



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

**TRABAJO DE GRADO, PREVIO A LA OBTENCIÓN DEL TÍTULO DE  
INGENIERO EN SISTEMAS COMPUTACIONALES**

**TEMA**

**“SISTEMA DE CONTROL DE PROCESOS DE INSPECCIONES PARA LA  
DIRECCIÓN DE PLANIFICACIÓN DEL GOBIERNO AUTÓNOMO  
DESCENTRALIZADO DE SAN MIGUEL DE IBARRA”**

**Autor:** Rivadeneira Terán Edison Raúl

**Director:** Ing. Irving Reascos.

**Ibarra – Ecuador**

**2014**



## 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 expreso mi voluntad de participar en este proyecto, para lo cual ponemos a disposición la siguiente información:

DATOS DE CONTACTO	
<b>CÉDULA DE IDENTIDAD</b>	100367207-6
<b>APELLIDOS Y NOMBRES</b>	RIVADENEIRA TERÁN EDISON RAÚL
<b>DIRECCIÓN</b>	Ibarra – Barrio La Victoria
<b>EMAIL</b>	<a href="mailto:edison6453@hotmail.com">edison6453@hotmail.com</a>
<b>TELÉFONO FIJO</b>	062615533
<b>TELÉFONO MÓVIL</b>	0994415646

DATOS DE LA OBRA	
<b>TÍTULO</b>	“SISTEMA DE CONTROL DE PROCESOS DE INSPECCIONES PARA LA DIRECCIÓN DE PLANIFICACIÓN DEL GOBIERNO AUTÓNOMO DESCENTRALIZADO DE SAN MIGUEL DE IBARRA”
<b>AUTOR</b>	RIVADENEIRA TERÁN EDISON RAÚL
<b>FECHA</b>	04 DE NOVIEMBRE DEL 2014
<b>PROGRAMA</b>	PREGRADO
<b>TÍTULO POR EL QUE OPTA</b>	INGENIERÍA EN SISTEMAS COMPUTACIONALES
<b>DIRECTOR</b>	ING. IRVING REASCOS

## 2. AUTORIZACIÓN DE USO A FAVOR DE LA UNIVERSIDAD

Yo, RIVADENEIRA TERÁN EDISON RAÚL, con cédula de identidad Nro. 100367207-6, 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 144.



.....

Firma

Nombre: RIVADENEIRA TERÁN EDISON RAÚL.

Cédula: 100367207-6

Ibarra a los 04 días del mes de Noviembre del 2014



## UNIVERSIDAD TÉCNICA DEL NORTE

### CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE GRADO A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

Yo, RIVADENEIRA TERÁN EDISON RAÚL, con cédula de identidad Nro. 100367207-6, 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 DE PROCESOS DE INSPECCIONES PARA LA DIRECCIÓN DE PLANIFICACIÓN DEL GOBIERNO AUTÓNOMO DESCENTRALIZADO DE SAN MIGUEL 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: RIVADENEIRA TERÁN EDISON RAÚL

Cédula: 100367207-6


Ibarrá a los 04 días del mes de Noviembre del 2014

## CONSTANCIAS

El autor manifiesta que la obra objeto de la presente autorización es original y se la desarrolló, sin violar derechos de autor de terceros, por lo tanto la obra es original y que es el titular de los derechos patrimoniales, por lo que asume la responsabilidad sobre el contenido de la misma y saldrá en defensa de la Universidad en caso de reclamación por parte de terceros.

Ibarrá a los 04 días del mes de Noviembre del 2014

**EL AUTOR:**



---

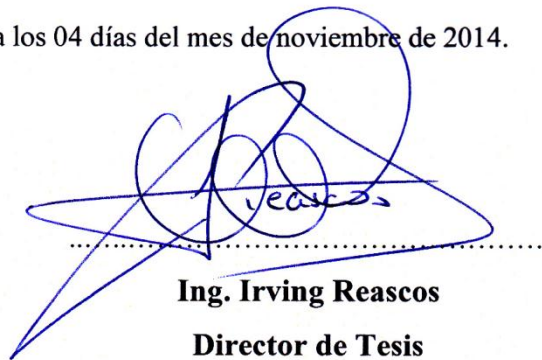
RIVADENEIRA TERÁN EDISON RAÚL

C.I.: 100367207-6

## CERTIFICACIÓN DIRECTOR DE TESIS

En mi calidad de Director del trabajo de grado presentado por el egresado: **Edison Raúl Rivadeneira Terán**, para optar por el título de **INGENIERO EN SISTEMAS COMPUTACIONALES** con el tema del proyecto de titulación: **“SISTEMA DE CONTROL DE PROCESOS DE INSPECCIONES PARA LA DIRECCIÓN DE PLANIFICACIÓN DEL GOBIERNO AUTÓNOMO DESCENTRALIZADO DE SAN MIGUEL DE IBARRA”**. Considero que el presente trabajo reúne los requisitos y méritos suficientes para ser sometido a presentación pública y evaluación por parte del tribunal examinador que se designe.

En la ciudad de Ibarra a los 04 días del mes de noviembre de 2014.



**Ing. Irving Reascos**  
**Director de Tesis**

# CERTIFICACIÓN

Ibarra, 04 de Noviembre de 2014

Señores

UNIVERSIDAD TÉCNICA DEL NORTE

Presente

De mis consideraciones.-

Siendo auspiciantes del proyecto de tesis del Egresado RIVADENEIRA TERÁN EDISON RAÚL con CI: 100367207-6 quien desarrolló su trabajo con el tema **“SISTEMA DE CONTROL DE PROCESOS DE INSPECCIONES PARA LA DIRECCIÓN DE PLANIFICACIÓN DEL GOBIERNO AUTÓNOMO DESCENTRALIZADO DE SAN MIGUEL 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 egresado RIVADENEIRA TERÁN EDISON RAÚL. Una vez que hemos recibido la capacitación y documentación respectiva, nos comprometemos a continuar utilizando el mencionado aplicativo en beneficio de nuestra empresa/institución.

El egresado RIVADENEIRA TERÁN EDISON RAÚL puede hacer uso de este documento para los fines pertinentes en la Universidad Técnica del Norte.

Atentamente,



---

Lcda. Sonia Bossano

RESPONSABLE DE SOFTWARE.

GOBIERNO AUTÓNOMO DESCENTRALIZADO DE IBARRA

## **DEDICATORIA**

*Dedico el presente trabajo de tesis principalmente a mis padres **Martha** y **Raúl** que con su cariño, esfuerzo, trabajo y consejos me han enseñado a valorar la vida y a luchar por nuestras metas propuestas.*

*A mis **hermanos** por estar siempre conmigo en las buenas y en las malas apoyándonos y valorando el gran esfuerzo de nuestros padres por hacer de cada uno de nosotros unas personas profesionales.*

*A mi **sobrina** que ha llenado de alegría a mi hogar y que ha sido otra razón por la cual sonreírle a la vida; con quien he pasado buenos momentos que siempre van estar presentes en mis recuerdos.*

***Edison***



## **AGRADECIMIENTO**

*Agradezco a Dios por darme vida y salud.*

*A mis padres porque mediante su trabajo me han apoyado a estudiar y culminar mi carrera profesional.*

*A mi Universidad y Docentes por ser el lugar donde estudie y con quienes me forme mediante sus conocimientos los cuales me han ayudado a ser un buen profesional.*

*Al Ing. Irving Reascos por ser mi director de tesis y ser la persona que me ayudo a realizar y culminar mi proyecto de grado.*

*Al Municipio de Ibarra en especial a todas las personas que trabajan en la Dirección de Sistemas por haberme apoyado con sus conocimientos en la realización de mi proyecto de tesis y por ser buenas personas con las cuales aprendí y gane mucha experiencia de ellos.*

*A mis amigos de la Universidad con los cuales hemos vivido y pasado buenos momentos, además del apoyo y compañía que me han brindado en el transcurso de mi etapa universitaria.*

***Edison***

# ÍNDICE DE CONTENIDOS

AUTORIZACIÓN DE USO Y PUBLICACIÓN A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE .....	II
CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE GRADO.....	IV
CONSTANCIAS.....	V
CERTIFICACIÓN DIRECTOR DE TESIS .....	VI
CERTIFICACIÓN .....	VII
DEDICATORIA.....	VIII
AGRADECIMIENTO.....	IX
ÍNDICE DE CONTENIDOS.....	X
ÍNDICE DE FIGURAS.....	XIV
ÍNDICE DE TABLAS.....	XVII
RESUMEN .....	XVIII
SUMMARY .....	XIX
INTRODUCCIÓN .....	1
1. ANTECEDENTES.....	1
2. GOBIERNO AUTÓNOMO DESCENTRALIZADO DE SAN MIGUEL DE IBARRA.....	1
❖ Misión .....	1
❖ Visión .....	1
❖ Objetivos Generales .....	2
❖ Objetivos Específicos.....	2
3. ORGANIGRAMA ESTRUCTURAL POR PROCESOS .....	3
4. DIRECCIÓN DE PLANIFICACIÓN Y DESARROLLO .....	5
❖ Misión .....	5
❖ Funciones .....	5
5. PROBLEMA .....	6
❖ Situación actual del problema .....	6
❖ Planteamiento del Problema .....	7
❖ Prospectiva del problema.....	7
6. OBJETIVOS .....	7
❖ Objetivo General.....	7
❖ Objetivos Específicos.....	7
7. JUSTIFICACIÓN.....	8
8. ALCANCE.....	8
CAPÍTULO I.....	10
MARCO TEÓRICO .....	10
1.1. HERRAMIENTAS DE DESARROLLO.....	10
1.1.1 Servidores de Aplicaciones .....	10
❖ HTTP Apache .....	10
❖ Tomcat.....	10
1.1.2 Base de Datos.....	11

❖	<i>Base de Datos PostgreSQL</i> .....	11
❖	<i>PostGIS</i> .....	14
1.1.3	<i>Lenguajes de Programación</i> .....	14
❖	<i>PHP</i> .....	14
❖	<i>JAVASCRIPT</i> .....	15
1.1.4	<i>Frameworks</i> .....	16
❖	<i>Symfony 1.4.6</i> .....	16
❖	<i>Doctrine ORM</i> .....	19
❖	<i>Ext JS 4.0.7</i> .....	21
❖	<i>Sencha Touch 2</i> .....	23
1.1.5	<i>Mapas</i> .....	26
❖	<i>MapServer</i> .....	26
❖	<i>MapFile</i> .....	26
❖	<i>WMS (Web Map Services)</i> .....	26
❖	<i>OpenLayers</i> .....	27
❖	<i>GeoExt2</i> .....	27
1.1.6	<i>Reportes</i> .....	27
❖	<i>JasperReports</i> .....	27
❖	<i>iReport</i> .....	29
❖	<i>PHP-JRU</i> .....	30
❖	<i>PHP/Java Bridge</i> .....	31
<b>1.2.</b>	<b>METODOLOGÍA DE DESARROLLO</b> .....	<b>32</b>
1.2.1	<i>Definición</i> .....	32
1.2.2	<i>Objetivo</i> .....	32
1.2.3	<i>Alcance</i> .....	32
1.2.4	<i>Definiciones y abreviaturas</i> .....	32
1.2.5	<i>Documentos de referencia</i> .....	33
1.2.6	<i>POLÍTICAS</i> .....	33
1.2.7	<i>Diagrama de flujo</i> .....	35
1.2.8	<i>Descripción del procedimiento</i> .....	36
1.2.9	<i>Documentación de la metodología GAD-I</i> .....	38
❖	<i>Formato de Proyecto Tecnológico</i> .....	39
❖	<i>Formato de Acta de Reunión</i> .....	39
❖	<i>Formato de Historia de Usuario</i> .....	39
❖	<i>Formato de Pila de Producto</i> .....	41
❖	<i>Formato de Pila de Iteración</i> .....	41
❖	<i>Formato de Manual de Usuario</i> .....	42
❖	<i>Formato de Manual Técnico</i> .....	42
❖	<i>Formato de Acta de Término de Proyectos</i> .....	42
❖	<i>Formato de Caracterización de Aplicaciones</i> .....	44
<b>CAPÍTULO II</b> .....		<b>47</b>
<b>PROCESO DE INSPECCIONES</b> .....		<b>47</b>
<b>2.1.</b>	<b>OBJETIVO</b> .....	<b>47</b>
<b>2.2.</b>	<b>ALCANCE</b> .....	<b>47</b>
<b>2.3.</b>	<b>ROLES Y RESPONSABILIDADES</b> .....	<b>47</b>
<b>2.4.</b>	<b>ABREVIATURAS Y DEFINICIONES</b> .....	<b>48</b>
<b>2.5.</b>	<b>DIAGRAMA DE FLUJO</b> .....	<b>48</b>

2.6.	DESCRIPCIÓN DEL PROCEDIMIENTO .....	50
2.7.	DOCUMENTOS DE REFERENCIA .....	51
CAPÍTULO III.....		54
DISEÑO Y DESARROLLO DEL SISTEMA DE INSPECCIONES .....		54
3.1.	ROLES.....	54
3.2.	RECEPCIÓN DE SOLICITUD PARA DESARROLLO DE SOFTWARE .....	56
3.3.	ASIGNACIÓN Y PLANIFICACIÓN DEL PROYECTO TECNOLÓGICO.....	56
3.4.	ELABORACIÓN DEL PROYECTO TECNOLÓGICO .....	56
3.5.	OBTENCIÓN DE REQUERIMIENTOS DEL SISTEMA.....	57
3.6.	DESARROLLO DE HISTORIAS DE USUARIO .....	57
3.7.	DISEÑO DEL SISTEMA.....	72
3.7.1	Arquitectura del Sistema Integrado GAD-I.....	72
3.7.2	Core de la vista del Sistema Integrado GAD-I.....	74
3.7.3	Módulos del Sistema de Inspecciones.....	75
3.7.4	Arquitectura Funcional del Sistema de Inspecciones .....	77
3.8.	PILA DE PRODUCTO .....	79
3.9.	PILAS DE ITERACIÓN .....	80
3.9.1	PILA DE ITERACIÓN 1 .....	82
3.9.2	PILA DE ITERACIÓN 2 .....	85
3.9.3	PILA DE ITERACIÓN 3 .....	89
3.9.4	PILA DE ITERACIÓN 4 .....	92
3.9.5	PILA DE ITERACION 5 .....	97
3.9.6	PILA DE ITERACION 6 .....	98
3.10.	DESARROLLO DE HISTORIAS DE USUARIO POR ITERACIÓN.....	100
3.10.1	Iteración 1 (Módulo Horarios de Inspecciones).....	100
❖	Historia N°1: Crear Horarios de Inspección .....	100
❖	Historia N°2: Asignación de arquitectos a parroquias.....	105
3.10.2	Iteración 2 (Módulo Fichas de Inspección).....	108
❖	Historia N°3: Crear Ficha de Inspección.....	108
❖	Historia N°4: Crear categorías y tipos de variables en la ficha de inspección.....	112
3.10.3	Iteración 3 (Módulo Agenda de Inspecciones) .....	120
❖	Historia N°5: Ordenar trámites para generar inspecciones.....	120
❖	Historia N°6: Ver información del trámite.....	124
❖	Historia N°7: Enviar e-mail al ciudadano informando el día de inspección .....	127
3.10.4	Iteración 4 (Módulo Registro de Inspecciones).....	129
❖	Historia N°8: Registro de Inspecciones.....	129
❖	Historia N°9: Recortar fotos de inspección .....	133
❖	Historia N°10: Consultar las inspecciones realizadas.....	135
❖	Historia N°11: Registrar el Personal responsable de la Inspección .....	138
❖	Historia N°12: Ver Informe de Inspección Realizada .....	141
3.10.5	Iteración 5 (Módulo Registro de Inspecciones en el Móvil) .....	142
❖	Historia N°13: Registro de Inspecciones en Dispositivo Móvil .....	142
3.10.6	Iteración 6 (Módulo de Reportes).....	149
❖	Historia N°14: Reportes con Gráficos Estadísticos .....	149
3.11.	ELABORACIÓN DE MANUALES Y ACTA DE TERMINO DE PROYECTO.....	153
CAPÍTULO IV .....		155

<b>CONCLUSIONES Y RECOMENDACIONES .....</b>	<b>155</b>
<b>4.1. CONCLUSIONES.....</b>	<b>155</b>
<b>4.2. RECOMENDACIONES .....</b>	<b>157</b>
<b>BIBLIOGRAFÍA .....</b>	<b>158</b>
<b>ANEXOS .....</b>	<b>161</b>

# ÍNDICE DE FIGURAS

FIGURA 1: ORGANIGRAMA ESTRUCTURAL POR PROCESOS DEL ILUSTRE MUNICIPIO DE IBARRA.....	4
FIGURA 2: ORGANIGRAMA ESTRUCTURAL POR PROCESOS DENTRO DE LA DIRECCIÓN DE PLANIFICACIÓN .....	5
FIGURA 3: ESTRUCTURA GENERAL DE LOS COMPONENTES EN UN SISTEMA POSTGRESQL.....	12
FIGURA 4: EL PATRÓN MVC DE SYMFONY.....	18
FIGURA 5: FUNCIONAMIENTO DE ORM.....	19
FIGURA 6: MODELO, VISTA, CONTROLADOR EXTJS .....	22
FIGURA 7: ESTRUCTURA DE ARCHIVOS DE UNA APLICACIÓN EN EXTJS 4.....	23
FIGURA 8: FUNCIONAMIENTOS TÁCTILES DE UNA APLICACIÓN SENCHA TOUCH.....	24
FIGURA 9: EMPAQUETADO NATIVO DE UNA APLICACIÓN SENCHA TOUCH .....	25
FIGURA 10: FUNCIONAMIENTO DE JASPERREPORTS .....	28
FIGURA 11: FUNCIONAMIENTO DE IREPORT.....	29
FIGURA 12: GRAFICO DE LA ESTRUCTURA DE PHP/JAVA BRIDGE .....	31
FIGURA 13: PROCESO DE DESARROLLO DE SOFTWARE GAD-I .....	35
FIGURA 14: DIAGRAMA DE FLUJO DE PROCESO DE INSPECCIÓN .....	49
FIGURA 15: ARQUITECTURA DEL SISTEMA .....	72
FIGURA 16: ESTRUCTURA CORE GAD-I.....	74
FIGURA 17: MÓDULOS DEL SISTEMAS DE INSPECCIONES .....	76
FIGURA 18: FUNCIONALIDAD DEL SISTEMA DE INSPECCIÓN EN BASE AL USUARIO ADMINISTRADOR .....	77
FIGURA 19: FUNCIONALIDAD DEL SISTEMA DE INSPECCIÓN EN BASE AL USUARIO INSPECTOR.....	78
FIGURA 20: DISEÑO INICIAL DE LA BASE DE DATOS - TABLAS PARA LA CREACIÓN DE HORARIOS .....	101
FIGURA 21: CREACIÓN DE LA APLICACIÓN DE INSPECCIONES Y MÓDULO DE HORARIOS.....	101
FIGURA 22: DIAGRAMA ENTIDAD RELACIÓN DE LAS TABLAS USADAS PARA PERMISOS Y ACCESOS .....	102
FIGURA 23: ESCRITORIO DEL CORE DEL SISTEMA INTEGRADO Y SUS APLICACIONES.....	102
FIGURA 24: INTERFAZ VENTANA DEL SISTEMA CONTROL DE INSPECCIONES Y SUS MENÚS.....	103
FIGURA 25: ESQUEMA DE LA APLICACIÓN DE INSPECCIONES Y MÓDULO DE HORARIOS EN EXTJS .....	103
FIGURA 26: VENTANA Y PANELES PARA LA ASIGNACIÓN DE PARROQUIAS A DÍAS .....	104
FIGURA 27: VENTANA PARA ASIGNAR UNA PARROQUIA A UN DÍA DEL HORARIO.....	104
FIGURA 28: USO Y CONFIGURACIÓN DEL PLUGIN TCPDF .....	105
FIGURA 29: ARCHIVO PDF VISUALIZANDO EL HORARIO DE INSPECCIONES ACTUAL.....	105
FIGURA 30: DIAGRAMA DE LAS TABLAS USADAS PARA ASIGNACIÓN DE PARROQUIAS-ARQUITECTO .....	106
FIGURA 31: CREACIÓN DE MÓDULO ARQUITECTO PARROQUIA EN SYMFONY .....	106
FIGURA 32: CREACIÓN Y ESTRUCTURA DEL MÓDULO ARQUITECTO PARROQUIA EN EXTJS .....	107
FIGURA 33: PARTE DE LA VENTANA Y PANELES DEL MÓDULO ARQUITECTO-PARROQUIA .....	107
FIGURA 34: VENTANA PARA ASIGNAR UN ARQUITECTO A UNA PARROQUIA.....	108
FIGURA 35: TABLA INICIAL DE LAS VARIABLES O DATOS DE UNA FICHA .....	109
FIGURA 36: GENERAR SCHEMA Y MODEL EN SYMFONY .....	109
FIGURA 37: VENTANA INICIAL DEL MÓDULO FICHA DE INSPECCIONES .....	110
FIGURA 38: LISTADO DE PROCESOS MOSTRADOS EN UN COMBOBOX.....	110
FIGURA 39: GRID-PANEL CON EL LISTADO DE VARIABLES .....	111
FIGURA 40: VENTANA PARA CREAR UNA NUEVA VARIABLE EN LA FICHA DE INSPECCIÓN .....	111
FIGURA 41: VENTANA PARA AGREGAR UNA VARIABLE YA EXISTENTE EN LA FICHA DE INSPECCIÓN .....	112
FIGURA 42: DIAGRAMA ENTIDAD RELACIÓN DE LAS TABLAS USADAS PARA LA FICHA DE INSPECCIÓN .....	113
FIGURA 43: CÓDIGO DE UN MODEL EN EXTJS.....	113
FIGURA 44: VENTANA Y GRID-PANEL CON LAS CATEGORÍAS DE LA FICHA DE INSPECCIÓN .....	114
FIGURA 45: VENTANA PARA CREAR UNA NUEVA CATEGORÍA EN LA FICHA DE INSPECCIÓN .....	115
FIGURA 46: VENTANA PARA AGREGAR UNA CATEGORÍA YA EXISTENTE EN LA FICHA DE INSPECCIÓN .....	115

FIGURA 47: VENTANA PARA ELEGIR EL TIPO DE VARIABLE A CREARSE .....	116
FIGURA 48: EJEMPLOS DE UNA VARIABLE DE SELECCIÓN ÚNICA Y SELECCIÓN MÚLTIPLE .....	116
FIGURA 49: VENTANA PARA CREAR UNA NUEVA VARIABLE DE SELECCIÓN ÚNICA .....	117
FIGURA 50: VENTANA PARA VER LAS OPCIONES DE LAS VARIABLES DE SELECCIÓN ÚNICA Y MÚLTIPLE .....	118
FIGURA 51: VENTANA CON LA OPCIÓN DE BUSCAR VARIABLES PARA AGREGAR A LA FICHA .....	118
FIGURA 52: VENTANA PARA VER LAS FICHAS SEGÚN DEPARTAMENTOS Y SUS PROCESOS.....	119
FIGURA 53: ARCHIVO PDF VISUALIZANDO EL DISEÑO DE UNA FICHA DE INSPECCIÓN.....	119
FIGURA 54: DIAGRAMA ENTIDAD RELACIÓN DE LAS TABLAS USADAS PARA LA AGENDA DE INSPECCIONES .....	120
FIGURA 55: VENTANA PARA ORDENAR Y FILTRAR LOS TRÁMITES CON INSPECCIÓN .....	121
FIGURA 56: CONFIGURACIÓN DE LA CONEXIÓN A LA BASE DE DATOS GIS.....	122
FIGURA 57: VENTANA CON LA UBICACIÓN DE LOS PREDIOS DE LOS TRÁMITES A INSPECCIONAR .....	122
FIGURA 58: ACTIVACIÓN DE ORDENACIÓN DE PREDIOS PARA RUTA DE INSPECCIÓN.....	123
FIGURA 59: ARCHIVO PDF VISUALIZANDO LA AGENDA E INSPECCIONES GENERADAS.....	124
FIGURA 60: UBICACIÓN Y ESTRUCTURA DEL CONTROL DATOS TRÁMITE.....	125
FIGURA 61: VENTANA Y TAB-PANEL CON DATOS DEL TRÁMITE .....	125
FIGURA 62: VENTANA Y TAB-PANEL CON DATOS DEL PREDIO .....	126
FIGURA 63: VENTANA DE ARCHIVOS ADJUNTOS DE UN TRÁMITE.....	126
FIGURA 64: VENTANA CON MAPA DE LA UBICACIÓN DEL PREDIO DE UN TRÁMITE .....	127
FIGURA 65: CONFIGURACIÓN DE SFMAILER PARA ENVIÓ DE CORREOS ELECTRÓNICOS .....	128
FIGURA 66: CÓDIGO PARA LA CREACIÓN DE ARCHIVO PARA ENVIÓ DE E-MAIL .....	128
FIGURA 67: CORREO ELECTRÓNICO ENVIADO A UN CIUDADANO .....	128
FIGURA 68: DIAGRAMA ENTIDAD RELACIÓN DE LAS TABLAS PARA REGISTRAR UNA INSPECCIÓN .....	129
FIGURA 69: VENTANA MAESTRO-DETALLE CON LAS AGENDAS E INSPECCIONES GENERADAS .....	130
FIGURA 70: VENTANA PARA REGISTRAR UNA INSPECCIÓN .....	131
FIGURA 71: VENTANA PARA SELECCIONAR Y BUSCAR EL DATO DE VEHÍCULO DE MOVILIZACIÓN .....	131
FIGURA 72: VENTANA PARA REGISTRAR LA FICHA DE INSPECCIÓN .....	132
FIGURA 73: VENTANA PARA ADJUNTAR FOTOS DE LA INSPECCIÓN .....	133
FIGURA 74: VENTANA PARA VER Y EDITAR IMAGEN ADJUNTADA DE LA INSPECCIÓN .....	134
FIGURA 75: LIBRERÍAS JQuery PARA EL USO DE RECORTE DE IMAGEN .....	134
FIGURA 76: USANDO LA FUNCIÓN DE RECORTE DE UNA IMAGEN .....	135
FIGURA 77: VENTANA DE CONSULTAR INSPECCIONES CON OPCIÓN DE VER POR AGENDA.....	136
FIGURA 78: VENTANA CON LISTADO DE INSPECCIONES REGISTRADAS .....	136
FIGURA 79: FILTROS POSIBLES PARA CONSULTAR INSPECCIONES REGISTRADAS .....	137
FIGURA 80: VENTANA PARA VISUALIZAR LAS FOTOS ADJUNTADAS DE LA INSPECCIÓN .....	137
FIGURA 81: VENTANA PARA VISUALIZAR LA FICHA DE INSPECCIÓN CON LOS DATOS REGISTRADA.....	138
FIGURA 82: DIAGRAMA ENTIDAD RELACIÓN DE LAS TABLAS USADAS PARA EL REGISTRO DEL PERSONAL .....	139
FIGURA 83: GRID EN VENTANA DE REGISTRO PARA AGREGAR PERSONAL RESPONSABLE.....	139
FIGURA 84: VENTANA PARA REALIZAR BÚSQUEDA DE EMPLEADOS DEL GAD-I .....	140
FIGURA 85: GRID CON PERSONAL AGREGADO Y COLUMNAS DE ACCIONES .....	140
FIGURA 86: COLUMNA DE ACCIÓN PARA VER EL INFORME DE INSPECCIÓN .....	141
FIGURA 87: VENTANA PARA VISUALIZAR EL INFORME DE INSPECCIÓN.....	141
FIGURA 88: VISTA INICIAL DE LA APLICACIÓN MÓVIL.....	143
FIGURA 89: VISTA DE LISTADO DE SISTEMAS DEL USUARIO .....	143
FIGURA 90: VISTA DE LISTADO DE PARROQUIAS DEL USUARIO .....	144
FIGURA 91: VISTA CON LISTA DE AGENDAS DE UNA PARROQUIA .....	144
FIGURA 92: VISTA CON LISTA DE TRÁMITES DE UNA AGENDA .....	145
FIGURA 93: VISTA CON INFORMACIÓN DEL TRÁMITE Y FOTO PREDIO.....	146
FIGURA 94: VISTA DE UBICACIÓN DE PREDIO EN MAPA.....	146
FIGURA 95: VISTA PARA CARGAR FOTOS DE LA INSPECCIÓN .....	147

FIGURA 96: VISTA PARA REGISTRAR UNA FICHA DE INSPECCIÓN .....	148
FIGURA 97: VISTA PARA GUARDAR LOS DATOS DE INSPECCIÓN .....	148
FIGURA 98: VENTANA DE REPORTE DE INSPECCIONES REALIZADAS .....	150
FIGURA 99: VENTANA DE REPORTE DE INSPECCIONES POR PARROQUIAS .....	150
FIGURA 100: VENTANA DE REPORTE DE INSPECCIONES POR PROCESOS.....	151
FIGURA 101: VENTANA CON REPORTE IREPORT DE INSPECCIONES REALIZADAS.....	151
FIGURA 102: VENTANA CON REPORTE IREPORT DE INSPECCIONES POR PARROQUIAS .....	152
FIGURA 103: VENTANA CON REPORTE IREPORT DE INSPECCIONES POR PROCESOS.....	152



# ÍNDICE DE TABLAS

TABLA 1: DEFINICIONES Y ABREVIATURAS DE LA METODOLOGÍA DE DESARROLLO .....	33
TABLA 2: DESCRIPCIÓN DEL PROCESO DE DESARROLLO DE SOFTWARE .....	38
TABLA 3: FORMATO DE HISTORIA DE USUARIO .....	39
TABLA 4: GUÍA DE LLENADO DE HISTORIA DE USUARIO .....	40
TABLA 5: FORMATO DE PILA DE PRODUCTO .....	41
TABLA 6: GUÍA DE LLENADO DE PILA DE PRODUCTO .....	41
TABLA 7: FORMATO DE PILA DE ITERACIÓN .....	41
TABLA 8: GUÍA DE LLENADO DE PILA DE ITERACIÓN .....	42
TABLA 9: FORMATO DE ACTA DE TERMINO DE PROYECTO .....	43
TABLA 10: GUÍA DE LLENADO DE ACTA DE TERMINO DE PROYECTO .....	44
TABLA 11: FORMATO DE DOCUMENTO DE CARACTERIZACIÓN DE APLICACIONES .....	44
TABLA 12: ROLES Y RESPONSABILIDADES DENTRO DEL PROCESO DE INSPECCIÓN .....	47
TABLA 13: ABREVIATURAS Y DEFINICIONES DENTRO DEL PROCESO DE INSPECCIÓN .....	48
TABLA 14: DESCRIPCIÓN DEL PROCEDIMIENTO DE INSPECCIONES .....	51
TABLA 15: ROLES PARTICIPANTES EN EL DESARROLLO DEL SISTEMA .....	56
TABLA 16: HISTORIA DE USUARIO N°1: CREAR HORARIOS DE INSPECCIÓN .....	58
TABLA 17: HISTORIA DE USUARIO N°2: ASIGNACIÓN DE ARQUITECTOS A PARROQUIAS .....	59
TABLA 18: HISTORIA DE USUARIO N°3: CREAR FICHA DE INSPECCIÓN .....	60
TABLA 19: HISTORIA DE USUARIO N°4: CREAR CATEGORÍAS Y TIPOS DE VARIABLES EN LA FICHA DE INSPECCIÓN .....	61
TABLA 20: HISTORIA DE USUARIO N°5: ORDENAR TRÁMITES PARA GENERAR INSPECCIONES .....	62
TABLA 21: HISTORIA DE USUARIO N°6: VER INFORMACIÓN DEL TRÁMITE .....	63
TABLA 22: HISTORIA DE USUARIO N°7: ENVIAR E-MAIL AL CIUDADANO INFORMANDO EL DÍA DE INSPECCIÓN .....	64
TABLA 23: HISTORIA DE USUARIO N°8: REGISTRO DE INSPECCIONES .....	65
TABLA 24: HISTORIA DE USUARIO N°9: RECORTAR FOTOS DE INSPECCIÓN .....	66
TABLA 25: HISTORIA DE USUARIO N°10: CONSULTAR LAS INSPECCIONES REALIZADAS .....	67
TABLA 26: HISTORIA DE USUARIO N°11: REGISTRAR EL PERSONAL RESPONSABLE DE LA INSPECCIÓN .....	68
TABLA 27: HISTORIA DE USUARIO N°12: VER INFORME DE INSPECCIÓN REALIZADA .....	69
TABLA 28: HISTORIA DE USUARIO N°13: REGISTRO DE INSPECCIONES EN DISPOSITIVO MÓVIL .....	70
TABLA 29: HISTORIA DE USUARIO N°14: REPORTES CON GRÁFICOS ESTADÍSTICOS .....	71
TABLA 30: PILA DE PRODUCTO DEL SISTEMA DE INSPECCIONES .....	79
TABLA 31: PILA DE ITERACIÓN 1 – HISTORIA DE USUARIO N°1 .....	83
TABLA 32: PILA DE ITERACIÓN 1 – HISTORIA DE USUARIO N°2 .....	84
TABLA 33: PILA DE ITERACIÓN 2 – HISTORIA DE USUARIO N°3 .....	86
TABLA 34: PILA DE ITERACIÓN 2 – HISTORIA DE USUARIO N°4 .....	88
TABLA 35: PILA DE ITERACIÓN 3 – HISTORIA DE USUARIO N°5 .....	90
TABLA 36: PILA DE ITERACIÓN 3 – HISTORIA DE USUARIO N°6 .....	91
TABLA 37: PILA DE ITERACIÓN 3 – HISTORIA DE USUARIO N°7 .....	91
TABLA 38: PILA DE ITERACIÓN 4 – HISTORIA DE USUARIO N°8 .....	93
TABLA 39: PILA DE ITERACIÓN 4 – HISTORIA DE USUARIO N°9 .....	94
TABLA 40: PILA DE ITERACIÓN 4 – HISTORIA DE USUARIO N°10 .....	95
TABLA 41: PILA DE ITERACIÓN 4 – HISTORIA DE USUARIO N°11 .....	96
TABLA 42: PILA DE ITERACIÓN 4 – HISTORIA DE USUARIO N°12 .....	96
TABLA 43: PILA DE ITERACIÓN 5 – HISTORIA DE USUARIO N°13 .....	98
TABLA 44: PILA DE ITERACIÓN 6 – HISTORIA DE USUARIO N°14 .....	99

# Resumen

El presente trabajo de tesis está basado en el desarrollo de un sistema para el control de procesos en la Dirección de Planificación del Gobierno Autónomo Descentralizado de San Miguel de Ibarra, específicamente para el registro y control de inspecciones que se realizan a los predios para la posterior aprobación de un trámite, esto con la finalidad de mejorar el proceso y agilizar los trámites de los ciudadanos, además de ayudar en la toma de decisiones a los Arquitectos y funcionarios dentro del departamento.

Para su desarrollo se trabajó con la metodología existente en la Dirección de TICs la cual está basada en metodologías de desarrollo ágil como son XP<sup>1</sup> y SCRUM<sup>2</sup> el cual su principal ventaja es el trabajo directo y continuo con los usuarios del sistema facilitando la obtención de requerimientos y creación del sistema.

Las herramientas de software usadas para la elaboración de este proyecto están basadas en la Arquitectura MVC (Modelo Vista Controlador) donde se trabaja con el framework Symfony en la parte del servidor y con ExtJS4 para las vistas del cliente. Adicionalmente se utilizó el sistema manejador de base de datos PostgreSQL.

El funcionamiento en general del sistema es la de ordenar predios según sus ubicaciones en el mapa GIS y generar una agenda de inspecciones con la cual se asigna según el horario actual de inspección la fecha de visita a cada uno de los predios en la cual principalmente se registran los datos de la ficha y fotos obtenidas que luego servirán de apoyo para futuras consultas y generación de reportes relacionadas con las inspecciones registradas.

En conclusión se determina que el desarrollo y uso del sistema genera un mejor desempeño en las labores del Departamento en cuanto a la realización de los procesos dado que existe un control y registro de las inspecciones realizadas.

## Palabras Clave:

GAD-I, Sistema de Inspecciones, PostgreSQL, Symfony, ExtJS4, Desarrollo Ágil

---

<sup>1</sup> XP: eXtreme Programming

<sup>2</sup> SCRUM: es un marco de trabajo para la gestión y desarrollo de software basada en un proceso iterativo e incremental utilizado comúnmente en entornos basados en el desarrollo ágil de software.

## Summary

This thesis is based on the development of a system for process control in the “Dirección de Planificación del Gobierno Autónomo Descentralizado de San Miguel de Ibarra”, specifically for the registration and control of inspections carried out at the premises for later approval of a process, this in order to improve the process and expedite the process of citizens, and help in decision-making at the Architects and officials within the department.

For its development with the existing methodology worked in the “Dirección de TICs” which is based on agile development methodologies like XP and SCRUM which its main advantage is the direct and continuous work with system users facilitating requirements elicitation and building the system.

The software tools used for the development of this project are based on the MVC (Model View Controller) which works with the Symfony framework on the server side and ExtJS4 for customer views. Additionally handler system was used PostgreSQL database.

The general operation of the system is to sort properties according to their location on the GIS map and generate a schedule of inspections which is allocated according to the current schedule of the date of inspection visit to each of the premises in which mainly recorded data and pictures obtained record who serve as support for future reference and reporting related to the registered inspection.

In conclusion it is determined that the development and use of the system generates better performance in the work of the Department as to the realization of the process because there is a monitoring and recording of inspections.

### **Keywords:**

GAD-I, Inspections System, PostgreSQL, Symfony, ExtJS4, Agile Development

# Introducción



## CONTENIDO DEL CAPÍTULO

- ANTECEDENTES
- JUSTIFICACIÓN
- GADI
- ALCANCE
- PROBLEMA
- OBJETIVOS

# INTRODUCCIÓN

El presente capítulo contiene la información sobre los antecedentes de la institución y Dirección donde se realiza el proceso de inspecciones, además de dar a conocer el problema y los objetivos con los que se plantea la solución mediante el desarrollo del sistema.

## **1. Antecedentes**

Uno de los procesos que se llevan a cabo dentro de la Dirección de Planificación son las inspecciones de las construcciones que se realizan en el territorio comprendido dentro de los límites del Cantón Ibarra, esto se lo ejecuta por medio de los inspectores los cuales están encargados de verificar que el desarrollo de una obra o cualquier construcción se está llevando a cabo en conformidad con los planos, parámetros y especificaciones aprobados según normas establecidas en Ordenanzas, Código de Arquitectura y Urbanismos y demás leyes pertinentes.

## **2. Gobierno Autónomo Descentralizado de San Miguel de Ibarra**

La siguiente información sobre el GAD-I fue tomada del documento Resolución administrativa y Gestión Organizacional por Procesos:

### **❖ Misión**

El Gobierno Autónomo Descentralizado de San Miguel de Ibarra 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**

Ser 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<sup>3</sup> 2010-2014, por una Ibarra hacia el futuro, segura, productiva y social, determina cuatro temas estratégicos en base al análisis FODA<sup>4</sup>, 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 e ibarreñas 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.

---

<sup>3</sup> GAD-I: Gobierno Autónomo Descentralizado de Ibarra

<sup>4</sup> 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.

### **3. Organigrama Estructural por Procesos**

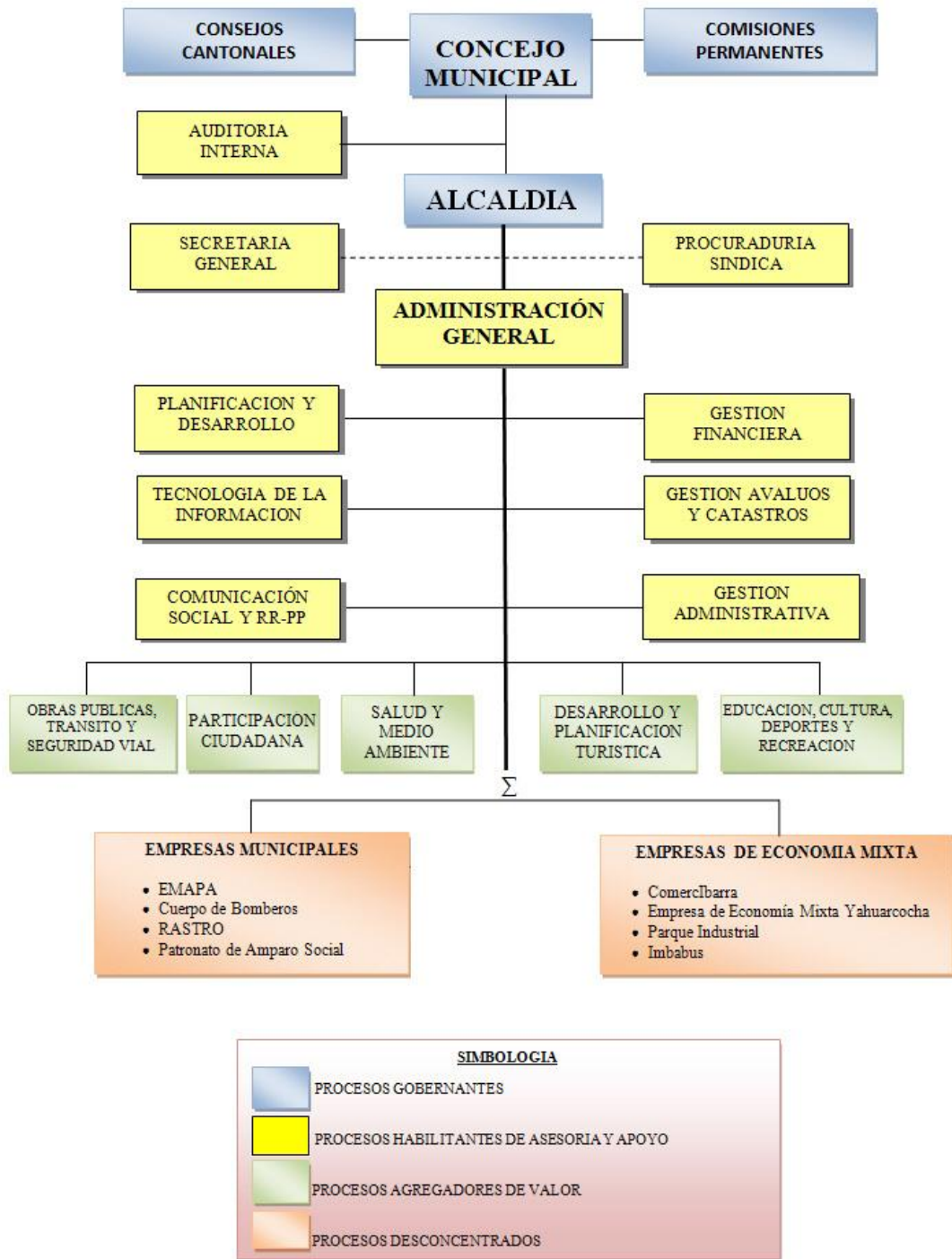
La estructura Administrativa de la Ilustre Municipalidad de Ibarra responde a las necesidades peculiares que debe satisfacer, la importancia de los servicios públicos a prestarse y con la finalidad de ejercer todas y cada una de las funciones que a ésta le competen para el mejor cumplimiento de sus fines, con apego a lo establecido en la COOTAD<sup>5</sup>.

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.

---

<sup>5</sup> COOTAD: Código Orgánico de Ordenamiento Territorial, Autonomía y Descentralización



**Figura 1: Organigrama estructural por procesos del Ilustre Municipio de Ibarra.**  
**Fuente: Resolución administrativa – Gestión Organizacional por Procesos - IMI 2010**



#### 4. Dirección de Planificación y Desarrollo

De igual manera la información sobre la misión y funciones de la Dirección de Planificación fue tomada del documento “Resolución administrativa - Gestión Organizacional por Procesos”.



**Figura 2: Organigrama estructural por procesos dentro de la Dirección de Planificación**

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

##### ❖ Misión

Promover y contribuir al desarrollo territorial del cantón Ibarra, tendiendo a mejorar la calidad de vida de la población y considerando a grupos vulnerables.

##### ❖ Funciones

1. Estudiar los planes de desarrollo cantonal y proponer como acción inmediata y dinamizadora de la administración Municipal, el plan operativo que se encuentra en marcha.
2. Elaborar con todo el equipo de la Dirección de Planificación, y bajo las directrices de la Coordinadora de proyectos, un plan de respuestas inmediatas a las necesidades de la comunidad, según presupuesto y gestiones a realizarse.
3. Diseñar normas y reglamentos para la administración y control del desarrollo físico del cantón, uso de suelo e implantación de nuevas urbanizaciones y lotizaciones.
4. Realizar estudios y diseños de planes viales, equipamiento comunitario implantación industrial y de servicio.
5. Coordinación de acciones con organismos del sector público y privado para lograr la concertación interinstitucional.

6. Emitir informes técnicos sobre temas que tengan que ver con su función.
7. Mantener actualizado el archivo de los planos de edificaciones y urbanizaciones que se han ejecutado.
8. Establecer normas y procedimientos para la elaboración del catastro predial del cantón.
9. Actualizar permanentemente el plan de ordenamiento urbano y rural del cantón
10. Planificación, coordinación, elaboración, ejecución y control de plan de desarrollo territorial del cantón Ibarra.
11. Asesoramiento en la elaboración de planes, programas y proyectos a los clientes internos y externos en el cantón.
12. Preparar un plan de desarrollo Municipal destinado a prever, dirigir, ordenar y estimular su desenvolvimiento en el orden social, económico, físico, y administrativo.
13. Elaboración de propuestas de políticas de desarrollo urbano y rural que posibiliten recuperar y crear nuevos hábitats que eleven el nivel de vida de los ciudadanos.
14. Realizar estudios y proyectos que incorporen en los avalúos y catastros aspectos espaciales, demográficos, económicos, sociales y legales vinculados a la prospectiva y programación territorial y de ordenamiento del cantón.
15. Conservación del patrimonio cultural, tangible e intangible, natural y paisajístico del Cantón Ibarra.
16. Otorgar permisos para la rehabilitación y conservación del patrimonio edificado en el centro y áreas históricas del cantón.
17. Conservación de los sitios históricos arqueológicos del cantón Ibarra.
18. Conservación del patrimonio natural del cantón.
19. Conservación de los bienes muebles e inmuebles religiosos del cantón.
20. Conservación y recuperación de la memoria del patrimonio tangible e intangible del cantón.

## **5. Problema**

### **❖ Situación actual del problema**

La información sobre las construcciones, informes de inspecciones y demás documentos relacionados con este proceso son llevados en su mayoría de una forma manual e impresa dentro de carpetas lo cual dificulta llevar un control y manejo correcto, rápido y eficaz por parte de los inspectores al verificar obras y construcciones que se están llevando a cabo dentro de la Ciudad de Ibarra por lo cual da como resultado la construcción ilegal de edificaciones que no han cumplido con las normas y requisitos para llevarlas a cabo.

Además que el proceso de Inspección es descoordinado dado que se presentan casos en las que una misma construcción es inspeccionada más de una vez por un inspector lo cual tiene como consecuencia la pérdida de tiempo y el no control de otras edificaciones dentro de la ciudad.

#### ❖ **Planteamiento del Problema**

El manejo de la información de Inspecciones y Control de construcciones que se ejecutan en el territorio comprendido dentro de los límites de la Ciudad de Ibarra es llevado manualmente además que este proceso es descoordinado y el resultado de las inspecciones no es digitalizado y se lo lleva en un registro impreso dentro de carpetas por lo que existe el alto riesgo de pérdida de información no recuperable.

#### ❖ **Prospectiva del problema**

En el caso de que no se desarrolle este proyecto seguiría existiendo irregularidad y descoordinación en las inspecciones, por lo cual existirán muchas construcciones ilegales pasadas por alto teniendo como consecuencia la inseguridad en la población al no realizar edificaciones que cumplan con normas para mejorar y asegurar la calidad de vida.

Además que se presentaría la demora en trámites dentro de la Dirección de Planificación donde se requiere la Inspección dentro de algunos procesos que se desarrollan dentro de la misma.

## **6. Objetivos**

#### ❖ **Objetivo General**

Desarrollar e implementar un Sistema de Control de Procesos de Inspecciones para la Dirección de Planificación del Gobierno Autónomo Descentralizado de Ibarra GAD-I y de esta forma facilitar el control, manejo de la información y resultados de las inspecciones dentro de los procesos que se ejecutan en esta dirección.

#### ❖ **Objetivos Específicos**

1. Estudiar los procesos que se llevan a cabo para ejecutar las inspecciones y control de construcciones.
2. Analizar los tipos de inspección que se ejecutan.

