

Capítulo 1 Sistema de encuestas georreferenciadas utilizando Java Spring Framework con el modelo de vista del controlador

Chapter 1 Implementation of a georeferenced survey System with Java Spring Framework using controller view model

VALDEZ-MENCHACA, Alicia Guadalupe†*, CASTAÑEDA-ALVARADO, Sergio Raúl, CORTES-MORALES, Griselda y VÁZQUEZ-DE LOS SANTOS, Laura Cristina

Universidad Autónoma de Coahuila. Barranquilla S/N C.P. 25700 Monclova Coahuila.

ID 1^{er} Autor: *Alicia Guadalupe, Valdez–Menchaca* / **ORC ID:** 0000-0002-3494-4830, **Researcher ID Thomson:** S-4551-2018, **CVU CONACYT ID:** 292172

ID 1^{er} Coautor: *Sergio Raúl, Castañeda-Alvarado* / **ORC ID:** 0000-0002-7396-2804, **CVU CONACYT ID:** 256334

ID 2^{do} Coautor: *Griselda, Cortes–Morales* / **ORC ID:** 0000-0002-2567-7056, **CVU CONACYT ID:** 617827

ID 3^{er} Coautor: *Laura Cristina, Vázquez–De Los Santos* / **ORC ID:** 0000-0002-0291-7774, **Researcher ID Thomson:** S-6543-2018, **CVU CONACYT ID:** 615088

DOI: 10.35429/H.2019.1.1.14

A. Valdez, S. Castañeda, G. Cortes y L. Vásquez

*aliciavaldez@uadec.edu.mx

A. Marroquín, J. Olivares, P. Díaz, L. Cruz. (Dir.). Mujeres en la tecnología. Handbooks-©ECORFAN-Mexico, Queretaro, 2019.

Resumen

Los sistemas de información apoyan varios propósitos en organizaciones de cualquier tipo. A veces, los sistemas de información se utilizan para conocer la tendencia de la población sobre diversos temas; las encuestas se han convertido en un tema público de relevancia para las personas. En este proyecto se ha desarrollado e implementado un sistema de información web de encuestas georreferenciadas para conocer la intención de voto de la población de la Región Central de Coahuila. Se ha utilizado el entorno de desarrollo integrado de Netbeans con el patrón del Modelo de Vista del Controlador y el software MySQL para la gestión de la base de datos. Como resultado, la aplicación ha sido diseñada para identificar coordenadas geográficas de las áreas de interés para la aplicación de la encuesta a través de waypoints y tracks; que registran datos exactos de la ubicación del entrevistador; se utilizaron los módulos Spring Web Framework para el desarrollo completo de la aplicación. Se aplicaron aproximadamente 2800 encuestas en un lapso de 3 meses y permitiendo la grabación de la sesión de audio para cada uno de los encuestadores, de acuerdo con el área seleccionada y la sección correspondiente; con el fin de obtener datos precisos y exactos sobre las opiniones de los encuestados para su posterior análisis y gráficos.

Sistema georreferenciado, Patrón Modelo Vista de Controlador, Spring framework, Java, MySql

Abstract

Information systems support many purposes in organizations of any kind. Sometimes the information systems are used to know the tendency of the population on diverse subjects; surveys have become a public issue of relevance to people. In this project a web information system of georeferenced surveys has been developed and implemented to know the vote intention of the population of the Central Region of Coahuila. Using the Netbeans Integrated Development Environment with the Controller View Model pattern and MySQL for the database management. As a result, the application has been designed to identify geographic coordinates of the areas of interest for the application of the survey through waypoints and tracks; which record exact data of the location of the interviewer; the spring web framework modules was used for the complete development of the application. Approximately 2800 surveys were applied in a 3-month span and it allows recording of the audio session for each one of the pollsters, according to the selected area and the corresponding sectional; obtaining precise and exact data on the opinions of the respondents for subsequent analysis and graphing.

Georeferenced system, MVC Pattern, Spring framework, Java, MySql

Introducción

El software es uno de los pilares estratégicos de las organizaciones y la sociedad, ya que muchos de sus procesos, productos y servicios dependen de un alto grado de operación adecuada (Cardona, 2010). Los cambios progresivos en los paradigmas de desarrollo a través de los paradigmas clásicos, a los paradigmas modernos, se presentaron con gran importancia en la historia del desarrollo de software. En este sentido, Méndez et al., (Y. Méndez et al., 2009) sostienen que existe un avance significativo en la industria del software para proporcionar mecanismos de reutilización para aumentar la productividad del software con un aumento de la calidad, lo que dice que uno de los aspectos es importante.

Las técnicas de desarrollo de software en el presente y futuro son la usabilidad o la reutilización de código. En este sentido, Méndez (Y. Méndez et al., 2009) menciona que en el desarrollo de software, la facilidad de uso es un tema que está ganando cada vez más importancia por la capacidad de crear software en menos tiempo.

Además, (Pressman, 2010) agrega que el software debe responder cada vez más rápido a los cambios que provienen del mundo exterior, al aumentar la competencia y porque los nuevos dominios donde el software juega un papel importante no permiten demoras en esas actualizaciones. Una mayor interoperabilidad con otros sistemas también hará que los sistemas que no se actualicen de inmediato queden obsoletos y no puedan seguir formando parte de otros "sistemas de sistemas".

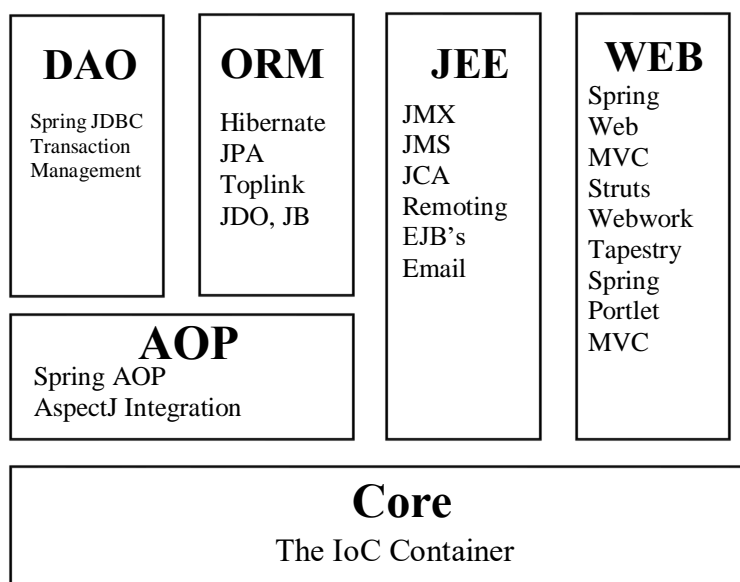
Para el análisis y desarrollo de las aplicaciones web, se han desarrollado numerosas técnicas que satisfacen esta expectativa. El patrón deseable para el desarrollo de software web es el Modelo de Vista del Controlador (MVC), que considera separar en tres elementos los componentes de un proyecto o capas a lo largo del proyecto, siendo: lógica de control (saber qué elementos tienen el proyecto y qué hacer, pero no cómo fue implementado), la lógica empresarial, know how, cómo se desarrolla la aplicación y la lógica de presentación, o cómo interactúa el usuario con la aplicación.

