

**UNIVERSIDAD POLITÉCNICA DE PACHUCA**

**MEMORIA TÉCNICA DE ESTADÍA**

**DESARROLLO DE LA APLICACIÓN TUZOBUS APP**

PRESENTADO POR:

**DERAHIN ROJO FUNETES**

PARA OBTENER EL TÍTULO DE

**INGENIERO EN SOFTWARE**

ASESORADO POR:

**BARTOLOMÉ FERNANDO MORENO VERA**

ASESOR ACADÉMICO

**ING. GELIER JONATHAN REYES VITE**

ASESOR TÉCNICO

# **AGRADECIMIENTOS**

*A mis padres*

# **RESUMEN**

Al día de hoy podemos encontrar en el mercado una gran cantidad de dispositivos móviles los cuales son utilizados por personas de cualquier rango de edad; es una realidad que actualmente el *internet of things,* hace que cada dispositivo tenga una conexión a internet independiente, la cual no se limita por el tamaño del mismo, esto facilita y cubre las necesidades del usuario para consultar información en línea.

En los siguientes párrafos de esta memoria de estadía se presenta el diseño y desarrollo de la aplicación “**TUZOBUS APP**”, en la cual se explica al usuario la manera en la que se desempeña dicho servicio. Al ser un transporte público de nueva introducción al municipio, se decide el desarrollo de esta herramienta para facilitar el aprendizaje de éste en donde se puede consultar las diferentes rutas que existen, dependiendo de la necesidad del usuario.

Se decide el desarrollo visual *user experience* donde se utiliza la normativa de diseño enfocado a la visualización del sistema operativo Android, la cual es metodología visual desarrollada por Google, donde se encuentra el lenguaje de diseño en el que predominan animaciones, transiciones, efectos de profundidad, así como iluminación y sombras, El entorno de *Material Design* es un elemento muy importante que juega dentro de la aplicación ya que sobre esta herramienta se pretende captar una mayor atención en el entorno visual de la aplicación.

El desarrollo del proyecto se administra mediante una herramienta móvil llamada *Trello,* la cual ayudó a mantener el equipo comunicado y un orden en los procesos que se estaban desarrollando para tener una misma sintaxis permitiendo que éste fuera monitoreado por cualquier integrante del equipo y dividiendo los módulos en un panel de información que constaba de: Planeación, Diseño, Desarrollo, Revisión y Finalizados.

Es utilizada la metodología *Scrum,* la cual fue ágil y flexible para gestionar el desarrollo ya que fue aumentando el número de módulos durante el mismo, los cuales eran inestables o cambiantes; esta metodología se adapta a los proyectos con requisitos inestables y para los que se requieren rapidez y flexibilidad, la cual fue perfecta para el desarrollo de la aplicación móvil.

**RESUMEN**

RESUMEN 2

# **CAPÍTULO 1**

En esta memoria de estadía se presenta el proyecto “Tuzobus App”, el cual es diseñado y desarrollado en oficinas de Creative Labs S. de R.L. de C. V., una empresa mexicana ubicada en la ciudad de Pachuca, Hidalgo, con dirección en calle Lázaro Cárdenas No. 100 Col. Adolfo López Mateos, C.P. 42094, la cual es innovadora en campos de realidad virtual y realidad aumentada; además ofrece servicios web y aplicaciones móviles, realizando investigaciones de tecnologías de la información así como visión computacional, ofreciendo su sello en cada uno de sus productos los cual son responsabilidad con el cliente así como la creatividad única en cada uno de sus proyectos.

En el año 2012, *Xólotl creative labs* decide crecer y posicionarse como una de las principales proveedoras de tecnología en el estado, comenzando con el desarrollo de videojuegos educativos adoptando nuevas tecnologías y diseños, los cuales lograron captar la atención de clientes potenciales que ayudarían al crecimiento de la misma, desafiándose cada vez más en proyectos los cuales conllevaban más nivel de dificultad en el desarrollo.

En este mismo año la empresa desarrolló diversas campañas de realidad aumentada mediante cámaras web así como Kinect para el seguimiento de objetos; con estas nuevas tecnologías logran desarrollos educativos más interactivos, demostrando su trabajo y creatividad en ferias de tecnología como *Expo Innova Universitaria*, gracias a esto se logra la venta a uno de sus principales clientes: *Kronos innovación*.

A principios de 2013 *Xólotl creative labs* decide aumentar sus capacidades de desarrollo invirtiendo tiempo en proyectos en el Game Engone de unity 3D, permitiendo exportar sus proyectos a nivel multiplataforma, tales como: Android, IOS. Cabe destacar que estos son los principales sistemas operativos para dispositivos móviles, así como sistemas operativos de escritorio como Linux, Windows y Mac.

La empresa vio la oportunidad de presentarles a más personas su producto, el cual constaba de una aplicación que ayudara al usuario a comprender cómo funcionaba el nuevo sistema de transporte público del municipio y al mismo tiempo ayudar a crecer las ventas de otros servicios de la empresa, realizando estudio de cuántas personas descargaban la aplicación, así como el rango de edad de las y los usuarios para segmentar mercado en otros productos.

# **CAPÍTULO 2**

# Marco contextual

Actualmente se encuentran aplicaciones con poca tecnología y falta de experiencia de usuario sobre herramientas que ayuden a entender cómo funciona el servicio de transporte público. Siendo Pachuca una ciudad de crecimiento y de gran turismo donde hay que promover el servicio de transporte público y educar a los nuevos usuarios sobre el uso correcto del mismo, aunque para personas que han utilizado servicios parecidos no sea necesario el conocimiento de las rutas, se incluirán publicaciones de información en este medio sobre el transporte público.

El nuevo servicio de transporte que ingresa a la ciudad de Pachuca es innovador, organizado y se encuentra la oportunidad del desarrollo de una aplicación móvil para el uso y conocimiento del mismo, ya que el 85% de la población tiene un dispositivo móvil donde puede consultar la información que le sea de interés; de tal manera se plantea utilizar ese porcentaje de usuarios para abarcar a más clientes sobre el desarrollo de aplicaciones móviles de nueva generación, con más aspecto visual y experiencia de usuario más atractiva.

Con base en esta modernización al sistema de transporte público, se pretende el desarrollo de una aplicación accesible a todos los usuarios, sin necesidad de una conexión a internet en todo momento, contando con modo *offline*, en la que el usuario pueda consultar las diferentes rutas y paradas que tiene el servicio de transporte TuzoBus, contando con una interfaz amigable y con temas de accesibilidad para los usuarios con debilidad visual.

TuzoBus App es una aplicación para dispositivos móviles con sistema operativo Android, enfocada al sistema de transporte público de la ciudad de Pachuca.

## **Descripción de problema**

El sistema de transporte público de Pachuca llegó sin suficiente información hacia los usuarios, lo que derivó el retraso de destinos por desconocimiento del servicio, así como de las rutas y las paradas que realiza en cada estación.

## **Objetivos**

El objetivo del proyecto es crear una herramienta que ayude a la población a conocer el sistema del transporte público, al contar con esta información, el usuario reduce el tiempo de traslado, ya que conocerá los puntos de ascenso y descenso que las unidades tienes especificadas; de igual manera la aplicación mostrara un aproximado en los tiempos que el usuario puede realizar de su ubicación actual a la estación más cercana.

### **Objetivo General**

Desarrollar una aplicación móvil utilizado en lenguaje de programación Java y Android Studio como entorno de desarrollo de la aplicación, y las herramientas de desarrollo de vida de software, para una correcta implementación en la evolución del proyecto, el cual facilitara a los usuarios en el conocimiento sobre cómo y dónde abordad las unidades de transporte público para llegar a su destino así como las correctas rutas del sistema de transporte metropolitano como sus horarios, conocer las rutas troncales y rutas alimentadoras que ofrece el servicio.

### **Objetivo específicos**

1. Identificar las limitaciones que tiene el usuario con el sistema de transporte público, y las áreas de oportunidad para el desarrollo.
2. Diseñar diagramas de casos de uso y de actividades dándole solución al problema.
3. Diseño y desarrollo de los módulos previamente evaluados.
4. Desarrollo y trazado de mapas con librería de google.
5. Diseño y desarrollo de la base de datos que se encargara de alimentar la información de la aplicación.

## **2.3** **Justificación**

El desarrollo de este proyecto surge de la innovación que pretende el área metropolitana ante un nuevo sistema de trasporte público el cual llega con desinformación por parte de los usuarios del mismo, de tal manera surge un área de oportunidad para el desarrollo de una aplicación móvil beneficiando así a los usuario para que puedan consultar las diferentes rutas que ofrece el sistema así como los horario y precios , contando con accesibilidad para abarcar a más usuarios adaptando el tamaño de letra y colores para un entorno visual más apropiado y cómodo dependiendo el tipo de usuario, así como con una sección de noticias donde se mantengan informados sobre los cambios repentinos o interrupciones del servicio.

La accesibilidad a las nuevas tecnologías, permite la correcta implementación de este proyecto ya que ahora se cuenta con una mejor conexión inalámbrica a las redes, así como con dispositivos inteligentes capaces de soportar aplicaciones robustas haciendo una disponibilidad de las misma en cualquier lugar y a cualquier hora sin tener que estas físicamente frente a una computadora o una conexión obligada a datos móviles.

## **2.4. Alcances y Limitaciones**

La aplicación “***Tuzo Bus***” está enfocada a informar al usuario sobre el funcionamiento del servicio de trasporte público ***Tuzo Bus***, encontrándose con un diseño atractivo y adaptable a la vista, así como una fluidez en la experiencia de usuario en la cual podrán consultarse las rutas, así como los horarios. Los alcances del proyecto son:

1. Los usuarios que utilicen la aplicación podrán consultar las rutas que deberán abordar para disminuir sus tiempos de traslado hacia su destino.
2. Los usuarios obtendrán la información necesaria acerca de horarios y los lugares donde hace parada cada unidad del ***Tuzo Bus***.
3. La aplicación contara con un módulo de noticias para que se mantengan informados sobre los cambios repentinos o interrupciones del servicio.
4. La aplicación está diseñada con diferentes temas para adaptar la experiencia visual en usuarios que no distingan a primera vista colores claros.
5. La aplicación esta optimizada para trabajar mejor con dispositivos Android 4.0 o superiores

Las limitaciones del proyecto son:

1. La aplicación solo es desarrollada para el sistema de transporte ***Tuzo Bus*** que se encuentra en la zona metropolitana de Hidalgo.
2. En su primer inicio se requiere una conexión a internet para la descarga de mapas necesarios para su correcto funcionamiento.
3. La aplicación solo se desarrolla bajo el sistema operativo Android

# **CAPÍTULO 3**

# Marco teórico

Dentro del proceso y desarrollo de la aplicación “***Tuzo Bus***” es necesario entender algunos términos, así como las herramientas utilizadas que fueron parte fundamental para el correcto desarrollo de dicha aplicación tales como: Material Design, Experiencia de usuario, Servicios Web basados en REST y JSON, Fragments, Actividades, Versión de Android, Librerías de google, Uml, por mencionar los más importantes.

## **3.1 Sistema operativo para dispositivos móviles**

Los sistema operativos de la actualidad se diseñan visualmente más atractivos y simplificados para el usuario aunque cuantas con más opciones y configuraciones que los SO anteriores los grandes sistemas se han enfocado en hacer su interfaz más agradable para el usuario, para otorgar así un funcionamiento más fluido y darle al usuario lo que él requiera con menos procedimientos o instrucciones, de igual manera estos SO se enfocan en las conexiones inalámbricas como las redes 4G LTE así como redes WiFI, la existencia de estos servicios hace más completo un dispositivo móvil al grado de que un usuario normal puede realizar actividades que comúnmente realizaría en una computadora personal eliminando con cada actualización limitaciones que pudieran limitar actividades.

Los sistemas operativos utilizados en los dispositivos móviles están basados en el modelo de capas.

Estas capas son:

**Kernel**: Es el que proporciona el acceso a los distintos elementos del hardware del dispositivo. Ofrece distintos servicios a las superiores como son los controladores o drivers para el hardware, la gestión de procesos, el sistema de archivos y el acceso y gestión de la memoria.

**Middleware**: Es el conjunto de módulos que hacen posible la propia existencia de aplicaciones para móviles. Es totalmente transparente para el usuario y ofrece servicios claves como el motor de mensajería y comunicaciones, códec multimedia, intérpretes de páginas web, gestión de dispositivos y seguridad.

Entorno de ejecución de aplicaciones: consiste en un gestor de aplicaciones y un conjunto de interfaces programables abiertas para facilitar la creación de software.

**Interfaz de usuario**: Facilita la interacción con el usuario y el diseño de la presentación visual de la aplicación. Los servicios que incluye son el de componentes gráficos (botones, pantallas, listas, etc.) y el del marco de interacción.

Los sistemas operativos móviles más usados son:

### ***Android***

El sistema operativo el cual es el líder del mercado con una actualización constante debido a estas actualizaciones es el sistema operativo más fragmentado hasta el momento teniendo 15 versiones publicadas desde el 2008 hasta la fecha siendo la última versión *Oreo* con Android 8.0 manejando un nivel de API 26, Fue diseñado principalmente para dispositivos móviles con pantalla táctil, como teléfonos inteligentes, tabletas y también para relojes inteligentes y televisores. Gracias a esto y a su agradable interfaz visual cuenta con una cuota de mercado cercana al 85% el sistema operativo de Google se caracteriza por ser abierto y disponible para cualquier fabricante interesando en utilizarlo para sus dispositivos móviles.

### ***IOS***

Es un sistema operativo más exclusivo y con más restricciones, ya que no permite la modificación de sus equipos lo cual asegura la integridad y la capacidad de proteger los datos del usuario ante algún robo de información haciendo sus quipos muy seguros, ofreciendo siempre una experiencia más estable y segura, Por otro lado, este sistema está limitado solo a sus unidades y no se puede instalar a empresas de terceros

### ***Windows*** ***Phone***

Microsoft que está realizando un gran esfuerzo financiero para posicionar Windows Phone como una tercera opción interesante para los consumidores después de que llegara tarde a la fiesta de los Smartphone. Su alianza con Nokia y su posterior compra le ha ayudado a darse a conocer mejor e ir arañando cuota de mercado a los dos líderes. Los últimos datos hablan de un 2,5% a nivel mundial.

Windows Phone destaca por su pantalla de inicio personalizable que ofrece las notificaciones de las apps de una manera sencilla y limpia. Además, ofrece una experiencia de usuario muy buena independientemente del tipo y gama de terminal en que se esté usando.

