FUNDACION UNIVERSITARIA LOS LIBERTADORES

FACULTAD DE INGENIERÍA Y CIENCIAS BÁSICAS

INGENIERÍA DE SISTEMAS

PROTOTIPO SOFTWARE PARA LA ENTREGA DE DOMICILIOS

(Quick Deliver)

SEBASTIÁN CAMILO SANABRIA BOCANEGRA

HUGO FERNEY CELIS SOTELO

BOGOTÁ D.C.

2.019

FUNDACION UNIVERSITARIA LOS LIBERTADORES

FACULTAD DE INGENIERÍA Y CIENCIAS BÁSICAS

INGENIERÍA DE SISTEMAS

PROTOTIPO SOFTWARE PARA LA ENTREGA DE DOMICILIOS

(Quick Deliver)

ELABORADO POR:

SEBASTIÁN CAMILO SANABRIA BOCANEGRA

HUGO FERNEY CELIS SOTELO

Trabajo de grado presentado para optar al título de

Ingeniero de Sistemas

Asesor:

JAVIER DAZA PIRAGAUTA

BOGOTÁ D.C.

2.019

LOS LIBERTADORES
FUNDACIÓN UNIVERSITARIA

Nota de Aceptación

Firma del jurado

Firma del Jurado

Director (a) de Tesis

Bogotá D.C., Julio de 2.019

LOS LIBERTADORES
FUNDACIÓN UNIVERSITARIA

DEDICATORIA

En primera instancia a Dios por permitirnos la oportunidad de culminar nuestra carrera por vocación hacia lo que nos gusta.

Nuestros padres que estuvieron en cada uno de los instantes de la carrera apoyándonos en todo momento con sus consejos, paciencia y amor para cumplir nuestro deseo del cual fue posible realizarlo.

LOS LIBERTADORES
FUNDACIÓN UNIVERSITARIA

AGRADECIMIENTOS

A Javier Daza Piragauta, Ingeniero de sistemas, director del proyecto, por el apoyo durante el desarrollo de este trabajo de grado.

A nuestros profesores docentes que nos acompañaron aportándonos los conocimientos necesarios para la finalización de nuestra carrera en cada una de sus enseñanzas como profesionales y personas integras.

LOS LIBERTADORES
FUNDACIÓN UNIVERSITARIA

Tabla de Contenido

Pág.

1. ASPECTOS DE LA INVESTIGACIÓN 16

1.1. PLANTEAMIENTO DEL PROBLEMA 16

1.2. DESCRIPCIÓN DEL PROBLEMA 16

1.3. JUSTIFICACIÓN DEL PROBLEMA 17

1.4. DELIMITACIÓN DEL PROBLEMA 18

1.5. OBJETIVOS 18

1.5.1. Objetivo General 18

1.5.2. Objetivos Específicos 19

2. MARCO REFERENCIAL 20

2.1. MARCO TEÓRICO 20

2.1.1. UML (Unified Modeling Language) 20

2.1.2. Tipos de Programación 20

2.1.3. IDE (Interated Development Enviroment) 21

2.1.4. Software 21

2.1.4.1. Lenguajes Utilizados 21

2.1.4.1.1. HTML 22

2.1.4.1.2. CSS 22

2.1.4.1.3. JavaScript 23

2.1.4.1.4. TypeScript 24

2.1.4.1.5. Framework 24

2.1.4.1.6. MVC 25

2.1.4.1.7. SPA 27

2.1.4.1.8. Angular 27

2.1.4.1.9. Node.JS 28

2.1.4.1.10. NPM 29

2.2. ANTECEDENTES TEÓRICOS 29

2.2.1. Legales 29

2.3. MARCO CONCEPTUAL 30

2.3.1. Comercio Electrónico 30

2.3.2. Problema 31

3. DESARROLLO DEL PROYECTO 32

3.1. METODOLOGÍAS DE DESARROLLO 32

3.1.1. GIT 33

3.1.2. SCRUM 34

4. LEVANTAMIENTO DE REQUERIMIENTOS 35

4.1. REQUERIMIENTOS DEL SISTEMA 35

4.1.1. Requerimientos Funcionales 35

4.1.2. Requerimientos no Funcionales 38

4.2. DISEÑO DE APLICACIÓN 41

4.2.1. Diagrama De Clases 41

4.2.2. Diagrama De Componentes 42

4.2.3. Diagrama De Implementación 43

4.2.4. Diagrama Uml Comportamiento 44

4.2.5. Diagrama Casos De Uso 44

4.2.6. Diagrama De Secuencia 45

4.2.7. Diagrama De Actividades 45

4.3. IMPLEMENTACIÓN 46

5. MANUALES 47

5.1. MANUAL DE USUARIO 47

5.2. MANUAL USUARIO ADMIN 50

5.3. MANUAL TÉCNICO 55

5.4. MANUAL MONTAJE SOFTWARE 57

6. CONCLUSIONES 63

REFERENCIAS BIBLIOGRÁFICAS 64

ANEXOS 65

Lista de Tablas

Pág.

