

Sistema inteligente para el monitoreo automatizado del transporte público en tiempo real

Yadira Quiñonez¹, Carmen Lizarraga¹, Juan Peraza¹, Oscar Zatarain¹

yadiraqui@uas.edu.mx, carmen.lizarraga@uas.edu.mx, jfperaza@uas.edu.mx, ozatarain@uas.edu.mx

¹ Universidad Autónoma de Sinaloa, Facultad de Informática, Av. Leonismo Internacional S/N y Av. de los Deportes, Ciudad Universitaria, 80000 Mazatlán, Sinaloa, México.

DOI: 10.17013/risti.31.94-105

Resumen: Este artículo propone un prototipo totalmente funcional para el monitoreo del transporte público, el principal objetivo es proporcionar información en tiempo real a los usuarios que utilizan los autobuses como medio de transporte, y al mismo tiempo, proporcionar datos históricos de cada unidad para conocer la información relacionada con los horarios y retrasos de las rutas realizadas en los diferentes circuitos; para lograr el objetivo se ha utilizado un conjunto de recursos y medios tecnológicos que se centran en las capacidades de una computadora de placa reducida Raspberry PI. De manera general, el prototipo está formado por cuatro módulos: el rastreador de GPS, que consta de dos componentes principales: una Raspberry PI y un módulo GPS NEO-6M Ublox. El servicio web y la base de datos se implementan en PostgreSQL, que almacena la información a través de un servicio web RestFul desarrollado en PHP. La aplicación web para la gestión de seguimiento de bus está desarrollada en PHP y utiliza el marco Bootstrap para la interfaz de usuario. Finalmente, la aplicación móvil para Android destinada a usuarios finales, se desarrolló en la plataforma de Visual Studio 2015 con Xamarin y se programó en C#.

Palabras-clave: Raspberry Pi, GPS, Base de Datos para el Monitoreo de Transporte, Servicio Web.

Intelligent system for automated monitoring of public transport in real time

Abstract: This article proposes a fully functional prototype for the monitoring of public transport, the main objective is to provide information in real time to users who use buses as a means of transport, and at the same time, provide historical data for each unit to know the information related to the schedules and delays of the routes made in the different circuits; to achieve the objective, a set of resources and technological means have been used that focus on the capabilities of a reduced-scoreboard computer Raspberry PI. In general, the prototype consists of four modules: the GPS tracker, which consists of two main components: a Raspberry PI and a GPS module NEO-6M Ublox. The web service and the database are

implemented in PostgreSQL, which stores the information through a RestFul web service developed in PHP. The web application for bus tracking management is developed in PHP and uses the Bootstrap framework for the user interface. Finally, the mobile application for Android intended for end users, was developed on the Visual Studio 2015 platform with Xamarin and was programmed in C #.

Keywords: Raspberry Pi, GPS, Transportation Monitoring Database, Web Service.

1. Introdução

En las últimas décadas, han surgido diferentes trabajos que abordan el desarrollo de estrategias e incorporación infraestructura para mejorar la seguridad y movilidad del transporte utilizando nuevas tecnologías. De manera colectiva, a estas tecnologías se conocen como Sistemas de Transporte Inteligente (STI), o ITS en sus siglas en inglés Intelligent Transportation Systems, que tiene como objetivo promover el desarrollo inteligente y sostenible, mejorar la eficacia y la productividad del transporte, mediante el uso e incorporación nuevas tecnologías de informática, información y telecomunicación (Lin, 2017, Piedra, 2018). De acuerdo con Chaturvedi y Srivastava (2017) proponen un sistema de STI que utiliza la red celular y la señal GPS para estimar la velocidad del vehículo bajo condiciones de tráfico heterogéneas haciendo uso infraestructura STI limitada. En otro trabajo (Quintero, Patino Alvarez & Aedo, 2016) proponen utilizar algoritmos de seguimiento de trayectorias para verificar si los vehículos de transporte están siguiendo las rutas definidas previamente, mediante el rastreo de la posición actual de la unidad de transporte.

En la actualidad, el transporte público es el medio de movilidad más utilizado, entonces, brindar información más precisa sobre horarios de llegada o retrasos de los vehículos ayuda a los usuarios a tomar buenas decisiones sobre los desplazamientos y mejorar el servicio brindado. Para lograr el objetivo de este desarrollo, la metodología empleada consistió en seleccionar los componentes adecuados que cumplieran con las características específicas y que fuera compatible para programar el algoritmo que permitiera la interacción correcta de todas las partes del sistema.