### ***Firefox OS***

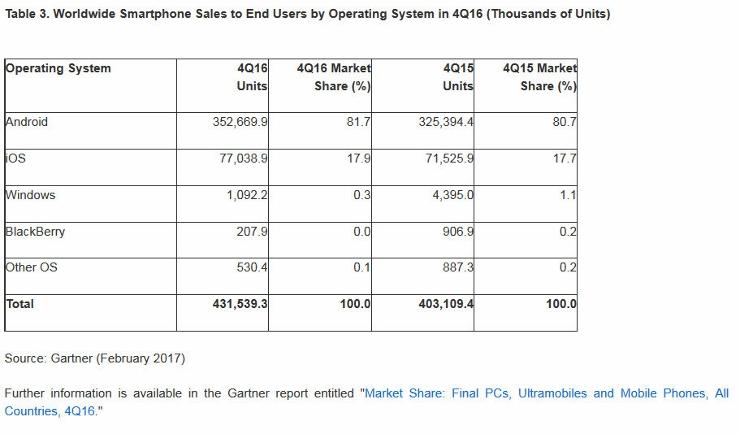
Un sistema operativo basado en HTML5 con núcleo Linux, de código abierto. Desarrollado por Mozilla Corporatión con apoyo de empresas como Telefónica. El sistema operativo está basado en Linux y usa la tecnología de Mozilla, Gecko. Se basa en estándares abiertos como por ejemplo HML5, CSS3 y JavaScript.

Conclusiones

Como conclusión puedo decir que todos los sistemas operativos móviles tienen sus ventajas y desventajas, actualmente Android ocupo gran parte del mercado debido a que sus aplicaciones tienen un entorno amigable y a su vez muchos usuarios pueden desarrollar sus propias aplicaciones, iOS por otro lado es más estable y presenta menos vulnerabilidades, Windows phone así mismo tiene una interfaz amigable, los demás sistemas operativos quizá menos utilizados pero dependiendo de la necesidad del usuario brindan sus respectivas ventajas.

## **3.2. Mercado de los sistemas operativos móviles**

El mercado de sistemas operativos (OS) móviles ha sido una carrera de sólo dos caballos desde hace tiempo, con la plataforma de Google, Android, en la delantera, seguido de Apple iOS. Eso es más cierto que nunca ahora que [Gartner ha publicado data](http://www.gartner.com/newsroom/id/3609817" \t "_blank) que apunta a que estas dos plataformas combinadas representan el 99.6% de la participación del mercado total.



## Sistema Operativo Android

En julio de 2005, Google adquirió Android Inc., una pequeña compañía de Palo Alto, California fundada en 2003. En aquel entonces, poco se sabía de las funciones de Android Inc. fuera de que desarrollaban software para teléfonos móviles. Esto dio pie a rumores de que Google estaba planeando entrar en el mercado de los teléfonos móviles.

En Google, el equipo liderado por Rubin desarrolló una plataforma para dispositivos móviles basada en el [núcleo Linux](https://es.wikipedia.org/wiki/N%C3%BAcleo_Linux) que fue promocionado a fabricantes de dispositivos y operadores con la promesa de proveer un sistema flexible y actualizable. Se informó que Google había alineado ya una serie de fabricantes de hardware y software y señaló a los operadores que estaba abierto a diversos grados de cooperación por su parte.

### 3.3.1 Arquitectura de Android

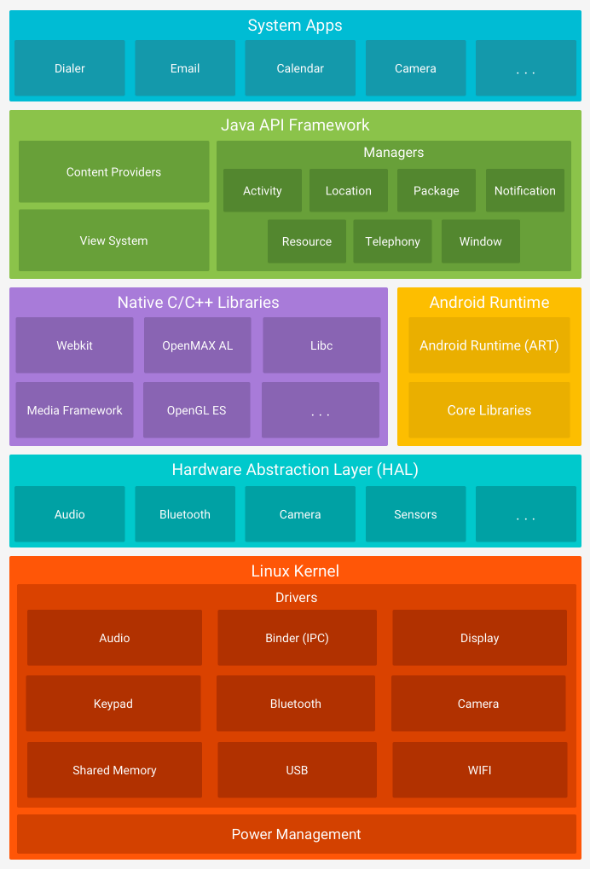
Android es una pila de software de código abierto basado en Linux creada para una variedad amplia de dispositivos y factores de forma. En el siguiente diagrama se muestran los componentes principales de la plataforma Android.

Figura 3.2: Arquitectura Android (Fuente https://developer.android.com/)

Apps del sistema: en Android se incluye un conjunto de apps centrales para correo electrónico, mensajería SMS, calendarios, navegación en Internet y contactos, entre otros elementos. Las apps incluidas en la plataforma no tienen un estado especial entre las apps que el usuario elije instalar; por ello, una app externa se puede convertir en el navegador web, el sistema de mensajería SMS o, incluso, el teclado predeterminado del usuario (existen algunas excepciones, como la app Settings del sistema).

Framework de la Java API: todo el conjunto de funciones del SO Android está disponible mediante API escritas en el lenguaje Java. Estas API son los cimientos que necesitas para crear apps de Android simplificando la reutilización de componentes del sistema y servicios centrales y modulares

Bibliotecas C/C++ nativas: muchos componentes y servicios centrales del sistema Android, como el ART y la HAL, se basan en código nativo que requiere bibliotecas nativas escritas en C y C++. La plataforma Android proporciona la API del framework de Java para exponer la funcionalidad de algunas de estas bibliotecas nativas a las apps.

Tiempo de ejecución de Android: para los dispositivos con Android 5.0 (nivel de API 21) o versiones posteriores, cada app ejecuta sus propios procesos con sus propias instancias del [tiempo de ejecución de Android (ART)](http://source.android.com/devices/tech/dalvik/index.html?hl=es-419). El ART está escrito para ejecutar varias máquinas virtuales en dispositivos de memoria baja ejecutando archivos DEX, un formato de código de bytes diseñado especialmente para Android y optimizado para ocupar un espacio de memoria mínimo. Crea cadenas de herramientas, como [Jack](https://source.android.com/source/jack.html?hl=es-419), y compila fuentes de Java en código de bytes DEX que se pueden ejecutar en la plataforma Android.

Capa de abstracción de hardware (HAL): la [capa de abstracción de hardware (HAL)](https://source.android.com/devices/index.html?hl=es-419#Hardware%20Abstraction%20Layer) brinda interfaces estándares que exponen las capacidades de hardware del dispositivo al [framework de la Java API](https://developer.android.com/guide/platform/index.html?hl=es-419" \l "api-framework) de nivel más alto. La HAL consiste en varios módulos de biblioteca y cada uno de estos implementa una interfaz para un tipo específico de componente de hardware, como el módulo de la cámara o de [bluetooth](https://source.android.com/devices/bluetooth.html?hl=es-419).

Kernel de Linux: la base de la plataforma Android es el kernel de Linux. Por ejemplo, el tiempo de ejecución de Android (ART) se basa en el kernel de Linux para funcionalidades subyacentes, como la generación de subprocesos y la administración de memoria de bajo nivel.

### 3.3.2. Versiones

El historial de versiones del sistema operativo Android se inició con el lanzamiento de Android beta en noviembre de 2007. La primera versión comercial (de prueba), Android 1.0, fue lanzada en septiembre de 2008 y la última hasta el momento en agosto del 2017 han pasado 10 años ya y hasta el momento son 10 versiones de Android cada una con el nombre de un postre en inglés y alfabéticamente. En la tabla 3.1 Versiones Android, encontraras un listado de las versiones publicadas con sus detalles

Tabla 3.1: Versión Android

|  |  |  |  |
| --- | --- | --- | --- |
| **NOMBRE DE VERSION** | **VERSION ANDROID** | **FECHA DE LANZAMIENTO** | **API** |
| Android 1.01​ | 1.0 | 23 de septiembre 2008 | 1 |
| Android 1.11​ | 1.1 | 9 de febrero 2009 | 2 |
| Cupcake | 1.5 | 27 de abril de 2009 | 3 |
| Donut | 1.6 | 15 de septiembre de 2009 | 4 |
| Eclair | 2.0–2.1 | 26 de octubre de 2009 | 5-7 |
| Froyo | 2.2–2.2.3 | 20 de mayo 2010 | 8 |
| Gingerbread | 2.3–2.3.7 | 6 de diciembre 2010 | 9–10 |
| Honeycomb2​ | 3.0–3.2.6 | 22 de febrero de 2011 | 11–13 |
| Ice Cream Sandwich | 4.0–4.0.5 | 18 de octubre 2011 | 14–15 |
| Jelly Bean | 4.1–4.3.1 | 9 de julio de 2012 | 16–18 |
| KitKat | 4.4–4.4.4, 4.4W–4.4W.2 | 31 de octubre de 2013 | 19–20 |
| Lollipop | 5.0–5.1.1 | 12 de noviembre de 2014 | 21–22 |
| Marshmallow | 6.0–6.0.1 | 5 de octubre de 2015 | 23 |
| Nougat | 7.0 - 7.1.2 | 15 de junio de 2016 | 24-25 |
| Oreo | 8.0 | 21 de agosto de 2017 | 26 |

## Arquitectura cliente-servidor

Este modelo Cliente/Servidor empezó a ser aceptado a finales de los 80’s. Su funcionamiento es sencillo: se tiene una máquina cliente, que requiere un servicio de una máquina servidor, y éste realiza la función para la que está programado.

Cliente

El cliente es el proceso que permite al usuario formular los requerimientos y pasarlos al servidor, se le conoce con el término front-end [15].

El Cliente normalmente maneja todas las funciones relacionadas con la manipulación y despliegue de datos, por lo que están desarrollados sobre plataformas que permiten construir interfaces gráficas de usuario (GUI), además de acceder a los servicios distribuidos en cualquier parte de una red.

Las funciones que lleva a cabo el proceso cliente se resumen en los siguientes puntos:

• Administrar la interfaz de usuario.

• Interactuar con el usuario.

• Procesar la lógica de la aplicación y hacer validaciones locales.

• Generar requerimientos de bases de datos.

• Recibir resultados del servidor.

• Formatear resultados.

Servidor

Es el proceso encargado de atender a múltiples clientes que hacen peticiones de algún recurso administrado por él. Al proceso servidor se le conoce con el término back-end [15].

El servidor normalmente maneja todas las funciones relacionadas con la mayoría de las reglas del negocio y los recursos de datos.

Las funciones que lleva a cabo el proceso servidor se resumen en los siguientes puntos:

• Aceptar los requerimientos de bases de datos que hacen los clientes.

• Procesar requerimientos de bases de datos.

• Formatear datos para trasmitirlos a los clientes.

• Procesar la lógica de la aplicación y realizar validaciones a nivel de bases de datos.

## Servicios web basados librería OKHttp REST

HTTP es la herramienta moderna para las aplicaciones que necesitan conexión. Una forma para intercambiar datos y medios. HTTP te ayuda a cargar datos de una forma eficiente con mayor velocidad y ahorrando datos de trasferencia.

OkHttp es un cliente HTTP que ofrece:

* La compatibilidad con HTTP/2 permite que todas las solicitudes en el mismo host compartan un socket.
* La agrupación de conexiones reduce la latencia de las solicitudes (si HTTP/2 no está disponible).
* GZIP reduce el tamaño de las descargas.
* Con el cache se omiten consultas repetidas.

OkHttp persiste cuando la red es problemática: se recuperará silenciosamente de problemas de conexión comunes. Si su servicio tiene múltiples direcciones IP, OkHttp intentará direcciones alternativas si falla la primera conexión. Esto es necesario para IPv4 + IPv6 y para servicios alojados en centros de datos redundantes. OkHttp inicia nuevas conexiones con las características modernas de TLS (SNI, ALPN), y vuelve a TLS 1.0 si falla el protocolo de enlace.

## JSON

JSON (JavaScript Object Notation - Notación de Objetos de JavaScript) es un formato ligero de intercambio de datos. Leerlo y escribirlo es simple para humanos, mientras que para las máquinas es simple interpretarlo y generarlo. Está basado en un subconjunto del [Lenguaje de Programación JavaScript](http://javascript.crockford.com/), [Standard ECMA-262 3rd Edition - Diciembre 1999](http://www.ecma-international.org/publications/files/ecma-st/ECMA-262.pdf). JSON es un formato de texto que es completamente independiente del lenguaje pero utiliza convenciones que son ampliamente conocidos por los programadores de la familia de lenguajes C, incluyendo C, C++, C#, Java, JavaScript, Perl, Python, y muchos otros. Estas propiedades hacen que JSON sea un lenguaje ideal para el intercambio de datos.

JSON está constituido por dos estructuras:

* Una colección de pares de nombre/valor. En varios lenguajes esto es conocido como un *objeto*, registro, estructura, diccionario, tabla hash, lista de claves o un arreglo asociativo.
* Una lista ordenada de valores. En la mayoría de los lenguajes, esto se implementa como arreglos, vectores, listas o secuencias.

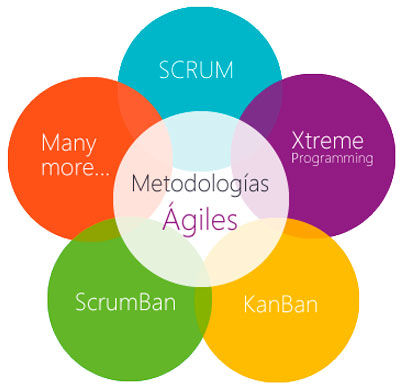
Estas son estructuras universales; virtualmente todos los lenguajes de programación las soportan de una forma u otra. Es razonable que un formato de intercambio de datos que es independiente del lenguaje de programación se base en estas estructuras.

## Desarrollo agil

El desarrollo ágil de software envuelve un enfoque para la toma de decisiones en los proyectos de software, que se refiere a métodos de ingeniería del software basados en el desarrollo iterativo e incremental, donde los requisitos y soluciones evolucionan con el tiempo según la necesidad del proyecto. Así el trabajo es realizado mediante la colaboración de equipos auto-organizados y multidisciplinarios, inmersos en un proceso compartido de toma de decisiones a corto plazo.

Cada iteración del ciclo de vida incluye: planificación, análisis de requisitos, diseño, codificación, pruebas y documentación. Teniendo gran importancia el concepto de "Finalizado" (Done), ya que el objetivo de cada iteración no es agregar toda la funcionalidad para justificar el lanzamiento del producto al mercado, sino incrementar el valor por medio de "software que funciona" (sin errores).

