



Vol. 27, No 1, 2
Enero - Junio 2019

CIENTIFICA



An International Refereed Scientific Journal
of the Facultad Experimental de Ciencias
at the Universidad del Zulia

Esta publicación científica en
formato digital es continuidad
de la revista impresa

Depósito Legal: pp 199302ZU47

ISSN: 1315-2076

CIENCIA 27 (1,2), 24 - 30, 2019
Maracaibo, Venezuela

DOI: <https://www.doi.org/10.5281/zenodo.5592980>

ZENDEROZ: Planificador de rutas de transporte público en Maracaibo (Venezuela)

Miguel Celedón, Amerindia Aguiar, Gerardo Pirela Morillo()*

Laboratorio de Lenguajes y Modelos Computacionales. Facultad Experimental de Ciencias.
La Universidad del Zulia
miguelceledon@outlook.com, amerindia.aguiar@gmail.com, gepirela@fec.luz.edu.ve

Recibido: 18-02-2019 Aceptado: 01-03-2019

Resumen

Los sistemas de información geográfico son en la actualidad tan ubicuos como las redes sociales o el uso de Internet. Su impacto desde la planificación urbana hasta la gestión de crisis es innegable. La presente investigación presenta la aplicación de SIG en la planificación de desplazamiento con rutas de transporte público en Maracaibo. A lo largo de las fases del ciclo de desarrollo de un software, se realizó un análisis preliminar de una herramienta base que usa grafos para la representación de las redes de vialidad y de transporte público en Maracaibo y el algoritmo de Dijkstra para calcular eficientemente rutas más cortas; se construyó un módulo para el rastreo y estimación de tiempos de llegada de unidades de transporte público; se desarrolló una interfaz web adaptativa para la integración de las diversas funcionalidades; y se realizaron pruebas funcionales al sistema resultante. El producto final cumple con estándares de calidad de SIG y su funcionamiento es computacionalmente óptimo.

Palabras clave: Sistemas de información geográfica, Sistema de Transporte público, Planificación automática de rutas, Grafos.

ZENDEROZ: Public transport route planner for Maracaibo (Venezuela)

Abstract

Geographic Information Systems are now as ubiquitous as social networks or the use of the Internet. Its impact from urban planning to crisis management is undeniable. This research presents the application of GIS in commute planning with public transport routes in Maracaibo. Throughout the phases of the software development cycle, a preliminary analysis was carried out of a base tool that uses graphs to represent the road and public transport networks in Maracaibo and the Dijkstra algorithm to efficiently calculate shortest routes; a module was built for tracking and estimating arrival times of public transport units; an adaptive web interface was developed for the integration of the various functionalities; and functional tests were carried out on the resulting system. The final product complies with GIS quality standards and its performance is computationally optimal.

Keywords: Geographic Information Systems, Public Transport System, Automatic Route Planning, Graphs

Introducción

Un sistema de información geográfica (SIG) es un sistema computacional que permite la captura, almacenamiento, consulta, análisis y visualización de datos geoespaciales. Los SIG conforman uno de los pilares esenciales de la tecnología geoespacial, la cual se relaciona con una multitud de campos, como sistemas de posicionamiento global (GPS – por sus siglas en inglés), sensores remotos, cartografía, geoestadística, entre otros (1).

Un área en la que los SIG han tenido crucial protagonismo es en el manejo de crisis. El instituto de investigaciones sobre sistemas ambientales (ESRI – por sus siglas en inglés) reportó, durante el terremoto que sacudió Haití en 2010, que se estaba usando SIG para la toma de decisiones respecto al despacho de asistencia donde más se necesitaba: identificación de áreas con daños extremos, evaluación de impacto en infraestructura crítica, localización de áreas aptas para la distribución de agua y comida (2). Por su parte, Patric Meier, reportero para National Geographic (3), resaltó que el *crowdsourcing* (colaboración abierta distribuida y externalización abierta de tareas) permitió completar la información geográfica que para entonces se tenía sobre el terreno, las calles, las construcciones y otras georreferencias cruciales durante la crisis. El uso de información satelital, herramientas SIG abiertas y fuentes de datos mixtas (como tuits, emails, SMS) permitió traducir y priorizar las necesidades durante el manejo de la crisis y fueron cruciales en la selección de sitios para la reconstrucción de instalaciones eléctricas, redes de transporte público, entre otros servicios.