Tabla 1. Versiones Software a usar 21

Tabla 2. Paquetes de node a usar 29

Tabla 3. Requerimiento Funcional RF01 35

Tabla 4. Requerimiento Funcional RF02 36

Tabla 5. Requerimiento Funcional RF03 36

Tabla 6. Requerimiento Funcional RF04 37

Tabla 7. Requerimiento Funcional RF05 37

Tabla 8. Requerimiento Funcional RF06 38

Tabla 9. Requerimiento No Funcional RNF01 38

Tabla 10. Requerimiento No Funcional RNF02 39

Tabla 11. Requerimiento No Funcional RNF03 39

Tabla 12. Requerimiento No Funcional RNF04 40

Tabla 13. Requerimiento No Funcional RNF05 40

LOS LIBERTADORES
FUNDACIÓN UNIVERSITARIA

Lista de Imágenes

Pág.

Imagen 1. Ingreso a la aplicación 47

Imagen 2. Listado de pedidos 48

Imagen 3. Creación de los pedidos 48

Imagen 4. Ingreso del domiciliario 49

Imagen 5. Listado de pedidos asignados 49

Imagen 6. Notificación de novedades 50

Imagen 7. Ingreso administrador de la cuenta 50

Imagen 8. Creación de usuarios 51

Imagen 9. Listado de usuarios agregados 51

Imagen 10. Listado de pedidos solicitados 52

Imagen 11. Creación de la zonas 52

Imagen 12. Asignación de zonas 53

Imagen 13. Listado de zonas 53

Imagen 14. Ingreso a la zona 54

Imagen 15. Asignación de la zona 54

Imagen 16. Pantalla ingreso a Google Play 60

Imagen 17. Configuración Google Play 61

LOS LIBERTADORES
FUNDACIÓN UNIVERSITARIA

Lista de Figuras

Pág.

Figura 1. Modelo Vista Controlador 26

Figura 3. Procesos en Scrum 34

Figura 4. Diagrama de clases 41

Figura 5. Diagrama de componentes 42

Figura 6. Diagrama de implementación 43

Figura 7. Diagrama de casos de uso 44

Figura 8. Diagrama de secuencia 45

Figura 9. Diagrama de actividades 45

LOS LIBERTADORES
FUNDACIÓN UNIVERSITARIA

ABSTRACT

Con la incursión de la tecnología que se ha presentado en los últimos años ha sido de forma necesaria que las empresas que ofrecen sus servicios a clientes por medio de aplicaciones para ofrecer sus servicios de comida y que sean de forma accesible y/o de fácil manejo. Aquí en Colombia se tiene una serie de aplicaciones para el servicio de domicilio, donde cada uno de los clientes puede recurrir a solicitar comida o bebidas de su preferencia desde su hogar, oficina u otra ubicación.

La aplicación que se desarrolla está en busca de la innovación del producto de domicilios, en esta se busca que la empresa del Grupo Cosechar ofrezca sus servicios de domiciliarios a cada empresa que esté en búsqueda de mejorar sus tiempos de entrega y atención al usuario cliente que solicita los servicios.

Por tal razón se define que esta aplicación o recurso tecnológico estará a la vanguardia y al alcance de las empresas que se asocien a la empresa Grupo Cosechar para el servicio domiciliario.

En el presente documento se presentará la información de investigación, desarrollo, metodologías y requerimientos para el óptimo funcionamiento para cada una de las empresas asociadas.

Palabras Clave: App, Metodologías, comercio electrónico, Consumidores, Mercado, java, CSS, HTML.

LOS LIBERTADORES
FUNDACIÓN UNIVERSITARIA

ABSTRACT

With the incursion of technology that has been presented in recent years, it has been necessary for companies that offer their services to customers through applications to offer their food services and that are accessible and / or easy to use. . Here in Colombia there is a series of applications for the domicile service, where each of the clients can resort to request food or drinks of their choice from their home, office or other location.

The application that is developed is looking for the innovation of the product of domiciles, in this one it looks for that the company of the Cosechar Group offers its services of domiciliary to each company that this in search of improving its times of delivery and attention to the user client that request the services.