En este sentido, el objetivo de este desarrollo tecnológico es proporcionar información en tiempo real a los usuarios que utilizan los autobuses como medio transporte, y al mismo tiempo, facilitar datos históricos de cada unidad para conocer toda la información relacionada con los recorridos realizados en los diferentes circuitos y tomar decisiones inteligentes sobre los desplazamientos. Esto supone un mejor servicio atendiendo la problemática social como lo es el transporte público, en relación a la seguridad, se tendrá un control de cada vehículo en tiempo real, obteniendo la velocidad instantánea con el fin de evitar excesos de velocidad en zonas urbanas y reducir accidentes.

El presente trabajo aborda la temática de rastreo de unidades de transporte urbano mediante la utilización de diferentes dispositivos y recursos tecnológicos, tales como: la Raspberry Pi, GPS, Servicios Web y Base de Datos, que en conjunto conforman un sistema de monitoreo automatizado. Se abordan puntos técnicos del funcionamiento del sistema, así como los recursos utilizados para su desarrollo.

De manera general, la automatización de procesos busca reducir costes, incrementar la calidad y mejorar la eficiencia ante las necesidades presentadas en el mercado (González-

Filgueira, 2018). Actualmente, hay diversas aplicaciones que utilizan la tecnología raspberry pi para llevar a cabo diferentes tareas de automatización. En los trabajos (Sandeep, 2015) y (Jamil, 2015) proponen un sistema para el funcionamiento automático y control de aparatos electrodomésticos de forma remota utilizando la raspberry pi. En (Dong, 2015) presentan un método para la detección de inactividad de una alarma en tiempo real implementada en la raspberry pi. En (Ikhankar, 2015) utilizan la raspberry pi para crear un robot con un sistema de vigilancia en tiempo real. En (Nguyen, Loan, Mao & Huh, 2015) proponen la implementación de un sistema de monitoreo de bajo costo mediante una raspberry pi, reduciendo los defectos de los actuales sistemas de vigilancia, tales como: imagen borrosa o anomalías de identificación.

Este artículo se encuentra estructurado de la siguiente manera sección 2 se describen las herramientas de implementación que se han utilizado para llevar a cabo el sistema de monitoreo automatizado. En la sección 3 se describe brevemente la descripción de cada uno de los módulos del sistema. La sección 4 describe el esquema general de funcionamiento del sistema de monitoreo. En la sección 5 se presentan los resultados experimentales y las pruebas realizadas del funcionamiento del sistema en tiempo real. Finalmente, la sección 6 consta las conclusiones sobre el sistema y sobre los trabajos que se tiene planeados para el futuro.

2. Herramientas de Implementación

En la actualidad, existen diferentes tecnologías de bajo coste que permite desarrollar diferentes sistemas de una manera sencilla, algunas de estas tecnologías son: Arduino (Arduino, 2018), Raspberry Pi (RaspberryPi, 2018), OpenPicus (OpenPicus, 2018), Cubieboard (Cubieboard, 2018), Udoo (Udoo, 2018), entre otras. Sin embargo, las tecnologías más utilizadas y conocidas son Arduino, OpenPicus y Raspberry Pi.

Este trabajo se enfoca en el uso de la tecnología raspberry pi (ver Fig.1). La placa raspberry pi (RaspberryPi, 2018) fue creada en Reino Unido por la Fundación Raspberry Pique, es básicamente una microcomputadora de bajo coste que cuenta con un sistema operativo con tres diferentes distribuciones basadas en Linux, con memoria RAM, cuenta con entradas y salidas para teclado, monitor, ratón y cable Ethernet. Al ser software libre, el código fuente está accesible para cualquier usuario pueda estudiarlo o modificarlo,

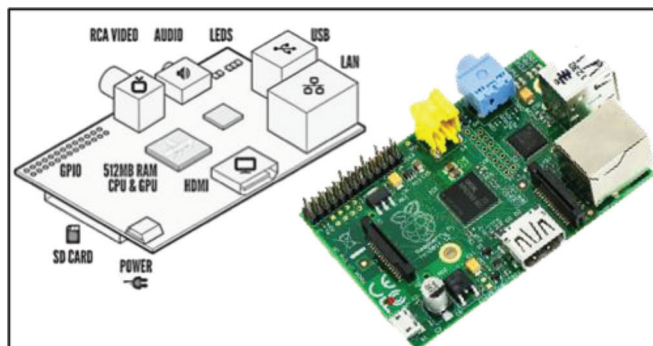


Figura 1 – Componentes de la raspberry pi

por lo tanto, la instalación de la distribución adecuada se realiza de manera sencilla a partir de una tarjeta SD con conocimientos básicos de Linux, solo es necesario seguir las instrucciones indicadas.

