línea para darle más facilidad a que el ciudadano pueda comprar saldo acumulativo de SISMERT. Además el sistema permitirá a los inspectores verificar si un vehículo estacionado en una plaza (sitio de parqueo) cuenta con tiempo disponible para usar el parqueadero.

3.6. Diseño del Sistema.

Con las historias de usuario que se han podido recopilar y haciendo uso de su información, ahora se procede a diseñar la arquitectura del sistema y la arquitectura funcional. Además se procede a especificar cada uno de los módulos y su funcionalidad con los que contara el sistema.

3.6.1. Arquitectura del Sistema.

El Sistema de Control de Parqueo Tarifado se va a desarrollar a través de una Tecnología Web. El patrón o modelo de abstracción de desarrollo de software está basada en N-capas que está conformado por:

El modelo, vista y controlador en la capa del cliente y en la capa del servidor, permitiendo que la aplicación sea escalable y forme una arquitectura robusta.

Modelo: Esta capa integra el acceso a los objetos relacionales de la Base de Datos PostgreSQL, con la herramienta de mapeo Doctrine ORM (Mapeo-Objeto Relacional) que permite obtener la información de una tabla a través de un objeto, permitiendo de esta manera acceder a los datos de una forma más fácil y sencilla.

El Doctrine permite crear los patrones de diseño de la aplicación para el acceder y modificar los datos de un objeto relacional. El Doctrine está compuesto por:

- DTO: Objetos de transferencia de datos representados en los .class.
- DAO: Acceso a los datos de un objeto estos están representados por las clases (TABLE).
- Manager Connection: Archivo de configuración de conexión a la base de datos (database.yaml).

En esta capa también se encuentra la Lógica del Negocio en donde se ejecutan todas las operaciones que va hacer el software, y toda la funcionalidad que este tendrá.

El Manager administra las funciones de la capa de lógica del negocios en esta capa se puede integrar diversas funcionalidades, que serán ejecutadas por el controlador de la aplicación, en Symfony representadas como las acciones de los módulos.

Controlador: La capa del controlador es el que interactúa entre el cliente y el servidor porque controla el flujo de las solicitudes realizadas a la aplicación, en esta capa se utiliza el framework Symfony 1.4.16.

Para el desarrollo de la presente aplicación se ha utilizado las acciones de los módulos que se encargan de responder a las llamadas que el usuario envía en una petición, estas acciones están conformadas por las funciones de la capa del Manager; toda acción que cumpla con la funcionalidad del sistema debe estar en esta capa para poder interactuar con el cliente.

Vista: Para el desarrollo de la interfaz grafica en la parte del cliente se utiliza el framework Ext JS 4.1.6, que permite crear una interfaz de escritorio, interactiva con páginas web dinámicas sin necesidad de recargarla cuando existe algún cambio de su contenido, es de fácil uso y más atractiva hacia el usuario. También nos permitirá optimizar el rendimiento a través del control de solicitudes manejadas por Ajax, las solicitudes del cliente son receptadas por las acciones que se encuentra en los módulos que están contenidos en la capa del controlador de la aplicación.

El framework está basado en el patrón de diseño de desarrollo web MVC, que contiene tres capas para separar el acceso a los datos, lógica de negocios y interfaz gráfica en la parte dl cliente.

- ❖ Modelo: La capa del modelo está constituido por un Store²⁶ que contiene un conjunto de datos que pueden ser desplegados en combos y grillas accedidas por los patrones de diseño DAO y DTO.
- ❖ Vista: Son los componentes que el usuario puede visualizar Ejem: textfield, combo, check.
- ❖ Controlador: Son las acciones o peticiones solicitadas con Ajax al servidor.

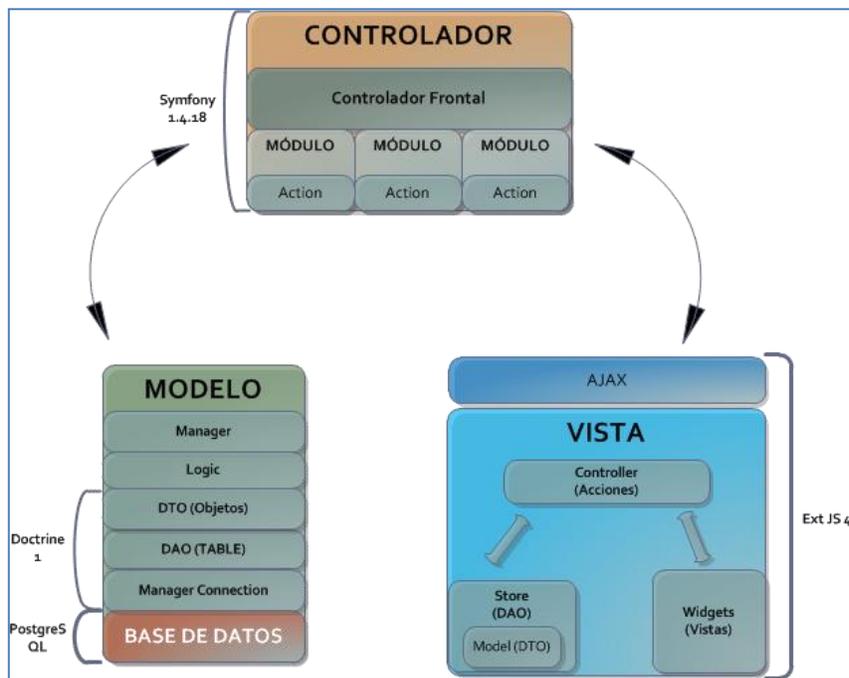


Figura 3. 1: Arquitectura del Sistema.
Fuente: Core GAD-I

²⁶ Store: Conjunto o Almacén de Datos.

Datos: Como base de datos alfanumérica y binaria se utiliza PostgreSQL, la misma que es utilizada por los sistemas del GAD-I.

Esta arquitectura fue propuesta pensando en el enfoque e integración que tiene la aplicación con los demás sistemas que se desarrollan en el Core del GAD-I, permitiendo que la aplicación sea escalable de acuerdo a nuevas operaciones o funcionalidades que el usuario solicite.

3.6.2. Arquitectura Funcional.

El Sistema de Control de Parqueo Tarifado se va a desarrollar con el servidor web Apache, como servidor de aplicaciones Apache Tomcat para generar reportes desde IReport, el repositorio de datos alfanuméricos PostgreSQL.

El Sistema será subido a producción sobre la plataforma Linux como sistema operativo Debian, como servidor de aplicaciones Apache, Apache Tomcat como servidor de aplicaciones y por ultimo como repositorio de datos alfanuméricos y Binarios PostgreSQL.

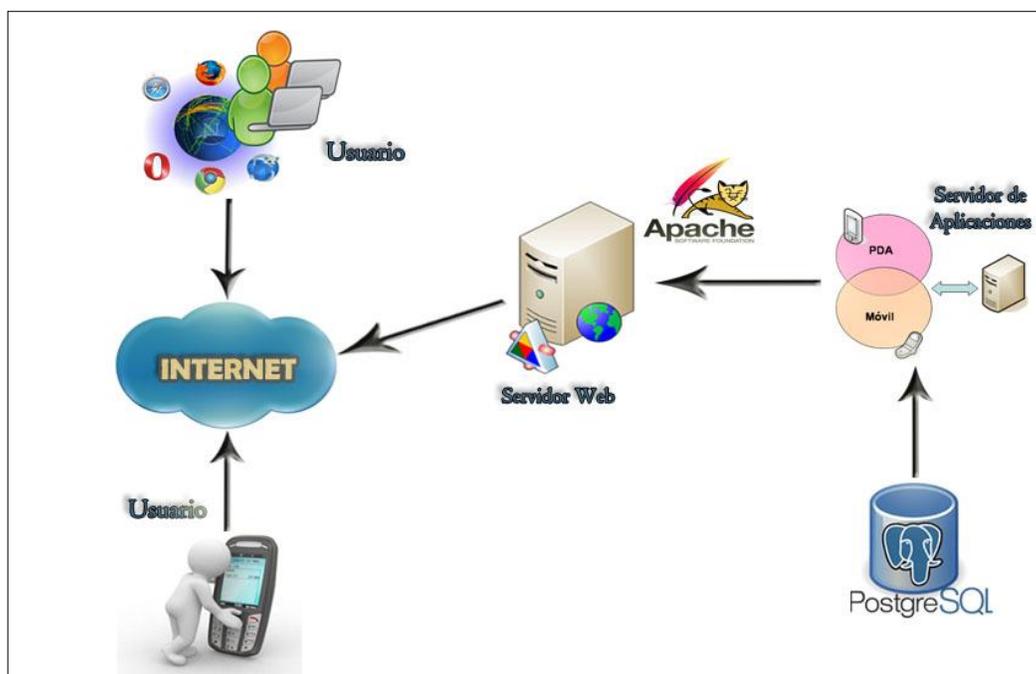


Figura 3. 2: Arquitectura Funcional del Sistema.
Fuente: Propia.

3.7. Pilas de Iteración

Las pilas de iteración nos ayudan a cumplir los compromisos adquiridos en las historias de usuario. A continuación se presenta la recopilación de cada iteración que fue necesaria para desarrollar la aplicación.

3.7.1. Pila de Iteración I

A continuación se muestra la Iteración con las historias correspondientes a su desarrollo.

PILA ITERACIÓN: 1			
ID. HISTORIA	NOMBRE HISTORIA	ESTIMACIÓN (Días)	PRIORIDAD
1	Requerimientos para uso del Parqueadero Sismert	18	ALTA
2	Repositorio de Datos SISMERT	10	ALTA
3	Seguridad y Control de Acceso	4	ALTA
4	Contenido del SMS para uso del Parqueadero	4	MEDIA
5	Registro de vehículos	4	ALTA

Tabla 3. 20: Pila Iteración 1

Fuente: Propia

3.7.2. Pila de Iteración II

PILA ITERACIÓN: 2			
ID. HISTORIA	NOMBRE HISTORIA	ESTIMACIÓN (Días)	PRIORIDAD
5	Registro de vehículos	4	ALTA
7	Compras de Saldo, Información Registros	8	ALTA

Tabla 3. 21: Pila Iteración 2

Fuente: Propia

3.7.3. Pila de Iteración III

PILA ITERACIÓN: 3			
ID. HISTORIA	NOMBRE HISTORIA	ESTIMACIÓN (Días)	PRIORIDAD
8	Cambiar Lugar de Estacionamiento	5	MEDIO

PILA ITERACIÓN: 3			
ID. HISTORIA	NOMBRE HISTORIA	ESTIMACIÓN (Días)	PRIORIDAD
9	Notificaciones Inspector y Ciudadano	7	ALTA

Tabla 3. 22: Pila Iteración 3

Fuente: Propia

3.7.4. Pila de Iteración IV

PILA ITERACIÓN: 4			
ID. HISTORIA	NOMBRE HISTORIA	ESTIMACIÓN (Días)	PRIORIDAD
6	Establecer Identificación de Sitios para el Parqueadero	14	MEDIO

Tabla 3. 23: Pila Iteración 4

Fuente: Propia

3.7.5. Pila de Iteración V

PILA ITERACIÓN: 5			
ID. HISTORIA	NOMBRE HISTORIA	ESTIMACIÓN (Días)	PRIORIDAD
5	Registro de vehículos	5	ALTA

Tabla 3. 24: Pila Iteración 5

Fuente: Propia

3.7.6. Pila de Iteración VI

PILA ITERACIÓN: 6			
ID. HISTORIA	NOMBRE HISTORIA	ESTIMACIÓN (Días)	PRIORIDAD
10	Plantear nueva solución de notificación Hacia el inspector	3	ALTA
11	Creación Aplicación Móvil y Seguridad.	10	ALTA
12	Consulta Tiempo Parqueo y Notificaciones.	9	ALTA
13	Registro y Compras de Saldo en Aplicación Móvil.	15	ALTA

Tabla 3. 25: Pila Iteración 6

Fuente: Propia

3.7.7. Pila de Iteración VII

PILA ITERACIÓN: 7			
ID. HISTORIA	NOMBRE HISTORIA	ESTIMACIÓN (Días)	PRIORIDAD
14	Recaudación de Compras de Saldo SISMERT	5	ALTA

Tabla 3. 26: Pila Iteración 7

Fuente: Propia

3.7.8. Pila de Iteración VIII

PILA ITERACIÓN: 8			
ID. HISTORIA	NOMBRE HISTORIA	ESTIMACIÓN (Días)	PRIORIDAD
15	Servicios en la web para el Ciudadano	19	ALTA
16	Recargas SISMERT en Línea	3	BAJA

Tabla 3. 27: Pila Iteración 8

Fuente: Propia

3.7.9. Pila de Iteración IX

PILA ITERACIÓN: 9			
ID. HISTORIA	NOMBRE HISTORIA	ESTIMACIÓN (Días)	PRIORIDAD
7	Compras de Saldo, Información Registros	4	ALTA
14	Recaudación de Compras de Saldo SISMERT	6	ALTA

Tabla 3. 28: Pila Iteración 9

Fuente: Propia

3.7.10. Pila de Iteración X

PILA ITERACIÓN: 10			
ID. HISTORIA	NOMBRE HISTORIA	ESTIMACIÓN (Días)	PRIORIDAD
9	Notificaciones Inspector y Ciudadano	6	ALTA

Tabla 3. 29: Pila Iteración 10

Fuente: Propia

3.8. Descripción de Iteraciones

En las iteraciones se encuentran cada una de las tareas que se debe completar para cumplir con el desarrollo de un proyecto de desarrollo de software, estas tareas son las pruebas de aceptación que se adquiere en las historias de usuario como requisitos para tener el enfoque de la aplicación.

El desarrollo de la aplicación por Iteración se presenta a continuación, conjuntamente con las tareas que se desarrolla en cada Historia de usuario.

3.8.1. Iteración I

La iteración presenta el análisis de requisitos para el desarrollo de la aplicación, herramientas de desarrollo de software, y capacitación de las mismas.

Además cuenta con la realización del esquema de Base de Datos y el análisis e integración de acceso y seguridades a la base de Datos centralizada del GAD-I, realizar pruebas de su correcta funcionalidad, para registrar nueva aplicación y menus que posee, y posteriormente desarrollar interfaces graficas y funcionales de registros en esta aplicación.

PILA DE ITERCIÓN I				03/05/2012	04/05/2012	07/05/2012	08/05/2012
ID	Tareas	Tipo	Estado	Esfuerzo estimado			
3 Seguridad y Control de Acceso							
1	Analizar el esquema actual de seguridad que maneja la Dirección TIC	ANÁLISIS	TERMINADA	4h			
2	Analizar los tipos de usuarios que alberga el sistema actual	ANÁLISIS	TERMINADA		1h		
3	Registrar los Menus de la aplicación y publicar esta en el sistema Integrado del GAD-I	ANÁLISIS Y CODIFICACIÓN	TERMINADA	3h	1h	1h	
4	Comprobar si la aplicación esta publicada con los respectivos permisos asignados.	PRUEBA Y CODIFICACIÓN	TERMINADA				2h

Tabla 3. 32: Iteración 1-Historia de usuario 3

Fuente: Propia

PILA DE ITERACIÓN: I				24/05/2012	25/05/2012	28/05/2012	29/05/2012
ID	Tareas	Tipo	Estado	Esfuerzo estimado			
4 Contenido del SMS para uso del Parquadero							
1	Analizar posibles contenidos para SMS.	ANÁLISIS, REUNIÓN	TERMINADA	8h			
2	Verificar alternativa de Identificación de un vehiculo para Contenido SMS.	ANÁLISIS	TERMINADA		3h		
3	Identificar sintaxis que contendrá el SMS para el tiempo de uso del parquadero	ANÁLISIS	TERMINADA		3h		
4	Establecer identificación de SMS para Parqueo(P) en el contenido SMS	ANÁLISIS	TERMINADA			4h	
5	Verificar y establecer las distintas formas que el usuario podrá escribir en el contenido SMS.	ANÁLISIS	TERMINADA			4h	
6	Rediseñar Base de Datos para formar Contenidos SMS válidos.	ANÁLISIS, CODIFICACIÓN	TERMINADA				6h
7	Analizar contenido final de un SMS para el proceso de respuestas hacia el usuario.	ANÁLISIS	EN CURSO				2h

Tabla 3. 33: Iteración 1-Historia de usuario 4

Fuente: Propia

PILA DE ITERACIÓN: I				30/05/2012	31/05/2012	01/06/2012	04/06/2012
ID	Tareas	Tipo	Estado	Esfuerzo estimado			
5 Registro de Vehículos							
1	Crear la clase para el Controlador de Registro de vehículos.	CODIFICACIÓN	TERMINADA	2h			
2	Crear clase para modelo y Store del registro y reporte de vehículos.	CODIFICACIÓN	TERMINADA	1h			
3	Diseño GUI para formulario Registro de vehículos	PROTOTIPADO	TERMINADA	3h			
4	Implementar acciones de consulta de teléfonos registrados en el lado del servidor.	CODIFICACIÓN	TERMINADA	2h			
5	Implementar llamadas de Base Datos desde la UI y comprobar si el teléfono es válido.	ANÁLISIS,CODIFICACIÓN	TERMINADA		2h		
6	Codificar las respectivas acciones en la clase manager para registro de un vehículo.	CODIFICACIÓN	TERMINADA		3h		
7	Rediseño de la GUI para Registro de vehículos.	PROTOTIPADO	TERMINADA		3h		
8	Implementar acciones en el cliente para envío de datos al servidor.	CODIFICACIÓN	TERMINADA			4h	
9	Mostrar los resultados de éxito o fracaso al realizar el registro de vehículos.	ANÁLISIS,PRUEBAS	TERMINADA			4h	
10	Diseñar GUI de un usuario con sus respectivos vehículos registrados.	PROTOTIPADO	TERMINADA				6h
11	Realizar llamadas al Servidor para comprobación de un teléfono válido.	CODIFICACIÓN	EN CURSO				
12	Mostrar la información de los vehículos registrados dependiendo de un teléfono válido.	ANÁLISIS,CODIFICACIÓN	EN CURSO				
13	Crear la acción de eliminación de un vehículo en el lado del servidor y cliente.	CODIFICACIÓN	EN CURSO				
14	Mostrar los resultados después de la acción realizada por el usuario.	ANÁLISIS,PRUEBAS	EN CURSO				

Tabla 3. 34: Iteración 1-Historia de usuario 5

Fuente: Propia

3.8.2. Iteración II

En la presente Iteración se continúa con el registro de vehículos pendiente de la anterior iteración. Además se desarrolla el servicio de compras o recargas de saldo SISMERT, con la presentación de datos de los registros creados.

PILA DE ITERACIÓN: II				05/06/2012	06/06/2012	07/06/2012
ID	Tareas	Tipo	Estado	Esfuerzo estimado		
1 Registro de Vehículos						
1	Realizar llamadas al Servidor para comprobación de un teléfono válido.	CODIFICACIÓN	TERMINADA	6h		
2	Mostrar la información de los vehículos registrados dependiendo de un teléfono válido.	ANÁLISIS,CODIFICACIÓN	TERMINADA	2h	6h	
3	Crear la acción de eliminación de un vehículo en el lado del servidor y cliente.	CODIFICACIÓN	TERMINADA		2h	
4	Mostrar los resultados después de la acción realizada por el usuario.	ANÁLISIS,PRUEBAS	TERMINADA			4h

Tabla 3. 35: Iteración 2-Historia de usuario 5

Fuente: Propia

PILA DE ITERACIÓN: II				07/06/2012	08/06/2012	11/06/2012	12/06/2012	13/06/2012	14/06/2012	15/06/2012	18/06/2012
ID	Tareas	Tipo	Estado	Esfuerzo estimado							
2 Compras de Saldo, Información Registros											
1	Crear la clase para el Controlador de Compras de Saldo.	CODIFICACIÓN	TERMINADA	1h							
2	Crear clase para modelo y Store del para información de registro de Teléfonos.	CODIFICACIÓN	TERMINADA	1h							
3	Diseño GUI para formulario de Compras de Saldo.	PROTOTIPO, CODIFICACIÓN	TERMINADA	2h							
4	Implementar funciones para compras de Saldo en las clases Manager y Logic y acciones Correspondientes.	CODIFICACIÓN	TERMINADA		6h	6h					
5	Implementar llamadas de Base Datos desde la UI y comprobar si el teléfono es esta registrado para la recarga de saldo.	CODIFICACIÓN	TERMINADA		2h	2h					
6	Implementar acciones en el cliente para envío de datos de la recarga a realizarse.	CODIFICACIÓN	TERMINADA				2h				
7	Realizar las pruebas de envío y recepción de los datos de una recarga.	CODIFICACIÓN	TERMINADA				4h				
8	Corregir datos erróneos y mostrar resultados de las recargas de saldo.	PRUEBAS, CODIFICACIÓN	TERMINADA				2h				
9	Crear la clase para el Controlador de Consultas de Saldo.	CODIFICACIÓN	TERMINADA					1h			
10	Diseñar GUI para la consulta de saldos.	PROTOTIPADO, CODIFICACIÓN	TERMINADA					2h			
11	Realizar llamadas al Servidor para comprobación de un teléfono válido para consultar Saldo.	CODIFICACIÓN	TERMINADA					1h			

PILA DE ITERACIÓN: II				07/06/2012	08/06/2012	11/06/2012	12/06/2012	13/06/2012	14/06/2012	15/06/2012	18/06/2012
12	Crear funciones para consultar el saldo disponible de un teléfono, en Manager y Logic.	CODIFICACIÓN	TERMINADA					4h			
13	Crear acciones para la consulta de saldos e interacción con el cliente(Controlador)	CODIFICACIÓN	TERMINADA						2h		
14	Crear las respectivas funciones en el controlador para realizar las llamadas hacia las acciones del servidor.	CODIFICACIÓN	TERMINADA						3h		
15	Mostrar los resultados y realizar pruebas de su funcionalidad.	PRUEBAS, CODIFICACIÓN	TERMINADA						3h		
16	Crear la clase para el Controlador de Reporte de información de registro de teléfonos.	CODIFICACIÓN	TERMINADA							1h	
17	Diseñar GUI para la consulta de Registro de Teléfonos.	PROTOTIPADO, CODIFICACIÓN	TERMINADA							3h	
18	Crear funciones para consultar los registros de teléfonos, en Manager y Logic y sus respectivas acciones.	CODIFICACIÓN	TERMINADA							4h	1h
19	Crear las respectivas funciones en el controlador para realizar las llamadas hacia las acciones del servidor.	CODIFICACIÓN	TERMINADA								4h
20	Mostrar los resultados y realizar pruebas de su funcionalidad.	PRUEBAS, CODIFICACIÓN	TERMINADA								3h
21	Diseñar reporte para Uso del Parquadero por año en IReport	PROTOTIPO, CODIFICACIÓN	EN CURSO								
22	Integrar el reporte Uso Parqueo en la aplicación.	CODIFICACIÓN	EN CURSO								
23	Diseñar reporte para registro de Teléfonos en IReport.	PROTOTIPO, CODIFICACIÓN	EN CURSO								
24	Integrar el reporte Registro Teléfonos en la aplicación.	CODIFICACIÓN	EN CURSO								

Tabla 3. 36: Iteración 2-Historia de usuario 7

Fuente: Propia

3.8.3. Iteración III

La Iteración se desarrolla para el análisis de envío de notificaciones al ciudadano y hacia el Inspector cuando se inicia el servicio de parqueo, y cuando un vehículo cambia de sitio y tiene todavía tiempo disponible para utilizar una zona regulada del SISMERT. También crear interfaz funcional para administración de sitios de Inspección para los Inspectores notificándoles a sus números de teléfono por medio de un SMS.

