

UNIVERSIDAD NACIONAL AUTÓNOMA DE NICARAGUA

UNAN-LEÓN

FACULTAD DE CIENCIAS Y TECNOLOGÍA



Sistema de apoyo al servicio de transporte urbano colectivo de la ciudad de León mediante la API de Google Maps v3.

Monografía para optar al Título de Ingeniero en Telemática.

Autores:

Br. Wanderlee Joseph Cano Gríos.

Br. Jorge Alí Urbina Reyes.

Br. Nelson José Velásquez Castillo.

Tutor:

Msc. Julio César González Moreno.

León, Diciembre del 2012

**Sistema de apoyo al servicio de transporte urbano colectivo
de la ciudad de León mediante la API de Google Maps v3**

Resumen

La evolución de la tecnología informática y las telecomunicaciones ha otorgado a Internet y al World Wide Web un papel predominante en el mundo tecnológico y económico. Las aplicaciones en Internet han modificado la forma cotidiana de obtener información de un determinado sitio, realizar negocios e intercambios comerciales y comunicarse.

En este trabajo se presenta el desarrollo de un sistema web de apoyo al servicio de transporte urbano colectivo de la ciudad de León, empleando la API de Google Maps en su versión 3 (GMaps v3), el cual ha sido nombrado como SaTUC.

Este sistema implementa acciones como: visualización de recorridos, visualización de puntos de parada, información asociada a la ruta que cubre dicho recorrido, creación de nuevos recorridos, edición de recorridos, creación de puntos de parada, eliminación de puntos de parada, entre otras, sobre una determinada ruta de transporte urbano colectivo, esto permite a las personas obtener una información detallada sobre una ruta específica.

Cabe destacar que el sistema está basado en la implementación de las acciones antes mencionadas a través de mapas interactivos, para ello se hace uso de la API de GMaps v3 para poder incorporar dichas funcionalidades. La visualización de los mapas se realizará mediante un navegador web a través de una dirección web disponible en Internet para cualquier ordenador o dispositivo móvil.

Los usuarios potenciales del sistema serán todos aquellos que cuenten con acceso a Internet y posean una necesidad demostrada de conocer u obtener información sobre el recorrido y los puntos de parada ofrecidos por el servicio de transporte urbano colectivo de la ciudad de León.

El sistema se encuentra dividido en dos módulos:

- **El módulo de administración:** este módulo está dirigido a la administración y gestión de la información asociada a cada ruta. El administrador puede manipular la información del sistema para realizar cambios en una o todas las rutas pre-cargadas; así como también agregar nueva información o nuevas rutas.
- **El módulo de interfaz pública:** este módulo está dirigido a la visualización de la información, asociada a las rutas, que es cargada por el administrador del sistema desde el módulo antes descrito. El usuario puede interactuar con diversas opciones que le permiten mostrar u ocultar información asociada a las rutas; así como también una visualización general de todas las rutas o una visualización de una ruta específica.

La información asociada a cada una de las rutas que han sido pre-cargadas en el sistema fue suministrada según los planos de recorrido oficial de las rutas de transporte urbano colectivo de

la ciudad de León para la fecha: 25 de enero del 2012, los cuales fueron facilitados por la Alcaldía de ésta ciudad.

Agradecimientos

Le agradecemos primeramente a Dios; por darnos la vida y por permitirnos llegar a culminar nuestro trabajo.

A nuestros padres por su ayuda incondicional, dedicación y confianza en nosotros, sin ellos no podríamos haber llegado hasta aquí.

A nuestro tutor Msc. Julio César Moreno por su esfuerzo, dedicación. Sus orientaciones, su manera de trabajar, su persistencia.

A nuestros Profesores ya que nos brindaron sus conocimientos, ayuda y paciencia en lo largo de este camino universitario.

A nuestros familiares y amigos por su apoyo y motivación para que saliéramos adelante.

Dedicatoria

Le agradecemos **primeramente a Dios** por habernos regalado la vida y la fuerza que nos impulsaron a seguir adelante a pesar de todas las dificultades.

Con mucho amor a **nuestros padres** por habernos dado todo lo que necesitábamos y estar a nuestro lado brindándonos su amor, apoyo y fortaleza.

A nuestras **familias**, porque simple y sencillamente se lo merecen.

A nuestras amistades, por apoyarnos siempre.

A nuestros **maestros**, por transmitirnos la sabiduría y el conocimiento para triunfar en la vida.

Índice General.

1. INTRODUCCIÓN	10
1.1 Definición del problema.....	10
1.2 ANTECEDENTES.....	11
1.3 JUSTIFICACIÓN.....	13
1.4 OBJETIVOS.....	14
1.4.1 OBJETIVO GENERAL.....	14
1.4.2 OBJETIVOS ESPECÍFICOS	14
2. MARCO CONCEPTUAL	15
2.1 Google Maps	15
2.2 Google Maps JavaScript API.....	16
2.2.1 Tipos de mapas	16
2.2.2 Aspectos generales de las superposiciones	18
2.2.2.1 Marcadores.....	18
2.2.2.2 Polilíneas.....	19
2.2.2.3 Ventanas de información	19
2.3 JavaScript	20
2.4 PHP.....	21
2.4.1 Características	21
2.5 MySQL.....	21
2.5.1 Características	22
2.6 JQuery	22
2.6.1 Características	23
2.7 HTML.....	23

2.7.1	Elementos.....	23
2.8	JSON	24
2.9	XML.....	24
2.9.1	Ventajas del XML.....	25
2.9.2	Estructura de un documento XML.....	25
2.10	CSS	27
2.11	Patrón de diseño Model View Controller (MVC)	27
3.	Análisis y Diseño del sistema	29
3.1	Descripción general del sistema.....	29
3.2	Funcionalidades generales del sistema.....	30
3.2.1	Funciones enfocadas al módulo de interfaz pública	30
3.2.2	Funcionalidades enfocadas al módulo administrador	31
3.3	Diagrama de casos de uso.....	33
3.3.1	Módulo de interfaz pública	34
3.3.2	Módulo administrador.....	36
3.4	Arquitectura del sistema.....	37
3.5	Secuencia del sistema.....	39
3.5.1	Diagramas de secuencia.....	39
3.6	Base de datos del sistema	42
4.	Conclusiones	45
5.	Recomendaciones	46
6.	Bibliografía	47
Anexos	48
1.	Interfaces principales del sistema	48

Tabla de ilustraciones.

Ilustración 1:	Tipo de mapa ROADMAP(mapa).....	17
Ilustración 2:	Tipo de mapa SATELLITE(satélite).....	17
Ilustración 3:	Icono que representa una parada en el sistema.	18
Ilustración 4:	Inicio del recorrido de la ruta 103 en forma de una polilínea.	19
Ilustración 5:	Infowindow que muestra una imagen y descripción de una parada.....	20
Ilustración 6:	Patrón de diseño MVC.	28
Ilustración 7:	Vista del módulo de interfaz pública.....	30
Ilustración 8:	Menú principal.	31
Ilustración 9:	Vista del modulo administrador.	32
Ilustración 10:	Diagrama de casos de uso del módulo interfaz pública.....	33
Ilustración 11:	Diagrama de casos de uso módulo administrador.	34
Ilustración 12:	Arquitectura MVC.....	38
Ilustración 13:	Diagrama de Base de datos.....	43
Ilustración 14:	Vista de todos los recorridos y un Infowindow desplegado.	48
Ilustración 15:	Visualización de una ruta.	49
Ilustración 16:	Como se agregar una parada a un recorrido.	49
Ilustración 17:	Como se agrega un recorrido al sistema.....	50
Ilustración 18:	Tabla para editar los recorridos.	50
Ilustración 19:	Formulario para editar un recorrido.....	51

1. INTRODUCCIÓN

1.1 Definición del problema

Hasta antes de la fecha de culminación del presente trabajo, si una persona necesitaba obtener información referente a los recorridos de todas las unidades de transporte urbano colectivo de la ciudad de León tenía que ir a las oficinas de la Alcaldía de dicha ciudad y solicitar la información. El departamento de transporte de la municipalidad posee una serie de mapas digitales en los que se muestran fotografías satelitales del casco urbano de la ciudad junto con los recorridos de las unidades de transporte, además de los puntos de paradas de éstas. Estas imágenes poseen una gran resolución y por lo tanto son archivos de cientos de mega bytes, razón por la cual pasar estos mapas hacia la persona que solicita la información toma una gran cantidad de tiempo, sin mencionar los problemas que enfrentará para poder visualizar el mapa en su ordenador.

En base a los argumentos antes planteados queda demostrado que existe la necesidad de ofrecer la información de los recorridos de las rutas del transporte urbano colectivo de una forma más eficaz, veloz, pública y sobretodo interactiva.

Esto nos lleva a plantearnos las siguientes interrogantes:

¿Cómo crear un sistema sencillo, intuitivo e interactivo que dé apoyo al servicio de transporte urbano colectivo de la ciudad de León mediante la API de GMaps v3?

¿Cómo desplegar el recorrido del transporte urbano colectivo de la ciudad de León en los mapas ofrecidos por la API de Gmaps v3?

¿Cuál es la manera adecuada y eficiente para guardar la información que se va a utilizar?

¿Qué herramientas son necesarias para facilitar la inserción de los datos que utilizará el sistema para poder mostrar la información en los mapas?

¿Qué mecanismos de seguridad utilizará el sistema para mantener la integridad de los datos almacenados en el mismo?

1.2 ANTECEDENTES

Existen en la actualidad muchas soluciones a nivel mundial que podemos clasificar como antecedentes de este proyecto, todas y cada una de ellas desarrolladas para trabajar sobre una zona geográfica específica, cabe destacar que todas estas soluciones implementan la API de Google Maps como eje central de su funcionamiento.

