

Desarrollo de una aplicación móvil para visitas
dentro el campus de la UMSS usando
Geolocalización

Edmundo Figueroa Herbas

12 de junio de 2016

Índice general

1. Introducción	1
1.1. Antecedentes	1
1.2. Descripción del problema	2
1.3. Objetivo general	2
1.4. Objetivos Específicos	2
1.5. Justificación:	2
2. Marco Referencial	5
2.1. Node JS	5
2.2. Express JS	5
2.2.1. Middleware	6
2.2.2. Routing	6
2.3. Base de Datos	7
2.3.1. PostgreSQL	7
2.3.2. PostGis	8

Capítulo 1

Introducción

El presente proyecto consiste en el desarrollo de una aplicación móvil que permita ubicar y encontrar una locación dentro del campus de la UMSS, la aplicación deberá localizar la ubicación actual del usuario y permitir especificar un punto de destino, mostrando a continuación el camino más corto para llegar a destino.

El campus universitario abarca más de 214.000 m² y encierra varias facultades y oficinas administrativas, para estudiantes nuevos y antiguos o personas que necesitan hacer trámites administrativos, incluso si solo se quiere conocer el campus, es necesario contar con un mapa donde ubicarse.

Las aplicaciones móviles tienen una gran demanda por parte de la población ya que la gran mayoría posee un smartphone o teléfono inteligente con capacidad de ejecutar aplicaciones muy fácilmente, los smartphones cuenta también con GPS, el cual se usa para conocer la ubicación del usuario con un margen de error de 3 metros, usando puntos de referencia geo-localizados se puede determinar la ruta óptima para llegar a destino.

Es una desventaja para nuestra Universidad que no exista información confiable de fácil acceso para poder desplazarse por el campus.

1.1. Antecedentes

Actualmente Google Maps ofrece una solución a este problema pero lo ofrecen para las ciudades pero dentro del campus de la UMSS, google no

cuenta información interna de la UMSS.

Existen blogs o aplicaciones con Información acerca de lugares turísticos o de interés para visitar en nuestra ciudad como ser TripAdvisor, pero estas aplicaciones tampoco cuentan con información interna de la UMSS.

1.2. Descripción del problema

La Universidad Mayor de San Simón no cuenta con un mapa interactivo que muestre la ubicación de los puntos o lugares que se encuentran dentro del campus universitario, este mapa sería de gran ayuda para desplazarse dentro del campus universitario, la falta de un mapa con estas características genera malestar entre la estudiantes o personas que quieren realizar trámites administrativos, ya que al no contar con una aplicación que muestre los puntos de interés geolocalizados se pierde tiempo al tratar de encontrarlos.

1.3. Objetivo general

Desarrollar una aplicación web móvil responsive para optimizar la ubicación de lugares y el Desplazamiento al interior del Campus Universitario de la Universidad Mayor de San Simon.

1.4. Objetivos Específicos

- Gestionar lugares geolocalizados dentro del campus Universitario.
- Permitir encontrar lugares dentro el campus.
- Priorizar lugares visitados.
- Generar mapa geolocalizado.
- Gestionar usuarios.

1.5. Justificación:

El Campus Universitario es bastante extenso y está en constante reestructuración, cada vez hay más aulas, las oficinas se mueven de lugar, etc. gracias a esto es que los mapas, que son escasos y están impresos sobre