En la tabla 1 se muestra un resumen de las características del modelo B de raspberry pi, esta plataforma ofrece capacidades avanzadas de procesamiento debido a la potencia del CPU y GPU, recomendable cuando se va a procesar una n cantidad de datos.

Características	Descripción
<i>CPU</i>	ARM 1176JZF a 900 MHz
<i>GPU</i>	Broadcom VideoCore IV, OpenGL ES 2.0, MPEG-2 y VC-1 (con licencia), 1080p 30H.264/MPEG-4 AVC
<i>Puertos USB 2.0</i>	2 (vía hub USB integrado)
<i>Entradas de vídeo</i>	Conector MIPI CSI que permite instalar un módulo de cámara desarrollado por la RP
<i>Salidas de vídeo</i>	Conector RCA (PAL y NTSC), HDMI (rev1.3 y 1.4), Interfaz DSI para panel LCD
<i>Salidas de audio</i>	Conector de 3.5 mm, HDMI
<i>Almacenamiento Integrado</i>	SD / MMC / ranura para SDIO
<i>Conectividad de la red</i>	10/100 Ethernet (RJ-45) vía hub USB
<i>Consumo energético</i>	700 mA, (3.5 W)
<i>Dimensiones</i>	85.60mm × 53.98mm
<i>Sistemas operativos soportados</i>	GNU/Linux: Debian (Raspbian), Fedora (Pidora), Arch Linux (Arch Linux ARM), Slackware Linux. RISC OS2

Tabla 1 – Descripción de las características de la Raspberry Pi

3. Descripción de los módulos del sistema

El prototipo para el monitoreo del transporte público consta de manera general de cuatro módulos principales, a continuación hace una descripción de cada uno de los módulos.

Rastreador GPS: el sistema para el monitoreo automatizado de transporte público que tiene como objetivo principal proporcionar información en tiempo real a los usuarios que utilizan los autobuses como su medio transporte, y al mismo tiempo, facilitar datos históricos de cada unidad para conocer toda la información relacionada con los recorridos realizados en los diferentes circuitos y tomar decisiones inteligentes sobre los desplazamientos; todo esto se logra utilizando un conjunto de recursos y medios tecnológicos que se centran en las capacidades de una computadora de placa reducida raspberry pi.

El rastreador GPS funciona bajo un algoritmo desarrollado en Python, el cual obtiene, analiza, procesa y almacena los datos emitidos por el módulo GPS en el servidor. Además, de llevar a cabo una serie de controles con el fin de brindar estabilidad al sistema. Básicamente está compuesto por una computadora de placa reducida raspberry pi (Fig.2), la cual se encuentra cargada con un sistema operativo Rasbian en su versión

Jessie en conjunto con un módulo GPS Ublox modelo neo-6m. Está implementado en distintas plataformas de desarrollo, cada una de ellas atendiendo a la necesidad de cada aplicación. Se utilizaron lenguajes como PHP, JavaScript, C#, Python y Frameworks como Bootstrap.

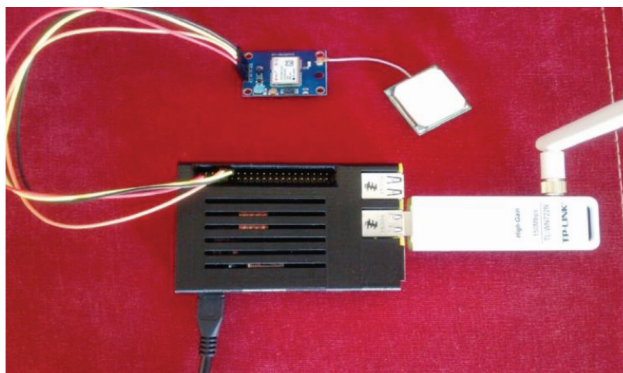


Figura 2 – Dispositivo raspberry pi con módulo GPS Ublox conectado

Con la tecnología GPS se logra un alto grado de precisión y aplicado a diversas áreas (Linli & Zhangyi, 2011). El rastreo por GPS es una de las herramientas indispensables para distintos usos como encontrar un automóvil extraviado, rastrear las unidades de paquetería e inclusive los dispositivos móviles cuentan con la opción de rastreo con el fin de saber tu ubicación o encontrar el dispositivo en caso de extravío. En este sentido, el módulo GPS está enlazado a la raspberry mediante los pines de conexión que esta posee y mediante la instalación de paquetes GPS en la raspberry se obtiene la interfaz donde se puede apreciar los datos de rastreo como latitud, longitud, altitud, velocidad, entre otros datos.