PILA DE ITERACIÓN: III				19/06/2012	20/06/2012	21/06/2012	22/06/2012	25/06/2012
ID	Tareas	Tipo	Estado	Esfuerzo estimado				
1 Cambiar Lugar de Estacionamiento								
1	Analizar la manera de que el ciudadano pueda cambiarse de sitio de parqueo.	ANÁLISIS	TERMINADA	4h				
2	Analizar el contenido del SMS que el ciudadano envía para cambio de sitio de parqueo.	ANÁLISIS	TERMINADA	4h				
3	Analizar el contenido del SMS para notificar al ciudadano e Inspector sobre el parqueo.	ANÁLISIS	TERMINADA		2h			
4	Analizar la factibilidad del cambio de sitio de parqueo.	ANÁLISIS	TERMINADA		4h			
5	Crear un controlador para ingresar y editar el número de teléfono de los Inspectores.	CODIFICACIÓN	TERMINADA		2h			
6	Crear clase para modelo y Store de los Inspectores.	CODIFICACIÓN	TERMINADA			1h		
7	Diseño GUI para formulario Ingreso y edición de teléfono de Inspectores.	PROTOTIPO,CODIFICACIÓN	TERMINADA			4h		
8	Codificar las respectivas acciones en la clase manager para ingreso y edición de un teléfono del Inspector.	CODIFICACIÓN	TERMINADA			3h	6h	
9	Implementar acciones en el cliente para envío de datos al servidor.	CODIFICACIÓN	TERMINADA				2h	2h
10	Mostrar los resultados de éxito o fracaso al realizar el registro y edición de datos.	ANÁLISIS,PRUEBAS	TERMINADA					4h

Tabla 3. 37: Iteración 3-Historia de usuario 8

Fuente: Propia

PILA DE ITERACIÓN: III				25/06/2012	26/06/2012	27/06/2012	28/06/2012	29/06/2012	02/07/2012	03/07/2012
2 Notificaciones Inspector y Ciudadano										
1	Analizar la forma de notificación hacia el inspector.	ANÁLISIS	TERMINADA	2h						
2	Analizar el contenido del SMS hacia el inspector cuando se inicia un parqueo.	ANÁLISIS	TERMINADA		2h					
3	Establecer el contenido de la notificación cuando el ciudadano se parquea.	ANÁLISIS	TERMINADA		2h					
4	Crear un controlador para Asignar y editar un Sitio de inspección (I) para el Inspector	CODIFICACIÓN	TERMINADA		1h					
5	Crear clase para modelo y Store de Sitios de Inspección (I).	CODIFICACIÓN	TERMINADA		2h					

PILA DE ITERACIÓN: III			25/06/2012	26/06/2012	27/06/2012	28/06/2012	29/06/2012	02/07/2012	03/07/2012
6	Diseño GUI para formulario Asignación y edición de Sitios de Inspección.	PROTOTIPO,CODIFICACIÓN	TERMINADA			6h	2h		
7	Codificar las respectivas acciones en la clase manager para asignación y edición de un sitio de Inspección.	CODIFICACIÓN	TERMINADA			2h	6h	6h	4h
8	Implementar acciones en el cliente para envío de datos al servidor.	CODIFICACIÓN	TERMINADA					2h	2h
9	Mostrar los resultados de éxito o fracaso al asignar y editar un sitio de Inspección.	ANÁLISIS,PRUEBAS	TERMINADA						4h

Tabla 3. 38: Iteración 3-Historia de usuario 9

Fuente: Propia

3.8.4. Iteración IV

La Iteración se desarrolla para identificar las calles de las plazas del SISMERT, para después establecer los sitios del uso del parqueadero asignándole un código de identificación para luego ser asignados dentro de una zona y así controlar el uso del parqueadero. Además diseñar interfaz funcional para su administración.

				Julio del 2012													
PILA DE ITERACIÓN: IV				04	05	06	09	10	11	12	13	16	17	18	19	20	23
ID	Tareas	Tipo	Estado	Esfuerzo estimado													
1 Establecer Identificación de Sitios para el Parqueadero																	
1	Identificar calles de las plazas del SISMERT.	ANÁLISIS	TERMINADA	2h													

Sistema de Control de Parqueo tarifado SISMERT

				Julio del 2012													
PILA DE ITERACIÓN: IV				04	05	06	09	10	11	12	13	16	17	18	19	20	23
2	Analizar y Crear código de identificación de un sitio de parqueo.	ANÁLISIS	TERMINADA	6h													
3	Rediseñar Base de Datos para la implementación de Sitios de Parqueo.	CODIFICACIÓN	TERMINADA		4h												
4	Ingresar datos de prueba, formando sitios de parqueo.	CODIFICACIÓN	TERMINADA		4h												
5	Agrupar los sitios por zonas para el control del parqueo.	ANÁLISIS, CODIFICACIÓN	TERMINADA			3h											
6	Implementar estructura de datos y realizar pruebas ingresando datos.	PRUEBAS, CODIFICACIÓN	TERMINADA			4h											
7	Crear controlador para ingresar y editar un Sitios de parqueo.	CODIFICACIÓN	TERMINADA				1h										
8	Crear clase para modelo y Store.	CODIFICACIÓN	TERMINADA				1h										
9	Diseño GUI para formulario Ingreso y edición de sitios de parqueo.	PROTOTIPO, CODIFICACIÓN	TERMINADA				6h										
10	Codificar las respectivas acciones en la clase manager para ingreso y edición de un sitio.	CODIFICACIÓN	TERMINADA					8h	8h								
11	Implementar acciones en el cliente para envío de datos al servidor.	CODIFICACIÓN	TERMINADA							8h							

				Julio del 2012													
PILA DE ITERACIÓN: IV				04	05	06	09	10	11	12	13	16	17	18	19	20	23
12	Mostrar los resultados de éxito o fracaso al realizar el ingreso y edición de datos.	ANÁLISIS,PRUEBAS	TERMINADA								2h						
13	Crear controlador para ingresar zonas y asignación de sitios.	CODIFICACIÓN	TERMINADA								1h						
14	Crear clase para modelo y Store.	CODIFICACIÓN	TERMINADA								1h						
15	Diseño GUI para formulario Ingreso y asignación de sitios en una zona.	PROTOTIPO,CODIFICACIÓN	TERMINADA								4h	2h					
16	Codificar las respectivas acciones en la clase manager para asignación de sitios en una zona.	CODIFICACIÓN	TERMINADA									6h	8h				
17	Implementar acciones en el cliente para envío de datos al servidor.	CODIFICACIÓN	TERMINADA											8h			
18	Mostrar los resultados de éxito o fracaso al realizar la asignación en una zona.	ANÁLISIS,PRUEBAS	TERMINADA												2h		
19	Crear controlador para editar asignación de zonas.	CODIFICACIÓN	TERMINADA												2h		
20	Crear clase para modelo y Store.	CODIFICACIÓN	TERMINADA												1h		
21	Diseño GUI para formulario de edición.	PROTOTIPO,CODIFICACIÓN	TERMINADA												3h		

				Julio del 2012													
PILA DE ITERACIÓN: IV				04	05	06	09	10	11	12	13	16	17	18	19	20	23
22	Codificar las respectivas acciones en la clase manager para asignación de sitios en una zona.	CODIFICACIÓN	TERMINADA													8h	
23	Implementar acciones en el cliente para envío de datos al servidor.	CODIFICACIÓN	TERMINADA														4h
24	Realizar las pruebas y corrección de edición de zonas.	ANÁLISIS, PRUEBAS	TERMINADA														4h

Tabla 3. 39: Iteración 4-Historia de usuario 6

Fuente: Propia

3.8.5. Iteración V

La Iteración se desarrolla para obtener datos de los vehículos existentes tras haber pasado por la certificación de SISMERT, estos datos serán reutilizados para el registro de vehículos en un número de teléfono para el servicio del parqueadero, además creara la opción de ingreso de vehículos y mostrar los existentes.

PILA DE ITERACIÓN: V				24/07/2012	25/07/2012	26/07/2012	27/07/2012	30/07/2012
ID	Tareas	Tipo	Estado	Esfuerzo estimado				
1 Registro de vehículos								
1	Revisar y analizar vehículos certificados en SISMERT.	REVISIÓN, ANÁLISIS	TERMINADA	4h				
2	Crear función para tomar datos de vehículos y su propietario.	CODIFICACIÓN	TERMINADA	4h	2h			
3	Pruebas y Revisión de información de datos de vehículos.	PRUEBAS	TERMINADA		4h			

PILA DE ITERACIÓN: V				24/07/2012	25/07/2012	26/07/2012	27/07/2012	30/07/2012
4	Registrar menu de revisión y registro de nuevos vehículos certificados.	CODIFICACIÓN	TERMINADA		2h			
5	Crear controlador para ingreso y actualización de datos de vehículos.	CODIFICACIÓN	TERMINADA			1h		
6	Crear modelo y Store para informe de vehículos.	CODIFICACIÓN	TERMINADA			2h		
7	Implementar funciones que manipulen el ingreso de nuevos datos.	CODIFICACIÓN	TERMINADA			5h	5h	
8	Implementar acciones en el servidor y cliente para ingreso y recepción de datos.	CODIFICACIÓN	TERMINADA				3h	4h
9	Realizar pruebas y verificar la funcionalidad.	PRUEBAS, CODIFICACIÓN	TERMINADA					4h

Tabla 3. 40: Iteración 5-Historia de usuario 5

3.8.6. Iteración VI

La Iteración se desarrolla para analizar la forma de revisar si un vehículo tiene tiempo de parqueo, también la creación de una aplicación móvil con su respectiva seguridad para el control de acceso de los usuarios. Además se desarrolla las interfaces de usuario funcionales para el control del parqueo y registros, notificaciones y la alternativa de iniciar el servicio del parqueadero desde la aplicación móvil.

PILA DE ITERACIÓN: VI				04/09/2012	05/09/2012	06/09/2012
ID	Tareas	Tipo	Estado	Esfuerzo estimado		
1 Plantear nueva solución de notificación Hacia el inspector						
1	Analizar la forma de para revisar si un vehículo tiene tiempo de parqueo.	ANÁLISIS	TERMINADA	6h		
2	Analizar y verificar si es posible crear una aplicación con el framework Extjs4.	ANÁLISIS	TERMINADA	2h	6h	
3	Analizar la posibilidad de crear una aplicación móvil para el control de estacionamiento.	ANÁLISIS	TERMINADA		2h	4h
4	Analizar la manera de integrar la nueva aplicación con la actual.	ANÁLISIS	TERMINADA			4h

Tabla 3. 41: Iteración 6-Historia de usuario 10

Fuente: Propia

PILA DE ITERACIÓN: VI				07/09/2012	10/09/2012	11/09/2012	12/09/2012	13/09/2012	14/09/2012	17/09/2012	18/09/2012	19/09/2012	20/09/2012
ID	Tareas	Tipo	Estado	Esfuerzo estimado									
2 Creación Aplicación Móvil y Seguridad.													
1	Analizar el Framework de Sencha Touch para crear aplicación móvil.	ANÁLISIS	TERMINADA	8h	8h								
2	Revisar el tipo de componentes que ofrece el framework.	ANÁLISIS, CODIFICACIÓN	TERMINADA			6h							
3	Realizar pruebas para comprobación de la funcionalidad del framework.	CODIFICACIÓN	TERMINADA			2h							
4	Analizar la aplicación actual para reutilizar las fuentes de acceso y seguridad a la nueva aplicación.	ANÁLISIS, CODIFICACIÓN	TERMINADA				4h						
5	Crear la nueva aplicación móvil Genérica para varias sub-aplicaciones.	PROTOTIPO, CODIFICACIÓN	TERMINADA				4h						
6	Integrar la seguridad de acceso a la aplicación con la respectiva creación del Core en el cliente.	CODIFICACIÓN	TERMINADA					8h	8h	8h	8h		
7	Realizar pruebas de acceso y corrección de posibles errores en la aplicación móvil.	PRUEBAS, CODIFICACIÓN	TERMINADA									8h	8h

Tabla 3. 42: Iteración 6-Historia de usuario 11

Fuente: Propia

PILA DE ITERACIÓN: VI				21/09/2012	24/09/2012	25/09/2012	26/09/2012	27/09/2012	01/10/2012	02/10/2012	03/10/2012	04/10/2012
ID	Tareas	Tipo	Estado	Esfuerzo estimado								
3 Consulta Tiempo Parqueo y Notificaciones en Aplicación Móvil.												
1	Crear y configurar archivo de aplicación para SISMERT.	CODIFICACIÓN	TERMINADA	6h	4h							
2	Crear controlador genérico para los servicios que se ofrecerá en la aplicación móvil.	CODIFICACIÓN	TERMINADA		4h							
3	Diseñar formularios para consultas de tiempo de Parqueo.	PROTOTIPADO, CODIFICACIÓN	TERMINADA			8h	8h					
4	Diseñar formulario para ingresar una notificación.	PROTOTIPADO, CODIFICACIÓN	TERMINADA					6h				
5	Crear función o funciones para ejecutar la acción de consultas de tiempo.	CODIFICACIÓN	TERMINADA					2h	8h	2h		
6	Crear funciones en el controlador genérico para acceso al servidor de consultas de tiempo.	CODIFICACIÓN	TERMINADA							4h		
7	Crear función o funciones para ejecutar la acción de ingreso de notificaciones.	CODIFICACIÓN	TERMINADA							2h	6h	
8	Crear funciones en el controlador genérico para acceso al servidor de ingreso de notificaciones.	CODIFICACIÓN	TERMINADA								2h	
9	Realizar pruebas y corrección de errores en la revisión de la funcionalidad.	PRUEBAS, CODIFICACIÓN	TERMINADA									8h

Tabla 3. 43: Iteración 6-Historia de usuario 12

Fuente: Propia

Sistema de Control de Parqueo tarifado SISMERT

				Septiembre del 2012														
PILA DE ITERACIÓN: VI				05	08	09	10	11	15	16	17	18	19	22	23	24	25	26
ID	Tareas	Tipo	Estado	Esfuerzo estimado														
4 Registro y Compras de Saldo en Aplicación Móvil.																		
1	Diseñar formulario para registro de teléfonos.	PROTOTIPADO, CODIFICACIÓN	TERMINADA	8h	3h													
2	Diseñar formulario para registro de vehículos.	PROTOTIPADO, CODIFICACIÓN	TERMINADA		5h	4h												
3	Diseñar formulario para realizar recargas de Saldo.	PROTOTIPADO, CODIFICACIÓN	TERMINADA			4h	4h											
4	Diseñar formulario para iniciar servicio de Parqueo.	PROTOTIPADO, CODIFICACIÓN	TERMINADA				4h	2h										
5	Crear función o funciones para ejecutar la acción de registro de nuevos teléfonos.	CODIFICACIÓN	TERMINADA					6h	6h									
6	Crear funciones en el controlador genérico para acceder al servidor e ingresar un registro de teléfono.	CODIFICACIÓN	TERMINADA						2h	4h								
7	Crear función o funciones para ejecutar la acción de registro de nuevos vehículos.	CODIFICACIÓN	TERMINADA							4h	4h							
8	Crear funciones en el controlador genérico para acceder al servidor e ingresar un nuevo vehículo.	CODIFICACIÓN	TERMINADA								4h	4h						
9	Crear función o funciones para ejecutar la acción de recargas de saldo SISMERT.	CODIFICACIÓN	TERMINADA									4h	4h					
10	Crear funciones en el controlador genérico para acceder al servidor y realizar una recarga.	CODIFICACIÓN	TERMINADA										4h	4h				
11	Crear función o funciones para ejecutar la acción de iniciar el servicio de parqueo.	CODIFICACIÓN	TERMINADA											4h	4h			

				Septiembre del 2012														
PILA DE ITERACIÓN: VI				05	08	09	10	11	15	16	17	18	19	22	23	24	25	26
12	Crear funciones en el controlador genérico para acceder al servidor e iniciar el servicio de parqueo.	CODIFICACIÓN	TERMINADA												4h	4h		
13	Realizar pruebas y corrección de errores en la revisión de la funcionalidad.	PRUEBAS, CODIFICACIÓN	TERMINADA													4h	8h	6h

Tabla 3. 44: Iteración 6-Historia de usuario 13

Fuente: Propia

3.8.7. Iteración VII

La Iteración se desarrolla para registrar las recaudaciones de las recargas por el responsable que ha acreditado el saldo en un número de teléfono, ayudando al control interno de las recaudaciones obtenidas en el día.

PILA DE ITERACIÓN: VII				30/10/2012	31/10/2012	05/11/2012	06/11/2012	07/11/2012
ID	Tareas	Tipo	Estado	Esfuerzo estimado				
1 Recaudación de Compras de Saldo SISMERT								
1	Registrar sub-menu de revisión de recaudaciones.	CODIFICACIÓN	TERMINADA	1h				
2	Analizar y Rediseñar esquema de base de datos para controlar el estado de las recaudaciones.	ANÁLISIS, CODIFICACIÓN	TERMINADA	4h				
3	Crear controlador para la revisión de recaudaciones y reportes.	CODIFICACIÓN	TERMINADA	1h				
4	Crear modelo y Store de las recaudaciones.	CODIFICACIÓN	TERMINADA	1h				
5	Crear la vista para mostrar las recaudaciones realizadas.	PROTOTIPADO, CODIFICACIÓN	TERMINADA		4h			
6	Crear formulario para obtener lo recaudado por responsable.	PROTOTIPADO, CODIFICACIÓN	TERMINADA		2h			

PILA DE ITERACIÓN: VII				30/10/2012	31/10/2012	05/11/2012	06/11/2012	07/11/2012
7	Implementar funciones en el manager y acciones para llamadas desde el cliente.	CODIFICACIÓN	TERMINADA		2h	8h	4h	
8	Implementar acciones en el controlador del cliente para realizar peticiones al servidor.	CODIFICACIÓN	TERMINADA				4h	
9	Revisar, corregir y Probar la funcionalidad de lo implementado.	PRUEBAS, CODIFICACIÓN	TERMINADA					6h
10	Diseñar reporte para recaudaciones de los inspectores por día.	PROTOTIPO, CODIFICACIÓN	EN CURSO					
11	Integrar el reporte de recaudaciones de inspectores.	CODIFICACIÓN	EN CURSO					
12	Diseñar reportes varios sobre las recaudaciones.	PROTOTIPO, CODIFICACIÓN	EN CURSO					
13	Integrar los reportes de las recaudaciones.	CODIFICACIÓN	EN CURSO					

Tabla 3. 45: Iteración 7-Historia de usuario 14

Fuente: Propia

3.8.8. Iteración VIII

Iteración creada para el Análisis y creación de una aplicación externa que será publicada en la web para ofrecer servicios al ciudadano del área de SISMERT, esta aplicación contiene la seguridad de acceso por cada usuario (ciudadano). Se tiene acceso al ingreso de registros y la verificación de uso del parqueadero ingresando un número de teléfono registrado. También la creación de un prototipo para las recargas en Línea.

