IX Jornadas de Ciencia y Tecnología
de Facultades de Ingeniería del NOA
Santiago del Estero, 3 y 4 de Octubre de 2013

Transporte Urbano Georreferenciado

José V. Zapana[1], Elsa D. Ramirez, Maria L. Aybar & Felipe F. Mullicundo

(1) Facultad de Ingeniería, Universidad Nacional de Jujuy.
jose@josezapana.com

RESUMEN: Google maps y Google Earth son herramientas que nos brindan excelentes imágenes y nos permite georreferenciar de forma gráfica, navegar virtualmente por el mundo, conocer caminos tomando datos de referencia como nombre de calles o marcar un camino o ruta. Ya no se trata solamente de geodatos limitados a los especialistas de las geociencias y SIG. Ahora la georreferenciación tiene un impacto sociológico puesto que se realiza sobre todos los contenidos sociales presentes en el mundo. La propuesta del presente proyecto muestra cómo aplicar la georreferenciación para el recorrido de transporte urbano haciendo uso de las herramientas que nos brinda Google maps, en base a esto se desarrolló un prototipo que permite, entre otras cosas, seleccionar origen y destino de un pasajero y realizar una consulta georreferenciada del circuito de la línea de transporte elegida. Otro aspecto importante a destacar es el uso de Scrum como metodología ágil de desarrollo, implementada para llevar adelante el proyecto.

1 INTRODUCCIÓN

En la tarea de seleccionar nuevas áreas de estudio que puedan ser aplicadas a los contenidos mínimos de las cátedras de la facultad, se consideraron diferentes ámbitos, tales como las áreas relacionadas a tecnologías móviles, domótica, etc. Luego de ver las tendencias tecnológicas actuales, se decidió la incursión en un área en la cual no se ha avanzado en el ámbito académico, se trata del "Desarrollo de aplicaciones web con la implementación de georreferenciación".
La georreferenciación (Ref. 1), en primer lugar, posee una definición tecnocientífica aplicada a la existencia de las cosas en un espacio físico, mediante el establecimiento de relaciones entre las imágenes de raster o vector sobre una proyección geográfica o sistemas de coordenadas. Consiste en asignar mediante cualquier medio técnico apropiado, una serie de coordenadas geográficas procedentes de una imagen de referencia conocida, a una imagen digital de destino. Por ello la georreferenciación se convierte en central para los modelados de datos realizados por los Sistemas de Información geográfica (SIG).

2 OBJETIVOS

El desarrollo del proyecto tiene los siguientes objetivos:

- Investigar sobre diferentes tecnologías para arquitecturas web robustas con N capas aplicables a SIG.
- Implementar una API para el manejo de mapas en el desarrollo de la aplicación Web.
- Crear de un prototipo funcional de una aplicación Web para el recorrido de transporte urbano de pasajeros en la ciudad de San Salvador de Jujuy.

3 FUNDAMENTO TEÓRICO

Social

En los últimos años nuestra ciudad tuvo un crecimiento de población significativa, lo que implicó también que haya un crecimiento en las distintas áreas relacionadas directamente, tal es el caso de los comercios, turismo, transporte, etc. La información sobre estos recorridos NO es de acceso público para los usuario que no utilizan frecuentemente el servicio (turistas, vecinos de otros sectores, etc.), ocasionando inconvenientes a la hora de buscar el medio de transporte adecuado ya que en la actualidad NO existe ninguna aplicación disponible, en el ejido municipal de San Salvador de Jujuy, capaz de resolver este problema.

Tecnológico

La aplicación web se desarrollada utilizando una arquitectura robusta incluyendo diversos frameworks que permitirán aplicar patrones de diseño tales como MVC, DAO, IoC, DTO, etc. e integrarlos con tecnologías de georreferenciación mediante un diseño de N capas que permitirán

lograr una arquitectura robusta y aplicable a otras aplicaciones que hagan uso de la georreferenciación.

Académico

El desarrollo del prototipo web georreferencial nos permitirá incursionar en una nueva área de desarrollo de sistemas de software, para luego trasladar los conocimientos y experiencia adquirida en los contenidos mínimos de las cátedras de nuestra unidad académica. Por otro lado, también nos permitirá incursionar en el desarrollo de este tipo de aplicaciones para otro tipo de dispositivos tales como los dispositivos móviles con diferentes tipos de interfaces.

