



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

TESIS PROFESIONAL

SISTEMA DE GEOPOSICIONAMIENTO DEL PUMABÚS

**QUE PARA OBTENER EL TÍTULO DE INGENIERO EN
COMPUTACIÓN**

PRESENTAN:

CÉSAR MIGUEL BARRIENTOS ROBLES

ISMAEL HERNÁNDEZ MILLARES

DIRECTORA DE TESIS:

M.C MA. JAQUELINA LÓPEZ BARRIENTOS



CIUDAD UNIVERSITARIA OCTUBRE 2015



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

Índice

Tabla de contenido

Introducción	8
Objetivo general	10
Objetivos particulares	10
Capítulo 1. Marco Teórico	11
1.1. Mercado de Telefonía Celular en México	12
1.2. Información estadística del Sistema de Transporte Interno PumaBús 14	
1.3. Sistemas operativos para dispositivos móviles.	16
1.3.1. Android.....	18
1.3.2. iOS.....	23
1.3.3. Windows Phone	26
1.4. Lenguajes de programación	28
1.4.1. Android Java	28
1.4.2. Objective C.....	30
1.4.3. HTML 5.....	31
1.5. Dispositivo Receptor.....	31
1.5.1. Arduino.....	31
1.5.2. Raspberry Pi.....	33
1.5.3. Smartphone	34
Capítulo 2. Soluciones Propuestas	35
2.1. Propuesta 1. Pantallas de localización satelital.....	36
2.1.1. Requisitos.....	37
2.1.2. Ventajas.....	37
2.1.3. Desventajas	38
2.2. Propuesta 2. Pantallas de localización mediante sensores.....	38
2.2.1. Requisitos.....	39
2.2.2. Ventajas.....	39
2.2.3. Desventajas	39
2.3. Propuesta 3. Uso de SmartPhones como dispositivos receptores y emisores	40
2.3.1. Requisitos.....	41
2.3.2. Ventajas.....	41
2.3.3. Desventajas	41
2.4. Selección de la mejor alternativa	42
2.4.1. Justificación de elección.....	42
2.4.2. Requerimientos para la implementación de la solución	43

Cápitulo 3. Diseño y Desarrollo	44
3.1. Diseño de la aplicación	45
3.1.1. Phonegap.....	45
3.1.2. NodeJS.....	45
3.1.3. Heroku.....	47
3.1.4. Herramientas de desarrollo	47
3.1.5. Servidor web.....	47
3.1.6. Estructura de la aplicación	48
3.2. Realización de la aplicación	49
3.2.1. Diseño HTML y CSS.....	49
3.2.2. Sockets web y NodeJS.....	49
3.2.3. Google Maps	50
3.2.4. JavaScript.....	50
3.2.5. Íconos y Logos	50
3.3. Documentación de la aplicación	52
3.3.1. Adaptación del Servidor.....	53
3.3.2. Cálculo de tiempos	54
Cápitulo 4. Pruebas y Resultados	55
4.1. Producto Final.....	56
4.1.1. Información.....	57
4.1.2. ¿Dónde estoy?.....	61
4.1.3. Localizar.....	62
4.2. Pruebas de funcionalidad del sistema.....	65
4.2.1. Prueba 1	66
4.2.2. Prueba 2	68
4.2.3. Prueba 3	70
4.2.4. Prueba 4	72
Conclusiones.....	73
Glosario	74
Referencias	79
Anexos	81
Anexo 1. Diagrama UML de Casos de Uso	81
Anexo 2. Diagrama de Flujo	82
Anexo 3. Diagrama UML de	83

Índice de Figuras

Figura 1. 1 Tendencias del consumidor Móvil en México 2013.....	12
Figura 1. 2 Gráfica de Búsquedas es Google Trends.....	13
Figura 1. 3 Circuito de Ciudad Universitaria antes del PumaBús	14
Figura 1. 4 Logotipo Android.....	18
Figura 1. 5 Estructura de Sistema Operativo Android.....	19
Figura 1. 6 Proceso de runtime de Android.....	21
Figura 1. 7 Logotipo de iOS versión 7	23
Figura 1. 8 Arquitectura del Sistema Operativo IOS	23
Figura 1. 9 Arquitectura Windows NT.....	26
Figura 1. 10 Logotipo de Java	28
Figura 1. 11 Logotipo de Arduino.....	31
Figura 1. 12 Esquema de integración de GPS con Arduino.....	32
Figura 2. 1 Pantalla 1	36
Figura 2. 2 Pantalla 2	36
Figura 2. 3 Esquema general primera alternativa	37
Figura 2. 4 Esquema general de la segunda alternativa.....	39
Figura 2. 5 Esquema general de tercera alternativa	41
Figura 3. 1 Evolución de JavaScript.....	46
Figura 3. 2 Operación de JavaScrip y Node.js.....	46
Figura 3. 3 Diagrama de navegación de la aplicación.....	48
Figura 3. 4 Ejemplo de llamado a un CSS.....	49
Figura 3. 5 Asignación de un puerto a un socket.....	49
Figura 3. 6 Conexión del cliente al socket.....	49
Figura 3. 7 Respuesta a la conexión del cliente.....	49

Figura 3. 8 Llamado a la librería de Google Maps desde HTML.....	50
Figura 3. 9 Redirección de páginas con JavaScript	50
Figura 3. 10 Icono de la aplicación	50
Figura 3. 11 Icono de inicio de la Aplicación.....	51
Figura 3. 12 Icono del botón de Localizar	51
Figura 3. 13 Icono del botón de información	51
Figura 3. 14 Icono del botón de ¿Dónde estoy?	51
Figura 3. 15 Icono de contacto.....	51
Figura 3. 16 Iconos de punteros para los mapas	52
Figura 3. 17 Iconos de letreros de rutas.....	52
Figura 3. 18 Cálculo de tiempo de Google Maps	54
Figura 3. 19 Tiempos para llegar a la siguiente parada.....	54
Figura 3. 20 Suma de tiempos.....	54
Figura 4. 1 Interfaz principal de la aplicación.....	56
Figura 4. 2 Logotipo de la aplicación	56
Figura 4. 3 Botón “Información”	57
Figura 4. 4 Información general del servicio PumaBús.....	57
Figura 4. 5 Menú de rutas.....	58
Figura 4. 6 Interfaz de información de ruta 8.....	59
Figura 4. 7 Mapa de ubicación de paradas de la ruta 8.....	59
Figura 4. 8 Paradas ruta 8.....	60
Figura 4. 9 Vista satelital de la ubicación de las paradas de la ruta 8	60
Figura 4. 10 Botón “¿Dónde Estoy?”	61
Figura 4. 11 Ubicación de usuario.....	61
Figura 4. 12 Botón “Localizar”	62
Figura 4. 13 Menú de Rutas	62
Figura 4. 14 Petición de nombre a usuario.....	63

Figura 4. 15 Petición de datos.....	64
Figura 4. 16 Petición de datos completo.....	64
Figura 4. 17 Tiempo Total.....	65
Figura 4. 18 Prueba 1.....	66
Figura 4. 19 Imagen de localización del usuario.....	67
Figura 4. 20 Tiempo de llegada al destino.....	67
Figura 4. 21 Prueba 2.....	68
Figura 4. 22 Ubicación de PumaBús.....	69
Figura 4. 23 Tiempo estimado de llegada.....	69
Figura 4. 24 Prueba 3.....	70
Figura 4. 25 Posición del PumaBús.....	71
Figura 4. 26 Tiempo estimado de prueba 3.....	71
Figura 4. 27 Prueba 4.....	72

Índice de Tablas

Tabla 1. 1 Relación Kilómetros Vueltas de 2013.....	15
Tabla 1. 2 Tabla de usuarios	16
Tabla 1. 3 Diferencias entre Android y Java	29
Tabla 2. 1 Comparación de costos	42

Introducción

En un mundo de desarrollo constante donde la tecnología va avanzando día a día extremadamente rápido, desde la invención del transistor a la fecha, el mundo ha cambiado y revolucionado tecnológicamente. Afortunada o desafortunadamente, dependiendo del punto de vista, la vida se ha vuelto muy activa, muy rápida, y gracias a la tecnología el estilo de vida se ha vuelto más sencillo y más cómodo.

Actualmente los tiempos que se utilizan para realizar las actividades cotidianas son muy cortos, ya que con internet se puede facilitar la realización de cualquier actividad. Por la misma razón, la vida actual se basa en medición de tiempos, en tratar de acelerar todo y realizar las cosas lo más pronto posible. Es entonces cuando se encuentra la dificultad de llegar a algún destino en el menor tiempo posible, o donde a veces se encuentran ciertas adversidades ya que no es tarea fácil la administración del tiempo.

Es importante ser capaz de administrar el tiempo de manera adecuada. Pero en ocasiones el tener un buen manejo del tiempo no depende de la persona sino de factores externos como lo son los medios de transporte, el tráfico de la ciudad, las manifestaciones, entre otras.

Existen distintas rutas para llegar a un lugar, pero es necesario saber identificar que ruta es más conveniente, debido a que no necesariamente la ruta habitual puede ser la más rápida o la más corta, ni tampoco puede ser la óptima en ese momento.

Este problema puede ser llevado a la vida universitaria, si se habla de la UNAM, específicamente el área de Ciudad Universitaria, una región geográfica ubicada en la Ciudad de México, que consta de un área mayor a 700 hectáreas y que está dividida en tres grandes áreas:

Estadio Olímpico, donde la universidad cuenta con estacionamiento para gente perteneciente a la universidad y visitantes, con capacidad de 2 mil 618 vehículos.

Zona Escolar, teniendo como elemento central de esta área la explanada conocida como “las islas”, alrededor de la cual se agrupan los edificios de diversas facultades y escuelas. La zona escolar está subdividida en cinco grandes grupos:

- Gobierno y Sociedad
- Humanidades
- Ciencias Biológicas
- Ciencias
- Artes y Museo

Campos Deportivos, es uno de los espacios de educación superior en el continente americano con mayor número de campos deportivos, tanto de entrenamiento como de exhibición. Esta zona cuenta con campos para la práctica de fútbol, softbol, fútbol

americano, béisbol, basquetbol y tenis, entre otros, a su vez, cuenta con la alberca olímpica, el estadio de prácticas, la pista de calentamiento y los frontones.

Teniendo un panorama amplio acerca de los espacios de Ciudad Universitaria, en algunas ocasiones se requiere que las personas acudan a diferentes facultades o institutos. Para responder a esta y otras problemáticas se creó el sistema de transporte PumaBús. Este sistema consta de camiones que circulan a lo largo de CU por un carril exclusivo de forma gratuita, con 12 rutas diferentes y más de 60 camiones que ayudan a transportarse a lo largo de Ciudad Universitaria, y por otro lado se tiene la opción de poder tomar la BiciPuma, que para ésta la universidad cuenta con 5980 metros de ciclo pista, y 12 módulos ubicados a lo largo del campus de manera estratégica.

Es entonces, cuando se estaría hablando de que las personas que se encuentran dentro del campus puedan tener diversas opciones para poder tomar una ruta óptima o un medio de transporte distinto.

Tomando como opción el PumaBús, se encuentra con que el sistema funciona de manera adecuada pero debido a las eventualidades que ocurren en el transcurso de la ruta y a la cantidad de camiones disponibles no siempre se sabe cuánto tiempo pasará para poder hacer uso del servicio. Esto resulta problemático para tener una administración adecuada del tiempo, que en este momento es muy preciado para toda la sociedad.

Una manera de ayudar a las personas a evitar contratiempos es tener un sistema que se encargue de avisar a los usuarios del transporte público universitario el tiempo de llegada de estos, hasta el punto donde se encuentra el usuario y ser capaz de avisarles de alguna eventualidad en tiempo real, ayudándoles a poder tomar una decisión correcta en cuanto a qué medio de transporte es el más conveniente de utilizar y calcular el tiempo de llegada a su destino.

Este sistema, podría ser observado en un teléfono móvil por su practicidad y economía, específicamente en un Smartphone, ya que gran parte de la población cuenta con uno.

El poder gestionar el tiempo de manera adecuada se ha vuelto un factor clave en la sociedad actual. Es imposible que se tenga un control total de todos los acontecimientos tanto positivos como negativos que ocurren en la vida diaria pero lo cierto es que en la mayoría de los casos dichos acontecimientos repercuten en el tiempo de las personas.

Una manera en la que se puede tener un mejor control del tiempo es tener opciones ante distintas situaciones. Por lo que el tener información para poder tomar decisiones se vuelve de suma importancia. Debido a esto, un sistema que sea capaz de permitir tomar una decisión informada, dará como resultado menor grado de incertidumbre, evitando el azar y ayudando a prever cualquier posible resultado, optimizando los tiempos del usuario.

Objetivo general

Realizar un sistema que brinde la posición de los PumaBús a los usuarios del transporte público en Ciudad Universitaria.

Objetivos particulares

- Analizar la viabilidad de realizar una aplicación para Smartphone que permita ubicar la posición geográfica de un PumaBús en un mapa y un tiempo estimado de llegada.
- Operar la aplicación que con la ayuda de un localizador GPS, proporcione las coordenadas del transporte universitario.
- Diseñar el sistema de localización de los PumaBús
- Gestionar pruebas en los PumaBús de CU.

Para alcanzar los objetivos planteados, se realizaron cuatro capítulos a lo largo del presente documento.

En el capítulo 1, “Marco teórico” se describen conceptos importantes para la realización del sistema, como lenguajes de programación, y sistemas operativos, mencionando ventajas y desventajas de los mismos, además se habla del uso de la telefonía celular, específicamente de los smartphones, para de esta manera, comprender hacia qué público está enfocada la aplicación, también se muestran cifras estadísticas del sistema PumaBús con la finalidad de entender la importancia del sistema de transporte y el porqué es necesaria su optimización. Por último se analizan distintos dispositivos receptores y transmisores para la realización del sistema.

En el capítulo 2, se exponen distintas propuestas que pueden dar solución a la problemática expuesta. Se abordan las ventajas y desventajas de cada una y finalmente se detallan motivos por los cuales se seleccionó la propuesta a implementar.

En el siguiente capítulo, “Diseño y Desarrollo” se reúne la información necesaria para llevar a cabo el análisis de los requerimientos para el desarrollo del proyecto, posteriormente se muestra la planeación del proyecto y al final se presenta la realización de la aplicación tanto lógica como física.

En el cuarto y último capítulo se engloban los resultados del desarrollo de la aplicación. Así como tiempos y forma de ejecución de la misma.

En las conclusiones se hace una breve descripción de los resultados obtenidos, así como los logros adicionales que se tuvieron al momento de llevar a cabo la realización de la aplicación.

Capítulo 1

Marco Teórico

En el marco teórico se abarcan los conceptos importantes para la realización del sistema. Se habla sobre el uso de la telefonía celular, a fin de comprender al público al que va dirigida esta aplicación, así como mostrar estadísticas del sistema de transporte interno PumaBús explicando por qué es importante su realización, y cómo es que pretende ayudar; partiendo de la descripción de distintos sistemas operativos con sus respectivas características, ventajas y desventajas, así como algunos lenguajes de programación que se utilizan para desarrollar aplicaciones móviles. Por último se analizan distintos dispositivos receptores para su realización.

1.1. Mercado de Telefonía Celular en México

El teléfono celular es un avance tecnológico en los medios que puede utilizar la sociedad para trabajar en forma más eficiente y eficaz. La vida de las personas se rige en estar comunicado con el exterior. Además, provee la posibilidad de una comunicación más rápida, cómoda, eficaz y confiable. Las personas pueden comunicarse entre sí sin importar en qué lugar del mundo se encuentren.

Esta necesidad de comunicación constante, ha hecho del teléfono celular uno de los elementos indispensables para las personas, gracias a lo mencionado con anterioridad, además de la comodidad que brinda debido a su practicidad, portabilidad y usabilidad, con el avance de la tecnología, un teléfono celular ya puede traer integradas muchas funcionalidades más, aparte de los usos básico de un teléfono como son las llamadas y los mensajes de texto, permitiéndole ser conocido como Smartphone.

Un Smartphone, puede llegar a tener tantas utilidades, que se puede llegar a olvidar que es un teléfono celular. Dentro de esas utilidades se puede encontrar además de hablar y escribir mensajes, aplicaciones para navegar en la red, jugar, tomar fotos y videos, estar conectado con la sociedad mediante redes sociales, entre muchas otras más que se especificarán a lo largo del presente documento.

En México existen 102.6 millones de suscriptores a teléfonos celulares (hasta junio de 2013), de los cuales solo el 37% son Smartphones y el resto corresponde al mercado de Feature y Basic Phones, según consta en el artículo “Tendencias del Consumidor de Telefonía Móvil en México en 2013” publicado en isopixel.net el 24 de Octubre de 2013. Con base en el mercado de Smartphones, se tiene una variedad de opciones para este tipo de tecnologías ya que permite el desarrollo de aplicaciones donde no sólo existe un sistema operativo, lo anterior es mostrado en la figura 1.1.



Figura 1. 1 Tendencias del consumidor Móvil en México 2013

Para la gente aficionada a estas tecnologías un tema relevante es el de los Sistemas Operativos, pues es donde entra una lucha constante de las empresas desarrolladoras de estas tecnologías para buscar quién es la que logra captar la mayor parte del mercado; cabe mencionar que dentro de los principales competidores destacan el Sistema Operativo iOS, Android, Windows Phone, BlackBerry y Asha.

Basándose en las tendencias de búsqueda de Google Trends durante el 2013, se arroja la gráfica que se muestra en la figura 1.2.

