



UNIVERSIDAD TÉCNICA DEL NORTE

FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES

RESUMEN EJECUTIVO

TEMA:

“SISTEMA DE CONTROL DE PROCESOS DE INSPECCIONES PARA LA DIRECCIÓN DE PLANIFICACIÓN DEL GAD-I”

Autor: Edison Raúl Rivadeneira Terán

Director: Ing. Irving Reascos

Ibarra – Ecuador

2014

RESUMEN GENERAL

El presente trabajo está basado en el desarrollo de un sistema para el control de procesos en la Dirección de Planificación del GAD-I, 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 y agilizar el proceso de 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 y SCRUM donde su principal ventaja es el trabajo directo y continuo con los usuarios del 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. Además 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.

INTRODUCCIÓN

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.

Misión GAD-I

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 GAD-I

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.

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.

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.

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

- Estudiar los procesos que se llevan a cabo para ejecutar las inspecciones y control de construcciones.
- Analizar los tipos de inspección que se ejecutan.
- Contribuir con el análisis y diseño de los procesos.
- Realizar una reingeniería de procesos para poder automatizarlos, organizarlos y disminuir el tiempo de ejecución.
- Estudiar y aplicar una metodología de desarrollo rápido para la elaboración del sistema.
- Capacitación, aprendizaje y uso del framework Symfony como herramienta de desarrollo del sistema.
- Capacitar al personal involucrado dentro de los proceso de Inspecciones.
- Informar al ciudadano la fecha de inspección al predio.

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.

Alcance

En base a la información que debe ser almacenada, controlada y gestionada por el sistema se determino la creación de 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

1. 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 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. [1]

❖ Tomcat

Tomcat funciona como un contenedor de servlets desarrollado bajo el proyecto Jakarta en la Apache Software Foundation e 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. [2]

1.1.2. Base de Datos

❖ PostgreSQL

PostgreSQL es un sistema de gestión de bases de datos objeto relacional, distribuido bajo licencia BSD y con su código fuente disponible libremente. Es el sistema de gestión de BDD de código abierto más potente del mercado. [3]

❖ PostGIS

Es una extensión al sistema de base de datos objeto-relacional PostgreSQL y permite el uso de objetos GIS. Con PostGIS podemos usar todos los objetos que aparecen en la especificación OpenGIS como puntos, líneas, polígonos, multilíneas, multipuntos, y colecciones geométricas para la creación de mapas. [4]

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. [5]

❖ **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 se convierte en unas páginas más dinámicas. [6]

1.1.4. Frameworks

❖ **Symfony**

Es un framework PHP que facilita y optimiza el desarrollo de las aplicaciones web. 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. Además separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web. [7]

❖ **Ext JS 4.0.7**

Es una biblioteca de JavaScript para el desarrollo de aplicaciones web interactivas usando tecnologías como AJAX, DHTML y DOM. Fue desarrollada por Sencha. [8]

❖ **Sencha Touch 2**

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, Android 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. [9]

1.1.5. Mapas

❖ **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. [10]

❖ **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. [11]

❖ **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. [12]

❖ **OpenLayers**

Es una biblioteca de Javascript de uso libre para acceder, manipular y mostrar mapas en páginas web. Proporciona una API 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 agregado controles diversos como zoom, medida de distancias y muchas otras herramientas. [13]

❖ **GeoExt2**

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. [14]

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. [15]

❖ **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. [16]

❖ **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. [17]

❖ **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. Además 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. [18]

1.2. Metodología de Desarrollo

1.2.1. Definición

La metodología GAD-I usada es de rápido desarrollo y está constituida por las metodologías XP y SCRUM, de las cuales se ha tomado los procesos más eficientes de cada una, con el fin de obtener proyectos más eficientes y en corto tiempo.

Los documentos que forman parte de la metodología de desarrollo de software son los siguientes:

❖ **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.

❖ **Acta de Reunión**

Documento que contiene los temas tratados en una reunión de trabajo que puede ser con gente dentro del área de sistemas o con los usuarios donde el objetivo obtener requerimientos para el desarrollo del sistema y luego especificar los compromisos adquiridos que dieron como resultado de la reunión.

❖ **Historias de Usuario**

Documento que contiene las peticiones del usuario final sobre el sistema informático a desarrollarse.

❖ **Pila de Producto**

Documento que enlista todas las historias de usuario determinando la prioridad de cada una de ellas.

❖ **Pila de Iteración**

Documento que contiene las tareas a ser realizadas en base a las historias de usuario y detallando en un cronograma los tiempos determinados para la realización de las mismas.

❖ **Manual de Usuario**

Documento enfocado principalmente para el usuario final en el cual se describe detalladamente la funcionalidad del sistema en cada uno de sus módulos.