En resumen, el dispositivo raspberry pi tiene como tarea interpretar los datos obtenidos por el dispositivo GPS, esto se realiza mediante un algoritmo desarrollado en Python, la funcionalidad de este algoritmo radica en obtener los datos del GPS y enviarlos al servidor de base de datos, la transmisión de la información se realiza mediante una comunicación inalámbrica mediante un módulo Gsm/Gprs utilizando datos móviles, esto es, una vez que el algoritmo procesa la información, es enviada a la base de datos para almacenarlo. En caso de que los datos móviles fallen el dispositivo de monitoreo está preparado para estar guardando de manera local un archivo de texto con las posiciones que le proporcione el GPS, esto para que no se pierda ningún dato y en el momento en el que se restablezca la señal se envían los datos al servidor.

Servicio web y servidor de base de datos: el funcionamiento básico del servicio web se da a partir del intercambio de datos o archivos en formato XML alojados en el servidor a través de servicios computacionales en la nube (Palos-Sánchez, 2017). Utiliza el protocolo HTTP para establecer la comunicación entre el servidor y el navegador. En este sentido, los servicios RestFul han demostrado que tienen buen potencial para el funcionamiento con este tipo dispositivos con un menor consumo de energía en los

servicios web (Nunes, 2014). En el sistema se ha utilizado RestFul como servicio web el cual se encuentra desarrollado en PHP, que permite obtener los datos del servidor de base de datos realizando una petición, donde devuelve los datos obtenidos en formato JSON y estos pueden ser utilizados por distintas aplicaciones.

Existen diversas alternativas a utilizar como bases de datos, las más utilizadas son: MySQL (MySQL, 2018), SQLite (SQLite, 2018) y PostgreSQL (PostgreSQL, 2018), entre otras. Para este trabajo se ha optado por utilizar la base de datos de tipo PostgreSQL porque es de código abierto, utiliza un modelo cliente/servidor y usa multiprocesos en lugar de multi-hilos con fin de garantizar la estabilidad, robustez y facilidad de administración. Entonces, una vez que se obtiene la información por el dispositivo de monitoreo, se envía al servidor de base de datos alojado en PostgreSQL que es un motor de base de datos muy estable, en él se guardan datos como latitud, longitud, velocidad entre otras cosas, con fin de que la información requerida para el funcionamiento del sistema es almacenada y obtenida por los distintos dispositivos y aplicaciones mediante un Servicio Web RestFul desarrollado en PHP.

Aplicación para la administración del rastreo de autobuses: Es necesario que cada unidad de transporte cuente con un dispositivo de rastreo, para que pueda ser monitoreada desde una aplicación web, esta aplicación web está desarrollada en PHP y se utiliza el framework Bootstrap para la interfaz de usuario. La aplicación web está segmentada en dos partes Autobuses Históricos y Monitoreo en tiempo real. Cabe mencionar, que las rutas de los autobuses se encuentran definidas y están guardadas en la base de datos y las cuales se encuentran divididas en IDA y VUELTA.

Autobuses Histórico. En la Figura 3 se muestra un listado de los autobuses registrados en el sistema realizando una petición al servicio web y las rutas realizadas por cada unidad. Al seleccionar un autobús se solicita la fecha y después muestra un mapa que

