APLICATIVO MÓVIL PARA LA ADMINISTRACIÓN DE SOLICITUDES DE TAXI.

ANDRÉS RODRÍGUEZ ESCUDERO.

UNIVERSIDAD CATÓLICA DE PEREIRA.
FACULTAD DE CIENCIAS BÁSICAS E INGENIERÍA.
PROGRAMA DE INGENIERÍA DE SISTEMAS Y TELECOMUNICACIONES.
PRÁCTICAS ACADÉMICAS.
PEREIRA.
2012.
APLICATIVO MÓVIL PARA LA ADMINISTRACIÓN DE SOLICITUDES DE TAXI.

ANDRÉS RODRÍGUEZ ESCUDERO.

TUTOR:
ING. RICARDO ALONSO HURTADO.

UNIVERSIDAD CATÓLICA DE PEREIRA.
FACULTAD DE CIENCIAS BÁSICAS E INGENIERÍA.
PROGRAMA DE INGENIERÍA DE SISTEMAS Y TELECOMUNICACIONES.
PRÁCTICAS ACADémICAS.
PEREIRA.
2012.
<table>
<thead>
<tr>
<th>TOPIC</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>INTRODUCCIÓN</td>
<td>6</td>
</tr>
<tr>
<td>PRESENTACIÓN DE LA ORGANIZACIÓN O SITIO DE PRÁCTICA</td>
<td>7</td>
</tr>
<tr>
<td>ETAPA JUNIOR</td>
<td>8</td>
</tr>
<tr>
<td>ETAPA ADVANCED</td>
<td>8</td>
</tr>
<tr>
<td>ETAPA SENIOR</td>
<td>8</td>
</tr>
<tr>
<td>SOMVI</td>
<td>9</td>
</tr>
<tr>
<td>LAS SOLUCIONES MÓVILES SON IMPACTADAS DESDE</td>
<td>9</td>
</tr>
<tr>
<td>DEFINICIÓN DE LAS LÍNEAS DE INTERVENCIÓN</td>
<td>10</td>
</tr>
<tr>
<td>DESCRIPCIÓN DEL PROBLEMA</td>
<td>11</td>
</tr>
<tr>
<td>JUSTIFICACIÓN</td>
<td>12</td>
</tr>
<tr>
<td>OBJETIVOS</td>
<td>13</td>
</tr>
<tr>
<td>OBJETIVO GENERAL</td>
<td>13</td>
</tr>
<tr>
<td>OBJETIVOS ESPECÍFICOS</td>
<td>13</td>
</tr>
<tr>
<td>CRONOGRAMA DE ACTIVIDADES</td>
<td>14</td>
</tr>
<tr>
<td>MARO TEÓRICO</td>
<td>16</td>
</tr>
<tr>
<td>¿QUÉ ES UN TELÉFONO MÓVIL?</td>
<td>18</td>
</tr>
<tr>
<td>“SMARTPHONE” O TELÉFONOS INTELIGENTES</td>
<td>18</td>
</tr>
<tr>
<td>COMPONENTES DE UN SISTEMA OPERATIVO</td>
<td>19</td>
</tr>
<tr>
<td>Procesos</td>
<td>19</td>
</tr>
<tr>
<td>Administración de memoria</td>
<td>19</td>
</tr>
<tr>
<td>Entradas y salidas</td>
<td>20</td>
</tr>
<tr>
<td>Archivos</td>
<td>20</td>
</tr>
<tr>
<td>SISTEMAS OPERATIVOS PARA DISPOSITIVOS MÓVILES</td>
<td>20</td>
</tr>
<tr>
<td>Symbian O.S</td>
<td>20</td>
</tr>
<tr>
<td>BlackBerry OS</td>
<td>21</td>
</tr>
<tr>
<td>IPhone OS</td>
<td>21</td>
</tr>
<tr>
<td>Windows Mobile</td>
<td>21</td>
</tr>
<tr>
<td>Android</td>
<td>22</td>
</tr>
<tr>
<td>INGENIERÍA DE SOFTWARE</td>
<td>22</td>
</tr>
<tr>
<td>INGENIERÍA DE REQUISITOS</td>
<td>23</td>
</tr>
</tbody>
</table>
7.7 MODELADO DE ANÁLISIS. ........................................................................................................ 23
7.8 ARQUITECTURA DE CLASES .................................................................................................. 25
  7.8.1 Clases con estereotipos .................................................................................................... 26
7.9 INGENIERÍA DEL DISEÑO ..................................................................................................... 27
8 PRESENTACIÓN Y ANÁLISIS DE LOS Resultados ................................................................. 28
  8.1 MODELO DE REQUISITOS. .................................................................................................. 28
    8.1.1 Descripción del problema ............................................................................................... 28
    8.1.2 Modelo de casos de uso ............................................................................................... 29
    8.1.3 Modelo de interfaces .................................................................................................... 30
  8.2 MODELO DE ANALISIS ......................................................................................................... 32
    8.2.1 Caso de uso No. 2 Registrar pasajero ............................................................................ 33
  8.3 MODELO DEL DISEÑO ............................................................................................................. 34
    8.3.1 Descripción de los casos de uso .................................................................................... 34
    8.3.2 Comunicación entre las partes ....................................................................................... 36
    8.3.3 Definición de la estructura de almacenamiento de datos ............................................. 38
    8.3.4 Definición de la plataforma tecnológica ....................................................................... 39
9 CONCLUSIONES. ....................................................................................................................... 40
10 RECOMENDACIONES. ............................................................................................................... 41
### SÍNTESIS

<table>
<thead>
<tr>
<th>Síntesis</th>
<th>Abstract</th>
</tr>
</thead>
<tbody>
<tr>
<td>En este trabajo se refleja todo el proceso de ingeniería del software sobre el proyecto que corresponde a una aplicación para dispositivos móviles que gestiona servicios de taxi, Donde los presuntos pasajero pueden hacer solicitud de servicios, y los taxistas pueden acudir a estos servicios.</td>
<td>This paper reflects the entire software engineering process on the draft corresponding to an application for mobile devices that manages taxi services, where the alleged passenger can request services, and taxi drivers can attend these services.</td>
</tr>
</tbody>
</table>

**Descriptores:** Android, Ingeniería del software, Móvil, Smartphone, Taxistas, Twitteros.

**Descriptors:** Android, Software Engineering, Mobile, Smartphone, Taxis drivers, Twitter.
1 INTRODUCCIÓN.

Un servicio de transporte dentro de una sociedad es severamente importante, ofrecer un medio de transporte a una comunidad ayuda a que dicha sociedad tenga facilidad de trasladarse rápidamente de un lado a otro. Un taxi es un medio de transporte público que permite desplazamientos rápidos, confortables y directos principalmente en áreas urbanas; el usuario paga una tarifa al conductor a cambio del servicio de transporte prestado. Cada conductor tiene su metodología para gestionar un pasajero. Pero especialmente, existe una comunidad de taxistas que tienen una forma muy particular de gestionar sus propios pasajeros.

La comunidad de taxistas twitteros está compuesta por un conjunto de taxistas de la ciudad de Bogotá que se dedican a promover un servicio de calidad a sus usuarios por medio de una de las redes sociales más populares en el momento. Básicamente hacen un ofrecimiento del servicio de taxi desde Twitter, con el fin de brindar un servicio confiable, amable y respetuoso. No todos los taxistas de la ciudad pueden ser taxistas twitteros, para ser miembro de la comunidad deben pasar por un proceso de selección hecho por profesionales idóneos para garantizar que los conductores ofrecerán un buen servicio. Esta comunidad marca la diferencia con las empresas convencionales, ya que para ellos lo primordial es la seguridad del usuario, el comportamiento de gente de bien por parte del conductor, quien es el que presta sus servicios.