Estas soluciones presentan diferencias entre sí; diferencias que pueden ir desde el funcionamiento, cantidad de información hasta el mecanismo de introducción de datos, pero todas persiguen un mismo objetivo desplegar información acerca del sistema de transporte de una determinada ciudad, las más importantes encontradas hasta la fecha de finalización del presente trabajo se muestran en la siguiente tabla:

	Descripción	Ventajas	Desventajas
<p>Rutas Jalisco</p> <p>http://www.rutasjalisco.gob.mx/</p>	<p>Portal web que permite visualizar y acceder a la información de las rutas de transporte público del área metropolitana de Guadalajara Jalisco, México.</p>	<p>Intuitivo y fácil de usar, integra todas las rutas oficiales de la ciudad. Presenta información dividida en categorías (Deportes, iglesias, etc.)</p>	<p>Las rutas se encuentran definidas de forma estática; evitando que nuevas rutas se vean reflejadas sobre el portal web.</p>
<p>Mapa de Rutas de Transporte Público Zona Metropolitana de Monterrey.</p> <p>http://www.nl.gob.mx/?P=aet_mapa_rutas_interactivo</p>	<p>Portal web que permite visualizar y acceder a la información de las rutas de transporte público del área metropolitana de la ciudad de Monterrey, México.</p>	<p>Forma fácil de seleccionar las rutas, información detallada acerca de cada ruta (Distancia, frecuencia, horarios.)</p>	<p>Las rutas se encuentran definidas de forma estática, interfaz simple poco agradable con pocas opciones de interacción en el menú.</p>

<p>Ruta Directa</p> <p>http://www.rutadirecta.com/</p>	<p>Sitio web gratuito que ofrece a los usuarios las rutas o trayectos del transporte urbano de Monterrey y su área metropolitana; incluidos tanto el sistema de autobuses como la red del metro.</p>	<p>Cuenta con herramientas útiles y de fácil uso, además de una serie de opciones que permiten trazar un recorrido entre dos puntos específicos.</p>	<p>No cuenta con una forma de visualizar las diferentes paradas para cada recorrido, la información que se muestra se generaliza para todo el recorrido, no para un punto específico.</p>
--	--	--	---

Tabla 1. Comparativa de sistemas antecedentes encontrados.

Los sistemas antes descritos fueron tomados como base para el presente trabajo retomando ideas/soluciones interesantes y acoplables al sistema planteado en este documento, así mismo se consideraron las debilidades encontradas con el objetivo de tratar de solucionarlas y mejorar las características de la aplicación.

Cabe destacar que a nivel local no se cuenta con ningún tipo de trabajo que se acople o sirva de referencia para el sistema planteado en esta tesis.

1.3 JUSTIFICACIÓN

El transporte público o transporte colectivo es el medio más utilizado por los Leoneses para desplazarse dentro la ciudad; a través de este la mayoría de los ciudadanos, que no tienen un medio de transporte propio, llevan a cabo su necesidad de movilización desde y hacia un determinado punto.

El mínimo de suscriptores que tienen acceso a Internet en Nicaragua es alrededor de 600,000 usuarios¹. Adicionalmente, habría otros 250,000 usuarios que regularmente acceden a Internet en puntos públicos como los telecentros gratuitos, los cibercafés, los laboratorios escolares y demás².

Con este trabajo se pretende explotar de la mejor manera posible este constante crecimiento de conexiones a Internet y a la vez eliminar el problema que acarrea el no tener una forma adecuada de mostrar los recorridos de las distintas rutas de transporte público de la ciudad. Con este trabajo se obtendrá un sistema de apoyo completamente funcional que permitirá a todas las personas acceder a dichos recorridos de una manera fácil e interactiva. Este sistema cumplirá con las siguientes características:

- Estable y robusto: para lo cual se planifico una serie de pruebas que garanticen el correcto funcionamiento y su recuperación ante errores.
- Seguro: el sistema estará compuesto por un módulo de administración que resulta indispensable que disponga de políticas de seguridad que garanticen la integridad de los datos.
- Sencilla: esta es una de las características en la que se hará mayor énfasis, debido a que es un elemento importante para el usuario final.
- Interactiva: importante para que el usuario final se sienta atraído y cómodo de utilizar el sistema.

Este sistema tendrá efecto sobre todos los ciudadanos locales o extranjeros, cuyo propósito sea el de hacer uso del transporte público de la ciudad de León, así mismo de las personas que se dispongan a administrar la aplicación (usuarios debidamente acreditados por la Alcaldía de León). Para los ciudadanos representará una forma rápida y eficaz de saber los distintos recorridos del transporte, así como sus respectivas paradas; a la Alcaldía se le proporcionará con este trabajo una manera de controlar y administrar todos los recorridos del transporte de la ciudad.

¹<http://www.exitoeexportador.com/central.htm>

²<http://www.cinco.org.ni//archive/398.pdf>

1.4 OBJETIVOS

1.4.1 OBJETIVO GENERAL

Desarrollar un sistema de apoyo al servicio de transporte urbano colectivo de la ciudad de León mediante la API de Google Maps v3.

1.4.2 OBJETIVOS ESPECÍFICOS

- Determinar y analizar las tecnologías web que permitan el desarrollo del sistema informático integrando la API de Google Maps v3.
- Estudiar y comprender la funcionalidad ofrecida por la API de Google Maps v3 para desplegar la información de los recorridos del servicio de transporte urbano colectivo mediante el uso de los mapas interactivos.
- Crear una cuenta de usuario administrador que posea todos los privilegios y mecanismos necesarios para la correcta gestión del sistema.
- Definir políticas de seguridad que garanticen que el sistema sea lo suficientemente robusto para evitar el acceso al módulo de administración por parte de usuarios sin privilegios.
- Implementar tecnologías que permitan trasladar e intercambiar información desde la base de datos hacia el sistema final y viceversa, necesario para la creación de los mapas interactivos.

2. MARCO CONCEPTUAL

2.1 Google Maps

Google Maps es el nombre de un servicio de Google. Es un conjunto de servidores que ofrecen un servicio de aplicaciones de mapas a través de la Web. Ofrece imágenes de mapas desplazables, así como fotos satelitales del mundo e incluso la ruta entre diferentes ubicaciones o imágenes a pie de calle (Street View). Desde el 6 de octubre del 2005, Google Maps es parte de Google Local.

Google Maps dispone de una amplia matriz de API que le permite a desarrolladores de sistemas insertar las funciones más completas y la utilidad diaria de Google Maps en su propio sitio web y en sus propias aplicaciones, así como superponer sus propios datos sobre ellas.

Desde su lanzamiento en febrero de 2005, la aplicación cartográfica de Google ha conmocionado a la comunidad de desarrolladores. Si bien sus principios técnicos de base eran ya conocidos, incluso utilizados desde hacía tiempo, la aplicación de Google los combina de manera inteligente, y sobre todo ofrece una accesibilidad sin igual.

Varios aspectos de Google Maps son responsables de la facilidad de uso por parte de cualquier usuario: el sistema de deslizamiento de imagen, acoplado a la carga dinámica de nuevas imágenes; la adaptación del mapa al tamaño de ventana del navegador; la interfaz minimalista; la posibilidad de cambiar de tipo de mapa en un clic, entre otras.

Como todas las demás aplicaciones Google, Maps descansa poderosamente sobre la utilización de JavaScript. La carga y el deslizamiento de imagen no podrían efectuarse sin la utilización de este lenguaje. Según el nivel de zoom, un mapa podría descomponerse en varias decenas de miles de cuadrados. Esta descomposición se realiza automáticamente por un script del lado servidor. Cuando el utilizador hace deslizar la imagen en un sentido, JavaScript calcula cuáles son los cuadrados involucrados, envía al servidor una pregunta sobre las imágenes en cuestión, y las coloca alineadas junto a las otras. Cada cuadrado es almacenado en un fichero cuyo nombre indica su longitud, su latitud, y el valor del zoom. Recuperar estas informaciones para todos los cuadrados a colocar, no es sino una cuestión de derivación de los datos conocidos para un solo cuadrado. Todo ello necesita, por tanto, una sincronización perfecta entre lo que puede preguntar el código JavaScript, y lo que está efectivamente disponible en el servidor. El programa de recorte inicial de la imagen completa es por tanto extremadamente importante, porque es el responsable, para la precisión del corte y el rigor de su etiquetamiento, del buen desenvolvimiento de los acontecimientos desencadenados por las acciones del usuario.

2.2 Google Maps JavaScript API

Permite insertar Google Maps en páginas web. La versión 3 de esta API está especialmente diseñada para proporcionar una mayor velocidad y que se pueda aplicar más fácilmente tanto a móviles como a aplicaciones de navegador de escritorio tradicionales. El API proporciona diversas utilidades para manipular mapas y para añadir contenido al mapa mediante diversos servicios, permitiendo crear sólidas aplicaciones de mapas en sitios web.

Google Maps provee a los desarrolladores un API capaz de aprovechar los datos disponibles a través del servicio, en el seno de las propias aplicaciones. Y más interesante para el funcionamiento mismo del servicio, Google da acceso a AjaXSLT, un proyecto Open Source que propone una biblioteca JavaScript.

Con las API de Google Maps publicadas, numerosos sitios han aparecido proponiendo sus propias maneras de extraer los datos de los mapas de Google y presentarlos en una página a través de scripts específicos, proponiendo un servicio particular.

A continuación se detallan algunos aspectos básicos de los objetos y funcionalidades de Google Maps que fueron empleadas para el desarrollo del presente trabajo.

2.2.1 Tipos de mapas

El API utiliza un objeto `MapType` para contener información sobre los diferentes tipos de mapas. Un objeto `MapType` es una interfaz que define la visualización y el uso de mosaicos de mapas, así como la traducción de sistemas de coordenadas de pantalla a coordenadas reales (en el mapa). Cada objeto `MapType` debe contener algunos métodos para gestionar la recuperación y el suministro de mosaicos y propiedades que definan su comportamiento visual.

El funcionamiento interno de los tipos de mapas en el API de Google Maps constituye un tema de nivel avanzado. La mayoría de los desarrolladores puede utilizar simplemente los tipos de mapas básicos que se describen a continuación. Sin embargo, también es posible definir mosaicos de mapas propios mediante tipos de mapas personalizados o modificar la presentación de tipos de mapas existentes a través de mapas con estilos. El sistema solamente utiliza los tres primeros tipos de mapas básicos que van descritos a continuación.

Tipos de mapas básicos:

- `MapTypeId.ROADMAP`: muestra la vista de mapa de carreteras predeterminada.
- `MapTypeId.SATELLITE`: muestra imágenes de satélite de Google Earth.
- `MapTypeId.HYBRID`: muestra una mezcla de vistas normales y de satélite.
- `MapTypeId.TERRAIN`: muestra mosaicos de mapas físicos a partir de la información de relieve.

