

CIS1710AP02

AppShipment:

Aplicación Multiplataforma para el envío de mercancías, basada en economía  
colaborativa

Diego Felipe Ramírez González

Johann Sebastián Moreno Mora

PONTIFICIA UNIVERSIDAD JAVERIANA  
FACULTAD DE INGENIERÍA  
CARRERA DE INGENIERÍA DE SISTEMAS  
BOGOTÁ, D.C.

2017

CIS1710AP02

AppShipment:

Aplicación Multiplataforma para el envío de mercancías, basada en economía  
colaborativa

**Autor(es):**

Diego Felipe Ramírez González

Johann Sebastián Moreno Mora

MEMORIA DEL TRABAJO DE GRADO REALIZADO PARA CUMPLIR UNO  
DE LOS REQUISITOS PARA OPTAR AL TÍTULO DE INGENIERO DE  
SISTEMAS

**Director**

Ing. Luis Guillermo Torres Ribero

**Jurados del Trabajo de Grado**

Ing. Edgar Enrique Ruiz

Ing. Diego Alberto Rincón

**Página web del Trabajo de Grado**

<http://pegasus.javeriana.edu.co/~CIS1710AP02/>

PONTIFICIA UNIVERSIDAD JAVERIANA

FACULTAD DE INGENIERÍA

CARRERA DE INGENIERÍA DE SISTEMAS

BOGOTÁ, D.C.

Mayo, 2017

**PONTIFICIA UNIVERSIDAD JAVERIANA  
FACULTAD DE INGENIERÍA  
CARRERA DE INGENIERÍA DE SISTEMAS**

**Rector Magnífico**

Jorge Humberto Peláez Piedrahita, S.J.

**Decano Facultad de Ingeniería**

Ingeniero Jorge Luis Sánchez Téllez

**Director de la Carrera de Ingeniería de Sistemas**

Ingeniera Mariela Josefina Curiel Huérfano

**Director Departamento de Ingeniería de Sistemas**

Ingeniero Efraín Ortiz Pabón

**Artículo 23 de la Resolución No. 1 de junio de 1946**

*“La Universidad no se hace responsable de los conceptos emitidos por sus alumnos en sus proyectos de grado. Sólo velará porque no se publique nada contrario al dogma y la moral católica y porque no contengan ataques o polémicas puramente personales. Antes bien, que se vean en ellos el anhelo de buscar la verdad y la Justicia”*

## **AGRADECIMIENTOS**

*“En agradecimiento a nuestras familias, profesores y amigos por los logros obtenidos a lo largo de nuestro trayecto académico y personal”*

**TABLA DE CONTENIDO**

TABLA DE CONTENIDO	6
INTRODUCCIÓN	12
I - DESCRIPCIÓN GENERAL	14
1. Oportunidad, Problemática, Antecedentes	14
1.1. Formulación del problema	17
1.2. Justificación del problema	17
1.3. Impacto Esperado	18
2. Descripción del Proyecto	18
2.1. Objetivo general	18
2.2. Objetivos específicos	19
3. Metodología	19
3.1 Fase Metodológica 1	19
3.1.1 Método	20
3.1.2 Actividades	21
3.1.3 Resultados Esperados	21
3.2 Fase Metodológica 2	22
3.2.1 Método	22
3.2.2 Actividades	22
3.2.3 Resultados Esperados	22
3.3 Fase Metodológica 3 Documentación	23
3.3.1 Método	23

3.3.2 Actividades	23
3.3.3 Resultados Esperados	23
II – MARCO TEÓRICO	24
1. Marco Contextual	24
2. Marco Conceptual	30
III – ANÁLISIS	37
3.1 Análisis de la competencia	37
3.1.1 Lista de Criterios de Mercado	39
3.2 Análisis de Herramientas	39
3.2.1 Análisis de herramientas alternativas y justificación	42
3.3 Especificación de Requerimientos	43
3.3.1 En relación a las condiciones de desarrollo	44
3.3.2 En relación al usuario	45
3.3.3 En relación a la aplicación	45
3.3.4 Lista de Requerimientos	46
3.3.5 Priorización de Requerimientos	51
3.4 Suposiciones y dependencias	54
3.4.1 Suposiciones	54
3.4.2 Dependencias	55
IV – DISEÑO	55
4.1 Persistencia de Datos	56
4.2 Vista de desarrollo del Sistema	57

4.3 Vista Física del Sistema	59
V – DESARROLLO DE LA SOLUCIÓN	61
5.1 Medición de cambios repentinos en el acelerómetro	61
5.2 Metodología del Desarrollo	65
5.3 Funcionalidades del Sistema	66
VI – RESULTADOS	77
VI – CONCLUSIONES	78
6.1 Análisis de Impacto del Desarrollo	78
6.2 Conclusiones y Trabajo Futuro	79
VII - REFERENCIAS Y BIBLIOGRAFÍA	81
VIII - ANEXOS	4
Anexo 1: Software Requirements Specification	4
Anexo 2: Requerimientos y Casos de Uso	4
Anexo 3: Software Design Description	4
Anexo 4: Manual de Usuario	4
Anexo 5: Manual de Instalación	4
Anexo 6: Bibliografía	5
Anexo 7: Colección de enlaces	5
Anexo 8: TDD Pruebas y Validación	5
IX – ANEXOS BIBLIOTECA	6



## **Lista de Tablas**

Tabla 1 Análisis de la Competencia.....	38
Tabla 2 Análisis de Herramientas .....	41
Tabla 3 Herramientas alternativas .....	43
Tabla 4 Lista de Requerimientos .....	51
Tabla 5 Requerimientos Implementados.....	54
Tabla 6 Tabulación de resultados del acelerómetro.....	64
Tabla 7 Árbol de navegación AppShipment .....	76

## Lista de Figuras

Ilustración 1 MER AppShipment.....	57
Ilustración 2 Vista de desarrollo del Sistema.....	59
Ilustración 3 Vista Física del Sistema.....	60
Ilustración 4 Ecuación de Sobre Aceleración.....	62
Ilustración 5 Toma del acelerómetro implementado.....	63
Ilustración 6 Login.....	68
Ilustración 7 Home.....	69
Ilustración 8 Registro.....	70
Ilustración 9 Crear Servicio.....	71
Ilustración 10 Iniciar servicio de envío de encomiendas.....	72
Ilustración 11 Tracking de Paquete.....	73
Ilustración 12 Calificación de Transportador.....	74
Ilustración 13 Alerta Sobre Aceleración.....	75
Ilustración 14 Terminar servicio.....	76

## **ABSTRACT**

*AppShipment is a multiplatform application, which aims to provide an alternative to the process of Shipment. Collaborative economies base this application to offer to natural people the maximum use of their goods, to provide services to the community to obtain any economic remuneration. Mobile and web technologies allow to exploit this alternative given its easy accessibility and portability, which is common in the real technological context thus, describe a solution as an optimal and innovative way.*

## **RESUMEN**

AppShipment es un aplicativo multiplataforma, que pretende brindar una alternativa más al proceso de envío de encomiendas. Las economías colaborativas fundamentan esta aplicación con el ánimo de ofrecer a personas naturales el máximo aprovechamiento de sus bienes para brindar un servicio a la comunidad y así mismo obtener remuneraciones económicas. Las tecnologías móviles y web permiten explotar dicha alternativa dado a su fácil accesibilidad y portabilidad la cual es común en el contexto actual tecnológico por lo cual, se describe una solución al problema adecuada e innovadora.

AppShipment fue desarrollado en Scala (Java) bajo el framework de Play, junto con tecnologías Front-End de uso actual tales como Ionic 2, Angular y TypeScript para lograr el alcance propuesto en el trabajo de grado y así mismo, lograr una experiencia de usuario satisfactoria y fluida.

## INTRODUCCIÓN

La presente memoria incluye todo el proceso que fue desarrollado a partir de la propuesta de grado “AppShipment: Aplicación multiplataforma para el envío de mercancías, basada en la economía colaborativa”. En los párrafos posteriores se explica cómo se encuentra distribuido y organizado el presente documento.

La primera sección busca explicar el contexto sobre el cual está desarrollado el trabajo de grado; como un aplicativo multiplataforma puede brindar una alternativa al proceso común de enviar encomiendas, la razón de porqué es importante atacar dicha problemática y así mismo, detallar la manera en que se buscó una solución a la alternativa planteada por AppShipment, incluyendo la metodología descrita en la propuesta de grado que fue usada para lograr el alcance establecido. La segunda sección busca contextualizar al lector con la terminología, conceptos y demás asociados al trabajo de grado y así, garantizar la comprensión de la solución elaborada.

La tercera sección indica las consideraciones adquiridas antes de proceder al desarrollo de la solución, incluyendo el análisis de mercado que justifica el alcance y actores del trabajo de grado, elección y justificación de las herramientas a usar para el desarrollo de la solución. Así mismo, se incluye el análisis de las variables relacionadas con los requerimientos y el plan de pruebas implementado.

En la cuarta sección se detalla la funcionalidad del sistema a un alto nivel, se muestra la Arquitectura de software la cual proporciona un marco definido y claro para interactuar con el código fuente de AppShipment. Así mismo, refleja las restricciones planteadas y es diseñada con base a los objetivos descritos del trabajo de grado.

En la quinta sección se describe el desarrollo de la solución propuesta en el trabajo de grado, en la cual se especifica la metodología de implementación durante el tiempo de calendario establecido. Además del flujo de navegación de AppShipment soportado por pantallas que muestran la ejecución e implementación del prototipo final.

Ya en la sexta sección se presentan los resultados y conclusiones del trabajo de grado. Se presenta el reporte de pruebas realizado por el equipo de desarrollo a cada una de las principales funcionalidades del prototipo final de AppShipment esto con ánimo de contextualizar al lector en cómo se garantiza el correcto desempeño y ejecución del sistema. Además, se presenta el análisis de impacto en diferentes ámbitos para justificar el posible trabajo futuro que AppShipment deja como legado o como idea como tal.

## **I - DESCRIPCIÓN GENERAL**

Esta sección introduce al lector a entender el problema planteado en este Trabajo de Grado, cómo y por qué fue formulado, así mismo el impacto esperado en los futuros usuarios una vez concluya el desarrollo e investigación de AppShipment.

### **1. Oportunidad, Problemática, Antecedentes**

El transporte de mercancías es un proceso logístico bastante robusto en el cual se tienen en cuenta muchas variables para que el envío de mercancías sea exitoso. Este éxito se alcanza siempre y cuando los paquetes se adecuen a los procesos de negocio y logística establecidos. En un proceso básico de envío de encomiendas el usuario debe llevar su mercancía a centros de atención donde la recepción estará sujeta a restricciones como: disponibilidad de transporte, rutas, centros de atención de la franquicia en los lugares de destino o intermediarios que continúen con el transporte entre otros, una vez el usuario no se vea afectado por dichas restricciones, deberá diligenciar y firmar documentos para que su mercancía sea empacada y empiece su recorrido [1]. Este proceso en logística y planeación es innecesariamente extenso a la hora de transportar mercancías de pequeñas dimensiones y peso por lo que podría desarrollarse un proceso y una logística adecuada para estas situaciones que facilite a las personas el envío de mercancías pequeñas basándose en la economía colaborativa.

La economía colaborativa es un modelo económico que ha tomado fuerza a nivel mundial [2] ya que facilita el desarrollo de diversas actividades económicas, favoreciendo a personas naturales, cuyo tiempo y/o recursos se ponen a disposición y servicio de una manera fácil y rápida para la satisfacción de necesidades de otras personas. Este modelo se basa en redes de confianza donde la confianza entre los usuarios es primordial y ambas partes se ven beneficiadas sin tener que mediar con

empresas que pueden prestar los mismos servicios, pero cuyos procesos y logística podría llegar a ser más extensa y complicada. [2]

En Colombia actualmente, el desarrollo y uso de aplicaciones web y/o móviles de este tipo no son aceptadas con facilidad dado que suele causar desconfianza, sin embargo, aplicaciones como Uber, AirBnB, BitCoin, entre otras, han empezado a abrir brecha [3] y han comenzado a cambiar la percepción de los colombianos frente a este tipo de aplicaciones además de que ha permitido generar ingresos económicos extras a sus usuarios.

A nivel nacional ha surgido un potencial competidor ante el trabajo de grado planteado, **Mensajería Urbana (MensajerosUrbanos.com)** nace en Julio de 2013 como una iniciativa tecnológica en la industria de la mensajería. Su modelo de negocio se basa en la contratación formal de mensajeros a los cuales, se les brinda toda infraestructura para prestar el servicio a la ciudadanía, enfocados al sector informal.

Disponen de una plataforma tecnológica para brindar el servicio, el cual brinda como factor crítico de éxito:

- Tracking del paquete o mensaje
- Costos e información previa a la ejecución del servicio
- Declaración obligatoria de la mercancía a transportar como seguro al cual responde la aplicación por su valor total
- Geolocalización de punto de partida y entrega
- Posee propia infraestructura y personal numeroso

AppShipment como aplicación pretende, a diferencia de sus posibles competencias, aprovechar al máximo los recursos de cualquier tipo de persona que desee adquirir

una remuneración económica al transportar encomiendas, y hacer uso de una plataforma tecnológica que se lo facilite brindando una alternativa diferente a lo común.

AppShipment es la aplicación resultante de este proyecto la cual busca fomentar prestaciones de servicios basados en la confianza entre usuarios y economías colaborativas a la hora de enviar mercancías, donde a través de una plataforma las personas, usuarios prestadores del servicio, puedan poner a disposición su tiempo y recursos para transportar encomiendas de otras personas, que son potenciales usuarios demandantes del servicio, y obtener remuneración económica.

Para AppShipment existen básicamente 2 necesidades a satisfacer:

- **Usuarios prestadores del servicio:**

Los usuarios prestadores del servicio son todos aquellos que cuentan con los recursos, ya sea espacio en un camión de transporte, maleta o vehículo, y que podrían llevar la mercancía de otra persona con el objetivo de optimizar el uso y sacarle un mejor provecho a sus recursos obteniendo una remuneración económica.

La aplicación AppShipment se encargaría de satisfacer la necesidad de encontrar la mercancía ideal, es decir, que cumpla con las condiciones de peso, dimensiones, ruta o destino para que el usuario prestador del servicio pueda transportarla y evidenciar su remuneración una vez se haya entregado la mercancía exitosamente.

- **Usuarios demandantes del servicio:**

AppShipment quiere ofrecer una alternativa diferente a la hora de enviar sus mercancías frente a las que les ofrecen compañías ya existentes como Servientrega, Coordinadora, DHL, FedEx, entre otras. AppShipment les



permitirá encontrar una persona, que posee el tiempo y los recursos, para llevar esta mercancía a su destino.