Se pretende construir una aplicación para la comunidad de taxistas twitteros y sus pasajeros, con el fin de notificar a los taxistas sobre una petición de algún pasajero en particular. Generando entre ellos un sistema de comunicación para determinar la confirmación de dicha solicitud.
En el año de 1999 surgió en Cali bajo el liderazgo de Orlando Rincón una iniciativa cuyo objeto era la creación de un espacio para jóvenes emprendedores de la industria del Software. Orlando Rincón, un reconocido líder de la industria, había fundado en 1984 Open Systems Ltda., una de las empresas más representativas de la industria de software colombiana. Durante todos estos años acumuló experiencias y conocimiento acerca de cómo consolidar una empresa de software.

En 1997 visitó dos países transformados en líderes globales de esta industria y con condiciones similares a Colombia: Irlanda y la India. Orlando observó que era viable construir, con muy poca inversión, un Parque Tecnológico de Software y que ésta podría ser una excelente oportunidad para la ciudad de Cali, sumida entonces en una grave crisis económica y de identidad social, debido al funesto impacto del narcotráfico.

Para ello, en junio de 1999 y aprovechando el cambio de sede de Open Systems, Orlando acordó con ésta la donación de la infraestructura avaluada en USD $30.000 para el inicio del Parque Tecnológico de Software de Cali.

En años anteriores, Orlando había desarrollado un proceso de incubación de dos empresas de software: VIANet, dedicada a crear páginas y software WEB y Apedi, empresa a la cual Open había entregado su software de propósito comercial cuando decidió especializarse en software para servicios públicos y telecomunicaciones, brindándoles apoyo económico, coaching y asesoría permanente en tecnología y situaciones de negocios. En 1998 se había incorporado Innova Systems, especializada en el desarrollo de software para gestión documental. Estas empresas se trasladaron en septiembre de 1999 a las instalaciones donadas por Open Systems, en calidad de empresas base, fundadoras de este proyecto. En diciembre de ese mismo año se creó la Fundación Parque Tecnológico del Software con el objetivo de facilitar la creación de empresas de software por parte de emprendedores jóvenes en la ciudad de Cali, en ese entonces con capacidad para residenciar 11 proyectos de emprendimiento con espacio para tres personas por proyecto, así nació ParqueSoft.

Actualmente, ParqueSofta consolidando un corredor de ciencia y tecnología en las ciudades de Cali, Popayán, Pasto, Buga, Tuluá, Palmira, Armenia, Manizales, Pereira, Buenaventura, Ibagué, Villavicencio y Sincelejo.

En la ciudad de Pereira, ParqueSoft inició labores hace cinco (5) años, gracias al empuje de varios emprendedores de la empresa Fastec de Colombia, quienes luego de conocer y valorar el modelo implementado en Cali, fueron vinculados como miembros de la Fundación Parque Tecnológico de Software en Agosto de 2002.

En el 2004, el proyecto fue vinculado al plan de desarrollo de la administración de Juan Manuel Arango, alcalde de Pereira de ese entonces; también fue incorporado en la Política de Desarrollo Regional del programa Ciencia, Tecnología e innovación, bajo la cual se proporcionaron rubros económicos para los primeros tres años de funcionamiento de ParqueSoft. En alianza entre Alcaldía de Pereira y la Universidad Tecnológica de Pereira (UTP) se entregó en comodato a la
UTP el espacio físico en el cual a la fecha, opera la Fundación, fue así como el 15 de Marzo de 2005, se constituyó la Fundación Parque Tecnológico de Pereira –ParqueSoft Pereira- siguiendo los lineamientos filosóficos de ParqueSoft Cali. Hoy ParqueSoft Pereira cuenta con un total de cuarenta y ocho (48) empresas y Noventa y cuatro (94) emprendedores y colaboradores, desarrollando proyectos de base tecnológica e investigación en el área de software. Sigue siendo apoyado por la Alcaldía de Pereira y la Universidad Tecnológica de Pereira, además de Media Commerce, Cámara de Comercio de Pereira, entre otros.

ParqueSoft es una fundación sin ánimo de lucro cuyo propósito es facilitar a jóvenes emprendedores la creación y desarrollo de empresas de base tecnológica que provean al mercado de productos y servicios de tecnología informática. El proceso de crecimiento y desarrollo de la actividad empresarial de los emprendedores está apoyada por 3 etapas, éstas con el fin de representar el nivel de evolución y responsabilidad que va adquiriendo cada empresa según los logros y experiencias adquiridas.

2.1 **ETAPA JUNIOR:**

Tiene duración de 6 meses e inicia al momento del ingreso a la comunidad ParqueSoft. Es la primera etapa y representa el proceso inicial de todo emprendedor en donde se consolida, se da forma a la idea de negocio y se desarrolla el producto o servicio.

2.2 **ETAPA ADVANCED**

Se inicia una vez culminada la etapa Junior y tiene duración de año y medio. Esta etapa representa el nivel de evolución de la actividad empresarial, en la que se tiene un prototipo del producto o servicio y se está preparado para realizar prueba piloto.

2.3 **ETAPA SENIOR**

Se considera la etapa de mayor nivel y tiene duración indefinida, ya que culmina únicamente con la desvinculación del parque. En esta etapa, las empresas están plenamente consolidadas, en donde su producto o servicio ya está probado y en la que ya se cuenta con varios clientes.
2.4 SOMVI.
Una de las 48 empresas vinculadas en ParqueSoft, es Somvi, dedicada a la creación de experiencias de usuario a través aplicaciones móviles en plataformas RIM, Android, iOS y HTML5.
En las aplicaciones móviles podemos generar entretenimiento, información, interactividad, portabilidad y finalmente experiencia de usuario directa.
Trabajamos sobre las tres (3) principales plataformas a nivel mundial RIM (BlackBerry), Android y iOS (iPhone), sin embargo utilizamos HTML5 para proyectos especiales.

2.5 LAS SOLUCIONES MÓVILES SON IMPACTADAS DESDE:

Aplicaciones corporativas, comerciales, de gestión de contenidos y/o entretenimiento, acceso a bases de datos, integración a redes sociales, geolocalización, medios de pagos como NFC, web para móviles, entre otras.

El proceso con nuestros clientes inicia con una etapa de consultoría con el fin de realizar el acompañamiento necesario para la toma de requerimientos inicial que transformará una idea o necesidad a una Aplicación Móvil.
Nuestras soluciones son testeadas y pensadas para generar Apps óptimas, la creación y desarrollo de la interfaz gráfica de nuestros desarrollos son elaboradas profesional y cuidadosamente, teniendo en cuenta el tamaño de las pantallas, el tipo de aplicación requerida, el público objetivo y la(s) plataforma(s) requerida.

Actualmente Somvi cuenta con Siete (7) empleados, trabajando en las diferentes áreas de la organización: área ejecutiva y comercial, área de proyectos, área administrativa y financiera.
3 DEFINICIÓN DE LAS LÍNEAS DE INTERVENCIÓN

El proyecto se enmarca en un sistema móvil que permite generar una comunicación directa desde sus dispositivos Smartphones, entre una comunidad de taxistas llamada Taxistas Twitteros y sus posibles pasajeros, con el fin de que los pasajeros puedan hacer peticiones de servicios de taxi, y los taxistas puedan estar al tanto de cuando hay solicitudes y cuando se requiere de la prestación de un servicio. El sistema también ofrece información necesaria para que los taxistas puedan saber cual es el pasajero solicitante, donde desea el servicio, y para cuando lo desea.
3.1 **DESCRIPCIÓN DEL PROBLEMA:**

En la ciudad de Bogotá existe una comunidad de taxistas que se conocen como Taxistas Twitteros, y tienen una forma de ofrecer servicios de transporte usando una de las redes sociales mas usadas a nivel mundial, Twitter. Básicamente esta comunidad está compuesta por un número de taxistas que poseen cada uno su cuenta en Twitter, donde son visibles para cualquier usuario que desee solicitarlos por un medio diferente al radio-teléfono tradicional. Twitter es una red social que posee demasiados usuarios, y ofrece servicios como medio de comunicación rápido entre sus mismos usuarios, por esta razón es que la comunidad de taxistas Twitteros recurre a usar Twitter, para lograr una comunicación directa con sus presuntos pasajeros con el fin de divulgar que son un conjunto de taxistas dispuestos a ofrecer sus servicios con calidad. Son demasiadas las peticiones que reciben los miembros de esta comunidad, que se torna difícil cumplir con todas ellas y a veces los taxistas se ven expuestos a rechazarlas.