				Noviembre 2012															Diciembre 2012			
PILA DE ITERACIÓN: VIII				08	09	12	13	14	15	16	19	20	21	22	23	26	27	28	29	30	03	04
ID	Tareas	Tipo	Estado	Esfuerzo estimado																		
1 Servicios en la web para el Ciudadano																						
1	Analizar la forma segura de publicar la aplicación en la web.	ANÁLISIS	TERMINADA	6h																		

Sistema de Control de Parqueo tarifado SISMERT

				Noviembre 2012																Diciembre 2012		
PILA DE ITERACIÓN: VIII				08	09	12	13	14	15	16	19	20	21	22	23	26	27	28	29	30	03	04
2	Crear una nueva aplicación similar a la aplicación interna, donde se publicaran servicios para el ciudadano.	CODIFICACIÓN	TERMINADA	2h	6h																	
3	Modificar el módulo de seguridad para el acceso de usuario Ciudadano.	CODIFICACIÓN	TERMINADA		2h	6h																
4	Analizar el tipo de seguridades al acceder a la aplicación, como el bloqueo de cuenta.	ANÁLISIS	TERMINADA			2h																
5	Analizar la forma de registro de un Usuario y restauración de clave de acceso.	ANÁLISIS	TERMINADA				4h															
6	Modificar estructura de datos de ciudadano para crear campos del control de seguridad de usuario.	CODIFICACIÓN	TERMINADA				4h															
7	Realizar pruebas a nivel de base de datos con los nuevos campos creados.	PRUEBAS, CODIFICACIÓN	TERMINADA					2h														
8	Codificar el acceso de usuario en módulo de seguridades de la aplicación.	CODIFICACIÓN	TERMINADA					6h														
9	Realizar pruebas de acceso a la aplicación.	PRUEBAS, CODIFICACIÓN	TERMINADA						8h													
10	Diseñar formulario para el registro de un nuevo usuario.	PROTOTIPO, CODIFICACIÓN	TERMINADA							1h												
11	Crear acciones en el cliente y servidor, para el registro de un nuevo usuario.	CODIFICACIÓN	TERMINADA							7h	6h											
12	Realizar pruebas del registro de usuario.	PROTOTIPO, CODIFICACIÓN	TERMINADA								2h	4h										

				Noviembre 2012																Diciembre 2012			
PILA DE ITERACIÓN: VIII				08	09	12	13	14	15	16	19	20	21	22	23	26	27	28	29	30	03	04	
13	Crear acciones en el cliente y servidor, para restaurar clave de usuario.	CODIFICACIÓN	TERMINADA									4h	8h										
14	Realizar pruebas de restauración de clave de acceso.	PRUEBAS, CODIFICACIÓN	TERMINADA											6h									
15	Registrar la aplicación web SISMERT para el ciudadano.	CODIFICACIÓN	TERMINADA											1h									
16	Registrar Menus de la nueva Aplicación.	CODIFICACIÓN	TERMINADA											1h									
17	Crear Controlador genérico para los servicios en la web del ciudadano.	CODIFICACIÓN	TERMINADA												2h								
18	Crear y analizar contenido de ventana principal para mostrar servicios de esta aplicación.	PROTOTIPO, ANÁLISIS, CODIFICACIÓN	TERMINADA												6h								
19	Acoplar Vistas de la aplicación interna (Registro teléfono, vehículo, consulta Saldo) en esta nueva aplicación del ciudadano.	CODIFICACIÓN	TERMINADA													8h	4h						
20	Reutilizar la capa de Manager y Logic de la aplicación interna para los servicios del ciudadano.	CODIFICACIÓN	TERMINADA														4h						
21	Crear acciones en el controlador genérico para las peticiones hacia el servidor.	CODIFICACIÓN	TERMINADA															8h	2h				
22	Crear ventana para consulta del uso del parqueadero.	PROTOTIPO, CODIFICACIÓN	TERMINADA																4h				

				Noviembre 2012															Diciembre 2012			
PILA DE ITERACIÓN: VIII				08	09	12	13	14	15	16	19	20	21	22	23	26	27	28	29	30	03	04
23	Crear acciones en el cliente y Servidor para revisar el uso del parqueadero.	CODIFICACIÓN	TERMINADA																2h			
24	Crear reporte PDF en Symfony para mostrar el uso del parqueadero.	CODIFICACIÓN	TERMINADA																	8h		
25	Realizar pruebas de funcionalidad, y corregir errores existentes.	PRUEBAS, CODIFICACIÓN	TERMINADA																		8h	6h

Tabla 3. 46: Iteración 8-Historia de usuario 15

Fuente: Propia

PILA DE ITERACIÓN: VIII				05/12/2012	06/12/2012	07/12/2012
ID	Tareas	Tipo	Estado	Esfuerzo estimado		
2 Recargas SISMERT en Línea						
1	Diseñar el formulario para realizar las compras en línea de recargas de saldo SISMERT.	PROTOTIPO, CODIFICACIÓN	TERMINADA	6h		
2	Crear acciones en el controlador genérico para mostrar la ventana de recargas en línea.	CODIFICACIÓN	TERMINADA		4h	
3	Revisar el contenido de la ventana para el envío de datos al proveedor de transacciones electrónicas.	ANÁLISIS, CODIFICACIÓN	TERMINADA		2h	6h

Tabla 3. 47: Iteración 8-Historia de usuario 16

Fuente: Propia

3.8.9. Iteración IX

Iteración desarrollada para el diseño y creación de los reportes presentados en esta aplicación como: Reporte de recaudaciones, Reportes estadísticos y de los registros de teléfonos en el sistema. Además la integración en la aplicación estándar del Control de Parqueo Tarifado.

PILA DE ITERACIÓN: IX				10/12/2012	11/12/2012	12/12/2012	13/12/2012
ID	Tareas	Tipo	Estado	Esfuerzo estimado			
1 Compras de Saldo, Información Registros							
1	Registrar Menus para Reportes	CODIFICACIÓN	TERMINADA	1h			
2	Diseñar reporte para Uso del Parquero por año en IReport	PROTOTIPO, CODIFICACIÓN	TERMINADA	6h			
3	Integrar el reporte Uso Parqueo en la aplicación.	CODIFICACIÓN	TERMINADA		3h		
4	Diseñar reporte para registro de Teléfonos en IReport.	PROTOTIPO, CODIFICACIÓN	TERMINADA			6h	
5	Integrar el reporte Registro Teléfonos en la aplicación.	CODIFICACIÓN	TERMINADA				3h

Tabla 3. 48: Iteración 9-Historia de usuario 7

Fuente: Propia

PILA DE ITERACIÓN: IX				14/12/2012	17/12/2012	18/12/2012	19/12/2012	20/12/2012	21/12/2012
ID	Tareas	Tipo	Estado	Esfuerzo estimado					
2 Recaudación de Compras de Saldo SISMERT									
1	Diseñar reporte para recaudaciones de los inspectores por día.	PROTOTIPO, CODIFICACIÓN	TERMINADA	6h					
2	Integrar el reporte de recaudaciones de inspectores.	CODIFICACIÓN	TERMINADA		4h				
3	Diseñar reportes varios sobre las recaudaciones.	PROTOTIPO, CODIFICACIÓN	TERMINADA			6h	4h		
4	Integrar los reportes de las recaudaciones.	CODIFICACIÓN	TERMINADA					4h	4h

Tabla 3. 49: Iteración 9-Historia de usuario 14

Fuente: Propia

3.8.10. Iteración X

La Iteración se desarrolla para implementar el servicio de notificaciones SMS hacia el ciudadano a través de un servicio web que accede a la aplicación de mensajería para posteriormente enviar un SMS cuando el usuario (ciudadano) hace uso del Sistema de Control de Parqueo Tarifado.

PILA DE ITERACIÓN: X				19/12/2012	20/12/2012	21/12/2012	07/01/2013	08/01/2013	09/01/2013
ID	Tareas	Tipo	Estado	Esfuerzo estimado					
1 Notificaciones Inspector y Ciudadano									
1	Crear servicio web para interpretar el servicio que solicita el usuario en un SMS.	CODIFICACIÓN	TERMINADA	4h	4h	4h			
2	Realizar pruebas de funcionalidad del servicio web.	PRUEBAS, CODIFICACIÓN	TERMINADA				4h		
3	Integrar el servicio web en la aplicación existente de envío de SMS.	CODIFICACIÓN	TERMINADA				2h		
4	Realizar pruebas de integración de recepción del SMS e interpretación del servicio solicitado.	PRUEBAS, CODIFICACIÓN	TERMINADA				2h		
5	Integrar el servicio web de envío de SMS al servicio de interpretación y recepción de peticiones.	CODIFICACIÓN	TERMINADA					4h	
6	Realizar pruebas de recepción y envío de SMS a la petición de un usuario.	PRUEBAS, CODIFICACIÓN	TERMINADA					4h	2h

Tabla 3. 50: Iteración 10-Historia de usuario 9

Fuente: Propia

3.9. Desarrollo de Historias de Usuario por Iteración.

Las Historias de Usuario contienen tareas que se desarrollan en las iteraciones de una aplicación. A continuación se presenta las tareas que se obtuvieron en las pruebas de aceptación que contienen las historias de usuario para el desarrollo de sistema del Control de Parqueo Tarifado.

3.9.1. Iteración I (Módulo de Registro y Compras de Saldo)

La Iteración presenta el desarrollo de las historias 1, 2, 3, 4, y 5 para fase inicial de análisis de requerimientos y desarrollo e implementación de la estructura de Datos para ingreso de registros. Además se desarrolla el control de acceso al Sistema, utilizando el esquema de administración de Sistemas del GAD-I.

También se realiza el análisis del contenido del contenido del SMS que el ciudadano debe enviar cuando desea utilizar una zona regula del SISMERT, estableciendo el modelo relacional con el que se controlara este servicio.

➤ Historia 2: Repositorio de Datos SISMERT

La Historia presenta 4 tareas, para el desarrollo de la estructura de datos y almacenamiento de los registros de teléfonos y vehículos como fase inicial del sistema, ingreso de datos de prueba para verificar su funcionalidad e integración a la Base de Datos Centralizada del GAD-I.

Tareas:

1. Diseño y modelado de Base de Datos.
2. Ingresar Datos de Prueba y verificar su funcionalidad.
3. Verificar estructura de BDD para su integración a la base actual GAD-I.
4. Integrar estructura de datos en esquema SISMERT.

A continuación se presenta el modelo relacional para el Módulo de Registros y Compras de Saldo

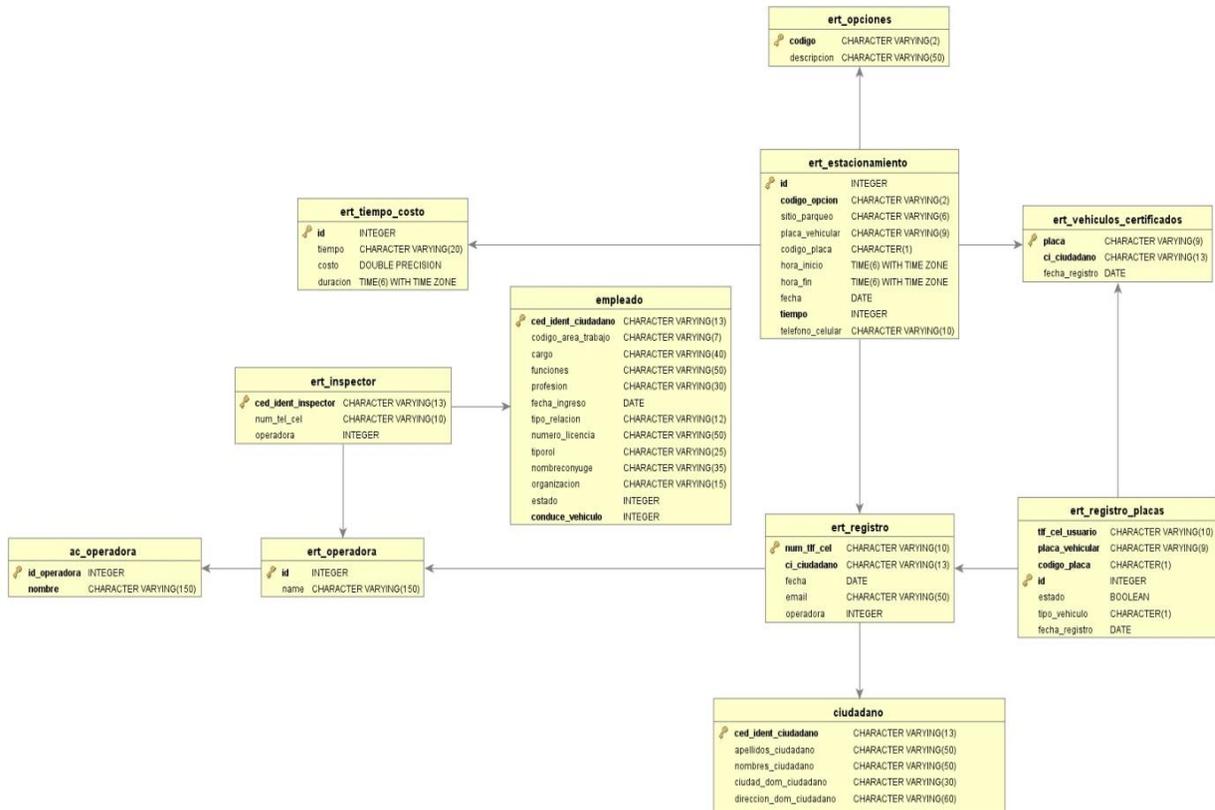


Figura 3. 3: Modelo Relación Registros y Contenido SMS para Parqueo. Fuente: Propia

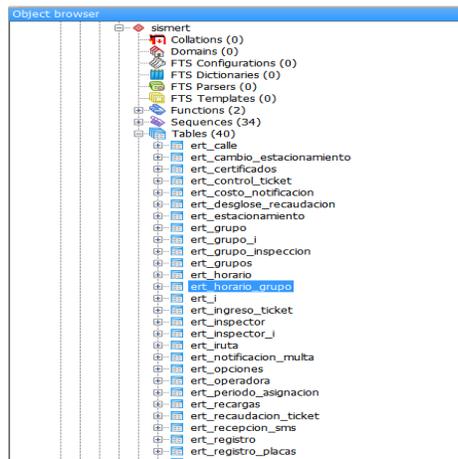


Figura 3. 4: Integración de estructura de Datos en Esquema SISMERT. Fuente: Propia

El modelo entidad relación para el desarrollo de aplicación se creó en el esquema existente del SISMERT, reutilizando la información de esta estructura de Datos.

En la figura anterior también se puede observar las tablas correspondientes al análisis realizado para el contenido del SMS, cuando el usuario (Ciudadano) inicia el servicio del parqueadero.

➤ Historia 3: Seguridad y Control de Acceso

Las Tareas 1, 2, 3 y 4 se desarrollan para el control de acceso a la aplicación, realizando el análisis necesario del esquema **adm** que es utilizado para la Administración de los sistemas del GAD-I.

Después de haber realizado este análisis se procede a registrar la aplicación y sus Menus.

Tareas:

1. Analizar el esquema actual de seguridad que maneja la Dirección TIC.
2. Analizar los tipos de usuarios que alberga el sistema actual.
3. Registrar los Menus de la aplicación y publicar esta en el sistema Integrado del GAD-I.
4. Comprobar si la aplicación esta publicada con los respectivos permisos asignados.

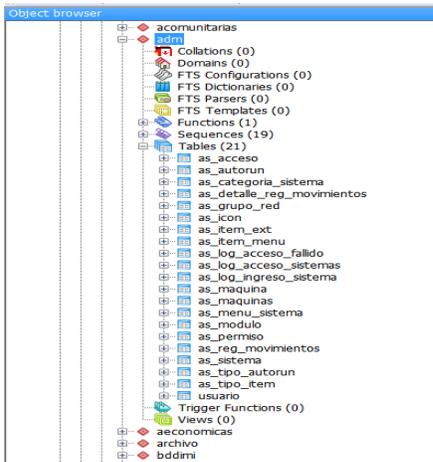


Figura 3. 5: Esquema de datos de Administración de Sistemas GAD-I.

Fuente: Propia

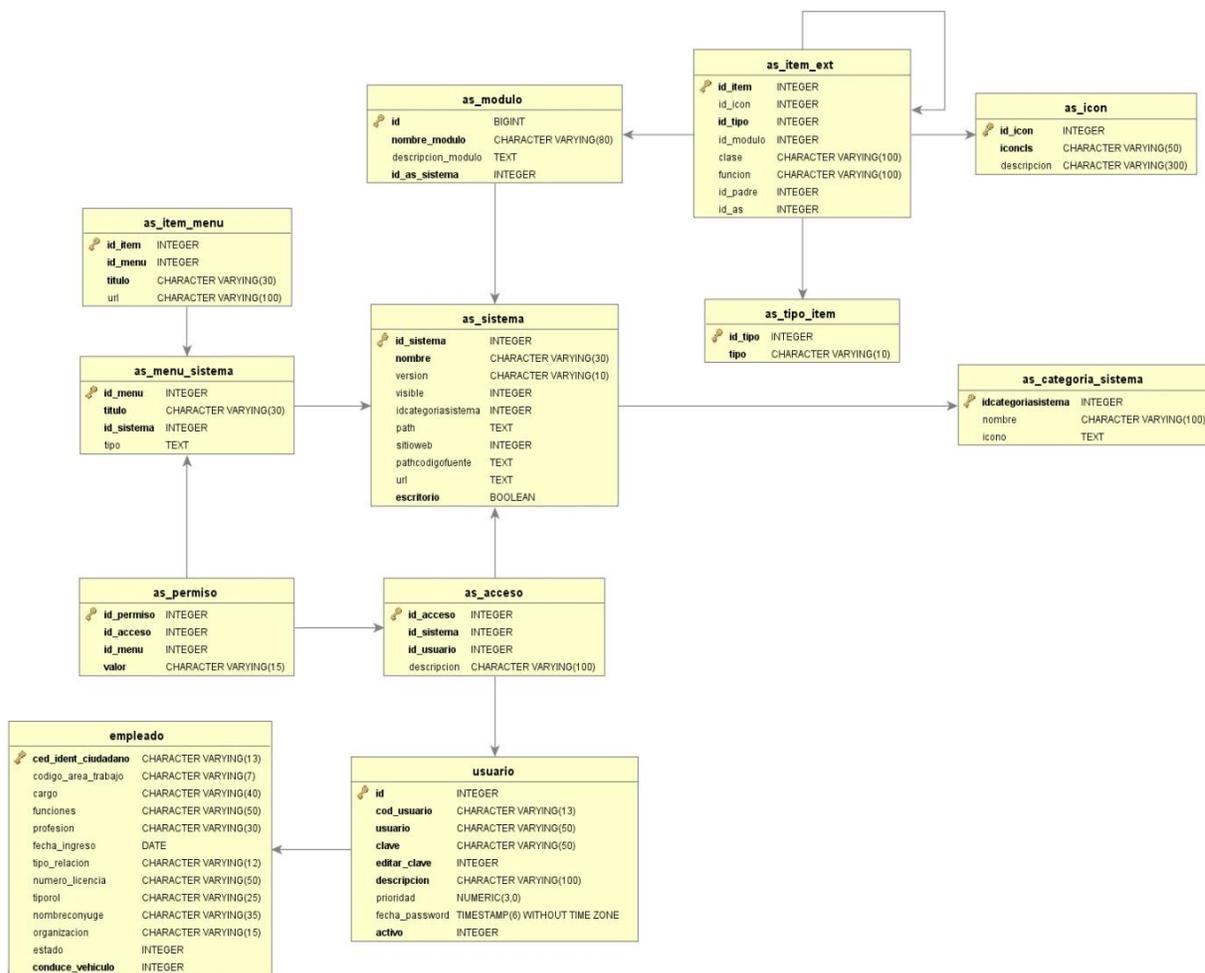


Tabla 3. 51: Modelo Relacional de Control de Acceso al Sistema
Fuente: Propia

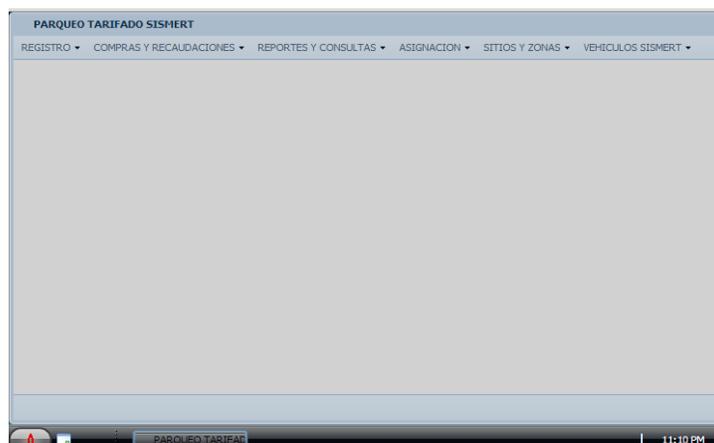


Figura 3. 6: Menus para Interfaz de control de la Aplicación.
Fuente: Propia

El control de acceso a la interfaz de usuario de la aplicación esta implementado en el módulo sfGuardAuth de la Aplicación Integral (Core) del GAD-I, este módulo utiliza el plugin **BasesfGuardAuthActions**²⁷ de symfony, el mismo que fue modificado para controlar el acceso a las aplicaciones que se encuentran en el Core.

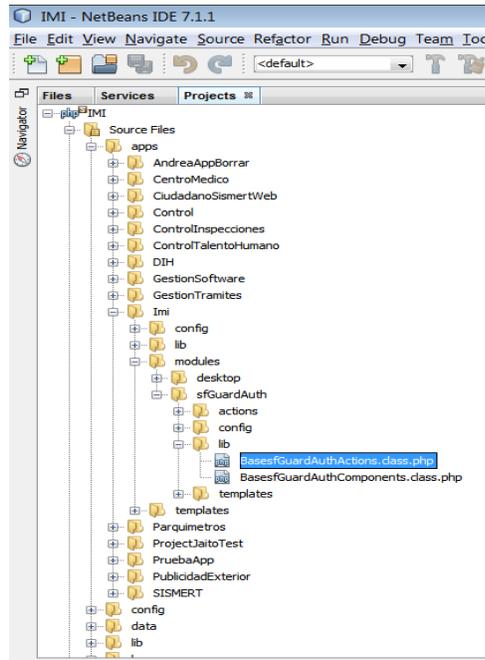


Figura 3. 7: Estructura del Módulo de Acceso sfGuardAuth del Core (GAD-I)
Fuente: Propia

El control de Acceso se complementa con el módulo **desktop**, el que se encarga de mostrar al usuario la interfaz de escritorio con las aplicaciones que se le permitió visualizar después de autenticarse.



Figura 3. 8: Interfaz de Acceso a la Aplicación Integral (Core)
Fuente: Propia

²⁷ BasesfGuardAuthActions : Clase Controlador, permite el control de ingreso al sistema.



Figura 3. 9: Interfaz de Escritorio de las Aplicaciones Integradas en el Core.
Fuente: Propia

➤ **Historia 1: Requerimientos para uso del Parqueadero Sismert**

Tarea 3 y 4: La Historia inicia con el análisis de requerimientos y creación de la aplicación, generando también el esquema y modelo de datos para el desarrollo de la aplicación.

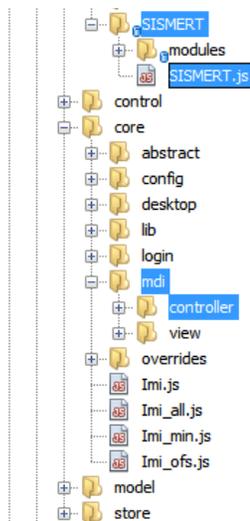
Después del análisis realizado conjuntamente con el usuario, se propuso que para el uso del parqueadero es necesario crear el registro (teléfonos, vehículos) y Compras de Saldo, por lo que se dispondrá de un mejor control al crear este módulo.