Otros autores analizados para la realización de esta aplicación, han utilizado sistemas de información geográfica con fines culturales, arquitectónicos y ecológicos (Galeano, 2017), mientras que otros han usado este tipo de sistemas para determinar zonas de franquicias funcionales para estaciones de servicio (Andrade, 2018) y para apoyo en el diagnóstico integral comunitario mediante una cartografía participativa con sistemas de información georreferenciados (Fagalde & Herrera, 2018).

Al implementar este patrón se obtiene: mayor calidad, mejor mantenibilidad y no iniciar de cero en un proyecto. En la capa de presentación se manejó el Framework Spring de Java, el cual permite agilizar y estilizar de una forma más sencilla la interfaz; para la programación de las dos capas restantes se necesita generar todo el código necesario, que en general es extenso y repetitivo, lo cual genera retraso al programador (Diaz, Queiruga, & Fava, 2009).

En la capa del modelo se requiere generar una clase de acceso a datos llamada DAO (Data Access Object), por cada estructura de consulta que se requiera, y una clase DTO (Data Transfer Object), con el objetivo de intercambiar la comunicación con las demás capas, ver Fig. 1.1.

Figura 1.1 Módulos del Spring Framework

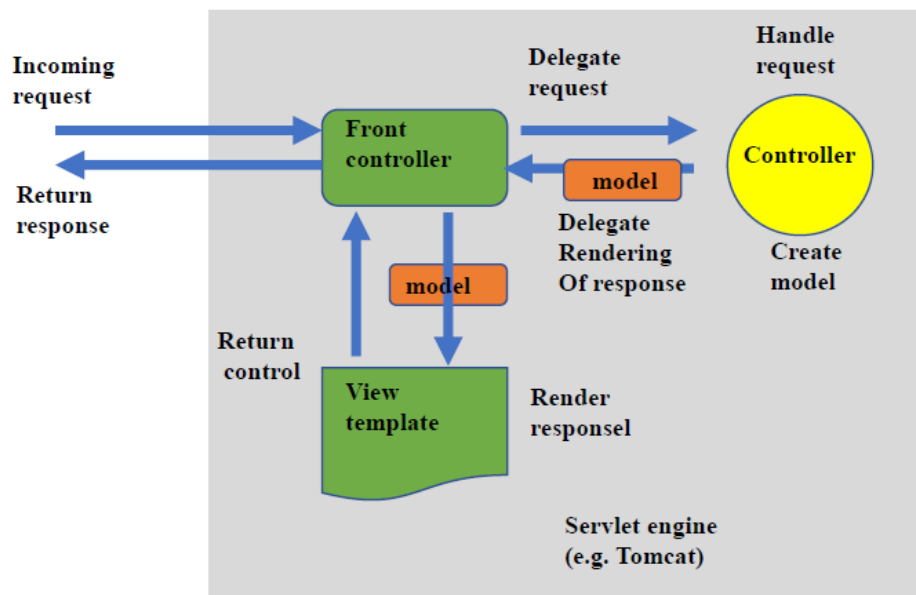


Fuente: (Johnson & Hoeller, 2018)

Para el caso del procesamiento de una petición HTTP, en Spring MVC, en el controlador Front, se manejó la clase DispatcherServlet, para servir a la petición de URL, se manejó la clase HandlerMapping, se llama al Controller que ejecuta la lógica de negocio, obtiene los resultados y los devuelve al Servlet, encapsulados en un objeto del tipo Model. Un ViewResolver se encarga de averiguar el nombre físico de la vista, que corresponde con el nombre lógico del paso anterior.

Finalmente, el DispatcherServlet redirige la petición hacia la vista, que muestra los resultados de la operación realizada, ver la Fig. 1.2.

Figura 1.2 Ejemplo del MVC



Fuente: (Santos, 2014)

Considerando estos antecedentes se cree necesario desarrollar varios autómatas que permitan la codificación de la estructura del patrón MVC, con lo que se puede disminuir el tiempo de programación de un 60 a 70%, estos valores se obtienen al evaluar el tiempo que se llevó programar una aplicación en internet en donde se manejaron alrededor de 25 tablas en un proyecto, empleando el patrón MVC sin utilizar la aplicación y después ya recurriendo a la aplicación (Trueba & Martínez, 2010).

El 10% de holgura se debe a la destreza de los programadores, cabe mencionar que todos los programadores involucrados tenían un nivel junior en el lenguaje Java. Considerando dicha evaluación se observa que hay reducción de costos por honorarios de programadores y tiempo de desarrollo, no sólo es considerable la reducción de tiempos de desarrollo para proyectos medianos, sino que a mayor tamaño del proyecto mayor es la mejora (J. Méndez, 2008).

Este capítulo está compuesto de varias secciones, siendo estas: Introducción, conceptos fundamentales, metodología, resultados, prospectiva, conclusiones y recomendaciones.

Objetivo Principal

Diseñar y generar un autómata que implemente el patrón MVC en el proyecto orientado a web en el manejo de encuestas georreferenciadas para la toma de datos de campo, utilizando GPS de Google Maps.

Objetivos Particulares

1. Diseñar el modelo del autómata que permita la codificación del patrón MVC, para recolectar información sobre la posición, longitud y área de elementos geográficos.
2. Procesar la información recolectada en el campo.
3. Desarrollo e implementación de un autómata para la homogenización de scripts SQL.
4. Diseño e implementación de una estructura dinámica que permita identificar las relaciones en el esquema de las bases de datos, su diccionario de datos y las vistas que existan.
5. Desarrollo e implementación del analizador sintáctico de scripts SQL para crear la estructura dinámica.
6. Desarrollo e implementación de algoritmos para generar las capas del patrón MVC.