### **1.1. Formulación del problema**

¿Cómo facilitar el envío de mercancías mediante un aplicativo multiplataforma, basado en modelos de economía colaborativa?

### **1.2. Justificación del problema**

El reciente crecimiento de mercados P2P (Persona a Persona) en varias regiones del mundo, conocidos popularmente como economías colaborativas, ha emergido como una alternativa para personas naturales. Esto con el objetivo de poder obtener ingresos a raíz de servicios y/o bienes de los cuales no encontraban un posible potencial. Este tipo de alternativa a los servicios que, tradicionalmente son exhibidos por industrias, están siendo impactadas por este nuevo auge de plataformas basadas en estos modelos económicos. [4]

Se quiere reconocer a la persona natural como potencial competidor frente a estas empresas grandes del sector, explotando todo bien o recurso que este pueda poseer. En este caso, el proceso de envío de mercancías en Colombia es manejado actualmente por varias empresas de este sector tales como Servientrega, Coordinadora, DHL, etc.

Los aplicativos multiplataforma, basados en modelos de economía colaborativa, serían la herramienta que se quiere dar a conocer en Colombia, además de fomentar el uso de las mismas para un beneficio común entre los usuarios. Un ejemplo de ello son los casos de éxito en todo el mundo como AirBnB, Uber y BlaBlaCar. Tales aplicaciones justifican por qué hacer uso del mismo modelo en nuestro país.

AppShipment es una solución basada en el mismo modelo explicado en la sección 1, el cual brinda una alternativa al proceso común y molesto de enviar mercancía o encomiendas. [5]

Se tiene como factor crítico de éxito el reciente auge en el uso de aplicaciones web y móviles en Colombia [6]. Adicionalmente, existe la posibilidad de innovación al momento de brindar tal alternativa hacia los usuarios que, en este caso, hace referencia a personas con posibilidad de brindar medios y recursos para transportar mercancías de tamaño mediano/pequeño para otras personas a cambio de una remuneración económica. Dicha remuneración será uno de los posibles valores agregados de AppShipment para los usuarios. [7]

### **1.3. Impacto Esperado**

Se espera que a mediano y a largo plazo, AppShipment sea un aplicativo integral y de uso esencial a toda persona involucrada en procesos de envío de encomiendas y así mismo, a cualquier persona que desee obtener un ingreso económico extra haciendo uso de sus productos, convirtiéndolos en servicios.

## **2. Descripción del Proyecto**

Esta sección aborda la descripción general del Trabajo de grado. Incluye los objetivos tanto generales como específicos, los cuales son fundamentales para lo que fue el desarrollo e implementación del Trabajo de Grado.

### **2.1. Objetivo general**

- Construir una aplicación multiplataforma basada en economías colaborativas que permitan a las personas encontrar una alternativa con base tecnológica

para él envío de mercancías a nivel nacional.

## **2.2. Objetivos específicos**

- Realizar proceso de ingeniería de Requerimientos para construir el aplicativo AppShipment
- Diseñar un aplicativo multiplataforma con base en tecnologías web y móvil
- Implementar el aplicativo AppShipment basado en el diseño realizado anteriormente
- Realizar procesos de validación y pruebas orientadas a TDD

## **3. Metodología**

Para la realización de esta aplicación práctica, se va a utilizar el Framework Disciplined Agile Delivery (DAD). Este Framework está basado en metodologías ágiles y nace para poder sacar provecho a las metodologías existentes tales como Scrum, XP y Lean. Gracias a esto DAD nos permitió combinar las metodologías anteriormente mencionadas y proporcionar un esquema ágil personalizado a nuestras necesidades.

Siguiendo el Framework propuesto se definen las siguientes fases metodológicas

- Fase metodológica 1 (Concepción)
- Fase metodológica 2 (Construcción)
- Fase metodológica 3 (Documentación)

### **3.1 Fase Metodológica 1**

La primera fase consiste en la búsqueda de fuentes de información necesarias para dar inicio al proyecto. Una parte de las fuentes de información tendrá un enfoque

cuantitativo, mientras que la otra un enfoque cuantitativo. En el enfoque cualitativo se pretende conocer a fondo el proceso del envío de encomiendas y un análisis de consumidores para entender aspectos del diario vivir en cuanto a decisiones de envíos de encomiendas. En el enfoque cuantitativo se pretende conocer datos aproximados a la realidad sobre el mercado del envío de encomiendas, tales como tamaño de población objetivo (Segmento de Mercado), cantidad de vehículos activos circulando a nivel nacional, tarifas promedio por envíos de encomiendas y usuarios con Smartphone en Colombia.

Posteriormente con la información encontrada en ambos enfoques se hace un análisis completo y minucioso con el cual lograr tener una abstracción del estudio de mercado. Finalmente, se lleva a cabo el levantamiento de requerimientos, planificación del proyecto y asignación de tareas de cada integrante del proyecto de grado.

### **3.1.1 Método**

Para llevar a cabo el enfoque cuantitativo, se hará una investigación usando fuentes de información tales como periódicos, revistas, libros y publicaciones científicas en internet que estén relacionadas con el tema de envío de encomiendas.

Para llevar a cabo la recolección de datos con un enfoque cualitativo se realizan entrevistas y encuestas al segmento de mercado seleccionado. Adicionalmente, se efectúan reuniones y entrevistas con personas relacionadas al tema, para conocer el estado del arte del proceso actual del envío de encomiendas. Así mismo, se realiza observación directa sobre la competencia para diferenciar el proceso propuesto por AppShipment frente a la competencia estudiada.

### **3.1.2 Actividades**

- Consultar fuentes de información cualitativas y cuantitativas acerca del Mercado de envío de encomiendas
  - Realizar Entrevistas y Encuestas a clientes o especialistas en el tema
  - Hacer observación directa a la competencia
  - Buscar Fuentes secundarias:
    - Periódicos
    - Revistas
    - Libros
    - Publicaciones Científicas en Internet
- Segmentar Mercado Objetivo
- Analizar la información
  - Gráficos y tablas de datos acerca del mercado
- Generar un Estudio de mercado
  - Proyección de ventas
  - Planeación del proyecto y tareas
- Levantamiento de requerimientos
- Elaboración del documento SRS

### **3.1.3 Resultados Esperados**

- Estudio de mercado
- Diagrama de Gantt del proyecto
- Encuestas tabuladas
- Reportes de entrevistas y reuniones
- Resúmenes de datos

- Gráficos y tablas de los resúmenes de datos
- Documento SRS
  
- Marco teórico parcial de la propuesta de grado

### **3.2 Fase Metodológica 2**

Esta fase se realiza la implementación del prototipo cumpliendo con los requerimientos levantados previamente en la **fase metodológica 1**.

#### **3.2.1 Método**

Durante la etapa de construcción se realiza la validación de los requerimientos y de los casos de uso planeados e implementados en iteraciones de una semana. También se contempla el desarrollo de pruebas unitarias y de integración para validar que el desarrollo incremental de la aplicación cumpla con las funcionalidades definidas.

#### **3.2.2 Actividades**

1. Refinar especificación de requerimientos.
2. Implementar los requerimientos funcionales.
3. Realizar las pruebas unitarias y de integración a las funcionalidades desarrolladas cada semana.
4. Validar si las funcionalidades implementadas están acordes con los requerimientos, en caso de no ser así, realizar los ajustes pertinentes en el siguiente ciclo de desarrollo.

#### **3.2.3 Resultados Esperados**

Al final de esta etapa se cuenta con un prototipo funcional, en el cual se encuentran desarrolladas las funcionalidades especificadas en los requerimientos. Este prototipo estará listo para pasar a un ambiente de pruebas para retroalimentación de los usuarios.

### **3.3 Fase Metodológica 3 Documentación**

La tercera y última fase consiste en crear y complementar los documentos necesarios para el trabajo de grado. Estos documentos son importantes debido a que son el medio por el cual se puede comprender el proyecto a desarrollar y el prototipo de AppShipment que se implementó.

#### **3.3.1 Método**

Para llevar a cabo esta fase de documentación se realizan revisiones a cada uno de los entregables y documentos realizados anteriormente, además de realizar los documentos faltantes como los son los manuales y la documentación del código fuente.

#### **3.3.2 Actividades**

- Verificación de documentos usando una plantilla
- Completar documentos realizados en las demás fases metodológicas.
- Creación de manuales de usuario y de instalación

#### **3.3.3 Resultados Esperados**

- Documento SRS final
- Documento SDD final
- Documentación del código
- Documento de reporte de pruebas
- Manuales de usuario y de instalación
- Memoria de trabajo de grado
- Propuesta de trabajo de grado final

## II – MARCO TEÓRICO

Los términos teóricos que se exponen en esta sección, contextualizan y acercará al lector a entender la problemática expuesta en la sección 1.

### 1. Marco Contextual

A continuación, se exponen los diferentes términos y elementos conceptuales y contextuales que sirvieron como base para el desarrollo de este proyecto.

#### **Economías Colaborativas**

Dado a que la economía evoluciona, está en constante cambio, nace en el ámbito tecnológico lo que se conoce como economía colaborativa [11].

Basándose en la cultura del compartir, reutilizar y ahorrar todo recurso que se posea. La economía colaborativa pretende transformar dichos recursos. Hace referencia al fenómeno que actualmente tiene un impacto de gran escala en nuestra actual sociedad que ha hecho que las personas puedan resolver y cubrir sus necesidades basándose en la oportunidad de aprovechar y poner a disposición aquellos recursos y servicios que disponga o posea una persona. Este fenómeno ha cambiado la perspectiva de las personas y encontrar una alternativa para viajar, comer, vestir, escuchar música, conducir, transportarse, etc.

En esencia, cuando se habla de economía colaborativa y su dimensión global gracias a la tecnología, concluye en el modelo de consumo basado en el **intercambio de bienes y servicios entre particulares** [11]. Gracias a la era digital en la que vivimos, toda persona natural puede organizar dicho intercambio mediante plataformas digitales y aportar a esta actividad comercial un valor agregado en diferentes



aspectos:

- Mayor alcance y oportunidad de acceder a nichos de mercados amplios.
- Gestión Logística transparente para el usuario
- Crear confianza entre usuarios
- Escalabilidad y oportunidad de negocio sin necesidad de una gran inversión

Actualmente existen dos grandes enfoques para hacer uso de modelos de economía colaborativa [8]: i) el primero agrupa a toda persona que quiere sumar esfuerzos para lograr un objetivo común, con casos de éxito como Kickstarter, Goteo y Verkami, los cuales brindan una alternativa para financiación en donde se incentiva a la comunidad y no a los bancos. ii) existe el enfoque donde se organiza el préstamo, intercambio o donación entre particulares, con casos de éxito como Airbnb y Uber, los cuales brindan sus plataformas tecnológicas para convertir productos en servicios.

### **Posicionamiento Geográfico**

Para ubicar dos puntos en el globo terráqueo, usaremos las coordenadas geográficas, estas coordenadas se interpretan con dos valores: la longitud y la latitud. Para representar rutas utilizaremos conjuntos de puntos ordenados, que unidos por líneas, nos permiten representar rutas sobre un mapa [9].

- Latitud: La latitud se define cuanto al sur o al norte nos encontramos en el globo terráqueo.
- Longitud: La longitud se define cuanto al este o al oeste nos encontramos en

el globo terráqueo.

Existen dos formas habituales de representar las coordenadas geográficas la notación DMS y la notación decimal, Para este proyecto haremos uso de la notación decimal. En esta notación simplemente se especifica un número decimal. En el caso de las latitudes, si el número es positivo, significa que se trata de una latitud del hemisferio norte, mientras que si es un número negativo, significa que la latitud es del hemisferio sur. Igualmente, si una longitud es negativa la longitud está al oeste del meridiano Greenwich, y si es positiva, la longitud se encuentra en el este. Ejemplos de notación decimal se mostrarán a continuación [9]:

- Notación decimal de coordenadas geográficas de Bogotá: (4.6482976,-74.1078071)
- Notación decimal de coordenadas geográficas de Medellín: (6.2686734,-75.6664331)

### **Sistema de información Geográfica**

Integración de software y datos geográficos diseñados para capturar, almacenar, analizar y desplegar información geográfica con el fin de resolver problemas complejos de planificación y gestión, actualmente los datos geográficos que se manejan en el envío de mercancías a nivel nacional son [13]:

- Coordenada geográfica en notación decimal
- Dirección, que es una combinación de caracteres que describen un sitio o lugar en el que se encuentra una cosa u objeto
- Barrio, parte o extensión de tierra medianamente grande que agrupa un área

de casas.

- Sector, concepto urbanístico que designa a una división espacial en una ciudad.
- Ciudad, concepto urbanístico que designa a una división espacial en un país.

### **Datos Actuales del Mercado de Envío de Mercancías**

Para definir correctamente los datos de del mercado, se debe hacer previa investigación de mercado para dar cuerpo a una descripción detallada de un usuario final, después de definir el usuario se debe proceder a conocer a los clientes (quienes pagan por un servicio o producto) usando un estudio de mercado y analizando las problemáticas diarias que tienen los clientes, es recomendable saber que la aplicación se construye basado en las necesidades de los posibles clientes [16]. Actualmente existen empresas de envío de encomiendas como Servientrega, FedEx, DHL [17][18], etc. las cuales para lograr sus operaciones diarias usan las siguientes datos en su negocio:

- El remitente es quien envía el paquete y posee los siguientes datos para hacer uso del servicio:
  - Nombres y apellidos
  - Dirección
  - Teléfono
  - Ciudad
- El destinatario es a quien le llegara el paquete enviado y posee los siguientes

datos para hacer uso del servicio:

- Nombres y apellidos
  - Dirección
  - Teléfono
  - Ciudad
- El repartidor o distribuidor de paquetes, quién es actualmente contratado por cada una de las agencias de envío de paquetes, el cual debe tener la siguiente información para hacer uso del vehículo a nivel nacional:
    - Nombre completo
    - Cedula
    - Numero de Celular
    - Sexo
    - Placas del carro
    - Seguro Obligatorio de Accidentes de Tránsito (SOAT): es un seguro obligatorio establecido por ley, su objetivo es asegurar la atención inmediata de víctimas de accidentes de tránsito [19].
    - Licencia de conducción: documento el cual autoriza a una persona para la conducción de vehículos con validez en todo el territorio nacional [20].