En el área de viabilidad y transporte público, los SIG también fungen un rol esencial en los procesos de planificación, diseño, definición de políticas, metas e inversiones para satisfacer la presente o futura necesidad de transportar bienes o movilización de la población a sus destinos de manera eficiente y sostenible (4).

En la actualidad y a nivel mundial, Google Maps (5) se perfila con una herramienta primordial para las satisfacer necesidades de ubicación de sitios georreferenciados y planificación de rutas para llegar de un punto a otro, bien sea de forma pedestre, con un vehículo o con el uso de la red de transporte público (en caso de que esta información esté contenida en Google Maps), brindando además información sobre rutas alternas, distancia y tiempo estimado. Además, ofrece una serie de bibliotecas, en la forma de interfaz para programar aplicaciones (API – según sus siglas en inglés) para desarrollo de soluciones personalizadas.

En muchas ciudades del mundo, dependiendo de la entidad que administre las unidades de transporte

público, también existen herramientas alternas a Google Maps que permiten el rastreo de unidades y muestran su posición, dirección y tiempo estimado de llegada a cada parada.

En el caso latinoamericano, Google Maps contiene la información de las redes de transporte público de la mayoría de las grandes ciudades. Sin embargo, en Venezuela, este no es el caso y Google Maps solo habilita las opciones de rutas pedestres y de vehículos particulares. Es por ello que los usuarios presentan quejas comunes entre los que están con mayor frecuencia la falta de un horario de servicio, la caótica regularidad de las unidades que genera incertidumbre en los tiempos estimados de llegada de éstas a las paradas y la dificultad de planificar las rutas a tomar que optimice la inversión de tiempo o dinero del transeúnte al momento de trasladarse a su destino.

En Maracaibo se han realizado trabajos de investigación y prototipado de sistemas para planificación de rutas. En (6) se describe un sistema para la planificación de rutas de transporte terrestre, utilizando mapas de la ciudad obtenidos en formato AutoCAD y .NET como herramienta de desarrollo de software. En (7) se describe un prototipo de planificador de rutas de transporte público, migrando las capas de mapas utilizados por (6) a los formatos requeridos por la API de Google Maps y usando Java como herramienta de desarrollo. Aguiar describe en (6) dos herramientas similares desarrolladas en el Distrito Capital: “SIG – Vargas” e “Infomapa Chacao”. Sin embargo, tal como se expresa en (8), no se cuenta en Maracaibo con una plataforma única que integra ambos planificadores, que pueda ir creciendo en servicios y que esté al alcance del público general.

Más aún, tal como se expresa en (9) y (10), a mediados del 2014, la Empresa Socialista Metro de Maracaibo, C.A. implanta en el municipio un nuevo sistema de transporte llamado Bus Metromara, el cual está conformado por unidades colectivas de nueva generación que cuentan con GPS y paradas. Sin embargo, los usuarios del Bus Metromara encuentran dificultades al momento de esperar las unidades por las largas colas a la hora pico y el tiempo tardío que tienen estas unidades. El conocimiento de la ubicación, dirección y tiempo estimado de llegada de las unidades a las paradas específicas facilitaría al usuario la toma de decisión y planificación al momento de movilizarse por la ciudad.

Se plantea entonces extender los trabajos descritos en (7), (9) y (10) en una herramienta SIG web que permitan la planificación de rutas y el rastreo de unidades de transporte público.