Tareas:

- 3. Generar la aplicación en Symfony y construir la estructura MVC en extjs4.
- 4. Generar Esquema y modelo de Datos en Symfony.

126ymfony app:routes SISMERT Creación de la Aplicación	126ymfony doctrine:build- schema Generar Esquema de datos	126ymfony doctrine:build- model Generar modelo de datos
--	---	---

La estructura de la aplicación en el cliente ExtJS se lo crea manualmente, y para acceder la interfaz grafica se crea un archivo de configuración que se extiende del controlador (MDI) del Core.



```
Ext.define("Imi.apps.SISMERT.SISMERT",  
{  
  extend:"Imi.mdi.controller.ControladorMdi"  
});
```

Archivo de Configuración de la Aplicación SISMERT.js

Figura 3. 10: Esquema inicial al crear la Aplicación en el Core.
Fuente: Propia

La estructura MVC en la parte del cliente ExtJS se crea dependiendo de los módulos que tenga la aplicación a continuación se muestra la estructura de la aplicación SISMERT con el módulo de registros y compras de saldo.

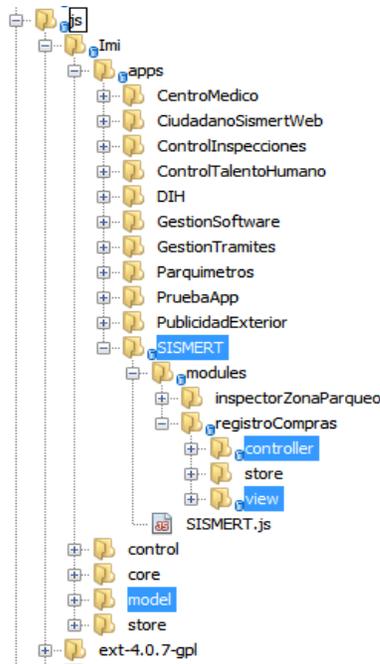


Figura 3. 11: Estructura MVC de la Aplicación SISMERT en ExtJS.
Fuente: Propia

Ahora se debe crear el módulo inicial en el servidor (Symfony) para desarrollar la aplicación, este módulo va ofrecer el servicio de registro y compras de saldo.

symfony doctrine:generate-module registroCompras SISMERT
Creación del módulo de Registro y Compras de Saldo

A continuación se crea la interfaz grafica y funcional para el registro de teléfonos que serán los usuarios para acceder al servicio del parqueadero del SISMERT.

Importante (Controlador de un Item Menu):

Para acceder a una ventana del sistema se debe crear una clase JavaScript .js que va hacer el controlador que se encarga de realizar peticiones y traer respuestas desde el servidor.

El controlador de JavaScript actúa como un disparador en una acción del ítem de un menú como se muestra a continuación:



Figura 3. 12: Acción de un Item-Menu desde el Controlador ExtJS.
Fuente: Propia

```
Ext.define('Imi.apps.SISMERT.modules.registroCompras.controller.RegPersonaController',
{
    extend: 'Imi.abstract.Controller',
    onLaunch: function()
    {
        Ext.create('Imi.apps.SISMERT.modules.registroCompras.view.registroCompras.RegistroPersona', {controller: this}).show();
        this.callParent();
    }
});
```

Tabla 3. 52: Estructura del controlador de un Item Menu (ExtJS)
Fuente: Propia

El Controlador se extiende desde la clase Abstracta del Core IMI, y su primera acción a ejecutar es el que se encuentre en la función **OnLaunch**.

Tareas:

5. Diseño GUI del formulario para registro de usuarios (teléfonos) Sismert.
6. Implementar las acciones en Symfony y ExtJS para el envío de datos de un registro de usuario Sismert.
7. Realizar pruebas de funcionalidad de un registro Sismert.

REGISTRO USUARIOS SISMERT

Teléfono Celular :*

Cédula Identidad :*

Operadora Movil:*

CLARO
 MOVISTAR
 ALEGRO
 CNT

Email:*

Fecha Registro:*

Guardar Cancelar

Figura 3. 13: Interfaz de Registro de Teléfonos SISMERT.
Fuente: Propia

El Registro de un teléfono también se puede editado los campos de la Operadora Móvil, Email, además si el número ingresado es incorrecto puede ser eliminado si aun no tiene registros que dependen de él.

Tareas:

8. Rediseñar GUI de registro Sismert para búsqueda y edición de un registro.
9. Implementar acciones pertinentes a búsqueda y edición de un registro.
10. Verificar funcionalidad.

REGISTRO DE USUARIOS SISMERT

Nuevo Editar Expandir Datos Retraer Datos Buscar 0989778000 1705594958

Datos Registros

	Telefono Celular	Cédula Ciudadano	Fecha Registro	Operadora Movil	Email
1	0989778000	1705594958	2012/12/10	MOVISTAR	k@j.com

Datos Personales Del Registro

Apellidos: TENECORA ZHISHPUIN

Nombres: JOSE POLIVIO

Eliminar

Figura 3. 14: Ventana para Búsqueda y Edición de un Registro (Teléfono)
Fuente: Propia

REGISTRO USUARIOS SISMERT

Teléfono Celular :*

Cédula Identidad :*

Operadora Movil:*

CLARO
 MOVISTAR
 ALEGRO
 CNT

Email:*

Fecha Registro:*

Guardar Edicion Cancelar

Figura 3. 15: Interfaz de Edición de un Registro de Teléfono SISMERT.
Fuente: Propia

Para la recepción de un SMS que el ciudadano envía al iniciar el servicio del parqueadero se cuenta con un sistema de envío y recepción de mensajes, que será reutilizado y acoplado para las diferentes notificaciones por uso del servicio. En el análisis de este sistema se ha comprobado que es factible utilizarlo para la recepción y envío de mensajes, cumpliendo la **Tarea 11**: Verificar servicios de Sistema SMS GAD-I.

➤ **Historia 4: Contenido del SMS para uso del Parqueadero**

El contenido del SMS se lo analizó minuciosamente de tal manera que el ciudadano al enviar su mensaje de texto no sea muy complicado de recordarlo y pueda identificar el código que se debe enviar para iniciar el servicio del parqueadero.

Tarea 1: Analizar posibles contenidos para SMS.

Ahora se procede a la asignación de códigos para identificar el servicio de parqueo, asignación de una letra a una placa vehicular y la identificación del tiempo que se va estacionar en una zona regulada.

Tareas:

2. Verificar alternativa de identificación de un vehículo para contenido SMS.
3. Identificar sintaxis que contendrá el SMS para el tiempo de uso del parqueadero.
4. Establecer identificación de SMS para parqueo (P) en el contenido SMS.

<p>PCC5705 → A ACC0908 → B Asignación de un código a una Placa vehicular.</p>	<p>1 → 0.30 Minutos 2 → 1.00 Horas 3 → 1.30 Horas 4 → 2.00 Horas 5 → 2.30 Horas 6 → 3.00 Horas Identificación de Tiempo de Estacionamiento.</p>	<p>Código: P Identificación del servicio solicitado en un SMS.</p>
--	--	---

Identificación de placa vehicular y tiempo de parqueo

Ahora se procede a realizar el análisis de la estructura del contenido del mensaje con la identificación y asignación de códigos creados anteriormente. Después del análisis se ha seleccionado el contenido final del mensaje de texto y posteriormente el rediseño de la estructura de datos.

Tareas:

5. Verificar y establecer las distintas formas que el usuario podrá escribir en el contenido SMS.
6. Rediseñar Base de Datos para formar contenidos SMS válidos.
7. Analizar contenido final de un SMS para el proceso de respuestas hacia el usuario.

Después de verificar y analizar las formas del contenido del SMS se ha podido establecer lo siguiente:

Contenido: P A SBPM 1 ‘o’ P AA0000 SBPM 1 P: Servicio de Parqueo A(AA0000): Código o placa del vehículo SBPM: Sitio(Simón Bolívar y Pedro Moncayo) 1: Tiempo(0.30min) SMS con Sitio de Parqueo	Contenido: P A 1 ‘o’ P AA0000 1 P: Servicio de Parqueo A(AA0000): Código o placa del vehículo 1: Tiempo(0.30min) SMS sin Sitio de Parqueo
---	--

Tabla 3. 53: Contenido SMS para iniciar Servicio de Estacionamiento
Fuente: Propia

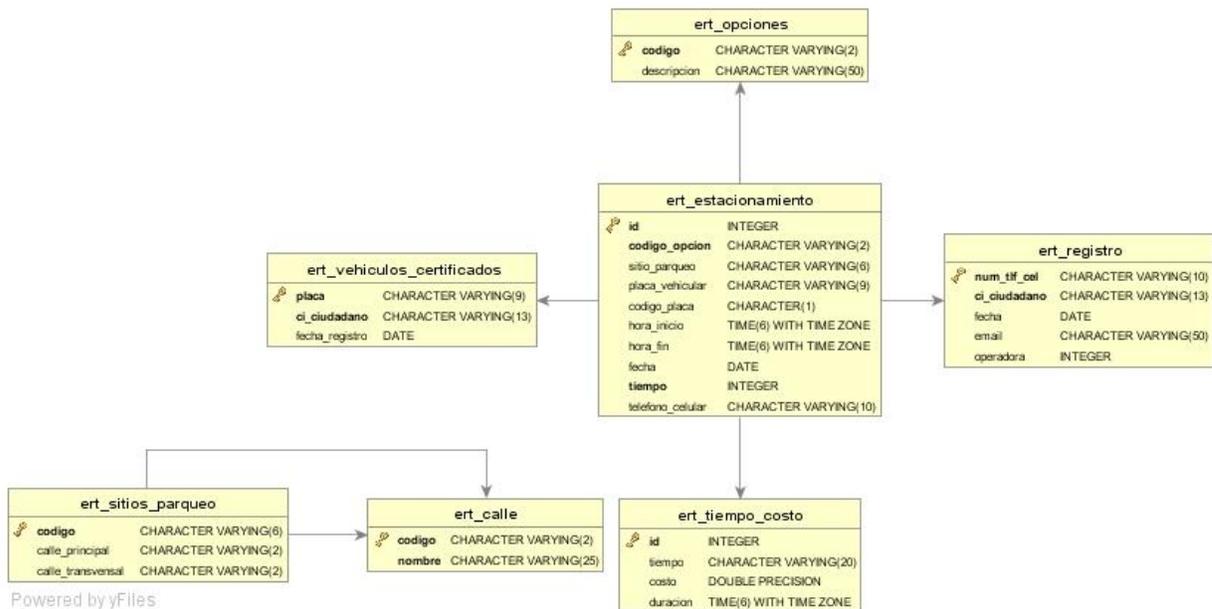


Figura 3. 16: Modelo Relacional Contenido SMS - Iniciar Servicio Estacionamiento.
Fuente: Propia

➤ **Historia 5: Registro de Vehículos**

Después de haber realizado el registro de teléfonos, ahora se debe continuar con el registro de vehículos en un número de teléfono que ya conste en el sistema.

Tareas:

1. Crear la clase para el controlador de registro de vehículos.
2. Crear clase para modelo y store del registro y reporte de vehículos.
3. Diseño GUI para formulario registro de vehículos

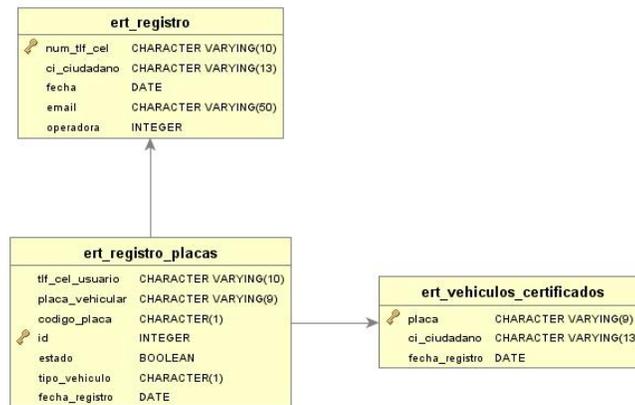


Figura 3. 17: Modelo relacional de Registro de Vehículos
Fuente: Propia

El controlador para registro de vehículos es similar a la estructura del controlador de registro de teléfonos explicado en la Iteración 1 (Ver Tabla 3.50), por lo que se procede a crear la clase para el modelo y store.

<pre>Ext.define('Imi.apps.SISMERT.modules.registroComp ras.store.RegistroPlacaStore', { extend: 'Imi.abstract.Store', model: 'Imi.model.RegistroPlaca', proxy: { reader: { type: 'json', root: 'registroPlacas' } } });</pre>	<pre>Ext.define('Imi.model.RegistroPlaca', { extend: 'Ext.data.Model', fields: [{name: 'placa_vehicular', type: 'string'}, {name: 'codigo_placa', type: 'string'}] });</pre>
---	--

Tabla 3. 54: Modelo y Store para registro de placas vehiculares (ExtJS).
Fuente: Propia

Figura 3. 18: Interfaz de registro de placas vehiculares.
Fuente: Propia

Para la clase modelo y store de las tareas que se presenten en las historias restantes se hará referencia a las clases creadas en la Tabla: 3.52 porque su estructura es similar.

Tareas:

4. Implementar acciones de consulta de teléfonos registrados en el lado del servidor.
5. Implementar llamadas de Base Datos desde la UI y comprobar si el teléfono es válido.

Figura 3. 19: Búsqueda de un Teléfono Registrado
Fuente: Propia

Tareas:

6. Codificar las respectivas acciones en la clase manager para registro de un vehículo.
7. Rediseño de la GUI para registro de vehículos.
8. Implementar acciones en el cliente para envío de datos al servidor.
9. Mostrar los resultados de éxito o fracaso al realizar el registro de vehículos.

Placa Vehicular	Código Asignado De Placa	Apellidos Propietario	Nombres Propietario	Tipo Vehículo	Fecha Registro
PCC5705	A	DIAZ FARINANGO	LUIS EDGAR	LIVIANO	2013/01/20

Figura 3. 20: Ventana principal de registro de vehículos
Fuente: Propia

En la figura anterior también se puede observar el código asignado a un vehículo después de ser registrado. Finalmente la tarea 10 es un rediseño de la interfaz de la ventana de registro de vehículos, esta nueva ventana se mostrara en la siguiente iteración para dar de baja a una placa vehicular.

3.9.2. Iteración II (Módulo de Registro y Compras de Saldo)

La iteración presenta el desarrollo de las historias 5 y 7, que permite dar de baja al registro de un vehículo. Además se realiza las compras de saldo y su respectiva consulta, también se muestra la información de los registros de teléfonos ingresados al sistema.

➤ Historia 5: Registro de vehículos

Tareas:

1. Realizar llamadas al Servidor para comprobación de un teléfono válido.
2. Mostrar la información de los vehículos registrados dependiendo de un teléfono válido.
3. Crear la acción de eliminación de un vehículo en el lado del servidor y cliente.
4. Mostrar los resultados después de la acción realizada por el usuario.

Placa Vehicular	Codigo Asignado De Placa	Apellidos Propietario	Nombres Propietario	Fecha Registro
1 PCC5705	A	DIAZ FARINANGO	LUIS EDGAR	2013/01/20

Figura 3. 21: Ventana de Administración de Vehículos Registrados en un Teléfono.
Fuente: Propia

➤ Historia 7: Compras de Saldo, Información Registros

Las Tareas de la historia inician con el proceso de compras de Saldo para un teléfono registrado.

Tareas:

1. Crear la clase para el Controlador de Compras de Saldo.
2. Crear clase para modelo y Store del para información de registro de teléfonos.
3. Diseño GUI para formulario de compras de saldo.

RECARGAS DE SALDO

Recarga: *

Fecha de Recarga: * 2013/01/20

Monto para Recargas de Saldo SISMERT

Comprar Saldo Cancelar

Figura 3. 22: Interfaz para Recargas de Saldo SISMERT.
Fuente: Propia

Las acciones para el control de recargas se realiza en el cliente y el servidor, como se explica en las siguientes tareas:

4. Implementar funciones para compras de saldo en las clases Manager y Logic y acciones correspondientes.
5. Implementar llamadas de Base Datos desde la UI y comprobar si el teléfono es esta registrado para la recarga de saldo.
6. Implementar acciones en el cliente para envío de datos de la recarga a realizarse.
7. Realizar las pruebas de envío y recepción de los datos de una recarga.
8. Corregir datos erróneos y mostrar resultados de las recargas de saldo.



Figura 3. 23: Ventana Funcional de Compras de Saldo SISMERT.
Fuente: Propia

La consulta del saldo disponible de un número de teléfono se lo desarrolla en las tareas 10, 11, 12, 13, 14, y 15, que es similar al proceso de las tareas que se realizaron anteriormente, a continuación se presenta el resultado de la interfaz funcional para las consultas de saldo.

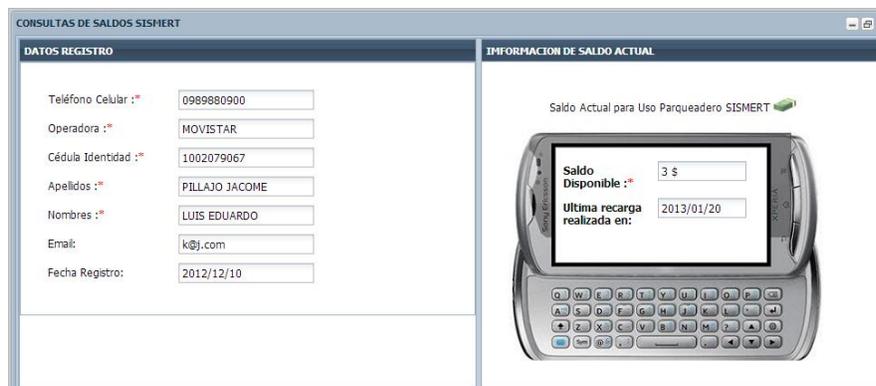
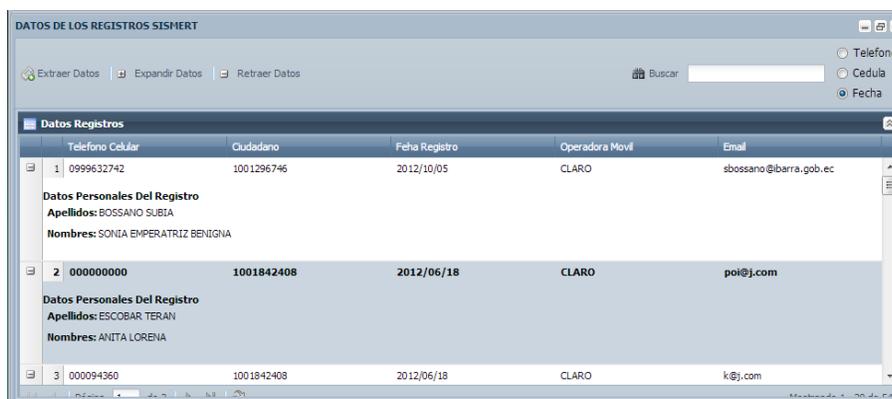


Figura 3. 24: Ventana Consultas de Saldo SISMERT
Fuente: Propia

La visualización de la información de todos los registros de teléfonos se realizan en las siguientes tareas: 16, 17, 18, 19 y 20 creando su clase controlador para el acceso a la interfaz de los registros.

Los registros pueden ser filtrados si el usuario desea revisar los datos de alguna fecha en específico, como se muestra a continuación:



The screenshot shows a software window titled "DATOS DE LOS REGISTROS SISMERT". At the top, there are buttons for "Extraer Datos", "Expandir Datos", and "Retraer Datos", along with a search bar labeled "Buscar". On the right, there are radio buttons for "Telefono", "Cedula", and "Fecha", with "Fecha" selected. Below this is a table with the following data:

	Telefono Celular	Ciudadano	Fecha Registro	Operadora Movil	Email
1	0999632742	1001296746	2012/10/05	CLARO	sbossano@barra.gob.ec
Datos Personales Del Registro Apellidos: BOSSANO SUBIA Nombres: SONIA EMPERATRIZ BENIGNA					
2	000000000	1001842408	2012/06/18	CLARO	poi@j.com
Datos Personales Del Registro Apellidos: ESCOBAR TERAN Nombres: ANITA LORENA					
3	000094360	1001842408	2012/06/18	CLARO	k@j.com

Figura 3. 25: Ventana de Información y Búsqueda de Registros.
Fuente: Propia

3.9.3. Iteración III (Módulo Inspección Zona Parqueo)

La Iteración consta de un análisis cuando el usuario tiene tiempo disponible de estacionamiento y desea usar este tiempo en otra plaza del SISMERT.

También se desarrolla la administración interna para el envío y recepción de SMS, a través de asignaciones de sitios de inspección a los Inspectores.

➤ Historia 8: Cambiar Lugar de Estacionamiento

El análisis para el cambio de un sitio a otro es a través del envío de mensajes de texto al Inspector indicándole la hora de finalización del servicio, y el sitio donde se encuentra estacionado.

Tareas:

1. Analizar la manera de que el ciudadano pueda cambiarse de sitio de parqueo.
2. Analizar el contenido del SMS que el ciudadano envía para cambio de sitio de parqueo.

3. Analizar el contenido del SMS para notificar al ciudadano e Inspector sobre el parqueo.
4. Analizar la factibilidad del cambio de sitio de parqueo.

<p>Contenido: CP A SBPM ‘o’ CP AA0000 SBPM</p> <p>CP: Cambio Sitio de Parqueo A(AA0000): Código o placa del vehículo SBPM: Sitio(Simón Bolívar y Pedro Moncayo)</p>

Tabla 3. 55: Contenido SMS al Cambiar el Sitio de Parqueo.
Fuente: Propia

<p>Contenido: AAA0000 12:00 12:30 SBPM</p> <p>AA0000: Placa Vehicular 12:00 12:30: Hora de Inicio y Finalización. SBPM: Sitio donde se estaciona un vehículo.</p> <p style="text-align: center;">Notificación Inspector</p>	<p>Contenido: 12:00 12:30 3.75</p> <p>12:00 12:30: Hora de Inicio y Finalización. 3.75: Saldo Actual.</p> <p style="text-align: center;">Notificación Ciudadano</p>
--	--

Tabla 3. 56: Contenido SMS Notificación Ciudadano e Inspector.
Fuente: Propia

La factibilidad de que un vehículo pueda cambiarse de un lugar a otro cuando aún le queda tiempo es necesaria pero a la vez implica que el ciudadano tenga que gastar más por el uso del parqueadero, por que el costo del mensaje tiene que asumirlo el usuario.