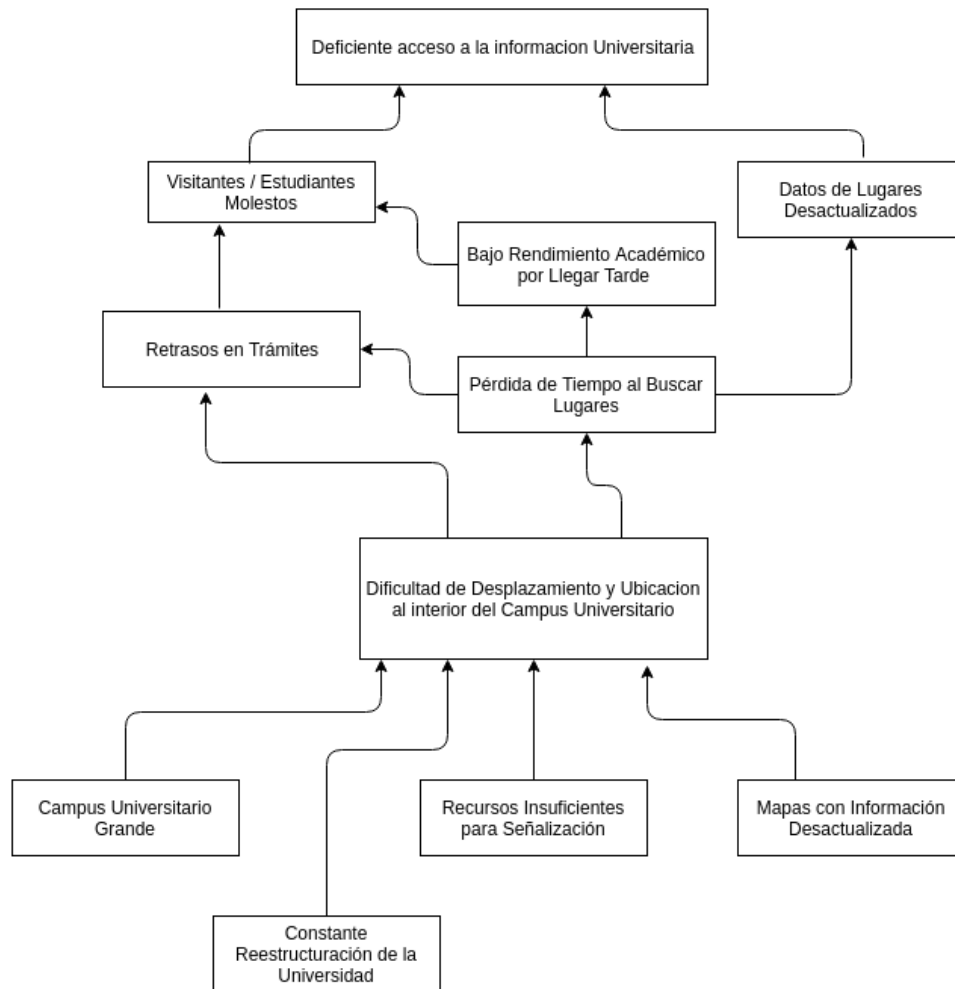
Árbol de Problemas

Figura 1.1: Diagrama Arbol de Problemas

banners estáticos, son difíciles de actualizar. Este hecho genera malestar en estudiantes que llegan tarde a sus clases o personas que necesitan hacer trámites administrativos y no encuentran con facilidad las oficinas a las que necesitan llegar.

Una aplicación que permita ubicar y encontrar locaciones además de proveer la ruta óptima dentro del campus de la Universidad Mayor de San Simón es de gran importancia para mejorar nuestra presentación a cualquier persona que necesite desplazarse por el campus Universitario.

Las Aplicaciones móviles y/o web demostraron ser el futuro del desarrollo de software y la gran mayoría de los países en el mundo consumen estas soluciones y nosotros necesitamos apuntar a esta tendencia.

Capítulo 2

Marco Referencial

2.1. Node JS

Node.js aparecio en 2009 y esta construido sobre el Motor de JavaScript de Google “V8” que fue sacado del browser y aplicado en el servidor.

Para desarrollar en el lado del browser (cliente) el programador solo tiene disponible JavaScript como lenguaje de desarrollo pero en el lado del servidor existen muchas alternativas (Ruby, C#, Phython, Java, etc.), JavaScript no estaba disponible.

Node se beneficia del Motor de JavaScript “V8” ya que éste es rápido y tiene integrado un sistema para manejar las instrucciones de forma asincrónica, pero el mayor beneficio y el porqué Node adquirió una gran popularidad es la facilidad de compartir código entre el cliente (browser) y el servidor.

Node.js provee características pero estas pueden parecer complicadas o que necesitan mas instrucciones de las necesarias para llevar a cabo acciones que ya son comunes en la creacion de aplicacion en lado del servidor, por ejemplo a la hora de crear un servidor web, Node se popularizo en gran medida por poder crear servidores web personalizables pero como ya dijimos esto tiene su grado de complejidad, aca es donde entra en accion Express.js.