4 DESCRIPCIÓN DEL SISTEMA DE TRANSPORTE GEORREFERENCIADO

El sistema de consulta de transporte urbano georreferenciado es un prototipo que permitirá a los usuarios de las líneas de colectivos de transporte urbano, conocer el recorrido de las diferentes líneas de colectivos de la ciudad de San Salvador de Jujuy. Básicamente posee 2 módulos uno para la configuración del sistema y otro módulo para la consulta de los recorridos mediante el uso de mapas, mediante esta interfaz el usuario podrá seleccionar puntos y consultar de las líneas disponibles entre los orígenes y destinos seleccionados.

Alcance

El prototipo desarrollado está sujeto a las siguientes características:

- Interfaz gráfica restringida al ejido municipal de la ciudad de San Salvador de Jujuy, se incluyen inicialmente para 5 líneas de transporte.
- Permite la selección del transporte más adecuado a sus necesidades, brindándole las alternativas al recorrido deseado, en frecuencia de tiempo y cercanía de su origen y destino.
- Módulo de Administración para la configuración de los recorridos de transporte, dicho módulo se encuentra restringido sólo para usuarios autorizados.
- El acceso para la consulta por parte de los usuarios se realizará a través de un sitio web público.
- Estadísticas de las líneas más consultadas por los usuarios

5 METODOLOGÍA DE DESARROLLO

El desarrollo del proyecto se realiza siguiendo las pautas de la metodología SCRUM, aprovechando las capacidades que nos brinda esta metodología, seleccionando los elementos necesarios y adecuados para este proyecto. Entre las principales características de Scrum aplicables al proyecto citamos:

- La organización de tareas con prioridades, determinado inicialmente el desarrollo de las que poseen mayor relevancia para el proyecto.
- Permite Inspeccionar rápida y periódicamente las etapas de desarrollo del software en la que se estará trabajando.
- Promueve la comunicación entre los miembros del equipo y el cliente.
- Otra característica significativa de Scrum, es que permite gestionar los cambios. Si ocurriesen cambios en los requisitos del producto o tecnológicos, los mismos se podrán contemplar de manera sencilla agregándolos al Backlog.

Roles Identificados

En base estos puntos se definieron los siguientes roles para el equipo de desarrollo:

- Product owner: se consiguió la colaboración de un representante de la municipalidad de San Salvador de Jujuy, con quien se definió el alcance y la lógica de los recorridos del transporte urbano de pasajeros.
- Scrum master: es el encargado de asegurar que el equipo siga las pautas definidas por la metodología y colabora para lograr el éxito del equipo en cada sprint definido. Este rol fue asumido por el director del proyecto.
- Team: es el equipo de desarrollo encargado de llevar adelante las tareas definidas en cada sprint. Conformada por los demás integrantes del grupo de desarrollo.

Actividades realizadas

Las actividades realizadas durante el desarrollo del prototipo fueron:

- Product backlog: lista priorizada de requisitos, fue definida entre el product owner y el equipo de desarrollo, la lista de requisitos básicamente se basó en el alcance definido para el proyecto.
- Sprint: son las iteraciones de trabajo con duraciones acotadas de tiempo, para el desarrollo de producto se consideró 4 semanas como la mejor opción para la duración de cada ciclo.

- Daily Scrum: es una reunión diaria breve del equipo que permite se comentan básicamente los avances, e inconvenientes del sprint en curso. De esta información pueden resultar, de ser necesario, una re planificación de las tareas del sprint que se está desarrollando. La daily scrum ha sido considerada por el equipo como fundamental para el éxito de los sprints desarrollados.
- Retrospective: otro de los componentes clave para el logro de los objetivos, ya que es la revisión que hace el equipo de todo lo realizado en el sprint que acaba de finalizar, se busca determinar los errores y aciertos producidos en pos de una mejora continua del team.