Más adelante en la Iteración 6 se planteará una nueva solución para las notificaciones hacia el Inspector cuando un ciudadano se cambia de un sitio a otro mientras tenga tiempo de parqueo.

Para la solución planteada mediante envío de SMS se necesita crear una interfaz para la administración de los teléfonos de los inspectores como se indica en las siguientes tareas.

Tareas:

5. Crear un controlador para ingresar y editar el número de teléfono de los Inspectores.
6. Crear clase para modelo y Store de los Inspectores.
7. Diseño GUI para formulario ingreso y edición de los teléfonos de Inspectores.

Cedula Identidad	Apellidos	Nombres	E-mail	Numero de Telefono	Operadora
1	1001667706	BAEZ FLORES	PABLO JAVIER	08888453	CLARO
2	1002447538	BRITO GONZALEZ	ANGEL RUBEN	08888876	CLARO
3	1002515094	MOREJON LLINA	PABLO AMADOR	08888888	CLARO
4	1001394905	JATIVA	PABLO JAVIER	08888865	MOVISTAR
5	1002661518	ROSETO MOLINA	JONNY VINICIO	08888888	CLARO
6	1002688461	LOVATO TUSA	LAURA CRISTINA	08888888	CLARO
7	1002714408	GUBIO BARRIGA	MARGOTH DEL ROCIO	08888888	CLARO
8	1002741955	CRILLO ESTEVES	MARCO DANIEL	08888888	CLARO
9	1002815668	QUILUMBA ALVAREZ	FLOR MARIA	08888888	MOVISTAR
10	1002858551	LASCANO CHUMAPI	LENNIN ROBERTO	08888888	MOVISTAR
11	1002865150	SALAZAR ISIZAN	ALEXANDRA PAULINA	08888888	MOVISTAR
12	1002869541	ERAZO TENGANA	ANDRES MAURICIO	08888888	MOVISTAR
13	1002884771	RIVERA PEÑAFIEL	PABLO MAURICIO	08888888	CLARO
14	1003880768	CADENA SURTA	DAVID SANTIAGO	08888888	CLARO

Figura 3. 26: Interfaz Administración de Teléfonos Inspectores.
Fuente: Propia

La interfaz gráfica de administración se rediseñó para editar los teléfonos de cada Inspector. Esta administración se la realiza con el fin de tener registrados todos los números de teléfono de cada Inspector para enviarle el SMS de notificación cuando un vehículo está utilizando una plaza de estacionamiento.

Cedula Identidad	Apellidos	Nombres	E-mail	Numero de Telefono	Operadora
1	1001667706	BAEZ FLORES	PABLO JAVIER	08888453	CLARO
2	1002447538	BRITO GONZALEZ	ANGEL RUBEN	08888876	CLARO
3	1002515094	MOREJON LLINA	PABLO AMADOR	08888888	CLARO
4	1001394905	JATIVA	PABLO JAVIER	08888865	MOVISTAR
5	1002661518	ROSETO MOLINA	JONNY VINICIO	08888888	CLARO
6	1002688461	LOVATO TUSA	LAURA CRISTINA	0994366732	CLARO
7	1002714408	GUBIO BARRIGA	MARGOTH DELT	08888888	CLARO
8	1002741955	CRILLO ESTEVES	MARCO DANIEL	08888888	CLARO
9	1002815668	QUILUMBA ALVAREZ	FLOR MARIA	08888888	MOVISTAR
10	1002858551	LASCANO CHUMAPI	LENNIN ROBERTO	08888888	MOVISTAR
11	1002865150	SALAZAR ISIZAN	ALEXANDRA PAULINA	08888888	MOVISTAR
12	1002869541	ERAZO TENGANA	ANDRES MAURICIO	08888888	MOVISTAR
13	1002884771	RIVERA PEÑAFIEL	PABLO MAURICIO	08888888	CLARO
14	1003880768	CADENA SURTA	DAVID SANTIAGO	08888888	CLARO

Figura 3. 27: Ventana de Asignación y Edición de un Número de Teléfono al Inspector.
Fuente: Propia

Tareas:

8. Codificar las respectivas acciones en la clase manager para ingreso y edición de un teléfono del Inspector.
9. Implementar acciones en el cliente para envío de datos al servidor.
10. Mostrar los resultados de éxito o fracaso al realizar el registro y edición de datos.

➤ Historia 9: Notificaciones Inspector y Ciudadano

El Análisis para el contenido para las notificaciones hacia el inspector y ciudadano, cuando el sistema recibe una petición desde un SMS se lo puede ver en la Tabla 3.53, cumpliendo así las tareas 1, 2 y 3.

A continuación se desarrolla la asignación y edición de un sitio de inspección a un Inspector para el envío de mensajes de notificaciones a los Inspectores más cercanos cuando se inicia el servicio de estacionamiento.

Tareas:

4. Crear un controlador para asignar y editar un sitio de inspección (I) para el Inspector.
5. Crear clase para modelo y Store de sitios de Inspección (I).
6. Diseño GUI para formulario asignación y edición de sitios de inspección.

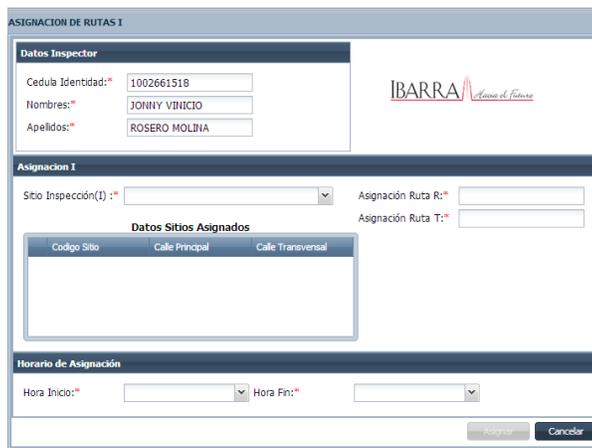


Figura 3. 28: Formulario de Asignación y Edición de un Sitio a un Inspector.
Fuente: Propia

Tareas:

7. Codificar las respectivas acciones en la clase manager para asignación y edición de un sitio de Inspección.
8. Implementar acciones en el cliente para envío de datos al servidor.
9. Mostrar los resultados de éxito o fracaso al asignar y editar un sitio de Inspección.

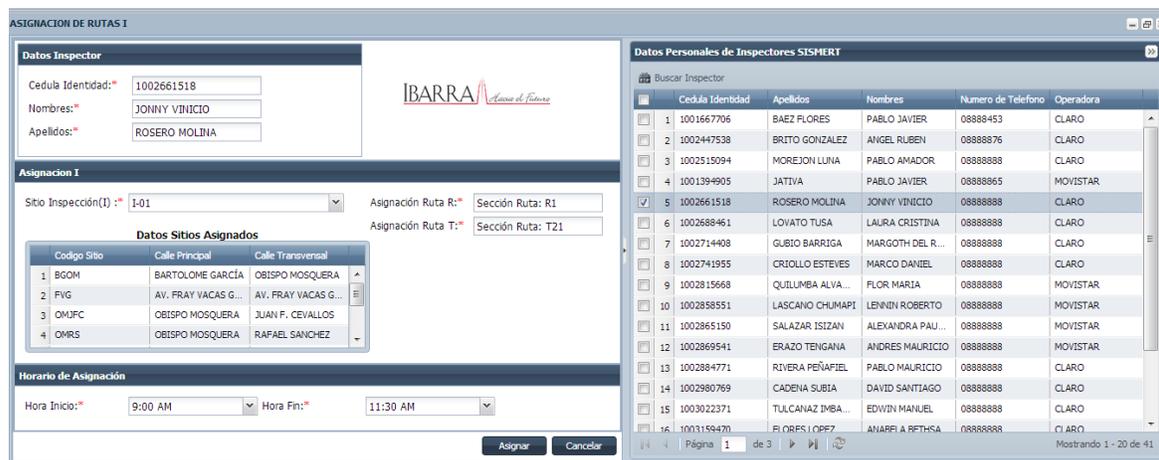


Figura 3. 29: Ventana Funcional de Asignación y Edición de un Sitio a un Inspector.
Fuente: Propia

3.9.4. Iteración IV (Módulo Inspección Zona Parqueo)

La Iteración se desarrolla para establecer los sitios de estacionamiento mediante la asignación de códigos que serán utilizados por el usuario cuando envíe un SMS solicitando el servicio de parqueo, así se podrá saber donde se encuentra estacionado.

Estos sitios son administrados por zonas que albergan varios grupos de inspección para saber que Inspectores están cerca del lugar donde se inició un servicio.

➤ Historia 6: Establecer Identificación de Sitios para el Parqueadero

Las tareas que se presenta a continuación sirven para establecer el código de un sitio o lugar de estacionamiento.

Tareas:

1. Identificar calles de las plazas del SISMERT.
2. Analizar y crear código de identificación de un sitio de parqueo.
3. Rediseñar Base de Datos para la implementación de sitios de parqueo.
4. Ingresar datos de prueba, formando sitios de parqueo.

Las calles de las plazas se encuentran registradas en el esquema actual del SISMERT, por lo que se procedió a reutilizar esta información y a crear los códigos de un sitio.

Calle Principal: Simón Bolívar
Calle Transversal: Pedro Moncayo
Sitio: Simón Bolívar y Pedro Moncayo.
Código: SMPM

Tabla 3. 57: Código de identificación de un sitio de estacionamiento.
Fuente: Propia

Nota: Los códigos asignados para el sitio están formados por las iniciales de los nombres de las calles.

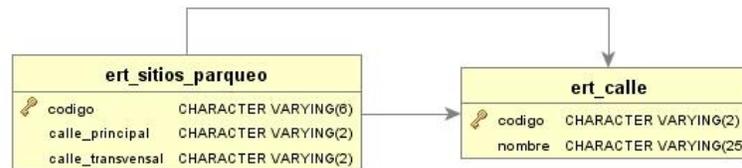


Figura 3. 30: Modelo Relacional Código Sitios Parqueo.
Fuente: Propia

Los sitios creados se los ha agrupado por rutas y zonas, porque un Inspector tiene asignado un lugar de Inspección que está formado una ruta R y ruta T.

Ruta R y Ruta T: Están Formadas por cuadras de diferentes calles.

Sitio Inspección (I): I-01 = R01 y T01

R01 comprende: Obispo Mosquera, Rafael Sánchez.

T01 Comprende: Av. Fray Vacas Galindo.

Para agrupar los sitios de Inspección (I) se crearon zonas, con esto ya es posible saber cuáles Inspectores son los más cercanos a un vehículo estacionado.

Tareas:

5. Agrupar los sitios por zonas para el control del parqueo.
6. Implementar estructura de datos y realizar pruebas ingresando datos.

Zona	Ruta R	Ruta T	I	Código Sitio
1	R01	T01	I01	RSOM(Rafael Sánchez y Obispo Mosquera)
1	R01	T01	I01	FVG(Avenida Fray Vacas Galindo)
1	R02	T02	I09	ZVJFC(Zenón Villacis y Juan francisco Cevallos)
1	R02	T02	I09	OMAC(Obispo Mosquera y Antonio Cordero)
2	R04	T03	I14	OBBG(Obispo Mosquera y Bartolomé García)
3	R11	T07	I23	CCSF(Cristóbal Colón y Sánchez y Cifuentes)

Tabla 3. 58: Agrupación de Sitios de Inspección (I) en Zonas.
Fuente: Propia

Si el usuario se encuentra estacionado con el código **RSOM**, entonces se sabe que está dentro de la zona 1 y las notificaciones serán enviadas a todos los Inspectores que están registrados en esta zona (I01, I09).

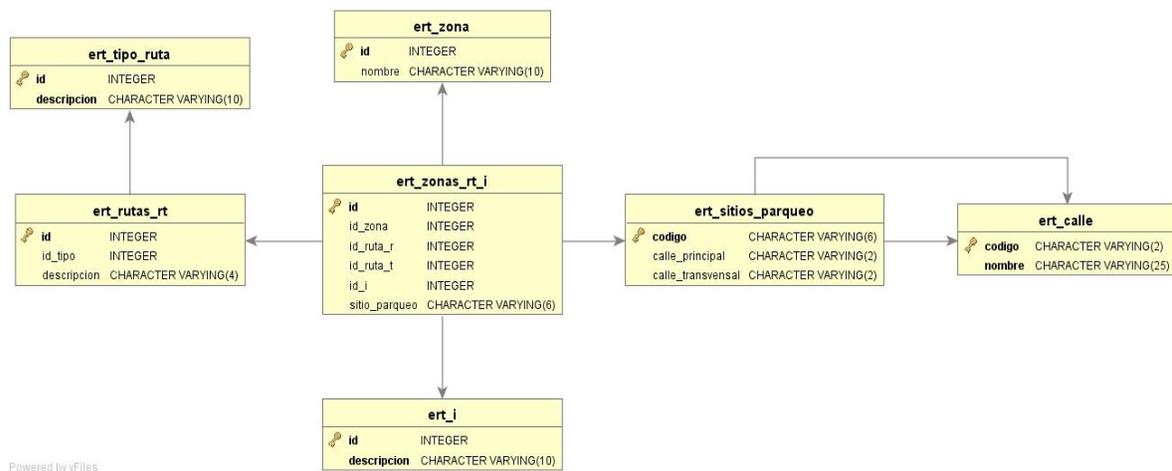


Tabla 3. 59: Modelo relacional agrupacion sitios (I) por zonas.
Fuente: Propia

Tareas:

7. Crear controlador para ingresar y editar un sitio de parqueo.
8. Crear clase para modelo y Store.
9. Diseño GUI para formulario ingreso y edición de sitios de parqueo.

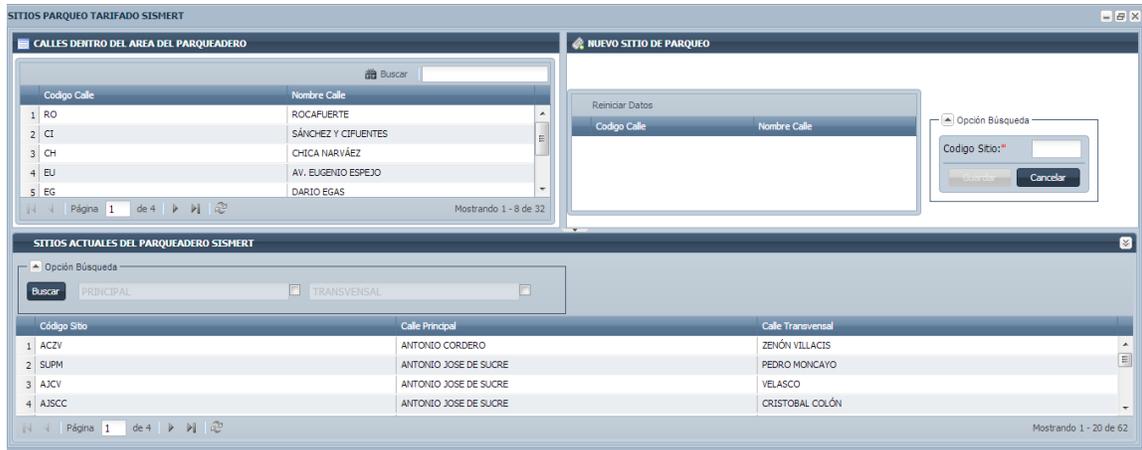


Figura 3. 31: Formulario inicia de Ingreso y edición de Sitios de Parqueo.
Fuente: Propia

Tareas:

- 10. Codificar las respectivas acciones en la clase manager para ingreso y edición de un sitio.
- 11. Implementar acciones en el cliente para envié de datos al servidor.
- 12. Mostrar los resultados de éxito o fracaso al realizar el ingreso y edición de datos.

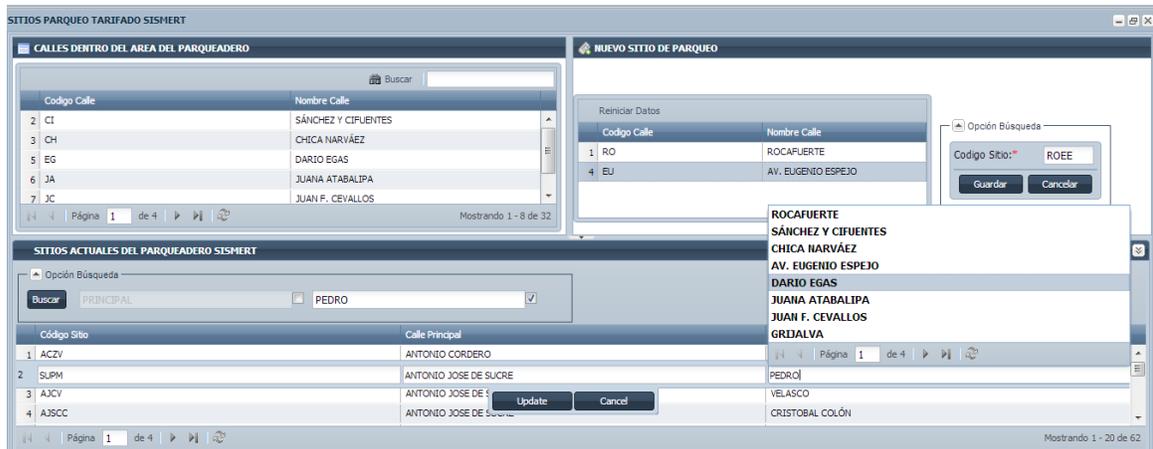


Figura 3. 32: Ventana Funcional Para Ingreso y edición de Sitios de Parqueo.
Fuente: Propia

Las Tareas 13, 14, 15, 16, 17 y 18 realizan el mismo proceso de creación de clases y acciones que se siguió anteriormente, pero ahora se desarrolla para ingresar una asignación de un sitio de inspección (I) dentro de una zona como se explico anteriormente en la **Tabla 3.56**.

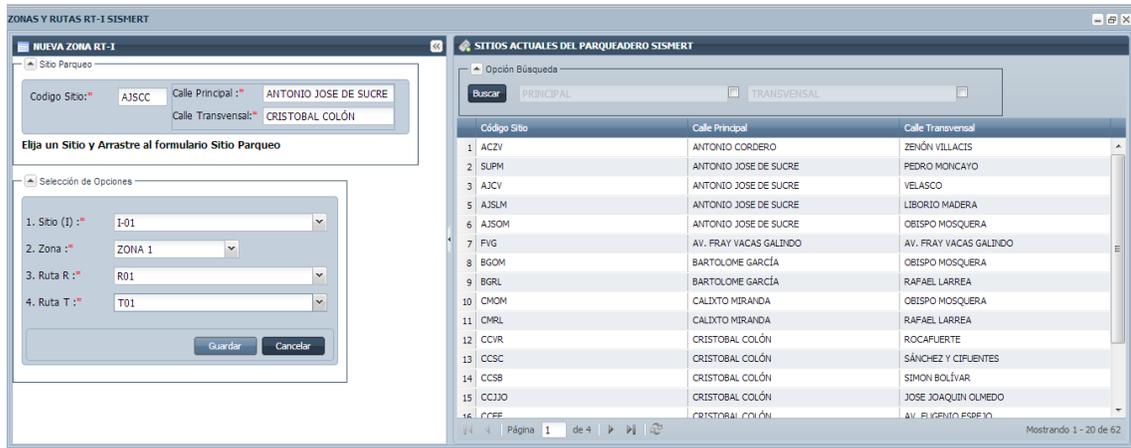


Figura 3. 33: Ventana de Asignación de Sitios de Inspección (I) en una Zona.
Fuente: Propia

Las Tareas 19, 20, 21, 22, 23 y 24 realizan el proceso de edición de la asignación de un sitio de Inspección en una zona, La edición muestra una lista de registros donde el usuario debe seleccionar el registro a editar y posteriormente se le cargan los datos similar a la ventana de la **Figura 3.33**

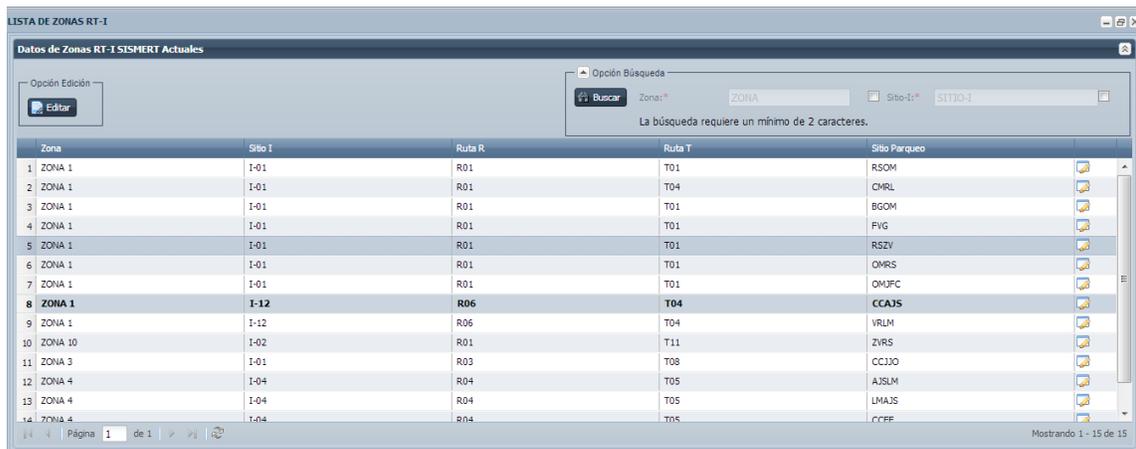


Figura 3. 34: Ventana Para Edición de Sitios (I) Asignados en una Zona
Fuente: Propia

3.9.5. Iteración V (Módulo de Registros y Compras de Saldo)

La Iteración de desarrolla con el fin de obtener los datos de los vehículos que obtienen el certificado para el proceso de matriculación vehicular.