2.2. Express JS

Express.js es un framework que esta construido sobre la funcionalidad de servidor web de Node.js, Express.js ayuda a simplificar el API de Node

y añadir nuevas características, diseñadas para mejorar y facilitar la organización de una aplicación *Express*.

El Cliente (navegador web, aplicación móvil, etc) envía una petición web y el servidor web de Node.js maneja los protocolos web, leyendolos y enviándolos a una aplicación *Express* que se encarga de añadir características a la petición y espera la respuesta del “Middleware Stack”, la función responde a la llamada y el servidor HTTP de Node envía la respuesta mediante los protocolos web al Cliente.

Para escribir un servidor web con Express no es necesario una gran función para manejar un request, Express contiene utilidades que permite escribir funciones más pequeñas para facilitar el manejo de las peticiones web, usando de “middleware” y “routing”.

2.2.1. Middleware

Node.js maneja una función para trabajar con una petición web, encambio *Express* maneja la llamada con varias funciones, cada función se encarga de una pequeña parte del trabajo. Estas pequeñas funciones que manejan la petición web se denominan *Middleware functions* o Middleware.

2.2.2. Routing

Muy parecido al Middleware, el Routing se encarga de partir una función de petición web monolítica en pequeñas piezas, pero a diferencia del Middleware, estos manejadores peticiones se ejecutan condicionalmente dependiendo del URL y el método HTTP (GET, POST, DELETE) que el cliente envía.

Express.js es bastante extensible y cuenta con gran popularidad en la comunidad de desarrollo, la cual provee herramientas para renderizar dinámicamente HTML o interfaces para comunicarse con Bases de Datos, por ejemplo para manejar la conexión y llamadas a una base de datos en PostgreSQL se usó la librería *pg-promise*.

```
database.any("SELECT * FROM users WHERE id = $1", [userId])
  .then(function (data) {
    response.send(data.name);
  });
```


2.3. Base de Datos

En una aplicación web es necesario alguna forma de persistencia de datos, en especial si se están usando datos complejos y en gran cantidad, para realizar esta tarea, la base de datos es un factor primordial. Rails maneja la base de datos mediante un ORM, por lo tanto la base de datos que se utilice no es tan excluyente, en este caso se utilizó *PostgreSQL* como base de datos relacional.

2.3.1. PostgreSQL

PostgreSQL es un sistema de gestión de bases de datos objeto-relacional, Open Source y distribuido bajo licencia BSD. PostgreSQL utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando. La última versión de PostgreSQL es la 9.5, su desarrollo comenzó hace más de 16 años, y cuenta con una gran comunidad que aporta con el desarrollo, testeo de nuevas versiones. PostgreSQL está considerada como una de los mejores *Sistemas de gestión de bases de datos*, es muy completo y está muy bien documentado¹. Entre sus características se pueden nombrar las siguientes.

- Es una base de datos 100 % ACID²
- Integridad referencial
- Replicación asincrónica/sincrónica
- Múltiples métodos de autenticación
- Disponible para Linux y UNIX en todas sus variantes
- Funciones/procedimientos almacenados
- Soporte a la especificación SQL

Personalmente se escogió trabajar con PostgreSQL como DBMS porque cuenta con una extensa documentación, y gracias a su carácter “Open Source”, y su gran flexibilidad en poder definir nuevos tipos de datos, se hace posible que empresas como **Refractions Research** puedan crear recursos como PostGIS, necesario para trabajar con datos geográficos ó espaciales.

¹ <http://www.postgresql.org/docs/9.5/static/>

² ACID es un acrónimo de Atomicity, Consistency, Isolation and Durability

2.3.2. PostGis

PostGIS es un módulo que añade soporte de objetos geográficos al DBMS PostgreSQL, convirtiéndola en una base de datos espacial para su utilización en un Sistema de Información Geográfica(SIG³).

El desarrollo de PostGIS está a cargo de **Refractions Research**, está liberada con la *Licencia pública general de GNU*, declarándola como software libre y lo protege de cualquier intento de apropiación.

PostGIS implementa la especificación “SFSQL” (Simple Features for SQL, define los tipos y funciones que necesita implementar cualquier base de datos espacial) de la OGC (Open Geospatial Consortium, es un consorcio internacional, formado por un conjunto de empresas, agencias gubernamentales y universidades, dedicado a desarrollar especificaciones de interfaces para promover y facilitar el uso global de la información espacial).