Desafortunadamente, la comunidad taxista Twitteros de Bogotá no posee un sistema que notifique cuando son solicitados los servicios que prestan por Twitter, esto se da porque Twitter no tiene la capacidad de administrar la filtración de los servicios que ofrece esta comunidad, trayendo como efecto que los miembros asociados a la comunidad sean sometidos a permanecer pendientes de Twitter desde sus dispositivos móviles para saber quien requiere de un servicio de taxi, presentando como consecuencia que los taxistas no logran captar quien requiere un servicio de taxi a tiempo.
4 JUSTIFICACIÓN

Desde un tiempo atrás, las redes sociales se han convertido de gran importancia para toda clases de personas, con el uso de ellas se puede encontrar amigos, gestionar información, buscar trabajo, etc. Por estas razones es que la gran mayoría de personas acude a esta clase de tecnología, con el fin de obtener empleo y/o ir en busca de posibilidades de trabajo.

La comunidad de los taxistas Twitteros de Bogotá optaron por abrir campo en ”Twitter”, una de las redes sociales más populares en el momento, con el objetivo de interactuar con posibles pasajeros y ofrecerles sus servicios de taxi con calidad. Desafortunadamente en el uso que ellos le dan a “Twitter” existen ciertas implicaciones que no permiten ofrecer una buena alternativa. Por eso, han querido mejorar su método de comunicación directa con los pasajeros y decidieron inclinarse por un sistema que pudiese atender solicitudes de una forma más rápida y directa.
5 OBJETIVOS

5.1 OBJETIVO GENERAL.

- Construir una aplicación móvil para Android que permite alarma a los miembros de la comunidad taxistas Twitteros cuando se requiera de sus servicios.

5.2 OBJETIVOS ESPECÍFICOS.

- Determinar los requerimientos que el cliente propone sobre la interacción de los taxistas Twitteros con el aplicativo.
- Diseñar el prototipo de la arquitectura del sistema de la aplicación mostrando la forma en que los taxistas interactúan con el aplicativo.
- Desarrollar el ejecutable de la aplicación.
- Realizar las pruebas significativas del aplicativo demostrando el funcionamiento estable.
6 CRONOGRAMA DE ACTIVIDADES.

<table>
<thead>
<tr>
<th>Actividad/Meses</th>
<th>Agosto</th>
<th>Septiembre</th>
<th>Octubre</th>
<th>Noviembre</th>
<th>Diciembre</th>
<th>Enero</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Sem1</td>
<td>Sem2</td>
<td>Sem3</td>
<td>Sem4</td>
<td>Sem1</td>
<td>Sem2</td>
</tr>
<tr>
<td>Introducción a la organización</td>
<td>x</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>formulación y definición del problema</td>
<td></td>
<td>x</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>modelación de análisis</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Modelación del diseño de arquitectura</td>
<td></td>
<td></td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>Elaboración de la aplicación</td>
<td></td>
<td></td>
<td></td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Implementación de pruebas</td>
<td></td>
<td></td>
<td></td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td></td>
<td>Sem1</td>
<td>Sem2</td>
<td>Sem3</td>
<td>Sem4</td>
<td>Sem1</td>
<td>Sem2</td>
</tr>
</tbody>
</table>

X = Actividad presente
<table>
<thead>
<tr>
<th>Asesorías de práctica</th>
<th>X</th>
<th>X</th>
<th>X</th>
<th>X</th>
<th>X</th>
<th>X</th>
<th>X</th>
<th>X</th>
<th>X</th>
<th>X</th>
<th>X</th>
<th>X</th>
</tr>
</thead>
</table>
7 MARO TEÓRICO.

Dada la plataforma la cual está enfocado este proyecto, existen algunas definiciones relevantes que hacen referencia a todo un marco tecnológico que funcionan como soporte teórico. Los dispositivos móviles son una plataforma tecnológica la cual escribe su historia en la humanidad, actualmente existen millones de dispositivos móviles, que funcionan ya sea como teléfono celular, como solo consumidor de internet.

Un dispositivo móvil se puede definir como un aparato de pequeño tamaño, con algunas capacidades de procesamiento, con conexión permanente o intermitente a una red, con memoria limitada, que ha sido diseñado específicamente para una función, pero que puede llevar a cabo otras funciones más generales. De acuerdo con esta definición existen multitud de dispositivos móviles, desde los reproductores de audio portátiles hasta los navegadores GPS, pasando por los teléfonos móviles, los PDAs o los Tablet PCs. En este trabajo nos centraremos fundamentalmente en los teléfonos móviles, por ser los tipos de dispositivos más utilizados y conocidos en la actualidad, los que ofrecen mayor variedad de aplicaciones multimedia y los que más posibilidades de evolución presentan en este sentido. (Alonso, Artime, Rodríguez, & Baniello)

Cada uno de los tipos de dispositivos móviles mencionados anteriormente nace de una historia que cuenta la necesidad que tenía la humanidad sobre lograr comunicarse desde Grandes distancias en tiempos instantáneos.

Desde el principio de las telecomunicaciones dos han sido las opciones principales para llevar a cabo una comunicación: con o sin hilos, por cable o por el aire. En realidad ambas pueden participar en un mismo proceso comunicativo. Por ejemplo la transmisión de un evento deportivo por televisión, en el que una cámara recoge la señal y la transmite, generalmente por cable, a una unidad móvil encargada de comunicarse vía radio con el centro emisor, que a su vez se comunica por cable con una antena emisora que la distribuye por el aire a la zona que cubra la cadena de televisión. De todas formas, en este caso se trata fundamentalmente de una transmisión vía radio, pues es así como se distribuye la señal que previamente ha producido la emisora (captar la señal con la cámara, llevarla al centro emisor y procesarla).

Esta monografía se centrará en las comunicaciones móviles, en las que emisor o receptor están en movimiento. La movilidad de los extremos de la comunicación excluye casi por completo la utilización de cables para alcanzar dichos extremos. Por tanto utiliza básicamente la comunicación vía radio.

Esta se convierte en una de las mayores ventajas de la comunicación vía radio: la movilidad de los extremos de la conexión. Otras bondades de las redes inalámbricas son el ancho de banda que proporcionan, el rápido despliegue que conllevan al no tener que llevar a cabo obra civil.

Sin embargo el cable es más inmune a amenazas externas, como el ruido o las escuchas no autorizadas, y no tiene que competir con otras fuentes por el espacio radioeléctrico. Dos, tres y más cables pueden ser tendidos a lo largo de la misma zanja, y tomando las medidas adecuadas,
no han de producirse interferencias. Imaginar cuatro o cinco antenas apuntando en la misma dirección. Resultado: un más que probable caos. Históricamente la comunicación vía radio se reservaba a transmisiones uno a muchos, con grandes distancias a cubrir. También era útil en situaciones en las que la orografía dificultase en exceso el despliegue de cables. Fundamentalmente se utilizaba para transmitir radio y TV.