Las placas registradas se tomaran para almacenarlas en una estructura de datos (ert_Vehiculos_certificados) del sistema que se está desarrollando, por que los vehículos que desean registrarse en el sistema deben constar en la Tabla de ert_Vehiculos_certificados.

Tareas:

1. Revisar y analizar vehículos certificados en SISMERT.
2. Crear función para tomar datos de vehículos y su propietario.
3. Pruebas y revisión de información de datos de vehículos.

La Función Implementada realiza lo siguiente:

- Tomar registros de placas de la estructura de datos de certificados del SISMERT, y migrarlas a la tabla ert_vehiculos_certificados formando una clave primaria de la placa vehicular.
- Actualiza el propietario del vehículo antes de migrarla.



Figura 3. 35: Modelo de Datos de un Vehículo a Registrarse.
Fuente: Propia

Las tareas 4, 5, 6, 7, 8 y 9 se realizan para mostrar la información de los datos de los vehículos que constan para que puedan ser registrados para el servicio del estacionamiento a través de este sistema.

También se desarrolla la interfaz funcional para el ingreso de nuevos vehículos.

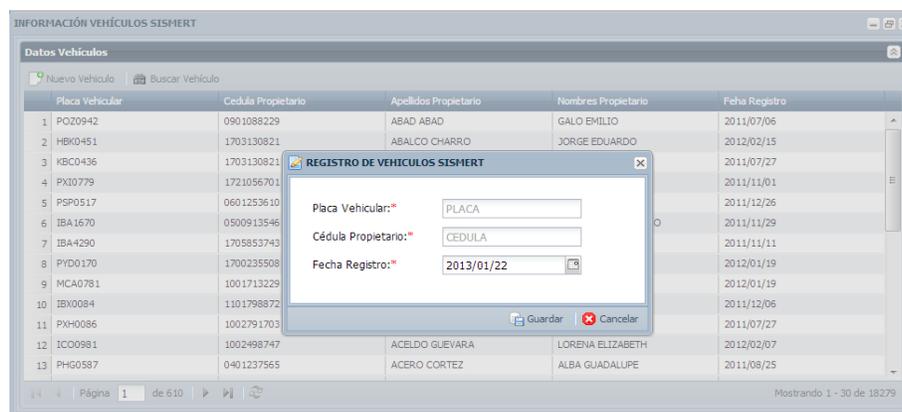


Figura 3. 36: Ventana de Revisión e Ingreso de Vehículos a ser Registrados.
Fuente: Propia

3.9.6. Iteración VI (Módulo Control Parqueo vía Aplicación Móvil)

La Iteración se desarrolla con el propósito de crear una aplicación móvil que realice el control del parqueadero, consultando el tiempo disponible de un vehículo estacionado, el registro de teléfonos y vehículos, compras de saldo e ingreso de notificaciones y la posibilidad de iniciar el servicio de estacionamiento de un vehículo.

La aplicación está proyectada para el uso exclusivo de un Inspector, porque anteriormente en el envío de notificaciones cuando un vehículo se encuentre estacionado no fue tan factible económicamente por que los SMS debían ser enviados a todos los inspectores para que cuando el vehículo cambia de un lugar a otro siempre que tenga tiempo disponible, no sea necesario que el usuario envíe un SMS avisándole al sistema que se cambió de lugar.

Con esta aplicación el Inspector simplemente digita la placa vehicular y comprueba la disponibilidad de tiempo de un vehículo.

➤ Historia 10: Plantear nueva solución de notificación Hacia el inspector

Las tareas se realizan para el análisis de creación de la aplicación móvil, análisis de integración con los datos del Módulo de Registros y Compras de Saldo.

Tareas:

1. Analizar la forma para revisar si un vehículo tiene tiempo de parqueo.

2. Analizar y verificar si es posible crear una aplicación con el framework Extjs4.
3. Analizar la posibilidad de crear una aplicación móvil para el control de estacionamiento.
4. Analizar la manera de integrar la nueva aplicación con la actual.

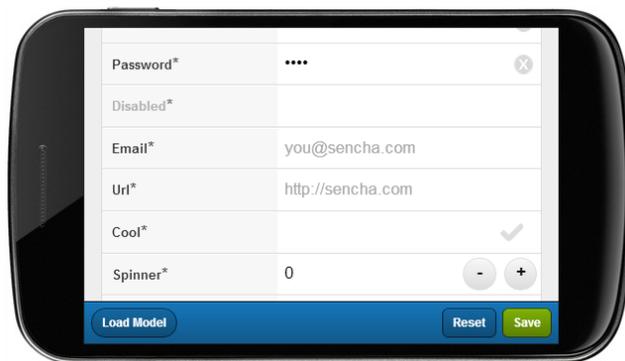
Realizar la aplicación móvil en ExtJS no es tan factible por que los componentes gráficos son demasiado pequeños para visualizarlos, por eso se opta por utilizar el framework de Sencha Touch para su realización.

➤ **Historia 11: Creación Aplicación Móvil y Seguridad.**

Las tareas que se desarrollan en la historia presentan el análisis de los componentes del framework de Sencha Touch, para la creación de la aplicación móvil e implantación de su seguridad.

Tareas:

1. Analizar el framework de Sencha Touch para crear aplicación móvil.
2. Revisar el tipo de componentes que ofrece el framework.
3. Realizar pruebas para comprobación de la funcionalidad del framework.



**Figura 3. 37: Componentes de un formulario en Sencha Touch.
Fuente: API Sencha Touch**

Tareas:

4. Analizar la aplicación actual para reutilizar las fuentes de acceso y seguridad a la nueva aplicación.

5. Crear la nueva aplicación móvil genérica para varias sub-aplicaciones.
6. Integrar la seguridad de acceso a la aplicación con la respectiva creación del Core en el cliente.
7. Realizar pruebas de acceso y corrección de posibles errores en la aplicación móvil.

La seguridad de la aplicación móvil es implementada con el módulo **sfGuardAuth** que ya fue explicado anteriormente en la Iteración 1, con las respectivas modificaciones para mostrar al usuario una lista de aplicaciones a las que tiene acceso.



Figura 3. 38: Autenticación para Acceso a la Aplicación Móvil.
Fuente: Propia

Después de la autenticación el usuario puede visualizar un listado de las aplicaciones a las que tiene acceso como se muestra en siguiente imagen.

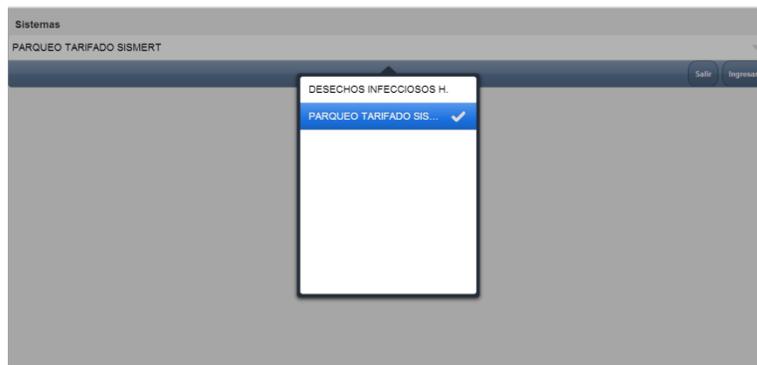


Figura 3. 39: Ventana de lista de aplicaciones en la aplicación móvil
Fuente: Propia

➤ **Historia 12: Consulta Tiempo Parqueo y Notificaciones.**

El Inspector tiene acceso a verificar el tiempo disponible que tiene un vehículo para estacionarse, si no tiene tiempo puede ingresar la notificación correspondiente.

Tareas:

1. Crear y configurar archivo de aplicación para SISMERT.
2. Crear controlador genérico para los servicios que se ofrecerá en la aplicación móvil.

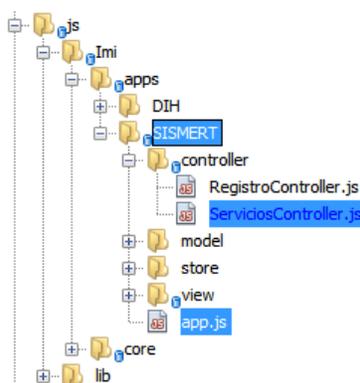


Figura 3. 40: Estructura MVC de la Aplicación Móvil SISMERT.
Fuente: Propia

Tareas:

3. Diseñar formularios para consultas de tiempo de parqueo.
4. Diseñar formulario para ingresar una notificación.

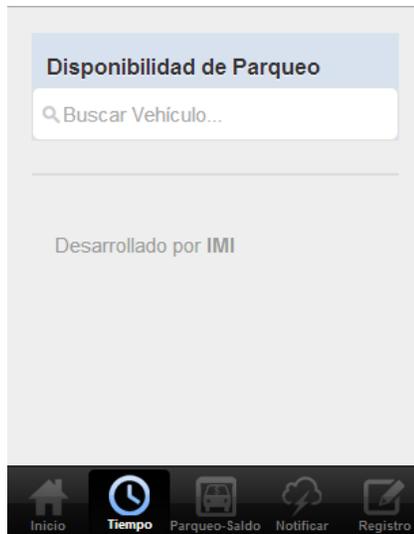


Figura 3. 41: Formulario de consulta de disponibilidad de estacionamiento.
Fuente: Propia

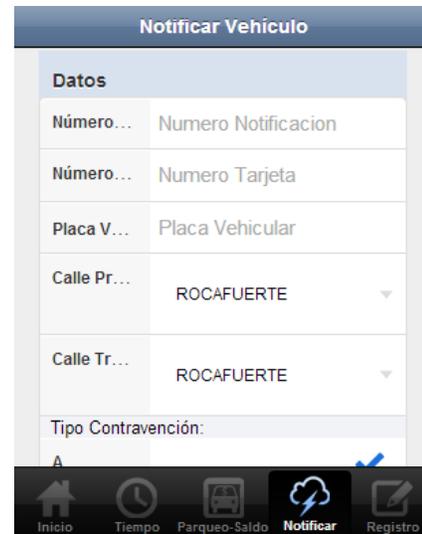


Figura 3. 42: Formulario de ingreso de notificaciones.
Fuente: Propia

En las Tareas 5, 6, 7, 8 y 9 se implementan las acciones en el servidor y cliente para en el envío y recepción de datos de consultas de tiempo e ingreso de notificaciones, a continuación se muestra las interfaces funcionales para estas opciones.

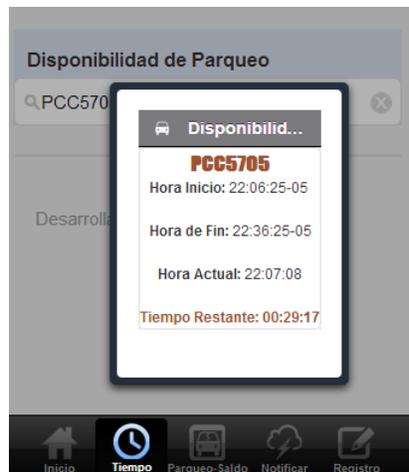


Figura 3. 43: Interfaz Funcional consultando Disponibilidad de Tiempo.
Fuente: Propia

La interfaz funcional para el ingreso de las notificaciones es similar al de la **Figura: 3.42**

➤ **Historia 13: Registro y Compras de Saldo en Aplicación Móvil.**

Al igual que el registro de teléfonos y vehículos realizados en las Iteraciones anteriores, ahora se crean estas opciones pero en la aplicación móvil reflejadas en las siguientes tareas.

Tareas 1, 5 y 6 creadas para el diseño de la interfaz e implementación de acciones para el registro de teléfonos.



**Figura 3. 44: Formulario Registro de Teléfonos en Aplicación Móvil.
Fuente: Propia**

Tareas 1, 7 y 8 creadas para el diseño de la interfaz e implementación de acciones para el registro de vehículos.



**Figura 3. 45: Formulario de registro de vehículos en aplicación móvil.
Fuente: Propia**

En las Tareas 3, 9 y 10 se crea la interfaz para realizar compras de saldo y las respectivas acciones para que sea funcional.

Compras de Saldo SISMERT

Datos

Teléfon... 0994364998

3 Dolares

Comprar

Inicio Tiempo Parqueo-Saldo Notificar Registro

Figura 3. 46: Formulario de compras de saldo en aplicación móvil.
Fuente: Propia

En las Tareas 4, 11 y 12 se desarrolla la opción para iniciar el servicio de estacionamiento de un usuario, esta petición lo realiza el inspector cuando el usuario así lo requiere.

Parquear Vehículo Ciudadano

Datos

Teléfon... 0994364998

Placa V... PCC5705

Tiempo ... 2

Tiempo: 1.00horas Costo: \$0.4

Estacionar

Inicio Tiempo Parqueo-Saldo Notificar Registro

Figura 3. 47: Formulario para iniciar servicio de estacionamiento en aplicación móvil.
Fuente: Propia

Después de realizar estas tareas se verificó su funcionalidad cumpliendo la Tarea 13. Esta es la solución más factible para que el Inspector verifique el tiempo de servicio de un vehículo.

3.9.7. Iteración VII (Módulo de Registro y Compras de Saldo)

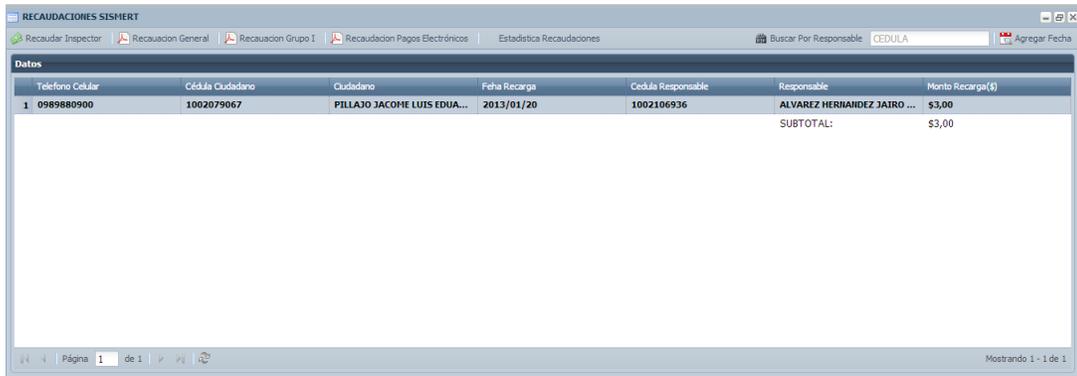
La Iteración se desarrolla para la revisión de las recaudaciones que se generan en el día por las compras o recargas de saldo. Esta revisión se realiza por que al momento de realizar una recarga esta transacción es ingresada en el modulo de recaudaciones del sistema integral del GAD-I, y al final del día el responsable del SISMERT tiene que cuadrar lo que se logró recaudar por medio de esta aplicación con el sistema de recaudaciones de la Institución.

➤ Historia 14: Recaudación de Compras de Saldo SISMERT.

En las Tareas 1 y 2 se realizó un análisis para aumentar los campos (responsable_recarga, tipo_recarga, estado) en la tabla ert_recargas para realizar el control de las recaudaciones, asignando un responsable por cada compra realizada y su estado, para que al final del día los registros que estén en estado 'I' son los que el inspector tiene que depositar al SISMERT para luego juntar todas las recaudaciones y depositarlas en el Sistema Integral de la Institución.

Tareas:

- 3.** Crear controlador para la revisión de recaudaciones y reportes.
- 4.** Crear modelo y Store de las recaudaciones.
- 5.** Crear la vista para mostrar las recaudaciones realizadas.



Telefono Celular	Cédula Ciudadano	Ciudadano	Fecha Recarga	Cédula Responsable	Responsable	Monito Recarga(\$)
1 0989880900	1002079067	PILLAJO JACOME LUIS EDUA...	2013/01/20	1002106936	ALVAREZ HERNANDEZ JAIRO ...	\$3,00
SUBTOTAL:						\$3,00

Figura 3. 48: Ventana de información y búsqueda de recaudaciones
Fuente: Propia

En las Tareas 6, 7, 8 y 9, se crea el formulario para recaudar al Inspector y las acciones correspondientes en el cliente (ExtJS) y el servidor (Symfony) para su funcionalidad.



Figura 3. 49: Ventana de recaudación diaria por Inspector.
Fuente: Propia

3.9.8. Iteración VIII (Módulo Web Ciudadano)

En la iteración se desarrolla el modulo de acceso web externo para que el ciudadano pueda ingresar a través de un cuenta de usuario a los servicios de registro de teléfonos y vehículos, reporte de uso del parqueadero, y la información para el uso del sistema.

➤ Historia 15: Servicios en la web para el ciudadano

En las Tareas 1 y 2 se realizó un análisis para verificar si es factible que la aplicación desarrollada para el ser usada en la Intranet, también sea publicada en la web para que el usuario acceda a los servicios

que ofrece la aplicación. Se clono la aplicación interna para realizar algunas modificaciones de acceso al sistema y pueda ser publicada en la web.

Para las Tareas 3, 4 y 5 se realizan el análisis de las seguridades que debe tener la aplicación, para que solo los usuarios registrados tengan acceso a este servicio.

Si el usuario intenta ingresar al servicio con una contraseña errónea tiene 4 intentos y a la próxima vez su cuenta se bloquea automáticamente durante 24 horas,

Para crear una cuenta de usuario primero hay que registrarse proporcionando los datos requeridos para el registro. Hay que tener en cuenta que solo las personas que están registrados como ciudadanos en el sistema Integral del GAD-I pueden crear una cuenta de usuario.

Para la restauración de clave de acceso se reutiliza los datos ingresados al momento del registro.

Tareas:

- 6. Modificar estructura de datos de ciudadano para crear campos del control de seguridad de usuario.
- 7. Realizar pruebas a nivel de base de datos con los nuevos campos creados.

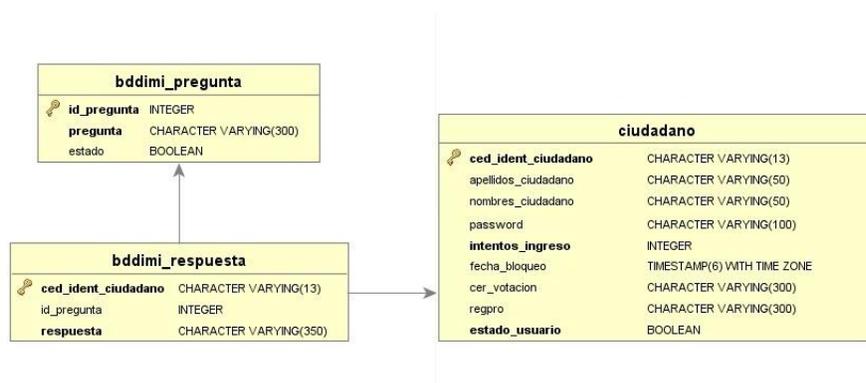


Figura 3. 50: Modelo relacional acceso web ciudadano
Fuente: Propia

Tareas:

- 8. Codificar el acceso de usuario en módulo de seguridades de la aplicación.
- 9. Realizar pruebas de acceso a la aplicación.

Para el acceso al sistema se ha modificado el modulo **sfGuardAuth** controlando que solo los usuarios registrados ingresen al sistema el formulario queda de la siguiente manera:

The image shows a login form titled "LOGIN" on a dark background. It features two input fields: "Usuario:" and "Contraseña:". Below these fields is a button labeled "Ingresar". At the bottom of the form, there are two links: "Crear cuenta de Usuario" with a "Registrar" button, and "Olvido su contraseña" with a "Restaurar" button.

Figura 3. 51: Formulario de Acceso a la Aplicación Web del Ciudadano.
Fuente: Propia

Tareas:

10. Diseñar formulario para el registro de un nuevo usuario.
11. Crear acciones en el cliente y servidor, para el registro de un nuevo usuario.
12. Realizar pruebas del registro de usuario.

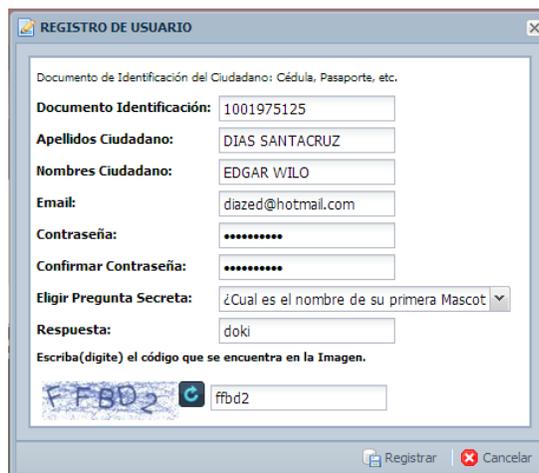
The image shows a registration form titled "REGISTRO DE USUARIO" in a window. It contains several input fields: "Documento Identificación:" (1001975125), "Apellidos Ciudadano:" (DIAS SANTACRUZ), "Nombres Ciudadano:" (EDGAR WILO), "Email:" (diazed@hotmail.com), "Contraseña:" (masked with dots), "Confirmar Contraseña:" (masked with dots), "Eligir Pregunta Secreta:" (dropdown menu with "¿Cual es el nombre de su primera Mascot"), "Respuesta:" (doki), and a CAPTCHA field with the text "Escriba (digite) el código que se encuentra en la Imagen." and the code "ffb2". At the bottom, there are "Registrar" and "Cancelar" buttons.

Figura 3. 52: Ventana de registro de Usuario Ciudadano.
Fuente: Propia

Tareas:

13. Crear acciones en el cliente y servidor, para restaurar clave de usuario.
14. Realizar pruebas de restauración de clave de acceso.