PostGIS al igual que PostgreSQL tiene una documentación bastante extensa, y cuenta con equipo de desarrollo que continuamente va sacando nuevas versiones, actualmente se encuentra la versión 2.0.1, pero para el desarrollo de la aplicación se hizo uso de la versión 1.5.5.

PostGIS es gratis, pero no por ello es una herramienta de baja calidad, al contrario se la considera una herramienta de nivel empresarial, y muchas instituciones la están usando de manera exitosa⁴, aparte de numerosas aplicaciones

Manejar los datos geográficos con PostGIS es sencillo y muy eficiente, por esta razón se utilizó esta herramienta, pero para conseguir la ruta óptima entre 2 puntos se necesitaba el uso del algoritmo de Dijkstra y para PostGIS existe el módulo **PgRouting**, que tiene implementado este algoritmo.

pgRouting

pgRouting es una extensión de PostGIS para proveer funcionalidades de ruteo espacial. pgRouting es un desarrollo posterior de pgDijkstra y actualmente está siendo mantenido por Georepublic, la última versión estable es la 2.1, y es la que fue usada para desarrollar el sistema.

³ Es bastante común utilizar el acrónimo en Inglés, Geographic Information System (GIS), de ahí viene el término de PostGIS = Postgres + GIS

⁴ <http://www.postgis.org/documentation/casestudies/>

Las ventajas del ruteo en la base de datos son:

- Los datos y atributos pueden ser modificados desde varios clientes, como Quantum GIS y uDig a través de JDBC, ODBC, o directamente usando Pl/pgSQL. Los clientes pueden ser PCs o dispositivos móviles.
- Los cambios pueden ser reflejados instantáneamente a través del motor de ruteo. No hay necesidad de hacer cálculos previos.
- El parámetro de “costo” puede ser calculado dinámicamente a través de SQL y su valor puede provenir de múltiples campos y tablas.

pgRouting provee funciones para:

- Camino mínimo (Dijkstra): algoritmo de ruteo sin heurística
- Camino mínimo (A-Star): ruteo para conjunto de datos grandes (con heurística)
- Camino mínimo (Shooting-Star): ruteo con restricciones de giro (con heurística)
- El problema del viajante (TSP: Traveling Salesperon Problem)
- Cálculo de ruta (Isolíneas)

Bibliografía

- [1] Sam Ruby, Dave Thomas, David Heinemeier Hansson, *Agile Web Development with Rails, Fourth Edition*
- [2] <http://trends.builtwith.com/topsites/Ruby-on-Rails>
- [3] tumblr.yasulab.jp/post/10271634919/5-question-interview-with-twitter-developer-alex-payne
- [4] http://www.artima.com/scalazine/articles/twitter_on_scala.html
- [5] <http://oreilly.com/web2/archive/what-is-web-20.html>
- [6] <http://radar.oreilly.com/2006/12/web-20-compact-definition-tryi.html>
- [7] <http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm>
- [8] <http://st-www.cs.illinois.edu/users/smarch/st-docs/mvc.html>
- [9] <http://msdn.microsoft.com/en-us/architecture/bb906060.aspx>
- [10] <http://www.json.org/json-es.html>
- [11] [http://kartoweb.itc.nl/geometrics/Coordinate %20systems/coordsys.html](http://kartoweb.itc.nl/geometrics/Coordinate%20systems/coordsys.html)
- [12] Introduction to Spatial Coordinate Systems: Flat Maps for a Round Planet,
[http://msdn.microsoft.com/en-us/library/cc749633\(v=sql.100\).aspx](http://msdn.microsoft.com/en-us/library/cc749633(v=sql.100).aspx)
- [13] <http://publib.boulder.ibm.com/infocenter/db2luw/v8/index.jsp?topic=/com.ibm.db2.udb.doc/opt/csb3022b.htm>
- [14] <http://msdn.microsoft.com/es-es/library/bb964707.aspx>
- [15] Introducción a la Teoría De Grafos, Alfredo Caicedo Barrero,
ISBN: 978-958-99325-7-5