Distribución de Sprints

La distribución de los sprints estuvo realizada del siguiente modo:

- Sprint 1 (Relevamiento): Relevamiento y documentación de las necesidades y lógica de recorridos del transporte de colectivos.
- Sprint 2 (Análisis): Elaboración de Requisitos funcionales, requisitos no funcionales, elaboración de casos de uso de análisis y prototipo estático.
- Sprint 3 (Diseño): Elaboración de diagramas de clase, definición del modelo de datos, elaboración de los casos de uso detallados,
- Sprint 4 (Diseño): Determinación de casos de uso críticos, diagramas de secuencias de búsquedas.
- Sprint 5 (Diseño y Codificación): Determinación de las herramientas de desarrollo y definición de las capas de la arquitectura del sistema.
- Sprint 6 (Codificación): Codificación del módulo de administración
- Sprint 7 (Codificación): Redefinición del modelo de datos, base de datos espaciales, Oracle spatial, inclusión del tipo de datos SDO_GEOMETRY.
- Sprint 8 (Codificación): Ajuste de la arquitectura del sistema, corrección de bugs de módulo de administración, desarrollo del módulo de consultas georreferenciado.
- Sprint 9 (Codificación): Redefinición de métodos para calcular la distancia entre 2 puntos
- Sprint 10 (Testing): En esta etapa se realiza el aseguramiento de la calidad de la aplicación basado en pruebas integrales y la corrección de bugs detectados.

6 DISEÑO

El diseño de la aplicación se realizó en base a un modelo de capas definido para aplicaciones JEE basado en el patrón de diseño Model View Controller (MVC) como lo muestra la Fig.1, en el mismo se puede observar como fluye la información atraves de las diferentes capas cuando se inicia una petición en la capa de la vista.

El patrón de diseño MVC es el patrón de arquitectura más usado en la ingeniería del software, y nos permitió definir las 3 capas principales de la aplicación (Ref. 2) es decir:

Capa de la vista

Incluye los archivos necesarios para mostrar las diferentes pantallas (xhtml, hojas de estilo, imágenes, otros recursos). Para la arquitectura diseñada se definieron Java Server Faces como tecnología base, y Prime Faces para lograr una interfaz con componentes ricos. En esta capa se definió una subcapa denominada "Servicios google maps" que nos permitió acceder a la vista basada en mapas de la aplicación.

Capa del modelo

En esta capa se definieron los componentes necesarios para el modelo de la aplicación. El mismo fue pensado para que pueda ser consumido por diferentes tipos de tecnologías en la capa de la vista, de este modo es posible utilizar el modelo para tecnologías Web o móviles. Básicamente se definieron:

- Servicios: está formado por un grupo de componentes organizados por categorías según las funcionalidades requeridas (roles, usuarios, empresas, recorridos, parada), básicamente consumen los servicios brindados por la capa DAO y devuelve los datos que fueron solicitados por el controlador.

- DAO: esta capa tiene como objetivo realizar las transacciones con la base datos, mediante el uso del framework de persistencia de mapeo objeto relacional hibernate.

- Utilidades: conformada con clases para el tratamiento de fechas, números, acceso a archivos de texto, etc.

- Entidades: en esta capa se encuentra la definición del dominio de la aplicación