Conceptos fundamentales

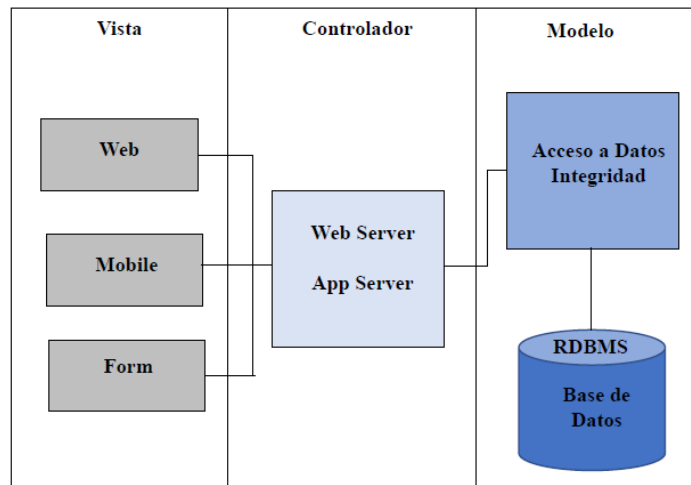
World Wide Web

El diseño de interfaces web es un tema complejo en el que no sólo intervienen procesos de diseño gráfico y programación, sino que también resultan imprescindibles aspectos de la arquitectura de la información, navegación, funcionalidad y, sobre todo, de la usabilidad (Belmonte, 2003). La ingeniería web hace referencia a las metodologías, técnicas y herramientas que se utilizan para el desarrollo de aplicaciones web complejas y de gran dimensión, en las que se apoya la evaluación, diseño, desarrollo, implementación y evolución de dichas aplicaciones. Ellas poseen determinadas características que lo hacen diferente del desarrollo de aplicaciones o software tradicional y sistemas de información.

Es importante considerar que es multidisciplinaria, aglutina contribuciones de: arquitectura de la información, ingeniería de hipermedia/hipertexto, diseño de interfaz de usuario, gráfico, usabilidad, análisis de sistemas, ingeniería de software, ingeniería de datos, indexado y recuperación de información, testeo, modelado. Así como simulación, despliegue de aplicaciones, operación de sistemas y gestión de proyectos (Y. Méndez et al., 2009).

El diseño web no es un clon o subconjunto de la ingeniería de software, aunque ambas incluyen desarrollo de software y programación, la ingeniería de la web utiliza principios de ingeniería de software (Stephens, 2015), incluye nuevos enfoques, metodologías, herramientas, técnicas, guías y patrones para cubrir los requisitos únicos de las aplicaciones, ver la Fig. 1.3.

Figura 1.3 Patrón MVC, asociado a la tecnología web



Fuente: (Johnson & Hoeller, 2018)

Arquitectura MVC

El patrón Modelo, Vista y Controlador (MVC) es el más extendido para el desarrollo de aplicaciones donde se manejan interfaces de usuarios, éste se centra en la separación de los datos o modelo, y la vista, mientras que el controlador es el encargado de relacionar a estos dos (Gallego, Parra, & Nuñez, 2012). Su principal característica es aislar la vista del modelo (Buhr, Pearce, & Casselman, 1996; Ferro, Sappia, Amo, & Liberal, 2003; Merlino, 2008). En la Fig. 1.3 se puede apreciar la separación de las tres capas y los componentes que la hacen funcional, por tener independencia entre capas, lo que hace que sea deseable para proyectos de grandes dimensiones.

Principales ventajas de usar el patrón MVC:

1. Permitir la sustitución de las interfaces de usuario.
2. Generar componentes de las interfaces.
3. Diseñar vistas simultáneas del mismo modelo.
4. Aplicar fácilmente cambios de las interfaces.

La vista y el modelo están muy acopladas, considerando el acoplamiento como el grado de interdependencia entre las unidades de software (módulos, funciones y subrutinas) de un sistema informático. En este sentido en el patrón MVC, el acceso a datos depende directamente del mismo modelo que se mapea por medio de la consulta SQL (vista), por tanto, los DTO y DAO corresponden a una estructura muy acoplada a la vista, ya que los objetos de intercambio (DTO) dependen directamente de los DAO y la generación de los mismos dependen directamente del modelo.

Software utilizado

La base del desarrollo está sustentada en Lenguaje Unificado de Modelado (UML) (Rumbaugh, Jacobson, & Booch, 2004), pero no se puede modificar un proyecto ya elaborado, dado que se basan en los diagramas UML del proyecto para generar el código del patrón MVC. En estas herramientas cuando requieren hacer cambios en el proyecto, no los pueden hacer, por no considerar retomar el proyecto desde el código ya generado sino del modelo UML. Si lo hacen tendrán que generar otro código, desechando todo el código generado anteriormente (incluyendo las vistas), en algunos casos perdiendo las correcciones, modificaciones y/o actualizaciones realizadas al código, lo que implica prácticamente la generación de un nuevo proyecto. Con lo que se trasgreden las reglas de usabilidad para el desarrollo de software.

El desarrollo de esta aplicación contempla generar el código del patrón MVC en lenguaje Java, que permita generar subproyectos o secciones de un proyecto y poder ser acoplado a un proyecto final. Con el fin de poder modificar un proyecto ya concluido (siempre y cuando se haya desarrollado con esta misma herramienta), brindando la posibilidad de expandir en algún futuro los proyectos ya generados, esto mediante la implementación de varios autómatas coordinados, que generen toda la estructura del patrón MVC necesaria de forma apropiada, uniforme y acorde con el proyecto requerido, además de la documentación inherente al proyecto. Todo esto a partir del script generado por una herramienta de modelación de base de datos.

Esto ayudara principalmente a que los desarrolladores pasen directo a la capa de negocio, alimentando la clase principal consiguiendo un incremento sustancial en la velocidad de desarrollo de cualquier proyecto con uniformidad del código obtenido, con su respectiva documentación, permitiendo actualizaciones futuras y fortalecer la usabilidad del software a futuro.

Requerimientos de hardware y software para la implementación de la aplicación

1. Para la programación de los autómatas se consideró el manejo del paradigma orientado a objetos en el lenguaje Java.
2. Software para 64 bits.
 - a. Entorno JDK de JAVA versión 2017.
 - b. Servidor de base de datos MySQL versión 2017.
 - c. Desarrollo de scripts con Workbench 2017.
 - d. Netbeans como IDE ver. 8.2.
 - e. Sistema Operativo Windows 10.
3. Hardware de Desarrollo 64 bits.
 - a. Laptop Dell Procesador Core I5 Intel.
 - b. Memoria Ram de 8 GB.
 - c. Disco Duro de 1 TB.
 - d. Pantalla de 14" HD 1280 MegaPixels.
 - e. Sistema Multimedia.
 - f. Sistema operativo Windows 10.
4. Hardware de Pruebas.
 - a. Tablet Samsung Galaxy SM-T560 32 bits con GPS.
 - i. Sistema operativo Android 4.4.4
 - ii. Memoria Ram de 4 Mb.
 - iii. Memoria de 64 Mb.
 - b. Celular HTC con GPS.
 - c. Sistema operativo Android ver. 5.2
 - d. Memoria Ram de 2 Mb.
 - e. Memoria de 54 Mb.