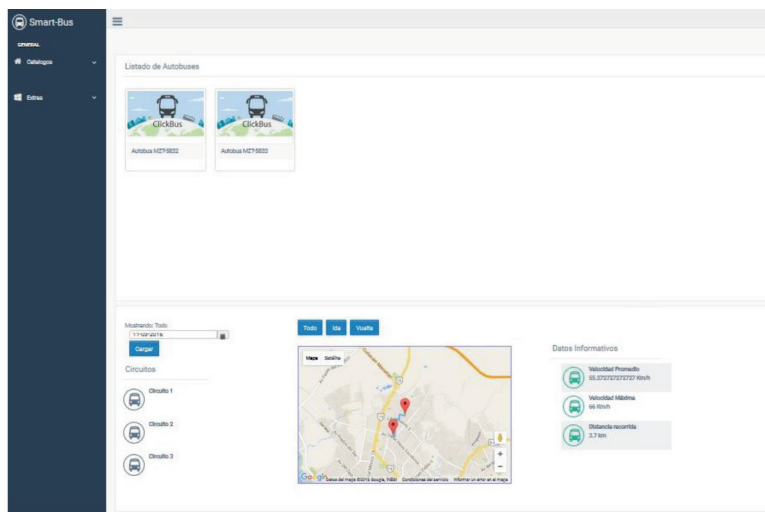


Figura 3 – Autobuses histórico

está implementado con la API de Google Maps con la ruta trazada del recorrido del autobús en el día especificado, se puede seleccionar la visualización del circuito completo del autobús, solo la IDA o en su caso solo la VUELTA, cada una de las opciones muestra la velocidad promedio, velocidad máxima y la distancia recorrida en km. Esto con el fin de saber si el autobús se desvió de la ruta o excedió el límite de velocidad permitido.

Monitoreo en Tiempo Real. Al igual que Autobuses Histórico muestra un listado de autobuses, pero en este caso solo los que se encuentren en servicio, al seleccionar algún autobús se carga el mapa con la posición actual de la unidad y se actualiza cada 5 segundos, al mismo tiempo, muestra la velocidad que lleva el autobús en ese momento.

Aplicación móvil para usuarios finales: se cuenta con una aplicación móvil llamada Smart BS destinada para usuarios finales, esta aplicación está desarrollada en la plataforma de Visual Studio 2015 utilizando Xamarin y programada en C#. Esta aplicación (Fig. 4) se encuentra estructurada en dos partes la primera es la visualización de las rutas de autobuses en tránsito y la segunda es activación de alarmas.

La visualización de rutas aparecen las rutas de autobuses de acuerdo a la ubicación geográfica, a partir de ahí, se selecciona la ruta de la cual se quiere obtener información. Una vez seleccionada la ruta se muestran las paradas oficiales del autobús, al seleccionar alguna de las paradas se carga un mapa mostrando la unidad próxima que va a pasar por el punto especificado y el tiempo estimado para la llegada del autobús.

La aplicación cuenta con alertas, que tienen como propósito avisarle al usuario cuando el autobús está próximo a llegar (Fig.5). Esto se realiza seleccionando la ruta y la parada del autobús, posteriormente, se tiene que especificar la hora que desea tomar vehículo y el tiempo de anticipación, para que el dispositivo le avise mediante una alerta cuando el vehículo le falte el lapso de tiempo especificado para pasar por la parada de autobús.



Figura 4 – Aplicación móvil para Android

Esto con el fin de que el usuario de los transportes públicos tenga una mejor organización del tiempo, de este modo, podrá conocer los tiempos de horarios y retrasos de los vehículos, de esta forma, decidir los tiempos en que desea salir a la parada de autobús.

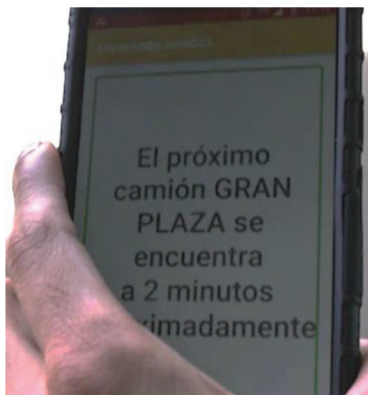


Figura 5 – Alertas para usuarios

4. Esquema general del desarrollo del sistema

El desarrollo del sistema está compuesto por tres etapas: la investigación para el desarrollo de la sistema, estructura o módulos del sistema y la planeación de procesos para su correcto desarrollo. A continuación, en la Figura 6 se muestra el diagrama de desarrollo del sistema para el monitoreo automatizado del transporte público y una breve explicación sobre cada segmento.

