

**Universidad Autónoma Metropolitana Unidad
Azcapotzalco**

División de Ciencias Básicas e Ingeniería

Licenciatura en Ingeniería en Computación

Proyecto de Integración

**Avent-UAM: Aplicación para
compartir automóviles de la UAM
Azcapotzalco**

Alejandra Rojas Bautista
208204951
Trimestre 2014 Invierno

Asesora
Silvia Beatriz González Brambila
Departamento de Sistemas

Yo, Dra. Silvia Beatriz González Brambila, declaro que aprobé el contenido del presente Reporte de Proyecto de Integración y doy mi autorización para su publicación en la Biblioteca Digital, así como en el Repositorio Institucional de UAM Azcapotzalco.

Handwritten signature of Silvia Beatriz González Brambila in cursive script.

Yo, Alejandra Rojas Bautista, doy mi autorización a la Coordinación de Servicios de Información de la Universidad Autónoma Metropolitana, Unidad Azcapotzalco, para publicar el presente documento en la Biblioteca Digital, así como en el Repositorio Institucional de UAM Azcapotzalco.

Handwritten signature of Alejandra Rojas Bautista in cursive script.

Resumen

Debido a la proliferación de automóviles es complicado circular en la ciudad, incluso dentro de nuestra propia universidad, la UAM- Azcapotzalco, encontrar un lugar de estacionamiento disponible es una odisea. Compartir un vehículo con personas que tengan una ruta similar a la nuestra, puede reducir el impacto de éste crecimiento, por lo que contar con una aplicación de cómputo que facilite el contacto entre los usuarios y proporcione la información necesaria para llevar a cabo el enlace, aumentara el uso de esta actividad.

Un ejemplo claro de este problema es el que vivimos los estudiantes del horario matutino, obtener un lugar de estacionamiento es cuestión de tiempo y mucha paciencia, ya que en ocasiones es necesario esperar a que alguien desocupe un lugar. Podemos observar vehículos estacionados en lugares prohibidos, incluso en el césped, esto obviamente ocasiona un caos para las salidas o entradas de otros vehículos. El uso del automóvil no solo es cuestión de estatus social, más bien es una necesidad para algunos universitarios, ya sea por las largas distancias que se recorren para trasladarse, inseguridad en el transporte público, por citar algunas. Una de las soluciones a este problema sería la ampliación del estacionamiento, pero esto ocasionaría gran deterioro de las áreas verdes, considerando que el espacio disponible a utilizar serían estas; pero hay que considerar que existen diferentes soluciones no solo la saturación del estacionamiento dentro de la Universidad sino también para disminuir los contaminantes. Haciendo estos cambios sin alterar a la naturaleza que mucha falta hace en nuestra comunidad.

El proyecto desarrollado es una aplicación web que permite compartir viajes en automóvil a profesores y alumnos de la UAM-Azcapotzalco, por medio de una metodología que utiliza la tecnología de *Google Maps*[1], y el algoritmo *M-Tree* [2] para determinar qué ruta trazada por un conductor es similar a la ubicación que marque el pasajero.

El servicio web de *Avent-UAM*, utiliza la tecnología de *Google Maps* que permite obtener los puntos geográficos de un punto de origen y punto de destino, tanto del conductor como del pasajero según se haya determinado el tipo de usuario que se desea. Estos puntos conforman la ruta, ésta presenta también los puntos intermedios los cuales son hitos que ayudan a determinar si la ruta es similar a la ubicación que marque el pasajero. La información obtenida en el trazado de cada una de las rutas es almacenada en la base de datos.

El trabajo fundamental del algoritmo *M-Tree* es determinar si son similares o no la ruta trazada por el conductor y el punto marcado por el pasajero con ello obtener la ruta más aproximada. Ya que el algoritmo divide los objetos en base a sus distancias relativas, estas se calculan a partir de una métrica, la cual dentro de nuestro proyecto son los puntos geográficos (latitud y longitud) y almacena estos objetos en nodos de tamaño fijo que corresponden a regiones cerradas de un espacio métrico. Esta estructura hace frente al procesamiento de consultas de similitud, con el objetivo de devolver la información en el menor tiempo posible evitando la evaluación de distancias innecesarias y tratando de minimizar los costos por acceso a disco.

Con la ayuda de este Servicio Web se pretende disminuir el número total de automóviles en el estacionamiento de la Universidad, de tal manera que la afluencia de vehículos sea más rápida y existan más espacios disponibles. El apoyo a la economía es inminente ya que el compartir un automóvil también implica compartir los gastos.

Índice General

Resumen	ii
Capítulo 1. Introducción	2
1.1 Objetivos del proyecto	3
1.2 Trabajos Relacionados	4
1.2.1 Referencias Internas	5
1.2.2 Referencias Externas	7
1.3 Panorama general del Proyecto Avent-UAM.....	13
Capítulo 2.-Metodología del Sistema	16
2.1 Modelado del Problema.....	17
2.2 Tecnología <i>Google Maps</i>	19
2.3 Algoritmo <i>M-Tree</i>	20
Capítulo 3.-Implementación del Sistema.....	23
3.1 Interfaz de Inicio del Sistema.....	26
3.2 Selección del Tipo de Usuario	27
3.2.1 Tipo de Usuario Conductor	28
3.2.2 Tipo de Usuario Pasajero	30
3.2.3 Perfil del Usuario	34
3.3 Implementación del Algoritmo <i>M-tree</i>	35
3.4 Estructura de la Bases de Datos	37
3.5 Notificaciones	38
Capítulo 4.-Pruebas al Sistema <i>Avent-UAM</i>	41
4.1 Interfaz y Administración.....	41
4.2 Usuario Conductor	48
4.3 Usuario Pasajero.....	50
4.4 Ejemplos.....	54
Ejemplo 1.....	54
Ejemplo 2	56
Ejemplo 3	59
4.5 Pruebas: Conclusiones.....	64
Capítulo 5.-Conclusiones y trabajos a futuro	67

<u>5.1 Conclusiones</u>	<u>67</u>
<u>5.2 Trabajos a futuro</u>	<u>71</u>
<u>Anexos.....</u>	<u>77</u>
<u>Anexo A: Manual de Administración</u>	<u>78</u>
<u>Anexo B: Instalación y configuración de Microsoft Outlook 2010</u>	<u>85</u>
<u>Anexo C: Manual de Usuario.....</u>	<u>89</u>

Índice de Figuras

Figura 1.1- Interfaz Carpoolworld.....	7
Figura 1.2- Acceso al sistema de Carpoolworld.....	8
Figura 1.3- Vista de rutas disponibles en Carpoolworld.	8
Figura 1.4- Interfaz Carpool.ca.	9
Figura 1.5 Diferentes usuarios de Carpool.ca.	10
Figura 1.6 Alertas de Carpool.ca.	10
Figura 1.7- Interfaz de Aventones.	11
Tabla 1- Tabla comparativa de características.	12
Figura 1.5 Componentes del Sistema <i>Avent-UAM</i>	14
Figura 2.1 Metodología de Espiral	16
Figura 2.2- Modelado del Problema.	18
Figura 2.3- Algoritmo <i>M-tree</i>	21
Figura 3.1- Diagrama de Flujo del Sistema <i>Avent-UAM</i>	24
Figura 3.2- Diagrama de Funcionamiento del Sistema <i>Avent-UAM</i>	25
Figura 3.3.- Diagrama de Secuencia de Interfaz de Inicio del Sistema.	27
Figura 3.4 Diagrama Usuario Conductor.	30
Figura 3.5 Etapas del Proceso del Usuario Conductor.	33
Figura 3.4–Diagrama Entidad-Relación del Sistema <i>Avent-UAM</i>	38
Figura 4.1- Interfaz para la administración del Servidor del Sistema <i>Avent-UAM</i> . ..	42
Figura 4.2- Interfaz para la administración de la base de datos del Sistema <i>Avent-UAM</i>	43
Figura 4.3- Pantalla de Inicio del Sistema <i>Avent-UAM</i>	44
Figura 4.4- Pantalla de Inicio de Sesión del Sistema <i>Avent-UAM</i>	45
Figura 4.5- Pantalla “Alta de Usuario”, Registro de Usuarios del Sistema <i>Avent-UAM</i>	46
Figura 4.6.- Pantalla de Tipo de Usuario del Sistema <i>Avent-UAM</i>	47
Figura 4.7- Interfaz de Usuario Conductor del Sistema <i>Avent-UAM</i>	48
Figura 4.8- Interfaz de usuario “Conductor” del Sistema <i>Avent-UAM</i> , trazado de ruta.	49
En la figura 4.9 observamos que después de que el UC trace su ruta éste puede guardarla, en la pantalla se muestra que está ha sido guardada con éxito.	49
Figura 4.9- Interfaz de Usuario Conductor del Sistema <i>Avent-UAM</i> . Salvar Ruta. ..	49
Figura 4.10- Interfaz de usuario “Pasajero” del Sistema <i>Avent-UAM</i>	50
Figura 4.11- Interfaz de UP del Sistema <i>Avent-UAM</i> , marca de posicionada en la ubicación selecciona por el UP.	51
Figura 4.12- Ruta seleccionada.	52
Figura 4.13- Envío de correo cuando la ruta ha sido aceptada.	53
Figura 4.14- Punto de inicio y fin de la ruta de Efraín.	54
Figura 4.15- Posicionamiento del destino.	54
Figura 4.16- Opciones mostradas por el sistema.	55
Figura 4.17- Mejor opción obtenida y seleccionada por el usuario.	55

Figura 4.18- Punto de inicio y fin de la ruta de Alex.	56
Figura 4.19- Posicionamiento de marca por parte del usuario Ale.	56
Figura 4.20a- Opciones mostradas por el sistema.	57
Figura 4.20b- Ruta cercana 2.....	57
Figura 4.20c- Ruta cercana 3.....	58
Figura 4.21- Mejor opción obtenida y seleccionada por el usuario (Ruta cercana 1).	58
Figura 4.22- ruta diaria de Alex.	59
Figura 4.23- Posicionamiento de marca por parte del usuario Ana.	60
Figura 4.24- Opciones mostradas por el sistema.....	60
Figura 4.25- Mejor opción obtenida y seleccionada por el usuario.	61
Figura 4.26- Mejor opción obtenida y seleccionada por el usuario.	62
Figura 4.27- Opciones Salir de Sesión y Perfil de Usuario.....	62
Figura 4.28- Envío de correo cuando la ruta ha sido aceptada.....	63
Figura 4.29- Interfaz de Microsoft Outlook 2010.	64
Figura 5.1- Ruta aproximada trazada por <i>Google Maps</i>	68
Figura 5.2- Ruta aproximada trazada utilizando <i>steps ways</i>	69
Figura A.1- Interfaz Usuario y contraseña para acceder al sistema de hospedaje gratuito.....	78
Figura A.2- Selección de servicio de Administración del Sistema.	79
Figura A.3- Selección de Admin (administrar la Base de Datos).	80
Figura A.4- Base de datos del Sistema Avent-UAM.	81
Figura A.5- Interfaz del administrador del Sistema <i>Avent-UAM</i>	82
Figura A.6- Administración del Servidor.	83
Figura B.1- Inicio de instalación de <i>Microsoft Outlook 2010</i>	85
Figura B.2- Cuenta de correo electrónico.....	86
Figura B. 3- Configuración de la cuenta de correo electrónico.....	86
Figura B.4- Llenado de registros.	87
Figura B.5- Configuración de información.	87
Figura B.6- Acceso a Microsoft Outlook 2010.	88
Figura C.1- Pantalla de Inicio del Sistema Avent-UAM.....	89
Figura C.2- Pantalla de Inicio de Sesión del Sistema Avent-UAM.	90
Figura C.3- Pantalla de Registro de Usuarios del Sistema <i>Avent-UAM</i>	91
Figura C.4.- Pantalla de Tipo de Usuario del Sistema <i>Avent-UAM</i>	92
Figura C.5- Interfaz de Usuario Conductor del Sistema Avent-UAM.....	93
Figura C.6- Interfaz de Usuario Conductor del Sistema <i>Avent-UAM</i> “Salvar mi ruta”.	94
Figura C.7- Interfaz de Usuario Conductor del Sistema <i>Avent-UAM</i> , opción “Salir de Sesión” y “Perfil de Usuario”.....	95
Figura C.8- Interfaz de Usuario Pasajero del Sistema <i>Avent-UAM</i>	96
Figura C.9- Interfaz de Usuario Pasajero del Sistema <i>Avent-UAM</i> , marca de la ubicación selecciona.	97
Figura C.10- Ruta seleccionada.....	98

Figura C.11- Envío de correo cuando la ruta ha sido aceptada..... 99
Figura C.10- Interfaz de Usuario “Pasajero” del Sistema *Avent-UAM*, opción “Salir
de “Sesión” y “Perfil de Usuario”. 100

|

Capítulo 1. Introducción

El gran avance tecnológico en esta época mejora la comunicación entre las personas, ayuda a contar con información rápida, segura y confiable, entre muchas otras cuestiones. No solo hemos crecido tecnológicamente, también la población ha ido en aumento. Actualmente se tienen problemas severos en cuestiones ambientales y saturación de vehículos, por mencionar algunos, afortunadamente se puede utilizar la tecnología a nuestro favor para disminuir los efectos de tal crecimiento. Algunos países e instituciones han adoptado alternativas para ayudar a combatir algunos de estos problemas. Uno de ellos es la compartición de automóviles, la cual consiste en que un grupo de personas con destinos similares, se contactan y comparten el medio de transporte. Uno de los mecanismos más comunes para realizar el contacto es por medio de páginas webs diseñadas para este fin, como *carpool.ca*, *aventones.com.mx*, por citar algunos. Esto con la finalidad de compartir los gastos de combustible y de los vehículos, con la seguridad de que la persona con quien se viaja es confiable.

La posibilidad de optimizar los recursos compartiendo el automóvil con otras personas que tengan el mismo destino, dividiendo los gastos que esto significa. Así también, una persona que no tenga automóvil puede buscar a otra que desee compartir el suyo.

Actualmente este tipo de iniciativas son mayormente utilizadas en países como Estados Unidos y Canadá [3] donde incluso existen paradas y carriles destinados exclusivamente para este tipo actividades. En México no se ha tenido un gran auge, al iniciar la investigación de este proyecto encontramos un Carpool¹ en México *www.dameunavento.com.mx*[4] el cual cuenta con más de 1500 usuarios, pero debido a la inseguridad que se vive en nuestra ciudad (Distrito Federal) no ha tenido el impacto que deseado.

Muchos de los estudiantes de la UAM-A deben recorrer grandes distancias desde su hogar hasta la universidad todos los días, ya sea en vehículo particular o en transporte público. El presente trabajo plantea el diseño e implementación de un sistema que facilite el contacto entre un pasajero que desea viajar y un conductor que desea compartir su auto, mediante el uso de un algoritmo que compare la similitud del origen y destino de ambos y seleccione las aproximaciones entre las rutas. Este tipo de iniciativas ayuda a los usuarios a viajar más seguros y disminuir sus gastos en viajes, solo por mencionar algunos.

¹Carpool. Es la práctica que consiste en que diversas personas compartan coche tanto para viajes periódicos como para trayectos de largas distancias.

1.1 Objetivos del proyecto

El objetivo general del proyecto se describe a continuación:

- ➔ Desarrollar un sistema de enlace entre conductores y pasajeros.

Se desarrolló un sistema que enlaza a usuarios conductores y a usuarios pasajeros para que exista una relación que satisfaga las necesidades de cada uno.

Objetivos específicos del proyecto:

- ➔ Establecer una conexión entre el sistema y el servicio de mapas de *Google*.

Se estableció la conexión del sistema al servicio de *Google Maps*, para realizar el trazado de las rutas que satisfagan la solicitud de cada uno de los usuarios. Una ruta aproximada para el conductor y la obtención de los puntos geográficos de la ubicación de inicio o fin de destino del pasajero.

- ➔ Implementar y adecuar los algoritmos de búsqueda de conductores y pasajeros.

Adecuación del algoritmo *M-Tree* al sistema para que cumpla con las características necesarias para su funcionamiento idóneo: centro con radio cobertor, solapamiento, métrica, árbol balanceado y paginado. Ya que éste es muy robusto y satisface necesidades de búsqueda para bases de datos muy grandes, ha sido necesario reducir su dimensionamiento y adecuar sus características.

- ➔ Diseñar la Base de Datos a utilizar en la aplicación.

Se ha diseñado una Base de Datos orientada a cubrir las necesidades del sistema, sin redundar en información, agregando solo lo necesario para satisfacer los requerimientos que son: 1 GB de almacenamiento, Sistema Operativo *Linux*, *Apache*, *PHP*, *MySQL*, arquitectura i686. El host *260MB* es gratuito, por ende es necesario administrar adecuadamente los recursos.

- ➔ Diseñar un módulo de perfil de usuario

El módulo de perfil de usuario ha sido diseñado para el apoyo en el sistema, contar con información que sea útil para los usuarios ya que cada uno cuenta con características particulares, diferentes hábitos, tipo de música, etc. con estos datos se pretende tener un perfil general de las personas.

- Diseñar una interfaz que permita utilizar la aplicación sin entrenamiento previo al usuario.

La interfaz se ha diseñado para que el usuario no requiera de capacitación previa a su uso, es sencilla, no requiere de mucho conocimiento, cada una de las ventanas nos indicaran el tipo de información requerida. Se anexa manual para el uso del sistema (Anexo C. Manual de Usuario).

- Verificar que la aplicación cumpla con los lineamientos mínimos establecidos en el diseño de la aplicación.

A lo largo del documento se muestra el cumplimiento de los lineamientos establecidos en la propuesta. Satisfaciendo las necesidades de los usuarios al obtener una ruta aproximada para cada una de las solicitudes. Ofreciendo un resultado satisfactorio de acuerdo a los resultados arrojados por el algoritmo.

1.2 Trabajos Relacionados

La compartición de automóviles no es una práctica que se lleve a cabo dentro de nuestro país de manera usual. Algunas empresas privadas y universidades privadas [3] de igual manera adoptan esta actividad, la cual les ha funcionado; es por ello que existen empresas privadas de diseño de *carpool* (compartición de automóviles).

Dentro de la UAM-A no se tienen registros de una aplicación similar de compartición de automóviles para estudiantes y/o profesores, podemos encontrar proyectos que tienen características similares[5,6,7]; esto es utilizando el algoritmo GT^2 el cual es utilizado para encontrar la ruta más corta, o también por medio de métricas. Observando que los proyectos fueron fijados para calcular las rutas, conocer distancias, saber si la ruta es la mejor alternativa para el usuario; pero no para determinar si un viaje puede ser compartido por más de una persona debido a que las rutas son similares.

Dentro de los proyectos externos encontrados, podemos observar que existen compañías de software especializadas en el diseño de los mismos. Empresas que impulsan a sus trabajadores a reducir contaminantes, también por la falta de espacios de estacionamiento en sus instalaciones. Problemas como estos son los que conllevan a que sea necesaria la implementación de este tipo de programas adecuándose a las características propias de cada situación y cultura.

² GT se asemeja a un recorrido en profundidad dentro de un grafo dirigido, no contiene ciclos.

1.2.1 Referencias Internas

Búsqueda de rutas más cortas entre diferentes sistemas de transporte en la Ciudad de México aplicando el enfoque GT [5].

En este proyecto se planteó diseñar e implementar una aplicación que permite encontrar la ruta más corta para llegar a una zona geográfica a partir de un punto específico aplicando el enfoque del Modelo GT.

Se calcula la ruta más corta entre un punto y una zona geográfica, usando solo medios de transporte públicos de gran relevancia en la ciudad, como lo es el Sistema de Transporte Colectivo Metro (incluyendo tren ligero), el *Metrobus*³ y las rutas del sistema *RTP*⁴, los cuáles son usados a diario por millones de personas para llegar a su trabajo, escuela, casa.

Presenta una interfaz gráfica en donde el usuario puede elegir el origen (Estación/Parada) y como destino una zona geográfica determinada, las posibles zonas geográficas destino se presentan en un menú para que el usuario elija la que desea. En base a la información introducida por el usuario sobre el origen y el destino, y a la información contenida en los grafos G, T y otras tablas construidas, aplica el algoritmo para resolver el problema de ruta más corta extendido generando la ruta que se presentará en pantalla. Después de haber encontrado la ruta más corta entre el origen y el destino, se despliega en pantalla lo que el usuario debe realizar para llegar a su destino, como pueden ser transbordos y cambios de sistema de transporte.

El proyecto solo muestra la información que necesita el usuario para llegar a su destino de manera más rápida en nuestro proyecto *Avent-UAM* obtenemos la ruta más aproximada y podemos adecuarla a nuestros requisitos nos proporciona no solo una opción si no un listado de las mejores.

Aplicación para dispositivos con Android⁵ que encuentre la mejor ruta entre dos estaciones del Metro [6].

Este proyecto plantea implementar una aplicación para un dispositivo con sistema operativo *Android* que permita obtener la mejor ruta entre una estación origen y una estación destino del Sistema de Transporte Colectivo Metro.

³Metrobus Sistema efectivo de transporte publico http://www.metrobus.df.gob.mx/que_es_metrobus.html

⁴RTP Red de transporte de Distrito Federal <http://www.rtp.gob.mx/>

⁵Android Sistema operativo de código abierto para dispositivos móviles.

Este proyecto también utilizará la tecnología *Google Maps*, pero es enfocado a las rutas del Metro de la Ciudad de México. El usuario proporcionará la estación origen, la estación destino y la métrica elegida para obtener la mejor ruta en cuanto al mínimo número de estaciones a recorrer, el mínimo kilometraje a recorrer, o el mínimo tiempo estimado que cueste dicho trayecto. Además, si el usuario desea podrá introducir estaciones por las cuales desee o no pasar en su recorrido. O también se puede hacer de por medio de una conexión a internet; el usuario no introducirá la estación origen ya que está se obtendría automáticamente. En nuestro proyecto podemos determinar el punto de origen y punto de destino, pero solo uno ya que es parte del sistema tener configurado un punto en particular que es la universidad. Así como modificaciones intermedias de la ruta realizándose desde cualquier punto para satisfacer los requisitos de los usuarios.

Sistema para obtener información de distancias y recorridos de un par de elementos de un conjunto de sitios localizados en un mapa [7].

Este proyecto plantea una aplicación que permite obtener la matriz de distancias entre dos puntos de una ciudad a partir de un mapa de vialidades, por medio del Modelo SIG⁶; así como también el Algoritmo Dijkstra⁷ este es aplicado sobre la pareja de puntos a seleccionar sobre el mapa digital obtenido.

El sistema permite separar la información en diferentes capas temáticas y las almacena independientemente, permitiendo trabajar con ellas de manera rápida y sencilla, facilitando al profesional la posibilidad de relacionar la información existente a través de la topología de los objetos, con el fin de generar otra nueva que difícilmente podríamos obtener de otra forma.

El sistema tiene como datos de entrada la información entera del mapa digital (número de casa, calle, colonia, municipio ó delegación).

El sistema permite al usuario elegir los puntos al calcular la ruta sobre el mapa.

A diferencia de nuestro proyecto, permite no solo obtener la información de la ruta, si no manipularla de acuerdo a la necesidad de cada usuario y cumple la finalidad de enlazar dos usuarios con requisitos similares.

⁶ Modelo SIG es una integración organizada de hardware, software y datos geográficos diseñada para capturar, almacenar, manipular, analizar y desplegar en todas sus formas la información geográficamente referenciada.

⁷ Algoritmo Dijkstra es un algoritmo para la determinación del camino más corto dado un vértice origen al resto de vértices en un grafo con pesos en cada arista.

1.2.2 Referencias Externas

Carpoolworld [8].

Es un sistema mundial con buena presencia en Europa y Norteamérica, se especializa en viajes de largas distancias y está basado en *Google Maps*. Se desconoce el motor de búsqueda por ser de uso comercial (Figura 1.1).

Trabaja por medio de grupos, hace referencia a clubs que pueden crearse para una mayor eficiencia, más personas dentro de estos grupos, menor será el costo de cada viaje para cada uno. Contiene la sección gratuita y la sección Premium. La primera nos presenta un listado de aquellos destinos con lugares disponibles dentro de cada unidad, se puede ingresar obteniendo usuario y contraseña, el servicio web se deslinda de cualquier problema que se suscité en el contacto con usuarios gratuitos, ya que la forma de registro es general. El servicio Premium se focaliza más en conjuntos estos son escuelas, empresas públicas y privadas así como comunidades. Donde la interacción es privada y más segura por consiguiente. El servicio Premium tiene costo.

El Sistema *Avent-UAM* se asemeja a la parte Premium de este servicio ya que solo es para usuarios de la Universidad, de la misma manera se utiliza el sistema de *Google Maps* para la obtención de rutas; pero utilizando las marcadas por default en el sistema.



Figura 1.1- Interfaz Carpoolworld.

La figura 1.2 muestra la forma de acceso que utiliza el sistema CarpoolWorld, se necesita un usuario y contraseña previamente establecida, también se puede acceder mediante otros sistemas de mensajería.



Figura 1.2- Acceso al sistema de Carpoolworld.

En la figura 1.3 podemos observar como el sistema muestra las rutas disponibles al momento, espacios disponibles, costo parcial, agenda del viaje si es diariamente, semanal o mensual.

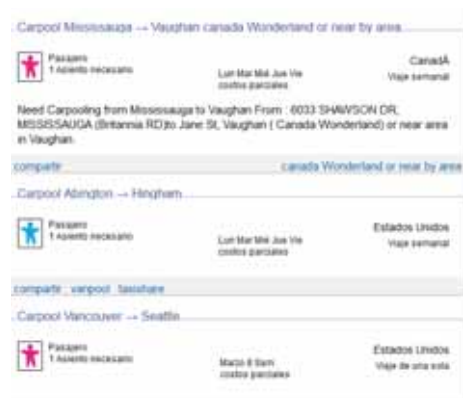


Figura 1.3- Vista de rutas disponibles en Carpoolworld.

Carpool.ca [9]

Es un sistema con presencia en Estados Unidos y Canadá con un total de más de 30,000,000 de usuarios. Anteriormente se tenía presencia en México, pero la demanda no fue la esperada, por el momento trabaja con número no mayor a 1000 usuarios. El buscador se basa en direcciones y poblaciones, por lo que tiene buen desempeño en largas distancias pero pobre desempeño en cortas, por ejemplo, viajes dentro de una ciudad a otra, utilizando la aplicación de *Google Maps*. Se desconoce el motor de búsqueda por ser de uso comercial. De igual manera trabaja de manera gratuita y para conjuntos como escuelas, empresas privadas y públicas así como comunidades que requieran de este servicio. Principalmente en aquellas que cubren largas distancias para llegar a sus destinos, de tal manera que pueden disminuir

gastos. En estos países el gobierno proporciona apoyo a las personas que tienen este tipo de iniciativas, ver Figura 1.4.

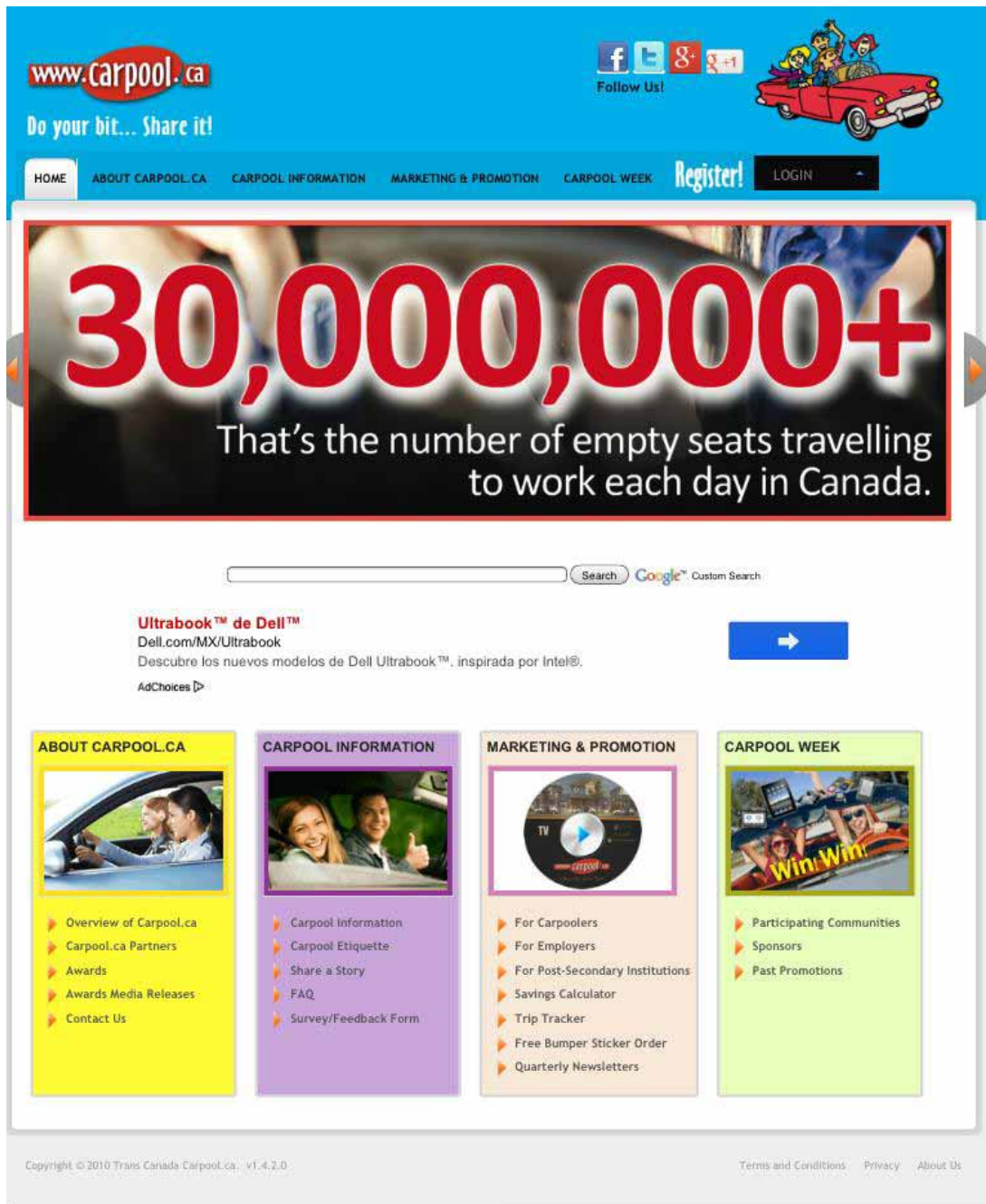


Figura 1.4- Interfaz Carpool.ca.