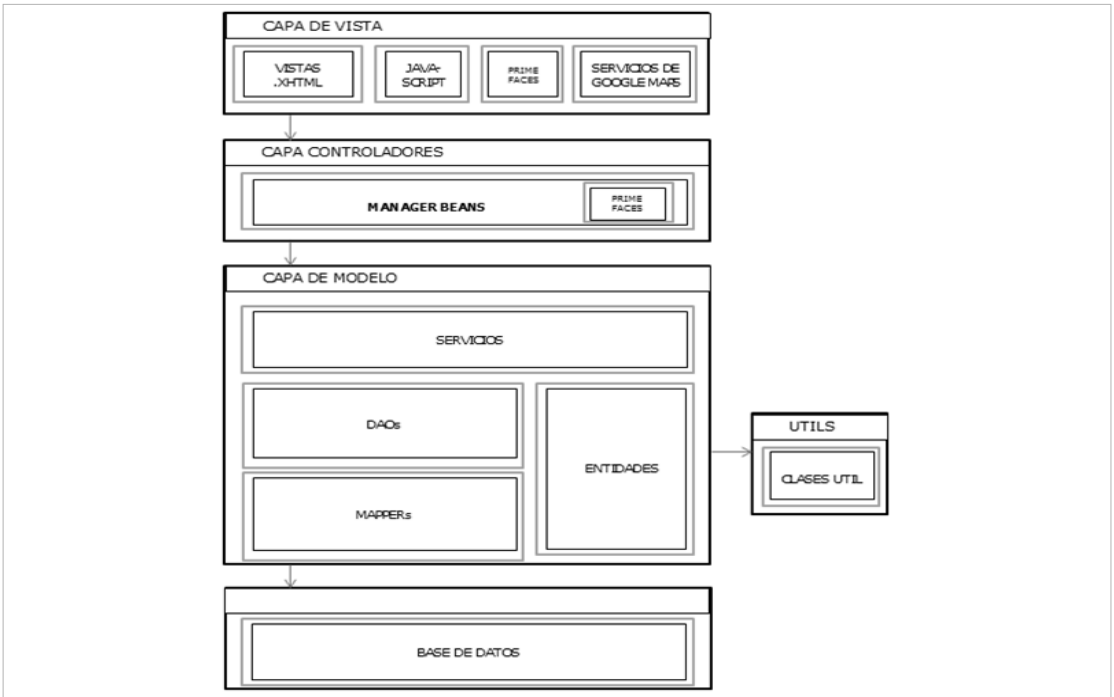

Figura 1. Diagrama en capas Model View Controller

Capa del controlador

Esta capa es la encargada de responder a los eventos que sucedan en la capa de la vista, en cada petición del usuario realiza la correspondiente solicitud de información al modelo para luego devolver la información en el formato que la vista pueda interpretar. Aquí se definieron todos los ManagedBeans que son los encargados de trasladar la petición de la vista a la capa de servicios y luego brindar la respuesta nuevamente a la capa de la vista.

7 TECNOLOGÍAS Y HERRAMIENTAS SELECCIONADAS

Uno de los aspectos más relevantes para el desarrollo del proyecto fue la definición de las tecnologías a utilizar en cada capa de la arquitectura, y las herramientas necesarias para el desarrollo de la aplicación. Luego de un minucioso análisis de las diferentes opciones se determinaron las siguientes tecnologías y herramientas

Tecnologías seleccionadas

- JSF (Ref. 3): nos permite desarrollar rápidamente aplicaciones de negocio dinámicas en las que toda la lógica de negocio se implementa en java, creando páginas para la vista con formato xhtml.
La principal función del controlador JSF es asociar a las pantallas, clases java que recogen la información introducida y que disponen de métodos que responden a las acciones del usuario. JSF nos resuelve de manera muy sencilla y automática muchas tareas por medio de los managed beans que se encuentran asociados a los formularios de la vista.

- Primefaces: Es una librería de componentes para JSF de código abierto con varias extensiones entre las cuales se implementaron:
  * Los componentes enriquecidos de la vista como html, editor, dialogs, charts, Ajax, etc.
  * themes integrados a la vista.

  Como así lo describe Cagatay (2009).

- API Google Maps: es la encargada de interpretar la información georreferenciada brindada por la capa del modelo en formato de mapas. En la arquitectura se definió una capa especial que se capaz de interactuar con los servicios de esta API.

- Spring framework: es un framework open source que consideramos muy necesario para implementar el patrón de diseño “Inversion of Control” (IoC) que nos permitió, entre otras cosas, resolver la inyección de dependencias. Posee un gran número de módulos que ofrecen una diversidad de servicios de los cuales se implementaron:
  * Contenedor de inversión de control: que cumple la función de una bean factory administrando el ciclo de vida de los objetos que necesita la aplicación atraves de la inyección de dependencias.
  * Programación orientada a aspectos: que permitió definir los escenarios de en los cuales se controlarán las transacciones a la base de datos.
  * Acceso a datos con RDBMS: utilizando Java Database Connectivty (JDBC) que permite la ejecución de operaciones sobre la base de datos, para esto se integró el framework hibernate para el acceso objeto relacional
  * Modelo Vista Controlador: que facilitó la configuración de la aplicación en relación a la organización de los beans que son consumidos por la aplicación.
- Hibernate (Ref. 4): es un framework también de código abierto que facilitó el mapeo de atributos entre la base de datos relacional y los objetos del dominio de la aplicación. La configuración se realizó por medio de archivos xml para evitar código invasivo en la arquitectura.