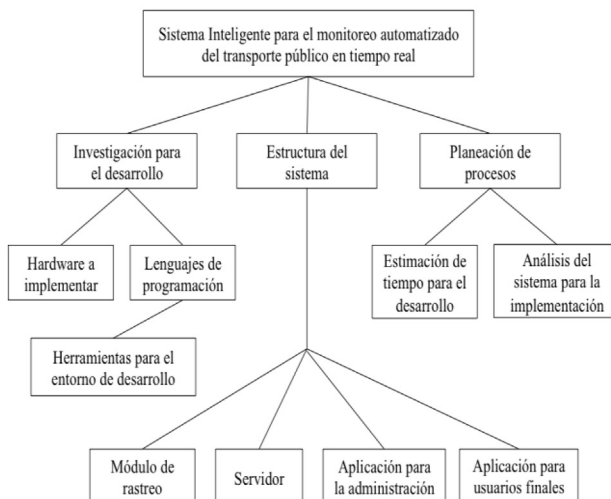


Figura 6 – Esquema general

La investigación del desarrollo se refiere a los dispositivos y al entorno de desarrollo a utilizar, es decir, al estudio del Hardware y al Software que se han implementado, esto se realiza mediante la definición del problema y las herramientas adecuadas a utilizar que cubran las necesidades de la manera más acorde de acuerdo al requerimiento.

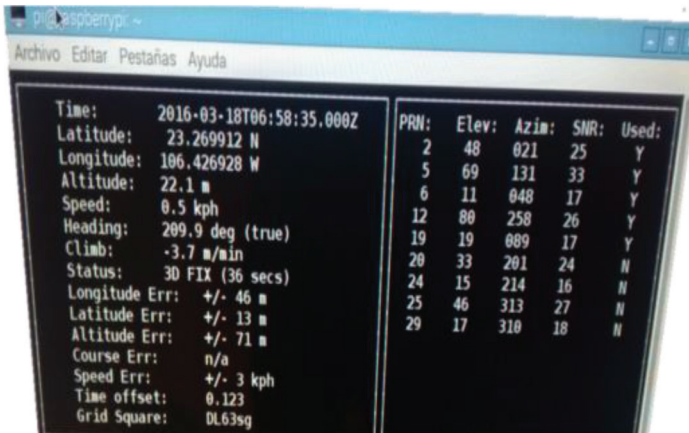
La estructura de sistema se refiere a los módulos que lo conforman, ya que el sistema esta segmentado en diversas aplicaciones, cada una de ellas se encargan de realizar los procesos y funciones necesarias para el correcto funcionamiento.

Al final, se encuentra la planeación de procesos que consiste en estimar tiempos de desarrollo para cada módulo, y su respectiva asignación a la persona que se encarga del desarrollo, al igual que el análisis del sistema, como la funcionalidad deseada y que es lo que se quiere obtener al finalizarla.

5. Resultados experimentales

Para conseguir el correcto funcionamiento del sistema y visualizar la información del trayecto en tiempo real del autobús, así como velocidad y frecuencia aproximada (Fig. 7); se han realizado diversos experimentos para evaluar estabilidad del algoritmo implementado y la respuesta del servidor a las aplicaciones que consumen sus servicios, con el fin de conocer el rendimiento del sistema.

Los primeros experimentos realizados fueron dentro del campus universitario con vehículo personal, posteriormente se hicieron en la ciudad con las rutas propias de los autobuses para medir el rendimiento del algoritmo utilizado y comprobar la viabilidad del proyecto.



Time: 2016-03-18T06:58:35.000Z	PRN: Elev: Azim: SNR: Used:
Latitude: 23.269912 N	2 48 021 25 Y
Longitude: 106.426928 W	5 69 131 33 Y
Altitude: 22.1 m	6 11 048 17 Y
Speed: 0.5 kph	12 80 258 26 Y
Heading: 209.9 deg (true)	19 19 009 17 Y
Climb: -3.7 m/min	20 33 201 24 N
Status: 3D FIX (36 secs)	24 15 214 16 N
Longitude Err: +/- 46 m	25 46 313 27 N
Latitude Err: +/- 13 m	29 17 310 18 N
Altitude Err: +/- 71 m	
Course Err: n/a	
Speed Err: +/- 3 kph	
Time offset: 0.123	
Grid Square: DL63sg	

Figura 7 – Funcionamiento del GPS en la placa raspberry pi

Primero, se han realizado pruebas para corroborar el funcionamiento correcto del dispositivo raspberry pi en conjunto con el rastreador GPS (Fig. 7), en este sentido, se conectó una pantalla a la raspberry pi para comprobar que los datos obtenidos de la señal

GPS correspondían a la posición en la que se encontraba el dispositivo, para visualizar de manera gráfica la posición se ha utilizado Google Maps (Fig. 8) y se ha verificado la posición actual era la correcta.