- Antecedente judicial o pasado judicial: dato que se validará haciendo uso del sistema de información de la policía, se consultará con un cedula y así conocer los antecedentes penales y judiciales asociados a una persona [21].
- Las características asociados al paquete que se enviará son la siguientes [17] [18]:
  - Contenido: es el contenido del paquete, el usuario describe que contiene el paquete.
  - Tipo: clasificación del paquete dependiendo del contenido.
  - Peso: medida o valor numérico el cual define la masa de un objeto
  - Longitud de cada arista: Línea que resulta de la intersección de dos superficies. Se usan actualmente aristas rectangulares, tales como cajas con dimensiones estandarizadas de 30x15x15, 30x30x30 y 40x40x40, respectivamente, las cajas se usarán dependiendo el tipo del paquete.
  - Volumen: medidas de espacio que ocupará un paquete en tres dimensiones, el volumen se usará dependiendo el tipo del paquete.
- Dentro de los paquetes que están permitidos según regulaciones internacionales y locales se encuentran [22] [23]:
  - **EXPLOSIVOS:** Fuegos artificiales – Petardos – Municiones – Pólvora - Bengalas
  - **GASES COMPRIMIDOS:** Mecheros - Tanques de buceo – Camping gas

- Aerosoles – Extintores.
- o **LÍQUIDOS INFLAMABLES:** Encendedores de gasolina - Líquidos para encendedor - Pinturas al óleo - Sustancias adhesivas - Perfumes inflamables.
- o **SUSTANCIAS INFLAMABLES:** Fósforos (Cerillas) – Carbones.
- o **SUSTANCIAS OXIDANTES:** Químicos generadores de oxígeno - Peróxidos/Polvos blanqueadores.
- o **SUSTANCIAS TÓXICAS E INFECCIOSAS:** Pesticidas - Químicos agrícolas - Compuestos de mercurio – Bacterias – Virus
- o **MATERIAS RADIOACTIVAS**
- o **MATERIAS CORROSIVAS:** Baterías húmedas – Mercurio (Termómetros) - Ácido clorhídrico.
- o **OTROS:** Productos magnetizados – Motores - Dióxido de carbono sólido (hielo seco).
- Existen restricciones para los paquetes anteriormente mencionados para que sean transportadas correctamente estas mercancías deberán ir acompañadas de documentos como: licencias, certificados, autorizaciones de transporte [22].

## 2. Marco Conceptual

En esta sección se detallan los conceptos tecnológicos que el lector debe entender para entender por qué AppShipment como aplicación es desarrollado bajo estos conceptos y herramientas.

## **Aplicación Multiplataforma**

En la actualidad existen una gran variedad de conexiones, lenguajes y dispositivos (Smartphone, Tablet, Pc) los cuales son incompatibles entre sí, y también desarrollar para cada uno de estos lenguajes y dispositivos cuesta demasiado, por esa razón nació el concepto de desarrollo multiplataforma en el que desarrolladores y organizaciones hacen un solo desarrollo y lo exportan a los diferentes dispositivos móviles, entre las herramientas que facilitan el desarrollo multiplataforma se encuentra, Ionic 2; sin embargo no todo es una maravilla ya que el desarrollo puede presentar algunos inconvenientes con algunos sensores de los dispositivos y los datos que arroja no son tan exactos, por ejemplo la precisión del GPS usando una herramienta multiplataforma no es tan exacta como lo es un desarrollo en lenguaje nativo del dispositivo.

- Ionic 2 hace parte de la solución como un Framework que permite el desarrollo de aplicaciones híbridas, aunque inicialmente fue pensado para móviles y tablets, permite al equipo de desarrollo implementar aplicaciones web. Es basado en el framework Front-End de Angular 2 y TypeScript. Permite el desarrollo de aplicaciones realizando un solo código base que puede ser fácilmente desplegado en otras plataformas, evitando la replicación de código [12].

## **Desarrollo Orientado a Pruebas (TDD)**

Es una disciplina ágil de desarrollo de software, donde los desarrolladores podrán hacer casos pruebas correctas y automatizadas para mejorar o adicionar nuevas características antes que los desarrolladores escriban algo de código. La premisa básica de TDD es que se comienza por escribir una prueba de fallos para la más

simple pieza de funcional que se desea implementar, después se escribe una premisa que tendrá que pasar la prueba, una vez hecho esto, el nuevo código se cambia o se mejora en otras palabras se logra asegurar que cumpla los requerimientos estándar del código, pero cuales son los beneficios de usar TDD [24]:

- Asegurar la calidad, enfocarse en los requerimientos antes de escribir una línea de código.
- Mantiene el código claro, simple y fácil de probar.
- Provee documentación para los diferentes miembros del equipo, conocer cómo trabaja el sistema para todos los miembros, además facilita la comunicación entre los miembros del equipo.
- Pruebas repetibles.
- Permite Cambios rápidos.
- Evita sobre-diseñar
- Incrementa la productividad

## **Angular2**

Framework de desarrollo para aplicaciones web, que ha evolucionado para brindar proyectos más robustos en el ámbito empresarial y así lograr aplicaciones web híbridadas, que facilitan y dan una nueva experiencia desarrollando del lado del cliente, angular es el mejor lenguaje para lograr un Front-End perfecto ya que este brinda una nueva tecnología que funciona como una caché evitando volver a cargar páginas anteriormente cargadas en otras palabras la eficiencia de carga de contenido es más



rápida[25].

### **Servidor Amazon Web Services (AWS)**

Amazon Web Services (AWS) es una plataforma de servicios en la nube que ofrece potencia de cómputo, almacenamiento de base de datos, entrega de contenido y otras funcionalidades para ayudar a las empresas a escalar y crecer[26], en otras palabras es una herramienta que puede servir como solución a la hora de iniciar un emprendimiento o una nueva idea de negocio haciendo uso de bootstrapping, por que la inversión inicial es mínima ya que AWS brindara un servicio gratuito hasta un año pero con unas restricciones de servicio mensuales.

### **APIs de sistemas de información geográfica**

Actualmente existen empresas con mucha trayectoria e investigación en temas relacionados con sistemas de información geográfica, adicionalmente llevan mucho tiempo capturando información geográfica, la cual está en constante actualización. Herramientas de SIG que existen y son muy usadas se encuentran Google Maps, Bing y OpenStreetMaps, las cuales brindan una API para hacer uso de todas las funcionalidades que nos ofrecen tales como acceso a todo tipo de mapas, servicios de geolocalización, datos e información acerca de puntos en un mapa y muchos más servicios. La selección de que API usar depende de qué funcionalidad se desee para el desarrollo. El API de Google maps es el más usado a nivel mundial gracias a su interfaz intuitiva y fácil de usar y sus búsquedas rápidas, también de ser el sistema de información geográfica con más información y mapas, ya que están en constante modificación [27].

- API Google Places, ofrece acceso ilimitado a la base de datos mundial Google

con más de cien millones de fichas de empresas y lugares de interés

- API Directions, ofrece instrucciones para llegar en automóvil en 199 países, le permite ayudar a sus usuarios a encontrar la forma de llegar a sus lugares de destino.
- Permite a sus usuarios ver mapas de gran precisión con visual Street View.

Por otra parte también existen métricas para calcular distancias tales como distancia Manhattan, distancia Haversine, las cuales tienen sus ventajas y desventajas por ejemplo, la distancia Manhattan es una forma de geométrica con una nueva métrica que reemplaza la distancia euclidiana ya que la distancia más corta de dos puntos se calcula basándose en variaciones que tiene el terreno como ciudades, este concepto es muy usado para resolver problemas en los que se involucran distancias en carreteras y vías, y mostrar la ruta más corta para los conductores [28]. Por otro lado, la distancia Haversine es muy usado para calcular distancia entre dos puntos sobre la esfera terrestre, es decir calcularemos la distancia lineal entre dos posiciones dando como datos de entrada la latitud y longitud de dos puntos en el globo terrestre contemplando la curvatura del globo terrestre, la desventaja de usar esta fórmula es que calcula una distancia lineal sin tener en cuenta los obstáculos y variaciones de los diferentes tipos de terreno [29].

### **Trabajos Importantes en el área**

Este trabajo de grado gira en torno a 2 grandes grupos: trabajos que aplican las **economías colaborativas en otros ámbitos** como en el transporte de pasajeros y el **modelo de transporte actual** como Servientrega, DHL, FedEx, entre otros, para dar una solución unificada que favorezca el transporte de mercancías tomando lo mejor

de cada uno de los grupos.

### **Economías Colaborativas en otros ámbitos**

Esta sección contextualiza a AppShipment frente a otros casos de economía colaborativa actuales que han tenido éxito en ámbitos diferentes al proceso de envío de mercancías y encomiendas.

- **UBER** **[30]:**  
 Se toma Uber en representación de los trabajos similares ya que actualmente es uno de los máximos exponentes en su área. Uber busca dar una nueva alternativa tecnológica para dar solución a los problemas de transporte de personas en todo el mundo basándose en economías colaborativas y redes de confianza. Uber actualmente se encuentra en aproximadamente 529 ciudades alrededor del mundo **[30]**, en las cuales permite a los propietarios de los vehículos aprovechar sus recursos para prestar un servicio y obtener una remuneración económica mejor que en los servicios clásicos de transportes de personas.

Ciertamente el servicio que presta Uber abre la posibilidad a los usuarios de enviar con un conductor paquetes o mercancías pequeñas y/o medianas, sin embargo, se presentan varios inconvenientes a la hora de realizar este proceso los cuales se expondrán a continuación:

- **Área de cobertura:** La cobertura de Uber a nivel nacional de es aproximadamente 9 ciudades **[30]** en las cuales se encuentran Bogotá, Medellín, Cali, Cartagena, Barranquilla, Cúcuta, entre otras, lo cual deja sin cobertura a varios pueblos y ciudades del territorio colombiano.

- Precios: Al ser un servicio de transporte de pasajeros se basa en unas tarifas adecuadas para este tipo de servicio y no el de transporte de mercancías por lo cual los precios tienen a ser más elevados a la hora de ir a distintas ciudades en comparación a empresas prestadoras de este servicio.

Por estas razones AppShipment busca brindar una solución diseñada específicamente para este tipo de mercado lo que le brindara al usuario mayor cobertura y un margen de precios más ajustados al tipo de mercado.

### **Modelo de transporte actual**

Las empresas que actualmente manejan el transporte de mercancías emplean un modelo estricto a la hora de recoger, enviar y entregar mercancías, ya que esto debe estar sujeto a los procesos de optimización en cada una de las etapas para que se obtengan las máximas ganancias a los menores costos [1]. Sin embargo, este modelo no permite que otras personas que cuentan con recursos para prestar el servicio entren a las empresas fácilmente y mucho menos están diseñados para que estas personas presten el servicio con flexibilidad de tiempos y horarios.

Lo que se busca con este trabajo de grado no es modificar ni mejorar estos procesos, sino permitirles a las personas que cuentan con los recursos entrar a un sistema diseñado para prestar este servicio con flexibilidades de horarios y obtener ganancias directas por este servicio.

### III – ANÁLISIS

Esta sección está compuesta por el análisis de las variables relacionadas de AppShipment, así mismo, los requerimientos planteados a nivel general y el plan de pruebas propuesto para el prototipo final. Por otro lado, se especifican las restricciones de la solución propuesta que limitan, pero no obstaculizan, el alcance propuesto.

#### 3.1 Análisis de la competencia

El analizar a la posible competencia del producto resultante de este trabajo de grado, AppShipment, es un paso previo crucial para establecer el posible factor de éxito como aplicación para suplir las necesidades de los usuarios. Es importante resaltar que el fin de esta sección es en gran medida:

- Saber dónde y con quien se está compitiendo realmente
- Identificar posibles oportunidades de negocio
- Detectar elementos diferenciadores

	AppShipment	<u>mensajerosurbanos.com</u>	Tappsi - Mensajería	Uber - Rush
<b>Reputación General</b>	No es conocido	Medianamente conocido	Altamente conocido	Altamente conocido
<b>Modo de transporte</b>	Multivehicular ( Automóvil,	Moto	Taxis	Automóviles

	Camión, Moto)			
<b>Precio</b>	Definido para trabajos futuros	De acuerdo a la distancia establecida.  Paquetes empresariales: <ul style="list-style-type: none"> <li>● MiniUrbanos (400.000 a 600.000 COP)</li> <li>● Semiurbanos (800.000 a 1'000.000 COP)</li> <li>● UltraUrbanos (1'200.000 a 1'500.000 COP)</li> </ul>	De acuerdo a la distancia recorrida, precio en base al taxímetro y costo de envío(10000 )	De acuerdo a las tarifas establecidas por Uber en referencia a la distancia recorrida

*Tabla 1 Análisis de la Competencia*

En base a la tabla 1, se concluye que AppShipment como factor diferenciador posee un modo de transporte multi vehicular. Esto permite que cualquier persona natural con disposición de su vehículo, pueda acceder a los servicios que ofrece AppShipment, y obtener los beneficios establecidos por la misma.

Es importante resaltar que AppShipment, poseerá ventajas a nivel técnico. El hecho de que la aplicación sea multiplataforma, es decir, accesible desde dispositivos Android y aplicaciones Web. Esto ayuda al usuario a tener disponibilidad del servicio en cualquier lugar que se encuentre, siempre y cuando, posea la propiedad de ser geolocalizado.

### **3.1.1 Lista de Criterios de Mercado**

Durante el desarrollo de proyecto, junto con el levantamiento de requerimientos, se plantea una lista de criterios a comparar con las entidades descritas en la **Sección 3.1**, con el motivo de formular aspectos en cuanto a calidad interna y externa, calidad en uso, de las correspondientes aplicaciones con el fin de identificar el factor diferenciador y de éxito de AppShipment frente a su competencia.

En el anexo **Matriz de Criterios** se detalla qué criterios son relevantes a comparar y cómo logran identificar lo descrito anteriormente.

### **3.2 Análisis de Herramientas**

El desarrollo multiplataforma hoy en día se ve sesgado por las múltiples tecnologías que surgen casi a diario. Lo que se busca realmente para el desarrollo del trabajo de grado a nivel tecnológico, son herramientas que faciliten plasmar y ejecutar los objetivos específicos planteados y que puedan materializarse en una aplicación multiplataforma.

En general, dado al conocimiento previo de los integrantes del trabajo de grado y experiencia en cuanto a desarrollo web y móvil, se plantean tecnologías que cumplen con los siguientes criterios de aceptación ante lo que AppShipment como trabajo de grado necesita a nivel tecnológico para facilitar la construcción e implementación del mismo:

- Permitir empaquetar aplicaciones web y desplegar a diferentes plataformas móviles.
- Lenguaje Back-End robusto que cuente con comunicación simple hacia las tecnologías Front-End
- Herramientas para creación de pruebas unitarias orientadas al marco ágil de **TDD**
- Motor de base de datos **relacional**
- Desarrollo colaborativo y versionamiento de código
- IDE de manejo simple, compatible con las herramientas escogidas
- Framework que brinde el manejo de transacciones, concurrencia y aspectos de seguridad informática y de la información.

Con base a lo descrito anteriormente, el equipo de desarrollo planteó las herramientas que logaron el alcance establecido para AppShipment como trabajo de grado y aplicación.