For this reason it is defined that this application or technological resource will be at the forefront and within reach of the companies that are associated with the company Group Cosechar for home service.

In this document we will present the research, development, methodologies and requirements for optimal operation for each of the associated companies.

Keywords: App, Methodologies, e-commerce, Consumers, Market, java, CSS, HTML.

INTRODUCCIÓN

El mundo de hoy vive conectado, con la incursión de la tecnología todas las personas tienen a disposición un teléfono celular que le permite incursionar en todo tipo de procesos, que unos años atrás eran complicados de hacer. Las personas tenían que salir de sus casas hacer filas o muchas veces tener que esperar días para realizar un trámite, o tenía limitaciones a la hora de comprar, dado que solo podían tener la visual de una tienda o de las que estuvieran cerca de su ubicación. Hoy en día con la facilidad de tener los teléfonos inteligentes estos procesos se han vuelto más fáciles de manejar debido a que se encuentran conectados a la red, actividades como pagar un recibo, consultar saldos bancarios, realizar transferencias electrónicas entre otras cosas sencillas como realizar una compra y que llegue a la puerta de la casa, etc.

De esta forma nace la necesidad de crear una plataforma móvil para la solicitud de domiciliarios. Las grandes y pequeñas empresas que deseen adquirir este servicio con la empresa Grupo Cosechar, compañía a la que se quiere impactar tecnológicamente en el proceso en el que ofrecerá domiciliarios para la entrega de los productos comprados por medio de la plataforma. Es necesario que no sea un problema a la horaen el momento de la solicitud y empresas asociadas como McDonald’s, Presto, el corral y demás empresas puedan facilitar el proceso de entrega de sus ventas. Tan solo con tener la aplicación del Grupo cosechar puedan solicitar un domiciliario y brindar un mejor servicio y más rápido a cada cliente usuario que requiere de los servicios para dar por terminado el proceso de su venta.

El método empleado para llevar a cabo este proyecto será el diseñar un prototipo utilizando tecnologías ágiles como Scrum, levantamiento de requerimientos y lenguajes de desarrollo que permitan elaborar un proyecto como el que es requerido por la empresa Grupo Cosechas. Más sin embargo como en todo proyecto se tiene la limitación de tiempo dado que el prototipo se diseñará dentro del semestre académico en curso.

1. ASPECTOS DE LA INVESTIGACIÓN

1.1. PLANTEAMIENTO DEL PROBLEMA

La empresa Grupo Cosechar desea implementar una plataforma tecnológica virtual que permita ofrecer el servicio principal de domiciliario a empresas que desean mejorar la atención en la entrega de sus productos, por tanto, en este proyecto se elabora la idea y el prototipo de la aplicación móvil y página web para la administración de solicitudes, yque empresas como KFC, Prestó, el Corral entre otras empresas que ofrecen comida, u otros productos y servicios. Soliciten los servicios de domiciliario por medio de la App y que se reduzcan tiempos y se mejore la calidad en la entrega del producto para dar por terminada la venta en el momento de entregar el pedido al cliente (Usuario) por parte del domiciliario asignado porla empresa Grupo cosechar.

1.2. DESCRIPCIÓN DEL PROBLEMA

En Colombia existen varias aplicaciones para domicilios, cada persona que tiene a disposición y facilidad para el manejo de las aplicaciones usan este tipo de APP´S.Por ello las grandes empresas de comidas o servicios han acudido a empresas externas (terceros) para realizar la entrega de sus pedidos, en Bogotá nacieron aplicaciones para domicilios, dado la necesidad de suplir la gran demanda que al pasar los días sigue en aumento facilitando a los usuarios y evitando desplazamientos para realizar tareas que las tienen a una simple solicitud.

Teniendo en cuenta lo anterior tenemos factores que influyen en el diseño del prototipo que se mostrará al Grupo Cosechar, como las distancias largas, los trancones, el horario que tendrían **que** tener los domiciliarios para poder cubrir toda la demanda, las prestaciones sociales que ellos infieren y que deben ser tenidas en cuenta por la empresa Grupo Cosechar.