3. Contribuir con el análisis y diseño de los procesos.
4. Realizar una reingeniería de procesos para poder automatizarlos, organizarlos y disminuir el tiempo de ejecución.
5. Estudiar y aplicar una metodología de desarrollo rápido para la elaboración del sistema.
6. Capacitación, aprendizaje y uso del framework Symfony como herramienta de desarrollo del sistema.
7. Capacitar al personal involucrado dentro de los proceso de Inspecciones.
8. Informar al ciudadano la fecha de inspección al predio.

## **7. Justificación**

El proceso de inspecciones tiene como beneficiarios principales a los ciudadanos porque ayuda a controlar que las construcciones cumplan con normas de seguridad que aseguren la calidad de vida de las personas; además de verificar que no existan construcciones ilegales las cuales no cumplan con los requisitos existentes en la Dirección de Planificación.

También serán beneficiados los arquitectos que llevaran un control y manejo de las inspecciones realizadas para ayudar a coordinar de forma más eficaz los procesos de la Dirección de Planificación al poder llevar estadísticas y tomar decisiones rápidas, ágiles y oportunas.

## **8. Alcance**

En base a la información que debe ser almacenada, controlada y gestionada por el sistema y analizando los procesos más importantes dentro de la Dirección de Planificación donde se realiza inspección se han determinado los siguientes módulos:

- Horarios de Inspecciones
- Fichas de Inspección
- Agenda de Inspecciones
- Registro de Inspecciones (PC/Móvil)
- Reportes

Además el sistema tiene permisos para dos tipos de usuarios; un rol administrador que podrá crear los horarios y fichas de inspección y el rol arquitecto o inspector que es el encargado de crear inspecciones, registrarlas y visualizar reportes estadísticos.

# Capítulo I: Marco Teórico



## CONTENIDO DEL CAPÍTULO

- HERRAMIENTAS DE DESARROLLO
- METODOLOGÍA DE DESARROLLO

# CAPÍTULO I

## MARCO TEÓRICO

En el presente capítulo se describen las herramientas y metodología usadas para el desarrollo del sistema dando a conocer conceptos de base de datos, lenguajes, frameworks, reportes y metodología de desarrollo.

### 1.1. HERRAMIENTAS DE DESARROLLO

#### 1.1.1 Servidores de Aplicaciones

##### ❖ HTTP Apache

Es un servidor web HTTP <sup>6</sup>de 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 de la Apache Software Foundation. (Wikispaces, 2013)

##### 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<sup>7</sup>. Esta licencia permite hacer lo que quiera con el código fuente siempre y cuando se reconozca su trabajo.

##### ❖ Tomcat

Apache Tomcat es una implementación de software de código abierto de Java Servlet y tecnologías JavaServer Pages que son desarrollados bajo la Java Community Process. (Foundation, 2013)

---

<sup>6</sup> HTTP: HyperText Transfer Protocol.

<sup>7</sup> GPL: Licencia Pública General de GNU (en inglés GNU General Public License).

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 y se lo puede usar en cualquier sistema operativo que disponga de la máquina virtual Java. (Wikispaces A. , 2013)

### **Características Tomcat 7.x:**

- Implementado de Servlet 3.0 JSP 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

### **1.1.2 Base de Datos**

Una base de datos es un “almacén” que nos permite guardar grandes cantidades de información de forma organizada para que luego podamos encontrar y utilizar fácilmente.

Desde el punto de vista informático, la base de datos es un sistema formado por un conjunto de datos almacenados en discos que permiten el acceso directo a ellos y un conjunto de programas que manipulen estos datos. Cada base de datos se compone de una o más tablas que guarda un conjunto de datos. Cada tabla tiene una o más **columnas** y **filas**. Las columnas guardan una parte de la información sobre cada elemento que queramos guardar en la tabla, cada fila de la tabla conforma un registro. (Pérez, 2007)

#### **❖ Base de Datos PostgreSQL**

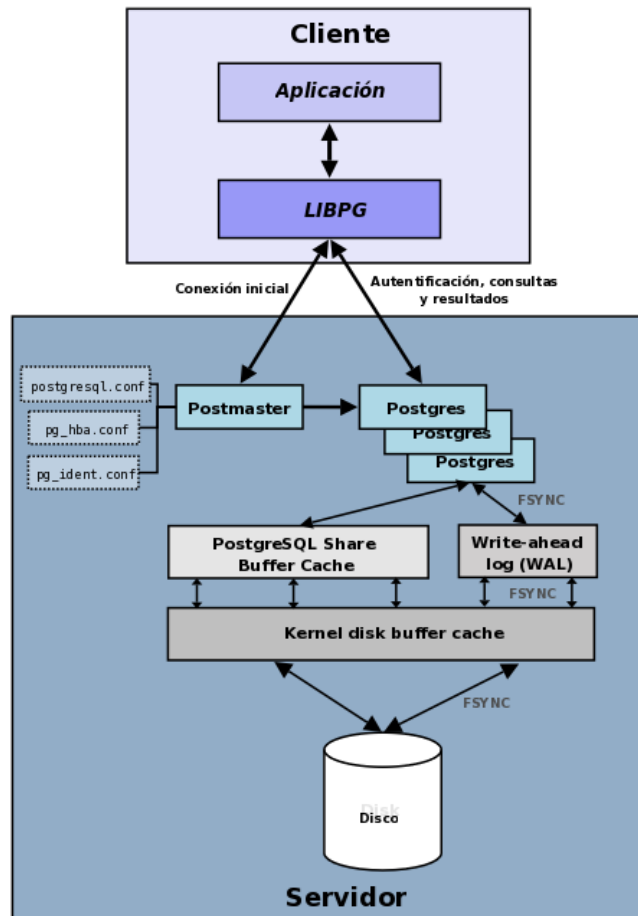
PostgreSQL es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD<sup>8</sup> y con su código fuente disponible libremente. Es el sistema de gestión de bases de datos de código abierto más potente del mercado.

PostgreSQL utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando. (Martínez, 2010)

---

<sup>8</sup> La **licencia BSD** es la licencia de software otorgada principalmente para los sistemas BSD (*Berkeley Software Distribution*)

En el siguiente gráfico se ilustra de manera general los componentes más importantes en un sistema PostgreSQL.



**Figura 3: Estructura General de los componentes en un sistema PostgreSQL**

Fuente: (Martínez, 2010)

- **Aplicación cliente:** Esta es la aplicación cliente que utiliza PostgreSQL como administrador de bases de datos. La conexión puede ocurrir vía TCP/IP<sup>9</sup> o sockets<sup>10</sup> locales.
- **Demonio postmaster:** Este es el proceso principal de PostgreSQL. Es el encargado de escuchar por un puerto/socket por conexiones entrantes de clientes. También es el encargado de crear los procesos hijos que se encargaran de autenticar estas peticiones, gestionar las consultas y mandar los resultados a las aplicaciones clientes.

<sup>9</sup> TCP/IP: Protocolo de Control de Transmisión/Protocolo de Internet (en inglés Transmission Control Protocol/Internet Protocol)

<sup>10</sup> Sockets: Son una interfaz que provee el sistema operativo para usar los servicios de la Capa de Transporte de la red



- **Ficheros de configuración:** Los 3 ficheros principales de configuración utilizados por PostgreSQL son: *postgresql.conf*, *pg\_hba.conf* y *pg\_ident.conf*.
- **Procesos hijos postgres:** Procesos hijos que se encargan de autenticar a los clientes, de gestionar las consultas y mandar los resultados a las aplicaciones clientes.
- **PostgreSQL share buffer cache:** Memoria compartida usada por PostgreSQL para almacenar datos en caché<sup>11</sup>.
- **Write-Ahead Log (WAL):** Componente del sistema encargado de asegurar la integridad de los datos (recuperación de tipo REDO<sup>12</sup>).
- **Kernel disk buffer cache:** Caché de disco del sistema operativo.
- **Disco:** Disco físico donde se almacenan los datos y toda la información necesaria para que PostgreSQL funcione.

### Características:

Algunas de las características más importantes que hacen que PostgreSQL funcione muy bien con grandes cantidades de datos y una alta concurrencia de usuarios accediendo a la vez al sistema son:

#### Generales:

- Es una base de datos 100% ACID<sup>13</sup>
- Integridad referencial
- Replicación asincrónica/sincrónica / Streaming replication - Hot Standby
- Copias de seguridad en caliente (Online/hot backups)
- Múltiples métodos de autenticación
- Acceso encriptado vía SSL<sup>14</sup>

#### Programación y Desarrollo:

- Funciones y procedimientos almacenados en numerosos lenguajes de programación, como PL/pgSQL (similar al PL/SQL de oracle), PL/Perl, PL/Python y PL/Tcl.

---

<sup>11</sup> Caché: es un conjunto de datos duplicados de otros originales, dado que los datos originales son costosos de acceder, normalmente en tiempo, respecto a la copia en la caché

<sup>12</sup> REDO: Archivos de Recuperación

<sup>13</sup> ACID: Atomicity, Consistency, Isolation and Durability (Atomicidad, Consistencia, Aislamiento y Durabilidad).

<sup>14</sup> SSL: Capa de Conexión Segura (del inglés Secure Socket Layer)

- APIs para programar en C/C++, Java, .Net, Perl, Python, Ruby, Tcl, ODBC, PHP, Lisp, Scheme, Qt y muchos otros.

## SQL:

- Llaves primarias y foráneas, check, unique y not null constraints
- Columnas auto-incrementales
- Índices compuestos, únicos, parciales y funcionales en cualquiera de los métodos de almacenamiento disponibles, B-tree, R-tree, hash ó GiST<sup>15</sup>
- Sub-selects y consultas recursivas
- Funciones “Windows”
- Vistas (views)
- Disparadores (triggers) comunes, por columna, condicionales.

### ❖ PostGIS

Es una extensión al sistema de base de datos objeto-relacional PostgreSQL. Permite el uso de objetos GIS<sup>16</sup>, además incluye soporte para índices GiST basados en R-Tree, y funciones básicas para el análisis de objetos GIS. Esta creado por Refrations Research Inc, como un proyecto de investigación de tecnologías de bases de datos espaciales y publicado bajo licencia GNU<sup>17</sup>.

Con PostGIS podemos usar todos los objetos que aparecen en la especificación OpenGIS<sup>18</sup> como puntos, líneas, polígonos, multilíneas, multipuntos, y colecciones geométricas para la creación de mapas. (Ramsey & Martín, 2012)

### 1.1.3 Lenguajes de Programación

#### ❖ PHP

PHP es uno de los lenguajes del lado servidor más extendidos en la Web. Permite embeber fragmentos de código dentro de la página HTML y realizar determinadas acciones de una forma fácil y eficaz sin tener que generar programas en un lenguaje distinto. Ofrece varias funciones para la explotación de bases de datos de una manera sencilla. (Sabana Mendoza, 2010)

---

<sup>15</sup> GiST: Índices para datos espaciales.

<sup>16</sup> GIS: Sistema de Información Geográfica (Geographic information systems).

<sup>17</sup> GNU: (Gnus Not UNIX) Proyecto patrocinado por Free Software Foundation.

<sup>18</sup> OpenGIS: es un estándar internacional orientado a Sistemas de Información Geográfica.

### **Características Generales:**

- Lenguaje exclusivo para el diseño de aplicaciones Web.
- Simplicidad en su sintaxis porque está basada en C y es fácil de aprender.
- Script de código abierto para servidores.
- Utiliza una amplia gama de servidores Web, como Apache, IIS<sup>19</sup>, etc.
- Conexión a diversas bases de datos, como PostgreSQL, MySQL, Ingres, Oracle, Sybase, Firebird, etc.
- Posee protocolos HTTP, POP3<sup>20</sup>, SNMP<sup>21</sup>, LDAP<sup>22</sup> e IMAP<sup>23</sup>
- No requiere grandes cantidades de hardware

### **Características de PHP 5:**

- Presenta el SQLite, sistema de gestión de base de datos con capacidad de 2TB.
- Manejo de excepciones.
- Posee un analizador de XML, llamado SimpleXML.
- Su nuevo modelo orientado a objetos presenta una sintaxis parecida a Java.
- PHP<sup>24</sup> 5 hace uso de los manejadores de objetos (Object handles), que son punteros que apuntan hacia los espacios en memoria donde residen los objetos.

### **❖ JAVASCRIPT**

El lenguaje de programación JavaScript se utiliza en las páginas Web para cubrir las carencias que deja el HTML a secas. Incorporando este lenguaje a nuestras páginas logramos incrementar la funcionalidad de las mismas y la interacción con el usuario, lo que traduce en unas páginas más dinámicas. Algunos ejemplos de su utilidad podrían ser:

- Validar los datos introducidos en un formulario.
- Detectar el navegador que está utilizando el usuario.
- Almacenar información del usuario para que no tenga que volver a ser introducida en la siguiente vista (Cookies).

---

<sup>19</sup> IIS (Internet Information Services): es un servidor web y un conjunto de servicios para el S.O Windows.

<sup>20</sup> POP3: Protocolo de Oficina Postal (del inglés Post Office Protocol).

<sup>21</sup> SNMP: Protocolo Simple de Administración de Red (del inglés Simple Network Management Protocol).

<sup>22</sup> LDAP: Protocolo Ligero de Acceso a Directorios (del inglés Lightweight Directory Access Protocol).

<sup>23</sup> IMAP: Internet Message Access.

<sup>24</sup> PHP: es un acrónimo recursivo que significa PHP Hypertext Pre-processor (inicialmente PHP Tools, o, Personal Home Page Tools).

El código que escribamos en nuestras páginas será interpretado directamente por el navegador que estemos usando, de modo que el servidor Web no interpreta ni ejecuta nuestro código. Esto lo convierte en un lenguaje del lado cliente, al contrario de lo que ocurre con lenguajes como PHP, donde todo se ejecuta en el lado del servidor. (Parra, 2012)

### **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.
- 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. (Uzuay, 2013)

#### **1.1.4 Frameworks**

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. (Paszniuk, 2013)

##### **❖ Symfony 1.4.6**

Es un framework PHP que facilita y optimiza el desarrollo de las aplicaciones web. Symfony se encarga de todos los aspectos comunes y aburridos de las aplicaciones web, dejando que el programador se dedique a aportar valor desarrollando las características únicas de cada proyecto.

Symfony separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web. (Symfony Es, 2013)

### **Características:**

- 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<sup>25</sup> 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.

### **El patrón MVC**

Symfony está basado en un patrón clásico del diseño web conocido como arquitectura MVC, que está formado por tres niveles:

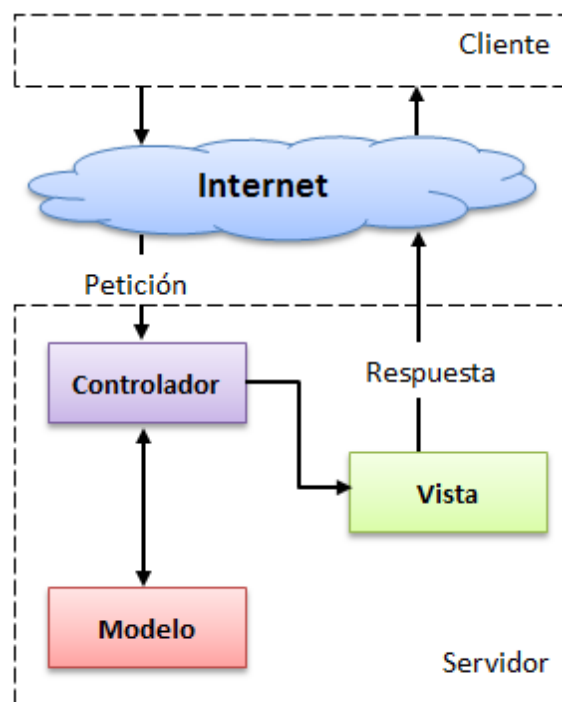
---

<sup>25</sup> SGBD: Sistema de Gestión de Bases de Datos

- El Modelo representa la información con la que trabaja la aplicación, es decir, su lógica de negocio.
- La Vista transforma el modelo en una página web que permite al usuario interactuar con ella.
- El Controlador se encarga de procesar las interacciones del usuario y realiza los cambios apropiados en el modelo o en la vista.

La **Figura 4** ilustra el funcionamiento del patrón MVC.

La arquitectura MVC separa la lógica de negocio (el modelo) y la presentación (la vista) por lo que se consigue un mantenimiento más sencillo de las aplicaciones. Si por ejemplo una misma aplicación debe ejecutarse tanto en un navegador estándar como un navegador de un dispositivo móvil, solamente es necesario crear una vista nueva para cada dispositivo; manteniendo el controlador y el modelo original. El controlador se encarga de aislar al modelo y a la vista de los detalles del protocolo utilizado para las peticiones (HTTP, consola de comandos, email, etc.). El modelo se encarga de la abstracción de la lógica relacionada con los datos, haciendo que la vista y las acciones sean independientes de, por ejemplo, el tipo de gestor de bases de datos utilizado por la aplicación. (Potencier & Zaninotto, 2010)



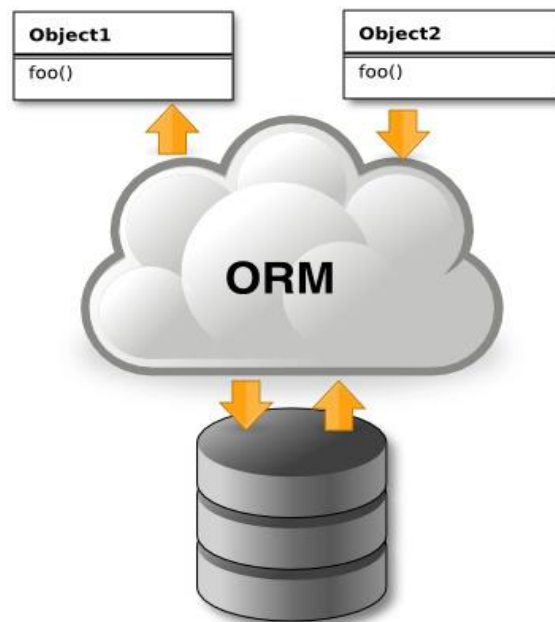
**Figura 4: El patrón MVC de Symfony**  
**Fuente: (Potencier & Zaninotto, 2010)**

## ❖ Doctrine ORM

Para entender claramente el concepto lo dividimos en 2 partes, primero que es un ORM y luego la definición de Doctrine.

### ¿Qué es un ORM?

Un ORM<sup>26</sup> o (Object Relation Mapper) es una técnica de programación que nos permite convertir datos entre el sistema de tipos utilizado en un lenguaje de programación orientado a objetos y el utilizado en una base de datos relacional, es decir, las tablas de nuestra base de datos pasan a ser clases y los registros objetos que podemos manejar con facilidad; Ver Figura 5 (Perez, 2013).



**Figura 5: Funcionamiento de ORM**

**Fuente: (Perez, 2013)**

Utilizar un ORM tiene una serie de ventajas que nos facilitan enormemente tareas comunes y de mantenimiento:

- a) **Reutilización:** La principal ventaja que aporta un ORM es la reutilización permitiendo llamar a los métodos de un objeto de datos desde distintas partes de la aplicación e incluso desde diferentes aplicaciones.

---

<sup>26</sup> ORM: Mapeador de objetos-relacional

- b) **Encapsulación:** La capa ORM encapsula la lógica de los datos pudiendo hacer cambios que afectan a toda la aplicación únicamente modificando una función.
- c) **Portabilidad:** Utilizar una capa de abstracción nos permite cambiar en mitad de un proyecto de una base de datos MySQL a una Oracle sin ningún tipo de complicación. Esto es debido a que no utilizamos una sintaxis MySQL, Oracle o SQLite para acceder a nuestro modelo, sino una sintaxis propia del ORM utilizado que es capaz de traducir a diferentes tipos de bases de datos.
- d) **Seguridad:** Los ORM suelen implementar mecanismos de seguridad que protegen nuestra aplicación de los ataques más comunes como SQL Injections.
- e) **Mantenimiento del código:** Gracias a la correcta ordenación de la capa de datos, modificar y mantener nuestro código es una tarea sencilla.

### ¿Qué es Doctrine?

Doctrine es un mapeador objeto relacional (ORM) para PHP 5.2.3 y posterior que proporciona persistencia transparente de objetos PHP. Se sitúa en la parte superior de una poderosa capa de abstracción de base de datos (DBAL por DataBase Abstraction Layer). La principal tarea de los es la traducción entre objetos (PHP) y las filas relacionales de la base de datos.

Una de las características clave de Doctrine es la opción de escribir las consultas de base de datos en un dialecto SQL propio orientado a objetos llamado Lenguaje de Consulta Doctrine (DQL por Doctrine Query Language), inspirado en Hibernate HQL. Además DQL difiere ligeramente de SQL en que abstrae considerablemente la asignación entre las filas de la base de datos y objetos, permitiendo a los desarrolladores escribir poderosas consultas de una manera sencilla y flexible. (Pacheco, 2013)

### Características:

- Agrega una capa de abstracción que se sitúa justo encima de un SGBD.
- 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.
- No es necesario generar o mantener complejos esquemas XML<sup>27</sup> de base de datos como en otros frameworks.

---

<sup>27</sup> XML: eXtensible Markup Language ('lenguaje de marcas extensible')



- 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).

#### ❖ Ext JS 4.0.7

Es una biblioteca de JavaScript para el desarrollo de aplicaciones web interactivas usando tecnologías como AJAX<sup>28</sup>, DHTML<sup>29</sup> y DOM<sup>30</sup>. Fue desarrollada por Sencha.

La versión actual en la que se está desarrollando es la EXTJS 4 que incluye una refactorización de todo el framework entre lo que cabe destacar una nueva estructura de clases y carga dinámica de objetos, paquete de datos, nuevos gráficos y temas. (Villa, 2013)

#### **Características:**

- 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.

---

<sup>28</sup> AJAX: acrónimo de Asynchronous JavaScript And XML, es una técnica de desarrollo web para crear aplicaciones interactivas o RIA.

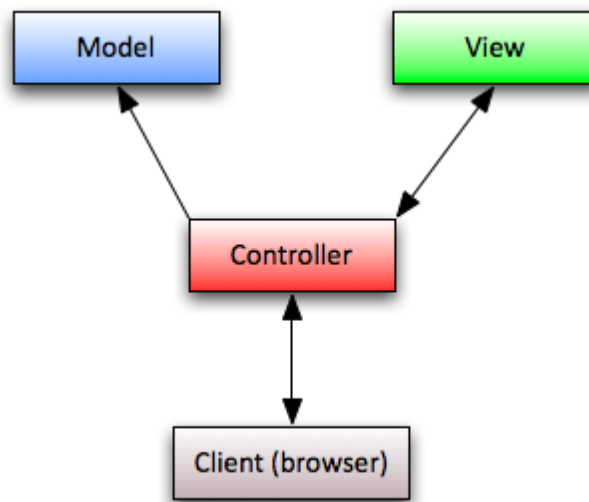
<sup>29</sup> DHTML: Dynamic HTML

<sup>30</sup> DOM: *Document Object Model* o DOM ('Modelo de Objetos del Documento' o 'Modelo en Objetos para la Representación de Documentos')

- ✓ 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.

### Modelo-Vista-Controlador

ExtJS4 nos permite implementar de una forma sencilla el patrón MVC para organizar nuestro código y separarlo en capas delegando responsabilidades específicas en cada una de ellas.



**Figura 6: Modelo, Vista, Controlador ExtJS**

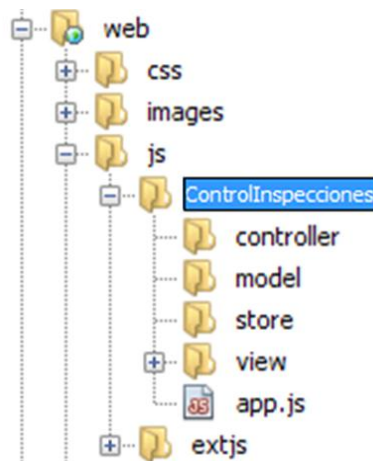
**Fuente: Propia**

- Un **Modelo** es una colección de campos y sus datos (por ejemplo, un modelo de usuario con nombre de usuario y las contraseñas. Los modelos se utilizan normalmente con los Stores para presentar los datos en las grillas y otros componentes.

- La **Vista** es cualquier tipo de componente como los grids, trees, panels, windows son todos ellos vistas.
- Los **Controladores** son lugares especiales para poner todo el código que hace que su aplicación funcione, ya sea el renderizado de vistas, creación de instancias de Modelos, eventos de los componentes o cualquier otra lógica de aplicación.

### Estructura de archivos

Las aplicaciones ExtJS4 siguen una estructura de directorio unificada que es el mismo para cada aplicación. En el diseño MVC, todas las clases se colocan en la carpeta app, que contiene a su vez sub-carpetas para generar el espacio de nombres de los modelos, las vistas, controladores y almacenes (stores).



**Figura 7: Estructura de Archivos de una Aplicación en ExtJS 4**

**Fuente: Propia**

### ❖ Sencha Touch 2

Sencha Touch es un framework MVC JavaScript que utiliza los estándares web de HTML5 y CSS3 para crear aplicaciones web móviles para dispositivos con pantalla táctil con la apariencia de ser nativos de los sistemas IOS<sup>31</sup>, Android<sup>32</sup> y Blackberry. Sencha Touch también permite a los desarrolladores crear aplicaciones para plataformas móviles que cuentan con navegadores que implementan las normas más recientes, como el motor de navegador WebKit. (Kosmaczewski, 2013)

<sup>31</sup> IOS: iPhone OS es un sistema operativo móvil de la empresa Apple Inc

<sup>32</sup> Android: es un sistema operativo basado en Linux, diseñado principalmente para móviles con pantalla táctil

Este framework ha sido creado por la empresa Sencha (los creadores de 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. (Meléndez, 2013)

### 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:** Está 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.

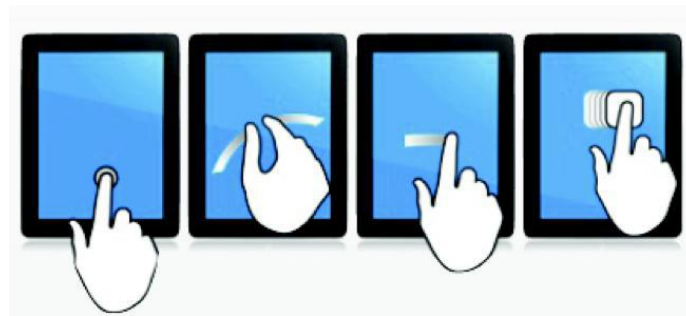


Figura 8: Funcionamientos táctiles de una aplicación Sencha Touch

Fuente: (Meléndez, 2013)

### 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.

### **Navegadores y dispositivos soportados**

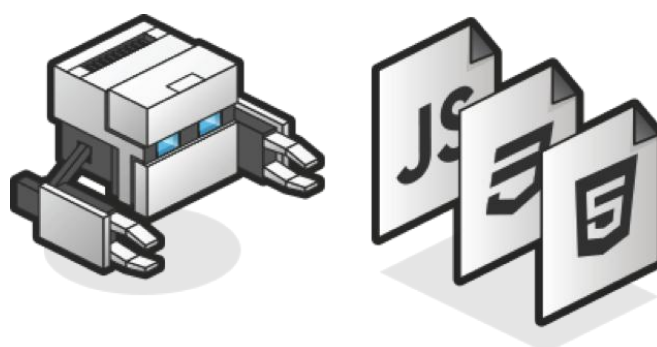
Sencha Touch, actualmente soporta las siguientes plataformas móviles:

- IOS desde la versión 3
- Android desde la versión 2.3
- BlackBerry OS desde la versión 6(solo para dispositivos con navegadores WebKit-Powered)

Además se tiene pensando tener próximamente soporte para Windows Phone 8, así como la versión preliminar de un tema para BlackBerry 10. Sencha Touch es un framework 100% orientado para navegadores, por lo tanto es independiente del servidor; puede implementar sus aplicaciones usando cualquier tecnología del lado del servidor, tales como PHP, Java, Ruby on Rails, .NET o cualquier otra tecnologia de su eleccion. (Kosmaczewski, 2013)

### **Convertir la aplicación web en nativa**

Sencha Touch 2 tiene su propio Empaquetado Nativo para convertir nuestra aplicación a IOS o Android, pero también se puede convertir en una aplicación nativa a través PhoneGap<sup>33</sup> Build que utiliza un mecanismo que la empotra en una Shell nativa y se puede acceder a elementos nativos del móvil como la cámara, el GPS, batería, etc. (Meléndez, 2013)



**Figura 9: Empaquetado nativo de una aplicación Sencha Touch**

**Fuente: (Meléndez, 2013)**

---

<sup>33</sup> PhoneGap: es un framework para el desarrollo de aplicaciones móviles que permiten convertir una aplicación web móvil en aplicación nativa

### 1.1.5 Mapas

#### ❖ MapServer

MapServer es un motor de renderización de datos geográficos Open Source escrito en C. Más allá de la navegación por datos GIS, MapServer permite crear “mapas de imágenes geográficas”, es decir, mapas de contenidos para los usuarios. Por ejemplo, en Minnesota DNR Recreation Compass se proporciona a los usuarios más de 10.000 páginas Web, informes y mapas, a través de una única aplicación. La misma aplicación hace de “motor de mapas”, proporcionando los contenidos espaciales a otras partes del sitio Web que las necesitan.

MapServer fue desarrollado inicialmente por el proyecto *ForNet* de la Universidad de Minnesota (UMN) en cooperación con la NASA, y el Departamento de Recursos Naturales de Minnesota, el MNDR (Minnesota Department of Natural Resources). Más tarde, se alojó en el proyecto *TerraSIP project*, un proyecto financiado por la NASA, entre la UMN y un consorcio de actores con intereses en gestión del territorio.

MapServer es ahora un proyecto de OSGeo<sup>34</sup>, y lo mantiene un grupo creciente de desarrolladores distribuidos por todo el mundo. Un diverso grupo de organizaciones dan soporte al proyecto, proporcionando mejoras y mantenimiento, y se administra dentro de OSGeo por el *MapServer Project Steering Committee*, formado por los desarrolladores y otros contribuidores. (MapServer, 2013)

#### ❖ MapFile

El archivo principal de configuración de MapServer es un archivo de texto, con extensión “.map”, en el que se incluye una serie de parámetros que definen las capas disponibles en el servicio, el estilo con que se representarán, su simbología, formato se generará la imagen, el sistema de referencia, etc. (Manso & Ballari, 2013)

#### ❖ WMS (Web Map Services)

El servicio Web Map Services interactúa con los clientes a través de peticiones HTTP, produce mapas referenciados espacialmente de forma dinámica, a partir de información geográfica. Estos mapas son enviados a los clientes en forma de imagen digital.

---

<sup>34</sup> OSGeo: La Open Source Geospatial Foundation (OSGeo), es una organización cuya misión es dar soporte y promover el desarrollo colaborativo de tecnologías geoespaciales y datos abiertos.

Los mapas producidos por WMS se generan normalmente en un formato de imagen como PNG, GIF o JPEG, se utilizan los formatos transparentes para sobreponer capas de diferentes servidores, también se puede generar archivos TIFF que necesitan software adicional para su visualización y ocasionalmente se generan como gráficos vectoriales en formato SVG (Scalable Vector Graphics) o WebCGM (Web Computer Graphics Metafile). (Moncayo, 2013)

#### ❖ **OpenLayers**

OpenLayers es una biblioteca de Javascript de uso libre para acceder, manipular y mostrar mapas en páginas web. Proporciona un interfaz de programación de aplicaciones, API (del inglés Application Programming Interface) que permite la creación de clientes web para acceder y manipular información geográfica proveniente de muy variadas fuentes, y que permite incorporar mapas de forma dinámica dotados con controles diversos como zoom, medida de distancias y muchas otras herramientas. (Higuera, 2013)

#### ❖ **GeoExt2**

GeoExt es de código abierto y permite la creación de aplicaciones GIS de escritorio, como a través de la web. Se trata de un framework de JavaScript que combina la funcionalidad GIS de OpenLayers con la interfaz de usuario de la biblioteca ExtJS proporcionada por Sencha.

La versión 2 de GeoExt es la sucesora de la serie GeoExt 1.x y está construida sobre las nuevas entregas oficiales de las bibliotecas base: OpenLayers 2.13.1 y ExtJS 4.2.1. GeoExt2 está basado en la versión más reciente de Ext JS 4. Esto significa que GeoExt se puede utilizar como cualquier otro componente de Ext4, y las aplicaciones que hacen uso de GeoExt2 también se benefician de las mejoras de Ext JS como la cartografía, además de una API compatible con Sencha Touch y una herramienta de construcción de un solo archivo sofisticado. (GeoExt, 2013)

### **1.1.6 Reportes**

#### ❖ **JasperReports**

JasperReports es una poderosa herramienta de reporte Open Source que tiene la habilidad de presentar reportes en pantalla, impresora o en formato PDF, HTML, XLS, CSV y XML. Es completamente escrita en Java y puede usarse en una infinidad de aplicaciones Java habilitadas para generación dinámica de contenido. Su propósito principal es ayudar a crear reportes escritos,

listos para imprimir de una manera simple y flexible. JasperReports organiza datos recuperados de una base de datos relacional a través de JDBC<sup>35</sup> según el diseño del reporte definido en un archivo XML. Para llenar un informe, dicho diseño debe primero ser compilado.

A través de la compilación, el diseño del reporte se carga en un objeto serializado y es guardado en el disco (net.sf.jasperreports.engine.JasperReport). Este objeto serializado se usa cuando la aplicación requiere llenar el diseño del reporte especificado con los datos obtenidos. De hecho, la compilación del diseño implica la compilación de todas las expresiones Java definidas en el archivo XML que representa el diseño del reporte. Varias comprobaciones son hechas en tiempo de compilación para verificar la consistencia del diseño. El resultado es un diseño final compilado que se usará para generar los reportes cargando los diferentes datos obtenidos. (Torres Arcos, 2010)

### Funcionamiento de JasperReports

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. (MyGnet, 2013)

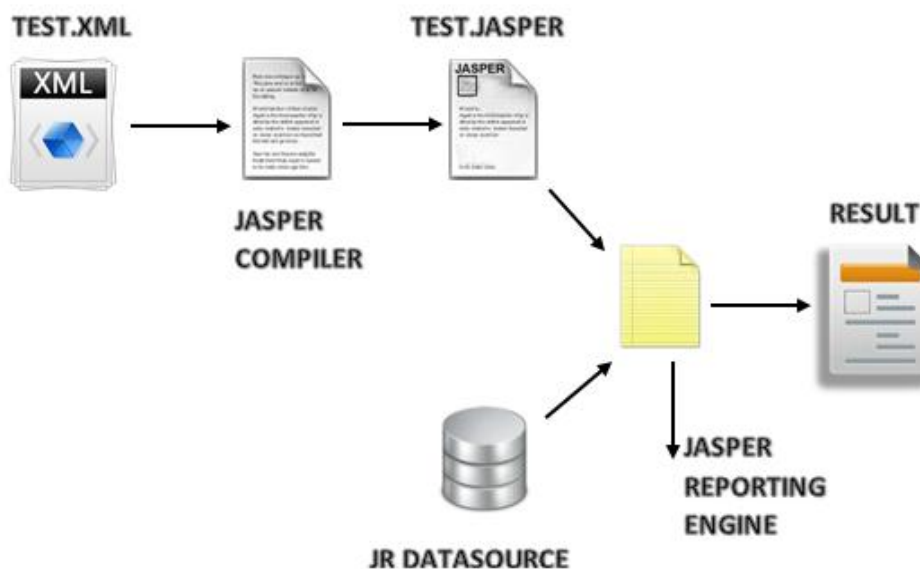


Figura 10: Funcionamiento de JasperReports

Fuente: (MyGnet, 2013)

<sup>35</sup> JDBC: Java Database Connectivity, es una API que permite ejecutar operaciones sobre BDD desde Java.



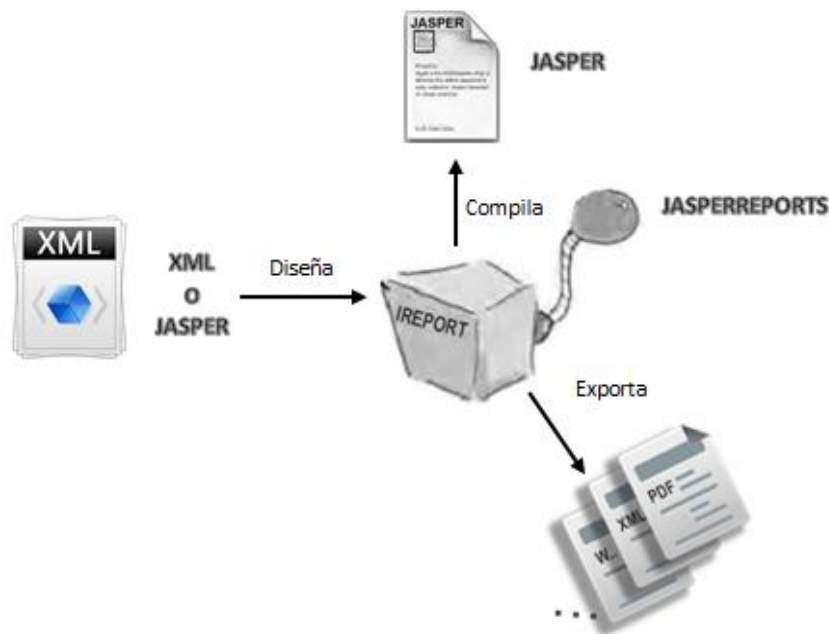
Este archivo fuente XML debe ser compilado para obtener un reporte real. La versión compilada del fuente es nombrada "archivo *Jasper*" (este termina con *.jasper*). Un Archivo *Jasper* es el compilado de un código fuente.

### ❖ 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 puede ayudar a la gente que no conoce la sintaxis XML a generar reportes de JasperReports. (MyGnet, 2013)

### Funcionamiento de iReport

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. (MyGnet, 2013)



**Figura 11: Funcionamiento de iReport**

Fuente: (MyGnet, 2013)

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, etc.

### **Características principales de iReport**

La lista siguiente describe algunas de las características importantes del iReport:

- 100% escrito en Java
- Open Source gratuito
- Maneja el 98% de las etiquetas de JasperReports
- Permite diseñar con sus propias herramientas: rectángulos, líneas, elipses, campos, textfields, cartas, sub-reportes, etc.
- Ayuda para el cambio de lengua no latina (ruso, chino, coreano, etc.)
- Browser de la estructura del documento
- Recopilador y exportador integrados
- Soporte de JDBC
- Soporta Java Beans como orígenes de datos (se debe implementar la interface JRDataSource)
- Incluye Wizard's (asistentes) para crear informes y reportes automáticamente
- Asistentes para manejo de sub-reportes
- Asistente para plantillas
- Fácil Instalación (EcuRed, 2013)

#### **❖ PHP-JRU**

PHP-JRU (PHP Jasper Report Utils) es una librería pensada para generar reportes diseñados bajo la herramienta iReport desde una simple aplicación escrita en PHP.

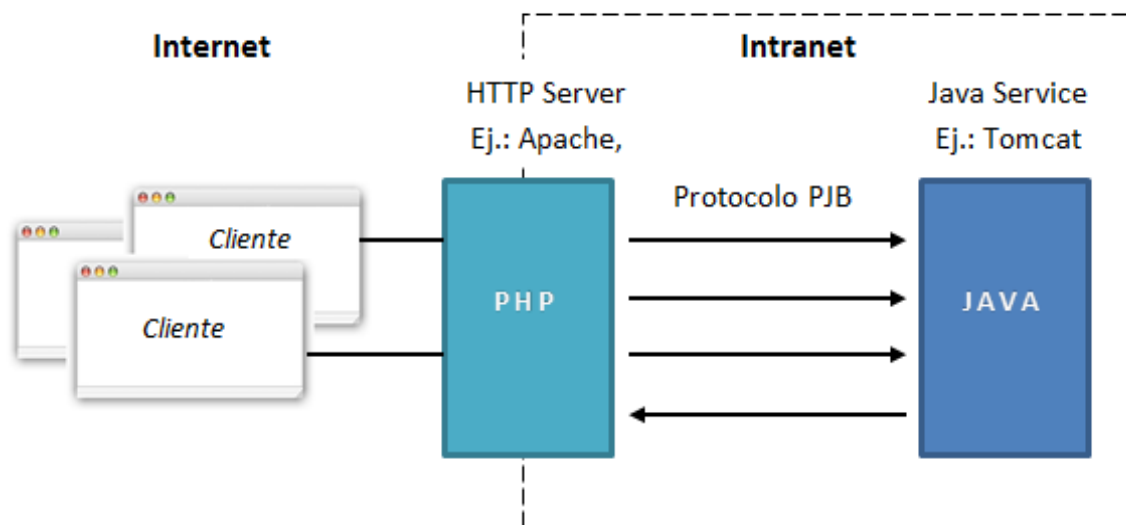
Para ello esta librería se comunica con la JVM<sup>36</sup> a través de PHP Java/Bridge, y de esta manera poder cargar las librerías provista por JasperReport he indicarle las tareas que deben realizarse para generar los reportes tal cual como si se estuviera haciendo desde una aplicación escrita en java. El proceso es el de diseñar un reporte con la herramienta iReport para crear el archivo .jrxml o el.jasper y haciendo uso de PHP-JRU llamar a las funciones necesarias de Java para generar el reporte. (Bruno R. , 2013)

---

<sup>36</sup> JVM: Máquina Virtual de Java(Java Virtual Machine)

## ❖ PHP/Java Bridge

PHP / Java Bridge es un puente basado en protocolo de red XML, que puede ser usado para conectar el motor de script nativo de PHP con Java o la máquina virtual. PHP / Java Bridge permite a los desarrolladores el acceso de clases Java en PHP. También permite el acceso a los scripts PHP dentro de clases Java. Debido a esta interoperabilidad, los desarrolladores pueden desarrollar páginas web interactivas utilizando PHP, pero delegar funciones en clases Java, lo que permite usar cada tecnología por sus puntos fuertes y eludir sus puntos débiles. (PHP-JRU, 2013)



**Figura 12: Grafico de la Estructura de PHP/Java Bridge**

**Fuente: (PHP-JavaBridge, 2013)**

Es hasta 50 veces más rápido que usar RPC (Remote Procedure Call, requiere menos recursos en el lado servidor web. Es más rápido y más fiable que la comunicación directa a través de la interfaz nativa de Java, y no requiere de componentes adicionales para invocar procedimientos de Java. (PHP-JavaBridge, 2013)

## 1.2. METODOLOGÍA DE DESARROLLO

### 1.2.1 Definición

Es una metodología de rápido desarrollo ya que está constituida por la metodología XP y SCRUM, ya que se ha tomado 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. (Rea, 2012)

### 1.2.2 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. (Rea, 2012)

### 1.2.3 Alcance

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

### 1.2.4 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	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:

N°	TÉRMINO	DEFINICIÓN
3	<b>MÉTODO X.Y.Z</b>	<ul style="list-style-type: none"> <li>• X.Y.Z: Software Nuevo, Cambios muy drásticos en el desarrollo, Reescritura o la incompatibilidad con versiones anteriores.</li> <li>• 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.</li> <li>• X.Y.Z: Cuando se hacen correcciones, añadir o eliminar comentarios, renombrar variables, pero no se ha añadido ni eliminado nada relevante.</li> </ul> <p><b>NOTA:</b> Siempre que se hace un cambio en una cifra de la izquierda, la cifra de la derecha se reinicia en 0.</p>
4	<b>PILA DE ITERACIÓN</b>	Lista de funcionalidades o requerimientos seleccionados para realizar durante una iteración.
5	<b>PILA DE PRODUCTO</b>	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: Definiciones y Abreviaturas de la Metodología de Desarrollo**

**Fuente: (Rea, 2012)**

### 1.2.5 Documentos de referencia

<b>DOCUMENTOS INTERNOS</b>	
N°	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 ( <a href="http://wiki.imi.gob.ec">http://wiki.imi.gob.ec</a> intranet)

<b>DOCUMENTOS EXTERNOS</b>	
N°	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.

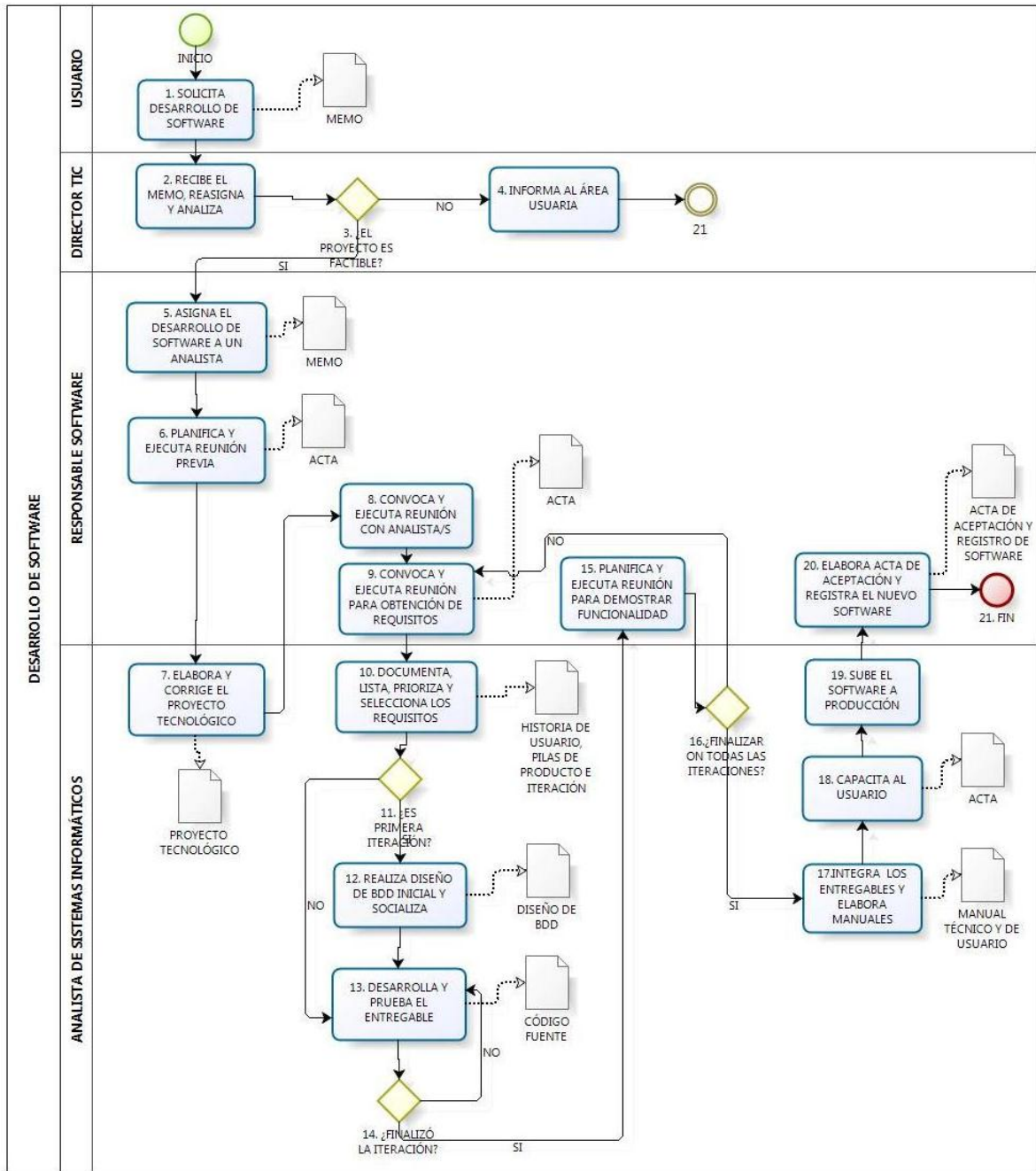
### 1.2.6 POLÍTICAS

- 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.
- 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”

**1.2.7 Diagrama de flujo**



**Figura 13: Proceso de Desarrollo de Software GAD-I**

Fuente: (Rea, 2012)

### 1.2.8 Descripción del procedimiento

N°	ACTIVIDAD	DESCRIPCIÓN	RESPONSABLE
1	<b>SOLICITA DESARROLLO DE SOFTWARE</b>	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	<b>RECIBE EL MEMO, REASIGNA Y ANALIZA</b>	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	<b>¿EL PROYECTO ES FACTIBLE?</b>	<ul style="list-style-type: none"> <li>▪ <b>Si el proyecto no es factible:</b> Va a la actividad N° 4.</li> <li>▪ <b>Si el proyecto es factible:</b> Continúa en la actividad N° 5.</li> </ul>	Director de TIC
4	<b>INFORMA AL ÁREA USUARIA</b>	Informa al área usuaria solicitante mediante un memorándum los motivos por los cuales no es factible el desarrollo del software.	Director de TIC
5	<b>ASIGNA EL DESARROLLO DE SOFTWARE A UN ANALISTA</b>	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	<b>PLANIFICA Y EJECUTA REUNIÓN PREVIA</b>	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	<b>ELABORA Y CORRIGE EL PROYECTO TECNOLÓGICO</b>	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	<b>CONVOCA Y EJECUTA REUNIÓN CON ANALISTAS</b>	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	<b>CONVOCA Y EJECUTA REUNIÓN PARA OBTENCIÓN DE REQUISITOS</b>	Convoca a una reunión al Usuario y Analista/s para determinar los requisitos del software. Registra el Acta de Reunión y de apoyo durante la obtención de requisitos, se puede elaborar una lista de preguntas.	Responsable Software



N°	ACTIVIDAD	DESCRIPCIÓN	RESPONSABLE
9		<b>NOTA:</b> <i>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>	
10	<b>DOCUMENTA, LISTA, PRIORIZA Y SELECCIONA LOS REQUISITOS</b>	<p>Documenta, con la participación del usuario, los requisitos obtenidos durante la Reunión de obtención de Requisitos en el Formato de Historia de Usuario.</p> <ul style="list-style-type: none"> <li>▪ Ver <b>FOR-3.9.4.1</b></li> </ul> <p>Elabora la pila del producto y la pila de iteración de acuerdo a las prioridades del sistema.</p> <ul style="list-style-type: none"> <li>▪ Ver <b>FOR-3.9.4.2</b> y <b>FOR-3.9.4.3</b></li> </ul>	Analista de Sistemas Informáticos
11	<b>¿ES PRIMERA ITERACIÓN?</b>	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	<b>REALIZA DISEÑO INICIAL DE BDD Y SOCIALIZA</b>	<p>Realiza el diseño inicial de la base de datos analizando los requerimientos obtenidos, y la reutilización de tablas existentes en el ambiente de producción.</p> <p>Socializa con los analistas de sistemas.</p> <p>Realiza las correcciones necesarias al diseño.</p>	Analista de Sistemas Informáticos
13	<b>DESARROLLA Y PRUEBA EL ENTREGABLE</b>	<p>Diseña, desarrolla y prueba el entregable de acuerdo a los requisitos de la Pila de Iteración. Coordina constantemente reuniones con el Área Usuaria para aclarar dudas acerca de los requerimientos del software.</p> <p><b>NOTA:</b> <i>Durante este paso, puede realizar correcciones al diseño de la base de datos.</i></p>	Analista de Sistemas Informáticos
14	<b>¿FINALIZÓ LA ITERACIÓN?</b>	<p>Si aún no finaliza la iteración, va al paso N° 13.</p> <p>Si finalizó la iteración va al paso N° 15.</p>	Analista de Sistemas Informáticos
15	<b>PLANIFICA Y EJECUTA REUNIÓN PARA DEMOSTRAR FUNCIONALIDAD</b>	Planifica una reunión con el Área Usuaria y el Analista de Sistemas para demostrar la funcionalidad realizada durante la iteración. Se registra el Acta de Reunión en el Módulo de Reuniones.	Responsable de Software
16	<b>¿FINALIZARON LAS ITERACIONES?</b>	<p>Si no finalizaron las iteraciones, va al paso N° 9.</p> <p>Si finalizaron las iteraciones, va al paso N° 17.</p> <p><b>NOTA:</b> <i>Las iteraciones finalizan cuando ya no existen historias de usuario por desarrollar.</i></p>	Responsable de Software

N°	ACTIVIDAD	DESCRIPCIÓN	RESPONSABLE
17	<b>INTEGRA LOS ENTREGABLES Y ELABORA MANUALES</b>	Integra los entregables realizados durante todo el proceso. Realiza ajustes y configuraciones necesarias. Elabora el Manual Técnico y de Usuario. ▪ <b>Ver FOR-3.9.4.4 y FOR-3.9.4.5</b>	Analista de Sistemas Informáticos
18	<b>CAPACITA AL USUARIO</b>	Convoca al Área Usuaria 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	<b>SUBE EL SOFTWARE A PRODUCCIÓN</b>	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	<b>ELABORA ACTA DE ACEPTACIÓN Y REGISTRA EL NUEVO SOFTWARE</b>	Elabora el Acta de Término de Proyectos o Requerimientos de acuerdo al formato. Entrega al usuario a través de memorándum para que la firme. ▪ <b>Ver FOR-3.9.4.6</b> Registra el nuevo software en la Caracterización de Software. Ver <b>FOR-3.9.4.8</b>	Responsable de Software
21	<b>FIN</b>		

**Tabla 2: Descripción del Proceso de Desarrollo de Software**

**Fuente: (Rea, 2012)**

### 1.2.9 Documentación de la metodología GAD-I

A continuación se presentan todos los documentos que se elaboran dentro de la Metodología GAD-I, indicando su formato y la guía de llenado.