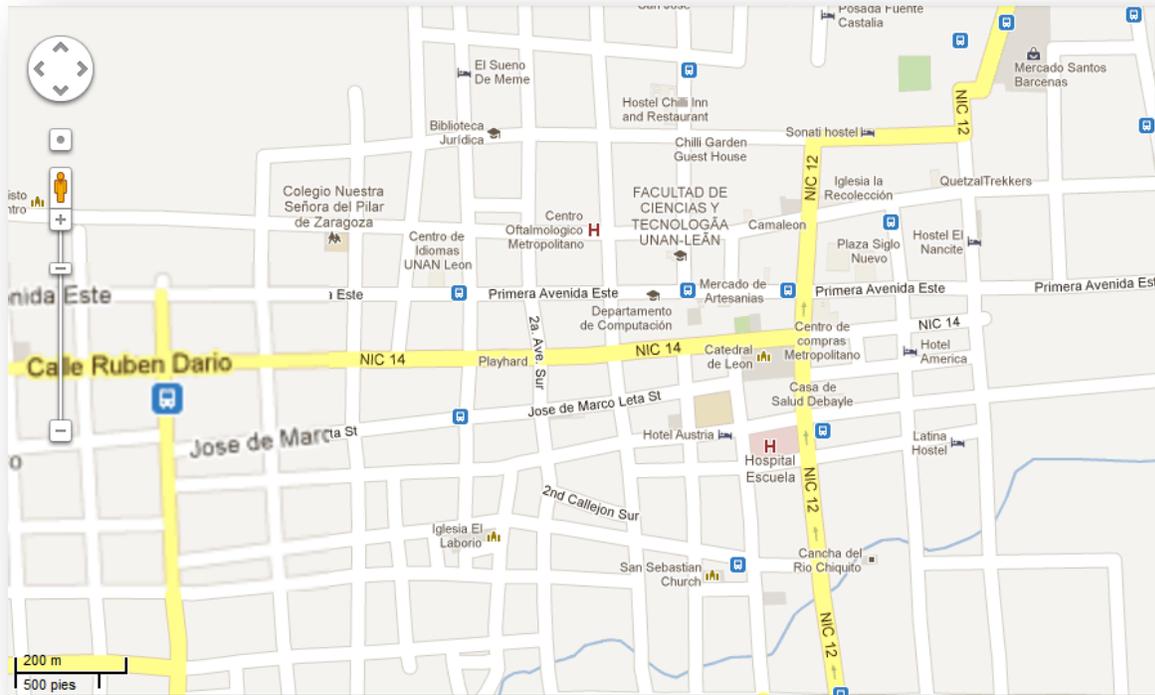


Ilustración 1: Tipo de mapa ROADMAP(mapa).



Ilustración 2: Tipo de mapa SATELLITE(satélite).

2.2.2 Aspectos generales de las superposiciones

Las superposiciones son objetos del mapa que están vinculados a coordenadas de latitud y longitud, por lo que se mueven al arrastrar o aplicar el zoom sobre el mapa. Las superposiciones son objetos que se añaden al mapa para designar puntos, líneas, áreas o grupos de objetos.

Google Maps API incorpora varios tipos de superposiciones, aquí las principales que utiliza el sistema:

2.2.2.1 Marcadores

Los marcadores identifican ubicaciones en el mapa (paradas). En algunas ocasiones, los marcadores pueden mostrar imágenes de iconos personalizados, que se denominan normalmente iconos. Los marcadores e iconos son objetos de tipo `Marker`.

Los siguientes campos son especialmente importantes y se suelen establecer al crear un marcador:

- `position` (obligatorio) especifica un valor de `LatLng`(coordenadas en latitud y longitud) que identifica la ubicación inicial del marcador.
- `map` (opcional) especifica el objeto `Map` en el que se sitúa el marcador.

Los marcadores se han diseñado para que sean interactivos. Por ejemplo, reciben eventos 'click' de forma predeterminada y se utilizan a menudo con detectores de eventos para proporcionar ventanas de información.



Ilustración 3: Icono que representa una parada en el sistema.

2.2.2.2 Polilíneas

La clase `Polyline` define una superposición de segmentos lineales conectados sobre el mapa. Un objeto `Polyline` incluye un conjunto de ubicaciones `LatLng` y crea una serie de segmentos lineales que conectan dichas ubicaciones en una secuencia ordenada.

Los objetos `Polyline` se representan en el mapa como una serie de líneas rectas. Es posible especificar colores, grosores y niveles de opacidad personalizados que se aplicarán al trazo de la línea en un objeto.

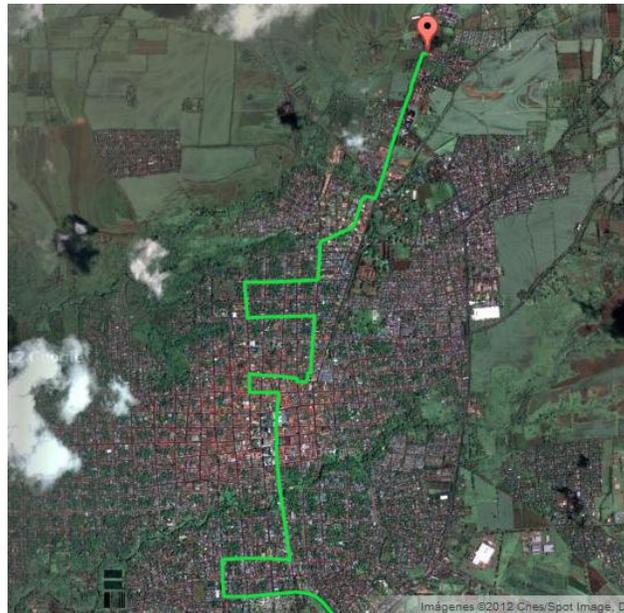


Ilustración 4: Inicio del recorrido de la ruta 103 en forma de una polilínea.

2.2.2.3 Ventanas de información

Las ventanas de información (`InfoWindow`) muestran contenido en una ventana flotante situada encima del mapa. La ventana de información tiene un aspecto ligeramente parecido al de los bocadillos de los cómics. Tiene un área de contenido y un pico afilado, cuyo extremo se encuentra en una ubicación especificada en el mapa.

El contenido de un `InfoWindow` puede incluir una cadena de texto, un fragmento HTML o un elemento DOM. Las ventanas de información (`InfoWindow`) pueden adjuntarse a objetos `Marker` (en cuyo caso su posición se basa en la ubicación del marcador) o al propio mapa en un objeto `LatLng` especificado o una polilínea.



Ilustración 5: Infowindow que muestra una imagen y descripción de una parada.

2.3 JavaScript

JavaScript es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico.

Se utiliza principalmente en su forma del lado del cliente (client-side), implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas, en bases de datos locales al navegador aunque existe una forma de JavaScript del lado del servidor (Server-side JavaScript o SSJS). Su uso en aplicaciones externas a la web, por ejemplo en documentos PDF, aplicaciones de escritorio (mayoritariamente widgets) es también significativo.

JavaScript se diseñó con una sintaxis similar al lenguaje C, aunque adopta nombres y convenciones del lenguaje de programación Java. Sin embargo Java y JavaScript no están relacionados y tienen semánticas y propósitos diferentes.

Todos los navegadores modernos interpretan el código JavaScript integrado en las páginas web. Para interactuar con una página web se provee al lenguaje JavaScript de una implementación del Document Object Model (DOM).

Tradicionalmente se venía utilizando en páginas web HTML para realizar operaciones y únicamente en el marco de la aplicación cliente, sin acceso a funciones del servidor. JavaScript se interpreta en el agente de usuario, al mismo tiempo que las sentencias van descargándose junto con el código HTML.

2.4 PHP

PHP es un lenguaje interpretado del lado del servidor que se caracteriza por su potencia, versatilidad, robustez y modularidad. Los programas escritos en PHP son embebidos directamente en el código HTML y ejecutados por el servidor web a través de un intérprete antes de transferir al cliente que lo ha solicitado un resultado en forma de código HTML puro.

Por su flexibilidad, PHP resulta un lenguaje muy sencillo de aprender; especialmente para programadores familiarizados con lenguajes como C o Java, debido a las similitudes de sintaxis entre ellos.

Es un lenguaje multiplataforma; los programas funcionan igual sobre diferentes plataformas, trabajando sobre la mayoría de servidores web y estando preparado para interactuar con más de 20 tipos de sistemas gestores de bases de datos, en comparación con otros tipos de tecnologías similares, PHP resulta más rápido, independiente de la plataforma y más sencillo de aprender y utilizar.

2.4.1 Características

A continuación se detallan las características más relevantes y que fueron tomadas en cuenta para la incorporación del lenguaje PHP como parte del desarrollo del sistema:

- Orientado al desarrollo de aplicaciones web dinámicas con acceso a información almacenada en una base de datos.
- El código fuente escrito en PHP es invisible al navegador web y al cliente ya que es el servidor el que se encarga de ejecutarlo y enviar el resultado en formato HTML al navegador. Esto hace que la programación en PHP sea segura y confiable.
- Capacidad de conexión con la mayoría de los motores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL y PostgreSQL.
- Capacidad de expandir su potencial utilizando módulos (llamados ext's o extensiones).

Posee una amplia documentación en su sitio web oficial (www.php.net/), entre la cual se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda.

2.5 MySQL

MySQL es un sistema de gestión de bases de datos relacional, multihilo y multiusuario.

MySQL es una base de datos muy rápida en la lectura cuando utiliza el motor no transaccional MyISAM, pero puede provocar problemas de integridad en entornos de alta concurrencia en la modificación. En aplicaciones web hay baja concurrencia en la modificación de datos y en cambio el entorno es intensivo en lectura de datos, lo que hace a MySQL ideal para este tipo de

aplicaciones. Sea cual sea el entorno en el que se va a utilizar MySQL, es importante monitorizar de antemano el rendimiento para detectar y corregir errores tanto de SQL como de programación.

2.5.1 Características

Inicialmente, MySQL carecía de elementos considerados esenciales en las bases de datos relacionales, tales como integridad referencial y transacciones. A pesar de ello, atrajo a los desarrolladores de páginas web con contenido dinámico, justamente por su simplicidad.

Poco a poco los elementos de los que carecía MySQL están siendo incorporados tanto por desarrolladores asociados al proyecto, como por desarrolladores de software libre. Entre las características disponibles en las últimas versiones se pueden destacar:

- Amplio subconjunto del lenguaje SQL. Algunas extensiones son incluidas igualmente.
- Disponibilidad en gran cantidad de plataformas y sistemas.
- Posibilidad de selección de mecanismos de almacenamiento que ofrecen diferente velocidad de operación, soporte físico, capacidad, distribución geográfica, transacciones, etc.
- Transacciones y claves foráneas.
- Conectividad segura.
- Replicación.
- Búsqueda e indexación de campos de texto.

