



UNIVERSIDAD CENTRAL DE VENEZUELA
FACULTAD DE CIENCIAS
ESCUELA DE COMPUTACIÓN
CENTRO DE ENSEÑANZA ASISTIDA POR COMPUTADOR - CENEAC

Desarrollo de una aplicación para la gestión de envíos entre clientes corporativos

**Trabajo Especial de Grado ante la
Ilustre Universidad Central de Venezuela por el Bachiller
Vincenzo Antonio Bianco Verdu C.I. 21.117.244
para optar al título de Licenciado en Computación**

Tutora: Profa. Yusneyi Y. Carballo Barrera

Ciudad Universitaria de Caracas, 11 de Octubre de 2017

ACTA

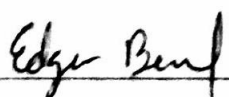
Quienes suscriben, miembros del jurado designado por el Consejo de la Escuela de Computación, para examinar el Trabajo Especial de Grado titulado **"Desarrollo de una aplicación para la gestión de envíos entre clientes corporativos"**, presentado por el Bachiller **Vincenzo Antonio Bianco Verdu (C.I. V-21.117.244)**, a los fines de optar al título de Licenciado en Computación, se deja en constancia de lo siguiente:

Leído como fue dicho trabajo, por cada uno de los miembros del jurado, se fijó el día **Miércoles, 11 de Octubre de 2017**, a las **10:30** horas, para que el autor lo defendiera en forma pública, lo que éste hizo en la **Sala planta** SALA 1 de la Escuela de Computación, mediante una presentación oral de su contenido, luego de lo cual respondió a las preguntas formuladas. Finalizada la defensa pública del Trabajo Especial de Grado, el jurado decidió aprobar con la nota de 19 puntos.

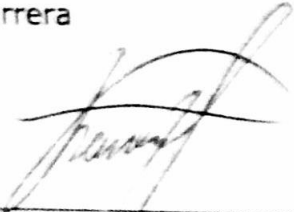
En fe de lo cual se levanta la presente Acta, en Caracas el día 11 de Octubre de 2017



Prof. Yusneyi Carballo Barrera
(Tutora)



Prof. Edgar Bernal
(Jurado)



Prof. Franky Uzcátegui
(Jurado)



Universidad Central de Venezuela
Facultad de Ciencias
Escuela de Computación
Centro de Enseñanza Asistida por Computador - CENEAC

Desarrollo de una aplicación para la gestión de envíos entre clientes corporativos

Autor:

Vincenzo Antonio Bianco Verdu, vincenzobianco1993@gmail.com

Tutora:

Profa. Yusneyi Carballo Barrera, yusneyi.carballo@ciens.ucv.ve

Fecha:

11 de Octubre de 2017

RESUMEN

En este Trabajo Especial de Grado se describe el desarrollo de dos aplicaciones que trabajan en conjunto para gestionar el envío de mensajes en clientes corporativos. Se creó una aplicación web que permite a los usuarios realizar envíos a destinatarios dentro un directorio empresarial, con etiquetas de identificación únicas, asignación de mensajeros y acceso a funcionalidades según distintos roles de usuario. También se describe el desarrollo de una aplicación móvil que permitirá la gestión del seguimiento de los envíos que son asignados a los mensajeros, donde los usuarios reciben notificaciones cuando tienen un nuevo envío asignado, pueden verificar la validez de una etiqueta mediante un escáner, cambiar el estado de un envío o reportar inconvenientes con el mismo. Ambos desarrollos apoyan los procesos de negocio de una empresa líder en el área de mensajería corporativa, respondiendo a necesidades reales e incorporando aplicaciones que fueron probadas en ambientes reales de trabajo. Para el diseño y desarrollo de las aplicaciones se usaron diversas herramientas y tecnologías, entre ellas JavaScript, Node.js, MongoDB, AngularJs, Android, Bootstrap y Ajax. Se trabajó con la metodología de desarrollo ágil Scrum y se siguieron los estándares actuales de diseño web, lineamientos de usabilidad y accesibilidad.

Palabras clave: Envíos corporativos, Gestión de mensajería, Servicio de Mensajería, Point-to-Point Messaging Service.

Agradecimientos

A mi familia, quienes siempre me han guiado y apoyado de una forma incondicional durante toda mi vida, a ustedes les debo todo, este logro es para ustedes.

A la memoria de mi abuelo Antonio Bianco y mi abuelo Verdu, quienes en vida fueron los mejores abuelos que pude tener, siempre me consintieron y me dieron mucho amor, desde el cielo sé que me siguen guiando y espero que desde allá se puedan sentir orgullosos de mí.

A mi Padre Vincenzo Bianco, por darme tantas oportunidades para que pueda lograr grandes cosas, por enseñarme que en esta vida nos caemos para aprender a levantarnos, gracias por ser mi ejemplo a seguir.

A mi Madre Morella Verdu, por su apoyo incondicional en los momentos más difíciles, gracias por mostrarme que la fe mueve montañas.

A mi Abuela Aide, por ser la mejor abuela que puedo tener, por siempre creer en mí, por tenerme tanta paciencia, gracias por enseñarme tantas cosas valiosas desde muy pequeño.

A todos los profesores que han dejado una huella en mí de alguna manera, a aquellos que creyeron en mí y me dieron aliento cuando más lo necesitaba, estaré eternamente agradecido con ustedes por haberme acompañado en todo mi camino.

A mi Tutora Yusneyi Carballo, por ser de esas profesoras que hacen las cosas con amor y dan un ejemplo a seguir, gracias a su forma especial de impartir clases fue que logré tener la mejor base para programar durante toda la carrera, y ahora que la estoy finalizando, es ella quién se dedicó a apoyarme para poder lograr esta meta.

A todas esas amistades que me ha regalado la vida, gracias por hacer mis días tan especiales.

Vincenzo Bianco

ÍNDICE

RESUMEN.....	3
ÍNDICE	5
INTRODUCCIÓN.....	I
CAPÍTULO I. PLANTEAMIENTO DEL PROBLEMA.....	3
1.1 PLANTEAMIENTO DEL PROBLEMA.....	3
1.2 OBJETIVO GENERAL.....	4
1.3 OBJETIVOS ESPECÍFICOS.....	4
1.4 REQUERIMIENTOS FUNCIONALES.....	4
1.5 REQUERIMIENTOS NO FUNCIONALES	5
1.5 ALCANCE	5
1.6 ANTECEDENTES.....	5
<i>Fedex</i>	5
<i>Lead Logística</i>	6
<i>MRW</i>	6
CAPÍTULO II. MARCO TEÓRICO	8
2.1. BASES CONCEPTUALES	8
<i>Servidor Web</i>	8
<i>Arquitectura Cliente/Servidor</i>	8
<i>Marco de Trabajo</i>	9
<i>Sistema de Gestión de Base de Datos</i>	9
<i>Bases de Datos NoSQL</i>	9
<i>Tipos de base de datos NoSQL</i>	10
<i>Modelo Vista Controlador (MVC)</i>	11
<i>Interfaz de Programación de Aplicaciones (API, Application Programming Interface)..</i>	12
<i>Transferencia de Estado Representacional (REST, Representational State Transfer)...</i>	13
<i>JavaScript</i>	13
<i>Objeto de Notación JavaScript (JSON)</i>	14
<i>Android</i>	14
<i>Sistema de Control de Versiones</i>	15
2.2. TECNOLOGÍAS DEL LADO DEL CLIENTE	15
<i>Lenguaje de Marcado de Hipertexto HTML</i>	15
<i>Hojas de Estilo en Cascada CSS</i>	15
<i>jQuery</i>	15
<i>Bootstrap</i>	16
<i>Sass</i>	16
<i>AngularJs</i>	16
2.3. TECNOLOGÍAS DEL LADO DEL SERVIDOR	17
<i>Node.js</i>	17
<i>Express</i>	17
<i>MongoDB</i>	18
<i>Modelado de datos en MongoDB</i>	18
<i>Amazon Elastic Compute Cloud (Amazon EC2)</i>	19
<i>Compose</i>	19
2.4. HERRAMIENTAS DE APOYO AL DESARROLLO DE SOFTWARE	20
<i>Gulp</i>	20
<i>Git</i>	20

<i>RoboMongo</i>	20
<i>Swagger UI</i>	20
CAPÍTULO III. MARCO METODOLÓGICO	22
3.1 METODOLOGÍA DE DESARROLLO DE SOFTWARE	22
3.2 METODOLOGÍAS ÁGILES	24
3.3 JUSTIFICACIÓN PARA LA SELECCIÓN DE SCRUM	25
3.4 SCRUM	25
3.4.1 ROLES	26
<i>Dueño del Producto</i>	26
<i>Equipo de Desarrollo</i>	26
<i>Scrum Master</i>	27
3.4.2 ARTEFACTOS	27
<i>Lista de Producto (Product Backlog)</i>	27
<i>Historias de Usuario</i>	27
<i>Lista de Requerimientos Pendientes (Sprint Backlog)</i>	27
<i>Tablero Scrum</i>	28
3.4.3 EVENTOS	28
<i>Sprints</i>	28
<i>Reunión Diaria</i>	29
<i>Reunión de Planificación de Sprint</i>	29
<i>Revisión de Sprint</i>	29
<i>Retrospectiva de Sprint</i>	30
CAPÍTULO IV. DESARROLLO DE LA SOLUCIÓN	32
4.1. DESCRIPCIÓN DE LA SOLUCIÓN	32
4.2. TECNOLOGÍAS Y LENGUAJES	33
4.3 PERFILES DE USUARIO	34
4.3 PRINCIPALES PROCESOS DE NEGOCIO	37
<i>Proceso de asignación de etiquetas a compañías y sucursales</i>	37
<i>Proceso de invitación a nuevos usuarios</i>	38
<i>Proceso de envío de correo corporativo</i>	39
4.4 APLICACIÓN DE LA METODOLOGÍA SCRUM	40
4.4.1 OBJETIVOS DE LOS SPRINTS (<i>SPRINTS GOALS</i>)	40
4.4.2 ITERACIONES	40
4.5 CASOS DE USO MÁS IMPORTANTES	43
4.6 MODELO DE DATOS	65
4.7 PRINCIPALES INTERFACES DE LA APLICACIÓN	67
4.8 PRUEBAS REALIZADAS	86
4.9 MEJORAS INCORPORADAS	93
RESULTADOS	96
CONCLUSIONES	98
RECOMENDACIONES Y TRABAJOS FUTUROS	100
REFERENCIAS BIBLIOGRÁFICAS Y DIGITALES	101
GLOSARIO DE TÉRMINOS	104

ÍNDICE DE FIGURAS

Figura 1. Diagrama Cliente-Servidor	9
Figura 2. Diagrama Modelo Vista Controlador	12
Figura 3. Diferencias entre Metodologías Ágiles y no Ágiles	23
Figura 4. Arquitectura de la solución	32
Figura 5. Proceso de asignación de etiquetas a compañías y sucursales	37
Figura 6. Proceso de asignación de etiquetas a compañías y sucursales	38
Figura 7. Proceso de envío de correo corporativo	39
Figura 8. Diagrama de casos de uso. Nivel 0	43
Figura 9. Diagrama de casos de uso de la aplicación web Nivel 1: Administrador de compañía y Administrador de aplicación	43
Figura 10. Diagrama de casos de uso de la aplicación web Nivel 1: Administrador de aplicación	44
Figura 11. Diagrama de casos de uso de la aplicación web Nivel 1: Administrador de compañía.....	44
Figura 12 .Diagrama de casos de uso de la aplicación web Nivel 1: Usuario de compañía.....	45
Figura 13. Diagrama de casos de uso de la aplicación móvil Nivel 1: Mensajero...	45
Figura 14. Diagrama de casos de uso de la aplicación web Nivel 2: Administrador de aplicación	50
Figura 15. Diagrama de casos de uso de la aplicación web Nivel 2: Administrador de aplicación. Parte 2	52
Figura 16. Diagrama de casos de uso de la aplicación web Nivel 2: Administrador de aplicación. Parte 3	54
Figura 17. Diagrama de casos de uso de la aplicación web Nivel 2: Administrador de aplicación y Administrador de compañía	55
Figura 18. Diagrama de casos de uso de la aplicación web Nivel 2: Administrador de aplicación y Administrador de compañía	56
Figura 19. Diagrama de casos de uso de la aplicación web Nivel 2: Administrador de aplicación. Parte 4	58
Figura 20. Diagrama de casos de uso de la aplicación web Nivel 2: Administrador de compañía.....	58
Figura 21. Diagrama de casos de uso de la aplicación web Nivel 2: Administrador de aplicación. Parte 5	59
Figura 22. Diagrama de casos de uso de la aplicación web Nivel 2: Administrador de compañía y Usuario de compañía.....	60
Figura 23. Diagrama de casos de uso de la aplicación web Nivel 2: Administrador de compañía y Usuario de compañía. Parte 2.....	62
Figura 24. Diagrama de casos de uso de la aplicación web Nivel 2: Mensajero	64
Figura 25. Modelo de datos orientado a documentos	66
Figura 26. Página inicio de sesión	67
Figura 27. Recuperación de contraseña	68
Figura 28. Edición de perfil.....	68
Figura 29. Sección de compañías	69
Figura 30. Formulario para nuevas compañías	69
Figura 31. Datos de una compañía	70

Figura 32. Buscador de compañías	70
Figura 33. Sucursales de una compañía	71

Figura 34. Sección de etiquetas	71
Figura 35. Lotes de etiquetas asignadas a una compañía	72
Figura 36. Formulario de asignación de etiquetas.....	72
Figura 37. Sección de mensajeros.....	73
Figura 38. Sección de usuarios	73
Figura 39. Emisión de facturas	74
Figura 40. Facturas emitidas	74
Figura 41. Directorio de una compañía	75
Figura 42. Lista de usuarios favoritos	75
Figura 43. Formulario para carga de múltiples usuarios	76
Figura 44. Sección de envíos	76
Figura 45. Formulario de nuevo envío.....	77
Figura 46. Asignación múltiple.....	77
Figura 47. Reporte de etiquetas y compañías	78
Figura 48. Reporte de mensajeros	78
Figura 49. Inicio de sesión aplicación móvil	79
Figura 50. Inicio de sesión aplicación móvil	80
Figura 51. Menú de la aplicación móvil	81
Figura 52. Sección mis envíos	82
Figura 53. Escáner	83
Figura 54. Detalle de un envío	84
Figura 55. Formulario de seguridad	85

ÍNDICE DE TABLAS

Tabla 1. Objetivos de los Sprints (<i>Sprints Goals</i>)	40
Tabla 2. Pruebas funcionales - Aplicación web. Parte 1	87
Tabla 3. Pruebas funcionales - Aplicación web. Parte 2	87
Tabla 4. Pruebas funcionales - Aplicación web. Parte 3	88
Tabla 5. Pruebas funcionales - Aplicación web. Parte 4	89
Tabla 6. Pruebas funcionales - Aplicación web. Parte 5	90
Tabla 7. Pruebas funcionales - Aplicación web. Parte 6	91
Tabla 8. Pruebas funcionales - Aplicación web. Parte 7	92
Tabla 9. Pruebas funcionales - Aplicación móvil	93

INTRODUCCIÓN

El gran avance de la tecnología ha dado paso a la evolución informática. Las tecnologías han tenido que evolucionar de la mano a las necesidades humanas para solventar nuevos problemas que se han presentado.

La expansión de la información a través de internet ha traído como consecuencia el almacenamiento, organización y representación de los datos, para que los usuarios puedan acceder a ellos de una forma fácil y amena.

Como consecuencia del crecimiento de internet y sus tecnologías derivas, las empresas en la actualidad se ven en la necesidad de hacer uso de estas tecnologías con el fin de mejorar sus servicios. Tealca C.A una empresa venezolana con más de 30 años de experiencia en el servicio de entrega de encomiendas, paquetería y carga en Venezuela. Tiene como principal objetivo brindar a sus clientes la mayor seguridad en el traslado de sus encomiendas brindando un servicio acorde con sus expectativas.

Con el tiempo a Tealca C.A. le surge la necesidad de ofrecer un servicio de envíos para clientes corporativos, para el cual requería una solución tecnológica que le permitiera administrar sus clientes, los mensajeros que manejarán los envíos y las etiquetas que identificarán de forma única a los envíos.

Para dar cumplimiento a las necesidades de la empresa objeto de estudio, se desarrollaron dos aplicaciones, una aplicación web, donde se gestionan las compañías, sucursales, etiquetas, envíos, mensajeros y reportes, y una aplicación móvil para que los mensajeros gestionen los envíos asignados. A la solución desarrollada con la combinación de estas aplicaciones se le dio el nombre de Aquiles.

Las tecnologías utilizadas fueron: JavaScript (lenguaje de programación), Express y AngularJS (marcos de trabajo), MongoDB (sistema manejador de base de datos), Java (Android nativo) y la metodología Scrum modificada.

Para describir el desarrollo de los objetivos enmarcados en esta investigación se ha estructurado este documento de la siguiente manera:

En el **capítulo I** se expone el objetivo general del Trabajo Especial de Grado, objetivos específicos, requerimientos funcionales, requerimientos no funcionales, alcance de la aplicación y antecedentes.

En el **capítulo II** se definen los conceptos relacionados a la investigación que corresponden a lo que es el marco conceptual y las herramientas de desarrollo.

En el **capítulo III** se presenta la metodología de desarrollo usada.

En el **capítulo IV** se describe la solución realizada, las tecnologías utilizadas, se muestran los procesos principales de la aplicación, se muestran los principales casos de uso de la aplicación, se describen los perfiles existentes dentro de la aplicación, se explica cómo se usó Scrum como metodología para el desarrollo de las soluciones, se describen los objetivos y el desarrollo de cada Sprint y se muestran la principales interfaces.

Finalmente, se presentan los resultados, conclusiones, recomendaciones y las fuentes bibliográficas consultadas.

CAPÍTULO I. Planteamiento del Problema

En este capítulo se presenta el planteamiento del problema, el objetivo general del Trabajo Especial de Grado, los objetivos específicos, requerimientos funcionales, no funcionales, el alcance de la aplicación y algunos antecedentes.

1.1 Planteamiento del Problema

En instituciones y empresas de gran tamaño, en adelante referidas como corporaciones, es común que las personas necesiten enviar documentos, paquetes o encomiendas entre sí. Se estima que en las grandes corporaciones se realizan miles de envíos por mes, lo que eleva los gastos y los esfuerzos en logística, pues en la mayoría de los casos éstas corporaciones optan por contratar más personal para que se encargue de realizar los envíos.

La cantidad de envíos que se realizan en las corporaciones es variante, puede que en un momento el volumen de los envíos que se realizan disminuya, dejando que el personal contratado para esta tarea específica quede en un estado de ocio, o el caso contrario en donde el volumen de envíos aumenta y las corporaciones se vean en necesidad de contratar más personal para esta labor.

Dadas las complicaciones que genera para las corporaciones gestionar sus propios envíos, Tealca C.A una empresa venezolana con más de 30 años de experiencia en el servicio de entrega de encomiendas, paquetería y carga en Venezuela, identifica la oportunidad de ofrecer un servicio de envíos a la medida para sus clientes, ya que cuenta con el personal y la experiencia necesaria para gestionar esta problemática. Sin embargo, requiere de soluciones informáticas que apoyen la gestión de este proceso

Vale destacar que con los años de experiencia que tiene Tealca C.A. sus procesos internos ya se encuentran bien definidos, es por ello que se ven en la necesidad de tener una solución tecnológica que se ajuste a su forma de trabajar, puesto que hacer uso de alguna aplicación ya existente implicaría adaptarse a esta, lo cual no es viable. Tener una aplicación propia le brindará una gran ventaja competitiva pues con esto se reforzaran los procesos ya existentes y permitirá manejar los nuevos procesos necesarios para la gestión de envíos internos en múltiples compañías de una forma centralizada.

Para poder cubrir las necesidades tecnológicas de la empresa objeto de estudio, se desarrollaron dos aplicaciones, una aplicación web, donde se gestionan las compañías, sucursales, etiquetas, envíos, mensajeros y reportes, y una aplicación móvil para que los mensajeros gestionen los envíos asignados. A la solución desarrollada con la combinación de estas aplicaciones se le dio el nombre de Aquiles.

1.2 Objetivo General

Desarrollar aplicaciones para la gestión de envíos a medida realizados entre clientes corporativos.

1.3 Objetivos Específicos

1. Analizar y modelar los procesos principales del negocio utilizando notación y diagramas BPMN.
2. Utilizar una metodología que garantice un proceso ágil y eficiente en el tiempo de desarrollo de solución.
3. Diseñar y desarrollar funcionalidades basándose en arquitecturas que permitan servir la información adecuada entre la aplicación web de gestión de los envíos y la aplicación móvil de gestión de mensajeros.
4. Diseñar las aplicaciones incorporando estándares y lineamientos de usabilidad, con el fin de lograr una interfaz clara y limpia para el usuario final.
5. Documentar el proceso de desarrollo y las aplicaciones.
6. Realizar pruebas de funcionalidad, navegabilidad y accesibilidad para verificar el correcto funcionamiento de las aplicaciones con casos de prueba reales.

1.4 Requerimientos Funcionales

Para la aplicación web:

- Administrar de manera independiente distintas compañías, para poder gestionar los envíos que se realizan dentro las mismas.
- Gestionar sucursales dentro de cada compañía.
- Debe estar internacionalizada.
- Manejar roles con distintos niveles de acceso a las funcionalidades.
- Manejar notificaciones por correo.
- Directorio empresarial, cada compañía registrada debe poder administrar un directorio con todos los empleados o usuarios que serán partícipes en la aplicación, almacenando información relevante como departamento y locación.
- Los usuarios registrados en directorio empresarial deben poder crear nuevos envíos, para ello deben indicar el usuario destino y una etiqueta única que servirá para identificar su envío.
- Permitir a los administradores gestionar los mensajeros que prestarán sus servicios de envío.
- Manejar reportes con los cuales se puedan visualizar el uso de las etiquetas y el rendimiento de los mensajeros.

- Permitir la administración de las etiquetas que son usadas para los envíos en las compañías.

Para la aplicación móvil:

- Permitir a los mensajeros manejar el estado de un envío. Estos estados se modifican al momento de confirmar la recepción de los envíos o cuando son entregados a su destinatario.
- Verificar que las etiquetas asignadas a los envíos sean válidas.

1.5 Requerimientos no funcionales

- **Disponibilidad:** Las aplicaciones deben estar en un estado operable cada vez que se necesiten.
- **Escalabilidad:** Las aplicaciones deben de manejar una creciente carga de trabajo, por ejemplo un mayor número de compañías que aumente la cantidad de usuarios.
- **Usabilidad:** Las aplicaciones deben ser de fácil uso y aprendizaje.
- **Rendimiento:** Las aplicaciones deben responder con rapidez las solicitudes que hagan los usuarios.

1.5 Alcance

- Permitir la administración de la información de las compañías, sucursales, usuarios, mensajeros, envíos y etiquetas.
- En la aplicación móvil los mensajeros solo podrán gestionar los envíos que les son asignados.
- Las notificaciones a los usuarios se realizará mediante correo electrónico.
- El pase a producción debe realizarse utilizando Amazon Web Services (AWS) y el servicio web Compose.

1.6 Antecedentes

El presente Trabajo Especial de Grado surge a partir de la necesidad de ofrecer un servicio de envíos a la medida para grandes corporaciones. Se realizó un proceso de investigación de otras empresas que ofrecen servicios similares, observando que las herramientas que usan estas empresas para la gestión de sus procesos no son accesibles al público. Las empresas investigadas fueron:

FedEx ®

Es una compañía que transporta paquetes a nivel mundial, gestionando algunos de sus servicios a través de aplicaciones web. Para poder realizar envíos con Fedex los usuarios deben completar el proceso de solicitud de un envío, en donde se indica la información del paquete, la ubicación del emisor y la ubicación del receptor.

Cuando un usuario realiza la solicitud de un envío, FedEx se encarga de localizar el medio de transporte y el mensajero más adecuado para atender la solicitud realizada, posteriormente el mensajero se dirige a la ubicación del usuario para recoger el paquete, finalmente cuando el mensajero lleva el paquete a su destino, se debe evaluar la entrega indicando si el servicio fue o no satisfactorio.

Durante todo el proceso de traslado se posee información del paquete, de esta manera se puede determinar si llegará con demora a su destino final, si se extravió o si tuvo algún daño en el proceso. Esto se logra gracias a un dispositivo llamado Supertracker®, que es un computador pequeño que permite rastrear los paquetes que manipula FedEx (FedEx, 2017)

Lead Logística

Es una compañía argentina especializada en todo tipo de movimiento de cargas, ya sea con camiones completos o cargas parciales, coordinando viajes punto a punto o repartos programados.

Lead Logística analiza las necesidades de cada cliente y administra los medios necesarios para brindar las mejores soluciones en todos los procesos que conforman la cadena de abastecimiento. Esta compañía cuenta con una gran variedad de camiones y camionetas para poder satisfacer las necesidades de sus clientes. Además, sus unidades de transporte cuentan con modernos sistemas de seguimiento satelital, siendo optativo y a requerimiento del cliente la posibilidad de disponer de custodia armada en el caso de transporte de cargas costosas.

Esta compañía no posee una aplicación web con acceso al público, es por ello que desconoce cómo apoyan informáticamente los procesos internos para el transporte de carga de sus clientes registrados (Lead Logística, 2017).

MRW

MRW es una corporación venezolana que ofrece un servicio de envíos masivos a nivel internacional y nacional a través de sus centros de atención u oficinas. Entre las características principales de su servicio de correspondencia se encuentra la asignación de prioridad de entrega de acuerdo a las necesidades del usuario y la distribución de los paquetes por vía terrestre y por correo.

El proceso de envío en MRW tiene como prioridad ofrecer a sus clientes la alternativa más económica y eficaz. Para ello, el individuo debe acudir a la oficina MRW más cercana a su domicilio (la ubicación de estos centros de atención pueden ser consultadas en el portal oficial de MRW, habilitado para el uso público) y aportar los datos de envío. Posteriormente, el personal le facilitará al cliente la alternativa más eficaz y conveniente.

Aunado a esto, MRW también ofrece la alternativa de resguardar las encomiendas por medio de "Mensajeros Radio Worldwide, CA" de tal manera que se pueda garantizar nuevamente el envío seguro del paquete. Este sistema de resguardo no aplica para los artículos prohibidos según la ley vigente (MRW, 2017).

CAPÍTULO II. Marco Teórico

La finalidad de este capítulo es presentar las bases conceptuales que sirvieron de fundamento para el análisis y desarrollo de este trabajo de investigación, así como describir brevemente las tecnologías y herramientas de apoyo utilizadas para el desarrollo de las soluciones de software planteadas en este trabajo.

2.1. Bases Conceptuales

Servidor Web

Es un programa que gestiona cualquier aplicación en el lado del servidor realizando conexiones bidireccionales y/o unidireccionales y síncronas o asíncronas con el cliente generando una respuesta en cualquier lenguaje en el lado del cliente. El código recibido por el cliente suele ser compilado y ejecutado por un Navegador Web. Para la transmisión de datos se utiliza algún protocolo, el más usado es el protocolo HTTP¹. (Ecured.cu, Servidor Web, s.f.)

Arquitectura Cliente/Servidor

La arquitectura cliente/servidor es un modelo para el desarrollo de sistemas de información en el que las transacciones se dividen en procesos independientes que cooperan entre sí para intercambiar información, servicios o recursos. (Ecured.cu, Servidor Web, s.f.)

En la manera de describir la forma de trabajo entre los clientes y los ordenadores se define:

- **Cliente:** Es el ordenador que pide información a otro, mediante la aplicación de un programa llamado cliente. Este contacta con el servidor y da formato a la petición de la información y da formato a la respuesta.
- **Servidor:** Es el ordenador que ofrece la información mediante la aplicación de un programa llamado servidor que recibe la información y la procesa y responde enviando la petición al cliente.

¹ El Protocolo de Transferencia de Hipertexto (*HTTP*) es un protocolo de la capa de aplicación para la transmisión de documentos hipermedia. (Mozilla Developer Network, s.f.)