<b>DOCUMENTOS METODOLOGIA GAD-I</b>	
<b>Número</b>	<b>Título del Documento</b>
1	FORMATO DE PROYECTO TECNOLÓGICO
2	FORMATO DE ACTA DE REUNIÓN
3	FORMATO DE HISTORIA DE USUARIO
4	FORMATO DE PILA DE PRODUCTO
5	FORMATO DE PILA DE ITERACIÓN
6	FORMATO DE MANUAL DE USUARIO
7	FORMATO DE MANUAL TÉCNICO
8	FORMATO DE ACTA DE TÉRMINO DE PROYECTOS
9	FORMATO DE CARACTERIZACIÓN DE APLICACIONES

❖ **Formato de Proyecto Tecnológico**

Documento donde se describe el conjunto de actividades dirigidas a conseguir un objetivo específico de investigación, desarrollo e innovación de tecnología en un intervalo de tiempo y costo definido además de especificar el alcance, justificación, beneficiarios entre otros aspectos relacionados con el sistema a desarrollar. **Ver Anexo N°1**

❖ **Formato de Acta de Reunión**

Documento que contiene detalladamente los temas tratados en una reunión de trabajo el cual puede ser con gente dentro del área de sistemas o con los usuarios que suelen tener como objetivos obtener requerimientos para el desarrollo del sistema y luego especificar los compromisos adquiridos que dieron como resultado de la reunión. **Ver Anexo N°2**

❖ **Formato de Historia de Usuario**

<b>HISTORIA DE USUARIO</b>			
<b>Fecha:</b>			
<b>Proyecto:</b>			
<b>Nombre de la historia:</b>			
<b>ID de historia:</b>		<b>Iteración Asignada:</b>	
<b>Dirección:</b>		<b>Funcionario:</b>	
<b>Unidad:</b>		<b>Cargo:</b>	
<b>Prioridad:</b>		<b>Estimación:</b>	
<b>Analista Programador Responsable:</b>		<b>Dependencia:</b>	
<b>Desarrollador:</b>			
<b>Descripción</b>			
<b>Pruebas de Aceptación</b>			
<b>Observaciones</b>			

**Tabla 3: Formato de Historia de Usuario**

Fuente: (Rea, 2012)

<b>GUÍA DE LLENADO</b>		
<b>Título del Formato:</b>	Historia de Usuario	
<b>Código del Formato:</b>	FOR-3.9.4.1	
<b>Descripción de cómo llenar el Formato</b>		
<b>N°</b>	<b>Campo</b>	<b>Instrucción</b>
1	Fecha	Fecha en que se documenta la historia
2	Proyecto	Colocar el nombre del proyecto.
3	Nombre de la Historia	Colocar un nombre representativo de la historia de usuario de acuerdo a los requerimientos que contenga.
4	ID de historia	Colocar un identificador para la historia de usuario. Debe ser único y arábigo: Ejemplo: 1, 2, 3, 4, 5 ...
5	Iteración Asignada	Colocar el número de la iteración estimada en la cual se va a ejecutar la historia. Ejemplo: 1, 2, 3, 4, 5 ...
6	Dirección	Dirección dentro de la Estructura Orgánica del GAD-I a la que pertenece el funcionario entrevistado.
7	Unidad	Unidad dentro de la Estructura Orgánica del GAD-I a la que pertenece el funcionario entrevistado.
8	Funcionario	Nombre del usuario que participó en el levantamiento de la historia.
9	Cargo	Cargo del funcionario
10	Prioridad	Para ello colocar la prioridad asignada a la historia de usuario de acuerdo a: <ul style="list-style-type: none"> <li>• <b>ALTA:</b> Describe al conjunto de requisitos que deben ser satisfechos en la solución final para que esta se considere un éxito.</li> <li>• <b>MEDIA:</b> Representa al conjunto de requisitos que deberían ser incluidos en la solución si es posible.</li> <li>• <b>BAJA:</b> Describe al conjunto de requisitos que se consideran deseables pero no necesarios. Deberían incluirse si los recursos y el tiempo lo permiten.</li> </ul>
11	Estimación	Colocar la estimación del esfuerzo en horas que tome ejecutar toda la historia. Por ejemplo 2.
12	Analista Programador Responsable	Colocar el nombre del Analista que desarrollará los requisitos definidos en la historia de usuario.
13	Dependencia	Una historia de usuario no debería ser dependiente de otra historia, pero a veces es inevitable. En este apartado se indicarían los IDs de las historias de las que depende la historia actual.
14	Desarrollador	Colocar el nombre del Desarrollador del Sistema en caso de no ser el Analista Programador Responsable.
15	Descripción	Listar y describir todos los requisitos en lenguaje del usuario, de acuerdo al alcance de la historia.
16	Pruebas de Aceptación	Listar y describir cuáles serán los criterios de aceptación de la presente Historia de Usuario de acuerdo a la descripción de los requisitos.
17	Observaciones	Colocar alguna observación acerca de la Descripción o Pruebas de Aceptación. (opcional)

**Tabla 4: Guía de llenado de Historia de Usuario**

**Fuente: (Rea, 2012)**

❖ **Formato de Pila de Producto**

<b>PILA DE PRODUCTO</b>		
<b>ID</b>	<b>Nombre Historia de Usuario</b>	<b>Prioridad</b>

**Tabla 5: Formato de Pila de Producto**

Fuente: (Rea, 2012)

<b>GUÍA DE LLENADO</b>		
<b>Título del Formato:</b>	Pila de Producto	
<b>Código del Formato:</b>	FOR-3.9.4.2	
<b>Descripción de cómo llenar el Formato</b>		
<b>N°</b>	<b>Campo</b>	<b>Instrucción</b>
1	ID	Colocar el ID de la Historia de Usuario.
2	Nombre Historia de Usuario	Colocar el nombre de la historia de Usuario.
3	Prioridad	Colocar la prioridad asignada a la Historia de Usuario, ya sea ALTA, MEDIA o BAJA.

**Tabla 6: Guía de llenado de Pila de Producto**

Fuente: (Rea, 2012)

❖ **Formato de Pila de Iteración**

<b>PILA DE ITERACIÓN</b>				<b>Día 1</b>	<b>Día2</b>	<b>DíaN</b>
<b>ID</b>	<b>Tareas</b>	<b>Tipo</b>	<b>Estado</b>	<b>Esfuerzo estimado</b>		
<b>Coloque aquí el Nombre de Historia de Usuario</b>						

**Tabla 7: Formato de Pila de Iteración**

Fuente: (Rea, 2012)

<b>GUÍA DE LLENADO</b>		
<b>Título del Formato:</b>	Pila de Iteración	
<b>Código del Formato:</b>	FOR-3.9.4.3	
<b>Descripción de cómo llenar el Formato</b>		
<b>N°</b>	<b>Campo</b>	<b>Instrucción</b>
1	ID	Colocar el ID de la Historia de Usuario.
2	Nombre Historia de Usuario	Colocar el nombre de la historia de Usuario.
3	Tareas	Listar las tareas en las que se ha dividido esa historia de usuario.
4	Tipo	Colocar el tipo de tarea de acuerdo a: ANÁLISIS, PROTOTIPADO, CODIFICACIÓN, PRUEBAS, REUNIÓN.
5	Estado	Colocar el estado de ejecución actual de la tarea de acuerdo a: PENDIENTE, EN CURSO, TERMINADA, ELIMINADA.
6	Día1, Día2, DíaN	Colocar las fechas de los días ascendentes de acuerdo al tamaño de la iteración.
7	Esfuerzo Estimado	Colocar el esfuerzo estimado de lo que falta de hacer de esa tarea en horas.

**Tabla 8: Guía de llenado de Pila de Iteración**

**Fuente: (Rea, 2012)**

❖ **Formato de Manual de Usuario**

Documento enfocado principalmente para el usuario final del sistema en el cual se describe detalladamente la funcionalidad del sistema en cada uno de sus módulos. **Ver Anexo N°3**

❖ **Formato de Manual Técnico**

Documento enfocado para los especialistas en el área de sistemas en el cual se describe la arquitectura del sistema, instalación, configuración, requerimientos del sistema entre otros datos técnicos. **Ver Anexo N°4**

❖ **Formato de Acta de Término de Proyectos**

<b>ACTA DE TÉRMINO DE PROYECTOS Ó REQUERIMIENTOS</b>	
<b>Sistema/Aplicación:</b>	
<b>Módulo(s):</b>	
<b>Departamento Solicitante:</b>	
<b>Elaboró:</b>	

<b>ACTA DE TÉRMINO DE PROYECTOS Ó REQUERIMIENTOS</b>				
<b>Analista Responsable:</b>				
<b>Fecha de entrega:</b>				
<b>Número de Orden:</b>				
<b>Fecha inicio:</b>				
<b>Director Sistemas:</b>				
<b>Responsable de Unidad:</b>				
<b>Descripción breve de las actividades:</b>				
<p>Estoy de acuerdo en que todas las actividades de este proyecto o requerimiento han terminado y estoy completamente satisfecho con la operación actual de los procesos y módulos implementados, por lo cual no tengo inconveniente en aceptar este proyecto y utilizarlo, en el entendido que cualquier adición o cambio será tratado como otro requerimiento.</p>				
<b>Participante</b>	<b>Área o Departamento</b>	<b>Firma</b>	<b>Fecha</b>	<b>Observaciones</b>

**Tabla 9: Formato de Acta de Termino de Proyecto**

**Fuente: (Rea, 2012)**

<b>GUÍA DE LLENADO</b>		
<b>Título del Formato:</b>	Acta de Término de Proyectos o Requerimientos	
<b>Código del Formato:</b>	FOR-3.9.4.6	
<b>Descripción de cómo llenar el Formato</b>		
<b>Nº</b>	<b>Campo</b>	<b>Instrucción</b>
1	Sistema/Aplicación	Colocar el nombre del Sistema/Aplicación desarrollado.
2	Módulo (s)	Colocar el nombre del Módulo(s) desarrollado(s).
3	Elaboró	Colocar el nombre de la persona que desarrolló el sistema en caso de que no sea funcionario de la Dirección. De no ser así dejar el campo vacío.
4	Analista Responsable	Colocar el nombre del analista responsable que guió/desarrollo el sistema.
5	Fecha de Entrega	Colocar la fecha en la que se entregó el sistema y/o módulo al área usuaria.
6	Número Orden	Colocar el número de memo en el que se solicitaba el desarrollo del sistema.
7	Fecha inicio	Colocar la fecha en que se dio inicio a la elaboración del sistema.
8	Director de Sistema	Colocar el nombre del Director de Sistemas.
9	Jefe de Unidad	Colocar el nombre del Jefe de la Unidad de Software.

Nº	Campo	Instrucción
10	Descripción breve de las actividades	Describir brevemente la funcionalidad y alcance del sistema y/o módulo que se desarrolló.
11	Participante	Colocar el nombre del usuario que participó y recibió la capacitación para el uso del sistema y/o módulo.
12	Área o departamento	Colocar el nombre del área o departamento que va a usar el sistema
13	Firma	Firma del usuario.
14	Fecha	Colocar la fecha en que firmó el usuario.
15	Observaciones	Colocar algunas observaciones al sistema y/o módulo realizadas por el usuario.

**Tabla 10: Guía de llenado de Acta de Termino de Proyecto**

Fuente: (Rea, 2012)

❖ **Formato de Caracterización de Aplicaciones**

<b>CARACTERIZACIÓN DE APLICACIONES</b>	
<b>Número:</b>	
<b>Nombre de la Aplicación:</b>	
<b>Nombre corto (SIGLAS):</b>	
<b>Programador/es Responsable/es:</b>	
<b>Unidad/es en la/s que se aplica:</b>	
<b>Descripción de la aplicación:</b>	
<b>Plataforma de Desarrollo:</b>	
<b>Tipo de Aplicación</b> <ul style="list-style-type: none"> <li>• Sistema <input type="checkbox"/></li> <li>• Subsistema <input type="checkbox"/></li> <li>• Aplicación Web <input type="checkbox"/></li> </ul>	<b>La aplicación fue desarrollada por</b> <ul style="list-style-type: none"> <li>• La institución <input type="checkbox"/></li> <li>• Empresa Externa <input type="checkbox"/></li> </ul>
<b>Uso del Sistema</b> <ul style="list-style-type: none"> <li>• Interno <input type="checkbox"/></li> <li>• Externo <input type="checkbox"/></li> <li>• Ambos <input type="checkbox"/></li> </ul>	<b>El sistema trabajará en red</b> <ul style="list-style-type: none"> <li>• Si <input type="checkbox"/></li> <li>• No <input type="checkbox"/></li> <li>• Nro. De usuarios <input type="checkbox"/></li> </ul>
<b>Dispone de Documentación</b> <ul style="list-style-type: none"> <li>• Manual de Usuario <input type="checkbox"/></li> <li>• Manual Técnico <input type="checkbox"/></li> </ul>	<b>Servidor</b>
<b>Sistemas relacionados</b>	

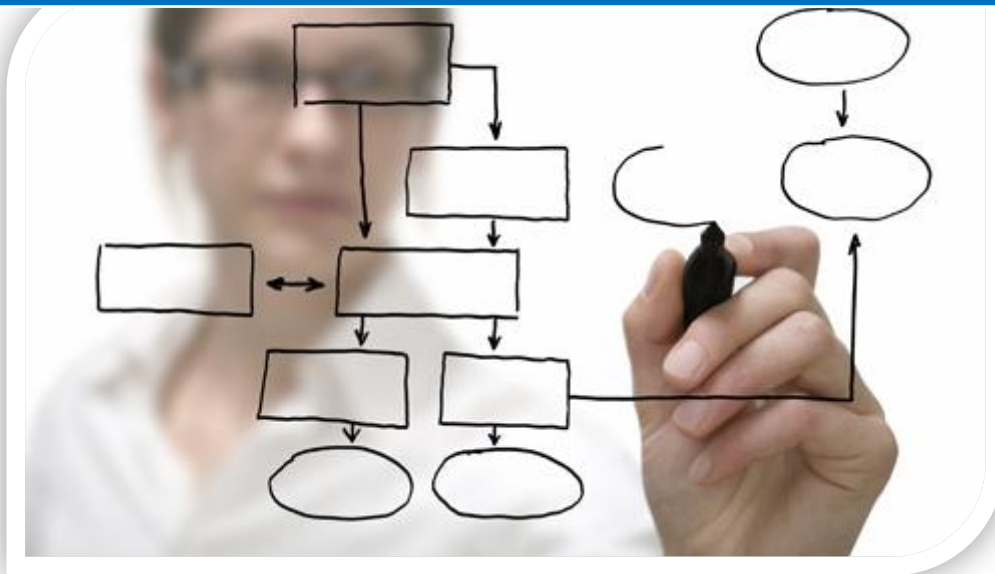
**Tabla 11: Formato de Documento de Caracterización de Aplicaciones**

Fuente: (Rea, 2012)





# Capítulo II: Proceso de Inspección



## CONTENIDO DEL CAPÍTULO

- DESCRIPCIÓN DEL PROCESO DE INSPECCIÓN
- OBJETIVO
- ALCANCE
- ROLES Y RESPONSABILIDADES
- DIAGRAMA DE FLUJO

## CAPÍTULO II

### PROCESO DE INSPECCIONES

En el presente capítulo se identifica y describe el proceso ejecutado dentro de la Dirección de Planificación para la realización de visitas de inspecciones a los predios.

#### 2.1. OBJETIVO

Controlar que las obras de infraestructura se realicen de acuerdo a los requerimientos y especificaciones técnicas dadas por la Dirección de Planificación, y sus respectivas ordenanzas.

#### 2.2. ALCANCE

El procedimiento aplica a las inspecciones que se realizan dentro del cantón Ibarra, ejecutadas por la Dirección de Planificación y Desarrollo. Inicia con el requerimiento interno o externo de inspección a un predio hasta la elaboración del informe técnico.

#### 2.3. ROLES Y RESPONSABILIDADES

<b>ROLES Y RESPONSABILIDADES</b>	
<b>Rol</b>	<b>Responsabilidad</b>
Director Planificación	<ul style="list-style-type: none"> <li>• Lidera el trabajo del grupo administrativo y técnico de la unidad.</li> <li>• Aprueba documentos técnicos para la construcción.</li> </ul>
Técnico OO.PP	<ul style="list-style-type: none"> <li>• Apoya en la supervisión, inspección y control de obras.</li> <li>• Realiza permisos de habitabilidad y definitivos de construcción.</li> </ul>
Arquitecto/a Planificación	<ul style="list-style-type: none"> <li>• Ejecutar de manera alternativa la aplicación de políticas, normas y estándares de ordenamiento y expansión urbana y de proyectos de desarrollo integral local y regional.</li> <li>• Organiza y ejecuta la revisión de documentos, planos y preparación de informe de aprobación de planos, sean arquitectónicos, declaración de propiedad horizontal, conjuntos habitacionales, actualizaciones y centro histórico.</li> </ul>
Secretaria OO.PP/ Planificación	<ul style="list-style-type: none"> <li>• Apoya en la recepción y despacho de trámites administrativos de la Unidad.</li> </ul>

**Tabla 12: Roles y Responsabilidades dentro del Proceso de Inspección**

**Fuente: Procesos GAD-I**

## 2.4. ABREVIATURAS Y DEFINICIONES

<b>ABREVIATURAS</b>		
<b>N°</b>	<b>TÉRMINO</b>	<b>DEFINICIÓN</b>
1	Tasa S.P.A	Tasa por Servicios públicos administrativos
2	GAD-I	Gobierno autónomo descentralizado municipal de Ibarra

<b>DEFINICIONES</b>		
<b>N°</b>	<b>TÉRMINO</b>	<b>DEFINICIÓN</b>
1	Cliente	Persona natural o jurídica que recibe los servicios del GAD-I.
2	Quipux	Sistema de Gestión Documental que soporta el registro, control, circulación y organización de los documentos digitales y/o impresos que se envían y reciben en el GAD-I.
3	Especies valoradas	Son documentos pre-elaborados y numerados, que están valorados y sirven para dar trámite a los diferentes servicios administrativos que realiza la Municipalidad de Ibarra.
4	Comprobante de pago	Es el documento que entrega la Municipalidad de Ibarra, a los contribuyentes que han pagado por impuesto, tasas, contribución u otros ingresos a la Municipalidad, en la ventanillas designadas para el efecto
5	Procedimiento	Documento donde se detalla los pasos a seguir para el cumplimiento de una actividad.

**Tabla 13: Abreviaturas y Definiciones dentro del Proceso de Inspección**

**Fuente: Procesos GAD-I**

## 2.5. DIAGRAMA DE FLUJO

Documentos:

D1: Cronograma de inspecciones
D2: Mapa del predio
D3: Solicitud de transporte
D4: Datos de la inspección.
D5: Plano topográfico / Planimétrico

DIAGRAMA DE FLUJO PARA EL PROCEDIMIENTO INSPECCIONES DE CAMPO

Fase

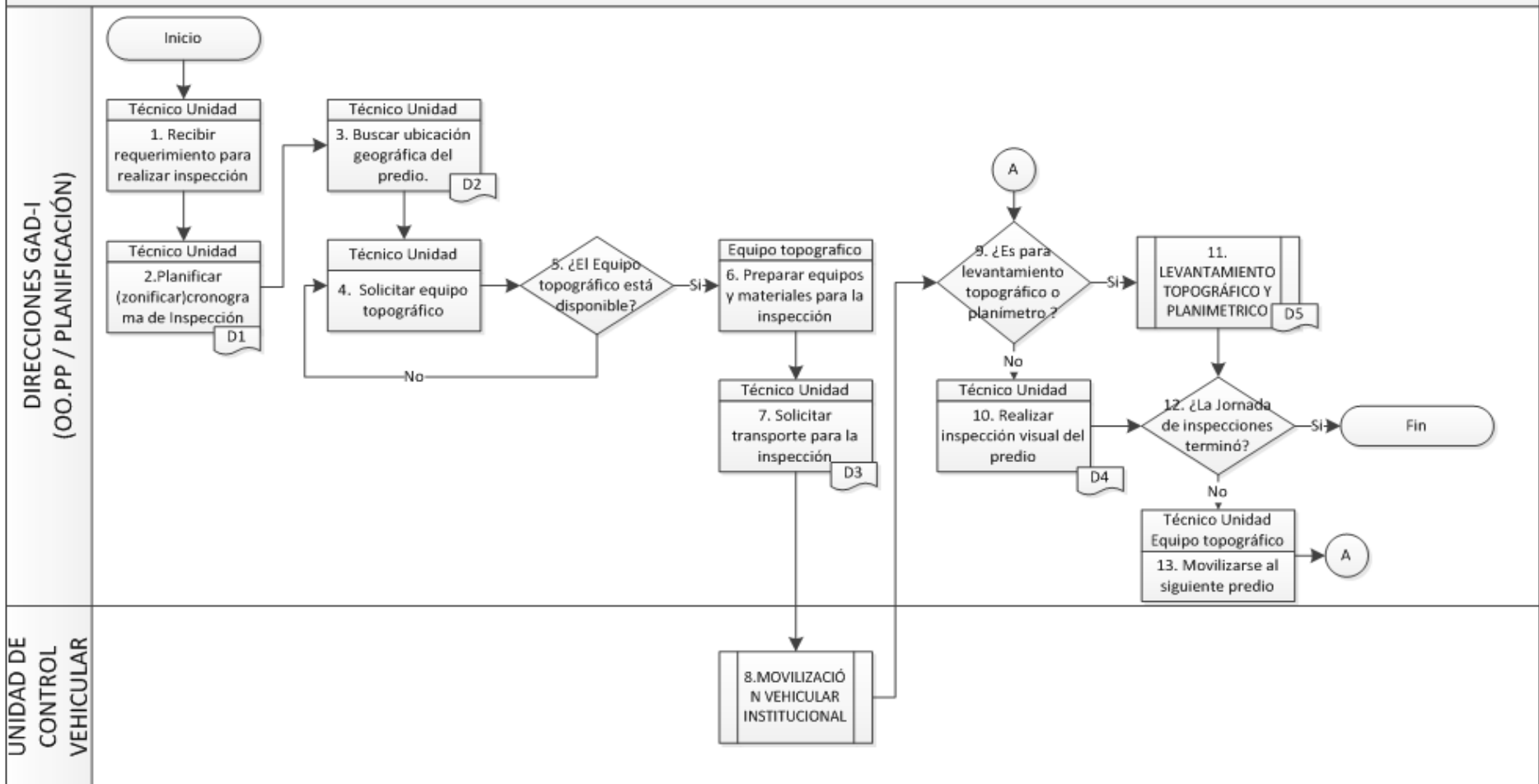


Figura 14: Diagrama de Flujo de Proceso de Inspección

Fuente: Procesos GAD-I

## 2.6. DESCRIPCIÓN DEL PROCEDIMIENTO

N°	ACTIVIDAD	DESCRIPCIÓN	RESPONSABLE
1	<b>RECIBIR REQUERIMIENTO PARA REALIZAR INSPECCIÓN</b>	Recibir la documentación y verificar el requerimiento de inspección.	Arquitecto Planificación
2	<b>BUSCAR UBICACIÓN GEOGRÁFICA DEL PREDIO.</b>	Buscar en el mapa la ubicación de los predios a inspeccionar	Arquitecto Planificación
3	<b>PLANIFICAR (ZONIFICAR) CRONOGRAMA DE INSPECCIÓN</b>	Elaborar el cronograma y ruta de inspecciones según zonas o conveniencia	Arquitecto Planificación
4	<b>SOLICITAR EQUIPO TOPOGRÁFICO</b>	Solicitar de forma verbal o escrita el equipo topográfico para cumplir las inspecciones.	Arquitecto Planificación
5	<b>¿EL EQUIPO TOPOGRÁFICO ESTÁ DISPONIBLE?</b>	<ul style="list-style-type: none"> <li>◆ Si el equipo topográfico está disponible continuar al paso N 6</li> <li>◆ Si el equipo topográfico no está disponible regresar al paso N 4</li> </ul>	Arquitecto Planificación
6	<b>PREPARAR EQUIPOS Y MATERIALES PARA LA INSPECCIÓN</b>	Verificar la lista de inspecciones a realizarse y preparar el equipo y materiales necesarios para ejecutar la inspección	Arquitecto Planificación Topógrafo Asistente Topografía
7	<b>SOLICITAR TRANSPORTE PARA LA INSPECCIÓN</b>	Elaborar solicitud de transporte para movilización del personal	Arquitecto Planificación
8	<b>MOVILIZACIÓN VEHICULAR INSTITUCIONAL</b>	Ver procedimiento para obtener movilización vehicular.	UNIDAD DE CONTROL VEHICULAR
9	<b>¿ES PARA LEVANTAMIENTO TOPOGRÁFICO O PLANÍMETRO?</b>	<ul style="list-style-type: none"> <li>◆ Si la inspección es para un levantamiento topográfico o planímetro continuar al paso N 11</li> <li>◆ Si la inspección no es para un levantamiento topográfico o planímetro continuar al paso N 10</li> </ul>	Arquitecto Planificación Topógrafo Asistente Topografía

N°	ACTIVIDAD	DESCRIPCIÓN	RESPONSABLE
10	<b>REALIZAR INSPECCIÓN VISUAL DEL PREDIO</b>	Realizar la inspección visual del predio y revisión de parámetro técnicos de acuerdo a ordenanzas municipales o normas de construcción, funcionalidad o habitabilidad, según sea el requerimiento.	Arquitecto Planificación Topógrafo Asistente Topografía
11	<b>LEVANTAMIENTO TOPOGRÁFICO Y PLANIMETRICO</b>	Ver procedimiento para los levantamientos topográficos y planimétricos.	Arquitecto Planificación Topógrafo Asistente Topografía
12	<b>¿LA JORNADA DE INSPECCIONES TERMINÓ?</b>	<ul style="list-style-type: none"> <li>◆ Si la jornada de inspecciones a terminado, procesar datos y elaborar informe de la inspección.</li> <li>◆ Si la jornada de inspecciones no ha terminado continuar al paso N 13</li> </ul>	Arquitecto Planificación Topógrafo Asistente Topografía
13	<b>MOVILIZARSE AL SIGUIENTE PREDIO</b>	Movilizarse al siguiente predio según el cronograma para ejecutar la inspección solicitada. Continuar al paso N 9	Arquitecto Planificación Topógrafo Asistente Topografía
	FIN		

**Tabla 14: Descripción del Procedimiento de Inspecciones**

**Fuente: Procesos GAD-I**

## 2.7. DOCUMENTOS DE REFERENCIA

<b>DOCUMENTOS INTERNOS</b>	
N°	Título del Documento
1	Resolución administrativa N° 44, Gestión Organizacional por procesos IMI 2010
2	Requisitos Trámites Municipales 2010
3	Reglamento Orgánico Funcional 2010
4	Ordenanza que regula el uso y ocupación del suelo 2012

<b>DOCUMENTOS EXTERNOS</b>	
N°	Título del Documento
1	Normas de control Interno
2	Código de Arquitectura y Urbanismo
3	COOTAD





# Capítulo III: Desarrollo del Sistema



## CONTENIDO DEL CAPÍTULO

- ELABORACIÓN DE PROYECTO DE SOFTWARE
- DISEÑO DEL SISTEMA
- DESARROLLO DEL SISTEMA

# CAPÍTULO III

## DISEÑO Y DESARROLLO DEL SISTEMA DE INSPECCIONES

En el presente capítulo se desarrolla el software siguiendo el proceso establecido por la metodología de desarrollo GAD-I de la Dirección de TICs que está basado en las dos metodologías de desarrollo ágil más conocidas XP y SCRUM.

### 3.1. ROLES

El desarrollo de un Sistema Informático es una tarea que está compuesta por varios elementos como procesos, metodologías, tecnologías de desarrollo y cada una tiene su importancia y objetivo que se llevan a cabo con la participación de un grupo de personas con sus diferentes roles y responsabilidades.

En la siguiente **tabla** describimos los roles que han formado parte del desarrollo del sistema:

Nombre Rol	Descripción	Responsabilidad
Usuario	Persona(s) encargada del proceso que va ser automatizado y quien va a determinar los requerimientos y funciones para el desarrollo del sistema.	<ul style="list-style-type: none"> <li>• Solicitar desarrollo del sistema.</li> <li>• Ayudar a identificar el proceso relacionado con el sistema.</li> <li>• Ayudar al levantamiento del proceso relacionado con el sistema.</li> <li>• Describir los requerimientos del sistema.</li> <li>• Ayudar a crear las historias de usuario.</li> </ul>
Director TIC	Principal Responsable del Departamento de Sistemas y encargado de gestionar el desarrollo del proyecto.	<ul style="list-style-type: none"> <li>• Recibir Solicitud de Desarrollo de Software.</li> <li>• Analizar y Reasignar el Proyecto.</li> <li>• Informar al Área de Sistemas y Usuaría.</li> </ul>

Nombre Rol	Descripción	Responsabilidad
Responsable de Software	Persona encargada de asignar un Proyecto a un Analista y realizar las pruebas de aceptación.	<ul style="list-style-type: none"> <li>• Asignar el Desarrollo de Software a un Analista y Programador.</li> <li>• Convocar y ejecutar Reuniones.</li> <li>• Elaborar Memo y Acta de Reunión.</li> <li>• Realiza pruebas de aceptación, asegurando que el sistema funcione según los requerimientos del usuario.</li> </ul>
Analista de Sistemas Informáticos	Persona encargada de dar seguimiento y apoyo al programador en el proceso de desarrollo del sistema.	<ul style="list-style-type: none"> <li>• Revisión y Análisis del Diseño de Base de Datos.</li> <li>• Revisión de historias de usuario.</li> <li>• Revisión de Iteraciones completadas.</li> <li>• Revisión y apoyo en el desarrollo del sistema.</li> <li>• Comunicar el avance del sistema al Responsable de Software.</li> </ul>
Programador	Persona responsable del desarrollo del sistema.	<ul style="list-style-type: none"> <li>• Participar en reuniones para la descripción y levantamiento del proceso.</li> <li>• Crear Actas de Reuniones.</li> <li>• Crear historias de usuario.</li> <li>• Crear documentos existentes en el proceso de desarrollo.</li> <li>• Realizar Prototipos para las pruebas de aceptación.</li> <li>• Programar código.</li> <li>• Diseñar Base de Datos.</li> </ul>

Nombre Rol	Descripción	Responsabilidad
Programador		<ul style="list-style-type: none"> <li>• Participar en reuniones para la planificación y demostración de funcionalidad del Sistema.</li> <li>• Integrar los Entregables y elaborar Manuales.</li> <li>• Capacitar al Usuario.</li> </ul>

**Tabla 15: Roles Participantes en el Desarrollo del Sistema**

**Fuente: Propia**

### **3.2. RECEPCIÓN DE SOLICITUD PARA DESARROLLO DE SOFTWARE**

La Dirección de Planificación al ver la necesidad de mejorar su proceso de inspecciones solicita a la Dirección de TIC el desarrollo de un sistema informático que automatice y registre las inspecciones realizadas por parte de los arquitectos.

El Director de TIC mediante el apoyo del Responsable de Software convoca a una reunión en la cual se analiza y verifica la factibilidad de desarrollar el “Sistema de Control de Procesos de Inspecciones para la Dirección de Planificación del GAD-I”.

### **3.3. ASIGNACIÓN Y PLANIFICACIÓN DEL PROYECTO TECNOLÓGICO**

Comprobado la factibilidad en el desarrollo del Sistema de Inspecciones se asigna el Proyecto a un Analista de Sistemas quien junto con un programador son los encargados de la realización del mismo.

Luego se realiza una reunión con las áreas de Sistemas y Planificación para comunicar sobre la inicialización del nuevo sistema informático y la asignación del proyecto al plan operativo anual.

### **3.4. ELABORACIÓN DEL PROYECTO TECNOLÓGICO**

Luego de la aceptación para dar inicio al desarrollo del sistema informático se elabora el documento de Proyecto Tecnológico en el cual se especifican los objetivos, alcance, beneficios entre otros aspectos importantes relacionados con este proyecto.

Este documento es revisado por el responsable de software quien junto con los Analistas de Sistemas de la Dirección de TIC dan su criterio y recomendaciones para la corrección del mismo en el cual queda especificado la información detallada para la creación del “Sistema de Control de Procesos de Inspecciones para la Dirección de Planificación del GAD-I”. **Ver Anexo N°5**

### **3.5. OBTENCIÓN DE REQUERIMIENTOS DEL SISTEMA**

Para la obtención de requerimientos del sistema se tuvo que realizar reuniones con los usuarios directos en ejecutar el proceso de inspección; que en este caso son los Arquitectos que trabajan dentro de la Dirección de Planificación del GAD-I.

Estas reuniones ayudaron a conocer el proceso que se lleva a cabo para ejecutar una inspección y cuáles son los requisitos pedidos por el Usuario Arquitecto para que el sistema sirva de apoyo en el control y registro de inspecciones; la información y temas tratados son documentados en Actas de Reuniones y los requerimientos obtenidos de estas constan en las Historias de Usuario. **Ver Anexo N°6**

Luego de presentarse el problema de no tener claro cuál o cuáles eran las fichas existentes para el registro de los datos de inspección se creó una encuesta dirigida hacia algunos arquitectos de la Dirección de Planificación lo cual dio como resultado que no existía una ficha definida por trámite o proceso por lo que se decidió agregar y desarrollar el módulo Fichas de Inspección. **Ver Anexo N°7**

### **3.6. DESARROLLO DE HISTORIAS DE USUARIO**

A continuación se muestran todas las historias de usuario creadas durante el desarrollo del proyecto en las cuales consta la información de los requisitos determinados por el usuario del sistema y como se demostrará su funcionalidad mediante las pruebas de aceptación.

<b>HISTORIA DE USUARIO</b>			
<b>Fecha:</b>	22 de Octubre del 2012		
<b>Proyecto:</b>	Sistema de Control de Procesos de Inspecciones para la Dirección de Planificación del Gobierno Autónomo Descentralizado de San Miguel de Ibarra		
<b>Nombre de la historia:</b>	Crear Horarios de Inspección		
<b>ID de historia:</b>	01	<b>Iteración Asignada:</b>	1
<b>Dirección:</b>	Planificación	<b>Funcionario:</b>	Arq. Lucia Iturralde
<b>Unidad:</b>	Control y Gestión Territorial	<b>Cargo:</b>	Arquitecta
<b>Prioridad:</b>	ALTA	<b>Estimación:</b>	6 semanas
<b>Analista Programador Responsable:</b>	Manuel Lara	<b>Dependencia:</b>	Ninguna
<b>Desarrollador:</b>	Edison Rivadeneira		
<b>Descripción</b>			
<ul style="list-style-type: none"> <li>▪ Como usuario es necesario que se pueda crear un horario de inspección donde se asigne las parroquias del Cantón Ibarra a un día de la semana y que sea modificable en cualquier momento.</li> <li>▪ Tomar en cuenta que se puede asignar una parroquia a dos días de la semana.</li> <li>▪ Los días de la semana para inspecciones es de lunes a viernes.</li> <li>▪ Además que se pueda imprimir el horario de inspecciones con el que actualmente se trabaje.</li> </ul>			
<b>Pruebas de Aceptación</b>			
<ul style="list-style-type: none"> <li>✓ Se creará una ventana donde estén paneles con los días lunes a viernes.</li> <li>✓ En cada panel se podrá agregar una nueva parroquia al día correspondiente.</li> <li>✓ Al dar click en el botón agregar incluido en cada panel se abrirá una nueva ventana donde este un listado de todas las parroquias del cantón.</li> <li>✓ Dentro de la ventana principal se incluirá un botón para abrir un PDF que muestre y se pueda imprimir el cuadro de horario de inspección actual.</li> </ul>			
<b>Observaciones</b>			

**Tabla 16: Historia de Usuario N°1: Crear Horarios de Inspección**

**Fuente: Propia**

<b>HISTORIA DE USUARIO</b>			
<b>Fecha:</b>	3 de Diciembre del 2012		
<b>Proyecto:</b>	Sistema de Control de Procesos de Inspecciones para la Dirección de Planificación del Gobierno Autónomo Descentralizado de San Miguel de Ibarra		
<b>Nombre de la historia:</b>	Asignación de arquitectos a parroquias		
<b>ID de historia:</b>	02	<b>Iteración Asignada:</b>	1
<b>Dirección:</b>	Planificación	<b>Funcionario:</b>	Arq. Lucia Iturralde
<b>Unidad:</b>	Control y Gestión Territorial	<b>Cargo:</b>	Arquitecta
<b>Prioridad:</b>	ALTA	<b>Estimación:</b>	3 semanas
<b>Analista Programador Responsable:</b>	Manuel Lara	<b>Dependencia:</b>	1
<b>Desarrollador:</b>	Edison Rivadeneira		
<b>Descripción</b>			
<ul style="list-style-type: none"> <li>▪ Como usuario es necesario que se pueda asignar uno o más arquitectos en cualquiera de las parroquias del Cantón Ibarra para que sean los encargados de realizar las inspecciones.</li> <li>▪ Las asignaciones de arquitectos a parroquias sean modificables en cualquier momento.</li> <li>▪ Al asignar un arquitecto se pueda elegir a uno de los que pertenecen o trabajan dentro de la Dirección de Planificación.</li> </ul>			
<b>Pruebas de Aceptación</b>			
<ul style="list-style-type: none"> <li>✓ Se creará una ventana para las asignaciones de arquitectos a parroquias.</li> <li>✓ Dentro de la ventana habrá un panel por cada parroquia.</li> <li>✓ Dentro de cada panel habrá un botón para al dar click se muestre un pequeña ventana donde se podrá elegir solo los arquitectos que trabajan en la dirección de planificación.</li> <li>✓ Se pondrá un botón junto al nombre del arquitecto asignado a la parroquia para poder quitar dicha asignación y así sea modificable en cualquier momento.</li> </ul>			
<b>Observaciones</b>			

**Tabla 17: Historia de Usuario N°2: Asignación de arquitectos a parroquias**

**Fuente: Propia**

<b>HISTORIA DE USUARIO</b>			
<b>Fecha:</b>	24 de Diciembre del 2012		
<b>Proyecto:</b>	Sistema de Control de Procesos de Inspecciones para la Dirección de Planificación del Gobierno Autónomo Descentralizado de San Miguel de Ibarra		
<b>Nombre de la historia:</b>	Crear Ficha de Inspección		
<b>ID de historia:</b>	03	<b>Iteración Asignada:</b>	2
<b>Dirección:</b>	Planificación	<b>Funcionario:</b>	Arq. Lucia Iturralde- Ing. Manuel Lara
<b>Unidad:</b>	Control y Gestión Territorial	<b>Cargo:</b>	Arquitecta
<b>Prioridad:</b>	ALTA	<b>Estimación:</b>	4 semanas
<b>Analista Programador Responsable:</b>	Manuel Lara	<b>Dependencia:</b>	Ninguna
<b>Desarrollador:</b>	Edison Rivadeneira		
<b>Descripción</b>			
<ul style="list-style-type: none"> <li>▪ Como usuario quiero que existe una ficha de inspección por trámite o proceso.</li> <li>▪ Que el usuario administrador pueda crear la ficha, la cual tenga sus respectivas variables o datos para registrar en la inspección.</li> <li>▪ También que el usuario administrador pueda modificar la ficha en cualquier momento, esto quiere decir que se pueda crear y quitar cualquier variable que pertenezca a una ficha.</li> </ul>			
<b>Pruebas de Aceptación</b>			
<ul style="list-style-type: none"> <li>✓ Según las necesidades del usuario se creará un módulo llamado Ficha de Inspección, el cual dentro de una ventana tendrá un componente <b>comboBox</b> donde se liste todos los procesos de la dirección de planificación.</li> <li>✓ Además habrá un <b>grid</b> con el listado de las variables pertenecientes al proceso seleccionado en el combo.</li> <li>✓ También habrá 2 botones para poder crear o agregar una variable a la ficha del proceso seleccionado.</li> <li>✓ Al crear una variable se abrirá una ventana donde se podrá escribir el nombre de la variable y una descripción de la misma.</li> <li>✓ Al agregar una variable se abrirá un grid donde se listará todas las variables ya creadas en otras fichas y se podrá seleccionar las que quiera agregar a la ficha del proceso seleccionado.</li> </ul>			
<b>Observaciones</b>			

**Tabla 18: Historia de Usuario N°3: Crear Ficha de Inspección**

**Fuente: Propia**



<b>HISTORIA DE USUARIO</b>			
<b>Fecha:</b>	14 de Enero del 2013		
<b>Proyecto:</b>	Sistema de Control de Procesos de Inspecciones para la Dirección de Planificación del GAD-I		
<b>Nombre de la historia:</b>	Crear categorías y tipos de variables en la ficha de inspección		
<b>ID de historia:</b>	04	<b>Iteración Asignada:</b>	2
<b>Dirección:</b>	Planificación	<b>Funcionario:</b>	Arq. Lucia Iturralde
<b>Unidad:</b>	Control y Gestión Territorial	<b>Cargo:</b>	Arquitecta
<b>Prioridad:</b>	ALTA	<b>Estimación:</b>	6 semanas
<b>Analista Programador Responsable:</b>	Manuel Lara	<b>Dependencia:</b>	3
<b>Desarrollador:</b>	Edison Rivadeneira		
<b>Descripción</b>			
<ul style="list-style-type: none"> <li>▪ Como usuario quiero poder crear categorías en la ficha de inspección para poder clasificar los datos registrados. Alguna de las categorías por ejemplo puede ser: datos prediales, salud y de medio ambiente entre otros.</li> <li>▪ Se debe poder crear variables que pertenezcan a una categoría de la ficha.</li> <li>▪ Además quiero tener la opción de que existan variables o datos que para registrar sean de selección única y múltiple.</li> <li>▪ También quiero poder visualizar e imprimir la ficha actual de inspección perteneciente a un trámite.</li> </ul>			
<b>Pruebas de Aceptación</b>			
<ul style="list-style-type: none"> <li>✓ De la misma forma como se crea las variables se tendrá la opción de crear una categoría por cada ficha de inspección perteneciente a un proceso.</li> <li>✓ Una ficha tendrá categorías y cada categoría tendrá sus variables.</li> <li>✓ Un cambio que se realizará es que para crear una variable se podrá dar doble click en cualquiera de las categorías y se abrirá la ventana de creación de variables.</li> <li>✓ Al crear variables se tendrá la opción de elegir qué tipo de variable se quiere crear, los cuales serán 3: simple, selección única y selección múltiple. <ul style="list-style-type: none"> <li>- La variable simple es la que tendrá un <i>textfield</i> para el ingreso de la información.</li> <li>- La variable de selección única es la que tendrá un grupo de <i>radiofield</i> para poder seleccionar solo una opción.</li> <li>- La variable de selección múltiple es la que tendrá un grupo de <i>checkbox</i> para seleccionar una o más opciones.</li> </ul> </li> <li>✓ También se creará una ventana donde se listará todos los procesos y al elegir uno se visualizará la ficha creada. Dentro de esta ventana habrá un botón para poder visualizar la ficha en un archivo PDF que a su vez se podrá imprimir.</li> </ul>			