Tecnologías requeridas para la implementación

Spring Framework

Es un marco de trabajo formado por una serie de módulos, que se utilizan y aplican para el desarrollo de sistemas empresariales, además se proporciona una compatibilidad con otros frameworks, como, por ejemplo, EJB, JSB, Struts, tal como se muestra en la Tabla 1.1.

Tabla 1.1 Tecnologías utilizadas

Tecnología	Descripción
Java	Lenguaje de programación Orientada a Objetos.
Spring Framework	Framework de desarrollo de software en el lenguaje de programación Java.
JMS	Software para el llamado de aplicaciones de forma asincrónica en Java Message Serving.
MySQL	Base de datos relacional Open Source.
Apache	Message Broker para el manejo de mensajes de comunicación asíncrona entre aplicaciones.
QuartzScheduler	Planificador de aplicaciones Java.
NetBeans 8.2	IDE para el manejo de aplicaciones Java.

Fuente: Elaboración Propia

JMS (Java Message Service)

En el desarrollo de aplicaciones de software es muy importante el manejo de procedimientos o funciones remotas, necesarias para esta aplicación sobre todo para el manejo de contenidos que son bajados desde el GPS. En JMS, se provee la comunicación asíncrona mediante JMS, el corredor de mensajes o Message Broker, es una aplicación intermedia que se encuentra operativa en un servidor para procesar y redirigir los mensajes que le son enviados. En esta aplicación se manejó el corredor de mensajes “Active MQ”, el cual es desarrollado por la comunidad Apache, brindando un amplio soporte y documentación.

Planificador (Quartz-Scheduler)

Excelente aplicación de programación de tareas ejecutando el trabajo en cierta cantidad de tiempo en cierta hora del día, se integra al Framework de Spring permitiendo un rápido desarrollo y configuración de la aplicación.

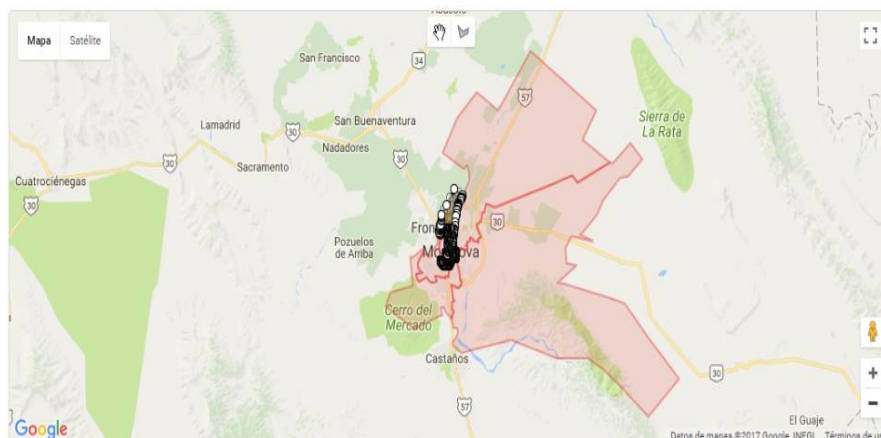
Seguridad de la Información

Se definieron los criterios siguientes para garantizar la seguridad en la información.

- Se cuenta con un Administrador cuya función principal es asignar y configurar usuarios, establecer políticas de seguridad, evaluar los logs y tomar medidas respecto al rendimiento de la base de datos derivado del manejo de archivos multimedia (audio).
- Se cuenta con personal que activa los dispositivos móviles como tabletas y celulares que deben de contar para que la aplicación de Encuestas, la cual está basada en el sistema operativo en Android pueda operar.
- Existe personal que apoya en la generación de encuestas y la asignación de tareas de la encuesta vigente para la carga en los dispositivos móviles que apoyan a los usuarios del sistema.

Metodología

Se determinan las coordenadas geográficas de un punto dado, como resultado de la recepción de información de señales generadas en Google Maps, provenientes de la clasificación de zonas y seccionales, determinadas por el Instituto Nacional Electoral (INE) en la ciudad de Monclova, Coahuila. Para fines de seguimiento de rutas, localización y monitoreo de personas bajo la cartografía determinada. En la Fig. 1.4 se muestra el mapa de la zona 1, el cual para que sea funcional con la aplicación requiere que se seleccione el área con un color determinado para que al momento de realizar las encuestas nada más se trabaje en la zona correspondiente asignada por el administrador (La plataforma permite la relación de N tipos de encuestas, N Encuestadores y 6 áreas como máximo y cada área con su seccional correspondiente de acuerdo al mapa que proporciona el INE).

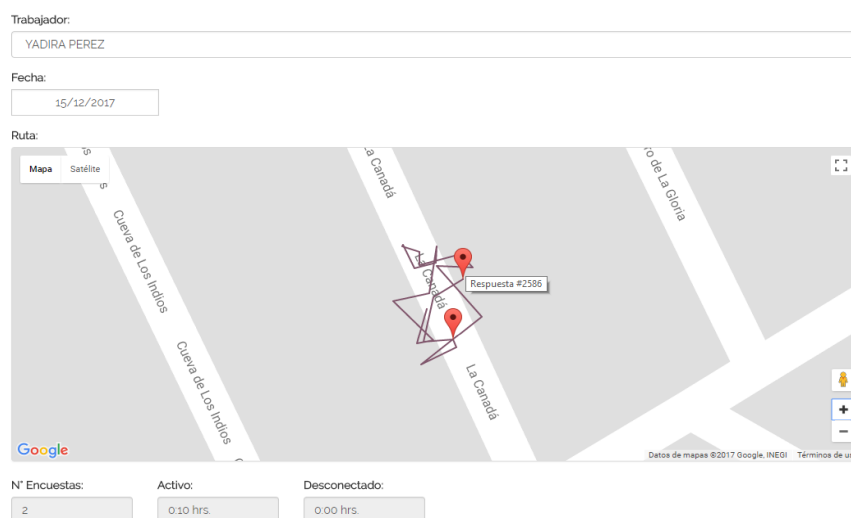
Figura 1.4 Mapa del área

Fuente: Google Maps

Algunos de los conceptos fundamentales para la funcionalidad de esta aplicación fueron los siguientes: Identificar las coordenadas geográficas, que nos permiten ubicar con exactitud un lugar en la superficie de la tierra, estos conjuntos de líneas corresponden a los meridianos y paralelos. Los meridianos corresponden a los círculos máximos que pasan por los polos, divididos en 180 meridianos al este y 180 meridianos al oeste, existen un total de 360 meridianos en total. La longitud es la distancia en grados, entre cualquier meridiano y el meridiano de Greenwich que es un punto universal de referencia, los meridianos se han trazado en intervalos de 10°. Ejemplo: La ciudad de Xalapa es de 19°32' latitud norte y 96°55' longitud oeste.