❖ **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.

❖ **Acta de Término de Proyectos**

Documento donde se detalla que se ha dado por finalizado las actividades del proceso de desarrollo de la aplicación y es aprobado por las personas a las que se les entrega el proyecto.

❖ **Caracterización de Aplicaciones**

Documento donde se especifica el tipo de aplicación que se ha desarrollado con algunas de sus características detalladas en resumen.

CAPÍTULO II

2. PROCESO DE INSPECCIONES

En el presente capítulo se identifica y describe el proceso ejecutado dentro del Departamento 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

- Director Planificación
- Técnico OO.PP
- Arquitecto/a Planificación

- Secretaria OO.PP/ Planificación

2.4. Descripción del Proceso de Inspección

A continuación se describe los pasos que se realizan en el proceso de inspección:

- Recibir la documentación y verificar el requerimiento de inspección.
- Buscar en el mapa la ubicación de los predios a inspeccionar.
- Elaborar el cronograma y ruta de inspecciones según zonas o conveniencia.
- Solicitar el equipo topográfico para cumplir las inspecciones.
- Verificar la lista de inspecciones a realizarse y preparar el equipo y materiales necesarios para ejecutar la inspección.
- Elaborar solicitud de transporte para movilización del personal.
- Si la inspección no es para un levantamiento topográfico o planímetro realizar la inspección visual del predio y revisión de parámetros técnicos de acuerdo a ordenanzas y normas.
- Si la inspección es para un levantamiento topográfico o planímetro ver los procedimientos

que se deben realizar para la ejecución.

- Si la jornada de inspecciones no ha terminado movilizarse al siguiente predio según el cronograma para ejecutar la inspección solicitada.
- Si la jornada de inspecciones a terminado, procesar datos y elaborar informe de la inspección.

CAPÍTULO III

3. 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.

A continuación se enlista los roles que han formado parte del desarrollo del sistema:

- Usuario
- Director TIC
- Responsable de Software
- Analista de Sistemas Informáticos
- Programador

3.2. 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.

Además se realizó una encuesta a los Arquitectos para obtener información sobre las fichas donde se registran los datos de la inspección.

3.3. Desarrollo de Historias de Usuario

En esta sección se describen 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.

3.4. 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.

3.4.1. Arquitectura del Sistema Integrado

La arquitectura del Sistema Integrado GAD-I, está basada principalmente en el patrón de diseño MVC, 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.

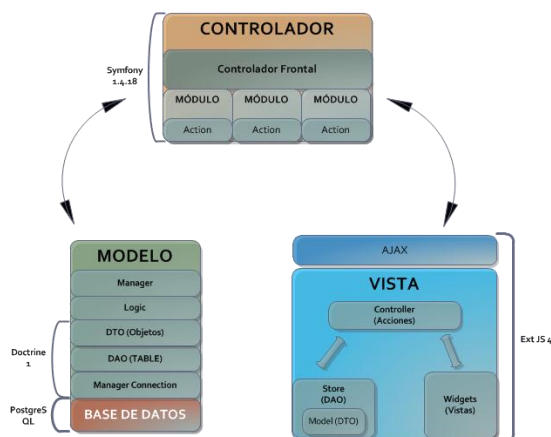


Figura 1: Arquitectura del Sistema GAD-I

Modelo: Esta capa mediante el uso del ORM Doctrine permite la comunicación y obtención de datos a través de las clases DTO creadas a partir de las tablas de la base de datos y el uso de los DAO para la ejecución de consultas SQL y DQL.

El manager 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.

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.

3.4.2. Core de la vista del Sistema Integrado

El Core del Sistema Integrado GAD-I relacionado 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 esta cuenta con varias aplicaciones que tienen funciones diferentes dependiendo de los permisos del usuario logueado.

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

Imi Ajax Connect: Permite la comunicación entre la vista del cliente y la lógica que se encuentra en el servidor (Controlador Frontal).

Api Application: Es la instancia principal del Sistema Integral a nivel de la vista que

permite la gestión de los controladores y vistas creadas en la ejecución de la aplicación, es encargada de eliminar instancias y listeners activos al 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 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 capas: Windows Manager y Módulo.

3.4.3. Módulos del Sistema de Inspecciones

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

Usuario Administrador: Es el encargado de crear los Horarios de Inspección y las Fichas de Inspección por proceso.

Usuario Arquitecto/Inspector: Es el encargado de generar, registrar, consultar y obtener reportes de las inspecciones.

A continuación se describen los módulos desarrollados del sistema de inspecciones.