Los métodos ágiles enfatizan las comunicaciones cara a cara en vez de la documentación. La mayoría de los equipos ágiles están localizados en una simple oficina abierta, a veces llamadas "plataformas de lanzamiento" (bullpen en inglés). La oficina debe incluir revisores, escritores de documentación y ayuda, diseñadores de iteración y directores de proyecto. Los métodos ágiles también enfatizan que el software funcional es la primera medida del progreso. Combinado con la preferencia por las comunicaciones cara a cara, generalmente los métodos ágiles son criticados y tratados como "indisciplinados" por la falta de documentación técnica.



### 12 principios de agilidad

1. La prioridad es satisfacer al cliente mediante tempranas y continuas entregas de software que le aporten valor.
2. Dar la bienvenida a los cambios de requisitos. Se capturan los cambios para que el cliente tenga una ventaja competitiva.
3. Liberar software que funcione frecuentemente, desde un par de semanas a un par de meses, con el menor intervalo de tiempo posible entre entregas.
4. Los miembros del negocio y los desarrolladores deben trabajar juntos diariamente a lo largo del proyecto.
5. Construir el proyecto en torno a individuos motivados. Darles el entorno y apoyo que necesiten y confiar a en ellos para conseguir finalizar el trabajo.
6. El diálogo cara a cara es el método más eficiente y efectivo para comunicar información dentro de un equipo de desarrollo.
7. El software que funciona es la principal medida de progreso.
8. Los procesos ágiles promueven un desarrollo sostenible. Los promotores, desarrolladores y usuarios deberían ser capaces de mantener una paz constante.
9. La atención continua a la calidad técnica y al buen diseño mejora la agilidad.
10. La simplicidad es esencial.
11. Las mejores arquitecturas, requisitos y diseños surgen de los equipos que se auto organizan.
12. En intervalos regulares el equipo debe reflexionar sobre cómo ser más efectivo y según estas reflexiones ajustar su comportamiento.

Estos principios marcan el ciclo de vida de un desarrollo ágil, así como las prácticas y procesos a utilizar.

### ***SCRUM***

proceso en el que se aplican de manera regular un conjunto de buenas prácticas para trabajar colaborativamente, en equipo, y obtener el mejor resultado posible de un proyecto. Estas prácticas se apoyan unas a otras y su selección tiene origen en un estudio de la manera de trabajar de equipos altamente productivos.

En Scrum se realizan entregas parciales y regulares del producto final, priorizadas por el beneficio que aportan al receptor del proyecto. Por ello, Scrum está especialmente indicado para proyectos en entornos complejos, donde se necesita obtener resultados pronto, donde los requisitos son cambiantes o poco definidos, donde la innovación, la competitividad, la flexibilidad y la productividad son fundamentales.

## **Lenguaje Unificado de Modelado**

El Lenguaje Unificado de Modelado (UML) fue creado para forjar un lenguaje de modelado visual común y semántica y sintácticamente rico para la arquitectura, el diseño y la implementación de sistemas de software complejos, tanto en estructura como en comportamiento. UML tiene aplicaciones más allá del desarrollo de software, p. ej., en el flujo de procesos en la fabricación

## ***Diagramas del UML***

está compuesto por diversos elementos gráficos que se combinan para conformar diagramas. Debido a que el UML es un lenguaje, cuenta con reglas para combinar tales elementos. La finalidad de los diagramas es presentar diversas perspectivas de un sistema, a las cuales se les conoce como modelo. Recordemos que un modelo es una representación simplificada de la realidad; el modelo UML describe lo que supuestamente hará un sistema, pero no dice cómo implementar dicho sistema.

## ***Diagrama de Clases***

Volver Los diagramas de clases describen la estructura estática de un sistema. Las cosas que existen y que nos rodean se agrupan naturalmente en categorías. Una clase es una categoría o grupo de cosas que tienen atributos (propiedades) y acciones similares. Un ejemplo puede ser la clase Aviones que tiene atributos como el modelo de avión, la cantidad de motores, la velocidad de crucero y la capacidad de carga útil. Entre las acciones de las cosas de esta clase se encuentran: acelerar, elevarse, girar, descender, desacelerar.

## ***Diagrama de Actividades***

ilustra la naturaleza dinámica de un sistema mediante el modelado del flujo ocurrente de actividad en actividad. Una actividad representa una operación en alguna clase del sistema y que resulta en un cambio en el estado del sistema. Típicamente, los diagramas de actividad son utilizados para modelar el flujo de trabajo interno de una operación.

## ***Diagrama de Casos de Uso***

Volver Un caso de uso es una descripción de las acciones de un sistema desde el punto de vista del usuario. Es una herramienta valiosa dado que es una técnica de aciertos y errores para obtener los requerimientos del sistema, justamente desde el punto de vista del usuario. Los diagramas de caso de uso modelan la funcionalidad del sistema usando actores y casos de uso. Los casos de uso son servicios o funciones provistas por el sistema para sus usuarios.

## ***Slack***

es una herramienta de colaboración en la que tu equipo y tú podéis trabajar para llevar a cabo vuestros proyectos. Slack te permite abarcar todos los detalles, desde los primeros pasos de proyectos hasta conversaciones sobre presupuestos.

Slack combina la comunicación y colaboración en equipo en un mismo lugar, haciendo que aumente la productividad en el trabajo, tanto en grandes empresas como en pequeños negocios. Vete tachando los elementos de tu lista de tareas pendientes y haz que avancen los proyectos, juntando las personas, conversaciones, herramientas e información que necesitas. Slack se encuentra disponible en cualquier dispositivo, de forma que puedas encontrar y acceder a tu equipo y tu trabajo, tanto si estás sentado en tu escritorio como si te encuentras fuera de la oficina.

# **CAPÍTULO 4**

# **Herramientas utilizadas**

En este capítulo se describe de manera general las herramientas y tecnologías utilizadas para el desarrollo de la aplicación “Tuzo Bus”.

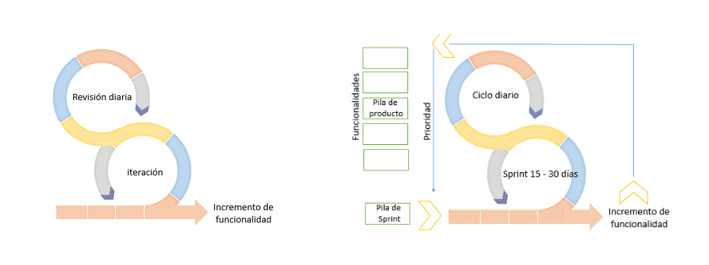


## **Metodología de desarrollo Ágil: SCRUM**

La metodología SCRUM se utilizó para la realización del proyecto “Tuzo Bus”. Ya que es una metodología ágil y flexible para gestionar el desarrollo de software, fue aplicada para cada tarea planeada mediante iteraciones, denominadas sprints, con una duración de 15 días. El resultado de cada sprint es el incremento ejecutable para ser mostrado.

Esta metodología es simple, no se basa en seguir un procedimiento fijo, ya que se va adaptando a las circunstancias de la evaluación del proyecto. Y la manera en que se orientada a las personas más que a los procesos, principalmente con el objetivo de que los miembros del equipo trabajen juntos y de forma eficiente obteniendo productos complejos y sofisticados.

En la Figura 4.1: Estructura y ciclo SCRUM, muestra las tareas planeadas que serán la pila de sprint, las cuales pasarán por un proceso central Scrum, donde se realiza el desarrollo de la tarea planeada, una vez concluida, se documenta y pasa a la pila de proceso.



## **Android Studio**

Android Studio es el entorno de desarrollo integrado (IDE) oficial para el desarrollo de aplicaciones para Android y se basa en [IntelliJ IDEA](https://www.jetbrains.com/idea/). Además del potente editor de códigos y las herramientas para desarrolladores de IntelliJ, Android Studio ofrece aún más funciones que aumentan tu productividad durante la compilación de apps para Android, como las siguientes:

* Un sistema de compilación basado en Gradle flexible
* Un emulador rápido con varias funciones
* Un entorno unificado en el que puedes realizar desarrollos para todos los dispositivos Android
* Instant Run para aplicar cambios mientras tu app se ejecuta sin la necesidad de compilar un nuevo APK
* Integración de plantillas de código y GitHub para ayudarte a compilar funciones comunes de las apps e importar ejemplos de código
* Gran cantidad de herramientas y frameworks de prueba
* Herramientas Lint para detectar problemas de rendimiento, usabilidad, compatibilidad de versión, etc.
* Compatibilidad con C++ y NDK
* Soporte incorporado para [Google Cloud Platform](http://developers.google.com/cloud/devtools/android_studio_templates/?hl=es-419), lo que facilita la integración de Google Cloud Messaging y App Engine

## **Jira Software**

es una herramienta ágil de gestión de proyectos compatible con cualquier metodología ágil, ya sea scrum, kanban o la tuya propia. Desde tableros hasta informes ágiles, puedes planificar, supervisar y gestionar todos los proyectos de desarrollo de software ágil con una sola herramienta. Elige una metodología para ver cómo Jira Software puede hacer que tu equipo publique software de calidad con mayor rapidez.

## **GitHub**

es una [forja](https://es.wikipedia.org/wiki/Forja_(software)) (plataforma de desarrollo colaborativo) para alojar proyectos utilizando el sistema de [control de versiones](https://es.wikipedia.org/wiki/Control_de_versiones) [Git](https://es.wikipedia.org/wiki/Git" \o "Git). Se utiliza principalmente para la creación de código fuente de programas de computadora. El software que opera GitHub fue escrito en [Ruby on Rails](https://es.wikipedia.org/wiki/Ruby_on_Rails). Desde enero de [2010](https://es.wikipedia.org/wiki/2010), GitHub opera bajo el nombre de GitHub, Inc. Anteriormente era conocida como Logical Awesome LLC. El código de los proyectos alojados en GitHub se almacena típicamente de forma [pública](https://es.wikipedia.org/wiki/C%C3%B3digo_abierto), aunque utilizando una cuenta de pago, también permite hospedar repositorios privados.

## **StarUML**

Para la aplicación se requirió utilizar la herramienta StarUML para analizar bien el proyecto, entender lo que el desarrollador realizaría, examinado los requerimientos del cliente, mediante diagramas de casos de uso. Siendo un lenguaje gráfico permitiendo visualizar, especificar, construir y documentar la aplicación “NEEDS”. (Bennett, 2007)

## **Lenguaje de programación JAVA**

El lenguaje para el desarrollo de la aplicación NEEDS es Java, un lenguaje multiplataforma con el cual se pueden desarrollar programas que se ejecuten sin problemas en sistemas operativos como Windows, Linux, Mac, Unix, permite crear programas que se ejecuten en varios dispositivos como computadoras, teléfonos móviles, es el único lenguaje soportado por el 100% de los teléfonos móviles. Java es uno de los lenguajes más demandados por las empresas y mejor remunerado. (Allende, 2001)

# **CAPÍTULO 5**

# **Desarrollo de actividades**

En este capítulo se detallan los procesos y las actividades realizadas en el proceso de desarrollo de la aplicación Tuzo Bus, la cual se basa en consultas sociales en donde se utilizaron herramientas específicas las cuales son mencionadas anteriormente utilizando la ISO/IEC 12207 Software life Cycle Processes en el estándar para los procesos de ciclo de vida, utilizando los procesos de desarrollo de software.

## **Adquisición de requerimientos**

Estos requerimientos se fueron obteniendo en juntas con el líder del proyecto de la empresa Xólotl Creative Labs S. de R. L. de C. V., para el desarrollo de la APP “Tuzo Bus”:

1. Al abrir la aplicación por primera vez se tendrán que seleccionar unas opciones, con el fin de recabar datos para marketing.

* Sexo
* Edad

1. El usuario continuara con la pantalla principal de la aplicación donde se muestra en un mapa todas las rutas troncales y alimentadoras del servicio de transporte público marcando cada ruta con colores diferentes e imágenes de estaciones
2. El usuario podrá encontrar dentro del menú principal accesos a diferentes módulos tales como:

* Mapa de red muestra el mapa completo de las estaciones por donde pasa la unidad del Tuzo Buz.
* Ruta Exprés muestra una lista despegable por donde para cada ruta exprés, mostrando las estaciones donde hace parada.
* Ruta Alimentadora muestra una lista con cada ruta alimentadora y con la información de los lugares donde hace parada, así como la opción de poder visualizar el mapa de la ruta alimentadora seleccionada.
* Horarios muestra información con los diferentes horarios de los días de la semana que opera el servicio de transporte Tuzo Bus
* En el módulo Sugerencias se da la opción de poder escribir una sugerencia anónima para que todos los usuarios puedan compartir información.
* Información conecta al Twitter de la página oficial del tuzo bus, para que los usuarios estén informados sobre posibles cambios que pudieran ocurrir sobre el servicio.
* El módulo Compartir App nos permite compartir el link de la tienda de google en aplicaciones sociales, para poder descarar la aplicación
* Configuración muestra diferentes ajustes para que el usuario navegue por la aplicación con mayor accesibilidad contando con ajustes para tamaño de letra, tema de colores, sugerencia de ruta, tipo de mapa.

1. Los usuarios que utilicen la aplicación tendrán conocimiento de las diferentes rutas troncales, así como de las rutas exprés y conocer cuál es la ruta más cercana, así como una sugerencia de cómo llegar a la estación desde su ubicación ya sea en automóvil o caminando informando también el tiempo de recorrido y la distancia de la estación.
2. La información obtenida (email, edad, sexo) será recabada con fines publicitarios y de marketing.

## **Administración del personal del proyecto**

La administración del personal del proyecto es importante para la realización de cualquier software ayuda a desarrollar y motivar una fuerza de trabajo socialmente representativa y competente para que trabaje eficazmente y llegar a concluir la realización de este. En el proyecto “Tuzo Bus”, la organización fue el elemento principal para su desarrollo los roles de la empresa son los siguientes:

DESARROLLADOR 1

DESARROLLADOR 2

LÍDER DE PROYECTO

 ÁREA DE MODELADO

 ÁREA DE DISEÑO WEB

ÁREA DE DESARROLLO MOBIL

ÁREA DE VIDEOJUEGOS

MARKETING

SOCIOS DE LA EMPRESA

 DESARROLLO

MERCADOTECNIA

Figura 5.1: **Organigrama de la empresa.**

### Selección de herramientas

Para la realización de la aplicación “Tuzo Bus” fue necesario la selección de las herramientas a utilizar para el desarrollo, ya mencionadas en el capítulo 4.

### 5.2.2 Requerimientos de hardware y software

Para el desarrollo óptimo de la aplicación “Tuzo bus” es necesario especificar los requerimientos de hardware y software.

* Software
* Android Studio.
* GitHub Desktop.
* Java Development kit (JDK).
* Genymotion Desktop.
* FileZilla.
* Hardware
* **Windows** 7/8/10 (32 o 64 bits).
* 4 GB de RAM (8 GB de RAM recomendado).
* 15 GB de espacio libre mínimo (20 GB recomendado).
* Resolución minima de 1.280 x 800.
* Java 8.
* 64 bits y procesador Intel (emulador).