La figura 1.5 muestra las diferentes variantes de usuarios, beneficios, a quien se enfoca más, preguntas frecuentes, contactos.



Figura 1.5 Diferentes usuarios de Carpool.ca.



La figura 1.6 muestra las alertas que envía el sistema carpool.ca para sus usuarios, con este tipo de alternativas los usuarios están informados de viajes disponibles.

Figura 1.6 Alertas de Carpool.ca.

Aventones [10].

Es un sistema mexicano diseñado para distancias cortas (ver Figura 1.7), sin embargo la participación está limitada a las empresas públicas o privadas así como instituciones educativas, no es de uso gratuito solo puede ser utilizado por aquellos que hayan adquirido los derechos. Está basada en *Google Maps*, se desconoce el motor de búsqueda. Este proyecto ha tenido en los últimos dos años un gran auge en los países de habla hispana y se pueden adquirir ya en países como Colombia y Argentina, entre otros. El proyecto propuesto es de uso gratuito el cual trabaja tanto para distancias cortas y largas, el usuario pasajero decide desde que punto iniciar el viaje o terminarlo, y en el caso del conductor este puede manipular la ruta según lo requiera.

Figura 1.7- Interfaz de Aventones.

La tabla 1 resume las características de las diferentes aplicaciones mencionadas y permite compararlas con el proyecto *Avent-UAM*.

	Algoritmo	Distancias cortas	Distancias largas	Fines lucrativos	Interfaz	Registro de usuarios para su uso	Publicidad En la web	Servicios Premium	Manipulación de Rutas
Carpoolworld	Privado	si	si	Si	Compleja	No	Si	Si	No
Carpool.ca	Privado	no	si	Si	Compleja	Si	Si	No	No
Aventones	Privado	Si	Si	Si	Simple	Si	Si	No	No
Sistema para obtener información de distancias y recorridos de un par de elementos de un conjunto de sitios localizados en un mapa	Se desconoce	Limitado a distancias cortas	No se utiliza en distancias largas	No	Se desconoce	Si	No	no	No
Búsqueda de rutas más cortas entre diferentes sistemas de transporte en la Ciudad de México aplicando el enfoque GT	Algoritmo GT	Limitado a distancias cortas (solo Ciudad de México)	No	No	Se desconoce	Si	No	No	No
Aplicación para dispositivos con <i>Android</i> que encuentre la mejor ruta entre dos estaciones del Metro	Se desconoce	Limitado a distancias cortas(solo estaciones del metro de la Ciudad de México)	No	No	Se desconoce	No	No	No	No
Avent-UAM: Aplicación para compartir automóviles de la UAM Azcapotzalco	M-Tree	Si	Si	No	Simple	Si	Si	no	Si

Tabla 1- Tabla comparativa de características.

Como podemos observar en la tabla de características, el sistema *Avent-UAM* sobrepasa el sistema de manipulación de la ruta trazada en los proyectos similares a éste. Puede trabajar tanto en largas y cortas distancias no está limitado. En los proyectos similares de la universidad si existe diferente, pues estos se enfocan en un solo punto estratégico, obtención de información, menor distancia; pero de igual manera el usuario no puede manipular información.

1.3 Panorama general del Proyecto Avent-UAM.

Esta sección detalla la metodología utilizada en el desarrollo del Sistema *Avent-UAM*, los procedimientos desarrollados y características peculiares, así como la implementación de cada uno de los componentes que se detallan en los capítulos 2 y 3.

Los componentes principales que conforman el sistema *Avent-UAM*, son:

Interfaz de Inicio del Sistema: Permite dar de alta a nuevos usuarios, o ingresar al sistema en caso de ser miembro activo.

Tipo de Usuario: Permite seleccionar entre los dos diferentes tipos de usuario que existen dentro del sistema (pasajero y conductor), cada uno con sus respectivas características.

Base de Datos: Almacena la información de los módulos. Sirve como fuente de datos para estadísticas.

Perfil de Usuario: Permite capturar las características relevantes del usuario, como son datos de contacto e información necesaria para formar un perfil básico de la persona con quien se compartirá el viaje.

Notificaciones: Permite comunicar al conductor con el pasajero y mantener actualizada la información por medio de correos electrónicos con información como:

- Citas pendientes
- Modificaciones

Enrutador: Traza la ruta mediante la aplicación de *Google Maps*, para su publicación. Obtiene los vectores de la ruta para su evaluación en los puntos origen o destino del pasajero mediante la implementación del algoritmo *M-Tree*.

En la Figura 1.8 se observa la relación entre cada uno de ellos, la correspondencia y la pertenencia.

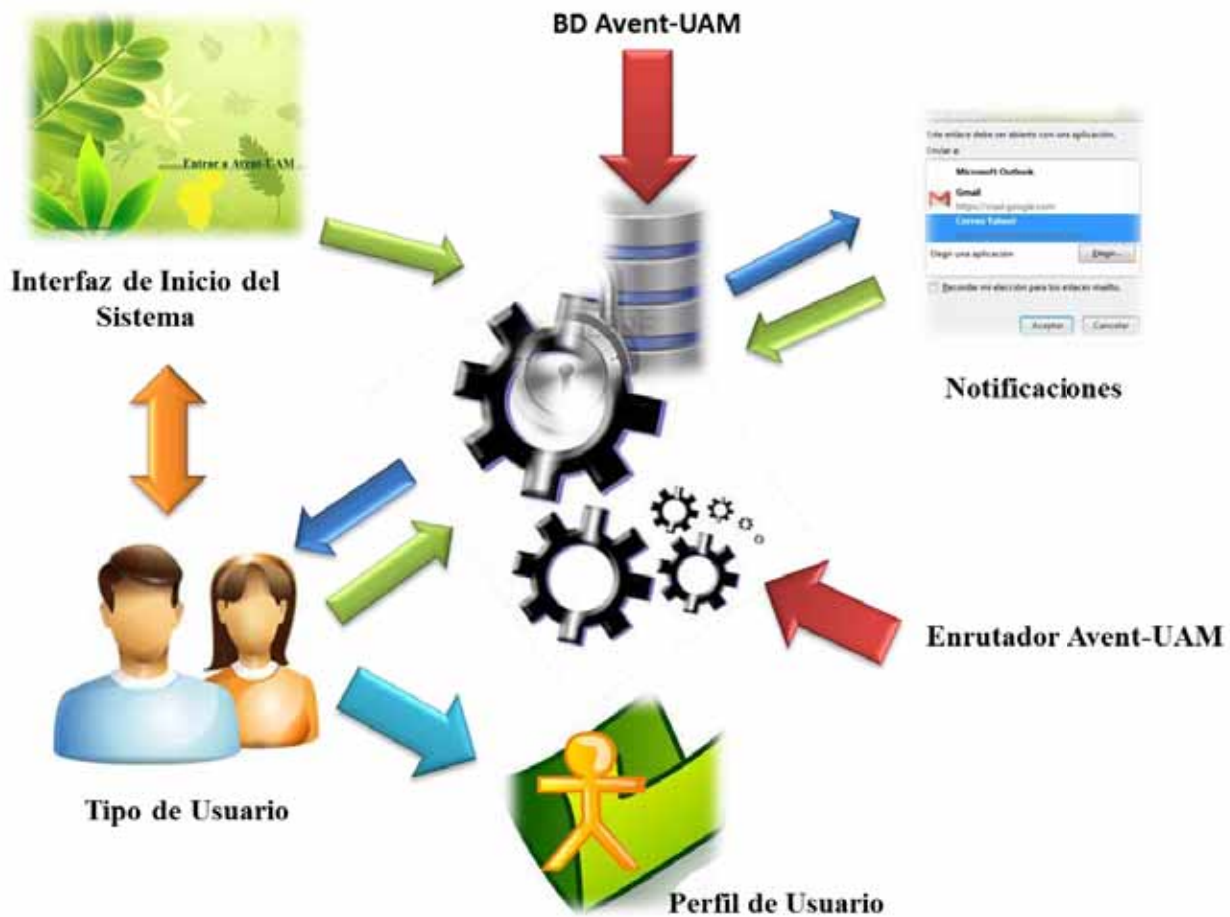


Figura 1.5 Componentes del Sistema *Avent-UAM*.

El siguiente capítulo plantea el problema en el que se basa el diseño del sistema, refiere a los procedimientos para cumplir los objetivos establecidos.

Capítulo 2.-Metodología del Sistema

Este capítulo hace referencia al conjunto de procedimientos que se desarrollaron para alcanzar el objetivo del proyecto, plantear una aplicación que permita a usuarios que tengan automóvil compartir su vehículo con otros que en algún punto o puntos de la ruta encuentren alguna similitud esto es, ya sea que su origen de partida sea la UAM-A para ambos y que en la ruta hacia su destino final exista uno o más puntos similares en el cual el usuario (pasajero) pueda acercarse más a su destino; o que el punto de destino sea la UAM-A, y que de igual manera exista un punto de similitud para que el conductor intercepte al usuario.

A continuación se explica el modelado del problema, el porqué del desarrollo de este sistema; así como el uso de la API *Google Maps* que es una herramienta de uso gratuito en el cual se basa el sistema. El algoritmo *M-tree* el cual es el corazón del sistema *Avent-UAM*, su estructura y su complejidad.

Se utilizó la metodología de Espiral [11], la cual establece que la etapa de Diseño del proyecto fue mediante la propuesta del mismo, así como el planteamiento del problema y conocimiento previo del mismo. La etapa de Desarrollo, se realizó mediante la exposición del proceso donde se establecieron las herramientas necesarias para cubrir cada uno de los objetivos planteados. La etapa de Elaboración consistió en realizar cada parte del proyecto, programación de código en diferentes lenguajes Java, Java Script, HTML, PHP, así como la interoperabilidad entre ellos. La metodología de Espiral también establece la etapa de mantenimiento, pero esto se llevara a cabo en un futuro. En la fig. 2.1 aparece la forma genérica de la metodología en Espiral, ver Figura 2.1.

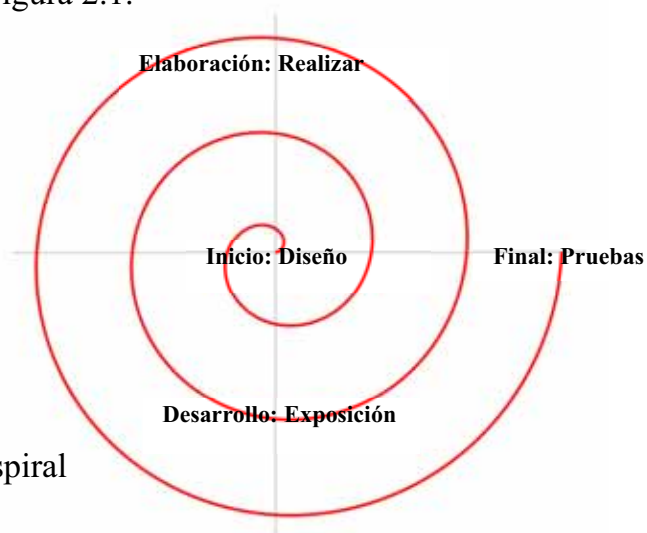


Figura 2.1 Metodología de Espiral

2.1 Modelado del Problema

El diseño del sistema en una parte fundamental del proyecto, es el inicio, partiendo de una idea creada por una necesidad, la necesidad de vivir mejor. Aunque la tecnología logra satisfacer cada día más nuestros requerimientos, actualmente el problema del tránsito desmedido, la inseguridad, el recorrido de distancias considerablemente largas conlleva a desarrollar servicios que cumplan este tipo de requerimientos. Esta problemática la acotamos a los estudiantes y profesores de la UAM-A, que debido al ascenso vehicular dentro de las instalaciones es más complicado encontrar un espacio disponible ya que el estacionamiento continúa siendo el mismo espacio designado para ello.

El proyecto está desarrollado para dos tipos de usuario. El primer tipo (pasajero) que necesita dirigirse a un punto en particular ya sea partiendo desde la Universidad o hacia ella, y un segundo (conductor) que presta su servicio de compartición de automóvil para disminuir sus gastos, o simplemente tener un acompañante. Se limita a dos ya que por el momento el sistema se encuentra en su primera fase, en miras a futuro encontrando satisfacer necesidades que surjan con el uso del mismo, de tal manera que el servicio sea más completo.

El encontrar un punto inicial, intermedio o final similar entre la ruta que el usuario (conductor) utiliza para llegar a su destino y un punto geográfico marcado por el usuario (pasajero) que requiere para llegar a su destino. La búsqueda de la opción aproximada para cada usuario pasajero. La relación que se establece entre los usuarios, la comunicación previa establecida mediante correos electrónicos, o por medio de la lectura de los perfiles de cada usuario que se encuentran dado de alta en el sistema la cual en conjunto nos proporciona mayor seguridad en el uso del servicio. Cubriendo con ello la necesidad de reducir gastos, viajar seguro, disminuir contaminantes, etc.

La figura 2.2 muestra el modelado del problema. Un usuario pasajero que necesita compartir gastos, que proporciona un viaje seguro, que sufre los problemas de falta de espacios en el estacionamiento de la Universidad; ya sea que su punto de origen sea la UAM-A o sea su punto destino. Y un usuario pasajero que igualmente necesita compartir gastos, tener un viaje seguro, y que por falta de vehículo o ahorro de dinero necesita un servicio de *carpooling*. En la parte central de la figura observamos la distancia a recorrer por cada uno de los usuarios así como el punto similar que existe entre las dos rutas.

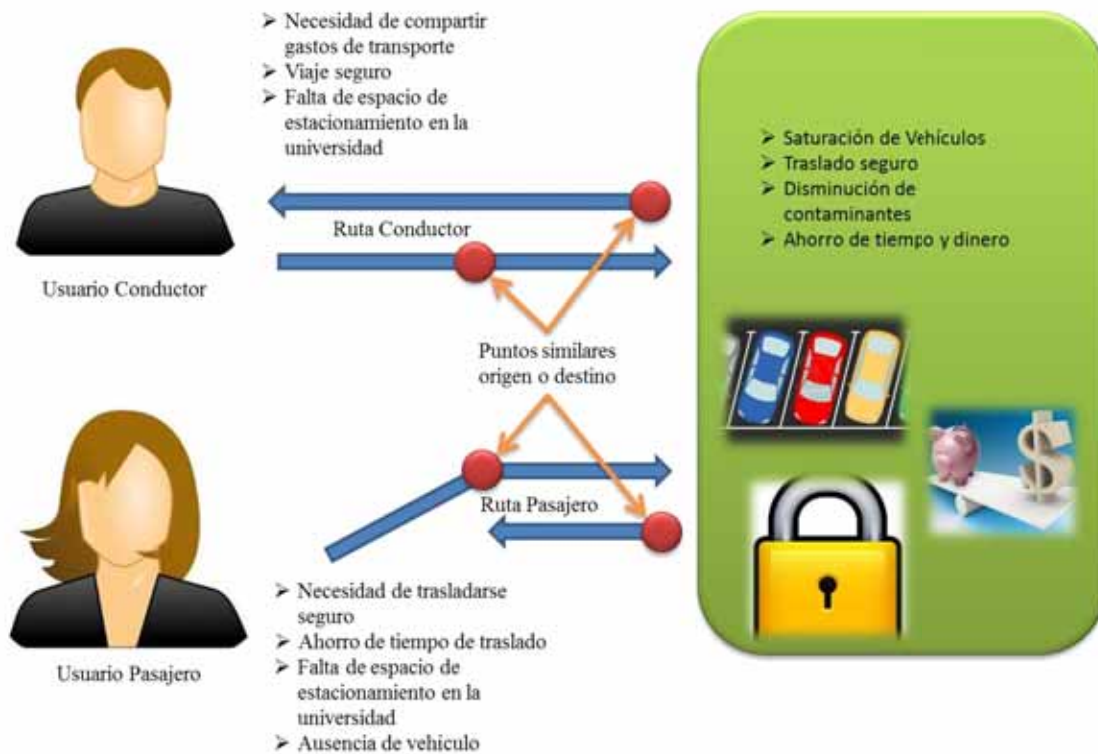


Figura 2.2- Modelado del Problema.

2.2 Tecnología *Google Maps*

Google Maps es un servidor de aplicaciones para visualizar mapas en la Web. Los usuarios pueden ingresar una dirección, una intersección o un área en general para buscar en el mapa. El API de JavaScript de *Google Maps* permite insertar mapas en páginas web. La versión 3 está especialmente diseñada para proporcionar una mayor velocidad y que se pueda aplicar más fácilmente. Por estas razones se decidió utilizarla en el Sistema *Avent-UAM*.

Esta API es utilizada dentro de este proyecto para realizar el trazado de las rutas tanto del usuario pasajero como del usuario conductor.

Google Maps contiene funciones diseñadas, las cuales únicamente se introducen dentro del código, por tanto es necesario tener una conexión a internet establecida para el uso del servicio. Algunas de las funciones son restringidas ya que ofrece también servicios más completos o extendidos para uso empresarial.

Las coordenadas de la aplicación están en el sistema WGS84¹ y se mostrará la latitud y la longitud, positiva para Norte y Este, negativa para Sur y Oeste.

El servicio cuenta con diferentes API'S:

- de rutas
- de matriz de distancia
- de elevación
- de codificación geográfica
- de Google Places

Dentro de nuestro proyecto utilizaremos la API de rutas. El API de rutas de *Google* es un servicio que utiliza una solicitud HTTP para calcular rutas para llegar de una ubicación a otra. Se pueden buscar rutas de varios métodos de transporte, como en transporte público, en coche, a pie o en bicicleta. Las rutas pueden especificar los orígenes, los destinos y los hitos como cadenas de texto. En éste proyecto solo es necesario el método de transporte del coche.

¹WGS84 es un sistema de coordenadas geográficas mundial que permite localizar cualquier punto de la Tierra (sin necesitar otro de referencia) por medio de tres unidades dadas. Su desarrollo fue obtenido a partir de todos los avances en teoría, sofisticación en la tecnología computacional y contando con la disponibilidad de las nuevas y más completas series de datos. <http://www.cocesna.org/subpagina.ais.php?id=274&lng=0>

Por lo general, este servicio está diseñado para calcular rutas a partir de direcciones estáticas (conocidas previamente) para la ubicación del contenido de la aplicación en un mapa. Sin embargo, este servicio no está diseñado para responder en tiempo real a la información introducida por el usuario, por ejemplo calcular rutas dinámicas (elemento de interfaz de usuario).

El cálculo de indicaciones es un proceso que consume mucho tiempo y muchos recursos. Siempre que sea posible, realiza un cálculo previo de las direcciones conocidas y almacena los resultados en una memoria caché temporal que se haya diseñado con tal fin, por lo cual este servicio únicamente se realiza cuando se tiene una conexión a internet.

El uso del API de rutas de Google está sujeto a un límite de 2,500 solicitudes de rutas al día. Todas las búsquedas de indicaciones contarán como una única solicitud respecto al límite diario cuando el modo de transporte es en coche.

2.3 Algoritmo *M-Tree*

“M-tree es un método que permite organizar y realizar búsquedas en grandes conjuntos de datos, la proximidad entre objetos se define a través de una función de distancia. Los datos se representan en un árbol balanceado y paginado, es decir, usa páginas de tamaño fijo con cantidad de nodos u objetos variables, para su construcción se elige un conjunto de centros, los cuales almacenarán su radio cobertor” [12].

El algoritmo *M-Tree* permite construir un árbol de n nodos donde cada nodo está conformado por un arreglo bidimensional que representa un punto geográfico.

Puntos geográficos = (latitud, longitud)

El algoritmo *M-Tree* puede tener un gran solapamiento dependiendo de la ruta trazada y el punto marcado por el usuario. Considere la Figura 2.3, donde se muestra la simulación del algoritmo M-tree, la línea azul dentro del círculo rojo es el solapamiento que podemos observar en las dos rutas que son similares en un punto, el punto similar está marcado en rojo.

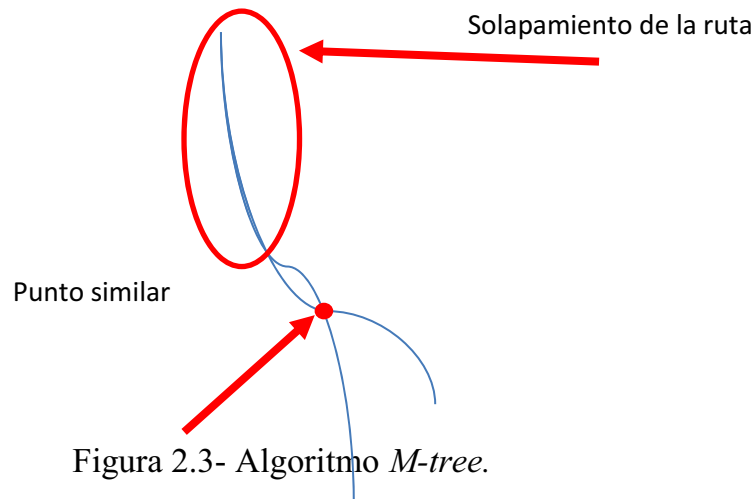


Figura 2.3- Algoritmo *M-tree*.

El algoritmo *M-Tree* tiene componentes y sub-componentes:

Los componentes son todas las rutas de los conductores existentes en la base de datos. Éstas forman cada uno de los subárboles del *M-tree*, teniendo como raíz el punto de partida de todas las rutas, la UAM-A.

Otro componente es el método para comparar una ruta con un punto, y entregar la distancia mínima.

Los subcomponentes son los puntos que forman cada ruta, los puntos que conforman la ruta y generan cada subárbol. Cada punto está formado por dos datos, latitud y longitud.

Al solicitar las indicaciones de la ruta se obtienen los puntos, con cada punto se va generando el árbol:

Aquellos nodos que no son hojas.

Un conjunto de objetos de enrutamiento.

Puntero al padre objeto RAIZ (UAM-A) del nodo.

Aquellos nodos que son hojas.

Un conjunto de puntos (puntos geográficos).

Puntero al padre objeto RAIZ del nodo.

El objeto de enrutamiento.

Distancia de un nodo al nodo raíz.

Radio cobertor distancia entre cada uno de los nodos al nodo raíz tomando como raíz (*UAM Azcapotzalco*).

El puntero de nuestro árbol es la *UAM Azcapotzalco*.

Objeto

En nuestro caso utilizaremos como objeto el punto geográfico marcado por el usuario pasajero.

Si bien no se cuenta con servicios web que apoyen este tipo de iniciativas para disminuir conglomeración vehicular en diferentes puntos estratégicos de forma gratuita, o el apoyo por parte de autoridades para fomentar el uso de compartición de automóviles. Podemos diseñar un sistema que apoye a los estudiantes y profesores de la UAM-Azcapotzalco. No es tarea fácil, las herramientas para realizar este tipo de sistemas las podemos encontrar de forma gratuita, tienen un grado de complejidad cada parte que comprende el sistema; aunque realizando una interoperabilidad de forma adecuada se obtendrá un resultado favorable.

El siguiente capítulo detalla paso a paso la implementación del Sistema *Avent-UAM*, cada uno de los procedimientos y métodos utilizados en el desarrollo del mismo. Estructura de la base de datos, implementación de la API de *Google Maps*, así como el diseño y modificaciones al algoritmo *M-tree* para cumplir con los requerimientos que el Sistema *Avent-UAM* así requiera.

Capítulo 3.-Implementación del Sistema

Este capítulo detalla paso a paso los procesos y métodos que se llevaron a cabo para la implementación del sistema, herramientas de la API de *Google Maps* utilizadas, así como la adecuación del Algoritmo *M-Tree*. Diseño de la base de datos y especificaciones de cada módulo.

El servicio utiliza el motor de Base de Datos *SQL*[13], para una administración adecuada del servicio.

El diseño del sistema permite familiarizarse con él de una manera fácil y rápida. La interfaz nos ayuda a explorar el contenido y sus características para un mejor desempeño del mismo.

La figura 3.1 muestra el proceso de entrada de un usuario pasajero o conductor en el Sistema *Avent-UAM*, y su correspondiente salida. Selecciona la opción deseada de acuerdo a sus requerimientos, obteniendo una ruta que concuerde con su solicitud.

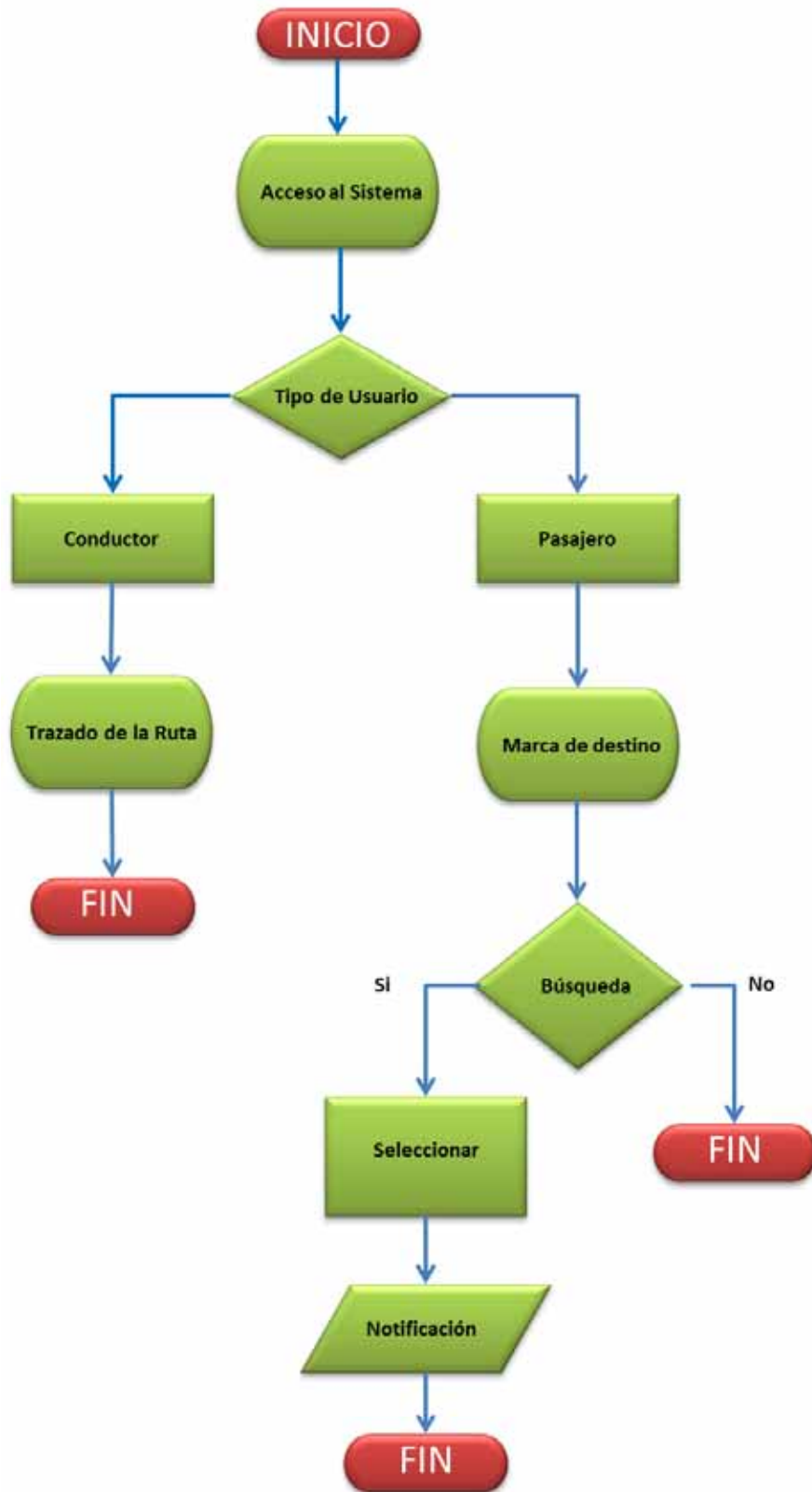


Figura 3.1- Diagrama de Flujo del Sistema *Avent-UAM*.