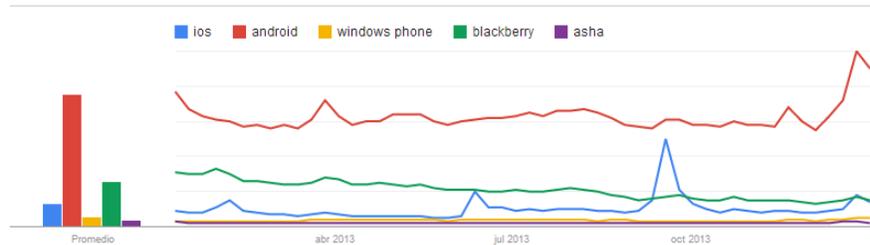


Figura 1. 2 Gráfica de Búsquedas en Google Trends

Como se puede ver en la gráfica anterior, Android arrasó en el año 2013 con un promedio de 81 respecto al resto de los sistemas operativos en México. El segundo lugar es de cierta manera engañoso, ya que como se aprecia en la gráfica en este lugar aparece BlackBerry; sin embargo, si se toma en cuenta que BlackBerry es el nombre de los teléfonos y al mismo tiempo del sistema operativo, sale beneficiado enormemente en la estadística. Se puede entonces tomar iOS como la segunda posición real con un promedio de 13.

En tercer lugar se encuentra Windows Phone con un promedio de 5, y por último se encuentra el sistema operativo Asha, que aunque tiene el mismo inconveniente que BlackBerry en el sentido de que su plataforma y sus teléfonos llevan el mismo nombre, mostraron tener un número lo suficientemente grande en búsquedas figurando en ésta estadística.

Es importante aclarar que las búsquedas en Google no están relacionadas de ninguna manera con las ventas o ingresos que generan las marcas, pero al menos da una idea de los temas en que la gente estuvo interesada por ellas en 2013. Por lo tanto, y por razones mostradas anteriormente, se llega a la conclusión de que el Sistema Operativo más popular en México es Android, y este resultado está respaldado por un estudio realizado por CNN México que lleva por nombre “Android es el sistema operativo más popular en el mundo, según encuesta” publicado el 12 de Noviembre de 2013. Además, con base en el artículo “Con 36.5% del mercado, Android es número uno” de mediatelecom.com publicado el 1 de Octubre de 2013 arrojó como resultados que en México, el 36.5% de los Smartphone cuentan con sistema operativo Android.

1.2. Información estadística del Sistema de Transporte Interno PumaBús

El sistema de transporte interno PumaBús nace como una solución a la alta densidad vehicular que existía en los circuitos de Ciudad Universitaria, ya que circula a lo largo de los circuitos por un carril exclusivo, quedando éstos libres de vehículos estacionados, como solía hacerse antes de que existiera este sistema (véase figura 1.3).



Figura 1. 3 Circuito de Ciudad Universitaria antes del PumaBús

El Sistema de Transporte Interno PumaBús es un servicio gratuito que ofrece la UNAM a toda persona que desee desplazarse dentro del campus de Ciudad Universitaria con facilidad, seguridad y rapidez.

Este sistema de transporte se encuentra en constante crecimiento, ejemplo claro de ello es que en el año 2000 el servicio contaba con 19 camiones para formar la flotilla vehicular, y para el año 2002 se fortaleció el transporte hasta llegar a 35 vehículos ofrecían el servicio, posteriormente en el año 2009 se tenían ya 60 camiones que cubrían las 11 rutas establecidas hasta entonces con 2 paraderos principales: el paradero oriente, situado en la estación del metro CU, de donde parten las rutas 1, 2, 3, 4 y 5; y el paradero poniente, ubicado en el estacionamiento 1 del Estadio Olímpico Universitario (EOU) para las rutas 6, 7 y 8, y un tercer punto de partida con dos paraderos ubicados en ambas salidas de la nueva estación del MetroBús para las rutas 9, 10 y 11. ¹

En la actualidad, el sistema cuenta con 65 unidades, 96 paraderos y traslada a 146,000 usuarios al día. Los datos mostrados en la tablas 1.1 y 1.2 fueron proporcionados por personal del Sistema de Transporte PumaBús.

¹ <http://www.pumabus.unam.mx/> (Última consulta en Marzo de 2014)

Tabla 1. 1 Relación Kilómetros Vueltas de 2013

<i>Mes</i>	<i>Kilómetros recorridos</i>	<i>Numero de Vueltas</i>
<i>Enero</i>	179,481	24,928
<i>Febrero</i>	196,992	27,360
<i>Marzo</i>	141,696	19,680
<i>Abril</i>	248,371	34,496
<i>Mayo</i>	232,243	32,256
<i>Junio</i>	198,144	27,520
<i>Julio</i>	66,355	9,216
<i>Agosto</i>	197,704	27,456
<i>Septiembre</i>	184,320	25,600
<i>Octubre</i>	217,267	30,176
<i>Noviembre</i>	175,104	24,320
<i>Diciembre</i>	80,870	11,232

Estos datos permiten ver la utilidad que se le da al sistema de transporte, además muestra la importancia que ha tomado este sistema y la cantidad de usuarios que tiene al día.

Tabla 1. 2 Tabla de usuarios

<i>Ruta</i>	<i>Usuarios por día</i>	<i>Paraderos</i>	<i>Longitud de ruta</i>
1	16.50%	16	7.2
2	17.00%	11	4.2
3	9.00%	20	8.8
4	3.50%	15	9.7
5	13.00%	18	8.3
6	3.50%	24	10.2
7	11.00%	17	4.8
8	10.50%	20	5.8
9	14.00%	15	4.1
10	1.30%	18	11
11	0.30%	14	5.8
12	0.04%	7—8	5.1

1.3. Sistemas operativos para dispositivos móviles.

Un sistema operativo es una capa de software que se encarga de proporcionar a los programas de usuario un modelo de computadora más simple.

Las funciones de un sistema operativo pueden ser divididas en Abstracción y Administración de Recursos.

a) Abstracción.

Una abstracción es un proceso donde se ignora una parte de un todo, con el fin de facilitar la comprensión. Se encarga de solo tomar en cuenta lo que es importante desde la perspectiva que se está utilizando. Algunos ejemplos de abstracción son:

- Abstracción de control: Las estructuras de control se encargan de controlar el flujo de la ejecución en un programa.

- Abstracción de datos: Las estructuras de datos es una forma de almacenar y organizar los datos.
- Archivo: Es una pieza útil de información que es más sencilla de manipular que manipular directamente datos en memoria o en algún medio de almacenamiento.

Una de las principales tareas de un sistema operativo es ocultar el hardware y mostrar una abstracción agradable. Esto es muy importante para poder lidiar con la complejidad, ya que se encarga de convertir tareas muy complicadas en tareas simples. El trabajo del sistema operativo es crear buenas abstracciones.

El crear una abstracción consta de dos partes:

- Definir y crear la abstracción
- Implementar la abstracción

b) Administración de recursos

Un sistema operativo se encuentra para administrar los recursos de hardware y de esta manera proporcionar una asignación ordenada y controlada de los procesadores, memorias y dispositivos de E/S. Debe ser capaz de llevar un registro de que el programa se encuentra utilizando algún recurso.

En la actualidad los sistemas operativos deben ser capaces de realizar multiplexaje de recursos en tiempo y en espacio. El multiplexaje en tiempo es que los distintos usuarios o programas toman turnos para poder acceder a los recursos. El multiplexaje en espacio, es que el recurso que se comparte entre los programas o usuarios que lo necesiten.

1.3.1. Android



Figura 1. 4 Logotipo Android

Android es un sistema basado en Linux para dispositivos móviles, el código de Android es libre y se encuentra bajo licencia de software libre. Fue desarrollado inicialmente por Android Inc. Y fue comprado en 2005 por Google. Pertenece a la OHA (Open Handset Alliance) y es el principal producto de esta.

En el tercer cuarto del 2013, cuatro de cada cinco Smartphones ya contaban con el sistema operativo Android. Actualmente se encuentra en su versión 6.0 Marshmallow y su logo se puede observar en la figura 1.4.

Algunas de sus características más sobresalientes son:

- Interfaz de usuario intuitiva y elegante
- Conectividad con distintos protocolos de comunicación como Bluetooth o Wi-Fi
- Soporte Multimedia en distintos formatos
- Navegador web con soporte para nuevas tecnologías como HTML 5
- Muti-Task que es la capacidad de que varias aplicaciones pueden ser ejecutadas simultáneamente
- Multilenguaje

1.3.1.1. Arquitectura de Android

El sistema operativo Android es una pila de software que se puede dividir en 4 capas las que se observan en la figura 1.5. Cada capa se encarga de brindar funciones específicas.

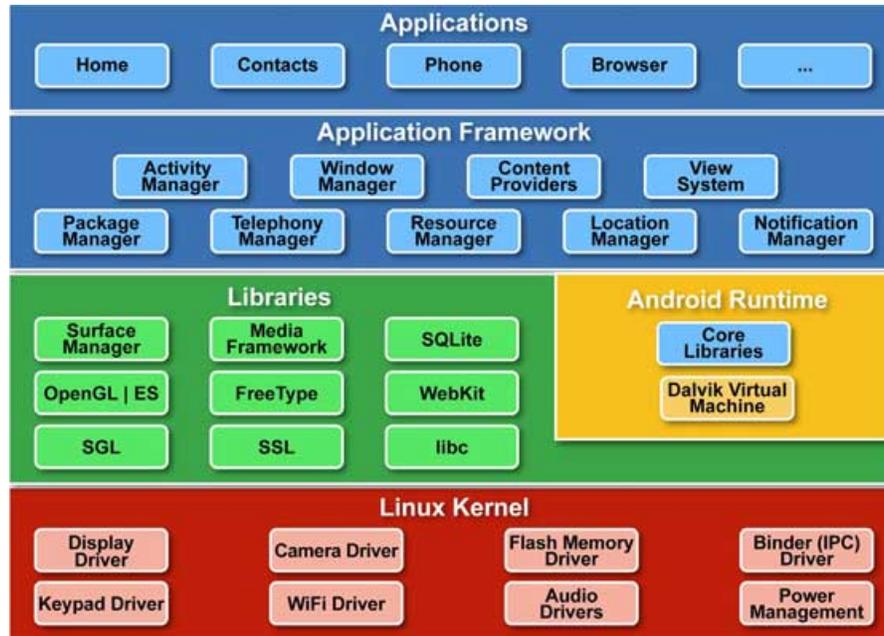


Figura 1. 5 Estructura de Sistema Operativo Android

1.3.1.2. Kernel de Linux

La base de Android es el Kernel de Linux, este es el encargado de que el software y el hardware puedan trabajar en conjunto. Android utiliza el Kernel Linux 2.6 con varios parches para añadir características y corregir fallos en programación.

Las funciones más importantes de esta capa son:

- Administración de Memoria para los programas y procesos en ejecución
- Administración de tiempo de Procesador que es utilizado por los programas y procesos que se están ejecutando.
- Manejo de Dispositivos.

El kernel de Linux es el que interactúa directamente con el hardware y permite la interacción con los dispositivos de E/S.

En el nivel más bajo contiene manejadores de interrupciones que son los encargados de interactuar con el despachador.

En un nivel superior encontramos tres componentes:

- Componentes de E/S

- Componentes de Administración de la Memoria
- Componentes de administración de los procesos

En la parte superior están las llamadas al sistema, que pasan el control a uno de los componentes del Kernel.

1.3.1.3. Bibliotecas

Sobre del Kernel de Linux existen bibliotecas. Las bibliotecas son una serie de implementaciones de comportamiento, éstas, proporcionan funcionalidad a las aplicaciones para tareas que se repiten con frecuencia, evitando codificarlas cada vez y haciendo de esta forma más eficiente el proceso. Algunos ejemplos importantes son:

- Apache HTTP: Es un servidor web de código abierto
- Harmony subset
- OpenGL ES: Es una API que se utiliza para crear aplicaciones 2-d y 3-d
- SQLite: Es una biblioteca que implementa un motor de base de datos
- Bouncy Castle: Es una API de criptografía utilizada por java y C#
- Open SSL: Un conjunto de herramientas utilizadas para implementar SSL
- SAX: es una API para XML
- Zlib: Biblioteca de compresión de datos
- FreeType: Biblioteca para el uso de distintas fuentes
- SGL: biblioteca de 2D

1.3.1.4. Entorno de Ejecución

En este mismo nivel se encuentra “Android runtime” (Entorno de ejecución), este se encuentra en el mismo nivel y no en uno separado debido a que también se encuentra formado por bibliotecas. Aquí se encuentran las bibliotecas con funcionalidades de Java así como otras específicas de Android.

La parte de principal del sistema operativo Android es la máquina virtual Dalvik, que se encarga de interpretar los archivos en formato Dalvik Executable (.dex). El formato Dex esta optimizado para el almacenamiento eficiente y ejecución correcta en memoria. Dalvik permite que el código sea compilado a nivel bytecode independiente de la máquina virtual en la que se va a ejecutar, para después interpretar este bytecode en el momento de ejecutar el programa. En la figura 1.6 se aprecia este proceso.

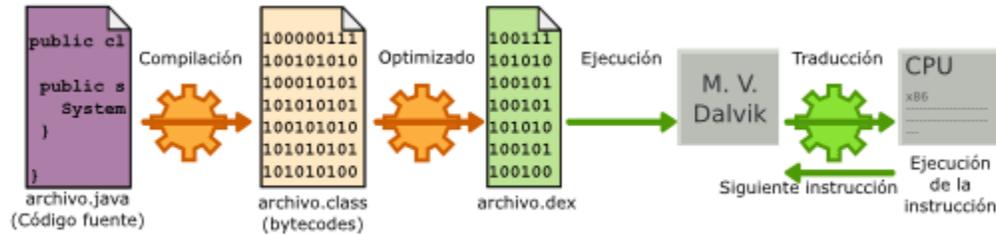


Figura 1. 6 Proceso de runtime de Android

Aunque Dalvik es muy parecido a la Máquina virtual de Java, cuenta con optimizaciones orientadas a dispositivos móviles. Algunas diferencias importantes son:

- Las bibliotecas de java en Android tiene ligeras diferencias con Java SE y Java ME
- Dalvik utiliza registros a diferencia de la máquina de java que utiliza pilas
- El recolector de basura de Java se optimiza en Dalvik

Java se usa únicamente como lenguaje de programación, y los ejecutables que se generan con el SDK de Android tienen la extensión *.dex. Esta es la razón por la que no se pueden correr aplicaciones Java en Android.

1.3.1.5. Framework de aplicaciones

Se encarga de brindar servicios mediante clases de Java, la mayoría de los componentes de esta capa accede a los recursos de la capa anterior mediante la Máquina virtual Dalvik. Algunos ejemplos de elementos de esta capa son:

- Manejador de actividades
- Manejador de ventanas
- Vistas
- Manejador de notificaciones
- Manejador de paquetes
- Manejador de telefonía
- Manejador de recursos
- Cámara
- Multimedia

1.3.1.6. Aplicaciones

La última capa incluye las aplicaciones, tanto nativas como administradas. Cualquier aplicación que se desarrolla corre en esta capa, por ejemplo: juegos, contactos, libros, por mencionar algunos. La aplicación principal del sistema Inicio o launcher se encuentra en esta capa, esta es la aplicación que permite ejecutar otras aplicaciones.

1.3.1.7. Android SDK

El SDK (Software Development Kit) es un conjunto de herramientas de desarrollo, utilizadas para crear aplicaciones de Android. En el SDK se incluyen:

- Bibliotecas necesarias
- Debugger
- Emulador
- Documentación de la API
- Códigos de ejemplo
- Tutoriales para Programar aplicaciones en Android

Conforme se liberan las nuevas versiones de Android, también se liberan nuevas versiones del SDK con el fin de brindar herramientas más poderosas a los desarrolladores. Debido a que las aplicaciones Android se escriben en código Java es necesario tener el JDK de Java.

1.3.2. iOS

Es un sistema operativo de Apple Inc. Fue lanzado en el 2007 para los dispositivos iPhone y Ipod touch. La versión actual de este sistema operativo es la 9.0.2 cuyo logo se puede observar en la figura 1.7. Para desarrollar y colocar las aplicaciones desarrolladas en la App Store se necesita estar registrado con el programa de desarrollo de iOS el cual tiene un costo de 99 dólares y contar con una computadora Mac con sistema operativo Mountain Lion o superior.



Figura 1. 7 Logotipo de iOS versión 9

1.3.2.1. Arquitectura iOS

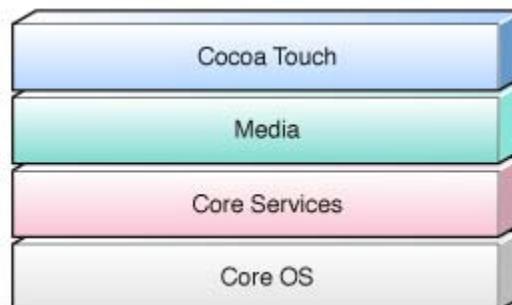


Figura 1. 8 Arquitectura del Sistema Operativo IOS

La arquitectura del sistema operativo iOS está basado en capas. Consta de 4 capas (Figura 1.8):

- Cocoa Touch
- Media
- Core Services
- Core OS

1.3.2.1.1. Core OS

Esta capa contiene las características de bajo nivel sobre las cuales se construyen todas las demás. Las ventajas de utilizar estas características y no desarrollar las propias es que se encuentran optimizadas para las configuraciones de hardware. Esto es un beneficio para los desarrolladores ya que pueden escribir código sin preocuparse por estas interacciones. La capa se divide a su vez en “frameworks” que ayudan al desarrollo:

- Accelerate Framework
 - o Es el encargado del procesamiento de señales digitales, álgebra lineal, procesamiento de imágenes y cálculos.
- Core Bluetooth Framework
 - o Ayuda a interactuar con la interfaz de Bluetooth usando la menor cantidad de energía posible.
- External Accesory Framework
 - o Se encarga de administrar hardware externo que es conectado al dispositivo.
- Generic Security Services Framework
 - o Es el encargado del manejo de credenciales
- Security Framework
 - o Es utilizado para brindar distintos tipo de seguridad a los datos utilizados en las aplicaciones.
- System
 - o Aquí es donde se encuentra el Kernel y las interfaces de bajo nivel del sistema operativo UNIX. El kernel es el encargado de manejar la memoria virtual del sistema, los hilos, el sistema de archivos, la red y la comunicación entre los procesos internos.
 - o Al principio iOS solo aceptaba la arquitectura de 32 bits pero desde la actualización a su versión 7 se incorporó la arquitectura de 64 bits.