Herramienta/Lenguaje	Descripción	Referencia
<b>Play Framework</b>	Framework para aplicaciones web escritas en Java y Scala, que integra componentes y las APIs necesarias para el desarrollo de aplicaciones web modernas.	<a href="https://www.playframework.com/documentation/2.5.x/Home">https://www.playframework.com/documentation/2.5.x/Home</a>
<b>Java</b>	Lenguaje de programación Orientado a Objetos	<a href="https://www.java.com/es/">https://www.java.com/es/</a>



<b>Angular 2</b>	Framework MVC para el desarrollo de páginas web dinámicas y consumo de servicios REST asincrónicamente.	<a href="https://angular.io/">https://angular.io/</a>
<b>ionic 2</b>	Framework que de la mano de Angular 2 permite crear aplicaciones híbridas multiplataforma respetando la guía de estilo de cada plataforma (Android, IOS y Windows).	<a href="http://ionic.io/">http://ionic.io/</a>
<b>Jasmine - Node.js</b>	Componente de node.js para realizar pruebas basadas en TDD (Test Driven Development)	<a href="https://jasmine.github.io/2.0/node.html">https://jasmine.github.io/2.0/node.html</a>
<b>IntelliJ - IDE</b>	IDE de programación robusto con soporte para las herramientas escogidas	<a href="https://www.jetbrains.com/idea/">https://www.jetbrains.com/idea/</a>
<b>Git</b>	Sistema de control de versiones	<a href="https://git-scm.com/">https://git-scm.com/</a>
<b>Github</b>	Plataforma para desarrollo colaborativo basado en git	<a href="https://github.com/">https://github.com/</a>

*Tabla 2 Análisis de Herramientas*

En el transcurso de desarrollo del proyecto, el equipo nota que la curva de aprendizaje para AngularJS, crece moderadamente debido a las diferencias existentes entre la versión anterior de Angular junto con la actual. Debido a esto, se toma la decisión de ajustar el calendario para incluir actividades de aprendizaje del framework y así continuar con el flujo normal del proyecto.

### 3.2.1 Análisis de herramientas alternativas y justificación

Se evalúa la posibilidad de varias herramientas y lenguajes ya especificadas en la sección anterior, las diferentes alternativas evaluadas que fueron descartadas por los integrantes del trabajo de grado, con el fin de justificar la escogencia de las herramientas trabajadas y el porqué las siguientes herramientas planteadas no cumplían con los criterios listados en la sección anterior.

Herramienta/Lenguaje	Descripción	Razón de descarte	Referencia
<b>Ruby</b>	Lenguaje de programación interpretado y orientado a objetos	Curva de aprendizaje alta dado al no conocimiento de los integrantes	<a href="https://www.ruby-lang.org/es/">https://www.ruby-lang.org/es/</a>
<b>Ruby On Rails</b>	Framework basado en el lenguaje de programación Ruby, enfocado a desarrollo web basado en MVC	Curva de aprendizaje alta dado al no conocimiento de los integrantes	<a href="http://rubyonrails.org/">rubyonrails.org/</a>

<b>Vue.js</b>	Framework progresivo para desarrollo de interfaces de usuario	No cumple con las expectativas requeridas para el desarrollo de Front-End	<a href="https://vuejs.org/">https://vuejs.org/</a>
<b>Phonegap</b>	Framework para desarrollo de aplicaciones móviles	Curva de aprendizaje extensa debido al no conocimiento de los integrantes	<a href="http://phonegap.com/">http://phonegap.com/</a>

*Tabla 3 Herramientas alternativas*

Las herramientas descritas anteriormente, son descartadas debido al poco conocimiento por parte de los integrantes de las tecnologías, basándose en la curva de aprendizaje por herramienta se estima que puede generar un impacto grave respecto a la calendarización del proyecto. No obstante, el equipo de desarrollo llega al acuerdo de que el lenguaje respectivo al Back-End será Java (Scala), debido al conocimiento adquirido durante la trayectoria universitaria la cual es común y robusta.

### **3.3 Especificación de Requerimientos**

Los requerimientos que se describen son el resultado del análisis y consideraciones basados en el tipo de usuario final, el alcance propuesto y lo que AppShipment como aplicación multiplataforma debe aportar a nivel tecnológico. Las funciones y acciones que desempeña el usuario dentro de la aplicación deberían poder realizarse de manera rápida y sencilla, lo cual debe verse reflejado con base a los requerimientos levantados. Así mismo, las restricciones especificadas en el documento SRS debe reflejarse en el prototipo final que está basado en dichos requerimientos.

### 3.3.1 En relación a las condiciones de desarrollo

Se espera que AppShipment como aplicación multiplataforma, sea una alternativa a considerar por los usuarios a la hora de realizar envíos, es por eso que la herramienta final es capaz de no solo de agilizar dicho proceso, sino también de crear un ambiente fluido y con una excelente experiencia de usuario gracias a las tecnologías escogidas.

Es por esto, que AppShipment se enfoca directamente en el ámbito de plataformas móvil y web, factor crítico y de aceptación en la comunidad tecnológica en la que nos encontramos actualmente. Es por eso que las tecnologías escogidas ayudan a lograr dicho enfoque; Ionic 2 permitirá la creación del aplicativo y poder desplegarlo en varios sistemas operativos, que, para un trabajo futuro, será una ventaja.

Por otro lado, Scala(Java) permite la construcción de las principales funcionalidades de AppShipment de manera robusta y escalable, que junto con el motor de bases de datos Oracle 12c, contribuirán al correcto desempeño y eficiencia de AppShipment.

A nivel visual y experiencia de usuario, el framework de Angular 2 junto con sus complementos(API Google y Ionic 2) permitirán desplegar por pantalla, los principales componentes de AppShipment de manera agradable tanto visualmente como desempeño.

Se hace uso de Amazon Web Services (AWS) como herramienta de despliegue de aplicaciones mediante servicios de Cloud Computing, así mismo, almacenamiento de bases de datos y un posible escalamiento a futuro [14], contribuyendo en el desarrollo de requerimientos de alto nivel o no funcionales. Permite el acceso hacia la aplicación por parte de los usuarios alojando a AppShipment en sus servidores.

### **3.3.2 En relación al usuario**

Una persona que busque hacer uso de AppShipment con fines de agilizar procesos de envío de encomiendas, o así mismo, buscar una remuneración económica con los bienes que posee y exponerlos a servicio de la comunidad reunida por AppShipment, debe ser capaz de comprender y utilizar de manera correcta las funcionalidades expuestas por la aplicación.

Desde el punto de vista de un usuario de AppShipment, el flujo de la aplicación debe ser entendible y sencillo, que cumpla con los objetivos que se proponga el usuario y tome a AppShipment como una real alternativa al envío de encomiendas y de esta manera ser capaz tanto de finalizar exitosamente un servicio dentro de la aplicación como recomendar la misma a otros potenciales usuarios.

### **3.3.3 En relación a la aplicación**

Es indispensable que AppShipment cumpla con lo especificado en los requerimientos de software, para definir y construir de manera correcta lo que se espera como comportamiento de la aplicación.

Por lo tanto, teniendo en cuenta los objetivos tanto generales como específicos que establecen el alcance del trabajo de grado, se espera que para lograr dicho alcance, AppShipment debería ser capaz de:

- Realizar un correcto manejo de sesiones.
- Desempeñar tareas de Geolocalización como ubicación de usuarios, cálculo de trayectos y filtrado de transportadores con base a distancias.
- Solicitar y finalizar un servicio de transporte de encomiendas.
- Generar pines de seguridad para finalización de un servicio.
- Realizar tracking de un paquete

AppShipment brinda una manera alternativa de resolver dichas necesidades, sin embargo, el usuario debe seguir tanto la estructura definida por la aplicación como la responsabilidad exigida en los términos y condiciones propuestos en un trabajo futuro.

### 3.3.4 Lista de Requerimientos

Fueron identificados 47 requerimientos los cuales satisfacen el comportamiento descrito en las secciones anteriores que deberá adoptar AppShipment.

Código del Requerimiento	Especificación del Requerimiento	Tipo
R001	El sistema debe contar con manejo de sesiones	Funcional
R002	El sistema debe contar con un registro de usuarios	Funcional
R003	El sistema debe permitir al usuario leer Términos y condiciones de uso de la aplicación	Funcional
R004	El sistema debe contar con geolocalización	Funcional
R005	El sistema debe permitir al contratante escoger un tipo de Vehículo	Funcional
R006	El sistema debe contar con una sección que permita ver información de noticias a fines de la aplicación	Funcional

R007	El sistema debe contar con servicio de calificación de usuario	Funcional
R008	El sistema debe contar con la función de solicitar servicio	Funcional
R009	El sistema debe contar con el ingreso de la dirección de destino	Funcional
R010	El sistema debe contar con el ingreso del nombre del destinatario	Funcional
R011	El sistema debe contar con el ingreso de la declaración de contenido correspondiente al paquete	Funcional
R012	El sistema debe permitir al contratante elegir el tamaño del paquete	Funcional
R013	El sistema debe permitir al contratante filtrar la búsqueda de servicio por localización	Funcional
R014	El sistema debe permitir al contratante elegir asegurar su paquete	Funcional
R015	El sistema debe contar con un radio de búsqueda para seleccionar al transportador prestador del servicio	Funcional
R016	El sistema debe tener configuraciones basadas	Funcional

	en tiempo	
R017	El sistema debe aumentar el radio de búsqueda	Funcional
R018	El sistema debe permitir al transportador rechazar una solicitud de servicio	Funcional
R019	El sistema debe permitir ingresar el valor declarado del paquete	Funcional
R020	El sistema de permitir al transportador completar servicios	Funcional
R021	El sistema debe permitir al transportador aceptar una solicitud de servicio	Funcional
R022	El sistema debe contar con alertas de emergencia	Funcional
R023	El sistema debe medir la aceleración del vehículo de un transportador	Funcional
R024	El sistema debe permitir al contratante conocer la información del perfil del transportador	Funcional
R025	El sistema debe permitir al contratante cancelar un servicio	Funcional



R026	El sistema debe contar con el servicio de recomendación de usuarios	Funcional
R027	El sistema debe contar con el servicio de carga de archivos	Funcional
R028	El sistema debe especificar el tamaño máximo de un archivo a subir	Funcional
R029	El sistema debe generar un PIN aleatorio de seguridad	Funcional
R030	El sistema debe contar con validación de tarjetas de crédito	Funcional
R031	El sistema debe permitir al transportador ofrecer un servicio de transporte	Funcional
R032	El sistema debe contar con un servicio de tracking de paquetes	Funcional
R033	El sistema debe contar con cálculo aproximado de tiempo de servicio con base a la distancia a recorrer	Funcional
R034	El sistema debe permitir al transportador confirmar la entrega de la encomienda	Funcional
R035	El sistema debe utilizar la aplicación de llamadas del dispositivo	Funcional

R036	El sistema debe permitir cambiar el medio de pago	Funcional
R037	El sistema debe permitirle al transportador cancelar un servicio	Funcional
R038	El sistema debe permitir al contratante seleccionar servicios programados	Funcional
R039	El sistema debe permitir al transportador agendar un servicio programado	Funcional
R040	El sistema debe permitir al contratante aceptar un servicio programado	Funcional
R041	El sistema debe permitir al transportador aceptar una solicitud a un servicio programado	Funcional
R042	El sistema debe permitir al transportador cambiar a estado Activo	Funcional
R043	El sistema debe permitir al transportador cambiar a estado Inactivo	Funcional
R044	El sistema debe enviar solicitudes de servicio sólo si el transportador está activo	Funcional
R045	El sistema debe notificar al contratante si su encomienda sufrió una alteración	Funcional

R046	El sistema debe alertar al transportador si existe un cambio significativo en la aceleración de su vehículo	Funcional
R047	El sistema debe contar con un mapa	Funcional
R49	El sistema debe mantener la disponibilidad a pesar que el número de usuarios aumente notablemente.	Confiabilidad
R50	El sistema debe estar integrado con una herramienta que permita la entrega e integración continua	Operación

*Tabla 4 Lista de Requerimientos*

### 3.3.5 Priorización de Requerimientos

Para la correspondiente priorización de requerimientos, el equipo de desarrollo plantea una matriz donde se refleje dicha priorización basada en cuatro (4) criterios correspondientes a:

- **Importancia:** Describe qué tan importante es el requerimiento dentro de la aplicación, que tan indispensable es dentro de la misma.
- **Costo/Tiempo:** Describe el valor asociado al costo aproximado de implementación y al tiempo que incurra desarrollarlo.
- **Riesgo:** Describe el valor asociado a que tan riesgoso es no implementar dicho requerimiento, así mismo, qué riesgo representa que el requerimiento falle o no esté disponible dentro de la aplicación.

- Volatilidad/Variabilidad: Describe el valor asociado a que tan cambiante puede ser un requerimiento en un futuro, dado que un cambio significativo a un requerimiento puede implicar a la consumación de los riesgos identificados en la propuesta de grado, así mismo, la calendarización establecida y el core de AppShipment como aplicación.

Detallado así, se procede a asignar valores a cada uno de los criterios mencionados anteriormente, basándose en la experticia tanto técnica como ingenieril de cada uno de los integrantes del equipo de desarrollo. Dicho valor, se encuentra en el rango de uno (1) a diez (10), siendo 1 identificado como **Baja importancia** y 10 como **Alta importancia**. Por lo tanto, una vez ingresados los valores se logra identificar el cálculo de priorización basándose en la asignación de porcentajes a cada uno de los criterios de la siguiente manera:

- Importancia - 30%
- Costo/Tiempo - 30%
- Riesgo - 30%
- Volatilidad/Variabilidad - 10%

Posteriormente, se logran identificar los requerimientos que poseen el mayor valor de priorización para continuar con su implementación junto con sus requerimientos asociados y así, lograr el alcance especificado dentro del trabajo de grado. Sin embargo, existen requerimientos los cuales serán implementados y tratados para trabajos futuros, justificando su importancia dentro de la aplicación, aunque para el ámbito de trabajo de grado no sea susceptible o alcanzable en el tiempo establecido.