Por el contrario, las comunicaciones telefónicas utilizaban cables. Todo esto nos lleva a la actual situación, en la que ya no está tan claro cuando es mejor una u otra opción. En cuanto a las comunicaciones móviles, no aparecen a nivel comercial hasta finales del siglo XX. Los países nórdicos, por su especial orografía y demografía, fueron los primeros en disponer de sistemas de telefonía móvil, eso sí, con un tamaño y unos precios no muy populares. Radio-búsquedas, redes móviles privadas o Trunking, y sistemas de telefonía móvil mejorados fueron el siguiente paso. Después llegó la telefonía móvil digital, las agendas personales, miniordenadores, laptops y un sinfín de dispositivos dispuestos a conectarse vía radio con otros dispositivos o redes. Y finalmente la unión entre comunicaciones móviles e Internet, el verdadero punto de inflexión tanto para uno como para otro. (Uribe, 2006)

Las comunicaciones móviles, consiste en generar una comunicación entre un emisor y un receptor mientras están en movimiento. La movilidad de estos dos factores que se encuentran en los extremos de la comunicación, le da importancia a la poca utilización de hilos (cables) para realizar la comunicación en dichos extremos. Por lo tanto utiliza básicamente la comunicación vía radio. Esta es una gran ventaja de la comunicación vía radio por la movilidad de los extremos de la conexión.

Existe una clasificación de los dispositivos móviles, estos se categorizan según el tipo de consumo de datos. Dado el variado número de niveles de funcionalidad asociado con dispositivos móviles, era necesario hacer una clasificación de los mismos, por ello en el 2005, T38 y DuPont Global Mobility Innovation Team propusieron los siguientes estándares para la definición de dispositivos móviles.

- **Dispositivo Móvil de Datos Limitados (Limited Data Mobile Device):** teléfonos móviles clásicos. Se caracterizan por tener un pantalla pequeña de tipo texto. Ofrecen servicios de datos generalmente limitados a SMS y acceso WAP.

- **Dispositivo Móvil de Datos Básicos (Basic Data Mobile Device):** se caracterizan por tener una pantalla de mediano tamaño, menú o navegación basada en iconos, y ofrecer acceso a emails, lista de direcciones, SMS, y, en algunos casos, un navegador web básico. Un típico ejemplo de este tipo de dispositivos son los teléfonos inteligentes (“Smartphone”).

- **Dispositivo Móvil de Datos Mejorados (Enhanced Data Mobile Device):** se caracterizan por tener pantallas de medianas a grandes (por encima de los 240 x 120 pixels), navegación de tipo stylus, y que ofrecen las mismas características que el "Dispositivo Móvil de Datos Básicos" (Basic Data Mobile Devices) más aplicaciones nativas como
7.1 ¿QUÉ ES UN TELÉFONO MÓVIL?
El teléfono móvil es un dispositivo inalámbrico electrónico basado en la tecnología de ondas de radio, que tiene la misma funcionalidad que cualquier teléfono de línea fija. Su principal característica es su portabilidad, ya que la realización de llamadas no es dependiente de ningún terminal fijo y no requiere ningún tipo de cableado para llevar a cabo la conexión a la red telefónica. Aunque su principal función es la comunicación de voz, como el teléfono convencional, su rápido desarrollo ha incorporado funciones adicionales como mensajería instantánea (SMS), agenda, juegos, cámara fotográfica, agenda, acceso a Internet, reproducción de video e incluso GPS y reproductor mp3. La evolución del teléfono móvil ha permitido disminuir su tamaño y peso, desde el Motorola DynaTAC, el primer teléfono móvil en 1983 que pesaba 780 gramos, a los actuales más compactos y con mayores prestaciones de servicio. Además a lo largo de estos años se ha llevado a cabo el desarrollo de baterías más pequeñas y de mayor duración, pantallas más nítidas y de colores, la incorporación de software más amigable. Inicialmente los teléfonos móviles sólo permitían realizar llamadas de voz y enviar mensajes de texto. Conforme la tecnología fue avanzando se incluyeron nuevas aplicaciones como juegos, alarma, calculadora y acceso WAP (acceso a Internet mediante páginas web especialmente diseñadas para móviles). (Alonso, Artime, Rodríguez, & Baniello).

7.2 “SMARTPHONE” O TELÉFONOS INTELIGENTES.
Un “Smartphone” (teléfono inteligente en español) es un dispositivo electrónico que funciona como un teléfono móvil con características similares a las de un ordenador personal. Es un elemento a medio camino entre un teléfono móvil clásico y una PDA ya que permite hacer llamadas y enviar mensajes de texto como un móvil convencional pero además incluye características cercanas a las de un ordenador personal. Una característica importante de casi todos los teléfonos inteligentes es que permiten la instalación de programas para incrementar el procesamiento de datos y la conectividad. Estas aplicaciones pueden ser desarrolladas por el fabricante del dispositivo, por el operador o por un tercero. Los teléfonos inteligentes se distinguen por muchas características, entre las que destacan las pantallas táctiles, un sistema operativo así como la conectividad a Internet y el acceso al correo electrónico. El completo soporte al correo electrónico parece ser una característica indispensable encontrada en todos los modelos existentes y anunciados en 2007, 2008 y 2009. Otras aplicaciones que suelen estar presentes son las cámaras integradas, la administración de contactos, el software multimedia para reproducción de música y visualización de fotos y video-clips y algunos programas de navegación así como, ocasionalmente, la habilidad de leer...
documentos de negocios en variedad de formatos como PDF y Microsoft Office. Una característica común a la mayoría de “Smartphone” es una lista de contactos capaz de almacenar tantos contactos como la memoria libre permita, en contraste con los teléfonos clásicos que tienen un límite para el número máximo de contactos que pueden ser almacenados. Casi todos los teléfonos inteligentes también permiten al usuario instalar programas adicionales. (Alonso, Artime, Rodríguez, & Baniello)

Cada dispositivo móvil, ya sea Smartphone o de gamas inferiores, tienen un sistema operativo que controla todos los componentes hardware del equipo y permite que el usuario final los use. Un sistema operativo es un programa informático altamente robusto que administra todos los componentes hardware de una computadora, también se caracteriza por ser plataforma de soporte para los programas de aplicación. De alguna manera, un sistema operativo actúa como intermediario entre el usuario y el hardware de la computadora. P. Martínez, M. Cabello, J.C. Díaz Martín. Presentan en su obra “SISTEMAS OPERATIVOS Teoría y práctica” diversas definiciones que describen el término de Sistema Operativo. (Martínez, Cabello, & Díaz Marín, 1997)

7.3 COMPONENTES DE UN SISTEMA OPERATIVO.

Tanenbaum define en su obra “sistemas operativos modernos” los mecanismos que componen un sistema operativo como: procesos, archivos, memoria, etc. Con el fin de contrastar con los diferentes componentes que hacen parte de un sistema operativo de un dispositivo móvil. (Tanenbaum, 2003)

7.3.1 Procesos.

Un proceso básicamente es un programa en ejecución. Cada proceso tiene asociado un espacio de direcciones: una lista de posiciones de memoria desde algún mínimo (0) hasta algún máximo, que el proceso puede leer y en las que puede escribir. El espacio de direcciones contiene el programa ejecutable, sus datos y su pila. Cada proceso está asociado también con algún conjunto de registros, incluido el contador de programa, el apuntador de la pila, y otros registros de hardware, así como toda la demás información necesaria para ejecutar el programa. (Tanenbaum, 2003)

7.3.2 Administración de memoria.

Toda computadora tiene una memoria principal que usa para contener los programas en ejecución. En un sistema operativo muy simple, solo hay un programa a la vez en la memoria. Para ejecutar un segundo programa, es preciso desalojar el primero y colocar el segundo en la memoria. Los sistemas operativos más avanzados permiten que haya varios programas en la memoria al mismo tiempo. Para evitar que se interfieran (e interfieran con el sistema operativo), se requiere algún tipo de mecanismo de protección. Aunque este mecanismo debe estar en el hardware, está bajo el control del sistema operativo. (Tanenbaum, 2003)
7.3.3 Entradas y salidas

Todas las computadoras tienen dispositivos físicos para obtener entradas y producir salidas. Después de todo, ¿de qué serviría una computadora si los usuarios no pudieran decirle que hacer y no pudieran obtener los resultados en vez realizado el trabajo requerido? Existen muchos tipos de dispositivos de entrada y salida, que incluyen teclados, monitores, impresoras, etc. y al sistema operativo le corresponde administrarlos. Por eso todo sistema operativo cuenta con un sistema de E/S para administrar sus dispositivos de E/S. (Tanenbaum, 2003).