1.3.2.1.2. Core Service Layer

Esta capa contiene piezas fundamentales para el desarrollo de aplicaciones ya que es donde está definido el tipo básico de aplicación. En esta capa también se incluyen características que pueden ser utilizadas por las aplicaciones, algunos ejemplos de estas tecnologías son:

- iCloud storage
- Peer to Peer services
- Automatic reference Counting
- Block Objects
- Data protection
- SQLite
- XML Support

1.3.2.1.3. Media

Esta capa se encarga del contenido de multimedia como gráficas, audio, video, y ayuda a que sea más fácil el construir apps con contenido multimedia ya que se encarga de manejar este contenido y optimizar su uso. Cuenta con distintas tecnologías y que a su vez se subdividen en distintos componentes. Las tecnologías con las que cuenta la capa de media son:

- Tecnología Gráfica
- Tecnología de Audio
- Tecnología de video
- AirPlay

En esta capa también se cuenta con distintos frameworks que ayudan al desarrollo de las aplicaciones.

1.3.2.1.4. COCOA

Esta capa es donde se define la apariencia de la aplicación que se va a realizar. Se provee a la aplicación de tecnologías básicas como lo son el Touch o el multitasking. Los frameworks de esta aplicación están orientados al desarrollo de aplicaciones.

Esta capa está formada por dos Frameworks fundamentales:

- UIKit: contiene las clases necesarias para el desarrollo de una interfaz de usuario.
- Foundation Framework: define las clases básicas, acceso y manejo de objetos.

El desarrollo de aplicaciones en iOS se hace en el lenguaje de programación Objective C, el cual es un lenguaje de programación orientado a objetos y requiere que la interfaz e implementación de una clase estén en bloques de código separados.

1.3.3. Windows Phone

Es un sistema operativo para Smartphones desarrollado por Microsoft que sucedió a Windows mobile. Actualmente se encuentra en su versión Windows Phone 8 con la tercera actualización y utiliza como Kernel Windows NT que a su vez es utilizado para ejecutar el Sistema operativo Windows 8 en ambientes no móviles.

La arquitectura Windows NT se basa en dos capas principales la capa de modo usuario y la capa de modo Kernel. A su vez estas capas se subdividen en distintos componentes como se observa en la figura 1.9.

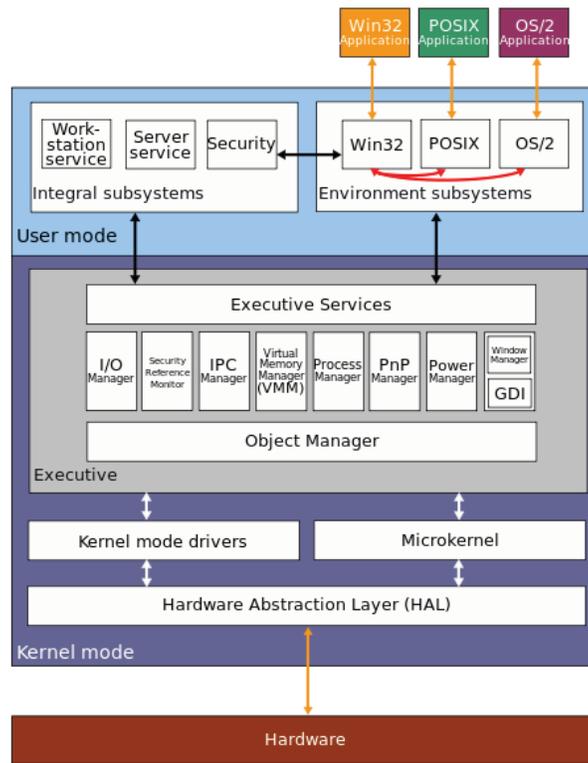


Figura 1. 9 Arquitectura Windows NT

1.3.3.1. Modo Usuario

Este es un subsistema que se encarga de ejecutar las aplicaciones, pero no tiene acceso directo al hardware del sistema. Para acceder a este debe realizar una petición a la capa de Kernel. Algunos ejemplos son peticiones de I/O que deben ser pasadas al Manejador de I/O que se encuentra en la capa de Kernel y el acceso a la memoria es designado por el Manejador de memoria virtual.

El modo usuario se puede subdividir en 2 capas más:

1. El subsistema integral se encarga de funciones específicas como lo son la seguridad del sistema y lo referente a redes del sistema.
2. Los subsistemas de ambiente fueron desarrollados para poder ejecutar aplicaciones para distintos sistemas operativos. Los tres principales son: WIN32, OS/2 y POSIX.

1.3.3.2. Modo Kernel

En la capa de Kernel se tiene acceso a todos los recursos del sistema. Es el encargado de los controles de acceso, el asignar prioridad a los hilos, el manejo de memoria y la interacción con el hardware. Su principal objetivo es el prevenir el acceso áreas críticas del sistema por parte del modo usuario, es por esta misma razón que el modo usuario debe pedir el uso de recursos al modo kernel.

1.4. Lenguajes de programación

1.4.1. Android Java

Java es un lenguaje de programación orientado a objetos que fue desarrollado por Sun Microsystems en 1995 y en la actualidad le pertenece a ORACLE. El logo de Java es una taza de café (Figura 1.10) y la versión actual es Java Standard Edition 8.

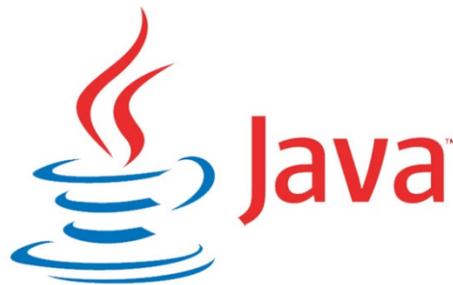


Figura 1. 10 Logotipo de Java

Dentro de las características más importantes de java se encuentran el que es un lenguaje que está basado en el paradigma de programación orientado a objetos lo que le permite hacer uso de los siguientes conceptos:

- Polimorfismo: Brinda la posibilidad de que un identificador puede tener distintas formas dependiendo del contexto de estas.
- Herencia: Es la relación entre una clase general y una más específica, esto permite compartir automáticamente métodos y atributos.
- Encapsulamiento: Buscar organizar los atributos y métodos, evitando el acceso a datos por cualquier otro medio.
- Abstracción: Es separar los elementos de un objeto para poder analizar el objeto en su pura esencia.
- Clases: Es la estructura del objeto, representa a un grupo de objetos que comparten una misma estructura.
- Objetos: Es una entidad con un conjunto de atributos y métodos que tiene la posibilidad de relacionarse con otros objetos.
- Instancias: es la creación de un objeto tomando como referencia la clase a la que pertenece.
- Métodos: Son el comportamiento del objeto
- “Message parsing”: Es la habilidad de pasar mensajes entre objetos.

Está diseñado para ser un lenguaje sencillo de entender y portable, esto último se refiere a que el compilador de Java genera un archivo obj que es independiente de la

arquitectura que se está utilizando lo que permite que este pueda ser ejecutado en distintos sistemas dependiendo solo del Java Runtime system.

Java tiene 5 principios sobre los cuales está pensado el lenguaje de programación. Y estos son que Java debe ser:

- Un lenguaje simple orientado a objetos
- Seguro y robusto
- De arquitectura neutra y portable
- De alto desempeño
- Interpretado, dinámico y capaz de usar Hilos

Diferencias entre Android y Java

Java es un lenguaje de programación mientras que Android es un sistema operativo. La relación que comparten es que en la mayoría de los casos el desarrollo para Android se hace en Java. Aun así existen diferencias entre el desarrollo de una aplicación en Java y una en Android, por ejemplo las aplicaciones en Android no tienen una función Main sino que tienen las funciones onCreate, onResume, onPause, onDestroy que deben ser sobrescritas. También es importante destacar que el código Java se compila en Java bytecode mientras el Android se compila en código objeto de Dalvik.

Es importante el Android Manifest y los componentes esenciales de una aplicación en Android. El Android Manifest es un archivo XML donde se colocan las configuraciones básicas de la aplicación como los permisos de la aplicación, las actividades y los servicios por mencionar algunos. Las aplicaciones constan de 4 componentes principales:

- **Actividades:** Es el componente encargado de la interfaz del usuario y la interacción que este tiene con el Smartphone
- **Servicios:** son los encargados de correr procesos en segundo plano que utiliza la aplicación.
- **Broadcast Receivers:** Es el encargado de la comunicación entre la aplicación y el sistema operativo Android.

Proveedores de contenido: se encarga del manejo de los datos.

Tabla 1. 3 Diferencias entre Android y Java

JAVA	ANDROID
Lenguaje de programación orientado a objetos	Es un Sistema operativo basado en el Kernel de Linux

Inicialmente desarrollado por Sun Microsystems, después adquirido por Oracle.	Desarrollado por Google
Es multiplataforma	Se ejecuta sobre un Kernel monolítico (Una modificación del Kernel de linux)
Requiere de la máquina virtual de Java para realizar la compilación	Utiliza la máquina virtual Dalvik; especializada para compilar programas en ambiente móviles.

1.4.2. Objective C

Es un lenguaje de programación orientado a objetos y basado en mensajes, altamente dinámico. Fue creado como una extensión de C añadiéndole: clases, objetos, métodos, protocolos, excepciones, propiedades y categorías. Este lenguaje consiste en un pequeño número de incrementos a ANSI C, y está caracterizado por su aplazamiento a muchas decisiones hasta que se cumple el tiempo de corrida del programa creado, soportando envíos dinámicos, escritura y carga dinámica.

Estas características soportan muchos de los patrones de diseño que utiliza Cocoa, incluyendo la delegación, notificación, y Modelo Vista Controlador. Debido a que es una extensión del lenguaje C, y al contener código y librerías de C, incluso aquellas que están basadas en C++, puede trabajar con aplicaciones basadas en Cocoa, sin perder nada del desarrollo original.

Originalmente fue creado por Brad Cox y la corporación Stepstone en 1980. En 1988 fue adoptado como lenguaje de programación de NEXTSTEP y en 1992 fue liberado bajo licencia GPL para el compilador GCC (GNU Compiler Collection).

Actualmente es el principal lenguaje utilizado para el desarrollo de aplicaciones en iOS y OS X. Su estructura está basada en:

- Objetos
- Escritura dinámica
- Escritura estática
- Mensajes
- Estructura de un mensaje
- Clases
- Definición de las clases
- Categorías
- Protocolos
- Tiempo de corrida

1.4.3. HTML 5

No es un lenguaje de programación, es un sistema de etiquetas que comunican al Browser que información se desplegará en pantalla. El significado de sus siglas es HyperText Markup Language, versión 5, es la actualización del lenguaje básico de la World Wide Web, HTML.

HTML5 proporciona mecanismos para simplificar el trabajo y facilitar la incorporación de elementos multimedia. Además tiene la capacidad de admitir diferentes sintaxis; HTML y XHTML.

HTML5 establece una serie de nuevos elementos y atributos que hacen posible la visualización de los sitios web modernos. Algunos de ellos son técnicamente similares a la versión antigua de HTML. Otros elementos proporcionan nuevas funcionalidades a través de una interfaz estandarizada; existen otros que mejoran el elemento, capaz de renderizar elementos 3D en los navegadores más usados (Mozilla, Chrome, Opera, Safari e IE).

Algo importante de este lenguaje es que existe un renovado énfasis en la importancia del scripting DOM (Documento Object Model) para el impulso de la web. 2.0

1.5. Dispositivo Receptor.

1.5.1. Arduino.

Es una plataforma electrónica abierta para la creación de prototipos basada en software y hardware flexibles y fáciles de usar. Se creó para artistas, diseñadores, aficionados y cualquier persona interesada en crear entornos u objetos interactivos, se distingue por el logotipo mostrado en la figura 1.11.



Figura 1. 11 Logotipo de Arduino

Arduino puede tomar información del entorno a través de sus pines de entrada de toda una gama de sensores y puede afectar aquello que le rodea controlando luces, motores y otros actuadores. El microcontrolador en la placa Arduino se programa mediante el lenguaje de programación Arduino (basado en Wiring) y el entorno de desarrollo Arduino (basado en Processing). Los proyectos hechos con Arduino pueden ejecutarse sin necesidad de conectar a un ordenador, si bien tienen la posibilidad de hacerlo y comunicar con diferentes tipos de software como Flash, Processing, MaxMSP.

Las placas pueden ser hechas a mano o compradas montadas de fábrica; el software puede ser descargado de forma gratuita. Los ficheros de diseño de referencia están disponibles bajo una licencia abierta, por lo que se es libre para adaptarlo a las propias necesidades.

Existen diversos tipos de placas Arduino, aproximadamente 20 placas registradas en la página oficial de Arduino, consultada el 24 de Febrero de 2014, que oscilan de entre los 20 euros hasta los 189 euros, dependiendo de la necesidad del usuario. Desafortunadamente, no existe ninguna placa con uso de GPS, por lo que si se quiere hacer el uso de Arduino como GPS, se le tienen que añadir otros dispositivos para su correcto funcionamiento.

Para el funcionamiento de Arduino como GPS, se le tiene que agregar un módulo llamado, “Parallax GPS module” y hacer las debidas conexiones hacia una pantalla LCD que también se le agrega como se muestra en la figura 1.12.

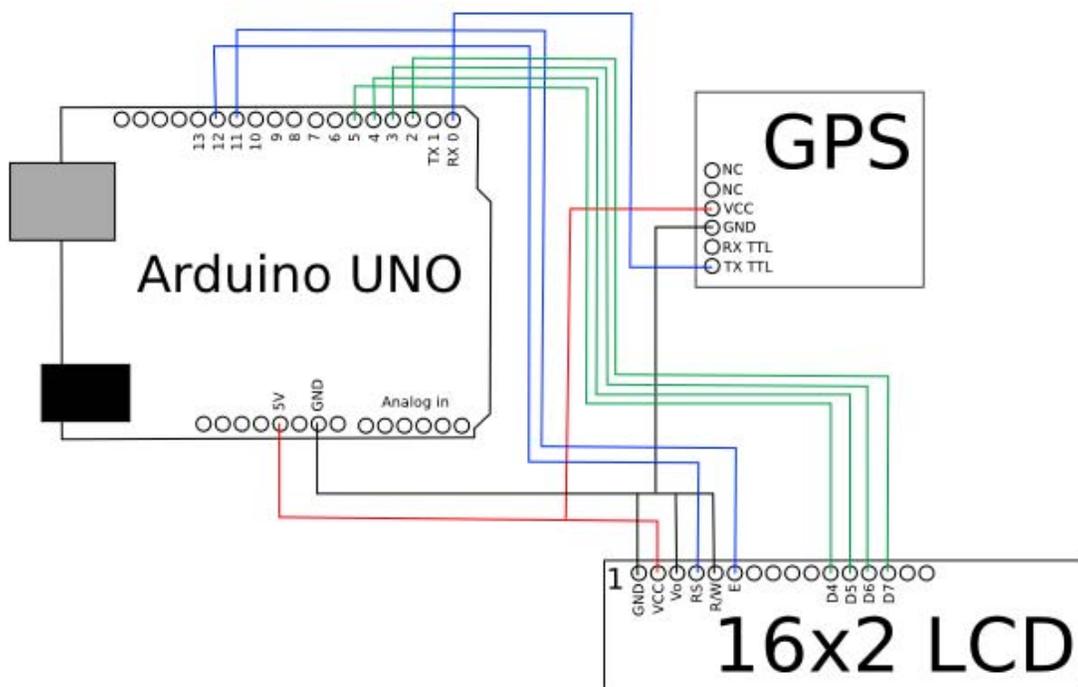


Figura 1. 12 Esquema de integración de GPS con Arduino

Al tener las conexiones anteriores hechas correctamente, lo que hace Arduino es regresar un par de número decimales, que son su latitud y longitud, siendo estos últimos su localización geográfica.

1.5.2. Raspberry Pi

Es una placa computadora de bajo costo desarrollada en Reino Unido por la Fundación Raspberry Pi, con el objetivo de estimular la enseñanza de ciencias de la computación en las escuelas.

El diseño de la Raspberry Pi incluye un System-on-a-chip Broadcom BCM2835, que contiene un procesador central (CPU) ARM1176JZF-S a 700 MHz, un procesador gráfico (GPU) VideoCore IV, y 512 MB de memoria RAM. El diseño no incluye un disco duro o una unidad de estado sólido, ya que usa una tarjeta SD para el almacenamiento permanente; tampoco incluye fuente de alimentación o carcasa. El modelo B se vende a 40\$ dólares y el modelo A 25\$ dólares, de acuerdo a la página oficial de Raspberry PI consultada el 24 de Febrero de 2014.

Desafortunadamente, este sistema no cuenta con un módulo oficial para obtener la localización geográfica del dispositivo, sin embargo, existen diversos, módulos de empresas externas como Cooking Hacks o Xataka Smart Home. Se elegirá el módulo de Cooking Hack debido a las razones que se muestran posteriormente.

El módulo que ofrece la compañía es llamado GSM/GPRS SIM900. Es una placa que se conecta directamente encima de la Raspberry Pi (de ahí su nombre de escudo) en sus conexiones GPIO, y los distintos módulos que se conectan se apilan encima de ella. La ventaja es que pueden además usarse los GPIO de la Raspberry Pi, ya que el escudo replica dichas conexiones, aunado a que su costo es aproximadamente de 11,50 euros, según la página del fabricante consultada el 24 de Febrero de 2014.

Al hacer las conexiones correctas, el dispositivo muestra su posicionamiento geográfico

1.5.3. Smartphone

En la actualidad un Smartphone es la combinación de un PDA con un celular. Un dispositivo es considerado un celular cuando es capaz de comunicarse a una red inalámbrica mediante radiofrecuencia o comunicación satelital. Además la mayoría cuenta con comunicación por voz, envío de mensajes cortos (SMS), envío de mensajes por voz (MMS) y en los casos más nuevos algunos servicios de internet. Por otra parte una PDA es una computadora de mano que brinda la posibilidad de realizar distintas tareas como crear documentos, jugar, revisar correo electrónico y navegar por internet entre otras.