Representación computacional básica de un SIG

Los SIG utilizan información que se captura de diversas fuentes, se almacena en bases de datos especializadas y se representa computacionalmente por capas que se van sobreponiendo según las

necesidades de análisis y visualización (4). En el caso particular de redes de transporte público, se comienza con capas cartográficas (mapas) de la región sobre la cual se representan las vías (calles y avenidas) y las intersecciones de éstas, tal como lo muestra la figura 1.

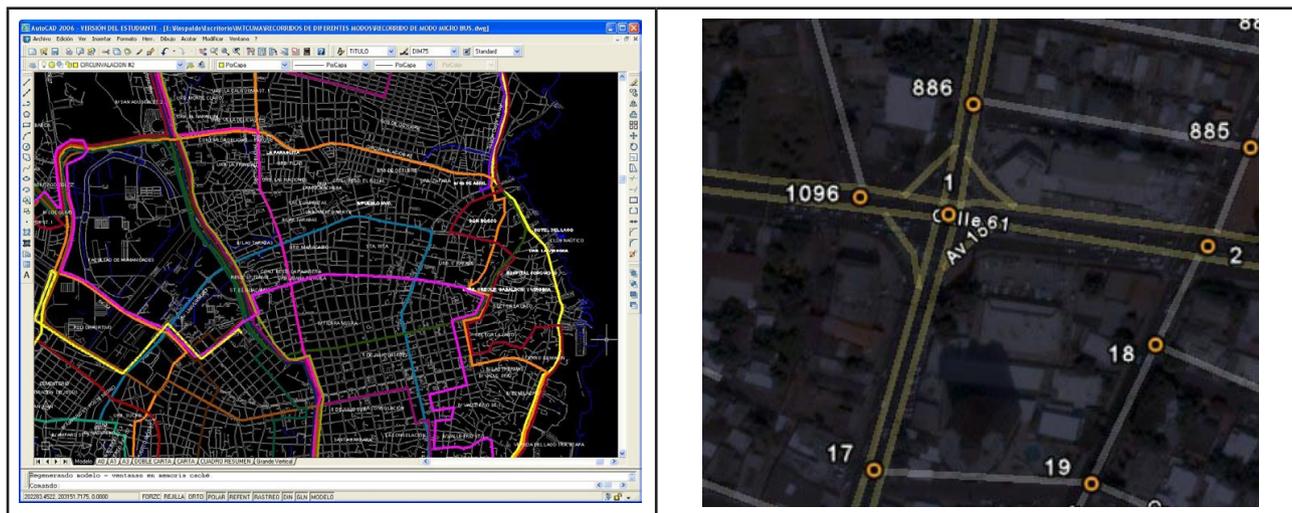


Figura 1 – Mapa de la ciudad de Maracaibo con capas sobrepuestas de vías e intersecciones etiquetadas.

La estructura de datos más adecuada para representar estas redes viales es el grafo. En (11) se define un grafo G como la tupla: $G = \langle V, A \rangle$ con $V = \{\text{vértices}\}$ y $A \subseteq V \times V = \{\text{arcos o aristas}\}$. Sobre estas estructuras ya existen representaciones computacionales y algoritmos eficientes conocidos para realizar análisis como estudio de conectividad y cálculo de rutas óptimas: el conocido algoritmo de Dijkstra, usando montículos de Fibonacci para su cola de prioridad, es el más eficiente para hallar caminos más cortos entre dos nodos cualesquiera de un grafo, reportándose una complejidad temporal lineal-logarítmica sobre la cantidad de nodos ($|V|$) y arcos ($|A|$) del grafo G : $T(\text{Dijkstra}(G, u \in V)) \in O(|A| + |V| \cdot \log|V|)$

Sistema de posicionamiento global

El sistema de posicionamiento global (GPS por sus siglas en inglés) es un sistema que permite determinar la posición de un objeto en la Tierra con una precisión generalmente de unos pocos metros de la posición real (12).