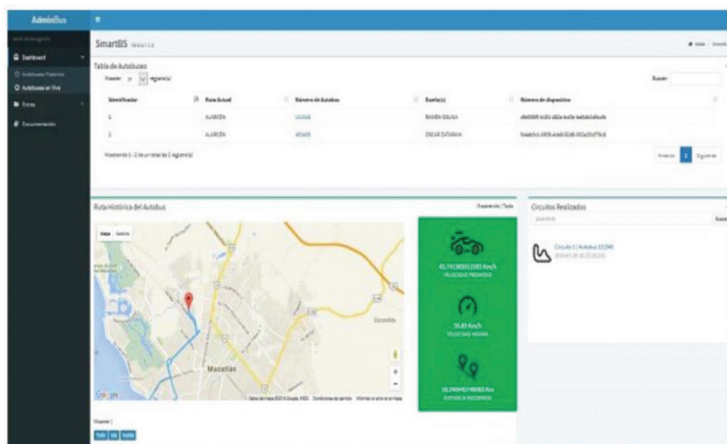


Figura 8 – Visualización de las rutas en Google Maps

Se ha realizado un análisis sobre el consumo de los datos móviles del dispositivo de rastreo con el fin de conocer el promedio de datos (megabyte) utilizados por el rastreador GPS, en la Fig. 9 se muestran los resultados obtenidos en relación a los megabytes consumidos por horas, como se puede observar, el consumo de datos del dispositivo es variable porque depende totalmente del nivel de uso, es decir, el consumo promedio en megabyte depende de las peticiones que se hacen al servidor.

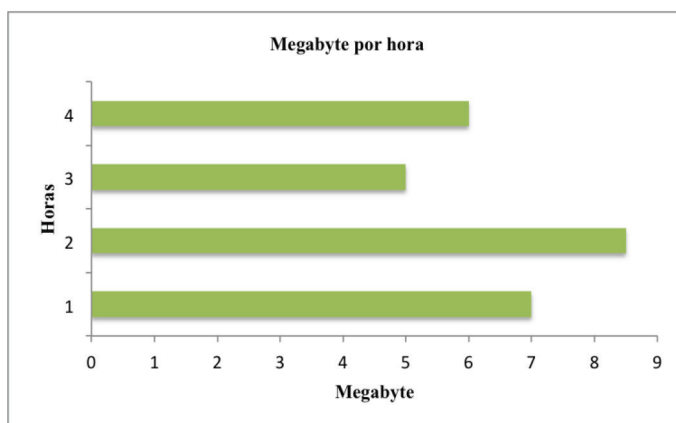


Figura 9 – Megabytes consumidos por hora.

6. Conclusiones

En la actualidad, el transporte público en México y en otras partes del mundo se está convirtiendo en un medio de transporte cada vez más importante y complejo, en este sentido, existe la necesidad de desarrollar tecnología adecuada para proporcionar información de los tiempos de desplazamiento a los usuarios en tiempo real para que puedan trasladarse por las ciudades utilizando el transporte público. Consideramos que este prototipo de monitoreo automatizado es de gran utilidad para los usuarios ya que se tendrá acceso a la red de autobuses en lo que respecta a trayectos, horarios y retrasos.

Con este sistema de monitoreo automatizado se intenta contribuir a llevar un control sobre cada unidad activa con el fin de poder brindar un mejor servicio, es importante mencionar que los componentes utilizados son de bajo coste y se obtiene buen rendimiento, solo que es fundamental llevar a cabo pruebas exhaustivas para poder determinar la estabilidad del sistema a implementar. Todo esto con la finalidad de optimizar los largos periodos de espera de los usuarios, donde los principales beneficiarios serán todas aquellas personas que utilizan los autobuses públicos como medio de transporte.

Smart BS es una aplicación móvil de desarrollo propio bajo la plataforma Android, la cual utiliza un diseño estándar de acuerdo a los lineamientos de Google Material Design para que los usuarios identifiquen los elementos de la interfaz de manera fácil e intuitiva. Al utilizar este estándar de diseño se garantiza que la aplicación sea muy familiar para los usuarios tales como las interfaces gráficas de aplicaciones más populares como WhatsApp y Facebook se encuentran diseñadas bajo estas reglas.