Los Smartphones además de tener características de los PDA y los celulares tienen otras características importantes como lo son los Sistemas Operativos que permiten que se corran aplicaciones. Algunos ejemplos de sistemas operativos son iOS y Android. Las Aplicaciones son las que le dan vida al Smartphone al permitir al usuario realizar distintas tareas.

Capítulo 2

Soluciones Propuestas

Se exponen distintas propuestas que pueden dar solución al traslado de los usuarios en PumaBús, siendo esta la problemática expuesta, así como sus ventajas y desventajas. Todo esto con la finalidad de brindar los motivos por los cuales se seleccionó la propuesta a implementar.

2.1. Propuesta 1. Pantallas de localización satelital

Una manera de solucionar la problemática expuesta es la creación e implementación de pantallas que muestren la última parada donde estuvo el PumaBús, estas pantallas estarían ubicadas en las Paradas del sistema de transporte universitario.

La complejidad de lo que se mostraría en estas pantallas, está ligada a la infraestructura que se quiera implementar, ya sea desde un simple letrero diciendo el número de camión y en qué parada se encuentra, como se ve en la imagen 2.1, hasta un mapa donde muestre donde se encuentra el PumaBús más próximo y un tiempo estimado de llegada, como se muestra en la figura 2.2.

	Parada 1	Parada 2	Parada 3	Parada 4	Parada 5	Parada 6	Parada 7	Parada 8	Parada 9	Parada 10
Camión 1										
Camión 2										
Camión 3										

Figura 2. 1 Pantalla 1

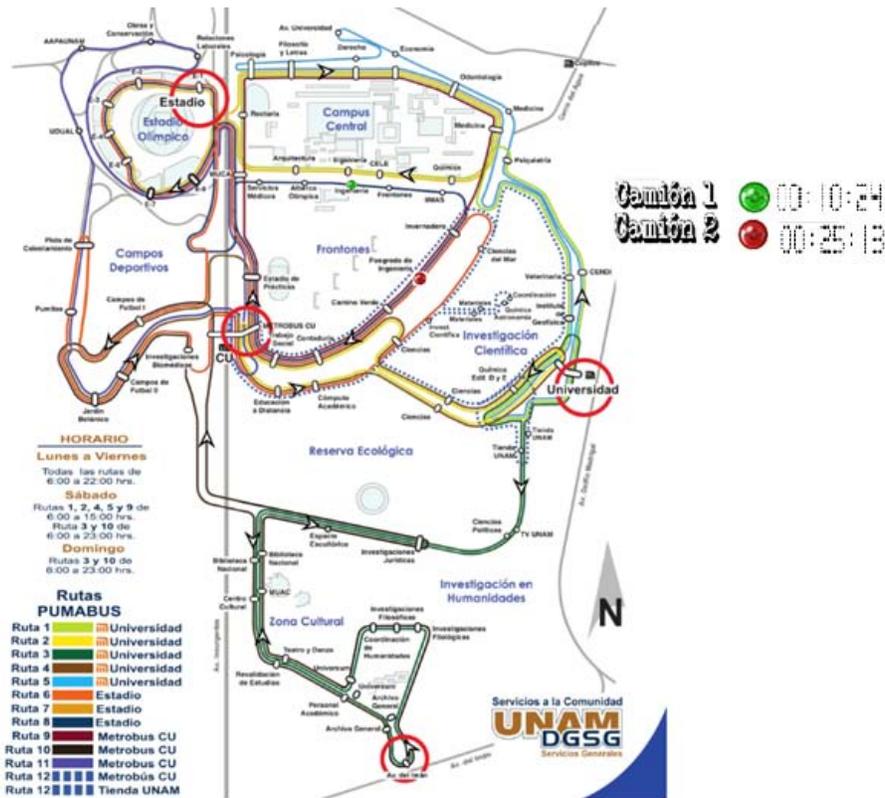


Figura 2. 2 Pantalla 2

Lo anterior, pensando en algo básico pero que cumpla con el propósito de dar información al usuario sobre la localización de los camiones del PumaBús y con esto brindarle argumentos para una toma de decisiones.

Dicha alternativa es una implementación que requiere poca infraestructura por parada y su implementación consiste en colocar un sistema que, apoyándose en la arquitectura cliente-servidor, desde un dispositivo (mencionados en el capítulo anterior), envíe la posición del PumaBús vía satelital a un servidor donde se procese y se reenvíe a las pantallas que tengan integrado un dispositivo de este estilo, de tal manera que la información se vea reflejada en la pantalla iluminando con LEDS la posición de cada camión. Una idea general del funcionamiento se puede apreciar en la figura 2.3.



Figura 2.3 Esquema general primera alternativa

2.1.1. Requisitos

1. Dispositivo para enviar información de localización PumaBús
2. Servidor para procesar la información recibida y reenviarla
3. Creación de pantallas
4. Microcontrolador para hacer uso de las pantallas.
5. Leds

2.1.2. Ventajas

El principal beneficio que se obtendría por este método es que todos los usuarios del servicio de transporte interno PumaBús, podrán tener acceso a la información de la ubicación, ya que se mostraría en cada parada y de esta manera los usuarios estarían en condiciones de tomar decisiones con más argumentos sobre su trayectoria.

2.1.3. Desventajas

La principal desventaja de esta propuesta es el costo que se tendría en infraestructura, puesto que cada estación debería contar con infraestructura propia capaz de recibir información y desplegarla. Otra desventaja importante es que la infraestructura no es susceptible a ser modificada con facilidad, lo que provoca que cualquier modificación realizada al sistema ya sea en rutas o en número de unidades disponibles para dar el servicio representa un cambio en la infraestructura de este sistema.

Los costos estimados para la realización de las pantallas para desplegar diseño fueron los siguientes:

- 2400 dólares por metro cuadrado 10 milímetros de separación por Led.
- 2800 dólares por metro cuadrado 8 milímetros de separación por Led.

Estos costos fueron obtenidos de la empresa Codelsa, empresa encargada del diseño de pantallas de Leds para exteriores. Se debe tomar en cuenta que debería existir una pantalla por parada del PumaBús; esto significaría tener 96 pantallas para que el sistema funcione en su totalidad; a su vez se debe considerar el costo del servidor, el del Smartphone y de las modificaciones de la infraestructura de red actual.

2.2. Propuesta 2. Pantallas de localización mediante sensores

Una solución alternativa a dicha problemática es nuevamente la creación e implementación de pantallas que muestren la última parada en la que fue ubicada una unidad del PumaBús, instalando dichas pantallas en las paradas del mismo.

Lo que se pueda mostrar en estas pantallas va de acuerdo a la necesidad de infraestructura que se desee implementar, se podría agregar una simple tabla con relación Camión-Parada como en la figura 2.1, o una pantalla más sofisticada como en la figura 2.2.

Esta alternativa sigue teniendo el propósito de dar información al usuario sobre la localización de los camiones del PumaBús y así brindarle información para que tenga argumentos para hacer una buena toma de decisión.

La implementación requiere de más infraestructura por cada parada ya que la alternativa funciona colocando un sistema de sensores RFID en las paradas de los camiones PumaBús y apoyándose en la arquitectura cliente-servidor junto con un microcontrolador como el de los dispositivos mencionados en el capítulo anterior, de manera que se envíe la posición del PumaBús vía satelital a un servidor donde se procesaría y se reenviaría a las pantallas para que fuese mostrada la información. El funcionamiento de esta alternativa se puede apreciar en la figura 2.4.

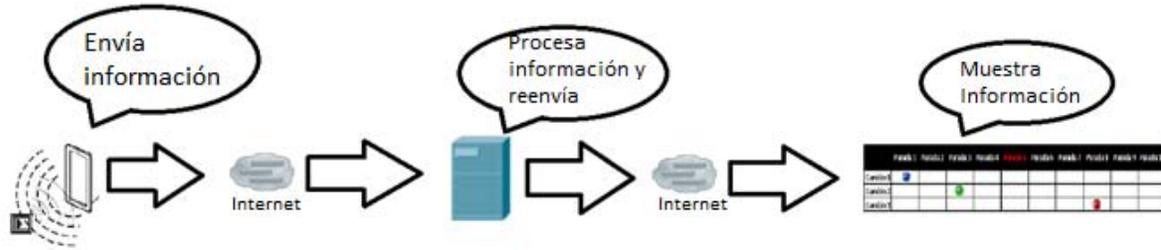


Figura 2. 4 Esquema general de la segunda alternativa

2.2.1. Requisitos

1. Dispositivo para enviar información de localización PumaBús
2. Servidor para procesar la información recibida y reenviarla
3. Creación de pantallas
4. Microcontrolador para hacer uso de las pantallas.
5. Leds
6. Sistema RFID

2.2.2. Ventajas

El beneficio más grande de esta propuesta es que, todo público ubicado dentro del campus tendría acceso a la información debido a la ubicación de las pantallas en cada parada del PumaBús y, de esta manera tomar decisiones con más argumentos sobre su trayectoria. Otra gran ventaja es que no necesita de un sistema de localización satelital.

2.2.3. Desventajas

La principal desventaja de esta propuesta es el costo que se tendría en infraestructura, ya que cada estación necesita contar con su propia infraestructura capaz de detectar el paso de las unidades del sistema de transporte. Otra desventaja importante y ya mencionada en la propuesta anterior es la dificultad de cambios en cuanto a infraestructura.

Al igual que en la propuesta anterior la cotización de las pantallas es la misma:

- 2400 dólares por metro cuadrado 10 milímetros de separación por Led.
- 2800 dólares por metro cuadrado 8 milímetros de separación por Led.

Para esta propuesta se agrega un sensor RFID el cual debe considerarse que tendrá un costo de instalación tanto en las paradas como en las Unidades lo que hace que este incremente su costo. El costo estimado es de:

- 4,500 \$ por la instalación del módulo lector de RFID
- 300 \$ por las etiquetas RFID para las unidades.

2.3. Propuesta 3. Uso de SmartPhones como dispositivos receptores y emisores

Haciendo uso de Smartphones como dispositivos receptores y emisores es posible conocer la localización del PumaBús en tiempo Real, de esta manera los usuarios estarían enterados de la ubicación del mismo y tendrían la posibilidad de evaluar si les resulta más conveniente hacer uso del sistema de transporte o buscar otra manera de llegar a su destino. La propuesta se basa en el crecimiento que ha tenido el mercado de Smartphones en México como se mencionó en el capítulo 1.

Al aprovechar sus capacidades tanto de envío de datos como de localización se puede implementar un sistema que se valga de estas capacidades para beneficiar a los usuarios. En este sistema sería posible ver digitalmente la ubicación de los camiones más cercanos junto con su tiempo estimado de llegada.

La vista de la aplicación que se realizaría para el usuario final, mostraría el mapa de las rutas de los camiones, así como un tiempo aproximado de llegada.

Para beneficiar a todos los usuarios y no solo a los que hacen uso de algún Smartphone, se pretende realizar la aplicación final en una Tablet, que este una en cada parada del PumaBús, de esta manera, la Tablet fungiría como la pantalla en la que se mostrarían los datos como se propusieron en las opciones anteriores. Además, se tiene la posibilidad de descargar la aplicación a cualquier Smartphone, aunque con algunas limitantes en cuanto a su funcionamiento, ya que la aplicación estaría optimizada para funcionar en cada parada del PumaBús.

En esta alternativa, lo interesante es que todo es en tiempo real, además de que mostraría la ubicación real de la unidad.

Un ejemplo de su funcionamiento a rasgos generales se muestra en la figura 2.5.



Figura 2. 5 Esquema general de tercera alternativa

2.3.1. Requisitos

1. Smartphone Receptor
2. Smartphone Emisor
3. Servidor para procesar información

Se propone un Smartphone debido a que actualmente cualquiera cuenta con un GPS integrado, es relativamente económico ya que se pueden encontrar smartphones desde 1000 pesos y en cuestión de espacio físico, no requiere demasiado y se puede colocar fácilmente en el camión, para enviar la posición del PumaBús a un servidor que se encuentre en línea, éste es el encargado de procesar la información recibida y desplegarla al usuario final en su Smartphone.

2.3.2. Ventajas

El gasto en infraestructura es muy reducido lo que ayuda a su fácil implementación. Además el que pueda brindar la información en tiempo real sobre la ubicación de las unidades ayuda a los usuarios a tomar una decisión basado en lo que está ocurriendo en todo el sistema.

2.3.3. Desventajas

El sistema solo está dirigido a los usuarios que tengan un Smartphone, esto limita a que no todos los usuarios del PumaBús tengan acceso a esta información, sin embargo, como ya se analizó en el capítulo anterior, el mercado de Smartphones es muy grande y se encuentra creciendo de forma exponencial por lo que aunque la aplicación solo se encuentre dirigida a este tipo de usuarios; esta cubriría a gran parte de los mismos.

2.4. Selección de la mejor alternativa

Tomando en cuenta las ventajas y desventajas de las diferentes propuestas se llegó a la conclusión de que la propuesta 3. Uso de SmartPhones como dispositivos receptores y emisores es la que brinda más beneficios a menor costo y a la mayor cantidad de usuarios.

2.4.1. Justificación de elección

La decisión tomada se basa en varios puntos, los cuales se mencionan enseguida.

De acuerdo a los costos presupuestados de cada alternativa se escribe con claridad en la tabla 2.1

Tabla 2.1 Comparación de costos

	Costo instalación pantallas	Costo Sensores	Costo Servidor	Costo Smartphone
Opción 1	1200 dólares (por cada parada)Costo promedio	N/A	100\$ (Mensuales)	2,000\$ (por camión)
Opción 2	1200 dólares (por cada parada)Costo promedio	4,500\$ (por cada parada), 300 \$ por cada unidad.	100\$ (Mensuales)	N/A
Opción 3	N/A	N/A	100\$ (Mensuales)	2,000\$ (por camión)

Tomando la información de la tabla anterior y utilizando el número de paradas (96) y de PumaBús actuales (65) tenemos los siguientes resultados:

- Opción 1 costo total: 1730100 pesos
- Opción 2 costo total: 2178100 pesos
- Opción 3 costo total: 120000 pesos

Los costos de la Tabla 2.1 son con Smartphones, Sensores y pantallas que soportan las características requeridas.

Al analizar la tabla se puede concluir que los costos de la opción 3 son significativamente menores. Este es un punto importante a tomar en cuenta para la realización e implementación del proyecto.

Otra razón importante por la que se tomó la decisión de desarrollar la opción 3 es la capacidad de ofrecer información en tiempo real. Esta característica es de gran importancia para los usuarios ya que pueden tomar una mejor decisión.

También es importante considerar el posible crecimiento del PumaBús (escalabilidad), ya que es más escalable la opción 3 al tener que realizarse modificaciones únicamente dentro del software.

De las pocas desventajas que tiene la opción 3 es que solo las personas que conozcan la existencia de esta aplicación podrán acceder a ella. Esto puede ser mitigado colocando carteles de información en las paradas de los PumaBús para dar a conocer a la comunidad la aplicación y de esta forma se informen sobre el paradero de PumaBús.

Dentro de la escalabilidad propuesta para el diseño, se puede lograr que cada usuario utilice su Smartphone como vía de acceso a la aplicación, de esta manera se consigue que no se dañe la propiedad de la universidad.

Al colocar tablets al alcance de los usuarios, también se debe considerar el proteger dicha infraestructura. Es una realidad que existen actos vandálicos en el Distrito Federal y dentro de ciudad Universitaria. Por eso al momento de que cada usuario haga uso de su propio dispositivo se espera que cada quien sea responsable del uso que se le da.

2.4.2. Requerimientos para la implementación de la solución

Debido a que la aplicación se realiza con software gratuito, los requerimientos para el desarrollo de la aplicación son básicos y éstos son:

- Se utilizaron computadoras con la capacidad de soportar ejecutar procesos como servidor, un ambiente de desarrollo que soporte JavaScript.
- Se requiere de una IP Pública, o del uso de un servidor VPC para poder acceder a este desde una IP pública.
- Se utilizó un servidor para alojar en él los servicios de la aplicación. Las características de dicho servidor fueron:
 - Sistema operativo: Linux Debian
 - Ram: 512 MB
 - SSD: 20 GB
 - Procesador: Intel(R) Xeon(R) CPU E5-2630L v2 @ 2.40GHz (1 core virtualizado)

Capítulo 3

Diseño y Desarrollo

En este capítulo se reúne la información necesaria para llevar a cabo el análisis de los requerimientos para el desarrollo del sistema, posteriormente se hace la planeación del proyecto y al final se presenta la realización de la aplicación tanto lógica como física.

3.1. Diseño de la aplicación

En el capítulo 1 se abordó el marco teórico, que rodeaba la realización del sistema, como se mencionó en los objetivos, existe la intención de que la aplicación abarque el mayor mercado posible. Teniendo esto como objetivo, se encontró la manera de realizar una aplicación multiplataforma, donde se empezará generando una aplicación web, siendo ese mismo código traducido a aplicaciones nativas automáticamente, llegando así a tener una aplicación instalable en los sistemas operativos mencionados en el “Marco Teórico”.

3.1.1. Phonegap

Para llevar a cabo la aplicación haciéndola llegar a la mayoría de plataformas posibles se utiliza PhoneGap.

PhoneGap es un framework de código abierto y gratis que permite crear aplicaciones móviles usando lenguajes web comunes, como son HTML, JavaScript y CSS.

PhoneGap compila la aplicación en la nube, sólo es suficiente tener una cuenta de Adobe (ya que PhoneGap es propiedad de Adobe), subir a sus servidores una carpeta comprimida en zip y PhoneGapBuild regresa los instalables para los diversos sistemas operativos.