Figura 1. Diagrama Cliente-Servidor
Fuente: (Ecured, 2016)

Marco de Trabajo

Un Marco de Trabajo (*Framework*) es una estructura conceptual y tecnológica de soporte definida, normalmente con artefactos o módulos de software concretos, en base a la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, puede incluir soporte de programas, librerías y un lenguaje interpretado entre otros programas para ayudar a desarrollar y unir los diferentes componentes de un proyecto.

Representa una arquitectura de software que modela las relaciones generales de las entidades del dominio. Provee una estructura y una metodología de trabajo la cual extiende o utiliza las aplicaciones del dominio. (EcuRed, Framework, s.f.)

Sistema de Gestión de Base de Datos

Un Sistema Manejador de Base de Datos (SMBD) consiste en una colección de datos interrelacionados y un conjunto de programas para acceder a dichos datos y su objetivo principal es proporcionar una forma de almacenar y recuperar la información de una base de datos de manera que sea tanto práctica como eficiente. (Silberschatz A., 2002)

Bases de Datos NoSQL

Según (MongoDB, 2017) las bases de datos NoSQL abarcan una amplia variedad de diferentes tecnologías de bases de datos, que se desarrollaron en respuesta a demandas que fueron surgiendo en el desarrollo de aplicaciones modernas, algunas de esas razones son las siguientes:

- Aplicaciones que crean grandes volúmenes de datos que cambian rápidamente, datos estructurados, semiestructurados, no estructurados y polimórficos.
- Aplicaciones que antes atendían a una audiencia finita ahora se ofrecen como servicios que deben estar siempre disponibles, accesibles desde muchos dispositivos y con escalabilidad de millones de usuarios.
- Las organizaciones están recurriendo ahora a arquitecturas escalables usando software de código abierto, servidores básicos y la computación en la nube, en lugar de grandes servidores monolíticos e infraestructuras de almacenamiento.
- Las bases de datos relacionales no fueron diseñadas para hacer frente a los desafíos de escalabilidad y agilidad que enfrentan las aplicaciones modernas, ni fueron construidas para aprovechar la capacidad de almacenamiento y procesamiento de productos básicos disponibles hoy en día.

Entre los aspectos más importantes que diferencian las bases de datos relacionales y las NoSQL tenemos que: las bases de datos NoSQL no usan SQL como lenguaje principal de consultas, los datos almacenados no requieren estructuras fijas como tablas, no garantizan completamente ACID (*Atomicity, Consistency, Isolation and Durability*: atomicidad, consistencia, aislamiento y durabilidad), y habitualmente escalan bien horizontalmente (LITH, 2010).

En comparación con las bases de datos relacionales, las bases de datos NoSQL son más escalables y proporcionan un rendimiento superior, hay que destacar que la escalabilidad no sólo se trata de velocidad, sino de tres métricas diferentes, que a menudo trabajan juntas:

- Escala cluster

Las bases de datos se pueden distribuir en más de cien nodos, a menudo en diferentes centros de datos.

- Escala en rendimiento

Son capaces de soportar más de cientos de miles de operaciones por segundo.

- Escala en Datos

Las bases de datos pueden almacenar más de un billón de documentos.

Tipos de base de datos NoSQL

Según (MongoDB, 2017), hasta ahora se reconocen cuatro tipos de base de datos, las cuales brindan distintos beneficios que pueden ser aprovechados dependiendo de las necesidades que se tengan para los proyectos, las mayores

diferencias que tienen estas bases de datos entre sí, son la forma en que almacenan los datos y el cómo se consultan estos mismos. Estos tipos de base de datos son:

- **Orientadas a documentos**

Asocian cada clave con una estructura de datos compleja conocida como documento. Los documentos pueden contener muchos pares clave-valor diferentes, pares de matrices de claves o incluso documentos anidados.

- **Orientadas a grafos**

Se utilizan para almacenar información sobre redes de datos, como las conexiones sociales, algunos ejemplos de estas son Neo4J y Giraph.

- **Clave valor**

Son bases de datos más simples. Cada elemento de la base de datos se almacena como un nombre de atributo (o clave), junto con su valor. Dentro las bases de datos de este tipo destacan: Redis, Riak y Berkeley DB.

- **Orientas a columnas anchas**

Se optimizan para consultas sobre conjuntos de datos grandes y almacenan columnas de datos en lugar de filas, Cassandra y HBase son ejemplos de estas bases de datos.

Modelo Vista Controlador (MVC)

El patrón de arquitectura Modelo Vista Controlador es un patrón que define la organización independiente del Modelo (objetos de negocio), la Vista (interfaz con el usuario u otro sistema) y el Controlador (controlador del flujo de la aplicación).

De esta forma, se divide el sistema en tres capas donde tenemos la encapsulación de los datos, la interfaz o vista por otro y por último la lógica interna o controlador. (Gomez, s.f.)

El patrón de arquitectura MVC, es una filosofía de diseño de aplicaciones, compuesta por:

- **Modelo:**

- Contiene el núcleo de la funcionalidad (dominio) de la aplicación.
- Encapsula el estado de la aplicación.
- Es independiente del Controlador y de la Vista.

- **Vista:**

- Es la presentación del Modelo.

- Puede acceder al Modelo pero nunca cambiar su estado.
- Puede ser notificada cuando hay un cambio de estado en el Modelo.
- **Controlador:**
 - Reacciona a la petición del Cliente, ejecutando la acción adecuada y creando el modelo pertinente.

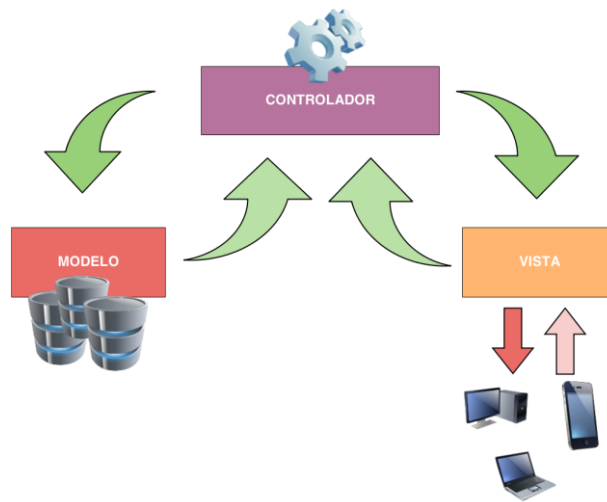


Figura 2. Diagrama Modelo Vista Controlador
Fuente: (FROAC, 2012)

Interfaz de Programación de Aplicaciones (API, *Application Programming Interface*)

Es un conjunto particular de reglas de código, protocolos y especificaciones que los programas de software pueden utilizar para comunicarse entre sí. Sirve como una interfaz entre diferentes programas de software y facilita su interacción, similar a la forma en que la interfaz de usuario facilita la interacción entre los seres humanos y las computadoras.

Según (Clarke, 2004) al igual que una interfaz gráfica de usuario facilita a las personas el uso de programas, las API's facilitan a los desarrolladores el uso de ciertas tecnologías en la creación de aplicaciones. Al abstraer la implementación subyacente y sólo exponer objetos o acciones que necesita el desarrollador, una API simplifica la programación. Así como una interfaz gráfica para un cliente de correo electrónico puede proporcionar al usuario un botón que realiza todos los pasos para buscar y resaltar nuevos correos electrónicos, una API para la entrada / salida de archivos podría dar al desarrollador una función que copia un archivo de una ubicación a otra sin requerir que el desarrollador entienda las operaciones del sistema de archivos que ocurren internamente.

Vale la pena resaltar que las API's pueden estar escritas en cualquier lenguaje de programación sin ningún tipo de problema, ya que al seguir estándares para la

comunicación, al momento en que un tercero escrito en un lenguaje diferente, desee comunicarse con la misma, los datos que el API le retornará, vendrán en un formato que cualquiera puede entender, comúnmente estos datos vienen en forma de XML o JSON.

Transferencia de Estado Representacional (REST, *Representational State Transfer*)

Es un estilo de arquitectura software para sistemas hipermedia distribuidos. El término se originó gracias a (Fielding, 2000) quien fue el que lo introdujo por primera vez en su tesis doctoral, actualmete se usa para describir cualquier interfaz web que utilice XML o JSON a traves de protocolo HTTP. Posteriormente, el termino RESTful surgió para hacer referencia a un servicio web que implementa la arquitectura REST.

La web ha disfrutado de escalabilidad como resultado de una serie de diseños fundamentales clave:

- Un protocolo cliente/servidor sin estado, donde no se pueden recordar comunicaciones previas, sin embargo existen mecanismos, como el uso de *tokens*, *cookies* o sesiones que permiten distinguir a los clientes que realizan las peticiones.
- Un conjunto de operaciones bien definidas que se aplican a todos los recursos de información. El protocolo HTTP en sí define un conjunto verbos para realizar operaciones, de los más destacados están: POST, GET, PUT y DELETE. Los cuales tienen una correspondencia con las operaciones básicas que se pueden realizar con los recursos que se manejan en las aplicaciones, como lo son crear, leer, actualizar y borrar.
- Una sintaxis universal para identificar los recursos. En un sistema REST, cada recurso es direccionable únicamente a través de su URI (Identificador Uniforme de Recurso).

Gracias a esta arquitectura de software el cliente puede interactuar con cualquier recurso de información conociendo su URI y realizar cualquier acción requerida sobre el mismo, sin tener que conocer si existen *proxys* o *firewalls*, el cliente debe estar en la capacidad de comprender la información retornada, puesto que generalmente la respuesta viene en formato XML o JSON.

JavaScript

Es el lenguaje interpretado orientado a objetos desarrollado por Netscape que se utiliza en millones de páginas web y aplicaciones de servidor en todo el mundo. JavaScript de Netscape es un superconjunto del lenguaje de scripts estándar de la edición de ECMA-262 3 (ECMAScript) que presenta sólo leves diferencias respecto a la norma publicada (Mozilla Developer Network, s.f.).

Contrariamente a la falsa idea popular, JavaScript no es "Java interpretativo". En pocas palabras, JavaScript es un lenguaje de programación dinámico que soporta construcción de objetos basado en prototipos. La sintaxis básica es similar a Java y C++ con la intención de reducir el número de nuevos conceptos necesarios para aprender el lenguaje.

Es un lenguaje considerado multi-paradigma, basado en prototipos, dinámico, soporta estilos de programación funcional, orientada a objetos e imperativa. Vale la pena mencionar que este lenguaje está en constante evolución pues cuenta con una comunidad de desarrollo bastante grande.

Objeto de Notación JavaScript (JSON)

Es un formato ligero de intercambio de datos. Es fácil de leer y escribir para los seres humanos, así como también es fácil para las máquinas analizar y generar este formato. JSON es un formato de texto que es completamente independiente del lenguaje, pero utiliza convenciones que son familiares a los programadores de la familia de lenguajes C, incluyendo C, C++, C#, Java, JavaScript, Perl, Python y muchos otros. Estas propiedades hacen de JSON un lenguaje de intercambio de datos ideal.

JSON está construido sobre dos estructuras:

- Una colección de pares nombre / valor. En varios lenguajes, esto es considerado como un objeto, registro, estructura, diccionario, tabla hash, lista con clave o arreglo asociativo.
- Una lista ordenada de valores. En la mayoría de los lenguajes, esto es conocido como un arreglo, vector, lista o secuencia.

Estas son estructuras de datos universales. Prácticamente todos los lenguajes de programación modernos las soportan de una forma u otra. Tiene sentido que un formato de datos que sea intercambiable con los lenguajes de programación también se base en estas estructuras (JSON, 2008).

Android

Android es un sistema operativo para teléfonos que se basa en el *kernel* o núcleo de Linux. Utiliza una máquina virtual personalizada diseñada para optimizar recursos de memoria y hardware en un entorno móvil. Android es de código abierto; Puede ampliarse para incorporar nuevas tecnologías de vanguardia a medida que surjan. La plataforma seguirá evolucionando a medida que la comunidad de desarrolladores trabaje para crear aplicaciones móviles innovadoras.

Proporciona acceso a una amplia gama de librerías útiles y herramientas que se pueden utilizar para crear aplicaciones ricas (Alliance, 2017).

Las aplicaciones de Android se escriben en lenguaje de programación Java. Las herramientas de Android SDK compilan tu código, junto con los archivos de recursos y datos, en un APK (Android Package, Paquete de Android), que es un

archivo de almacenamiento con el sufijo .apk. Un archivo de APK incluye todos los contenidos de una aplicación de Android y es el archivo que usan los dispositivos con tecnología Android para instalar la aplicación (Android, 2017).

Sistema de Control de Versiones

El control de versiones es un sistema que registra los cambios realizados sobre un archivo o conjunto de archivos a lo largo del tiempo, de modo que puedas recuperar versiones específicas más adelante.

Permite revertir archivos a un estado anterior, revertir el proyecto entero a un estado anterior, comparar cambios a lo largo del tiempo, ver quién modificó por última vez algo que puede estar causando un problema, quién introdujo un error y cuándo. (Git - Acerca del control de versiones, s.f.)

2.2. Tecnologías del lado del cliente

Lenguaje de Marcado de Hipertexto HTML

HTML, que significa Lenguaje de Marcado para Hipertextos (*HyperText Markup Language*) es el elemento de construcción más básico de una página web y se usa para crear y representar visualmente una página web. Determina el contenido de la página web, pero no su funcionalidad.

HTML es el lenguaje que describe la estructura y el contenido semántico de un documento web. El contenido dentro de una página web es etiquetado con elementos HTML como , <title>, <p>, <div>, y así sucesivamente. Estos elementos conforman los bloques de construcción de un sitio web. (Mozilla Developer Network, s.f.)

Hojas de Estilo en Cascada CSS

Hojas de Estilo en Cascada CSS (*Cascading Style Sheets*) es el lenguaje utilizado para describir la presentación de documentos HTML. CSS describe como debe ser renderizado el elemento estructurado en pantalla, en papel, hablado o en otros medios. (Mozilla Developer Network, s.f.)

jQuery

jQuery es una librería de JavaScript rápida, pequeña y con una gran cantidad de funcionalidades. Permite simplificar la manipulación y navegación de documentos HTML, manejo de eventos, y agregar interacción con la técnica AJAX² a través del

² **JavaScript Asíncrono y XML (AJAX)** no es una tecnología por sí misma, es un término que describe un nuevo modo de utilizar conjuntamente varias tecnologías existentes. Permite que las aplicaciones web capaces de actualizarse continuamente sin tener que volver a cargar la página completa.

uso de una API³ que es compatible con los navegadores más utilizados. (jQuery, s.f.)

Bootstrap

Bootstrap es un *framework* muy popular de HTML, CSS, and JavaScript para el desarrollo de sitios web adaptables a distintos dispositivos. Bootstrap logra que el proceso de desarrollo sea más rápido y sencillo, gracias al uso de sus clases de CSS predefinidas. (Otto, s.f.)

Sass

Sass es una extensión de CSS que le añade funcionalidades como el uso de variables, reglas anidadas, mixins⁴, funciones, y más. Sass ayuda a mantener las hojas de estilo de gran tamaño bien organizadas, y a las pequeñas funcionando de manera rápida. (Sass-lang, s.f.)

Características:

- Compatible con CSS.
- Extensión del lenguaje como variables, anidación y mixins.
- Muchas funciones útiles para manipular el color y otros valores.
- Características avanzadas como el control de directivas para librerías.
- Salida con un formato personalizable.

AngularJs

AngularJs brinda la posibilidad de escribir aplicaciones web del lado del cliente. Permite usar HTML como lenguaje de plantilla y permite extender su sintaxis para expresar los componentes de la aplicación de manera clara. Permite sincronizar automáticamente los datos de las vistas con sus objetos JavaScript a través de dos vías de enlace de datos para ayudar a estructurar la aplicación. (Angularjs, s.f.)

³ La **interfaz de programación de aplicaciones**, abreviada como **API** (del inglés: *Application Programming Interface*), es el conjunto de subrutinas, funciones y procedimientos (o métodos, en la programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro *software* como una capa de abstracción.

⁴ Un **mixin** es una sección de código que contiene código Sass. Cada vez que se llama un mixin en el proceso de conversión el contenido del mismo es insertado en el lugar de la llamada.

2.3. Tecnologías del lado del servidor

Node.js

Node.js es un entorno de ejecución para JavaScript construido con el motor de JavaScript V8 de Chrome. Node.js usa un modelo de operaciones E/S sin bloqueo y orientado a eventos, que lo hace liviano y eficiente (Node.js, 2017).

Node está diseñado para construir aplicaciones en red escalables, una aplicación simple de node puede manejar muchas conexiones concurrentes, pero se diferencia del modelo de concurrencia más común hoy en día, donde se usan hilos del Sistema Operativo. Las operaciones de redes basadas en hilos son relativamente ineficientes y son muy difíciles de usar. Además, los usuarios de Node están libres de preocupaciones sobre el bloqueo del proceso, ya que no existe. Casi ninguna función en Node realiza E/S directamente, así que el proceso nunca se bloquea. Debido a que no hay bloqueo es muy razonable desarrollar sistemas escalables en Node.

Solo porque Node esté diseñado sin hilos, no significa que no pueda aprovechar los múltiples núcleos de un sistema. Los procesos hijos pueden ser ejecutados usando el API `child_process.fork()`, la cual está diseñada para comunicarse fácilmente con el proceso principal. Construida sobre la misma interfaz está el módulo `cluster`, el cual incluso permite compartir sockets entre procesos para activar el balanceo de cargas en sus múltiples núcleos.

Es importante aclarar que cuando se dice que Node.js es no bloqueante, significa que mientras se realiza una operación de E/S y con estas operaciones hablamos principalmente de interacciones con el disco del sistema y redes (Libuv, 2017), se puede seguir ejecutando código JavaScript, todo esto gracias al ciclo de eventos que maneja Node.js, el cual le permite colocar en una cola todos los eventos, dado que la mayoría de los kernels modernos son de múltiples hilos, pueden manejar múltiples operaciones ejecutándose en segundo plano. Cuando una de estas operaciones se completa, el kernel le dice a Node.js para que este de forma apropiada ejecute otro evento de la cola.

Express

Es un marco de trabajo minimalista, rápido y flexible para realizar aplicaciones en Node.js, Express no es dogmático, es decir que permite el desarrollador tome sus propias decisiones al momento de programar con este marco de trabajo, puesto que no será restringido por ningún tipo de dogma impuesta por Express, por otro lado también proporciona un conjunto robusto de funciones para aplicaciones web y móviles (Expressjs, 2017).

Express cuenta con un gran número de librerías y métodos para manipular las peticiones HTTP que permiten la creación de API's robustas de forma rápida y sencilla.

La comunidad de Node.js y Express es tan grande, que estas tecnologías no paran de evolucionar, permitiendo que los programadores puedan hacer más cosas con sus aplicaciones de una forma bastante comprensible.

MongoDB

Es una base de datos orientada a documentos, con una buena escalabilidad y flexibilidad, que incluye unos métodos sencillos para hacer consultas sobre los datos.

MongoDB almacena los datos en documentos, estos son como objetos de JavaScript (JSON), lo que significa que los campos de estos documentos pueden variar entre ellos, así como su estructura puede cambiar con el tiempo, manipular estos documentos es muy sencillo gracias a que MongoDB les hace una correspondencia a objetos dentro de la aplicación.

En cuanto a las formas de acceder y analizar los datos, MongoDB ofrece consultas eficientes, indexación y distintas maneras de agrupar datos. Otra característica que vale la pena resaltar de esta base de datos, es que tiene un núcleo distribuido que le permite escalar de forma horizontal, además de su alta disponibilidad.

Es importante destacar que MongoDB es una base de datos de código abierto, que cuenta con una gran comunidad de desarrolladores, que pueden hacer uso de esta en más de 10 diferentes lenguajes (MongoDB, 2017).

Modelado de datos en MongoDB

Los datos en MongoDB tienen un esquema flexible, a diferencia de las bases de datos relacionales, donde se debe tener un esquema para las tablas antes de insertar datos en ellas.

Al momento de modelar los datos hay que tener presentes distintos factores, el cómo se mostrará la data en la aplicación y qué tanto va a variar esta en el tiempo, existen dos formas de representar las relaciones en los documentos: de forma referenciada o embebida.

Referencia

Al hacer usar referencia, se almacenan las relaciones entre datos agregando enlaces o referencias de un documento a otro. Las aplicaciones pueden resolver estas referencias para acceder a los datos relacionados. En términos generales, estos son modelos de datos normalizados (MongoDB, 2017).

Detalles a considerar al momento de hacer uso de referencias en documento:

- Si varios documentos, tienen datos en común que podrían cambiar, es conveniente separar esos datos en común en documento distinto, ya que cuando estos cambien, es mejor actualizar un solo documento a tener que actualizar múltiples, con esto se puede garantizar consistencia en los datos.

- Hacer mucho uso de referencia, puede hacer que la base de datos no tenga el rendimiento esperado, puesto que para encontrar datos que son referenciados al menos se deben hacer dos consultas.

Embebido

Los documentos embebidos mantienen las relaciones entre datos almacenando datos relacionados en un solo documento estructurado. Los documentos de MongoDB permiten integrar estructuras de documentos en un campo o arreglo de otro documento. Estos modelos de datos desnormalizados permiten a las aplicaciones obtener y manipular datos relacionados en una sola operación de base de datos (MongoDB, 2017).

Detalles a considerar al momento de usar documentos embebidos:

- Cuando varios documentos, tienen datos en común que no suelen cambiar, es conveniente embeber esos datos en un solo documento, de esa manera garantizamos que todas la operaciones realizadas en esos documentos serán atómicas, a pesar de que esto genera datos duplicados se le da poco valor por los beneficios brinda en rendimiento.
- Las bases de datos orientadas a documentos, principalmente fueron diseñadas para evitar el uso de los *joins* de las bases de datos relacionales, es por ello que al modelar datos lo ideal es colocar todo lo posible en un solo documento, teniendo en cuenta que estos tienen un límite de 16MB.

El cómo modelar los datos es un tópico muy discutido entre los programadores, es por ello hay que entender que MongoDB busca tener un balance entre consistencia y eficiencia, con los distintas formas de modelar datos que ofrece, queda de parte de los programadores y las necesidades que tengan sus aplicaciones, hacer un uso inteligente las opciones que proporciona MongoDB.

Amazon Elastic Compute Cloud (Amazon EC2)

Es un servicio web de Amazon Web Services (AWS) que proporciona una capacidad de cómputo segura y redimensionable en la nube.

Amazon EC2 proporciona un control completo de los recursos informáticos y reduce el tiempo necesario para obtener e iniciar nuevas instancias de servidor a minutos, lo que le permite aumentar rápidamente la capacidad, tanto hacia arriba como hacia abajo, a medida que cambian las necesidades de cómputo de nuestras aplicaciones alojadas en las instancias (Amazon, 2017).

Compose

Es un servicio web que permite desplegar, monitorear, administrar y consultar bases de datos en la nube, a través de una sencilla interfaz. Además que permite agregar usuarios y establecer permisos, agregar nuevos nodos y acceder a los archivos de monitoreo diario con facilidad.

La asignación de recursos se ajusta automáticamente a medida que crece la base de datos. También se pueden ajustar manualmente las instancias para asignar recursos adicionales (Compose, 2017).

Mandrill

Es un API de correo electrónico transaccional. Es fiable e ideal para enviar correos electrónicos con plantillas personalizadas, posee una extensa documentación con ejemplos de cómo se debe integrar en distintos lenguajes de programación. Adicionalmente, Mandrill monitorea cada uno de los correos electrónicos que envía, permitiendo saber si el mismo llegó, cuantas veces el correo ha sido abierto, entre otras cosas (Mandrill, 2017).

2.4. Herramientas de Apoyo al desarrollo de Software

Gulp

Gulp es un sistema de construcción que permite automatizar tareas comunes de desarrollo, tales como la reducción de código JavaScript, recarga del navegador, compresión de imágenes, validación de sintaxis de código, entre otras tareas. Está construido con JavaScript, funciona sobre Node.js y es de código abierto. (Sánchez, 2014)

Git

Git es un sistema de control de versiones gratis y de código abierto, diseñado para manejar desde pequeños hasta grandes proyectos con eficiencia. (Git-scm, s.f.)

RoboMongo

Es una herramienta que brinda una forma más fácil de explorar y manipular los datos en MongoDB, cuenta con una interfaz muy limpia que permite visualizar los documentos almacenados. Se pueden ejecutar consultas en segundos, interactuar con los documentos con funcionalidades CRUD completas. Además de Ver y optimizar el rendimiento de la consulta (Robomongo, 2017).

Swagger UI

Swagger UI es un marco de trabajo que permite visualizar e interactuar con los recursos de las API's sin tener que hacer una implementación, pues genera automáticamente una documentación que facilita el consumo del lado del cliente (Swagger, 2017).

Las tecnologías seleccionadas fueron de gran ayuda para este desarrollo, es importante resaltar que la curva de aprendizaje de las mismas es alta, por lo que se recomienda primero aprender a programar con JavaScript, entender cómo funciona y trabaja este lenguaje de programación, para luego una vez que se tenga dominio, empezar a trabajar con los *frameworks* Express y AngularJs, los cuales poseen una

excelente documentación y cuentan con grandes comunidades de desarrollo que los apoyan.

Es importante resaltar que los servicios de Mandrill, Compose y AWS; hacen que las aplicaciones sean altamente escalables, ya que permiten añadir capacidad de cómputo o espacio de almacenamiento en el momento que sea requerido. Sin embargo, estos servicios son privados y pueden llegar a generar grandes gastos, por lo que se recomienda hacer uso de estos servicios cuando el desarrollo este por terminar. En igual forma, todos los servicios mencionados pueden ser usados de forma gratuita por un corto periodo de tiempo, esto es de gran ayuda cuando se desea empezar a hacer uso de los mismos.

CAPÍTULO III. Marco Metodológico

Para evitar que el desarrollo de software se convierta en una actividad caótica es conveniente hacer uso de una metodología de desarrollo, para asegurar el éxito del mismo, la metodología de desarrollo nos provee de una dirección a seguir para la producción de un software completo y de calidad.

Por esta razón, es de vital importancia seleccionar la metodología adecuada, cuya selección dependerá del tipo de software a implementar junto con sus requerimientos.

En este capítulo se describe la metodología de desarrollo tomando el enfoque hacia metodologías ágiles, en particular la metodología de desarrollo Scrum.

3.1 Metodología de Desarrollo de Software

La Metodología para el desarrollo de software es un modo sistemático de realizar, gestionar y administrar un proyecto, de manera que pueda llevarse a cabo con altas posibilidades de éxito, esta comprende los procesos a seguir para idear, implementar y mantener un producto de software, desde que surge la necesidad hasta que se cumple el objetivo.

Según (CMS, 2008) Una metodología de desarrollo de software se refiere a un marco de trabajo que es usado para estructurar, planear y controlar el proceso de desarrollo en sistemas de información.

Una sola metodología puede no resultar adecuada para todos los desarrollos de software, tomando en cuenta consideraciones técnicas, organizacionales, de proyecto y de equipo, por este motivo se han generado una gran variedad de metodologías, en donde alguna más que otra se ajusta a los requerimientos.

Según su filosofía de desarrollo se pueden clasificar en dos grandes grupos:

- **Las metodologías orientadas al control de los procesos (Metodologías Tradicionales o Pesadas):**

Se establecen rigurosamente las actividades a desarrollar, herramientas a utilizar y notaciones que se usarán.

- **Las metodologías orientadas a la interacción con el cliente y el desarrollo incremental del software (Metodologías Ligeras o Ágiles):**

Se muestran versiones parcialmente funcionales al cliente, en intervalos cortos de tiempo, para que pueda evaluar y sugerir cambios en el producto según se va desarrollando.

Metodologías Ágiles	Metodologías Tradicionales
Basadas en heurísticas provenientes de prácticas de producción de código	Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo
Especialmente preparados para cambios durante el Proyecto	Cierta resistencia a los cambios
Impuestas internamente (por el equipo)	Impuestas externamente
Proceso menos controlado, con pocos principios	Proceso mucho más controlado, con numerosas políticas/normas
El cliente es parte del equipo de desarrollo	El cliente interactúa con el equipo de desarrollo mediante reuniones
Grupos pequeños (<10 integrantes) y trabajando en el mismo sitio	Grupos grandes y posiblemente distribuidos
Pocos artefactos	Más artefactos
Pocos roles	Más roles
Menos énfasis en la arquitectura del software	La arquitectura del software es esencial y se expresa mediante modelos

Figura 3. Diferencias entre Metodologías Ágiles y no Ágiles
Fuente: (Canós, s.f)

Actualmente, los proyectos suelen desarrollarse bajo exigencias cambiantes en los que, con frecuencia, se reducen los tiempos de desarrollo manteniendo una gran calidad. En este escenario, las Metodologías Ágiles constituyen una solución ajustada a medida, aportando una elevada simplificación que, a pesar de ello, no renuncia a las prácticas esenciales para asegurar la calidad del producto. A continuación detallaremos lo que contienen estas metodologías.