El GPS funciona mediante una red de 24 satélites en órbita sobre la Tierra, a 20.200 km de altura, con trayectorias sincronizadas para cubrir toda la superficie de la Tierra. Cuando se desea obtener la posición de un objeto, el receptor localiza automáticamente al menos tres satélites de la red,

de los que recibe señales con la identificación y la hora del reloj de cada uno y calcula el tiempo que tardan en llegar las señales al receptor, de tal modo que mide la distancia al satélite mediante el método de trilateración inversa, con el que se determina la distancia de cada satélite al punto de medición (el receptor GPS). Conocidas estas distancias, se determina fácilmente la posición relativa del receptor respecto a los satélites y se obtiene finalmente su posición absoluta o coordenadas reales en la Tierra (12).

Estimación de distancia entre puntos geográficos

Puesto que la Tierra es una forma aproximada a una esfera, la distancia entre dos puntos sobre la superficie terrestre no puede ser calculada a partir de una línea recta (distancia euclidiana) sino más bien debe ser calculada usando una distancia ortodrómica, que es el camino más corto entre dos puntos en la superficie de una esfera, es decir, el arco del círculo cuyo centro sea el mismo de la esfera (conocido como “círculo máximo”) que conecte ambos puntos (13).

Sean los puntos P_1 y P_2 , determinados por sus coordenadas geográficas de longitud y latitud, respectivamente: $P_1 = (\varphi^1, \lambda^1)$ y $P_2 = (\varphi^2, \lambda^2)$, y sean $\Delta\varphi$, $\Delta\lambda$ sus diferencias absolutas, entonces el ángulo

central entre ellos: $\Delta\sigma$, está dado por la ley esférica de cosenos:

$$\Delta\sigma = \cos^{-1}(\sin\varphi^1 \cdot \sin\varphi^2 + \cos\varphi^1 \cdot \cos\varphi^2 \cdot \cos\Delta\lambda)$$

Así, la longitud del arco d , para una esfera de radio r y $\Delta\sigma$ dado en radianes se obtiene por: $d=r\Delta\sigma$

Sin embargo, debido a la precisión finita de los números sistemas de cómputo, pueden surgir problemas de redondeo si la distancia es pequeña. Para distancias pequeñas es numéricamente mejor condicionada la conocida “fórmula haversine” o “fórmula del semiverseno”:

$$\Delta\sigma = 2 \sin^{-1} \sqrt{\sin^2(\Delta\varphi/2) + \cos\varphi^1 \cdot \cos\varphi^2 \cdot \sin^2(\Delta\varphi/2)}$$

Para el caso de la Tierra, si se desean calcular distancias pequeñas, puede asumirse sin problemas una forma esférica para ésta, ya que el error sería muy pequeño, y el radio para calcular la distancia final sea 6.371 Km, definido por la Unión Internacional de Geodésica y Geofísica (IUGG).

Metodología

Para el desarrollo de esta investigación se siguió la metodología utilizada por (6), planteada a lo largo de cuatro fases fundamentales: investigación preliminar, diseño, desarrollo de la herramienta y pruebas del producto final.

En la primera fase se realizó un estudio de la aplicación existente, llamada Zenderoz, y se determinaron los nuevos requerimientos del sistema adaptados a las nuevas funcionalidades: interfaz web, ampliación de rutas, rastreo de unidades de transporte público.

La aplicación base: Zenderoz es una aplicación de escritorio que utiliza Java Swing para su interfaz gráfica. Al convertir la aplicación en un servicio web, dicha librería ya no era necesaria ni su código relacionado, por lo que se optó en modificar dicho código y construir un servicio web REST-ful, con un método GET del protocolo HTTP que devuelve su salida en formato JSON, para ser procesado por el código del lado del cliente.

Adicionalmente, puesto que “ZenderozWeb” llevaría el seguimiento de unidades de transporte público, debe tener información de las rutas y paradas que dichas unidades recorren, por lo que ZenderozWeb necesita una base de datos también con tal información, el diagrama se presenta en la siguiente sección.