Waypoint

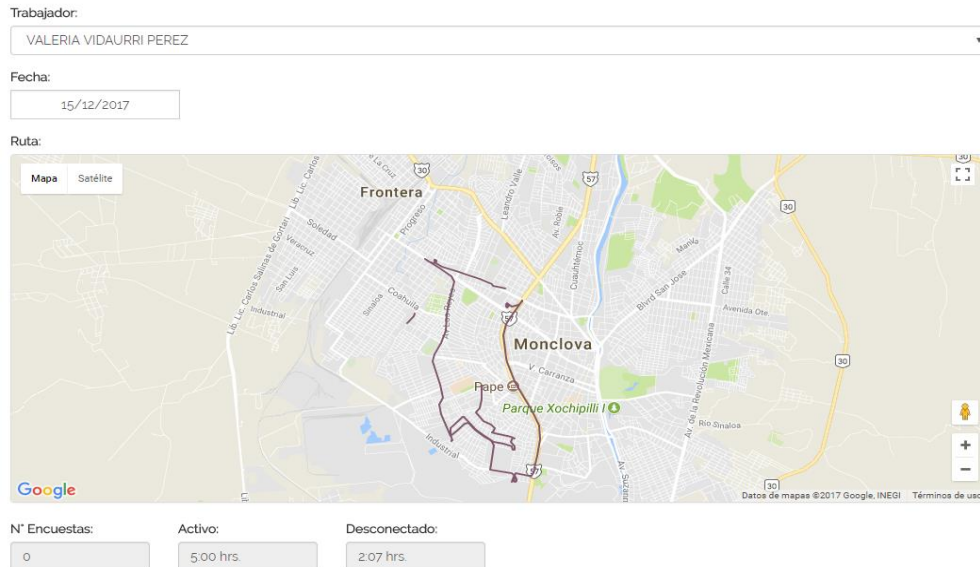
En la memoria del receptor del GPS, un waypoint contiene las coordenadas de la posición que define en que parte del mundo está localizado, en este caso aplica únicamente para la ciudad de Monclova y sus áreas aledañas. Normalmente al waypoint se le asocian valores de altitud, latitud, asigna la fecha, la hora y un comentario, en este caso relacionado con los datos del encuestador, el área, el seccional y el número de respuesta que está realizando, tal como se muestra en la Fig. 1.5.

Figura 1.5 Waypoint

Fuente: Google Maps

Track

Un track es una concatenación de waypoints, unos detrás de otros, para definir un recorrido, algunos de los GPS guardan desde 10,000 hasta 50,000. Tal como se muestra en la Fig. 1.6, donde se indica el recorrido realizado por medio del track, mostrando el recorrido realizado por la persona que va a realizar la encuesta, acorde a la fecha, la zona y los seccionales que le correspondan.

Figura 1.6 Track

Fuente: Google Maps

Diseño de la aplicación mediante Lenguaje Unificado de Modelado (UML)

Para el diseño de la aplicación se utilizó el lenguaje UML, que permitirá un entendimiento completo del proyecto, así como la generación de documentación de la arquitectura de software para futuros mantenimientos.

Diagrama de despliegue

Se usan para mostrar la configuración de los elementos del proceso en tiempo de ejecución y los componentes de software, artefactos y procesos que se encuentran en ellos, los cuales se forman por nodos y rutas de comunicación. Para el menú de opciones de la aplicación, en el apartado de carga de encuesta (Se define el tipo de encuesta y la carga de preguntas), se carga el área de trabajo a desarrollar la encuesta (ya mencionada con anterioridad), se definen los encuestadores y el tipo de encuesta a realizar, entre otros, el nodo principal sería el Message Broker, el servidor de aplicaciones de MySQL, para la base de datos principal. Además de definir la ruta, la ubicación y el rol que seguirá el usuario dentro con la nube en donde está cargada toda la aplicación para registro de trazabilidad del mapa.

Arquitectura

La arquitectura lógica de un sistema permite definir claramente la forma en que los principales componentes se deberán de desarrollar. Para el sistema de Encuestas, se definió por la arquitectura de 3 capas: Presentación (Escuchar a la gente), Service (Lógica de negocio) y DAO (Acceso a datos). Este tipo de arquitectura nos permite separar el ingreso de peticiones en audio o encuestas mediante estas capas.

Estructura de automatización para el patrón MVC

Posteriormente, una vez que se configura la estructura dinámica: las tablas, sus atributos y vistas que conformarán el proyecto o subproyecto. También, se decide si el proyecto se manejará como servicio web, proyecto web o aplicación local, enseguida se establece la conexión a la base de datos.