3.2 Metodologías Ágiles

Las metodologías ágiles son un conjunto de métodos de Ingeniería de Software, que se basan en el desarrollo iterativo e incremental, teniendo presente cambios y respondiendo a los mismos, ya que al final de cada iteración se evalúan las prioridades del proyecto en aras de cumplir los objetivos, durante todo el proceso de desarrollo se cuenta con un grupo de desarrolladores auto-organizados y multidisciplinares.

En la creación del modelado ágil se establecieron los siguientes valores:

- Individuos e interacciones sobre procesos y herramientas.
- Software funcionando sobre documentación extensiva.
- Colaboración con el cliente sobre negociación contractual.
- Respuesta ante el cambio sobre seguir un plan.

Los valores anteriores inspiran doce principios que son características que diferencian un proceso ágil de uno tradicional. Los dos primeros principios son generales y resumen gran parte del espíritu ágil. El resto tienen que ver con el proceso a seguir y con el equipo de desarrollo, en cuanto metas a seguir y organización del mismo.

Los principios son:

1. La prioridad es satisfacer al cliente mediante tempranas y continuas entregas de software que le aporten un valor.
2. Dar la bienvenida a los cambios. Se capturan los cambios para que el cliente tenga una ventaja competitiva.
3. Entregar frecuentemente software que funcione, la frecuencia va desde un par de semanas a un par de meses, con el menor intervalo de tiempo posible entre entregas.
4. La gente del negocio y los desarrolladores deben trabajar juntos a lo largo del proyecto.
5. Construir el proyecto en torno a individuos motivados. Darles el entorno y el apoyo que necesitan y confiar en ellos para conseguir la finalización del trabajo.
6. El diálogo cara a cara, como el método más eficiente y efectivo para comunicar información dentro de un equipo de desarrollo.
7. El software que funciona es la medida principal de progreso.

8. Los procesos ágiles promueven un desarrollo sostenible. Los promotores, desarrolladores y usuarios deberían ser capaces de mantener una paz constante.
9. La atención continua a la calidad técnica y al buen diseño mejora la agilidad.
10. La simplicidad es esencial.
11. Las mejores arquitecturas, requisitos y diseños surgen de los equipos organizados por sí mismos.
12. En intervalos regulares, el equipo reflexiona respecto a cómo llegar a ser más efectivo, y según esto ajusta su comportamiento.

Varias metodologías encajan bajo la clasificación de ágiles, todas ellas comparten muchas características y también hay algunas diferencias significativas. De estas, se seleccionó la metodología de desarrollo Scrum.

3.3 Justificación para la selección de Scrum

Esta metodología fue seleccionada porque es sencilla de usar, ofrece gran valor añadido sin demasiado esfuerzo, centrándose en la productividad a través de iteraciones cortas, el valor del equipo, la gestión de las estimaciones y la relación con el cliente.

Manejar el concepto de entregas parciales es bastante beneficioso dado que se obtienen comentarios y observaciones por parte del cliente, los cuales son importantes tener en cuenta para el correcto desarrollo del proyecto.

Scrum está especialmente indicado para proyectos en entornos complejos, que requieren ser hechos en menor tiempo, donde los requisitos del cliente son cambiantes y donde la innovación, la competitividad y la productividad son fundamentales.

A continuación se describen los fundamentos de esta metodología.

3.4 Scrum

Es una metodología ágil y flexible de gestión de proyectos cuyo objetivo primordial es desarrollar primero las funcionalidades de mayor importancia para el cliente y elevar al máximo la productividad de un equipo, fue desarrollada por Jeff Sutherland y elaborada más formalmente por Ken Schwaber.

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

definidos, donde la innovación, la competitividad, la flexibilidad y la productividad son fundamentales. (Proyectos Ágiles, 2015)

3.4.1 Roles

El equipo Scrum está compuesto por el Dueño de Producto, el Equipo de Desarrollo y el Scrum Master. Tiene como características principales ser un equipo autoorganizado y multifuncional, no son dirigidos por personas externas las decisiones son tomadas por el equipo, al ser multifuncional el equipo cuenta con las competencias necesarias para llevar a cabo el trabajo sin depender de otras personas que no son parte del equipo. El modelo de equipo en Scrum está diseñado para optimizar la flexibilidad, la creatividad y la productividad (Schwaber, 2013).

Dueño del Producto

El Dueño de Producto es la única persona responsable de gestionar la Lista del Producto. La gestión de la Lista del Producto incluye:

- Expresar claramente los elementos de la Lista del Producto.
- Ordenar los elementos en la Lista del Producto para alcanzar los objetivos de la mejor manera posible.
- Optimizar el valor del trabajo desempeñado por el Equipo de Desarrollo.
- Asegurar que la Lista del Producto es visible, transparente y clara para todos, y que muestra aquello en lo que el equipo trabajará a continuación.
- Asegurar que el Equipo de Desarrollo entiende los elementos de la Lista del Producto al nivel necesario.
- Revisa junto con otros involucrados el producto final en cada Sprint.

El Dueño del Producto podría hacer el trabajo anterior, o delegarlo en el Equipo de Desarrollo. Sin embargo, en ambos casos el Dueño de Producto sigue siendo el responsable de dicho trabajo.

Equipo de Desarrollo

El Equipo de Desarrollo consiste en los profesionales que desempeñan el trabajo para poder entregar un Incremento de producto que potencialmente se pueda poner en producción, al final de cada Sprint.

Los Equipos de Desarrollo tienen las siguientes características:

- Son autoorganizados. Nadie (ni siquiera el Scrum Master) indica al Equipo de Desarrollo cómo convertir elementos de la Lista del Producto en Incrementos.

- Los Equipos de Desarrollo son multifuncionales, contando como equipo con todas las habilidades necesarias para crear un Incremento de producto.
- Scrum no reconoce subequipos en los equipos de desarrollo.
- Los Miembros individuales del Equipo de Desarrollo pueden tener habilidades especializadas y áreas en las que estén más enfocados, pero la responsabilidad recae en el Equipo de Desarrollo como un todo.

El tamaño del Equipo de Desarrollo es lo suficientemente pequeño como para permanecer ágil y lo suficientemente grande como para completar una cantidad de trabajo significativa, el tamaño ideal puede variar entre 4 a 9 personas dependiendo de la carga que se tenga en la Lista del Producto (Schwaber, 2013).

Scrum Master

Los Scrum Masters se aseguran de que el Equipo Scrum trabaja ajustándose a la teoría, prácticas y reglas de Scrum. Es un líder que está para ayudar a los otros miembros del equipo a desempeñar sus actividades de una manera más óptima.

Es ideal que su perfil sea un balance entre desarrollador y gerente, es el moderador de las ceremonias de Scrum, puede ayudar al Dueño de Producto de varias formas como por ejemplo gestionando la Lista del Producto de una manera más efectiva o haciendo que los elementos de la Lista sean más claros. De la misma manera el Scrum Master debe guiar al Equipo de Desarrollo, eliminar los impedimentos que tengan los mismos para que sus productos sean de alto valor (Schwaber, 2013).

3.4.2 Artefactos

Lista de Producto (*Product Backlog*)

Es el plan que contiene las metas que se deben cumplir en el desarrollo del proyecto, es creado por el dueño del producto y es representado a través de una lista con todos los requerimientos iniciales del cliente, también describe el funcionamiento del negocio. (Schwaber, 2013)

Historias de Usuario

Son los requerimientos del cliente, estas historias tienen puntos de dificultad, puntos de importancia y puntos de aceptación, cuando se planifica un sprint los miembros del equipo de desarrollo toman estas historias de usuario y crean las subtarefas que implican el desarrollo de la misma.

Lista de Requerimientos Pendientes (*Sprint Backlog*)

La Lista de Pendientes del Sprint es el conjunto de requerimientos de la Lista de Producto seleccionados para el Sprint, más un plan para entregar el Incremento de producto y conseguir el Objetivo del Sprint.

La Lista de Pendientes del Sprint muestra todo el trabajo que el Equipo de Desarrollo identifica como necesario para alcanzar el Objetivo del Sprint, esta lista puede cambiar durante el Sprint, esto ocurre a medida que el Equipo de Desarrollo trabaja sobre el plan y aprende más acerca del trabajo necesario para conseguir el Objetivo del Sprint (Schwaber, 2013).

Tablero Scrum

Es un tablero usado para poder visualizar con transparencia el trabajo que se hace en el proyecto, ayuda a enfocarse y observar el flujo de trabajo, es una herramienta muy eficaz para mejorar la productividad del equipo.

3.4.3 Eventos

Sprints

Según (Schwaber, 2013) un Sprint se puede ver como un pequeño proyecto el cual debe lograr un objetivo en específico, este posee un plan para saber cómo se va a construir y tiene un tiempo estimado para su desarrollo no mayor a un mes.

Cuando se finaliza un Sprint se crea un incremento de producto, utilizable y potencialmente desplegable, cada Sprint comienza inmediatamente después de la finalización del Sprint anterior.

Los Sprints provienen de la Reunión de Planificación del Sprint, y contemplan reuniones diarias (*Daily Scrums*), trabajo de desarrollo, Revisión del Sprint (*Sprint Review*), y Retrospectiva del Sprint (*Sprint Retrospective*).

Durante el Sprint:

- No se realizan cambios que puedan afectar al Objetivo del Sprint.
- Los objetivos de calidad no disminuyen.
- El alcance puede ser clarificado y renegociado entre el Dueño de Producto y el Equipo de Desarrollo.
- Su duración son de una a cuatro semanas.
- Todos duran la misma cantidad de tiempo.

El objetivo del Sprint es una meta establecida para el mismo, que puede ser alcanzada mediante la implementación de elementos de la Lista de Producto. Es creado durante la reunión de Planificación del Sprint. Los requerimientos seleccionados de la Lista del Producto ofrecen una funcionalidad coherente. A medida que el Equipo de Desarrollo trabaja, se mantiene el objetivo del Sprint en mente (Schwaber, 2013).

Un incremento es la suma de todos los requerimientos de la Lista de Producto completados durante un Sprint. Al final de un Sprint, el nuevo incremento debe estar "Terminado", lo cual significa que está en condiciones de ser utilizado.

Reunión Diaria

La reunión diaria o *Daily Scrum* se realiza con el fin de mostrar los avances del Sprint, su duración no debería pasar de 15 minutos, su propósito es tener una retroalimentación sobre las tareas en el Sprint y los obstáculos que se presentan en las mismas. En esta reunión se responden preguntas como: ¿Qué has hecho desde el último encuentro? ¿Qué obstáculos hay para cumplir la meta? ¿Qué harás antes del próximo encuentro?

Las reuniones diarias mejoran la comunicación, eliminan la necesidad de mantener otras reuniones, identifican y eliminan impedimentos relativos al desarrollo, resaltan y promueven la toma de decisiones rápida, y mejoran el nivel de conocimiento del Equipo de Desarrollo (Schwaber, 2013).

Reunión de Planificación de Sprint

El trabajo a realizar durante el Sprint se planifica en la Reunión de Planificación de Sprint. Este plan se crea mediante el trabajo colaborativo del Equipo Scrum completo.

La Reunión de Planificación de Sprint tiene un máximo de duración de ocho horas para un Sprint de un mes. Para Sprints más cortos, el evento es usualmente toma menos tiempo. El Scrum Master se asegura de que el evento se lleve a cabo y que los asistentes entiendan su propósito. El Scrum Master enseña al Equipo Scrum a mantenerse dentro del bloque de tiempo.

El Equipo de Desarrollo trabaja para proyectar la funcionalidad que se desarrollará durante el Sprint. El Dueño de Producto discute el objetivo que el Sprint debería lograr y los elementos de la Lista de Producto que, si se completan en el Sprint, se deberían lograr, el equipo Scrum completo colabora en el entendimiento del trabajo del Sprint, una vez que se aclaren las funcionalidades a desarrollar se elabora el Objetivo del Sprint (*Sprint Goal*) (Schwaber, 2013).

Revisión de Sprint

Al final del Sprint se lleva a cabo una Revisión para inspeccionar el incremento sobre el proyecto y adaptar la lista de producto si es necesario. Durante la Revisión de Sprint, el Equipo Scrum comunica lo que se hizo durante el mismo.

El Dueño de Producto explica qué requerimientos de la Lista de Producto se han terminado y cuáles no. El Equipo de Desarrollo habla acerca de qué fue bien durante el Sprint, qué problemas aparecieron y cómo fueron resueltos esos problemas. El grupo completo colabora acerca de qué hacer a continuación, de modo que la Revisión del Sprint proporcione información de entrada valiosa para Reuniones de Planificación de Sprints subsiguientes (Schwaber, 2013).

Retrospectiva de Sprint

Según (Schwaber, 2013) la Retrospectiva de Sprint es una oportunidad para el Equipo Scrum de inspeccionarse a sí mismo y crear un plan de mejoras que sean abordadas durante el siguiente Sprint.

La Retrospectiva de Sprint tiene lugar después de la Revisión de Sprint y antes de la siguiente Reunión de Planificación de Sprint, su propósito es:

- Inspeccionar cómo fue el último Sprint en cuanto a relaciones, procesos y herramientas.
- Identificar y ordenar los requerimientos más importantes que salieron bien y las posibles mejoras.

Crear un plan para implementar las mejoras a la forma en la que el Equipo Scrum desempeña su trabajo.

CAPÍTULO IV. Desarrollo de la Solución

En este capítulo se describen los principales elementos de la solución planteada, las tecnologías, actividades y etapas que fueron realizadas durante su desarrollo, siguiendo las pautas de la metodología Scrum.

4.1. Descripción de la solución

Para abordar la problemática planteada en el capítulo I, se desarrollaron dos aplicaciones, una aplicación móvil y una aplicación web, ambas vinculadas por un API RESTful (ver **Figura 4**). A la solución que integra ambas aplicaciones se denominó Aquiles.

La aplicación web fue desarrollada con las siguientes tecnologías: MongoDB, Express, AngularJs y Node.js. Esta aplicación estará disponible para los usuarios de compañía, administradores de compañía y administradores de aplicación, permitiéndoles gestionar las compañías, sucursales, etiquetas, envíos, mensajeros y reportes, dependiendo de los permisos que tengan sus respectivos roles.

Por otro lado la aplicación móvil fue desarrollada de forma nativa en Android, para que pueda ser ejecutada en dispositivos con la versión 4.4 KitKat en adelante, esta aplicación estará disponible únicamente para los mensajeros, permitiéndoles gestionar los envíos que tienen asignados.

En la **Figura 4** se muestra cómo la aplicación web y la móvil se encuentran integradas al API, donde se encuentra la lógica del negocio, este API se encarga de procesar las peticiones realizadas por los clientes, comunicarse con la base de datos y otros servicios en la nube como por ejemplo Mandrill.

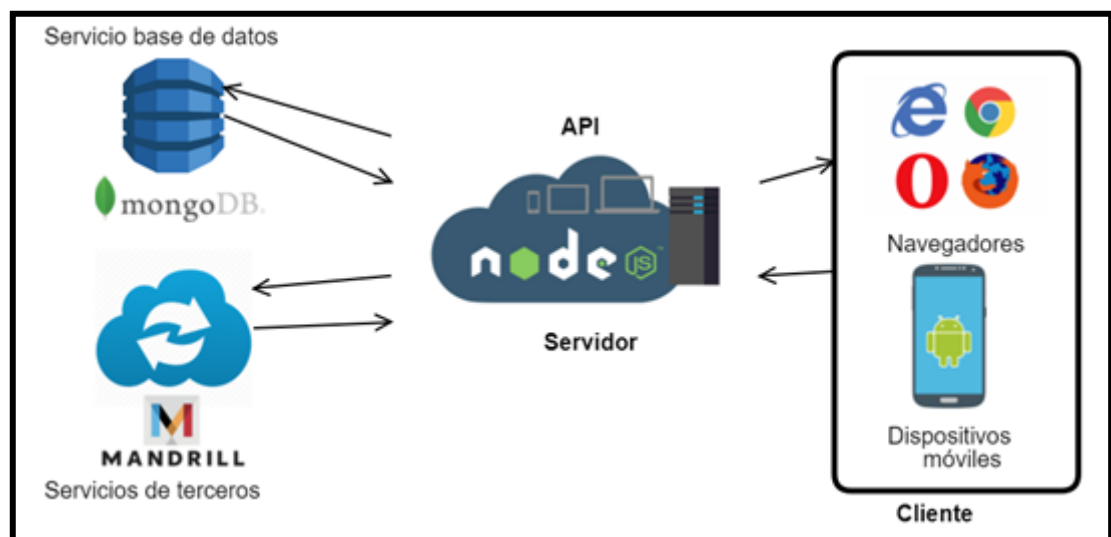


Figura 4. Arquitectura de la solución

A continuación se describe brevemente las tecnologías que fueron escogidas para que este desarrollo fuera posible.

4.2. Tecnologías y lenguajes

- **Aplicación móvil**

Fue desarrollada de forma nativa para Android, para obtener el máximo rendimiento de los dispositivos que cuentan con ese sistema operativo, además de que se contempló que los mensajeros harán uso de dispositivos con este sistema operativo, ya que los mismos son más económicos en el mercado. La aplicación se podrá ejecutar a partir de Android 4.4 KitKat en adelante.

- **Tecnologías en el servidor**

Se optó por hacer uso de: Node.js, Express y MongoDB, existen muchas razones por las que estas tecnologías se ajustan a las necesidades que este desarrollo requiere, entre ellas está la escalabilidad, tanto MongoDB como Node.js permiten que las aplicaciones crezcan de forma horizontal a través de mecanismos como *clusters*, ambas herramientas tienen un alto rendimiento en la forma como desempeñan sus tareas, además que cuentan con una comunidad de desarrollo muy grande que siempre está aportando mejoras.

- **Tecnologías en el lado del cliente**

Como marco de trabajo principal en el lado del cliente se eligió Angular.js, ya que cuenta con: Directivas que definen comportamientos dentro del DOM (*Document Object Model*, Modelo en Objetos para la Representación de Documentos), vínculos en dos direcciones entre la vista y modelo, que están en relación constante, gracias a esto, se logra que todo cambio visual, se actualice a tiempo real en el modelo y viceversa. Este tipo de características hacen de AngularJs una buena elección. Además, se usaron otras tecnologías de apoyo para el desarrollo, tales como Bootstrap para poder organizar los elementos dentro de las interfaces de una forma sencilla y JQuery que provee una gran cantidad de funcionalidades que son de ayuda para la manipulación del DOM. Es importante destacar que, al hacer uso de estos *frameworks* y librerías, se está haciendo uso de HTML5, CSS y JavaScript que son las tecnologías principales para el desarrollo aplicaciones web.

- **Documentación del API**

Se eligió Swagger como herramienta para documentar el API, ya que genera una interfaz donde se puede ver la descripción de cada ruta del API, así como probarlas directamente antes de integrarse a ellas, es muy sencillo generar estas vistas, pues son creadas a partir de comentarios en el código.

- **Plataforma para envío de correos electrónicos**

El manejo de notificaciones por correo electrónico estará integrado a Mandrill, una plataforma potente para enviar correos electrónicos personalizados, permitiendo la creación de correos con plantilla que pueden ser integrados con facilidad al código.

- **Servicios web**

Para el pase a producción se decidió hacer uso de AWS (Amazon Web Service) de forma que la aplicación web se ejecutara desde una instancia de AWS en la nube. Por otro lado se usaran los servicios de Compose, para alojar la base de datos. Vale destacar que tanto Compose como AWS permiten escalar de forma horizontal y vertical de una forma muy sencilla.

4.3 Perfiles de usuario

- **Administrador de aplicación**

Usuarios que necesitan gestionar las compañías y sucursales de los clientes corporativos, y además gestionar las etiquetas que son asignadas a las compañías. Los usuarios de este perfil podrán:

- Gestionar las compañías a las cuales se les va a brindar el servicio de envíos.
- Administrar los lotes de etiquetas que son asignadas a las compañías.
- Visualizar las facturas que son generadas a cada compañía.
- Gestionar los mensajeros que atenderán a las distintas compañías.
- Visualizar reportes sobre los mensajeros y el uso de las etiquetas.

- **Administrador de compañía**

Usuarios que se encargan de gestionar dentro de su compañía. Los usuarios de este perfil podrán:

- Visualizar los reportes asociados a los mensajeros y etiquetas de la compañía.
- Administrar los usuarios que pertenecen a la compañía.
- Gestionar sus envíos dentro de la compañía.
- Asignar envíos a los mensajeros.
- Visualizar los mensajeros que están asignados a la compañía.

- **Usuario de compañía**

Este perfil es para los empleados que pertenecen a una compañía a la cual se le presta el servicio de envíos. Los usuarios con este perfil podrán:

- Gestionar sus envíos con otros usuarios que formen parte del directorio de su compañía.

- **Mensajero**

Este perfil le corresponde al usuario que cumple el rol mensajero, cuyo trabajo consiste en transportar los envíos que le son asignados mediante la aplicación web, estos perfiles son gestionados por el administrador de la aplicación y son los únicos que hacen uso de la aplicación móvil. Los usuarios con este perfil podrán:

- Gestionar los envíos que están asociados a él.
- Verificar el código de una etiqueta mediante un escáner.

4.4 Principales procesos de negocio

4.4.1 Proceso de asignación de etiquetas a compañías y sucursales

En la **figura 5**, se representa el proceso de asignación de etiquetas a compañías y sucursales, el cual pertenece a la aplicación web, donde los administradores de la aplicación asignan lotes de etiquetas para las compañías, con estas etiquetas suministradas se mantiene un control sobre los envíos que se realizan dentro de las compañías y sucursales.

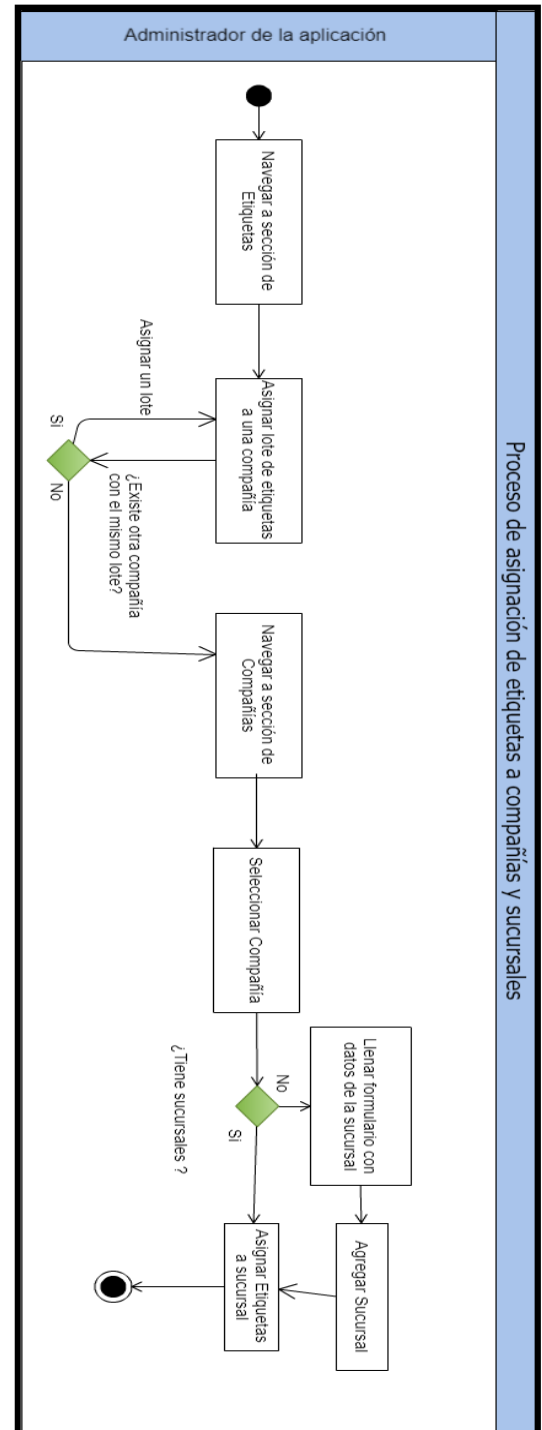


Figura 5. Proceso de asignación de etiquetas a compañías y sucursales

4.4.2 Proceso de invitación a nuevos usuarios

En la **figura 6**, se representa el proceso de invitación a nuevos usuarios, el cual pertenece a la aplicación web, se puede observar que en este proceso el administrador de la compañía, inicia el proceso llenando el formulario para crear nuevos usuarios, donde debe indicar información como el rol y los permisos que tendrá el nuevo usuario, una vez finalizado el formulario el usuario invitado recibirá un correo electrónico, el cual le permitirá completar su registro en la aplicación.