**Tabla 19: Historia de Usuario N°4: Crear categorías y tipos de variables en la ficha de inspección**

**Fuente: Propia**

<b>HISTORIA DE USUARIO</b>			
<b>Fecha:</b>	25 de Febrero del 2013		
<b>Proyecto:</b>	Sistema de Control de Procesos de Inspecciones para la Dirección de Planificación del Gobierno Autónomo Descentralizado de San Miguel de Ibarra		
<b>Nombre de la historia:</b>	Ordenar trámites para generar inspecciones		
<b>ID de historia:</b>	05	<b>Iteración Asignada:</b>	3
<b>Dirección:</b>	Planificación	<b>Funcionario:</b>	Arq. Lucia Iturralde
<b>Unidad:</b>	Control y Gestión Territorial	<b>Cargo:</b>	Arquitecta
<b>Prioridad:</b>	ALTA	<b>Estimación:</b>	6 semanas
<b>Analista Programador Responsable:</b>	Manuel Lara	<b>Dependencia:</b>	1,2
<b>Desarrollador:</b>	Edison Rivadeneira		
<b>Descripción</b>			
<ul style="list-style-type: none"> <li>▪ Como usuario quiero poder ver un listado de los trámites según mi parroquia correspondiente.</li> <li>▪ Los trámites que existan en ese momento quiero poder ordenarlos a mi manera viendo el mapa de su ubicación para crear la ruta.</li> <li>▪ También quiero que luego de haber ordenado los trámites se pueda imprimir en una hoja cada trámite con algunos datos del predio incluyendo la foto y mapa del mismo.</li> </ul>			
<b>Pruebas de Aceptación</b>			
<ul style="list-style-type: none"> <li>✓ Según las necesidades del usuario se creará un módulo llamado Agenda de Inspecciones, en el cual dentro de una ventana habrá un <b>comboBox</b> con las parroquias asignadas según el arquitecto que ingrese al sistema y un campo para ingresar la cantidad de trámites a realizar inspección.</li> <li>✓ Además habrá dos paneles, en el primero estará el listado de los trámites ordenados por fecha de creación y en el segundo el mapa para visualizar la ubicación de los predios.</li> <li>✓ Cada ubicación en el mapa tendrá un gráfico señalador para saber dónde queda exactamente el predio, además de tener el número según el orden de los trámites por fecha.</li> <li>✓ Se podrá elegir el orden de ruta de inspección directamente dando click en cada señalador de los predios ubicados en el mapa.</li> <li>✓ Luego de haber ordenado los trámites para la ruta de inspección habrá un botón para que se genere un PDF con la información respectiva y en cada una se visualice la foto y mapa del predio, además de que el documento podrá ser impreso.</li> </ul>			
<b>Observaciones</b>			

**Tabla 20: Historia de Usuario N°5: Ordenar trámites para generar inspecciones**

**Fuente: Propia**

<b>HISTORIA DE USUARIO</b>			
<b>Fecha:</b>	5 de Abril del 2013		
<b>Proyecto:</b>	Sistema de Control de Procesos de Inspecciones para la Dirección de Planificación del GAD-I		
<b>Nombre de la historia:</b>	Ver información del trámite		
<b>ID de historia:</b>	06	<b>Iteración Asignada:</b>	3
<b>Dirección:</b>	Planificación	<b>Funcionario:</b>	Arq. Lucia Iturralde
<b>Unidad:</b>	Control y Gestión Territorial	<b>Cargo:</b>	Arquitecta
<b>Prioridad:</b>	ALTA	<b>Estimación:</b>	3 semanas
<b>Analista Programador Responsable:</b>	Manuel Lara	<b>Dependencia:</b>	5
<b>Desarrollador:</b>	Edison Rivadeneira		
<b>Descripción</b>			
<ul style="list-style-type: none"> <li>▪ Como usuario quiero poder ver información del trámite al momento de realizar la ordenación de los trámites para crear la ruta de inspecciones.</li> <li>▪ Alguna de la información que me gustaría ver es el número del trámite, datos del predio, datos del peticionario, foto del predio y mapa del predio.</li> <li>▪ Además quiero poder visualizar y descargar los archivos adjuntos en los trámites para poder identificar claramente de que se trata el trámite.</li> <li>▪ También me gustaría que en el archivo generado para las rutas de inspecciones se visualice el nombre del día junto a la fecha en la que se realizará la visita al predio.</li> </ul>			
<b>Pruebas de Aceptación</b>			
<ul style="list-style-type: none"> <li>✓ Se creará una ventana con la información del trámite donde conste lo siguiente: <ul style="list-style-type: none"> <li>- Número de Trámite</li> <li>- Nombre del proceso del trámite</li> <li>- Fecha del Trámite</li> <li>- Datos del Peticionario</li> <li>- Datos del Propietario</li> </ul> <p>Datos del Predio:</p> <ul style="list-style-type: none"> <li>- Clave Catastral</li> <li>- Dirección</li> <li>- Foto del Predio</li> <li>- Mapa del Predio</li> </ul> </li> <li>✓ Dentro de la misma ventana habrá un botón para abrir un listado de los archivos adjuntos en el trámite en el cual se tendrá un botón para descargarlos.</li> <li>✓ Se agregará el nombre del día de inspección junto a la fecha en el PDF generado de inspecciones.</li> </ul>			

**Tabla 21: Historia de Usuario N°6: Ver información del trámite**

**Fuente: Propia**

<b>HISTORIA DE USUARIO</b>			
<b>Fecha:</b>	30 de Abril del 2013		
<b>Proyecto:</b>	Sistema de Control de Procesos de Inspecciones para la Dirección de Planificación del Gobierno Autónomo Descentralizado de San Miguel de Ibarra		
<b>Nombre de la historia:</b>	Enviar e-mail al ciudadano informando el día de inspección		
<b>ID de historia:</b>	07	<b>Iteración Asignada:</b>	3
<b>Dirección:</b>	Planificación	<b>Funcionario:</b>	Arq. Lucy Iturralde
<b>Unidad:</b>	Control y Gestión Territorial	<b>Cargo:</b>	Arquitecta
<b>Prioridad:</b>	MEDIA	<b>Estimación:</b>	2 semanas
<b>Analista Programador Responsable:</b>	Manuel Lara	<b>Dependencia:</b>	5
<b>Desarrollador:</b>	Edison Rivadeneira		
<b>Descripción</b>			
<ul style="list-style-type: none"> <li>▪ Para mejorar la comunicación al ciudadano sobre el día inspección designado a visitar el predio se quisiera enviarlo a través de un correo electrónico.</li> <li>▪ En el correo quisiera que se muestre la fecha incluido el día de la inspección, además de quien es el arquitecto encargado y algunos datos del trámite.</li> </ul>			
<b>Pruebas de Aceptación</b>			
<ul style="list-style-type: none"> <li>✓ Especificadas las necesidades del usuario se investigará y se creará la función de envío de correos electrónicos en el momento de generar las inspecciones la cual tenga la siguiente información: Día y fecha de inspección, arquitecto encargado, datos del trámite y predio.</li> <li>✓ Esto tiene como objetivo dar a conocer rápidamente al ciudadano la fecha de inspección a su predio.</li> </ul>			
<b>Observaciones</b>			

**Tabla 22: Historia de Usuario N°7: Enviar e-mail al ciudadano informando el día de inspección**

**Fuente: Propia**

<b>HISTORIA DE USUARIO</b>			
<b>Fecha:</b>	13 de Mayo del 2013		
<b>Proyecto:</b>	Sistema de Control de Procesos de Inspecciones para la Dirección de Planificación del GAD-I		
<b>Nombre de la historia:</b>	Registro de Inspecciones		
<b>ID de historia:</b>	08	<b>Iteración Asignada:</b>	4
<b>Dirección:</b>	Planificación	<b>Funcionario:</b>	Arq. Lucia Iturralde
<b>Unidad:</b>	Control y Gestión Territorial	<b>Cargo:</b>	Arquitecta
<b>Prioridad:</b>	ALTA	<b>Estimación:</b>	5 semanas
<b>Analista Programador Responsable:</b>	Manuel Lara	<b>Dependencia:</b>	3,5
<b>Desarrollador:</b>	Edison Rivadeneira		
<b>Descripción</b>			
<ul style="list-style-type: none"> <li>▪ Como usuario quiero registrar las inspecciones que se han realizado</li> <li>▪ Quiero poder registrar los siguientes datos: <ul style="list-style-type: none"> <li>- El vehículo con el que se realiza el transporte a la visita del predio</li> <li>- Datos obtenidos en la visita</li> <li>- Guardar fotos con alguna descripción si es necesario</li> <li>- Escribir observaciones si existen</li> <li>- Registrar si ha sido aprobada o no la inspección, esto quiere decir si cumple con los datos del trámite y con otros requerimientos.</li> </ul> </li> </ul>			
<b>Pruebas de Aceptación</b>			
<ul style="list-style-type: none"> <li>✓ Según las necesidades del usuario se creará un módulo llamado Registro de inspecciones.</li> <li>✓ En una ventana se podrá ver en forma de Maestro-Detalle las agendas u orden de inspecciones siendo el maestro y como detalles un listado de las inspecciones generadas en el módulo Agenda de Inspecciones.</li> <li>✓ Para registrar una inspección se dará doble click en un ítem del listado de inspecciones el cual abrirá una ventana para realizar el registro. <ul style="list-style-type: none"> <li>- En la ventana de registro se podrá elegir el código vehicular de movilización para el transporte al realizar las visitas.</li> <li>- También habrá un botón para abrir la ficha de inspección y registrar los datos obtenidos.</li> <li>- Existirá otro botón para poder adjuntar las fotos tomadas en la inspección y escribir una descripción de la misma.</li> <li>- También habrá un <i>textarea</i> para escribir observaciones de la inspección</li> <li>- Y un <i>radiobutton</i> para seleccionar si la inspección fue aprobada o no.</li> </ul> </li> </ul>			

**Tabla 23: Historia de Usuario N°8: Registro de Inspecciones**

**Fuente: Propia**

<b>HISTORIA DE USUARIO</b>			
<b>Fecha:</b>	17 de Junio del 2013		
<b>Proyecto:</b>	Sistema de Control de Procesos de Inspecciones para la Dirección de Planificación del Gobierno Autónomo Descentralizado de San Miguel de Ibarra		
<b>Nombre de la historia:</b>	Recortar fotos de inspección		
<b>ID de historia:</b>	09	<b>Iteración Asignada:</b>	4
<b>Dirección:</b>	Planificación	<b>Funcionario:</b>	Arq. Lucia Iturralde
<b>Unidad:</b>	Control y Gestión Territorial	<b>Cargo:</b>	Arquitecta
<b>Prioridad:</b>	BAJA	<b>Estimación:</b>	3 semanas
<b>Analista Programador Responsable:</b>	Manuel Lara	<b>Dependencia:</b>	8
<b>Desarrollador:</b>	Edison Rivadeneira		
<b>Descripción</b>			
<ul style="list-style-type: none"> <li>▪ Como usuario quisiera poder recortar la foto que se ha tomado en las inspecciones, esto es porque existen fotos que solo se requiere una parte de la misma y no toda.</li> </ul>			
<b>Pruebas de Aceptación</b>			
<ul style="list-style-type: none"> <li>✓ Para realizar el recorte de la fotografía se agregará un botón editar en la vista previa de la imagen cargada al momento de adjuntar las fotos.</li> <li>✓ Al dar click en el botón editar se abrirá una ventana con la imagen cargada y para recortar se seleccionará con el puntero del mouse la parte necesaria.</li> <li>✓ Al recortar se reemplazará la imagen original cargada por la modificada para luego ser guardada.</li> </ul>			
<b>Observaciones</b>			

**Tabla 24: Historia de Usuario N°9: Recortar fotos de inspección**

**Fuente: Propia**

<b>HISTORIA DE USUARIO</b>			
<b>Fecha:</b>	2 de Julio del 2013		
<b>Proyecto:</b>	Sistema de Control de Procesos de Inspecciones para la Dirección de Planificación del Gobierno Autónomo Descentralizado de San Miguel de Ibarra		
<b>Nombre de la historia:</b>	Consultar las inspecciones realizadas		
<b>ID de historia:</b>	10	<b>Iteración Asignada:</b>	4
<b>Dirección:</b>	Planificación	<b>Funcionario:</b>	Arq. Lucia Iturralde
<b>Unidad:</b>	Control y Gestión Territorial	<b>Cargo:</b>	Arquitecta
<b>Prioridad:</b>	ALTA	<b>Estimación:</b>	3 semanas
<b>Analista Programador Responsable:</b>	Manuel Lara	<b>Dependencia:</b>	8
<b>Desarrollador:</b>	Edison Rivadeneira		
<b>Descripción</b>			
<ul style="list-style-type: none"> <li>▪ Como usuario quiero consultar las inspecciones que ya se han realizado y registrado.</li> <li>▪ Quiero poder ver todo el listado de inspecciones y que además se pueda filtrar por número de trámite, agenda de inspección y fecha de inspección.</li> <li>▪ Además quiero ver las inspecciones que han sido favorables o cuáles no.</li> <li>▪ También quiero ver las observaciones, las fotos y la ficha registrada.</li> </ul>			
<b>Pruebas de Aceptación</b>			
<ul style="list-style-type: none"> <li>✓ Se creará una ventana donde se pueda ver todo el listado de las inspecciones registradas.</li> <li>✓ Se podrá realizar filtros de búsqueda con los siguientes criterios: <ul style="list-style-type: none"> <li>- Consultar por agenda de inspecciones</li> <li>- Consultar por número de trámite</li> <li>- Consultar entre fechas donde se elegirá la fecha inicial y final para ver las inspecciones</li> </ul> </li> <li>✓ Se tendrá dos botones en los ítems del listado de inspecciones registradas donde: <ul style="list-style-type: none"> <li>- El primero será para ver las fotos que de guardaron en la inspección.</li> <li>- El segundo será para ver un PDF con los datos de la ficha de inspección registrada.</li> </ul> </li> <li>✓ Las fotos estarán con su respectiva descripción y se verán paginadas de 3 en 3</li> </ul>			
<b>Observaciones</b>			

**Tabla 25: Historia de Usuario N°10: Consultar las inspecciones realizadas**

**Fuente: Propia**

<b>HISTORIA DE USUARIO</b>			
<b>Fecha:</b>	22 de Julio del 2013		
<b>Proyecto:</b>	Sistema de Control de Procesos de Inspecciones para la Dirección de Planificación del Gobierno Autónomo Descentralizado de San Miguel de Ibarra		
<b>Nombre de la historia:</b>	Registrar el Personal responsable de la Inspección		
<b>ID de historia:</b>	11	<b>Iteración Asignada:</b>	4
<b>Dirección:</b>	Planificación	<b>Funcionario:</b>	Arq. Lucia Iturralde
<b>Unidad:</b>	Control y Gestión Territorial	<b>Cargo:</b>	Arquitecta
<b>Prioridad:</b>	MEDIA	<b>Estimación:</b>	3 semanas
<b>Analista Programador Responsable:</b>	Manuel Lara	<b>Dependencia:</b>	8
<b>Desarrollador:</b>	Edison Rivadeneira		
<b>Descripción</b>			
<ul style="list-style-type: none"> <li>▪ Como usuario quiero opcionalmente poder registrar los datos de las personas presentes al realizar la inspección de un predio.</li> <li>▪ Además de poder especificar quien es el responsable del grupo presente en la inspección.</li> </ul>			
<b>Pruebas de Aceptación</b>			
<ul style="list-style-type: none"> <li>✓ Dado las necesidades del usuario dentro de la ventana de Registro de Inspección se creará un <i>grid-panel</i> en el cual se podrá agregar uno o más personas que trabajen en el GAD-I y que hayan estado presentes en la inspección realizada.</li> <li>✓ Además el grid tendrá una columna <i>radiobutton</i> para seleccionar quien es el responsable del grupo de inspección.</li> </ul>			
<b>Observaciones</b>			

**Tabla 26: Historia de Usuario N°11: Registrar el Personal responsable de la Inspección**

**Fuente: Propia**



<b>HISTORIA DE USUARIO</b>			
<b>Fecha:</b>	9 de Agosto del 2013		
<b>Proyecto:</b>	Sistema de Control de Procesos de Inspecciones para la Dirección de Planificación del Gobierno Autónomo Descentralizado de San Miguel de Ibarra		
<b>Nombre de la historia:</b>	Ver Informe de Inspección Realizada		
<b>ID de historia:</b>	12	<b>Iteración Asignada:</b>	4
<b>Dirección:</b>	Planificación	<b>Funcionario:</b>	Arq. Lucia Iturralde
<b>Unidad:</b>	Control y Gestión Territorial	<b>Cargo:</b>	Arquitecta
<b>Prioridad:</b>	MEDIA	<b>Estimación:</b>	2 semanas
<b>Analista Programador Responsable:</b>	Manuel Lara	<b>Dependencia:</b>	8,10
<b>Desarrollador:</b>	Edison Rivadeneira		
<b>Descripción</b>			
<ul style="list-style-type: none"> <li>▪ Como usuario en la ventana de ver inspecciones realizadas quisiera también poder visualizar un informe de la inspección realizada.</li> <li>▪ Este informe debe contener una información resumida sobre datos del trámite, predio e inspección incluyendo si se ha registrado el personal responsable.</li> <li>▪ Además de ser un archivo para imprimir y al final de la hoja conste el nombre del arquitecto y el responsable del grupo para las firmas respectivas.</li> </ul>			
<b>Pruebas de Aceptación</b>			
<ul style="list-style-type: none"> <li>✓ Dado las necesidades del usuario se agregará en el <i>grid</i> de inspecciones realizadas una columna con un botón para al dar click se abra un archivo PDF mostrando el informe de inspección.</li> <li>✓ En este documento PDF se podrá ver la información organizada sobre datos del trámite y del predio.</li> <li>✓ Además de la información de la inspección como fecha, si se han registrado fotos, ficha y el personal responsable.</li> <li>✓ Al final se mostrará los nombres del arquitecto y el responsable de realizar la inspección incluidos espacios para sus respectivas firmas.</li> </ul>			
<b>Observaciones</b>			

**Tabla 27: Historia de Usuario N°12: Ver Informe de Inspección Realizada**

**Fuente: Propia**

<b>HISTORIA DE USUARIO</b>			
<b>Fecha:</b>	23 de Agosto del 2013		
<b>Proyecto:</b>	Sistema de Control de Procesos de Inspecciones para la Dirección de Planificación del Gobierno Autónomo Descentralizado de San Miguel de Ibarra		
<b>Nombre de la historia:</b>	Registro de Inspecciones en Dispositivo Móvil		
<b>ID de historia:</b>	13	<b>Iteración Asignada:</b>	5
<b>Dirección:</b>	Planificación	<b>Funcionario:</b>	Arq. Lucia Iturralde, San Pedro
<b>Unidad:</b>	Control y Gestión Territorial	<b>Cargo:</b>	Arquitectos
<b>Prioridad:</b>	MEDIA	<b>Estimación:</b>	3 semanas
<b>Analista Programador Responsable:</b>	Manuel Lara	<b>Dependencia:</b>	5,8
<b>Desarrollador:</b>	Edison Rivadeneira		
<b>Descripción</b>			
<ul style="list-style-type: none"> <li>▪ Como usuario quisiera tener la opción de realizar el registro de la inspección desde un dispositivo móvil donde se incluye las observaciones, la ficha de inspección y fotos.</li> <li>▪ Además me gustaría poder ver la foto y mapa del predio desde el móvil para una rápida localización.</li> </ul>			
<b>Pruebas de Aceptación</b>			
<ul style="list-style-type: none"> <li>✓ Dado las necesidades del usuario se creará una aplicación móvil para el registro de las inspecciones.</li> <li>✓ Esta aplicación tendrá la misma idea ya existente para el registro de inspecciones desde un entorno web.</li> <li>✓ La funcionalidad será la siguiente: <ul style="list-style-type: none"> <li>- Según el usuario que ingreso al sistema se podrá ver el listado de sus parroquias asignadas.</li> <li>- Una vez escogido la parroquia se podrá ver el listado de las agendas creadas la cual tiene la información del día y fecha de inspecciones.</li> <li>- De igual forma al elegir una agenda se listará los trámites con inspecciones que pertenecen a la misma.</li> <li>- De estos trámites se podrá elegir uno para primero ver una pequeña información del trámite, la foto y mapa del predio correspondiente.</li> <li>- Se agregará un botón para poder registrar la inspección en donde se podrá agregar las imágenes tomadas con el dispositivo móvil e ingresar los datos de la ficha de inspección.</li> </ul> </li> </ul>			
<b>Observaciones</b>			

**Tabla 28: Historia de Usuario N°13: Registro de Inspecciones en Dispositivo Móvil**

**Fuente: Propia**

<b>HISTORIA DE USUARIO</b>			
<b>Fecha:</b>	13 de Septiembre del 2013		
<b>Proyecto:</b>	Sistema de Control de Procesos de Inspecciones para la Dirección de Planificación del Gobierno Autónomo Descentralizado de San Miguel de Ibarra		
<b>Nombre de la historia:</b>	Reportes con Gráficos Estadísticos		
<b>ID de historia:</b>	14	<b>Iteración Asignada:</b>	6
<b>Dirección:</b>	Planificación	<b>Funcionario:</b>	Arq. Lucia Iturralde
<b>Unidad:</b>	Control y Gestión Territorial	<b>Cargo:</b>	Arquitecta
<b>Prioridad:</b>	MEDIA	<b>Estimación:</b>	3 semanas
<b>Analista Programador Responsable:</b>	Manuel Lara	<b>Dependencia:</b>	8, 13
<b>Desarrollador:</b>	Edison Rivadeneira		
<b>Descripción</b>			
<ul style="list-style-type: none"> <li>▪ Como usuario quisiera ver reportes de medición para cumplir indicadores</li> <li>▪ Me gustaría ver un gráfico estadístico en relación a las inspecciones realizadas</li> <li>▪ Principalmente quisiera ver un reporte de las inspecciones aprobadas y no aprobadas, por parroquias y por proceso o tipo de trámite.</li> <li>▪ Además quiero tener la opción de consultar entre fechas</li> <li>▪ También me gustaría poder imprimir el reporte gráfico de la consulta</li> </ul>			
<b>Pruebas de Aceptación</b>			
<ul style="list-style-type: none"> <li>✓ Según los requerimientos se creará un módulo de reportes.</li> <li>✓ Este módulo tendrá las ventanas para ver un gráfico estadístico con la información de las inspecciones realizadas y cuantas fueron o no aprobadas.</li> <li>✓ También se podrá ver un gráfico estadístico con la información de las inspecciones realizadas por parroquias y otro por proceso.</li> <li>✓ Estos gráficos se podrán visualizar con el valor total de inspecciones realizadas y además se podrá consultar entre una fecha inicial y final.</li> <li>✓ Además se agregará un botón para poder visualizar el reporte en un PDF el cual se podrá imprimir.</li> </ul>			
<b>Observaciones</b>			

**Tabla 29: Historia de Usuario N°14: Reportes con Gráficos Estadísticos**

**Fuente: Propia**

### 3.7. DISEÑO DEL SISTEMA

Luego de haber obtenido y descrito los requisitos del sistema en las historias de usuario se define cuáles son los módulos y tipos de usuario del sistema; además a continuación se describe la Arquitectura y Core para la Vista del Sistema Integrado GAD-I y la arquitectura funcional del Sistema de Inspecciones.

#### 3.7.1 Arquitectura del Sistema Integrado GAD-I

La arquitectura del Sistema Integrado, está basado principalmente en el patrón de diseño MVC (Modelo, Vista, Controlador), donde cada uno de estos componentes se usó según las necesidades y características de los framework usados para el desarrollo del sistema; además de tener la posibilidad de agregar más capas para una mejor estructura del código y mejorar la escalabilidad.

En la Figura N°15 se puede observar la arquitectura del sistema integrado y como se interactúa entre la parte del servidor y el cliente.

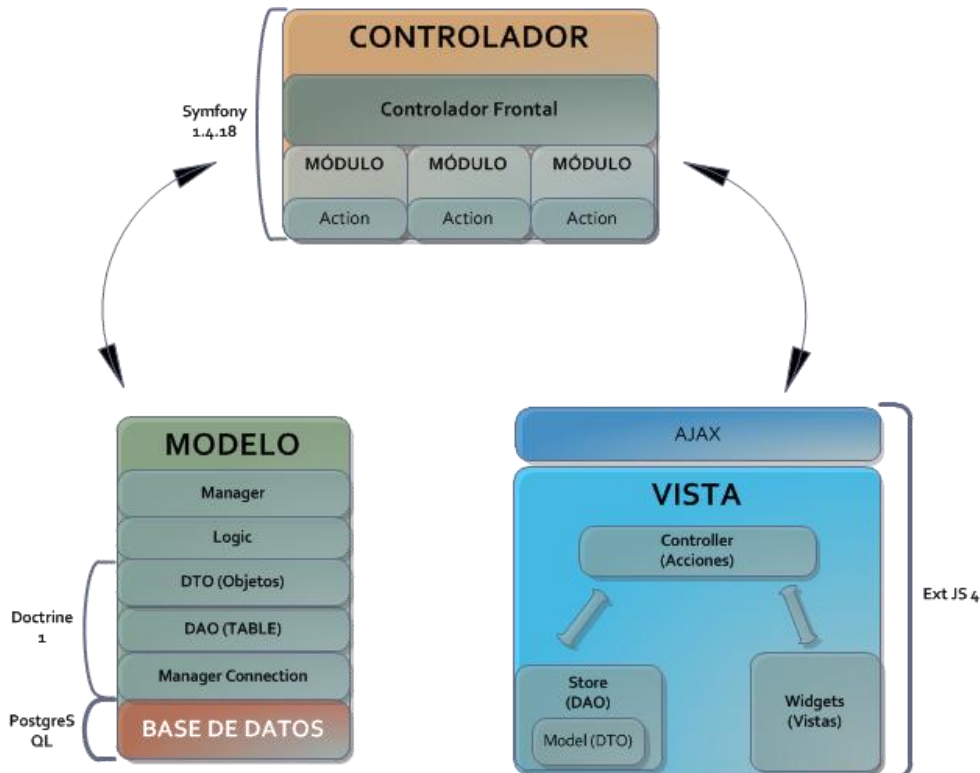


Figura 15: Arquitectura del Sistema

Fuente: Documentación GAD-I

**Modelo:** Esta capa mediante el uso del ORM Doctrine permite la comunicación y obtención de datos a través de las clases DTO<sup>37</sup> creadas a partir de las tablas de la base de datos PostgreSQL y el uso de los DAO (Table) para la ejecución de consultas SQL y DQL.

El manager fue creado para integrar y a la vez separar los componentes del controlador y los DAOs y funciona como intermediario por el cual se reciben solicitudes *request* desde el controlador y este llama a las consultas SQL o DQL para la obtención de datos a través de los objetos persistentes.

**Controlador:** Esta capa es la encargada de interactuar las consultas y datos obtenidos en el modelo para ser enviados y presentados en la vista. Basándose en el funcionamiento del framework Symfony esta capa nos provee de las siguientes características:

- El controlador frontal que tiene todo el enrutamiento de la aplicación y permite la asignación de los recursos de la misma, dependiendo del componente o modulo del sistema a ser solicitado.
- Los módulos existentes dentro de cada aplicación se encuentran fundamentado en el uso de sus acciones, donde cada una recibe el request de una solicitud del cliente y está a su vez invoca al manager encargado de gestionarla. Podríamos decir que estas acciones se convierten en los listeners de las solicitudes Ajax generadas por el cliente.

**Vista:** La capa de la vista está creada mediante el uso del framework ExtJS el cual nos provee de componentes ricos y dinámicos para un mejor manejo y representación de la información hacia el cliente.

Este framework utiliza como principal medio de solicitud de datos un conector Ajax, permitiendo mejorar el resultado de la respuesta y acceso a la misma. Además, la versión Ext JS 4.0.7 con la cual se desarrolló el sistema se fundamenta en el patrón de diseño MVC, el mismo que cuenta con las siguientes capas:

- **Widgets:** Estos son los componentes o vistas, que son mostradas al cliente, las mismas que contienen la información a ser presentada, mejorando la manipulación y acceso a la misma.
- **Model:** Esta capa es el mapeo u objeto de la entidad a ser presentada por la vista.

---

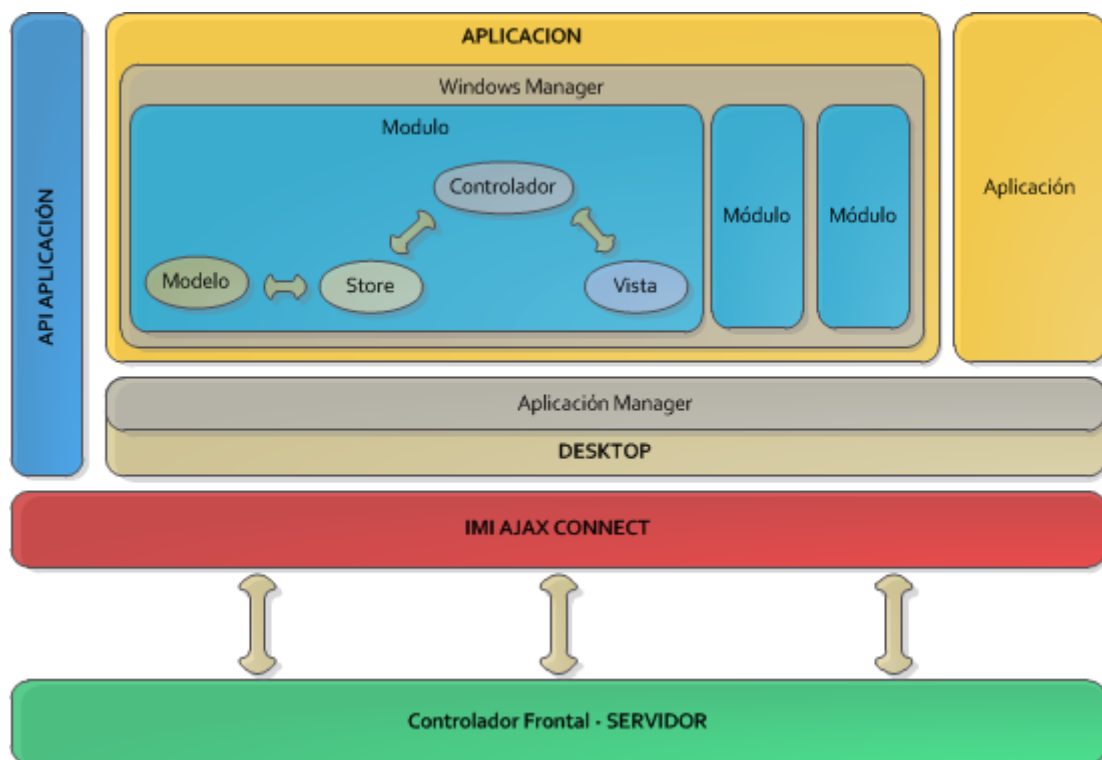
<sup>37</sup> DTO: Objeto de Transferencia de Datos (DTO por sus siglas en inglés)

- **Store:** Esta capa permite almacenar los datos usando la capa model para luego ser visualizado en componentes como comboBox, Grid, etc.
- **Controller:** Esta capa controla la lógica de la vista, sus métodos, eventos, así como también el envío de peticiones solicitadas con Ajax al servidor.

### 3.7.2 Core de la vista del Sistema Integrado GAD-I

El Core del sistema Integral relacionada con la vista se fundamenta a través de una arquitectura principal, con el fin de que esta sea más gestionable y nos permita obtener una única presentación y acceso al Sistema, ya que este cuenta con varias aplicaciones que tienen funciones diferentes dependiendo de los permisos del usuario logueado.

La Figura N°16 muestra la estructura y capas del Core para la vista.



**Figura 16: Estructura Core GAD-I**

**Fuente: Documentación GAD-I**

El Core de la vista está compuesto por las siguientes capas:

- **Imi Ajax Connect:** Este componente permite la comunicación entre la vista del cliente y la lógica que se encuentra en el servidor (Controlador Frontal), a su vez nos permite controlar

la sesiones creadas por el cliente y no tener inconsistencia al momento de comunicarse con el servidor.

- **Api Application:** Es la Instancia principal del Sistema Integral a nivel de la vista, el cual permite la gestión de todos los controladores y vistas creadas durante la ejecución de la aplicación, es la encargada de eliminar instancias y listeners activos al momento de cerrar una aplicación.
- **Desktop:** Es la vista principal del Sistema Integral, es la encargada de la administración de la carga de las aplicaciones, dentro de su entorno.
- **Aplicación Manager:** Este componente gestiona la carga de la vista principal (Desktop) y la de una aplicación asignando las opciones permitidas dentro de la misma dependiendo del usuario que ingreso al Sistema Integral.
- **Aplicación:** Este componente representa una aplicación en el nivel del servidor, en el cual cada una cuenta con las siguientes capas o componentes:
  - **Windows Manager:** Este componente permite gestionar la lógica de las ventanas a representarse dentro de una aplicación.
  - **Módulo:** Cada entidad dentro de una aplicación se gestiona a través de un módulo, tanto en la parte del cliente y servidor. El mismo que a su vez cumple con el patrón MVC del framework Ext JS, funcionalidad y capas que fueron descritas en la Arquitectura del Sistema Integral GAD-I.

### 3.7.3 Módulos del Sistema de Inspecciones

Basándose en las necesidades y requisitos obtenidos se determina la existencia de 2 tipos de usuarios o roles para el Sistema de Inspecciones los cuales son:

- **Usuario Administrador:** Este usuario es el encargado de crear los Horarios de Inspección y las Fichas de Inspección por proceso.
- **Usuario Arquitecto/Inspector:** Este usuario es el encargado de generar, registrar, consultar y obtener reportes de las inspecciones.

En la Figura N°17 se visualizan los módulos existentes en el Sistema de Inspecciones y sus permisos para los 2 tipos de usuario:

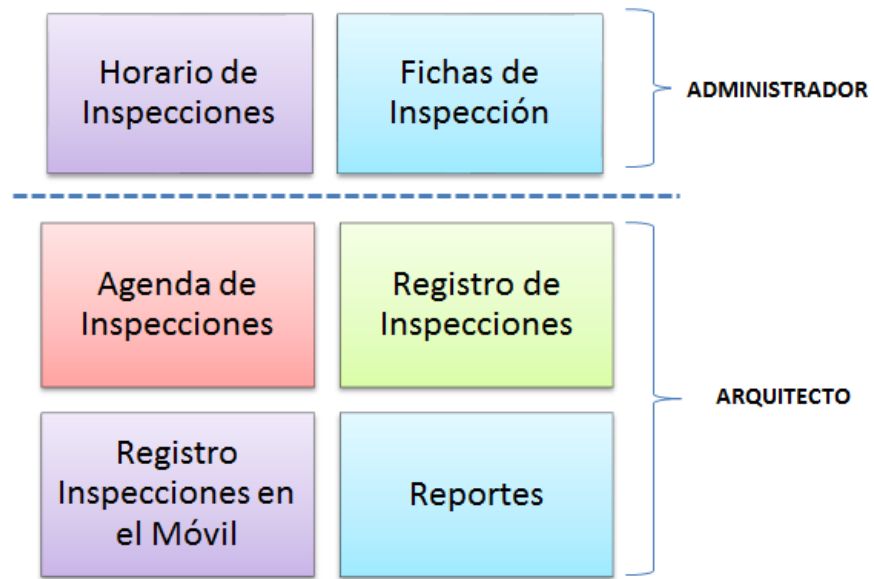


Figura 17: Módulos del Sistemas de Inspecciones

Fuente: Propia

- **Módulo Horario de Inspecciones:** En este módulo se asignan parroquias a días y arquitectos con lo cual se obtiene la información para generar el horario de inspecciones que luego sirve para asignar una fecha de visita a los predios.
- **Módulo Fichas de Inspección:** En este módulo se puede crear de forma dinámica una ficha de inspección por proceso o tipo de trámite el cual contiene categorías y dentro datos que pueden ser de 3 tipos: simple, selección única y selección múltiple. Esta ficha sirve para registrar los datos obtenidos al momento de realizar la inspección.
- **Módulo Agenda de Inspecciones:** En este módulo se receptan y visualizan los trámites que han sido seleccionados para realizar una inspección; además con la ayuda de un mapa donde se visualiza la ubicación de los predios se crea una ruta de inspección, el cual sirve de apoyo al momento de realizar las inspecciones.
- **Módulo Registro de Inspecciones:** En este módulo que tiene un entorno web se registran todos los datos de las inspecciones realizadas como son: fotos, ficha, observaciones y personas responsables de la inspección. Además de que las inspecciones realizadas luego se las puede buscar a través de algunos filtros como: fechas, agenda, número de trámite e inspecciones aprobadas.
- **Módulo Registro de Inspecciones en el Móvil:** Este módulo es similar al anterior con la diferencia de que su objetivo es ayudar a realizar un registro rápido de los datos obtenidos

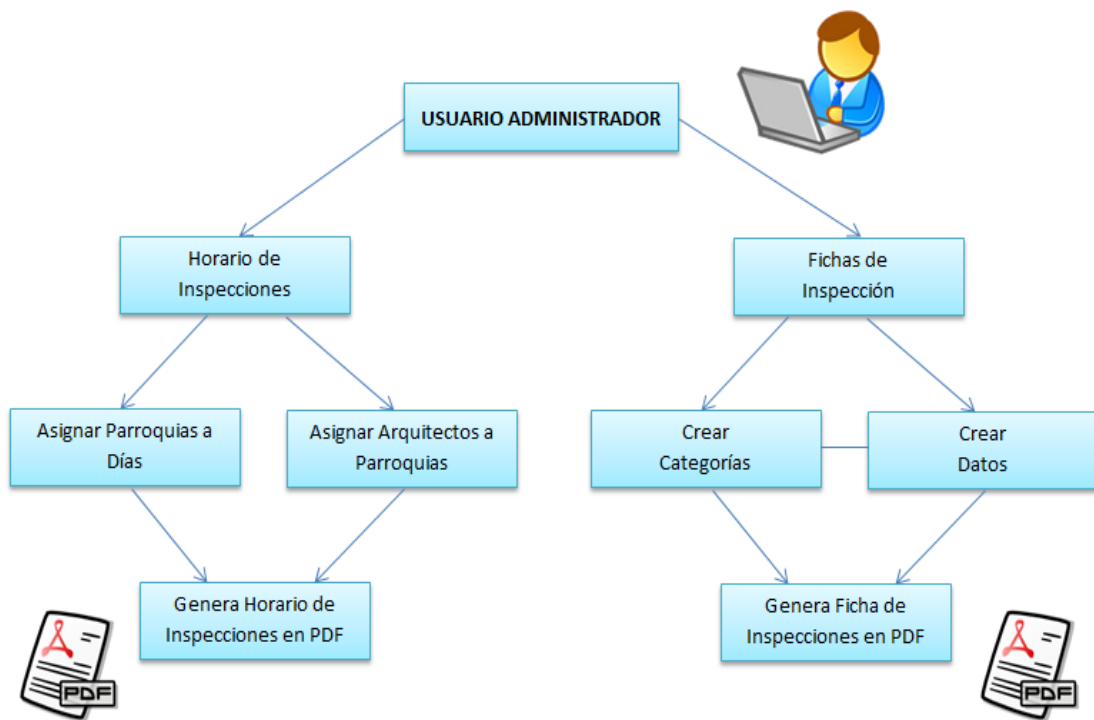


de inspección; esto es posible ya que fue creado para ejecutarlo en dispositivos móviles el cual se lo puede usar en el momento exacto de la visita a cada uno de los predios.

- **Módulo de Reportes:** Este módulo sirve para generar e imprimir reportes gráficos y estadísticos basándose en los datos de las inspecciones realizadas; además puede ser usado por los arquitectos para tener documentos de apoyo en el cumplimiento de indicadores existentes dentro del departamento.

### 3.7.4 Arquitectura Funcional del Sistema de Inspecciones

A continuación se visualiza el funcionamiento en general del Sistema de Inspecciones según la función de los 2 tipos de usuario:

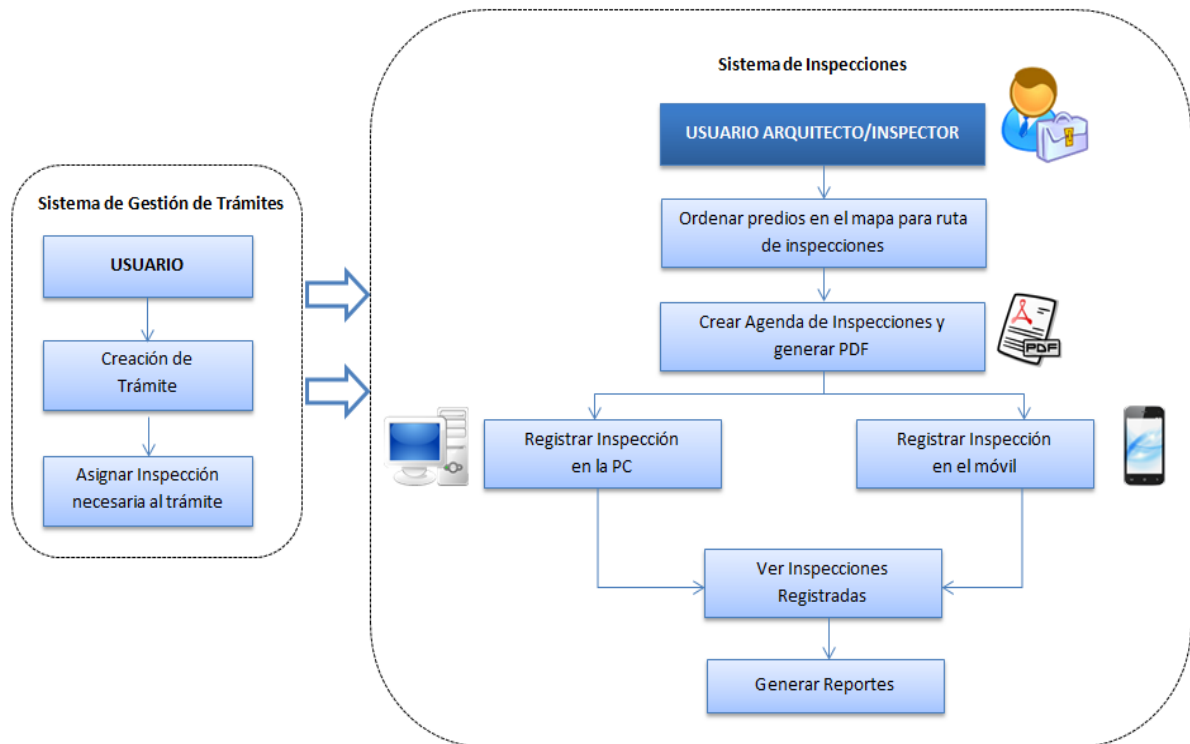


**Figura 18: Funcionalidad del Sistema de Inspección en base al Usuario Administrador**

**Fuente: Propia**

En la Figura N°18 podemos visualizar el funcionamiento del sistema según los permisos del usuario administrador el cual es el encargado de usar el Módulo de Horario de Inspecciones donde mediante la asignación de las parroquias a los días y arquitectos se puede crear e imprimir el Horario actual de inspecciones con el cual se trabaja dentro de la Dirección de Planificación.

Además este usuario tiene acceso al módulo de Fichas de Inspecciones donde puede visualizar y crear nuevas fichas por proceso o tipo de trámite en las cuales se pueden crear o agregar categorías y a su vez en estas crear los datos que va a contener la ficha que luego es usada para el registro de inspecciones. Esta ficha puede ser consultada por proceso y visualizada en formato PDF para imprimirla.



**Figura 19: Funcionalidad del Sistema de Inspección en base al Usuario Inspector**

**Fuente: Propia**

En la Figura N°19 podemos observar el funcionamiento del sistema según los permisos del usuario inspector o arquitecto el cual tiene una relación muy importante con el Sistema de Gestión de Trámites dado que es en él, donde se crean los trámites con toda la información necesaria como datos del trámite, predio, persona solicitante y dueño del predio; además en este sistema se eligen cuáles son los trámites que necesitan realizar una inspección.

Luego de haber creado y seleccionado los trámites que necesitan inspección, se reciben y visualizan en el Módulo Agenda de Inspecciones en el cual se determina el número de inspecciones a realizarse y con la ayuda de un mapa de ubicación de los predios, el usuario crea la ruta y agenda de inspecciones que da como resultado generar un archivo PDF con la información básica de los trámites como fecha de inspección, datos del trámite, foto y mapa del predio; además en el momento de generar una agenda de inspecciones se envía un correo electrónico a los ciudadanos

informando la fecha asignada automáticamente según el horario de inspecciones en el cual se va a visitar el predio a ser inspeccionado.

Después de haber generado las inspecciones se realiza su registro, el cual se lo puede hacer a través de un entorno web y móvil; en estos se registran los datos necesarios obtenidos en la inspección como: ficha, fotos, personal responsable, observaciones, entre otros.

Por ultimo este usuario puede consultar las inspecciones realizadas y basándose en éstas, pueda generar reportes estadísticos de: inspecciones realizadas, inspecciones por parroquia e inspecciones por proceso; estos se pueden visualizar en un archivo PDF e imprimirlos.

### 3.8. PILA DE PRODUCTO

La Pila de Producto es el artefacto donde se enlista todas las historias de usuario creadas a partir de la obtención de requerimientos, las cuales están priorizadas según las tareas que se tienen que realizar.

Mediante esta pila podemos observar de forma resumida las funcionalidades y características del sistema para luego clasificarlas según los módulos a desarrollarse.