### 5.2.3 Programación de tiempo

la programación el tiempo es una herramienta para administrar el proyecto limitando la entrega en tiempos establecidos en el cual se van desglosando los módulos que conforman el proyecto, administrando el tiempo de acuerdo a la complejidad de cada uno de ellos.

En la figura 5.1: Diagrama de Gantt tareas realizadas, se detallan la programación de módulos desarrollados en la aplicación de Tuza Bus y la tabla 5.2: programación de tiempos, se detalla el tiempo de desarrollo de cada actividad evidenciando los días de duración de cada una de ellas.

Figura 5.1: Diagrama de Gantt tareas realizadas.

Tabla 5.2: Programación de tiempo

|  |  |  |  |
| --- | --- | --- | --- |
| ACTIVIDADES | FECHA DE INICIO | DURACION  DIAS | FECHA DE TERMINACION |
| Registro de Usuarios | 21/01/2016 | 12 | 02/02/2016 |
| Trazo de rutas en Google Maps | 02/02/2016 | 12 | 13/02/2016 |
| Menú Delegable | 16/02/2016 | 12 | 27/02/2016 |
| Servicios Web | 02/03/2016 | 12 | 13/03/2016 |
| Base de Datos | 16/03/2016 | 18 | 03/04/2016 |
| ListView con rutas | 06/04/2016 | 12 | 17/04/2016 |
| Mapa de ubicación | 20/04/2016 | 12 | 01/05/2016 |
| Estación más cercana | 04/05/2016 | 12 | 15/05/2016 |
| Configuración de Aplicación | 18/05/2016 | 12 | 29/05/2016 |

Tabla 5.2: Programación de tiempo

|  |  |
| --- | --- |
| FECHA | |
| inicio | final |
| **21/01/2016** | 29/05/2016 |

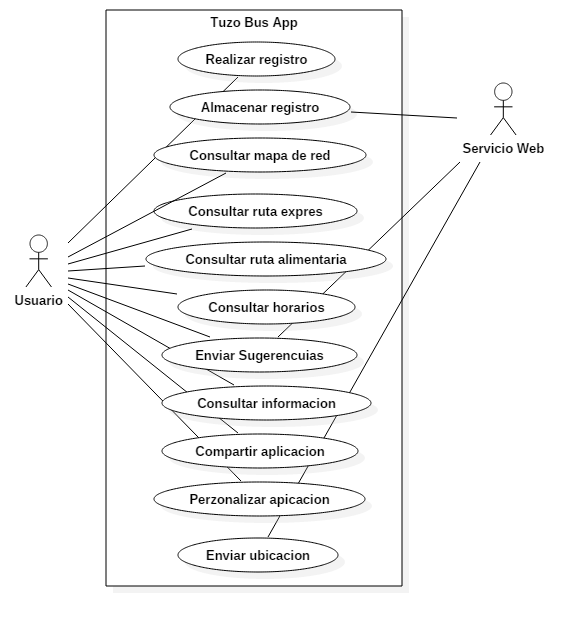
## Análisis de requerimientos

El análisis de requerimiento es el desarrollo de los procesos de la aplicación, es el estudio de las necesidades que va satisfacer la aplicación Tuzo Bus, organizados eficientemente en los siguientes diagramas.

### Diagrama de casos de usos

En la figura 5.3 se muestra el diagrama de casos de uso del proyecto Tuzo bus, donde se detallan de manera general los procesos que el usuario tiene que seguir con la aplicación tales como: Realizar el registro, consultar mapa de red, Consultar ruta Exprés Consultar ruta alimentadora, consultar horarios, enviar sugerencias, consultar información, compartir aplicación, personalizar aplicación, compartir ubicacion.

* Realizar registro: Este caso de uso consiste en que la primera vez que se inicia la aplicación saldrán un par de preguntas con respecto al sexo y la edad.
* Almacenar registro: Este caso de uso consiste en que una vez presionadas las opciones de registro el sistema enviará la información al servicio web, el cual se encargará de almacenar la información del usuario.
* Consultar mapa de red: Este caso de uso consiste en que el usuario podrá consultar con la ayuda de un mapa de google las ubicaciones de las estaciones del Tuzo Bus, así como las rutas que toman las unidades de alimentación, mostrando al usuario cual es la estación más cercana.
* Consultar ruta exprés: Ese caso de uso consiste en que el usuario podrá consultar mediante una lista desplegable la información las diferentes rutas exprés, así como las estaciones donde hacen parada.
* Consultar ruta alimentaria: Este caso de uso consiste en poder consultar las rutas alimentadoras donde se especifica las paradas de cada una de ellas, así como poder visualizar la ruta en el mapa.
* Consultar Horarios: Este caso de uso consiste en que el usuario puede consultar el horario de servicio de las unidades de transporte.
* Enviar Sugerencias: Este caso de uso consiste en que el usuario puede registrar una sugerencia que estará almacenada en la nube y podrá ser consultada por todos los usuarios.
* Consultar Información: Este caso de uso consiste en que el usuario podrá consultar información en la cuenta oficial de Twitter para enterarse de las ultimas noticias publicadas con respecto al servicio de transporte.
* Compartir ubicación: Este caso de uso consiste en que el usuario podrá compartir el link de descarga de la tienda Play Store, en aplicaciones de comunicación.
* Personalizar aplicación: Este caso de uso consiste en que el usuario podrá modificar el tipo de letra, así como los temas y los tipos de mapas que tiene la aplicación para una mayor accesibilidad.
* Enviar Ubicación. Este caso de uso consiste en que la aplicación detectará la ubicación del usuario y la compartirá con el servicio web para su almacenamiento.

Figura 5.3: Muestra el Diagrama de Casos de Uso de la aplicación Tuzo Bus.

### 5.3.2. Detalles de caso de uso

En las siguientes tablas se representan los detalles de casos de uso tales como Realizar el registro, consultar mapa de red, Consultar ruta Exprés Consultar ruta alimentadora, consultar horarios, enviar sugerencias, consultar información, compartir aplicación, personalizar aplicación, esto facilita al cliente observar el funcionamiento de la aplicación Tuzo Bus.

**Tabla 5.3.2.1:** Registro de usuario.

|  |  |
| --- | --- |
| **Nombre del caso de uso:** | **Registro de usuario** |
| **Actor:** | Usuario |
| **Precondición:** Contar con servicio de Internet, Abrir la APP “Tuzo Bus”, habilitar GPS. | |
| Flujo normal de los eventos:  1.- La aplicación muestra un mensaje de bienvenida.  2.- La aplicación solicita el registro del género mostrando dos opciones.  3.- El usuario selecciona el género de su preferencia.  4.- La aplicación envía el género al servicio web.  5.- La aplicación solicita el registro de la edad mostrando tres opciones.  6.- El usuario selecciona el rango de edad al que pertenece.  7.- La aplicación envía el rango de edad al servicio web. | |
| **Flujo Alternativo:**  8.- Si la APP no cuenta con servicio de internet, muestra el mensaje ***“No hay conexión a Internet”.***  9.- El usuario debe conectarse a internet para poder registrarse. | |
| ***Postcondición:***  Si el usuario ya se registró anteriormente al abrir la aplicación, se enviará a la pantalla principal.  El usuario cierra la APP “TuzoBuz”. | |

**Tabla 5.3:** consultar mapa de red.

|  |  |
| --- | --- |
| **Nombre del caso de uso:** | **consultar mapa de red** |
| **Actor:** | Usuario |
| **Precondición:** Contar con servicio de Internet, Abrir la APP “Tuzo Bus”, habilitar GPS. | |
| Flujo normal de los eventos:  1.- La aplicación muestra el mapa de google con las rutas alimentadoras trazada con colores diferentes cada una, así como las ubicaciones de las estaciones del Tuzo bus.  2.- el usuario podrá navegar dentro del mapa para para poder visualizar todas las rutas.  3.- la aplicación consulta la ubicación GPS y calcula la estación más cercada mostrando una ruta para llega a ella. | |
| **Flujo Alternativo:**  4.- la ubicación GPS debe estar activada.  5.- Si la APP no cuenta con servicio de localización GPS, muestra el mensaje ***“Activar GPS”.***  6.- Si la APP no cuenta con servicio de internet, muestra el mensaje ***“No hay conexión a Internet”.***  7.- El usuario debe conectarse a internet para poder registrarse. | |
| **Postcondición:**  El usuario tiene que estar registrado para poder acceder a la aplicación.  Si el usuario ya se registró anteriormente al abrir la aplicación, se enviará a la pantalla principal.  El usuario cierra la APP “TuzoBuz” | |

Tabla 5.5: Consultar ruta Exprés.

|  |  |
| --- | --- |
| **Nombre del caso de uso:** | **Consultar ruta Exprés** |
| **Actor:** | Usuario |
| **Precondición:** Contar con servicio de Internet, Abrir la APP “Tuzo Bus”, habilitar GPS. | |
| Flujo normal de los eventos:  1.- La aplicación muestra el mapa de red en la pantalla principal.  2.- El usuario da clic en el menú.  3.- La aplicación despliega el menú con los diferentes módulos.  2.- El usuario da clic en consultar ruta exprés.  3.- La aplicación muestra una lista despegable con cada una de las rutas exprés.  4.- El usuario da clic en la ruta a consultar.  5.- la aplicación despliega una lista con los nombres de las estaciones donde hace parada. | |
| **Flujo Alternativo:**  6.- Si la APP no cuenta con servicio de internet, muestra el mensaje ***“No hay conexión a Internet”.***  7.- El usuario regresa al menú principal. | |
| **Postcondición:**  El usuario tiene que estar registrado para poder acceder a la aplicación.  Si el usuario ya se registró anteriormente al abrir la aplicación, se enviará a la pantalla principal.  El usuario cierra la APP “TuzoBuz” | |

Tabla 5.4: Consulta de ruta alimentadora.

|  |  |
| --- | --- |
| **Nombre del caso de uso:** | **Consulta de ruta alimentadora.** |
| **Actor:** | Usuario |
| **Precondición:** Contar con servicio de Internet, Abrir la APP “Tuzo Bus”, habilitar GPS. | |
| **Flujo normal de los eventos:**  1.- la aplicación muestra la pantalla principal.  2.- El usuario da clic en el menú.  3.- La aplicación despliega el menú con los diferentes módulos.  2.- El usuario da clic en consultar ruta alimentadora.  3.- la aplicación muestra una lista desplegable de todas las rutas  4.- El usuario selecciona la ruta alimentadora a consultar.  5.- La aplicación despliega una lista con la ubicación de las paradas de la ruta alimentadora, así como la opción de poder consultar mapa.  6. El usuario selecciona la opción mostrar mapa.  7. la aplicación muestra la ruta trazada en un mapa . | |
| **Flujo Alternativo:**  8.- Si la APP no cuenta con servicio de internet, muestra el mensaje ***“No hay conexión a Internet”.***  9.- El usuario debe conectarse a internet para poder registrar los intereses personales. | |
| **Postcondición:**  El usuario tiene que estar registrado para poder acceder a la aplicación.  Si el usuario ya se registró anteriormente al abrir la aplicación, se enviará a la pantalla principal.  El usuario cierra la APP “TuzoBuz” | |

Tabla 5.5: consultar horarios

|  |  |
| --- | --- |
| **Nombre de casos de uso:** | **consultar horarios** |
| **Actor:** | Usuario |
| **Precondición:** Contar con servicio de Internet, Abrir la APP “Tuzo Bus”, habilitar GPS. | |
| **Flujo normal de los eventos**:  1.- la aplicación muestra la pantalla principal.  2.- El usuario da clic en el menú.  3.- La aplicación despliega el menú con los diferentes módulos.  4.- El usuario da clic en consultar horarios.  5.- la aplicación despliega un cuadro de alerta con la información de los horarios en los que prestan servicio las unidades del Tuzo Bus. | |
| **Flujo Alternativo**:  6.- El usuario regresa al menú principal. | |
| **Postcondición**:  El usuario tiene que estar registrado para poder acceder a la aplicación.  Si el usuario ya se registró anteriormente al abrir la aplicación, se enviará a la pantalla principal.  El usuario cierra la APP “Tuzo Bus” | |

Tabla 5.6: enviar sugerencias.

|  |  |
| --- | --- |
| **Nombre de casos de uso:** | **Enviar sugerencias** |
| **Actor:** | Usuario |
| **Precondición:** Contar con servicio de Internet, Abrir la APP “Tuzo Bus”, habilitar GPS. | |
| **Flujo normal de los eventos:**  1.- la aplicación muestra la pantalla principal.  2.- El usuario da clic en el menú.  3.- La aplicación despliega el menú con los diferentes módulos.  4.- El usuario da clic en sugerencias.  5.- la aplicación muestra un formulario con un cuadro de texto y un botón.  6.- el usuario escribe la sugerencia en el cuadro de texto.  7.- el usuario da clic en el botón de enviar que se encuentra en el formulario.  8.- La aplicación envía al servidor la información. | |
| **Flujo Alternativo:**  9.- Si la APP no cuenta con servicio de internet, muestra el mensaje ***“No hay conexión a Internet”.***  10.- El usuario debe conectarse a internet para poder envias sugerencias.  11.- El usuario regresa al menú principal. | |
| **Postcondición:**  El usuario tiene que estar registrado para poder acceder a la aplicación.  Si el usuario ya se registró anteriormente al abrir la aplicación, se enviará a la pantalla principal.  El usuario cierra la APP “TuzoBuz” | |

Tabla 5.7: consultar información.

|  |  |
| --- | --- |
| **Nombre de casos de uso:** | **consultar información** |
| **Actor:** | Usuario |
| **Precondición:** Contar con servicio de Internet, Abrir la APP “Tuzo Bus”, habilitar GPS. | |
| **Flujo normal de los eventos:**  1.- la aplicación muestra la pantalla principal  2.- El usuario da clic en el menú.  3.- La aplicación despliega el menú con los diferentes módulos.  2.- El usuario da clic en información  3.- la aplicación se conecta a la cuenta de Twitter oficial del Tuzo Bus #Tuzobus\_Oficial.  4.- el usuario puede navegar dentro de la cuenta de Twitter | |
| **Flujo Alternativo:**  14.- El usuario regresa al menú principal. | |
| **Postcondición:**  El usuario tiene que estar registrado para poder acceder a la aplicación.  Si el usuario ya se registró anteriormente al abrir la aplicación, se enviará a la pantalla principal.  El usuario cierra la APP “TuzoBuz” | |