Los requerimientos implementados se muestran en la siguiente tabla junto con

el valor de priorización correspondiente, para mayor detalle véase el anexo

### Especificación de requerimientos y Casos de Uso.

<b>Id Req.</b>	<b>Importancia</b>	<b>Importancia (ajustada)</b>	<b>Costo</b>	<b>Costo (ajustado)</b>	<b>Riesgo</b>	<b>Riesgo (ajustado)</b>	<b>Volatilidad</b>	<b>Volatilidad (ajustada)</b>	<b>Prioridad</b>
<b>R47</b>	10	3	9	2,7	8	2,4	5	1,5	9,6
<b>R45</b>	10	3	7	2,1	7	2,1	6	1,8	9
<b>R46</b>	10	3	7	2,1	7	2,1	6	1,8	9
<b>R13</b>	10	3	8	2,4	6	1,8	5	1,5	8,7
<b>R32</b>	10	3	10	3	6	1,8	2	0,6	8,4
<b>R22</b>	8	2,4	8	2,4	5	1,5	6	1,8	8,1
<b>R15</b>	10	3	8	2,4	6	1,8	1	0,3	7,5
<b>R04</b>	10	3	9	2,7	5	1,5	1	0,3	7,5
<b>R19</b>	8	2,4	6	1,8	5	1,5	6	1,8	7,5
<b>R01</b>	10	3	8	2,4	5	1,5	5	0,5	7,4
<b>R08</b>	10	3	5	1,5	4	1,2	5	1,5	7,2
<b>R02</b>	8	2,4	7	2,1	5	1,5	5	1	7
<b>R20</b>	9	2,7	6	1,8	5	1,5	3	0,9	6,9
<b>R07</b>	9	2,7	7	2,1	6	1,8	1	0,3	6,9
<b>R34</b>	10	3	5	1,5	4	1,2	3	0,9	6,6
<b>R21</b>	8	2,4	5	1,5	5	1,5	3	0,9	6,3
<b>R42</b>	9	2,7	4	1,2	4	1,2	4	1,2	6,3
<b>R43</b>	9	2,7	4	1,2	4	1,2	4	1,2	6,3
<b>R44</b>	8	2,4	6	1,8	4	1,2	1	0,3	5,7
<b>R14</b>	10	3	3	0,9	3	0,9	3	0,9	5,7

<b>R17</b>	7	2,1	5	1,5	3	0,9	3	0,9	5,4
<b>R11</b>	10	3	3	0,9	2	0,6	2	0,6	5,1
<b>R09</b>	10	3	2	0,6	2	0,6	2	0,6	4,8
<b>R12</b>	10	3	2	0,6	2	0,6	1	0,3	4,5
<b>R10</b>	8	2,4	1	0,3	1	0,3	1	0,3	3,3

*Tabla 5 Requerimientos Implementados*

### 3.4 Suposiciones y dependencias

Para el desarrollo de AppShipment a nivel de producto, el equipo de desarrollo debe tener en cuenta las siguientes Suposiciones y dependencias, con ánimo de establecer con los usuarios y Stakeholders las condiciones en las cuales AppShipment como producto y el equipo de desarrollo asumirá y así mismo liberarse en cuanto a consecuencias y responsabilidades de lo que se establezca en esta sección. Esto estará estrechamente relacionado con las pre condiciones y Términos de uso de la aplicación.

#### 3.4.1 Suposiciones

- Se asume que AppShipment será utilizado por actores implicados que conocen el funcionamiento de este tipo de aplicaciones, por lo que, a la hora de finalizar la aplicación, no se brindará capacitación alguna, solo se pretende brindar una guía acerca de la aplicación dentro de la misma.
- Se asume que AppShipment debe de funcionar de manera similar a aplicaciones de este tipo (Mapas, Economías Colaborativas, Tracking de paquetes), esto para que el usuario pueda familiarizarse de manera rápida con la aplicación, esto dentro del propósito y alcance para el trabajo de grado.

- Se asume que el cliente en este caso especial, el equipo de desarrollo, define el tamaño, alcance y funcionalidad de AppShipment, dado así, no habrá cambios en los requerimientos de la aplicación posterior a su entrega.
- Se asume la dependencia en la estabilidad de servidores (Web y Base de datos), para que el usuario obtenga una mejor experiencia de usuario.
- Se supone que el usuario tiene manejo pleno del dispositivo donde esté accediendo a AppShipment, esto para lograr una máxima experiencia con la aplicación.
- Se asume que el usuario usará a AppShipment con motivos únicamente correctos, el mal uso de la aplicación, fraudes y todo tipo de acción fraudulenta no concierne en ningún aspecto al equipo de desarrollo ni a AppShipment como aplicativo.
- El usuario final de AppShipment debe asumir toda la responsabilidad al hacer uso de la aplicación.

### **3.4.2 Dependencias**

- AppShipment depende en totalidad con que el usuario cuente con un dispositivo de gama media/alta (para el propósito de pruebas de software), así mismo como datos de navegación suficientes.
- AppShipment centra el core de negocio en la localización y uso de mapas, por lo tanto, la disponibilidad del GPS debe ser obligatoria para un correcto flujo de la aplicación.

## **IV – DISEÑO**

En esta sección se detalla el diseño de la arquitectura que representa la solución propuesta para el desarrollo de AppShipment como aplicación multiplataforma. Las

herramientas que finalmente fueron base para el cumplimiento de los requerimientos funcionales especificados (Ver Anexo – Software Requirements Specification), así mismo como se modelo la base de datos que persistirá los registros y transacciones que se realicen dentro de la aplicación.

#### **4.1 Persistencia de Datos**

En esta sección se detalla el funcionamiento de la capa de persistencia de datos, esta capa se encarga de acceder a una fuente de datos o base de datos, en este caso para la persistencia de los registros, queries y transacciones de AppShipment. El motor de base de datos implementado durante el desarrollo es Oracle 12.c, el cual permite la estandarización de tareas, consolidación y automatización de transacciones entre el aplicativo y la base de datos.

El propósito de uso de este motor es el de Realizar y facilitar todo el proceso de implementación, gestión, y uso de las bases de datos necesarias para AppShipment.

Con motivo de que el equipo de desarrollo y el lector entiendan como se almacenarán los datos, se modelo la base de datos usando un modelo entidad relación (MER), el cual representa las entidades relevantes para la persistencia de datos de AppShipment y permite el correcto diseño y normalización de la base de datos a usar por la aplicación, la cual asegura el correcto funcionamiento transaccional y de persistencia.

A continuación, se presenta el Modelo Entidad Relación (MER) de AppShipment:



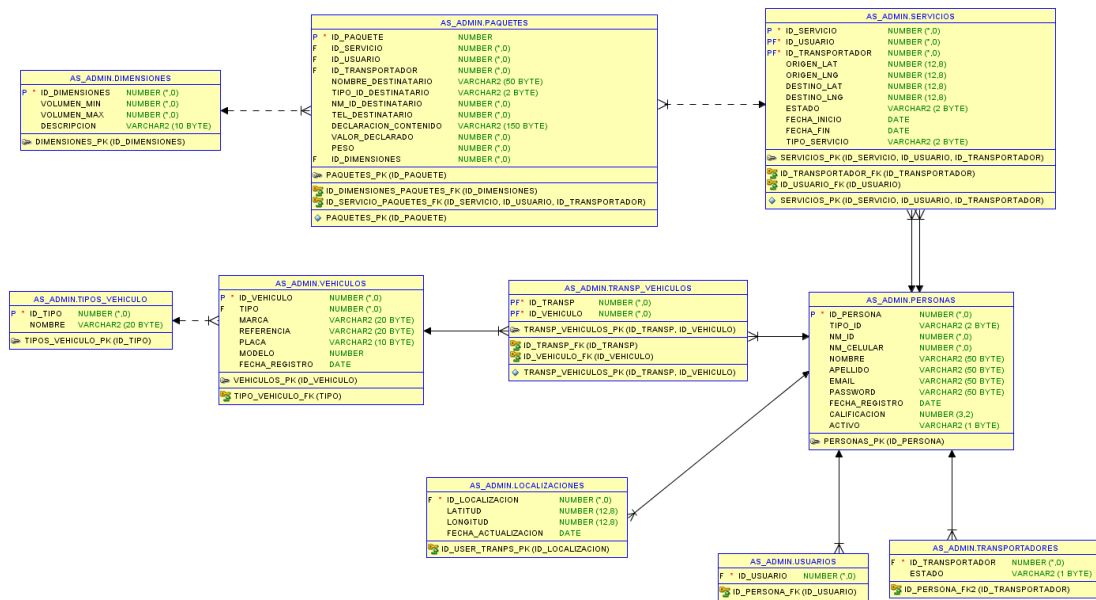


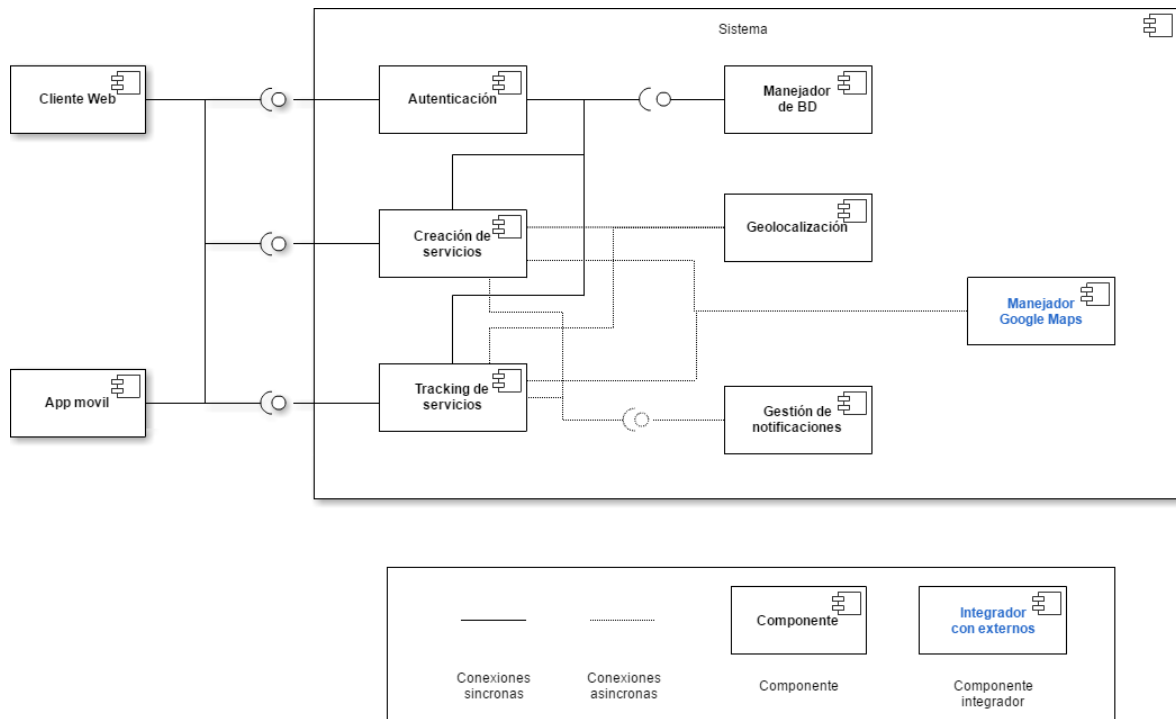
Ilustración 1 MER AppShipment

Así mismo el equipo de desarrollo, describe con la siguiente tabla el detalle de cada una de las entidades descritas en el Modelo Entidad Relación, con el fundamento de justificar el por qué la existencia de cada entidad, atributo y relación, esto puede detallarse en el anexo **Software Design Description , Sección 7.2.**

#### 4.2 Vista de desarrollo del Sistema

A continuación, se presentan los componentes presentes en AppShipment como aplicación y sus respectivas dependencias, estos componentes representan las funcionalidades comprendidas en los Casos de uso (Ver Anexo **Especificación de Requerimientos y Casos de Uso**).

- Cliente Web y App Móvil: componentes encargados de presentar la interfaz de usuario a los usuarios contratantes y transportadores.
- Autenticación: se encarga de validar credenciales de acceso a la aplicación.
- Creación de Servicios: se encarga de obtener los datos necesarios del paquete y destinatario para crear el servicio de envío de encomiendas, así como la asignación del transportador bajo filtros de cercanía.
- Tracking de Servicios: componente encargado de realizar el seguimiento del paquete y de presentar las notificaciones de alerta durante la ejecución del servicio.
- Manejador Google Maps: componente encargado del consumo de servicios asíncronos al API de Google Maps V3.
- Gestión de Notificaciones: componente encargado del envío de notificaciones al usuario contratante y transportador.
- Geolocalización: encargado de la obtención de coordenadas del usuario contratante y transportador.
- Manejador de Bases de Datos: este componente realiza la conexión con la base de datos.



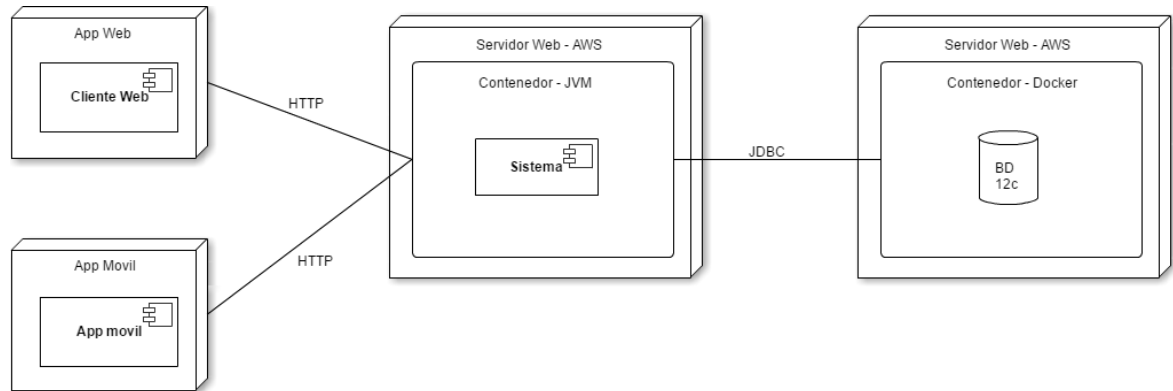
*Ilustración 2 Vista de desarrollo del Sistema*

### 4.3 Vista Física del Sistema

Para AppShipment como aplicación se optó por una arquitectura de tipo Cliente/Servidor de tres niveles, la lógica de negocio se encuentra desplegada en la instancia de un servidor AWS el cual corre una Java Virtual Machine JVM 8. En cuanto a la persistencia de datos, se encuentra desplegado en otra instancia de AWS en la cual se ejecuta un contenedor Docker que contiene una base de datos Oracle 12c.

Las conexiones de los clientes web y móvil al servidor de AWS donde se aloja la lógica de negocio, se manejan a través del protocolo HTTP. En cuanto a las conexiones del sistema hacia la base de datos se hace uso de JDBC como API de

consultas y transacciones.