A continuación se presenta la pila de producto obtenida para desarrollar el Sistema de Control de Procesos de Inspección:

<b>PILA DE PRODUCTO</b>		
<b>ID</b>	<b>Nombre de Historia de Usuario</b>	<b>Prioridad</b>
1	Crear Horarios de Inspección	ALTA
2	Asignación de arquitectos a parroquias	ALTA
3	Crear Ficha de Inspección	ALTA
4	Crear categorías y tipos de variables en la ficha de inspección	ALTA
5	Ordenar trámites para generar inspecciones	ALTA
6	Ver información del trámite	ALTA
7	Enviar e-mail al ciudadano informando el día de inspección	ALTA
8	Registro de Inspecciones	ALTA
9	Recortar fotos de inspección	BAJA
10	Consultar las inspecciones realizadas	ALTA
11	Registrar el Personal responsable de la Inspección	MEDIA
12	Ver Informe de Inspección Realizada	ALTA
13	Registro de Inspecciones en Dispositivo Móvil	MEDIA
14	Reportes con Gráficos Estadísticos	MEDIA

**Tabla 30: Pila de Producto del Sistema de Inspecciones**

**Fuente: Propia**

### 3.9. PILAS DE ITERACIÓN

A continuación se presentan todas las pilas de iteraciones agrupadas por los módulos del Sistema en las cuales luego se describen las actividades de sus respectivas historias de usuario:

#### Módulo 1: Horarios de Inspecciones

Pila Iteración: 1				
ID historia	Nombre de Historia	Estado	Estimación(semanas)	Prioridad
1	Crear Horarios de Inspección	TERMINADA	6 semanas	ALTA
2	Asignación de arquitectos a parroquias	TERMINADA	3 semanas	ALTA

#### Módulo 2: Fichas de Inspección

Pila Iteración: 2				
ID historia	Nombre de Historia	Estado	Estimación(semanas)	Prioridad
3	Crear Ficha de Inspección	TERMINADA	4 semanas	ALTA
4	Crear categorías y tipos de variables en la ficha de inspección	TERMINADA	6 semanas	ALTA

#### Módulo 3: Agenda de Inspecciones

Pila Iteración: 3				
ID historia	Nombre de Historia	Estado	Estimación(semanas)	Prioridad
5	Ordenar trámites para generar inspecciones	TERMINADA	6 semanas	ALTA
6	Ver información del trámite	TERMINADA	3 semanas	ALTA
7	Enviar e-mail al ciudadano informando el día de inspección	TERMINADA	2 semanas	ALTA

#### Módulo 4: Registro de Inspecciones

Pila Iteración: 4				
ID historia	Nombre de Historia	Estado	Estimación(semanas)	Prioridad
8	Registro de Inspecciones	TERMINADA	5 semanas	ALTA
9	Recortar fotos de inspección	TERMINADA	3 semanas	BAJA
10	Consultar las inspecciones realizadas	TERMINADA	3 semanas	ALTA

ID historia	Nombre de Historia	Estado	Estimación(semanas)	Prioridad
11	Registrar el Personal responsable de la Inspección	TERMINADA	3 semanas	MEDIA
12	Ver Informe de Inspección Realizada	TERMINADA	2 semanas	ALTA

**Módulo 5:** Registro de Inspecciones en el Móvil

Pila Iteración: 5				
ID historia	Nombre de Historia	Estado	Estimación(semanas)	Prioridad
13	Registro de Inspecciones en Dispositivo Móvil	TERMINADA	3 semanas	MEDIA

**Módulo 6:** Reportes

Pila Iteración: 6				
ID historia	Nombre de Historia	Estado	Estimación(semanas)	Prioridad
14	Reportes con Gráficos Estadísticos	TERMINADA	3 semanas	MEDIA

### 3.9.1 PILA DE ITERACIÓN 1

PILA DE ITERACIÓN 1				Esfuerzo estimado														
Historia N°1: Crear Horarios de Inspección				Octubre 2012				Noviembre 2012										
ID	Tareas	Tipo	Estado	V	L	M	M	J	L	M	M	J	V	L	M	M	J	
				26	29	30	31	1	5	6	7	8	9	12	13	14	15	
1	Capacitación y estudio de la base de datos del Municipio	ANÁLISIS	TERMINADO	4h														
2	Capacitación y revisión del framework EXTJS4	ANÁLISIS	TERMINADO	4h	4h													
3	Estudio de EXTJS4: Creación de clases, Herencia Múltiple, Definición de alias y xtype	ANÁLISIS	TERMINADO		4h	4h												
4	Visualización del Proceso de Inspecciones	ANÁLISIS	TERMINADO			4h												
5	Capacitación EXTJS4 integrado a Symfony	ANÁLISIS	TERMINADO				4h											
6	Capacitación EXTJS4 usando MVC: Controlador, Modelo, Store, Vista	ANÁLISIS	TERMINADO				4h											
7	Elaboración de Cuestionario para obtención de requerimientos	ANÁLISIS	TERMINADO					4h										
8	Ejecución de cuestionario para obtención de requerimientos	ANÁLISIS	TERMINADO					4h	4h									
9	Capacitación EXTJS4: Creación de un CRUD e integración final con Symfony	ANÁLISIS	TERMINADO						4h	4h								
10	Análisis de la Base de Datos para trámites de planificación	ANÁLISIS	TERMINADO							3h								
11	Elaboración y Diseño de la base de datos inicial para inspecciones	PROTOTIPADO	TERMINADO								4h	4h	3h					
12	Creación de la Aplicación Control de Inspecciones en Symfony dentro del proyecto integrado	CODIFICACIÓN	TERMINADO										1h	2h				
13	Creación del módulo Horarios de Inspecciones	CODIFICACIÓN	TERMINADO											2h				
14	Registro en la BDD la información del sistema, módulos, menús, submenús, permisos.	CODIFICACIÓN	TERMINADO												4h			
15	Creación de la aplicación Control de Inspecciones en EXTJS4	CODIFICACIÓN	TERMINADO													4h	4h	

Continuación Pila de Iteración 1 - Historia N°1				Noviembre 2012											Diciembre 2012			
ID	Tareas	Tipo	Estado	V	L	M	M	J	V	L	M	M	J	V	L	M	M	
				16	19	20	21	22	23	26	27	28	29	30	3	4	5	
16	Creación del módulo Horarios de Inspecciones en ExtJS	CODIFICACIÓN	TERMINADO	4h	2h													
17	Creación de controlador, models y stores para el módulo Horarios de Inspección	CODIFICACIÓN	TERMINADO		2h	4h												
18	Creación de ventana para asignar parroquias a días	CODIFICACIÓN	TERMINADO				4h											
19	Crear paneles para cada día de Lunes a Viernes	CODIFICACIÓN	TERMINADO					2h										
20	Crear consulta para obtener las parroquias del cantón Ibarra	CODIFICACIÓN	TERMINADO					2h										
21	Programar los métodos para asignar una parroquia a un día	CODIFICACIÓN	TERMINADO						4h									
22	Programar los métodos en Symfony para insertar las asignaciones en las tablas de la BDD	CODIFICACIÓN	TERMINADO							3h								
23	Agregar el boton para visualizar el calendario de inspecciones	CODIFICACIÓN	TERMINADO							1h								
24	Agregar, configurar y usar el puglin TCPDF en Symfony	CODIFICACIÓN	TERMINADO								4h	4h						
25	Crear consulta para obtener el horario de inspección actual	CODIFICACIÓN	TERMINADO										4h	4h				
26	Programar los métodos para visualizar el horario en PDF	CODIFICACIÓN	TERMINADO												4h	4h		
27	Realizar pruebas ingresando datos de asignación de parroquias a un día para inspección y visualizar el PDF para imprimir	REUNIÓN	TERMINADO															4h

**Tabla 31: Pila de Iteración 1 – Historia de Usuario N°1**

**Fuente: Propia**

PILA DE ITERACIÓN 1				Esfuerzo estimado												
Historia N°2: Asignación de arquitectos a parroquias				DICIEMBRE 2012												
ID	Tareas	Tipo	Estado	J	V	L	M	M	J	V	L	M	M	J	V	
				06	07	10	11	12	13	14	17	18	19	20	21	
1	Creación y uso de la tabla para asignación de arquitectos a parroquias	PROTOTIPADO	TERMINADO	4h												
2	Creación del módulo <i>ArquitectoParroquia</i> en Symfony y ExtJS	CODIFICACIÓN	TERMINADO		4h	1h										
3	Creación del controlador para el módulo <i>ArquitectoParroquia</i>	CODIFICACIÓN	TERMINADO			3h										
4	Creación de modelos para las tablas	CODIFICACIÓN	TERMINADO				2h									
5	Creación de stores en el módulo <i>ArquitectoParroquia</i>	CODIFICACIÓN	TERMINADO				2h									
6	Creación de ventana y paneles para asignar arquitectos a parroquias	CODIFICACIÓN	TERMINADO					4h	2h							
7	Crear consulta en Symfony para obtener los arquitectos que pertenecen a la dirección de planificación	CODIFICACIÓN	TERMINADO						2h	2h						
8	Programar los métodos en ExtJS para asignar un arquitecto a una parroquia	CODIFICACIÓN	TERMINADO								4h					
9	Programar los métodos en Symfony para insertar las asignaciones en las tablas de la BDD	CODIFICACIÓN	TERMINADO									3h				
10	Modificar los métodos para visualizar el horario de inspecciones en PDF incluido los arquitectos por parroquia	CODIFICACIÓN	TERMINADO											4h	4h	
11	Realizar pruebas ingresando datos de asignación de arquitectos a una parroquia y visualizar el PDF resultante de horarios de inspección	PRUEBAS	TERMINADO													4h

Tabla 32: Pila de Iteración 1 – Historia de Usuario N°2

Fuente: Propia



### 3.9.2 PILA DE ITERACIÓN 2

PILA DE ITERACIÓN 2				Esfuerzo estimado													
Historia N°3: Crear Ficha de Inspección				DIC. 2012			ENERO 2013										
ID	Tareas	Tipo	Estado	M	J	V	M	J	V	L	M	M	J	V	L	M	
				26	27	28	02	03	04	07	08	09	10	11	14	15	
1	Crear tablas en la BDD para la creación de fichas de inspección	PROTOTIPADO	TERMINADO	2h													
2	Generar schema y model en Symfony con las nuevas tablas	CODIFICACIÓN	TERMINADO	1h													
3	Creación del módulo Fichas de Inspecciones en Symfony y ExtJS	CODIFICACIÓN	TERMINADO	1h	3h												
4	Creación del controlador, models y stores en ExtJS para el módulo Fichas de Inspección	CODIFICACIÓN	TERMINADO		1h	4h	1h										
5	Creación de ventana y grid-panels en ExtJS para ver los procesos existentes y crear una ficha	CODIFICACIÓN	TERMINADO				3h	4h	1h								
6	Crear consulta SQL para obtener el listado de los procesos de la Dirección de Planificación	CODIFICACIÓN	TERMINADO						2h								
7	Programar métodos en ExtJS para visualizar la consulta del listado de procesos en un comboBox	CODIFICACIÓN	TERMINADO						1h	1h							
8	Crear consulta SQL para obtener el listado de variables según el proceso	CODIFICACIÓN	TERMINADO							3h							
9	Programar métodos en ExtJS para visualizar la consulta del listado de variables en el grid-panel	CODIFICACIÓN	TERMINADO								4h	2h					
10	Agregar dos botones en el grid de variables donde el uno es para crear una nueva y el segundo para agregar a la ficha una variable ya existente.	PROTOTIPADO	TERMINADO									2h					
11	Realizar ventana para crear nueva variable donde se escribe el nombre y descripción	PROTOTIPADO	TERMINADO										2h				
12	Programar métodos para enviar datos de la variable al servidor e insertarlos en las tablas respectivas de la base de datos	CODIFICACIÓN	TERMINADO											2h	1h		

ID	Tareas	Tipo	Estado	ENERO 2013												11	14	15
13	Crear ventana con un grid-panel de las variables ya existentes para agregar en la ficha	PROTOTIPADO	TERMINADO													3h		
14	Programar métodos para agregar una o más variables del grid a la ficha, enviar al servidor e insertar los datos en las tablas de la base de datos	CODIFICACIÓN	TERMINADO														4h	
15	Realizar pruebas creando una ficha en el módulo y verificar si los datos insertados en la tabla son correctos	PRUEBAS	TERMINADO															4h

Tabla 33: Pila de Iteración 2 – Historia de Usuario N°3

Fuente: Propia

PILA DE ITERACIÓN 2				Esfuerzo estimado					
Historia N°4: Crear categorías y tipos de variables en la ficha de inspección				Enero 2013					
ID	Tareas	Tipo	Estado	M	J	V	L	M	M
				16	17	18	21	22	23
1	Crear, modificar y agregar campos en las tablas para la creación de categorías y tipos de variables en una ficha de inspección	PROTOTIPADO	TERMINADO	3h					
2	Generar schema y model en Symfony con las nuevas tablas y campos agregados	CODIFICACIÓN	TERMINADO	1h					
3	Creación de models de las nuevas tablas en ExtJS	CODIFICACIÓN	TERMINADO		1h				
4	Realizar modificaciones en la ventana de fichas agregando otro grid para crear categorías	PROTOTIPADO	TERMINADO		3h	1h			
5	Crear consulta SQL para obtener el listado de las categorías según el proceso	CODIFICACIÓN	TERMINADO			2h			
6	Programar métodos en ExtJS para visualizar la consulta del listado de categorías en el grid-panel	CODIFICACIÓN	TERMINADO			1h	4h	1h	
7	Agregar dos botones en el grid de categorías donde el uno es para crear una nueva y el segundo para agregar a la ficha una categoría ya existente	PROTOTIPADO	TERMINADO					3h	
8	Realizar ventana para crear nueva categoría donde se escriba el nombre y una descripción	PROTOTIPADO	TERMINADO						4h

Continuación Pila de Iteración 2 - Historia N°4				Enero 2013						Febrero 2013						
ID	Tareas	Tipo	Estado	J	V	L	M	M	J	V	L	M	M	J	V	
				24	25	28	29	30	31	1	4	5	6	7	8	
9	Programar métodos para enviar datos de la categoría al servidor e insertarlos en las tablas respectivas de la base de datos	CODIFICACIÓN	TERMINADO	2h												
10	Crear ventana con un grid-panel de las categorías ya existentes para agregar en la ficha	PROTOTIPADO	TERMINADO	2h	2h											
11	Programar métodos para agregar una o más categorías del grid a la ficha, enviar al servidor e insertar los datos en las tablas de la base de datos	CODIFICACIÓN	TERMINADO		2h	4h										
12	Programar y modificar métodos o eventos ya existentes para crear o agregar una variable a una categoría según el proceso seleccionado	CODIFICACIÓN	TERMINADO				4h	2h								
13	Crear y modificar ventanas para tener la opción de crear 3 tipos de variables (Simple, Selección Única y Selección Múltiple) y que pertenezcan a una categoría	CODIFICACIÓN	TERMINADO					2h	4h							
14	Programar eventos en ExtJS para crear las opciones en las variables de tipo Selección Única y Múltiple	CODIFICACIÓN	TERMINADO							4h						
15	Programar eventos para enviar los datos de las variables y sus opciones al servidor e insertar los datos en las tablas respectivas de la base de datos	CODIFICACIÓN	TERMINADO								4h					
16	Realizar pruebas creando y agregando una categoría a un proceso; luego a este crear y agregar variables y verificar si los datos insertados son correctos	PRUEBAS	TERMINADO									4h				
17	Crear ventana en ExtJS para visualizar las opciones que tiene una variable de Selección Única y Múltiple	PROTOTIPADO	TERMINADO										4h	2h		
18	Crear consulta SQL para obtener las opciones según la id de variable	CODIFICACIÓN	TERMINADO											2h		
19	Programar métodos en ExtJS para visualizar la consulta SQL en la ventana de opciones	CODIFICACIÓN	TERMINADA													4h

Continuación Pila de Iteración 2 - Historia N°4				Febrero 2013												
ID	Tareas	Tipo	Estado	L	M	M	J	V	L	M	M	J	V	L	M	
				11	12	13	14	15	18	19	20	21	22	25	26	
20	Crear consultas SQL para realizar búsquedas de categorías y variables al momento de agregar una a la ficha	CODIFICACIÓN	TERMINADA	4h	2h											
21	Programar métodos para visualizar las búsquedas en las ventanas de agregar categorías y variables	CODIFICACIÓN	TERMINADA		2h	4h										
22	Realizar pruebas buscando, agregando categorías y variables con opciones para verificar si los datos insertados en las tablas son correctos	PRUEBAS	TERMINADA				4h									
23	Crear ventana para visualizar el diseño de la ficha según un proceso	PROTOTIPADO	TERMINADA					4h								
24	Crear consulta SQL para obtener los datos de los departamentos existentes en la institución GAD-I y sus procesos correspondientes	CODIFICACIÓN	TERMINADA						2h							
25	Programar métodos en ExtJS para visualizar la consulta de departamentos y procesos dentro de la ventana de ver ficha	CODIFICACIÓN	TERMINADA						2h							
26	Crear consulta SQL para obtener los datos de la ficha según el proceso seleccionado en la que se incluyen las categorías, variables y opciones.	CODIFICACIÓN	TERMINADA							4h						
27	Programar métodos en ExtJS para visualizar la ficha dentro de la ventana de ver ficha	CODIFICACIÓN	TERMINADA								4h	4h				
28	Programar métodos en Symfony y ExtJS para visualizar la ficha en un PDF	CODIFICACIÓN	TERMINADA										4h	4h		
29	Realizar pruebas creando una ficha completa y visualizar su diseño en el PDF	PRUEBAS	TERMINADA													4h

Tabla 34: Pila de Iteración 2 – Historia de Usuario N°4

Fuente: Propia

### 3.9.3 PILA DE ITERACIÓN 3

PILA DE ITERACIÓN 3				Esfuerzo estimado														
Historia N°5: Ordenar trámites para generar inspecciones				FEB 2013		MARZO 2013												
ID	Tareas	Tipo	Estado	M	J	V	L	M	M	J	V	L	M	M	J	V	L	
				27	28	1	4	5	6	7	8	11	12	13	14	15	18	
1	Creación de tablas para el registro de las agendas de inspecciones	PROTOTIPADO	TERMINADO	2h														
2	Generar schema y model en Symfony con las nuevas tablas	CODIFICACIÓN	TERMINADO	2h														
3	Creación del módulo Agenda de Inspecciones en Symfony y ExtJS	CODIFICACIÓN	TERMINADO		2h													
4	Creación del controlador, models y stores en ExtJS para el módulo Agenda de Inspecciones	CODIFICACIÓN	TERMINADO		2h	4h												
5	Creación de ventana en ExtJS para ver y ordenar los trámites que se han asignado inspección	PROTOTIPADO	TERMINADO				4h											
6	Crear consulta SQL para obtener las parroquias asignadas al arquitecto logueado y visualizar en un comboBox	CODIFICACIÓN	TERMINADO					2h										
7	Agregar numberfield para ingresar y determinar el número de inspecciones a realizarse	PROTOTIPADO	TERMINADO					2h										
8	Crear consulta SQL para obtener solo los trámites creados que necesitan inspección y visualizarlos en el grid-panel	CODIFICACIÓN	TERMINADO						2h									
9	Agregar conexión hacia la base de datos GIS del municipio en el archivo database.yml de Symfony	CODIFICACIÓN	TERMINADO						2h									
10	Crear control, ventana y layouts para ver el mapa en ExtJS	PROTOTIPADO	TERMINADO							4h	4h							
11	Programar métodos para consultar y visualizar la ubicación de los predios según la clave catastral de los trámites	CODIFICACIÓN	TERMINADO									4h	4h	4h	4h	4h	4h	

Continuación Pila de Iteración 3 - Historia N°5				Marzo 2013							Abril 2013							
ID	Tareas	Tipo	Estado	M	M	J	V	L	M	M	J	V	L	M	M	J	V	L
				19	20	21	22	25	26	27	28	29	1	2	3	4	5	8
12	Agregar boton para poder activar la creación de la ruta de inspecciones y programar métodos para elegir orden de predios (Al dar click en cada una de las ubicaciones)	PROTOTIPADO	TERMINADO	4h	4h	4h	4h											
13	Agregar boton y programar métodos para generar inspecciones	PROTOTIPADO	TERMINADO					4h	4h									
14	Programar métodos para identificar si el día de inspección es igual al día actual para que sea el usuario el que elige que día va ser la inspección	CODIFICACIÓN	TERMINADO							4h	4h							
15	Crear ventana para visualizar en un PDF las inspecciones generadas en la agenda incluido la foto y mapa respectivo del trámite	PROTOTIPADO	TERMINADO									4h	4h					
16	Crear métodos para consultar los datos del trámite, predio, propietario, peticionario, foto, mapa y presentar en el PDF	CODIFICACIÓN	TERMINADO										4h	4h	4h	4h		
17	Realizar pruebas creando trámites con inspección y luego ordenarlos para generar una agenda de inspecciones visualizando los datos en el PDF	PRUEBAS	TERMINADO															4h

Tabla 35: Pila de Iteración 3 – Historia de Usuario N°5

Fuente: Propia

PILA DE ITERACIÓN 3				Esfuerzo estimado															
Historia N°6: Ver información del trámite				Abril 2013															
ID	Tareas	Tipo	Estado	M	M	J	V	L	M	M	J	V	L	M	M	J	V	L	M
				09	10	11	12	15	16	17	18	19	22	23	24	25	26	29	30
1	Crear un control genérico en ExtJS para ver los datos de un trámite en cualquier módulo del sistema	CODIFICACIÓN	TERMINADO	4h	4h														
2	Crear controlador, models, stores y views del control genérico llamado datos trámite	CODIFICACIÓN	TERMINADO			4h													
3	Crear consultas SQL para obtener los datos del trámite	CODIFICACIÓN	TERMINADO				2h	2h											

ID	Tareas	Tipo	Estado	Abr 2013	12	15	16	17	18	19	22	23	24	25	26	29	30
4	Crear consulta SQL para obtener los datos y foto del predio	CODIFICACIÓN	TERMINADO			2h	2h	2h									
5	Agregar tab-panel para ver datos del trámite y predio	PROTOTIPADO	TERMINADO				2h										
6	Programar métodos para visualizar las consultas del trámite y predio en la ventana del control datos trámite	CODIFICACIÓN	TERMINADO					4h	2h								
7	Agregar dos botones en la ventana, uno para ver los archivos adjuntos y otro para ver el mapa del predio	PROTOTIPADO	TERMINADO						2h								
8	Crear consulta SQL para obtener los archivos adjuntos según el número del trámite	CODIFICACIÓN	TERMINADO							4h							
9	Crear grid-panel y programar métodos para mostrar el listado de los archivos adjuntos al dar click en el boton "Ver Archivos Adjuntos"	CODIFICACIÓN	TERMINADO								4h	4h					
10	Crear métodos para visualizar el mapa dentro del control de datos de trámite al dar click en el boton "Ver Mapa"	CODIFICACIÓN	TERMINADO										4h	4h	4h	2h	
11	Realizar pruebas creando trámites con inspecciones y verificar con un trámite del módulo Agenda de Inspecciones si los datos, foto y mapa son correctos.	PRUEBAS	TERMINADO													2h	4h

**Tabla 36: Pila de Iteración 3 – Historia de Usuario N°6**

**Fuente: Propia**

<b>PILA DE ITERACIÓN 3</b>				<b>Esfuerzo estimado</b>									
<b>Historia N°7: Enviar e-mail al ciudadano informando el día de inspección</b>				<b>Mayo 2013</b>									
ID	Tareas	Tipo	Estado	M	J	V	L	M	M	J	V	L	M
				01	02	03	06	07	08	09	10	13	14
1	Consultar como enviar e-mails con PHP y Symfony	ANÁLISIS	TERMINADO	4h	4h	4h							
2	Usar sfMailer y configurarlo en el archivo factories.yml	CODIFICACIÓN	TERMINADO				4h	4h					
3	Agregar dentro del método creación de Agenda de Inspecciones el código con el formato y envió de email	CODIFICACIÓN	TERMINADO						4h	4h	4h		
4	Realizar pruebas creando inspecciones y verificar si ha llegado el email enviado a una dirección electrónica	PRUEBAS	TERMINADO									4h	4h

**Tabla 37: Pila de Iteración 3 – Historia de Usuario N°7**

**Fuente: Propia**

### 3.9.4 PILA DE ITERACIÓN 4

PILA DE ITERACIÓN 4				Esfuerzo estimado									
Historia N°8: Registro de Inspecciones				Mayo 2013									
ID	Tareas	Tipo	Estado	M	J	V	L	M	M	J	V	L	M
				15	16	17	20	21	22	23	24	27	28
1	Creación de tablas para el registro de las inspecciones	PROTOTIPADO	TERMINADO	1h									
2	Generar schema y model en Symfony con las nuevas tablas	CODIFICACIÓN	TERMINADO	1h									
3	Creación del módulo Registro de Inspecciones en Symfony y ExtJS	CODIFICACIÓN	TERMINADO	2h	1h								
4	Creación del controlador, models y stores para el módulo Registro de Inspecciones	CODIFICACIÓN	TERMINADO		3h								
5	Creación de ventana, paneles y grid-panels en ExtJS para ver en maestro-detalle las agendas y sus inspecciones correspondientes generadas	PROTOTIPADO	TERMINADO			4h	2h						
6	Crear consulta SQL para obtener las agendas creadas y trámites con inspecciones correspondientes	CODIFICACIÓN	TERMINADO				2h	2h					
7	Programar métodos y eventos para presentar los datos de las consultas de agendas e inspecciones en forma de maestro-detalle	CODIFICACIÓN	TERMINADO					2h	4h	2h			
8	Agregar al grid de inspecciones (detalle) una columna con un boton para poder ver la información del trámite	CODIFICACIÓN	TERMINADO							2h			
9	Crear ventana para ingresar los datos y registro de la inspección donde además se puede registra la ficha y adjuntar fotos	PROTOTIPADO	TERMINADO								4h	4h	
10	Crear ventana para seleccionar cual fue el código de movilización para realizar la inspección	PROTOTIPADO	TERMINADO										2h
11	Crear consulta SQL para obtener los datos de movilización y además filtrar por <i>cod_movilizacion</i> y <i>cod_vehiculo</i>	CODIFICACIÓN	TERMINADO										2h



Continuación Pila de Iteración 4 - Historia N° 8				Mayo 2013			Junio 2013											
ID	Tareas	Tipo	Estado	M	J	V	L	M	M	J	V	L	M	M	J	V	L	
				29	30	31	3	4	5	6	7	10	11	12	13	14	17	
12	Programar métodos y eventos en ExtJS para presentar los datos de movilización y seleccionar uno del listado para agregar al registro de inspección	CODIFICACIÓN	TERMINADO	4h														
13	Crear consulta SQL para obtener la ficha respectiva según el proceso del trámite seleccionado	CODIFICACIÓN	TERMINADO		4h													
14	Crear ventana y programar métodos para mostrar la ficha respectiva	CODIFICACIÓN	TERMINADO			4h	4h	4h										
15	Programar métodos y eventos en Symfony y ExtJS para registrar los datos de la ficha, enviar al servidor e insertarlos en una variable de sesión temporal	CODIFICACIÓN	TERMINADO						4h	4h								
16	Crear ventana para adjuntar fotos tomadas en la inspección y escribir una descripción de la misma	PROTOTIPADO	TERMINADO								4h	4h						
17	Programar métodos y eventos en Symfony y ExtJS para adjuntar fotos y enviar al servidor con su descripción e insertarlos en una variable de sesión temporal	CODIFICACIÓN	TERMINADO										4h	4h				
18	Programar métodos y eventos para guardar todo el registro de la inspección donde consta la ficha, fotos, movilización, observaciones y si es o no favorable	CODIFICACIÓN	TERMINADO												4h	4h		
19	Realizar pruebas generando una agenda de inspección en el otro módulo y luego ver y registrar en este los datos de la inspección que incluya ficha y fotos	PRUEBAS	TERMINADO															4h

Tabla 38: Pila de Iteración 4 – Historia de Usuario N°8

Fuente: Propia

PILA DE ITERACIÓN 4				Esfuerzo estimado												
Historia N°9: Recortar fotos de inspección				Junio 2013								Julio 2013				
ID	Tareas	Tipo	Estado	M	M	J	V	L	M	M	J	V	L	M	M	J
				18	19	20	21	24	25	26	27	28	01	02	03	04
1	Crear ventana para ver y editar la imagen cargada	PROTOTIPADO	TERMINADO	4h	4h											
2	Programar métodos y eventos para cargar la imagen en la ventana de editar	CODIFICACIÓN	TERMINADO			4h	4h									
3	Consultar como recortar una imagen con PHP y JavaScript	ANALISIS	TERMINADO					4h	4h	4h						
4	Usar librería jQuery y programar métodos con PHP para realizar el recorte de la imagen	CODIFICACIÓN	TERMINADO								4h	4h	4h			
5	Programar métodos para reemplazar la imagen anterior por la nueva recortada	CODIFICACIÓN	TERMINADO											4h	4h	
6	Realizar pruebas adjuntando una imagen de inspección y luego recortarla	PRUEBAS	TERMINADO													4h

Tabla 39: Pila de Iteración 4 – Historia de Usuario N°9

Fuente: Propia

PILA DE ITERACIÓN 4				Esfuerzo estimado													
Historia N°10: Consultar las inspecciones realizadas				Julio 2013													
ID	Tareas	Tipo	Estado	V	L	M	M	J	V	L	M	M	J	V	L	M	M
				05	08	09	10	11	12	15	16	17	18	19	22	23	24
1	Crear controlador, stores y models de las tablas y funciones necesarias	PROTOTIPADO	TERMINADO	4h													
2	Crear ventana con el diseño para consultar las inspecciones realizadas	PROTOTIPADO	TERMINADO		4h	4h	4h										
3	Crear consultas SQL para obtener el listado de las agendas con sus respectivas inspecciones registradas	CODIFICACIÓN	TERMINADO					4h									

ID	Tareas	Tipo	Estado	Julio 2013							12	15	16	17	18	19	22	23	24
4	Crear consultas SQL para realizar filtros de búsquedas por número de trámite, fechas de registro y si las inspecciones son o no aprobadas	CODIFICACIÓN	TERMINADO							4h									
5	Programar métodos y eventos para visualizar las consultas con sus filtros	CODIFICACIÓN	TERMINADO								4h	4h							
6	Crear consultas SQL para obtener los datos de las inspecciones incluyendo las fotos y ficha registrada	CODIFICACIÓN	TERMINADO										4h	4h					
7	Programar métodos y eventos para visualizar las consultas de fotos y ficha	CODIFICACIÓN	TERMINADO												4h	4h	4h		
8	Realizar pruebas registrando una inspección y visualizar los datos en la ventana de consulta de inspecciones realizadas usando sus filtros	PRUEBAS	TERMINADO																4h

Tabla 40: Pila de Iteración 4 – Historia de Usuario N°10

Fuente: Propia

PILA DE ITERACIÓN 4				Esfuerzo estimado													
Historia N°11: Registrar el Personal responsable de la Inspección				Julio 2013					Agosto 2013								
ID	Tareas	Tipo	Estado	J	V	L	M	M	J	V	L	M	M	J	V	L	
				25	26	29	30	31	01	02	05	06	07	08	09	12	
1	Creación de tabla para el registro del personal responsable	PROTOTIPADO	TERMINADO	4h													
2	Agregar grid en la ventana Registro de Inspección con un boton para agregar personal del GAD-I	PROTOTIPADO	TERMINADO		4h												
3	Crear ventana donde se presente el listado de empleados del GAD-I y se pueda filtrar la búsqueda por cédula y nombres	PROTOTIPADO	TERMINADO			4h	4h										
4	Crear consultas SQL para obtener los empleados del GAD-I con y sin filtros	CODIFICACIÓN	TERMINADO					4h									
5	Programar el método para agregar un empleado como parte del personal responsable de inspección	CODIFICACIÓN	TERMINADO						4h	4h							

ID	Tarea	Tipo	Estado	Agosto 2013							05	06	07	08	09	12
6	En el grid de personal responsable agregar y programar una columna de radiobutton para poder seleccionar cuál de ellos es el responsable del grupo	CODIFICACIÓN	TERMINADO								4h	4h	4h			
7	En el grid agregar y programar otra columna de acción para poder quitar una de las personas agregadas	CODIFICACIÓN	TERMINADO											4h		
8	Agregar al boton de guardar registro de inspección el código de envío de los datos de las personas responsables para registrar la información en la BDD	CODIFICACIÓN	TERMINADO												4h	
9	Realizar pruebas registrando una inspección y agregando el personal responsable	PRUEBAS	TERMINADO													4h

**Tabla 41: Pila de Iteración 4 – Historia de Usuario N°11**

**Fuente: Propia**

PILA DE ITERACIÓN 4				Esfuerzo estimado											
Historia N°12: Ver Informe de Inspección Realizada				Agosto 2013											
ID	Tareas	Tipo	Estado	M	M	J	V	L	M	M	J	V	L		
				13	14	15	16	19	20	21	22	23	26		
1	Agregar al grid de Inspecciones Registradas una columna de acción para poder abrir el informe con datos de la inspección	PROTOTIPADO	TERMINADO	4h	4h										
2	Crear una consulta SQL para obtener todos los datos de la inspección registrada	CODIFICACIÓN	TERMINADO			4h	4h								
3	Crear el diseño del informe usando el plugin TCPDF	PROTOTIPADO	TERMINADO					4h	4h						
4	Programar un método en Symfony y ExtJS para visualizar los datos de la inspección en un archivo PDF	CODIFICACIÓN	TERMINADO								4h	4h	4h		
5	Realizar pruebas registrando una inspección con todos los datos posibles y visualizar en la ventana de inspecciones registradas el informe en PDF	PRUEBAS	TERMINADO											4h	

**Tabla 42: Pila de Iteración 4 – Historia de Usuario N°12**

**Fuente: Propia**

### 3.9.5 PILA DE ITERACION 5

PILA DE ITERACIÓN 5				Esfuerzo estimado															
Historia N°13: Registro de Inspecciones en Dispositivo Móvil				Agosto 2013					Septiembre 2013										
ID	Tareas	Tipo	Estado	M	M	J	V	L	M	M	J	V	L	M	M	J	V	L	
				27	28	29	30	02	03	04	05	06	09	10	11	12	13	16	
1	Estudio e investigación del framework Sencha Touch 2 para aplicaciones móviles	ANALISIS	TERMINADO	8h	8h	8h	2h	2h											
2	Crear aplicación de Inspecciones en Symfony y SenchaTouch	PROTOTIPADO	TERMINADO				6h	6h	6h	2h									
3	Agregar conexión hacia la base de datos de imágenes en el archivo schema.yml	CODIFICACIÓN	TERMINADO							1h									
4	Crear módulo para el registro de inspecciones	PROTOTIPADO	TERMINADO							3h									
5	Crear controlador, models y stores para las vistas de Sencha Touch	CODIFICACIÓN	TERMINADO							2h									
6	Crear menús y diseño del sistema en sencha Touch	PROTOTIPADO	TERMINADO								2h	3h							
7	Crear una lista donde muestre las parroquias asignadas al usuario logueado	CODIFICACIÓN	TERMINADO								2h	2h							
8	Crear una lista donde al escoger una parroquia muestre las agendas creadas para realizar inspecciones	CODIFICACIÓN	TERMINADO								2h	2h							
9	Crear una lista donde al escoger una agenda se muestre los trámites pertenecientes a la misma	CODIFICACIÓN	TERMINADO								2h								
10	Revisar y programar como mostrar la foto del predio en Sencha Touch	CODIFICACIÓN	TERMINADO									1h	5h						
11	Revisar y programar como ver el mapa del predio en Sencha Touch	CODIFICACIÓN	TERMINADO										3h	5h	2h				
12	Crear una vista donde se muestre una pequeña información del predio con foto y ubicación en el mapa.	CODIFICACIÓN	TERMINADO											2h	3h				

ID	Tareas	Tipo	Estado	Septiembre 2013											10	11	12	13	16			
13	Revisar y programar como cargar una foto en la aplicación desde el celular	CODIFICACIÓN	TERMINADO														1h	2h	2h			
14	Crear una vista donde se pueda agregar y cargar las fotos obtenidas en el registro de inspección y guardarlas	CODIFICACIÓN	TERMINADO															1h	2h			
15	Crear vista donde se visualice la ficha de inspección y crear el código para guardar sus datos	CODIFICACIÓN	TERMINADO																	4h	3h	
16	Programar métodos y eventos para registrar la inspección enviando los datos al servidor e insertarlos en las tablas de la base de datos	CODIFICACIÓN	TERMINADO																		5h	
17	Realizar pruebas haciendo el registro completo de la inspección desde un dispositivo móvil	PRUEBAS	TERMINADO																			2h

Tabla 43: Pila de Iteración 5 – Historia de Usuario N°13

Fuente: Propia

### 3.9.6 PILA DE ITERACION 6

PILA DE ITERACIÓN 6				Esfuerzo estimado																
Historia N°14: Reportes con Gráficos Estadísticos				Septiembre 2013											Octubre 2013					
ID	Tareas	Tipo	Estado	M	M	J	V	L	M	M	J	V	L	M	M	J	V	L	M	M
				17	18	19	20	23	24	25	26	27	30	1	2	3	4	7	8	9
1	Creación del módulo Reportes de Inspecciones en Symfony y ExtJS	PROTOTIPADO	TERMINADO	4h																
2	Crear controlador, models y stores en ExtJS para el módulo Reportes de Inspecciones	PROTOTIPADO	TERMINADO		4h															
3	Crear ventanas para ver los reportes de inspecciones realizadas, por parroquias y procesos.	PROTOTIPADO	TERMINADO			4h	8h	5h												
4	Crear consultas SQL para obtener el total de inspecciones realizadas, por parroquias y procesos, además de poder filtrar las mismas por fechas	CODIFICACIÓN	TERMINADO					3h	4h											

Continuación Pila de Iteración 6 – Historia N°14				Septiembre 2013										Octubre 2013						
ID	Tareas	Tipo	Estado	M	M	J	V	L	M	M	J	V	L	M	M	J	V	L	M	M
				17	18	19	20	23	24	25	26	27	30	1	2	3	4	7	8	9
5	Programar métodos y eventos para poder visualizar los datos de las consultas en gráficos estadísticos	CODIFICACIÓN	TERMINADO							4h	4h	4h	8h							
6	Consultar e investigar el uso de ireport desde PHP	ANÁLISIS	TERMINADO											4h	4h					
7	Crear reportes en ireport con las consultas SQL	CODIFICACIÓN	TERMINADO													4h				
8	Programar métodos para presentar los reportes PDF en ExtJS y Symfony	CODIFICACIÓN	TERMINADO													4h	6h			
9	Realizar pruebas visualizando los reportes totales y filtrando por fechas	PRUEBAS	TERMINADO															3h	4h	
10	Entrega del Sistema de Control de Procesos de Inspección	REUNIÓN	TERMINADO																	3h

**Tabla 44: Pila de Iteración 6 – Historia de Usuario N°14**

**Fuente: Propia**

### **3.10. DESARROLLO DE HISTORIAS DE USUARIO POR ITERACIÓN**

La elaboración de las Historias de Usuario y a su vez las pilas de iteración son la parte fundamental dentro de la metodología para el desarrollo del sistema ya que en ellos se describen cada una de las tareas o actividades realizadas para la creación del Sistema de Control de Procesos de Inspecciones.

A continuación se presenta detalladamente las iteraciones con sus respectivas tareas realizadas para el desarrollo de los módulos del sistema.

#### **3.10.1 Iteración 1 (Módulo Horarios de Inspecciones)**

En esta iteración perteneciente a la fase inicial del desarrollo del sistema se describen las 2 historias de usuario pertenecientes al Módulo de Horarios de Inspección donde entre otras tareas consta el inicio del desarrollo del sistema, estudio de herramientas y obtención de requisitos.

##### **❖ Historia N°1: Crear Horarios de Inspección**

En esta historia de usuario las tareas iniciales desde la 1 a 10 describen la capacitación y estudio de los frameworks Symfony y ExtJS; además del análisis de la base de datos del Municipio y que tablas están relacionadas o son usadas para el desarrollo del Sistema de Inspecciones.

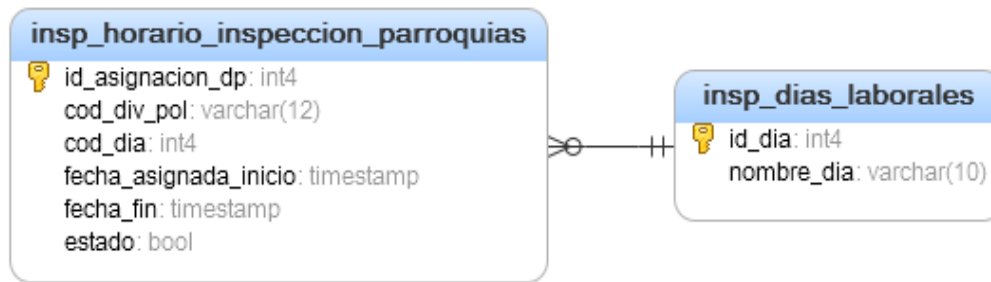
También dentro de estas tareas se realiza el estudio del proceso de inspecciones mediante el levantamiento del mismo y además haciendo uso de la elaboración y ejecución de cuestionarios dirigidos a los Arquitectos de la Dirección de Planificación para la aclaración y obtención de requerimientos.

##### **Tareas:**

- 11.** Elaboración y Diseño de la base de datos inicial para inspecciones.
- 12.** Creación de la app Control de Inspecciones en Symfony dentro del proyecto integrado
- 13.** Creación del módulo Horarios de Inspecciones.
- 14.** Registro en la BDD la información del sistema, módulos, menús, submenús, permisos.

En la **Figura 20** se visualiza el diseño inicial de las tablas de la base de datos en las cuales se registran los días asignados a las parroquias para su posterior creación y uso del horario de inspecciones.

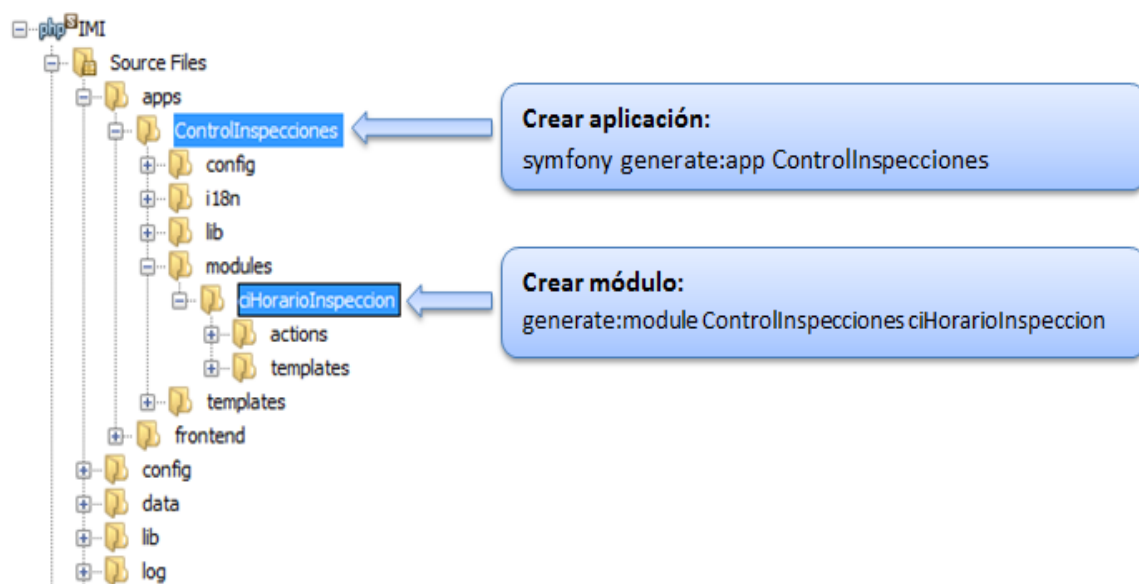




**Figura 20: Diseño inicial de la base de datos - Tablas para la creación de Horarios**

**Fuente: Propia**

En la siguiente imagen se muestra la estructura de carpetas del proyecto principal en Symfony y la respectiva creación de la aplicación **Control de Inspecciones** y el módulo **Horario de Inspección**.

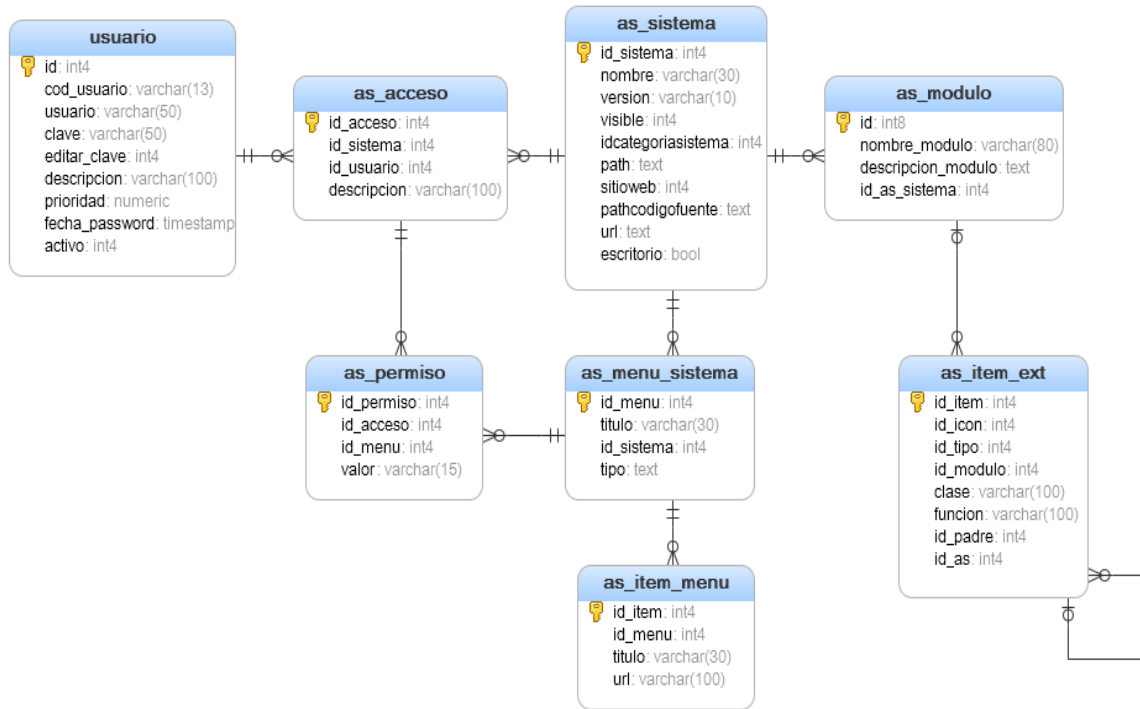


**Figura 21: Creación de la aplicación de inspecciones y módulo de horarios**

**Fuente: Propia**

Para el registro de los datos del sistema, módulos, menús, submenús y permisos de usuarios se hace uso de las tablas existentes en la base de datos IMI dentro del esquema *adm*.

A continuación se visualiza en la primera imagen (*Figura 22*) el diagrama de las tablas usadas, en la siguiente imagen (*Figura 23*) está el diseño del escritorio del sistema integrado junto con las aplicaciones o sistemas permitidos y en la última imagen (*Figura 24*) se observan los menús y submenús de la aplicación control de inspecciones.



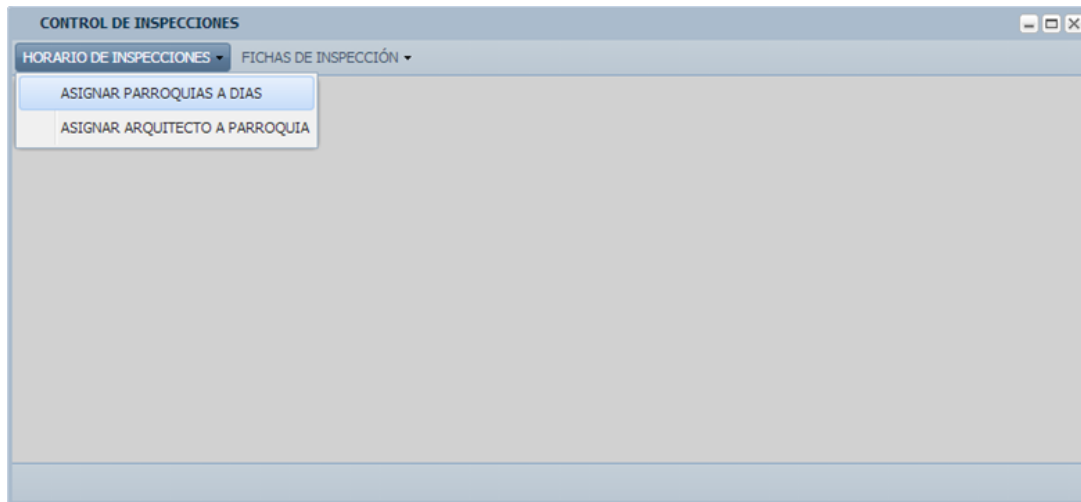
**Figura 22: Diagrama Entidad Relación de las tablas usadas para permisos y accesos**

**Fuente: Propia**



**Figura 23: Escritorio del Core del Sistema Integrado y sus Aplicaciones**

**Fuente: Propia**

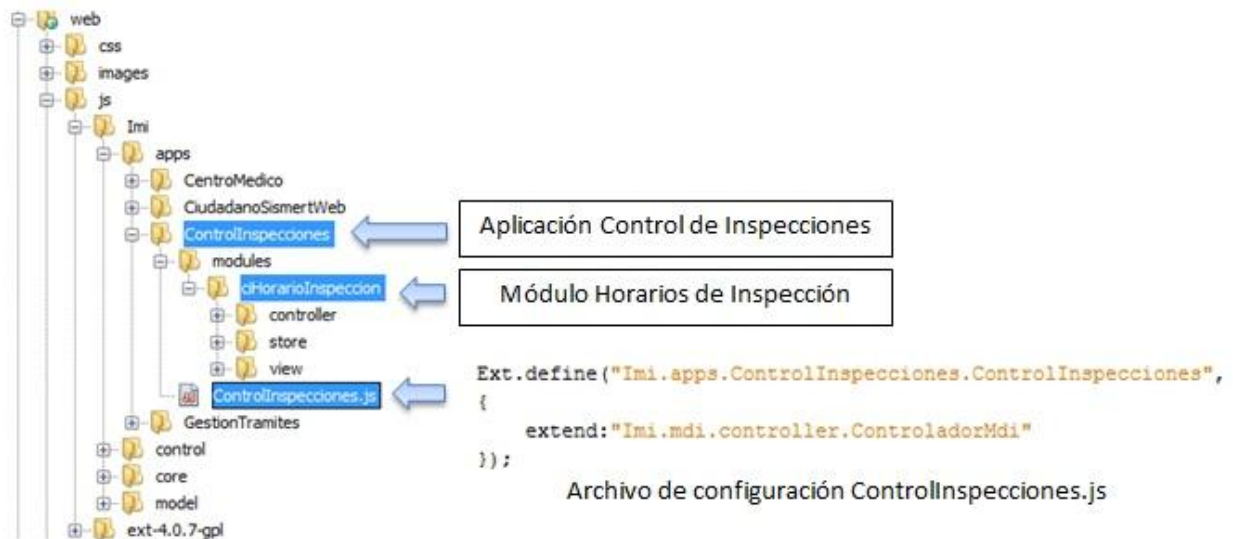


**Figura 24: Interfaz ventana del Sistema Control de Inspecciones y sus menús**

**Fuente: Propia**

**Tareas:**

15. Creación de la aplicación Control de Inspecciones en EXTJS4.
16. Creación del módulo Horarios de Inspecciones en ExtJS.
17. Creación de controlador, models y stores para el módulo Horarios Inspección.



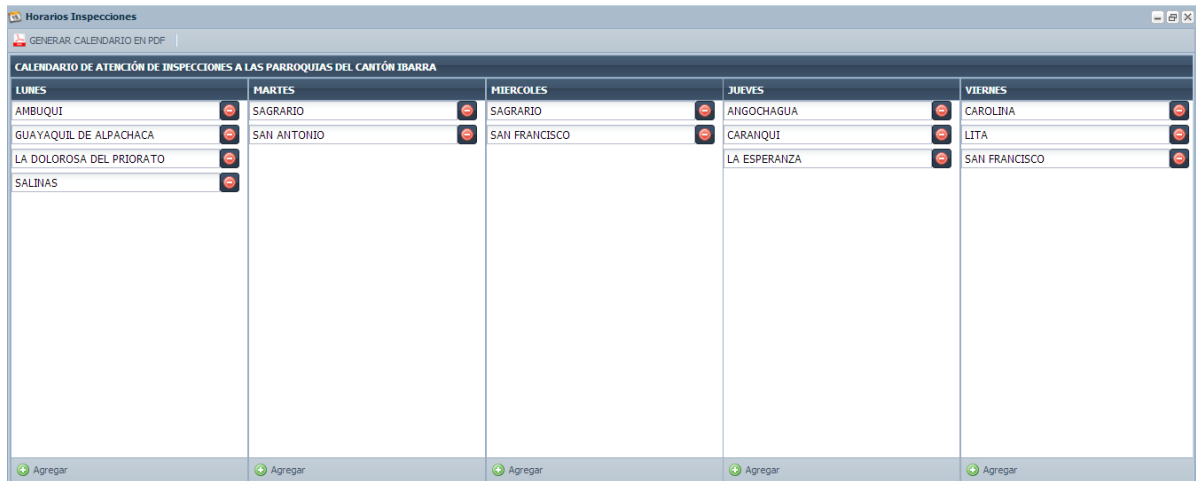
**Figura 25: Esquema de la aplicación de inspecciones y módulo de horarios en ExtJS**

**Fuente: Propia**

**Tareas:**

18. Creación de ventana para asignar parroquias a días.
19. Crear paneles para cada día de Lunes a Viernes.

20. Crear consulta para obtener las parroquias del cantón Ibarra.
21. Programar los métodos para asignar una parroquia a un día.
22. Programar los métodos en Symfony para insertar las asignaciones en tablas de la BDD.