Tabla 5.8: compartir aplicación

|  |  |
| --- | --- |
| **Nombre de casos de uso:** | **compartir aplicación** |
| **Actor:** | Usuario |
| **Precondición:** Contar con servicio de Internet, Abrir la APP “Tuzo Bus”, habilitar GPS. | |
| **Flujo normal de los eventos:**  1.- la aplicación muestra la pantalla principal  2.- El usuario da clic en el menú.  3.- La aplicación despliega el menú con los diferentes módulos.  2.- El usuario da clic compartir  3.- La aplicación muestra aplicaciones para compartir el link de descarga de google Play.  5.- El usuario selecciona la aplicación de su agrado.  6.- la aplicación re direcciona a la aplicación seleccionada para buscar el destino del mensaje. | |
| **Flujo Alternativo:**  **7**.- El usuario regresa al menú principal.  7.- Si la APP no cuenta con servicio de internet, muestra el mensaje ***“No hay conexión a Internet”.***  8.- El usuario debe conectarse a internet para poder compartir la aplicación. | |
| **Postcondición:**  El usuario tiene que estar registrado para poder acceder a la aplicación.  Si el usuario ya se registró anteriormente al abrir la aplicación, se enviará a la pantalla principal.  El usuario cierra la APP “TuzoBuz”. | |

Tabla 5.8: Personalizar aplicación

|  |  |
| --- | --- |
| **Nombre de casos de uso:** | **Personalizar aplicacion** |
| **Actor:** | Usuario |
| **Precondición:** Contar con servicio de Internet, Abrir la APP “Tuzo Bus”, habilitar GPS. | |
| **Flujo normal de los eventos:**  1.- la aplicación muestra la pantalla principal  2.- El usuario da clic en el menú.  3.- La aplicación despliega el menú con los diferentes módulos.  2.- El usuario da clic configuraciones.  5.- la aplicación muestra las opciones de configuración para el usuario: tamaño de letra, colores, tipo de mapa.  6.- El usuario configura la aplicación a su comodidad. | |
| **Flujo Alternativo:**  **7**.- El usuario regresa al menú principal.  8.- cada configuración regresa a la pantalla principal | |
| **Postcondición:**  El usuario tiene que estar registrado para poder acceder a la aplicación.  Si el usuario ya se registró anteriormente al abrir la aplicación, se enviará a la pantalla principal.  El usuario cierra la APP “TuzoBuz”. | |

Tabla 5.8: Compartir ubicación

|  |  |
| --- | --- |
| **Nombre de casos de uso:** | **Compartir ubicacion** |
| **Actor:** | Aplicación |
| **Precondición:** Contar con servicio de Internet, Abrir la APP “Tuzo Bus”, habilitar GPS. | |
| **Flujo normal de los eventos:**  1.- La aplicación ejecuta la búsqueda de coordenadas GPS.  2.- La aplicación Envía a servidor la ubicación.  3. Servicio web almacena ubicación de usuario. | |
| **Flujo Alternativo:**  **7**.- El usuario regresa al menú principal.  8.- cada configuración regresa a la pantalla principal  9.- La aplicación debe tener conexión a internet | |
| **Postcondición:**  Una vez enviados los parámetros mediante Post se detiene el servicio | |

### 5.3.3. Diagrama de actividades

En la Figura 5.4: Diagrama de actividades de la aplicación Tuzo Bus*,* donde se muestran las opciones con las que cuenta el usuario al utilizar la aplicación Tuzo Bus, en ellas se observan las actividades y decisiones que permiten descubrir lo que ocurre cuando el usuario elige una opción.

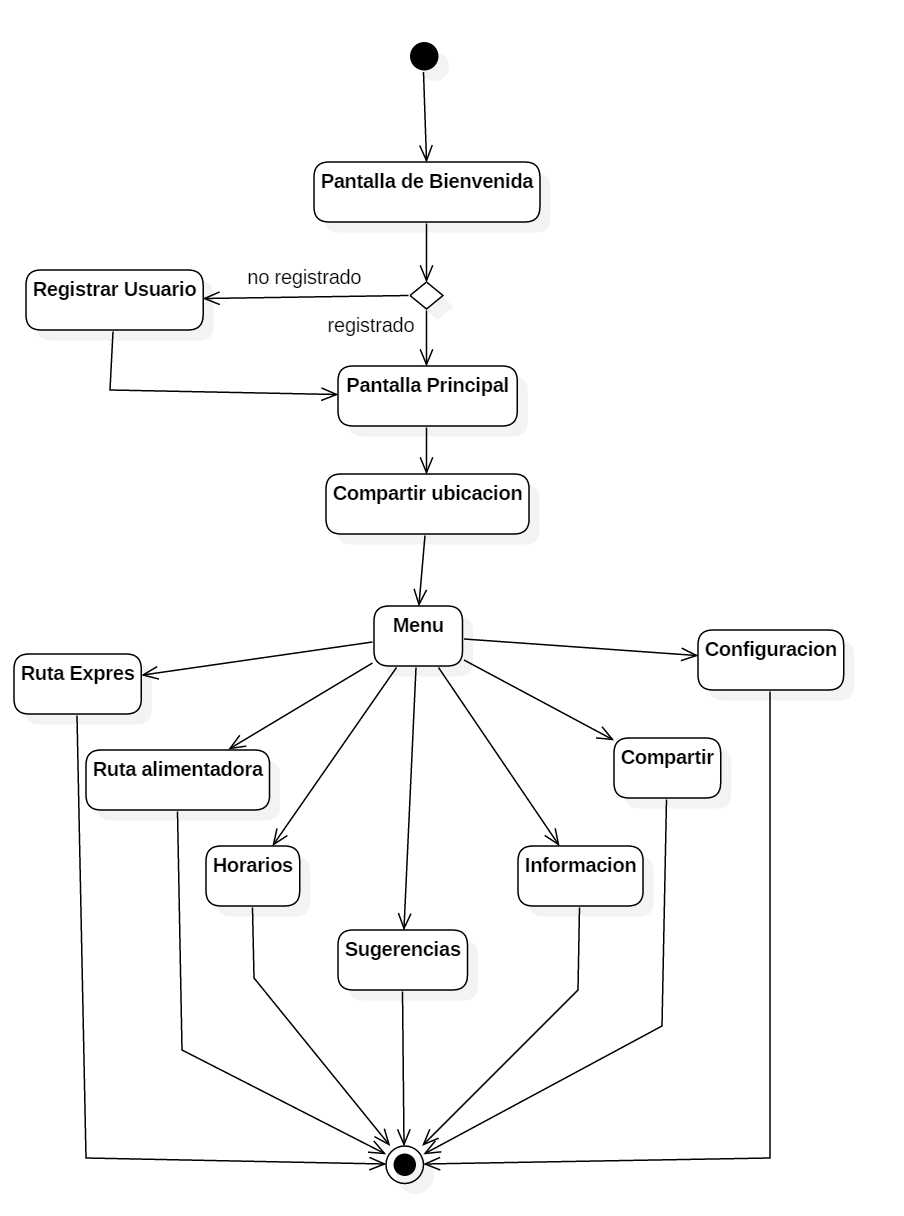


Figura 5.4: Diagrama de actividades de la aplicación Tuzo Bus.

### 5.3.4. Diseño preliminar de interfaces de usuario

Para una pre visualización en cuanto a diseño, se hicieron todos los bocetos de pantalla la aplicación “Tuzo Bus” para el diseño gráfico, fue necesario utilizar la herramienta *Moqups* un prototipo que proporciona al menos una parte de la funcionalidad del sistema y permite pruebas del diseño. En las siguientes figuras se podrá observa la secuencia.

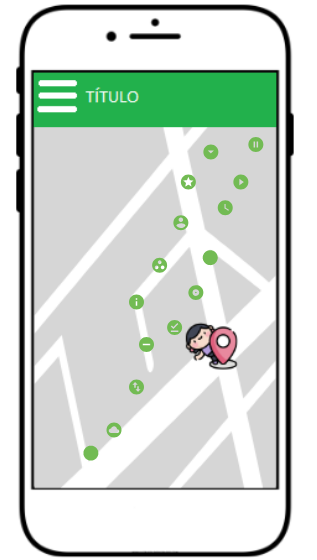
En la figura 5.5 Diseño preliminar de la pantalla de bienvenida, en la cual se pretende informar sobre el servicio con recomendaciones que se muestran aleatoriamente, así como horarios del servicio



Figura 5.5: Diseño preliminar de la pantalla de bienvenida.

En la figura 5.5 Diseño preliminar de la pantalla de registro en la cual el usuario podrá seleccionar su género y su rango de edad, siendo este proceso un registro rápido para que el usuario no demore en abrir la aplicación.



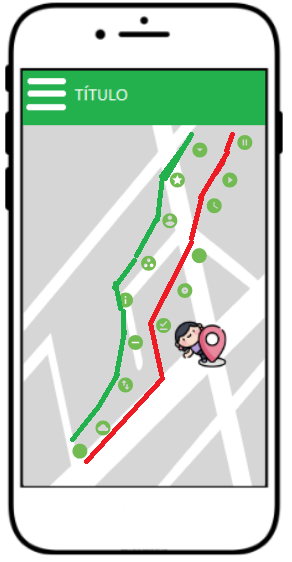
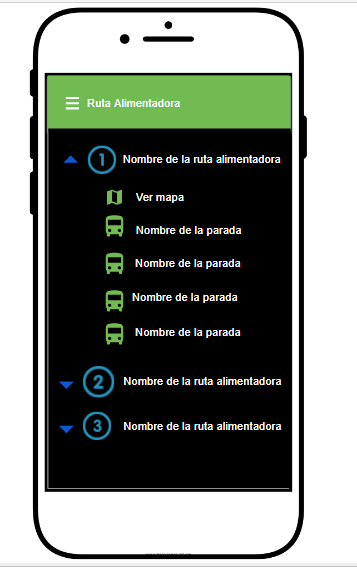
En la Figura 5.5 Diseño preliminar de la pantalla de Mapa de red, la cual cuenta con un mapa de Google con las ubicaciones de las estaciones del Tuzo Bus marcadas con la imagen característica de la estación.

En la figura 5.5: Diseño preliminar del Menú: Se muestra la distribución de los elementos dentro del menú y los módulos a los cuales se van a enlazadas las opciones.

En la figura 5.5: Diseño preliminar de Ruta Expres: Cuenta con una lista de las rutas existentes que al seleccionarla se despliega una lista con los nombres de las ubicaciones en los que hace parada la unidad.

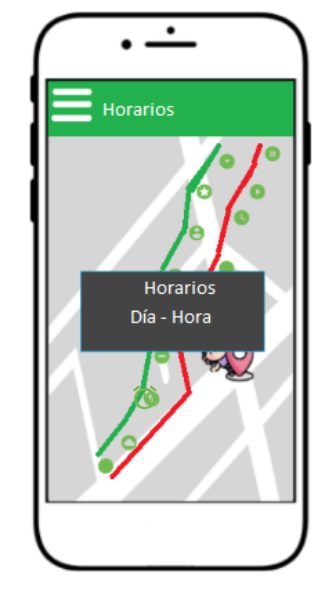
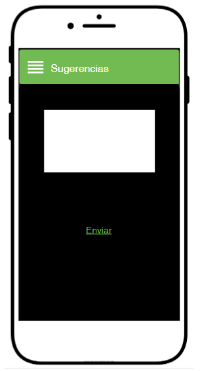


En la figura 5.5: Diseño preliminar de Ruta Alimentadora: Muestra una lista de las rutas alimentadoras, al seleccionar alguna de estas se despliega una lista con los nombres de las paradas, así como la opción de poder ver la ruta en el mapa.



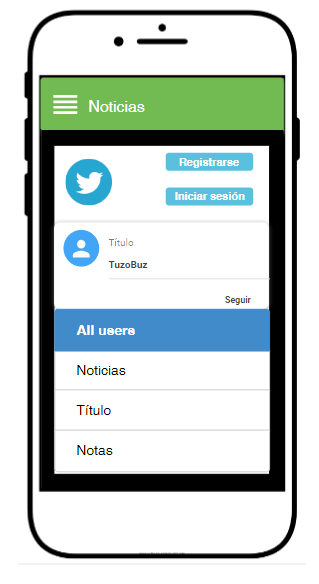
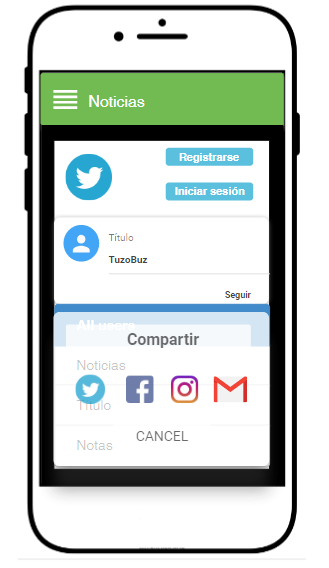
En la figura 5.5: Diseño preliminar de Horarios y Sugerencias: Se muestra un cuadro de alerta con la información de los horarios en que presta servicio el Tuzo Bus.

En la pantalla de sugerencia se muestra un formulario para publicar un texto como sugerencia.



En la figura 5.5 Diseño preliminar de Compartir e Información: Se muestran las opciones para compartir el enlace de descarga.

En el módulo de Información, se muestra el diseño de la aplicación conectándose a la cuenta de Twitter oficial del Tuzo Bus



En la figura 5.5 Diseño preliminar de Configuración: Se muestran las opciones en las que el usuario podrá configurar el tipo de letra, colores tipo de mapa, así como las sugerencias GPS.

## 5.4. Diseño de la base de datos

Para poder diseñar la base de datos fue necesario conocer la información completa del proyecto para su almacenamiento.

### Tabla de la base de Datos

En las siguientes tablas se muestra las entidades creadas para el diseño de la base de datos.

En la *tabla 5.9: Entidad de usuarios,* muestra información general de los usuarios en la base de datos conformada de cinco columnas. (id\_usuario, correo, sexo, edad, fecha\_registro).

Tabla 5.9: Entidad de usuarios.

|  |  |  |  |
| --- | --- | --- | --- |
| Campo | Tipo | Tamaño | Descripción |
| id\_usuario | Int | 11 | Identificador de usuario |
| correo | Varchar | 30 | Correo electrónico |
| sexo | Int | 1 | Genero usuario |
| edad | Int | 1 | Rango de edad usuario |
| fecha\_registro | Varchar | 50 | Fecha en timestamp unix |

En la *tabla 5.10: Entidad de* ***sugerencia***, muestra información general de las sugerencias registradas en la tabla sugerencia, conformada de cinco columnas (id\_sugerencia, id\_usuario, correo, mensaje, fecha).

Tabla 5.10: Entidad de sugerencia.

|  |  |  |  |
| --- | --- | --- | --- |
| Campo | Tipo | Tamaño | Descripción |
| id\_sugerencia | Int | 11 | Identificador de sugerencia |
| id\_usuario | Int | 11 | Identificador de usuarios |
| correo | Varchar | 50 | Correo de usuarios |
| mensaje | Varchar | 500 | Mensaje de Sugerencia |
| fecha | Varchar | 50 | Fecha en timestamp unix |

En la *Tabla 5.11*: *Entidad* ***ubicación,*** muestra información general de las ubicaciones almacenadas en la tabla ubicaciones conformada de cinco columnas (id\_ubicacion, id\_usuario, latitud, longitud, fecha\_conexion).