Los componentes principales y su interacción los observamos en la Figura 3.2. Iniciando con el acceso al sistema, seleccionado el tipo de usuario que se requiere. En el caso de usuario Conductor, este traza la ruta que cubra sus requerimientos, la cual es almacenada en la base de datos para su posterior consulta. Si la opción es usuario Pasajero, éste marca su ubicación en el mapa, y envía su solicitud al sistema para que el motor (algoritmo M-Tree) se ejecute y obtenga las rutas más similares a las requeridas por el usuario pasajero. Si éste selecciona una ruta puede visualizarla en la pantalla y decidir si es o no la mejor opción. En caso afirmativo se enviara una notificación vía correo electrónico al usuario Conductor de la ruta seleccionada. En caso contrario el usuario puede seleccionar otra ruta o salir del sistema.

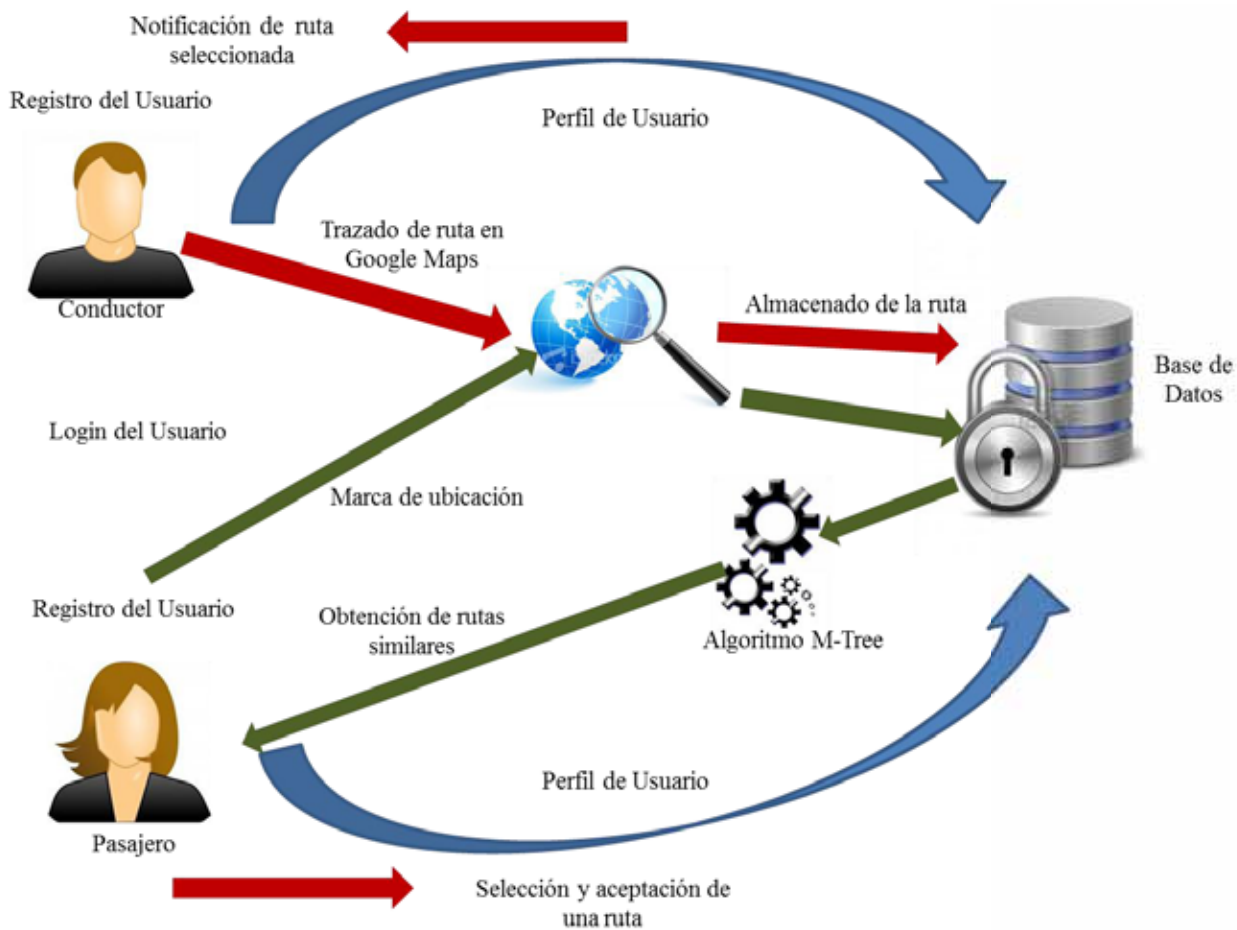


Figura 3.2- Diagrama de Funcionamiento del Sistema *Avent-UAM*.

3.1 Interfaz de Inicio del Sistema

La interfaz de inicio del sistema, proporciona dos opciones: registro y *login*.

Cuando el usuario no está registrado en el sistema, éste no podrá acceder al mismo; por tanto deberá registrarse en el sistema proporcionando los datos:

- Usuario
- Password
- Licenciatura
- Correo electrónico
- Observaciones

La información recabada será utilizada para tener la información de cada usuario almacenada en la base de datos y manipularla de acuerdo a las necesidades del sistema.

El módulo de Acceso al Sistema contiene las siguientes interfaces:

- **Index.html.** Interfaz gráfica inicial para acceso al sistema.
- **Login.html.** Interfaz gráfica en la cual se muestra una ventana donde el usuario podrá acceder o registrarse.
- **Login.php.** Introduce los datos del usuario, utilizando el programa `validar.php` para validar que la información sea correcta con la base de datos.
- **Validar.php.** El programa realiza la validación de los datos introducidos en la interfaz de login y la base de datos, si son correctos permite el acceso, de lo contrario lo deniega.
- **Conexión.php.** Se realiza la conexión a la Base de datos.
- **Registro.php.** Se registra y almacena la información en la base de datos.

El sistema accede a la siguiente tabla en la base de datos del sistema:

- `n260m_13554453_aventuam-> usuarios`

En la figura 3.3 observamos la secuencia que debe realizar el usuario al iniciar su sesión, es caso de tener ya registrado sus datos únicamente debe ingresar al sistema con su usuario y contraseña. En el caso de no estar en la base de datos es necesario dar de alta la información del nuevo usuario en el sistema.

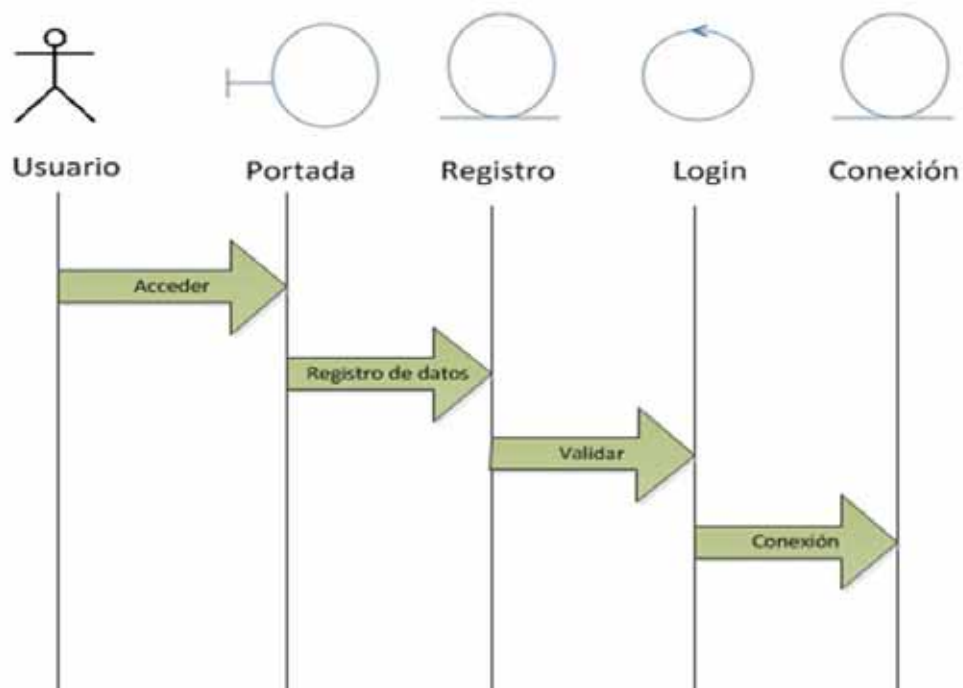


Figura 3.3.- Diagrama de Secuencia de Interfaz de Inicio del Sistema.

3.2 Selección del Tipo de Usuario

El módulo Tipo de Usuario, proporciona dos alternativas de usuario: conductor y pasajero. Proporciona una interfaz de selección la cual direcciona a la opción requerida. Hace seguimiento del usuario con la sesión activa y muestra el nombre del usuario.

El código que hace referencia a esta sección del sistema es:

➔ Tipo_usuario.php

El sistema accede a la siguiente tabla en la base de datos del sistema:

➔ n260m_13554453_aventuam-> tipo_usuario

3.2.1 Tipo de Usuario Conductor

Dentro del sistema existen dos opciones de usuario, usuario conductor (UC) y usuario pasajero (UP). El UC es aquel que cuenta con una interfaz “conductor.php” dentro de ésta se cuenta con el enlace al servidor, el acceso al servicio de rutas tiene lugar de forma asíncrona, la llamada se hace a un servidor externo por medio de la URL “<https://maps.googleapis.com/maps/api/js?v=3.exp&sensor=false>”.

El UC realiza el trazado de la ruta. Se crea un objeto de tipo *DirectionsService* y ejecuta *DirectionsService.route()* para iniciar una solicitud al servicio de las rutas, incluyendo un objeto literal *DirectionsRequest* que contiene los términos de entrada y un método de devolución de llamada que se ejecutará al recibir la respuesta. *DirectionsRenderer* controla la visualización de la polilínea¹ entre las ubicaciones indicadas, así como la colocación, si corresponde, de marcadores en el origen, en el destino y en los hitos (en nuestro proyecto obtendremos los hitos mediante el algoritmo *M-Tree*, ya que el servicio de *Google Maps* únicamente proporciona un máximo de 8 Hitos por cada ruta trazada.). Al enviar una solicitud de rutas a *DirectionsService*, se recibirá una respuesta que constará de un código de estado y un resultado, el cual es un objeto *DirectionsResult*. La función *calcRoute()* la cual generara el trazado de la ruta, esto con los valores proporcionados de origen y destino.

Se establece la propiedad *draggable* en *true* esta opción nos permite modificar en tiempo real la ruta optima mostrada por el servicio *Google Maps* en el caso de que el usuario desee personalizarla. Esta es una forma dinámica denominado objeto *DirectionsRenderer* en caso de rutas *arrastrables* (*draggable*: ruta que puede ser modificada por el usuario por medio de los hitos).

Una vez obtenida la ruta el UC guardara la ruta en la base de datos: *mapdir-mapdir*, en la cual se guardaran en el atributo *ruta* los puntos geográficos obtenidos del trazado (únicamente dos puntos inicio y fin). Ya que los puntos geográficos intermedios serán almacenados en la tabla *puntos*; esto es porque la ruta aproximada emitida por *Google Maps* únicamente proporciona dos puntos inicio y fin y de ahí partimos para obtener los puntos intermedios.

¹Una polilínea es un objeto en AutoCAD que consiste de uno o más segmentos de línea(rectas o arcos). Un rectángulo es un ejemplo de una polilínea con el que usted ya está familiarizado. <http://es.scribd.com/doc/17593519/POLILINEA>

El UC, puede terminar su sesión en cualquier momento, dando click en el botón “Salir de Sesión”. El sistema cuenta con la opción “Perfil de Usuario” esta permite que el usuario lea la información que introdujo al sistema al darse de alta.

Si su ruta ha sido seleccionada por un usuario pasajero, recibirá un correo electrónico informando la selección, así como información que el usuario pasajero personalice en el correo que crea conveniente para compartir el viaje.

Cabe mencionar que el inicio del viaje inicia en el Punto A y Termina en el Punto B

Contiene los programas:

- ➔ conductor.php
- ➔ perfil.html
- ➔ process.php
- ➔ logout.php

El sistema accede a las siguientes tablas en la base de datos del sistema:

- ➔ n260m_13554453_aventuam-> usuarios
- ➔ n260m_13554453_aventuam-> perfil
- ➔ n260m_13554453_aventuam-> mapdir
- ➔ n260m_13554453_aventuam-> puntos

Contiene las siguientes funciones del servicio Google Maps:

- ➔ function initialize(): Función de inicio.
- ➔ function calcRoute(): Calculo la ruta que el usuario especifico.
- ➔ function save_waypoints(): Almacena los puntos generados por la ruta
- ➔ DirectionsService(): Inicia una solicitud al servicio de rutas
- ➔ Geocoder(): Transforma direcciones en coordenadas geográficas.
- ➔ DirectionsRenderer(): Controla la visualización de la polilínea entre las ubicaciones indicadas, así como la colocación, si corresponde, de marcadores en el origen, en el destino y en los hitos.
- ➔ DirectionsResult(): Devolución de llamada de la solicitud del servicio.
- ➔ DirectionsRequest(): Contiene los términos de entrada y un método de devolución de llamada que se ejecutará al recibir la respuesta

La figura 3.4 muestra la secuencia que ejemplifica el proceso del usuario Conductor, el usuario conductor crea los elementos *directionsService* y envía la solicitud con los elementos y coordenadas necesarias para generar la ruta, de tal manera que mediante los *steps ways* se obtengan los puntos geográficos intermedios y toda la información sea almacenada en la base de datos.

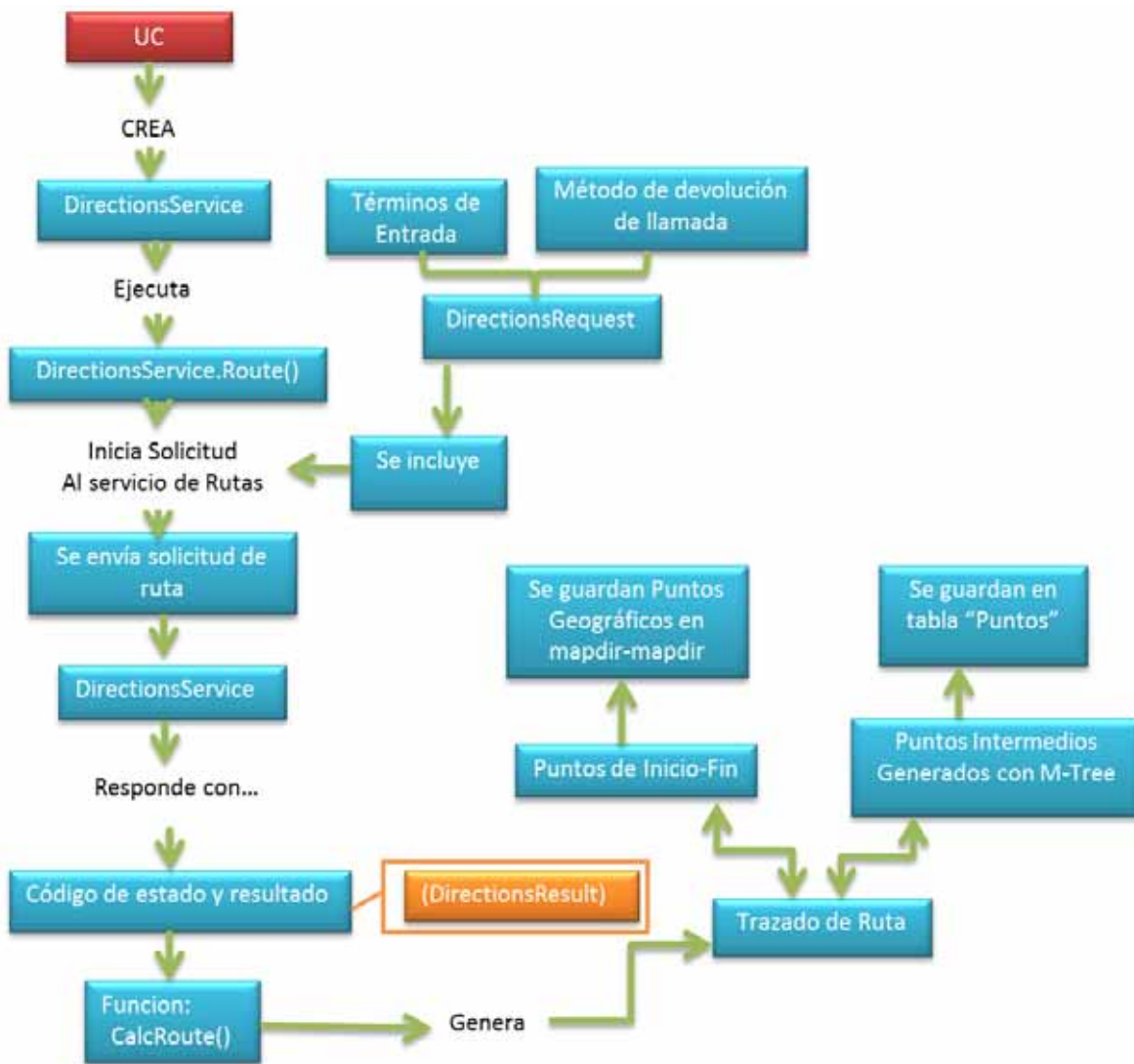


Figura 3.4 Diagrama Usuario Conductor.

3.2.2 Tipo de Usuario Pasajero

La otra opción disponible en el sistema es el usuario pasajero (UP). El UP es aquel que cuenta con una interfaz “pasajero.php” dentro de la interfaz se tiene el enlace al servidor, este acceso al servicio de rutas tiene lugar de forma asíncrona, esta llamada se hace a un servidor externo. Esto se lleva a cabo por medio del enlace a la dirección URL “<https://maps.googleapis.com/maps/api/js?v=3.exp&sensor=false>”.

El UP inserta una marca en el sitio del destino, por medio de la función de *google maps function place Marker(position, map)*. El sistema mostrara un marcador (el marcador en un globo rojo, el sistema solo permite una marca por sesión activa) en

el mapa mostrando la ubicación seleccionada por el usuario. El punto seleccionado por el usuario será almacenado en la base de datos con la función *function save_waypoints ()*. La marca o posición del UP será almacenada, en la base de datos para realizar la comparación entre los puntos geográficos con los puntos de la ruta trazada del UC. El usuario tiene la opción de eliminar la marca que posicionó en caso de que desee cambiar la de ubicación, utilizando la opción “Eliminar marca”.

Posteriormente, el algoritmo *M-Tree* se ejecutara al pulsar “Buscar Coincidencias”. El algoritmo realizara una búsqueda en el árbol que fue creado al momento de su ejecución tomando como nodo raíz la UAM Azcapotzalco, ya sea como origen o destino. El radio cobertor utilizado se ha programado de un punto geográfico a la raíz, esto nos arrojará la distancia que existe entre cada nodo (cada hito obtenido).

Como el algoritmo *M-Tree* trabaja con funciones de distancias obtendremos la mínima entre los nodos del árbol. Consiguiendo esas distancias, el algoritmo determina mediante el radio cobertor cuál es el nodo más similar al nodo de comparación (en este caso la ubicación que el usuario eligió). Esto se realiza para cada una de las rutas trazadas almacenadas en la base de datos, se hace por medio de un ciclo recorriendo cada una de las rutas almacenadas, obteniendo la más similar a la ubicación establecida por el usuario pasajero.

El sistema mostrara una alerta en caso de no existir coincidencias especificando la negatividad. De lo contrario muestra las rutas coincidentes con la ubicación establecida por el UP, el detalle de las opciones aparece en la parte inferior central de la ventana del “Pasajero”, observamos las coincidencias encontradas (empezando por coincidencia más similar).

El UP podrá hacer la selección de la ruta de su conveniencia dando un click en la que desee. Al dar click en el link se mostrara la ruta seleccionada, se puede decidir si es o no factible, en caso de no ser factible se puede regresar y seleccionar otra de la lista. Al seleccionar la ruta el usuario dará click en “Aceptar” el sistema mandara una notificación por medio de un correo electrónico, el cual podrá ser personalizado por el UP, indicando cualquier información detallada que este crea pertinente: información de contacto, nombre completo, teléfono, información relevante para los usuarios. En la ventana donde se esta la ruta se mostrara el perfil del usuario conductor, esta información servirá al usuario para conocer más a la persona con la que podría compartir el viaje.

Cabe mencionar que el inicio del viaje inicia en el Punto A y termina en el Punto B.

El UP, puede terminar su sesión en el momento que desee, únicamente dando click en el botón “Salir de Sesión”.

Contiene los programas:

- ➔ pasajero.php
- ➔ perfil.html
- ➔ process.php
- ➔ muestraruta.php
- ➔ logout.php
- ➔ json2.js² archivo de secuencia Script
- ➔ jsmysql
 - mtree
 - mysqlws³
 - mysqlwslib⁴

El sistema accede a las siguientes tablas en la base de datos del sistema:

- ➔ n260m_13554453_aventuam-> usuarios
- ➔ n260m_13554453_aventuam-> perfil
- ➔ n260m_13554453_aventuam-> mapdir
- ➔ n260m_13554453_aventuam-> puntos

Contiene las siguientes funciones del servicio *Google Maps*:

- ➔ function initialize() : Función de inicio
- ➔ function save_waypoints(): Almacena los puntos geográficos en la base de datos.
- ➔ function placeMarker(position, map): posiciona una marca de identificación.
- ➔ function setAllMap(map): Establece todos los marcadores que el usuario necesite.
- ➔ function clearOverlays(): Oculta el marcador especificado por el usuario
- ➔ function showOverlays(): Muestra el marcador especificado por el usuario
- ➔ function deleteOverlays(): Permite eliminar el marcador posicionado por el usuario, esto le permite en caso de no posicionar correctamente el marcador pueda realizar nuevamente el marcado.

²Json2 es un conjunto de la notación literal de objetos de JavaScript <https://github.com/douglascrockford/JSON-js>

³mysqlws Librería para acceder desde Javascript a la B.D. mysql (Utilizando un programa php en el servidor) <https://github.com/shiguera/mysqlws>

⁴mysqlw slib Forma parte del paquete de acceso a mysql junto con la librería mysqlwslib.js, <http://www.bahai.org.uy/AppTryouts/js/mysqlws.php>

La figura 3.5 se observa al usuario pasajero su punto destino o inicio en el mapa, el cual es almacenado en la base de datos. El sistema mediante la ejecución del Algoritmo *M-Tree* obtiene la mejor opción para el usuario pasajero, el cual al seleccionar alguna de las opciones exhibidas por el sistema envía mediante un correo electrónico la notificación al usuario conductor sobre su elección.

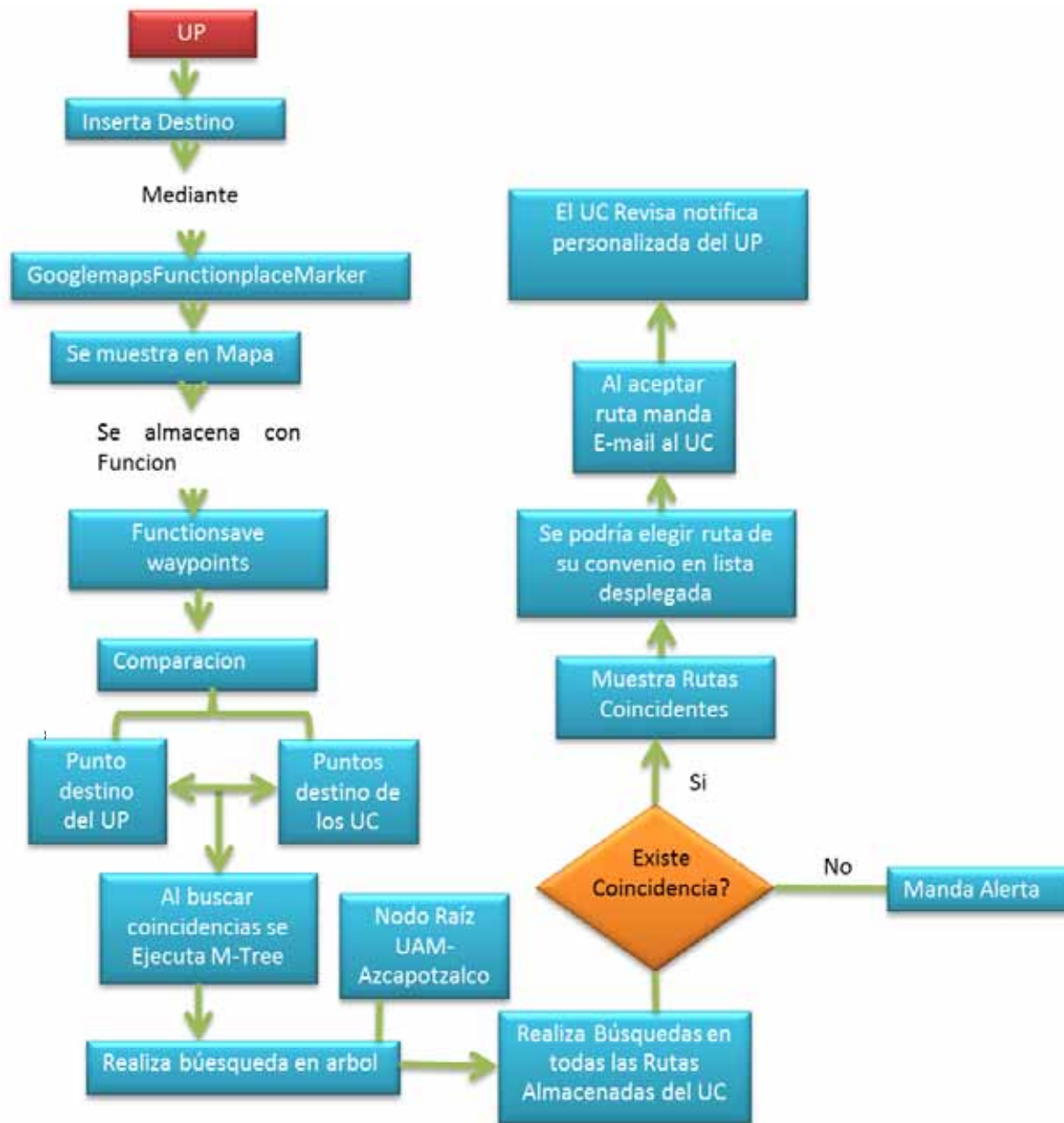


Figura 3.5 Etapas del Proceso del Usuario Conductor.

3.2.3 Perfil del Usuario

Permite visualizar la información del usuario. Muestra las características del usuario, nombre, licenciatura, datos del mismo. Esto para que los usuarios tengan conocimiento de la persona con la que realizaran el viaje, lo cual el usuario podrá ingresar al momento de su registro en el sistema.

El módulo del Perfil de Usuario contiene la información general del usuario que éste proporcionó al darse de alta:

- Nombre
- Licenciatura
- Observaciones: preferencias del usuario

En la sección de Observaciones el usuario puede incluir sus preferencias de viaje, y como sabemos existen diferentes horarios en la universidad, el usuario puede contactarse vía correo electrónico con los pasajeros y proporcionar sus diferentes horarios, eso permite mayor flexibilidad a cada una de las rutas trazadas.

En esta sección el usuario no podrá hacer modificaciones, ya que los resultados serán automáticamente proporcionados por la Base de Datos.

Contiene los programas:

- perfil.html
- logout.php

El sistema accede a las siguientes tablas en la base de datos del sistema:

- n260m_13554453_aventuam-> usuarios
- n260m_13554453_aventuam-> perfil
- n260m_13554453_aventuam-> mapdir

3.3 Implementación del Algoritmo *M-tree*

El algoritmo *M-Tree* será utilizado en el proceso cuando el usuario conductor trace su ruta obteniendo los puntos geográficos intermedios que se arrojaran como resultado de la petición al servicio; ya que *Google Maps* únicamente obtiene la ruta aproximada esto es, proporciona los puntos geográficos de origen y destino y estos no son suficientes para el logro del objetivo. Obtenidos los puntos intermedios el sistema *Avent-Uam* buscara por similitud las posibles rutas que coincidan con el punto geográfico especificado por el usuario pasajero, esto se llevara a cabo por medio del algoritmo *M-tree*, con las modificaciones adecuadas. Los puntos intermedios de cada ruta se obtienen del servicio de *Rutas* de *Google*, se solicita las indicaciones y obtenemos los puntos de cada indicación, esto se hace para conocer en qué punto o puntos se desvía la ruta de las calles, ya que los puntos *dragables* no son suficientes, cada punto es una marca de indicación.

Para la estructura del final del sistema, es necesaria la integración de cada uno de los componentes, la interoperabilidad se realiza por medio de los archivos que contiene el código para ello.

jsmsql: Dentro de esta carpeta se encuentra el archivo de los códigos *Mtree.js*, *Mysqlws.php* y *Mysqlwslib.js* los cuales se describen a continuación.

➤ *Mtree.js*.

Aquí se utilizan dos clases, una es *Mtree2Nodo* y la otra es el *Mtree2*. La clase *Mtree2Nodo* representa cada punto de una ruta en coordenadas x, y que corresponden a la latitud y longitud respectivamente. Sus atributos son:

- ➔ *parentDistance*. Distancia del punto a la raíz
- ➔ *puntoX*. Representa la latitud de un punto geográfico.
- ➔ *puntoY*. Representa la longitud de un punto geográfico.