Arquitectura de ZenderozWeb

Se planteó usar un servidor web para que funcione como portal para interconectar los clientes web con la aplicación Zenderoz. Ya que ésta está desarrollada en Java, lo mejor fue utilizar un servidor web desarrollado para este mismo lenguaje. Se optó por utilizar Apache Tomcat en el servidor web, el cual hospeda una única instancia de Zenderoz a la espera de peticiones de los usuarios mientras que, al mismo tiempo, actualiza periódicamente los datos GPS de las unidades de transporte registradas. La arquitectura final a alto nivel se aprecia en la figura 2.

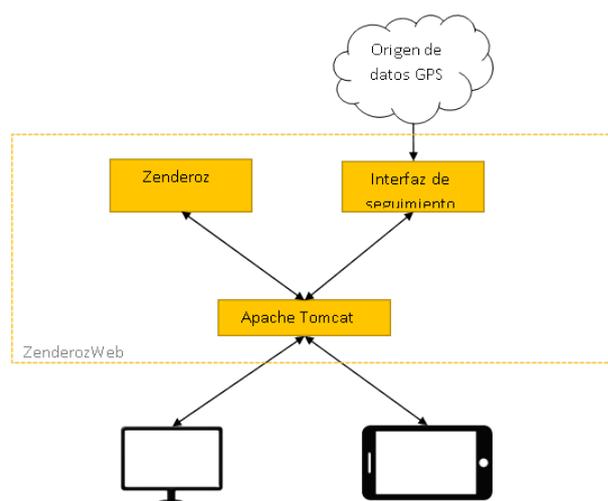


Figura 2 – Arquitectura de ZenderozWeb

Para efectos de su desarrollo, el componente externo “Origen de datos GPS” es un ambiente simulado, que imita unidades moviéndose entre puntos de una ruta preestablecida. La clase principal de la aplicación, contiene una instancia del gestor de GPS, que se encarga de llevar el control de las unidades y proveer el estado de todas y cada una de las unidades siempre que se requiera por petición de un usuario.

Además de recolectar y procesar en tiempo real la información de geoposicionamiento de las unidades vehiculares, ZenderozWeb también almacena información de las rutas, paradas y detalles de cada unidad para mayor control y organización, dicha información está almacenada en una base de datos MySQL.

De esta manera, resulta sencillo filtrar las paradas y unidades dependiendo la ruta de donde pertenezcan, para mostrarle al usuario sólo la información que le interesa, además de tener información útil que necesita colectarse una sola vez, tales como el punto de retorno de una ruta y la distancia entre una parada y la anterior. Esto

último hace posible calcular el tiempo estimado para la llegada de la unidad a la parada deseada, ya que teniendo la distancia a recorrer (calculada por haversine) y la velocidad (calculada por la diferencia de posiciones entre actualizaciones sucesivas del servicio de origen GPS), se puede calcular el tiempo estimado a una parada con la fórmula conocida: $\text{Tiempo} = \text{Distancia}/\text{Velocidad}$.

Resultados y discusión

ZenderozWeb tiene una estructura muy sencilla e intuitiva para el usuario, desde la vista principal (figura 3) puede empezar a interactuar con el mapa de las rutas en tiempo real, seleccionando las opciones que desee y un botón para planificar rutas llenando la información y recibiendo la respuesta (figura 4).