Dado que los computadores son muy buenos manejando grandes cantidades de información, los administradores de bases de datos juegan un papel central en computación, como aplicaciones independientes o como parte de otras aplicaciones.

2.6 JQuery

JQuery es una biblioteca de JavaScript, creada inicialmente por John Resig, que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas web.

Las aplicaciones en Internet son cada vez más complejas, ya que incorporan efectos visuales, funcionalidad drag and drop, auto-completar, animaciones etc. el desarrollar todos estos conceptos desde cero puede resultar complicado sobretodo si tenemos que presentar la solución en muy poco tiempo, en este tipo de situaciones el empleo de librerías como JQuery facilitan el desarrollo de una solución web. Otra ventaja paralela es despreocuparse cuando se codifica en la compatibilidad de navegadores, ya que la librería resolverá esto.

JQuery consiste en un único fichero JavaScript que contiene las funcionalidades comunes de DOM, eventos, efectos y AJAX.

La característica principal de la biblioteca es que permite cambiar el contenido de una página web sin necesidad de recargarla, mediante la manipulación del árbol DOM y peticiones AJAX. Para ello utilizan las funciones `$()` o `jQuery()`.

2.6.1 Características

A continuación se detallan las características más relevantes y que fueron tomadas en cuenta para la incorporación de JQuery como parte del desarrollo del sistema:

- Selección de elementos DOM.
- Interactividad y modificaciones del árbol DOM, incluyendo soporte para CSS 1-3 y un plugin básico de XPath.
- Eventos.
- Manipulación de la hoja de estilos CSS.
- Efectos y animaciones.
- Animaciones personalizadas.
- Ajax.

2.7 HTML

HTML, siglas de Hyper Text Markup Language (lenguaje de marcado de hipertexto), es el lenguaje de marcado predominante para la elaboración de páginas web. Es usado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes. El HTML se escribe en forma de «etiquetas», rodeadas por corchetes angulares (`<`, `>`). HTML también puede describir, hasta un cierto punto, la apariencia de un documento, y puede incluir un script (por ejemplo JavaScript), el cual puede afectar el comportamiento de navegadores web y otros procesadores de HTML.

HTML también es usado para referirse al contenido del tipo de MIME text/html o todavía más ampliamente como un término genérico para el HTML, ya sea en forma descendida del XML (como XHTML 1.0 y posteriores) o en forma descendida directamente de SGML (como HTML 4.01 y anteriores).

2.7.1 Elementos

Los elementos son la estructura básica de HTML. Los elementos tienen dos propiedades básicas: atributos y contenido. Cada atributo y contenido tiene ciertas restricciones para que se considere válido al documento HTML. Un elemento generalmente tiene una etiqueta de inicio (p.ej. `<nombre-de-elemento>`) y una etiqueta de cierre (p.ej. `</nombre-de-elemento>`). Los

atributos del elemento están contenidos en la etiqueta de inicio y el contenido está ubicado entre las dos etiquetas (p.ej. `<nombre-de-elemento atributo="valor">Contenido</nombre-de-elemento>`). Algunos elementos, tales como `
`, no tienen contenido ni llevan una etiqueta de cierre.

2.8 JSON

Formalizado y popularizado por Douglas Crockford, JSON es un formato ligero de intercambio de datos escrito usando objetos JavaScript y una sintaxis de array literal. Aquí hay un ejemplo de una lista de usuario escrito en JSON:

```
[
  {
    "id": 1, "username": "alice", "realname": "Alice Smith",
    "email": "alice@alicesmith.com"},
  {
    "id": 2, "username": "bob", "realname": "Bob Jones",
    "email": "bob@bobjones.com"},
  {
    "id": 3, "username": "carol", "realname": "Carol Williams",
    "email": "carol@carolwilliams.com"},
  {
    "id": 4, "username": "dave", "realname": "Dave Johnson",
    "email": "dave@davejohnson.com"}
]
```

Los usuarios son representados como objetos y la lista de usuarios está dentro de un array.

2.9 XML

XML, siglas en inglés de eXtensible Markup Language (lenguaje de marcas extensible), es un lenguaje de marcas desarrollado por el World Wide Web Consortium (W3C). Deriva del lenguaje SGML (Standard Generalized Markup Language) y permite definir la gramática de lenguajes específicos para estructurar documentos grandes.

XML es una tecnología que se propone como un estándar para el intercambio de información estructurada entre diferentes plataformas.

Tiene un papel muy importante en la actualidad ya que permite la compatibilidad entre sistemas para compartir la información de una manera segura, fiable y fácil.

2.9.1 Ventajas del XML

Es extensible: Después de diseñado y puesto en producción, es posible extender XML con la adición de nuevas etiquetas, de modo que se pueda continuar utilizando sin complicación alguna.

El analizador es un componente estándar, no es necesario crear un analizador específico para cada versión de lenguaje XML. Esto posibilita el empleo de cualquiera de los analizadores disponibles. De esta manera se evitan bugs y se acelera el desarrollo de aplicaciones.

Si un tercero decide usar un documento creado en XML, es sencillo entender su estructura y procesarla. Mejora la compatibilidad entre aplicaciones. Podemos comunicar aplicaciones de distintas plataformas, sin que importe el origen de los datos, es decir, podríamos tener una aplicación en Linux con una base de datos PostgreSQL y comunicarla con otra aplicación en Windows y una Base de Datos MS-SQL Server.

Transformamos datos en información, pues se le añade un significado concreto y los asociamos a un contexto, con lo cual tenemos flexibilidad para estructurar documentos.

2.9.2 Estructura de un documento XML

La tecnología XML busca dar solución al problema de expresar información estructurada de la manera más abstracta y reutilizable posible. Que la información sea estructurada quiere decir que se compone de partes bien definidas, y que esas partes se componen a su vez de otras partes. Entonces se tiene un árbol de trozos de información.

Una etiqueta consiste en una marca hecha en el documento, que señala una porción de éste como un elemento. Un pedazo de información con un sentido claro y definido. Las etiquetas tienen la forma <nombre>, donde nombre es el nombre del elemento que se está señalando.