7.3.4 Archivos.

Otro concepto clave que manejan casi todos los sistemas operativos es el sistema de archivos. Como ya señalamos, una función importante del sistema operativo consiste en calcular las peculiaridades de los discos y otros dispositivos de E/S, y presentar al programador un modelo abstracto, bueno y claro de archivos independientes de los dispositivos. Es obvio que se requieren llamadas al sistema para crear archivos, eliminarlos, leerlos y escribirlos. Para poder leer un archivo es preciso localizarlo en el disco y abrirlo, y una vez que se ha leído deberá cerrarse, así que se cuenta con llamadas para realizar estas actividades.

7.4 SISTEMAS OPERATIVOS PARA DISPOSITIVOS MÓVILES

Partiendo de la definición de sistema operativo: Capa compleja entre el hardware y el usuario, concebible también como una máquina virtual, que facilita al usuario o al programador las herramientas e interfaces adecuadas para realizar sus tareas informáticas, abstrayéndole de los complicados procesos necesarios para llevarlas a cabo. Podemos deducir que el uso de uno u otro S.O determinarán las capacidades multimedia de los dispositivos, y la forma de éstas de interactuar con el usuario. Existen multitud de opciones, si bien las más extendidas son Symbian, BlackBerry OS, Windows Mobile, y recientemente iPhone OS y el sistema móvil de Google, Android, además por supuesto de los dispositivos con sistema operativo Linux. Las características básicas de cada uno son las siguientes:

7.4.1 Symbian O.S

Este es el sistema operativo para móviles más extendido entre “Smartphone”, y por tanto el que más aplicaciones para su sistema tiene desarrolladas. Actualmente Symbian copa más del 65% del mercado de sistemas operativos. Su principal virtud es la capacidad que tiene el sistema para adaptar e integrar todo tipo de aplicaciones. Admite la integración de aplicaciones y, como sistema operativo, ofrece las rutinas, los protocolos de comunicación, el control de archivos y los
servicios para el correcto funcionamiento de estas aplicaciones. La tecnología del sistema operativo Symbian se ha diseñado teniendo en cuenta puntos clave como el poder proporcionar la energía, memoria y gestión de entrada y salida de recursos requeridos específicamente en los dispositivos móviles. También, supone una plataforma abierta, ésta es la clave, que aúna telecomunicaciones y los estándares globales de internet. Los usuarios de Symbian señalan como principal ventaja del sistema el hecho de que exista una amplia selección de aplicaciones disponibles para todo tipo de teléfonos móviles. Destacan también la compatibilidad con los estándares de conectividad y redes como Bluetooth, WiFi, GSM, GPRS, CDMA y WCDMA.

7.4.2 BlackBerry OS.

BlackBerry es un sistema operativo multitarea que está arrasando en la escena empresarial, en especial por sus servicios para correo y teclado QWERTY. Actualmente BlackBerry OS cuenta con un 11% del mercado. BlackBerry aparece en el mercado justo en el momento en que comenzaba a demandarse un sistema operativo que permitiera utilizar de una forma fácil, cómoda y rápida los servicios de correo electrónico. Hoy en día es también proveedor de servicios de correo electrónico a dispositivos que no son BlackBerry, gracias al programa BlackBerry Connect. Así, en líneas generales, en un dispositivo BlackBerry es posible redactar, enviar y recibir todo tipo de mensajes de correo electrónico, al igual que en el programa que se utiliza en un ordenador. Además, es posible realizar y contestar a las llamadas que se emitan a través de la red de telefonía móvil, lo que permite sustituir el teléfono móvil. También, como evolución lógica, los dispositivos de este fabricante permiten la navegación por internet en páginas HTML o WAP y tienen la capacidad de enviar o recibir mensajes SMS. Por lo demás, este sistema operativo incorpora múltiples aplicaciones y programas que convierten a los dispositivos en completos organizadores de bolsillo con funciones de calendario, libreta de direcciones, bloc de notas, lista de tareas, entre otras.

7.4.3 IPhone OS

IPhone OS es una versión reducida de Mac OS X optimizada para los procesadores ARM. Aunque oficialmente no se puede instalar ninguna aplicación que no esté firmada por Apple ya existen formas de hacerlo, la vía oficial forma parte del iPhone DeveloperProgram (de pago) y hay que descargar el SKD que es gratuito. IPhone dispone de un interfaz de usuario realmente interesante, la única pega es la cantidad de restricciones que tiene, aunque quizás Apple se dé cuenta que para triunfar mucho más es mejor liberar y dar libertad a su sistema. Aunque su tiempo de vida es corto ya copa casi el 7% del mercado.

7.4.4 Windows Mobile

Microsoft lanzó su propio Windows para móviles, antes conocido como Windows CE o Pocket PC, tiene una larga historia como segundón en el campo de los PDA u ordenadores de bolsillo, sin embargo hace pocos meses superó por primera vez al hasta entonces líder, Palm OS. Windows Mobile es un sistema operativo escrito desde 0 y que hace uso de algunas
convenciones de la interfaz de usuario del Windows de siempre. Una de las ventajas de Windows Mobile sobre sus competidores es que los programadores pueden desarrollar aplicaciones para móviles utilizando los mismos lenguajes y entornos que emplean con Windows para PC. En comparación, las aplicaciones para Symbian necesitan más esfuerzo de desarrollo, aunque también están optimizadas para cada modelo de teléfono.

7.4.5 Android

Google es otro de los desarrolladores que coge algo y es capaz de convertirlo en una referencia. Android es un sistema operativo móvil basado en Linux y Java que ha sido liberado bajo la licencia Apache versión 2. El sistema busca, nuevamente, un modelo estandarizado de programación que simplifique las labores de creación de aplicaciones móviles y normalice las herramientas en el campo de la telefonía móvil. Al igual que ocurriera con Symbian, lo que se busca es que los programadores sólo tengan que desarrollar sus creaciones una única vez y así ésta sea compatible con diferentes terminales. Google promete una plataforma de desarrollo gratuita, flexible, económica en el desarrollo de aplicaciones y simple, diferenciada de los estándares que ofrecen Microsoft o Symbian.

7.5 INGENIERÍA DE SOFTWARE

La práctica de ingeniería de software es un amplio arreglo de conceptos, principios, métodos y herramientas que deben considerarse cuando se planea y desarrolla el software. representa los detalles -las consideraciones técnicas y los cómomas- que están bajo la superficie del proceso de software: las cosas que se necesitarán para realmente construir software de computadora de alta calidad. Esta práctica de la ingeniería del software la aplican los ingenieros de software y sus gerentes. El proceso de software proporciona a todos los involucrados en la creación de un sistema o producto basado en computadora, un mapa del camino para llegar de manera exitosa a su destino. la práctica proporciona los detalles que se necesitan para transitar a lo largo del camino. Indica donde están ubicados los puentes, los bloqueos del camino y los obstáculos. Ayuda a entender los conceptos y principios que se deben comprender y seguir para conducir de manera segura y rápida. Enseña a cómo conducir, donde reducir y donde aumentar la velocidad. En el contexto de la ingeniería del software, la práctica es lo que se realiza a diario mientras el software evoluciona desde una idea hasta una realidad.