*Ilustración 3 Vista Física del Sistema*

La decisión arquitectural de un diseño de 3 niveles se tomó a partir de los siguientes criterios:

1. **Multiplataforma:** ya que la aplicación será desplegada en diferentes plataformas, es necesario que el cliente tenga acceso únicamente a la interfaz gráfica (GUI), así en posteriores desarrollos y lanzamiento de versiones estos no se vean afectados directamente en los cambios y tengan la necesidad de actualizar sus aplicaciones. Para los aplicativos móviles será necesario que en los cambios gráficos se descarguen nuevas versiones de las tiendas (App Store, Google Play, etc.)
2. **Escalabilidad:** Se espera que la aplicación crezca rápidamente en cantidad de usuarios, esta arquitectura permite ampliar fácilmente los nodos del backend, adicionando balanceadores de carga para mantener los requerimientos de disponibilidad (Ver Anexo – Software Requirements Specification), sin

afectar la aplicación de los usuarios. Además, los nodos serán alojados en servidores de Amazon Web Services, los cuales permiten realizar la escalabilidad horizontal de una manera simple y permite integrarse con aplicaciones de integración y entrega continua como lo es Jenkins, adicionalmente en etapa de desarrollo se cuenta con la capa gratuita de AWS, lo que reduce los costos de la aplicación.

## **V – DESARROLLO DE LA SOLUCIÓN**

En esta sección se detalla la metodología de desarrollo e implementación que fundamenta la solución propuesta por el Trabajo de Grado. Así mismo, se incluye a detalle las funcionalidades implementadas en el prototipo final de AppShipment junto con la muestra de pantallas como apoyo visual al entendimiento de dichas funcionalidades.

### **5.1 Medición de cambios repentinos en el acelerómetro**

En base a los requerimientos relacionados con alertas acordes a la sobre aceleración repentina del vehículo que transporta la encomienda, se implementa en AppShipment, un acelerómetro basado en el API de Córdova conocido como DeviceMotion con el fin de obtener la aceleración actual en los 3 ejes del sistema cartesiano y obtener el parámetro porcentual de la aceleración que condiciona el envío de alertas a los usuarios informando el posible comprometimiento del estado de la encomienda enviada.

Se plantean inicialmente 2 maneras para obtener dicho parámetro:

- Planteamiento Teórico basado en la sobre aceleración de un cuerpo.
- Planteamiento Práctico basado en prueba y error.

Se entiende como sobre aceleración a la tasa de deriva de la aceleración conocida también como sacudón o tirón. La sobre aceleración, dada por expresiones matemáticas está dada por la siguiente ecuación [15]:

$$\frac{d^3 x}{dt^3} = f\left(\frac{d^2 x}{dt^2}, \frac{dx}{dt}, x\right).$$

*Ilustración 4 Ecuación de Sobre Aceleración*

Definida como la segunda derivada de la velocidad con respecto al tiempo, permite hallar la magnitud de dicha sobre aceleración.

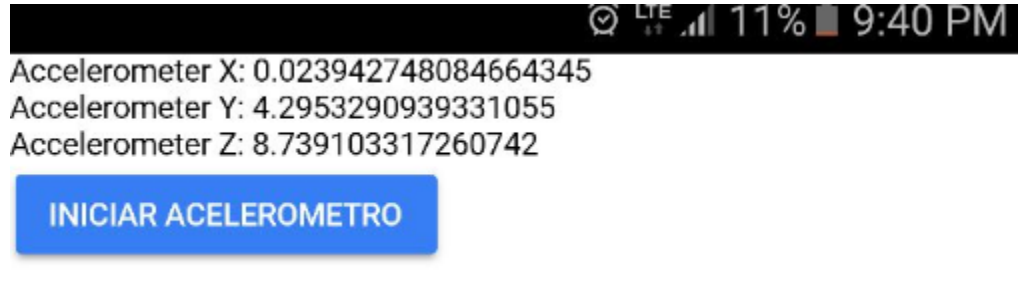
Conocida así la teoría se descarta la obtención de resultados de manera teórica debido al poco tiempo que se tiene para poder realizar una investigación exhaustiva y poder proceder con la implementación de los objetivos del trabajo de grado y lograr el alcance planteado. No obstante, el equipo de trabajo de grado opta por el planteamiento e implementación práctico para obtener un escenario que justifique el potencial trabajo futuro de esta funcionalidad basada en dicha teoría.

Posteriormente, se realizan unas primeras pruebas para analizar a manera de prueba y error, la aceleración basada en eventos específicos, creados para evaluar valores en cada uno de los ejes del plano cartesiano, teniendo en cuenta, que el dispositivo con el que se realiza la prueba está en posición horizontal (sobre una mesa, como referencia) :

- Reposo Total (Evaluar los 3 ejes)
- Lanzamiento del dispositivo desde 2 metros de altura, partiendo del reposo (Evaluar eje Y)
- Caminando en la calle (Evaluar eje X)

- Abordo de un Transmilenio (Evaluar Eje X y Z)

Se toma inicialmente los valores que arroja el acelerómetro en el dispositivo ilustrado así:



*Ilustración 5 Toma del acelerómetro implementado*

Se registran los atributos de interés tales como:

- Promedio de aceleración en el eje X
- Promedio de aceleración en el eje Y
- Promedio de aceleración en el eje Z
- Sobre aceleración (Posible tirón o sacudón)

Registrando así los primeros resultados, luego de 3 pruebas para cada escenario:

<b>Prueba</b>	<b>X Axis</b>	<b>Y Axis</b>	<b>Z Axis</b>	<b>Over Acceleration %</b>
	<b>Promedio</b>	<b>Promedio</b>	<b>Promedio</b>	<b>- 1 Second Later (After heavy</b>

				<b>acceleration Change)</b>
T-001 (Reposo)	- 0,0578885860 00	- 0,0981098266 21	9,8796577523 54	1,98823%
T-002 (Lanzamiento a 2 metros de altura)	- 0,6865061000 00	9,6464275610 00	1,4970794200 00	98,95756330036 %
T-003 (Caminando en calle)	0,0402251146 00	- 0,1463588020 00	9,7524412307 70	12,65843 %
T-004 (Abordo de Transmilenio)	0,5262977474 60	0,5000411060 00	9,9195771133 00	322,801239%

*Tabla 6 Tabulación de resultados del acelerómetro*

Finalmente se quiere deducir en un porcentaje en base a la aceleración luego de 1 segundo, aquella sobre aceleración que permita definir adecuadamente la palabra “brusco”. Aproximadamente, un valor estimado para dicho porcentaje **es de 95% en adelante** de la aceleración obtenida un segundo antes, indicando una desaceleración o aceleración de considerable magnitud en una unidad de tiempo bastante pequeña.

Este valor obtenido aplica para todo tipo de vehículo registrado en AppShipment, es decir, para automóvil, camión, moto o bicicleta. Dado a que la aceleración mide el cambio de velocidad por unidad de tiempo, el tipo de vehículo es indiferente para este



cálculo logrando así, establecer el valor para condicionar el envío de alertas al usuario.

## **5.2 Metodología del Desarrollo**

Al momento de iniciar a desarrollar AppShipment como aplicación, se tenía un estado relativamente avanzado en la parte investigativa, de planeación y contexto. AppShipment había sido definida tiempo anterior al inicio de la misma. Por lo tanto, la especulación de como AppShipment terminaría siendo como prototipo final, pasa a ser un factor de ayuda en cuanto a implementación.

Al iniciar el calendario de trabajo, el equipo de desarrollo decide iniciar el proceso de ingeniería de requerimientos los cuales abarcaban en su mayoría lo que se había establecido como alcance y así mismo con su respectiva priorización, se logra determinar el orden de implementación.

Previamente se definió la metodología como DAD (Disciplined Agile Delivery) el cual ayudo al equipo de desarrollo a implantar los valores y principios de las metodologías ágiles desde la definición de requerimientos, pasando por la implementación de las funcionalidades y hasta el despliegue de AppShipment para ser entregado como trabajo de grado. El equipo de desarrollo decide adoptar así mismo, el sistema de información Kanban como herramienta de ayuda para establecer:

- Cantidad de trabajo por iteración
- Asignación de tareas
- Control en tiempos necesarios
- Control de actividades

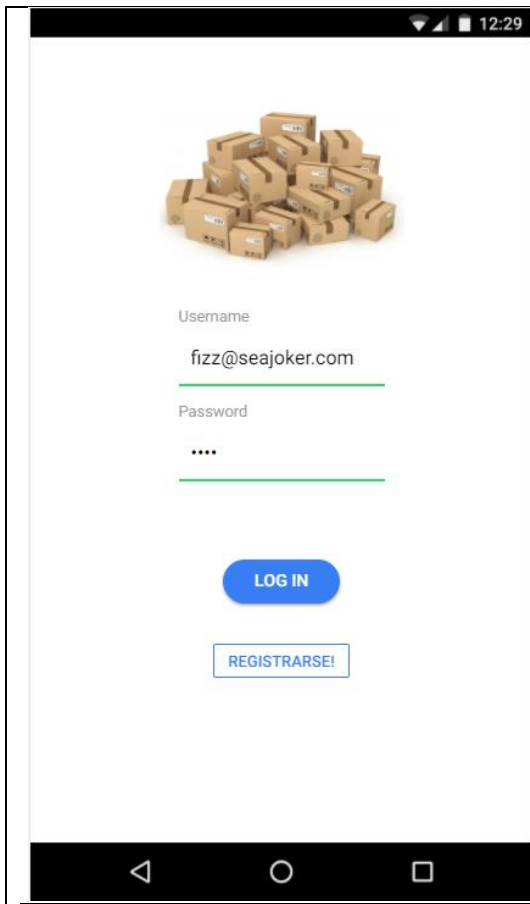
Cabe resaltar que cada iteración anteriormente mencionada correspondía a una semana del calendario en la cual se realizaba una constante retroalimentación tanto con el director de trabajo de grado como con los resultados obtenidos a lo largo de la iteración. Gracias a esto, se alcanza la tercera fase descrita en el comienzo de esta memoria, en donde se obtiene la aceptación por parte del equipo de desarrollo frente a cada requerimiento implementado. La manera en que AppShipment fue diseñada, requería implementar funcionalidades puntuales durante cada iteración y luego evaluar la correspondiente integración con los componentes ya realizados.

Sin embargo, debido a complicaciones con la curva de aprendizaje de las herramientas escogidas para el desarrollo de AppShipment, logra retrasarse la fase final descrita en la metodología dado que la implementación de los requerimientos asignados para las últimas iteraciones no permitían el avance esperado para la fase documental. No obstante, el equipo de desarrollo provee una recalendarización y priorización de los entregables descritos en esta memoria y finalmente se termina cada fase satisfactoriamente.

### **5.3 Funcionalidades del Sistema**

En esta sección se detalla mediante un árbol de navegación las diferentes vistas o pantallas de la interfaz gráfica con su respectiva relación (navegación) entre ellas. Esto con fin de que tanto los lectores como el desarrollador entiendan, usen y para el caso del desarrollador, implemente, lo que AppShipment como aplicación multiplataforma plantea como solución gráfica y funcional al proceso de envío de encomiendas basada en economías colaborativas, esto a su vez, realizando trazabilidad con los Casos de Uso que describen cada funcionalidad de AppShipment. Para mayor detalle en cuanto al árbol de navegación de AppShipment diríjase al anexo **Software Design Description sección 7.3**.

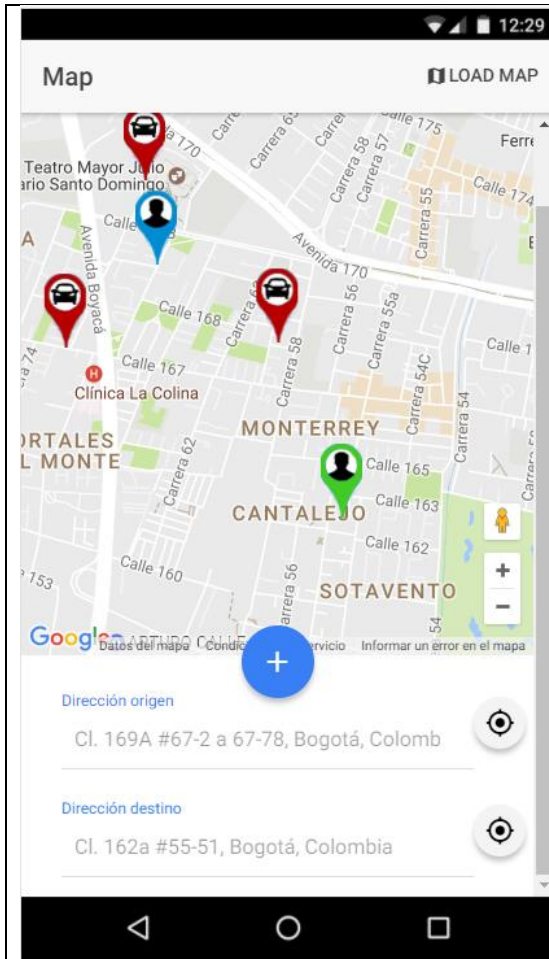
<b><u>PANTALLA APPSHIPMENT</u></b>	<b><u>DESCRIPCIÓN</u></b>
------------------------------------	---------------------------



*Ilustración 6 Login*

Esta pantalla muestra el manejo de sesiones de AppShipment, cada usuario posee credenciales de acceso a la aplicación con las cuales inicia sesión.

**Casos de uso Asociados: CU-001**



Pantalla referencia al home de AppShipment, se muestra la posición actual del usuario junto con los marcadores de los transportadores a su alrededor.