Figura 3 – Formularios para planificación de ruta: a la izquierda: Zenderoz; a la derecha: ZenderozWeb

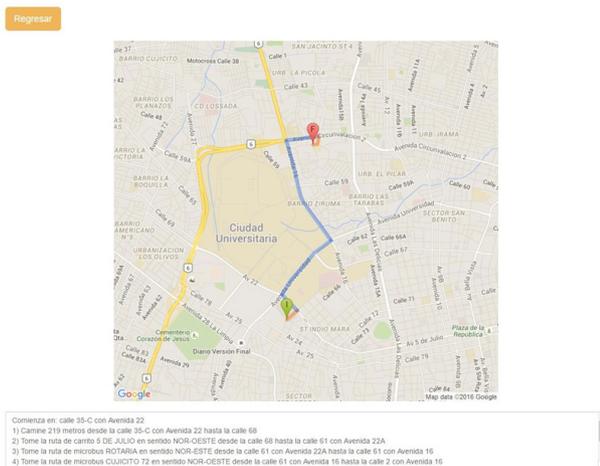


Figura 4 – Resultado de la planificación de ruta

La interfaz inicial tiene el botón “Planificar ruta”, al activar el cual se mostrará el formulario original de Zenderoz con un diseño más simple y amigable a dispositivos móviles. Una vez llenado el formulario: esquina origen, esquina destino y tipo de búsqueda,

con el botón “Planificar” se activará el llamado al servicio web de Zenderoz y se presentará el resultado de la planificación en dos modalidades: gráfica (pintando sobre el mapa las rutas resultantes) y textual (describiendo los pasos del plan: qué rutas tomar y cómo hacer los trasbordos).

Este resultado se obtiene invocando el servicio web que ejecuta el núcleo de Zenderoz. El grafo de rutas se representa computacionalmente con listas de adyacencia para minimizar la complejidad espacial, tal como se muestra en la figura 5. El grafo resultante de una red de vialidad urbana tiende a ser grande, por la gran cantidad de esquinas (nodos) y calles (aristas) que deben tomarse en cuenta. Debido a esto, una vez que se conocen las coordenadas de las esquinas entre las cuales se desea hallar un plan de rutas de transporte público óptimo, Zenderoz realiza un preprocesamiento del grafo, para eliminar las zonas irrelevantes del área urbana y recortar las esquinas intermedias que no incurran en transbordo o en potenciales segmentos pedestres entre una ruta de bus y otra. Esto requirió que, para el cálculo final de las rutas más cortas, se escribieran modificaciones específicas del algoritmo de Dijkstra con truncamientos tempranos por distancia requerida o por destino. Los algoritmos 1 y 2 a continuación muestran estas adaptaciones.

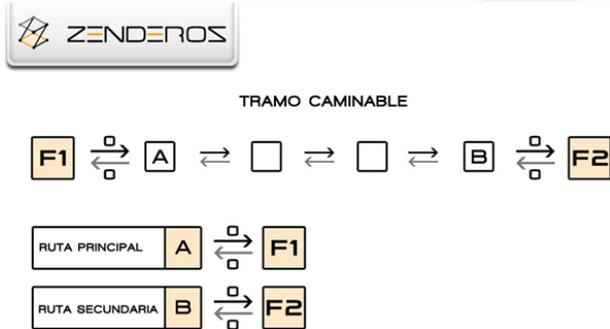
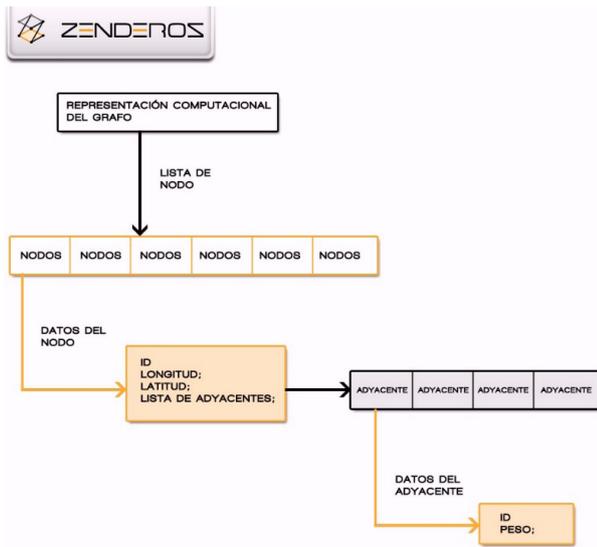