Diseño y construcción de la base de datos

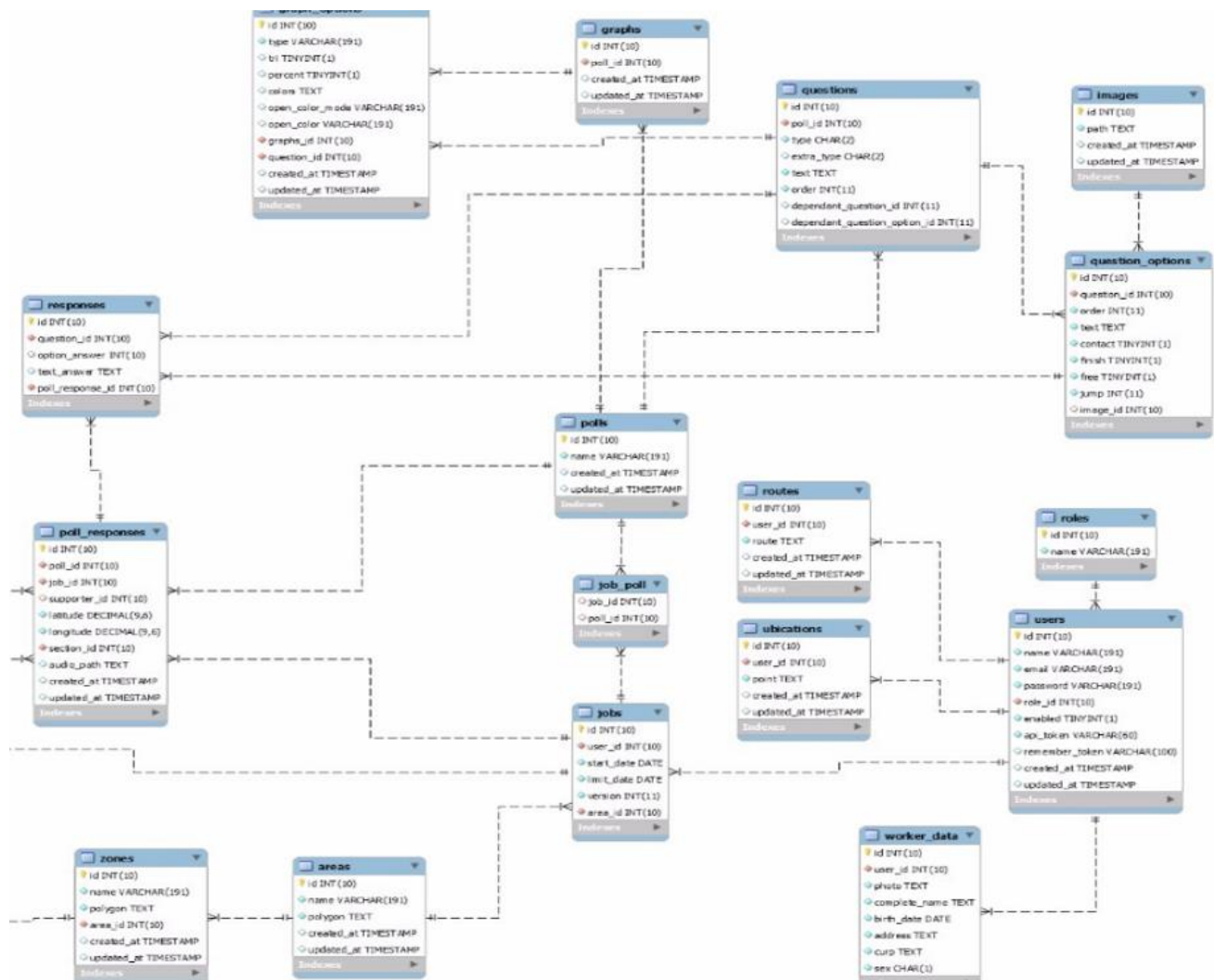
Se diseñan y posteriormente se generan los scripts requeridos para la base de datos creada físicamente con Mysql Workbench. Posteriormente se deben de homologar los archivos de SQL a un estándar predefinido, con el objetivo de sólo tener todos los elementos necesarios a ser utilizados por el autómatas. Ver Fig. 1.7. Es conveniente mencionar que las vistas SQL manejan la misma estructura de datos que las tablas.

Patrón MVC

Para cumplir con el último objetivo de crear el código del modelo MVC en Java, se parte de los generadores que hacen varios recorridos sobre la estructura dinámica transformada para extraer el esquema de la base de datos y crear los objetos DTO mapeándolos dentro de Java, de igual forma se procede para realizar las consultas y actualización del DAO, mientras que se toman los nombres de las tablas para los métodos de las clases Facade y Delegate.

Para la capa del modelo se requiere generar una clase DAO como mínimo por cada estructura de consulta que se requiera, y una clase DTO, como objeto de intercambio para la comunicación con las demás capas. Para la capa de control Service, se agrega una clase Facade por cada una de los DAO creados y se agrega una clase Delegate que agrupe a todas las Facade creadas, para posteriormente ligar el Delegate generado a la capa de vista con su Framework correspondiente. Para lograr la creación apropiada de las capas se consideró integrar un archivo XML de configuración por cada generador de código (gd, gf, ga, gt), y unas librerías de compatibilidad que bien pueden ser configuradas desde la interfaz o tomarse en su configuración predeterminada, que permite generar cada una de las capas del patrón MVC, tal como se muestra en la Fig. 1.8.

Figura 1.7 Modelo de datos



Fuente: Elaboración Propia en software MySQL

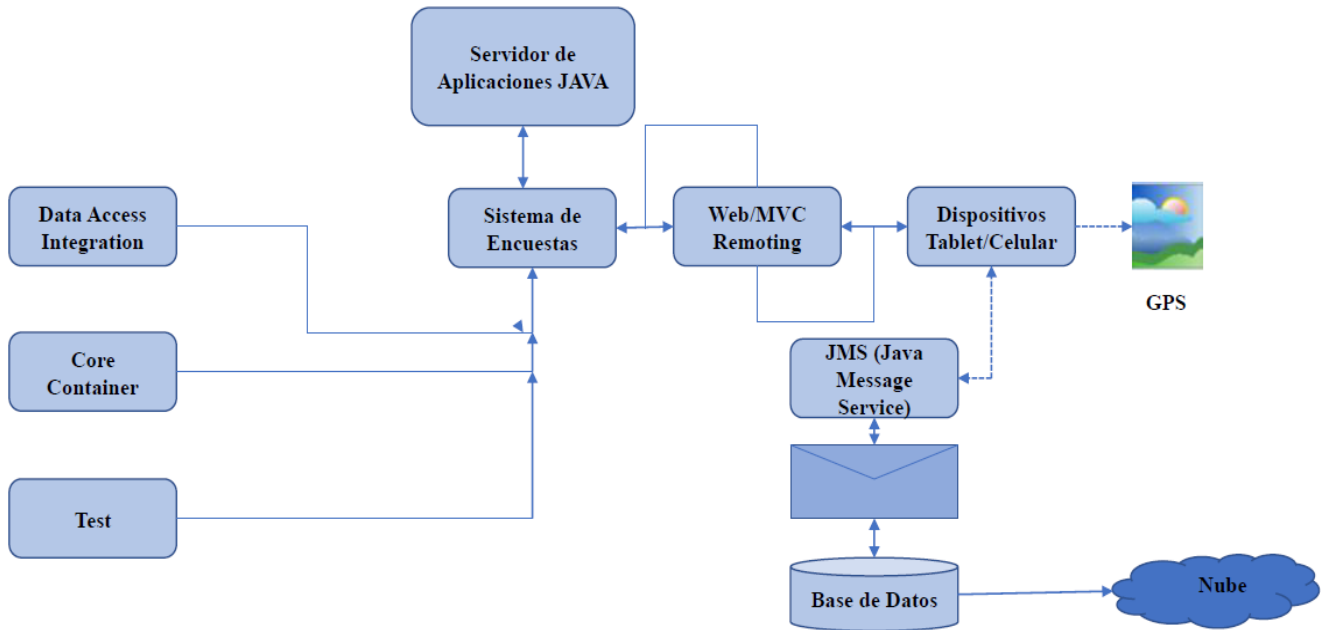
Resultados

Como resultado se obtuvo una aplicación que gestiona el control de Encuestas y permite la grabación de la sesión en audio por cada uno de los encuestadores, de acuerdo al área seleccionada y al seccional correspondiente. Esta propuesta implicó el trabajo con cero conocimientos de MVC de Spring, se manejó un total de 25 tablas, para generar el código del modelo MVC, se llevó un tiempo aproximado de 3 meses, desde que inició el proyecto.