A continuación se muestra un ejemplo para entender la estructura de un documento XML:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE Edit_Mensaje SYSTEM "Edit_Mensaje.dtd">
<Edit_Mensaje>
  <Mensaje>
    <Remitente>
      <Nombre>Nombre del remitente</Nombre>
      <Mail> Correo del remitente </Mail>
    </Remitente>
    <Destinatario>
      <Nombre>Nombre del destinatario</Nombre>
      <Mail>Correo del destinatario</Mail>
    </Destinatario>
    <Texto>
      <Asunto>
        Este es mi documento con una estructura muy sencilla
        no contiene atributos ni entidades...
      </Asunto>
      <Parrafo>
        Este es mi documento con una estructura muy sencilla
        no contiene atributos ni entidades...
      </Parrafo>
    </Texto>
  </Mensaje>
</Edit_Mensaje>
```

2.10 CSS

CSS es un lenguaje de hojas de estilos creado para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML y XHTML. CSS es la mejor forma de separar los contenidos y su presentación y es imprescindible para crear páginas web complejas.

Separar la definición de los contenidos y la definición de su aspecto presenta numerosas ventajas, ya que obliga a crear documentos HTML/XHTML bien definidos y con significado completo (también llamados "*documentos semánticos*"). Además, mejora la accesibilidad del documento, reduce la complejidad de su mantenimiento y permite visualizar el mismo documento en infinidad de dispositivos diferentes.

Al crear una página web, se utiliza en primer lugar el lenguaje HTML/XHTML para *marcar* los contenidos, es decir, para designar la función de cada elemento dentro de la página: párrafo, titular, texto destacado, tabla, lista de elementos, etc.

Una vez creados los contenidos, se utiliza el lenguaje CSS para definir el aspecto de cada elemento: color, tamaño y tipo de letra del texto, separación horizontal y vertical entre elementos, posición de cada elemento dentro de la página, etc.

2.11 Patrón de diseño Model View Controller (MVC)

Uno de los patrones de diseño web más conocido es la arquitectura **MVC** (Model-View-Controller), el cual es utilizado en la Ingeniería de Software y consta de tres niveles:

- El *Modelo* (**Model**) es la lógica del negocio, la cual representa la información con la que la aplicación opera. Para persistir esta información, muchas aplicaciones usan un mecanismo de almacenaje tal como una base de datos. Pero el patrón MVC no hace mención específicamente a la capa de acceso a la información porque está sobreentendido que está encapsulada por el modelo.
- La *Vista* (**View**) renderiza el modelo dentro de una página web apropiada para que el usuario pueda interactuar con ella. Cabe destacar que con esta arquitectura múltiples vistas pueden existir para un simple modelo.
- El *Controlador* (**Controller**) responde a las acciones del usuario e invoca cambios en el modelo ó genera la vista apropiada, dependiendo de las peticiones del usuario.

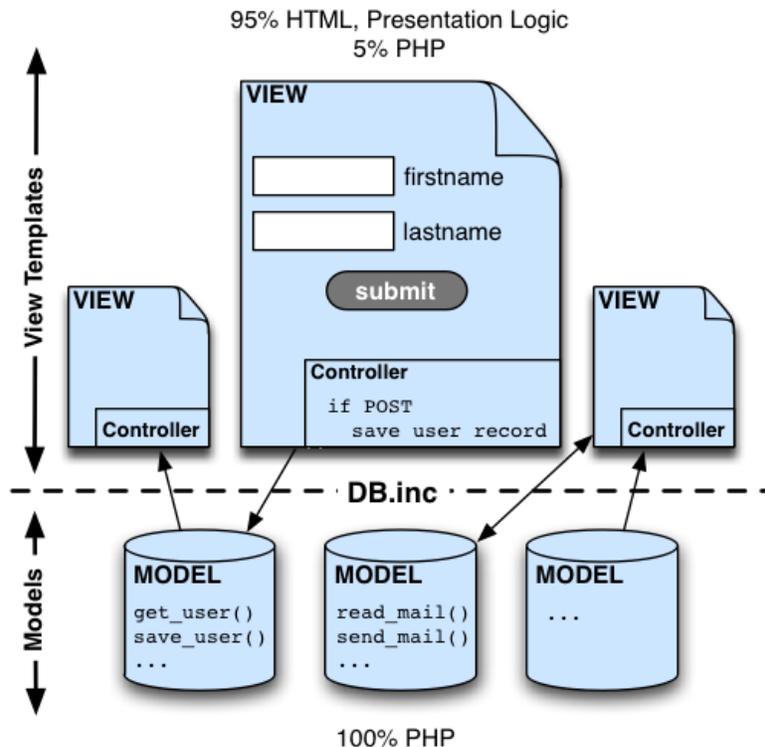


Ilustración 6: Patrón de diseño MVC.

Como se puede observar en la Ilustración 6 la **Vista** contiene un formulario para almacenar información en la base de datos, el usuario interactúa con la aplicación mediante esta capa. **Controlador** la capa que sirve como intermediario entre la vista y el modelo, cuando el usuario realiza una acción en la vista esta envía una información al controlador, este se encarga de enviarle esa información al modelo y de recibir la respuesta del modelo, a continuación esta respuesta es pasada a la vista para mostrársela al usuario y el **Modelo** agrega significado y el acceso a los datos detrás de la aplicación, es el encargado de realizar la conexión a la base de datos para insertar la información indicada por la **Vista**.

La arquitectura **MVC** separa la lógica de negocio (modelo) y la presentación (view), resultando en un código muy mantenible. Por ejemplo, si una aplicación debe ejecutarse tanto en un navegador web como en un dispositivo móvil, solo necesita de una nueva vista (view); pero conservando el controlador y modelo originales. El controlador ayuda a ocultar los detalles del protocolo usado en la petición del modelo y la vista, mientras que el modelo abstrae la lógica de la información, lo cual hace a la vista y la acción independientes de, por ejemplo, el tipo de base de datos usado por la aplicación.

3. Análisis y Diseño del sistema

En este capítulo se dará a conocer la descripción general del sistema y su funcionamiento en general, las partes que lo conforman y su funcionamiento dentro del mismo. También se dará a conocer la estructura del sistema y los elementos que conforman parte del mismo, como también el análisis del diseño del sistema.

3.1 Descripción general del sistema

El sistema al cual denominamos Satuc provee una forma a través de la cual todas las personas pueden visualizar los recorridos y paradas de las distintas rutas del sistema de transporte urbano de la ciudad de León, este es accesible únicamente desde la web, siendo estrictamente necesario disponer de una conexión a Internet. El sistema utiliza la API de Google Maps v3 para poder mostrar los recorridos y paradas asociadas a estos recorridos, de igual forma dispone de una serie de opciones desarrolladas para que el usuario final pueda interactuar de forma agradable, estas opciones van desde la manipulación de los recorridos, ocultar o visualizar paradas, selección de idioma en el que se presenta el sistema y mostrar información detallada de las paradas y recorridos.

El sistema se encuentra dividido en dos módulos:

- El módulo de administración: módulo dirigido a la administración y gestión de la información asociada a cada ruta. El administrador puede manipular la información del sistema para realizar cambios en una o todas las rutas pre-cargadas; así como también agregar nueva información o nuevas rutas. Para emplear la funcionalidad de este módulo es necesario autenticarse dentro del sistema y acceder a la sección de administración.
- El módulo de interfaz pública: módulo dirigido a la visualización de la información, asociada a las rutas, que es cargada por el administrador del sistema desde el módulo antes descrito. El usuario puede interactuar con diversas opciones que le permiten mostrar u ocultar información asociada a las rutas; así como también una visualización general de todas las rutas o una visualización de una ruta específica. Este módulo es el que se muestra por defecto al cargar el sistema desde el navegador web.

3.2 Funcionalidades generales del sistema