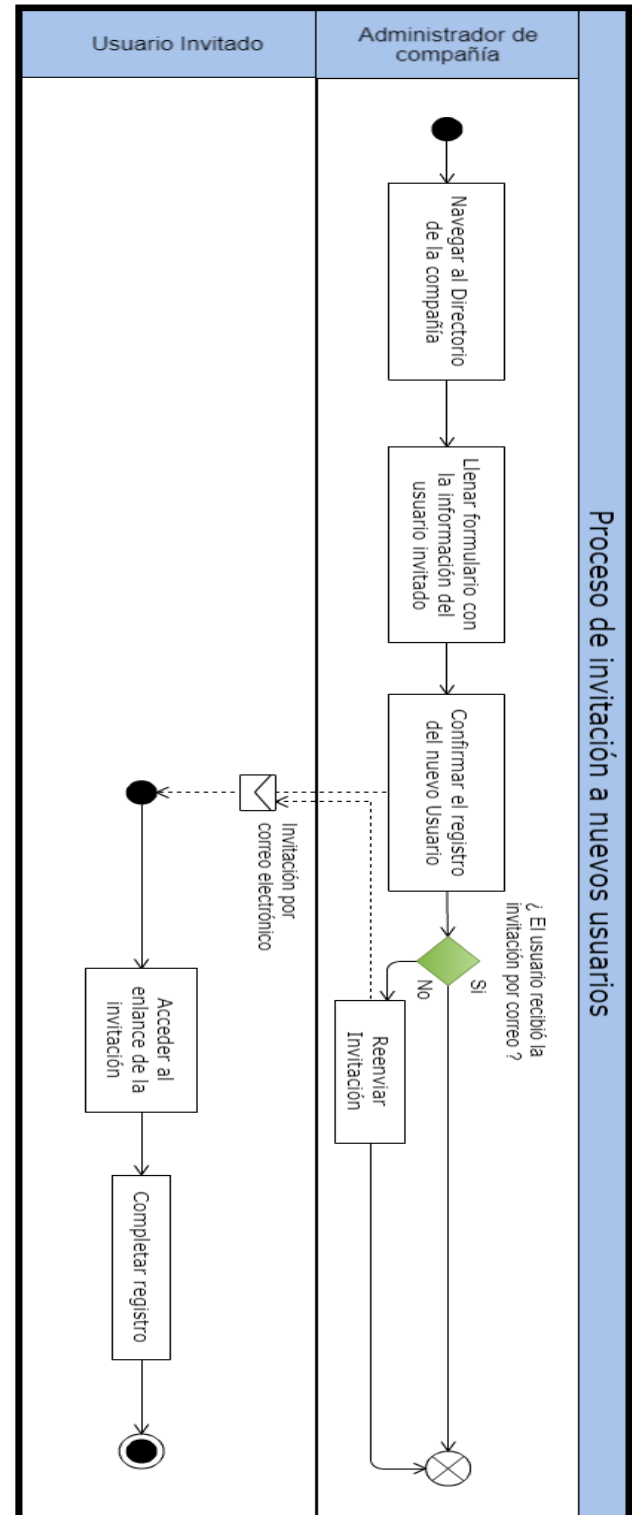
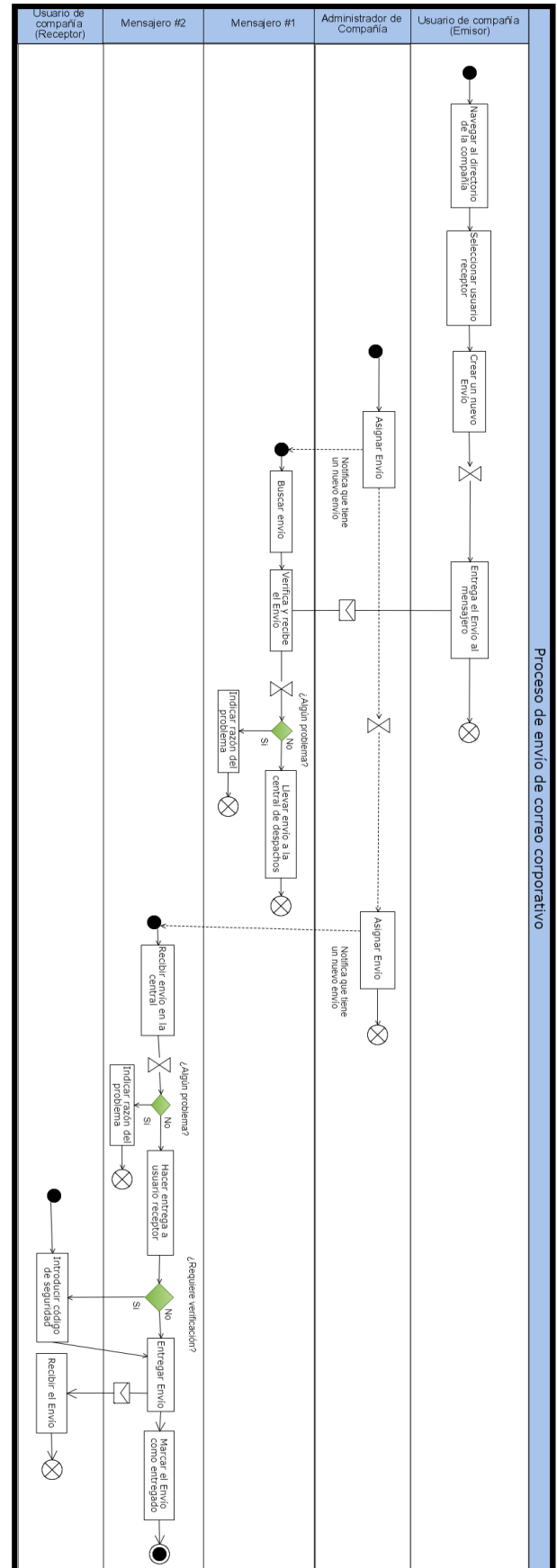


Figura 6. Proceso de asignación de etiquetas a compañías y sucursales

4.4.3 Proceso de envío de correo corporativo

En la **figura 7**, se representa el proceso de envío de correo corporativo, el cual involucra la aplicación web y la móvil. En este proceso se puede observar como un usuario de compañía crea un envío, luego el administrador de compañía se encarga de asignar el envío a un mensajero, para que este lo busque y verifique, posteriormente este procede a transportarlo hacia la central de despachos, luego el administrador de compañía asigna a un mensajero para que lleve el envío a su destino final, finalmente el mensajero entrega el envío al usuario final.

Figura 7. Proceso de envío de correo corporativo



4.5 Aplicación de la metodología Scrum

En este punto se encuentra documentado cada paso que se siguió para el desarrollo del proyecto. Se siguieron los lineamientos de Scrum, lo cual permitió que el desarrollo y evolución del proyecto fuera de manera organizada, eficiente y eficaz.

Cada iteración (*Sprint*) tuvo una duración de dos semanas y cada incremento fue probado durante el desarrollo.

4.5.1 Objetivos de los Sprints (*Sprints Goals*)

Iteración (<i>Sprint</i>)	Objetivos
1	Diseño de la base de datos e implementación del modelos de datos.
2	Implementación de registro de usuarios y control de accesos por permisos.
3	Implementación de sección de compañías y sucursales.
4	Implementación de sección de directorio de usuarios.
5	Implementación de sección de etiquetas.
6	Implementación de sección para el manejo de envíos.
7	Implementación de sección de mensajeros en la aplicación web e implementación de sección de escáner y envíos en la aplicación móvil.
8	Implementación de sección de reportes.
9	Pase a producción y corrección de errores.

Tabla 1. Objetivos de los Sprints (*Sprints Goals*)

4.5.2 Iteraciones

Iteración 1

Diseño de la base de datos e implementación del modelo de datos.

Se diseñó una base de datos que representara la información que se deseaba almacenar junto con sus relaciones, luego de terminar el diseño se procedió a implementar un modelo de datos, que hiciera correspondencia con las colecciones

definidas para la base datos, en donde se agregaron las validaciones y métodos correspondientes a cada modelo.

Iteración 2

Implementación de registro de usuarios y control de accesos por permisos.

Se desarrollaron en el API las rutas correspondientes para el proceso de registro, tanto en la aplicación móvil como en la aplicación web se realizaron las vistas correspondientes y se integraron con las rutas del API. También se implementó un sistema de seguridad usando *tokens*, el cual permitió restringir funcionalidades según los permisos del usuario.

Iteración 3

Implementación de sección de compañías y sucursales.

Se desarrollaron las rutas que permitieran gestionar las compañías y las sucursales, estas fueron restringidas a los usuarios que no tuviesen el rol de administrador de la aplicación, luego se desarrollaron las vistas de estas secciones en la aplicación web y se integraron con las rutas del API

Iteración 4

Implementación de sección de directorio de usuarios.

Para que los administradores de las compañías pudieran gestionar el directorio de usuarios, se desarrolló una sección en donde se pueden invitar a nuevos usuarios, agregar varios usuarios mediante un archivo con formato CSV, además de la funcionalidad de poder editar y eliminar los usuarios ya existentes.

Para los usuarios de las compañías, se desarrolló una sección donde en podrán visualizar a los otros usuarios pertenecientes al directorio, en donde tendrán la capacidad de hacer envíos a estos, también se desarrolló una funcionalidad para agregar usuarios a una lista de favoritos, para que de esta manera los usuarios que tengan contacto frecuente puedan ser ubicados de una manera más rápida.

Iteración 5

Implementación de sección de etiquetas.

En esta iteración se desarrolló una sección para administrar las etiquetas, a la cual solo tendrán acceso los administradores de la aplicación, en esta sección los administradores podrán asignar etiquetas a las compañías y a sus correspondientes sucursales.

Iteración 6

Implementación de sección para el manejo de envíos.

Se desarrollaron las rutas en el API que permitieran gestionar los envíos que los usuarios de la compañía realizan, a su vez se implementaron las vistas correspondientes a esta sección y posteriormente se integraron a las rutas desarrolladas en el API, en esta sección los usuarios podrán visualizar el estado de sus envíos, así como también podrán asignar mensajeros a los envíos que desean hacer.

Iteración 7

Implementación de sección de mensajeros en la aplicación web e implementación de sección de escáner y envíos en la aplicación móvil.

Se desarrollaron las rutas necesarias para gestionar los mensajeros y luego se integraron con las vistas de la aplicación web, luego se implementó la funcionalidad de escáner, para que los mensajeros puedan escanear los código que poseen las etiquetas de los envíos, también se implementó una vista en la aplicación móvil que permite visualizar los envíos asignados a un mensajero.

Iteración 8

Implementación de sección de reportes.

En esta iteración se desarrolló la sección de reportes, la cual solo está habilitada para usuarios administradores, en esta sección se podrán visualizar datos de interés, tales como: compañías activas, compañías inactivas, etiquetas asignadas, etiquetas usadas y mensajeros activos.

Iteración 9

Pase a producción y corrección de errores.

En esta iteración se migró el API y la aplicación web a una instancia de Amazon Web Service, se instalaron las dependencias necesarias para ejecutar la aplicación, se creó un dominio para la misma y se creó una instancia de MongoDB en Compose para alojar la base de datos, luego de esto se hizo una última revisión para verificar el correcto funcionamiento de las aplicaciones para luego corregir los errores detectados.

4.6 Casos de uso más importantes

En esta sección se presenta desde la **Figura 8** hasta la **Figura 24** los diagramas y explicación de los casos de uso más importantes contenidos en el sistema, los cuales especifican la comunicación y el comportamiento de la aplicación mediante su interacción con los usuarios.

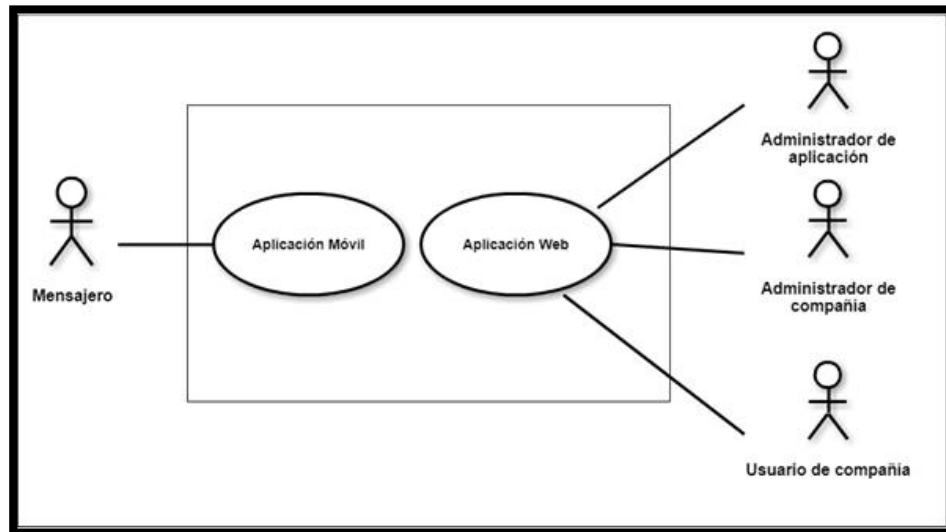


Figura 8. Diagrama de casos de uso. Nivel 0

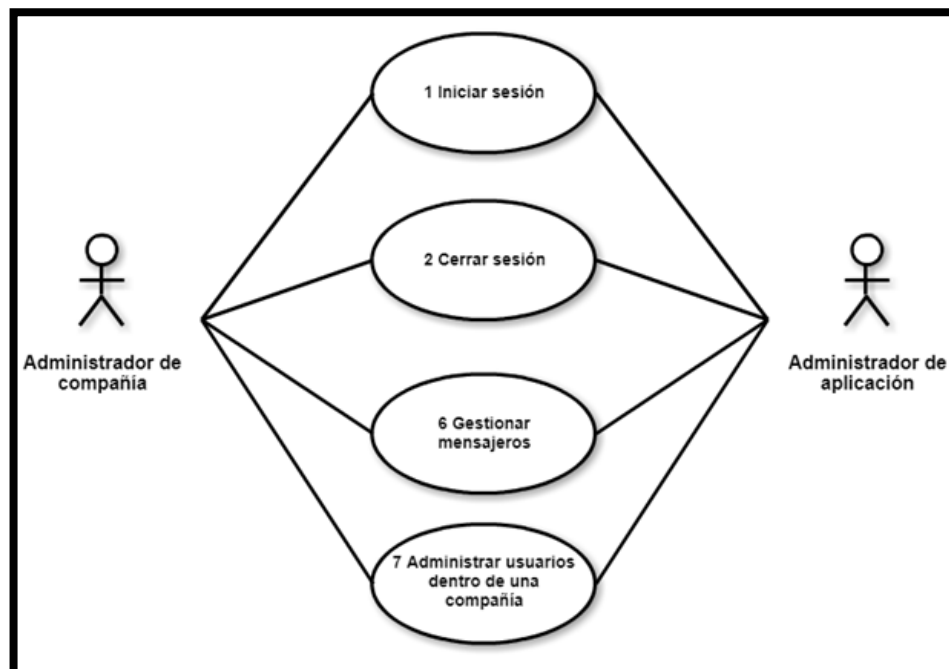


Figura 9. Diagrama de casos de uso de la aplicación web Nivel 1: Administrador de compañía y Administrador de aplicación

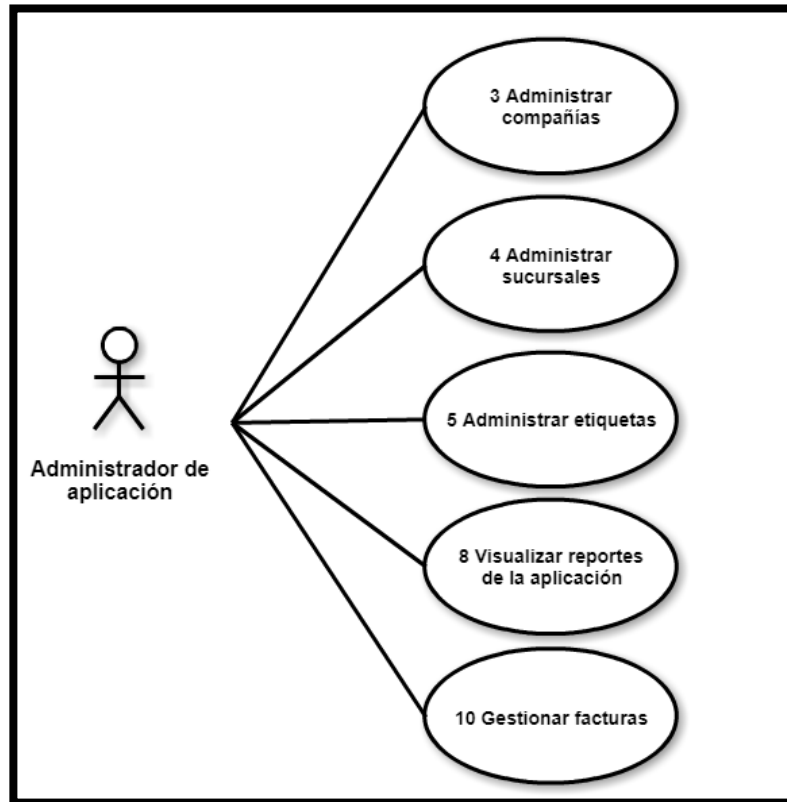


Figura 10. Diagrama de casos de uso de la aplicación web Nivel 1: Administrador de aplicación

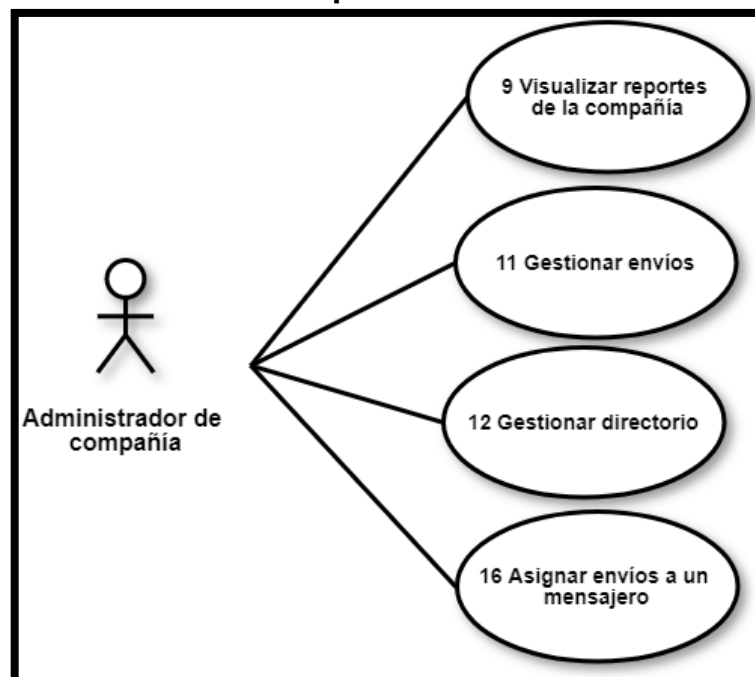


Figura 11. Diagrama de casos de uso de la aplicación web Nivel 1: Administrador de compañía

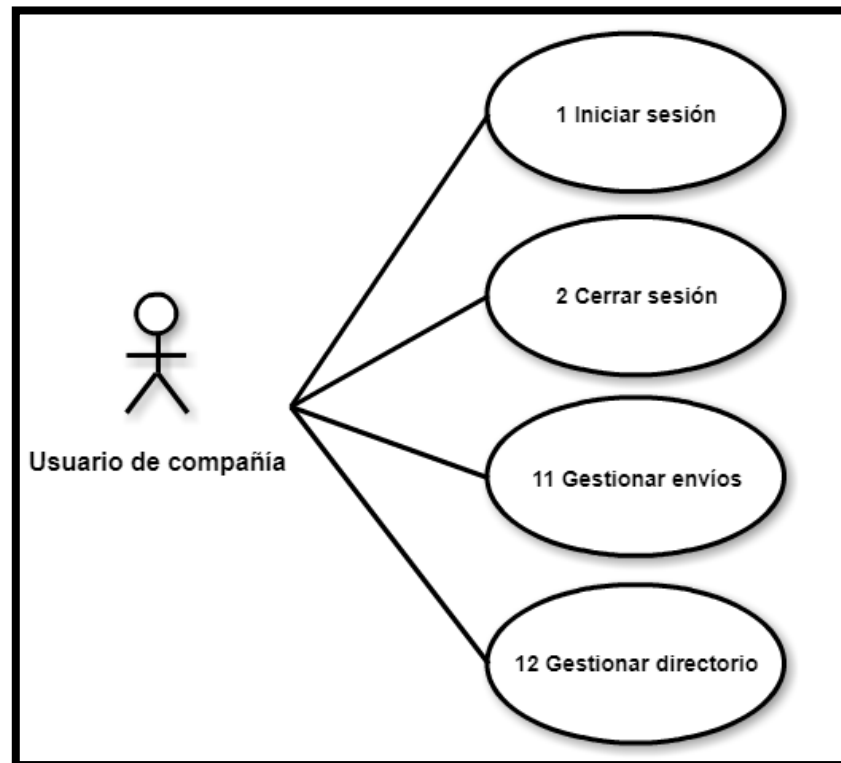


Figura 12 .Diagrama de casos de uso de la aplicación web Nivel 1: Usuario de compañía

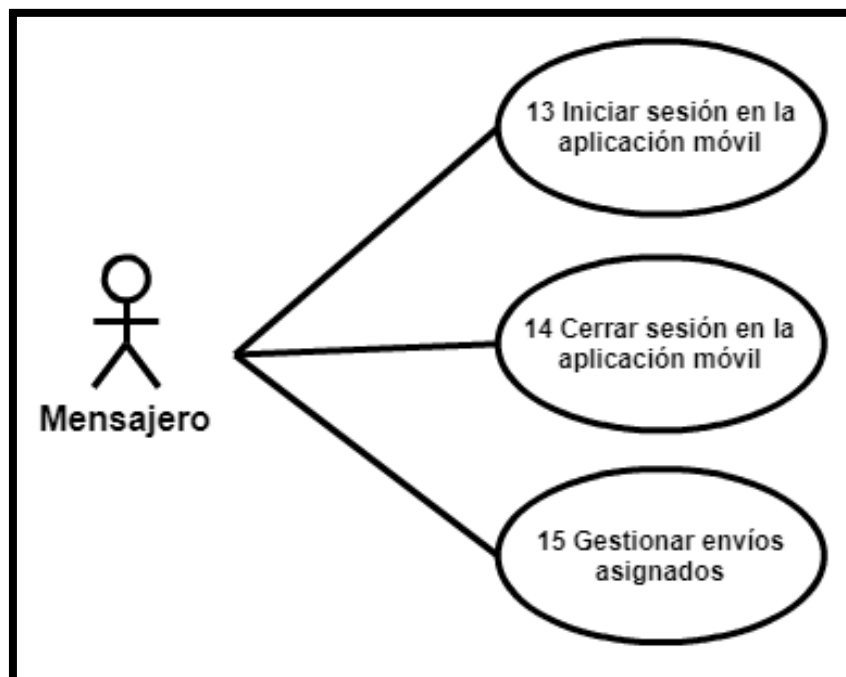


Figura 13. Diagrama de casos de uso de la aplicación móvil Nivel 1: Mensajero

Identificador	1
Nombre del Caso de Uso	Iniciar sesión
Actores	Administrador de aplicación, Administrador de compañía, Usuario de compañía
Descripción	A través del correo electrónico y contraseña válidos se otorga a los usuarios acceso a la aplicación web

Identificador	2
Nombre del Caso de Uso	Cerrar sesión
Actores	Administrador de aplicación, Administrador de compañía, Usuario de compañía
Descripción	Se otorga la salida de la aplicación web
Precondición	El usuario debe haber iniciado sesión

Identificador	3
Nombre del Caso de Uso	Administrar compañías
Actores	Administrador de aplicación
Descripción	Permite al usuario tener control de las compañías a las cuales se les prestará el servicio de envíos internos, a través de operaciones de agregación, edición y eliminación
Precondición	El usuario debe haber iniciado sesión

Identificador	4
Nombre del Caso de Uso	Administrar sucursales
Actores	Administrador de aplicación
Descripción	Permite al usuario tener control de las sucursales pertenecientes a una compañía, a través de operaciones de agregación, edición y eliminación
Precondición	El usuario debe haber iniciado sesión y haber creado una compañía

Identificador	5
Nombre del Caso de Uso	Administrar etiquetas

Actores	Administrador de aplicación
Descripción	Permite al usuario asignar lotes de etiquetas a las compañías, visualizar las asignaciones hechas a las compañías y filtrar por número de etiqueta.
Precondición	El usuario debe haber iniciado sesión y haber creado una compañía

Identificador	6
Nombre del Caso de Uso	Gestionar mensajeros
Actores	Administrador de aplicación, Administrador de compañía
Descripción	Permite al usuario añadir y visualizar mensajeros dentro de una compañía.
Precondición	El usuario debe haber iniciado sesión y haber creado una compañía

Identificador	7
Nombre del Caso de Uso	Administrar usuarios dentro de una compañía
Actores	Administrador de aplicación, Administrador de compañía
Descripción	Permite al usuario tener control de los usuarios pertenecientes a una compañía, a través de operaciones de agregación, edición y eliminación
Precondición	El usuario debe haber iniciado sesión y haber creado una compañía

Identificador	8
Nombre del Caso de Uso	Visualizar reportes de la aplicación
Actores	Administrador de aplicación
Descripción	Permite al usuario visualizar reportes con información relevante de la aplicación, como: compañías activas, compañías inactivas, compañías suspendidas, etiquetas asignadas y etiquetas usadas
Precondición	El usuario debe haber iniciado sesión

Identificador	9
Nombre del Caso de Uso	Visualizar reportes de la compañía

Actores	Administrador de compañía
Descripción	Permite al usuario visualizar reportes con información relevante de la compañía, como: mensajeros activos, mensajeros inactivos, envíos recogidos, envíos entregados, etiquetas asignadas y etiquetas usadas
Precondición	El usuario debe haber iniciado sesión

Identificador	10
Nombre del Caso de Uso	Gestionar facturas
Actores	Administrador de aplicación
Descripción	Permite al usuario realizar facturas a las compañías, así como visualizar el histórico de facturas realizadas
Precondición	El usuario debe haber iniciado sesión y debe existir al menos una compañía

Identificador	11
Nombre del Caso de Uso	Gestionar envíos
Actores	Administrador de compañía, Usuario de compañía
Descripción	Permite al usuario cancelar envíos, visualizar detalles del envío, buscar envíos y crear nuevos envíos.
Precondición	El usuario debe haber iniciado sesión

Identificador	12
Nombre del Caso de Uso	Gestionar directorio
Actores	Administrador de compañía, Usuario de compañía
Descripción	Permite al usuario visualizar el directorio de la compañía, crear envíos para usuarios del directorio, buscar usuarios y agregar usuarios a una lista de favoritos
Precondición	El usuario debe haber iniciado sesión

Identificador	13
Nombre del Caso de Uso	Iniciar sesión en la aplicación móvil
Actores	Mensajero

Descripción	A través del correo electrónico y contraseña válidos se otorga a los usuarios acceso a la aplicación móvil
--------------------	--

Identificador	14
Nombre del Caso de Uso	Cerrar sesión en la aplicación móvil
Actores	Mensajero
Descripción	Se otorga la salida de la aplicación móvil
Precondición	El usuario debe haber iniciado sesión

Identificador	15
Nombre del Caso de Uso	Gestionar envíos asignados
Actores	Mensajero
Descripción	Permite al usuario tener acceso a los envíos que tiene asignado, podrá ver todos los datos de un envío así como cambiar el estado del mismo, adicional el usuario podrá verificar la de etiqueta de un envío mediante un escáner
Precondición	El usuario debe haber iniciado sesión

Identificador	16
Nombre del Caso de Uso	Asignar envíos a un mensajero
Actores	Administrador de compañía
Descripción	Permite al usuario asignar un mensajero de la compañía a un envío creado previamente, para que este sea el encargado de transportar dicho envío
Precondición	El usuario debe haber iniciado sesión y haber creado un envío

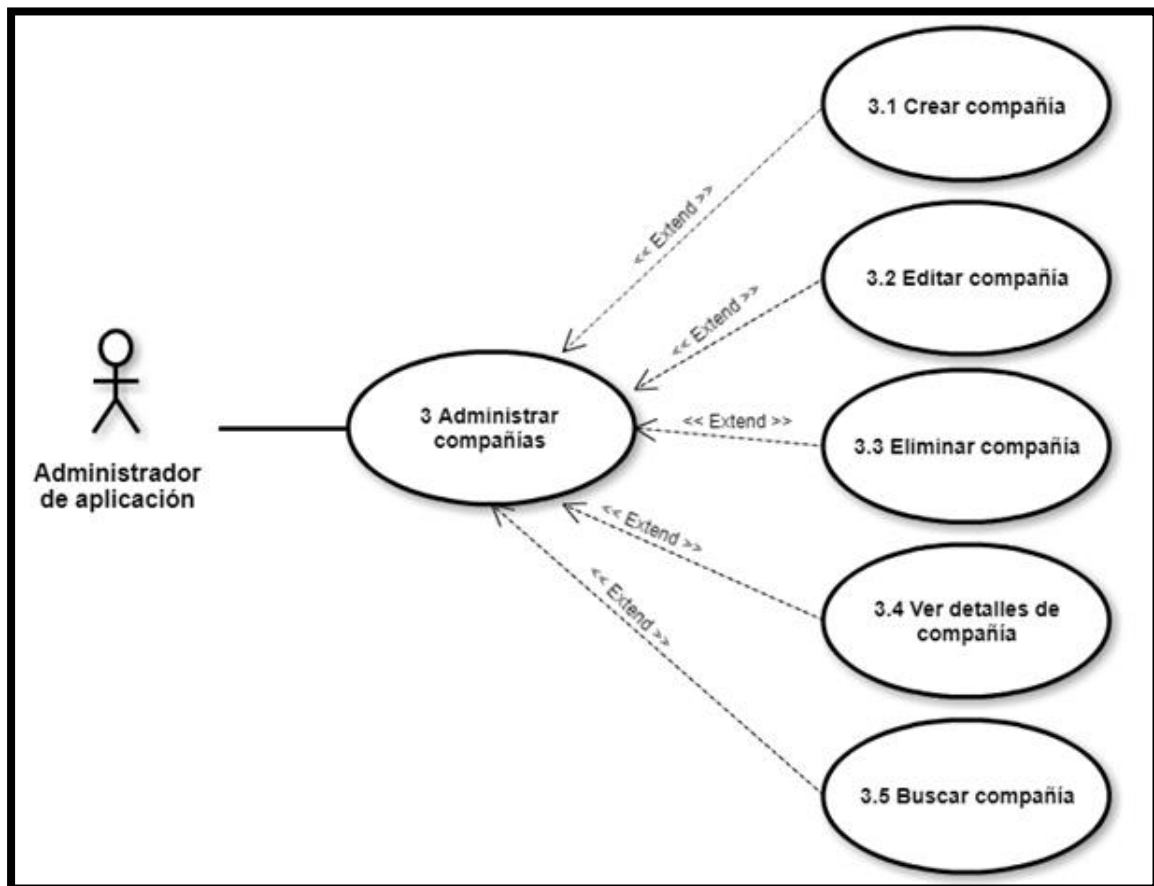


Figura 14. Diagrama de casos de uso de la aplicación web Nivel 2: Administrador de aplicación