Herramientas Seleccionadas

- IDE de desarrollo: Eclipse versión Juno elegida por su versatilidad para integrarse con todas las tecnologías seleccionadas, además de ser un entorno liviano y de código abierto (Ref. 5).

- Servidor: con el fin de configurar el entorno de desarrollo del grupo de investigación, fue necesario la configuración de un servidor en un hosting dedicado, el mismo fue instalado un sistema operativo CENTOS 5.5.
- Subversion: utilizada para el versionamiento del código mediante un repositorio centralizado creado en el servidor del grupo de investigación.
- Mantis: Herramienta de gestión de proyecto utilizada para el control y asignación de tareas

8 DISEÑO DEL MODELO DE DATOS

Durante las iteraciones realizadas y luego de investigar las distintas posibilidades para trabajar con datos geoespaciales, se definió el Modelo de Datos. Es importante destacar que Oracle 11g posee una extensión que permite manipular los datos geoespaciales (Oracle Spatial). Usando para ello un tipo de dato sdo_geometry compatible en java con Jgeometry.
Con ello el conjunto de puntos de un recorrido se almacenará en dos columnas (para el recorrido de ida y el recorrido de vuelta) de la tabla “Recorridos” cuyo tipo de dato es sdo_geometry.
Del modelo de datos se destaca la inclusión del manejo de datos espaciales (spatial database). Donde es imprescindible establecer un cuadro de referencia para definir la localización entre objetos, ya que los datos tratados en este tipo de bases de datos tienen un valor relativo, no es un valor absoluto. Los sistemas de referencia espacial pueden ser de dos tipos (Ref. 6):
- Georreferenciados: aquellos que se establecen sobre la superficie terrestre, son los que normalmente se utilizan, ya que es un dominio manipulable, perceptible y que sirve de referencia.
- No georreferenciados: son sistemas que tienen valor físico, pero que pueden ser útiles en determinadas situaciones.
La construcción de la base de datos georreferenciada implicó un proceso de abstracción para pasar de la complejidad del mundo real a una representación simplificada que pueda ser procesada por la arquitectura definida.

Oracle Spatial

Oracle Spatial proporciona un esquema SQL y funciones que facilitan el almacenamiento, recuperación, actualización y consultas de grupos de características espaciales en el SGBD de Oracle. Oracle Spatial tiene los siguientes componentes (Chuck, 2009):

- Un esquema (MDSYS): determina el almacenamiento, sintaxis y semántica de los tipos de datos geométricos soportados.
- Un mecanismo de indización espacial.
- Operadores y funciones para realizar consultas sobre áreas de interés, consultas sobre relaciones espaciales y otras operaciones espaciales.
- Utilidades administrativas.

Oracle Spatial soporta diversos tipos de elementos geométricos, tanto simples como compuestos. Entre los diferentes tipos de elementos que se pueden implementar están los siguientes elementos de dos dimensiones: Point, Line String, Polygon, Rectangle, Circle, etc.
Los puntos de dos dimensiones son elementos compuestos por dos coordenadas X e Y, que a menudo corresponden con la longitud y la latitud.
Las líneas están compuestas por uno o más pares de puntos que definen segmentos que pueden ser

rectos o arcos. Los polígonos están compuestos por líneas conectadas que forman un anillo cerrado, el área de los polígonos está implícita.
Para la representación de este sistema de coordenadas, se usa el tipo de dato SDO_GEOMETRY.

Tipo de dato SDO_GEOMETRY

Con Oracle Spatial, la descripción de los objetos espaciales es guardada en una fila simple, en una columna de tipo SDO_GEOMETRY de una tabla definida por el usuario. Cualquier tabla que tenga columnas del tipo SDO_GEOMETRY debe tener otra columna, o conjunto de columnas, que definan una única clave primaria para la tabla. Las tablas de este tipo algunas veces son llamadas tablas espaciales o tablas geométricas espaciales.