Figura 3. 53: Ventana Restaurar Usuario Ciudadano.
Fuente: Propia

En las Tares 15, 16, 17 y 18 se registra la aplicación ciudadano SISMERT con el menu de acceso al servicio, presentando una ventana más amigable al usuario como se presenta a continuación:

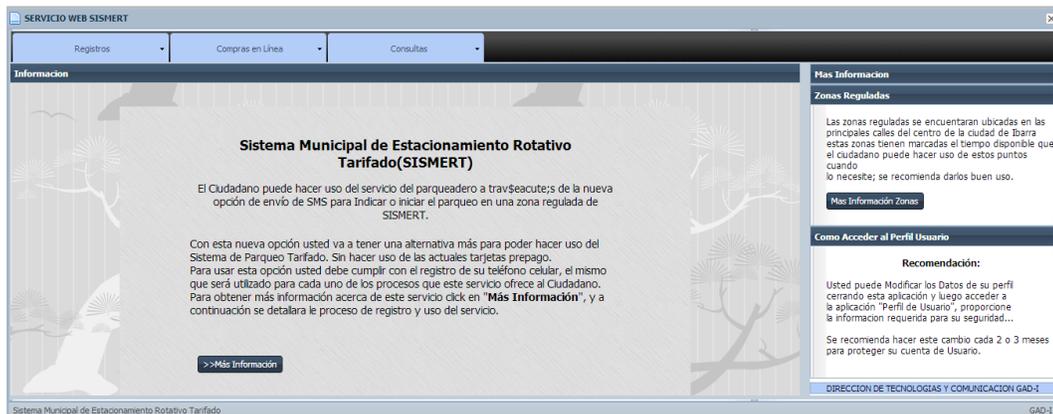


Figura 3. 54: Ventana Principal de la Aplicación Web Ciudadano.
Fuente: Propia

Las tareas 19, 20 y 21 realizan el mismo proceso que se realizó para el registro de teléfonos (Ver Figura 3.13) y registros de vehículos (Ver Figura 3.20). Además la opción de consultar el saldo disponible de un número de teléfono (Ver Figura 3.24)

Tareas:

22. Crear ventana para consulta del uso del parqueadero.
23. Crear acciones en el cliente y servidor para revisar el uso del parqueadero.



Figura 3. 55: Ventana para consulta de uso del parqueadero.
Fuente: Propia

Tareas:

- 24. Crear reporte PDF en Symfony para mostrar el uso del parqueadero.
- 25. Realizar pruebas de funcionalidad, y corregir errores existentes.

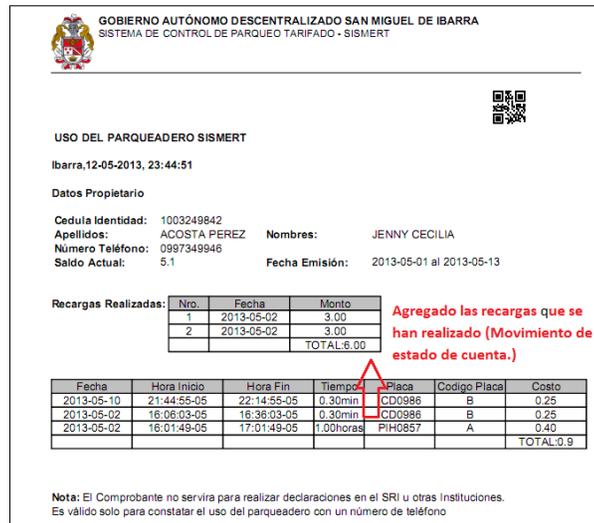


Figura 3. 56.: Ventana del Reporte en PDF de Uso del Parqueadero
Fuente: Propia

➤ Historia 16: Recargas SISMERT en Línea

Las tareas que cumple esta historia son creadas para crear de la interfaz grafica del proceso de compras en línea a través de una tarjeta de crédito.

El proceso no está completo por lo que únicamente llega hasta el envío de los datos al posible proveedor del Servicio.

Tareas:

1. Diseñar el formulario para realizar las compras en línea de recargas de saldo SISMERT.
2. Crear acciones en el controlador genérico para mostrar la ventana de recargas en línea.
3. Revisar el contenido de la ventana para el envío de datos al proveedor de transacciones electrónicas.



Figura 3. 57: Prototipo de Compras de Saldo en Línea.
Fuente: propia

3.9.9. Iteración IX (Módulo de Registro y Compras de Saldo)

La Iteración presenta el diseño de Reportes para Compras de Saldo, recaudaciones de los Inspectores y registros de teléfonos que constan en el sistema.

Todos los reportes se han creado con la herramienta iReport, y posteriormente se los integra en la aplicación desarrollada.

Para integrar los reportes creados en iReport hacia la aplicación PHP, el Core del GAD-I contiene una clase genérica para crear el Puente de conexión entre JAVA y PHP, utilizando la librería PHP-JRU para generar los reportes .jasper en PDF.

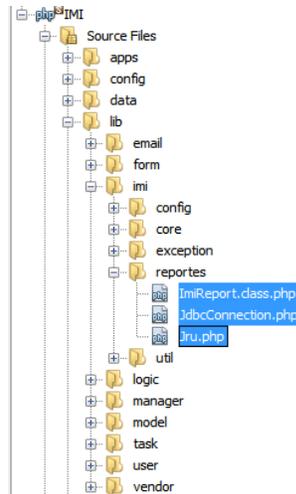


Figura 3. 58: Estructura de la Clase Genérica ImiReport para Generar Reportes.
Fuente: propia

➤ Historia 7: Compras de Saldo, Información Registros

En la Tarea 1 se registran submenús (ítems), en los menús que se ha venido registrando en el proceso de creación de la aplicación. Estos ítems ejecutan la acción para mostrar los reportes.

Tareas:

2. Diseñar reporte para uso del parqueadero por año en iReport.
3. Integrar el reporte uso parqueo en la aplicación.

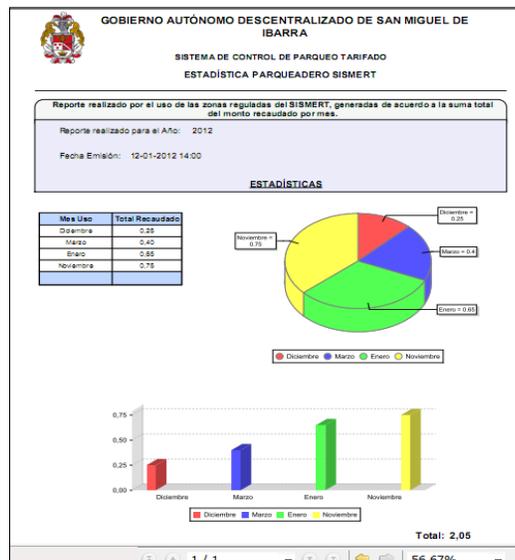


Figura 3. 59: Reporte Estadístico de Uso del Parqueadero por Año
Fuente: Propia

Tareas:

4. Diseñar reporte para registro de teléfonos en iReport.
5. Integrar el reporte registro teléfonos en la aplicación.

GOBIERNO AUTÓNOMO DESCENTRALIZADO DE SAN MIGUEL DE IBARRA
SISTEMA DE CONTROL DE PARQUEO TARIFADO
REGISTROS DE SISMERT

Datos de Los Registros, que se encuentran presentes en SISMERT, para el respectivo Uso del Parquero Tarificado de la ciudad de Ibarra

El Reporte Se Ha Realizado en las Siguietes Fechas
 Desde el: 2012-09-01 Hasta el: 2013-01-10
 Fecha Emisión: 12-01-2013 16:00

LISTADO DE REGISTROS

Fecha	Cédula	Apellidos Ciudadano	Nombres Ciudadano	Operadora	Num.	Email
1	101213000	SAVIOZ VADIA	EDWIN BARTAGO	CLARO	099827497	ibarras@smi.gov.ec
2	101213000	LUIS	QUIMAN	CLARO	0994398732	ibarras@smi.gov.ec
3	101213000	LUIS	QUIMAN	NOVIETAR	0994398732	ibarras@smi.gov.ec
4	101213000	LUIS	QUIMAN	CLARO	0994398732	ibarras@smi.gov.ec
5	101213000	BUPHEA	BINTOCHRY S.A.	CLARO	097797699	ibarras@smi.gov.ec
6	101213000	TEBODRA ZHISPUJUN	JOSÉ POLVIO	CLARO	0994398732	ibarras@smi.gov.ec
7	101213000	PILLAO JACQUE	LUIS EDUARDO	NOVIETAR	099830300	ibarras@smi.gov.ec
8	101213000	TEBODRA ZHISPUJUN	JOSÉ POLVIO	NOVIETAR	099778000	ibarras@smi.gov.ec
9	101213000	BUPHEA	BINTOCHRY S.A.	CLARO	099839847	ibarras@smi.gov.ec
10	101213000	ESCOBAR TERAN	RYTA LORENA	CLARO	09988389	ibarras@smi.gov.ec
11	101213000	HEA LOZADA	ROSA ANDREA	NOVIETAR	099748799	ibarras@smi.gov.ec
12	101213000	PILLAO JACQUE	LUIS EDUARDO	ONT	0994398732	ibarras@smi.gov.ec
13	101213000	NAVARRETE LARA	EDUARDO FRAVUJUN	CLARO	099203847	ibarras@smi.gov.ec
14	101213000	MEACOS PAREDES	RUIZ MARLON	CLARO	099830300	ibarras@smi.gov.ec
15	101213000	ESBANO SUBIA	EDYLA EUPENATREZ BUCUJA	CLARO	099830300	ibarras@smi.gov.ec
16	101213000	ESCOBAR TERAN	RYTA LORENA	CLARO	09988389	ibarras@smi.gov.ec

1 / 1 56.67%

Figura 3. 60: Reporte de Registros de Teléfonos SISMERT.
 Fuente: Propia

➤ **Historia 14: Recaudación de Compras de Saldo SISMERT**

Tareas:

1. Diseñar reporte para recaudaciones de los inspectores por día.
2. Integrar el reporte de recaudaciones de inspectores.

GOBIERNO AUTÓNOMO DESCENTRALIZADO DE SAN MIGUEL DE IBARRA
SISTEMA DE CONTROL DE PARQUEO TARIFADO
RECAUDACIONES INSPECTORES SISMERT

Datos de Recaudaciones Inspectores SISMERT, para el Uso del Parqueadero Tarifado de la ciudad de Ibarra.
 Fecha de Emisión: 12-01-2013 18:00 Venta de Tiempo: 1, 3, 5 y 10 Dólares
 Día Recaudación: 2012-12-20

	Cédula	Responsable	Recaudado
1	100475694	AQUIRRE ENRIQUEZ BRYAN ESTEBAN	10.0
2	1002106936	ALVAREZ HERNANDEZ JAIRO GUSTAVO	7.0
3	1001667706	BAEZ FLORES PABLO JAVIER	1.0
4	1003765731	BENAVIDES TAPIA KLEVER ROBERTO	1.0
5	1002453429	BOLAÑOS PUENTE DANIEL MANUEL	6.0
6	1003209466	BURGOS PANTOJA PATRICIO ALEXANDER	10.0
7	1002980769	CAJENA BUBIA DAVID SANTIAGO	3.0
8	1002208787	CHICAIZA CARRILLO BALERIA DEL ROSARIO	1.0
9	1003169470	FLORES LOPEZ ANABELASBETH BIBE	6.0
10	1002172854	FRACIOS JACOME CRISTIAN PIER	3.0
11	1001997921	PORTILLA CHACON JUAN FRANCISCO	10.0
12	1003693254	REALPEALVAREZ EDWIN RAMIRO	4.0
13	1002884771	RIVERA PEÑAFIEL PABLO MAURICIO	6.0
14	1001768636	SILAB MAPLA EDWIN EDUARDO	1.0
15	1002171600	SOLANO MORAN JORGE FERNANDO	1.0
16	1003022371	TULCANAZ IMBAQUINSO EDWIN MANUEL	1.0
17	1003166012	VALENCIA GORDILLO MICHAEL SANTIAGO	4.0
Total Recaudado: \$			73.00

Figura 3. 61: Reporte Recaudaciones a Inspectores.
Fuente: Propia

Finalmente en las Tareas 3 y 4 se diseñaron reportes estadísticos para saber cual mes durante un año se obtuvo una mayor recaudación. Además varios reportes para verificar las recaudaciones teniendo la opción de obtener los registros por responsable.

Figura 3. 62: Ventana de Búsqueda de Recaudaciones que Genera Reportes.
Fuente: Propia

3.9.10. Iteración X (Módulo de Registro y Compras de Saldo)

La Iteración se desarrolla para la implementación de envío de notificaciones SMS cuando se inicia el servicio de estacionamiento.

➤ **Historia 9: Notificaciones Inspector y Ciudadano**

El sistema de mensajería de la Dirección TIC del GAD-I es capaz de recibir un SMS que el usuario envía a un número de celular fijo, también realiza el envío de mensajes de texto cuando se recibe una petición en el Sistema.

En las Tareas 1, 2, 3, 4, 5 y 6 se desarrolla el envío y recepción de mensajes cuando el sistema recibe una petición solicitada al Sistema de Control de Parqueo Tarifado.

Para la integración de la aplicación de mensajería con las opciones que tiene el sistema de control de parqueo tarifado se creó dos servicios web.

El primer servicio web se encarga de recibir la petición de un usuario e interpretar el servicio que está solicitando.

Servicios interpretados por el sistema:

Servicio	Contenido SMS	Descripción
Servicio de Parqueo	P PCC5705 1 P A 1	Iniciar el servicio de estacionamiento: Donde : P: Código del servicio. A (PCC5705): Código o placa vehicular. 1: Tiempo 0.30min.
Registro de Vehículo	R PCC5705 L 1004590121 R PCC5705 P 1004590121	Registrar una placa vehicular R: Código de servicio (Registro). PCC5705: Placa a registrarse. L 'o' P: Tipo vehículo Liviano o Pesado. 1004590121: Cedula del propietario del vehículo.
Consulta de Saldo	S	Consultar saldo disponible S: Código de consulta

Tabla 3. 60: Servicios SMS interpretados por el Sistema.

Fuente: Propia

El sistema solo recibe las peticiones enviadas desde un número de teléfono registrado, caso contrario la petición no es atendida.

El contenido del mensaje texto debe estar separado por espacios en blanco para que el sistema pueda interpretar el servicio solicitado.

El segundo servicio web se encarga del envío de mensajes de texto luego de haber interpretado el servicio solicitado por el usuario.

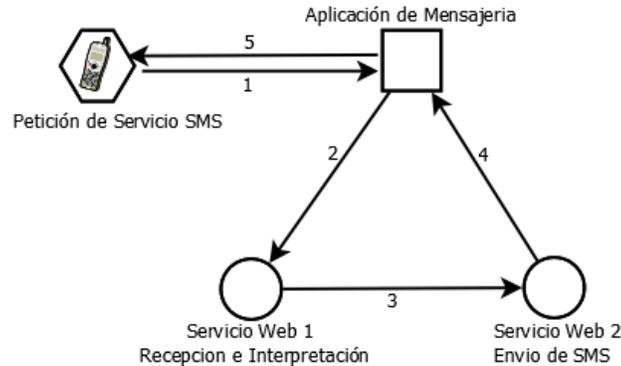


Figura 3. 63: Estructura funcional de envío y recepción de SMS.
Fuente: Propia

Como se puede ver en la figura anterior el usuario envía la petición que es receptada por la aplicación de mensajería, para activar el servicio web 1 que funciona como receptor e interpretador de la solicitud del usuario.

Luego de interpretar el servicio solicitado se activa el servicio web 2 para enviar el SMS de notificación ingresando nuevamente a la aplicación móvil, y finalmente el envío del mensaje de notificación al usuario.

Este es el proceso implementado para la recepción y envío de SMS, con el servicio desarrollado se completa los servicios propuestos para el control del parqueadero tarifado.

3.10. Subir cambios de versiones al repositorio GIT

Las aplicaciones que se encuentran en el core son versionadas con el repositorio para mantener un historial de versiones y mantener un respaldo de cada aplicación.

Luego de haber terminado una iteración es necesario subir el proyecto al repositorio GIT, con la finalidad de mantener un respaldo de la aplicación.

El Procedimiento para reflejar los cambios en el repositorio es el siguiente:

1. Verificar que los archivos a subir sean los correctos.
2. Ejecutar el comando **Commit** para guardar todos los cambios generados en la aplicación.

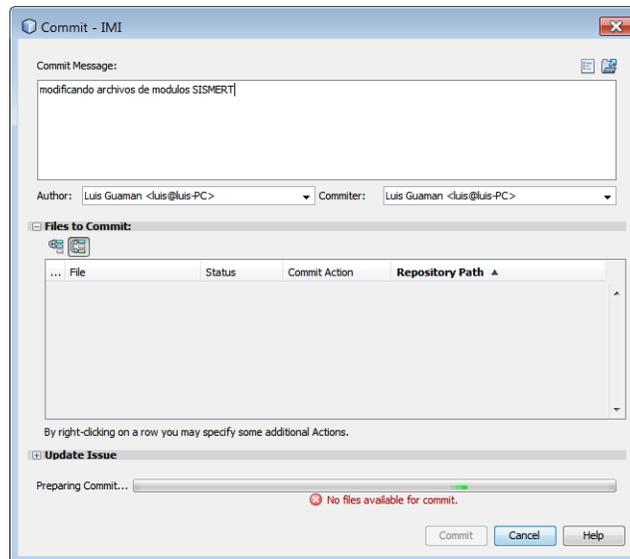


Figura 3. 64: Comando Commit para versionar el proyecto.
Fuente: Propia

3. Ejecutar el comando **Pull** para enviar y traer datos del servidor. El comando se lo puede dividir en dos tareas a raves de los comando **Fetch** y **Merge** que realizan la misma función.

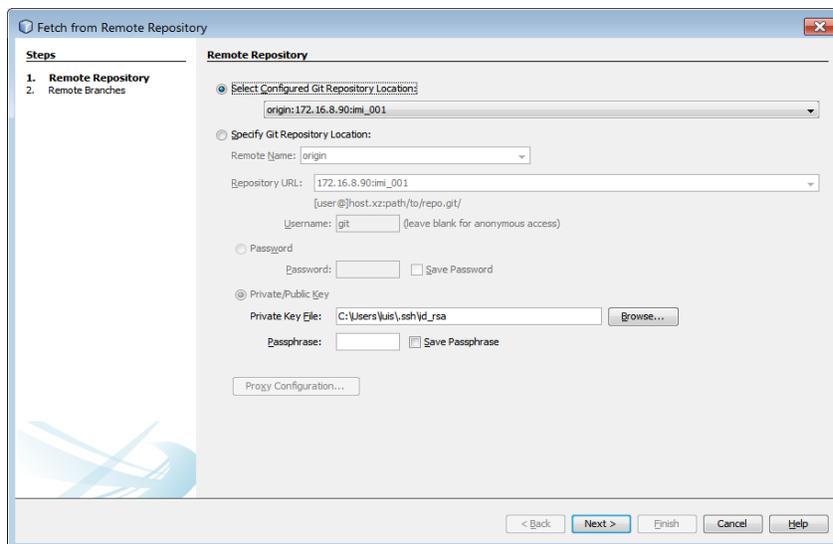


Figura 3. 65: Comando Fetch para obtener datos del repositorio remoto del GIT.
Fuente: Propia

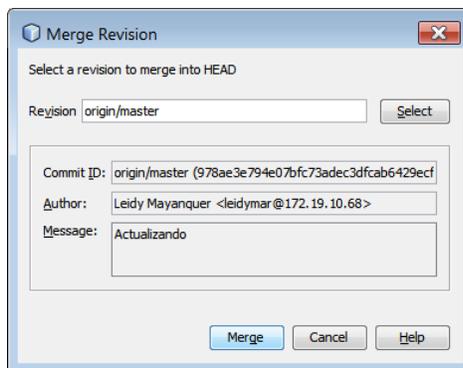


Figura 3. 66: Comando Merge para enviar datos al repositorio remoto del GIT.
Fuente: Propia

4. Ejecutar el comando **Push** para ingresar los cambios de la aplicación en el repositorio de versiones.

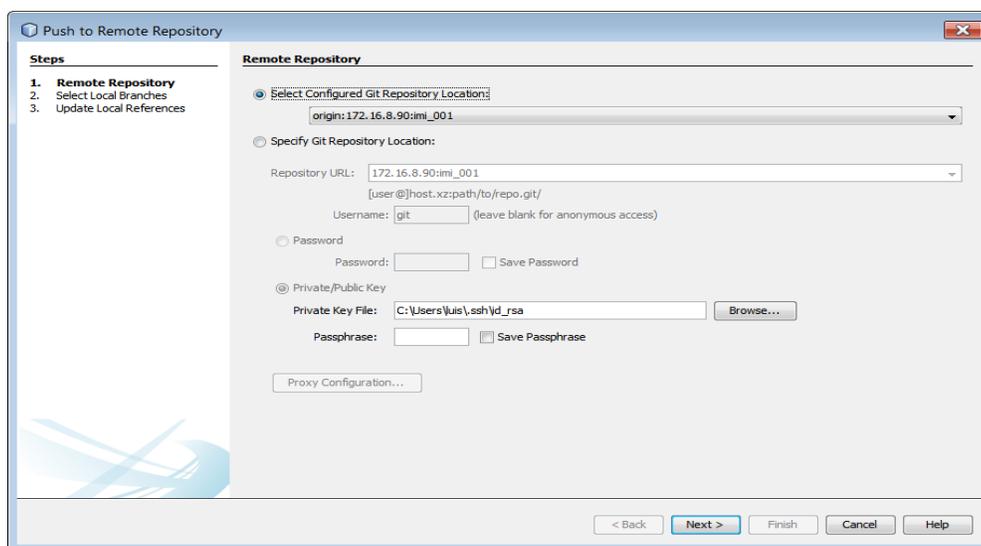


Figura 3. 67: Comando Push para ingresar cambios en el repositorio del GIT.
Fuente: Propia

Con los comandos explicados se realizó el proceso para versionar la aplicación, manteniendo la seguridad al tener un respaldo de los cambios realizados en el transcurso del desarrollo del sistema.

4. Capítulo IV

Conclusiones y Recomendaciones

4.1. Análisis Costo Beneficio

El análisis se lo realiza para determinar el costo que implica el desarrollo del sistema y los beneficios que presenta su realización.

La implementación del sistema se lo ha desarrollado con herramientas de software libre con la finalidad de que el costo de la aplicación no sea muy elevado. Además implica el beneficio de reducción del tiempo que un inspector se demora para comprobar si un vehículo tiene tiempo disponible para usar el servicio del SISMERT.

A continuación se presenta los datos de Costo-Beneficio que implica la creación del Sistema de Control de Parqueo Tarifado.