Sus métodos son:

- ➔ *distanciaRaiz*. Establece su distancia con respecto a la raíz, la UAM-A.
- ➔ *distanciaNodo*. Regresa la distancia a otro nodo de esta misma clase la cual es una distancia euclidiana, un punto de la ruta a un punto externo.
- ➔ *distanciaPunto*. Regresa la distancia a un punto geográfico.

La clase *Mtree* se encarga de crear un árbol para representar los puntos que componen una ruta, los puntos son insertados en orden y se calcula su distancia al nodo raíz. Cada ruta proporciona la distancia mínima a cualquier punto con el que se compare.

Sus atributos son:

- ➔ raizX. Latitud de la raíz.
- ➔ raizY. Longitud de la raíz.
- ➔ entries. Arreglo de nodos.
- ➔ tamaño. Número de nodos.

Sus métodos son:

- ➔ setRaiz. Establece el punto raíz.
- ➔ distanciaRaiz. Calcula y establece la distancia de cada nodo a la raíz.
- ➔ distMin. Regresa la distancia mínima de la ruta con un punto proporcionado

Para incluir como librería se cambia por "miruta" la ubicación del archivo *mtree.js*.

En el archivo, *mysqlws.php*, hay que configurar las variables:

\$host, \$user, \$pw, \$db

Las variables se configuraron donde *\$host* representa la dirección URL en donde se encuentra hospedada, *\$user* es el nombre del administrador, *\$pw* es el password y *\$db* es la base de datos donde se encuentran las tablas.

Para realizar la interoperabilidad en el sistema es necesario configurar los siguientes archivos, ya que al trabajar con diferentes lenguajes es necesario utilizar aplicaciones para que la integración se lleve adecuadamente.

Los siguientes dos archivos *Mysqlwslib* y *Mysqlws.php*.

- *Mysqlwslib.js* [14]. Forma parte del paquete de acceso a *mysql* junto con el archivo *mysqlws.php*, para utilizarlo se deben instalar en el servidor. En el

archivo *mysqlwslib.js* hay que configurar la ruta correcta para que encuentre este archivo y la base de datos por defecto. Para poder utilizar la librería desde nuestra página web habrá que poner un enlace en el fichero 'html':

```
<script src="miruta/mysqlwslib.js"></script>
```

- *Mysqlws.php* [14]. Forma parte del paquete de acceso a *mysql* junto con la librería *mysqlwslib*. se deben instalar en el servidor de igual manera. Ayuda a hacer consultas en *JavaScript*, a través de *PHP* enviando parámetros y recibiendo resultados en forma de arreglos.

3.4 Estructura de la Bases de Datos

La Base de Datos guarda información relevante de:

- ➔ **Usuario.** Presenta la información correspondiente a cada uno de los usuarios registrados en el servicio. Sus atributos son: *id(int)*, *usuario(varchar)*, *mail(varchar)*, *password(Blob)*, *licenciatura(varchar)* y *observaciones(varchar)*.
- ➔ **Tipo_usuario.** Permite seleccionar entre ser pasajero o conductor y guardar en la base de datos, esto para que en un futuro pueda ser seleccionada por algún usuario pasajero. Sus atributos son: *id(int)*, *usuario(varchar)*, *conductor(int)*, *pasajero(int)*.
- ➔ **Perfil.** Permite guardar información del usuario, sus atributos son *id(int)*, *calificación(varchar)* y *observaciones(varchar)*.
- ➔ **Map_dir.** Contiene la información de las rutas trazadas por los usuarios. Sus atributos son *id(int)*, *usuario(varchar)* y *ruta(varchar)*.
- ➔ **Ruta_pasajero.** Permite guardar la ubicación marcada por el usuario. Sus atributos son: *id(int)*, *usuario(varchar)*, *ubicación(varchar)*.
- ➔ **Rutas.** Presenta la información de las rutas que son concretas por los usuarios conductores, sus atributos son: *id*, *usuario*, *concretadas*.
- ➔ **Puntos.** Son los puntos geográficos almacenados de las rutas trazadas por los conductores, atributos: *id(int)*, *usuario(varchar)*, *puntos(varchar)*.

La Figura 3.4 muestra el diagrama Entidad-Relación del Sistema Avent-UAM; la correspondencia y la pertenencia de cada una de sus tablas, así como sus atributos.

Podemos observar en el diagrama la independencia lógica y física de los datos, existe una redundancia mínima para disminuir espacio de almacenamiento. La

integridad de datos es importante para que no se corrompa la información. Acceder a través de diferentes lenguajes de programación estándar, ya que nuestro sistema así lo requiere. Se realizaron en la ejecución del programa al analizar y verificar que los datos guardados en la base de datos correspondieran a lo que se esperaba almacenar, además de que cada usuario conductor tenga almacenada una única ruta, aun cuando esta sea modificada varias veces.

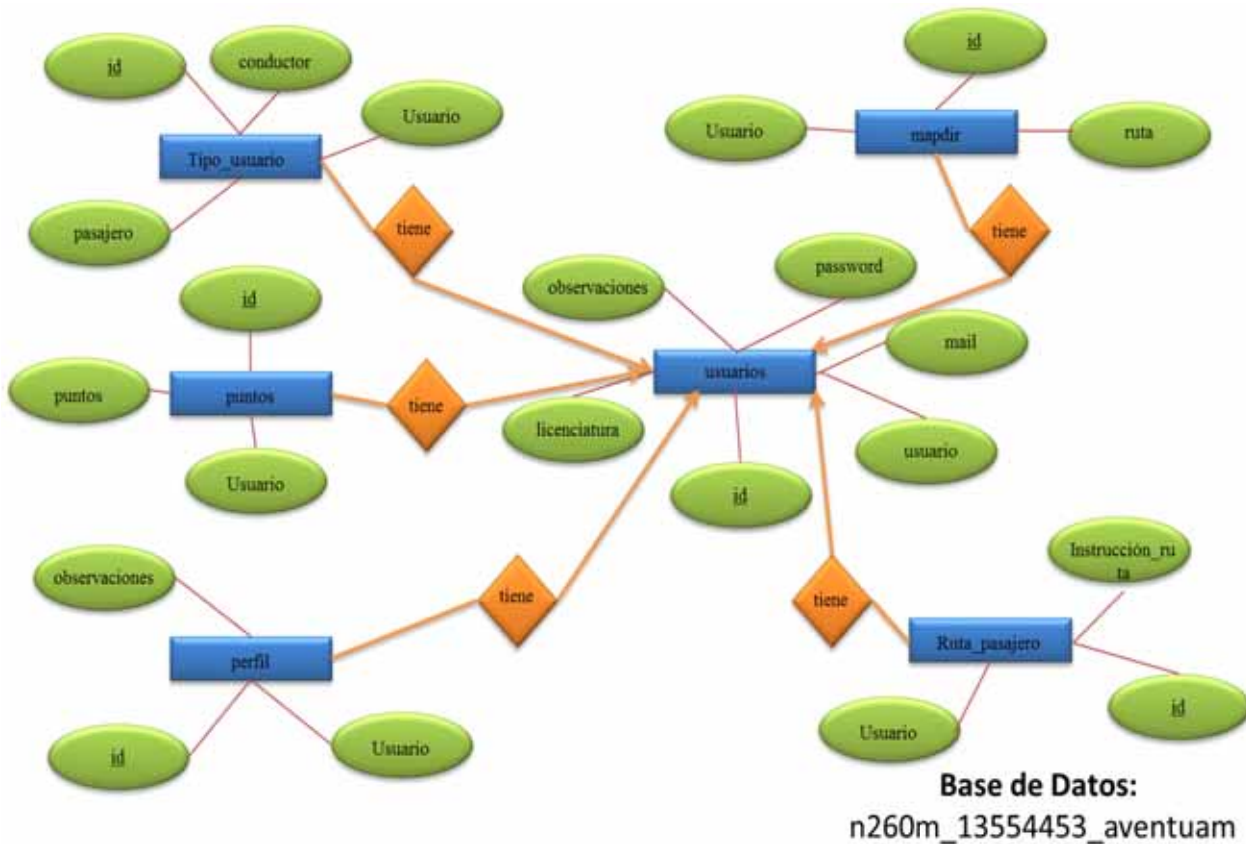


Figura 3.4–Diagrama Entidad-Relación del Sistema *Avent-UAM*.

3.5 Notificaciones

Permite por medio de correo electrónico notificar al usuario conductor que existe un usuario pasajero que desea compartir viaje con él. El usuario debe tener configurado el servicio de *Microsoft Outlook 2010*⁵ con anticipación para que esta función trabaje correctamente. (Anexo A: Manual de Administración).

⁵Microsoft Outlook es un programa de organización ofimática y cliente de correo electrónico de Microsoft, y forma parte de la *suite Microsoft Office*. http://es.wikipedia.org/wiki/Microsoft_Outlook

- función `mandaCorreo()`
- `muestraruta.php`: Muestra las rutas de los conductores recomendadas al pasajero.

Como funciones adicionales a cada una de las interfaces de los usuarios tenemos: *view Street*, *giro de imagen 45°*, *zoom del mapa*. Todas estas características son funciones de *Google Maps*.

Detallando cada uno de los módulos que comprenden el sistema, observando la complejidad en la implementación del algoritmo *M-Tree*, así como la adecuación de sus características ya que al ser un algoritmo robusto únicamente se consideró la parte que correspondiente a distancias, herramienta que fue utilizada en este servicio. Al tomar como referencias distancias dentro del desarrollo; El algoritmo *M-Tree* es muy utilizado para el tratamiento de imágenes ayudando a encontrar similitudes entre ellas al analizar sus diferentes características. En este proyecto se ocupó el mismo principio pero únicamente utilizado solo una característica, distancias euclidianas, de esta manera no ocupamos nada relacionado con tratamiento de imágenes y sus características y solo aprovechamos la función de comparar objetos más sencillos, que en este caso son rutas formadas por puntos.

Considero que la API de *Google Maps* fue un poco más sencilla de configurar y adecuar al sistema, ya que cuenta con funciones prediseñadas que aportan una mayor familiaridad en el uso de la API y esto conlleva a trabajar con resultados satisfactorias.

La base de datos se adecuo para obtener la información necesaria optimizando recursos, siendo una de estas las partes menos complejas.

La complejidad mayúscula radicó en la interoperabilidad de los diferentes tipos de lenguajes y servicios utilizados, y en obtener la interacción idónea para el funcionamiento correcto del sistema.

En el capítulo 4 se observan las pruebas realizadas y la ejecución de cada uno de los módulos que comprenden el Sistema *Avent-UAM*.

Capítulo 4.-Pruebas al Sistema *Avent-UAM*

4.1 Interfaz y Administración

La metodología descrita en el capítulo 3 detalla cada uno de los módulos que comprenden el Sistema *Avent-UAM*, a continuación se muestra el funcionamiento de cada uno de ellos.

Para la implementación de los módulos 2 y 3 se utilizó:

- MySQL *Versión 5.5.28*[15]
- JavaScript *Versión 1.1.0.4*[16]
- Java *Versión 6*[17]
- HTML *Versión 5*[18]
- PHP *Versión 5.3.5*[19]
- Apache *Versión 2.2.3*[20]
- phpAdmin *Versión 3.3.14*[21]
- *FileZilla* *Version 3.7.4.1* [22]

La información siguiente es proporcionada para la administración tanto del servidor como de la base de datos en que se aloja la información del sistema.

- *Panel de Control username: n260m_13554453*
- *Panel de Control password: Esten0*
- *Control panel URL: http://cpanel.260mb.net*
- *MySQL username: n260m_13554453*
- *MySQL password: Esten0*
- *MySQL hostname: sql310.260mb.net*
- *FTP username: n260m_13554453*
- *FTP password: Esten0*
- *FTP host name: 185.27.134.8*
- *Website URL: http://aventuam.260mb.net*

Se utiliza el programa *FileZilla* para administrar la base de datos del Sistema *Avent-UAM*.

El acceso al sistema es mediante el link:

➔ Página web: <http://aventuam.260mb.net/>

En la figura 4.1 podemos observar la interfaz del servidor del sistema *Avent-UAM*, Dentro del programa *FileZilla* podemos llevar a cabo actualizaciones al sistema. (Anexo A: Manual de Administración).

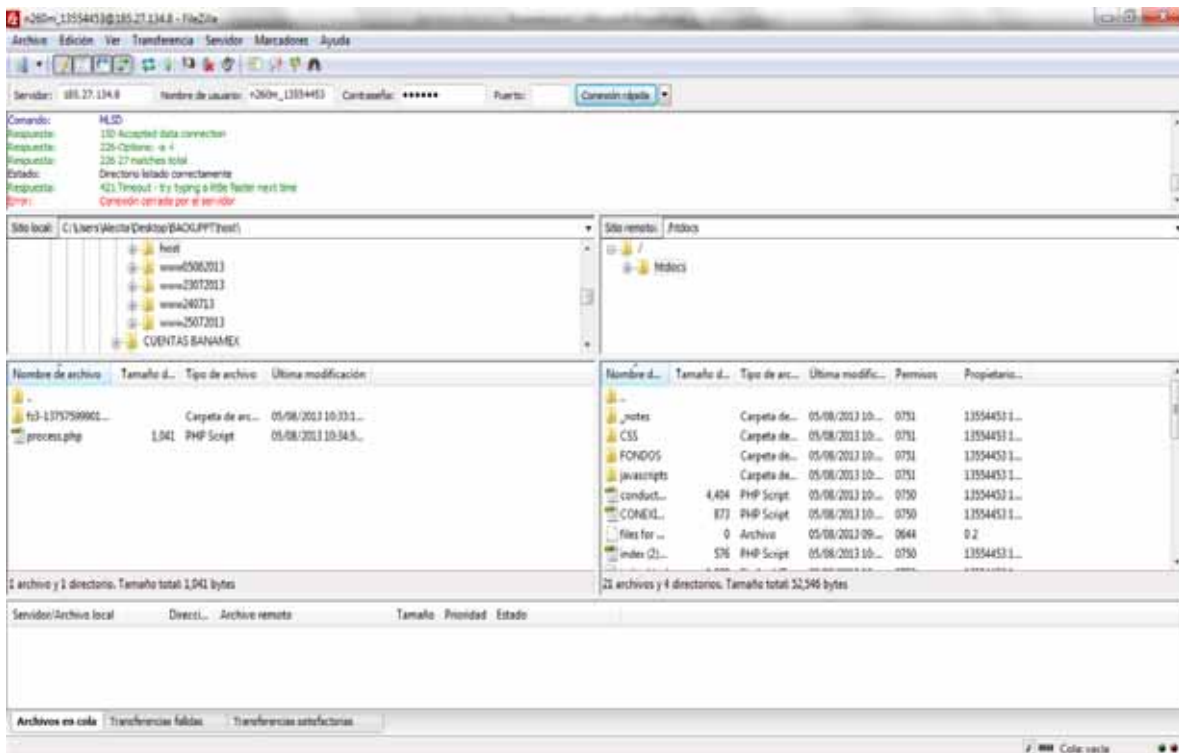


Figura 4.1- Interfaz para la administración del Servidor del Sistema *Avent-UAM*.

La figura 4.2 muestra la interfaz para la administración de la base de datos del Sistema *Avent-UAM*. Podemos hacer las modificaciones que sean necesarias a la base de datos y no es necesario almacenarlo dentro de un servidor físico, esta información está alojada en un host gratuito y podemos hacerlo desde cualquier equipo de cómputo con una conexión a internet. (Anexo A: Manual de Administración).

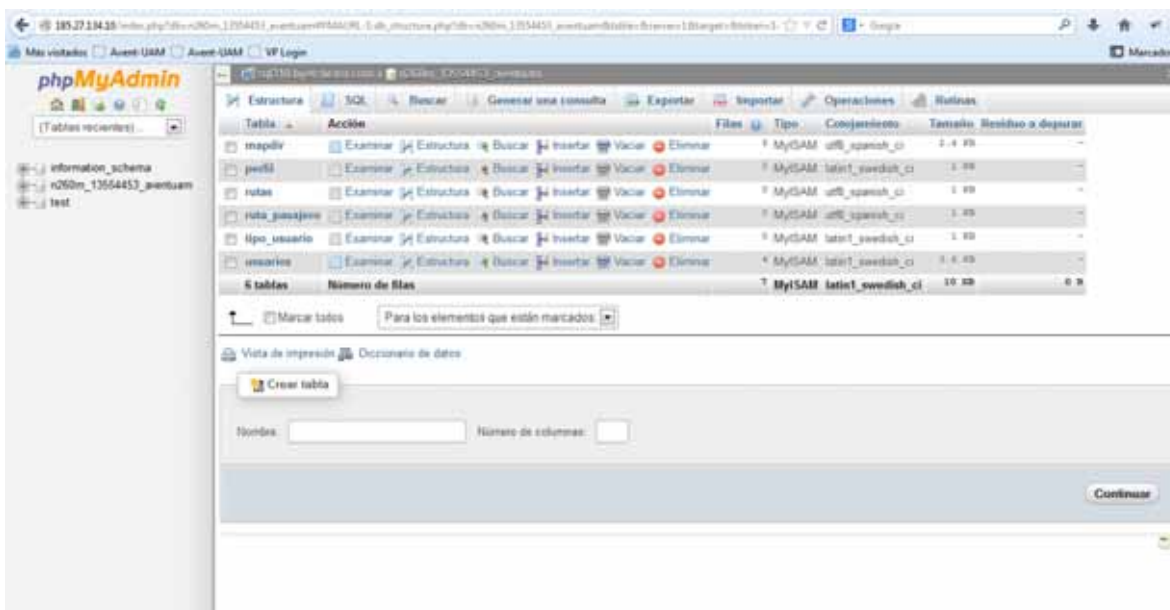


Figura 4.2- Interfaz para la administración de la base de datos del Sistema *Avent-UAM*.

La Pantalla de inicio que nos muestra la figura 4.3 realiza el re-direccionamiento a la pantalla de “Inicio de Sesión”. Dando *click* en “Entrar a *Avent-UAM*” este nos llevara a la primera parte del sistema.



Figura 4.3- Pantalla de Inicio del Sistema *Avent-UAM*.

La figura 4.4 muestra la pantalla de “Inicio de Sesión”, se debe introducir “Usuario” y “Password” en caso de que el usuario esté dado de alta en el sistema.



Figura 4.4- Pantalla de Inicio de Sesión del Sistema *Avent-UAM*.

La figura 4.5 muestra la pantalla de “Alta de Usuario” en el cual el usuario debe introducir los datos solicitados: Usuario, Password, Licenciatura, Correo Electrónico, Observaciones.

Al realizar el alta de usuario, regresamos a la ventana de inicio de sesión y será posible acceder al sistema.

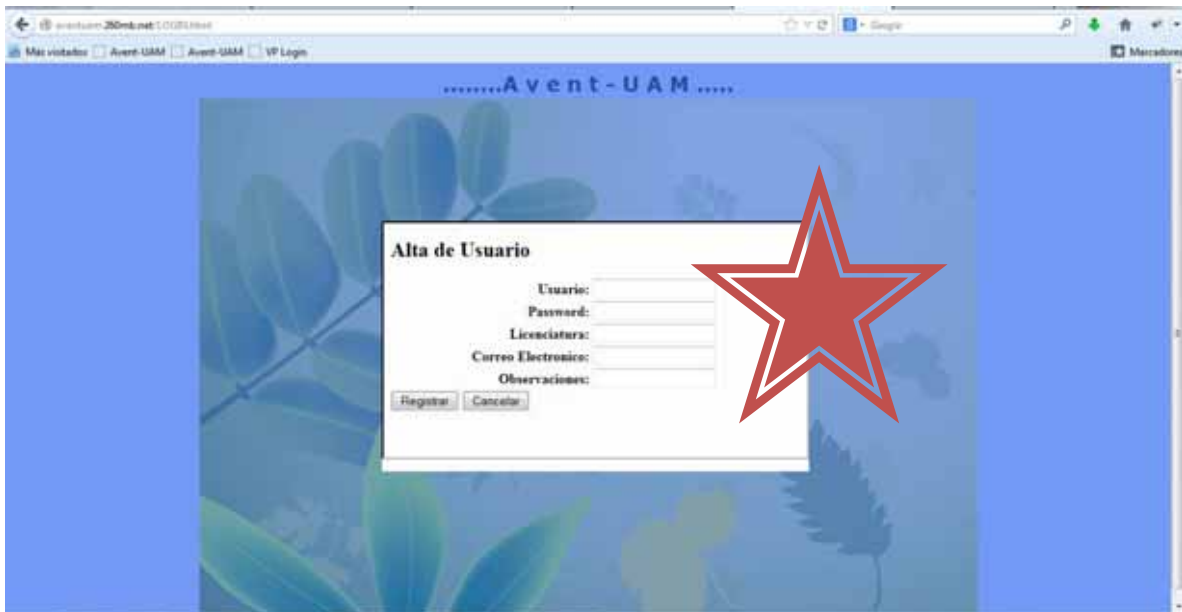


Figura 4.5- Pantalla “Alta de Usuario”, Registro de Usuarios del Sistema *Avent-UAM*.

En la figura 4.6 se nos indica que después de iniciar sesión el usuario podrá elegir entre dos opciones: ser usuario “Conductor” o “Pasajero”. Podemos observar que el usuario registrado se muestra informando que seguimos con sesión activa marcada con la flecha verde.



Figura 4.6.- Pantalla de Tipo de Usuario del Sistema *Avent-UAM*.

4.2 Usuario Conductor

En la figura 4.7 se muestra la ventana de usuario “Conductor” (UC), donde se deben proporcionar las direcciones de inicio y fin, posteriormente dar *click* en “Trazar ruta”, para que el sistema trace la ruta óptima que otorga el servicio de *Google Maps*.

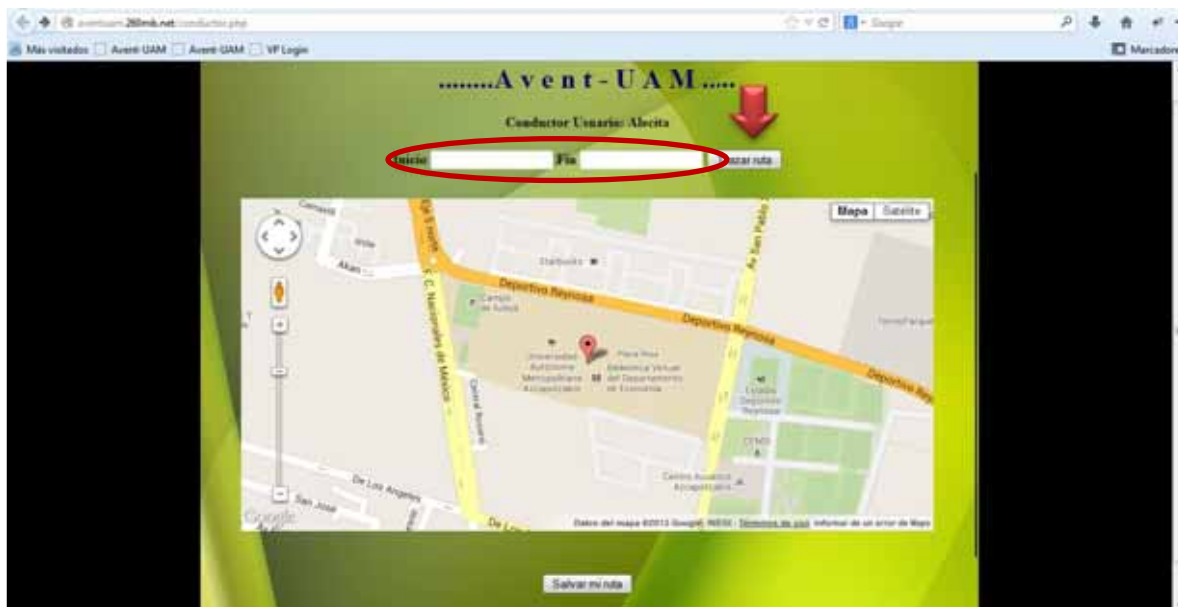


Figura 4.7- Interfaz de Usuario Conductor del Sistema *Avent-UAM*.

El UC traza su ruta, lo cual podemos observar en la figura 4.8 dando dos valores uno de “Inicio” y otro de “Fin”. Con estos parámetros se puede realizar el trazado de la ruta, lo cual se mencionó en la figura anterior, también se cuenta con una característica que ayuda al usuario a personalizar su ruta, en el caso que la ruta aproximada emitida por el servicio *Google Maps* no sea la mejor opción, el usuario puede por medio del ratón arrastrar la ruta y ubicarla en los puntos geográficos que sean más precisos a la ruta que él desea.

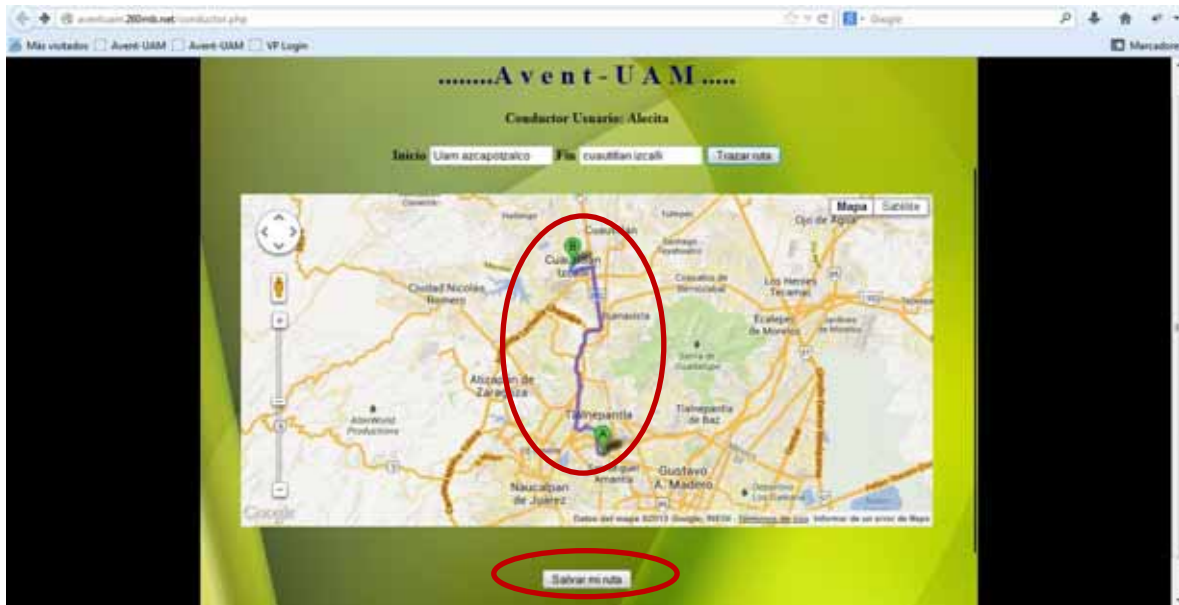


Figura 4.8- Interfaz de usuario “Conductor” del Sistema *Avent-UAM*, trazado de ruta.

En la figura 4.9 observamos que después de que el UC trace su ruta éste puede guardarla, en la pantalla se muestra que está ha sido guardada con éxito.

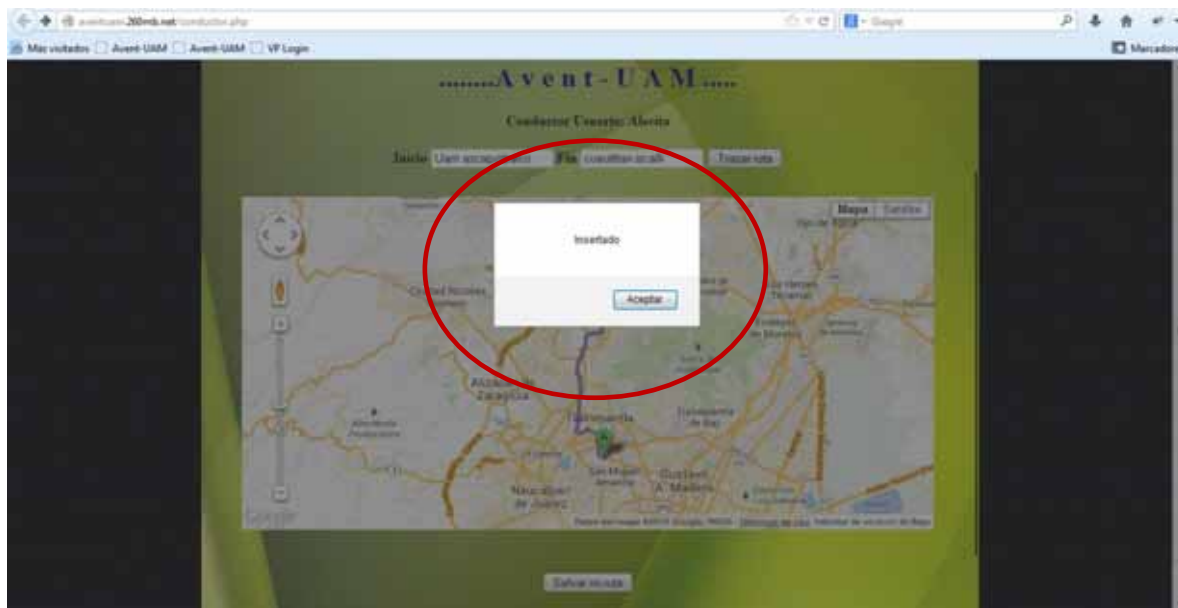


Figura 4.9- Interfaz de Usuario Conductor del Sistema *Avent-UAM*. Salvar Ruta.