**Figura 26: Ventana y paneles para la asignación de parroquias a días**

**Fuente: Propia**

Al dar click en el boton Agregar que hay debajo de cada panel de los días se muestra una ventana con el listado de las parroquias para asignar a uno de estos días.



**Figura 27: Ventana para asignar una parroquia a un día del horario**

**Fuente: Propia**

### Tareas:

23. Agregar el boton para visualizar el calendario de inspecciones.
24. Agregar, configurar y usar el puglin TCPDF en Symfony.
25. Crear consulta para obtener el horario de inspección actual.
26. Programar los métodos para visualizar el horario en PDF.
27. Realizar pruebas ingresando datos y visualizar el PDF para imprimir.

Para la creación y visualización del horario de inspecciones en PDF se lo realizó mediante el uso del Plugin TCPDF el cual luego de descargar se lo copia dentro de la ruta IMI/plugins, también se copia el archivo **pdf\_configs.yml** en la ruta IMI/apps/ControlInspecciones/config y por último se configura el uso del plugin en el archivo **ProjectConfiguration.class.php** que está en IMI/config.



**Figura 28: Uso y configuración del plugin TCPDF**

**Fuente: Propia**

Hora	LUNES	MARTES	MIERCOLES	JUEVES	VIERNES
8H00 HASTA 17H30	ALPACHACA Arq. Jenny Muenala		SAN FRANCISCO Arq. Cumanda Iturralde	CARANQUI Arq. Oscar Sanipatin	CAROLINA Arq. Cumanda Iturralde Arq. Jenny Muenala
	PRIORATO Arq. Jenny Muenala				LITA Arq. Jenny Muenala SAN FRANCISCO Arq. Cumanda Iturralde
	INSPECCIONES APROBACION PLANOS LEVANTAMIENTOS CERRAMIENTOS	INSPECCIONES APROBACION PLANOS LEVANTAMIENTOS CERRAMIENTOS	INSPECCIONES APROBACION PLANOS LEVANTAMIENTOS CERRAMIENTOS	INSPECCIONES APROBACION PLANOS LEVANTAMIENTOS CERRAMIENTOS	INSPECCIONES APROBACION PLANOS LEVANTAMIENTOS CERRAMIENTOS

**Figura 29: Archivo PDF visualizando el horario de inspecciones actual**

**Fuente: Propia**

### ❖ Historia N°2: Asignación de arquitectos a parroquias

En esta historia se especifica la creación de un módulo ArquitectoParroquia lo cual se lo hace para tener una mejor estructura del código pero en realidad este tiene relación y forma parte del módulo Horarios de Inspección dado que aquí es donde se asigna una o más parroquias a un arquitecto.

## Tareas:

1. Creación y uso de la tabla para asignación de arquitectos a parroquias.
2. Creación del módulo ArquitectoParroquia en Symfony y ExtJS.
3. Creación del controlador para el módulo ArquitectoParroquia.
4. Creación de models para las tablas.
5. Creación de stores en el módulo ArquitectoParroquia.

En la siguiente imagen se puede visualizar la estructura de las tablas para el registro y asignación de un Arquitecto a una parroquia donde la tabla *parroquia\_planificacion* tiene relación con 2 tablas ya existentes en la Base de Datos IMI, la primera es la tabla *empleado* para obtener los datos del Arquitecto y la segunda es la tabla *div\_pol* para consultar los datos de la parroquia.



Figura 30: Diagrama de las tablas usadas para asignación de parroquias-arquitecto

Fuente: Propia

En la Figura 31 se visualiza el comando en Symfony para la creación del módulo **Arquitecto Parroquia** dentro de la aplicación de inspecciones y en la Figura 32 se muestra la creación del mismo módulo dentro del proyecto ExtJS donde se encuentran las pantallas del cliente.

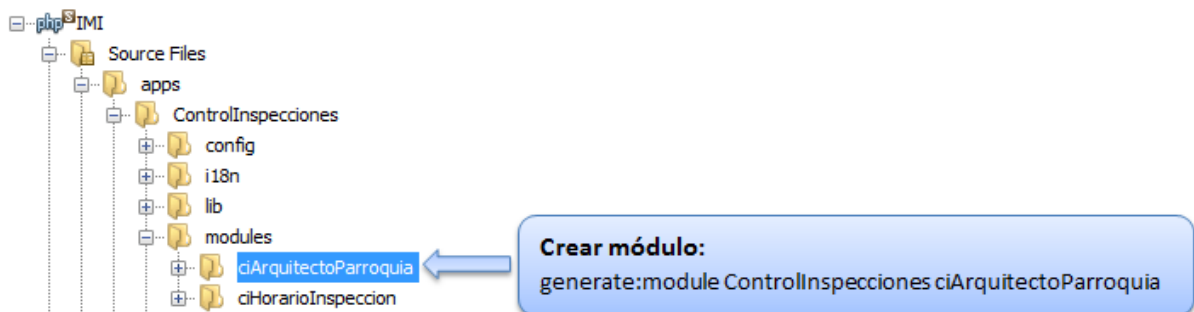
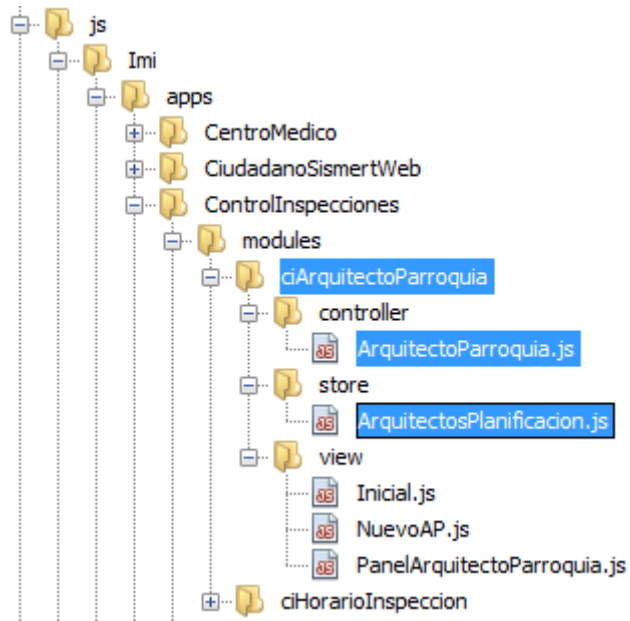


Figura 31: Creación de módulo Arquitecto Parroquia en Symfony

Fuente: Propia

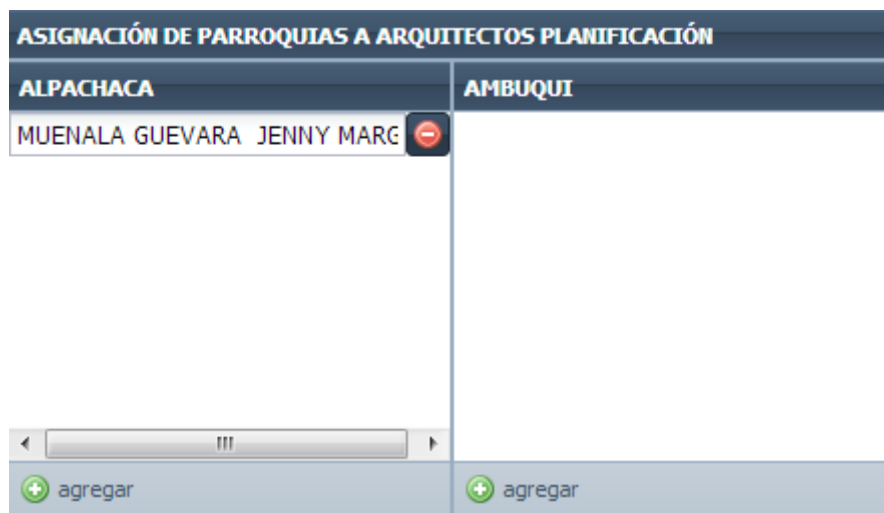


**Figura 32: Creación y estructura del módulo Arquitecto Parroquia en ExtJS**

**Fuente: Propia**

**Tareas:**

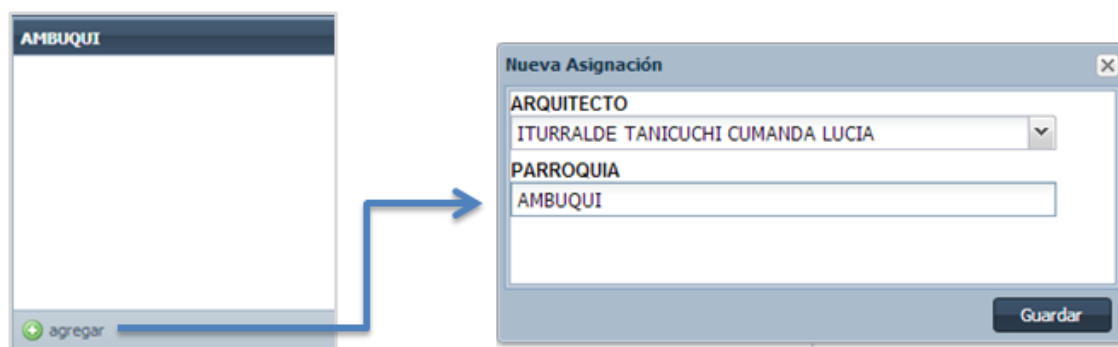
6. Creación de ventana y paneles para asignar arquitectos a parroquias.
7. Crear consulta en Symfony para obtener los arquitectos que pertenecen a la dirección de planificación.
8. Programar los métodos en ExtJS para asignar un arquitecto a una parroquia.
9. Programar los métodos en Symfony para insertar las asignaciones en las tablas de la BDD.



**Figura 33: Parte de la ventana y paneles del módulo arquitecto-parroquia**

**Fuente: Propia**

Al dar click en el boton agregar que está debajo de cada uno de los paneles de parroquias se abre la ventana para realizar una nueva asignación de un arquitecto con dicha parroquia.



**Figura 34: Ventana para asignar un arquitecto a una parroquia**

**Fuente: Propia**

Por último se agrega al método de generar el horario de inspecciones en PDF el código para poder visualizar los arquitectos asignados a cada una de las parroquias del cantón, el resultado y visualización de este archivo se lo muestra anteriormente en la Figura 29.

### **3.10.2 Iteración 2 (Módulo Fichas de Inspección)**

En esta iteración se describen las historias de usuario pertenecientes al Módulo Fichas de Inspección la cual se creó luego de haber comprobado mediante el uso de cuestionarios la no existencia de una ficha por proceso o tipo de trámite.

#### **❖ Historia N°3: Crear Ficha de Inspección**

En esta historia se describen las tareas para que un usuario administrador pueda crear una nueva ficha de inspección según el proceso las cuales contienen datos o variables.

#### **Tareas:**

1. Crear tablas en la BDD para la creación de fichas de inspección.
2. Generar schema y model en Symfony con las nuevas tablas.
3. Creación del módulo Fichas de Inspecciones en Symfony y ExtJS.
4. Creación del controlador, models y stores en ExtJS para el módulo Fichas de Inspección.



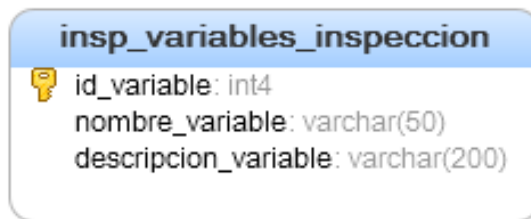


Figura 35: Tabla inicial de las variables o datos de una ficha

Fuente: Propia

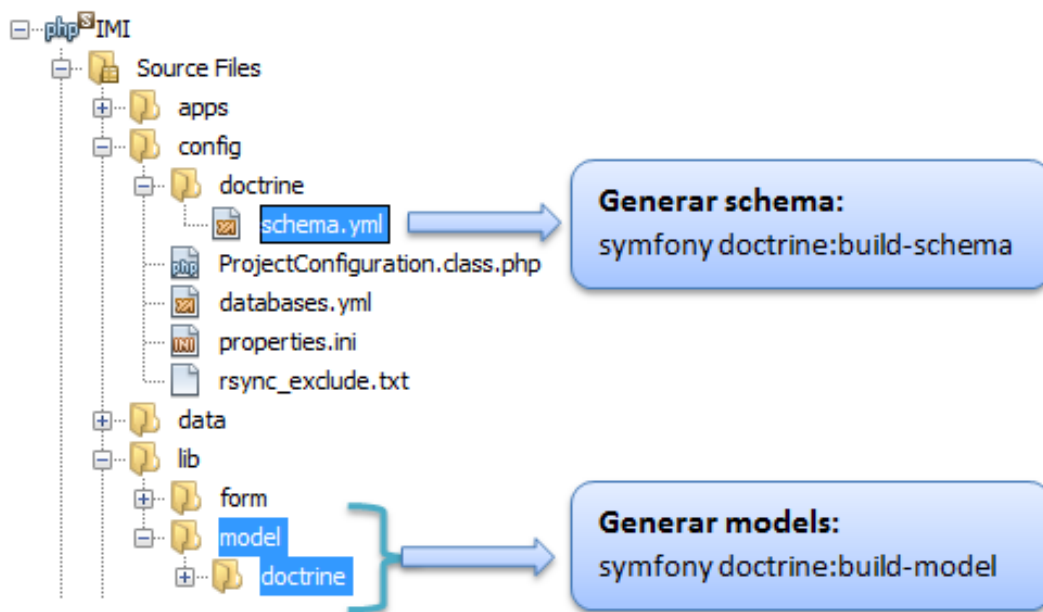


Figura 36: Generar schema y model en Symfony

Fuente: Propia

Luego se crea el módulo Fichas de Inspección en Symfony y ExtJS de la misma manera y estructura como se crearon los anteriores módulos en el cual consta el controlador, models y stores en el lado del cliente.

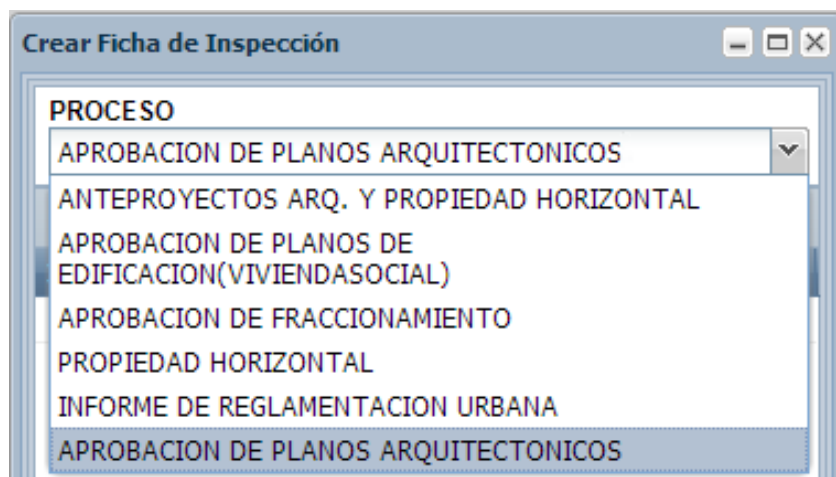
#### Tareas:

5. Creación de ventana y grid-panels en ExtJS para ver los procesos y crear una ficha.
6. Crear consulta SQL para obtener el listado de los procesos de la Dirección de Planificación.
7. Programar métodos en ExtJS para visualizar la consulta del listado de procesos en un comboBox.



**Figura 37: Ventana inicial del módulo Ficha de Inspecciones**

**Fuente: Propia**



**Figura 38: Listado de procesos mostrados en un ComboBox**

**Fuente: Propia**

**Tareas:**

8. Crear consulta SQL para obtener el listado de variables según el proceso.
9. Programar métodos en ExtJS para visualizar la consulta del listado de variables en el grid-panel.
10. Agregar dos botones en el grid de variables donde el uno es para crear una nueva y el segundo para agregar a la ficha una variable ya existente.

Id Variable	Nombre Variable	Descripción Variable	Quitar
83	SERVICIOS BASICOS	SERVICIOS BASICOS	-
89	AREA DE PARQUERO		-
90	PARQUEADERO		-

Mostrando 1 - 3 de 3

**Figura 39: Grid-Panel con el listado de variables**

**Fuente: Propia**

**Tareas:**

11. Realizar ventana para crear nueva variable donde se escribe el nombre y descripción.
12. Programar métodos para enviar datos de la variable al servidor e insertarlos en las tablas respectivas de la base de datos.
13. Crear ventana con un grid-panel de las variables ya existentes para agregar en la ficha.
14. Programar métodos para agregar una o más variables del grid a la ficha, enviar al servidor e insertar los datos en las tablas de la base de datos.

En la siguiente imagen se muestra la ventana realizada para la creación de nuevas variables simples donde es obligatorio ingresar el nombre de la variable y tipo de dato el cual puede ser texto o numérico; la descripción es opcional pero es recomendable registrar para tener una definición de las variables que se van a crear.

**Crear Variable**

**Datos**

Proceso: ANTEPROYECTOS ARQ. Y PROPIEDAD HORIZC

Nombre Variable:\*

Tipo de dato:\*

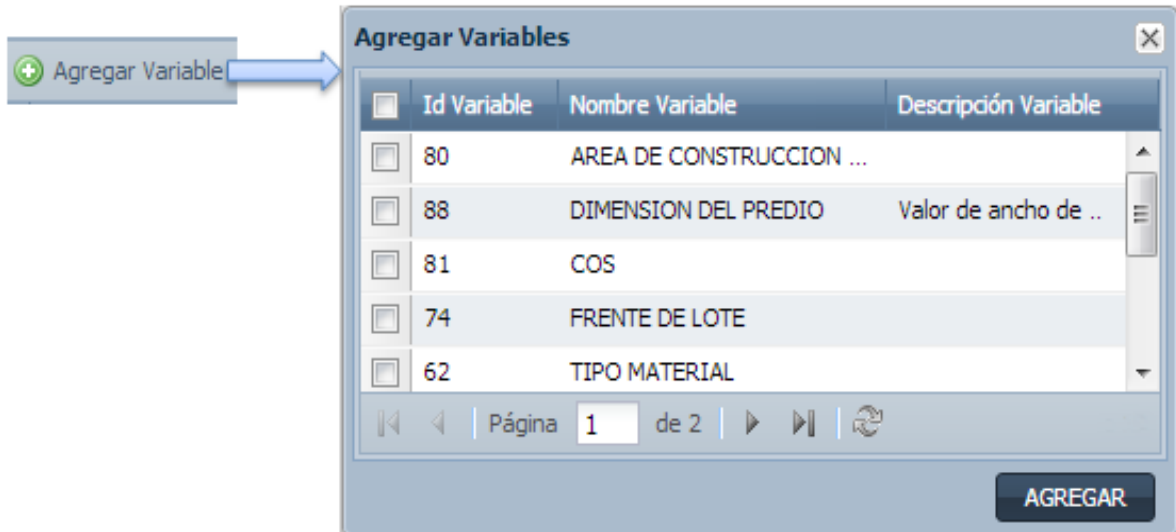
Descripción:

Guardar

**Figura 40: Ventana para crear una nueva variable en la ficha de inspección**

**Fuente: Propia**

En la siguiente imagen se muestra la ventana creada para poder seleccionar variables ya existentes o creadas para otras fichas con el objetivo de no crear variables repetidas y poder reusar o agregarlas a una ficha nueva o diferente.



**Figura 41: Ventana para agregar una variable ya existente en la ficha de inspección**

**Fuente: Propia**

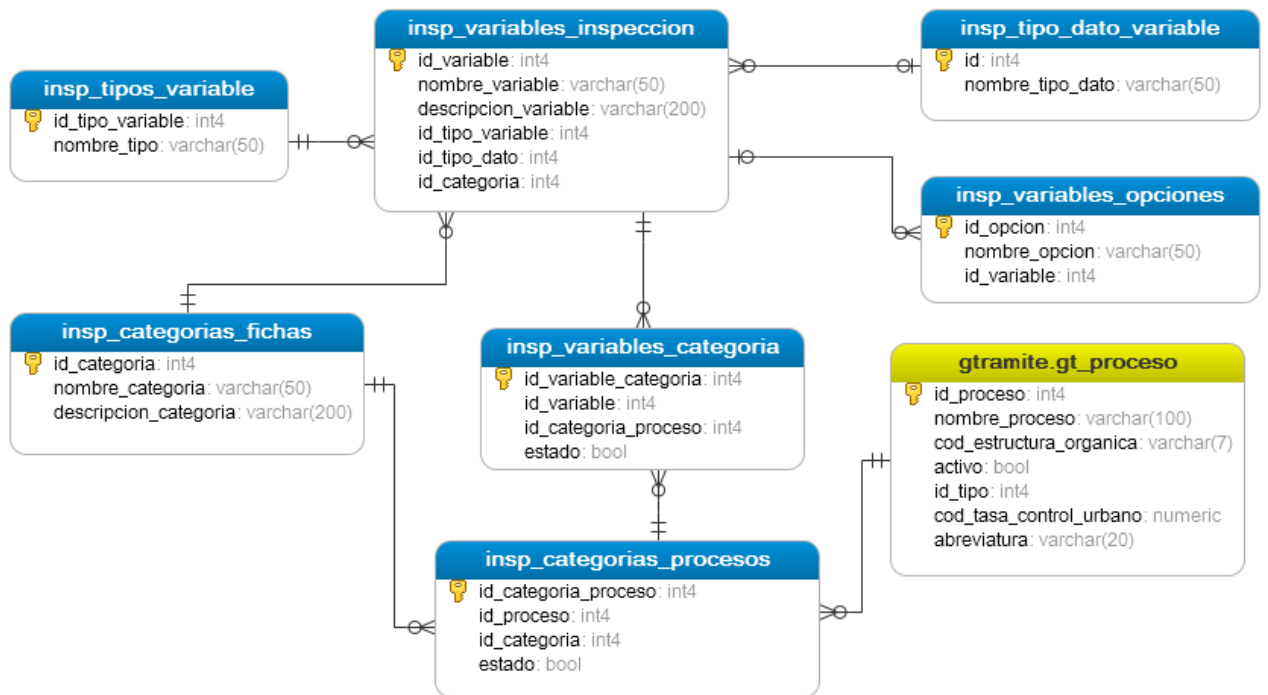
Por último se realiza pruebas creando una ficha con algunos datos o variables nuevas y agregadas y verificando si los datos insertados en las tablas son correctos.

#### ❖ **Historia N°4: Crear categorías y tipos de variables en la ficha de inspección**

Dado a que se presentó la necesidad de mejorar la estructura de una ficha de inspección, en esta historia se describe la posibilidad de crear categorías en las fichas y dentro de cada una de estas sus variables o datos correspondientes; además de que se implementó la posibilidad de crear 3 tipos de datos: simple, selección única y selección múltiple.

#### **Tareas:**

1. Crear, modificar y agregar campos en las tablas para la creación de categorías y tipos de variables en una ficha de inspección.
2. Generar schema y model en Symfony con las nuevas tablas y campos agregados.
3. Creación de models de las nuevas tablas en ExtJS.



**Figura 42: Diagrama Entidad Relación de las tablas usadas para la ficha de inspección**

**Fuente: Propia**

Luego de agregar las nuevas tablas que sirven para la creación de fichas de inspección se actualiza el schema.yml y models en Symfony de igual manera como se indicó anteriormente en la Figura 39; además en ExtJS también se agregan los models de las tablas correspondientes.

```
Ext.define('Imi.model.InspCategoriasFichas', {
    extend: 'Ext.data.Model',

    fields: [
        {name: 'id_categoria_proceso', type: 'int'},
        {name: 'id_categoria', type: 'int'},
        {name: 'nombre_categoria', type: 'string'},
        {name: 'descripcion_categoria', type: 'string'}
    ]
});
```

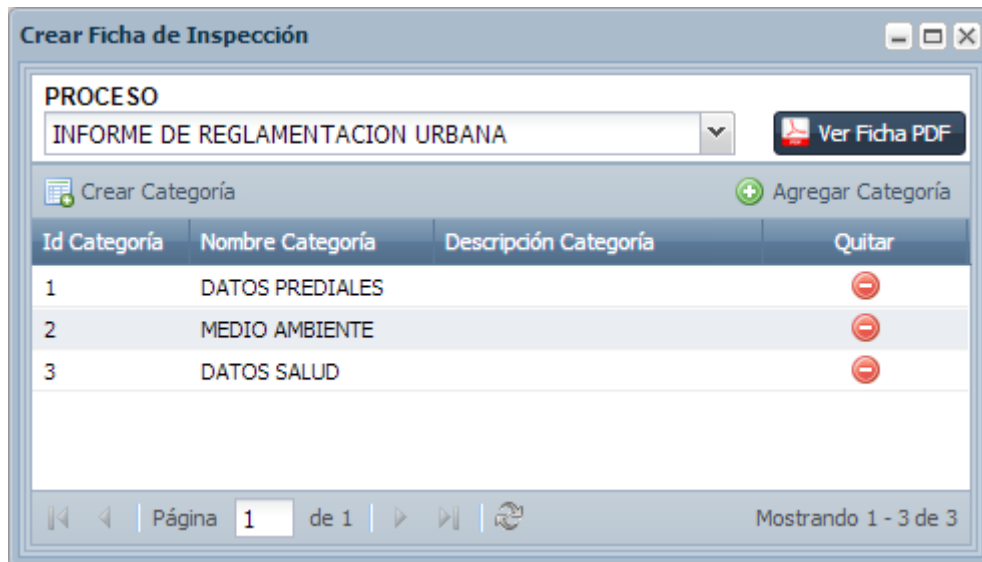
**Figura 43: Código de un model en ExtJS**

**Fuente: Propia**

**Tareas:**

4. Realizar modificaciones en la ventana de fichas agregando otro grid para crear categorías.
5. Crear consulta SQL para obtener el listado de las categorías según el proceso.

6. Programar métodos en ExtJS para visualizar la consulta del listado de categorías en el grid-panel.
7. Agregar dos botones en el grid de categorías donde el uno es para crear una nueva y el segundo para agregar a la ficha una categoría ya existente.



**Figura 44: Ventana y grid-panel con las categorías de la ficha de inspección**

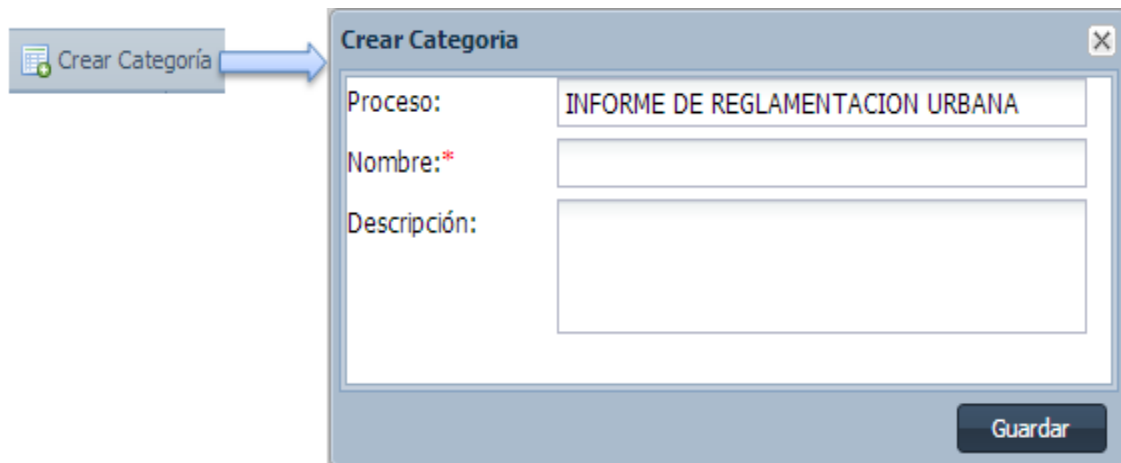
**Fuente: Propia**

#### **Tareas:**

8. Realizar ventana para crear nueva categoría donde se escriba el nombre y una descripción.
9. Programar métodos para enviar datos de la categoría al servidor e insertarlos en las tablas respectivas de la base de datos.
10. Crear ventana con un grid-panel de las categorías ya existentes para agregar en la ficha.
11. Programar métodos para agregar una o más categorías del grid a la ficha, enviar al servidor e insertar los datos en las tablas de la base de datos.

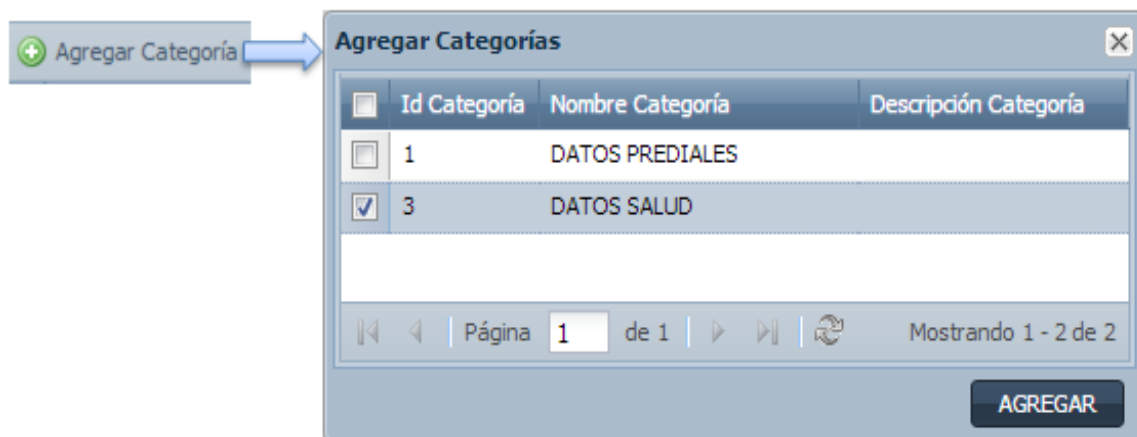
En las siguientes imágenes se muestra primero (*Figura 45*) la ventana realizada para la creación de nuevas categorías donde según el proceso seleccionado es obligatorio ingresar el nombre de la categoría y la descripción es opcional pero es recomendable registrar para tener una definición de las categorías que se van a crear.

En la segunda imagen (*Figura 46*) se muestra la ventana creada para poder seleccionar categorías ya existentes o creadas para otras fichas con el objetivo de no crear categorías repetidas y poder reusar o agregarlas a una nueva o diferente ficha.



**Figura 45: Ventana para crear una nueva categoría en la ficha de inspección**

**Fuente: Propia**



**Figura 46: Ventana para agregar una categoría ya existente en la ficha de inspección**

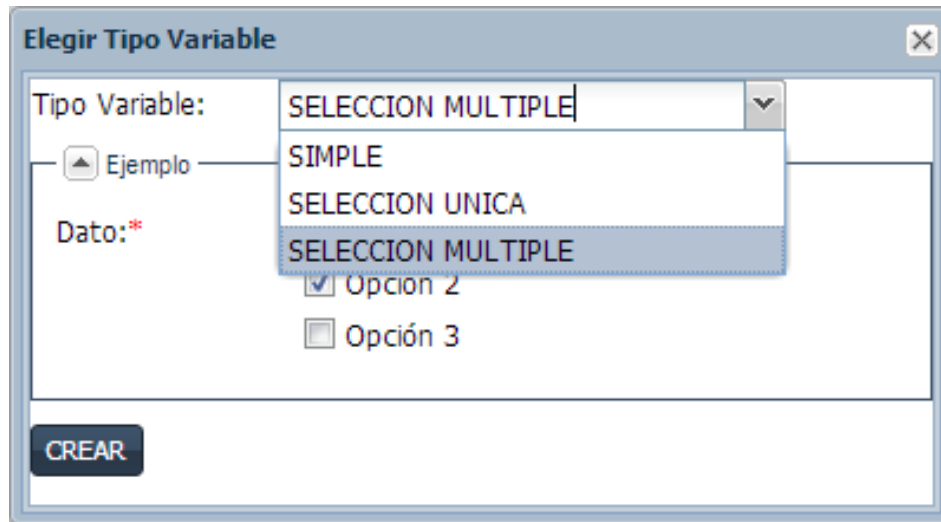
**Fuente: Propia**

En la tarea 12 se modifican algunos métodos haciendo que las variables o datos de una ficha pertenezcan a una categoría y esta categoría a un proceso cambiando la idea inicial donde una variable pertenecía directamente a un proceso.

**Tareas:**

13. Crear y modificar ventanas para tener la opción de crear 3 tipos de variables (Simple, Selección Única y Selección Múltiple) y que pertenezcan a una categoría.
14. Programar eventos en ExtJS para crear las opciones en las variables de tipo Selección Única y Múltiple.
15. Programar eventos para enviar los datos de las variables y sus opciones al servidor e insertar los datos en las tablas respectivas de la base de datos.

En la siguiente imagen se visualiza la ventana realizada para al momento de crear una nueva variable se tenga la opción de seleccionar en un comboBox el tipo de variable a crearse los cuales son 3: simple, selección única y selección múltiple. Además al seleccionar cada una de las opciones presentadas en el comboBox se muestra dentro de un panel el ejemplo del uso de la variable a crearse.



**Figura 47: Ventana para elegir el tipo de variable a crearse**

**Fuente: Propia**

La variable simple o normal es la que se mostró anteriormente en la Figura 40 la cual sirve para el registro de información tipo texto o numérico, los otros tipos de variables se muestra en la siguiente figura.

VARIABLE SELECCIÓN ÚNICA	VARIABLE SELECCIÓN MÚLTIPLE
Dato: <input type="radio"/> Opción 1 <input checked="" type="radio"/> Opción 2 <input type="radio"/> Opción 3	Dato: <input checked="" type="checkbox"/> Opción 1 <input checked="" type="checkbox"/> Opción 2 <input type="checkbox"/> Opción 3

**Figura 48: Ejemplos de una variable de selección única y selección múltiple**

**Fuente: Propia**

En la siguiente figura se muestra la ventana para crear una nueva variable de selección única, la cual es muy similar a la ventana de crear una variable de selección múltiple con la única diferencia de que en la primera además de crear opciones propias se puede escoger opciones predeterminadas como por ejemplo el SI/NO o V/F.



**Figura 49: Ventana para crear una nueva variable de selección única**

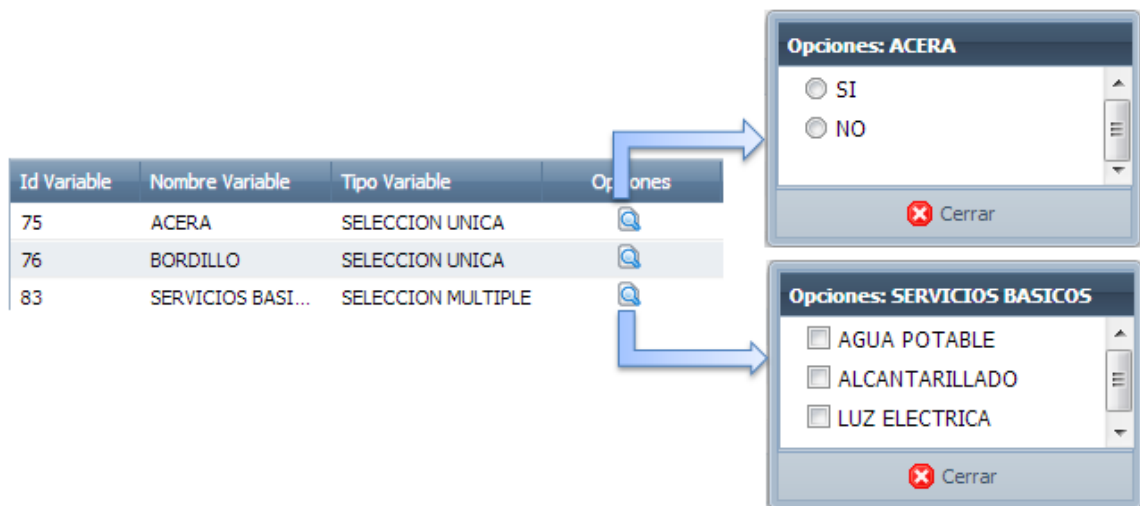
**Fuente: Propia**

En la tarea 16 se realizan pruebas creando una nueva ficha la cual conste de categorías y dentro de cada una se creen variables o datos de los 3 tipos y se verifica si los datos insertados existen en las tablas de la base y son correctos.

**Tareas:**

17. Crear ventana en ExtJS para visualizar las opciones que tiene una variable de Selección Única y Múltiple.
18. Crear consulta SQL para obtener las opciones según la id de variable.
19. Programar métodos en ExtJS para visualizar la consulta SQL en la ventana de opciones.
20. Crear consultas SQL para realizar búsquedas de categorías y variables al momento de agregar una a la ficha.
21. Programar métodos para visualizar las búsquedas en las ventanas de agregar categorías y variables.

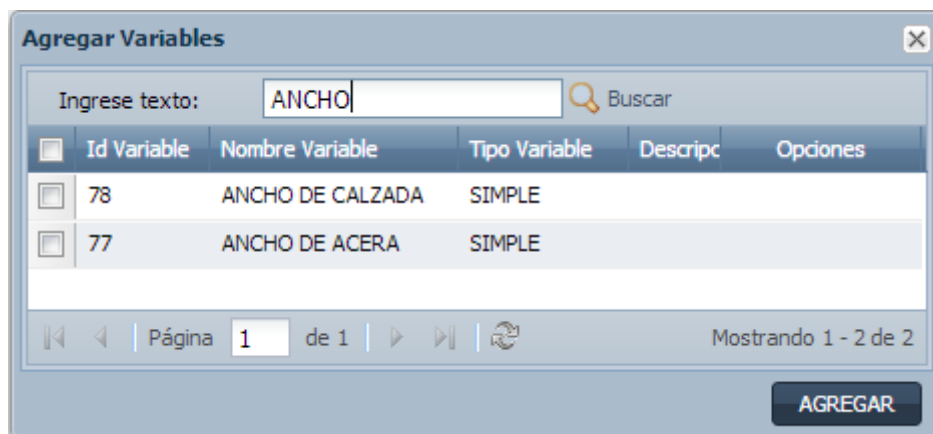
En la Figura 50 se visualizan las ventanas que se despliegan al dar click en el boton ver opciones de las variables tipo selección única y múltiple.



**Figura 50: Ventana para ver las opciones de las variables de selección única y múltiple**

**Fuente: Propia**

En la figura siguiente se muestra la ventana de agregar variables con la opción de realizar búsquedas sobre las mismas por el nombre o su descripción; esta ventana es idéntica y tiene la misma función al agregar categorías.



**Figura 51: Ventana con la opción de buscar variables para agregar a la ficha**

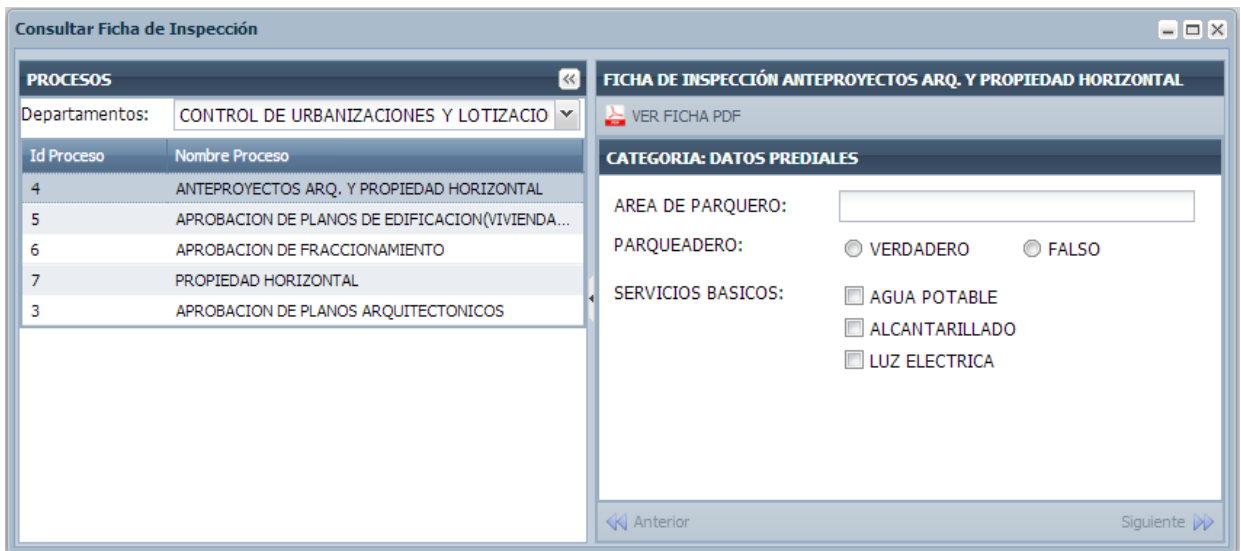
**Fuente: Propia**

En la tarea 22 se realiza pruebas agregando y buscando por nombre y descripciones categorías y variables y verificando si los datos ingresados en las tablas son correctos.

**Tareas:**

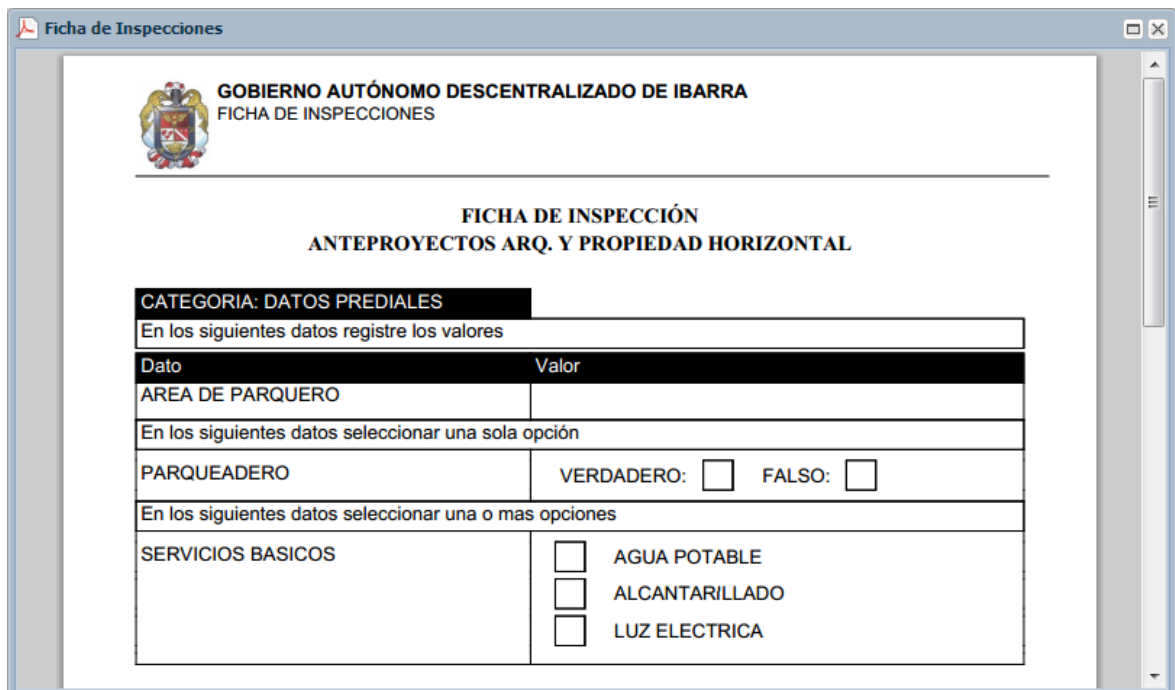
- 23. Crear ventana para visualizar el diseño de la ficha según un proceso.
- 24. Crear consulta SQL para obtener los datos de los departamentos existentes en la institución GAD-I y sus procesos correspondientes.

25. Programar métodos en ExtJS para visualizar la consulta de departamentos y procesos dentro de la ventana de ver ficha.
26. Crear consulta SQL para obtener los datos de la ficha según el proceso seleccionado en la que se incluyen las categorías, variables y opciones.
27. Programar métodos en ExtJS para visualizar la ficha dentro de la ventana de ver ficha.
28. Programar métodos en Symfony y ExtJS para visualizar la ficha en un PDF.



**Figura 52: Ventana para ver las fichas según departamentos y sus procesos**

**Fuente: Propia**



**Figura 53: Archivo PDF visualizando el diseño de una ficha de inspección**

**Fuente: Propia**

Por ultimo en el desarrollo de esta historia de usuario se crea una ficha completa con todas las opciones posibles y se verifica su correcto diseño visualizando el archivo PDF.

### 3.10.3 Iteración 3 (Módulo Agenda de Inspecciones)

En esta iteración se describe el desarrollo del Módulo Agenda de Inspecciones el cual en general es aquel donde se visualizan los trámites seleccionados a inspeccionar para luego mediante la creación de una ruta generar inspecciones.

#### ❖ Historia N°5: Ordenar trámites para generar inspecciones

En esta historia se describen las tareas para que un usuario arquitecto o inspector pueda ordenar los trámites según la ubicación de los predios y pueda generar inspecciones.

#### Tareas:

1. Creación de tablas para el registro de las agendas de inspecciones.
2. Generar schema y model en Symfony con las nuevas tablas.
3. Creación del módulo Agenda de Inspecciones en Symfony y ExtJS.
4. Creación del controlador, models y stores en ExtJS para el módulo Agenda de Inspecciones.

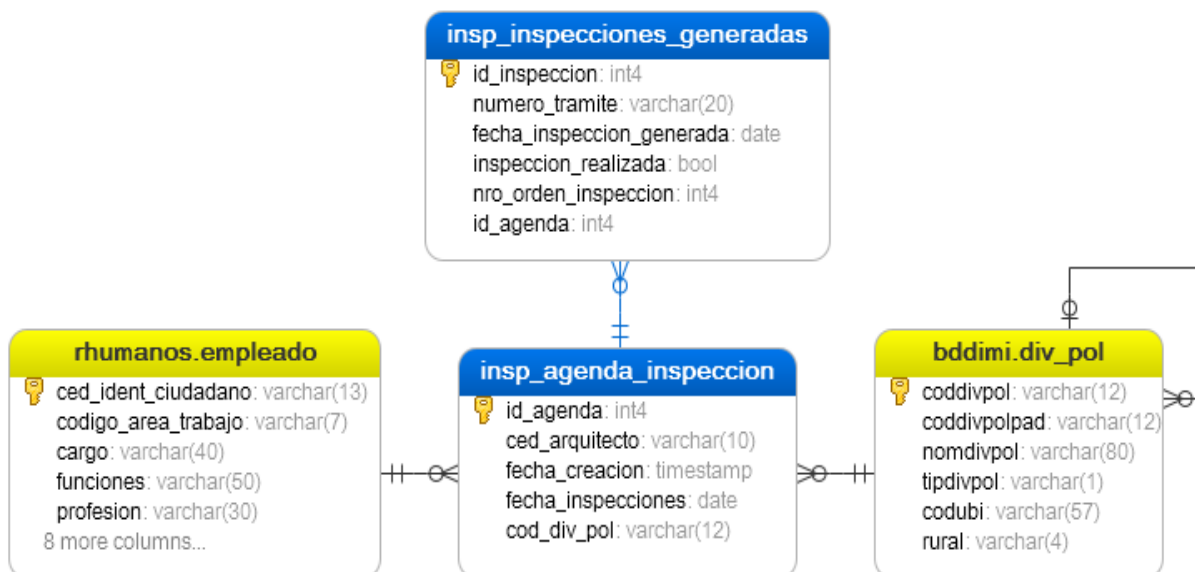


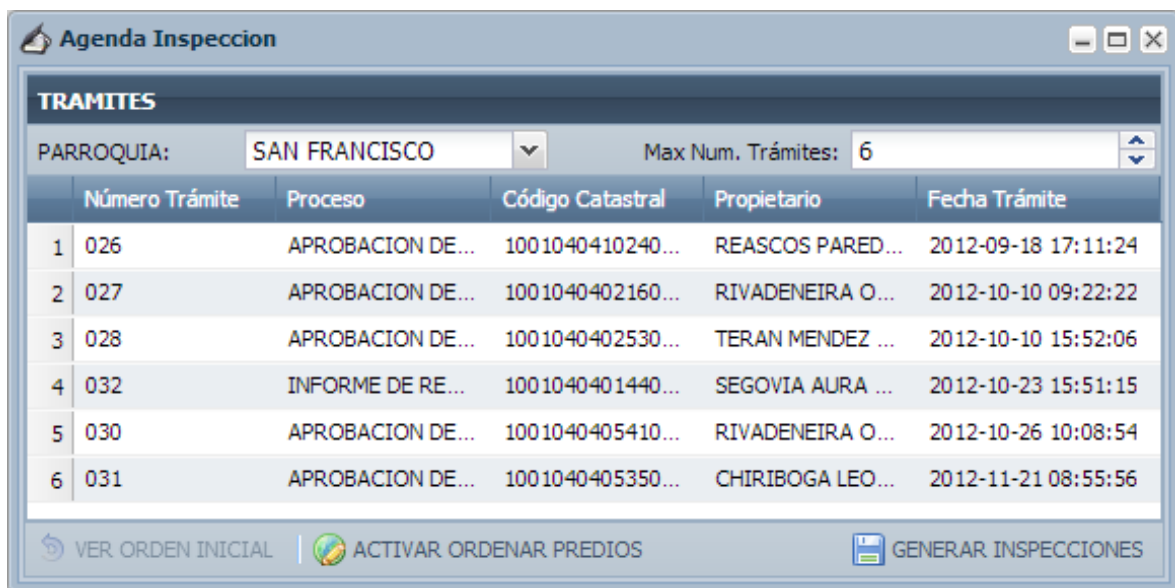
Figura 54: Diagrama Entidad Relación de las tablas usadas para la agenda de inspecciones

Fuente: Propia

Luego de crear las nuevas tablas, en las siguientes tareas 2 a 4 se vuelve a generar el schema.yml y los models en Symfony; además se crea el módulo Agenda de Inspecciones en Symfony y ExtJS en el cual consta el controlador, models y stores de la misma manera como se creó los módulos anteriores.