Identificador	3.1
Nombre del Caso de Uso	Crear compañía
Actores	Administrador de aplicación
Descripción	Permite la creación de compañías
Precondición	El usuario debe haber iniciado sesión

Identificador	3.2
Nombre del Caso de Uso	Editar compañía
Actores	Administrador de aplicación
Descripción	Permite al usuario modificar los datos de una compañía
Precondición	El usuario debe haber iniciado sesión

Identificador	3.3
----------------------	-----

Nombre del Caso de Uso	Eliminar compañía
Actores	Administrador de aplicación
Descripción	Permite al usuario borrar una compañía de la aplicación
Precondición	El usuario debe haber iniciado sesión

Identificador	3.4
Nombre del Caso de Uso	Ver detalles de compañía
Actores	Administrador de aplicación
Descripción	Permite al usuario visualizar todos los datos de una compañía
Precondición	El usuario debe haber iniciado sesión

Identificador	3.5
Nombre del Caso de Uso	Buscar compañía
Actores	Administrador de aplicación
Descripción	Permite al usuario realizar búsquedas a través del: nombre de la compañía, país o por el estado de la compañía
Precondición	El usuario debe haber iniciado sesión

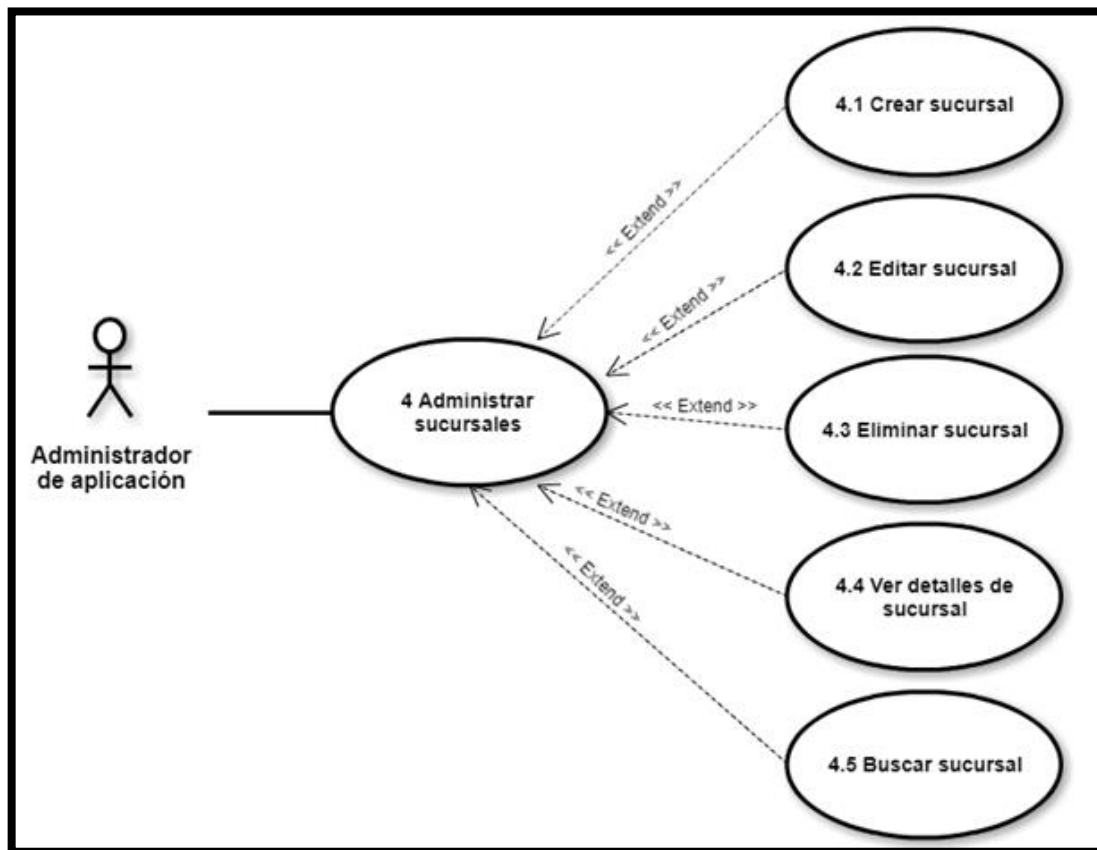


Figura 15. Diagrama de casos de uso de la aplicación web Nivel 2: Administrador de aplicación. Parte 2

Identificador	4.1
Nombre del Caso de Uso	Crear sucursal
Actores	Administrador de aplicación
Descripción	Permite la creación de sucursales dentro de una compañía
Precondición	El usuario debe haber iniciado sesión

Identificador	4.2
Nombre del Caso de Uso	Editar sucursal
Actores	Administrador de aplicación
Descripción	Permite al usuario modificar los datos de una sucursal
Precondición	El usuario debe haber iniciado sesión

Identificador	4.3
Nombre del Caso de Uso	Eliminar sucursal

Actores	Administrador de aplicación
Descripción	Permite al usuario borrar una sucursal de la aplicación
Precondición	El usuario debe haber iniciado sesión

Identificador	4.4
Nombre del Caso de Uso	Ver detalles de sucursal
Actores	Administrador de aplicación
Descripción	Permite al usuario visualizar todos los datos de una sucursal
Precondición	El usuario debe haber iniciado sesión

Identificador	4.5
Nombre del Caso de Uso	Buscar sucursal
Actores	Administrador de aplicación
Descripción	Permite al usuario realizar búsquedas a través del nombre de la sucursal o ciudad a la que pertenece
Precondición	El usuario debe haber iniciado sesión

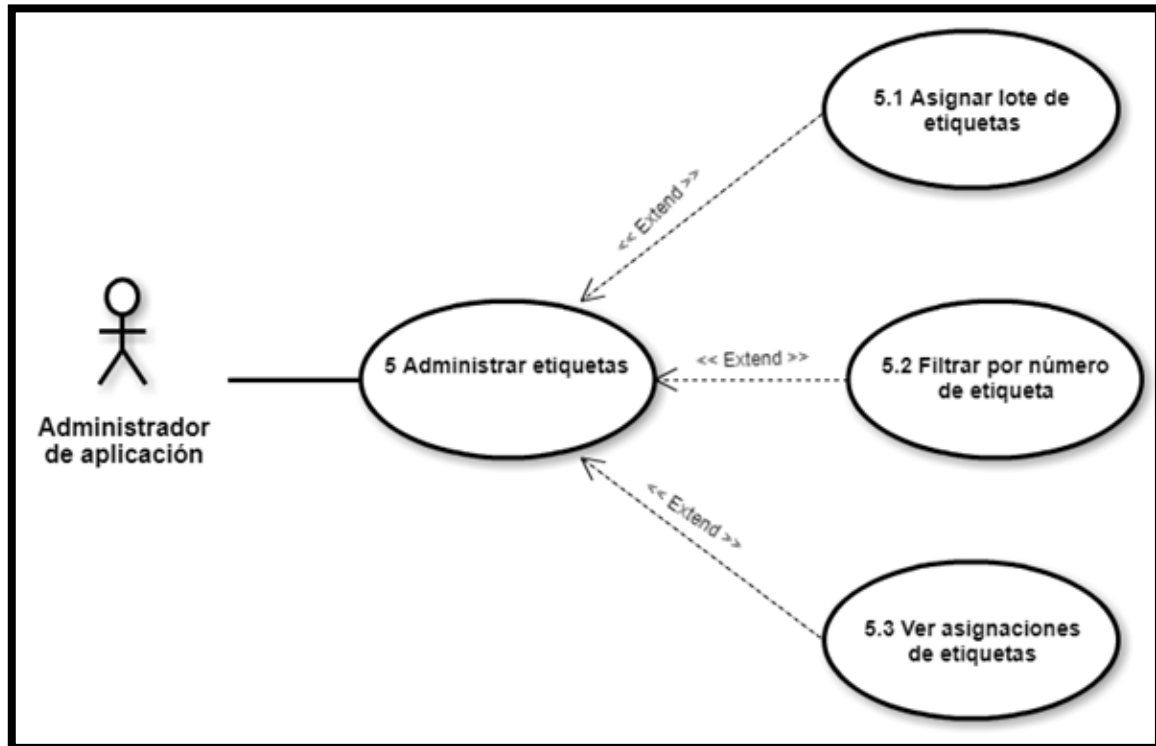


Figura 16. Diagrama de casos de uso de la aplicación web Nivel 2: Administrador de aplicación. Parte 3

Identificador	5.1
Nombre del Caso de Uso	Asignar lote de etiquetas
Actores	Administrador de aplicación
Descripción	Permite al usuario asignar lotes de etiquetas a una compañía
Precondición	El usuario debe haber iniciado sesión

Identificador	5.2
Nombre del Caso de Uso	Filtrar por número de etiquetas
Actores	Administrador de aplicación
Descripción	Permite al usuario buscar lotes asignados a través de un número de etiqueta
Precondición	El usuario debe haber iniciado sesión

Identificador	5.3
Nombre del Caso de Uso	Ver asignaciones de etiquetas
Actores	Administrador de aplicación
Descripción	Permite al usuario visualizar todas las

	asignaciones de etiquetas hechas a una compañía
Precondición	El usuario debe haber iniciado sesión

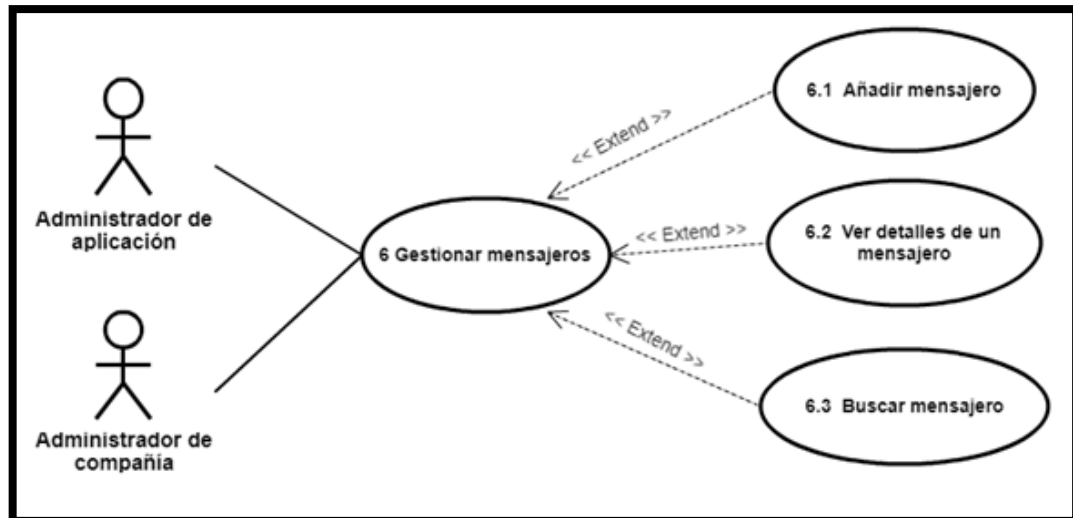


Figura 17. Diagrama de casos de uso de la aplicación web Nivel 2: Administrador de aplicación y Administrador de compañía

Identificador	6.1
Nombre del Caso de Uso	Añadir mensajero
Actores	Administrador de aplicación, Administrador de compañía
Descripción	Permite al usuario añadir un mensajero dentro de una compañía
Precondición	El usuario debe haber iniciado sesión

Identificador	6.2
Nombre del Caso de Uso	Ver detalles de un mensajero
Actores	Administrador de aplicación, Administrador de compañía
Descripción	Permite al usuario visualizar todos los datos de un mensajero
Precondición	El usuario debe haber iniciado sesión

Identificador	6.3
Nombre del Caso de Uso	Buscar mensajero
Actores	Administrador de aplicación, Administrador

	de compañía
Descripción	Permite al usuario buscar mensajeros a través del nombre o correo electrónico
Precondición	El usuario debe haber iniciado sesión

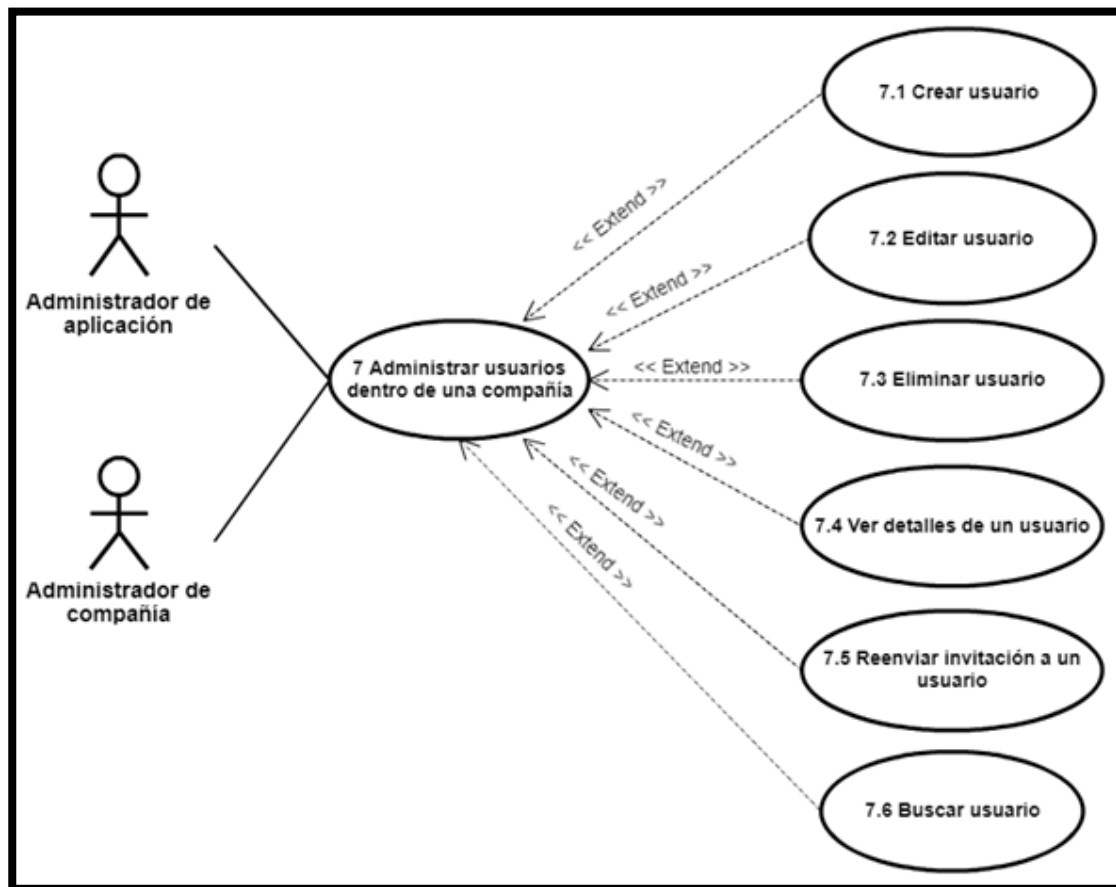


Figura 18. Diagrama de casos de uso de la aplicación web Nivel 2: Administrador de aplicación y Administrador de compañía

Identificador	7.1
Nombre del Caso de Uso	Crear usuario
Actores	Administrador de aplicación, Administrador de compañía
Descripción	Permite la creación de usuarios dentro una compañía, los tipos de usuarios que se pueden crear son: Administrador de compañía, mensajero y usuario de compañía
Precondición	El usuario debe haber iniciado sesión

Identificador	7.2
----------------------	-----

Nombre del Caso de Uso	Editar usuario
Actores	Administrador de aplicación, Administrador de compañía
Descripción	Permite modificar los datos de un usuario
Precondición	El usuario debe haber iniciado sesión

Identificador	7.3
Nombre del Caso de Uso	Eliminar usuario
Actores	Administrador de aplicación, Administrador de compañía
Descripción	Permite al usuario borrar a un usuario de la aplicación
Precondición	El usuario debe haber iniciado sesión

Identificador	7.4
Nombre del Caso de Uso	Ver detalle de un usuario
Actores	Administrador de aplicación, Administrador de compañía
Descripción	Permite al usuario visualizar todos los datos de un usuario
Precondición	El usuario debe haber iniciado sesión

Identificador	7.5
Nombre del Caso de Uso	Reenviar invitación a un usuario
Actores	Administrador de aplicación, Administrador de compañía
Descripción	Permite al usuario reenviar un correo electrónico a un usuario que no haya podido completar su registro en la aplicación
Precondición	El usuario debe haber iniciado sesión

Identificador	7.6
Nombre del Caso de Uso	Buscar usuario
Actores	Administrador de aplicación, Administrador de compañía
Descripción	Permite realizar búsquedas de usuarios a través del nombre o correo electrónico
Precondición	El usuario debe haber iniciado sesión



Figura 19. Diagrama de casos de uso de la aplicación web Nivel 2: Administrador de aplicación. Parte 4

Identificador	8.1
Nombre del Caso de Uso	Ajustar gráficos de etiquetas
Actores	Administrador de aplicación
Descripción	Permite al usuario seleccionar un rango de fechas, para poder visualizar los estados de las etiquetas en un gráfico
Precondición	El usuario debe haber iniciado sesión

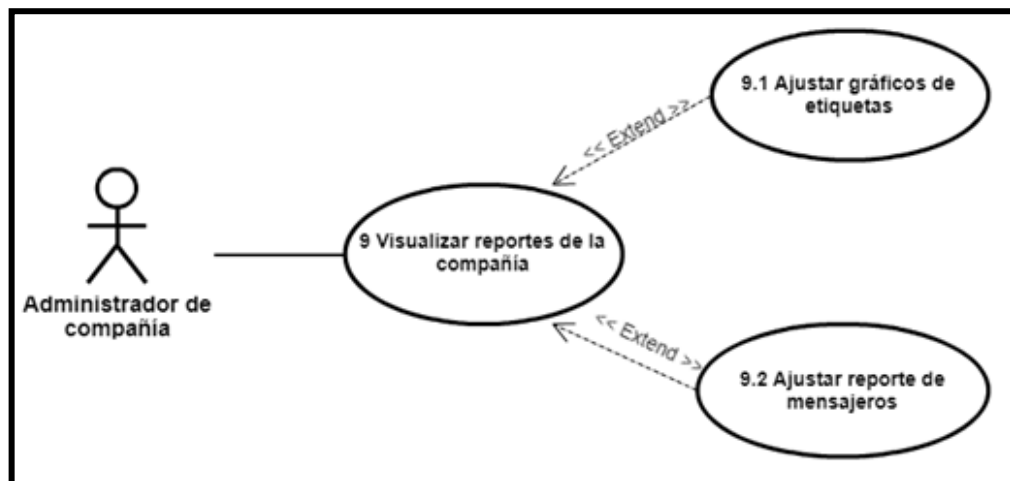


Figura 20. Diagrama de casos de uso de la aplicación web Nivel 2: Administrador de compañía

Identificador	9.1
Nombre del Caso de Uso	Ajustar gráficos de etiquetas
Actores	Administrador de compañía
Descripción	Permite al usuario seleccionar un rango de fechas, para poder visualizar los estados de

	las etiquetas en un gráfico
Precondición	El usuario debe haber iniciado sesión

Identificador	9.2
Nombre del Caso de Uso	Ajustar reporte de mensajeros
Actores	Administrador de compañía
Descripción	Permite al usuario seleccionar un rango de fechas, para poder visualizar la efectividad de los mensajeros
Precondición	El usuario debe haber iniciado sesión

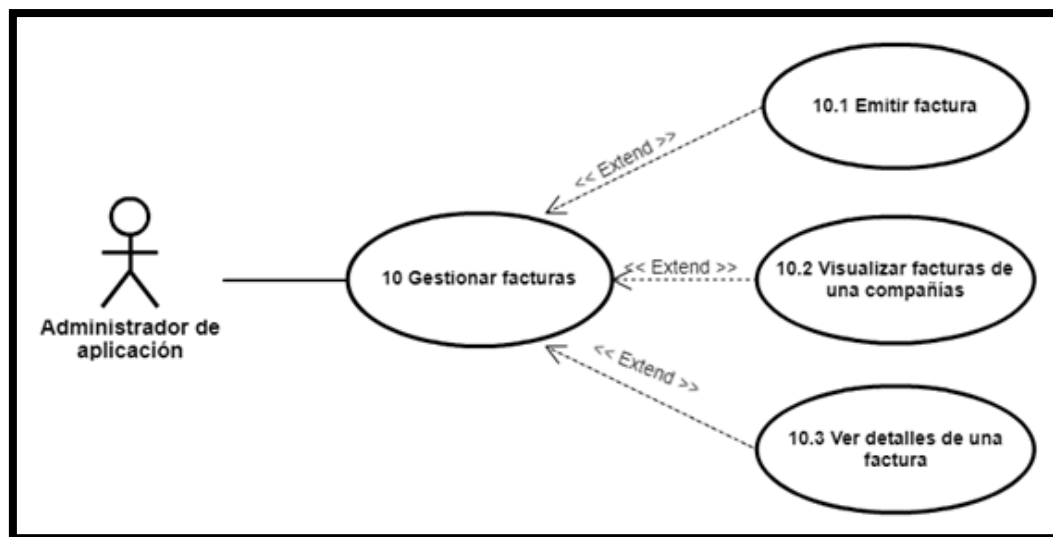


Figura 21. Diagrama de casos de uso de la aplicación web Nivel 2: Administrador de aplicación. Parte 5

Identificador	10.1
Nombre del Caso de Uso	Emitir factura
Actores	Administrador de aplicación
Descripción	Permite al usuario emitir una factura por las etiquetas asignadas a una compañía en un rango de fecha
Precondición	El usuario debe haber iniciado sesión

Identificador	10.2
Nombre del Caso de Uso	Visualizar facturas de una compañía
Actores	Administrador de aplicación
Descripción	Permite al usuario visualizar todas las

	facturas que fueron emitidas a una compañía
Precondición	El usuario debe haber iniciado sesión

Identificador	10.3
Nombre del Caso de Uso	Ver detalles de una factura
Actores	Administrador de aplicación
Descripción	Permite al usuario ver todos los datos asociados a una factura
Precondición	El usuario debe haber iniciado sesión

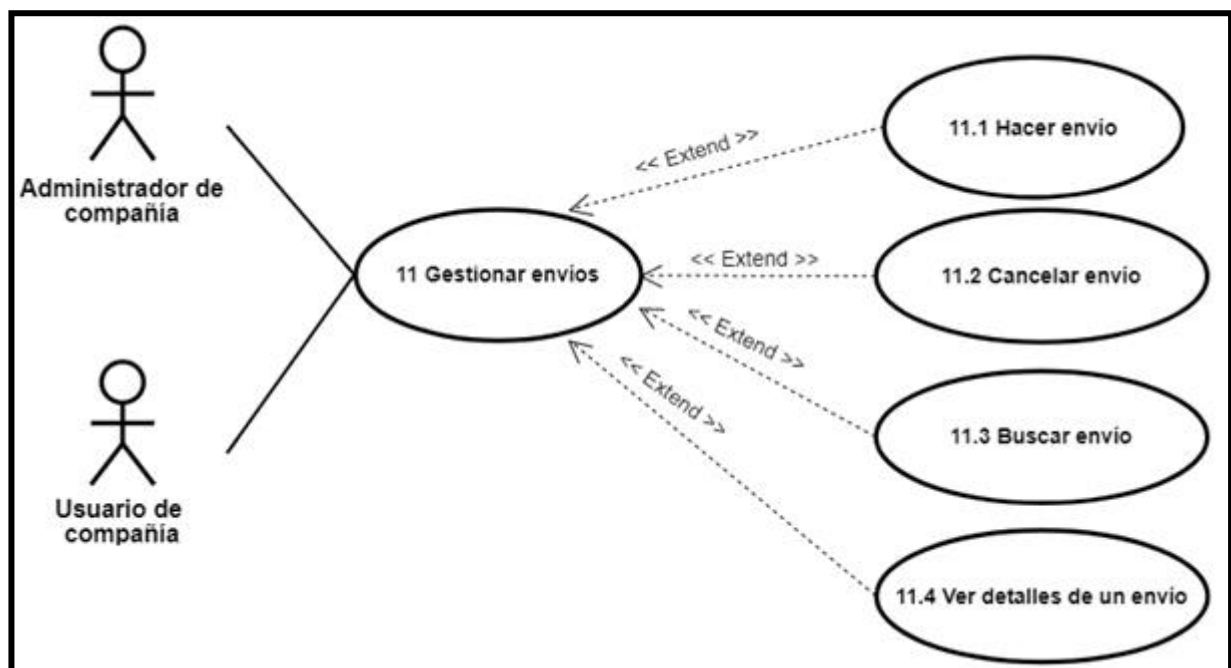


Figura 22. Diagrama de casos de uso de la aplicación web Nivel 2: Administrador de compañía y Usuario de compañía.