En esta sección se detallan las funciones enfocadas a cada uno de los dos grandes módulos desarrollados. Se describe de forma general los aspectos más relevantes así como también las funcionalidades ofrecidas.

3.2.1 Funciones enfocadas al módulo de interfaz pública

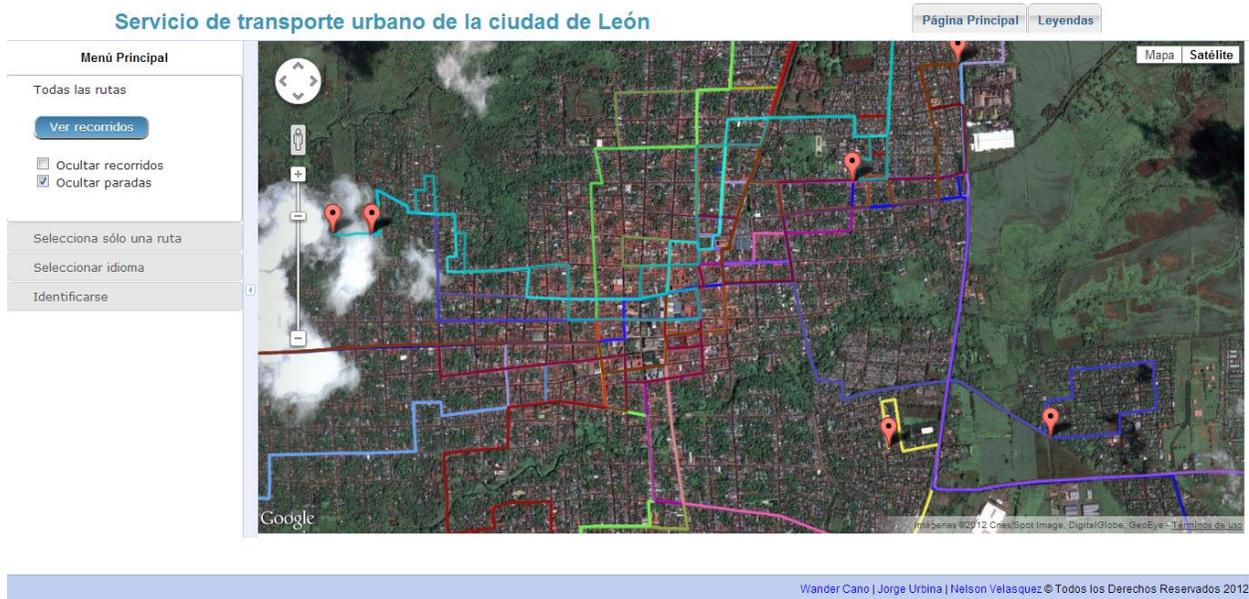


Ilustración 7: Vista del módulo de interfaz pública.

Para hacer uso del sistema se cuenta con un menú a través del cual el usuario tiene a disposición todas las opciones con las que cuenta este sistema, estas opciones son:

Todas las rutas: muestra sobre el mapa todas las rutas que se encuentran pre-cargadas en el sistema.

Ver recorrido: con este botón el usuario podrá visualizar todos los recorridos existentes en la base de datos.

- Ocultar recorridos: proporciona una forma de ocultar los recorridos.
- Ocultar paradas: oculta todas las paradas asociadas a un recorrido, tiene doble función ya que también se encarga de mostrarlas.

Seleccionar sólo una ruta: opción que carga una ruta específica de las que ya están dentro del sistema.

- Recorrido de ida: muestra el recorrido solo de ida para una ruta.
- Recorrido de vuelta: muestra el recorrido solo de vuelta para una ruta.
- Mostrar las paradas: muestra todas las paradas para una ruta.

- Terminales: muestra las terminales para una ruta. Las terminales son los puntos iniciales y finales de una ruta.

Seleccionar idioma: a través de esta opción el usuario puede seleccionar el idioma de su preferencia.

Identificarse: opción que permite al administrador autenticarse como tal dentro del sistema.

Todas estas opciones permiten al usuario interactuar con el sistema de forma fácil y agradable, una vez que de clic sobre estas opciones se muestra la información asociada a ese botón.



Ilustración 8: Menú principal.

3.2.2 Funcionalidades enfocadas al módulo administrador

Este módulo permite realizar una administración de los datos necesarios para cargar la información a ser visualizada en el mapa, las opciones de administración van desde agregar, editar y eliminar, esto se realiza utilizando funciones implementadas en PHP las cuales lanzan una consulta a la base de datos logrando una actualización de los datos en tiempo real, esta administración es manipulable desde una interfaz diseñada para que el administrador interactúe de manera fácil y funcional programada en PHP y haciendo uso de formularios para el envío de las diferentes consultas.



Ilustración 9: Vista del modulo administrador.

Agregar rutas: permite agregar una nueva ruta en el sistema.

Edita rutas: incluye las opciones de actualizar y eliminar una o todas las rutas.

Agregar paradas: permite agregar una nueva parada asociada a un recorrido en específico.

Editar paradas: comprende funciones como son actualizar y eliminar una paradas.

Agregar recorrido: permite agregar un nuevo recorrido, para esto es necesario establecer los puntos en el mapa, estos puntos serán luego las coordenadas para este nuevo recorrido.

Editar recorrido: permite la actualización de un recorrido y también se contempla la posibilidad de eliminarlo.

3.3 Diagrama de casos de uso

Los diagramas de caso de uso son una técnica que permite ayudar a definir cada una de las tareas que un tipo de usuario es capaz de realizar sobre el sistema, su propósito principal es comunicar la funcionalidad del sistema, así como su comportamiento ante los usuarios.

A continuación se detallan los diferentes casos de uso que se pueden realizar en el sistema.

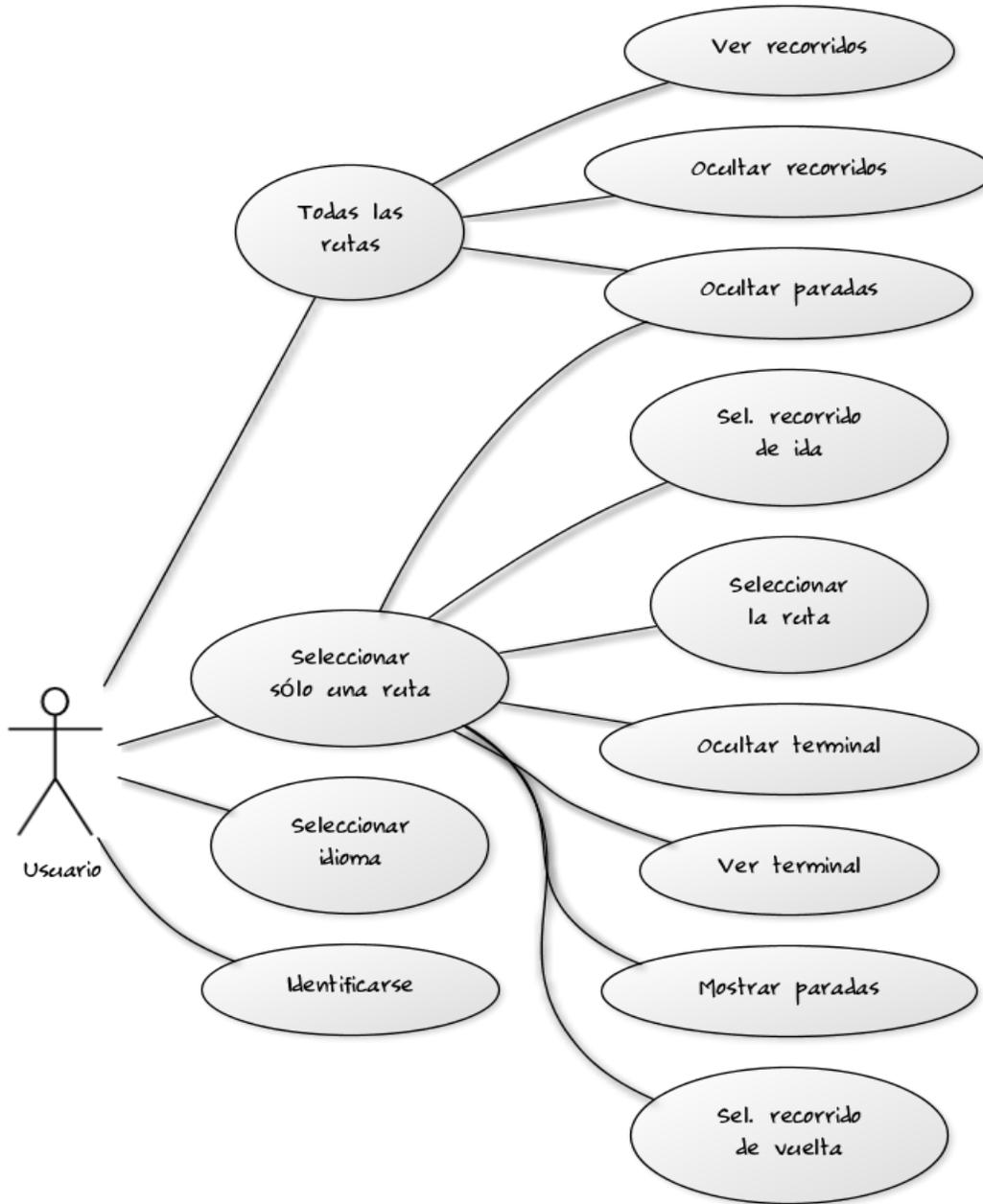


Ilustración 10: Diagrama de casos de uso del módulo interfaz pública.

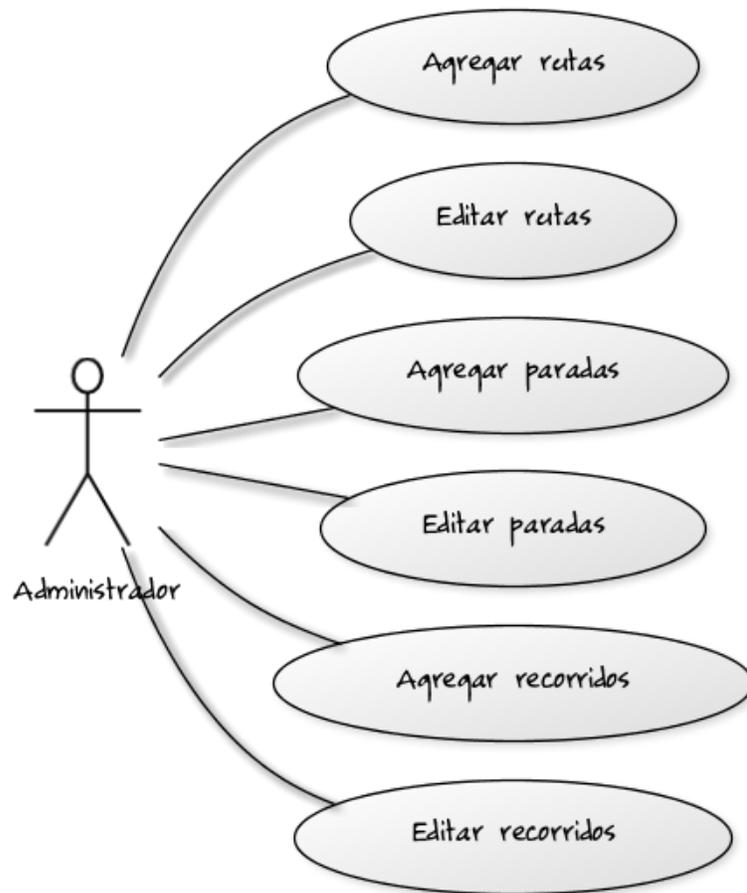


Ilustración 11: Diagrama de casos de uso módulo administrador.

3.3.1 Módulo de interfaz pública

Ver todos los recorridos:

Propósito: dar la posibilidad al usuario de ver todos los recorridos que realizan las unidades de transporte del casco urbano de la ciudad de León así como también todos los puntos de parada.

El usuario tiene la posibilidad de interactuar con el menú asociado y poder realizar una de las actividades siguientes.

- **Ver recorridos**

El sistema captura el evento clic asociado a este botón y envía una consulta a la base de datos, con el objetivo de extraer la referencia de todos los archivos xml los cuales poseen la latitud y longitud en grados de cada punto del recorrido en el mapa para luego enviarlos a la función que se encarga de pintar todas las líneas utilizando la API de Gmaps v3.

- **Ocultar recorridos**

Propósito: dar la posibilidad al usuario de ocultar todos los recorridos que realizan las unidades de transporte para que éste pueda interactuar con los puntos de paradas de éstas mismas unidades.

El usuario da clic sobre la casilla de verificación asociada a este caso de uso y el sistema oculta todos los recorridos utilizando un método de la API de Gmaps v3 esto manda a ocultar los recorridos en forma de polilíneas que se encuentran dibujados dentro del mapa.

- **Ocultar paradas**

Propósito: el sistema dispone de la característica de poder ocultar los puntos de parada. Por defecto cuando se carga el sistema por primera vez no se muestran las paradas.

El usuario da clic sobre la casilla de verificación asociada a este caso de uso y el sistema ejecuta la funcionalidad que permite ocultar las paradas representadas en forma de marcas.

Seleccionar sólo una ruta

Permite que el usuario pueda seleccionar por medio de una lista desplegable de manera individual el recorrido de solamente una unidad de transporte para poder ver con detenimiento el recorrido que realiza dentro del casco urbano de la ciudad, además de seleccionar el sentido del mismo, sus terminales y sus paradas.

- **Seleccionar ruta**

La funcionalidad se activa cuando se desencadena un evento de cambio dentro de una lista desplegable que contiene el nombre de cada unidad de transporte que tiene almacenado un recorrido en la base de datos, esto desencadenan la ejecución de dos consultas SQL utilizando como parámetro el código de ruta; la primer consulta devuelve como resultado todos los datos y referencias del archivo xml con los recorridos; la segunda regresará todas las paradas que posee dicho recorrido. Para luego mandarlas a mostrar en el mapa que posee el sistema.

- **Seleccionar recorrido de ida**

El sistema muestra solamente el(los) recorrido(s) de ida que realiza la unidad de transporte.

- **Seleccionar recorrido de vuelta**

El sistema muestra solamente el(los) recorrido(s) de vuelta que realiza la unidad de transporte.

- **Mostrar paradas**

Se muestran en el mapa del sistema los puntos de detención del recorrido.

- **Ocultar paradas**

Se ocultan en el mapa del sistema los puntos de detención del recorrido.