**Tareas:**

5. Creación de ventana en ExtJS para ver y ordenar los trámites que se han asignado inspección.
6. Crear consulta SQL para obtener las parroquias asignadas al arquitecto logueado y visualizar en un comboBox.
7. Agregar numberfield para ingresar y determinar el número de inspecciones a realizarse.
8. Crear consulta SQL para obtener solo los trámites creados que necesitan inspección y visualizarlos en el grid-panel.



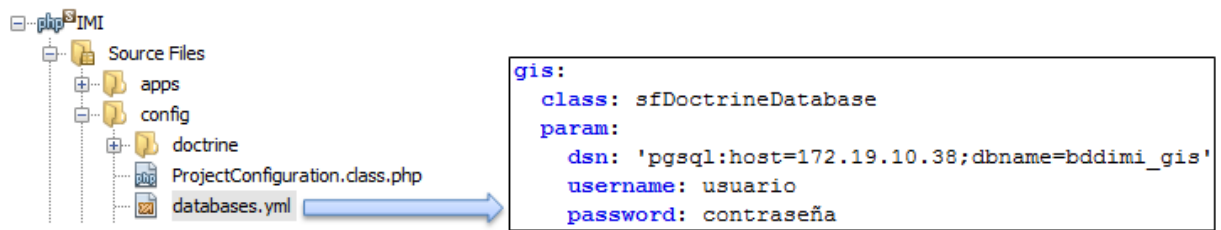
**Figura 55: Ventana para ordenar y filtrar los trámites con inspección**

**Fuente: Propia**

**Tareas:**

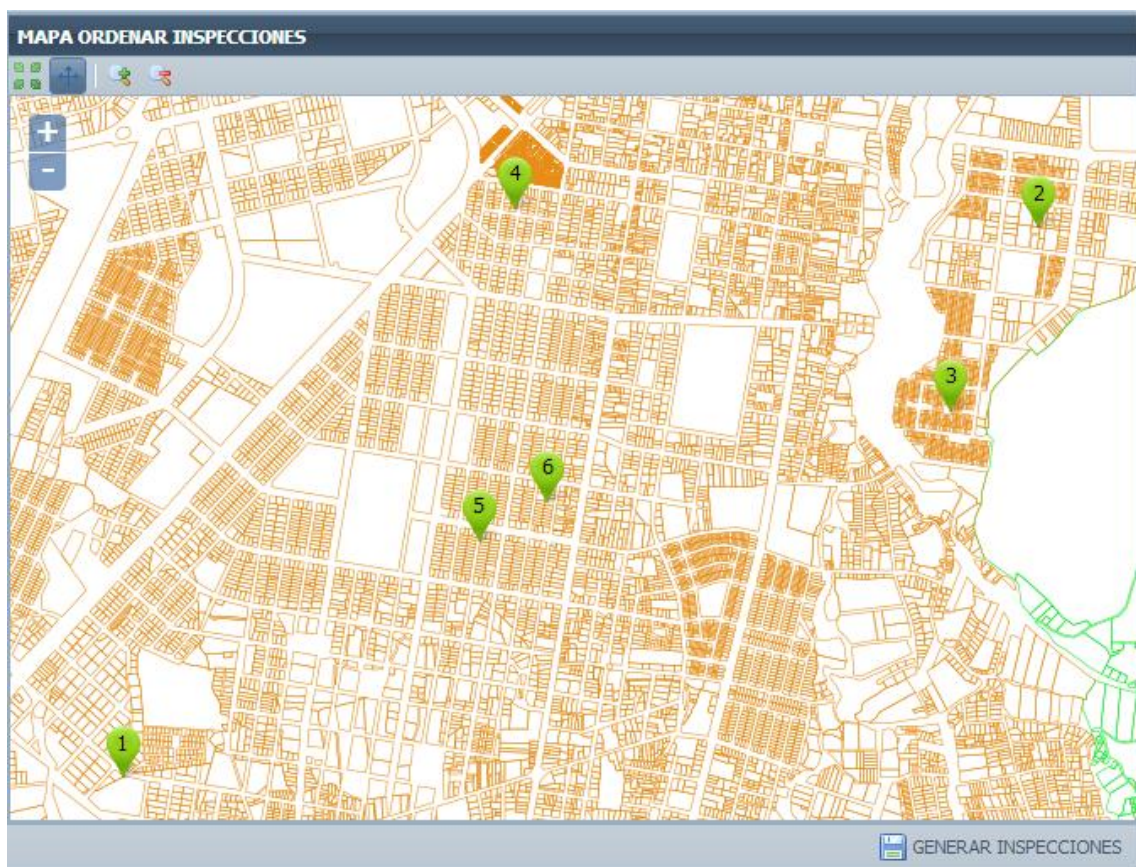
9. Agregar conexión hacia la base de datos GIS del municipio en el archivo database.yml de Symfony.
10. Crear control, ventana y layouts para ver el mapa en ExtJS.
11. Programar métodos para consultar y visualizar la ubicación de los predios en el mapa según la clave catastral de los trámites.

12. Agregar boton para poder activar la creación de la ruta de inspecciones y programar métodos para elegir orden de predios.



**Figura 56: Configuración de la conexión a la base de datos GIS**

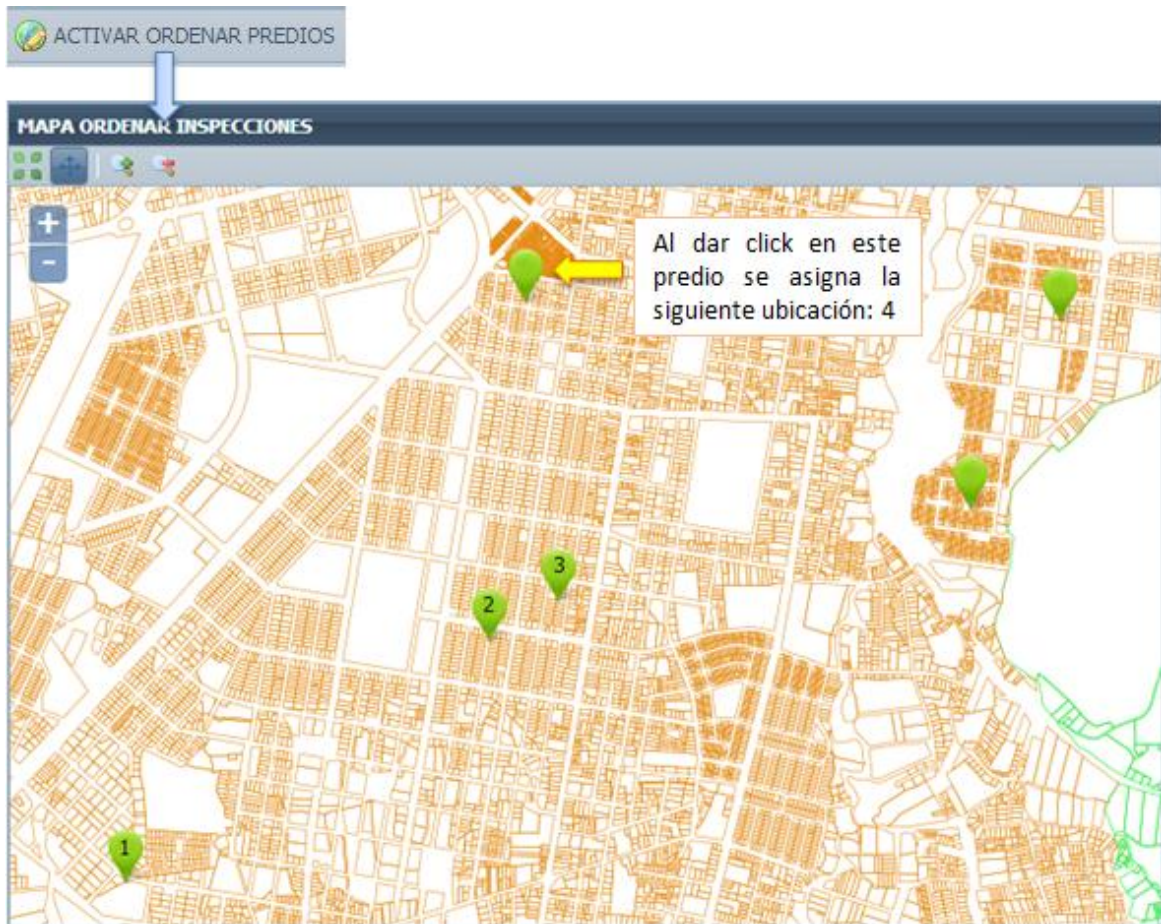
**Fuente: Propia**



**Figura 57: Ventana con la ubicación de los predios de los trámites a inspeccionar**

**Fuente: Propia**

En la siguiente figura se visualiza la ordenación de predios la cual se activa al dar click en el boton Activar Ordenar Predios con lo cual quedan vacíos los números que se encuentran dentro de los iconos de las ubicaciones de los mismos y al dar click en cada uno de ellos se va poniendo el orden para la ruta de inspección.



**Figura 58: Activación de Ordenación de Predios para ruta de inspección**

**Fuente: Propia**

#### **Tareas:**

13. Agregar boton y programar métodos para generar inspecciones.
14. Programar métodos para identificar si el día de inspección es igual al día actual para que sea el usuario el que elige que día va ser la inspección.
15. Crear ventana para visualizar en un PDF las inspecciones generadas en la agenda incluido la foto y mapa respectivo del trámite.
16. Crear métodos para consultar los datos del trámite, predio, propietario, peticionario, foto, mapa y presentar en el PDF.

En la Figura 59 se visualiza la ventana creada para presentar la información de las inspecciones generadas luego de ordenar los predios; dentro del PDF se detalla información de la fecha y orden de inspección al predio asignado automáticamente según el horario de inspección actual, también se muestra datos del propietario y predio incluyendo el mapa y foto si existe.

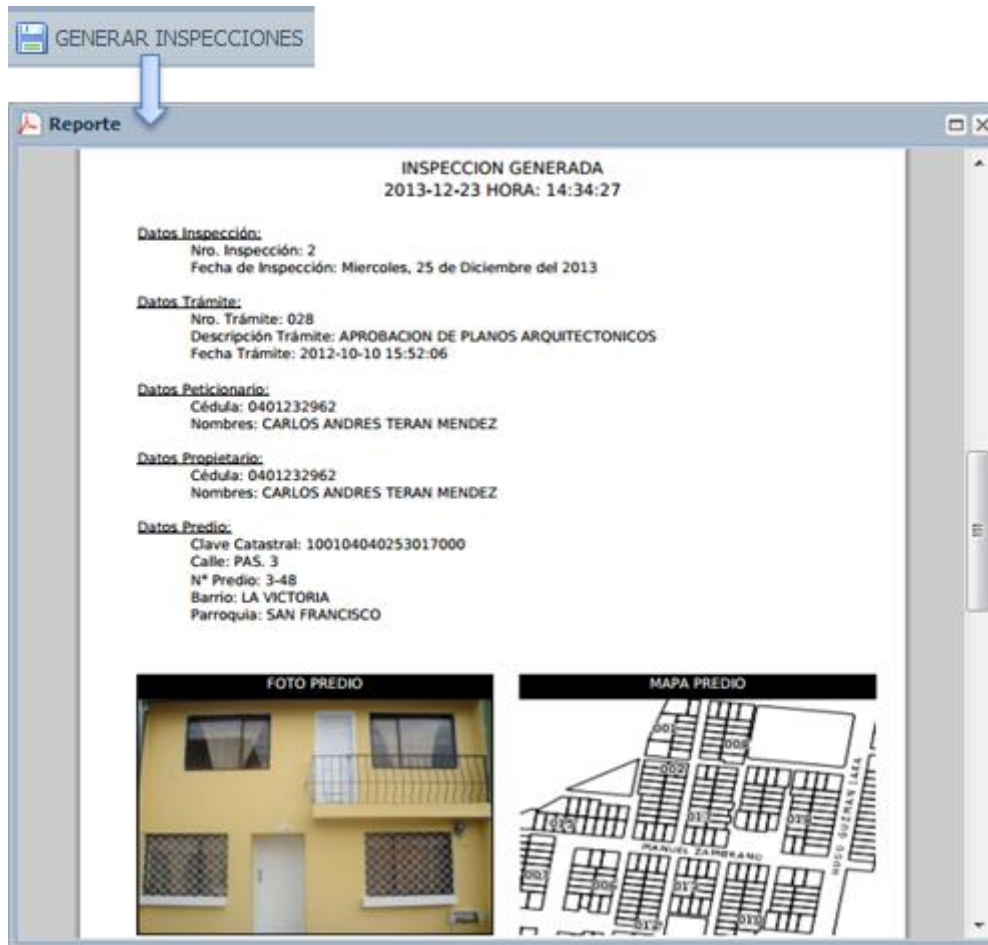


Figura 59: Archivo PDF visualizando la agenda e inspecciones generadas

Fuente: Propia

Por último se realizan pruebas mediante la generación de trámites con inspección luego ordenarlos para crear una ruta y generar una agenda de inspecciones visualizando si los datos son correctos en el archivo PDF.

#### ❖ Historia N°6: Ver información del trámite

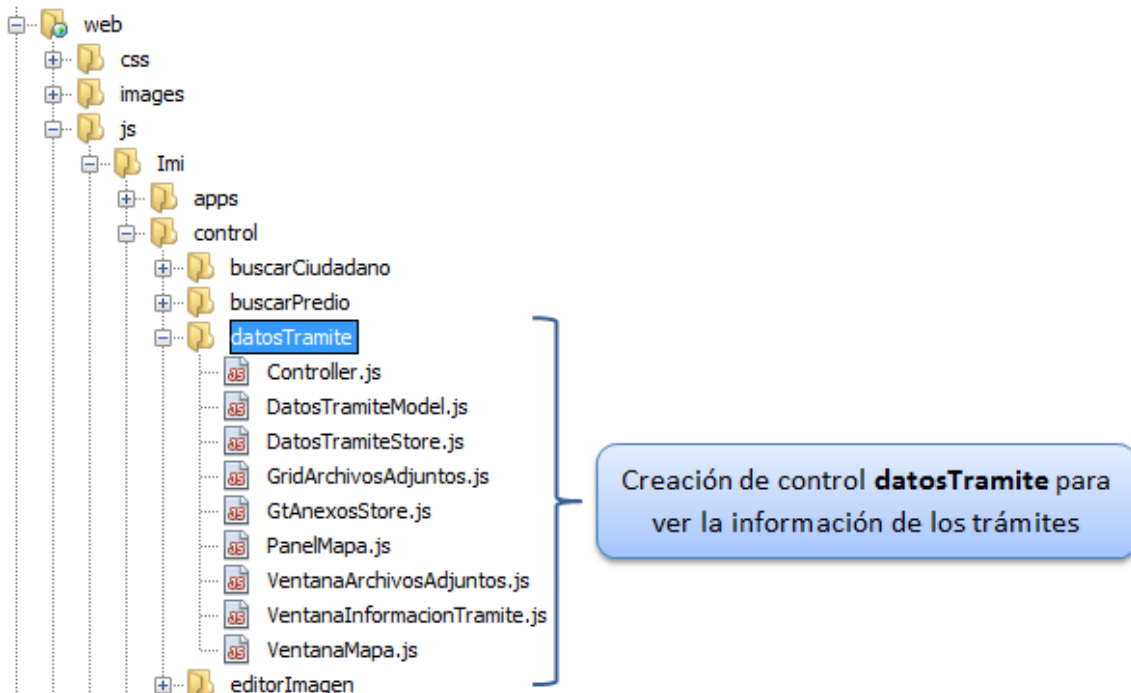
Dado la necesidad de visualizar una información más amplia sobre los trámites con inspección, en esta historia de usuario se describe la creación y uso de un control o ventana que muestre una información general del trámite.

#### Tareas:

1. Crear un control genérico en ExtJS para ver los datos de un trámite en cualquier módulo del sistema.
2. Crear controlador, models, stores y views del control genérico llamado datos trámite.

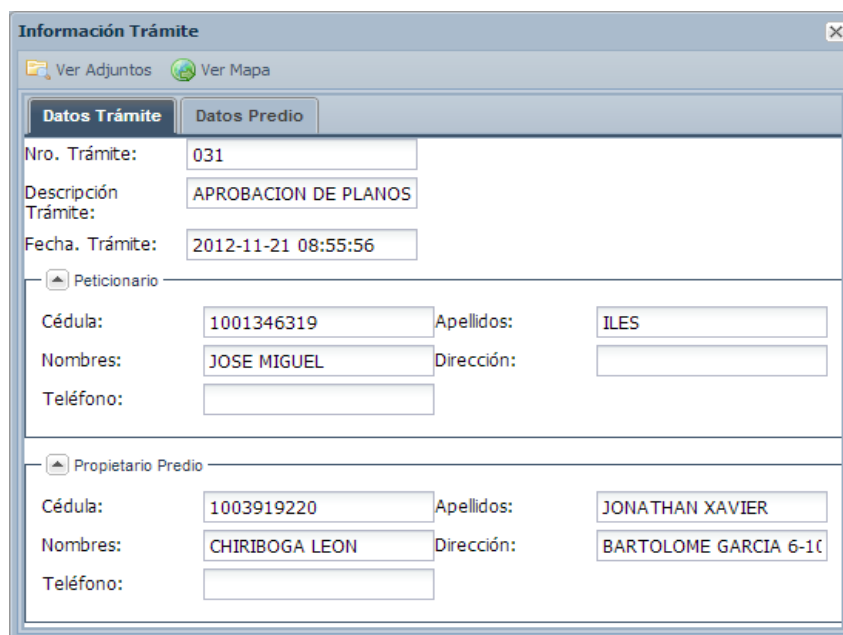


3. Crear consultas SQL para obtener los datos del trámite.
4. Crear consulta SQL para obtener los datos y foto del predio.
5. Agregar tab-panel para ver datos del trámite y predio.
6. Programar métodos para visualizar las consultas del trámite y predio en la ventana del control datos trámite.



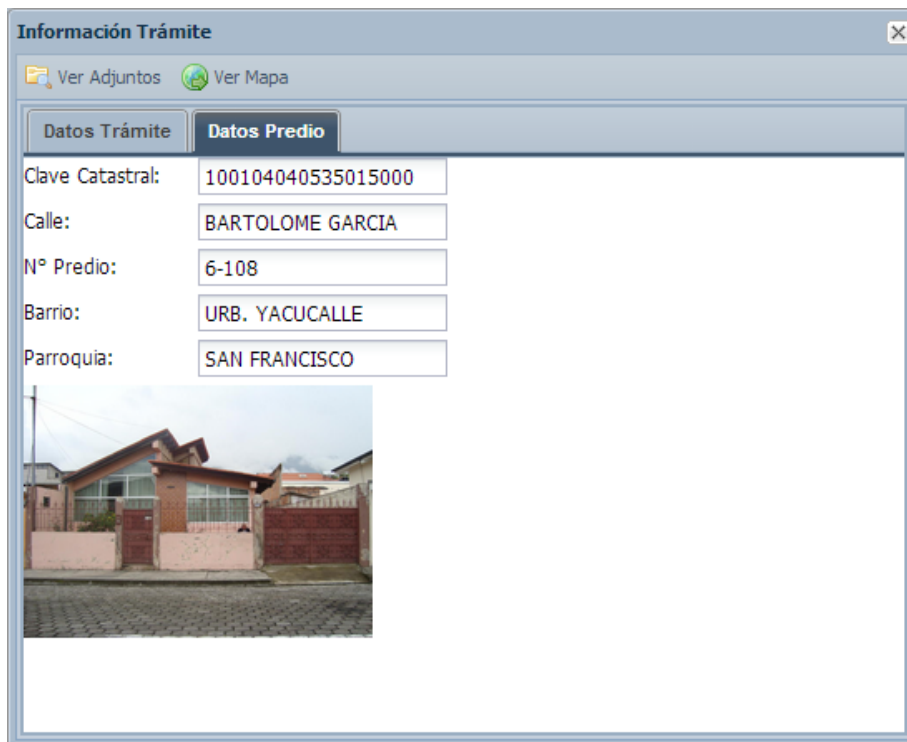
**Figura 60: Ubicación y estructura del control datos trámite**

**Fuente: Propia**



**Figura 61: Ventana y tab-panel con datos del trámite**

**Fuente: Propia**



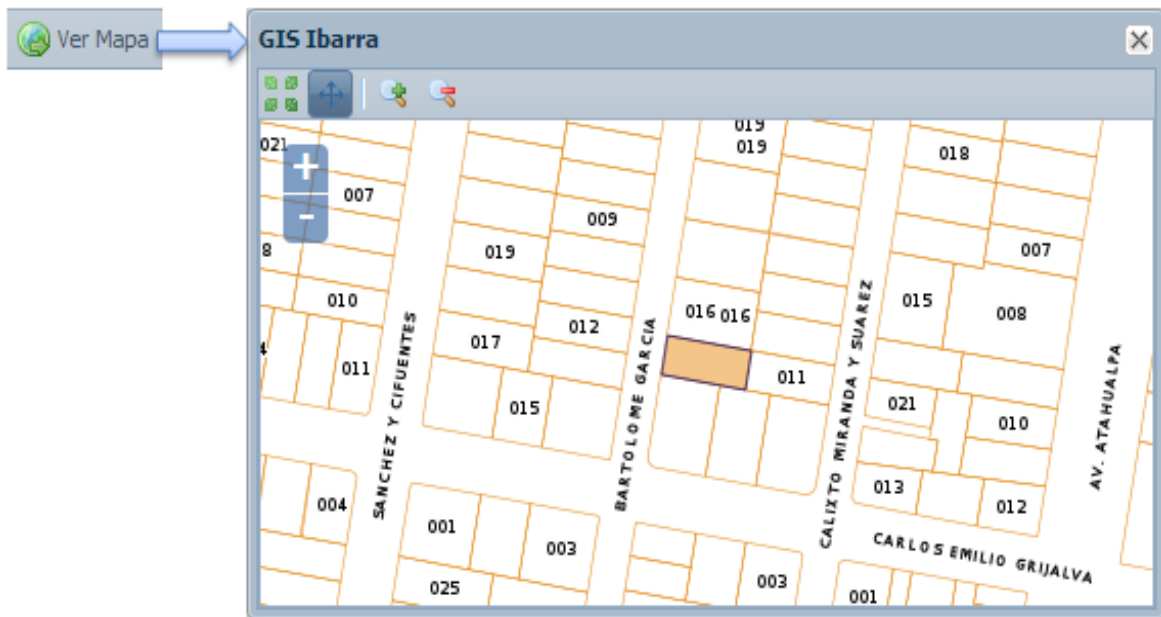
**Figura 62: Ventana y tab-panel con datos del predio**  
Fuente: Propia

**Tareas:**

7. Agregar dos botones en la ventana, uno para ver los archivos adjuntos y otro para ver el mapa del predio.
8. Crear consulta SQL para obtener los archivos adjuntos según el número del trámite.
9. Crear grid-panel y programar métodos para mostrar el listado de los archivos adjuntos al dar click en el boton "Ver Archivos Adjuntos".
10. Crear métodos para visualizar el mapa dentro del control de datos de trámite al dar click en el boton "Ver Mapa".



**Figura 63: Ventana de archivos adjuntos de un trámite**  
Fuente: Propia



**Figura 64:** Ventana con mapa de la ubicación del predio de un trámite

**Fuente:** Propia

Por ultimo en la tarea 11 se realizan pruebas mediante la creación de una columna en el grid de trámites para inspección en la cual al dar click en el boton Ver Información se despliega la ventana del control datos trámite y se verifica si la información es correcta incluyendo foto y mapa del predio.

❖ **Historia N°7: Enviar e-mail al ciudadano informando el día de inspección**

En esta historia de usuario se describe él envió de correos electrónicos al momento de generar las inspecciones informando al ciudadano la fecha de visita a su predio para la respectiva inspección según el tramite solicitado.

**Tareas:**

1. Consultar como enviar e-mails con PHP y Symfony.
2. Usar sfMailer y configurarlo en el archivo factories.yml.
3. Agregar dentro del método creación de Agenda de Inspecciones el código con el formato y envió de email.
4. Realizar pruebas creando inspecciones y verificar si ha llegado el email enviado a una dirección electrónica.

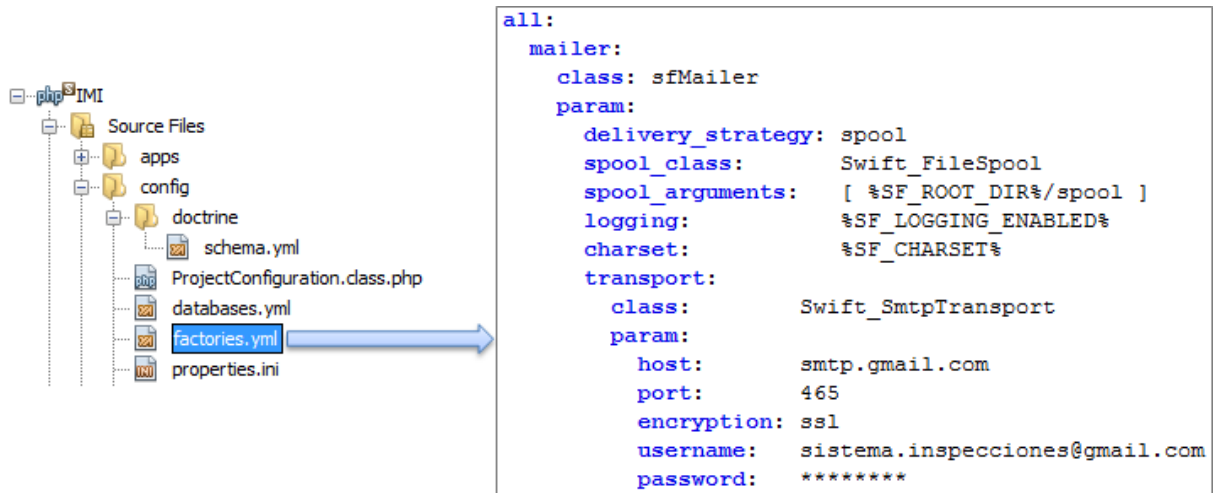


Figura 65: Configuración de sfMailer para envío de correos electrónicos

Fuente: Propia

```

public static function notify($mailer,$contenidoEmail) {
    $email_from = sfConfig::get('app_mail_sistemas');
    $mensaje = $mailer->compose($email_from, 'correo_ciudadano@hotmail.com',
        "Fecha de Inspección al Predio");
    $mensaje->setBody($contenidoEmail, 'text/html');
    $mailer->send($mensaje);
}

```

Figura 66: Código para la creación de archivo para envío de e-mail

Fuente: Propia



Figura 67: Correo electrónico enviado a un ciudadano

Fuente: Propia

### 3.10.4 Iteración 4 (Módulo Registro de Inspecciones)

En esta iteración se describen las historias de usuario para el registro de inspecciones con todos los datos necesarios y también para visualizar las inspecciones registradas.

#### ❖ Historia N°8: Registro de Inspecciones

En esta historia de usuario se describen las tareas para la creación de las ventanas donde se realiza el registro de las inspecciones incluyendo los datos de la ficha de inspección y fotos adjuntadas.

#### Tareas:

1. Creación de tablas para el registro de las inspecciones.
2. Generar schema y model en Symfony con las nuevas tablas.
3. Creación del módulo Registro de Inspecciones en Symfony y ExtJS.
4. Creación del controlador, models y stores para el módulo Registro de Inspecciones.

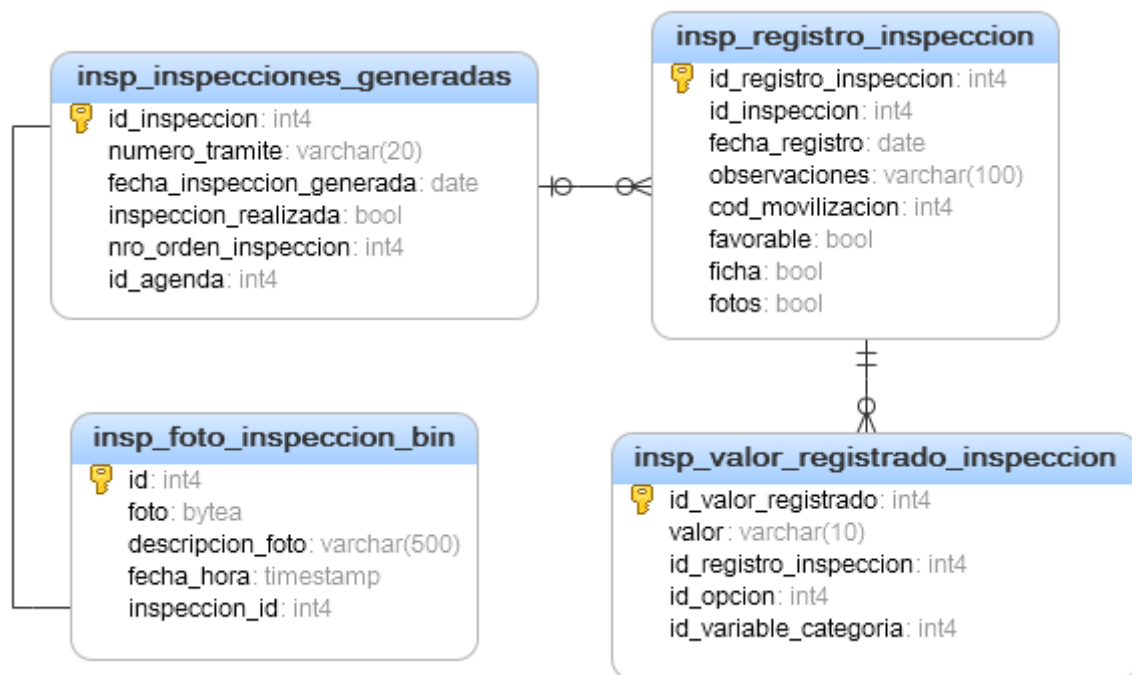



Figura 68: Diagrama Entidad Relación de las tablas para registrar una inspección







Fuente: Propia

En las tareas 2-4 se genera nuevamente el schema.yml y los models con las nuevas tablas; además se crea el nuevo módulo Registro de Inspecciones con su controlador, models y stores en ExtJS de la misma manera como ya se realizó en los módulos anteriores.

## Tareas:

5. Creación de ventana, paneles y grid-panels en ExtJS para ver en maestro-detalle las agendas y sus inspecciones correspondientes generadas.
6. Crear consulta SQL para obtener las agendas creadas y trámites con inspecciones correspondientes.
7. Programar métodos y eventos para presentar los datos de las consultas de agendas e inspecciones en forma de maestro-detalle.
8. Agregar al grid de inspecciones (detalle) una columna con un boton para poder ver la información del trámite.



Nro-Inspección	Numero Trámite	Proceso	Clave Catastral	Propietario	Fecha Inspección	Ver Información Trámite
1	027	APROBACION DE PLANOS ARQ...	100104040216012000	RIVADENEIRA ORTEGA EDGAR RAUL	2013-12-25	
2	028	APROBACION DE PLANOS ARQ...	100104040253017000	TERAN MENDEZ CARLOS ANDRES	2013-12-25	
3	032	INFORME DE REGLAMENTACIO...	100104040144003008	SEGOVIA AURA MARIA	2013-12-25	
4	031	APROBACION DE PLANOS ARQ...	100104040535015000	CHIRIBOGA LEON JONATHAN XAVIER	2013-12-25	
5	030	APROBACION DE PLANOS ARQ...	100104040541012000	RIVADENEIRA ORTEGA NANCY ELENA	2013-12-25	
6	026	APROBACION DE PLANOS ARQ...	100104041024003000	REASCOS PAREDES IRVING MARLON	2013-12-25	

**Figura 69: Ventana Maestro-Detalle con las agendas e inspecciones generadas**

**Fuente: Propia**

## Tareas:

9. Crear ventana para ingresar los datos y registro de la inspección donde además se puede registra la ficha y adjuntar fotos.
10. Crear ventana para seleccionar cual fue el código de movilización para realizar la inspección.
11. Crear consulta SQL para obtener los datos de movilización y además filtrar por cod\_movilizacion y cod\_vehiculo.
12. Programar métodos y eventos en ExtJS para presentar los datos de movilización como el código del vehículo, fecha de emisión, fecha de autorización, actividad y luego seleccionar uno del listado para agregar al registro de inspección.

**Registro Inspección**

Registrar Ficha | Adjuntar Fotos | Información Trámite

Número Trámite: 027

Fecha Registro: 26/12/2013

Cod-Movilización:

Observaciones:

Aprobado:\*\*  SI  NO

**GUARDAR**

**Figura 70: Ventana para registrar una inspección**

**Fuente: Propia**

**MOVILIZACION**

**Buscar**

Buscar por:  Código Movilización  Código Vehículo

Código Vehículo: 120

**Buscar**

**Listado de Movilizaciones**

Código	Código Vehículo	Fecha Emisión	Fecha Autorización	Actividad
4040	120	Thu Mar 27 2008 19:00:0...	Sat Mar 29 2008 08:00:00 GMT-050...	TRASLADO ...
10392	120	Tue Feb 01 2011 19:00:0...	Thu Feb 03 2011 06:00:00 GMT-05...	TRASLADAR...
15135	120	Mon May 14 2012 19:00:...	Wed May 16 2012 07:00:00 GMT-0...	TRASLADAR...

Página 1 de 1 | Mostrando 1 - 3 de 3

**Figura 71: Ventana para seleccionar y buscar el dato de vehículo de movilización**

**Fuente: Propia**

**Tareas:**

13. Crear consulta SQL para obtener la ficha respectiva según el proceso del trámite seleccionado.

14. Crear ventana y programar métodos para mostrar la ficha respectiva.
15. Programar métodos y eventos en la parte del servidor Symfony y el cliente ExtJS para registrar los datos de la ficha, enviar al servidor e insertarlos en una variable de sesión temporal.
16. Crear ventana para adjuntar fotos tomadas en la inspección y escribir una descripción de la misma.
17. Programar métodos y eventos en la parte del servidor Symfony y el cliente ExtJS para adjuntar fotos y enviar al servidor con su descripción e insertarlos en una variable de sesión temporal.

The screenshot shows a web application window titled "FICHA DE INSPECCIÓN". It contains the following information:

- Información Trámite:**
  - Nro. Trámite: 027
  - Clave Catastral: 100104040216012000
  - Proceso: APROBACION DE PLANOS ARQUITECTONICOS
  - Ver Más button
- CATEGORIA: DATOS PEDIALES:**
  - AREA DE LOTE: 23,34
  - COS: 20
  - COS TOTAL: 20
  - FRENTE DE LOTE: 90
  - ACERA:  SI  NO
  - BORDILLO:  SI  NO
  - SERVICIOS BASICOS:
    - AGUA POTABLE
    - ALCANTARILLADO
    - LUZ ELECTRICA

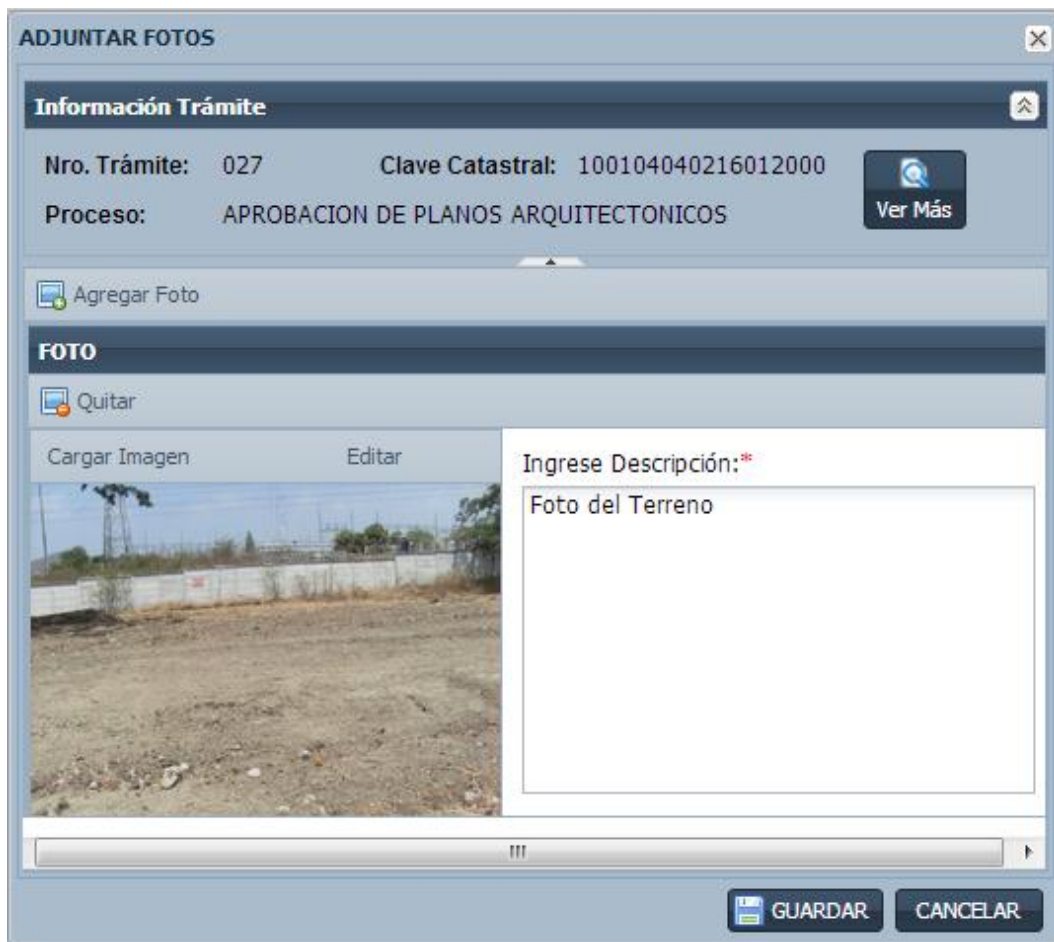
Navigation buttons: Anterior, Siguiente, and a GUARDAR button.

**Figura 72: Ventana para registrar la ficha de inspección**

**Fuente: Propia**

En la siguiente imagen (*Figura 73*) se visualiza la ventana creada para adjuntar al registro de inspección una o más fotos tomadas durante la visita al predio, además el usuario puede ingresar una descripción de cada una de las fotos adjuntadas.





**Figura 73: Ventana para adjuntar fotos de la inspección**

**Fuente: Propia**

En la tarea 18 se crean los métodos para el registro de una inspección con todos los datos necesarios y en la tarea 19 se realizan pruebas mediante la creación de una agenda para luego registrar una inspección incluyendo código de movilización, ficha, fotos y se verifica si los datos son guardados correctamente en las tablas correspondientes de la base de datos.

#### ❖ **Historia N°9: Recortar fotos de inspección**

Al presentarse la necesidad de querer recortar solo una parte de las fotos tomadas en una inspección se crea esta historia de usuario en la que se especifican las tareas para agregar dicha función la cual se hace uso al momento de adjuntar fotografías en el registro de inspección.

#### **Tareas:**

1. Crear ventana para ver y editar la imagen cargada.
2. Programar métodos y eventos para cargar la imagen en la ventana de editar.

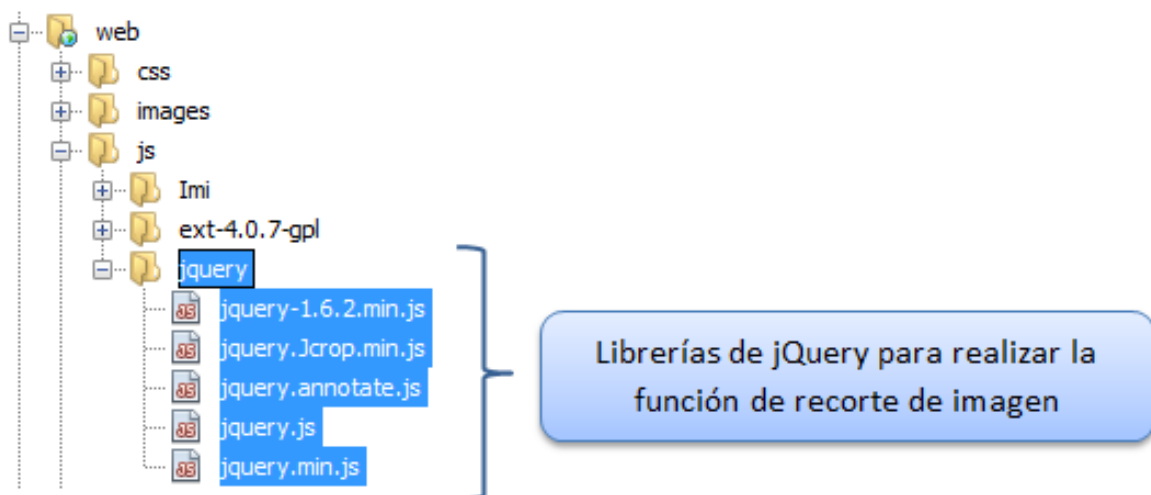
3. Consultar como recortar una imagen con PHP y JavaScript.
4. Usar librería jQuery y programar métodos con PHP para realizar el recorte de la imagen.
5. Programar métodos para reemplazar la imagen anterior por la nueva recortada.



**Figura 74: Ventana para ver y editar imagen adjuntada de la inspección**

**Fuente: Propia**

En la siguiente imagen se muestran las librerías jquery usadas para agregar en la ventana de edición la función de seleccionar dentro de la imagen un cuadro de recorte.



**Figura 75: Librerías jQuery para el uso de recorte de imagen**

**Fuente: Propia**

En la siguiente imagen se visualiza el funcionamiento y uso de las librerías para poder recortar una imagen, la cual se activa al dar click en el boton indicado en la imagen y luego se selecciona un cuadro y por último se aplica el recorte.



**Figura 76: Usando la función de recorte de una imagen**

**Fuente: Propia**

En la tarea 6 se realizan pruebas adjuntando una imagen en el registro de una inspección para luego recortarla y guardarla en la base de datos verificando si dicha función se aplicó correctamente.

#### ❖ **Historia N°10: Consultar las inspecciones realizadas**

En esta historia de usuario se describen las tareas realizadas para luego de haber registrado las inspecciones poder visualizarlas mediante algunos filtros como fecha, numero de trámite, y por agenda.

#### **Tareas:**

1. Crear controlador, stores y models de las tablas y funciones necesarias.
2. Crear ventana con el diseño para consultar las inspecciones realizadas.
3. Crear consultas SQL para obtener el listado de las agendas con sus respectivas inspecciones registradas.



**Figura 77: Ventana de consultar inspecciones con opción de ver por agenda**  
**Fuente: Propia**

En la siguiente imagen se muestra la ventana creada para consultar todas las inspecciones registradas las cuales pueden ser filtradas y consultadas según la agenda de inspección.

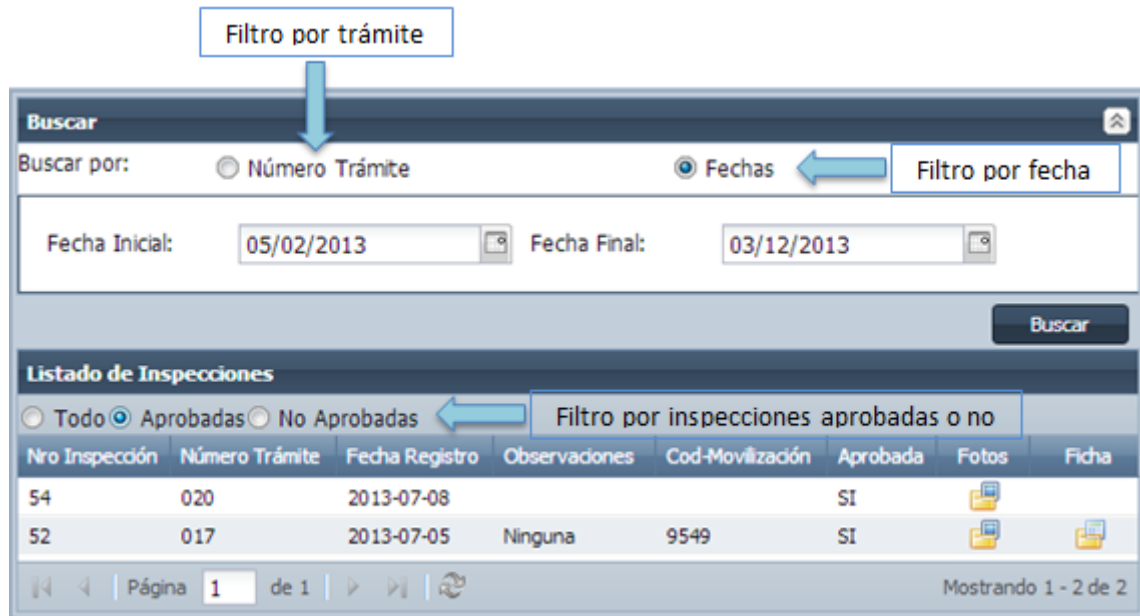


**Figura 78: Ventana con listado de inspecciones registradas**  
**Fuente: Propia**

**Tareas:**

4. Crear consultas SQL para realizar filtros de búsquedas por número de trámite, fechas de registro y si las inspecciones son o no aprobadas.

5. Programar métodos y eventos para visualizar las consultas con sus filtros.
6. Crear consultas SQL para obtener los datos de las inspecciones incluyendo las fotos y ficha registrada.
7. Programar métodos y eventos para visualizar las consultas de fotos y ficha.



**Figura 79: Filtros posibles para consultar inspecciones registradas**

**Fuente: Propia**

En la siguiente imagen se visualiza la ventana que se despliega al dar click en el boton Ver Fotos la cual muestra todas las fotos adjuntadas y su respectiva descripción al momento de registrar la inspección.



**Figura 80: Ventana para visualizar las fotos adjuntadas de la inspección**

**Fuente: Propia**

En la siguiente imagen se visualiza la ventana que se despliega al dar click en el boton Ver Ficha la cual muestra toda la ficha, categorías y datos registrados durante la visita e inspección del predio realizado según el trámite correspondiente.

The screenshot shows a window titled 'Ficha de Inspecciones' with a sub-header 'DATOS REGISTRADOS FICHA DE INSPECCIÓN'. Below this is a table with the following data:

CATEGORIA: DATOS PEDIALES		
Nro	Dato	Valor
1	AREA DE LOTE	12
2	FRENTE DE LOTE	10
3	ACERA	SI
4	BORDILLO	NO
5	ANCHO DE ACERA	18
6	ANCHO DE CALZADA	20
7	AREA DE CONSTRUCCION EN PLANTA BAJA	20
8	AREA DE CONSTRUCCION EN OTRAS PLANTAS	21
9	COS	21
10	COS TOTAL	20
11	SERVICIOS BASICOS	LUZ ELECTRICA, AGUA POTABLE

**Figura 81: Ventana para visualizar la ficha de inspección con los datos registrada**

**Fuente: Propia**

En la tarea 8 se realizan pruebas registrando algunas inspecciones y luego se consultas todos sus datos usando todos los filtros posibles.