Identificador	11.1
Nombre del Caso de Uso	Hacer envío
Actores	Administrador de compañía, Usuario de compañía
Descripción	Permite al usuario crear envíos a otros usuarios dentro del directorio de una compañía
Precondición	El usuario debe haber iniciado sesión

Identificador	11.2
Nombre del Caso de Uso	Cancelar envío
Actores	Administrador de compañía, Usuario de compañía
Descripción	Permite al usuario cancelar un envío
Precondición	El usuario debe haber iniciado sesión y haber creado un envío

Identificador	11.3
Nombre del Caso de Uso	Buscar envío
Actores	Administrador de compañía, Usuario de compañía
Descripción	Permite al usuario buscar envíos a través de la referencia del envío o el nombre del usuario destino
Precondición	El usuario debe haber iniciado sesión

Identificador	11.4
Nombre del Caso de Uso	Ver detalles de un envío
Actores	Administrador de compañía, Usuario de compañía
Descripción	Permite al usuario visualizar todos los datos relacionados a un envío
Precondición	El usuario debe haber iniciado sesión

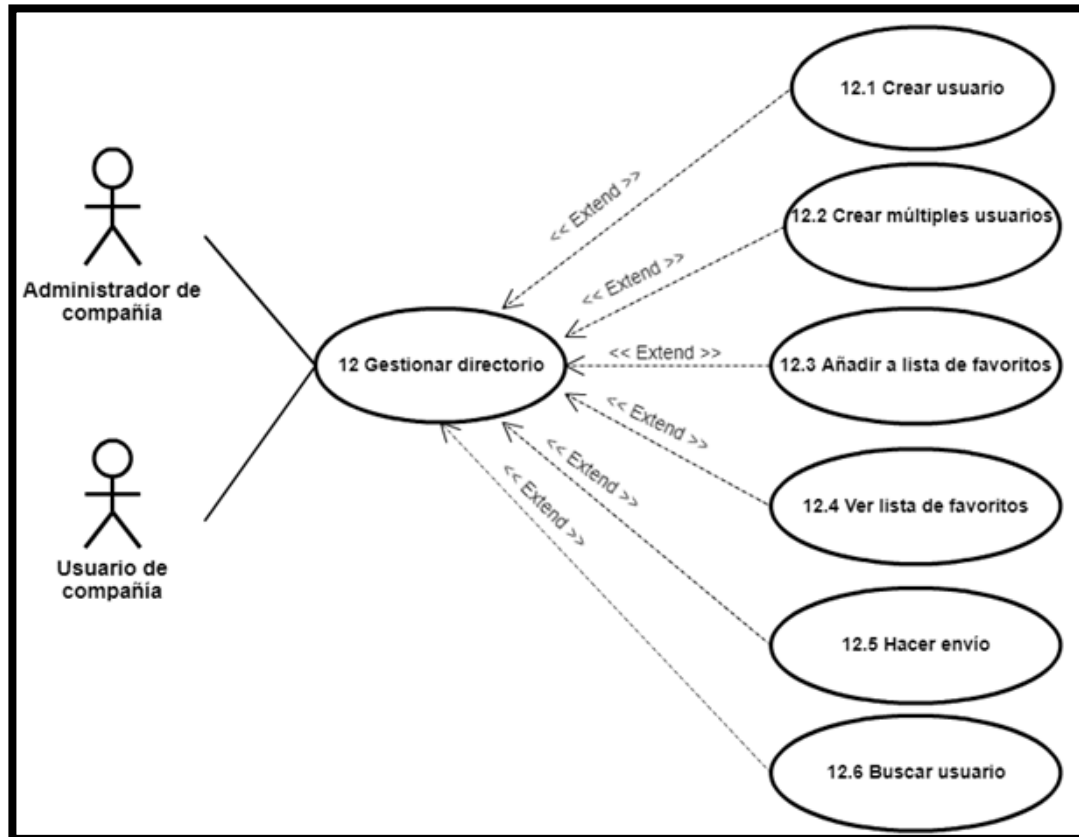


Figura 23. Diagrama de casos de uso de la aplicación web Nivel 2: Administrador de compañía y Usuario de compañía. Parte 2

Identificador	12.1
Nombre del Caso de Uso	Crear usuario
Actores	Administrador de compañía
Descripción	Permite la creación de usuarios dentro una compañía, los tipos de usuarios que se pueden crear son: Administrador de compañía, mensajero y usuario de compañía
Precondición	El usuario debe haber iniciado sesión

Identificador	12.2
Nombre del Caso de Uso	Crear múltiples usuarios
Actores	Administrador de compañía
Descripción	Permite al usuario crear múltiples usuarios a través de una operación, cargando la información de los usuarios mediante un archivo con formato CSV
Precondición	El usuario debe haber iniciado sesión

Identificador	12.3
Nombre del Caso de Uso	Añadir a lista de favoritos
Actores	Administrador de compañía, Usuario de compañía
Descripción	Permite al usuario añadir los usuarios más importantes para él, en una lista de favoritos
Precondición	El usuario debe haber iniciado sesión

Identificador	12.4
Nombre del Caso de Uso	Ver lista de favoritos
Actores	Administrador de compañía, Usuario de compañía
Descripción	Permite al usuario visualizar todos sus usuarios destacados o favoritos
Precondición	El usuario debe haber iniciado sesión

Identificador	12.5
Nombre del Caso de Uso	Hacer envío
Actores	Administrador de compañía, Usuario de compañía
Descripción	Permite al usuario crear envíos a otros usuarios dentro del directorio de una compañía
Precondición	El usuario debe haber iniciado sesión

Identificador	12.6
Nombre del Caso de Uso	Buscar usuario
Actores	Administrador de compañía, Usuario de compañía
Descripción	Permite realizar búsquedas de usuarios a través del nombre o correo electrónico
Precondición	El usuario debe haber iniciado sesión

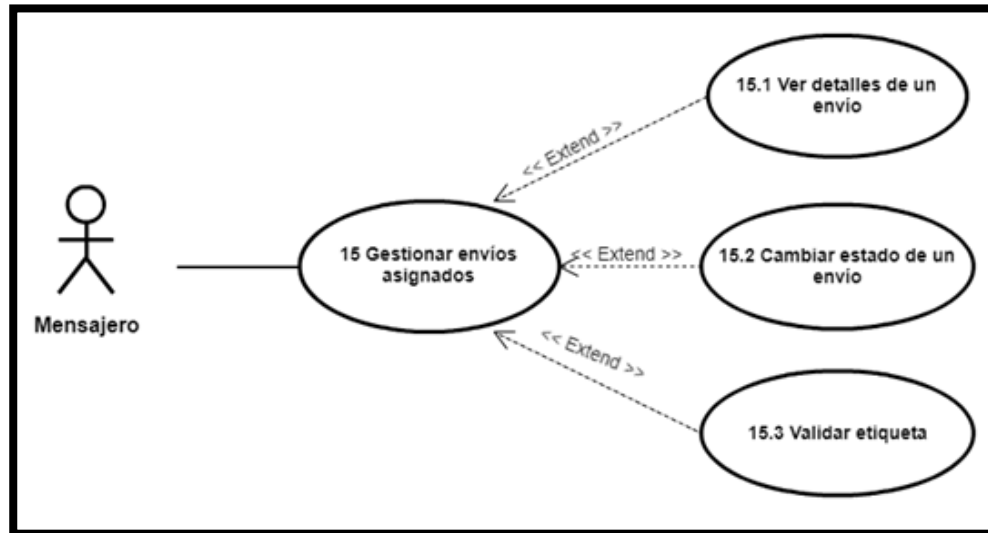


Figura 24. Diagrama de casos de uso de la aplicación web Nivel 2: Mensajero

Identificador	15.1
Nombre del Caso de Uso	Ver detalles de un envío
Actores	Mensajero
Descripción	Permite al usuario visualizar todos los datos relacionados a un envío
Precondición	El usuario debe haber iniciado sesión en la aplicación móvil

Identificador	15.2
Nombre del Caso de Uso	Cambiar estado de un envío
Actores	Mensajero
Descripción	Permite al usuario cambiar el estado del envío
Precondición	El usuario debe haber iniciado sesión en la aplicación móvil

Identificador	15.3
Nombre del Caso de Uso	Validar etiqueta
Actores	Mensajero
Descripción	Permite al usuario verificar un código QR correspondiente al número de una etiqueta, el usuario verifica los envíos asignados y cambia su estado automáticamente
Precondición	El usuario debe haber iniciado sesión en la aplicación móvil

4.7 Modelo de Datos

El Modelo de Datos elaborado para estas soluciones, representa una base de datos NoSQL orientada a documentos, donde se pueden visualizar los datos que son almacenados en cada documento y como se relacionan entre sí. Es importante destacar que también se incluyeron en el modelo colecciones relevantes para el desarrollo del API que se comunica con las aplicaciones (ver **Figura 25**).

Puesto que MongoDB es una base de datos NoSQL orientada a documentos y fue la que se escogió para este desarrollo, la mejor manera para visualizar que almacenará esta base de datos y como se relacionan los documentos entre sí, es con un modelo de datos.

Los documentos son estructuras de datos similares a un JSON, que son utilizadas por las bases de datos orientadas a documentos, para almacenar la información de una entidad, así como lo hacen las filas de una tabla de una base de datos SQL. La mayor diferencia entre estas dos es que los documentos son semiestructurados y pueden cambiar con el tiempo sin ningún problema, a diferencia de las filas de una tabla que son más rígidas. Dentro de los documentos más importantes están los de Usuario, que pueden representar desde un Mensajero hasta un Administrador de Aplicación, según el Rol que tenga asociado; Por otro lado, Los documentos de Envío almacenan el número de la etiqueta asignada, una descripción del contenido y también hacen referencia al usuario emisor, receptor, mensajero asignado y compañía a la que pertenece.

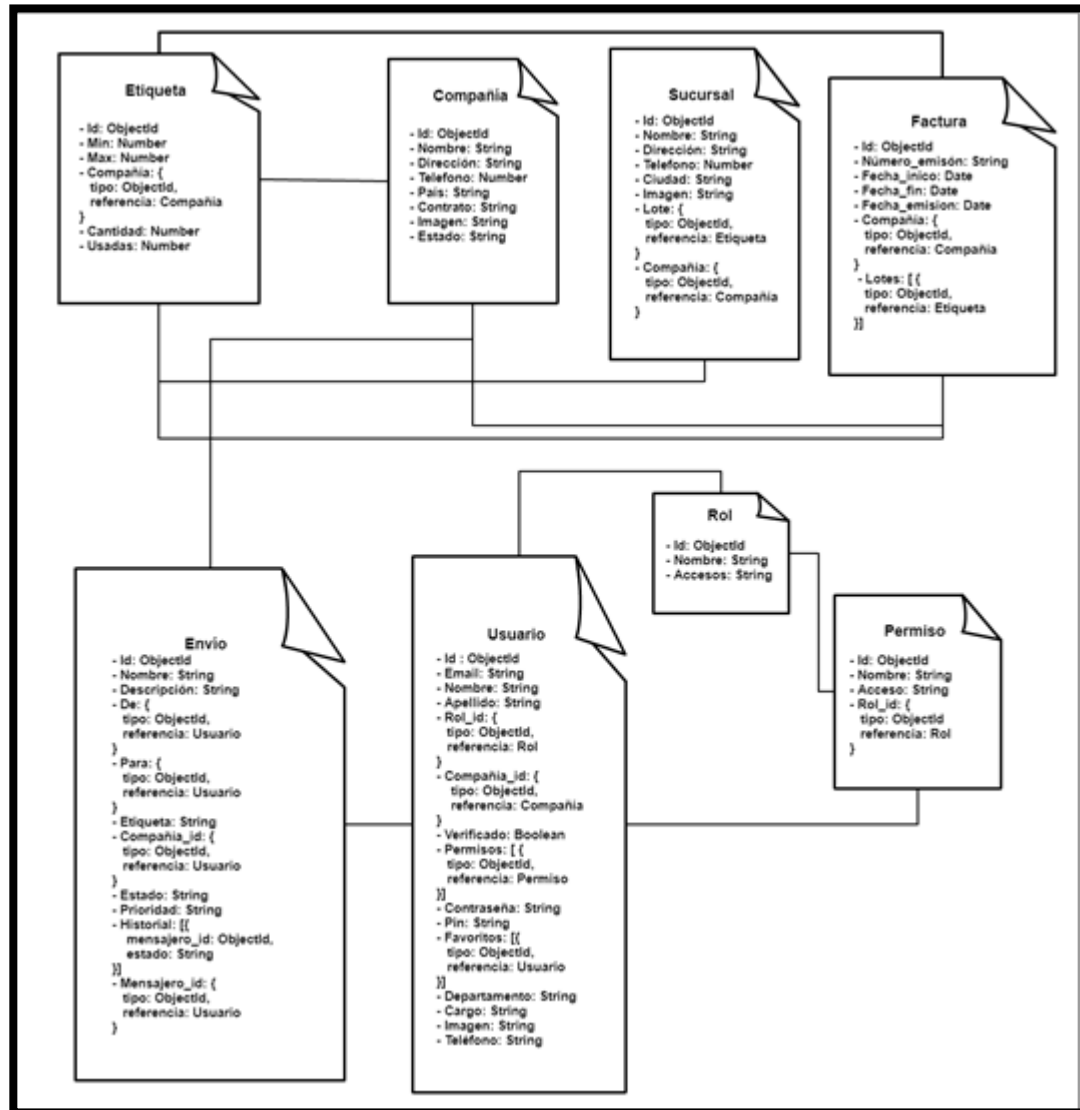


Figura 25. Modelo de datos orientado a documentos

4.8 Principales interfaces de la aplicación

A continuación se muestran las principales interfaces de la aplicación web y móvil



Figura 26. Página inicio de sesión

A través del formulario de inicio de sesión mostrado en la **Figura 26** los usuarios podrán acceder a la aplicación web.



Figura 27. Recuperación de contraseña

En la **figura 27** se muestra la interfaz donde el usuario podrá recuperar su contraseña, luego de introducir el correo electrónico y presionar el botón de enviar, el usuario recibirá un correo que le permitirá cambiar su contraseña.

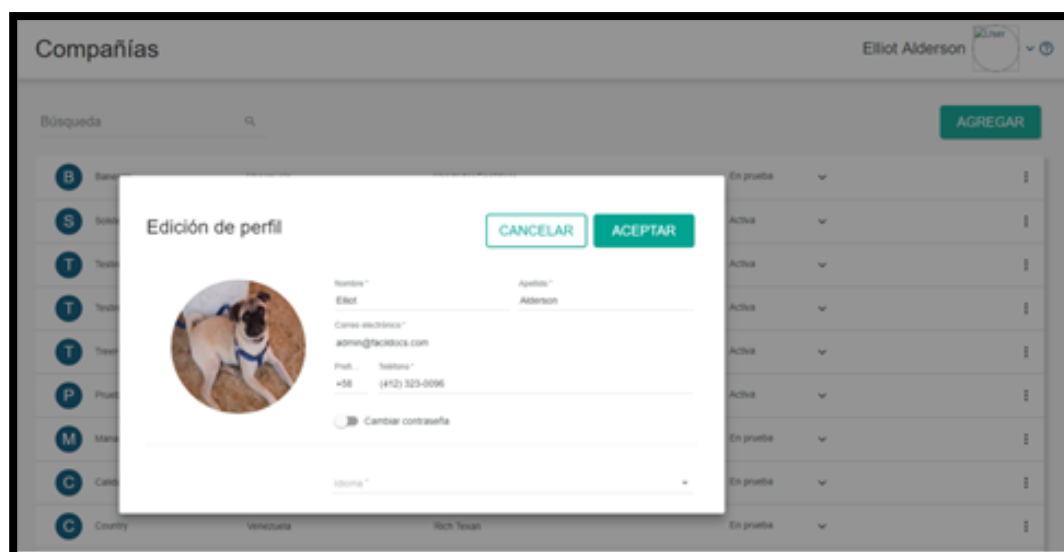


Figura 28. Edición de perfil

En la **figura 28** se muestra el formulario mediante el cual los usuarios podrán cambiar la información de su perfil.

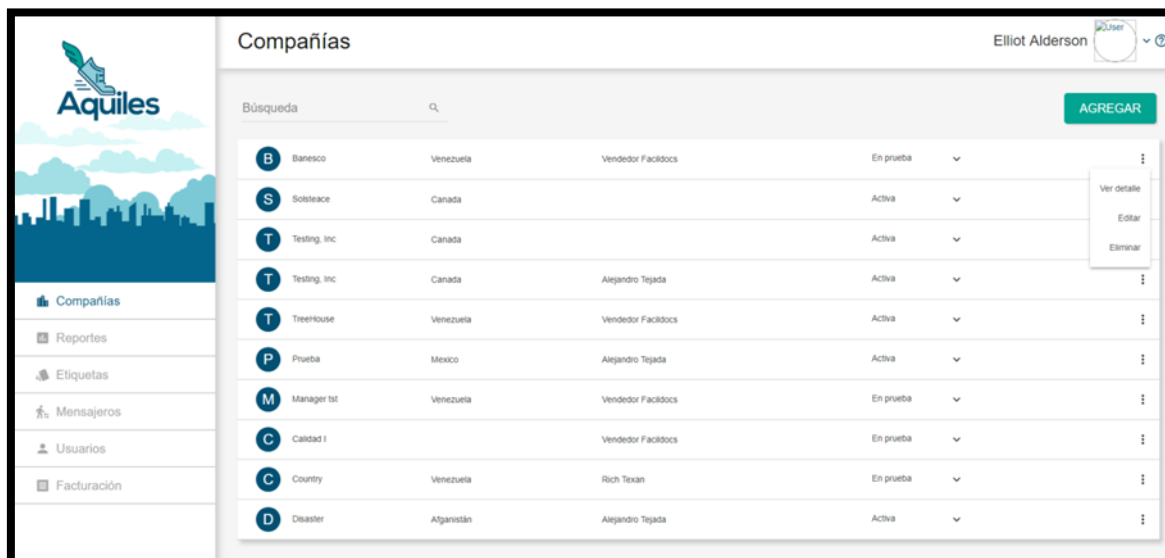


Figura 29. Sección de compañías

En la **figura 29** se observa la sección de compañías en donde el usuario podrá gestionarla, en la parte superior derecha se puede observar el botón que permite agregar nuevas compañías, así como las tres acciones que se pueden realizar sobre cada compañía (ver detalle, editar y eliminar), también en la parte superior izquierda se puede observar el buscador de compañías.

Nueva compañía

DATOS GENERALES

Nombre de la compañía * Estado *
País * Precio * Teléfono *

Sitio web Dirección *

DATOS DE FACTURACIÓN

Número de identificación fiscal Moneda de facturación *

Método de facturación * Monto acordado

☐ Facturación por centro de costos

DATOS DE LA OPERACIÓN

Tipo de conexión al directorio Mensajeros asignados por

Figura 30. Formulario para nuevas compañías

En la **figura 30** se observa el formulario para crear nuevas compañías, el cual aparece en forma de modal luego de que el usuario presiona el botón de "Agregar", una vez que usuario complete todos los campos requeridos en el formulario, se habilitara el botón que permite enviar y crear la nueva compañía.

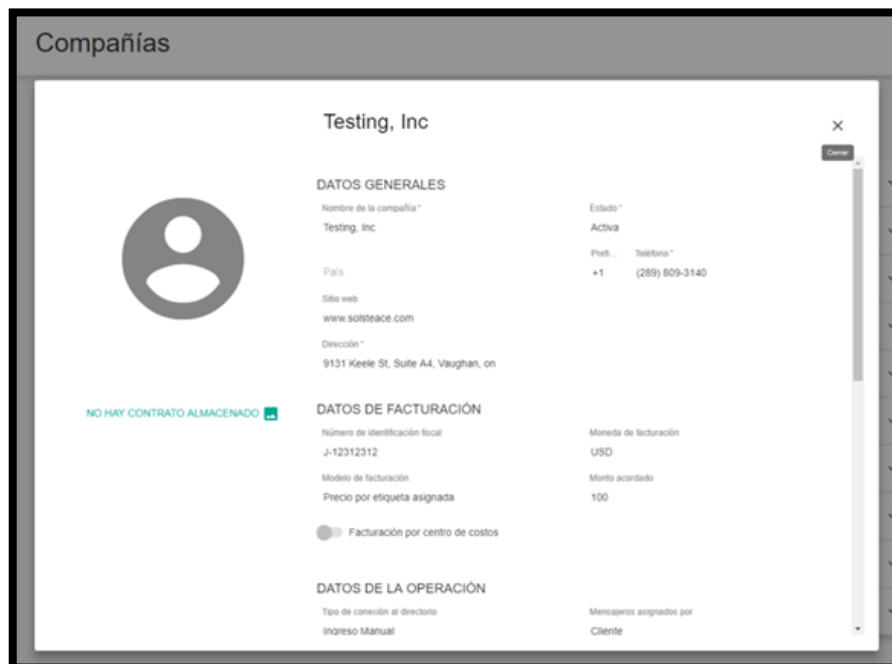


Figura 31. Datos de una compañía

En la **figura 31** se observa un modal el cual muestra toda la información asociada a una compañía.

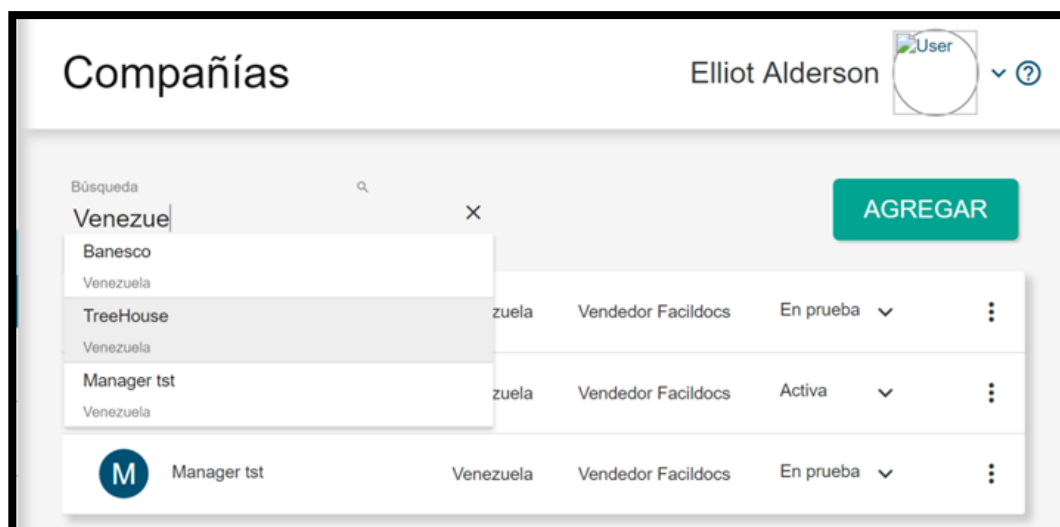


Figura 32. Buscador de compañías

En la **figura 32** se observa el buscado de compañías, este permite buscar por nombres o países, el buscador cuenta con una búsqueda predictiva.

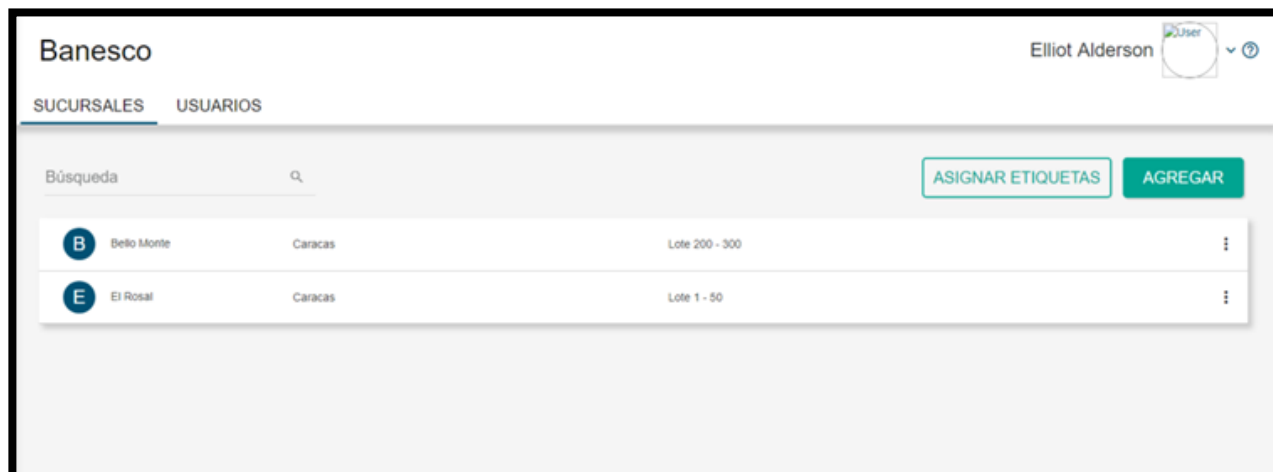


Figura 33. Sucursales de una compañía

En la **figura 33** se observa la sección de sucursales de una compañía, en la parte superior derecha se puede observar el botón que permite agregar nuevas sucursales y asignar etiquetas a una sucursal, de forma adicional en cada sucursal se pueden realizar las funciones de: Editar, ver detalle y eliminar, así como también se podrán hacer búsquedas en la sucursales.

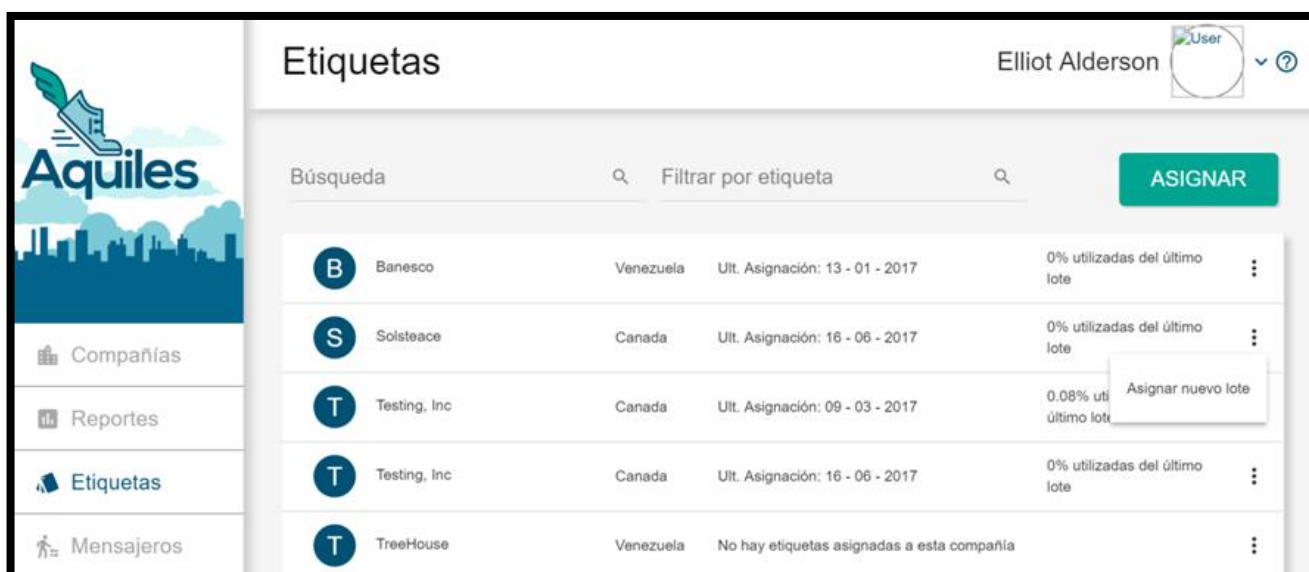
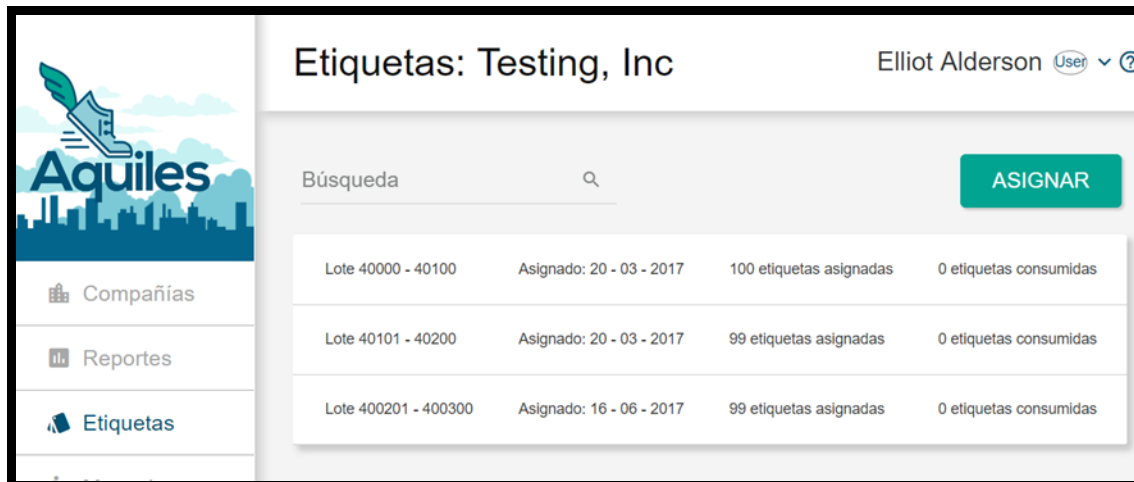


Figura 34. Sección de etiquetas

En la **figura 34** se observa la sección de etiquetas, donde el usuario podrá asignar lotes de etiquetas a las compañías, filtrar por etiquetas y buscar compañías.

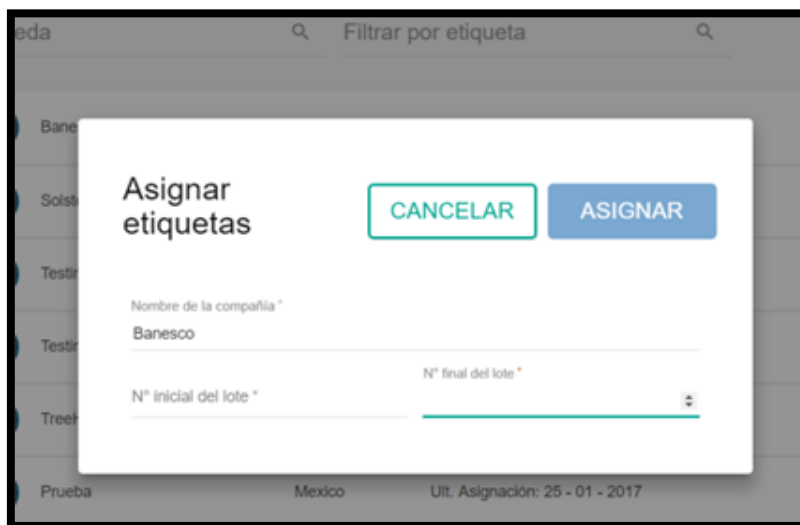


The screenshot shows a web application interface for 'Etiquetas: Testing, Inc'. The user 'Elliot Alderson' is logged in. The sidebar on the left contains navigation links: 'Compañías', 'Reportes', and 'Etiquetas'. The main content area has a search bar labeled 'Búsqueda' and a green 'ASIGNAR' button. Below this is a table listing assigned lots.

Lote	Asignado	Etiquetas asignadas	Etiquetas consumidas
Lote 40000 - 40100	Asignado: 20 - 03 - 2017	100 etiquetas asignadas	0 etiquetas consumidas
Lote 40101 - 40200	Asignado: 20 - 03 - 2017	99 etiquetas asignadas	0 etiquetas consumidas
Lote 400201 - 400300	Asignado: 16 - 06 - 2017	99 etiquetas asignadas	0 etiquetas consumidas

Figura 35. Lotes de etiquetas asignadas a una compañía

En la **figura 35** se observa la interfaz que permite visualizar los lotes que se han asignado a una determinada compañía, también desde esta pantalla se pueden asignar lotes de etiquetas.