3.1.2. NodeJS

Para establecer conexiones entre los dispositivos que pretendan enviar y recibir información de localización se utiliza NodeJS, que es una plataforma construida con JavaScriptV8. Permite correr código JavaScript en el backend, fuera del browser como se muestra en la figura 3.1

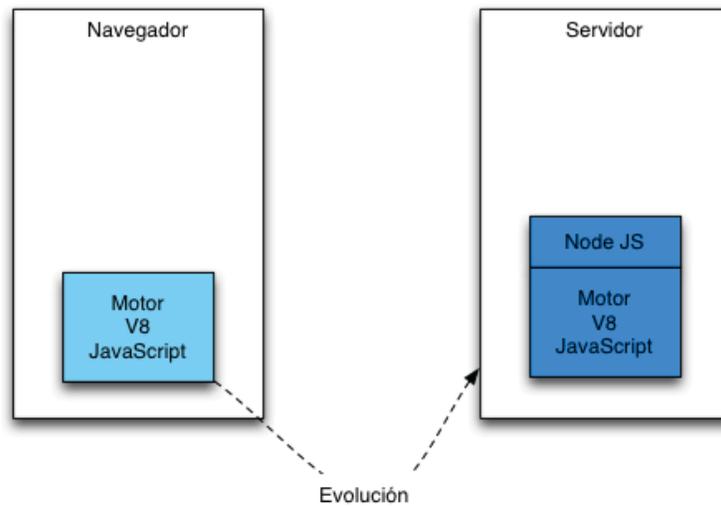


Figura 3. 1 Evolución de JavaScript

De los beneficios que brinda son la flexibilidad para el uso y trabajo de Hilos. Al ser una aplicación que requiere de múltiples conexiones, el manejo de estas debe ser tomado en cuenta. Node.js trabaja con un único hilo de ejecución que es el encargado de organizar todo el flujo de trabajo que se deba realizar, ya que dentro del código de programación se debe generar y montar un servidor. Un ejemplo gráfico de esto se muestra en la figura 3.2 donde se aprecia que JavaScript es capaz de recibir diversas peticiones y gracias a Node.js, las separa por hilos y pueden ser atendidas al mismo tiempo.

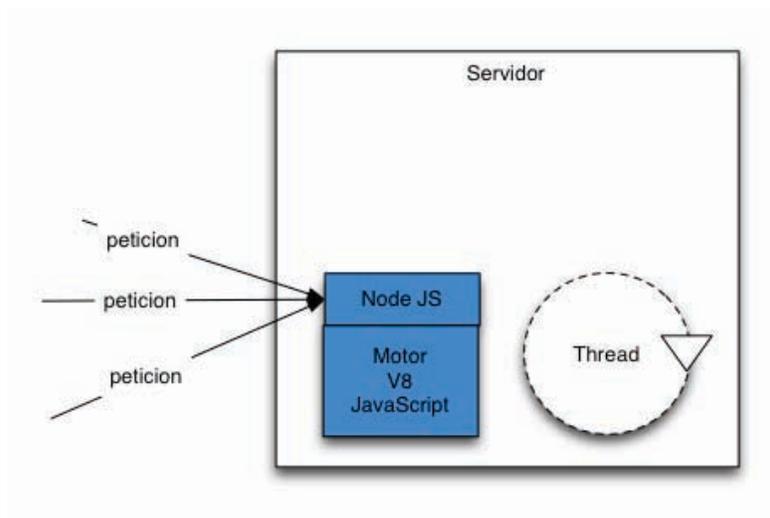


Figura 3. 2 Operación de JavaScript y Node.js

3.1.3. Heroku

Para alojar los servicios de NodeJS se ocupa Heroku, que es una plataforma capaz de alojar aplicaciones web desarrolladas en diversos lenguajes de programación y además ser consultados desde internet mediante un enlace que genera el propio servidor.

3.1.4. Herramientas de desarrollo

En el diseño de la aplicación se ocupa DreamWeaver, que es un entorno de desarrollo, muy enfocado al diseño. DreamWeaver se encuentra dentro de la suite de Adobe y se considera ideal para esta aplicación por la facilidad que da al momento de programar e ir evaluando el desarrollo físicamente.

Para programar la parte de JavaScript se utiliza NetBeans, que es un entorno de desarrollo muy útil y práctico para cualquier lenguaje de programación.

En la parte de la visualización de los mapas se utiliza el API de Google Maps ya que es una plataforma bastante estable y con documentación que la avala.

3.1.5. Servidor web

La aplicación al ser realizada en HTML, puede ser alojada en un servidor web y consultada desde una computadora. Para eso se ocupa el servidor Apache que es un servidor web HTTP de código abierto para la creación de páginas y servicios web; se aloja en Digital Ocean que es una plataforma de Hosting VPS. Ofrece una interfaz sencilla para su administración y el costo depende de las características que se necesiten y el tráfico que fluya por el servidor.

Para la aplicación se brinda una IP pública que permite realizar las conexiones desde distintos sitios a un servidor. Para la aplicación se cuenta con un servidor con las siguientes características:

- Debian 7
- 512 MB RAM
- 20 GB SSD
- IP pública: 104.236.24.186
- Nombre de dominio: lemuria.gq

3.1.6. Estructura de la aplicación

Para la realización de la aplicación se creó un diagrama de navegación web que es la base para el desarrollo de la aplicación. Dicho diagrama se muestra en la figura 3.3.

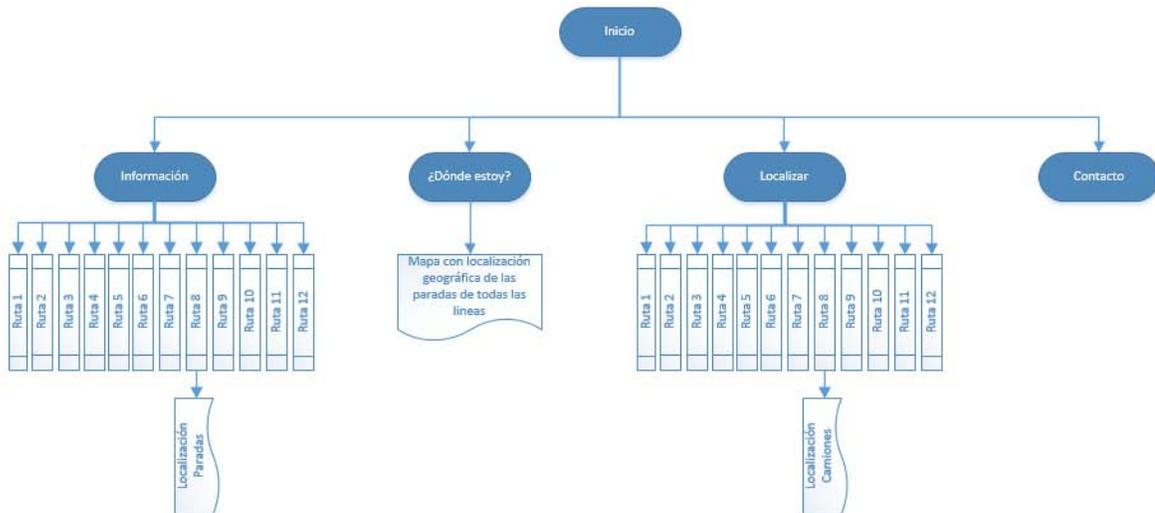


Figura 3. 3 Diagrama de navegación de la aplicación.

Como se muestra en el diagrama se tienen 4 opciones al inicio, que son; Información, ¿Dónde estoy?, Localizar y Contacto. Cada opción tiene una función específica además de tener diferentes submenús, los cuales son:

- Información.- Muestra la información detallada de cada ruta, existe la posibilidad de ver el mapa donde sólo muestre las paradas de la estación seleccionada.
- ¿Dónde estoy?- Muestra un mapa general con las paradas de todas las rutas.
- Localizar.- Es la opción dónde se elige una ruta, muestra los camiones disponibles de dicha ruta, permite elegir un punto de llegada y el origen del viaje, generando un tiempo estimado de llegada al destino.
- Contacto.- Muestra la información de contacto de los desarrolladores.

Con fines de lograr un buen diseño de la aplicación para su fácil edición, escalabilidad y mejora, se usa un modelo vista controlador (MVC). Donde el modelo son los métodos escritos en JavaScript, la vista son los diseños creados con HTML y CSS y el controlador lo genera NodeJS.

3.2. Realización de la aplicación

3.2.1. Diseño HTML y CSS

Para darle una interfaz agradable y llamativa se generan hojas de estilo, que son llamadas CSS, y son invocadas desde el código HTML como se muestra en la figura 3.4

```
<link rel="stylesheet" type="text/css" href="css/estilos.css">
```

Figura 3. 4 Ejemplo de llamado a un CSS

De esta manera se genera la misma línea de código para todos los archivos de la aplicación, con la finalidad de tener el mismo diseño en toda la aplicación.

3.2.2. Sockets web y NodeJS

Para que un socket funcione necesita principalmente que se le asigne un puerto para escuchar y enviar peticiones. Para lograr eso en NodeJS se logra como se muestra en la figura 3.5

```
server.listen(process.env.PORT || 8082);
```

Figura 3. 5 Asignación de un puerto a un socket.

Teniendo esto, el servidor está a la espera de peticiones para ser atendidas.

En la parte del cliente, las peticiones se mandan como se muestra en la figura 3.6

```
var socket = io.connect('/');
socket.on('connect', function ()
```

Figura 3. 6 Conexión del cliente al socket.

Y el servidor las responde como se muestra en la figura 3.7

```
io.sockets.on('connection', function (socket)
```

Figura 3. 7 Respuesta a la conexión del cliente

Este proceso se itera las veces que sea consultada la aplicación.

3.2.3. Google Maps

Para hacer uso de los mapas de google es necesario agregar en cada página de la aplicación donde muestre un mapa el fragmento de código mostrado en la figura 3.8

```
<script
src="https://maps.googleapis.com/maps/api/js?v=3.exp"></script>
```

Figura 3. 8 Llamado a la librería de Google Maps desde HTML.

Con esta línea ya se puede ocupar el API de Google Maps, que es básicamente JavaScript.

3.2.4. JavaScript

JavaScript es utilizada en toda la aplicación, pero principalmente el uso que se le da es para re-direccionar a otros archivos internos que se tienen como se muestra en la figura 3.9

```
function showMap() {
window.location = "Mapas/Ruta8.html";
}
```

Figura 3. 9 Redirección de páginas con JavaScript

3.2.5. Iconos y Logos

Para el diseño de esta aplicación, se realizaron logos e íconos con referencia a cada parte de la aplicación. Las siguientes imágenes (figuras de la 3.9 a la 3.16) muestran dichos iconos.



Figura 3. 10 Ícono de la aplicación



Figura 3. 16 Íconos de punteros para los mapas



Figura 3. 17 Íconos de letreros de rutas.

3.3. Documentación de la aplicación

La aplicación busca brindar más información a los usuarios del PumaBús para tomar decisiones más sólidas y de esta forma invertir su tiempo de la manera más eficiente.

Algunos aspectos importantes a ser mencionados son:

- Funcionamiento exclusivo de la ruta 8

Las únicas paradas mostradas en la aplicación son las de la ruta 8 del Puma Bus. Se seleccionó esta ruta debido a que es la ruta que une los complejos de ingeniería, además de tener puntos importantes a lo largo de la ruta, como es el estadio y el MetroBús. Para la aplicación desarrollada se cumple con el alcance, ya que agregar más rutas es una tarea donde se deben conocer las coordenadas de cada una de las paradas de las diferentes rutas.

- Número limitado de usuarios

Aunque la aplicación no limita el número de usuarios, se ve limitada por la capacidad del Virtual host contratado. La cantidad de memoria RAM con la que se cuenta es de 512 MB. Por esta razón si el número de usuarios sobrepasara las capacidades físicas de la máquina, ésta no podría manejarlo, sin embargo, si la aplicación es instalada y administrada por parte de la UNAM, sería la propia universidad la que decidiría que recursos brindarle a la aplicación para que se gestione de mejor manera a los usuarios.

Cabe destacar que la aplicación se encuentra funcionando y es capaz de realizar sus funciones, sin embargo no debe ser utilizada en sus condiciones actuales en ambientes de producción.

La aplicación desarrollada aunque brinda varias opciones para el usuario, también se encuentra limitada debido al alcance propuesto para el desarrollo de la misma.

Dicha aplicación fue desarrollada pensando en llegar a la mayor cantidad de usuarios. Por esta razón se alojó la aplicación en un servidor web por lo que con tener un navegador web es suficiente para poder acceder a la aplicación. Específicamente la aplicación fue probada con los siguientes sistemas operativos:

- Windows (7, 8, 8.1)
- Linux Debian
- Mac OS (Maverick)
- Android (2.3, 4.4, 5.0)
- IOS (7, 8)

Los navegadores que fueron probados y donde la aplicación funcionó de manera correcta son:

- Firefox
- Iceweasel
- Internet Explorer
- Google Chrome
- Safari

Esto demuestra que la aplicación puede ser usada por distintos tipos de usuarios, beneficiando a la mayor cantidad de los mismos.

3.3.1. Adaptación del Servidor

Para que la aplicación se ejecutará de manera adecuada, fue montada sobre un servidor con sistema operativo Debian 7, puesto que, se tenía que contar con acceso a una IP pública para realizar las peticiones vía Internet. Al realizar esta aplicación se pensaba no solo en los usuarios que pertenecen a la comunidad universitaria y que tienen acceso a la red interna, sino llegar a todos los usuarios que hicieran uso del PumaBús.

Se obtuvo una IP pública de Digital Ocean. La IP asignada fue 104.236.24.186. Una vez obtenida se consiguió un dominio para hacer referencia. El nombre de dominio utilizado es lemuria.gq.

3.3.2. Cálculo de tiempos

Para los cálculos de tiempo realizados por la aplicación se utilizaron los tiempos estimados de Google maps y tiempos reales entre las paradas del PumaBús tomados experimentalmente. La razón de utilizar dos tiempos es obtener un cálculo de tiempo más cercano a la realidad. El API de Google maps realiza los cálculos con base en tres factores:

- La distancia entre el origen y el destino
- La cantidad de tráfico
- La velocidad máxima permitida en ese lugar (40Km/h en Ciudad Universitaria)

Estos tres factores son muy útiles para realizar los cálculos en una ciudad, pero para el caso del PumaBús en Ciudad Universitaria se debían considerar otros aspectos:

- El PumaBús Realiza paradas
- Tiene un carril exclusivo para su circulación

La forma en la que Google calcula los tiempos se muestra en la figura 3.18

```
response.routes[0].legs[0].duration.value
```

Figura 3. 18 Cálculo de tiempo de Google Maps

Los tiempos obtenidos en la práctica entre cada parada se muestran en la figura 3.19

```
var times = [
301.8, 120.6, 69, 85.8, 145.8, 150, 90.6, 168, 231.6,
163.8, 135, 73.8, 156.6, 192.6, 127.8, 133.8, 75.6,
51.6, 75, 63.6
];
```

Figura 3. 19 Tiempos para llegar a la siguiente parada

Por esta razón se combinó la información para obtener un mejor resultado como se muestra en la figura 3.20.

```
full = final + parseInt(response.routes[0].legs[0].duration.value);
```

Figura 3. 20 Suma de tiempos

De esta manera se tiene un tiempo más preciso y aproximado.

Capítulo 4

Pruebas y Resultados

En este capítulo se muestran los resultados del desarrollo de la aplicación. Así como tiempos y forma de ejecución de las mismas.