4.3 Usuario Pasajero

La figura 4.10 muestra la interfaz del usuario “Pasajero” (UP), ésta interfaz cuenta con tres opciones, la primera “Eliminar Marca”, se utiliza cuando la marca se posicionó de manera incorrecta o el usuario prefiere otra ubicación, la segunda opción “Buscar Coincidencias”, donde el sistema ejecuta el algoritmo en proporciona una lista de rutas similares y la tercera opción “Regresar” cuando deseamos seleccionar otra opción.

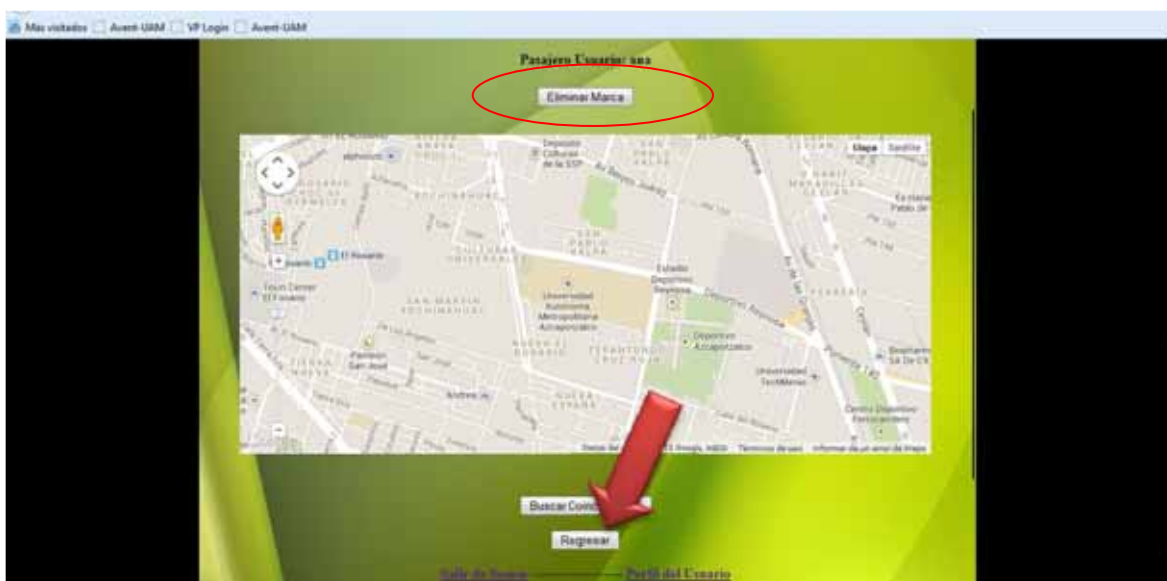


Figura 4.10- Interfaz de usuario “Pasajero” del Sistema *Avent-UAM*.

En la figura 4.11 el UP busca un punto geográfico que es la dirección a la que desea dirigirse el UP, la búsqueda se hace mediante el arrastre del mapa y posiciona la marca (globo rojo) cuando la elección cumpla los requerimientos personales de cada usuario. Pulsar opción “Buscar Coincidencias”, ésta opción mostrara las coincidencias encontradas por el sistema, desglosa una lista de las rutas similares con el punto geográfico establecido por el UP. Este listado está ordenado de la ruta más a la menos similar.

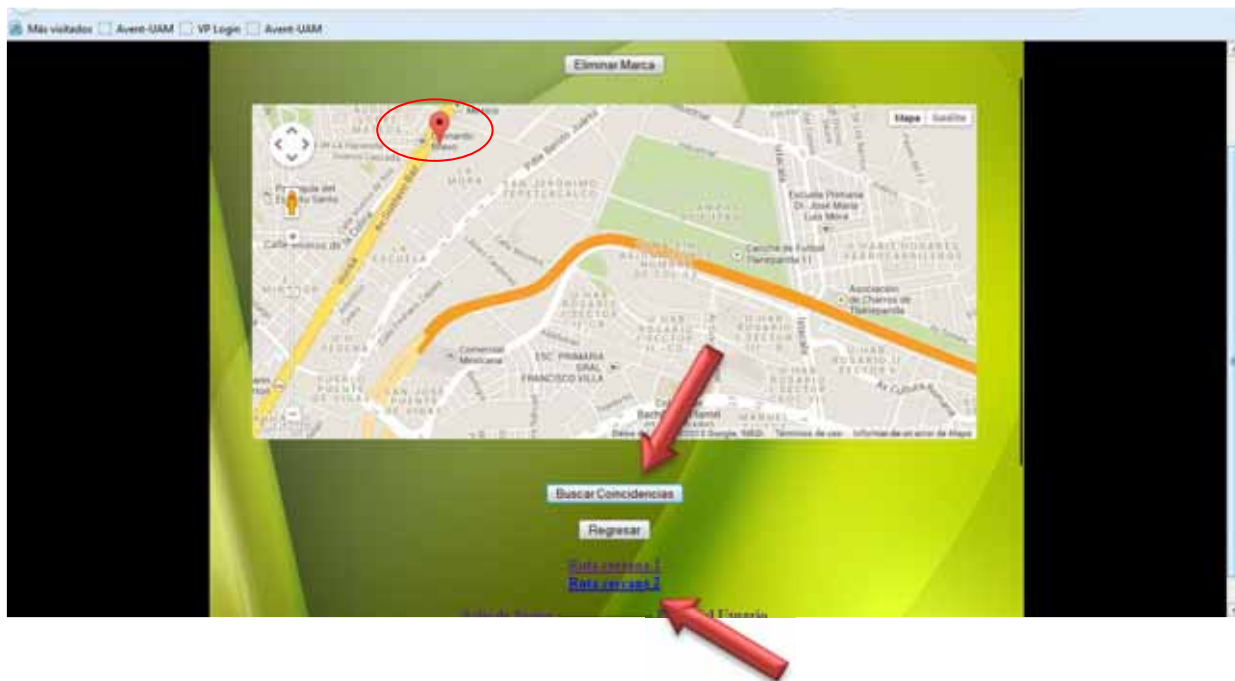


Figura 4.11- Interfaz de UP del Sistema *Avent-UAM*, marca de posicionada en la ubicación seleccionada por el UP.

La figura 4.12 muestra la ruta seleccionada por parte del usuario pasajero, la ruta puede o no cumplir con los requerimientos del UP, en caso afirmativo se pulsa en la opción y el sistema muestra en la pantalla la ruta. En el caso opuesto el usuario puede pulsar en la opción “Regresar” y escoger otra ruta.

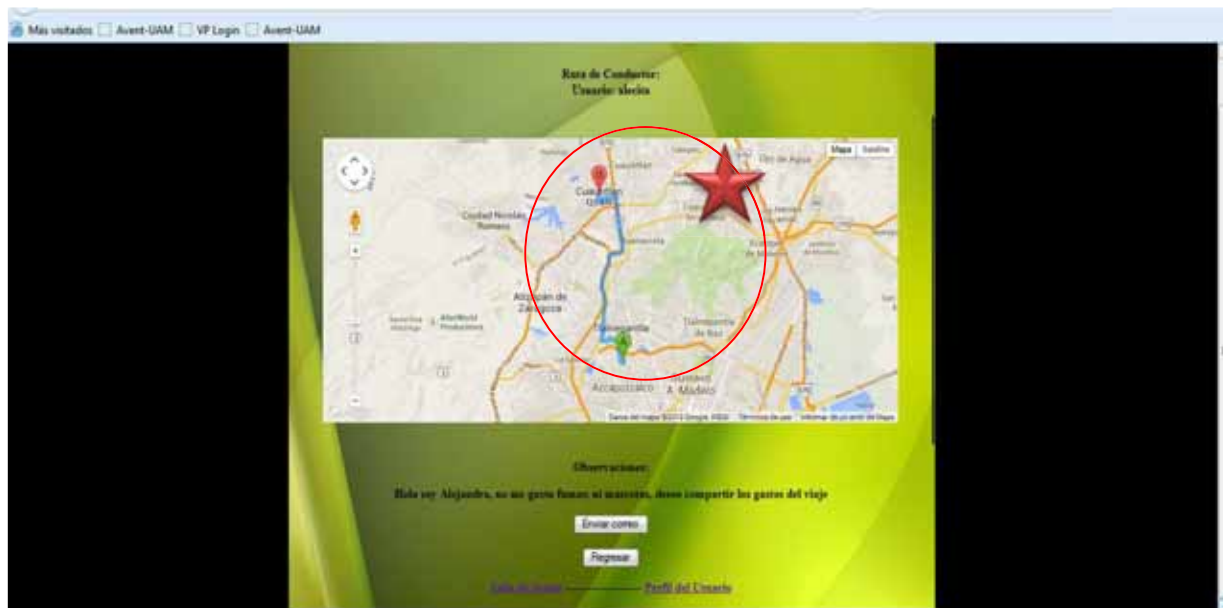


Figura 4.12- Ruta seleccionada.

En la figura 4.13, al seleccionar la opción correcta el sistema muestra una alerta en la cual se indica que la ruta ha sido aceptada y posteriormente se envía un correo electrónico al usuario conductor que trazó la ruta. Para poder realizar el envío del correo debemos tener configurado previamente el programa *Microsoft Outlook 2010* (Anexo B. Manual de instalación y configuración de Microsoft Outlook 2010) de lo contrario no se podrá hacer uso de esta opción. Podemos utilizar como lo indica la ventana entre el sistema de correo por default o si se desea escoger entre las alternativas que se muestran.

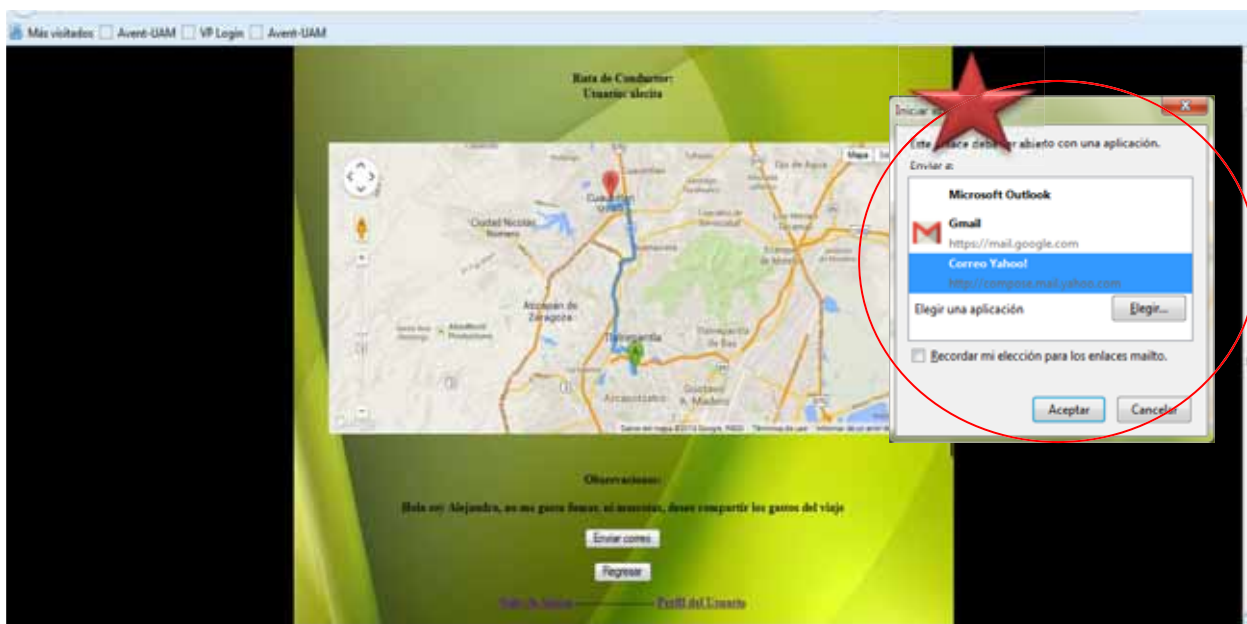


Figura 4.13- Envío de correo cuando la ruta ha sido aceptada.

4.4 Ejemplos

Ejemplo 1

El usuario Efraín, al tener auto propio será considerado usuario conductor, su ruta es de la UAM-Azcapotzalco a Teotihuacán de Arista y su recorrido es diario. La figura 4.14 muestra la ruta trazada.



Figura 4.14- Punto de inicio y fin de la ruta de Efraín.

Un usuario de nombre Alecita, al no contar con un automóvil propio se catalogará como usuario pasajero y ella desea aprovechar el viaje de un usuario conductor que tenga alguna ruta similar de su trayectoria UAM-Azcapotzalco hacia los Héroes Tecámac. El usuario pasajero coloca su marca en la posición adecuada a sus necesidades. En la figura 4.15 se observa la marca posicionada por el usuario Alecita.

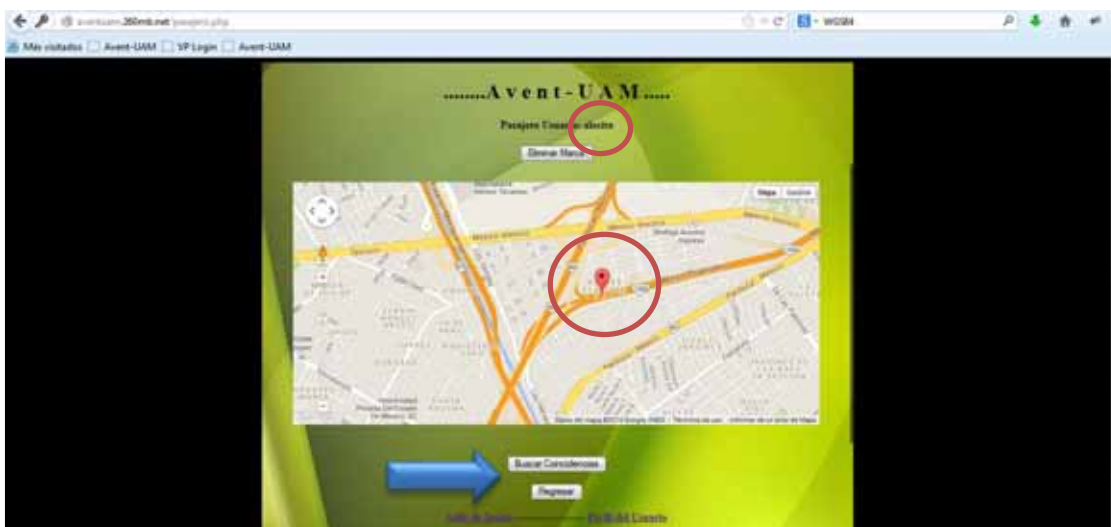


Figura 4.15- Posicionamiento del destino.

La figura 4.16 muestra las diferentes opciones que el usuario puede seleccionar iniciando con la más similar en primer lugar posterior aquellas rutas que fueron trazadas por otros usuarios pero que los puntos de ubicación son menos similares



Figura 4.16- Opciones mostradas por el sistema.

La figura 4.17 se muestra la ruta que el sistema obtuvo de la búsqueda (primera opción obtenida por el sistema), la cual encuentra un punto similar (marcado con negro) que corresponde al usuario Efraín.

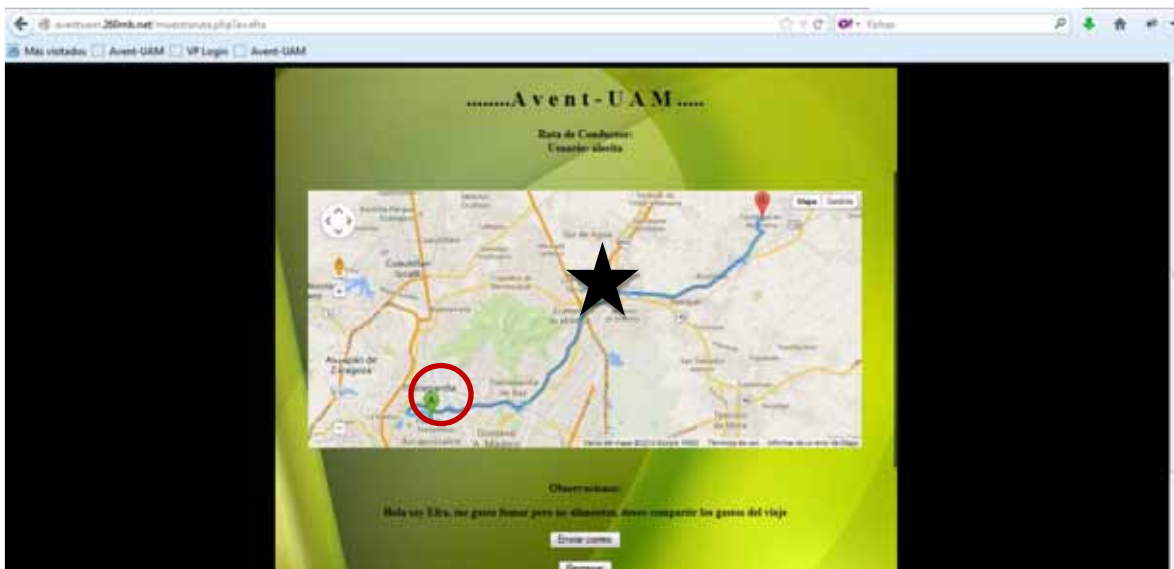


Figura 4.17- Mejor opción obtenida y seleccionada por el usuario.

Ejemplo 2

El usuario Alex cuenta con automóvil propio lo consideramos usuario conductor, ya que trazo una ruta de Plaza Galerías Toluca Lerma a UAM-Azcapotzalco, el cual es su recorrido lunes, miércoles y viernes, ver figura 4.18.

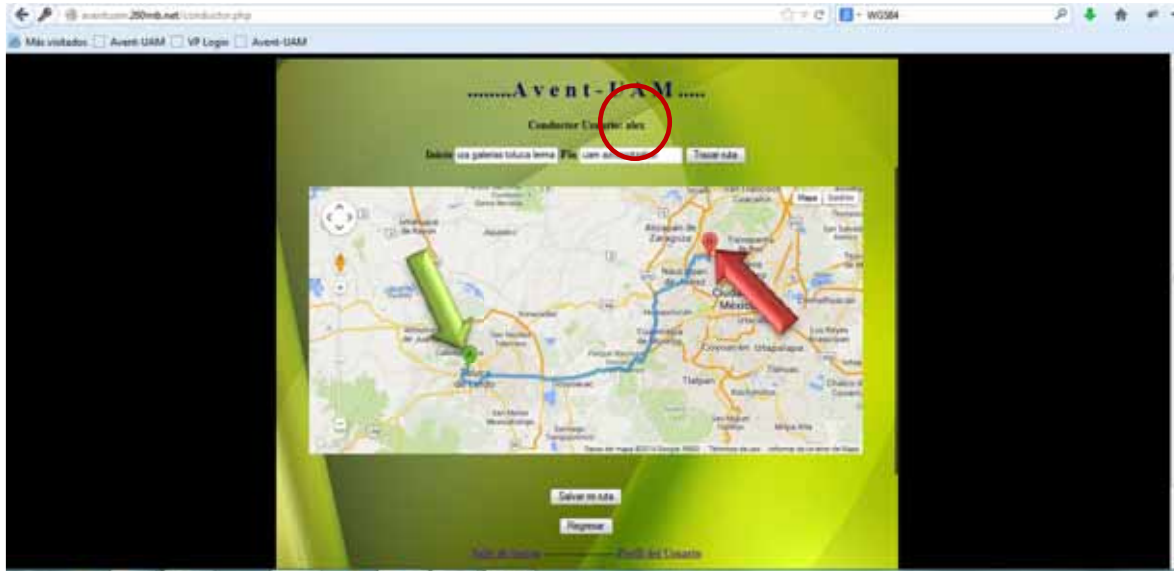


Figura 4.18- Punto de inicio y fin de la ruta de Alex.

Un usuario de nombre Ale al no tener transporte personal se cataloga como usuario pasajero y desea aprovechar el viaje de un usuario conductor que tenga alguna ruta similar de su trayectoria UAM-Azcapotzalco hacia El Capulín. El usuario pasajero coloca su marca en la posición adecuada a sus necesidades. En la figura 4.19 se observa la marca posicionada por el usuario Ale.

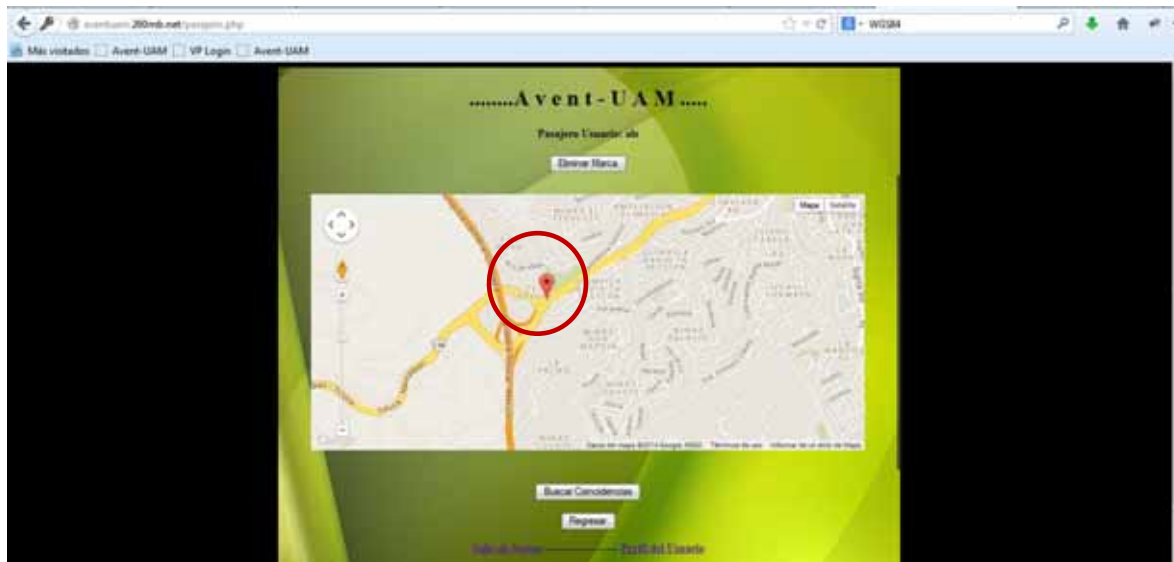


Figura 4.19- Posicionamiento de marca por parte del usuario Ale.

La figura 4.20a, 4.20b y 4.20c muestra las diferentes opciones que el usuario puede seleccionar iniciando con la opción más similar en primer lugar a la menos adecuada.



Figura 4.20a- Opciones mostradas por el sistema.



Figura 4.20b- Ruta cercana 2.



Figura 4.20c- Ruta cercana 3.

La figura 4.21 se muestra la ruta que el sistema obtuvo de la búsqueda (primera opción obtenida por el sistema), la cual encuentra un punto similar (marcado con negro) que corresponde al usuario conductor Alex.

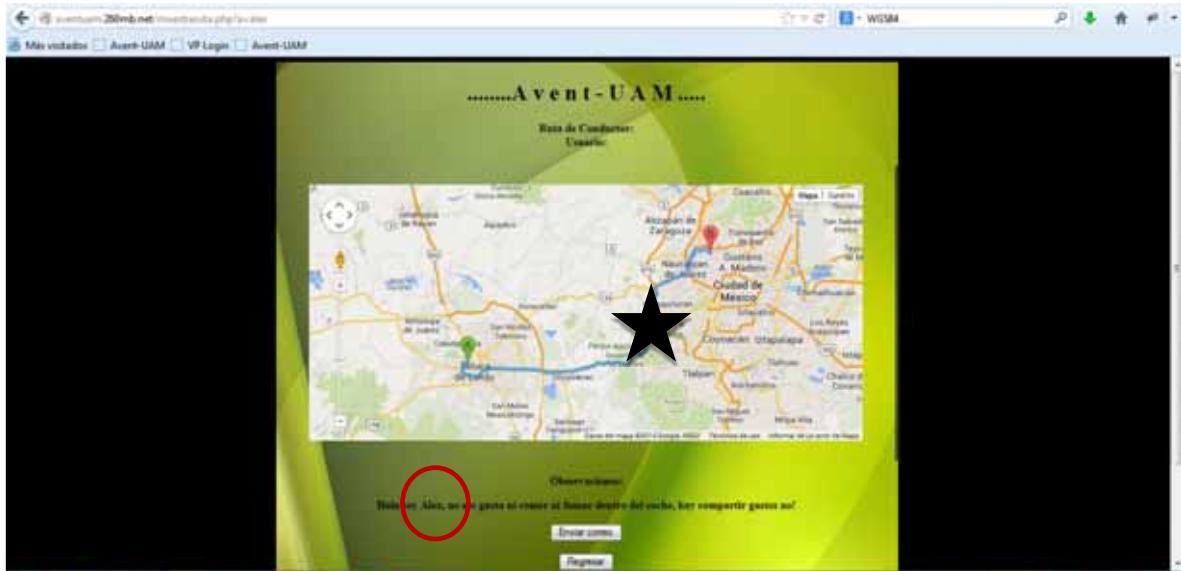


Figura 4.21- Mejor opción obtenida y seleccionada por el usuario (Ruta cercana 1).

Ejemplo 3

El usuario Conductor Silvia trazó una ruta de la UAM-Azcapotzalco a Teotihuacán de Arista, el cual es su recorrido diario. La figura 4.22 muestra la ruta trazada iniciando en la UAM-Azcapotzalco y culminando en Xochimilco.



Figura 4.22- ruta diaria de Alex.

Ana al no tener transporte personal se cataloga como usuario pasajero y desea aprovechar el viaje de un usuario conductor que tenga alguna ruta similar de su trayectoria UAM-Azcapotzalco hacia El metrobus Reforma. El usuario pasajero coloca su marca en la posición adecuada a sus requerimientos. En la figura 4.23 se observa la marca posicionada por el usuario Ana.

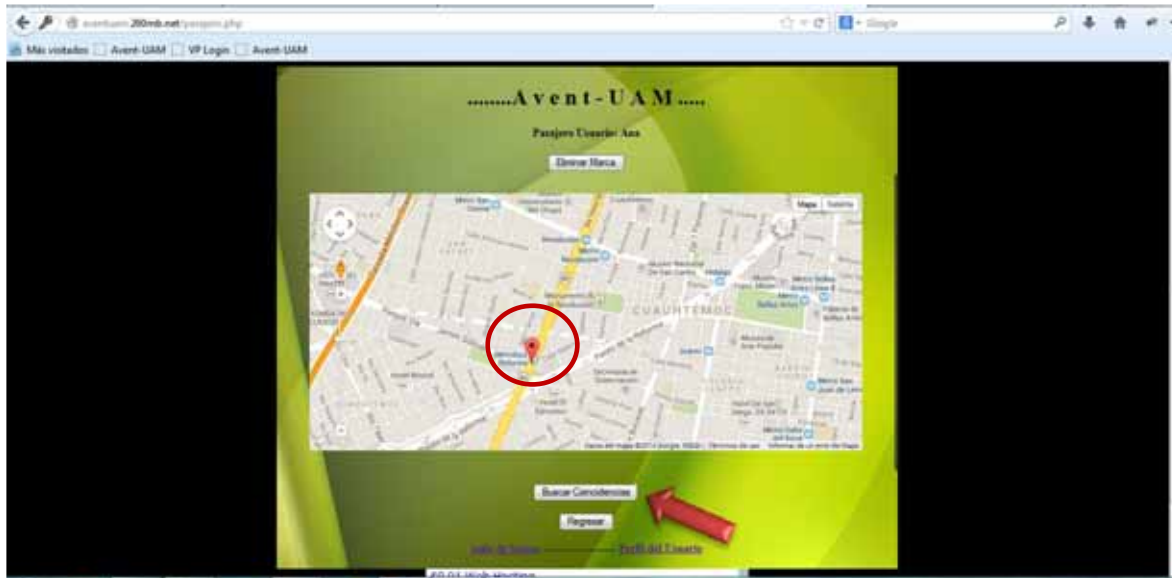


Figura 4.23- Posicionamiento de marca por parte del usuario Ana.

La figura 4.24 muestra las diferentes opciones que el usuario puede seleccionar iniciando con la opción más similar a la menos adecuada.