- **Mostrar terminales**

Se muestra, a través de una marca el punto inicial y final de un recorrido.

- **Ocultar terminales**

Se oculta el punto inicial y final de un recorrido.

Seleccionar idioma

A través de esta opción el usuario puede seleccionar el idioma de su preferencia, para ello se hace uso de un plugin proporcionado por Google Website Translator para utilizar este servicio se debe acceder a la siguiente dirección web: <https://translate.google.com/manager/add> y seguir los pasos necesarios, se copia el código proporcionado y se incrusta en el sistema.

El lenguaje por defecto es el español y los lenguajes que el usuario tiene la posibilidad de traducir son: alemán, francés, holandés, inglés, italiano y portugués.

Identificarse

Opción que permite al administrador autenticarse ante el sistema por medio de un formulario que pide el nombre de usuario y su contraseña.

3.3.2 Módulo administrador

Selecciona ruta

El administrador tiene la posibilidad de ver en el mapa el recorrido de cualquier ruta para trabajar agregar o editar paradas asociadas a éste.

Agregar ruta

Muestra un formulario donde se agrega una nueva ruta, es decir el nombre y la descripción de la misma, también es posible agregar el nombre de la cooperativa pero este campo es opcional.

Editar ruta

Muestra una tabla con todas las rutas definidas en el sistema; en este punto es posible editar los valores que posee cada una de las rutas o simplemente eliminarla de la base de datos.

Agregar paradas

Para agregar paradas primero se debe seleccionar el recorrido específico al cual van a ser agregadas las nuevas paradas, a continuación se coloca una marca en el mapa, luego se completa un formulario con el nombre de la estación, descripción de la misma y una imagen opcional de la parada.

Editar paradas

Muestra una tabla con todas las paradas asociadas a un recorrido, con la posibilidad de editar y/o eliminar las paradas del recorrido.

Agregar recorridos

Da la posibilidad al administrador de agregar un nuevo recorrido, primero va dibujando punto por punto todo el recorrido en el mapa, luego llena un formulario con los campos que posee ese recorrido como: nombre del recorrido, descripción, sentido y color asociado a dicho recorrido. Por último se selecciona el nombre de la ruta a la que pertenecerá dicho recorrido.

Editar recorridos

Muestra una tabla con todos los recorridos insertados en el sistema, con la posibilidad de editar los campos que posee o eliminarlo por completo.

3.4 Arquitectura del sistema

Aquí se describen todas las herramientas software necesarias para la implementación, así como el mecanismo de comunicación utilizado para almacenar y mostrar información que se ofrece y como están estructurados ciertos elementos dentro del sistema.

El sistema está basado en el patrón de arquitectura de software MVC para lograr una mejor organización de los datos. Para el desarrollo del sistema se hace uso del lenguaje de programación PHP, utilizando para la codificación **phpDesigner v8** el cual es un editor de código PHP de alta funcionalidad; las distintas interfaces gráficas que componen el sistema están desarrolladas en el lenguaje de marcado HTML y se hace uso de JavaScript para insertar la funcionalidad de los mapas.

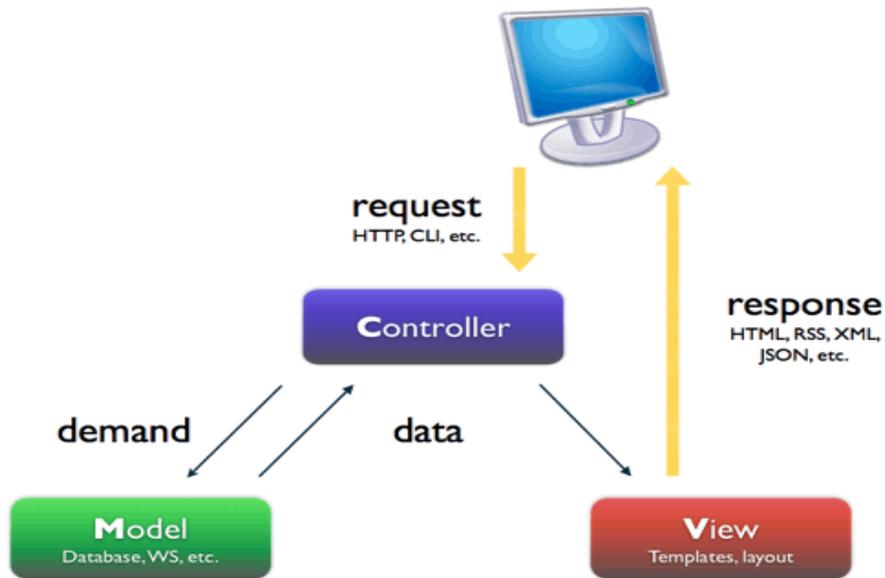


Ilustración 12: Arquitectura MVC.

Para el correcto funcionamiento del sistema se hace uso de una serie de clases las cuales llevan a cabo diferentes tareas como lo son la extracción e introducción de datos desde y hacia el servidor web. El sistema es desarrollado con una serie de herramientas cada una de ellas eficientes en su área, con el objetivo de implementar la arquitectura del sistema de la manera más adecuada. A continuación se explican cada una de ellas en base al módulo en que se utilizan.

Módulo interfaz pública

Es el módulo encargado de presentar la información y las opciones de interacción al usuario, éste es desarrollado con las siguientes herramientas:

- Toda la interfaz que se le muestra al usuario está desarrollada en HTML, esta se encarga de mostrar todas las opciones y controles para que el usuario interactúe de la mejor manera, la utilización de este está basada en que HTML es el lenguaje de marcado predominante para la elaboración de páginas webs.
- Para la carga del mapa y su respectiva API se utiliza el lenguaje de programación interpretado JavaScript; para la carga de toda la información asociada al mapa se utiliza PHP, esto es necesario para crear todas las clases que se encargaran de manipular dicha información, así mismo se hace uso del lenguaje de marcas extensible XML para el intercambio de información entre el servidor y el cliente.

Módulo administrador

Módulo encargado para la administración del sistema, este es desarrollado con las siguientes herramientas:

- Al igual que en el módulo de interfaz pública este utiliza HTML para el diseño de todos los controles necesarios para una correcta administración. Para la manipulación de los datos se hace uso de PHP y XML.

3.5 Secuencia del sistema

La secuencia del sistema involucra todas aquellas operaciones que se llevan a cabo desde los diferentes módulos del sistema siguiendo una secuencia ordenada.

La secuencia de operaciones que involucra este sistema en particular son las que se producen cuando un usuario interactúa con el sistema, así mismo cuando se realicen diferentes tareas de administración de este mismo.

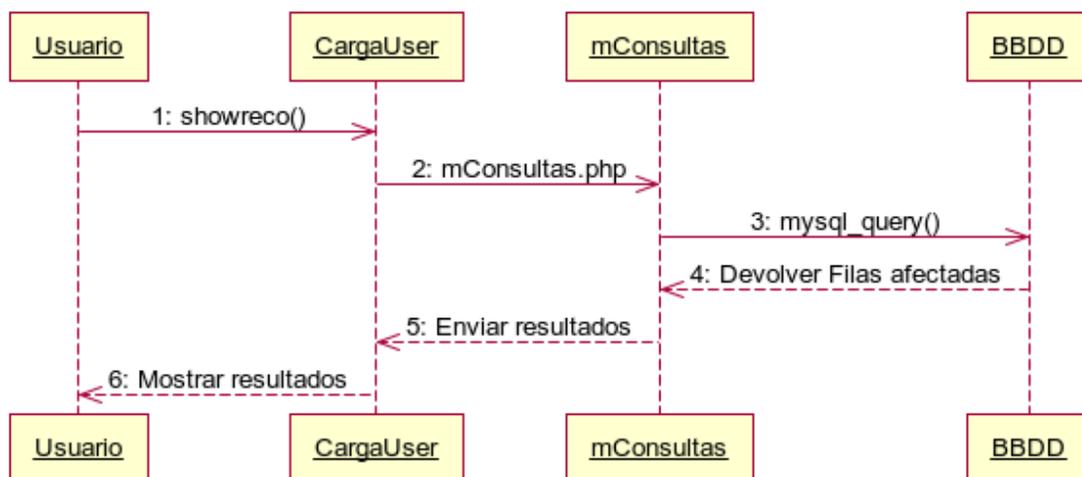
3.5.1 Diagramas de secuencia

Un diagrama de secuencia muestra la interacción de un conjunto de objetos en un sistema a través del tiempo y se modela para cada caso de uso.

Los diagramas de secuencia que se emplean en esta sección sirven para mostrar los detalles de implementación de cada uno de los escenarios, incluyéndose los objetos y las clases utilizadas en la implementación de dicho escenario, así como también se muestran los mensajes intercambiados entre los objetos.

A continuación se muestran los diagramas de secuencia más importantes para la correcta funcionalidad del sistema.

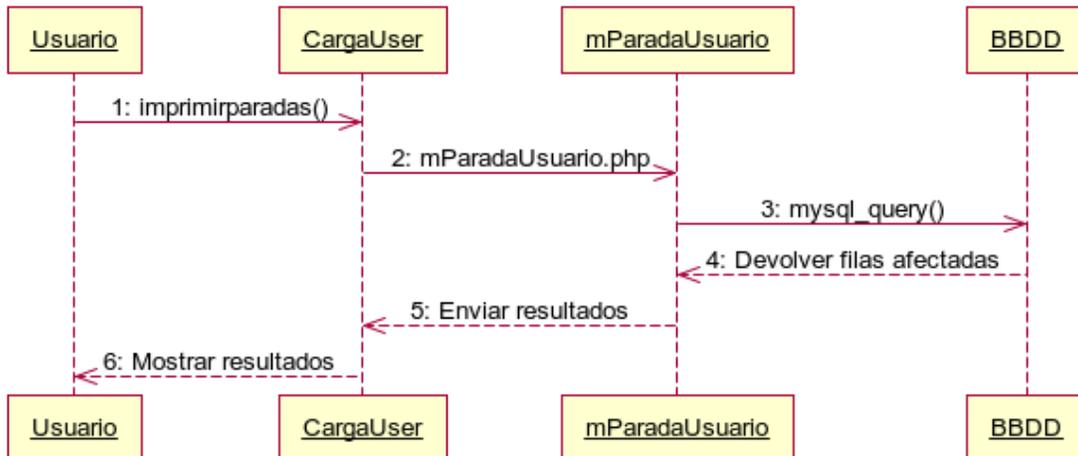
3.5.1.1 Extraer y mostrar recorridos



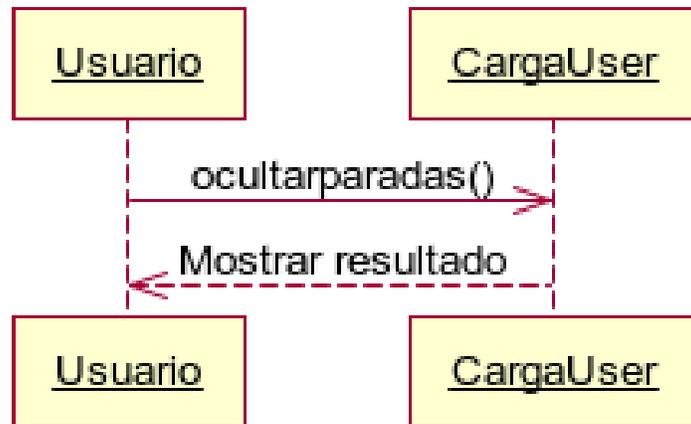
3.5.1.2 Ocultar recorridos



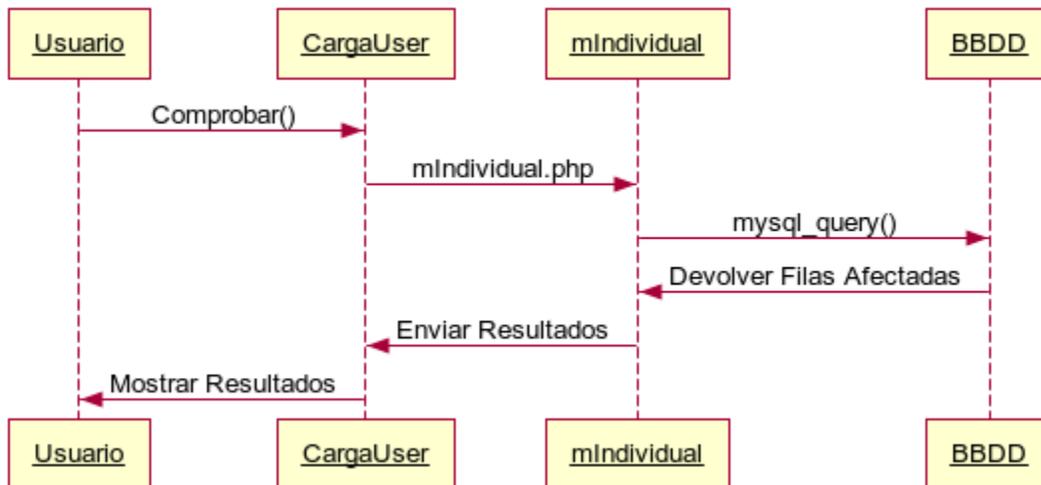
3.5.1.3 Extraer y mostrar paradas



3.5.1.4 Ocultar paradas



3.5.1.5 Seleccionar un recorrido específico



3.5.1.6 Extraer y mostrar terminales



3.5.1.7 Ocultar terminales



3.6 Base de datos del sistema

La base de datos del sistema es el actor de nuestro sistema que se encarga de almacenar toda la información referente a paradas, recorridos, rutas, etc. Para este sistema se utilizó el SGBD MySQL que es uno de los más populares bajo la filosofía de código abierto (open source).