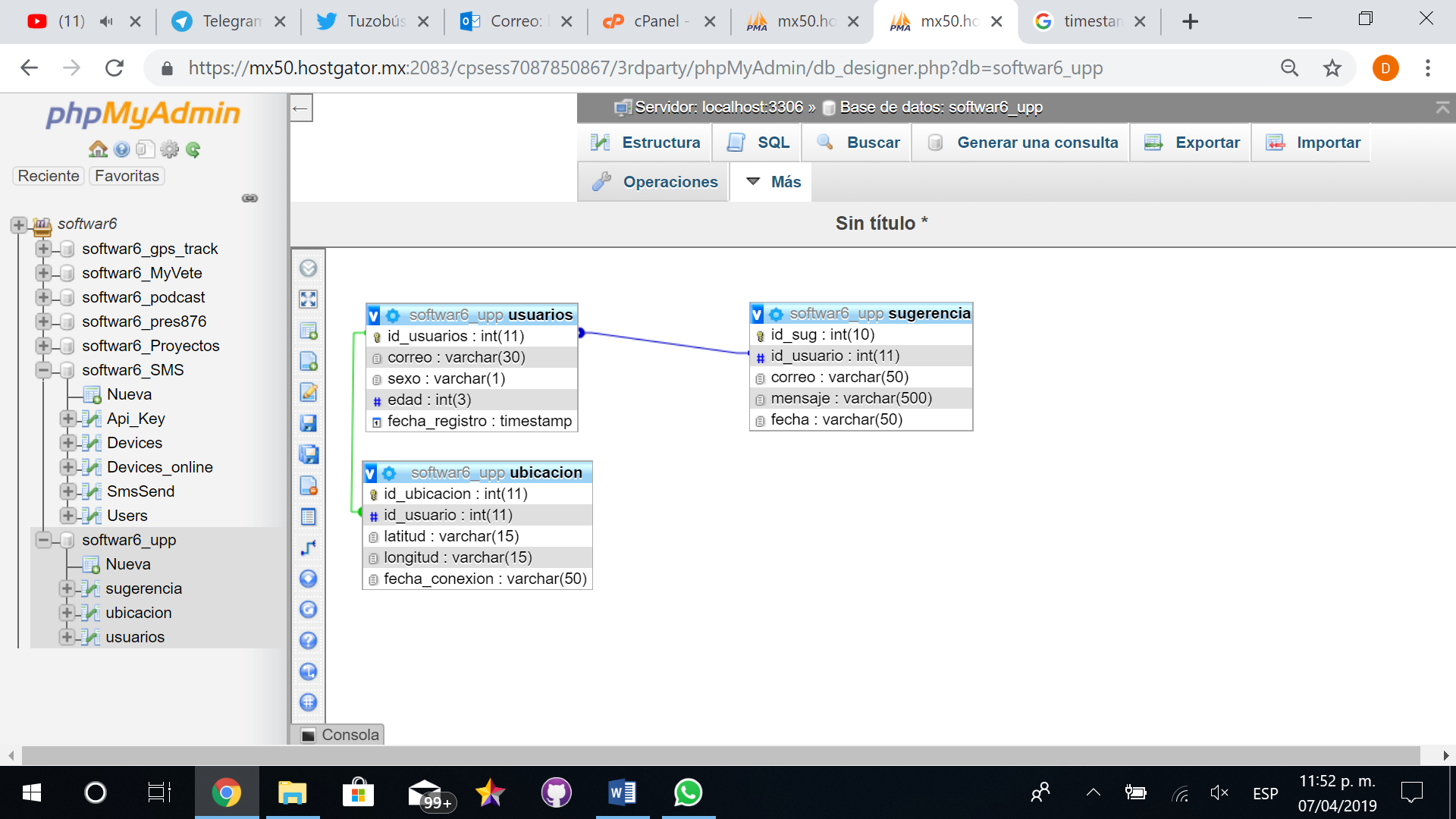
Tabla 5.11: Entidad de nee\_negocios.

|  |  |  |  |
| --- | --- | --- | --- |
| Campo | Tipo | Tamaño | Descripción |
| Id\_ ubicacion | Int | 11 | Identificador de ubicación |
| Id\_ usuario | Int | 11 | Identificador de usuario |
| latitud | Varchar | 15 | Latitud |
| longitud | Varchar | 15 | Longitud |
| fecha\_conexion | Varchar | 50 | Fecha en timestamp unix |

## Diagrama Relacional

En la Figura 5.5: Diagrama Racional de la aplicación Tuzo Bus, muestra la base de datos que será utilizada por la aplicación, la cual se elaboró con la herramienta grafica para la base de datos phpMyAdmin.

En el siguiente diagrama se muestra las diferentes tablas de la base de datos, donde se puede observar la relación que existe entre ellas donde el registro del usuario se relaciona con los registros de sugerencias, así como los registros de ubicación que se envía al momento de iniciar la aplicación Tuzo Bus.



|  |
| --- |
| Figura 5.10: Diagrama Racional de la Aplicación Tuzo Bus. |

## 5.6. **Interfaces graficas de usuario Aplicación NEEDS.**

A continuación, se explican las ventanas de la aplicación Tuzo Bus, como son: Bienvenida, Registro de usuario, Mapa de red, Ruta exprés, Ruta alimentadora, Horarios, Información, Compartir, Configuración

***SPLASH***

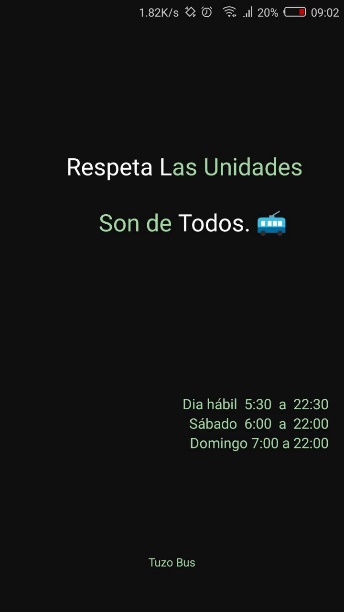
***Splash:*** *En la figura 5.11: Splash (Mensaje de bienvenida)* muestra un mensaje aleatorio con recomendaciones sobre el servicio del Tuzo Bus, así como los horarios**.**

Figura 5.11: Splash (Mensaje de bienvenida)

***REGISTRO DE USUARIO***

***Registro de usuario:*** En la **figura** 5.12: Registro de usuario, muestra las opciones que el usuario tiene que seleccionar con sus datos personales poder ser registrado en la base de datos para su almacenamiento.

|  |  |
| --- | --- |
| *C:\Users\r0jo\Pictures\tuzobos\4241274b-ac27-4e0b-8473-5fb1dbc6f37a.jpg* | *C:\Users\r0jo\Pictures\tuzobos\1d637bfc-0040-4690-b766-3c3a9d2a8199.jpg* |
|  |  |

Figura 5.12: Registro de usuario.

***MAPA DE RED***

En esta figura 5.12: Mapa de red, muestra la ruta de las unidades exprés del Tuzo bus, así como las ubicaciones de las estaciones donde hace parada, mostrando la estación más cercana a partir de la posición actual del usuario, diferenciándola con un circulo azul.

******

Figura 5.12: Registro de usuario.

***MENU***

En esta figura 5.12: Menú, muestra los módulos a los que puede acceder el usuario tales como: mapa de red, Ruta expres, Ruta alimentadora, horarios, sugerencias, información, compartir App, Configuración.

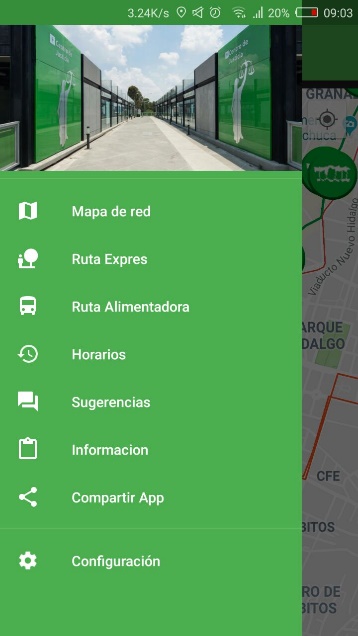
******

Figura 5.12: Registro de usuario

***RUTA EXPRES***

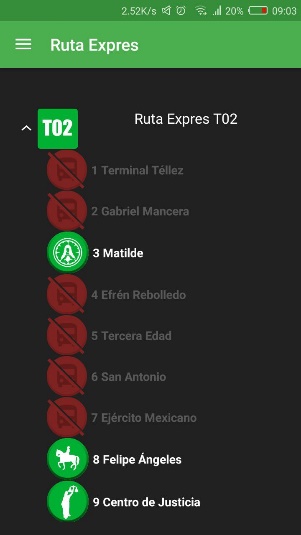
En esta figura 5.13: Ruta expres, muestra una lista de las diferentes rutas que utiliza el Tuzo Bus, al seleccionar una de ellas se despliega un listado con los detalles, mostrando los nombres de las estaciones donde hace parada en verde, y los nombres de las estaciones donde no hace parada en rojo.

Figura 5.12: Registro de usuario

***RUTA ALIMENTADORA***

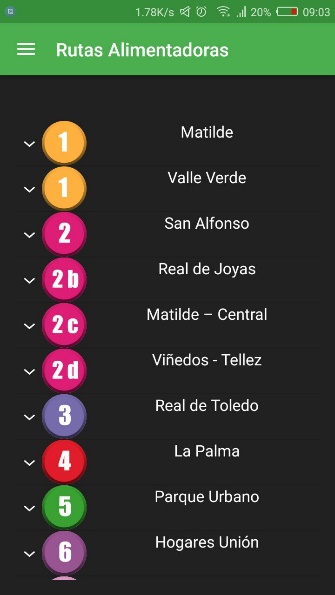
******En esta figura 5.14: Ruta alimentadora, muestra una lista de los nombres de las rutas alimentadoras, al seleccionar una, despliega un listado de los nombres de las bases donde hace parada, con la opción de poder visualizar en el mapa la ruta.

Figura 5.12: Registro de usuario

***HORARIOS***

En esta figura 5.14: Horarios, se muestra un cuadro de alerta con los horarios que presta servicio las unidades de transporte publico Tuzo Bus.

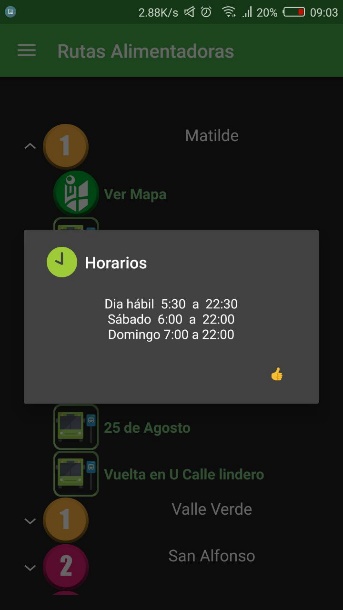


Figura 5.12: Registro de usuario

***SUGERENCIAS***

En esta figura 5.14: Sugerencias, se muestra un formulario para escribir una sugerencia y al servicio web para ser almacenada la información de quien escribió la sugerencia.

Figura 5.12: Sugerencias

***INFORMACION***

En esta figura 5.14: información, se muestra la cuenta de Twitter oficial Tuzo bus, para que el usuario pueda consultar las ultimas noticias con respecto al servicio publicadas de manera inmediata por los administradores del servicio.



Figura 5.12: información.

***COMPARTIR APLICACION***

En esta figura 5.14: Compartir aplicación, se muestra las aplicaciones en donde se puede compartir el texto que contiene el link de descarga, la cual va dirigida a los contactos del usuario.

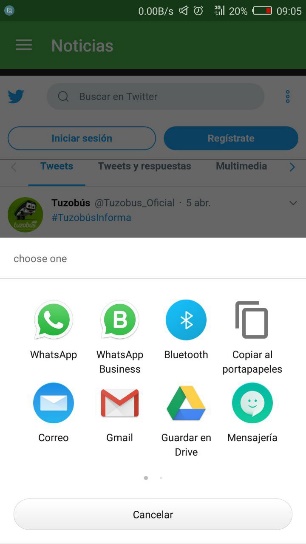


Figura 5.12: Compartir aplicación.

CONFIGURACION

En esta figura 5.14: Configuración, se muestran las opciones que se pueden configurar dentro de la aplicación, tales como: tres tamaños de letra, Mostrar rutas de unidades alimentadoras, recomendación GPS, temas de la aplicación, tipos de mapas.



## 5.7 Codificación

En la Aplicación Tuzo Bus se siguieron las especificaciones del ISO/12207 para comprobar su funcionalidad, se realiza la codificación de las clases que la componen, a continuación, se muestra la codificación.

Clase ***Registro.java***

Método público ***“Installation”,*** el cual se utiliza para enviar parámetros con los datos de registro mediante el método POST a una página PHP, la cual se encarga de hacer el Query (consulta), para almacenar la información en la base de datos y devolver el Identificador de la Base de datos.