¿Cuáles son los pasos? Existen tres elementos de la práctica que se aplican sin importar el modelo de proceso que se escoja. Estos son los conceptos, los principios y los métodos. un cuarto elemento de la práctica -las herramientas- apoya la aplicación de los métodos. ¿Cuál es el producto obtenido? La práctica incluye las actividades técnicas que producen todos los productos del trabajo definidos por el modelo de proceso del software que se ha elegido. ¿Cómo puedo estar seguro de que lo he hecho correctamente? Primero se deben comprender con firmeza los conceptos y principios aplicables al trabajo que se realiza en el momento (por ejemplo, el
diseño). Después es preciso asegurarse que se ha seleccionado un método apropiado para el trabajo; se debe tener la certeza que se ha entendido la forma de aplicar el método y el uso de las herramientas automatizadas cuando éstas son apropiadas para la tarea, y se debe ser firme en la necesidad de usar técnicas para asegurar la calidad de los productos de trabajo que se produzcan. (Pressman)

7.6 INGENIERÍA DE REQUISITOS (REQUIREMENTS).

la ingeniería de requisitos ayuda a los ingenieros de software a entender mejor el problema en cuya solución trabajarán. incluye el conjunto de tareas que conducen a comprender cuál será el impacto del software sobre el negocio, qué es lo que el cliente quiere y cómo interactuarán los usuarios finales con el software. Esta labor la realizan los ingenieros de software (algunas veces referidos como ingenieros de sistemas o analistas en el mundo de la TI) y otros interesados (gerentes, clientes y usuarios finales) participan en la ingeniería de requisitos.

El diseño y la construcción de un elegante programa de computadora que resuelva el problema incorrecto no satisfacen las necesidades de nadie. Por lo tanto, es muy importante entender lo que el cliente quiere antes de comenzar a diseñar y construir un sistema basado en computadora. ¿Cuáles son los pasos? La ingeniería de requisitos empieza con la fase de inicio, la cual es una tarea que define el ámbito y la naturaleza del problema que debe resolverse. después continúa con obtención, que es una tarea que ayuda al cliente a definir sus necesidades; posteriormente sigue con la elaboración, que es la fase donde se refinan y modifican los requisitos básicos. cuando el cliente ha definido el problema se lleva a cabo la negociación, donde se define cuáles son las prioridades, cuáles aspectos son esenciales y en qué momento se requieren. por último, el problema se especifica de alguna manera, y después es validado y revisado para asegurar que el la concepción del problema que tiene el ingeniero de software coincide con la percepción del cliente. ¿Cuál es el producto obtenido? El objetivo del proceso de la ingeniería de requisitos es darle a todas las partes una explicación escrita del problema. Esto puede lograrse por medio de varios productos de trabajo: escenarios de uso, listas de funciones y de características, modelos de análisis o laguna especificación. ¿Cómo puedo estar seguro de que lo he hecho correctamente? El ingeniero de software revisa los productos de trabajo de la ingeniería de requisitos junto con el cliente y los usuarios finales para asegurarse que haya entendido lo que en realidad pretendian decirle. Es necesario hacer una advertencia: aun después de que todas las partes están de acuerdo, las cosas cambian, y continuarán haciéndolo a través de la vida del proyecto. (Pressman)

7.7 MODELADO DE ANÁLISIS.

La palabra escrita es un vehículo maravilloso para la comunicación, pero no es, necesariamente, la mejor forma de representar los requisitos para el software de computadora. El modelado del
análisis utiliza una combinación de formatos en texto y diagramas para representar los requisitos de los datos, las funciones y el comportamiento de una manera que es relativamente fácil de entender y, aún más importante, conduce a una revisión para lograr la corrección, integridad y la consistencia. Esto lo hace un ingeniero de software (algunas veces llamado analista) construye el modelo empleando requisitos obtenidos del cliente.

Para validar los requisitos del software es necesario examinarlos desde algunos puntos de vista diferentes. El modelado del análisis representa los requisitos en múltiples "dimensiones", con lo que se incrementa la probabilidad de encontrar errores, de que surjan inconsistencias y de que se descubran omisiones.

Los requisitos de información, funcionales y de comportamiento se modelan mediante varios tipos de diagramas. El modelado basado en escenarios representa el sistema desde el punto de vista del usuario. El modelado orientado al flujo indica cómo se transforman los objetos de datos al realizarse las funciones del procesamiento. El modelado basado en clases define objetos, atributos y relaciones. El modelado del comportamiento presenta los estados del sistema y sus clases, así como el impacto de los eventos sobre sus estados. Una vez que se han creado los modelos preliminares, éstos se refinan y analizan para evaluar su calidad, integridad y consistencia. Después, el modelo de análisis final lo validan los interesados. ¿Cuál es el producto obtenido? Para el modelo de análisis es posible elegir una amplia variedad de tipos de diagramas. Cada una de estas representaciones ofrece una visión de uno o más de los elementos del modelo. ¿Cómo puedo estar seguro de que lo he hecho correctamente? Los productos de trabajo del modelado del análisis deben revisarse en lo relativo a su corrección, integridad y consistencia. Éstos deben reflejar las necesidades de todos los interesados y establecer una base desde la cual pueda conducirse el diseño. (Pressman)
7.8 ARQUITECTURA DE CLASES.

El modelo de análisis tiene como objetivo generar una arquitectura de objetos que sirva como base para el diseño del sistema. Existen diversas arquitecturas especialmente diseñadas para el manejo de los sistemas de información. Las cuales involucran la manipulación de la información guardada en bases de datos a partir de interfaces de usuario.

La arquitectura se distingue según la organización de los objetos de acuerdo a su funcionalidad. Esto es también conocido como la dimensión de la arquitectura. Por ejemplo: si existe un grupo de objetos para el manejo de la funcionalidad de la aplicación y otro para interactuar con las entidades externas de la aplicación, como el usuario y las bases de datos, entonces se considera que la arquitectura es de dos dimensiones. Por el contrario, si existe un solo grupo de objetos que maneja de manera distinta la funcionalidad junto con la interacción externa, entonces se considera que la arquitectura es de una sola dimensión.

En general, una arquitectura puede incluir cualquier número de dimensiones, algo que depende del tipo de aplicación que se desee desarrollar. En general, el planteamiento es: si se diseña un sistema con cierto número de dimensiones, ¿se obtendría un sistema más estable y fácil de extender que con un número menor o mayor? La respuesta depende de qué tan independiente sean los objetos de un eje de funcionalidad con los demás. Si se cuenta con ejes de funcionalidad completamente ortogonales, algo que es difícil de lograr, el efecto de cambios en una dimensión no debería afectar a las demás dimensiones. Sin embargo, si los grupos de objetos no son lo suficientemente independientes, aun se puede limitar el efecto de los posibles cambios. En el caso de los sistemas de información, una de las arquitecturas más utilizadas es la de Modelo, Vista, Controlador (MVC – Model, View, Control). Popularizada por los ambientes de desarrollo para los lenguajes de programación de Smalltalk. Esta arquitectura se basa en tres dimensiones principales. *Modelo* correspondiente a la información, *Vista* correspondiente a la presentación o interacción con los usuarios y *Control* correspondiente al comportamiento, como se ilustra en la siguiente figura.
Diagrama de tres dimensiones correspondientes a la arquitectura del modelo de análisis, basado en el modelo de casos de uso. En correspondencia con el modelo MVC. La arquitectura para el modelo de análisis se basará en tres tipos o estereotipos de objetos correspondientes a las tres dimensiones anteriores.

7.8.1 Clases con estereotipos

El tipo de funcionalidad o la razón de ser de un objeto dentro de una arquitectura se conoce como su estereotipo.

- El estereotipo Entidad (Entity) para los objetos que guardan información sobre el estado interno del sistema a corto y largo plazo. Estos objetos corresponden al dominio del problema.

- El estereotipo Borde (Boundary) para objetos que implementan las interfaces del sistema en el mundo externo, correspondientes a todos los actores, incluyendo a aquellos que no son humanos.