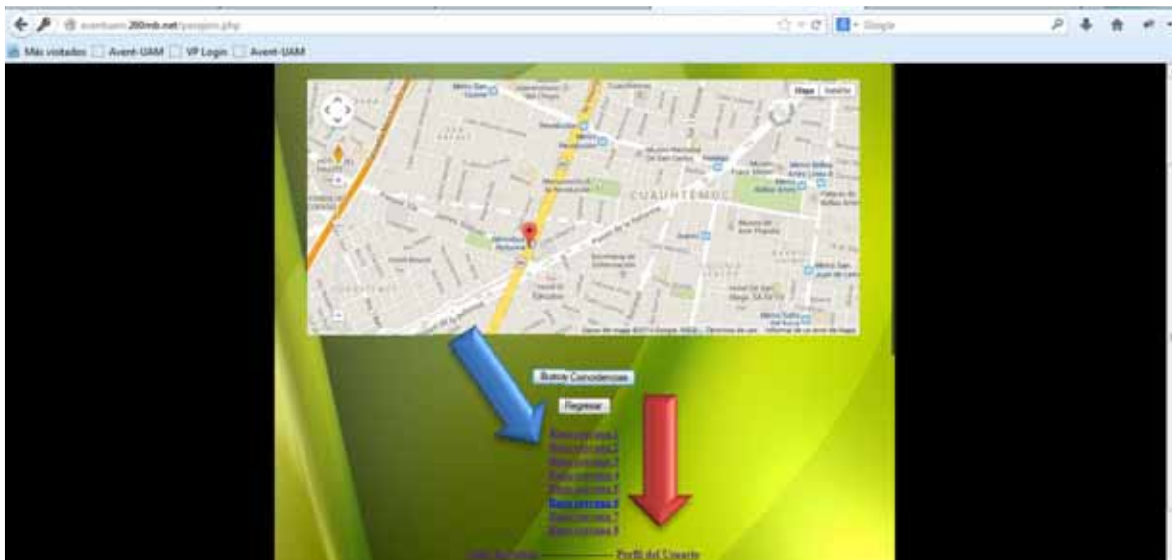


Figura 4.24- Opciones mostradas por el sistema.

La Figura 4.25 se muestra la ruta que el sistema obtuvo de la búsqueda (primera opción obtenida por el sistema), la cual encuentra un punto similar (marcado con negro) que corresponde al usuario conductor Alex.



Figura 4.25- Mejor opción obtenida y seleccionada por el usuario.

La Figura 4.26 se muestra la ruta que el sistema obtuvo de la búsqueda (primera opción obtenida por el sistema), la cual encuentra un punto similar (marcado con negro) que corresponde al usuario Alex (usuario conductor).



Figura 4.26- Mejor opción obtenida y seleccionada por el usuario.

El usuario puede terminar su sesión en el momento que lo desee, o explorar su perfil, en el “Perfil de Usuario”, encontramos la información relevante del usuario, información que fue solicitada al momento de registrar al usuario (Figura 4.27).

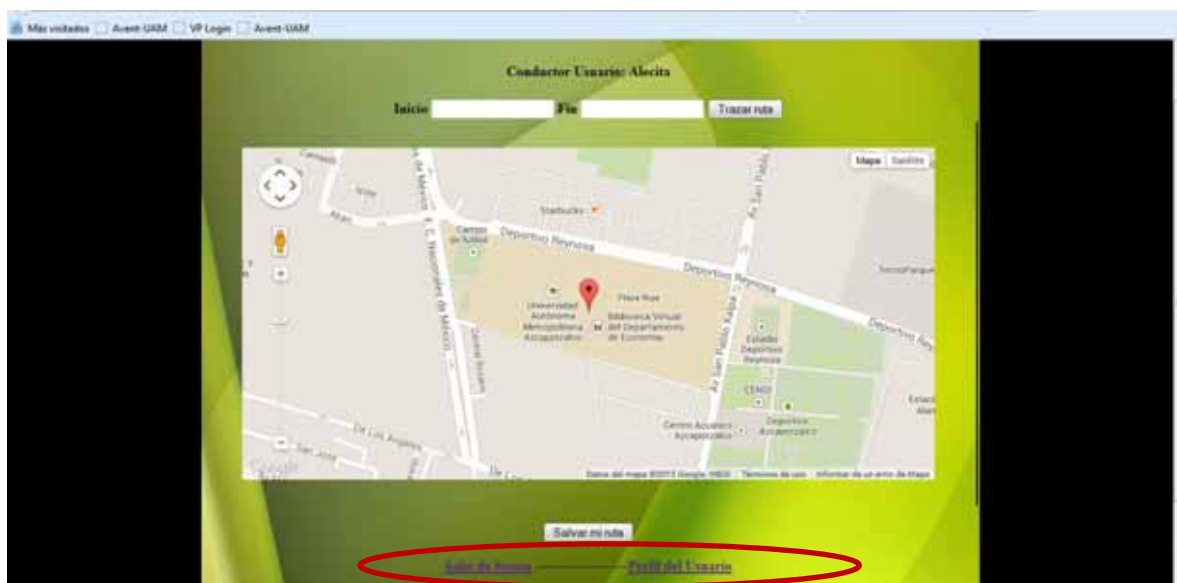


Figura 4.27- Opciones Salir de Sesión y Perfil de Usuario.

En la figura 4.28, cuando el usuario ha seleccionado la opción adecuada, aparecerá una alerta en la cual se indica que se ha aceptado la ruta y se enviará un correo al conductor que trazó la ruta. Este correo puede ser personalizado por parte del UP, indicando en el correo la información que considere necesario. Para poder realizar el envío del correo debemos tener configurado previamente el programa *Microsoft Outlook 2010* (Anexo B Manual de Instalación y Configuración de Microsoft Outlook 2010) de lo contrario no se podrá hacer uso de esta sección. Podemos utilizar como lo indica la ventana entre el sistema de correo por default o si se desea escoger entre las alternativas que se muestran.

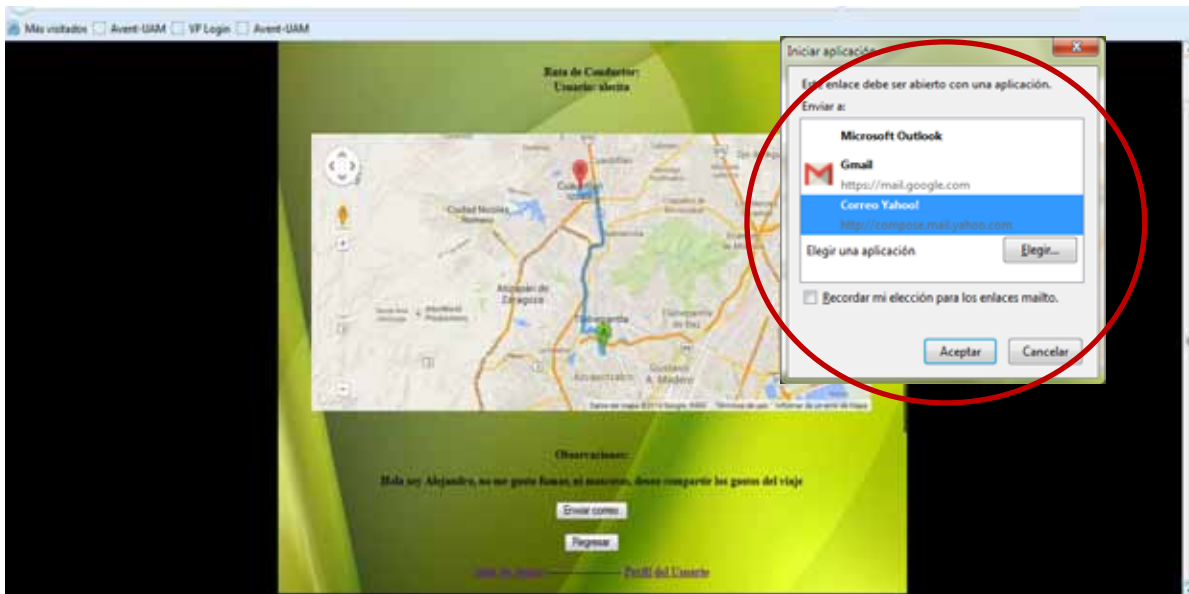


Figura 4.28- Envío de correo cuando la ruta ha sido aceptada

El envío del correo electrónico para realizar la solicitud por parte del UP se puede utilizar por medio de la opción de *Microsoft Outlook 2010* (Anexo B Manual de Instalación y Configuración de Microsoft Outlook 2010), o por medio de otro servicio de mensajería, cuando se haya enviado el correo al destinatario se hace doble *click* en regresar y retornamos al sistema *Avent-UAM*. (Figura 4.29)

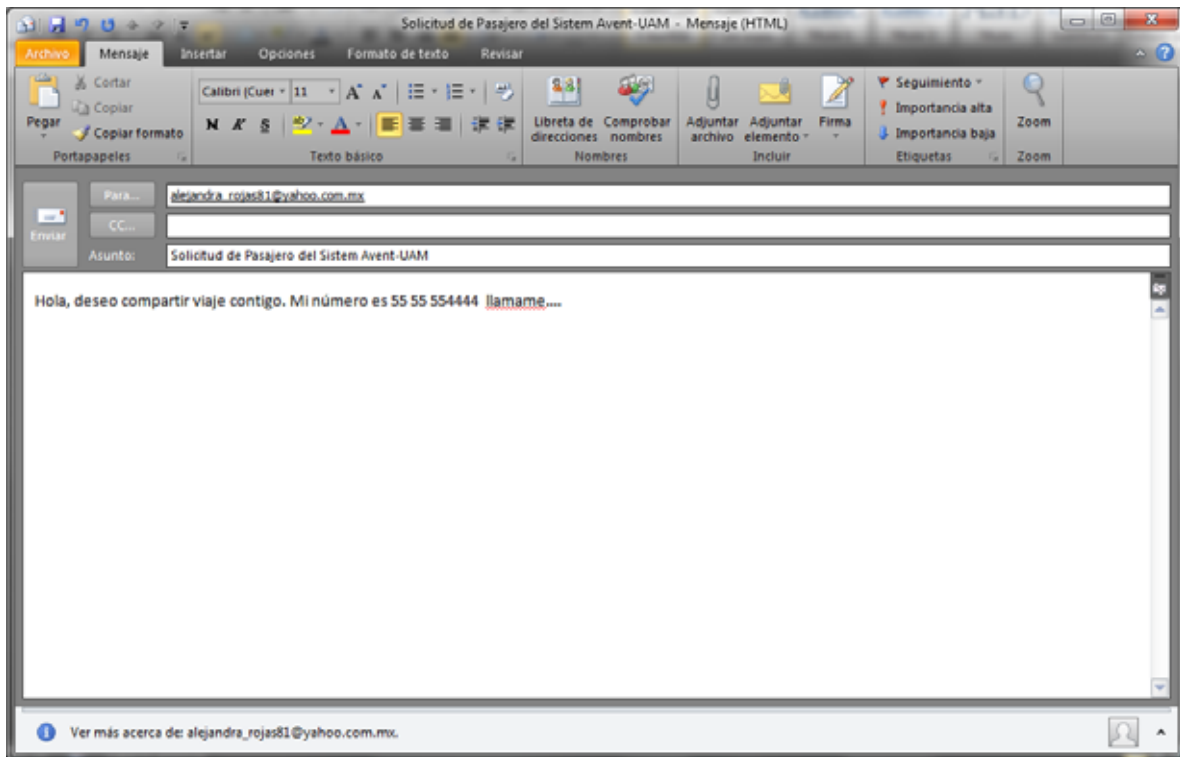


Figura 4.29- Interfaz de Microsoft Outlook 2010.

En *Microsoft Outlook* solo es necesario cerrar el servicio. (Anexo B: Manual de Instalación y Configuración de Microsoft Outlook 2010).

4.5 Pruebas: Conclusiones

Las pruebas realizadas al sistema arrojaron un resultado satisfactorio, al ejecutar cada una en diferentes puntos estratégicos con distintas distancias considerables, exponiendo de manera congruente el listado de las opciones obtenidas por el sistema. Con ello se comprueba el éxito del funcionamiento en la adecuación e implementación del algoritmo *M-Tree*, así como cada uno de los servicios y API's utilizados. Observando que la interoperabilidad realizada en el sistema funciona y proporciona resultados que cumplen con los requerimientos de los usuarios. Así mismo se expone que nuestro proyecto satisface más requerimientos que los sistemas similares privados. Como es la manipulación de la ruta por parte del usuario.

En el Capítulo 5 conclusiones y trabajos a futuro, se desglosan los puntos principales observados en el desarrollo, cada uno de los puntos clave que conllevaron a una ardua investigación para satisfacer los requerimientos del proyecto, así como miras a futuro para cubrir detalles observados en el desarrollo, y mejoras al mismo.

La información incluida en el CD de entrega de culminación del proyecto se enumera a continuación:

1.- Reporte y Anexos

2.- Códigos del Sistema

- Index.html
- jason2.js
- LOGIN.html
- login.php
- logout.php
- mostrarruta.php
- pasajero.php
- perfil.php
- process.php
- tipo_usuario.php
- validar.php.
- mtree.js
- CONEXIÓN.php
- registro.php

Capítulo 5.-Conclusiones y trabajos a futuro

5.1 Conclusiones

El presente proyecto *Avent-UAM* tuvo como objetivo “Diseñar e implementar una aplicación web que permita compartir viajes en automóvil a profesores y alumnos de la UAM-Azcapotzalco, mediante la API de *Google Maps* y el Algoritmo M-Tree, para contribuir al cuidado del medio ambiente”, mediante el avance tecnológico ayude a mejorar condiciones tanto ambientales, económicas, disminución de circulación vehicular.

Para desarrollar este sistema primero se analizó el problema, y las alternativas del mismo. Se estudiaron de manera individual y en conjunto tanto software, recursos y diferentes lenguajes de programación los cuales se verían reflejados en el desarrollo del mismo y en su interoperabilidad.

Observamos que la implementación del API de *Google Maps*, no presenta complicación si se utiliza y administra como lo establece la API de Google; pero estas características no cumplen con los requerimientos que el sistema necesita. Esto quiere decir que la información que la API muestra al solicitar el trazo de una ruta solo proporciona los puntos geométricos de origen y destino; por tal motivo los datos que se obtienen no son suficientes para lograr el objetivo del sistema. Este problema se solucionó por medio de “*steps ways*” estos son los hitos principales en la ruta, proporciona el mayor número de puntos estratégicos en cada trazó. Al conseguir esta información se obtienen los suficientes datos para que el sistema proporcione los puntos geográficos necesarios para que el motor del sistema pueda trabajar adecuadamente y al tener más información el resultado es más aproximado.

La figura 5.1 exhibe la ruta que es obtenida al ejecutar el API *Google Maps*, se observan los dos puntos de origen en verde y destino en rojo. Señalando únicamente dos puntos geográficos las cuales nos proporcionan la información de cada uno.

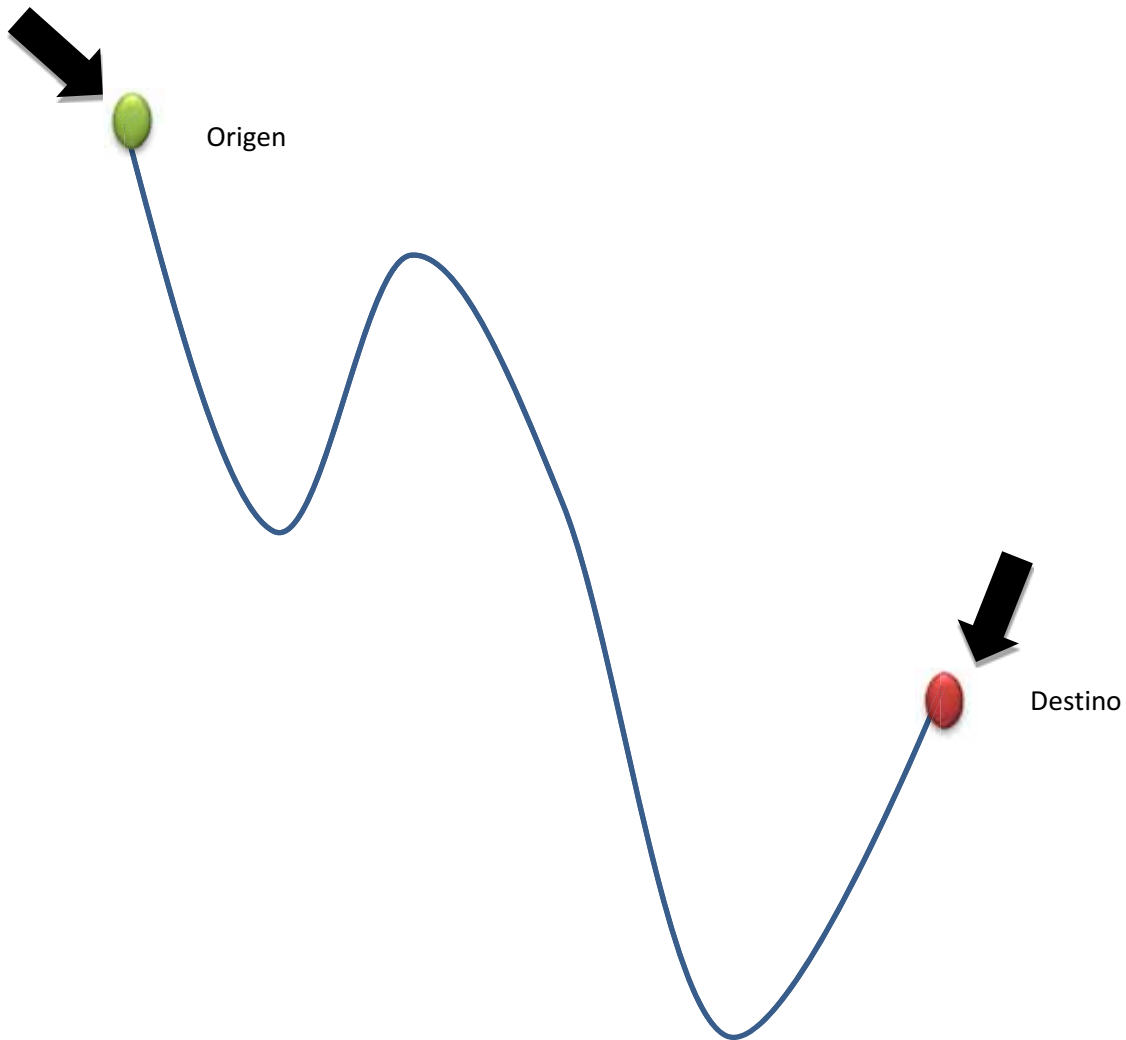


Figura 5.1- Ruta aproximada trazada por *Google Maps*.

La figura 5.2 observamos que el sistema obtiene mediante los *steps ways* que son las indicaciones principales en la ruta, más puntos geográficos que se utilizan como referencia para obtener datos suficientes para el árbol M que se crea con esta información. De tal manera que al ejecutarse el algoritmo obtenemos un resultado más fidedigno.

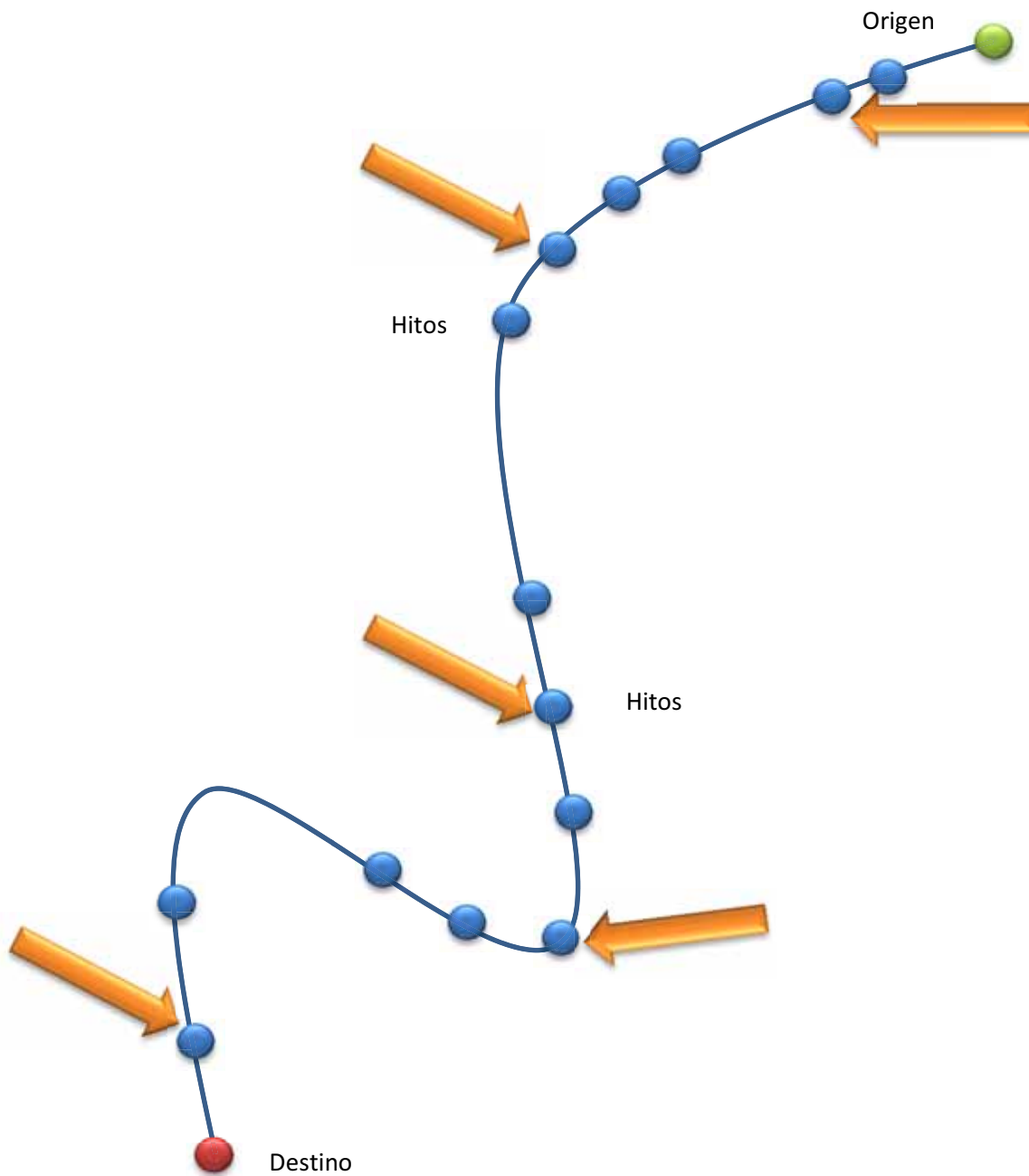


Figura 5.2- Ruta aproximada trazada utilizando *steps ways*.

La relevancia del proyecto se puede observar en la adecuación del algoritmo *M-Tree* al sistema, al ser éste un algoritmo robusto, esto es, trabaja para bases de datos

grandes, cuenta con características que satisfacen requerimientos que en nuestro proyecto no son necesarios, por tanto, es esencial delimitar las diferentes cualidades de este algoritmo y acotarlo al uso de distancias. El algoritmo se ajustó a trabajar con las distancias que lanzó como resultado el trazo de la ruta por medio de los *steps ways*.

Otro punto relevante que sobresale en el diseño del algoritmo es en el momento de almacenar los puntos geográficos arrojados del trazado de cada una de las rutas en arreglos, al iniciar la ejecución del *M-Tree* se obtuvo un desfase de tiempo, por lo cual se optó por utilizar una herramienta de *JavaScript* para retardar el inicio del algoritmo, esto es, existía una diferencia de más de 20 veces en tener la información en el arreglo que el tiempo que tarda el algoritmo en ejecutarse. La solución para resolver este conflicto de tiempo fue cambiar la información obtenida por el usuario conductor. En un inicio los *steps ways* se guardaban de manera temporal en un archivo, al momento de ser utilizados se ejecutaban desde la aplicación de *Google Maps* (en tiempo real), esto generaba que se utilizara un tiempo mayor que el usado por el algoritmo y por tanto existía un desfase de varios nanosegundos. Para alcanzar el objetivo y que el algoritmo se ejecutara simultáneamente con los datos, los *steps ways* son almacenados en una tabla de la base de datos llamada “puntos”. Con ello, cuando el algoritmo solicite la información la obtenga a la misma velocidad, de tal manera que al ejecutarse no exista un desfase y provoque errores de cálculo, obteniendo un resultado más aproximado.

Al diseñarse el sistema se consideró que sería necesario interactuar con diferentes lenguajes de programación así como el enlace con la base de datos, examinando cada uno de los lenguajes utilizados (*Java, JavaScript, HTML.PHP, MySQL*) los cuales deben hacer llamados a cada uno de los archivos subsecuentes, se decidió utilizar código libre “.js”, el cual facilitó la interoperabilidad del sistema. Así mismo se diseñó un código complementario .js para enlazar el algoritmo (motor) al sistema ya que éste por ser un código modificado especialmente para el proyecto no existe información o material para ello.

Si bien es un proyecto universitario, la implementación del mismo sería una gran contribución a nuestra Universidad. Como se observa la tecnológica no solo es diseñada y creada para fines lucrativos, también puede colaborar a crear conciencia en reducción de contaminantes, crear vínculos entre estudiantes, obtener más tiempo de calidad al no pasar tanto tiempo en el tráfico. Es necesario implementar este tipo de actividades ya que cada día somos más individuos en el mismo espacio.

5.2 Trabajos a futuro

El trabajo presentado satisface las necesidades planteadas en la propuesta, al realizar el diseño e implementación del servicio observamos características que se pueden mejorar en un futuro para una mayor eficiencia o simplemente para el mantenimiento del sistema

1. Manipulación de la ruta por parte del conductor en tiempo real, informando a los pasajeros de la modificación, por medio de una notificación. En el caso de que el usuario conductor encuentre la ruta planeada saturada pueda informar al usuario pasajero del cambio por medio de algún medio electrónico. De tal manera mantendrían contacto.
2. Mostrar en tiempo real el tránsito en la ruta trazada, para con ello buscar mejores alternativas en el momento en que se está realizando.
3. Aplicación para teléfonos celulares inteligentes.
4. Detección de la ubicación del usuario en tiempo real.
5. Establecer horario de cada ruta activa.
6. Establecer horarios para cada uno de las rutas trazadas (horarios fijos).
7. Modificaciones en las características de los usuarios.
8. Establecer lugares disponibles por parte del usuario conductor, manejo de diferentes usuarios.
9. Notificación cuando el viaje a realizar ya no tenga espacios disponibles para más usuarios.

Estas características permitirían a los usuarios manipular mejor la situación en tiempo real ya que se podrían controlar las características del sistema en el momento.

Bibliografía

[1]Google, Google Maps, Developers, Google Maps api V3, [Visitado el 23 de Septiembre de 2012] [En línea] Disponible en: <https://developers.google.com/maps/?hl=es>

[2] Barrientos, Ricardo, “Búsqueda por Similitud en Espacios Métricos Sobre Plataformas MultiCore (CPU y GPU)”, Tesis para obtención del grado de Maestría en Ciencias, Mención Computación, Facultad de Ciencias Físicas y Matemáticas, Escuela de Posgrado, Universidad de Chile, Chile, 2011.

[3] CNN ExpansionCarpool, compartamos auto,18 de junio de 2012, Trabajadores del Gobierno de Canadá. Disponible: <http://www.cnnexpansion.com/actualidad/2008/03/15/todos-para-uno-y-un-coche-para-todos>.

[4]El carpooling llega a México, 14 de noviembre de 2012, Estudiantes adoptan idea. Disponible: <http://www.empresayentorno.com/2012/11/el-carpooling-llega-a-mexico/>

[5] J. M. C. Guzmán, “Búsqueda de rutas más cortas entre diferentes sistemas de transporte en la Ciudad de México aplicando el enfoque GT”, Proyecto terminal de Ingeniería en Computación, DCBI, Universidad Autónoma Metropolitana Azcapotzalco, México, 2011.

[6] V. M. Hernández Muñoz, “Aplicación para dispositivos con Android que encuentre la mejor ruta entre dos estaciones del Metro”, Proyecto terminal de Ingeniería en Computación, DCBI, Universidad Autónoma Metropolitana Azcapotzalco, México, 2012.

[7] I. K. Zúñiga González, C. A. Blanno de la Vega, “Sistema para obtener información de distancias y recorridos de un par de elementos de un conjunto de sitios localizados en un mapa”, Proyecto terminal de Ingeniería en Computación, DCBI, Universidad Autónoma Metropolitana Azcapotzalco, México, 2012.

[8] Carpoolworld.com, Carpoolworld, interfaz y aplicación [Visitado el 23 de diciembre de 2012] [En línea] Disponible en: <http://www.carpoolworld.com/>

[9]Carpool.ca, CarpoolCánada, interfaz y aplicación [Visitado el 23 de diciembre de 2012] [En línea] Disponible en: <http://www.carpool.ca/>

[10] Aventones, Aventones, interfaz y aplicación [Visitado el 23 de diciembre de 2012] [En línea] Disponible en: <http://www.aventon.es/>

[11] Enríquez González, Ramón E. “Modelo de Espiral”, Administración Informática de las Organizaciones, UPIICSA, Instituto Politécnico Nacional [Visitado el 7 de marzo de 2014][En línea] Disponible en: http://www.sites.upiicsa.ipn.mx/polilibros/portal/Polilibros/P_externos/Administracion_informatica_de_las_organizaciones_Ramon_E_Enriquez_Gonzalez/AIO2_Mod_ESPIRAL.html

[12] Barrientos, Ricardo, “Búsqueda por Similitud en Espacios Métricos Sobre Plataformas MultiCore (CPU y GPU)”, Tesis para obtención del grado de Maestría en Ciencias, Mención Computación, Facultad de Ciencias Físicas y Matemáticas, Escuela de Posgrado, Universidad de Chile, Chile, 2011.