9 PROTOTIPO

En la fig. 2 se presenta una pantalla del prototipo diseñado, en la misma se muestra la selección de los puntos origen y destino, la información de la línea y empresa del recorrido encontrado como resultado de la consulta realizada.
En las etapas de diseño e implementacion de las búsquedas de Paradas (Parada de origen y destino) se planteó realizar las misma dentro una región circular delimitada por su centro (dado por el punto origen o destino, según corresponda) y un radio, el cual será ingresado por el usuario, los resultados arrojados corresponderán a las Paradas que se encuentran en la región interior de la circunferencia. A continuación se procede a calcular las distancias entre cada uno de los puntos de las paradas obtenidas y el centro la de circunferencia, utilizando la fórmula de Haversine que permite efectuar este calculo teniendo en cuenta la curvatura de la tierrra.
La fórmula necesaria para calcular la distancia es:

$$d = 2r\sin^{-1}\left(\sqrt{\sin^2\left(\frac{\varphi_2-\varphi_1}{2}\right)+\cos(\varphi_1)\cos(\varphi_2)\sin^2\left(\frac{\psi_2-\psi_1}{2}\right)}\right) \quad (1)$$

Donde φ es la latitud, ψ es la longitud, R es el radio de la tierra (radio medio = 6,371km)
d es la distancia entre dos puntos con la longitud y la latitud (ψ, φ) (Ref. 7).

Resultados

El prototipo logrado es el resultado de la investigación e implementación de las tecnologías citadas en el presente documento, donde se destacan los siguientes los siguientes logros:
- Se logró la implementación de las tecnologías seleccionadas en la arquitectura diseñada, con lo cual se puede afirmar que las mismas son una buena solución para trabajar con este tipo de aplicaciones basadas en MVC y mapas.
- Se lograron muy buenos resultados con la metodología seleccionada ya que la misma permitió realizar varios prototipos funcionales incrementales, y realizar cambios a medida que se iba avanzando en la investigación. Las daily scrum y las sprint retrospective resultaron muy valiosas ya que permitieron detectar errores a tiempo y plantear soluciones que eviten la reaparición de los mismos.
- Se logró una buena performance de respuesta a las consultas realizadas con la interfaz de mapas, gracias a que en la capa del modelo se combinaron tecnologías de muy buena respuesta, tales como Spring Framework e Hibernate

Grado de Avance

El prototipo se encuentra en un 80% de avance, ya que se cuenta con el módulo de configuración finalizado y se están realizando mejoras en el módulo de las búsquedas en el mapa. Respecto de este módulo inicialmente se planteó realizar búsquedas en una región cuadrada del mapa, lo que nos devuelve las paradas encontradas dentro de la región. Actualmente se está trabajando sobre áreas circulares en el mapa debido a que se detectó que en las regiones cuadradas los valores obtenidos entre los puntos tiene una discrepancia respecto del circuito real en el mapa. Se está trabajando también sobre un algoritmo que calcule la distancia entre el punto origen - destino y los puntos encontrados en la región de la búsqueda. Para poder tomar las decisiones más adecuada al planteo de las búsquedas se están realizando distintas pruebas y analizando los distintos casos que se presentan para los diferentes sectores donde se encontraría el usuario y los recorridos existentes.

10 CONCLUSIONES Y TRABAJO A FUTURO

El desarrollo del presente proyecto permitió conocer y aplicar todos los aspectos necesarios para el modelado y programación de una aplicación Web integrado con una API para mapas. Se encontraron varios inconvenientes a la hora de implementar las diferentes capas de la aplicación, tales como:
- La implementación de los tipos de datos SDO_GEOMETRY con el framework Hibernate en la capa del modelo
- La integración de Primefaces con la API de mapas en la capa de la vista.

Se experimentaron con diferentes alternativas hasta encontrar un modelo de capas robusto, flexible y fácil de mantener, ya que separa en forma clara la lógica de negocio de la vista a implementar. La capa del modelo también presenta una gran flexibilidad respecto del motor de base de datos ya que es posible configurar la conexión de cualquier otro tipo de base de datos

Como trabajo a futuro, se espera implementar en la capa de la vista a dispositivos móviles de diferentes fabricantes, se estima que no demandará demasiados esfuerzos ya que se cuenta con una arquitectura flexible que podrá también ser adaptada para aplicaciones con funcionalidades similares, tales como circuitos turísticos, aplicaciones relacionadas con la salud para determinar centros de salud cercanos a un punto determinado y el transporte más rápido para llegar a ellos etc. Se prevé también la ampliación del alcance del prototipo a otras ciudades o provincias donde se pueda contar con la logística de algún tipo de servicio de transporte.