- El estereotipo Control (Control) para objetos que implementan el comportamiento de la lógica de los casos de uso, especificando cuando y como el sistema cambia de estado.
El diseño es lo que casi cualquier ingeniero quiere hacer. es el sitio donde manda la creatividad, donde los requisitos del cliente, las necesidades de negocio y las consideraciones técnicas se unen en la formulación de un producto o sistema. el diseño crea una representación o modelo del software, pero a diferencia del modelo de análisis (que se enfoca en la descripción de los datos, las funciones y el comportamiento requeridos), el modelo de diseño proporciona detalles acerca de las estructuras de datos, las arquitecturas, las interfaces y los componentes del software que son necesarios para implementar el sistema. Esto lo hacen los ingenieros de software que encabezan cada una de las tareas de diseño.

El proceso de diseño es importante porque permite al ingeniero de software modelar el sistema o producto que se va a construir. Este modelo puede evaluarse en relación con su calidad y mejorarse antes de generar código, de realizar pruebas y de que los usuarios finales se vean involucrados a gran escala. El diseño es el sitio en el que se establece la calidad del software. Los pasos para el proceso de diseño son los siguientes: el diseño presenta el software de diferentes formas. Primero, debe representarse la arquitectura del sistema o producto. Después, se modelan las interfaces que conectan el software con los usuarios finales, con otros sistemas y dispositivos y con los propios componentes que lo constituyen. Por último, se diseñan los componentes del software que se utilizan en la construcción del sistema. Cada una de estas visiones representa una acción de diseño diferente, pero todas deben ajustarse a un conjunto de conceptos básicos del diseño que determinan todo el trabajo de diseño.

El producto obtenido es un modelo que abarca representaciones arquitectónicas, de interfaz, en el nivel de componentes y de despliegue. Se puede estar seguro de que el proceso está correctamente hecho cuando el modelo de diseño lo evalúa el equipo de software en un esfuerzo encaminado a determinar si éste contiene errores, inconsistencias u omisiones; si existe mejores alternativas; y si el modelo puede implementarse dentro de las restricciones, el itinerario y el costo que han sido establecidos. (Pressman)
8 PRESENTACIÓN Y ANÁLISIS DE LOS RESULTADOS

Para suplir la problemática que se presenta en este proyecto sobre la imposibilidad de controlar el tráfico de solicitudes que deben atender los miembros de la comunidad de taxistas llamada "TaxistasTwitteros" por medio de la red social Twitter; se escogió la metodología de ingeniería de software orientada a objetos que plantea el autor del libro “INGENIERÍA DE SOFTWARE ORIENTADA A OBJETOS CON UML, JAVA, E INTERNET”, Alfredo Weitzenfeld. El autor propone una serie de actividades o modelos básicos a seguir para lograr las metas dentro del proceso de desarrollo ingeniería de un proyecto de software; los modelos son: modelos de requisitos, modelos de análisis, modelos de diseño, modelos de implementación, modelos de integración, modelos de pruebas. El objetivo de usar la propuesta clásica que nos presenta Weitzenfeld, es documentar el proceso de ingeniería de software para cada uno de los modelos; teniendo en cuenta que se seguirá el modelo de ciclo de vida cascada.

Para este proyecto se optó por utilizar 3 del total de actividades planteadas, estas actividades son: Modelo de requisitos, Modelo de análisis, Modelo de Diseño; de las cuales cada modelo está definido por un conjunto de sub-modelos que ayudan a describir el proceso entendible para el equipo de trabajo.

Para el modelo de requisitos se trabajó con los siguientes sub-modelos.

8.1 MODELO DE REQUISITOS.

- Descripción del problema.
- Modelo de casos de uso.
- Modelo de interfaces.

El propósito del modelo de requisitos es comprender en su totalidad el problema y sus implicaciones. Los demás modelos, análisis, diseño, implementación, y pruebas dependen directa o indirectamente del modelo de requisitos. Asimismo, este modelo sirve de base para el desarrollo de las instrucciones operacionales y los manuales, ya que todo lo que el sistema deba hacer se describe aquí desde la perspectiva del usuario. (Weitzenfeld, 2005).

8.1.1 Descripción del problema.

En la descripción del problema, se hizo un resumen preliminar de las necesidades que sirvieron como punto de partida para comprender los requisitos del sistema (ir a la descripción del problema).
8.1.2 Modelo de casos de uso.
En el modelo de casos de uso se describe en términos gráficos, las posibles formas en el que los usuarios interactúan con el sistema. Cada usuario se representa como actor, que corresponden a: Pasajero, Taxista, Administrador, como lo muestra el siguiente gráfico. Cada uno de estos actores ejerce una interacción con almenos un caso de uso.
8.1.3 Modelo de interfaces.

EL modelo de interfaces describe la presentación de información entre los actores y el sistema. Se especifica en detalle cómo se verán las interfaces de usuario al ejecutar cada uno de los casos de uso. (Weitzenfeld, INGENIERIA DE SOFTWARE ORIENTADA A OBJETOS CON UML. JAVA E INTERNET, 2005)

Seleccionar el rol dentro del sistema.

Formulario de registro para pasajeros.
El pasajero pide el taxi.

La solicitud le llega al taxista.
El taxista responde a la solicitud y envía una respuesta al pasajero.

8.2 MODELO DE ANALISIS.

Luego de definir el modelo de casos de uso, concretar qué acciones iba a realizar los diferentes usuarios, y todo lo que abarca el modelo de requisitos se inicia el desarrollo del modelo de análisis, cuyo objetivo es comprender y generar una estructura del sistema con base a lo especificado en el modelo de requisitos.

Cuando se desarrolla el modelo de análisis, normalmente se trabaja con un caso de uso a la vez. (Weitzenfeld, INGENIERIA DE SOFTWARE ORIENTADA A OBJETOS CON UML, JAVA E INTERNET, 2005)

Para cada caso de uso se definió un conjunto de clases necesarias, cada clase corresponde a un estereotipo diferente, ya sea estereotipo borde, estereotipo de control, o estereotipo de Modelo. Y cada una tiene definido explícitamente que comportamiento es responsable dentro del caso de uso. Se comienza identificando los objetos borde necesarios, continuando con los objetos de entidad (Modelo) y finalmente, los objetos de control. Este proceso se aplica a los demás casos de uso.
La elección de los diferentes estereotipos por cada caso de uso se hizo con el fin de separar la presentación del procesamiento y de la persistencia de los datos. Si por alguna circunstancia se requiere hacer un cambio en el proyecto, que normalmente los cambios más comunes ocurren en las clases de funcionalidad (Control) y las clases que definen interfaces (Borde), podrá hacerse únicamente a la clase que necesite hacer el cambio, sin que sea afectado todo el proyecto.

Se puede visualizar la estructura del caso de uso “Registrar pasajero” con las diferentes clases de estereotipo que lo conforman. El actor, “Usuario pasajero” interactúa con una clase de tipo Borde, llamada “Formulario Registro Pasajero” Esta clase representa la interfaz gráfica que define el formulario encargado de capturar los datos necesarios del registro. Las clases encargadas del procesamiento y validación de datos son: la clase “manejador Datos Usuarios” y “Gestionar Direcciones”; adicional a eso, estas mismas clases son encargadas de transportar la información a las clases de tipo Identidad “Usuario” y “Direcciones” cuyas clases tienen funciones de almacenar los datos y extraer los datos cuando sean necesarios, desde la base de datos.

Este proceso se realizó con cada uno de los casos de uso, logrando identificar y definir las clases pertinentes para concretar la implementación de MVC.
8.3 MODELO DEL DISEÑO.

Una vez definido el modelo de análisis, y tener especificado parcialmente todo el sistema, pasamos a realizar el modelo del diseño, donde prácticamente se concretó más detalladamente cada interacción de los actores con el funcionamiento del sistema. También logró concretarse la manera en que se generaba la comunicación entre las diferentes partes (Sistema central y terminales).