[13] ¿Qué son las Bases de Datos? [Visitado el 29 de noviembre de 2013] [En línea] Disponible en: <http://www.maestrosdelweb.com/editorial/%C2%BFque-son-las-bases-de-datos/>

[14] Higuera, Santiago, “Librería para acceder desde Javascript a la B.D. mysql” [Visitado el 18 de marzo de 2013] [En línea] Disponible: <http://bicimap.es/wordpress/?p=1459>

[15] ¿Qué es MySQL? [Visitado el 29 de noviembre de 2013] [En línea] Disponible en: <http://www.desarrolloweb.com/articulos/262.php>

[16] JavaScript [Visitado el 16 de agosto de 2013] [En línea] Disponible en: <http://es.wikipedia.org/wiki/JavaScript>

[17] The Java EE 6 (4th Edition) Autores: Eric Jendrock , Ricardo Cervera-Navarro, Ian Evans Editorial: Addison-Wesley Professional (2013) ISBN-10: 0137081863 ISBN-13: 978-0137081868

[18] Luján Mora, Sergio. Programación de aplicaciones web: historia, principios básicos y clientes web. Alicante: Editorial Club Universitario, 2002. ISBN 978-84-8454-206-3, 321 p.

[19] PHP,[Visitado el 18 marzo de 2013] [En línea] Disponible en <http://www.php.net/>

[20] The Apache Software Foundation,[Visitado el 20 mayo de 2013] [En línea] Disponible en: <http://www.apache.org/>

[21] ¿Qué es phpMyAdmin?[Visitado el 17 mayo de 2013] [En línea] Disponible en: http://www.phpmyadmin.net/home_page/index.php

[22] FileZilla The free FTP Solution [Visitado el 18 agosto de 2013] [En línea] Disponible en: <https://filezilla-project.org/>

Anexos

A. Manual de Administración.

B. Manual de Instalación y Configuración de Microsoft Outlook 2010.

C. Manual de Usuario.

Anexo A: Manual de Administración

Este manual de administración tiene como objetivo ser una guía en la implementación y administración del Sistema *Avent-UAM*.

En su contenido podremos observar los aspectos más esenciales para el mantenimiento y administración del Sistema.

El sistema *Avent-UAM* fue diseñado para facilitar la comunicación entre un usuario conductor que desea compartir su vehículo con un usuario pasajero con el cual se tenga en común como punto de origen o de destino la UAM Azcapotzalco.

En la implementación del Sistema *Avent-UAM* no es necesario instalar una gama de programas únicamente se debe instalar una aplicación para administrar el servidor, ya que toda la administración se realiza mediante un servicio de hospedaje gratuito.

La Figura A.1 muestra la interfaz del sistema de hospedaje gratuito 260MB donde debemos acceder con nuestro usuario y contraseña.



Figura A.1- Interfaz Usuario y contraseña para acceder al sistema de hospedaje gratuito.

En la figura A.2 se muestra la pantalla que seleccionaremos para el servicio que vamos a utilizar en la administración del sistema *Avent-UAM*.



Figura A.2- Selección de servicio de Administración del Sistema.

En la figura A.3 se debe seleccionar la opción de ADMIN, que nos direccionara a la ventana de la administración de la Base de Datos de Sistema.

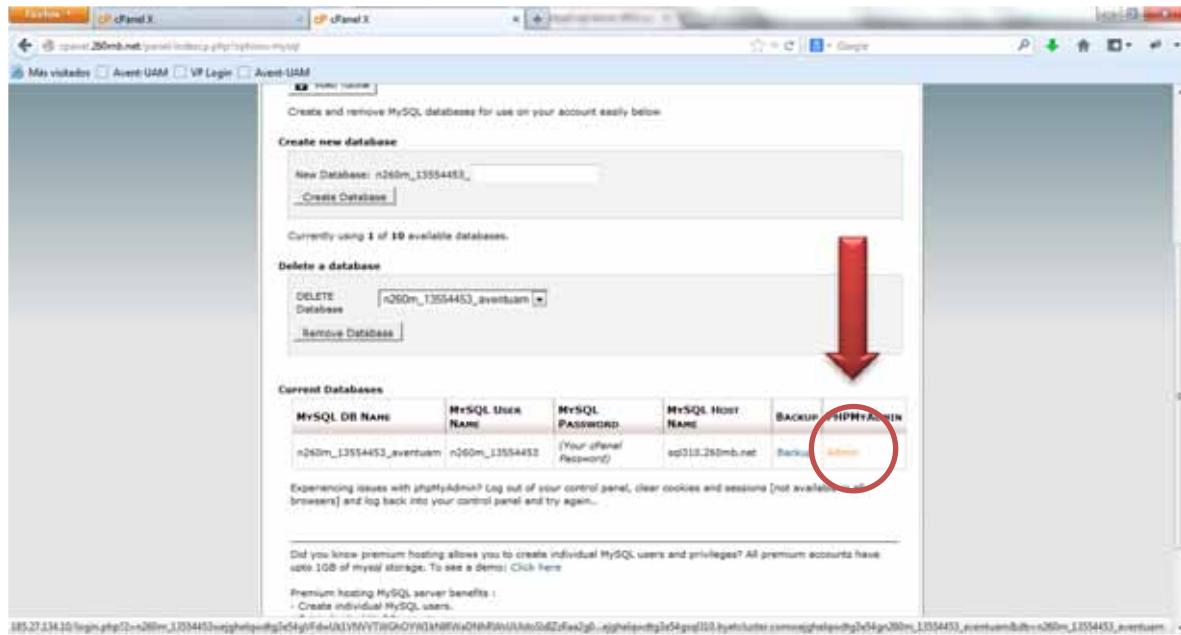


Figura A.3- Selección de Admin (administrar la Base de Datos).

En la figura A.4 se presenta la Base de Datos activa del Sistema, las tablas que contiene, cada una de las modificaciones que se realizaran se harán directamente a la base de datos (BD), esto es, si se desea eliminar un usuario se hará mediante esta aplicación, el proyecto no contempla modificaciones a la BD en esta etapa.

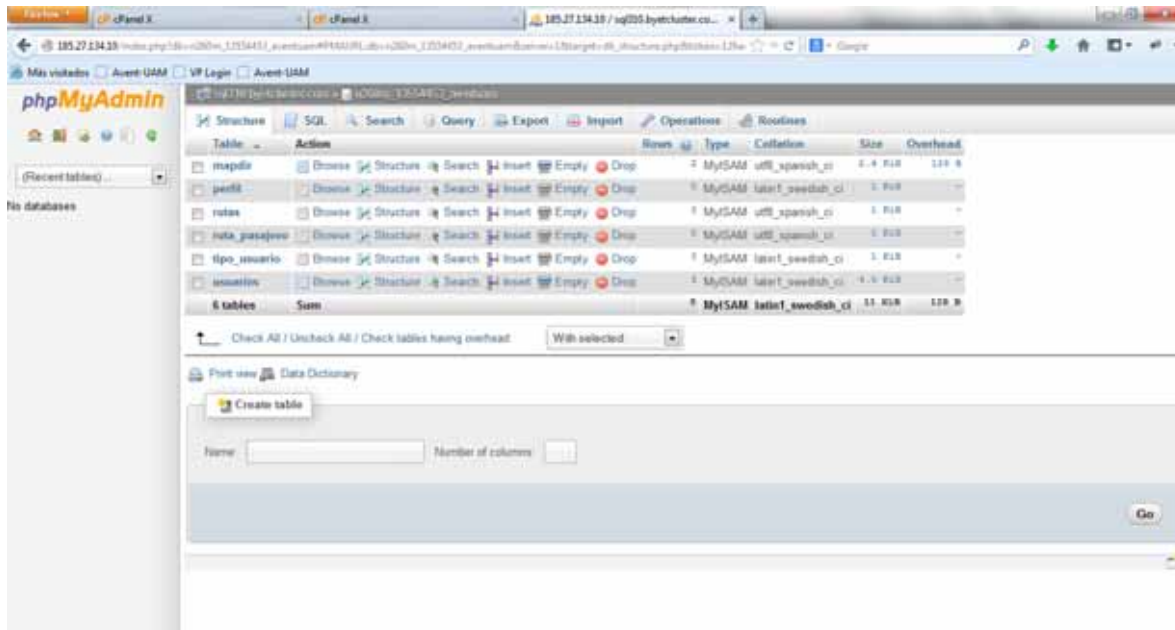


Figura A.4- Base de datos del Sistema Avent-UAM.

La figura A.5, muestra el administrador del servidor del Sistema *Avent-UAM*, la cual se realiza por medio de la interfaz *FileZilla*:

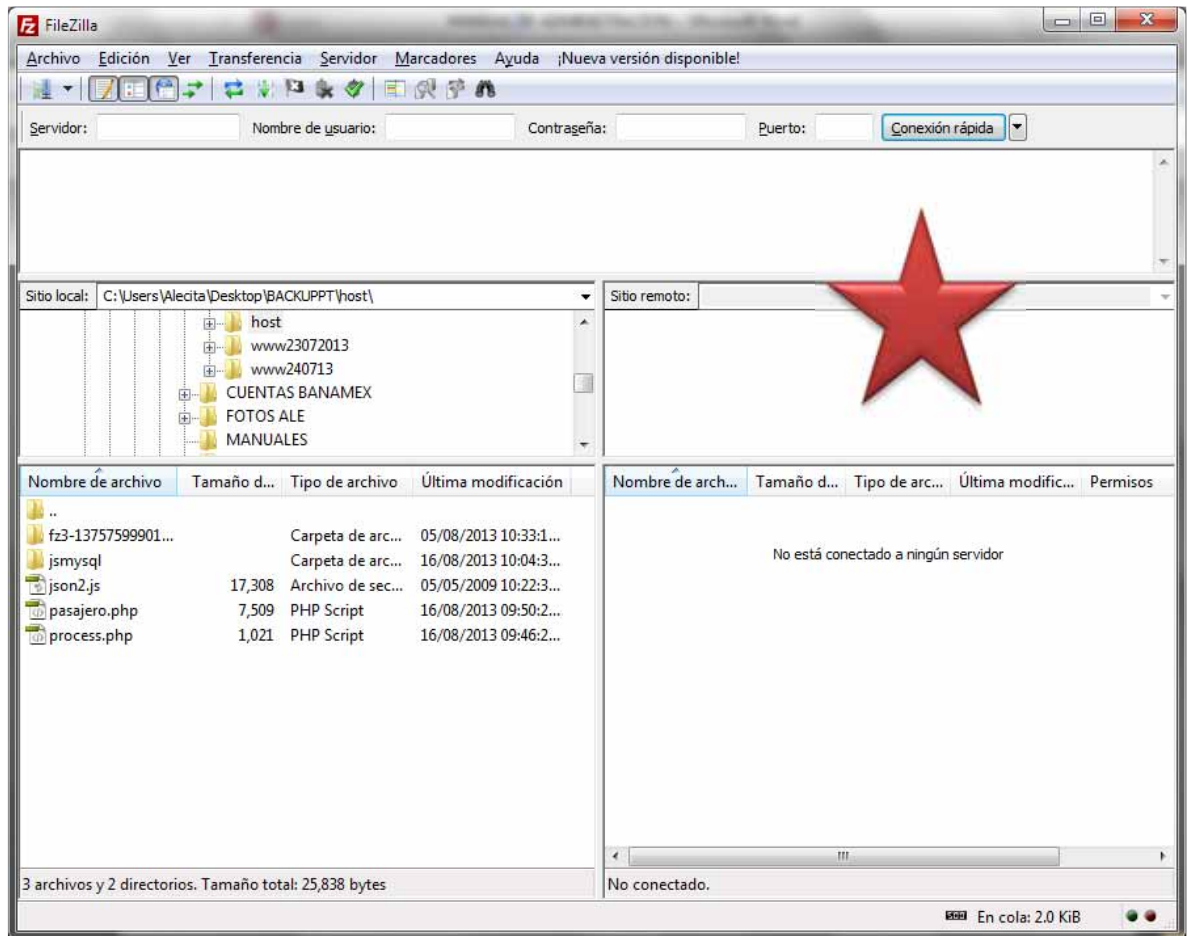


Figura A.5- Interfaz del administrador del Sistema *Avent-UAM*.

En la figura A.6 se visualiza la interfaz del servidor del sistema *Avent-UAM* el cuál está alojado en un host gratuito, pero es administrado mediante el programa denominado *FileZilla*, en el cual se actualizan y/o modifican al sistema. En la parte inferior derecha encontramos los archivos que conforman el proyecto, en ese apartado no se pueden hacer modificaciones, es necesario arrastrarlo a la parte izquierda de la pantalla y dentro de este recuadro se puede realizar todos los cambios. Existe una carpeta denominada *host*, a la cual podemos acceder y modificar los archivos que deseamos. Después de haber realizado los cambios, dar click derecho y seleccionar actualizar, arrastramos los archivos al recuadro derecho para que los cambios se visualicen en el servicio web.

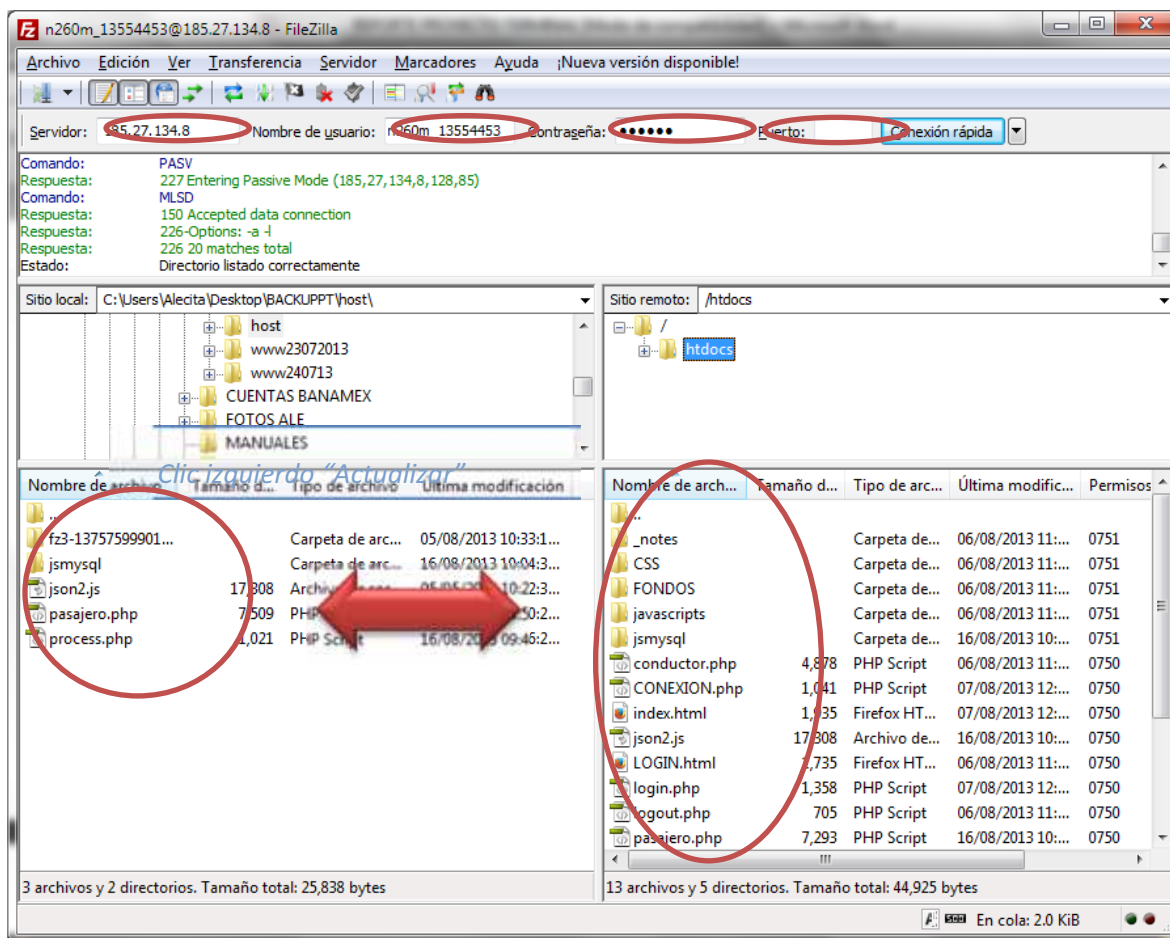


Figura A.6- Administración del Servidor.

En los círculos marcados en la figura A.6 se debe insertar la siguiente información:

- *FTP host name*: **185.27.134.8**
- *FTP username*: **n260m_13554453**
- *FTP password*: **Esten0**
- Puerto : 21

Cada una de las modificaciones del sistema se realizaran en los archivos correspondientes y serán actualizados en el servidor, en caso de ser necesario. Por el momento el proyecto queda concluido cubriendo cada uno de los puntos establecidos en la propuesta, por tal motivo no es necesario realizar modificaciones en el servidor.

Anexo B: Instalación y configuración de Microsoft Outlook 2010

El presente manual es una guía rápida para la instalación y configuración de *Microsoft Outlook 2010*.

Para el mejor desempeño del servicio web es necesario llevar a cabo la instalación previa de este programa. Cuando se ejecute el servicio web se muestra una ventana para seleccionar el servicio de correo electrónico seleccionado por el usuario.

Nota: No es necesario tener abierto el programa al ejecutar el servicio.

Dar click en “Next” para continuar con la instalación. (Figura B.1)

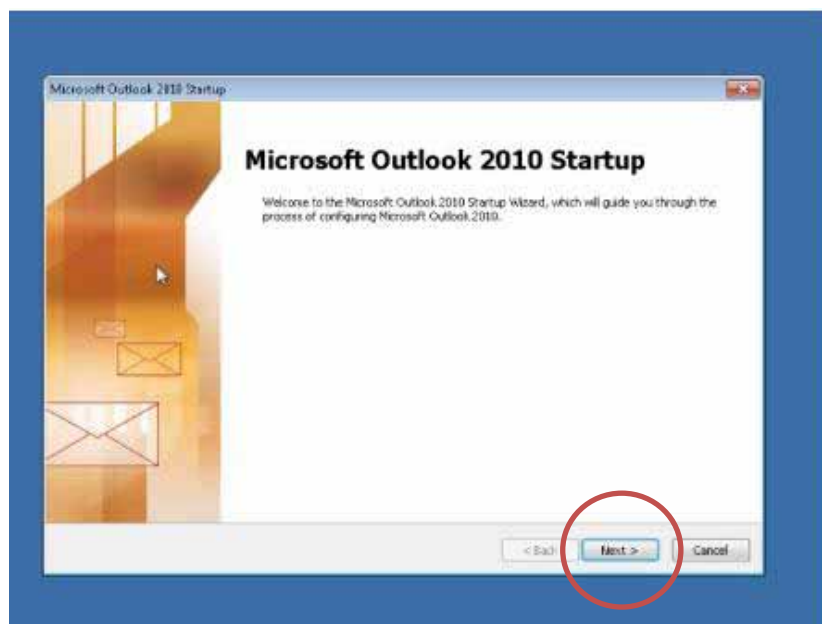


Figura B.1- Inicio de instalación de *Microsoft Outlook 2010*.

Haga click en “yes” y después seleccione en botón “Next” (Figura B.2).

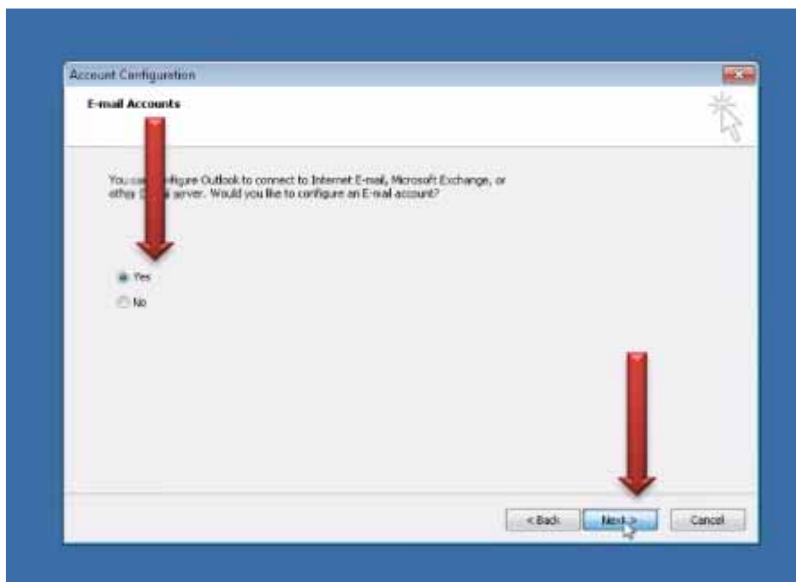


Figura B.2- Cuenta de correo electrónico.

A continuación aparecerá una ventana donde deberá insertar la información solicitada en cada uno de los registros (Figura B.3).

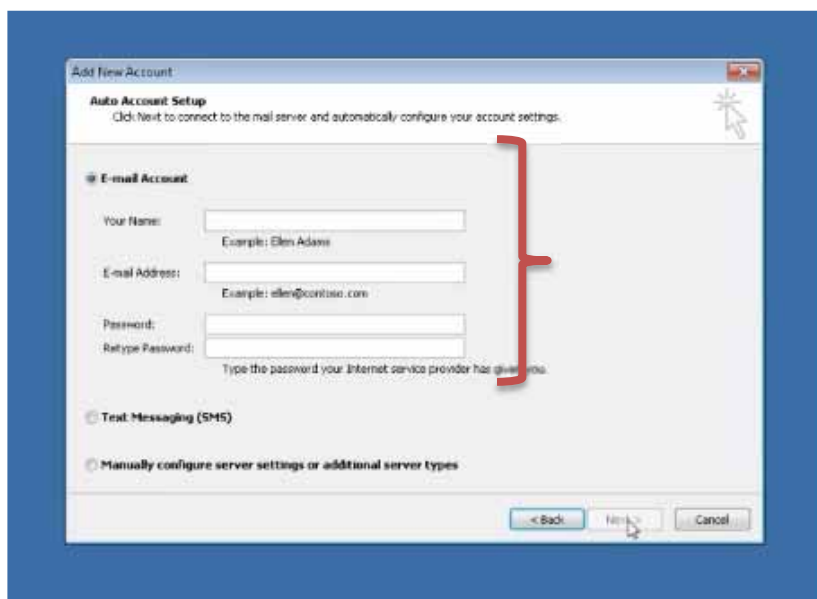


Figura B. 3- Configuración de la cuenta de correo electrónico.

Una vez que ha completado la información solicitada por el sistema de click en “Next” (Figura B.4).

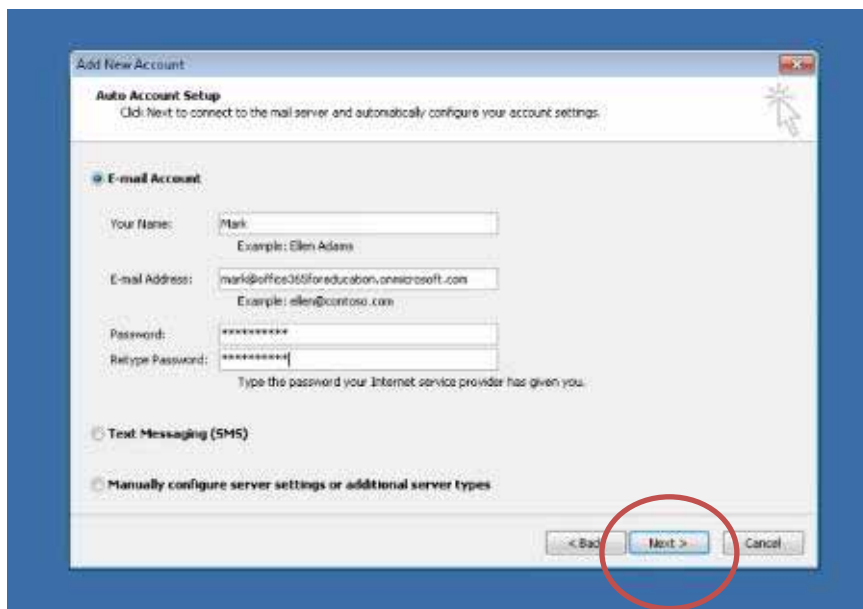


Figura B.4- Llenado de registros.

Microsoft Outlook 2010 realizara la configuración del sistema, dar click en “Finish”. Al finalizar este proceso estará listo para que usted pueda utilizar el programa (Figura B.5).

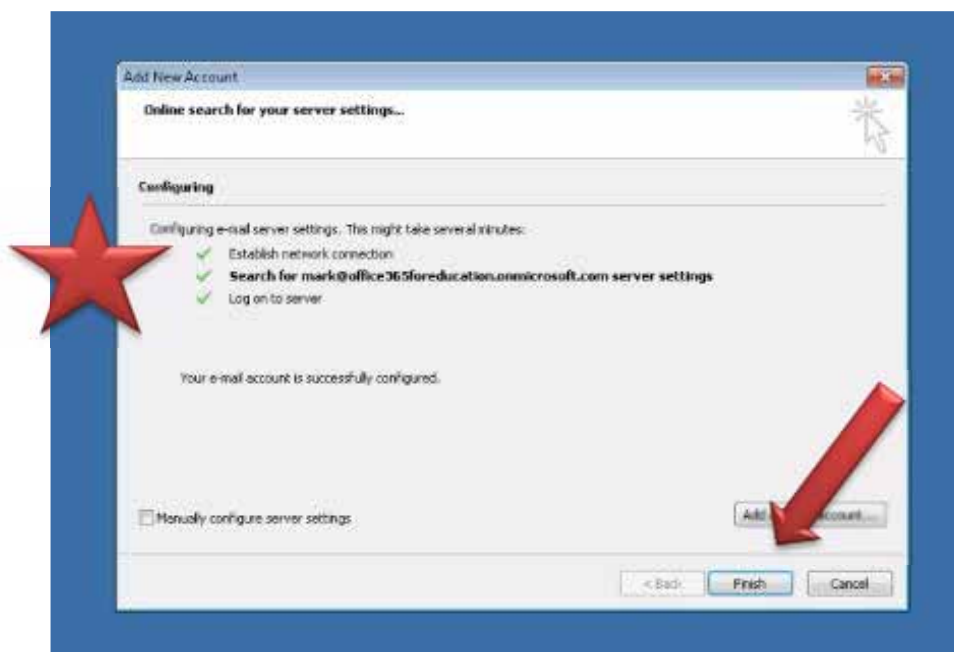


Figura B.5- Configuración de información.

Posteriormente en el escritorio aparecerá el icono de *Microsoft Outlook 2010* a partir del cual se podrá acceder a su correo electrónico (Figura B.6).

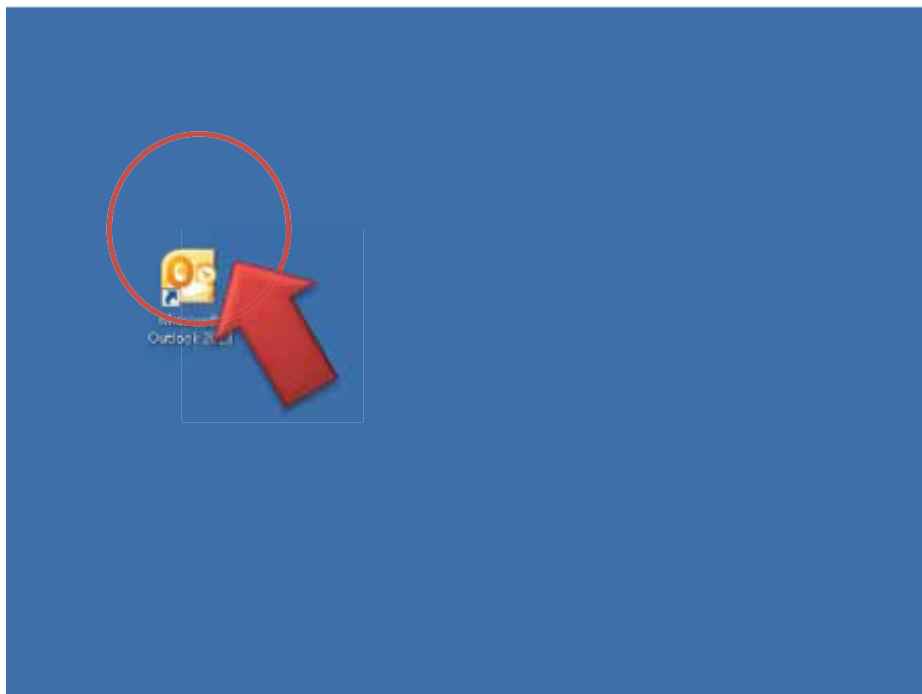


Figura B.6- Acceso a Microsoft Outlook 2010.

Anexo C: Manual de Usuario

El presente manual es una guía en el uso del Sistema Avent-UAM. En su contenido podremos observar los aspectos esenciales para utilizar de una manera muy fácil cada uno de los módulos.

Ingreso al Sistema *Avent-UAM*

Se debe acceder por medio por medio de la siguiente dirección:

<http://aventuam.260mb.net/>

La cual nos direccionara a la página de bienvenida del Sistema *Avent-UAM*.