Referencias

- Arduino. (2018). Recuperado de: <https://www.arduino.cc/>.
- Chaturvedi, M., & Srivastava, S. (2017). Multi-Modal Design of an Intelligent Transportation System. *Transactions on Intelligent Transportation Systems*, 18(8), 2017-2027.
- CubieBoard. (2018). A series of open source hardware. Recuperado de <http://cubieboard.org/>.
- Dong, Q., Yang, Y., Hongjun, W., & Jian-Hua, X. (2015). Fall alarm and inactivity detection system design and implementation on raspberry pi. In: *International Conference on Advanced Communication Technology*, (pp. 382-386). Seoul, South Korea: IEEE publishing.
- González-Filgueira, G., & Rodríguez, F.G. (2019). Automatización de una planta industrial de alimentación mediante control distribuido. *Revista Ibérica de Sistemas y Tecnologías de Información*, 27(6), 1-17.
- Ikhankar, R., Kuthe, V., Ulabhaje, S., Balpande, S., & Dhadwe, M. (2015). Pibot: The raspberry pi controlled multi-environment robot for surveillance and live streaming. In: *International Conference on Industrial Instrumentation and Control*, (pp. 1402-1405). Pune, India: IEEE publishing.

- Jamil, M. M. A., & Ahmad, M. S. (2015). A pilot study: Development of home automation system via raspberry Pi. In: International Conference on Biomedical Engineering, (pp. 1-4). Penang, Malaysia: IEEE publishing.
- Lin, Y., Wang, P., & Ma, M. (2017). Intelligent Transportation System (ITS): Concept, Challenge and Opportunity. In: International Conference on Big Data Security on Cloud, (pp. 167-172). Beijing, China: IEEE publishing.
- Linli, W. & Zhangyi, S., (2011). Difference analysis of GPS data base sources based on vehicle location system. In: International Conference on Computer Research and Development, (pp. 421-425). Shanghai, China: IEEE publishing.
- MySQL. (2018). Recuperado de: <https://www.mysql.com/>.
- Nguyen, H.Q., Loan, T. T. K., Mao, B. D., & Huh, E.N. (2015). Low cost real-time system monitoring using raspberry pi. In: International Conference on Ubiquitous and Future Networks, (pp. 857-859). Sapporo, Japan: IEEE publishing.
- Nunes, L.H., Nakamura, L.H.V., Vieira, H.F., Libardi, R.M.O., de Oliveira, E.M., Estrella, J.C., & Reiff-Marganiec, S. (2014). Performance and energy evaluation of RESTful web services in Raspberry Pi. In: International Performance Computing and Communications Conference, (pp. 1-9). Austin, Texas, USA: IEEE publishing.
- OpenPicus Microchip Technology. (2018). Recuperado de: <http://www.microchip/openpicus.com/>.
- Palos-Sánchez, P.R., Arenas-Márquez, F.J., & Aguayo-Camacho, M. (2017). La adopción de la tecnología cloud computing (SaaS): efectos de la complejidad tecnológica vs formación y soporte. Revista Ibérica de Sistemas y Tecnologías de Información, 22(6), 89-105.
- Piedra, N., & Suárez J.P. (2018). Hacia la interoperabilidad semántica para el manejo inteligente y sostenible de territorios de alta biodiversidad usando SmartLand-LD. Revista Ibérica de Sistemas y Tecnologías de Información, 26(3), 104-121.
- PostgreSQL (2018). PostgreSQL: The world's most advanced open source database. Recuperado de: <https://www.postgresql.org/>.
- Quintero, Y. A., Patino Alvarez, G. A., & Aedo, J. E. (2016). Path-Tracking Algorithm for Intelligent Transportation System. IEEE Latin America Transactions, 14(6), 2934-2939.
- Sandeep, V., Gopal, K. L., Naveen, S., Amudhan, A., & Kumar, L. S. (2015). Globally accessible machine automation using raspberry pi based on Internet of Things. In: International Conference on Advances in Computing, Communications and Informatics, (pp. 1144-1147). Kochi, India: IEEE publishing.
- SQLite (2018). SQLite Home Page. Recuperado de: <http://www.sqlite.org/>.
- Rapsberry Pi (2018). Teach, Learn, and Make with Rapsberry Pi. Recuperado de: <https://www.raspberrypi.org/>.
- UDOO (2018). UDOO: All you need mini PC Android + Linux + Arduino. Recuperado de: <https://www.udoo.org/>.