**Casos de uso Asociados:CU-002, CU-003, CU-006**

*Ilustración 7 Home*

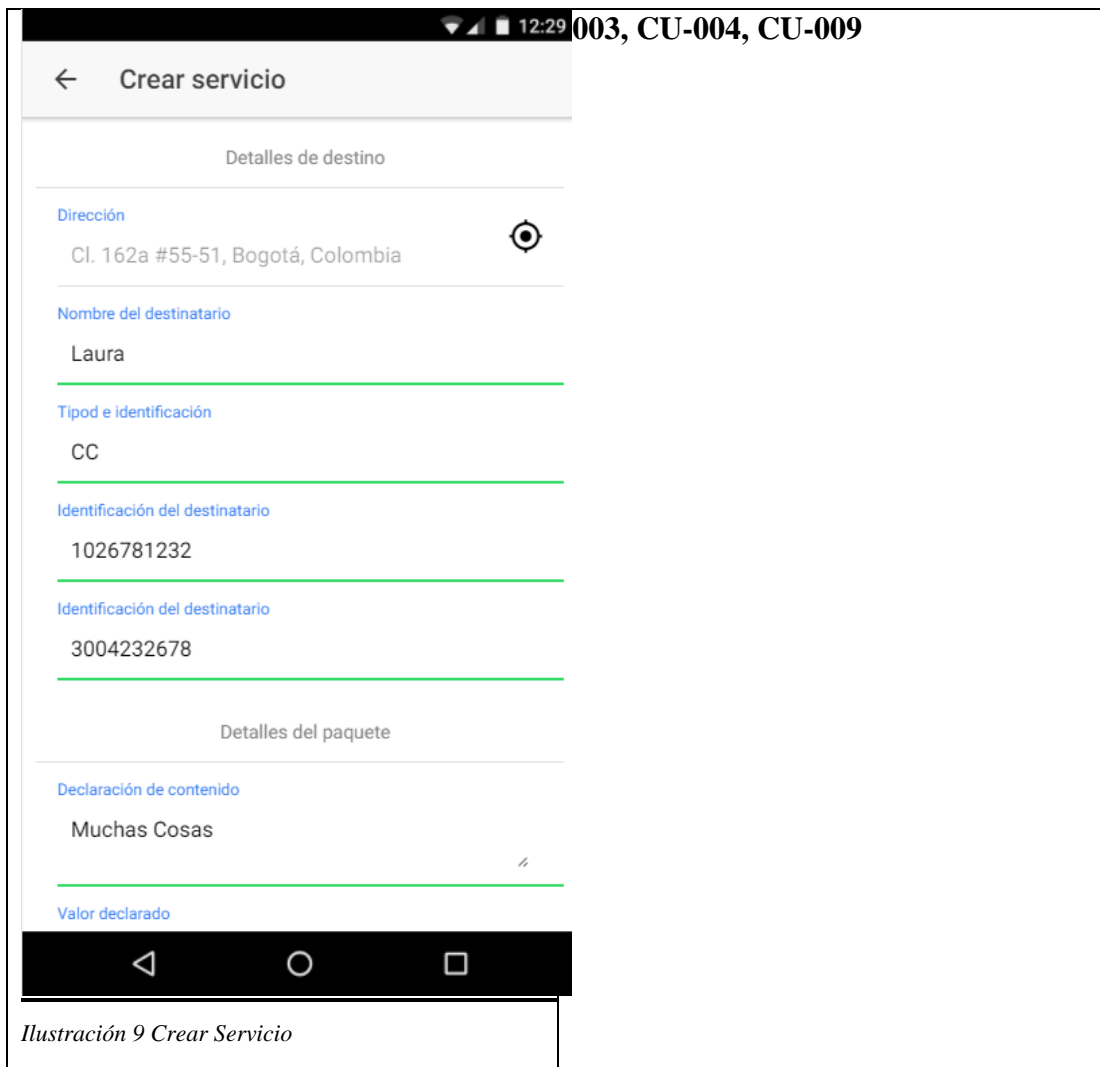
Formulario de registro de AppShipment. El usuario diligencia los atributos que se muestran por pantalla, para posteriormente realizar el registro.

**Casos de uso Asociados: CU-007**

*Ilustración 8 Registro*

El usuario ingresa a la opción de crear servicio. Ingresa los datos solicitados por el formulario para luego enviar la solicitud de servicio de envío de encomiendas.

**Casos de uso Asociados: CU-002, CU-**



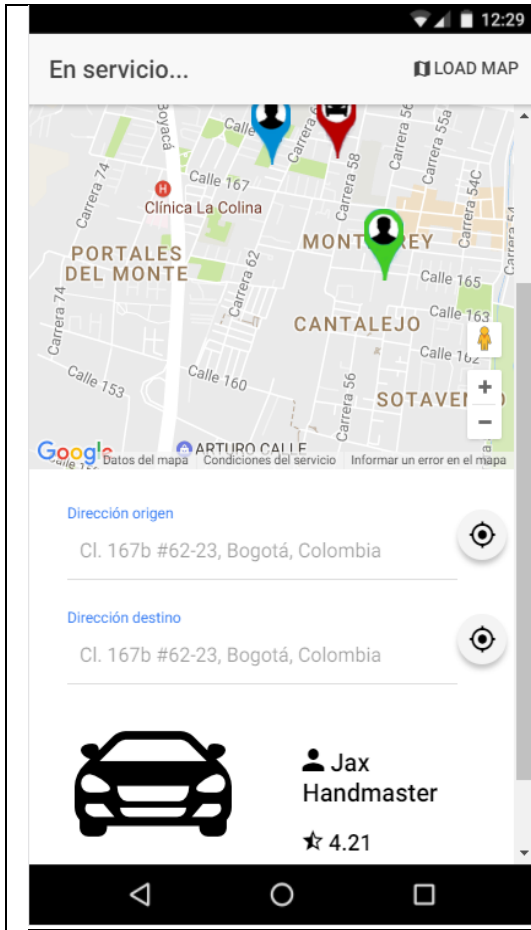


Ilustración 10 Iniciar servicio de envío de encomiendas

El usuario inicializa el servicio de envío de encomiendas. Se muestran los datos de perfil del transportador junto con la calificación del mismo.

**Casos de uso Asociados: CU-009, CU-008, CU-004, CU-003, CU-002**



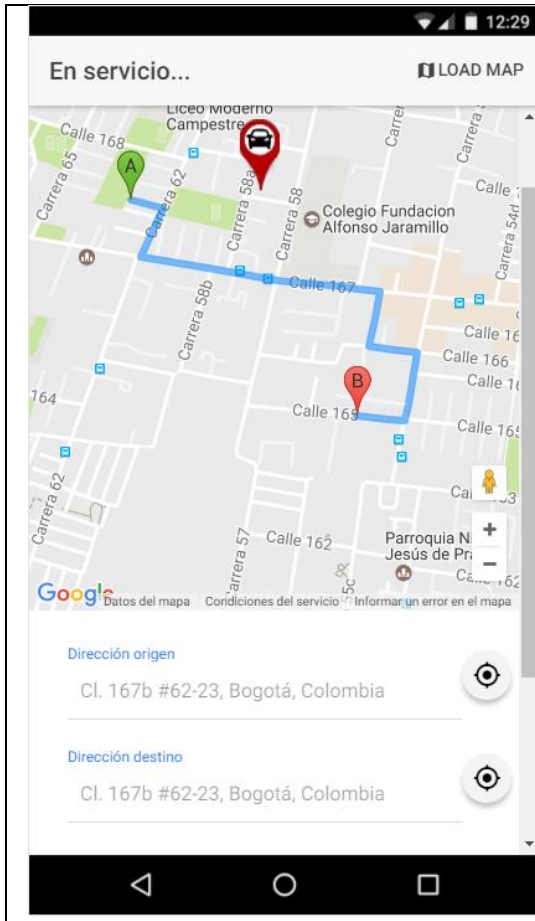


Ilustración 11 Tracking de Paquete

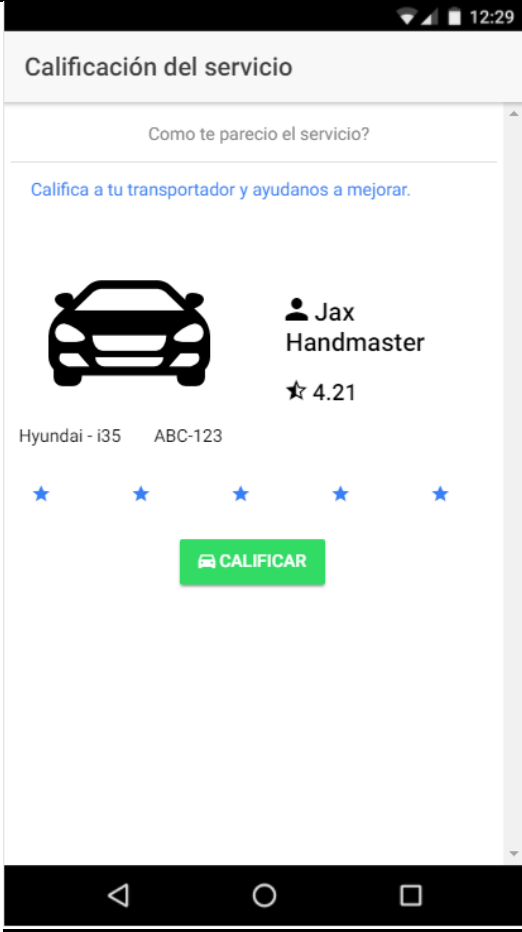
Una vez iniciado el servicio de envío de encomiendas, el sistema realiza un tracking o seguimiento del paquete acarreado por el transportador. Dicha posición es actualizada cada 10 segundos.

**Casos de uso Asociados:CU-004, CU-005**

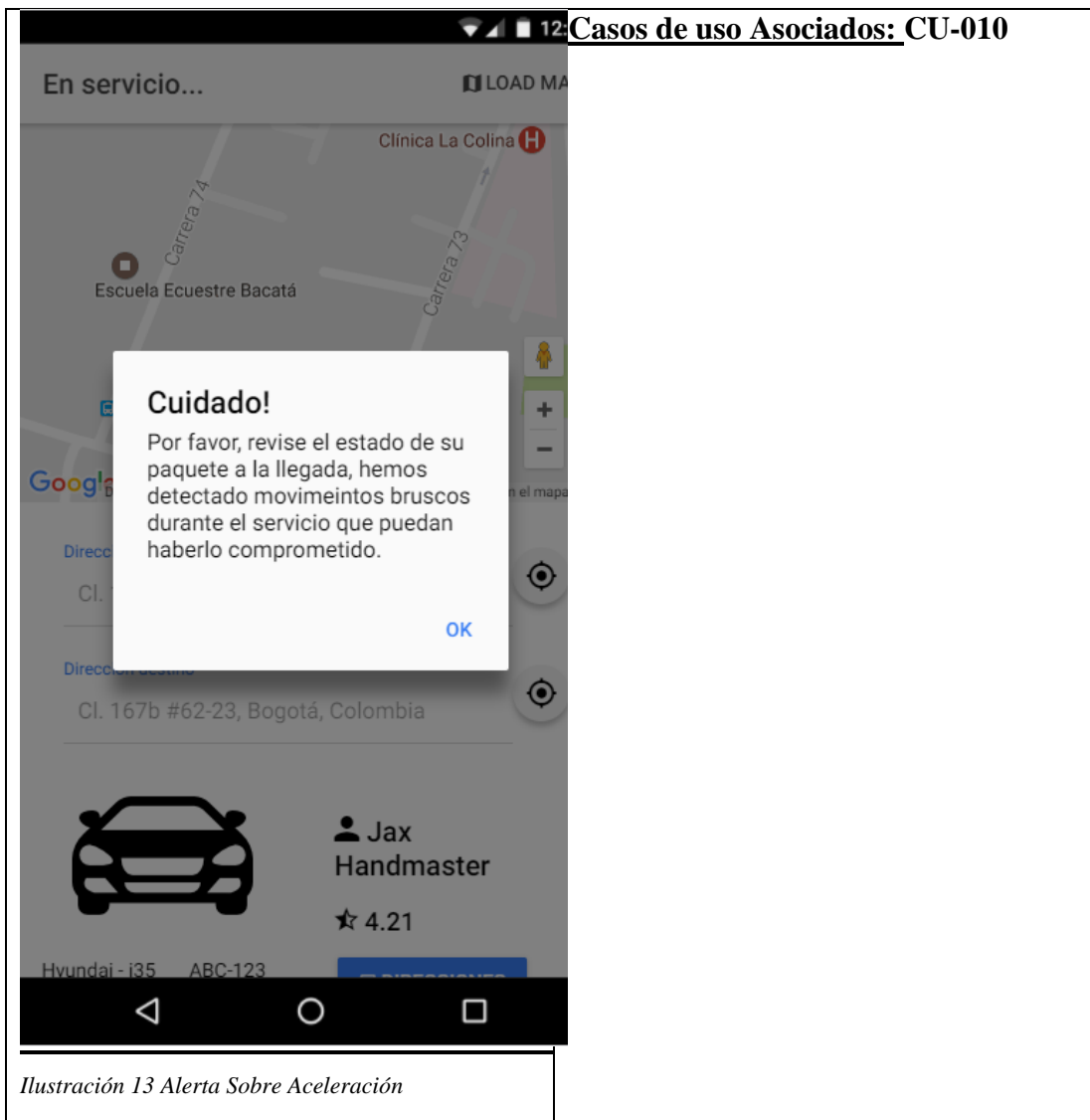
**(7)**

En esta pantalla el usuario es capaz de calificar a un transportador luego de haberse finalizado un servicio.

**Casos de uso Asociados: CU-008**

	
<p><i>Ilustración 12 Calificación de Transportador</i></p>	
	<p>El sistema envía una alerta a los usuarios correspondientes a un servicio, basado en la sobre aceleración del dispositivo móvil del transportador. Se muestra por pantalla una alerta indicando que el paquete pudo haberse visto comprometido luego de la sobre aceleración que sufrió el dispositivo.</p>

**Casos de uso Asociados: CU-010**



*Ilustración 13 Alerta Sobre Aceleración*

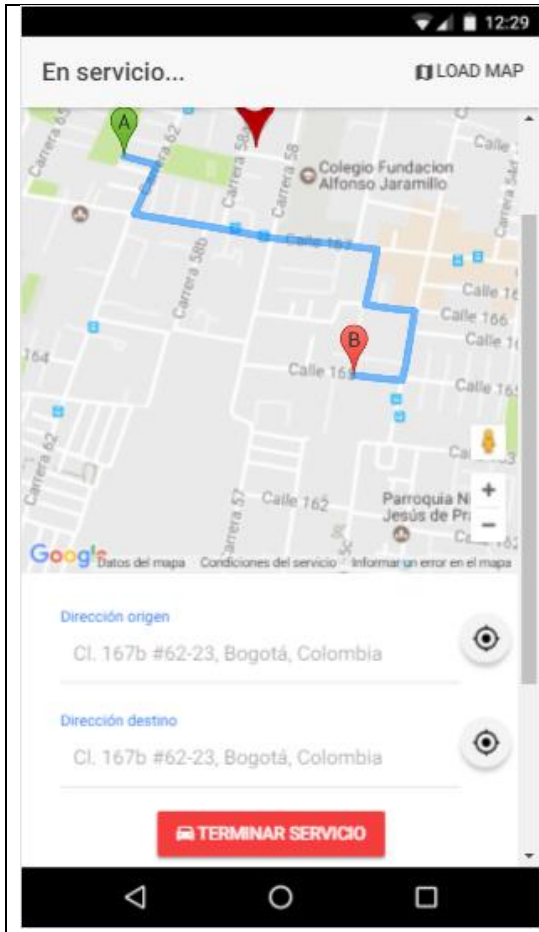


Ilustración 14 Terminar servicio

El usuario notifica al sistema la finalización de un servicio de envío de encomiendas, cuando los usuarios involucrados finalizan dicho servicio.