8.3.1 Descripción de los casos de uso.

La primera actividad con la que se comenzó, fue haciendo una descripción minuciosa de la interacción del actor con los diferentes casos de uso; narrando el flujo de eventos que debían realizarse, y estipulando cual era la respuesta del sistema según cada acción ejecutada, también se debía especificar una serie de pasos como flujo alternativo, que describían las acciones en el momento que no se siguiera el flujo de eventos normal.

Toda la información detallada que se pudo recolectar para obtener un buen flujo de eventos fue gracias al siguiente formato. Para cada caso de uso se realizó el mismo procedimiento de descripción, pero como ejemplo se mostrará la descripción del mismo caso de uso citado anteriormente.

CU2: Registrar pasajero.
Descripción:
Permitir al usuario crear un perfil dentro de la aplicación.

Actores:
Pasajero

Flujo de eventos
Básico:
Información previa:
Para que el actor llegue al formulario de registro, es necesario que tenga la aplicación instalada en el dispositivo y que posterior a este escoja el rol de pasajero.

<table>
<thead>
<tr>
<th>Acción de los actores</th>
<th>Respuesta del sistema</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. El actor comienza el registro llenando los datos correspondientes a: Nombre, Cuenta de usuario de Twitter, diez (10) direcciones posibles.</td>
<td>Se hace saber al actor antes de añadir la</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td><strong>Table #2</strong></td>
<td><strong>Flujo de eventos –Registrar Pasajero</strong></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>2.</strong> El actor introduce una de las direcciones posibles.</td>
<td><strong>4.</strong> El sistema almacena la dirección en la memoria volátil del celular, y habilita la casilla de direcciones para que se pueda introducir una nueva dirección.</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>3.</strong> El actor presiona el botón añadir para guardar la dirección introducida.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>5.</strong> El actor añade la contraseña</td>
<td></td>
</tr>
<tr>
<td><strong>6.</strong> El actor Ingresa de nuevo la contraseña para confirmar la contraseña que digitó previamente</td>
<td></td>
</tr>
<tr>
<td><strong>7.</strong> El actor presiona el botón guardar para almacenar los datos.</td>
<td><strong>8.</strong> El sistema valida que no haya casillas vacías.</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>9.</strong> El sistema valida que la contraseña coincida con la contraseña de confirmación.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>10.</strong> El sistema comprueba la validez de los datos y almacena él:</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Nombre</td>
</tr>
<tr>
<td></td>
<td>- Twitter</td>
</tr>
<tr>
<td></td>
<td>- Dirección.</td>
</tr>
<tr>
<td></td>
<td>- contraseña.</td>
</tr>
</tbody>
</table>

**Nota:** El punto 2, 3, y 4 puede llegar el caso de que se repitan hasta 10 veces en un mismo registro.
Flujo Alternativo:
- si algún campo está vacío, se avisa al actor que debe ingresar el dato faltante.
- si los campos de contraseña y confirmación de contraseña no coinciden, el sistema hará saber al actor que no son exactos y no terminará el registro hasta que introduzca las contraseñas adecuadamente.
- si los datos no almacenan, puede que exista una falla en la conexión con la base de datos.

Precondición:
- El actor debe tener la aplicación instalada
- Escoger el tipo de usuario pasajero.

Pos condición:
- Los datos son almacenados en la base de datos
- El sistema pasa a la siguiente ventana, pedir taxi.

8.3.2 Comunicación entre las partes.

Las partes corresponden a todo aquel dispositivo que va hacer uso del sistema, ya sea un servidor que controle el funcionamiento administrativo del sistema, y donde se alojaran los datos necesarios. O el terminal en este caso un dispositivo móvil (Smartphone) que trabajará como consumidor de los servicios que ofrecerá el servidor.

Cada terminal hará una solicitud al servidor, y este debe estar dispuesto a entregarle una respuesta. Cuando un usuario de rol pasajero hace una solicitud de un servicio de taxi, esta petición viaja vía Internet hacia el servidor, donde este tendrá la obligación de recibir la petición y enviarla en forma broadcast a todos los terminales que estén con el rol de taxista; cada usuario con rol de taxista estará en la disposición de recibir la petición siempre y cuanto se encuentre con la sesión iniciada, y así mismo estará en la disposición de contestar a la solicitud. El servidor recibe la respuesta de los taxistas, pero tiene la responsabilidad de asignar el servicio a un solo taxista, y esta decisión la toma verificando qué taxista fue el más rápido en contestar la petición. Al usuario pasajero le llega la respuesta del servidor, indicando que algún taxista, de todos los que fueron notificados, respondió a la solicitud. Este mismo pasajero debe hacer notificar al servidor de que confirma que ese taxista que ha aceptado acudir a la petición, sea el que atienda la petición. La confirmación es recibida por el servidor, e inmediatamente le notifica al taxista que el pasajero ha confirmado su petición. Y así mismo el taxista confirma que acudirá por la solicitud de servicio. La siguiente imagen muestra como es la comunicación entre las partes.
Enviar solicitud de servicio

1

llega el primero que respondió

4

Enviar confirmación al taxista

5

alarma de solicitud

2

responde solicitud

3

mensaje de confirmación pasajero

6

alarma de solicitud

2

responde solicitud

3

alarma de solicitud

3

alarma de solicitud

2

responde la solicitud

5
8.3.3 Definición de la estructura de almacenamiento de datos
8.3.4 Definición de la plataforma tecnológica.

Software:
- Para que el sistema funcione:

La aplicación se ejecutará sobre dispositivos móviles que tengan como sistema operativo Android desde la versión 2.2 en adelante. Estas plataformas tendrán acceso a la base de datos de la universidad católica de Pereira.

- Para construir el sistema:

Para la construcción de la aplicación, se necesitó todas las herramientas necesarias para construir aplicaciones en la plataforma Android, el entorno de desarrollo más adecuado es Eclipse, ya que la documentación sobre Android esta soportada para este IDE. También se trabajó con lenguajes de programación java y XML, java para realizar la lógica de procesamiento y manejadores lógicos necesarios, y XML para declarar la capa de presentación y construir toda la interfaz. Por lado de la administración y de gestión del servidor se utilizaron tecnologías como PHP, HTML, CSS, Javascript.

Hardware:

- Para que el sistema funcione:

Los dispositivos que se necesitan para hacer uso del sistema móvil, son dispositivos celulares de gamas Smartphone, algunos de los muchos manufactureros son: LG, Samsung, HTC.

Comunicaciones:

Para lograr una comunicación entre el sistema móvil, y el sistema central, se utilizó un protocolo de comunicación que consiste en envio de mensajes SOAP, (Protocolo SOAP) por medio de servicios web.
9 CONCLUSIONES.

Cuando se va a construir un software, es importante tener definido muy bien los requerimientos de lo que realmente se desea que haga el software; tanto como el cliente y el desarrollador deben ser conscientes de estos requerimientos. Si por alguna circunstancia es necesario modificarlos, es más conveniente hacerlo en la etapa de recolección de información, puesto que si se hace después de esto, se generará más costos y más retrasos en la entrega del producto.

No es conveniente estipular un tiempo de entrega ni el costo de lo que realmente vale la construcción del producto sin usar métodos o técnicas de estimación, puede llegar el caso de que realmente no sea el valor y el tiempo que consuma dicha construcción, y puede generar líos de cumplimientos de pólizas entre los clientes y el equipo desarrollador.

En el momento de la construcción del software, es importante tener en cuenta el manejo de estándares de codificación según el lenguaje de programación que se esté manejando, esto ayuda a estructurar el código y hacerlo entendible para terceras personas.
10 RECOMENDACIONES.

Para continuar con la culminación de la construcción del software, se debe tener en cuenta la comunicación entre el nodo central, que se encargara de gestionar la administración de los servicios, con los terminales móviles. Para este proceso se podrá usar cualquiera de los protocolos de comunicación existen como: (SOAP, HTTP)