❖ **Historia N°11: Registrar el Personal responsable de la Inspección**

Al presentarse la necesidad de agregar la información del grupo de personas responsables de realizar una inspección, en esta historia de usuario se describen las tareas realizadas para que al momento de registrar una inspección se pueda agregar personal que trabaja dentro del GAD-I.

**Tareas:**

1. Creación de tabla para el registro del personal responsable.
2. Agregar grid en la ventana Registro de Inspección con un boton para agregar personal del GAD-I.

3. Crear ventana donde se presente el listado de empleados del GAD-I y se pueda filtrar la búsqueda por cédula y nombres.
4. Crear consultas SQL para obtener los empleados del GAD-I con y sin filtros.

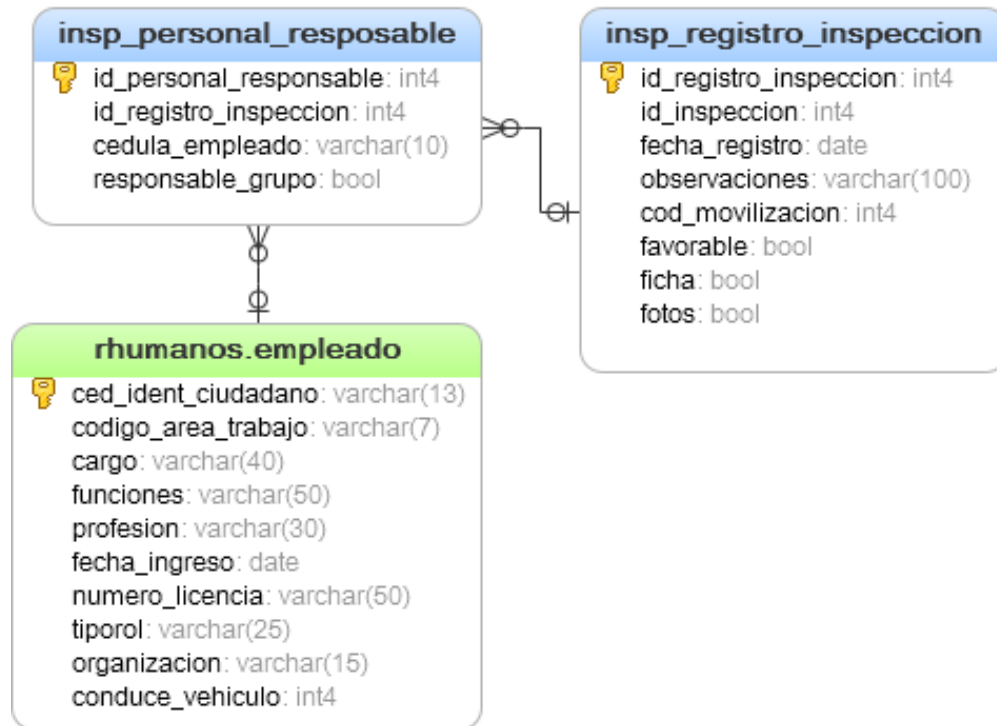
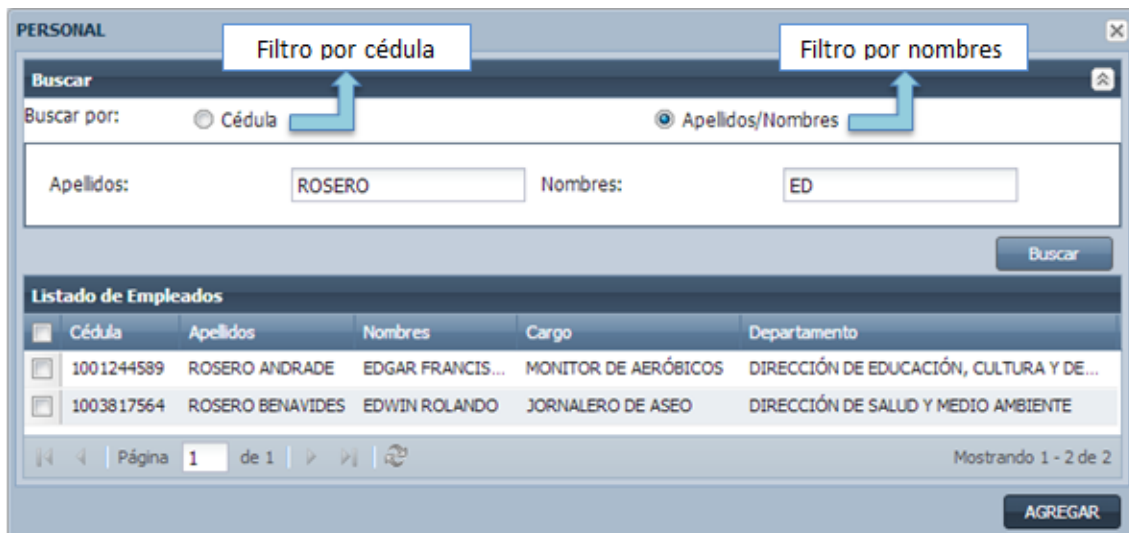


Figura 82: Diagrama Entidad Relación de las tablas usadas para el registro del personal

Fuente: Propia

Figura 83: Grid en ventana de registro para agregar personal responsable

Fuente: Propia



**Figura 84: Ventana para realizar búsqueda de empleados del GAD-I**

**Fuente: Propia**

**Tareas:**

5. Programar el método para agregar un empleado como parte del personal responsable de inspección.
6. En el grid de personal responsable agregar y programar una columna de radiobutton para poder seleccionar cuál de ellos es el responsable del grupo.
7. En el grid agregar y programar otra columna de acción para poder quitar una de las personas agregadas.

Personal Responsable					
Apellidos	Nombres	Cargo	Departamento	Responsable	Acción
ACOSTA ARMAS	FERNANDO XAVIER	ASISTENTE AVAL...	DIRECCIÓN DE AVALÚ...	<input type="radio"/>	<input type="button" value="-"/>
ACOSTA GARZON	VICTOR EDUARDO	ASISTENTE FINA...	DIRECCIÓN FINANCIE...	<input checked="" type="radio"/>	<input type="button" value="-"/>
ACOSTA ARIAS	DIEGO FERNANDO	DIRECTOR DE PL...	DIRECCIÓN PLANIFIC...	<input type="radio"/>	<input type="button" value="-"/>

Agregar

**Figura 85: Grid con personal agregado y columnas de acciones**

**Fuente: Propia**

En la tarea 8 se programa el código para enviar y guardar los datos del grupo de personas responsables en el registro de la inspección y en la tarea 9 se realizan pruebas mediante la creación y registro de una inspección donde se incluya los datos del personal responsable.



## ❖ Historia N°12: Ver Informe de Inspección Realizada

En esta historia de usuario se especifican las tareas que se realizaron para la creación de un informe de inspección donde se presentan algunos datos del registro de la misma y se visualizan en un archivo PDF.

### Tareas:

1. Agregar al grid de Inspecciones Registradas una columna de acción para poder abrir el informe con datos de la inspección.
2. Crear una consulta SQL para obtener todos los datos de la inspección registrada.
3. Crear el diseño del informe usando el plugin TCPDF.
4. Programar un método en Symfony y ExtJS para visualizar los datos de la inspección en un archivo PDF.

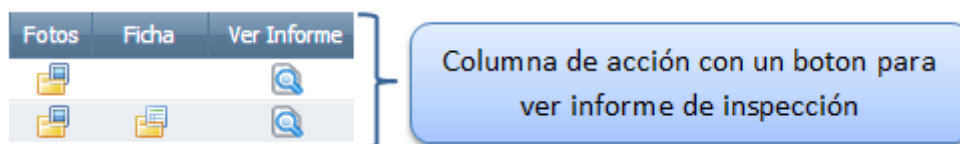


Figura 86: Columna de acción para ver el informe de inspección

Fuente: Propia

**Informe de Inspección**

**DATOS DE INSPECCIÓN**

Nro. Inspección: 52	
Fecha Inspección: 2012-12-05	Fecha Registro: 2013-07-05
Código Movillización: 9549	Fotos Registradas: SI
Ficha Registrada: SI	Inspección Aprobada: SI
Observaciones: Ninguna	

**PERSONAL PRESENTE EN INSPECCIÓN**

Cédula	Nombre	Cargo
1000034585	BOLIVAR RAMIRO ACOSTA ARIAS	DIRECTOR DE OO.PP. MUNICIPALES
1001896586	HENRY VICENTE ACELDO ARGOTI	AISISTENTE AVALUOS B

ITURRALDE CUMANDA  
ARQUITECTO

ACOSTA BOLIVAR  
RESPONSABLE INSPECCIÓN

Figura 87: Ventana para visualizar el informe de inspección

Fuente: Propia

En la tarea 5 se realizan pruebas mediante el registro de una inspección con todos los datos posibles como registro de ficha, fotos, datos del vehículo de movilización, datos de las personas presentes en la inspección al predio y luego comprobar si los mismos son visualizados correctamente en el informe PDF.

### **3.10.5 Iteración 5 (Módulo Registro de Inspecciones en el Móvil)**

En esta iteración se describen las historias de usuario realizadas para la creación de una aplicación para registrar las inspecciones desde un dispositivo móvil basándose en el módulo de registro de inspecciones realizado para un cliente de escritorio.

#### **❖ Historia N°13: Registro de Inspecciones en Dispositivo Móvil**

Al presentarse la necesidad de mejorar la rapidez del registro de inspecciones en esta historia de usuario se describen las tareas realizadas para la creación de una aplicación similar al módulo de la iteración 4 con la diferencia que en este todo se realiza desde un dispositivo móvil para agilizar los trámites con inspección.

#### **Tareas:**

1. Estudio e investigación del framework Sencha Touch 2 para el desarrollo de aplicaciones móviles.
2. Crear aplicación de Inspecciones en Symfony y Sencha Touch.
3. Agregar conexión hacia la base de datos de imágenes en el archivo schema.yml.
4. Crear módulo para el registro de inspecciones.
5. Crear controlador, models y stores para las vistas de Sencha Touch.
6. Crear menús y diseño del sistema en Sencha Touch.

En las siguientes imágenes se visualiza primero (*Figura 88*) la pantalla inicial para ingresar al sistema GAD-I móvil donde se puede loguear cualquier usuario que tenga acceso a los sistemas móviles creados, luego de realizar el ingreso correcto según el usuario logueado se puede visualizar en la pantalla de la *Figura 89* el listado de sistemas a los que tenga permisos; en este caso un usuario Arquitecto puede ingresar al sistema móvil de inspecciones y según las parroquias asignadas luego puede entrar a la pantalla de la *Figura 90* donde se encuentra el listado de parroquias en las que puede realizar un registro de inspección.



**Figura 88: Vista inicial de la aplicación móvil**

**Fuente: Propia**



**Figura 89: Vista de listado de sistemas del usuario**

**Fuente: Propia**

7. Crear una lista donde muestre las parroquias asignadas al usuario logueado.
8. Crear una lista donde al escoger una parroquia muestre las agendas creadas para realizar inspecciones.

9. Crear una lista donde al escoger una agenda se muestre los trámites pertenecientes a la misma.



Figura 90: Vista de listado de parroquias del usuario

Fuente: Propia



Figura 91: Vista con lista de agendas de una parroquia

Fuente: Propia

En la siguiente imagen se muestra la pantalla con el listado de trámites generados según la fecha y agenda seleccionada en la pantalla de la Figura 9; dentro de la pantalla de trámites se detalla la información del número de trámite, tipo de trámite y el ciudadano peticionario que está realizando la solicitud.



**Figura 92: Vista con lista de trámites de una agenda**

**Fuente: Propia**

#### **Tareas:**

10. Revisar y programar como mostrar la foto del predio en Sencha Touch.
11. Revisar y programar como ver el mapa del predio en Sencha Touch.
12. Crear una vista donde se muestre una pequeña información del predio con foto y ubicación en el mapa.

En las siguientes imágenes se muestra primero en la *Figura 93* la pantalla que se visualiza al seleccionar uno de los trámites de la lista anterior; dentro de esta pantalla se detalla información del predio como su dueño y una foto; además al dar click en el boton Ver Mapa se presenta la pantalla de la *Figura 94* donde se puede observar la ubicación del predio dentro de un mapa catastral.



Figura 93: Vista con información del trámite y foto predio

Fuente: Propia

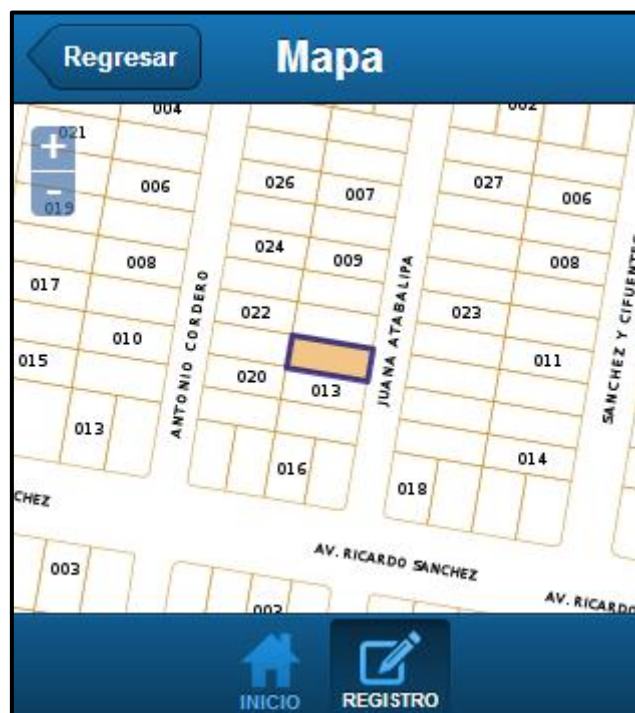


Figura 94: Vista de ubicación de predio en mapa

Fuente: Propia

**Tareas:**

13. Revisar y programar como cargar una foto en la aplicación desde el celular para posteriormente guardar en el servidor correspondiente.
14. Crear una vista donde se pueda agregar y cargar las fotos obtenidas en el registro de inspección y guardarlas.
15. Crear vista donde se visualice la ficha de inspección y programar el código para guardar sus datos.
16. Programar métodos y eventos para registrar la inspección enviando los datos al servidor e insertarlos en las tablas de la base de datos.



**Figura 95: Vista para cargar fotos de la inspección**

**Fuente: Propia**

En las siguientes imágenes podemos visualizar primero en la *Figura 96* la pantalla creada para el registro de los datos de una inspección los cuales se clasifican según las categorías existentes de la ficha correspondiente al proceso y en la *Figura 97* se presenta la pantalla inicial para el registro de una inspección donde se puede adjuntar fotos, registrar ficha y otros datos.

Cancelar **Ficha** Guardar

**DATOS PEDIALES**

AREA DE CONSTRUCCIO...	10	✕
BORDILLO	NO	▼
ACERA	SI	▼

**SERVICIOS BASICOS**

AGUA POT...	✓
ALCANTARI...	✓
LUZ ELECT...	✓

INICIO REGISTRO

Figura 96: Vista para registrar una ficha de inspección  
Fuente: Propia

Regresar **Registro** Ficha Fotos

**Datos**

Nro. Trámite:	030	
Fecha Regi...	2/1/2014	
Cod-Mov:	100	✕
Cédula Res...	1003672987	✕
Observación	Ninguna	✕
Favorable	<input checked="" type="checkbox"/>	

GUARDAR

INICIO REGISTRO

Figura 97: Vista para guardar los datos de inspección  
Fuente: Propia



En la tarea 17 se realizan pruebas mediante el registro completo de una inspección desde un dispositivo móvil en la que se incluyen fotos tomadas, datos registrados en la ficha y luego se comprueba que los datos registrados existan en las tablas de inspecciones correspondientes dentro de la base de datos.

### **3.10.6 Iteración 6 (Módulo de Reportes)**

En esta iteración se describen las historias de usuario realizadas para la creación del módulo de reportes.

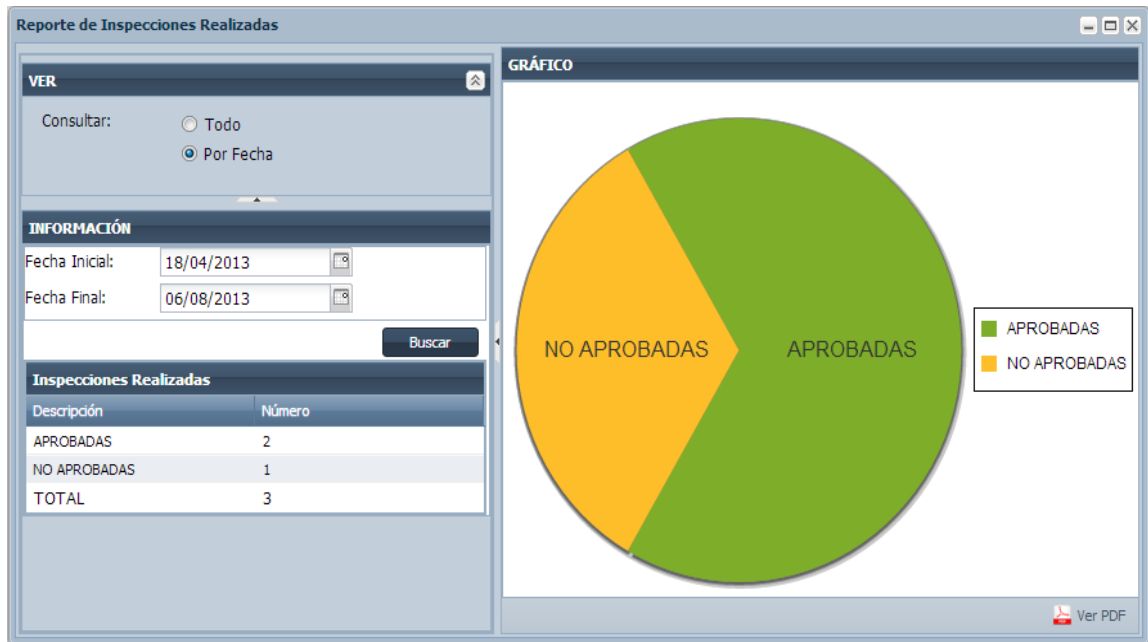
#### **❖ Historia N°14: Reportes con Gráficos Estadísticos**

En esta historia de usuario se describen las tareas para la creación de reportes basados en los registros de las inspecciones realizadas; los reportes generados son: inspecciones realizadas, inspecciones por parroquias y por procesos.

#### **Tareas:**

1. Creación del módulo Reportes de Inspecciones en la parte del servidor Symfony y el aplicativo para el cliente ExtJS.
2. Crear controlador, models y stores en el cliente ExtJS para el módulo Reportes de Inspecciones.
3. Crear ventanas para visualizar los reportes de inspecciones realizadas, por parroquias y procesos.
4. Crear consultas SQL para obtener el total de inspecciones realizadas, por parroquias y procesos, además de poder filtrar las mismas según el rango de fechas ingresadas por el usuario.
5. Programar métodos y eventos para poder visualizar los datos de las consultas en gráficos estadísticos.

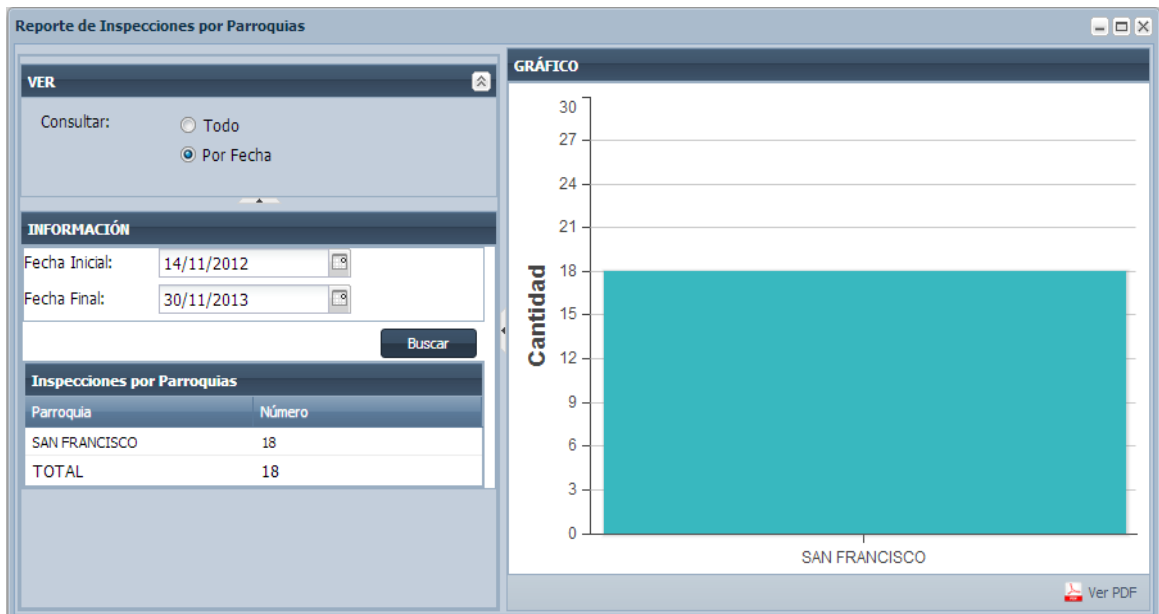
En la siguiente imagen (*Figura 98*) se visualiza la ventana para consultar el número de inspecciones aprobadas y no aprobadas en su totalidad o se puede filtrar según un rango de fechas ingresadas por el usuario; los datos son presentados en un gráfico estadístico tipo pastel que se obtiene según la información de las inspecciones registradas.



**Figura 98: Ventana de Reporte de Inspecciones Realizadas**

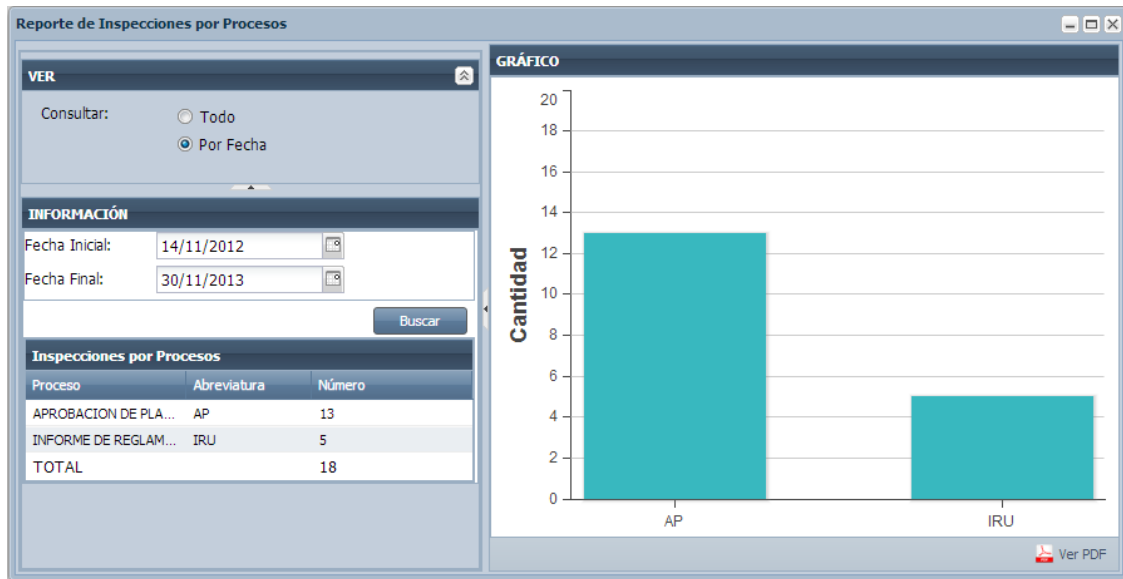
**Fuente: Propia**

En la siguiente imagen (*Figura 99*) se visualiza la ventana para consultar el número de inspecciones realizadas por parroquia en su totalidad o se puede filtrar según un rango de fechas ingresadas por el usuario; los datos son presentados en un gráfico estadístico de barras que se obtiene según la información de las inspecciones registradas



**Figura 99: Ventana de Reporte de Inspecciones por Parroquias**

**Fuente: Propia**

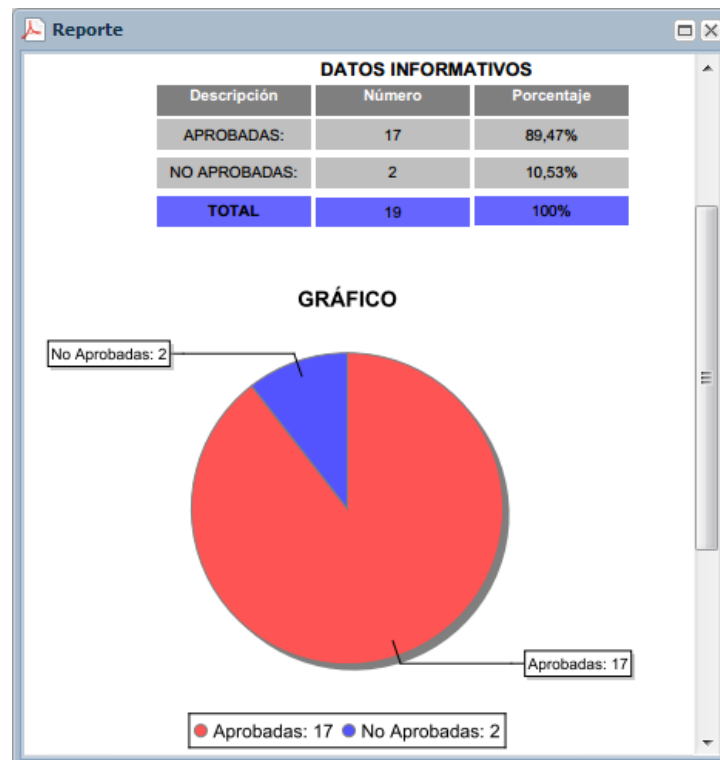


**Figura 100: Ventana de Reporte de Inspecciones por Procesos**

**Fuente: Propia**

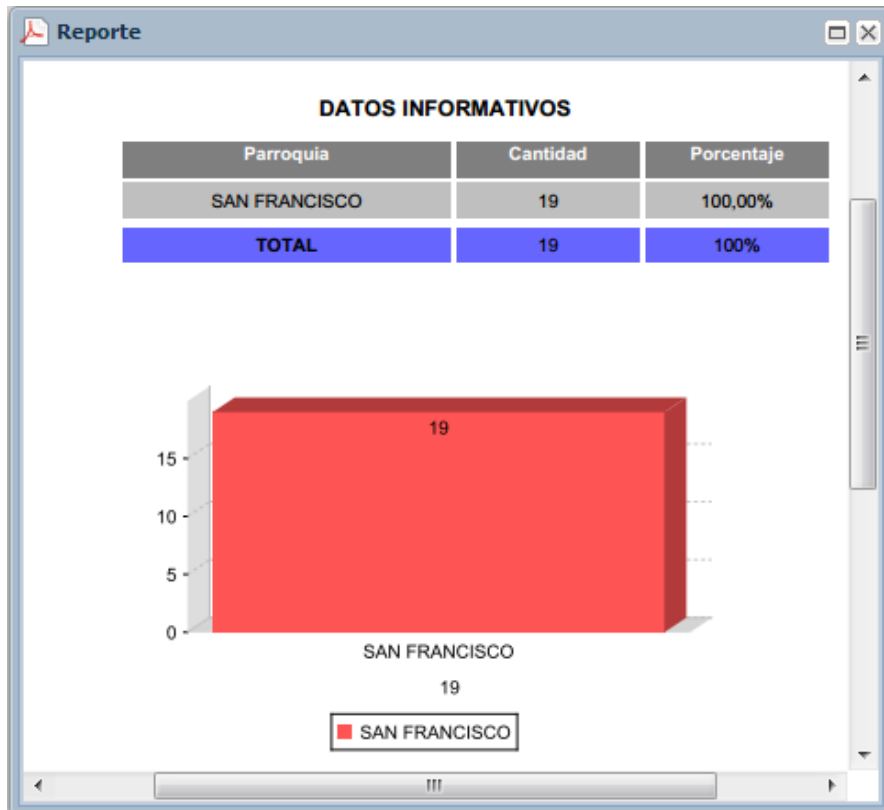
### Tareas:

6. Consultar e investigar el uso de ireport desde PHP.
7. Crear reportes en ireport con las consultas SQL.
8. Programar métodos para presentar los reportes PDF en ExtJS y Symfony.

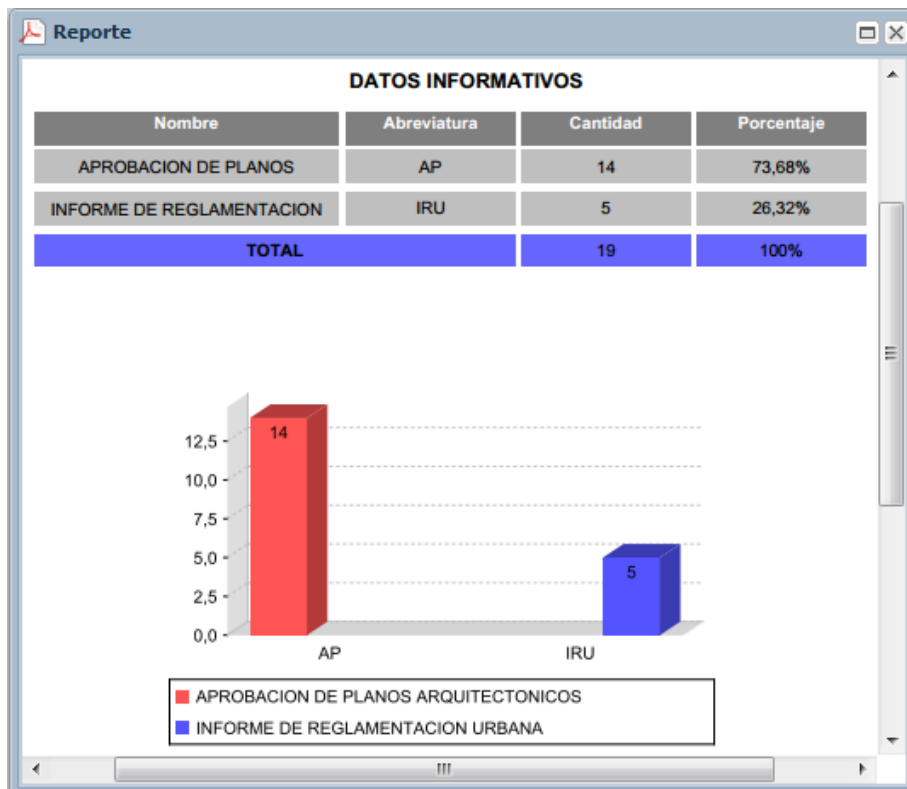


**Figura 101: Ventana con Reporte Ireport de Inspecciones Realizadas**

**Fuente: Propia**



**Figura 102: Ventana con Reporte Ireport de Inspecciones por Parroquias**  
**Fuente: Propia**



**Figura 103: Ventana con Reporte Ireport de Inspecciones por Procesos**  
**Fuente: Propia**

En la tarea 9 se realizan pruebas mediante la visualización de reportes con totales o con filtros por fechas y en la tarea 10 para finalizar se hace la presentación y entrega del Sistema de Control de Procesos de Inspección en la Dirección de Sistemas del GAD-I.

### **3.11. ELABORACIÓN DE MANUALES Y ACTA DE TERMINO DE PROYECTO**

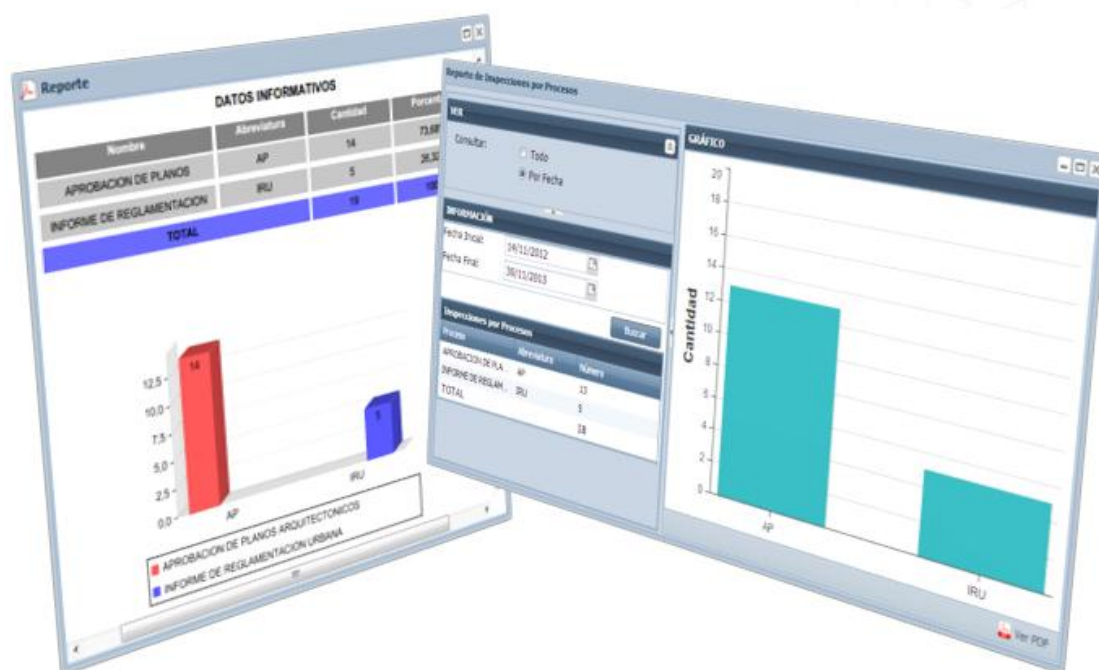
Al finalizar el desarrollo del sistema se elaboró el manual técnico que está enfocado para el uso de un especialista del área de sistemas en la cual se describe la arquitectura del sistema, su configuración, instalación, estructura del código, entre otros aspectos técnicos. **Ver Anexo N°8**

También se elabora el manual de usuario en el cual se describe el funcionamiento del sistema completo y se lo redacta en un lenguaje entendible para cualquier persona que pueda hacer uso del Sistema de Control de Procesos de Inspección. **Ver Anexo N°9**

Luego de realizar la presentación del sistema se elabora el acta de término y aceptación del proyecto en el cual se describe de forma general los módulos entregados y sus características, este documento es revisado y firmado por los miembros del área de sistemas del GAD-I. **Ver Anexo N°10**

Por último se registra el documento de caracterización de aplicaciones en el cual se describe básicamente las características principales del sistema. **Ver Anexo N°11**

# Capítulo IV: Conclusiones y Recomendaciones



## CONTENIDO DEL CAPÍTULO

- CONCLUSIONES
- RECOMENDACIONES

# CAPÍTULO IV

## CONCLUSIONES Y RECOMENDACIONES

En este capítulo final se describen las conclusiones y recomendaciones obtenidas en el desarrollo del trabajo de tesis “Sistema de Control de Procesos de Inspecciones”.

### 4.1. Conclusiones

Las siguientes conclusiones están descritas basándose en los objetivos planteados al inicio del proyecto:

- Mediante el uso del sistema se puede lograr obtener un manejo más eficaz y organizado de las inspecciones realizadas incluyendo sus datos registrados.
- Se logró estudiar algunos de los procesos y normas relacionadas con la ejecución de las inspecciones entre las cuales está la Aprobación de Planos e Informe de Reglamentación Urbana.
- Dentro de la Dirección de Planificación el tipo de inspección más realizado está relacionado con la Aprobación de Planos Arquitectónicos en la cual se realiza una visita al predio para verificar si cumple con los requisitos y documentos entregados para dicho trámite.
- A través del apoyo de personal del área de procesos se hizo un levantamiento y organización de todos los procesos que se ejecutan en la Dirección de Planificación en las que se incluyeron las inspecciones y trámites relacionados con el mismo.
- La metodología usada para el desarrollo del sistema de inspecciones es la implementada en la Dirección de Sistemas del GAD-I la cual ha sido creada mediante la revisión y unión de 2 conocidas metodologías de desarrollo ágil como son: XP y SCRUM.
- Para el desarrollo del sistema se hizo un estudio y capacitación del framework MVC Symfony 1. 4 aunque para la parte de las vistas de igual manera se estudió y usó el framework para creación de aplicaciones RIA ExtJS 4.0.
- Al finalizar el desarrollo del sistema se realizó una presentación al personal del área de sistemas y de planificación en la cual se indicó de forma detallada el funcionamiento y uso

del mismo; además de que se hace entrega de los manuales técnicos y de usuario que sirven de apoyo para el personal involucrado.

- Uno de los objetivos que se quería cumplir es poder informar al ciudadano que fecha va ser visitado su predio para realizar una inspección; esto se pudo cumplir principalmente mediante la creación del módulo Agenda de Inspecciones en el cual se determinan y generan las inspecciones creando una ruta según la ubicación de los predios en el mapa y enviando correos a los ciudadanos consultando la fecha asignada a su parroquia en el módulo Horarios de Inspección.
- Se determinó que para casi todos los trámites dentro de la Dirección de Planificación las inspecciones son similares por lo que no se crearon 3 módulos diferentes para Aprobación de Planos, IRU y Trabajos Varios como se había determinado inicialmente.
- También al presentarse el inconveniente de no tener claro un documento donde se registren los datos de inspección se decidió crear el módulo Fichas de Inspecciones en el cual un usuario administrador puede estructurar una ficha por proceso o tipo de trámite con los datos necesarios para registrar.
- Otro módulo que estaba determinado a desarrollarse era el de Administración de Usuarios pero debido a la existencia de un esquema en la base de datos del GAD-I donde se registra todo lo relacionado a los usuarios, permisos, menús, entre otros datos y además de la creación del Sistema Integrado GAD-I en el cual dentro se tiene acceso a una o más aplicaciones según los permisos dados a un usuario se determinó la no creación de este módulo.
- Luego de conocer más a fondo sobre el proceso de inspecciones a través del levantamiento de procesos que se realizó y la ejecución de un cuestionario a una parte de los arquitectos de Planificación se determinó la necesidad de crear los módulos Agenda y Registro de Inspecciones.
- Un módulo o aplicación extra creada es la de Registro de Inspecciones desde un dispositivo móvil la cual ayuda a disminuir el uso de papel y el tiempo para que el ciudadano continúe o finalice con el trámite respectivo, esto es porque se tiene la opción de registrar los datos desde y en el mismo momento de realizar una inspección a los predios.
- El sistema de inspecciones tiene una relación directa con el Sistema de Gestión de Trámites ya que es en el donde se crea un trámite y luego los arquitectos seleccionan los que



necesitan inspección que posteriormente son visualizados en el módulo Agenda de Inspecciones.

#### **4.2. Recomendaciones**

- Se podría crear otro módulo del sistema en el cual directamente se creen trámites con inspección para el uso del mismo en otros departamentos del GAD-I.
- Para disminuir el tiempo de desarrollo en ExtJS se recomienda el uso de un IDE en la cual se pueda crear rápidamente las interfaces de usuario y así el programador pueda dedicarse mayormente a la lógica de negocio.
- Para el uso del sistema y aplicaciones ExtJS 4 es recomendable hacer uso de navegadores en versiones más actuales para su normal ejecución.
- Debido a que la aplicación móvil no es una aplicación nativa y se ejecuta en un navegador web dependiendo directamente del uso de internet se recomienda hacer uso de dispositivo con paquete de datos para tener acceso al mismo.
- Otra recomendación sobre la aplicación móvil es investigar más sobre el uso de PhoneGap que tiene como principal característica convertir una aplicación móvil no nativa como lo hace Sencha Touch a una aplicación nativa para instalar y ejecutar en dispositivos con Android o iPhone sin depender obligatoriamente del internet.
- Se debe socializar al ciudadano sobre el nuevo sistema y envío de correos electrónicos comunicando la fecha de inspección al predio para evitar la acumulación de gente en la Dirección de Planificación consultando sobre dicha información.
- Antes de iniciar un proyecto de software se debe consultar y visualizar directamente con las personas encargadas de realizar el proceso relacionado para tener claro lo que se quiere automatizar y con esto evitar que se presenten muchos cambios a futuro.

# BIBLIOGRAFÍA

## Libros:

Kosmaczewski, A. (2013). *Sencha Touch 2 Up and Running*. Sebastopol (California): O'Reilly Media.

Martínez, R. (2010, Octubre 02). *Sobre PostgreSQL*. Retrieved Febrero 08, 2013, from PostgreSQL-es: [http://www.postgresql.org.es/sobre\\_postgresql](http://www.postgresql.org.es/sobre_postgresql)

Parra, A. (2012). *Guía Práctica de JavaScript*. Madrid: Anaya Multimedia.

Pérez, D. (2007, Octubre 26). *¿Qué son las base de datos?* Retrieved Febrero 09, 2013, from Maestros del Web: <http://www.maestrosdelweb.com/principiantes/¿que-son-las-bases-de-datos/>

Potencier, F., & Zaninotto, F. (2010). *Symfony, la guía definitiva*.

Ramsey, P., & Martín, M. (2012, Abril 04). *Manual de PostGIS de Paul Ramsey traducido al Español*. Retrieved Febrero 08, 2013, from PostGIS: <http://postgis.refractor.net/documentation/postgis-spanish.pdf>

Sabana Mendoza, M. (2010). *PHP 5 con PostgreSQL*. Lima: Grupo editorial Megabyte.

## Tesis:

Rea, A. (29 de Junio de 2012). Tesis: Normas de Control Interno emitidas por la Contraloría. Ibarra, Ecuador.

Torres Arcos, C. M. (2010). *Tesis Generación Dinámica de Reportes basado en Ireport y JasperReport*. Quito: Escuela Politécnica Nacional.

## Publicaciones en Línea:

Bruno, R. (12 de Octubre de 2013). *Manual de PHP-JRU*. Recuperado el 12 de Octubre de 2013, de PHP-JRU: <http://php-jru.sourceforge.net/docs/manual-PHP-JRU.pdf>

Bruno, R. (13 de Octubre de 2013). *Proceso de PHP-JRU*. Recuperado el 13 de Octubre de 2013, de Blog Robert Bruno: <http://robertbruno.wordpress.com/2008/06/25/php-jasperreport-utils-php-jru>

CodeMania. (20 de Octubre de 2013). *ExtJS*. Recuperado el 20 de Octubre de 2013, de Codemania: <http://codemania.cubava.cu/2013/09/que-es-extjs>

Doctrine. (15 de Noviembre de 2013). *Características de Doctrine*. Recuperado el 15 de Noviembre de 2013, de Wikipedia: [http://es.wikipedia.org/wiki/Doctrine\\_\(PHP\)](http://es.wikipedia.org/wiki/Doctrine_(PHP))

- EcuRed. (16 de Octubre de 2013). *Características de Ireport*. Recuperado el 16 de Octubre de 2013, de EcuRed: <http://www.ecured.cu/index.php/IReport>
- Foundation, A. (21 de Noviembre de 2013). *Apache Tomcat*. Recuperado el 21 de Noviembre de 2013, de Apache Tomcat: <http://tomcat.apache.org/>
- GeoExt. (15 de Noviembre de 2013). *¿Qué es GeoExt2?* Recuperado el 15 de Noviembre de 2013, de GeoExt: <http://geoext.github.io/geoext2>
- Higuera, S. (15 de Noviembre de 2013). *OpenLayers*. Recuperado el 15 de Noviembre de 2013, de Manual de OpenLayers: <http://openlayers.bicimap.es/manualOpenLayers.html>
- Manso, M., & Ballari, D. (15 de Octubre de 2013). *El Archivo Map*. Recuperado el 15 de Octubre de 2013, de IDEE: [http://www.ideo.es/resources/presentaciones/JIDEE07/ARTICULOS\\_JIDEE2007/Articulo1-Anexo-archivo-map.pdf](http://www.ideo.es/resources/presentaciones/JIDEE07/ARTICULOS_JIDEE2007/Articulo1-Anexo-archivo-map.pdf)
- MapServer. (12 de Noviembre de 2013). *MapServer*. Recuperado el 12 de Noviembre de 2013, de OSGeo Live: [http://live.osgeo.org/es/overview/mapserver\\_overview.html](http://live.osgeo.org/es/overview/mapserver_overview.html)
- Meléndez, C. (20 de Octubre de 2013). *Sencha Touch*. Recuperado el 20 de Octubre de 2013, de Maestros del Web: <http://www.maestrosdelweb.com/editorial/aplicaciones-moviles-sencha-touch>
- Moncayo, D. (15 de Octubre de 2013). *Web Map Services*. Recuperado el 15 de Octubre de 2013, de Repositorio UPS: <http://www.dspace.ups.edu.ec/bitstream/123456789/56/11/Capitulo5.pdf>
- MyGnet. (17 de Octubre de 2013). *Funcionamiento de JasperReport e iReport*. Recuperado el 17 de Octubre de 2013, de MyGnet: <http://mygnet.net/articulos/java/301>
- Pacheco, N. (12 de Noviembre de 2013). *¿Qué es Doctrine?* Recuperado el 12 de Noviembre de 2013, de Doctrine Project Team: [http://sf2-es.net16.net/\\_downloads/Doctrine2ORM.pdf](http://sf2-es.net16.net/_downloads/Doctrine2ORM.pdf)
- Paszniuk, R. (13 de Noviembre de 2013). *Frameworks*. Recuperado el 13 de Noviembre de 2013, de Programación: <http://www.programacion.com.py/noticias/infografia-los-mejores-frameworks-de-php>
- Perez, M. (22 de Octubre de 2013). *¿Qué es ORM?* Recuperado el 22 de Octubre de 2013, de Blog de Manel Pérez: <http://manelperez.com/programacion/que-es-doctrine-orm/>
- PHP-JavaBridge. (15 de Octubre de 2013). *Características de PHP/Java Bridge*. Recuperado el 15 de Octubre de 2013, de PHP-JavaBridge: <http://php-java-bridge.sourceforge.net/pjb>
- PHP-JRU. (15 de Octubre de 2013). *¿Que es PHP/Java Bridge?* Recuperado el 15 de Octubre de 2013, de PHP-JRU: [ftp://ftp.heanet.ie/disk1/sourceforge/p/ph/php-jru/legacy/manual\\_de\\_uso\\_PHP-JRU.pdf](ftp://ftp.heanet.ie/disk1/sourceforge/p/ph/php-jru/legacy/manual_de_uso_PHP-JRU.pdf)

Symfony Es. (18 de Octubre de 2013). *¿Qué es Symfony?* Recuperado el 18 de Octubre de 2013, de Symfony Es.: <http://www.symfony.es/que-es-symfony>

Uazuay. (15 de Noviembre de 2013). *Características de JavaScript*. Recuperado el 15 de Noviembre de 2013, de Manual de JavaScript: [http://www.uazuay.edu.ec/estudios/sistemas/lenguaje\\_iii/MAnualJavaScript/caracteristicas.htm](http://www.uazuay.edu.ec/estudios/sistemas/lenguaje_iii/MAnualJavaScript/caracteristicas.htm)

Villa, C. (15 de Noviembre de 2013). *Fundamentos de ExtJS*. Recuperado el 15 de Noviembre de 2013, de Quizzpot: <http://www.quizzpot.com/courses/fundamentos-de-ext-4>

Wikispaces. (20 de Noviembre de 2013). *Servidor HTTP Apache*. Recuperado el 20 de Noviembre de 2013, de CONCEPTOS-REDES: <http://conceptos-redes.wikispaces.com/Servidor+Web>

Wikispaces, A. (21 de Noviembre de 2013). *Características de Tomcat*. Recuperado el 21 de Noviembre de 2013, de Apache Software Foundation: <http://apachefoundation.wikispaces.com/Apache+Tomcat>

# ANEXOS

**Anexo N°1:** Formato del Documento de Proyecto Tecnológico

**Anexo N°2:** Formato del Documento de Acta de Reunión

**Anexo N°3:** Formato del Manual de Usuario

**Anexo N°4:** Formato del Manual Técnico

**Anexo N°5:** Proyecto Tecnológico

**Anexo N°6:** Actas de Reuniones

**Anexo N°7:** Encuestas realizadas para obtener requerimientos sobre inspecciones

**Anexo N°8:** Manual Técnico

**Anexo N°9:** Manual de Usuario

**Anexo N°10:** Acta de Término de Proyecto de Software

**Anexo N°11:** Documento de Caracterización de Aplicación