**Casos de uso Asociados: CU-004, CU-005**

Tabla 7 Árbol de navegación AppShipment

## VI – RESULTADOS

Para el cumplimiento de uno de los objetivos específicos del trabajo de grado, se orienta el desarrollo de AppShipment bajo el marco de Test Driven Development [32]. Este marco permitió al equipo de desarrollo escribir pruebas unitarias de software validando que las pruebas tanto fallen como funcionen, permitiendo el realizar *refactoring* en el código fuente.

Las pruebas escritas en el anexo **TDD Pruebas y Validación**, validan los requerimientos implementados en el prototipo final de AppShipment. El framework utilizado para la escritura y validación de dichas pruebas es Jasmine, dependencia alojada por npm de Node.js. Este framework basado en código JavaScript provee una sintaxis sencilla a la hora de escribir cada prueba unitaria, este se basa en el supuesto de “si esto sucede, esto debería pasar” [32]. Finalmente, una vez obtenidos los resultados de las pruebas realizadas a cada requerimiento implementado en el prototipo final de AppShipment, TDD evalúa y refleja la expectativa pensada al escribir cada prueba.

Sin embargo, la ejecución de esta metodología atrasa el tiempo de implementación dado al lento proceso de escribir pruebas unitarias antes de implementar cualquier requisito, por lo cual el equipo de desarrollo en las últimas iteraciones del calendario decide proceder a implementar los requerimientos restantes bajo el marco metodológico ágil Kanban para lograr cumplir con el alcance establecido, disminuyendo el tiempo implicado al marco de TDD.

Para mayor detalle de las pruebas realizadas diríjase al anexo **TDD Pruebas y Validación** y al código fuente de AppShipment.

## VI – CONCLUSIONES

En esta sección se describen las conclusiones obtenidas al finalizar las fases propuestas en esta memoria. Resaltar el impacto de la solución propuesta en diferentes ámbitos, identificar como se logran los objetivos propuestos de este trabajo de grado y como este podría repercutir en trabajos futuros.

### 6.1 Análisis de Impacto del Desarrollo

AppShipment como prototipo finalizado y funcional tendrá influencia en varias perspectivas, tal y como se presenta a continuación:

Desde una perspectiva económica, el equipo de desarrollo parte del supuesto de que AppShipment como aplicación permitirá a los usuarios obtener una remuneración económica a partir de los recursos que este mismo posee y posiblemente esta aplicación permitirá brindar el mismo como un servicio y obtener remuneraciones económicas. A partir de esto, mejoras en cuanto a calidad de servicio, despliegue del prototipo en otras plataformas y popularización del producto final, podría establecer a AppShipment como una alternativa factible al proceso de envío de encomiendas lo cual fue la real intención para iniciar este trabajo de grado.

A nivel social, el impacto de la solución propuesta e implementada por AppShipment se ve limitada pero no descartada, como alternativa a implementar a nivel nacional, esto debido a el posible rechazo de este tipo de aplicaciones en el contexto en el que se encuentra Colombia como país [31]. Sin embargo, desde el contexto estudiantil y tecnológico, AppShipment se centraría como aplicación que brinda oportunidades alternativas al proceso de envío de encomiendas ayudando a personas con

disponibilidad de tiempo y recursos, lo cuál puede ser aceptado socialmente.

Finalmente, desde un punto de vista disciplinar, el impacto recae tanto en el equipo de desarrollo como en los usuarios finales de AppShipment. Ya que el desarrollo de AppShipment es basado en herramientas nuevas con posibilidad de crecimiento y extensión de funcionalidades, existiría la posibilidad de implementar lo que en esta memoria se menciona como posible trabajo futuro, implementación de mecanismos de pagos, asegurar servicios de envío de encomiendas, mejoramiento del sistema de alertas para los usuarios y nuevas necesidades del usuario que vayan surgiendo en el posible camino de AppShipment. No obstante, eliminar las restricciones, suposiciones y dependencias planteadas durante el desarrollo, pueden hacer parte de un crecimiento disciplinar y aportar al mejoramiento continuo de aprendizaje del equipo de desarrollo y así mismo, de AppShipment como herramienta tecnológica.

## **6.2 Conclusiones y Trabajo Futuro**

La elección de las herramientas para el desarrollo de AppShipment, permitió orientar al prototipo final a evidenciar lo que se propone como alcance del trabajo de grado, se logra implementar la aplicación con un solo código base evitando la replicación de código y poder hacer despliegue en plataformas web y móvil para el sistema operativo Android, características importantes para el flujo de la aplicación.

AppShipment a su vez, logra evidenciar un modelo de economía colaborativa implementado en una aplicación multiplataforma de libre acceso y con posibilidad de extensión para trabajos futuros. Funcionalidades como métodos de pago y aseguramiento de encomiendas hacen parte esencial de la aplicación, las cuales pueden ser implementadas posteriormente dado a las buenas prácticas de desarrollo implementadas. Estas permiten que AppShipment cumpla con atributos de calidad

como escalabilidad, portabilidad y mantenibilidad las cuales aseguran la posible integración y adaptación de funcionalidades futuras.

La importancia de definir una diseño y arquitectura adecuado para AppShipment fundamenta la culminación satisfactoria del trabajo de grado, dicha arquitectura permitió evidenciar el correcto funcionamiento y comportamiento del sistema. Así mismo, el utilizar *DAD* como metodología de desarrollo ayudo a implementar un marco metodológico que se adaptaba a las diferentes necesidades del equipo de desarrollo y de AppShipment. Dicho marco ayudo a la toma de decisiones para enfrentar situaciones complejas tanto a nivel de equipo como profesional, logrando objetivos comunes de una manera ágil y eficaz.

Se logra realizar un proceso adecuado de Ingeniería de requerimientos, el cual permitió definir cómo el sistema debía comportarse y así evitar sobredimensionar el alcance establecido para el trabajo de grado. Se implementa el 60% de los requerimientos en total, evidenciando en el prototipo final de AppShipment la debida priorización de requerimientos junto con las funcionalidades que darían un valor agregado al trabajo de grado y a AppShipment como aplicación, aunque el porcentaje restante se tomará como trabajo futuro.

Por otro lado, las iteraciones realizadas dentro del desarrollo del trabajo de grado, al ser cortas (Semanales, terminando cada iteración con una reunión con el director de trabajo de grado) tuvieron como ventaja la constante retroalimentación por parte del director de trabajo de grado el cual sirvió de guía para la detección de errores y a su vez, llevar un correcto flujo de desarrollo. Sin embargo, fue desfavorable para el equipo de desarrollo la falta de tiempo dadas las condiciones laborales de los integrantes, el no poder distribuir el tiempo de manera habitual, aunque, por otro lado, el equipo de desarrollo supo compensar esta desventaja haciendo uso de lo



establecido en los documentos incurriendo así mismo, en el cambio de calendario y asignación de tareas.

## VII - REFERENCIAS Y BIBLIOGRAFÍA

[1]. R. L. Eugenio Tamayo, Recepción, expedición y envío de mercancías, Editex, 2012.

[2] Libre mercado, «Libre mercado,» 26 03 2016. [En línea]. Disponible en: <http://www.libremercado.com/2016-03-26/la-economia-colaborativa-en-grandes-cifras-1276570467/>.

[3] Semana, Revista, «La economía colaborativa, a pasos agigantados,» Revista Semana, 2016.

[4] Berger, S. and Gleisner, F. (2014). Emergence of Financial Intermediaries in Electronic Markets: The Case of Online P2P Lending. *Business Research*, [online] 2(1), pp.39-65. Disponible en: <http://link.springer.com/article/10.1007/BF03343528>

[5] 10 Aspectos Clave sobre derechos de autor para creación de apps. Available at: <https://apps.co/comunicaciones/tic-legal/10-aspectos-clave-sobre-derecho-de-autor-para-crea/>

[6] Aplicativos móviles , más y mucho más.(2014) Disponible en: <https://colombiadigital.net/actualidad/noticias/item/7783-aplicativos-moviles-mas-y-mucho-mas.html>

[7] Search.proquest.com. (2016). *Apps como una posibilidad más de comunicación entre la marca y su público: un análisis basado en la valoración de los usuarios/Apps as an additonal possibility for the communication between brand and his audience::*

*An analysis based on the users rating - ProQuest.* [online] Disponible en: <http://search.proquest.com/openview/c07d7f460b427f7ae8c01a0b3b6bfbb6/1?pq-origsite=gscholar>

[8] *Economía colaborativa: ¿Riesgo u oportunidad?* IESE Business School. Disponible en: <http://www.iese.edu/es/conoce-iese/prensa-noticias/noticias/2016/septiembre/economia-colaborativa-riesgo-u-oportunidad/>

[9] *Sistema de creación de rutas – Coordenadas geográficas,* Disponible en: <http://upcommons.upc.edu/bitstream/handle/2099.1/21780/92227.pdf>

[10] *Economía colaborativa: 10 ejemplos de modelo de negocio,* Disponible en: <http://www.youngmarketing.co/10-ejemplos-de-economia-colaborativa/>

[11] *¿Nueva economía? La Economía Colaborativa,* Disponible en: <http://www.esade.edu/web/cat/about-esade/today/esade-opinion/viewelement/315964/1/nueva-economia-la-economia-colaborativa>

[12] *Welcome to Ionic 2,* Disponible en: <http://ionicframework.com/docs/v1/guide/preface.html>

[13] *Data Base concept son GIS:* Disponible en: [http://www.colorado.edu/geography/gcraft/notes/datacon/datacon\\_f.html](http://www.colorado.edu/geography/gcraft/notes/datacon/datacon_f.html)

[14] *Cloud computing con Amazon Web Services,* Disponible en: <https://aws.amazon.com/es/what-is-aws/>

[15] *Ángulo de presión y radio de curvatura,* Disponible en : <https://upcommons.upc.edu/bitstream/handle/2117/93720/08Chap08de15.pdf?sequence=8&isAllowed=y>

- [16] *Build an End User Profile*, Disponible en: [http://gsl.mit.edu/media/programs/india-bms-summer-2013/materials/build an end user ---trepreneurship 101.pdf](http://gsl.mit.edu/media/programs/india-bms-summer-2013/materials/build_an_end_user_---trepreneurship_101.pdf)
- [17] *Servientrega*, Disponible en: <http://www.servientrega.com/wps/portal/inicio>
- [18] *FeDex*, Disponible en: <http://www.fedex.com/co/index-mobile.html>
- [19] *Seguro Obligatorio de Transporte, SOAT*, Disponible en: <http://www.soatmundial.com.co/>
- [20] *Secretaria de movilidad*, Disponible en: <https://www.medellin.gov.co/movilidad/temas-de-interes/licencia-conduccion>
- [21] *Pasado Judicial, Policía Nacional*, Disponible en: <http://antecedentes.policia.gov.co:7003/WebJudicial/index.xhtml>
- [22] *Leyes de objetos no permitidos para enviar*, Disponible en: [http://www.correos.es/ss/Satellite/site/informacion\\_cliente-1363187212945-inicio/includeTemplate=COR\\_DetalleGuiasAyuda-sidioma=es\\_ES](http://www.correos.es/ss/Satellite/site/informacion_cliente-1363187212945-inicio/includeTemplate=COR_DetalleGuiasAyuda-sidioma=es_ES)
- [23] *Organización mundial que regula en envío de encomiendas*, Disponible en: <http://www.upu.int/keepitsafe>
- [24] *TDD practice at IBM*, Disponible en: <http://collaboration.csc.ncsu.edu/laurie/Papers/agile07.pdf>
- [25] *Angular 2*, Disponible en: <https://angular.io/>
- [26] *Amazon Web Services about*, Disponible en: <https://aws.amazon.com/es/>
- [27] *Api Google SIG*, Disponible en: <https://developers.google.com/maps/web/?hl=es-419>

[28] *Manhattan algorithm*, Disponible en:  
<http://www.sciencedirect.com/science/article/pii/S0304397507007657>

[29] *Distance Between Point son the Earth's Surface*, Disponible en:  
<https://www.math.ksu.edu/~dbski/writings/haversine.pdf>

[30] *UBER*, [En línea], Disponible en: <https://www.uber.com/en-CO/>.

[31] “*Gremio transportador rechaza el servicio que prestan aplicaciones como Uber*”, Artículo disponible en :  
<http://www.vanguardia.com/actualidad/colombia/288627-gremio-transportador-rechaza-el-servicio-que-prestan-aplicaciones-como-ub>

[32] *Test-Driven Development by Example* , Author: *Kent Beck* , *Three Rivers Institute*. Disponible en :  
[http://cecs.wright.edu/~pmateti/Courses/7140/Lectures/Maintenance/KentBeck\\_TDD\\_byexample.pdf](http://cecs.wright.edu/~pmateti/Courses/7140/Lectures/Maintenance/KentBeck_TDD_byexample.pdf)

## VIII - ANEXOS

### **Anexo 1: Software Requirements Specification**

Este anexo presenta todo el proceso de ingeniería de Requerimientos realizado durante el desarrollo del trabajo de grado. Este documento sirve de base para la implementación y desarrollo de un prototipo funcional de AppShipment, donde se detallan lo que el software debe hacer dado un alcance y su respectiva calendarización

### **Anexo 2: Requerimientos y Casos de Uso**

Este anexo presenta a detalle el contexto en el que se realiza el levantamiento de requerimientos y casos de uso; incluyendo la identificación de actores, levantamiento de información y conceptos que soportan el resultado obtenido.

### **Anexo 3: Software Design Description**

Este anexo detalla todo lo que compone el diseño y Arquitectura, comportamiento del sistema, las interacciones entre componentes y el ambiente del desarrollo de AppShipment. A su vez el de obtener mayor entendimiento tanto de bajo nivel, como de alto nivel de diseño, acerca de que compone en si a AppShipment. En otras palabras, es un documento para que el equipo de desarrollo y los lectores, permitirá conocer el cómo se implementó el prototipo final de AppShipment.

### **Anexo 4: Manual de Usuario**

Este anexo presenta un manual para el usuario de AppShipment, el documento relacionado presenta la manera en que se puede acceder a cada una de las funciones implementadas por AppShipment.

### **Anexo 5: Manual de Instalación**

Este anexo presenta un manual para el usuario mediante el cual puede instalar en su dispositivo móvil a AppShipment como prototipo final.

## **Anexo 6: Bibliografía**

En este anexo se encuentra toda la bibliografía y referencias que ayudaron a la construcción de este documento y sus anexos.

## **Anexo 7: Colección de enlaces**

En este anexo se presentan todos los enlaces relacionados al presente trabajo de grado, los cuales complementa la información presentada en este documento con la intención de contextualizar al lector con trabajos relacionados a este campo.

## **Anexo 8: TDD Pruebas y Validación**

Este anexo describe las pruebas unitarias realizadas bajo el marco de TDD, para realizar la validación correspondiente entre el prototipo final de AppShipment y los requerimientos funcionales obtenidos en el proceso de ingeniería de requerimientos.