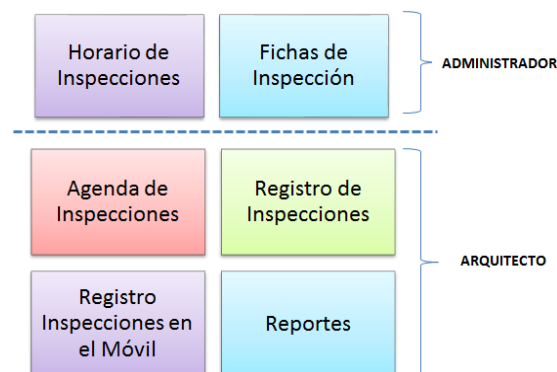


Figura 2: Módulos del Sistema de Inspecciones

Módulo Horario de Inspecciones: Módulo donde 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 Ficha de Inspección: En este módulo se crean fichas de inspección por proceso o tipo de trámite el cual contiene categorías y datos que pueden ser simples, selección única y múltiple. Esta ficha sirve para registrar los datos obtenidos al momento de realizar la inspección.

Módulo Agenda de Inspecciones: En este módulo se visualizan los trámites seleccionados a realizar una inspección;

además se hace uso de los mapas donde se visualiza la ubicación de los predios y se crea una ruta que sirve de apoyo al momento de realizar las inspecciones.

Módulo Registro de Inspecciones: En este módulo con entorno web se registran los datos de las inspecciones 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 realizar un registro rápido de los datos obtenidos de la inspección; esto es posible ya que fue creado para ejecutarlo en dispositivos móviles y usarlo en el lugar 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 y puede ser usado por los arquitectos como apoyo en el cumplimiento de indicadores existentes.

3.4.4. Descripción Funcional del Sistema de Inspecciones

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

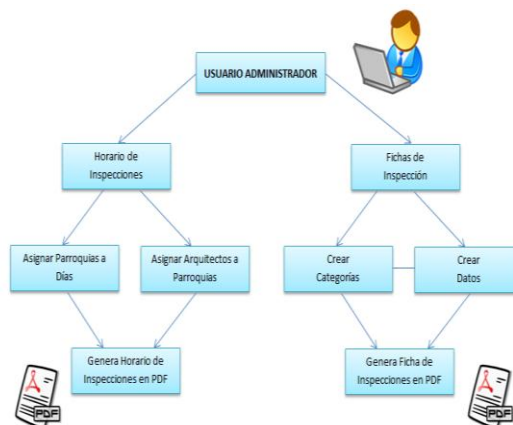


Figura 3: Funcionalidad del Sistema en base al Usuario Administrador

Administrador: El usuario administrador tiene acceso al 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.

Este usuario también tiene acceso al módulo de Fichas de Inspecciones donde puede visualizar y crear nuevas fichas por tipo de trámite en las cuales se pueden crear o agregar categorías y a su vez en estas crear los datos que va contener la ficha que es usada para el registro de

inspecciones. Esta ficha puede ser consultada y visualizada en formato PDF.

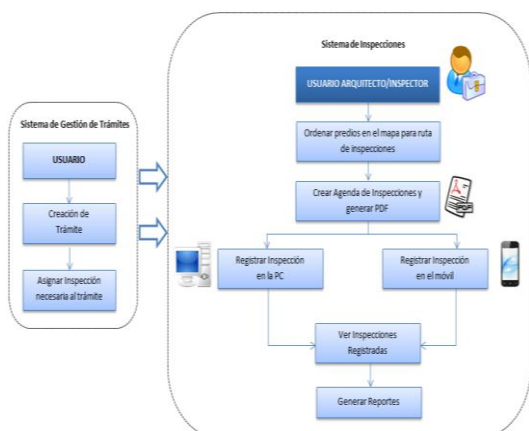


Figura 4: Funcionalidad del Sistema en base al Usuario Inspector

Inspector: Este usuario tiene acceso al módulo Agenda de Inspecciones que tiene una relación directa con el Sistema de Gestión de Trámites ya que es donde se crean los trámites con la información necesaria.

Luego de crear los trámites con inspección, se reciben y visualizan en el Módulo Agenda de Inspecciones y con la ayuda de un mapa de ubicación de los predios, el usuario crea la ruta y agenda que da como resultado generar un archivo PDF con la información básica de los trámites y se envía un e-mail a los ciudadanos informando la fecha asignada el cual se va a visitar el predio.

Después de haber generado las inspecciones se realiza su registro, que

puede ser a través de un entorno web y móvil; en estos se registran los datos necesarios obtenidos en la inspección y por ultimo este usuario puede consultar las inspecciones realizadas y generar reportes estadísticos con los datos de las inspecciones realizadas.