La pantalla de inicio nos envía a la página de “inicio de sesión” (Figura C.1).

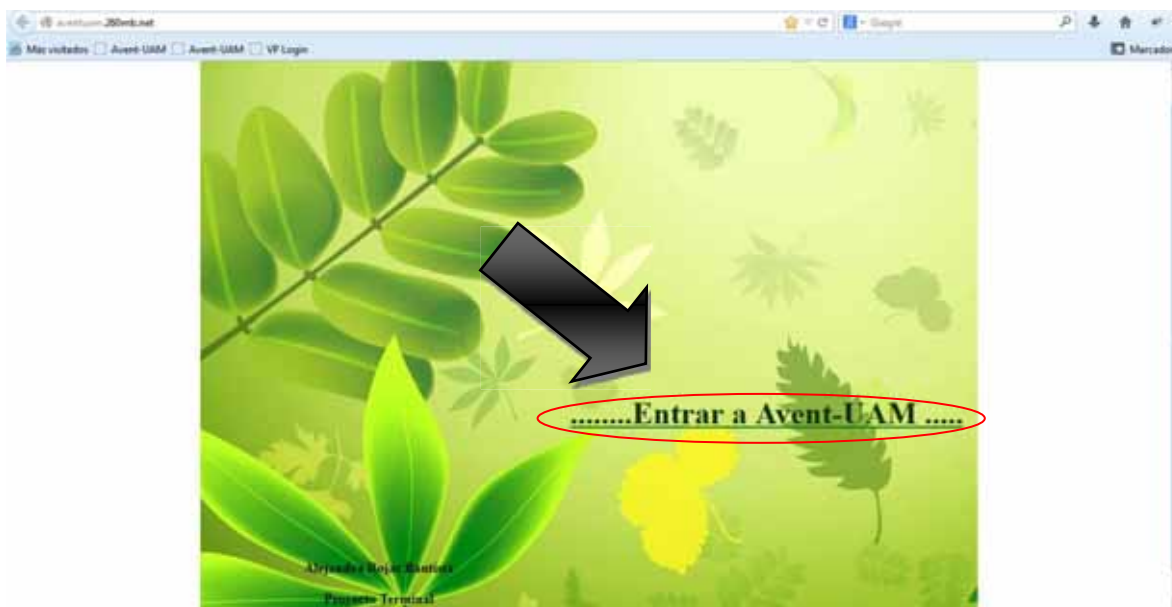


Figura C.1- Pantalla de Inicio del Sistema Avent-UAM.

La figura C.2 muestra la ventana de Inicio de Sesión. El usuario debe proporcionar su “Usuario” y “Contraseña”, dar *click* en “ingreso”, si éste no está registrado en el sistema deberá seleccionar “REGÍSTRATE” y proporcionar la información que se solicita.



Figura C.2- Pantalla de Inicio de Sesión del Sistema Avent-UAM.

El usuario deberá introducir los datos que el sistema solicita estos son:

- Usuario: Escribir el nombre de usuario que desea tener.
- Password: Escribir el password que considere, no mayor a 8 caracteres.
- Licenciatura: Se solicita esta información para que el usuario se familiarice más fácilmente con usuarios de su misma división o carrera.
- Correo Electrónico: Se solicita para enviar notificaciones a los usuarios.
- Observaciones: Información relevante sobre las características del viaje.

En la Figura C.3 se muestra la pantalla de inicio de alta al usuario y los datos que se solicitan.

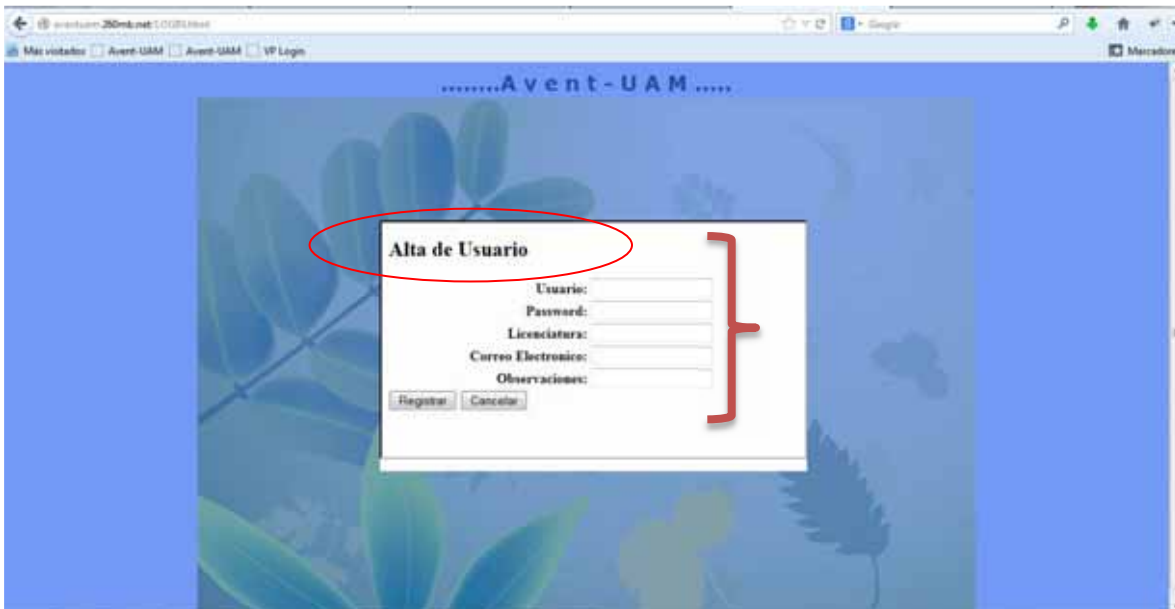


Figura C.3- Pantalla de Registro de Usuarios del Sistema *Avent-UAM*.

Una vez dentro del Sistema, se mostrara la pantalla denominada “Tipo de Usuario”, la cual nos proporciona las opciones de Tipo de Usuario “Conductor” y Tipo de Usuario “Pasajero”. Existe también la opción de regresar o salir de sesión.

Se puede observar que el usuario accede y se muestra el usuario que en el momento está activo (Figura C.4).

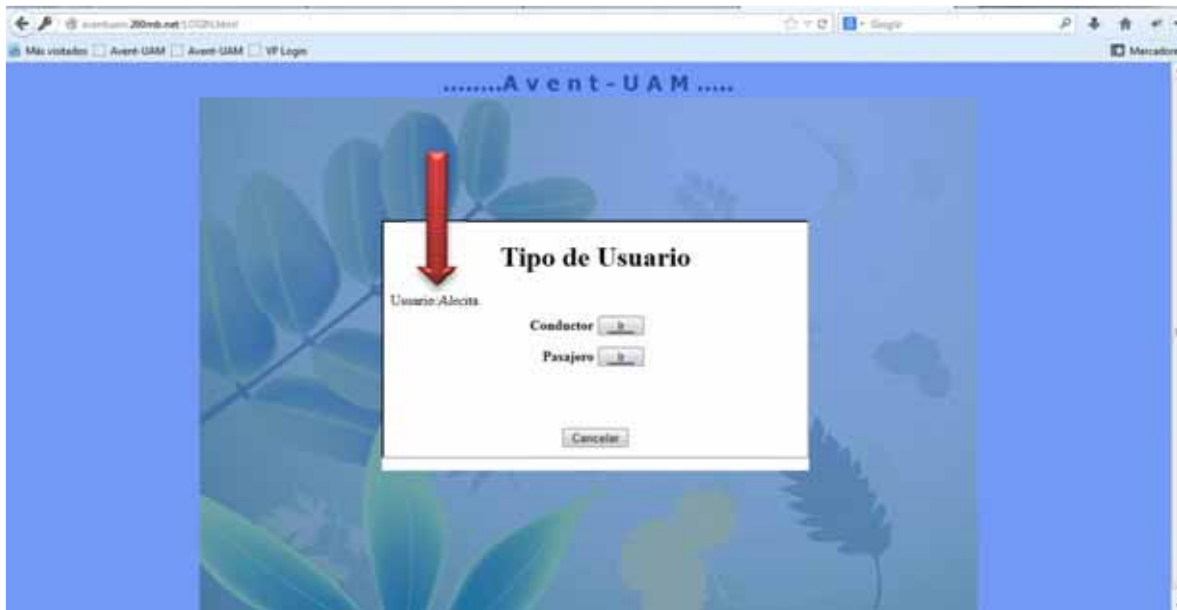


Figura C.4.- Pantalla de Tipo de Usuario del Sistema *Avent-UAM*.

La opción Usuario Conductor nos permite trazar una ruta que será almacenada en la base de datos, y con ello algún usuario pasajero pueda considerarla como opción.

El Usuario Conductor debe introducir los datos de inicio de viaje y fin del mismo, los cuales son correspondientes a las celdas “Inicio” y “Fin”. Pulsar la opción “Trazar ruta”. El sistema trazará la ruta óptima que el servicio de *Google Maps* muestra por default, si el usuario lo desea puede personalizar su recorrido, utilizando el mouse y arrastrando la ruta en los puntos que se desee. Obteniendo la ruta adecuada a las necesidades, para finalizar se debe dar click en “Salvar mi ruta” (Figura C.5).

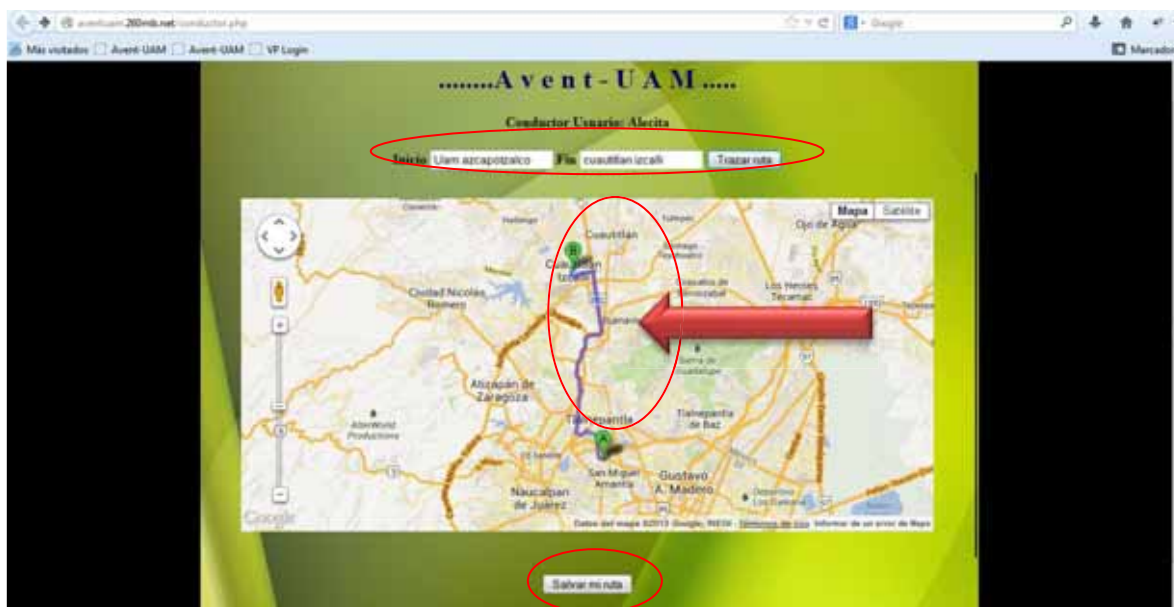


Figura C.5- Interfaz de Usuario Conductor del Sistema Avent-UAM.

En la figura C.6 observamos que la ruta ha sido almacenada con éxito, lo cual se muestra en el recuadro blanco.

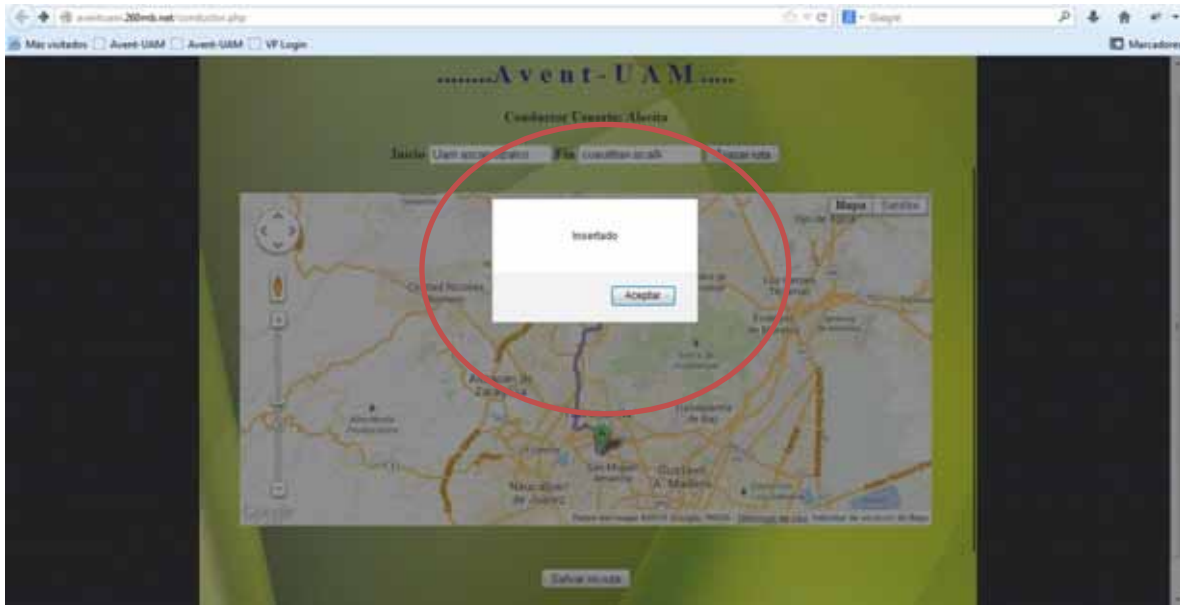


Figura C.6- Interfaz de Usuario Conductor del Sistema *Avent-UAM* “Salvar mi ruta”.

El Usuario Conductor tiene las opciones “Salir de Sesión” y “Perfil de Usuario” en la parte inferior de la página.

La opción de “Salir de Sesión” nos apoya cuando el usuario no desea continuar en el sistema o haya concluido.

En la opción “Perfil de Usuario” tenemos la información del usuario activo, información personal y preferencias en el viaje (Figura C.7).

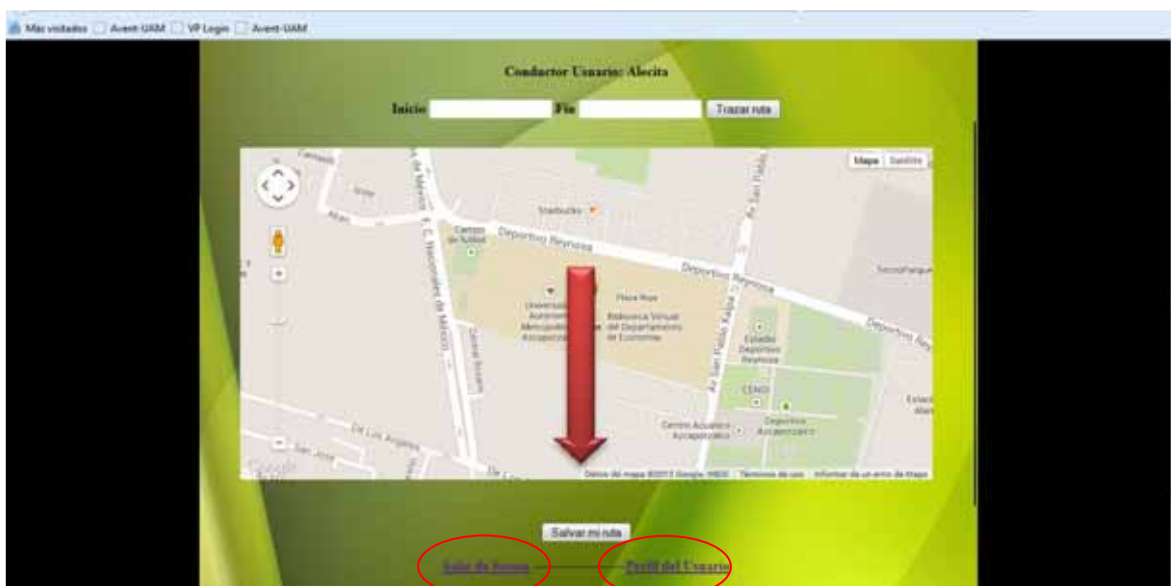


Figura C.7- Interfaz de Usuario Conductor del Sistema *Avent-UAM*, opción “Salir de Sesión” y “Perfil de Usuario”.

La figura C.8 muestra la interfaz del Usuario “Pasajero”, éste debe buscar en el mapa el punto de ubicación en el cual se basará la búsqueda de la ruta que será similar a sus necesidades. Ésta interfaz cuenta con tres opciones, la primera “Eliminar Marca”, se utiliza cuando la marca posicionada ha sido incorrecta o el usuario prefiere otra ubicación; la opción otra opción, “Buscar Coincidencias”, muestra las coincidencias proporcionadas por el sistema, “Regresar” cuando deseamos seleccionar otra opción.

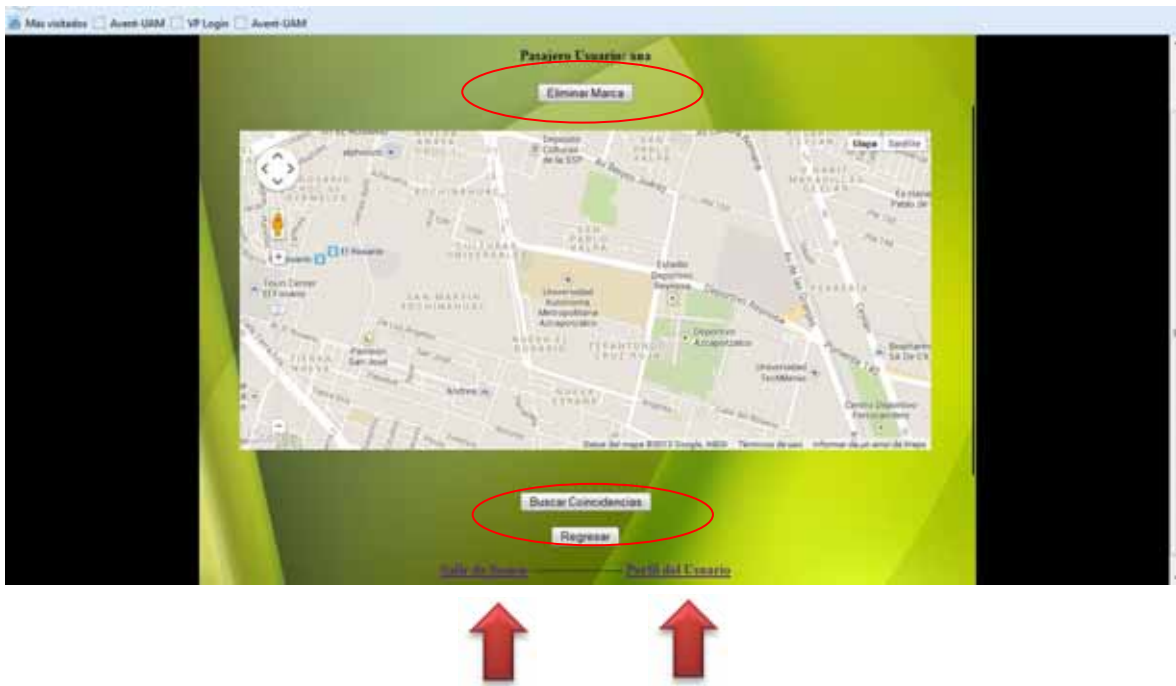


Figura C.8- Interfaz de Usuario Pasajero del Sistema *Avent-UAM*.

Cuando el usuario de un click con el *mouse* en la ubicación correcta aparecerá una marca roja (globo rojo) indicando el punto seleccionado. Al dar click en “Búsqueda de Coincidencias”, ésta despliega un listado de las rutas más similares hasta la menos de la ubicación que se seleccionó. (Figura C.9).

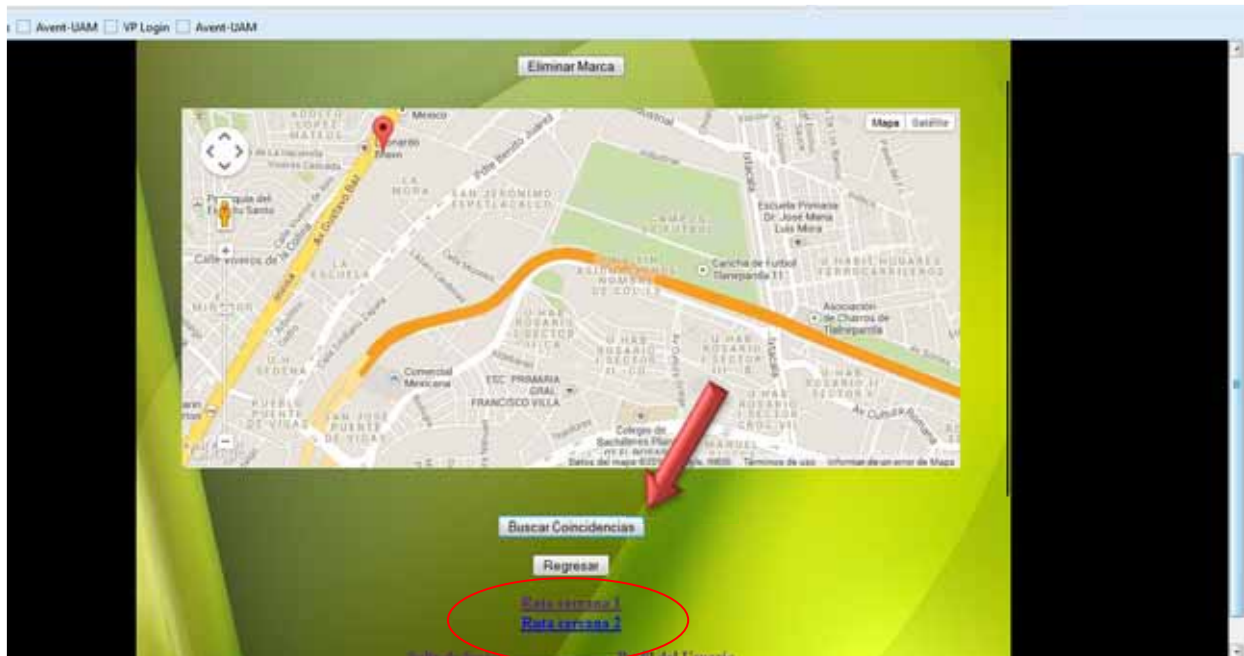


Figura C.9- Interfaz de Usuario Pasajero del Sistema *Avent-UAM*, marca de la ubicación selecciona.

La figura C.10 muestra la ruta seleccionada por parte del usuario pasajero, puede o no ser la ruta óptima, en el caso de que no sea la mejor, el usuario puede dar click en “Regresar” seleccionando otra opción.



Figura C.10- Ruta seleccionada

Cuando el usuario ha seleccionado la opción adecuada a sus necesidades, aparecerá una alerta en la cual se indica que se ha aceptado la ruta y se enviará un correo al conductor que trazó la ruta. Debemos corroborar que tenemos configurado *Microsoft Outlook 2010* (Anexo B. Instalación y configuración de *Microsoft Outlook 2010*) para poder ejecutar esta opción (Figura C.11).

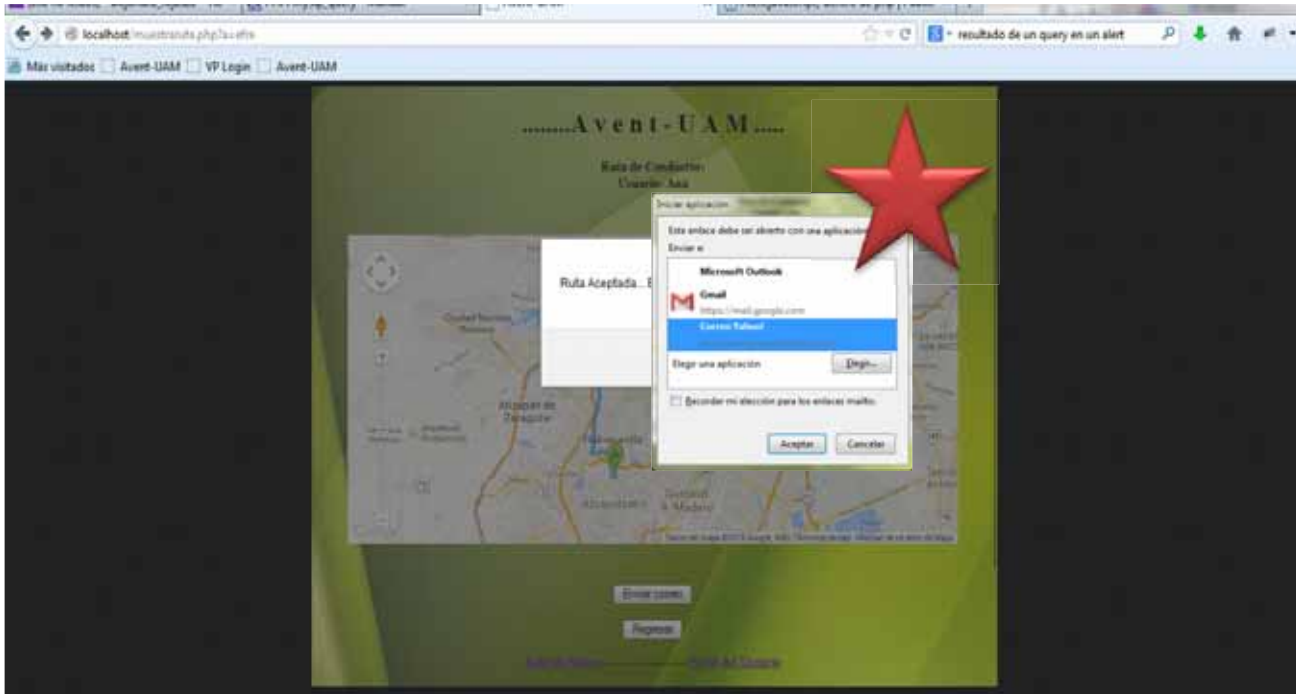


Figura C.11- Envío de correo cuando la ruta ha sido aceptada

El Usuario Pasajero tiene las opciones “Salir de Sesión” y “Perfil de Usuario” en la parte inferior de la página.

La opción de “Salir de Sesión” apoya cuando no desee continuar en el sistema o haya concluido.

En la opción “Perfil de Usuario” tenemos la información del usuario activo, nombre del usuario, correo electrónico, licenciatura, observaciones (información de viaje).

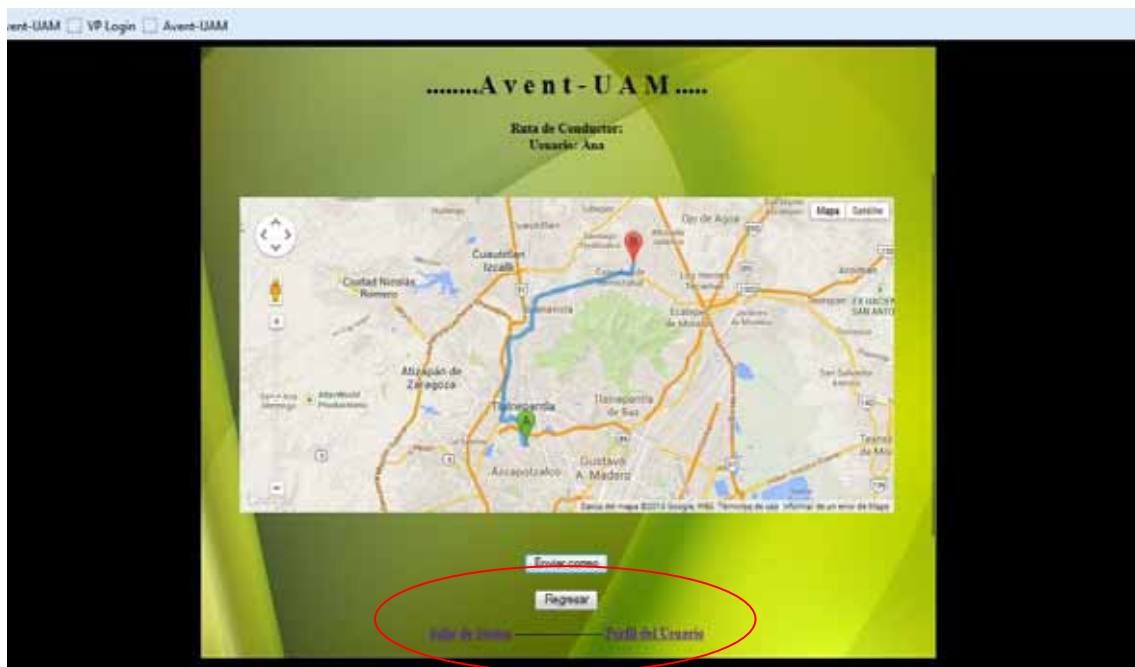


Figura C.10- Interfaz de Usuario “Pasajero” del Sistema *Avent-UAM*, opción “Salir de Sesión” y “Perfil de Usuario”.