Figura 5 – Representación computacional del grafo

```

DijkstraTruncadoPorDistancia(Grafo G, nodo_s,
distanciaMaxima)
for u ∈ V[G] do
  distancia[u] = ∞
  padre[u] = NULL
fin for
distancia[s] = 0
Encolar(cola, nodos(G))
mientras vacia(cola) ≠ true do
  u = extraer_minimo(cola)
  for v ∈ adyacencia[u] do
    if (distancia[v] > distancia[u] + peso(u,
v)) &&
      (distanciaMaxima > distancia[u] +
peso(u, v)) do
      distancia[v] = distancia[u] + peso(u, v)
      padre[v] = u
    fin if
  fin for
fin mientras
fin DijkstraTruncadoPorDistancia
    
```

Algoritmo 1 – Dijkstra truncado por distancia

```

DijkstraTruncadoPorDestino(Grafo G, nodo s,
nodo d)
for u ∈ V[G] do
  distancia[u] = ∞
  padre[u] = NULL
fin for
distancia[s] = 0
Encolar(cola, nodos(G))
mientras vacia(cola) ≠ true do
  u = extraer_minimo(cola)
  if (u==d)
    vaciar(cola)
  else
    for v ∈ adyacencia[u] do
      if (distancia[v] > distancia[u] + peso(u,
v)) do
        distancia[v] = distancia[u] + peso(u, v)
        padre[v] = u
      fin if
    fin for
  fin if
fin mientras
fin DijkstraTruncadoPorDestino
    
```

Algoritmo 2 – Dijkstra truncado por distancia

Las adaptaciones resultantes no cambian la complejidad temporal del algoritmo de Dijkstra, ya que introducen solo paradas tempranas, por lo que la cota superior de su complejidad, expresada en términos de O-grande, sigue siendo la misma teórica reportada por (11).

En cuanto al seguimiento de unidades de transporte, al ingresar en ZenderozWeb el usuario tiene presente la lista de rutas de transporte, cuya elección revela en el mapa las unidades de dicha ruta, el sentido y las paradas. Al hacer clic a uno de los marcadores que identifica las paradas, o al seleccionar una de la lista, el mapa se centra en la parada seleccionada y muestra el tiempo estimado para la llegada de la siguiente unidad (figura 6).

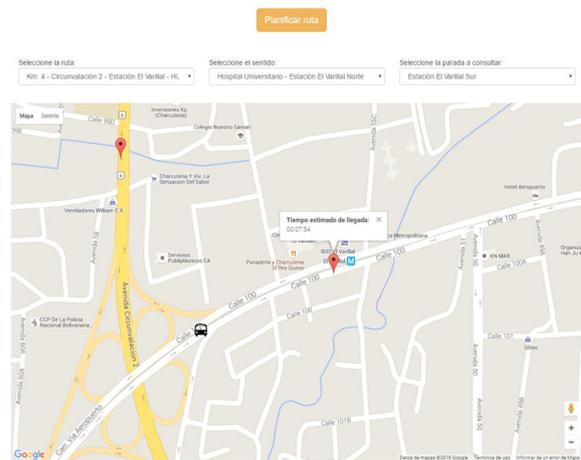


Figura 6 – Seguimiento de unidades vehiculares.

Debido a que no se contó con datos reales de las unidades para el momento del desarrollo y pruebas, se desarrolló un servicio que funcionaría aparte de ZenderozWeb, cuyo propósito es simular las posiciones (longitud y latitud) de cuatro unidades de transporte, siguiendo continuamente la ruta establecida, dada por una serie de puntos clave del mapa, cuyo trayecto en carretera fuese una línea recta e interpolando linealmente entre cada punto de manera que cada n segundos llegue al siguiente punto, simulando así también cambios de velocidades cuando los puntos están más lejos o más cerca que los anteriores. La intención es que pueda sustituirse este módulo por datos reales provistos por los institutos encargados de dichos servicios en Maracaibo, tales como IMTCUMA y METROMARA.