4.1. Producto Final

Para acceder a la aplicación desde un Smartphone sólo es necesario dar clic en el icono de la aplicación, o si se quiere entrar mediante una PC se utiliza la siguiente URL:

<http://lemuria.gq/Tesis/>

Al ingresar la URL se despliega la pantalla principal de la aplicación. Como se muestra en la figura 4.1

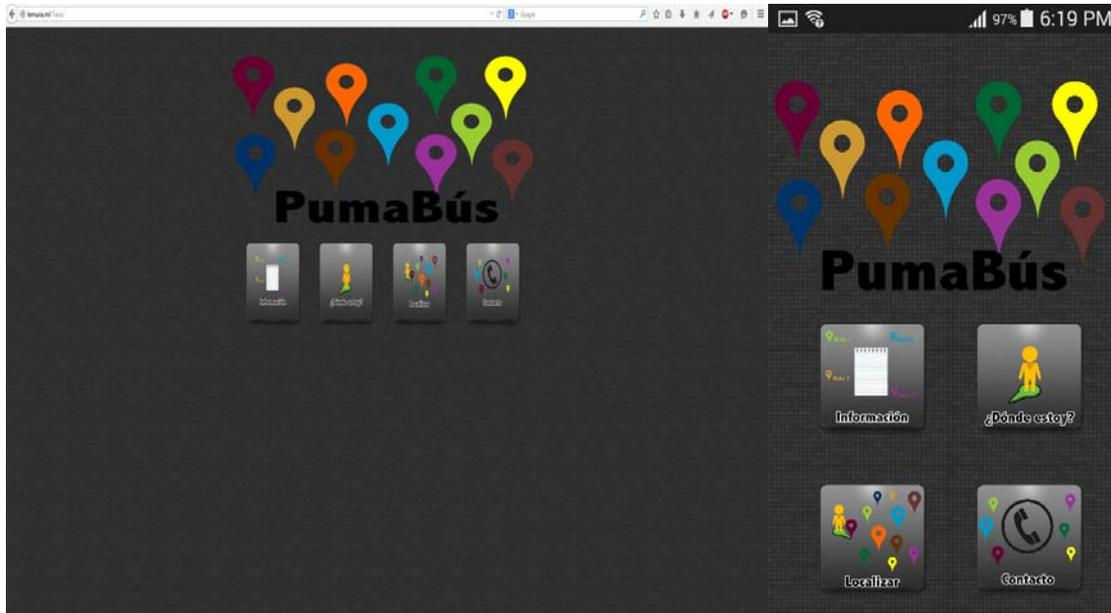


Figura 4. 1 Interfaz principal de la aplicación

Al ingresar a la aplicación se puede observar el logo y cuatro iconos que funcionan como ligas para acceder al contenido de la aplicación. Como se muestra en la figura 4.2



Figura 4. 2 Logotipo de la aplicación

4.1.1. Información



Figura 4. 3 Botón “Información”

Al dar clic en el botón información (Figura 4.3), la aplicación puede obtener información relevante de cada parada para el usuario que no está familiarizado con el PumaBús o información más específica como mapas u horarios de operación de las rutas, como se muestra en la figura 4.4

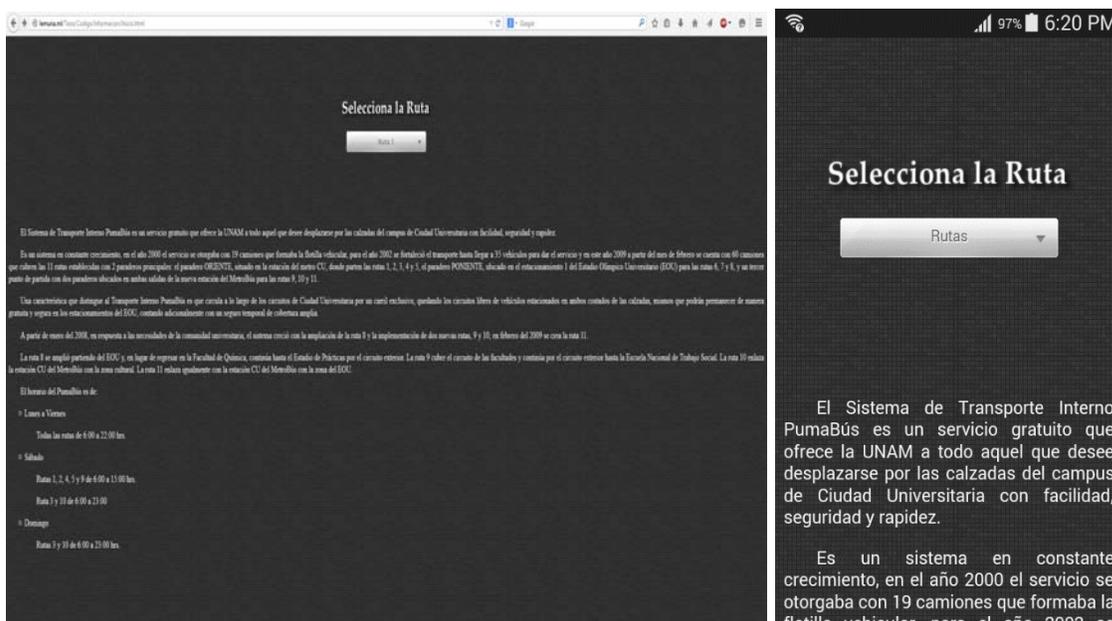


Figura 4. 4 Información general del servicio PumaBús

Cuando el usuario da clic sobre el botón de rutas se despliega un menú con las distintas rutas que ofrece actualmente el PumaBús. Una vez que se selecciona la ruta deseada se desplegará información específica de la ruta. Como se muestra en la figura 4.5



Figura 4. 5 Menú de rutas

Algunos de los datos que se pueden obtener en esta sección son:

- Número de la ruta
- Recorrido de la ruta
- Distancia
- Puntos de Referencia
- Número de paradas
- Mapa

La sección ver mapa despliega un mapa donde se muestran las paradas de la ruta indicada. Para la creación de este mapa se utiliza el API de Google Maps. La diferencia entre el primer mapa y el segundo es la posibilidad que ofrece al usuario para interactuar con el mapa, como hacer zoom, o localizar geográficamente las paradas. El primer mapa es una imagen estática que puede ser de gran utilidad para los usuarios que conocen o están familiarizados con Ciudad Universitaria. Se muestran puntos muy específicos sin muchas indicaciones. Como se muestra en la figura 4.6

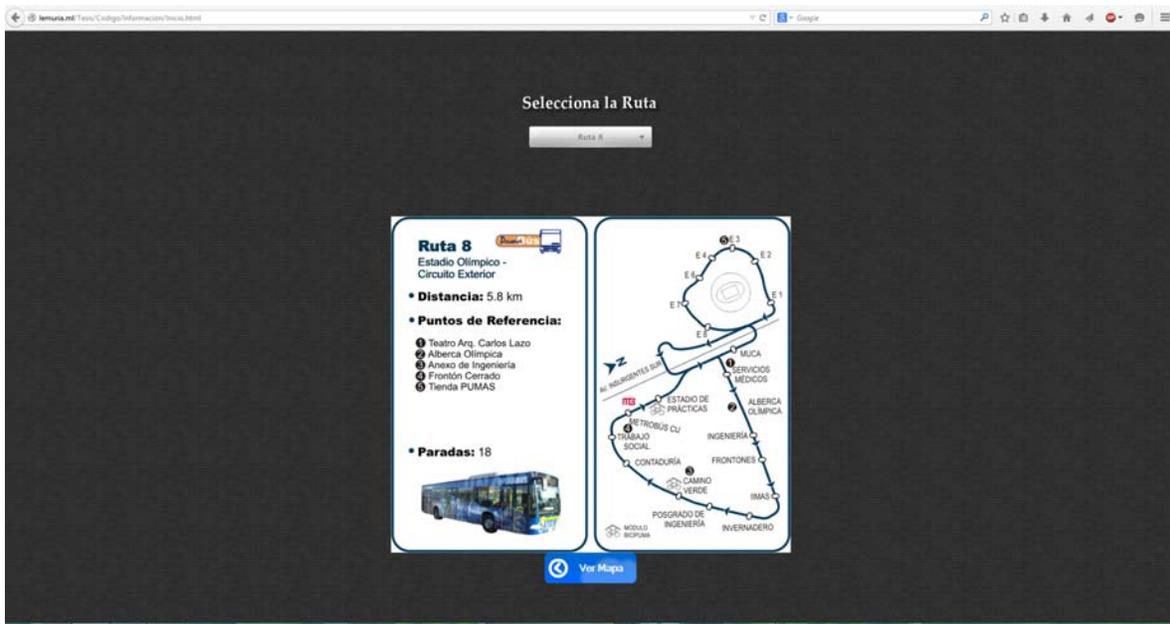


Figura 4. 6 Interfaz de información de ruta 8

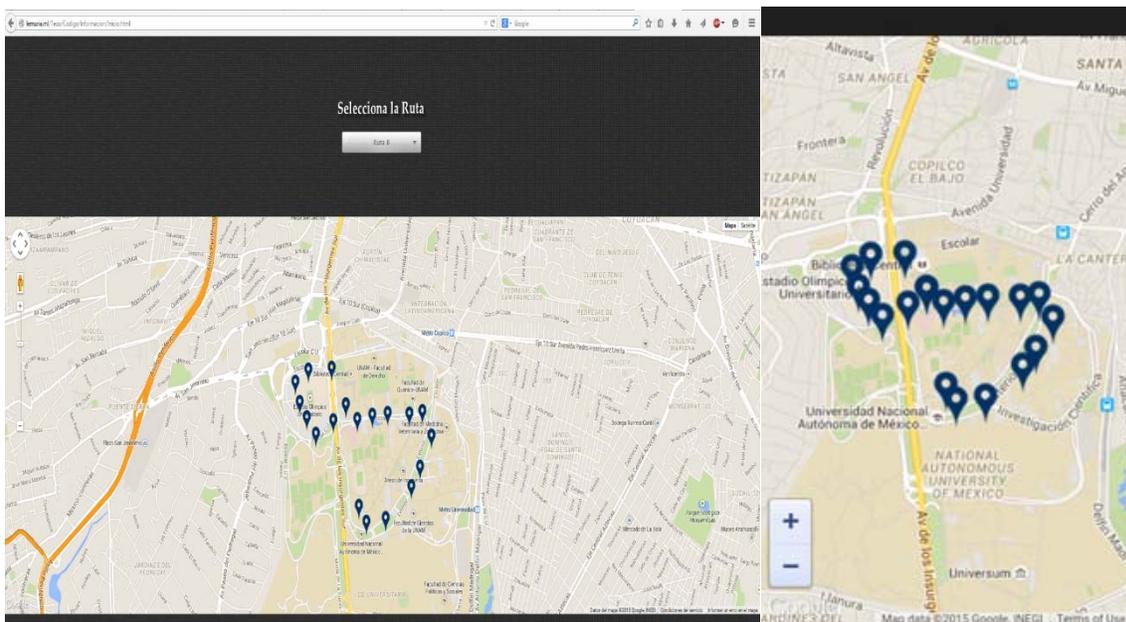


Figura 4. 7 Mapa de ubicación de paradas de la ruta 8

El segundo mapa (Figura 4.7) por su parte al permitir la interacción del usuario para acercar el mapa, alejarse y moverse no solo en el área de Ciudad Universitaria, sino a cualquier punto que el usuario desee. Esto facilita el entendimiento de las paradas del PumaBús, permitiendo que para quien utilice el servicio sea más fácil ubicar cada una de las paradas con base en distintas referencias que se encuentran en el mapa de Google

Maps, desde referencias de Ciudad Universitaria hasta el nombre de las calles. Como se muestra en la figura 4.8

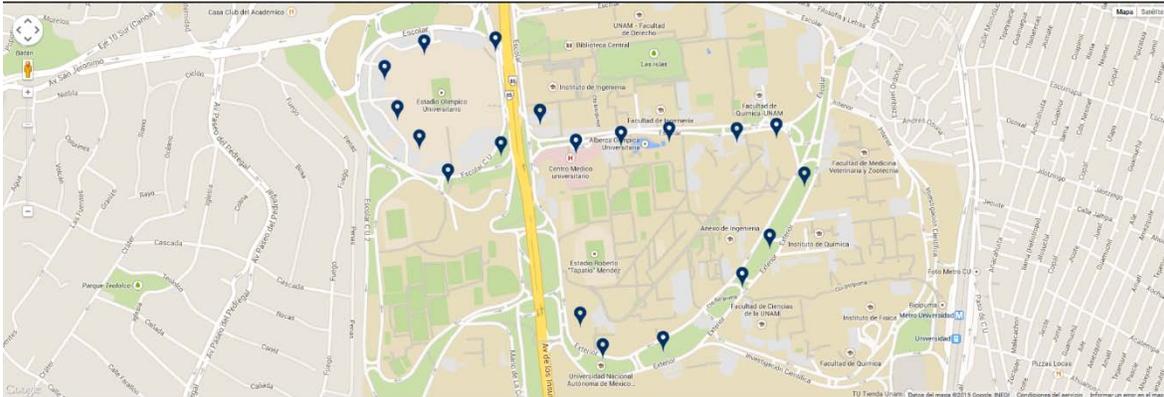


Figura 4. 8 Paradas ruta 8

Otra característica que ofrece este mapa es la posibilidad de una vista satelital. Con esta vista se puede tener otra perspectiva sobre la ubicación de las paradas. A diferencia del primer mapa que sólo ofrece nombres, este mapa permite ver qué hay cerca de la parada, permitiendo al usuario decidir qué parada es la más cercana al destino. A continuación se expone una situación donde esto ocurre:

Si lo que se busca es llegar del Estadio Olímpico Universitario a la Facultad de Ciencias y el usuario solo se basa en el primer mapa, es posible observar que solo la ruta 6 tiene una parada llamada “Ciencias”. El usuario podría tomar la ruta seis y llegar a su destino, pero esta no es la ruta más rápida. La ruta 6 pasa por varias paradas que se encuentran alejadas, por lo que la distancia es mayor y el tiempo de llegada al destino aumenta. Esto no es visible en el primer mapa. Por otro lado si se utiliza el segundo mapa, el usuario puede notar que si toma la ruta 8 y decide caminar un poco, llegará antes a su destino. Como se muestra en la figura 4.9



Figura 4. 9 Vista satelital de la ubicación de las paradas de la ruta 8

4.1.2. ¿Dónde estoy?



Figura 4. 10 Botón “¿Dónde Estoy?”

Al dar clic en esta sección (Figura 4.10) despliega al usuario la ubicación de todas las paradas del PumaBús y agrega la ubicación actual del usuario. El beneficio de esta función es la posibilidad de encontrar al usuario y con base en la información proporcionada tomar decisiones. Como se muestra en la figura 4.11



Figura 4. 11 Ubicación de usuario

4.1.3. Localizar



Figura 4. 12 Botón “Localizar”

Esta sección (Figura 4.12) es la encargada de localizar la ubicación del PumaBús. En esta sección se pide al usuario que seleccione la ruta que desea. Como se muestra en la figura 4.13

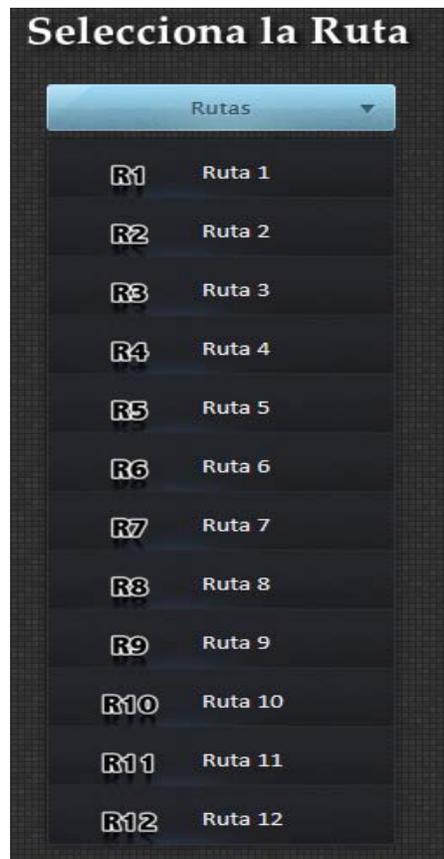


Figura 4. 13 Menú de Rutas

Una vez que el usuario selecciona la ruta, se despliega un mensaje solicitando al usuario su nombre o pseudónimo (Figura 4.14). Esto ayuda a la aplicación para identificar qué tipo de usuario es.

La aplicación entiende la existencia de dos usuarios. El usuario chofer y el usuario común. Las diferencias entre estos dos tipos de usuario son las siguientes:

- Chofer: El usuario chofer una vez que entra a la aplicación se encuentra operando como tal. No tendrá que realizar ninguna otra modificación. Aparecerá visible en el mapa y podrá ser visto por cualquier otro usuario que ingrese a la aplicación.
- Común: no es visible en el mapa, pero es capaz de ver en el mapa a los usuarios chofer. Permite la opción de localizar parada más cercana y conocer el tiempo de llegada.

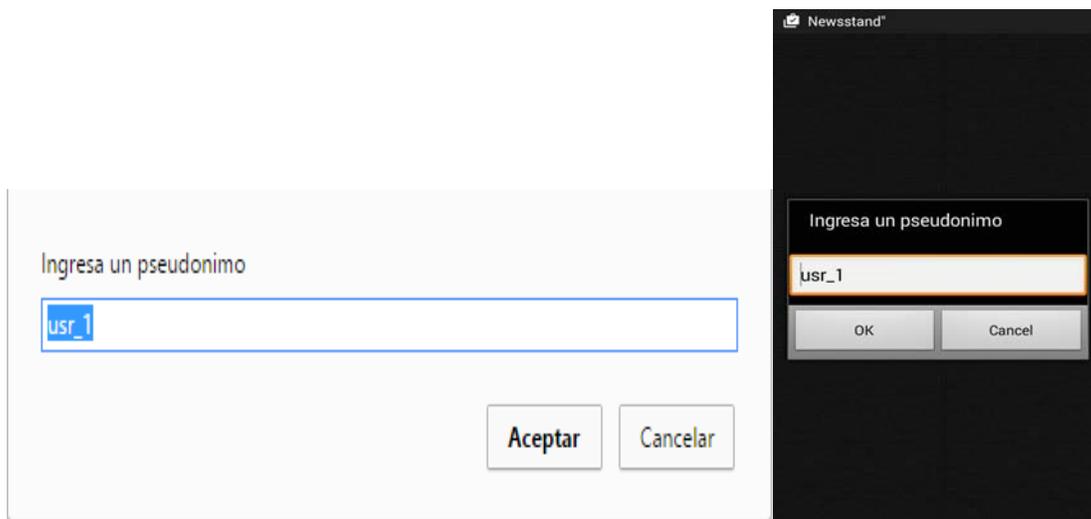


Figura 4. 14 Petición de nombre a usuario

Una vez que el usuario coloca su nombre aparece una pantalla donde se le solicitan al usuario más datos para calcular los tiempos correspondientes (Figura 4.15 y 4.16). Los campos a llenar son:

- Parada origen
- Parada destino
- Selección de camión

En caso de ser un usuario chofer no es necesario que se ingrese ningún dato.



Figura 4. 15 Petición de datos

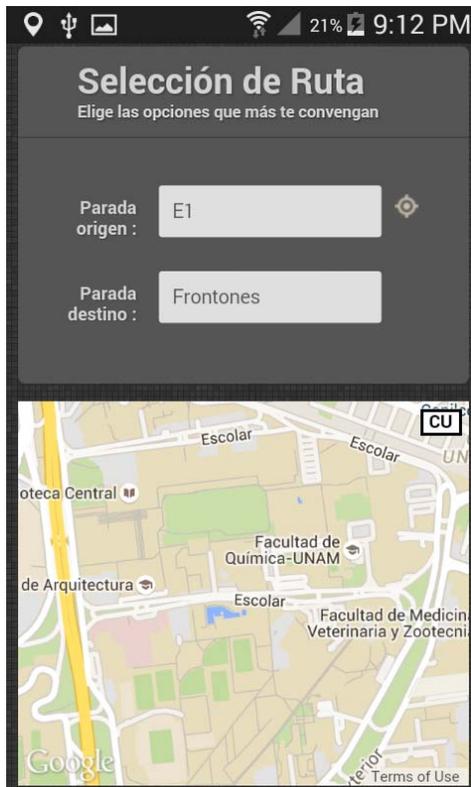


Figura 4. 16 Petición de datos completo

Al dar clic en el botón “Calcular Ruta” regresa un mensaje con el tiempo estimado, como se muestra en la figura 4.17.