3.4.5. 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:

PILA DE PRODUCTO		
ID	Nombre de Historia de Usuario	Prioridad
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

CAPÍTULO IV

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

3.4.6. Desarrollo de Historias de Usuario

Luego de la creación de todas las historias de usuario se crean las pilas de iteración donde se describen las tareas a realizarse para cumplir con las pruebas de aceptación detalladas en las historias; dichas tareas se organizan y visualizan en forma cronología. Finalmente se realiza una captura de las pantallas creadas en el aplicativo que sustenta la realización de las tareas asignadas en cada historia de usuario.

4. CONCLUSIONES Y RECOMENDACIONES

En este capítulo final se describen las conclusiones y recomendaciones obtenidas en el desarrollo del “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 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.

5. BIBLIOGRAFIA

- [1] Wikispaces, «Servidor HTTP Apache,» 20 Noviembre 2013. [En línea]. Available: <http://conceptos-redes.wikispaces.com/Servidor+Web>. [Último acceso: 20 Noviembre 2013].
- [2] A. Wikispaces, «Características de Tomcat,» 21 Noviembre 2013. [En línea]. Available: <http://apachefoundation.wikispaces.com/Apache+Tomcat>. [Último acceso: 21 Noviembre 2013].
- [3] R. Martínez, «Sobre PostgreSQL,» 02 Octubre 2010. [En línea]. Available: http://www.postgresql.org.es/sobre_postgresql. [Último acceso: 08 Febrero 2013].
- [4] P. Ramsey y M. Martín, «Manual de PostGIS de Paul Ramsey traducido al Español,» 04 Abril 2012. [En línea]. Available: <http://postgis.refractions.net/documentation/postgis-spanish.pdf>. [Último acceso: 08 Febrero 2013].

- [5] M. Sabana Mendoza, PHP 5 con PostgreSQL, Lima: Grupo editorial Megabyte, 2010. [Último acceso: 15 Octubre 2013].
- [6] A. Parra, Guía Práctica de JavaScript, Madrid: Anaya Multimedia, 2012.
- [7] S. E. «¿Qué es Symfony?,» 18 Octubre 2013. [En línea]. Available: <http://www.symfony.es/que-es-symfony>. [Último acceso: 18 Octubre 2013].
- [8] C. Villa, «Fundamentos de ExtJS,» 15 Noviembre 2013. [En línea]. Available: <http://www.quizzpot.com/courses/fundamentos-de-ext-4>. [Último acceso: 15 Novimebre 2013].
- [9] A. Kosmaczewski, Sencha Touch 2 Up and Running, Sebastopol (California): O'Reilly Media, 2013.
- [10] MapServer, «MapServer,» 12 Noviembre 2013. [En línea]. Available: http://live.osgeo.org/es/overview/mapserver_overview.html. [Último acceso: 12 Noviembre 2013].
- [11] M. Manso y D. Ballari, «El Archivo Map,» 15 Octubre 2013. [En línea]. Available: http://www.ideo.es/resources/presentaciones/JIDEE07/ARTICULOS_JIDEE2007/Articulo1-Anexo-archivo-map.pdf. [Último acceso: 15 Octubre 2013].
- [12] D. Moncayo, «Web Map Services,» 15 Octubre 2013. [En línea]. Available: <http://www.dspace.ups.edu.ec/bitstream/123456789/56/11/Capitulo5.pdf>.
- [13] S. Higuera, «OpenLayers,» 15 Noviembre 2013. [En línea]. Available: <http://openlayers.bicimap.es/manualOpenLayers.html>. [Último acceso: 15 Noviembre 2013].
- [14] GeoExt, «¿Qué es GeoExt2?,» 15 Noviembre 2013. [En línea]. Available: <http://geoext.github.io/geoext2>. [Último acceso: 15 Noviembre 2013].
- [15] C. M. Torres Arcos, Tesis Generación Dinámica de Reportes basado en Ireport y JasperReport, Quito: Escuela Politécnica Nacional, 2010.
- [16] MyGnet, «Funcionamiento de JasperReport e iReport,» 17 Octubre 2013. [En línea]. Available: <http://mygnet.net/articulos/java/301>. [Último acceso: 17 Octubre 2013].
- [17] R. Bruno, «Manual de PHP-JRU,» 12 Octubre 2013. [En línea]. Available: <http://php-jru.sourceforge.net/docs/manual-PHP-JRU.pdf>. [Último acceso: 12 Octubre 2013].
- [18] PHP-JRU, «¿Que es PHP/Java Bridge?,» 15 Octubre 2013. [En línea]. Available: ftp://ftp.heanet.ie/disk1/sourceforge/php/php-jru/legacy/manual_de_uso_PHP-JRU.pdf. [Último acceso: 15 Octubre 2013].