Costos	Beneficios
<ul style="list-style-type: none">Herramientas de software libre \$0Contratación de servicio de transacciones electrónicas (indefinido).Adquisición de dispositivos móviles $40(\text{dispositivos}) * \\$200 = \\$8000$.Paquete de mensajería a 1 centavo x SMS.Adquisición de dispositivos inalámbricos (Routers) $30(\text{dispositivos}) * 250 = 7500$.Programador $\\$700 * 9(\text{Meses}) = \\6300.	<ul style="list-style-type: none">Reducción de tiempo durante la verificación de disponibilidad de uso del servicio del SISMERT.Usuario es independiente del inspector para comprar tiempo de estacionamiento.Inspector ingresa notificaciones en tiempo real sin la necesidad de ingresar todas las notificaciones al final de la jornada de trabajo.El usuario será controlado de mejor manera y se recaudará un porcentaje mayor al actual por el costo de las infracciones.El control permite aumentar las ganancias al mantener un mejor control

TOTAL IMPLEMENTACION
EQUIPOS TECNOLOGICOS: \$15500
TALENTO HUMANO: \$6300
TOTAL =\$21800
Mantenimiento Mensual por SMS: \$60

- por que los tiempos serán más rotativos para acceder al servicio.
- Ciudadano (usuario) conforme con el servicio al tener una opción más para hacer uso del SISMERT.
 - El usuario puede tener almacenado su saldo para que lo pueda ocupar cuando va a utilizar el servicio de estacionamiento debitándose de acuerdo al tiempo de uso del servicio.

Análisis Costo Beneficio

La aplicación es publicada por el gestor de contenidos de la dirección de TIC, esto permite que este costo no sea reflejado. El costo para la contratación del servicio de transacciones electrónicas para pagos en línea no es tan beneficioso porque la gran mayoría de los usuarios puede que no accedan a realizar las recargas de saldo por este medio, de tal manera que el servicio de pagos en línea sería innecesario.

Para que el usuario tenga otro medio de compras o recargas de saldo lo puede hacer también por medio de los inspectores con la aplicación móvil que posee cada uno de ellos o acercándose al GAD-I. Este saldo es muy importante porque el usuario no tiene que comprar tiempo de parqueo a cada instante que necesite utilizar el servicio, si no que podrá almacenarlo y utilizarlo cuando lo necesite debitándose solo del tiempo que ha utilizado el servicio.

Para la adquisición de los dispositivos móviles e inalámbricos el costo que implica su compra se lo ha referenciado al valor anual que el SISMERT recauda teniendo los siguientes datos:

Recaudación Anual SISMERT	Compras dispositivos móviles e inalámbricos
\$300000 equivalentes al 100%	\$8000 equivalentes al 2.6 %
	\$7500 equivalente al 2.5%
	Costo Total Adquisición: \$15500
	Porcentaje Total Anual: 5.16%

Costos que implica comprar recursos tecnológicos.

Cabe señalar que el costo de la adquisición de estos dispositivos será reflejado solo en el primer año y no representa un costo muy elevado por los beneficios que se obtiene al realizar esta aplicación. Para descontar el valor del costo de estos dispositivos serán debitados de las recaudaciones que se realicen por el uso del servicio.

El costo que representan los SMS son para el envío de notificaciones luego que el usuario inicia el servicio de parqueo, este costo es igual al que se emplea para la impresión de las tarjetas prepago que se utilizan actualmente para el control de este servicio, justificando así este costo de 1ctvs por mensaje luego de que el usuario usa el servicio de estacionamiento.

Las recaudaciones del SISMERT pueden aumentar un 5% más de lo que se logra recaudar actualmente beneficiando al SISMERT no inmediatamente hasta que el sistema se regularice pero más adelante se obtendrán ganancias más altas por el uso de este servicio.

4.2. Impactos

El sistema desarrollado aumenta la efectividad del control del parqueadero, reduciendo el tiempo de comprobación cuando el Inspector acude al lugar donde se encuentra un vehículo estacionado para verificar el tiempo disponible de uso de una zona regulada del SISMERT.

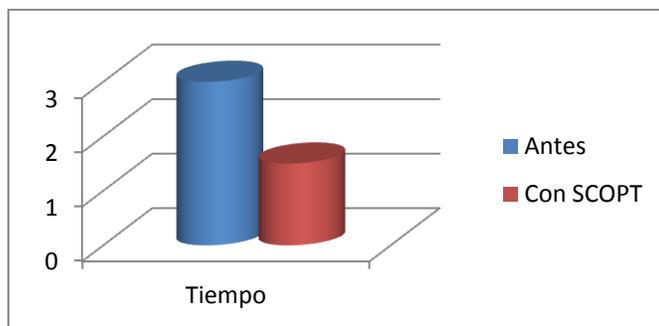


Figura 4. 1: Tiempo de Control del Parqueadero
Fuente: Propia

Las recaudaciones por el uso del estacionamiento son más seguras por que el dinero obtenido al comprar tiempo es reflejado automáticamente en el sistema integral de recaudaciones del GAD-I a través de la aplicación móvil.

El usuario (ciudadano) tiene la opción de almacenar el saldo de las recargas realizadas, para posteriormente hacer uso de este, cuando el usuario así lo requiera. Evitando tener contacto con el inspector para la compra de una tarjeta prepago.

Ejemplo: Si el usuario realiza una recarga de 3 dólares y va a permanecer estacionado 0.30 minutos el saldo actual será 2.75 dólares.

4.3. Conclusiones

- Para el pago electrónico es indispensable utilizar un proveedor de servicios de transacciones electrónicas y un firewall para que los datos de las transacciones sean seguros.
- Los protocolos de transferencia de datos en la red son indispensables para las transferencias electrónicas en un servicio.
- Los servicios web en una aplicación ayudan a ejecutar diferentes servicios entre aplicaciones como el envío de SMS desde la aplicación de mensajería de la dirección de TIC.
- Es importante interpretar el contenido enviado de un SMS para iniciar el uso del parqueadero o utilizar uno de los servicios que ofrece el sistema, para su posterior envío de notificaciones de acuerdo al servicio solicitado por el usuario.
- El control del parqueadero es más eficaz al reducir el tiempo de comprobación de la disponibilidad del tiempo de un vehículo estacionado.
- Con el desarrollo de la aplicación móvil se tiene acceso al ingreso de notificaciones en tiempo real, cuando un vehículo ha infringido una zona regulada del SISMERT.
- El envío de SMS para realizar una petición al sistema es muy importante porque el usuario tiene acceso al servicio de estacionamiento con tan solo enviar un mensaje de texto.
- La implementación de un reporte del uso del parqueadero al final de cada mes es muy útil para el usuario porque puede verificar la utilización del servicio de estacionamiento en un mes seleccionado.
- La arquitectura del sistema es flexible y escalable al tener N-Capas por lo que permite manejar el lado del cliente y servidor por separado, esto hace que el sistema sea más robusto.
- El uso de herramientas de software libre ha permitido que el costo del desarrollo del sistema sea reducido.

- Las aplicaciones que utilizan tecnología RIA son más rápidas, más livianas, y son muy amigables al tener una interfaz de usuario de fácil manipulación.
- La Metodología de Desarrollo implantada por el GAD-I se lo ha diseñado con la finalidad de ser eficiente y rápida para el desarrollo de proyectos Tecnológicos.
- El envío de SMS hacia los inspectores para informarles que un vehículo se ha estacionado en una zona regulada no es tan factible económicamente por que este mensaje debería llegar a los 32 Inspectores y el costo del SMS es muy elevado provocando pérdidas económicas al SISMERT por el uso del parqueadero.
- La mejor manera para comprobar que un vehículo está estacionado en una zona regulada y tiene tiempo disponible es a través de la aplicación móvil.
- Es importante que el usuario conozca cada uno de los servicios que presenta el sistema para su correcta funcionalidad.

4.4. Recomendaciones

- Se recomienda que para el proceso de pagos electrónicos la institución adquiriera un firewall para controlar la seguridad de los datos del usuario.
- Capacitar a los usuarios (ciudadanos) sobre el contenido de los SMS para que al momento de escribir su mensaje y solicitar un servicio, este no sea mal escrito y el sistema no podrá interpretar esta petición.
- Para que el ciudadano pueda acceder a este servicio es necesario informarle que el primer paso a seguir es el registro de sus teléfonos y placas vehiculares, una vez realizado el registro el ciudadano debe realizar una compra de saldo para que pueda utilizar el servicio de estacionamiento.
- Es necesario socializar con las personas (Ciudadanos) para que se enteren del nuevo servicio implantado por el SISMERT y el tipo de peticiones que pueden realizar a través de un mensaje de texto.
- El ciudadano debe contar con un documento de identificación (cédula o pasaporte) registrado en el GAD-I para crear una cuenta de usuario en el sistema desarrollado.
- El reporte que el usuario adquiere por el uso del parqueadero por mes no se lo debe utilizar para realizar declaraciones al SRI.
- Es importante que el usuario revise la documentación de los manuales de usuario para que el sistema tenga una correcta funcionalidad.

- Es necesario adquirir los equipos tecnológicos para que la aplicación móvil se suba a producción, porque el control solo con SMS no es suficiente.
- Hacer uso de buenas prácticas de programación y patrones de diseño, permiten implementar aplicaciones de fácil mantenimiento y sobre todo escalables.
- La Metodología de desarrollo del GAD-I debe ser más específica para su fácil uso en el llenado de ciertos campos de los artefactos utilizados en esta metodología.
- Si se va a realizar una aplicación integral es necesario que la institución que lo realiza establezca como primer punto primordial la estructura de la arquitectura del sistema.
- Se recomienda a la Institución del GAD-I reservar entre sus actividades un tiempo para la revisión y seguimiento de los proyectos de Tesis porque el tiempo brindado para esta aplicación no fue lo suficiente, y por esta razón hubo algunos cambios pero al final se concluyo con la aplicación.
- Se recomienda que la institución académica brinde mas información acerca de los frameworks de desarrollo web de software libre y también proporcione información acerca de las metodologías ágiles para el desarrollo de Software orientado a objetos.
- La Institución académica debería añadir en su malla curricular una materia sobre la programación orientada a dispositivos móviles, por que actualmente las aplicaciones móviles constituyen un eje muy importante en la sociedad y está en constante crecimiento.
- Es recomendable establecer los recursos tecnológicos de hardware y software para que una aplicación se construya en su totalidad y poder cumplir con todos los objetivos planteados en el anteproyecto de Tesis.

Glosario de Términos

ACID: Atomicidad, Consistencia, Aislamiento y Durabilidad.

BasesfGuardAuthActions : Clase Controlador, permite el control de ingreso al sistema.

BSD: Distribución de Software Berkeley, es un sistema operativo derivado del sistema Unix.

DDL: Lenguaje de Definición de Datos.

DHTML: Dynamic HTML Usado para crear efectos de páginas web Dinámicas.

DOM: Document Object Model.

FODA: Fortalezas, Oportunidades, Amenazas, Debilidades.

GAD-I: Gobierno Autónomo Descentralizado de Ibarra.

GPS: Global Positioning System: sistema de posicionamiento global.

HTML: Hypertext Markup Language.

HTTP: Protocolo de Transferencia de Hipertexto.

OASIS: Acrónimo de Organization for the Advancement of Structured Information Standards.

ORM: Object-Relational Mapping.

PhoneGap: Usado para distribuir aplicaciones en la App Store de Sencha Touch o en Android Marketplace.

PostgreSQL: Gestor de Base de Datos.

SASS: style with attitude, es una extensión de CSS.

SCOPT: Sistema de Control de Parqueo tarifado.

Servlet: La palabra servlet deriva de otra anterior, applet, que se refería a pequeños programas que se ejecutan en el contexto de un navegador web.

SISMERT: Sistema Municipal de Estacionamiento Rotativo Tarifado.

SMTP: (Simple Mail Transfer Protocol), para el correo electrónico.

SQL: lenguaje de consultas estructurado.

Store: Conjunto o Almacén de Datos.

TCP: Protocolo de Control de Transmisión.

TIC: Tecnologías de la información y Comunicaciones.

XML-RPC: Protocolo de llamada a procedimiento remoto, antecesor de SOAP.

Bibliografía

LIBROS

[LIB1]: Camps Paré, R., Casillas Santillán, L. A., Costal Costa, D., Gibert Ginestá, M., Martín Escofet, C., & Pérez Mora, O. (2007). *Base de Datos*. Barcelona: Fundación para la Universidad Oberta de Catalunya.

[LIB2]: Pérez, J. E. (2009). *CSS Avanzado*. librosWeb.

[LIB3]: Goodman, D. (2001). *JavaScript Examples Bible*. New York: JavaScript Bible.

[Lib4]: Chacon, S. (2009). *Git*.

[LIB5]: Francois Zaninotto, F. P. (2008). *Symfony La Guia Definitiva*. librosWeb.

[LIB6]: Corporation, J. (2010). *iReport Ultimate Guide*. Jaspersoft.

[LIB7]: Corporation, J. (2002-2010). *JasperReports Ultimate Guide*. Jaspersoft.

[LIB8]: Groner, L. (2011). *Ext JS 4 First Look*. Brazil: Usha Iyer.

PUBLICACIONES EN LÍNEA

[WEB1]: PostgreSQL. (10 de Febrero de 2010). *PostgreSQL-es*. Recuperado el 2012, de Sobre PostgreSQL: http://www.postgresql.org/es/sobre_postgresql

[WEB2]: Servidor HTTP Apache. (14 de Noviembre de 2012). *Wikipedia*. Recuperado el 2012, de Servidor HTTP Apache: http://es.wikipedia.org/wiki/Servidor_HTTP_Apache

[WEB3]: Tomcat. (29 de Octubre de 2012). *Wikipedia*. Recuperado el 2012, de Tomcat: <http://es.wikipedia.org/wiki/Tomcat>

[WEB4]: Eguiluz Pérez, J. (s.f.). *Introducción a CSS*. Obtenido de Capítulo 1 Introducción ¿Que es CSS?: <http://www.librosweb.es/css/capitulo1.html>

[WEB5]: Hojas de Estilo en Cascada, E. (19 de Octubre de 2012). *Wikipedia*. Recuperado el 2012, de Hojas de estilo en cascada: http://es.wikipedia.org/wiki/Hojas_de_estilo_en_cascada

[WEB6]: PHP. (17 de Noviembre de 2012). *Wikipedia*. Recuperado el 2012, de PHP: <http://es.wikipedia.org/wiki/PHP>

[WEB7]: JavaScript. (20 de Noviembre de 2012). *Wikipedia*. Recuperado el 2012, de JavaScript: <http://es.wikipedia.org/wiki/JavaScript>.

[WEB8]:Características JavaScript. (s.f.). *Manual JavaScript*. Recuperado el 2012, de Características: http://www.uazuay.edu.ec/estudios/sistemas/lenguaje_iii/MANUALJavaScript/caracteristicas.htm

[WEB9]:Framework. (31 de Octubre de 2012). *Wikipedia*. Recuperado el 2012, de Framework: <http://es.wikipedia.org/wiki/Framework>

[WEB10]:Eguiluz, J. (s.f.). *symfony.es*. Recuperado el 2012, de Que es Symfony: <http://www.symfony.es/que-es-symfony/>

[WEB11]:Doctrine (PHP). (10 de Julio de 2012). *Wikipedia*. Recuperado el 2012, de Doctrine (PHP): http://es.wikipedia.org/wiki/Doctrine_%28PHP%29

[WEB12]:Ext JS. (14 de Agosto de 2012). *Wikipedia*. Obtenido de Ext JS: http://es.wikipedia.org/wiki/Ext_JS

[WEB13]:Arquitectura MVC. (2012). *Sencha Ext JS*. Recuperado el 2012, de Whats new in ext js 4: <http://www.sencha.com/products/extjs/whats-new-in-ext-js-4/>

[WEB14]:Aplicaciones Sencha Touch. (11 de Octubre de 2012). *Maestros del Web*. Recuperado el 2012, de Aplicaciones móviles con Sencha Touch: <http://www.maestrosdelweb.com/editorial/aplicaciones-moviles-sencha-touch>

[WEB15]:Soporte Sencha Touch. (2012). *Sencha Touch*. Recuperado el 2012, de Que es Sencha Touch: <http://www.sencha.com/products/touch>

[WEB16]:Introducción Descripción Secha Touch. (02 de Septiembre de 2012). *Pharalax Blog*. Recuperado el 2012, de Introduccion, Descripcion de Sencha Touch y La Primera Aplicacion: <http://pharalax.com/blog/sencha-touch-introduccion-descripcion-de-sencha-touch-y-la-primera-aplicacion>

[WEB17]:JaspertRepor e Ireport. (09 de Marzo de 2010). *Programacion Visual*. Recuperado el 2012, de Manual de Ireport Basico: http://www.campus.fi.unju.edu.ar/main/document/document.php?curdirpath=%2Frecursos_jaser-ireport&cidReq=

[WEB18]:Wikipedia, l. e. (22 de Octubre de 2012). *Wikipedia*. Recuperado el 2012, de Servicio web: http://es.wikipedia.org/w/index.php?title=Servicio_web&oldid=60762191

[WEB19]:Wikipedia(XML), l. e. (11 de Noviembre de 2012). *Extensible Markup Language*. Recuperado el 2012, de XML: http://es.wikipedia.org/w/index.php?title=Extensible_Markup_Language&oldid=61275434

[WEB20]:Wikipedia(WSDL), l. e. (23 de Octubre de 2012). *Wikipedia*. Recuperado el 2012, de WSDL: <http://es.wikipedia.org/w/index.php?title=WSDL&oldid=60802059>

[WEB21]:Wikipedia(SOAP), l. e. (19 de Octubre de 2012). *Simple Object Access Protocol*.

Recuperado el 2012, de SOAP:

http://es.wikipedia.org/w/index.php?title=Simple_Object_Access_Protocol&oldid=60679323

[WEB22]:ExtJS Comunidad en español. (27 de Abril de 2012). *ExtJS*. Recuperado el 2012, de

Arquitectura de la aplicación MVC: <http://www.extjs.mx/2012/01/arquitectura-de-la-aplicacion-mvc>

[WEB23]:PRO GIT. (s.f.). *PRO GIT, Empezando*. Recuperado el 2012, de Acerca del control de

versiones: <http://git-scm.com/book/es/Empezando-Acerca-del-control-de-versiones>

[WEB24]:PRO GIT, F. (s.f.). *PRO GIT,Empezando* . Recuperado el 2012, de Fundamentos de Git:

<http://git-scm.com/book/es/Empezando-Fundamentos-de-Git>

[WEB25]:desarrolloweb. (s.f.). *DesarrolloWeb.com*. Recuperado el 2012, de Comenzamos a utilizar

NuSOAP: <http://www.desarrolloweb.com/articulos/1884.php>

[WEB26]:Proceso. (5 de 12 de 2012). *Wikipedia*. Recuperado el 2012, de Wikipedia, la enciclopedia

libre: <http://es.wikipedia.org/w/index.php?title=Proceso&oldid=61961747>

[WEB27]:INTERDIN. (2005). *INTERDIN*. Obtenido de Portal de Servicios:

<https://www.optar.com.ec/Optar.Static/empresa.html>

[WEB28]:Secure Electronic Transaction. (12 de 07 de 2012). *Wikipedia*. Obtenido de SET:

<http://es.wikipedia.org/wiki/SET>

[WEB29]: Secure Sockets Layer. (18 de 01 de 2013). *Wikipedia*. Obtenido de SSL:

<http://es.wikipedia.org/wiki/SSL>

[WEB30]: Paypal. (2013). *Paypal*. Obtenido de Que es Paypal: <http://www.paypal.com/ec>

[WEB31]:Martinez, R. (10 de Febrero de 2010). *PostgreSQL-es*. Recuperado el 2013, de Sobre

PostgreSQL: http://www.postgresql.org.es/sobre_postgresql

[WEB32]: Zulca M, J. (02 de Diciembre de 2008). *emagister*. Recuperado el 2013, de Arquitectura de

base de datos para la web: <http://www.emagister.com/curso-php-mysql-sql-8/arquitectura-base-datos-web>

[WEB33]: computing, C. (Jueves de Julio de 2010). *migrate2Cloud*. Recuperado el 2013, de Apache-

Tomcat Load Balanced Persistent Session Setup on Amazon EC2:
<http://www.migrate2cloud.com/blog/apache-tomcat-load-balanced-persistent-session-setup-on-amazon-ec2>

[WEB34]: Garcia, J. (10 de Septiembre de 2012). *Arquitecto IT*. Recuperado el 2013, de Ejemplo de

Layout Adaptable con Media Queries: <http://www.arquitectoit.com/>

[WEB35]:, d. (20 de Septiembre de 2008). *Dimuthus*. Recuperado el 2013, de SOA Way of Writing

PHP: <http://www.dimuthu.org/blog/2008/09/20/soa-way-of-writing-php/>

[WEB36]: Berners, T. (07 de Mayo de 2013). *Lipido's Lab*. Recuperado el 2013, de Interfaces de usuario HTML/CSS/Javascript ¿nuevo estándar?:
<http://sing.ei.uvigo.es/~lipido/blog/2013/05/07/interfaces-de-usuario-htmlcssjavascript-nuevo-estandar/>

[WEB 37]:, S. (2013). *LibrosWeb*. Recuperado el 2013, de El patrón MVC:
http://librosweb.es/symfony_1_2/capitulo_2/el_patron_mvc.html

[WEB38]: Pacheco, N. (22 de Abril de 2013). *Symfony*. Recuperado el 2013, de Bases de datos y Doctrine: <http://gitnacho.github.io/symfony-docs-es/book/doctrine.html>

ANEXOS

Anexo 1: Ordenanza de organización y establecimiento del SISMERT de Ibarra.

Anexo 2: Acta de reunión para obtención de requerimientos para el desarrollo del sistema.

Anexo 3: Proyecto Tecnológico.

Anexo 4: Acta de revisión y corrección del Proyecto Tecnológico.

Anexo 5: Actas de reunión de obtención de requisitos y revisión del sistema.

Anexo 6: Manual de Usuario.

Anexo 7: Manual Técnico.

Anexo 8: Artefactos de la Metodología de desarrollo GAD-I.

Anexo 9: Acta de entrega de Proyecto o Módulos del Sistema.