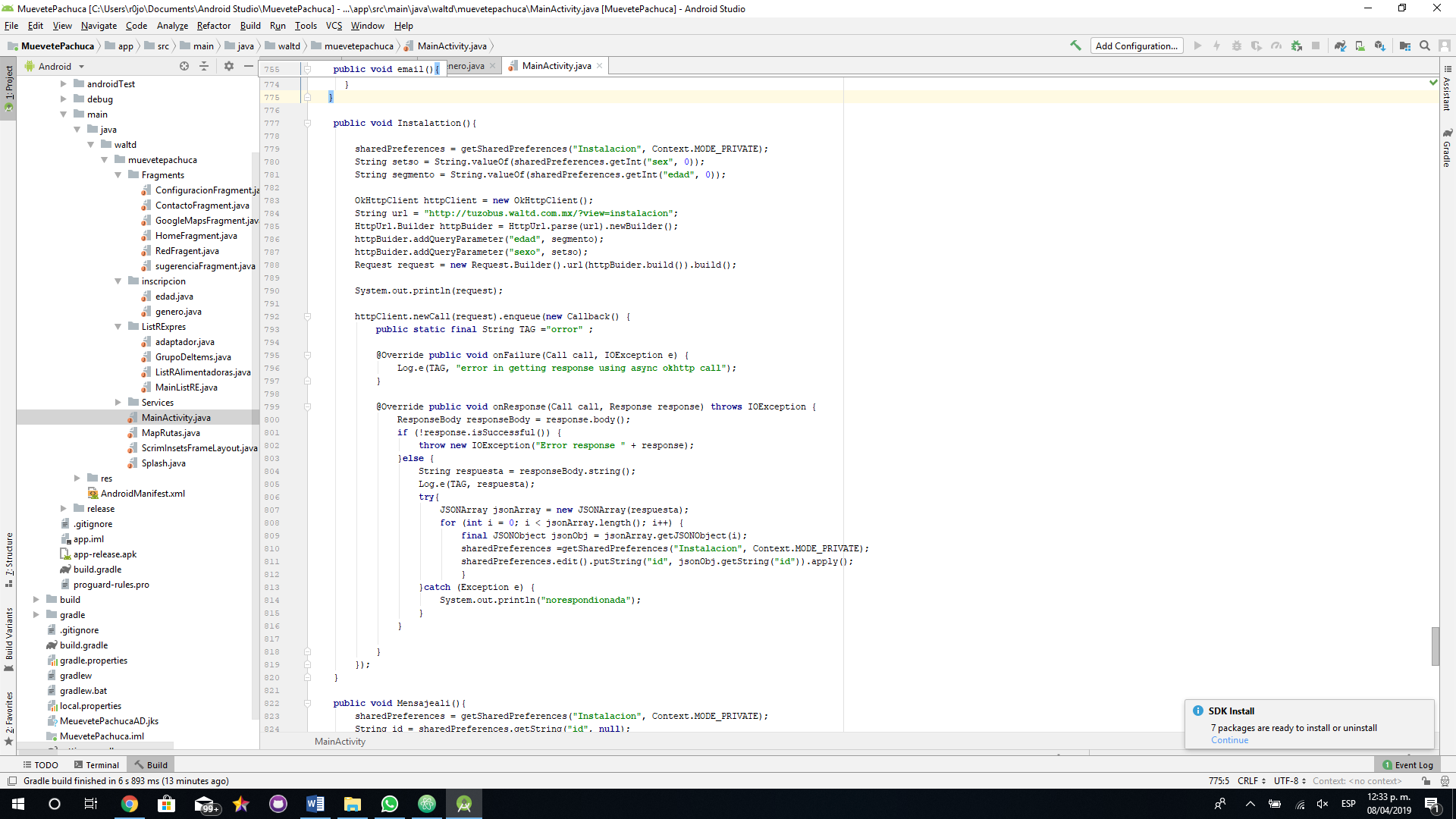
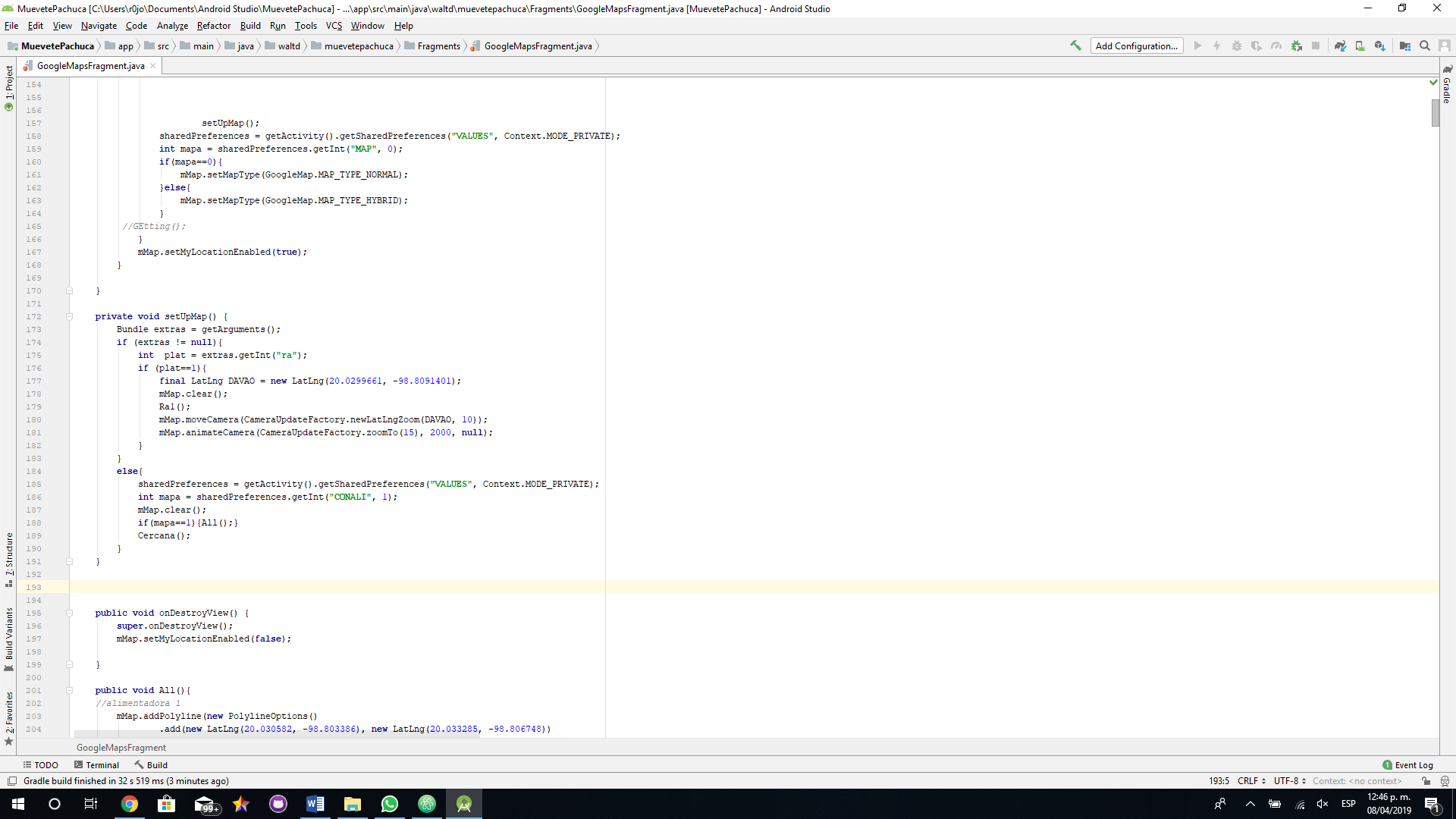


Figura 5.22: Clase Registro.java

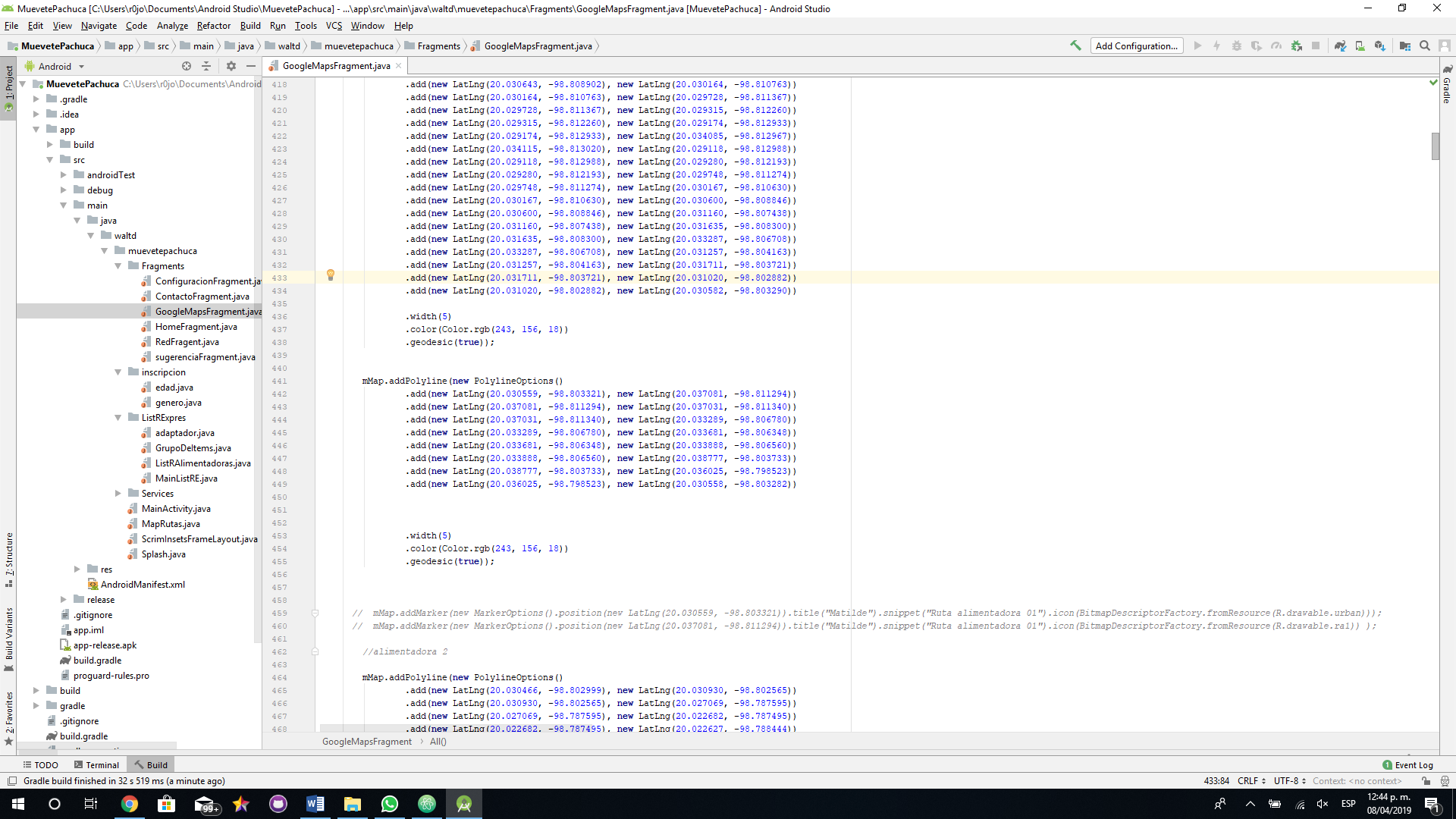
Clase ***MapaRed.java***

Método público ***“setMAp”,*** el cual se encarga de configurar el mapa, estableciendo la posición donde se va a centrar el mapa, estableciendo el tipo de mapa que se ha establecido en configuraciones.



Clase ***Polyline.Java***

La clase Polyline, se encarga de realizar trazos de líneas, en el mapa, con la ayuda de las coordenadas GPS para el trazo de rutas del Tuzo Bus, configurando el grosor de la línea, así como el color.



Clase ***DireccionJSONParser.java***

La clase DireccionJSONParser se encarga de realizar el trazo de ruta desde la ubicacion actual hasta la estacion del tuzo bus mas cercana, asi como de informar el tiempo estimado de llegada a la estacion.

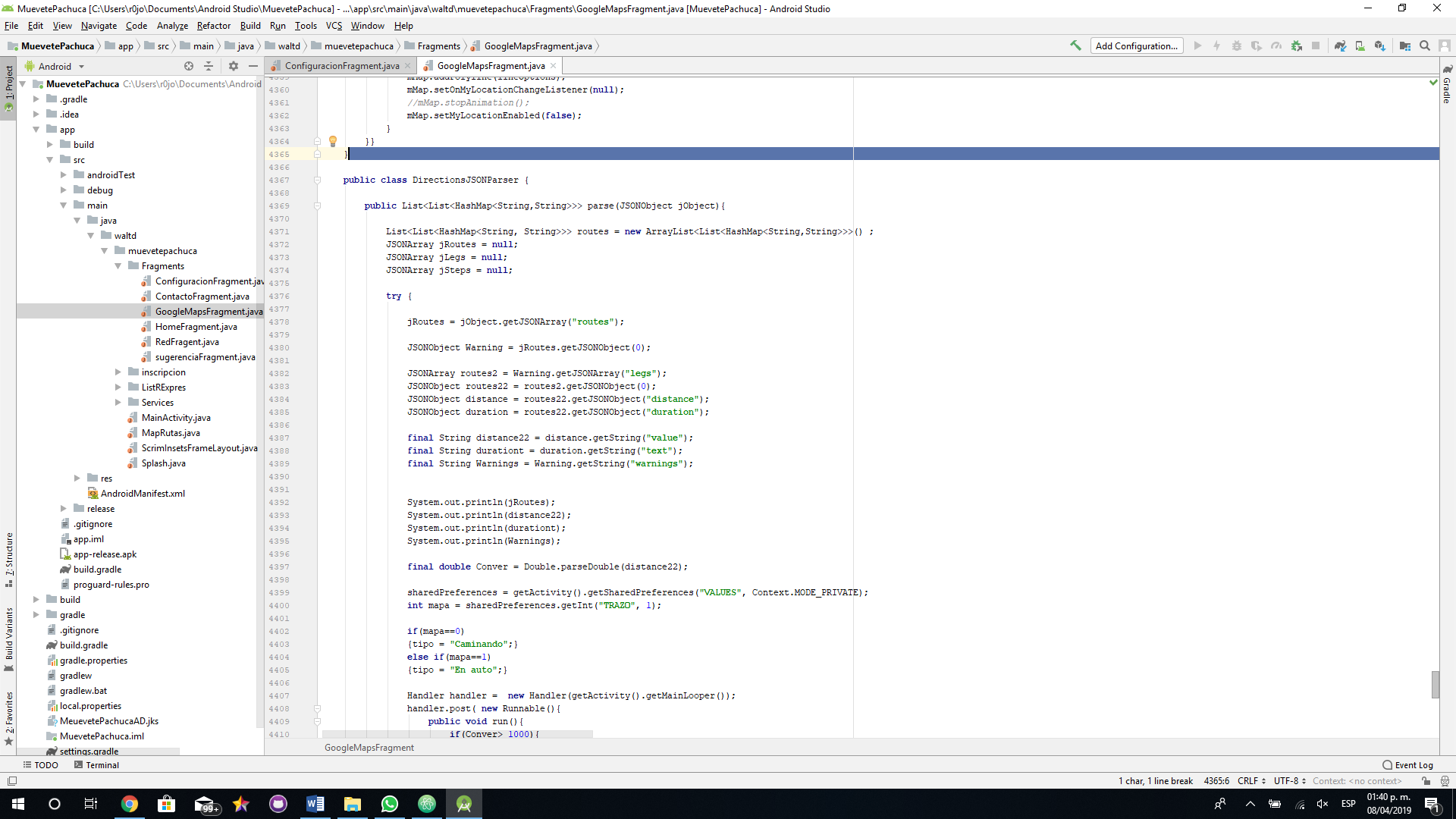
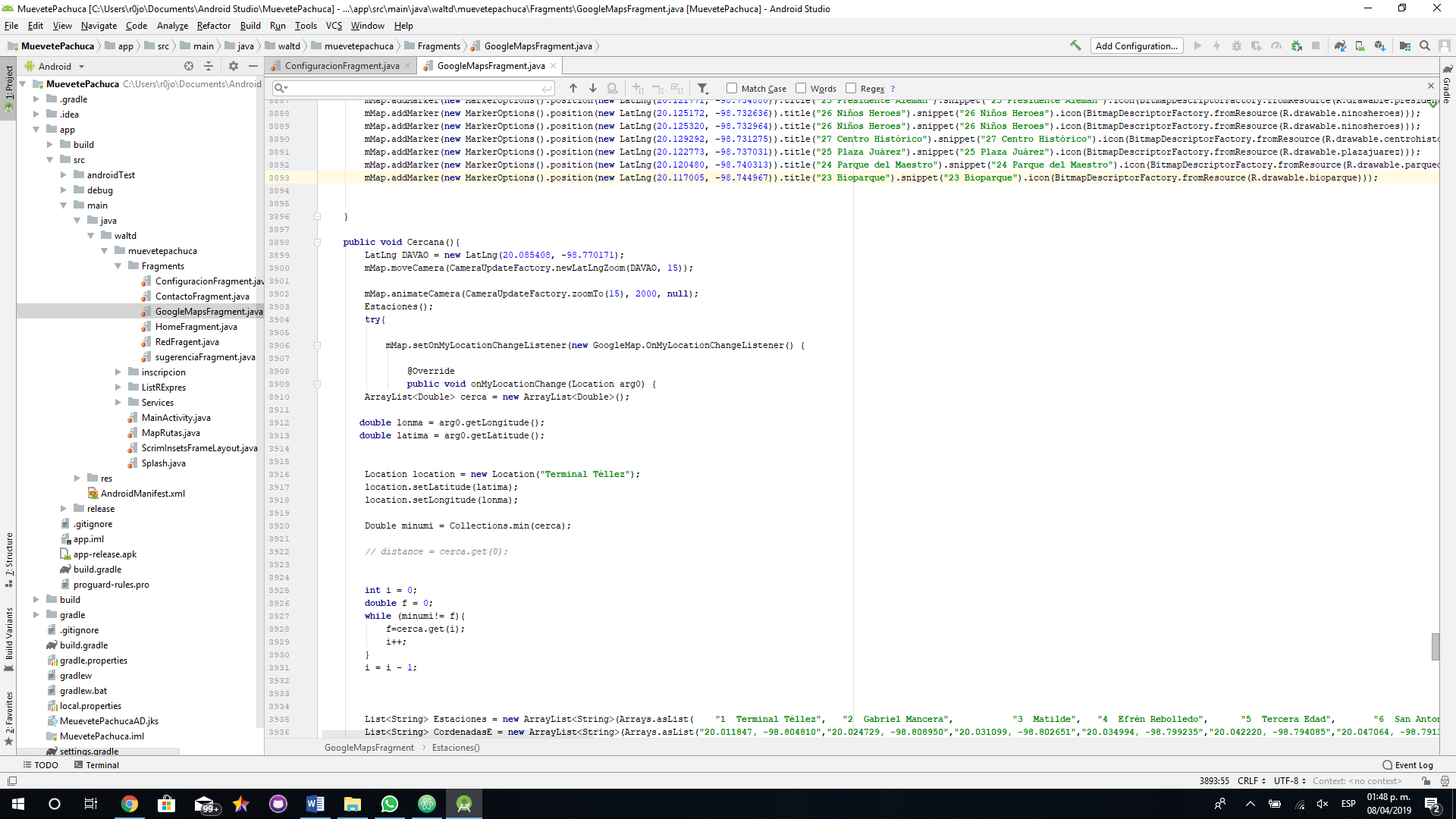