Considerando lo anterior la aplicación de Encuestas MVC, tiene un alto desempeño, con lo cual se puede ahorrar meses de programación en la estructura del patrón MVC y asegurar una estructura homogénea la cual está de momento libre de errores y operando con un total de más de 2,800 encuestas realizadas, tal como se muestra la Fig. 1.9.

Otro aspecto importante es la capacidad de reutilización de código ya que el sistema de Encuestas MVC permite agregar subproyectos a un proyecto existente, con lo cual la usabilidad del proyecto se extiende completamente, ya que con la interfaz se pueden modificar algunas propiedades y elegir sólo lo necesario para la sección de un subproyecto.

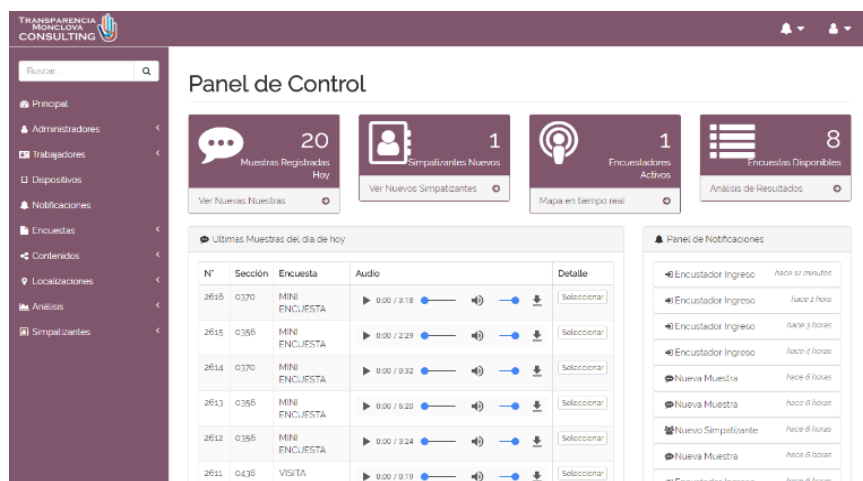
Figura 1.8 Patrón MVC del proyecto



Fuente: Elaboración Propia

De tal forma que si se tiene un proyecto existente se pueden generar los objetos necesarios para cada una de las capas del patrón MVC, acoplándose a lo ya generado con anterioridad, esto sólo ocurrirá si la primera parte se generara con esta misma herramienta.

Figura 1.9 Pantalla principal de la aplicación de encuestas



Fuente: Pantalla del sistema

Prospectiva

El software es uno de los pilares estratégicos de las organizaciones y de la sociedad, debido a que muchos de sus procesos, productos y servicios dependen en un alto grado de su correcto funcionamiento (Cardona, 2010). En la historia del desarrollo de software se han presentado con gran trascendencia cambios progresivos en los paradigmas de desarrollo pasando por los paradigmas clásicos, hasta paradigmas modernos. Al respecto Pressman (2010) argumenta que hay un avance significativo en la industria de software al proporcionar mecanismos de reutilización para aumentar la productividad de software con un aumento en la calidad, con lo que afirma que uno de los aspectos importantes en las técnicas de desarrollo de software actuales y futuras es la usabilidad o reutilización del código, el trabajo expuesto (Encuestas MVC) cumple con este mandato ya que considera agregar nuevas funcionalidades a los proyectos sin la necesidad de reescribir todo el software.

Al respecto Y. Mendez (2009) menciona que, en el desarrollo de software, la usabilidad es un tema que está cobrando importancia cada vez mayor por la capacidad de crear software en menos tiempo. Asimismo, Hölz y Wirsing (2008) agregan que el software deberá responder cada vez más rápido a los cambios que provengan del mundo exterior, por la creciente competencia y porque los nuevos dominios en los que el software juegue un papel relevante no permitirán demoras para realizar esas actualizaciones. La creciente interoperabilidad con otros sistemas también hará que los sistemas que no se actualicen inmediatamente queden obsoletos y no puedan seguir formando parte de otros “sistemas de sistemas”.

Los métodos de desarrollo que se usen, sobre todo los orientados a la evolución de aplicaciones existentes, deberán tener en cuenta estas crecientes presiones. Otro aspecto que se contempla en el desarrollo del Encuestas MVC expuesto en este trabajo es la automatización como un segundo precepto que deben considerar los desarrolladores de software de vanguardia para considerar trabajos futuros, al respecto otros autores (Hölz & Wirsing, 2008) comentan que en temas de diseño y programación como refactorización automática es importante y que el potencial que tienen estas técnicas en la automatización de actividades de los sistemas de información, ya que esto tendrá un fuerte impacto en lidiar con la creciente complejidad de la disciplina.

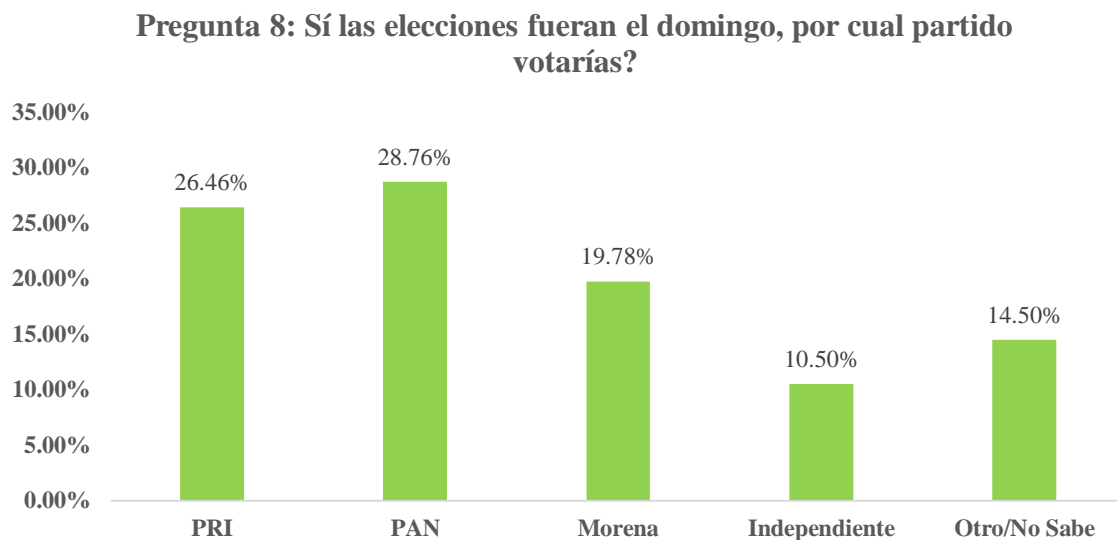
Por ejemplo, las problemáticas de líneas de productos y producción ciertamente ofrecen oportunidades para la aplicación de técnicas automatizadas y semiautomatizadas para el procesamiento de artefactos del proceso de desarrollo que permitan extraer, abstraer y refactorizar los artefactos disponibles y ayudar en la construcción de una familia de productos. Este mismo autor también considera que un problema clásico de la ingeniería de software es: cómo usar modelos de mayor nivel de abstracción para poder generar de manera automática o semiautomática distintas aplicaciones, además de lograr un análisis más oportuno sobre propiedades de lo que se está construyendo y facilitar las tareas de verificación.