The screenshot shows a modal form titled 'Asignar etiquetas'. It has two buttons at the top: 'CANCELAR' (green) and 'ASIGNAR' (blue). The form contains two input fields: 'Nombre de la compañía *' with the value 'Banesco' and 'N° inicial del lote *' with a range from 40000 to 40100. The 'N° final del lote *' field is empty. The background shows a list of companies with 'Banesco' selected.

Figura 36. Formulario de asignación de etiquetas

Con el formulario de asignación de etiquetas, que se puede observar en la **figura 36**, los administradores de la aplicación podrán asignar lotes de etiquetas a una compañía determinada.

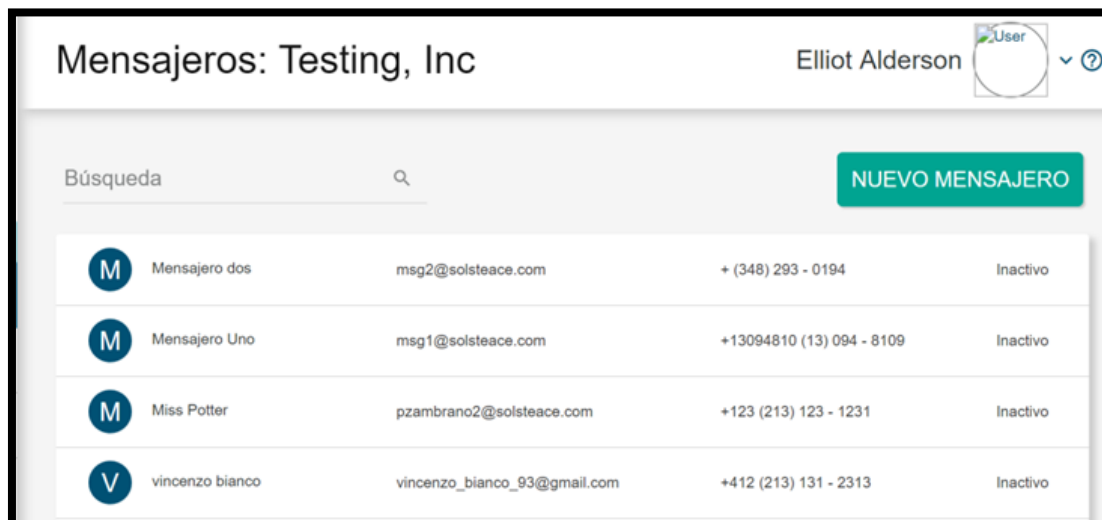


Figura 37. Sección de mensajeros

En la **figura 37** se observa la sección de mensajeros en donde el usuario podrá gestionar a los mensajeros, en la parte superior derecha se puede observar el botón que permite agregar nuevos mensajeros y en la parte superior izquierda encuentra el buscador de mensajeros.

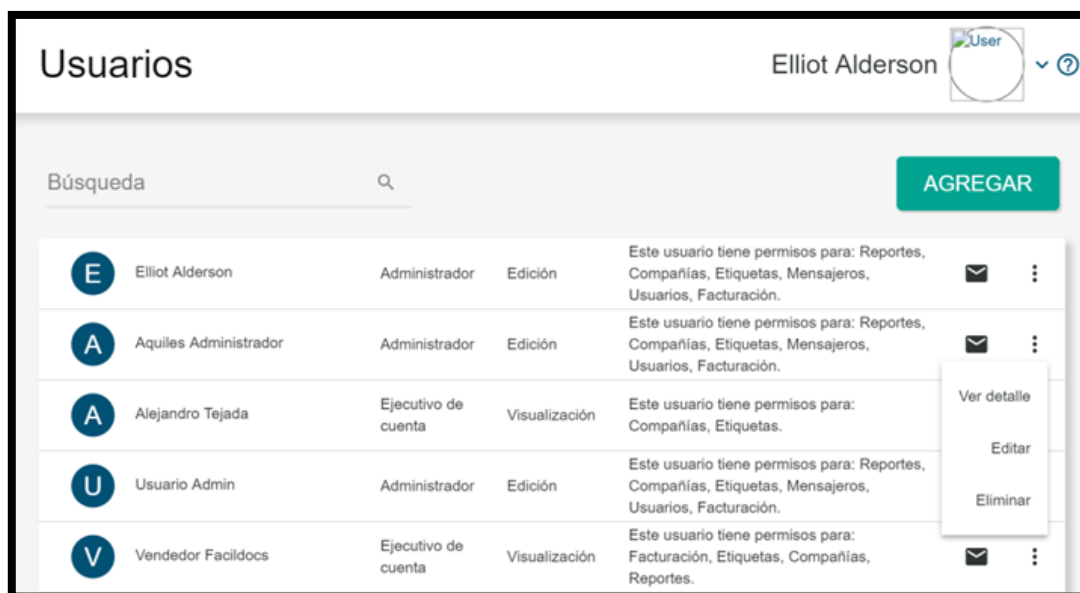


Figura 38. Sección de usuarios

En la **figura 38** se observa la sección de usuarios, en la parte superior derecha se puede observar el botón que permite agregar nuevos usuarios dentro de una compañía, así como las tres acciones que se pueden realizar sobre cada usuario (ver detalle, editar y eliminar), el icono de un sobre tiene la funcionalidad de reenviar una invitación al correo de un usuario que no haya completado su registro, también en la parte superior izquierda se puede observar el buscador de usuarios.

Facturación: Testing, Inc

Elliot Alderson

User

EMISIÓN

SOPORTES

Movimientos desde la fecha *

1/1/2017

Movimientos hasta la fecha *

1/1/2018

CANCELAR

EMITIR

Testing, Inc

Soporte N:

Emitido:

Precio por etiqueta asignada

Periodo: Domingo, enero 01, 2017 - Lunes, enero 01, 2018

Total: USD

Fecha	Lote	Etiquetas asignadas	Etiquetas movilizadas
Marzo 20, 2017	40000 - 40100	100	0
Marzo 20, 2017	40101 - 40200	99	0
Junio 16, 2017	400201 - 400300	99	0
		298	0

Figura 39. Emisión de facturas

En la **figura 39** se muestra la interfaz en donde el usuario podrá emitir facturas por las etiquetas asignadas a una compañía en un periodo de tiempo.

Facturación: Testing, Inc

Elliot Alderson

User

EMISIÓN

SOPORTES

Búsqueda

Soporte N 1

Emitido: Agosto 03, 2017

Periodo: Enero 01, 2017 - marzo 30, 2017

Soporte N 2

Emitido: Agosto 03, 2017

Periodo: Abril 01, 2017 - agosto 03, 2017

Figura 40. Facturas emitidas

En la **figura 40** se muestra la interfaz en donde el usuario podrá visualizar todas las facturas que han sido generadas a una compañía.

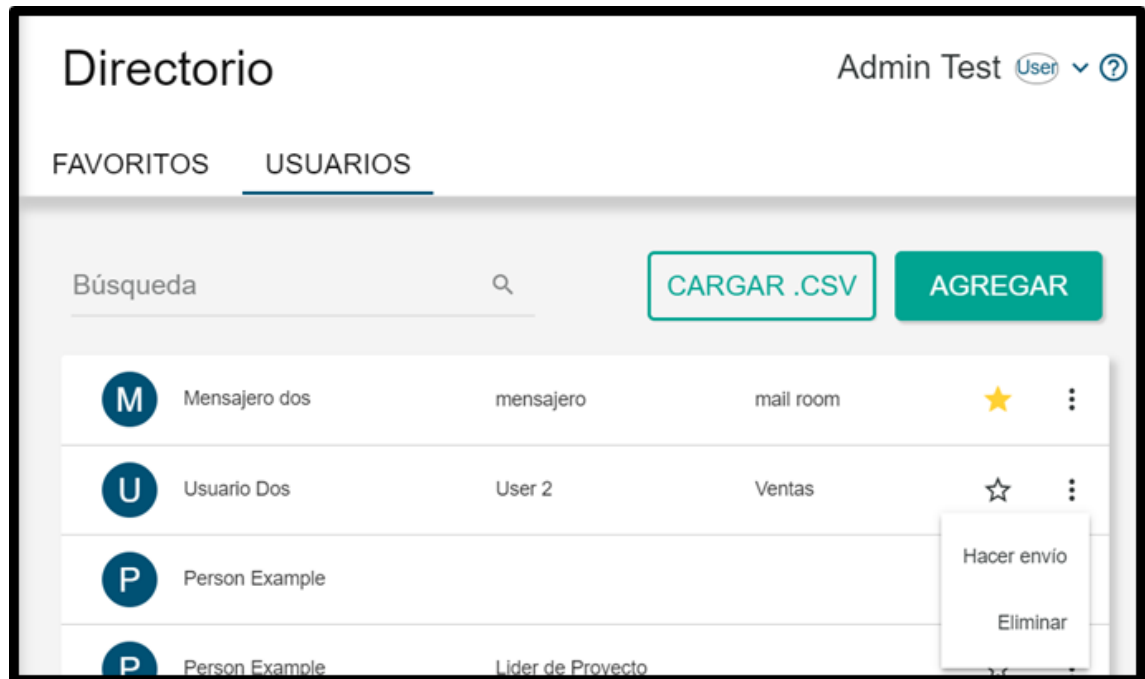


Figura 41. Directorio de una compañía

En la **figura 41** se observa el directorio de una compañía, donde se pueden hacer envíos a usuarios del directorio, agregar uno y múltiples usuarios (solo administradores), además de la funcionalidad de búsqueda de usuarios.

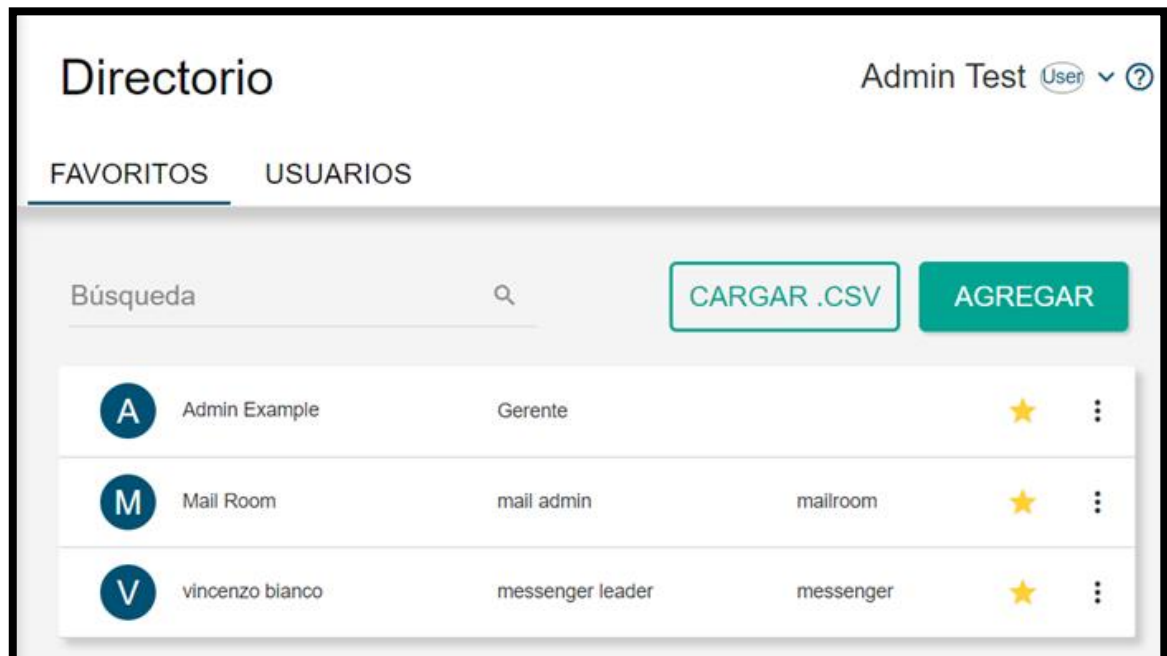


Figura 42. Lista de usuarios favoritos

En la **figura 42** se muestra la lista de usuarios favoritos, en donde el usuario podrá tener un acceso más rápido con sus contactos de preferencia.

Figura 43. Formulario para carga de múltiples usuarios

En la **figura 43** se muestra el formulario de carga de múltiples usuarios, el cual mediante una serie de campos y un archivo formato CSV con ciertas especificaciones, permitirá que el usuario cree nuevos usuarios dentro de una compañía en una sola operación.

Estado	Título	ID	Fecha	De	Para	Destinatario	Acciones
↓	Prueba tesis 3	1252	07 - 08 - 2017	De Admin Test a Usuario Dos	En entrega	Mensajero dos	⋮
⬆	Prueba tesis 2	1251	07 - 08 - 2017	De Admin Test a Usuario Dos	Salida	Mensajero dos	⋮
⬆	Referencia tesis 1	1250	07 - 08 - 2017	De Admin Test a Usuario Dos	Salida	Mensajero dos	⋮
⬇	Envío de Vincenzo	45003	19 - 06 - 2017	De Vinz Bianco a User Cinco	En central	Mensajero dos	⋮
⬆	Nomina de solsteace	45002	16 - 06 - 2017	De Admin Test a User Tres	Salida	Mensajero dos	⋮
⬆	Constancia de trabajo	45001	16 - 06 - 2017	De Admin Test a Mensajero dos	Salida	Mensajero Uno	⋮
⬇	Paquete de	45000	01 - 04 - 2017	De Usuario Dos a	Cancelado		⋮

Figura 44. Sección de envíos

En la **figura 44** se muestra la interfaz donde el usuario podrá gestionar sus envíos, con las funcionalidades de: crear un nuevo envío, cancelar un envío, ver detalle de un envío, asignar un envío a un mensajero y buscar envíos. Los envíos pueden tener prioridad baja, intermedia o alta; en la interfaz esto corresponde con

los colores verde, amarillo y rojo respectivamente. También, la flecha apuntando hacia arriba o hacia abajo indica si el envío va en dirección a la central o al usuario final.



Envío interno

CANCELAR ENVIAR

Contenido o referencia *

Remitente * Departamento del remitente

Admin Test Administración

Destinatario * Departamento del destinatario

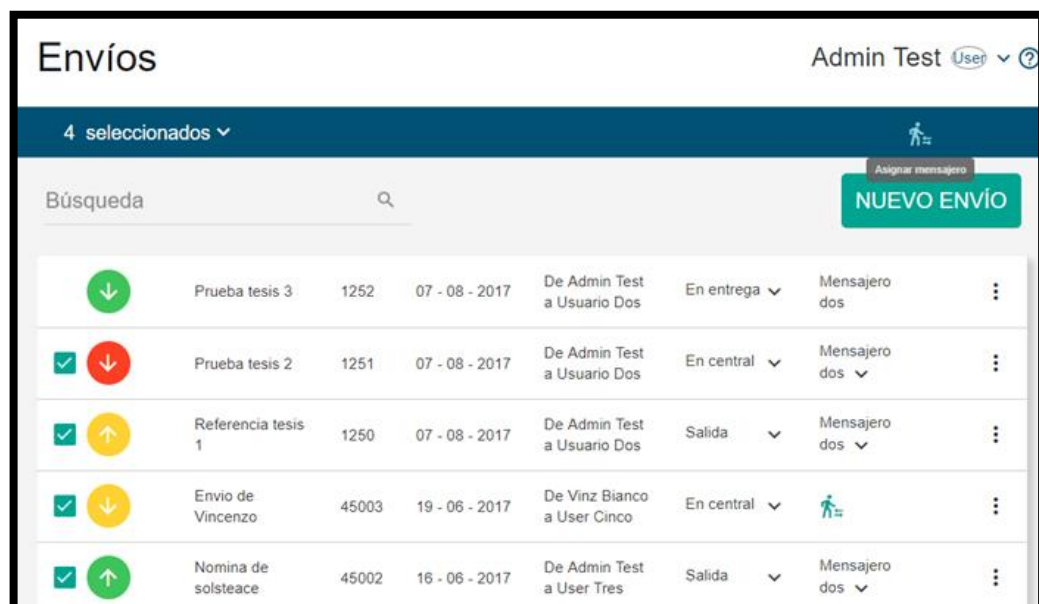
Etiqueta * Prioridad *

Descripción *

☐ Entrega garantizada

Figura 45. Formulario de nuevo envío

En la **figura 45** se muestra el formulario de nuevo envío, aquí el usuario podrá crear envíos para los compañeros que pertenezcan al directorio de la compañía.



Envíos Admin Test User ?

4 seleccionados

Búsqueda

Asignar mensajero

NUEVO ENVÍO

↓	Prueba tesis 3	1252	07 - 08 - 2017	De Admin Test a Usuario Dos	En entrega	Mensajero dos	⋮
✓ ↓	Prueba tesis 2	1251	07 - 08 - 2017	De Admin Test a Usuario Dos	En central	Mensajero dos	⋮
✓ ↑	Referencia tesis 1	1250	07 - 08 - 2017	De Admin Test a Usuario Dos	Salida	Mensajero dos	⋮
✓ ↓	Envío de Vincenzo	45003	19 - 06 - 2017	De Vinz Bianco a User Cinco	En central	Mensajero dos	⋮
✓ ↑	Nomina de solsteace	45002	16 - 06 - 2017	De Admin Test a User Tres	Salida	Mensajero dos	⋮

Figura 46. Asignación múltiple

En la **figura 46** se muestra la funcionalidad de asignación múltiple, donde el usuario podrá seleccionar varios envíos y asignarles un mensajero en una sola operación.

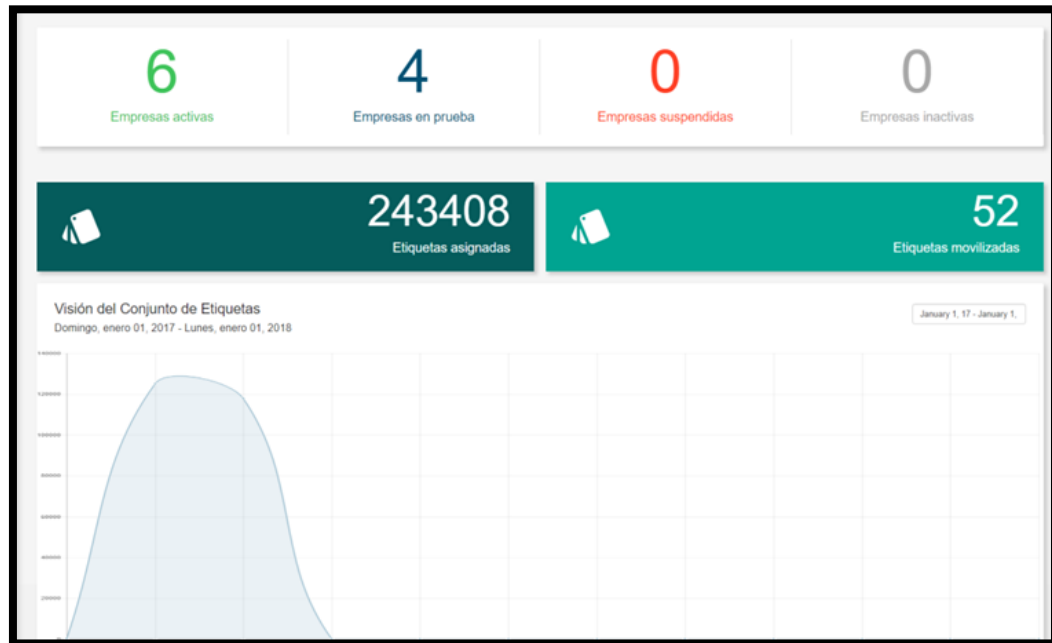


Figura 47. Reporte de etiquetas y compañías

En la **figura 47** se observa la sección de reportes, donde se puede visualizar información relevante sobre etiquetas y compañías.



Figura 48. Reporte de mensajeros

En la **figura 48** se observa la sección de reportes, en esta parte se puede visualizar información relevante sobre los mensajeros de una compañía.



Figura 49. Inicio de sesión aplicación móvil



Figura 50. Inicio de sesión aplicación móvil

En la **figura 49** y **figura 50** se muestran las pantallas mediante las cuales los mensajeros podrán ingresar en la aplicación móvil.

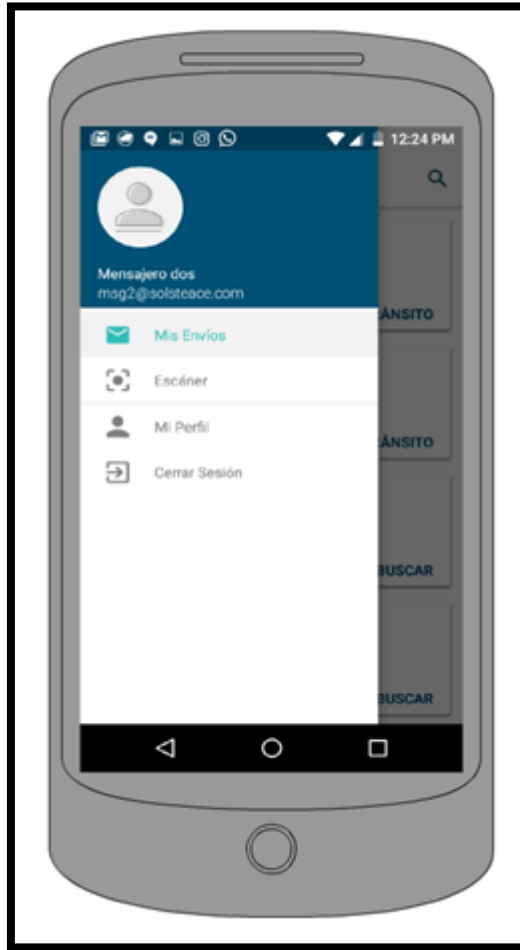


Figura 51. Menú de la aplicación móvil

En la **figura 51** se muestra el menú donde los mensajeros podrán: ver sus envíos, escanear etiquetas, ver su perfil y cerrar la sesión.



Figura 52. Sección mis envíos

En la **figura 52** se observa una lista con los envíos asignados a un mensajero, desde esta pantalla el mensajero podrá gestionar los envíos.



Figura 53. Escáner

En la **figura 53** se muestra el escáner con el cual los mensajeros podrán verificar si la etiqueta de un envío es válida, luego de que una etiqueta es escaneada por el mensajero, el envío cambia de estado, indicando que el envío ya está en manos del mensajero.



Figura 54. Detalle de un envío

En la **figura 54** se observa la pantalla en donde se pueden visualizar los datos de un envío, el mensajero podrá cambiar el estado de un envío desde esta pantalla.

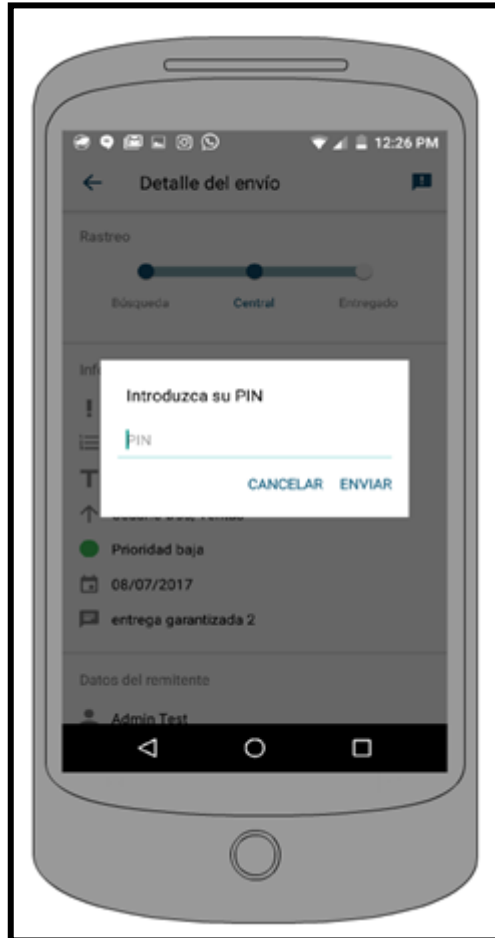


Figura 55. Formulario de seguridad

En la **figura 55** se observa el formulario que aparece cuando un envío tiene la opción de entrega garantizada, el usuario destino debe indicar su PIN de seguridad para indicar que recibió personalmente su envío.

4.9 Pruebas realizadas

Para verificar el correcto funcionamiento de las aplicaciones, se realizaron pruebas de caja negra, sobre las interfaces. Los casos y los resultados de las principales pruebas se presentan en las siguientes tablas.

Desde de la **tabla 2** a la **tabla 8**, se muestran las pruebas realizadas sobre la aplicación web, para verificar la compatibilidad, despliegue y correcto funcionamiento en varias plataformas, las pruebas se realizaron en los siguientes navegadores: Mozilla Firefox, Safari y Google Chrome. Para las pruebas no se utilizó ninguna herramienta de apoyo, se creó un ambiente en AWS para poder realizar las mismas sin interferir con el desarrollo del proyecto, durante la implementación se designó una especialista en el área de calidad, que se encargó de verificar el funcionamiento de los incrementos que se entregaban al finalizar cada Sprint. Es por ello que las fallas se iban corrigiendo durante todo el desarrollo y no al final del mismo.