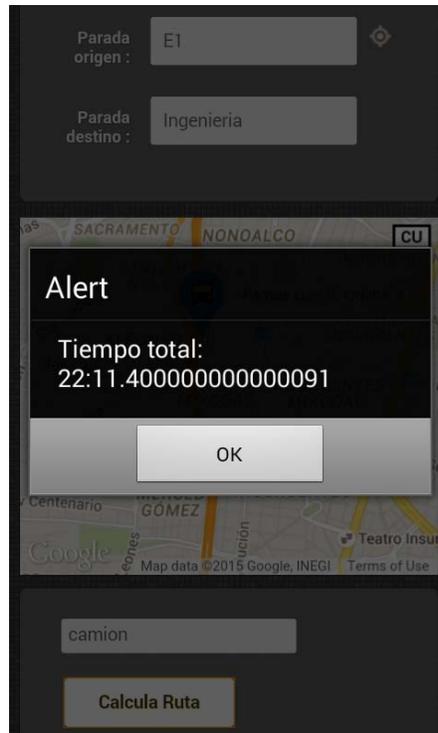


Figura 4. 17 Tiempo Total

4.2. Pruebas de funcionalidad del sistema

Las siguientes pruebas fueron realizadas como se describe a continuación. Una persona ingresaba al PumaBús y se autentificaba como chofer, mientras que la otra se colocaba en una parada de la ruta 8 y hacía uso de la aplicación.

4.2.1. Prueba 1

Origen: Alberca Olímpica

Destino: Posgrado de Ingeniería

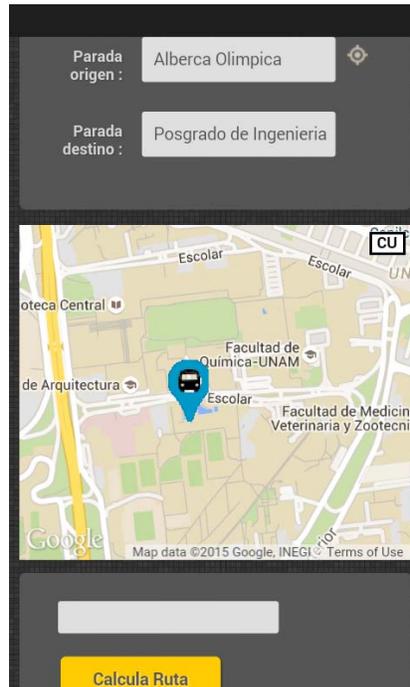


Figura 4. 18 Prueba 1

Mapa posición original del PumaBús (Figura 4.18)

Posición origen del PumaBús: Cerca de la parada Centro médico. (Figura 4.19)



Figura 4. 19 Imagen de localización del usuario

Información brindada por la aplicación:

Tiempo aproximado de llegada al destino 8 minutos con 28 segundos. (Figura 4.20)

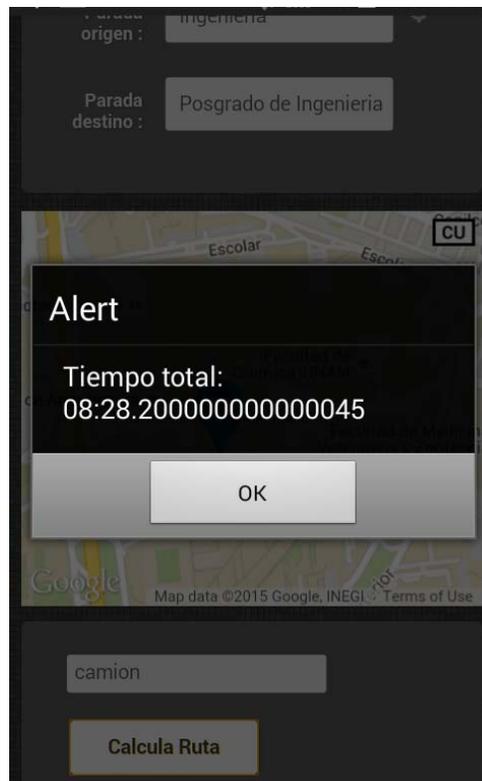


Figura 4. 20 Tiempo de llegada al destino

El tiempo real de llegada fue de:

6 minutos con 38 segundos.

Comparando el tiempo real con el estimado por la aplicación que fue de 8 minutos con 28 segundos, se observa que hay una variación de tiempo generando un posible margen de error, ya que hay condiciones que no son controladas por la aplicación, como el tiempo de espera en cada parada, semáforos o velocidad del camión.

4.2.2. Prueba 2

Origen: Contaduría

Destino: Estacionamiento 1 (Figura 4.21)

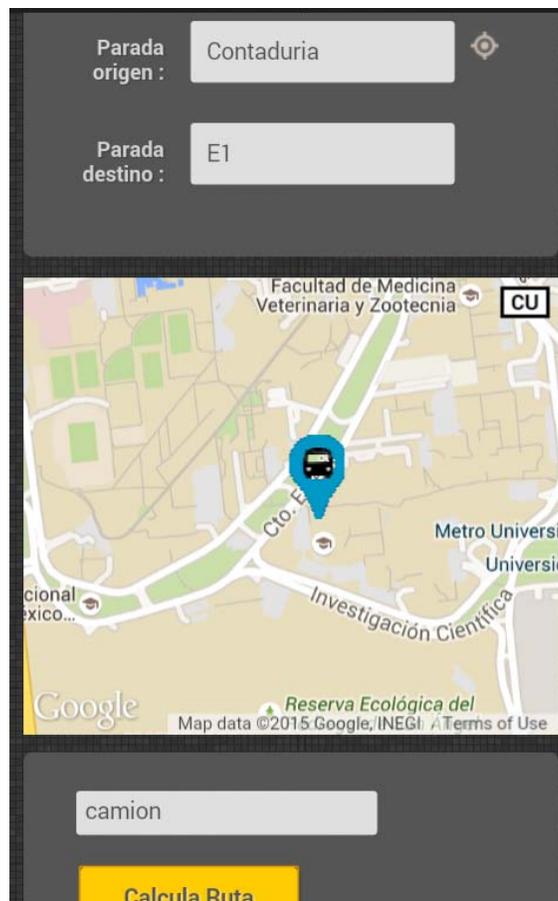


Figura 4. 21 Prueba 2

Punto de partida del PumaBús: Cerca del Anexo de Ingeniería



Figura 4. 22 Ubicación de PumaBús.

Información brindada por la aplicación:

Tiempo aproximado de llegada al destino 22 minutos con 07 segundos. (Figura 4.23)

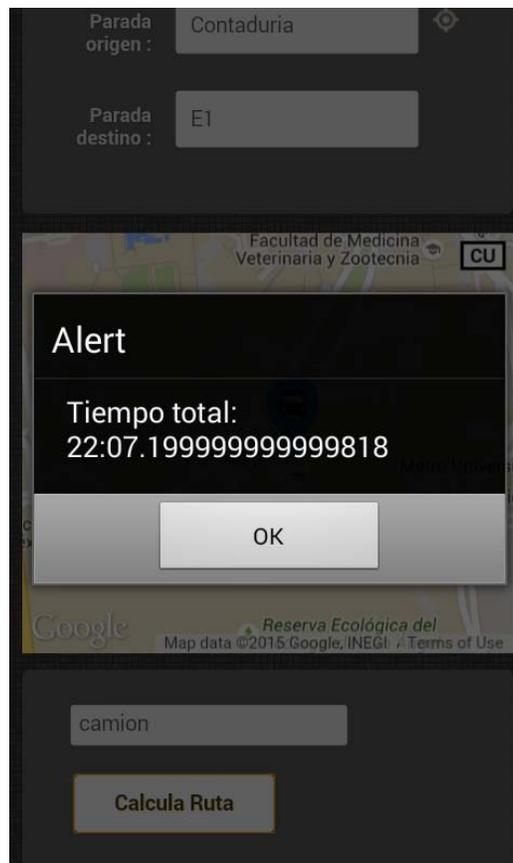


Figura 4. 23 Tiempo estimado de llegada

El tiempo real de llegada fue de:

18 minutos con 22 segundos.

Comparando el tiempo real con el estimado por la aplicación que fue de 22 minutos con 7 segundos, se observa que hay una variación de tiempo generando un margen de error, ya que la prueba se hizo en un horario que no es tan concurrido en las instalaciones universitarias, esto demuestra que existe una variación de tiempo dependiendo de la demanda del servicio de transporte universitario.

4.2.3. Prueba 3

Origen: Invernaderos

Destino: Estacionamiento 6 (Figura 4.24)

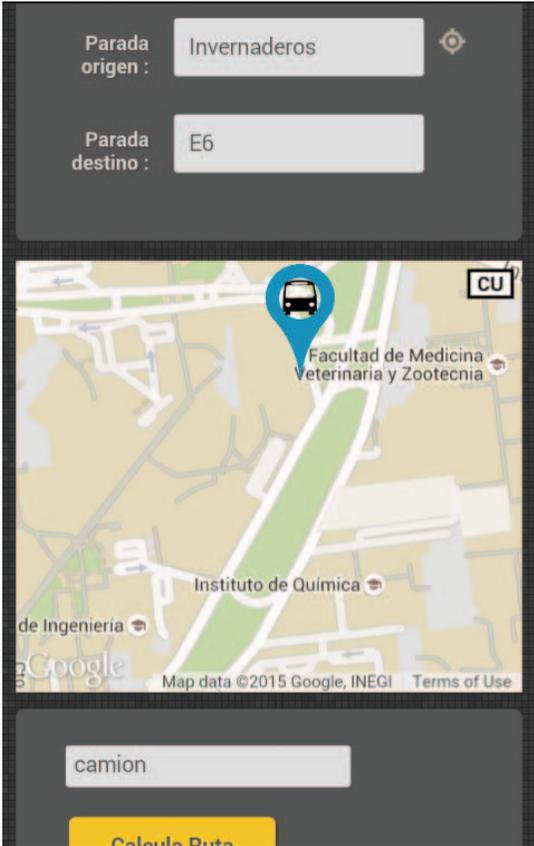


Figura 4. 24 Prueba 3

Punto de partida del PumaBús: Cerca de la parada de IIMAS (Figura 4.25)



Figura 4. 25 Posición del PumaBús.

Información brindada por la aplicación:

Tiempo aproximado de llegada al destino 24 minutos con 14 segundos. (Figura 4.26)



Figura 4. 26 Tiempo estimado de prueba 3

El tiempo real de llegada fue de:

25 minutos con 15 segundos.

Comparando el tiempo real con el estimado por la aplicación que fue de 24 minutos con 14 segundos, se observa que hay una variación de dónde se ganaron unos segundos de acuerdo al estimado que regreso la aplicación.

4.2.4. Prueba 4

Después de realizar las pruebas anteriores se decidió hacer una prueba en la que solo se documentara la diferencia entre el tiempo otorgado por la aplicación y tiempo real del trayecto el real. (Figura 4.25)

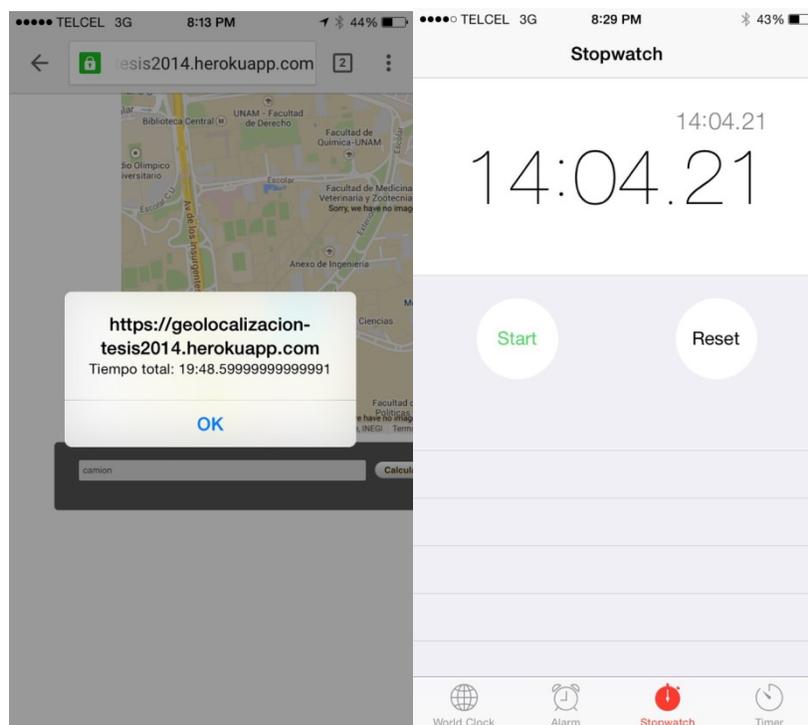


Figura 4. 27 Prueba 4

Nuevamente se genera una ganancia en el tiempo de acuerdo al estimado que marca la aplicación, esto es debido al horario en que se hicieron las pruebas y el horario en el que se tomaron los tiempos entre parada y parada, generándonos en la mayoría de los casos una ganancia en los tiempos estimados.

Conclusiones

El objetivo general de este trabajo fue crear una aplicación que tuviera impacto en la toma de decisiones de los usuarios del sistema de transporte PumaBús, mostrándoles la ubicación de los camiones del sistema, así tendrían un aproximado del tiempo de llegada a sus destinos dentro del campus Ciudad Universitaria. De esta manera el usuario tiene la facilidad de decidir entre esperar el camión PumaBús o tomar otro medio de transporte. Tomando ventaja de la tecnología con la que contamos, es posible interactuar de una manera bastante fluida con nuestros dispositivos, es por ello que se logró generar un sistema el cual muestra la ubicación de los camiones y además el tiempo estimado de llegada al destino del usuario.

A lo largo del presente trabajo se mostraron diferentes plataformas sobre las cuales podría operar la aplicación, sin embargo tuvimos la posibilidad de hacerla multiplataforma, de esta manera queda al alcance de las personas que cuenten con cualquier tipo de Smartphone o computadora, ya que se realizó la aplicación en un sistema web, pudiéndolo consultar desde cualquier browser.

Actualmente la aplicación se encuentra en una fase beta, donde solo está documentada la ruta 8. Tomando como punto de partida la realización de la ruta 8 y su efectividad en la misma, se desarrollarán las demás rutas logrando con ello ofrecer el servicio a todos los usuarios dentro del campus Ciudad Universitaria.

La aplicación puede ser mejorada e implementada en el campus universitario para beneficiar a las personas que haga uso del sistema PumaBús.

La tecnología avanza día con día y los dispositivos móviles no son la excepción. Cada vez más comodidades son brindadas a través de estos dispositivos que se han vuelto parte importante de la vida de las personas. Es importante no solo utilizarlos como medio de entretenimiento, sino buscar mejorar la productividad del trabajo humano, disminuir los esfuerzos físicos e incrementar la calidad de vida de la población.

Glosario

API (Interfaz de programación de aplicaciones):

Una API es un conjunto de funciones que permite al programador acceder a servicios de una aplicación a través del uso de un lenguaje de programación.

Backend:

Es una abstracción que sirve para mantener separadas las diferentes partes de un sistema. Se refiere al lado del servidor, hace referencia al sitio donde se procesa y envía información al navegador del usuario.

Basic Phone:

Telefonos móviles de gama baja o de características limitadas. Posen funciones esenciales como la posibilidad de llamar o de enviar mensajes de texto, el software es relativamente limitado.

Bit (Binary Digit):

Es la menor unidad de información digital con la que pueden trabajar los dispositivos electrónicos. Permite representar dos valores diferentes (como abierto/cerrado o verdadero/falso) y asignar dichos valores al estado de encendido (1) o apagado (0).

Browser:

Es un programa o software, por lo general gratuito, que permite visualizar páginas web a través de Internet además de acceder a otros recursos de información alojados en servidores web, como pueden ser videos, imágenes, audio y archivos XML.

Debugger:

El depurador 'debugger', tiene como misión principal ayudar a identificar errores de programación, al tiempo que permite visualizar los contenidos de la memoria de trabajo durante la ejecución del programa o la función en la ejecución del código. La depuración (debugging) es el proceso mediante el cual se aísla y determinan errores existentes en el código.

Emulador:

Un emulador es un *software* que imita al *hardware* o a un sistema operativo con el objetivo final de ejecutar un programa, aplicación, software, etc. sobre éste

Escalabilidad:

Es la propiedad de aumentar la capacidad de trabajo o de tamaño de un sistema sin comprometer el funcionamiento y calidad normales del mismo. En programas de

cómputo se dice que el sistema es escalable cuando puede aumentar el número de usuarios, de datos que procesa o de solicitudes que recibe, sin que se afecte significativamente su velocidad de respuesta.

Fase Beta:

Una versión beta o lanzamiento beta representa generalmente la primera versión completa del programa informático o de otro producto, que es probable que sea inestable pero útil para que las demostraciones internas y las inspecciones previas seleccionen a clientes.

Feature Phone:

Los feature phones poseen funciones esenciales como la posibilidad de llamar o enviar mensajes de texto, emplear archivos multimedia o navegar por internet usando conexiones GSM o WiFi. Los teléfonos con funciones limitadas son comercializados como alternativa de menor costo a los teléfonos inteligentes, específicamente en mercados emergentes.

Framework:

Es un esqueleto para el desarrollo de y/o la implementación de una aplicación. Un Framework es un entorno o ambiente de trabajo para desarrollo; dependiendo del lenguaje normalmente integra componentes que facilitan el desarrollo de aplicaciones como el soporte de programa, bibliotecas, plantillas y más.

Geoposicionamiento:

El objetivo del geoposicionamiento o geolocalización es ofrecer información de la ubicación geográfica de un recurso. Estos objetos pueden ser determinados de las formas siguientes:

- 1.- En relación a un sistema de coordenadas, generalmente tridimensional.
- 2.- En relación a otro punto, tomando uno como origen de un sistema de coordenadas locales.