3.6.1 Diagrama de base de datos

Este diagrama representa cada una de las tablas que conforman la base de datos del sistema, cada una de estas con sus respectivos campos además de todas las relaciones que aseguran la integridad de los datos.

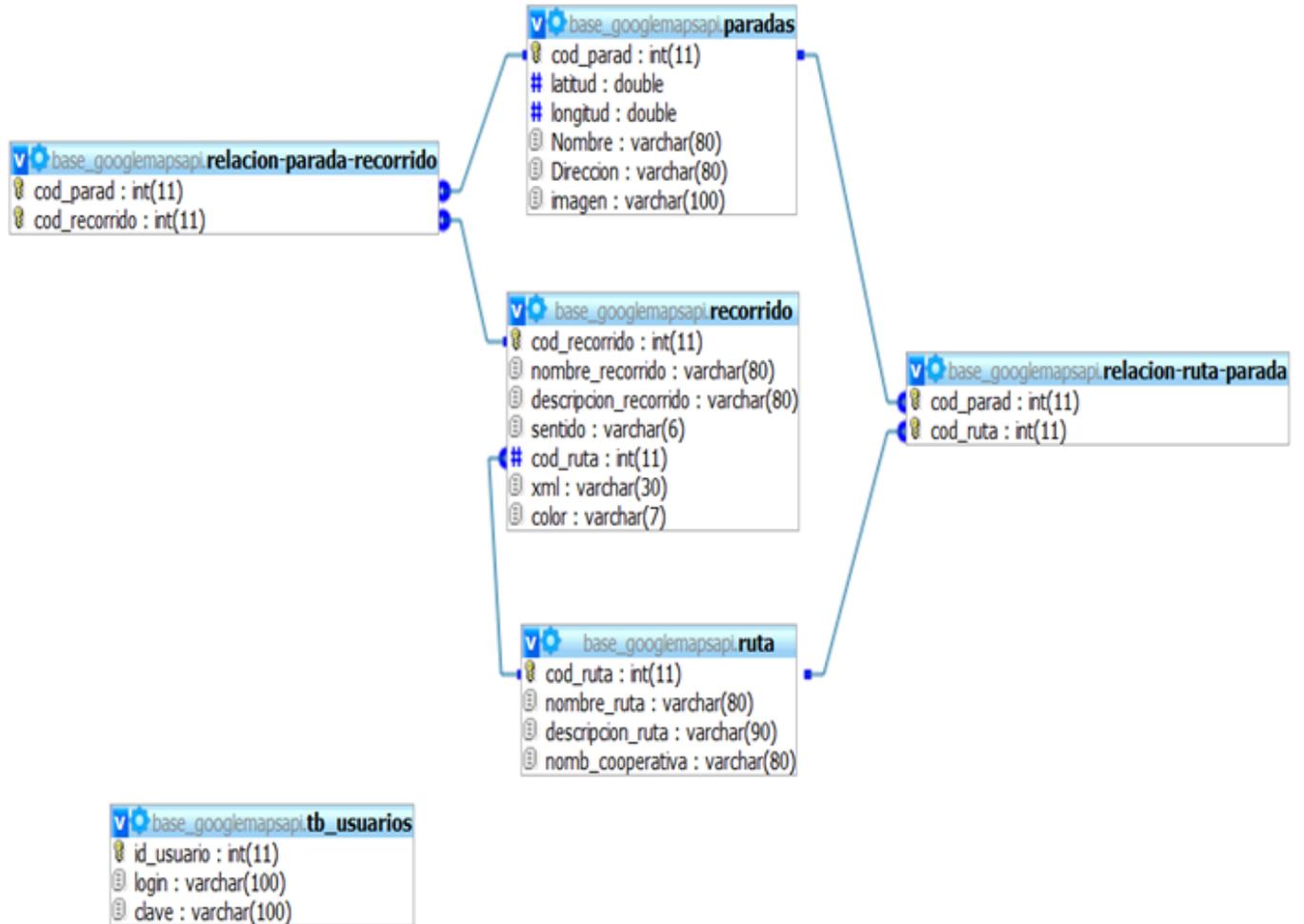


Ilustración 13: Diagrama de Base de datos.

3.6.2 Descripción de las tablas de la base de datos

Paradas: almacena los datos necesarios para poder representar una parada dentro del mapa, así mismo se encarga de almacenar la descripción de cada una de estas paradas.

Recorrido: almacena la información asociada a cada recorrido. Almacenan las coordenadas en formato latitud-longitud de cada recorrido; es aquí donde se establece el sentido de cada recorrido.

Relación parada-recorrido: se establece la relación entre una parada y un recorrido, es decir que paradas pertenecen a un recorrido específico.

Ruta: almacena la información de todas las rutas que componen el sistema.

Relación ruta-parada: se establece la relación entre una parada y la ruta que comprende dicha parada.

Tb_usuarios: esta tabla almacena los datos necesarios para lograr una correcta autenticación por parte del usuario administrador del sistema, esta tabla posee el campo **clave** que almacena por medio de cifrado MD5 la clave de acceso por parte del usuario administrador del sistema.

4. Conclusiones

Una vez que fueron realizadas las pruebas pertinentes se obtuvieron las siguientes conclusiones:

- La arquitectura del sistema permitió desarrollar un sistema que facilita la visualización de la información de las diferentes rutas como son: visualización, edición y eliminación de rutas, recorridos y paradas.
- Se logro la incorporación de diferentes tecnologías web con plataformas de trabajo flexibles para poder integrar la API de Google Maps v3.
- Se comprendió la funcionalidad ofrecida por la API de Google Maps v3 lo cual facilitó la forma de mostrar los recorridos y paradas en el mapa.
- Fue posible crear una cuenta de usuario administrador con todos los privilegios y con una seguridad que garantiza que el sistema sea lo suficientemente robusto.
- La combinación de tecnologías permitió intercambiar información con la base de datos para agregar, mostrar y ocultar los recorridos y paradas en el mapa de forma interactiva.

Por tanto con lo antes mencionado es posible afirmar que se ha diseñado y desarrollado un sistema sencillo y práctico que resuelve la problemática o la necesidad planteada en la definición del problema del presente trabajo.

5. Recomendaciones

Una vez finalizado el presente trabajo se proponen una serie de recomendaciones para la continuación y mejora del sistema aquí plasmado.

- Proponer un algoritmo que permita al usuario interactuar aun más con la aplicación haciendo que éste pueda seleccionar el lugar donde está ubicado y hacia a donde se quiere dirigir y que la aplicación le muestre las rutas a tomar para llegar a su destino.
- Que el usuario le dé clic en un lugar y que se muestren las rutas que pasan cerca de ese punto.
- Buscar la actualización del mapa o trabajar con un mapa que no presente zonas con poca o total visualización por nubes en la Ciudad de León.
- Mejorar el módulo de administración para que la inserción de un nuevo recorrido y parada sea mucho más amigable e intuitiva.
- Mostrar en el sistema las intersecciones entre las diferentes rutas.
- Agregar funcionalidades para que el sistema presente mayor interacción entre el módulo usuario y el módulo administrador.

6. Bibliografía

Andre Lewis, M. P. (2007). *Beginning Google Maps Applications with Rails and Ajax: From Novice to Professional*. USA: APress.

Cobo, A. (2005). *PHP y MySQL: Tecnología para el desarrollo de aplicaciones web*. Madrid, España.

Cornelio Hopmann, María Edith Arce, Leonor Zúniga Gutiérrez. (s.f.). *Centro de Investigación de la Comunicación (CINCO)*. Recuperado el 5/22/2012, de <http://www.cinco.org.ni/archive/398.pdf>

Crokford, D. (2008). *javascript: The good parts*. USA.

Digital Magic Productions Inc 2006 *Digital Magic Productions*

Fundación Wikimedia, Inc. (19 de Junio de 2012). *wikipedia*. Recuperado el 25 de Junio de 2012, de wikipedia: http://es.wikipedia.org/wiki/Extensible_Markup_Language

Google *Google developers*

Google. (10 de Febrero de 2012). *Google developers*. Recuperado el 12 de Abril de 2012, de Google developers: http://www.googlemaps.es/?page_id=3

Google. (16 de 04 de 2012). *Google developers*. Recuperado el 17 de 09 de 2012, de Google developers: <https://developers.google.com/maps/documentation/javascript/maptypes?hl=es>

High Performance JavaScript 2010 O'Reilly Media, Inc

Lengstorf, J. (2010). *Pro PHP and JQuery*. Apress.

Pontificia Universidad Catolica del Peru

Svennerberg, G. (2010). *Beginning Google Maps API v3*. Apress.

The jQuery Foundation *JQuery*

Anexos

1. Interfaces principales del sistema

En la ilustración 14. se presenta la forma en que el usuario visualiza todos los recorridos pre-cargados en el sistema, así como la información asociada a cada uno de estos, esta información se muestra en forma de *Infowindow* al dar clic sobre un recorrido.



Ilustración 14: Vista de todos los recorridos y un Infowindow desplegado.

Cuando se quiere ver los recorridos que realiza una única ruta del conjunto de rutas, los datos se visualizan como se muestra en la Ilustración 15.



Ilustración 15: Visualización de una ruta.

Cuando se desea agregar una parada, se selecciona la ruta y el recorrido ya sea ida o vuelta, como ese muestra en la Ilustración 16.



Ilustración 16: Como se agregar una parada a un recorrido.

En la Ilustración 17 se muestra la vista que aparecerá cuando el administrador desee agregar un recorrido.



Ilustración 17: Como se agrega un recorrido al sistema.

En la Ilustración 18 se muestra la tabla que se visualiza cuando se edita, ya sea la información o la línea perteneciente a un recorrido del mapa o si desea eliminar un recorrido.



Ilustración 18: Tabla para editar los recorridos.

En la Ilustración 19 se muestra el formulario donde el administrador puede editar la información de un recorrido.

Menú Principal

Edición de rutas

Agregar ruta

Editar ruta

Editar recorridos

Cerrar sesión

Ingrese los datos

Editar recorrido

Nombre del recorrido: Recorrido 102_vuelta

Descripción del recorrido: Agregar los barrios

Sentido: vuelta

Ruta a la que pertenece este recorrido: ruta_102

Color: #aaaaff

Guardar Cancelar

Ilustración 19: Formulario para editar un recorrido.