ID	Rango de entradas	Prueba efectuada	Perfiles de usuarios	Resultado esperado	Resultado obtenido	Pasó / Falló
1	4	Iniciar sesión con usuario y contraseña válida	-	El usuario tiene acceso a la aplicación	El usuario tiene acceso a la aplicación	Pasó
2	4	Iniciar sesión con un usuario registrado y contraseña incorrecta	-	Mensaje: "Usuario y contraseña no coinciden"	Mensaje: "Usuario y contraseña no coinciden"	Pasó
3	5	Iniciar sesión con un usuario no registrado	-	Mensaje: "Usuario no existe"	Mensaje: "Usuario no existe"	Pasó
4	6	Iniciar sesión mediante un correo con una sintaxis inválida	-	Botón de ingresar se bloquea	Botón de ingresar se bloquea	Pasó
5	4	Iniciar sesión con un usuario que no ha completado su registro	-	Mensaje: "Usuario y contraseña no coinciden"	Mensaje: "Usuario y contraseña no coinciden"	Pasó
6	4	Recuperar contraseña mediante un correo con una sintaxis inválida	-	Botón de enviar se bloquea	Botón de enviar se bloquea	Pasó
7	4	Recuperar contraseña con un usuario no registrado	-	Mensaje: "Usuario no existe"	Mensaje: "Usuario no existe"	Pasó

Tabla 2. Pruebas funcionales - Aplicación web. Parte 1

ID	Rango de entradas	Prueba efectuada	Perfiles de usuarios	Resultado esperado	Resultado obtenido	Pasó / Falló
8	6	Recuperar contraseña con un usuario registrado	-	Se envía un correo que permite cambiar la contraseña	Se envía un correo que permite cambiar la contraseña	Pasó
9	4	Crear una compañía ingresando todos los campos obligatorios	Administrador de aplicación	Se crea una nueva compañía	Se crea una nueva compañía	Pasó
10	6	Crear una compañía con datos incompletos	Administrador de aplicación	Botón de agregar se bloquea	Botón de agregar se bloquea	Pasó
11	8	Verificar que los campos del formulario para nuevas compañías funcionen correctamente	Administrador de aplicación	Todos los campos validan la información antes de enviarla al servidor	Todos los campos validan la información excepto el campo de teléfono	Falló
12	14	Agregar un usuario a una compañía	Administrador de aplicación y Administrador de compañía	El usuario recibe un correo electrónico para completar su registro	El usuario recibe un correo electrónico para completar su registro	Pasó
13	6	Agregar un usuario ya existente en una compañía	Administrador de aplicación y Administrador de compañía	Mensaje: " El correo ya está registrado"	Mensaje: " El correo ya está registrado"	Pasó

Tabla 3. Pruebas funcionales - Aplicación web. Parte 2

ID	Rango de entradas	Prueba efectuada	Perfiles de usuarios	Resultado esperado	Resultado obtenido	Pasó / Falló
14	8	Reenviar el correo para completar el registro de un usuario	Administrador de aplicación y Administrador de compañía	El usuario recibe un correo electrónico para completar su registro	El usuario recibe un correo electrónico para completar su registro	Pasó
15	14	Completar registro de usuario	Administrador de aplicación ,Administrador de compañía y Usuario de compañía	El usuario queda registrado junto con las credenciales que agregó al completar su registro	El usuario queda registrado junto con las credenciales que agregó al completar su registro	Pasó
16	7	Encontrar usuarios de una compañía mediante del buscador	Administrador de aplicación ,Administrador de compañía y Usuario de compañía	Al escribir un correo electrónico o nombre de usuario, aparecen los usuarios que coincidan con estos campos	Al escribir un correo electrónico o nombre de usuario, aparecen los usuarios que coincidan con estos campos	Pasó
17	10	Verificar que los campos del formulario para completar registro de un usuario funcionen correctamente	Administrador de aplicación ,Administrador de compañía y Usuario de compañía	Todos los campos validan la información antes de enviarla al servidor	Todos los campos validan la información excepto el campo de teléfono y además el espacio para ingresar el correo electrónico es muy pequeño	Falló
18	12	Asignar un nuevo lote de etiquetas a una compañía	Administrador de aplicación	Lote de etiquetas es asignado a una compañía	Lote de etiquetas es asignado a una compañía	Pasó

Tabla 4. Pruebas funcionales - Aplicación web. Parte 3

ID	Rango de entradas	Prueba efectuada	Perfiles de usuarios	Resultado esperado	Resultado obtenido	Pasó / Falló
19	8	Asignar un lote con etiquetas pertenecientes a otra ya asignado	Administrador de aplicación	Mensaje: " Rango inválido de etiquetas"	Permitió asignar el mismo lote de etiquetas a distintas compañías, sin embargo mostró el mensaje de error cuando se intentó asignar el mismo lote en una misma compañía	Falló
20	12	Buscar una etiqueta perteneciente a un lote asignado a una compañía	Administrador de aplicación	Se muestra la compañía que tiene asignada la etiqueta	Se muestra la compañía que tiene asignada la etiqueta	Pasó
21	8	Buscar una etiqueta que no pertenece a ningún lote asignado	Administrador de aplicación	Mensaje: " Etiqueta no asignada en ningún lote"	Mensaje: " Etiqueta no asignada en ningún lote"	Pasó
22	5	Realizar una factura a una compañía, por el cobro de etiquetas en un periodo de tiempo específico	Administrador de aplicación	Se emite una factura por las etiquetas asignadas en el periodo de tiempo indicado	Se emite una factura por las etiquetas asignadas en el periodo de tiempo indicado	Pasó
23	4	Realizar una factura por el cobro de etiquetas, en un período de tiempo donde no se asignó ninguna	Administrador de aplicación	No se muestran lotes por cobrar	No se muestran lotes por cobrar	Pasó

Tabla 5. Pruebas funcionales - Aplicación web. Parte 4

ID	Rango de entradas	Prueba efectuada	Perfiles de usuarios	Resultado esperado	Resultado obtenido	Pasó / Falló
24	4	Realizar una factura por el cobro de etiquetas, en un periodo de tiempo ya facturado	Administrador de aplicación	Mensaje: " Este rango de fechas ya ha sido facturado. Por favor, busque uno nuevo"	Mensaje: " Este rango de fechas ya ha sido facturado. Por favor, busque uno nuevo"	Pasó
25	8	Crear una sucursal ingresando todos los campos obligatorios	Administrador de aplicación	Una sucursal es creada dentro de una compañía	Una sucursal es creada dentro de una compañía	Pasó
26	4	Crear una sucursal con datos incompletos	Administrador de aplicación	Botón de agregar se bloquea	Botón de agregar se bloquea	Pasó
27	7	Encontrar una sucursal dentro de una compañía mediante el buscador de sucursales	Administrador de aplicación	Al escribir el nombre de la sucursal aparece la sucursal ingresada en el buscador	Al escribir el nombre de la sucursal aparece la sucursal ingresada en el buscador	Pasó
28	6	Ver reporte de compañías y etiquetas	Administrador de aplicación	Se visualiza la cantidad de compañías: activas, en prueba, suspendidas e inactivas. Así como también un contador de etiquetas asignadas	Se visualiza la cantidad de compañías: activas, en prueba, suspendidas e inactivas. Así como también un contador de etiquetas asignadas	Pasó

Tabla 6. Pruebas funcionales - Aplicación web. Parte 5

ID	Rango de entradas	Prueba efectuada	Perfiles de usuarios	Resultado esperado	Resultado obtenido	Pasó / Falló
29	6	Ver reporte de mensajero y etiquetas	Administrador de compañía	Se visualiza la actividad de los mensajeros y la cantidad de etiquetas asignadas y usadas dentro de la compañía	Se visualiza la actividad de los mensajeros y la cantidad de etiquetas asignadas y usadas dentro de la compañía	Pasó
30	28	Crear un envío con una etiqueta válida	Administrador de compañía y Usuario de compañía	Se crea un envío, listo para ser asignado a un mensajero	Se crea un envío, listo para ser asignado a un mensajero	Pasó
31	8	Crear un envío con una etiqueta inválida	Administrador de compañía y Usuario de compañía	Mensaje: " Etiqueta inválida"	Mensaje: " Etiqueta inválida"	Pasó
32	10	Verificar que los campos del formulario para crear un envío funcionen correctamente	Administrador de compañía y Usuario de compañía	Los campos funcionan correctamente y el campo destinatario se autocompleta, buscando usuarios dentro de la compañía	Los campos funcionan correctamente y el campo destinatario se autocompleta, buscando usuarios dentro de la compañía	Pasó
33	18	Asignar un envío a un mensajero	Administrador de compañía y Usuario de compañía	Un mensajero de la compañía queda como asignado al envío	Un mensajero de la compañía queda como asignado al envío	Pasó

Tabla 7. Pruebas funcionales - Aplicación web. Parte 6

ID	Rango de entradas	Prueba efectuada	Perfiles de usuarios	Resultado esperado	Resultado obtenido	Pasó / Falló
34	6	Cancelar un envío	Administrador de compañía y Usuario de compañía	El envío queda completamente anulado	El envío queda completamente anulado	Pasó
35	8	Asignar múltiples envíos a un mensajero, con la funcionalidad de asignación múltiple	Administrador de compañía y Usuario de compañía	Todos los envíos seleccionados son asignados a un mensajero de la compañía	Todos los envíos seleccionados son asignados a un mensajero de la compañía	Pasó
36	4	Editar el perfil de un usuario	-	Todos los campos se pueden editar y guardar	Se pueden editar y guardar los campos de texto, pero la imagen de perfil no se puede cambiar	Falló
37	6	Crear un mensajero	Administrador de aplicación y Administrador de compañía	Se crea un mensajero dentro de una compañía	Con el Administrador de compañía se puede crear el mensajero, pero con el Administrador de aplicación se produce un error	Falló
38	4	Añadir un usuario a la lista de favoritos	Administrador de compañía y Usuario de compañía	El usuario queda agregado a una lista de favoritos y se marca con una estrella amarilla	El usuario queda agregado a la lista de favoritos, pero la estrella no se marca de color amarillo.	Falló
39	8	Buscar un envío por su contenido de referencia en el buscador de envíos	Administrador de compañía Y Usuario de compañía	Al escribir el contenido de referencia en el buscador, aparecen los envíos que coincidan con este campo	Al escribir el contenido de referencia en el buscador, aparecen los envíos que coincidan con este campo	Pasó

Tabla 8. Pruebas funcionales - Aplicación web. Parte 7

En la **tabla 9** se muestran las pruebas realizadas en la aplicación móvil, las pruebas fueron realizadas con los siguientes dispositivos: Blu R1 HD - Android 6.0, Huawei Y-250 - Android 4.4 y Motorola Moto G5 - Android 6.0.

ID	Rango de entradas	Prueba efectuada	Perfiles de usuarios	Resultado esperado	Resultado obtenido	Pasó / Falló
40	16	Notificación de un envío asignado	Mensajero	El dispositivo móvil recibe una notificación con la información del envío asignado	El dispositivo móvil recibe una notificación con la información del envío asignado	Pasó
41	14	Cambiar el estado de un envío	Mensajero	El mensajero puede cambiar el estado de un envío para indicar si ya recibió el paquete o si ya lo entregó	El mensajero puede cambiar el estado de un envío para indicar si ya recibió el paquete o si ya lo entregó	Pasó
42	14	Escanear una etiqueta válida	Mensajero	El envío es verificado y su estado cambia a "En tránsito" para indicar que ya se encuentra en manos del mensajero	El envío es verificado y su estado cambia a "En tránsito" para indicar que ya se encuentra en manos del mensajero	Pasó
43	9	Escanear una etiqueta inválida	Mensajero	Mensaje "Etiqueta inválida"	Mensaje "Etiqueta inválida"	Pasó
44	6	Entregar un paquete con entrega garantizada	Mensajero	En el momento que el mensajero quiera cambiar el estado del envío a "Entregado" el dispositivo móvil solicitará el pin del usuario destinatario, para poder continuar	En el momento que el mensajero quiera cambiar el estado del envío a "Entregado" el dispositivo móvil solicitará el pin del usuario destinatario, para poder continuar	Pasó

Tabla 9. Pruebas funcionales - Aplicación móvil

Es importante destacar que, al comienzo de este desarrollo se implementaron pruebas unitarias, las cuales se utilizaron para garantizar la calidad de los entregables que se hacían en cada Sprint, sin embargo por el corto tiempo que había para hacer el desarrollo, se dejaron de implementar estas pruebas.

4.10 Corrección de fallas y mejoras incorporadas

Luego de realizar las pruebas de funcionalidad, con el fin de corregir posibles errores en las aplicaciones, se atendieron los errores encontrados, a continuación se muestran los ajustes realizados:

ID de la prueba	11
Observación	En el formulario de nuevas compañías, el campo para ingresar el número de teléfono acepta cualquier cantidad de números, cuando solo debería aceptar un prefijo de dos números y diez números para el resto del teléfono
Corrección realizada	Se dividió el campo en dos, uno que acepta un prefijo de dos dígitos y otro que acepta los 10 dígitos restantes del teléfono

ID de la prueba	17
Observación	En el formulario de nuevas compañías, el campo para ingresar el número de teléfono acepta cualquier cantidad de números, cuando solo debería aceptar un prefijo de dos números y diez números para el resto del teléfono
Corrección realizada	Se dividió el campo en dos, uno que acepta un prefijo de dos dígitos y otro que acepta los 10 dígitos restantes del teléfono

ID de la prueba	17
Observación	En el formulario de nuevas compañías, el campo para ingresar el correo electrónico es muy pequeño, lo cual hace incómodo ingresar correos que contengan más de veintidós caracteres
Corrección realizada	Se aumentó el tamaño del campo de correo electrónico

ID de la prueba	19
Observación	La aplicación permite asignar lotes de etiquetas exactamente iguales en distintas compañías, esto es incorrecto puesto que las etiquetas deben ser únicas en toda la aplicación, no solo en las compañías en las cuales son asignadas
Corrección realizada	Se añadieron validaciones extras en el proceso de asignación de lotes de etiquetas, para que la aplicación no permita que existan etiquetas duplicadas

ID de la prueba	36
Observación	No se puede editar la imagen del perfil de un usuario, pues se genera un error cuando se guarda
Corrección realizada	Cuando se realizó el cambio de servidor en el pase de producción, la aplicación web seguía apuntando al servidor de desarrollo, el problema se solventó cambiando el IP del servidor de desarrollo por el dominio de producción

ID de la prueba	37
Observación	Cuando se intenta crear un mensajero con el perfil de un Administrador de aplicación, se produce un error que impide crear al mensajero
Corrección realizada	La aplicación web no se encontraba bien integrada a la ruta de creación de usuarios del API, pues no estaba enviando el correo electrónico del mensajero a crear. Al ser un campo obligatorio el API no permitía crear al mensajero, se ajustó el campo de correo electrónico y con esto se solventó el problema

ID de la prueba	38
Observación	Los usuarios favoritos no quedaban marcados con la estrella amarilla que los identifica
Corrección realizada	Se ajustó la variable que permite marcar o no a un usuario favorito

Al probar la aplicación web y la móvil como un todo se verificaba de forma implícita la correcta integración con el API, puesto que cualquier error al momento de integrar estas tecnologías se ve reflejado en el resultado final. Verificar los requerimientos no funcionales viene de la mano con la selección de tecnologías y servicios. Por ejemplo, Compose y AWS permiten escalar de en cuanto a capacidad de cómputo y almacenamiento, Node.js y MongoDB son tecnologías conocidas por su alto rendimiento y por último seguir las normas de *material design* aseguran que las interfaces sean usables para el usuario final, además que el diseño de las interfaces fue previamente aprobado por la empresa de mensajería.

Resultados

El presente Trabajo Especial de Grado consistió en el desarrollo de una solución informática denominada Aquiles, compuesta por dos aplicaciones que trabajan en conjunto para gestionar el envío de mensajes entre clientes corporativos. La primera solución es una aplicación web a través de la cual se gestionan: compañías, sucursales, etiquetas, envíos, reportes y usuarios; la segunda es una aplicación móvil que permite a los mensajeros gestionar los envíos que les son asignados.

Se espera que estas aplicaciones mejoren los procesos de la empresa involucrada en la investigación en los siguientes aspectos:

- Reducción en los costos y la logística de los clientes corporativos ya que al usar Aquiles tendrán un servicio de envíos a la medida, donde solo pagarán por la cantidad de envíos que necesiten realizar.
- Los empleados de las corporaciones podrán crear y gestionar sus envíos de una forma directa a través de la aplicación web.
- Los administradores de la aplicación web tendrán control sobre las etiquetas que son asignadas a las compañías.
- Los mensajeros podrán gestionar los envíos que tienen asignados, ya que a través de la aplicación móvil tendrán el control de la información y el estado de estos.
- Se aumenta la ventaja competitiva con respecto a otras empresas de envío venezolanas, ya que con Aquiles se reforzarán los procesos necesarios para ofrecer un servicio de envíos a clientes corporativos.

Consideramos que los objetivos planteados en este Trabajo Especial de Grado fueron alcanzados exitosamente debido a las siguientes razones:

- Los principales procesos para la gestión de envíos de clientes corporativos, fueron analizados y modelados, utilizando notación y diagramas BPMN.
- Se utilizó una metodología ágil que permitió realizar el desarrollo de las soluciones en forma eficiente y organizada, específicamente la metodología Scrum.
- Para que las soluciones trabajaran en conjunto se usó una arquitectura basada en un API RESTful, gracias a esto tanto la aplicación web como la móvil pueden gestionar los datos en la solución integrada Aquiles, a través de las rutas que provee el API.
- Las interfaces de ambas aplicaciones siguen los lineamientos de *material design*, con lo cual se logra tener una interfaz intuitiva y agradable a la vista.
- El API fue documentado con Swagger, una librería que permitió documentar cada ruta que posee el API, para así describir como se utilizan y que funcionalidades tienen.

- Para verificar el correcto funcionamiento de Aquiles, se realizaron pruebas de funcionalidad, para luego corregir los errores encontrados.

Conclusiones

Este Trabajo Especial de Grado, surge de la necesidad de materializar una alternativa más eficiente para la gestión de envíos internos de clientes corporativos, quienes presentan complicaciones al momento de encargarse por su cuenta de estos procedimientos, debido a los costos elevados en el mantenimiento del personal, logística, materia prima, entre otros.

Para la solución de esta problemática se desarrollaron dos aplicaciones que trabajan en conjunto, una aplicación web y una móvil, haciendo uso de la metodología Scrum. Esta metodología ágil de desarrollo de software fue considerada la más adecuada debido a las características del equipo de desarrollo en la empresa participante en el estudio y por la necesidad de hacer entregas parciales en cortos períodos de tiempo. Aunado a esto, la naturaleza flexible de Scrum permitió realizar modificaciones en una etapa avanzada del desarrollo, donde por requerimientos iniciales poco específicos o no completados por parte cliente, se implementaron funcionalidades cuyo comportamiento no era el deseado. A manera de ejemplo, las etiquetas en un principio se implementaron para que fueran únicas en cada compañía cuando realmente el cliente deseaba que estas fueran únicas en toda la aplicación. Incorporar estos cambios en el desarrollo fue bastante sencillo, los ajustes se agregaban a la pila del producto y si el objetivo del Sprint se lograba antes de tiempo, se tomaban los ajustes pendientes que estuviesen en la Lista del Producto, según su prioridad.

Respecto a las tecnologías, la selección de MongoDB, AngularJs, Express y Node para el desarrollo de la aplicación web fue bastante acertada. En la actualidad, es bastante común que se usen en conjunto, dado que estas tecnologías se integran bastante bien entre sí. Una gran ventaja de escoger las tecnologías mencionadas es que tienen una comunidad de desarrollo muy grande, lo cual es de gran utilidad al momento de hacer una investigación, pues las mayoría de los desarrolladores tienen dudas en común, las cuales ya están resueltas por expertos, durante este desarrollo surgieron muchas dudas como por ejemplo: ¿Cómo realizar una búsqueda predictiva en MongoDB? ¿Cómo funciona la agregación en MongoDB?, entre otras más que fueron resueltas sin problema alguno gracias a la comunidad de desarrollo y la buena documentación. Además se contaba con experiencia parcial en el uso de algunas de estas tecnologías, complementando el conocimiento durante el desarrollo del TEG.

Por otra parte, la selección de Android como tecnología base para la aplicación móvil también fue adecuada, Android cuenta con una buena documentación y una gran comunidad de desarrollo. Ahora bien es importante resaltar que esta tecnología fue parte de los requerimientos solicitados por la empresa de mensajería, pues era necesario que la aplicación móvil se pudiera ejecutar en dispositivos con precios bajos, por esta razón no se optó por hacer un desarrollo IOS o un desarrollo híbrido en su defecto. También se tenía conocimiento previo sobre esta tecnología.

Para finalizar, se logró solventar la problemática con la gestión de envíos internos en las grandes corporaciones, ofreciéndoles un servicio de envíos a la medida, gracias al desarrollo de las aplicaciones tecnológicas descritas, integradas en la solución Aquiles.

Recomendaciones y Trabajos Futuros

Con la finalidad de aumentar la eficiencia de Aquiles, se podrían incorporar las siguientes mejoras en desarrollos futuros:

- Ampliar la sección de etiquetas complementando su funcionamiento por medio de la creación de una subsección, que permita crear las etiquetas directamente desde esta herramienta. De esta manera, podrán ser asignadas posteriormente a las compañías registradas dentro de la aplicación web. Es deseable que en esta subsección se puedan exportar los lotes de etiquetas creadas como formato .pdf o .doc, para que luego se puedan imprimir como códigos QR. Agregando esta subsección se lograría automatizar más la gestión de envíos para los clientes corporativos, además que es ideal que toda la lógica de negocio se encuentre en un solo lugar, en este caso la herramienta Aquiles.
- Con fin de tener más control sobre los envíos que se realizan dentro de las corporaciones, se puede añadir más información a los detalles de los envíos, como por ejemplo:
 - Historial para los cambios de estado de un envío, donde se puedan visualizar datos como: Fecha y hora en que cambió el estado del envío, y quien realizó el cambio de estado del mismo.
 - Historial de asignaciones de mensajeros a un envío, donde se puedan visualizar datos como: Fecha y hora en que se asignó un mensajero a un envío.
- En la sección de reportes de mensajeros, sería de utilidad añadir un histórico de envíos asignados a un mensajero, de esta manera sería sencillo saber cuáles envíos a gestionado un mensajero en particular.
- Es posible hacer que los envíos una vez que sean creados, se asignen a un mensajero de forma automática e inteligente, incorporando geolocalización a la aplicación móvil entre otras informaciones, de esta forma se puede determinar cuál es el mensajero más indicado para hacerse cargo de un envío, según la cercanía y la carga de asignaciones que posea en el momento.

Referencias bibliográficas y digitales

- Alliance, O. H. (2017). *Open Handset Alliance*. Obtenido de https://www.openhandsetalliance.com/android_overview.html
- Amazon. (2017). *Amazon web services*. Obtenido de <https://aws.amazon.com/ec2/>
- Android. (2017). *Android Deevelopers*. Obtenido de <https://developer.android.com/guide/components/fundamentals.html>
- Angularjs. (s.f.). *AngularJS — Superheroic JavaScript MVW Framework*. Recuperado el 16 de Junio de 2016, de Angularjs.org: <https://github.com/angular/angular.js>
- Canós, J. H. (s.f.). *Metodologías Ágiles en el Desarrollo de Software*. Valencia: Universidad Politécnica de Valencia.
- Chacaltana, G. (2014). *Una breve cronología de MySQL*. Recuperado el 20 de Junio de 2016, de <http://www.solocodigoweb.com/blog/2014/03/04/una-brevecronologia-de-mysql/>
- Clarke, S. (2004). *Measuring API Usability*.
- CMS. (27 de Marzo de 2008). Recuperado el 02 de Agosto de 2016, de cms: <https://www.cms.gov/Research-Statistics-Data-and-Systems/CMS-Information-Technology/XLC/Downloads/SelectingDevelopmentApproach.pdf>
- Compose. (2017). *Compose*. Obtenido de <https://www.compose.com/>
- Coursera. (Agosto de 2016). Recuperado el 10 de Agosto de 2016, de <https://www.coursera.org/>
- Ecured. (s.f.). Recuperado el 11 de Mayo de 2016, de Ecured: http://www.ecured.cu/Servidor_Web
- Ecured.cu. (s.f.). *Framework*. Recuperado el 20 de Julio de 2016, de <http://www.ecured.cu/Framework>
- Ecured.cu. (s.f.). *Servidor Web*. Recuperado el 16 de Junio de 2016, de http://www.ecured.cu/Servidor_Web
- Escuela Web. (s.f.). Recuperado el 02 de Agosto de 2016, de <https://escuelaweb.net/>
- Expressjs. (Jun de 2017). *Expressjs*. Obtenido de <http://expressjs.com>
- FedEx. (2017). *FedEx*. Recuperado el 20 de 08 de 2017, de FedEx: <http://www.fedex.com/us/service-guide/prepare-shipment/packing/express-ground/index.html>
- Fielding, R. T. (2000). *Architectural Styles and the Design of Network-based Software Architectures*. California.
- FROAC. (2012). Obtenido de Federación de Repositorios de Objetos de Aprendizaje Colombia: <http://froac.manizales.unal.edu.co/gloar/lib/images/mvc.png>
- Foundation, N. (june de 2017). *node.js*. Obtenido de <https://nodejs.org>
- Git - Acerca del control de versiones. (s.f.). Recuperado el 16 de Junio de 2016, de Git-scm.com: <https://git-scm.com/book/es/v1/Empezando-Acerca-del-control-de-versiones>
- Git-scm. (s.f.). <https://git-scm.com/>. Recuperado el 16 de Junio de 2016, de git-scm.com.

- Gomez, J. (s.f.). *Patrón de arquitectura Modelo Vista Controlador (MVC)*. Recuperado el 16 de Junio de 2016, de Lab.inf.uc3m.es: <http://www.lab.inf.uc3m.es/~a0080802/RAI/mvc.html>
- GulpJs. (s.f.). Recuperado el 11 de Mayo de 2016, de <http://gulpjs.com/>
- jQuery. (s.f.). Recuperado el 11 de Mayo de 2016, de <https://jquery.com/>
- jQuery jquery.org. (s.f.). *jQuery*. Recuperado el 07 de Junio de 2016, de Jquery.com: <https://jquery.com/>
- JSON. (2008). *JSON*. Recuperado el June de 2017, de <http://json.org/>
- Laravel. (s.f.). Recuperado el 20 de Julio de 2016, de GitHub: <https://github.com/laravel/laravel>
- Libuv. (jun de 2017). *libuv*. Obtenido de <http://libuv.org/>
- LITH, A. (2010). Investigating storage solutions for large data.
- Logística, L. (2017). *Lead Logística*. Recuperado el 20 de 08 de 2017, de Lead Logística: <http://www.leadlogistica.com/>
- Mandrill. (2017). *Mandrill*. Recuperado el 01 de Septiembre de 2017, de <https://www.mandrill.com/>
- MongoDB. (2017). *mongodb*. Obtenido de <https://www.mongodb.com/nosql-explained>
- Mozilla Developer Network. (s.f.). *AJAX*. Recuperado el 08 de Julio de 2016, de <https://developer.mozilla.org/es/docs/AJAX>
- Mozilla Developer Network. (s.f.). *CSS*. Recuperado el 16 de Junio de 2016, de Mozilla Developer Network: <https://developer.mozilla.org/es/docs/Web/CSS>
- Mozilla Developer Network. (s.f.). *HTML*. Recuperado el 16 de Junio de 2016, de Mozilla Developer Network: <https://developer.mozilla.org/es/docs/Web/HTML>
- Mozilla Developer Network. (s.f.). *HTTP*. Recuperado el 16 de Junio de 2016, de <https://developer.mozilla.org/es/docs/Web/HTTP>
- Mozilla Developer Network. (s.f.). *Javascript*. Recuperado el 16 de Junio de 2016, de Mozilla Developer Network: <https://developer.mozilla.org/es/docs/Web/JavaScript>
- MRW. (2017). *MRW*. Recuperado el 20 de 08 de 2017, de MRW: <http://www.mrw.com.ve/Servicios.aspx>
- MySQL. (s.f.). Recuperado el 16 de Junio de 2016, de <https://www.mysql.com/>
- MySQL. (s.f.). *MySQL :: MySQL Workbench*. Recuperado el 21 de Julio de 2016, de Mysql.com: <https://www.mysql.com/products/workbench/>
- Otto, M. (s.f.). *Bootstrap · The world's most popular mobile-first and responsive front-end framework*. Recuperado el 16 de Junio de 2016, de Getbootstrap.com: <https://getbootstrap.com/>
- PHP Group. (s.f.). *PHP: Prefacio - Manual*. Recuperado el 25 de Junio de 2016, de php.net: <https://secure.php.net/manual/es/preface.php>
- Proyectos Ágiles. (2015). Recuperado el 10 de Agosto de 2016, de <https://proyectosagiles.org/que-es-scrum/>
- RAE. (Octubre de 2014). *Real Academia Española*. Obtenido de RAE: <http://www.rae.es>
- Robomongo. (June de 2017). *Robomongo*. Obtenido de <https://robomongo.org/>
- Sanchez, J. (31 de Agosto de 2014). *Gulp.js en español - tutorial básico y primeros pasos*. Recuperado el 20 de Julio de 2016, de FrontEnd Labs: <https://frontendlabs.io/1669--gulp-js-en-espanol-tutorial-basico-primeros-pasos-y-ejemplos>

- Sass-lang. (s.f.). *SASS_REFERENCE — Sass Documentation*. Recuperado el 05 de Julio de 2016, de Sass-lang.com: http://sass-lang.com/documentation/file.SASS_REFERENCE.html
- Schwaber, K. (Julio de 2013). *Scrum Guides*. Recuperado el 10 de Agosto de 2016, de <http://www.scrumguides.org/docs/scrumguide/v1/Scrum-Guide-ES.pdf>
- Scrum.org*. (s.f.). Recuperado el 10 de Agosto de 2016, de www.scrum.org
- Silberschatz A., K. H. (2002). *Fundamentos de Bases de Datos*. (4ta ed.). (M. G. Hill, Ed.) Madrid, España.
- Swagger. (june de 2017). *Swagger.io*. Obtenido de <https://swagger.io/swagger-ui/>
- W3school*. (2013). Recuperado el 11 de Mayo de 2016, de <http://www.w3schools.com/bootstrap/>

Glosario de términos

- **Correo:** Es un servicio público que tiene por objeto el transporte de la correspondencia oficial y privada (RAE, 2014).
- **Envío:** Según la (RAE, 2014) es la acción y efecto de enviar, también es sinónimo de remesa, que significa conjunto de cosas enviadas o recibidas de una vez, es por esto que se hará uso de este término, para referirse a cosas que necesiten ser enviadas por los usuarios de la aplicación.
- **Mensajero:** Persona que tiene por oficio llevar y traer la correspondencia de un lugar a otro (RAE, 2014), en el caso de la aplicación estas personas son contratadas por los dueños de la misma, con el fin de que se encarguen de transportar envíos dentro de las compañías y sucursales afiliadas.