Conclusiones

El producto final presentado es un SIG que permitiría al público general en la ciudad de Maracaibo mejorar su vialidad cotidiana. La herramienta creada es capaz de planificar traslados con rutas de transporte público y seguimiento en tiempo real de unidades con GPS.

Se logró desarrollar un servicio web RESTful que responde los planes de la aplicación base: Zenderoz, la cual utiliza técnicas computacionales óptimas de representación de grafos, poda y cálculo de caminos más cortos.

Se desarrolló una interfaz web cómoda y sencilla para ser vista tanto en computadora de escritorio como en dispositivo móvil.

Se construyó un módulo de rastreo de unidades de transporte público capaz de seguir la posición y velocidad promedio de cada unidad.

Las pruebas del sistema fueron exitosas en cada una de sus funcionalidades.

Se recomienda el desarrollo de módulos de administración para la edición y extensión de las capas cartográficas y de las rutas de transporte público.

Finalmente, se recomienda también crear alianzas con los institutos gubernamentales pertinentes para promover la implantación de la herramienta creada y su uso por el público en general, así como alianzas para alcanzar la información contenida en este SIG a las bases de conocimiento de Google Maps.

Referencias bibliográficas

1. KANG-TSUNG C. *Introduction to Geographic Information Systems*. McGraw-Hill Higher Education. New York (USA). 464 pp. 2018.
2. <https://www.esri.com/news/arcwatch/0310/haiti.html>. Fecha de consulta: 05/12/2018.
3. <https://blog.nationalgeographic.org/2012/07/02/how-crisis-mapping-saved-lives-in-haiti/>. Fecha de consulta: 05/12/2018.
4. HARDER C. *The ArcGIS Book: 10 Big Ideas about Applying The Science of Where*. Esri Press. Redlands (USA). 172 pp. 2017.
5. <https://developers.google.com/maps/documentation>. Fecha de consulta: 05/12/2018.
6. CHÁVEZ A. Asistente Automatizado de Planeamiento de Rutas de Transporte Terrestre en Maracaibo (Para obtener el título de Licenciada en Computación). Facultad Experimental de Ciencias. Universidad del Zulia. Maracaibo (Venezuela). 64 pp. 2007.
7. AGUIAR A. Prototipo de un Planificador de Transporte Público Urbano (Para obtener el título de Licenciada en Computación). Facultad Experimental de Ciencias. Universidad del Zulia. Maracaibo (Venezuela). 77 pp. 2009.
8. <https://www.youtube.com/watch?v=d47JINsoNyo>. Fecha de consulta: 05/12/2018.
9. CELEDÓN M., PIRELA MORILLO G. V *Congreso Venezolano y VI Jornadas Nacionales de Investigación Estudiantil*. 2621-2628. Maracaibo (Venezuela). 2016.
10. CELEDÓN M. Herramienta web para planificación de rutas y seguimiento de unidades de transporte urbano (Para obtener el título de Licenciado en Computación). Facultad Experimental de Ciencias. Universidad del Zulia. Maracaibo (Venezuela). 64 pp. 2017.
11. CORMEN T., LEISERSON C., RIVEST R., STEIN C. *Introduction to Algorithms*. The MIT Press. Cambridge (USA). 1251 pp. 2010.
12. <https://www.GPS.gov>. Fecha de consulta: 05/12/2018.
13. <http://www.movable-type.co.uk/scripts/latlong.html>. Fecha de consulta: 05/12/2018.



UNIVERSIDAD
DEL ZULIA

CIENCIA

Vol.27 N°1, 2

*Esta revista fue editada en formato digital y publicada en junio de 2019, por el **Fondo Editorial Serbiluz**, Universidad del Zulia. Maracaibo-Venezuela*

www.luz.edu.ve
www.serbi.luz.edu.ve
produccioncientifica.luz.edu.ve