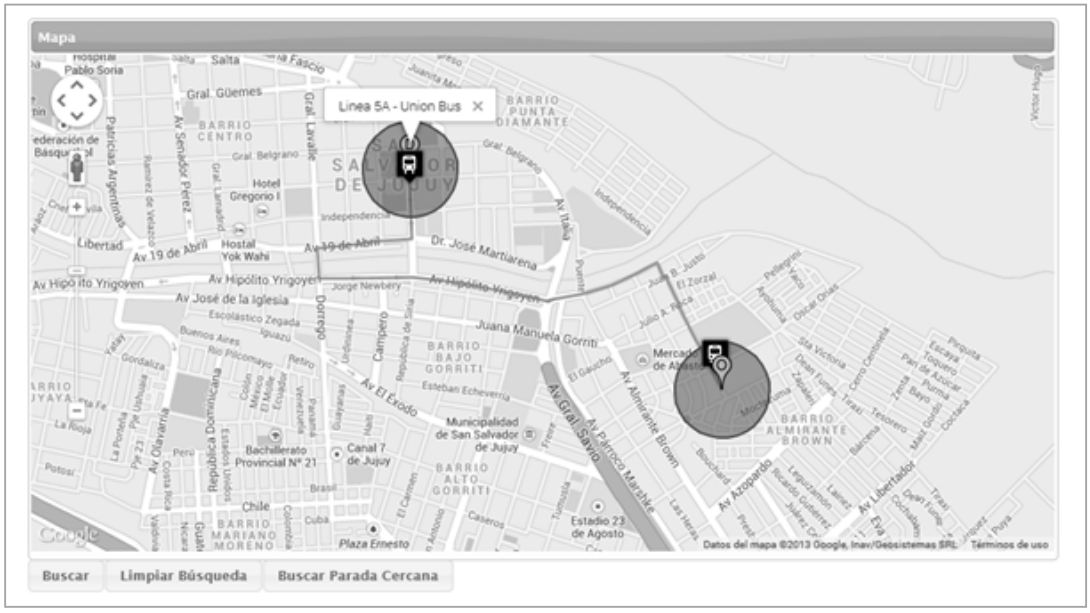

Figura 2. Resultado de las búsquedas

11 REFERENCIAS

(Cagatay): Cagatay Civici *"Primefaces user's guide 3.2". s.f.*

(Chuck 2009): Chuck Murray. *Oracle Spatial Developer's Guide, 11g*. Junio 2009

(Ref. 1): Georreferencion, http://es.wikipedia.org/wiki/Georreferenciaci%C3%B3n, Mayo de 2011

(Ref. 2): Modelo Vista Controlador y algunas variantes. http://www.neleste.com/modelo-vista-controlador-y-algunas-variantes/, Mayo de 2013

(Ref. 3): Tutorial de JavaServer Faces (s.f.). http://www.sicuma.uma.es/sicuma/Formacion/documentacion/JSF.pdf, Junio de 2012

(Ref. 4): Definición de Hibernate. https://www.google.com.ar/url?sa=t&rct=j&q=&esrc=s&source=web&cd=2&ved=0CC4QFjAB&url=http%3A%2F%2Fwww.unife.edu.pe%2Fing%2Fdesarrollo.doc&ei=RVqGT_b4OtLqtgf_4OjNBw&usg=AFQjCNEYWSGaKUVBmr9ZxZx3x-PwfI78IQ&sig2=slx7dOUOgEdwu8vDimg8K, Junio de 2012

(Ref. 5): Eclipse (Software). http://es.wikipedia.org/wiki/Eclipse_(software), Noviembre de 2012

(Ref. 6): Base de datos espacial, http://es.wikipedia.org/wiki/Base_de_datos_espacial, Abril de 2013

(Ref. 7): Calculate Haversine Distance in Java. http://reflectvicky.blogspot.com.ar/2013/04/calculate-haversine-distance-in-java.html, Julio 201