Figura 5.23: Intereses.java

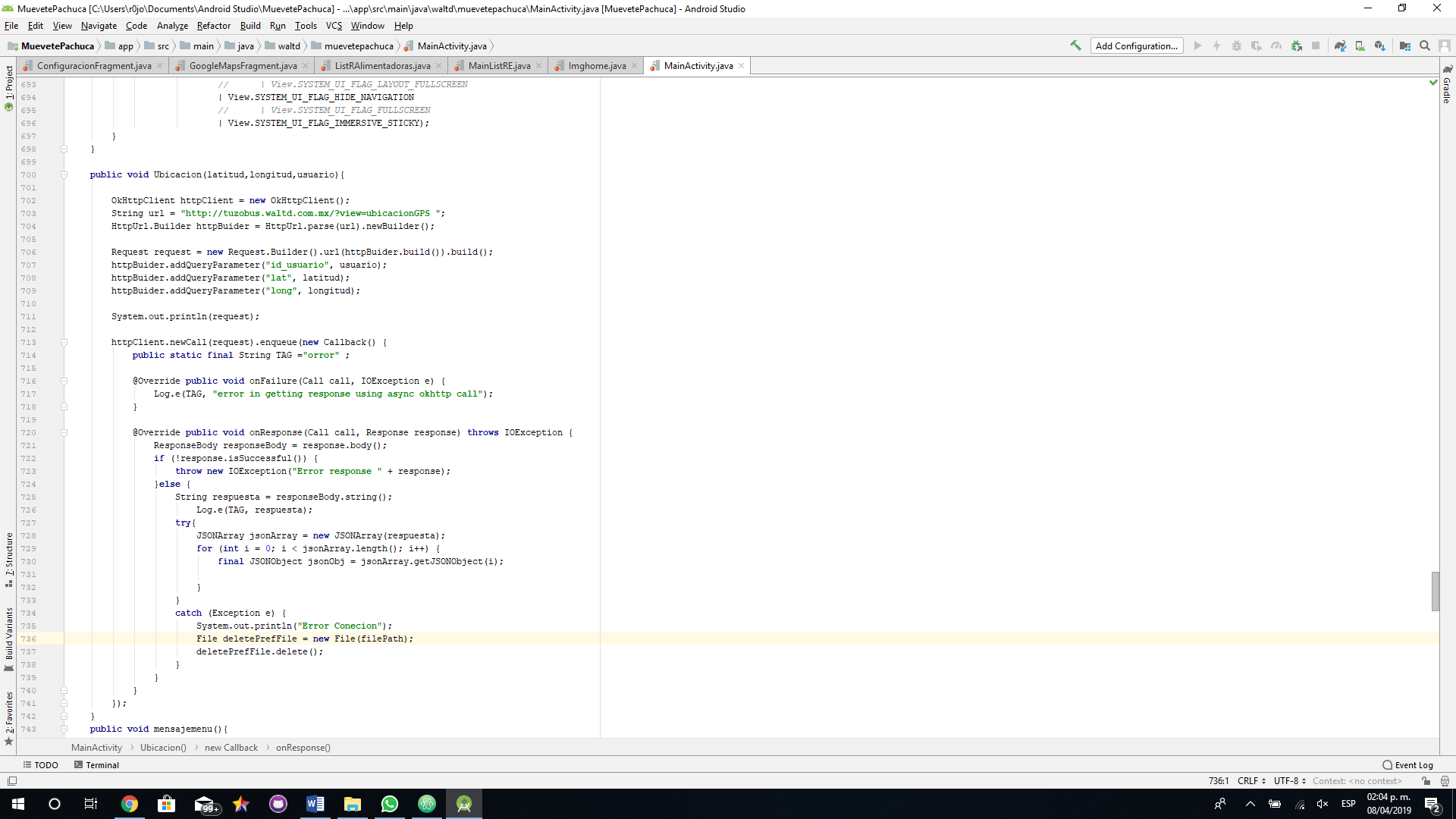
Clase ***Estaciones.java***

Método público “Cercana”, se encarga de realizar las operaciones de distancia de punto a punto a partir de un arreglo multidimensional, el cual contiene los nombres de las estaciones, así como su ubicación GPS, regresando como resultado la posición en el arreglo de la ubicación con la distancia más.



Clase MainActivity.java

Método público “Ubicacion”, se encarga de ejecutar un servicio que se inicia cuando se abre la aplicación, la cual obtiene la ubicación GPS (latitud, longitud) enviando la información a un servicio web para tener el registro de las ubicaciones de donde se abre la aplicación.



## 5.8 Pruebas

El objetivo de realizar las pruebas es verificar que funcione correctamente la aplicación, De esta manera se asegura y se garantiza que el cliente no tendrá ningún problema al utilizarla.

Para el desarrollo de este proyecto, se realizó la prueba de cada módulo, al finalizar las pruebas, se puso a disposición del líder del proyecto las versiones que se desarrollaron, para obtener la aprobación del módulo funcionando correctamente, para asegurar que cumpliera las funciones, de esta manera se cumplió en tiempo y forma con la planeación establecida por el líder del proyecto.

# **CAPÍTULO 6**

# Conclusiones y trabajos futuros

Una de las etapas más difíciles del desarrollo del proyecto, fue aprender a gestionar un desarrollo de software manejando las herramientas que se utilizaron para programar la aplicación, así como la configuración de servidores, pues en las practicas solo se habían trabajado con servidores locales

También me enfrenté a la problemática de los servicios web, donde aprendí los métodos de envió seguro, utilizando el método POST y librerías tales como OkHttp que facilitaban la implementación de estas tecnologías, donde los servicios web recibían la información y hacia un Query (consulta) para la comunicación a base de datos.

El tiempo fue limitado para la entrega de tareas de cada módulo realizado para la aplicación, tratando de respetar los tiempos y la hora de entrega.

Para el proyecto, se utilizó la metodología Ágil Scrum, la cual facilito manejar de manera adecuada los requerimientos, adecuándose a los cambios de cualquier fase del proyecto y a las necesidades del cliente.

Como tarea personal tuve que repasar algunos conceptos de programación orientada a objetos y el Lenguaje Unificado de Modelado (UML), ya que fueron herramientas indispensables para el desarrollo de la aplicación.

Trabajos futuros:

Como trabajo a futuro se pretende agregar el modulo para poder revisar el saldo de las tarjetas del Tuzo Bus, ya que estas tienen la capacidad de lectura y escritura con tecnología NFC (***Near field communication)*** de la cual algunos dispositivos tienen antena para lectura y escritura,

Aumentar las rutas de servicios de transporte universitarios, para aumentar la población de los usuarios d la aplicación, de igual manera el desarrollo de la aplicación en plataforma IOS con el propósito de adquirir conocimientos para el desarrollo de nuevas plataformas, así como agregar espacios publicitarios para la monetización de la aplicación.

# **GLOSARIO**

**API:** La interfaz de programación de aplicaciones, (Application Programming Interface), es el conjunto de subrutinas, funciones y procedimientos (o métodos, en la programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

**CASE:** Ingeniería de Software Asistida por Ordenador (Computer Aided Software Engineering), son aplicaciones informáticas destinadas a la productividad en el desarrollo del software, reduciendo costo en términos de dinero y tiempo.

**HOSTING**: es el servicio que provee a los [usuarios](https://es.wikipedia.org/wiki/Usuario_(inform%C3%A1tica)) de [Internet](https://es.wikipedia.org/wiki/Internet) un sistema para poder almacenar información, imágenes, vídeo, o cualquier contenido accesible vía web.

**HTML:** lenguaje de marcas de hipertexto (hiperText Markup Language), es un estándar que sirve de referencia para la elaboración de páginas web.

**MIME:** extensiones multipropósito de correo de internet (multipurpose Internet Mail Extensions,) son una serie de conversación o especificaciones dirigidas al intercambio a través de internet de todo tipo de archivo d forma trasparente para el usuario.

**UML:** lenguaje unificado de modelado (unified modeling language) lenguaje grafico para visualizar, especificar construir y documentar un sistema.

**JSON:** es un formato de intercambio de datos ligero. (JavaScript Object Notation) es fácil para los seres humanos a leer y escribir. Es fácil para las máquinas para analizar y generar.

**POST** es un [método de solicitud](https://en.wikipedia.org/wiki/HTTP#Request_methods) soportado por [HTTP](https://en.wikipedia.org/wiki/HTTP" \o "HTTP)utilizado por la [World Wide Web](https://en.wikipedia.org/wiki/World_Wide_Web" \o "Red mundial) . Por diseño, el método de solicitud POST solicita que un [servidor web](https://en.wikipedia.org/wiki/Web_server) acepte los datos incluidos en el cuerpo del mensaje de solicitud, lo más probable es que se almacene.

**PHPMYADMIN:** es una herramienta escrita en PHP con la intención de manejar la administración de MySQL a través de páginas web, utilizando un navegador web

**QUERY**: interacción con una [base de datos](https://es.wikipedia.org/wiki/Base_de_datos).

**REST:** transferencia de estado representacional (respresenttational state tranfer) es una técnica de arquitectura de software para sistemas hipermedia distribuidos como internet.

**SQL:** Lenguaje de consultas estructurado (Scrutured Query Language) es un lenguaje declarativo de acceso a base de datos relacionales que permite diversos tipos de operaciones en ellas).

**SCRUM**: nombre que proviene de cierta jugada que tiene lugar durante un partido de rugby, en donde se forma de jugadores alrededor del balón y todos trabajan juntos para moverlo a través del campo.

**SERVICIO WEB:** (web service o web services) es una tecnología que utiliza un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones.

**SPRINT:** periodo que comprende de una y cuatro semanas en proceso SCRUM

**URI:** identificador de recursos uniforme es una cadena de caracteres que indica los recursos de una red de forma univoca.

**XML:** lenguaje de marcas extensible es un lenguaje de marcas de desarrollo por el World Wide Web Consirtium (W2C) utilizado para almacenar datos de forma legible.

**SOAP:** (simple object access protocol), es un protocolo estándar que define como dos objetos en diferentes procesos puedes comunicarse por medio de intercambio de datos XML.

# **BIBLIOGRAFIA**

Allende, J. S. (2001). *Java 2. .* Madrid: McGRAW-HILL.

Bennett, S. (2007). *Anàlisis y diseño orientado a objetos de sistemas usando UML.* Madrid: McGrawHill.

Braude, E. J. ( 2003). *Ingenierìa de Software. .* Mèxico , D.F.:: alfaomega.

Castellano, I. L. (2013). *Bases de Datos Primera Ediciòn.* Mèxico: Alfaomega.

Cornelio, E. R. (2002 ). *Introducciòn al SQL para Usuarios y Programadores.* Madrid España: THOMSON.

Darwin, I. F. (2012). *Android Coobook.* Sebastopol: O'REILLY.

Gironés, J. T. (2013). *El gran libro de Android Tercera Edición.* México: Alfaomega.

JSON, I. (s.f.). Obtenido de www.json.org

Kniberg, H. (2007). *Scrum.* Estados Unidos de America.

Marsicano, K., Hardy, B., Stewart, C., & Phillips, B. (2015). *Programación con android.* México: ANAYA MULTIMEDIA .

Plesa, D. (2007). *Scrum.* Estados Unidos de América : Henrik Kniberg.

Riordan, R. M. (2000). *Diseño de bases de datos relacionales con Access y SQL Server.* Madrid: McGRAW-HILL.

Senn, J. A. (1992). *ANÀLISIS y DISEÑO de SISTEMAS de INFORMACIÒN.* Edo. de Mèxico: McGRAW-HILL.

Silberschatz, A. ( 2006). *Fundamentos de bases de datos Quinta ediciòn.* Madrid : McGRAW-HILL.

Sommerville, I. (2005). *Ingenierìa del Software. Sèptima ediciòn.* Madrid: Pearson Educaciòn.

Soriano, E. A. (2012). *Adroid Progración de Dispositivos Móviles a través de Ejemplos .* México: Alfaomega.

*Trello*. (s.f.). Obtenido de https://trello.com/

Wiley, J. (2013). *Android Application Development Cookbook.* Indeanapolis: Wei Meng Lee.

Zhou, Z. (2011). *WINDOWS PHONE 7 PROGRAMMING.* WILEY.