El primer modo de posicionamiento se conoce como posicionamiento absoluto, el segundo como posicionamiento relativo. Si el objeto es estacionario, se usa el término posicionamiento estático. Cuando el objeto es móvil, se denomina posicionamiento cinemático. Generalmente, el posicionamiento estático se usa en topografía y el cinemático en navegación.

Google Maps:

Google Maps es un servidor de aplicaciones de mapas en la web que pertenece a Google. Ofrece imágenes de mapas desplazables, así como fotografías por satélite del mundo e incluso la ruta entre diferentes ubicaciones o imágenes a pie de calle con Google Street View.

Google Trends:

Google Trends es una herramienta de acceso libre y gratuito brindada por Google, que permite comparar la popularidad de búsqueda de varias palabras o frases; de esta manera conocer el nivel de búsqueda de un determinado término (keyword) durante un período de tiempo determinado.

GPS:

Sistema de Posicionamiento Global (Global Positioning System) es un sistema de navegación basado en satélites

Hardware:

El hardware es la parte física de una computadora está formado por los componentes eléctricos, electrónicos, electromecánicos y mecánicos, tales como circuitos de cables y circuitos de luz, placas, utensilios, cadenas y cualquier otro material, en estado físico, que sea necesario para hacer que el equipo funcione.

Hosting/Servidor VPS:

Un VPS (Servidor privado virtual) es un método de particionar un servidor físico en varios servidores de tal forma que todo funcione como si se estuviese ejecutando en una única máquina. Cada servidor virtual es capaz de funcionar bajo su propio sistema operativo y además cada servidor puede ser reiniciado de forma independiente.

Internet:

Internet es un neologismo del inglés que significa red informática descentralizada de alcance global. Se trata de un sistema de redes informáticas interconectadas mediante distintos medios de conexión, que ofrece una gran diversidad de servicios y recursos, como, por ejemplo, el acceso a plataformas digitales.

Interfaz:

La interfaz de usuario es el medio con que el usuario puede comunicarse con una máquina, equipo, computadora o dispositivo, y comprende todos los puntos de contacto entre el usuario y el equipo. Las interfaces básicas de usuario son aquellas que incluyen elementos como menús, ventanas, contenido gráfico, cursor, los beeps y algunos otros sonidos que la computadora hace, y en general, todos aquellos canales por los cuales se permite la comunicación entre el ser humano y la computadora.

IP Pública:

Es la que tiene asignada cualquier equipo o dispositivo conectado de forma directa a Internet. Algunos ejemplos son: los servidores que alojan sitios web como Google, los router o modems que dan a acceso a Internet, otros elementos de hardware que forman parte de su infraestructura, etc.

LED:

Un led es un componente electrónico cuya función principal es convertir la energía eléctrica en una fuente luminosa, la palabra led proviene del acrónimo inglés Light

Emmiting Diode o diodo emisor de luz. Específicamente un led corresponde a un tipo especial diodo el cual transforma la energía eléctrica en luz, su principio de funcionamiento se basa en la emisión de fotones (luz) cuando los electrones portadores de la electricidad atraviesan el diodo, dicho fenómeno se conoce como electroluminiscencia.

Lenguajes de Programacion: Un lenguaje de programación consiste en todos los símbolos, caracteres y reglas de uso que permiten a las personas "comunicarse" con las computadoras.

Máquina Virtual:

Una máquina virtual (VM) es una implementación de software de un ambiente de computación en el que se puede instalar y ejecutar un sistema operativo (OS) o programa. La máquina virtual normalmente emula un ambiente de computación físico pero las demandas de CPU, memoria, disco duro, red y otros recursos de hardware son gestionadas por una capa de virtualización que traduce estas solicitudes a la infraestructura de hardware físico subyacente.

Memoria RAM:

La memoria RAM es la memoria principal de un dispositivo donde se almacena programas y datos informativos. Las siglas RAM significan "Random Access Memory". La memoria RAM es conocida como memoria volátil lo cual quiere decir que los datos no se guardan de manera permanente, es por ello, que cuando deja de existir una fuente de energía en el dispositivo la información se pierde. Asimismo, la memoria RAM puede ser reescrita y leída constantemente.

Microcontrolador:

Es un circuito integrado programable, capaz de ejecutar las órdenes grabadas en su memoria. Está compuesto de varios bloques funcionales, los cuales cumplen una tarea específica. Un microcontrolador incluye en su interior las tres principales unidades funcionales de una computadora: unidad central de procesamiento, memoria y periféricos de entrada/salida.

MVC (Modelo Vista Controlador):

Es un patrón de arquitectura de software que separa los datos y la lógica de negocio de una aplicación de la interfaz de usuario y el módulo encargado de gestionar los eventos y las comunicaciones.

Navegador:

Es un programa o software, por lo general gratuito, que permite visualizar páginas web a través de Internet además de acceder a otros recursos de información alojados en servidores web, como pueden ser videos, imágenes, audio y archivos XML.

Parche:

Son cambios aplicados a un programa para corregir errores, agregarle funcionalidad, actualizarlo, etc.

RFID:

Identificación por Radio Frecuencia, es una tecnología que permite la captura automática de datos identificando objetos mediante el uso Tags de ondas de radio frecuencia y readers o lectores manuales para la captura de información.

Servidor Web:

Un servidor web es un programa que se ejecuta continuamente en un computador, manteniéndose a la espera de peticiones de ejecución que le hará un cliente o un usuario de Internet. El servidor web se encarga de contestar a estas peticiones de forma adecuada, entregando como resultado una página web o información de todo tipo de acuerdo a los comandos solicitados.

Socket:

Un socket es un mecanismo que permite la conexión entre distintos procesos, habitualmente se utilizan para establecer comunicaciones entre distintas máquinas que estén conectadas a través de la red. Un identificador de socket es una pareja formada por una dirección IP y un puerto. Cuando un programa crea un socket puede solicitarle al sistema operativo que asocie un número de puerto con el socket.

Software:

Se considera que el software es el equipamiento lógico e intangible de una computadora. el concepto de software abarca a todas las aplicaciones informáticas, como los procesadores de textos, las planillas de cálculo y los editores de imágenes, etc.

Tablet:

Es una computadora portátil más grande que un smartphone pero más pequeña que una netbook. Se caracteriza por contar con pantalla táctil, no se necesita mouse ni teclado.

Transistor: El transistor es un dispositivo electrónico semiconductor utilizado para entregar una señal de salida en respuesta a una señal de entrada. Cumple funciones de amplificador, oscilador, conmutador o rectificador. El término transistor es la contracción en inglés de transfer resistor

ZIP:

ZIP es un formato de fichero que comprime cada uno de los archivos de forma separada. zip apela a diversos algoritmos de compresión y soporta un sistema de cifrado que se basa en una clave única.

Referencias

- [1] TANENBAUM, ANDREW S. Sistemas operativos modernos. Tercera edición. Pearson Educación, México, 2009
- [2] ANDREW S. TANENBAUM y DAVID J. WHETHERALL. Sistemas operativos modernos. Quinta edición. Pearson Educación, México, 2009
- www.codelsa.com
- <https://nodejs.org/>
- www.ibm.com/developerworks/ssa/opensource/library/os-nodejs/
- www.nodebeginner.org/index-es.html
- https://media.blackhat.com/bh-us-12/Turbo/Galbreath/BH_US_12_Galbreath_Libinjection_Slides.pdf
- <http://www.imperva.com/Products/DatabaseFirewall>
- <https://www.debian.org/doc/manuals/packaging-tutorial/packaging-tutorial.en.pdf>
- <http://www.linux.com/learn/tutorials/307110-easy-package-creation-with-checkinstall>
- <http://www.linux.com/learn/tutorials/307110-easy-package-creation-with-checkinstall>
- <http://revista.seguridad.unam.mx/numero-18/firewall-de-bases-de-datos>
- <http://rudiments.sourceforge.net/about.html>
- <http://www.darkreading.com/risk/hackers-choice-top-six-database-attacks/d/d-id/1129481?>
- <http://www.redseguridad.com/opinion/articulos/la-importancia-y-la-necesidad-de-proteger-la-informacion-sensible>
- <http://www.webupd8.org/2010/01/how-to-create-deb-package-ubuntu-debian.html>
- <http://www.webupd8.org/2010/01/how-to-create-your-own-gpg-key.html>
- <https://wiki.debian.org/HowToPackageForDebian>
- <http://www.linuxuser.co.uk/tutorials/make-your-own-deb-and-rpm-packages>
- <http://www.modsecurity.org/>
- <http://askubuntu.com/questions/140998/compiling-source-into-a-deb-package>
- <http://santi-bassett.blogspot.mx/2014/07/how-to-create-debian-package.html>
- <https://github.com/tonejo/firewall-sql>
- <http://www.cyberciti.biz/faq/ubuntu-linux-how-do-i-install-deb-packages/>

<http://www.linuxjournal.com/content/using-checkinstall-build-packages-source>

<http://www.postgresql.org.es/>

<https://www.android.com/>

http://www.tutorialspoint.com/android/android_architecture.htm

<https://source.android.com/devices/>

<http://www.javatpoint.com/android-software-stack>

<https://www.oracle.com/java/index.html>

<https://developer.apple.com/library/ios/documentation/Miscellaneous/Conceptual/iPhoneOSTechOverview/Introduction/Introduction.html>

<http://eve-ingsistemas-u.blogspot.mx/2012/04/sistemas-operativos-moviles-ios.html>

<http://dmovilesras.blogspot.mx/2012/05/ios-apple.html>

<https://www.debian.org/index.es.html>

<https://www.linux.com/>

<https://www.digitalocean.com/>

<http://www.iingen.unam.mx/es-mx/Investigacion/Proyectos/Documents/HiperPuma2013.pdf>

<http://www.sitepoint.com/working-with-geolocation-and-google-maps-api/>

<https://developers.google.com/maps/documentation/javascript/events?hl=es>

<http://www.funcion13.com/calculando-distancias-y-areas-con-google-maps-v3/>

<http://www.phonegapspain.com/topic/google-maps-y-calculo-de-distancia-entre-dos-puntos-geograficos/>

<http://robdodson.me/deploying-your-first-node-dot-js-and-socket-dot-io-app-to-heroku/>

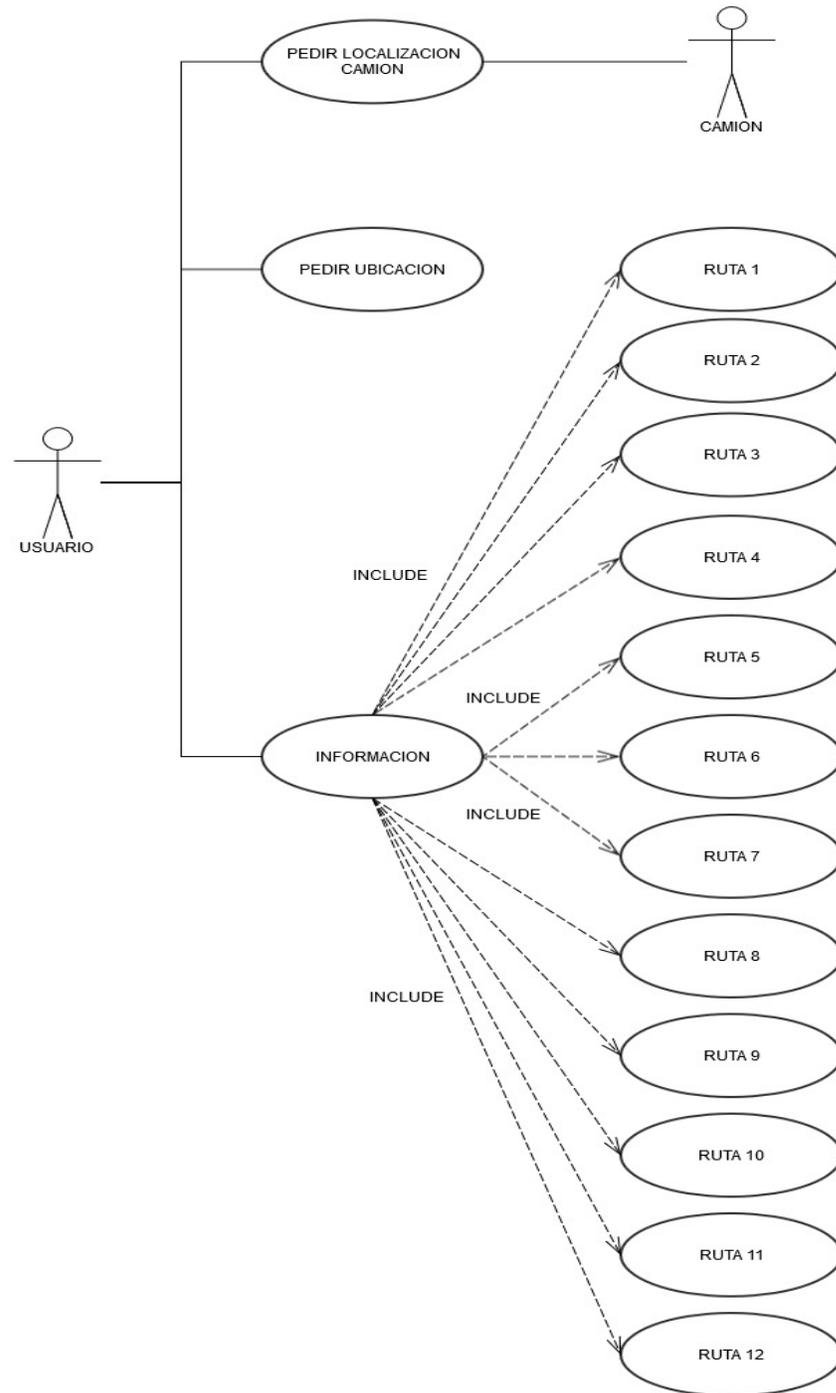
<https://nodejs.org/en/>

<http://socket.io/>

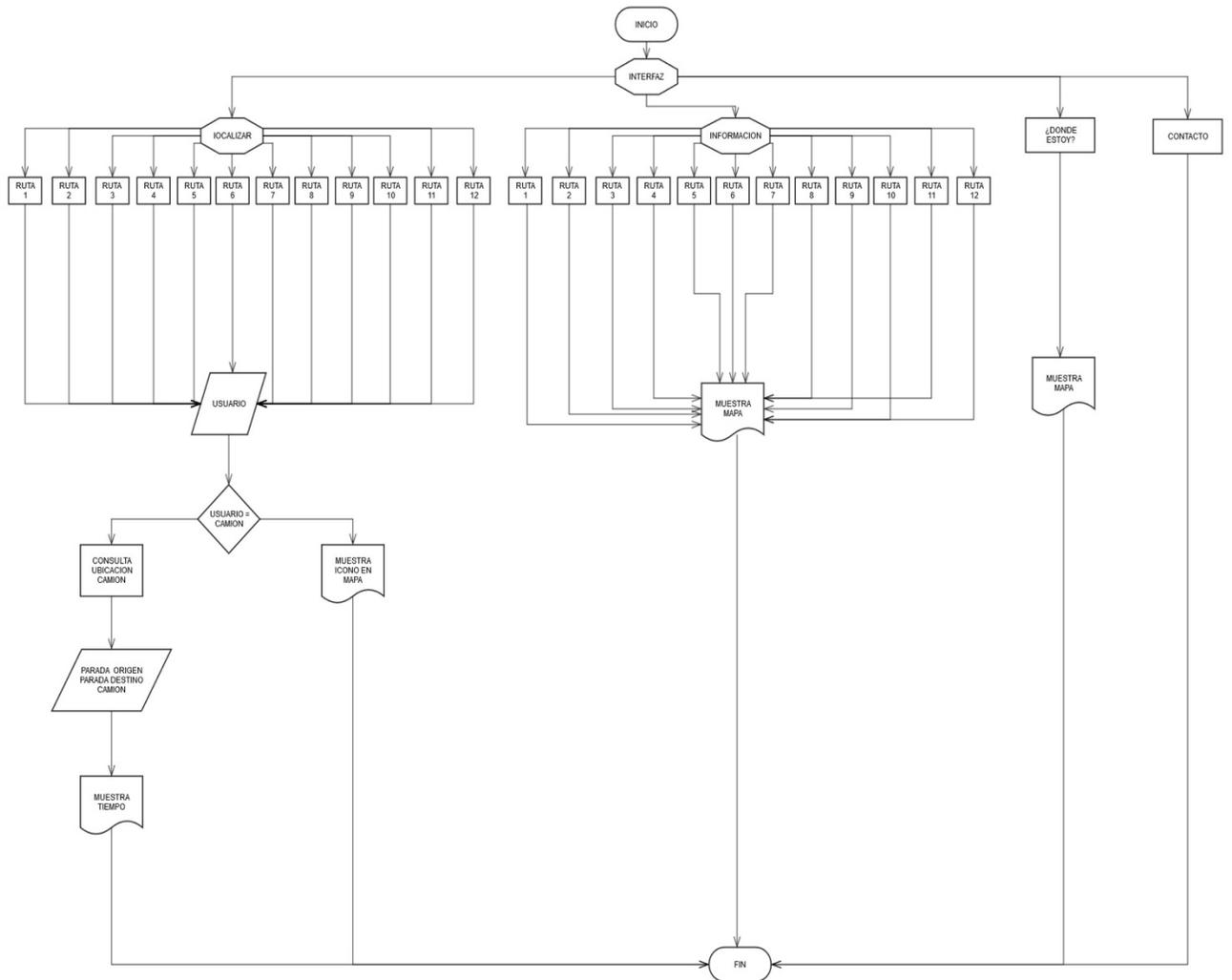
<http://www.javascripter.net/faq/index.htm>

Anexos

Anexo 1. Diagrama UML de Casos de Uso



Anexo 2. Diagrama de Flujo



Anexo 3. Diagrama UML de