Desde este punto de vista para la automatización del patrón MVC que se emplea en Encuestas MVC se justifica, ya que permite la automatización para el desarrollo de software.

Conclusiones

Encuestas MVC genera código del patrón MVC en lenguaje java, Encuestas MVC considera la usabilidad indeterminada del proyecto por lo que se pueden anexar subproyectos sin perder la continuidad, siempre y cuando el proyecto inicial se haya empezado con esta herramienta. El uso de Encuestas MVC, garantiza la consistencia de la programación del patrón MVC. Con Encuestas MVC se reduce el tiempo de programación que se empleará en la programación que soporta el patrón Encuestas MVC es de fácil utilización lo cual lo hace muy versátil con un impacto inmediato.

El software Encuestas MVC, está a la vanguardia para la creación de sistemas de información para la web y trabajar de manera distribuida, situación que lo hace ser una herramienta que se estará utilizando en proyectos futuros, además se está trabajando para mejorar el proyecto para dar soporte a características de futuras versiones en la creación de software; a las personas que utilizan la aplicación se les ha proporcionado una capacitación en el uso del software, la Fig. 1.10 muestra el análisis final realizado a partir de un grupo de personas encuestadas.

Figura 1.10 Análisis realizado a 2,056 encuestas

Fuente: Pantalla del sistema

Se reducen los tiempos de programación, pero se debe considerar la programación de la lógica de negocios y diseño gráfico de la página Web.

Recomendaciones

Como se ha expuesto en este capítulo, este sistema de información de Encuestas georreferenciadas, es un sistema complejo que muestra la información sobre la localización física del encuestador, así como grabaciones en tiempo real del audio y de la información proporcionada por los encuestados; por lo que representa una gran ventaja en la aplicación de este tipo de sistemas en tiempo real.

El desarrollo utilizando el framework Spring de Java, así como las librerías de conexiones de datos necesarias para el manejo de los datos ha representado un desafío en la integración de la información obtenida, para su posterior análisis, graficación y toma de decisiones.

Las futuras investigaciones sobre este tipo de aplicaciones son muy variadas, ya que se pueden utilizar los sistemas georreferenciados en múltiples campos como seguridad en general de personas y cosas, sistemas sanitarios, información geográfica de bienes, seguridad comunitaria, entre otros.

Referencias

Andrade, A. (2018). *Determinación de zonas de franquicia funcionales a partir de la georreferenciación de estaciones de servicio por medio de una herramienta SIG, y el diseño de una app móvil para su consulta*. Ingeniería, Universidad Distrital Francisco José de Caldas, Bogotá, Colombia.

Belmonte, J. (2003). *Ingeniería de Usabilidad, aplicado al desarrollo de un portal Web administrado dinámicamente*. Licenciatura, Universidad Politécnica de Cataluña España, Cataluña, España.

Buhr, R., Pearce, T., & Casselman, R. (1996). Design Patterns with Use Case Maps: A Case Study in Reengineering an Object-Oriented Framework. In D. o. S. a. C. Engineering (Ed.). Ontario, Canada: Carleton University.

Cardona, S. (2010). Perspectivas de la Ingeniería en Computación. In D. d. Ingeniería (Ed.). Colombia: Universidad de Quindío.

Díaz, F., Queiruga, C., & Fava, L. (2009). *Struts y Server Faces – Cara a Cara*. Paper presented at the XI Workshop de Investigadores en Ciencias de la Computación, Argentina.

Fagalde, M., & Herrera, M. (2018). Cartografía participativa, SIG e IDE, herramientas para la creación de un Diagnóstico Integral Comunitario: el caso de Barrio Tongui – Lomas de Zamora – Buenos Aires. In C. d. A. T. y. Estadística (Ed.). Buenos Aires, Argentina: Dirección Nacional del Observatorio Argentino de Drogas.

Ferro, V., Sappia, R., Amo, C. P. d., & Liberal, M. (2003). Herramienta de Gestión Automatizada para la elaboración de proyectos en entornos de formación. In D. d. E. y. Telecomunicaciones (Ed.). España: Universidad del País Vasco.

Galeano, E. (2017). *Sistema De Información Geográfica (SIG) del Patrimonio Cultural, Arquitectónico Y Ecológico Del Municipio De Zipaquirá Como Potencial Turístico*. Ingeniería, Universidad Distrital Francisco José de Caldas, Bogotá, Colombia.

Gallego, R., Parra, N., & Nuñez, P. (2012). AR-Learning: interactive book based on augmented reality, application in teaching. *Revista Monográfico*, 1(8), 74-89.

Hölz, M., & Wirsing, M. (2008). State of the Art for the Engineering of Software-Intensive Systems. In L. Maximilians-Universität (Ed.). Munich, Germany: Ludwig Maximilians-Universität.

Johnson, R., & Hoeller, J. (2018). Spring Framework – Reference Documentation 3.0.

Méndez, J. (2008). *Análisis comparativo de las plataformas J2EE, y .Net. aplicado a servicios Web*. Ingeniería de Sistemas, Universidad de Pamplona, Pamplona, Colombia.

Méndez, Y., Collazos, C., Granollers, T., Villegas, M., Ruiz, A., & Giraldo, W. (2009). Modelo para la creación de un laboratorio de usabilidad. *Revista Avances en Sistemas e Informática*, 6(2), 211-217.

Merlino, H. (2008). Patrones de diseño en vistas adaptables. *Ingeniería de Software*, 8(2), 30-35.

Pressman, R. (2010). *Ingeniería del Software. Un enfoque práctico* (Vol. 7a. Edición). México, D.F.: McGraw-Hill.

Rumbaugh, J., Jacobson, I., & Booch, G. (2004). *Unified Modeling Language Reference Manual, The (2nd Edition)*: Pearson Higher Education.

Santos, M. (2014). *Diseño de un módulo de carga de pagos en entidades públicas mediante mensajería con spring framework* (Vol. 15).

Stephens, R. (2015). *Beginning Software Engineering*. Canada: Wiley Publishers.

Trueba, E., & Martínez, R. (2010). Sistema de Información para el manejo de datos académicos administrativos de la DES, Texcoco. In C. U. UAEM (Ed.). México, D.F.: Universidad Autónoma del Estado de México.