1.3.JUSTIFICACIÓN DEL PROBLEMA

El impacto Social que el prototipo diseñado al que se llamara para los clientes y empresas asociadas Quick Deliver, generará oportunidades de empleo, la competencia en que el consumidor tendrá más opciones a la hora de elegir sus compras, dado que el proceso de entrega de los productos es transparente para los hipermercados. Esto también tendrá un impacto económico en las ventas que tienden a crecer dada la demanda y facilidad para adquirir productos. Los mercados y comerciantes no tienen que preocuparse por la entrega de los productos y el proceso de entrega dada la gran cantidad de demanda la técnica que usa Quick Deliver es diseñar el modelo de negocio en el cual el domiciliario puede ingresar a la plataforma móvil y estar listo para la asignación del domicilio, en el momento de alguna empresa haga una solicitud y sea asignado según la ubicación más cercana. Por ello el impacto organizacional es alto dado que las empresas que ofrecen comidas, productos y servicios no se preocupan por que solo tienen un domiciliario y la demanda es alta. La metodología a usar es que a través de la solución

tecnológica solo solicita cuantos domiciliarios sean necesarios para cubrir la demanda y optimizar las entregas.

1.4. DELIMITACIÓN DEL PROBLEMA

El limitante principal se basa en costos de diseño del prototipo, dado que el tiempo de elaboración es directamente proporcional al tiempo en que se debe entregar el proyecto documentado a la Fundación Universitaria Los Libertadores.

1.5. OBJETIVOS

1.5.1. Objetivo General

Diseñar un prototipo a la empresa Grupo Cosechar en el que permita administrar y realizar un seguimientode domiciliarios en el momento de solicitudes por parte de las empresas asociadas.

1.5.2. Objetivos Específicos

- Realizar levantamiento de requerimientos de las áreas implicadas de la empresa Grupo Cosechar.
- Diseñar módulo para la empresa que solicita el domiciliario.
- Elaborarpágina web para la administración de solicitudes por parte de las empresas asociadas.
- Diseñar módulo para el manejo del domiciliario en las entregas de los servicios asignados.

2. MARCO REFERENCIAL

2.1.MARCO TEÓRICO

2.1.1. UML (Unified Modeling Language)

El lenguaje de modelado estandarizado UML es un lenguaje que utiliza notaciones gráficas para expresar a gran escala el diseño de una aplicación o software es desarrollado para ayudar a las personas que desarrollan y diseñan aplicaciones ayudando así detallar , divisar, levantar y documentar todas aquellas partes que son importantes para el buen desarrollo de cualquier proyecto. UML a través de sus diagramas da ideas claras de cómo empezar y fluir a través del desarrollo del proyecto dado que tiene en la visual sus componentes, comunicaciones con otras aplicaciones, modelos de uso de todos los stakeholders y sus interacciones con la aplicación o software así como todos los proceso y árboles que el sistema puede perseguir entre otros aspectos que este modelo usa para tener una visual muy completa del proyecto(Qué es el lenguaje unificado de modelado (UML), 2019).

2.1.2. Tipos de Programación

La programación es básicamente definir una instrucciones organizadas y estructuradas adecuadamente para que la computadora la entienda existen diferentes tipos de programación como la programación estructurada es cuando la aplicación es desarrollada ordenadamente tiene un principio y un fin y está desarrollada de forma secuencial y por módulos, otra forma de programar en la programación orientada a

objetos la cual está desarrollada en técnicas como cohesión, abstracción encapsulamiento y polimorfismo en ella los objetos tienen atributos y clases y gracias ella se crean relaciones entre estos objetos(Qué es el lenguaje unificado de modelado (UML), 2019)

2.1.3. IDE (Interated Development Enviroment)

Es un entorno de desarrollo integrado es un programa creado para ayudar a programar que contiene editor de texto que nos permite indentar, autocompletar resaltado de errores, también contienen un plano depurador, un compilador y un diseñador de entornos gráficos entre otras funcionalidades (IDE de programación, 2012)

2.1.4. Software

2.1.4.1. Lenguajes Utilizados

Los siguientes serán los lenguajes, entornos y frameworks a usar en el desarrollo del proyecto del cual se está tratando.

Tabla 1. Versiones Software a usar

Lenguaje o Framework	**Versión**
Ionic	3
Node.js	>= 10.0.0
HTML	5
Angular	6
Css	3
JavaScript	6 ó ES6

Fuente: Los autores

2.1.4.1.1. HTML

HTML (*HyperText Markup Language*) es un lenguaje muy sencillo que permite describir hipertexto, es decir, texto presentado de forma estructurada y agradable, con enlaces (hyperlinks) que conducen a otros documentos o fuentes de información relacionadas, y con inserciones multimedia (gráficos, sonido...). La descripción se basa en especificar en el texto la estructura lógica del contenido (títulos, párrafos de texto normal, enumeraciones, definiciones, citas, etc.) así como los diferentes efectos que se quieren dar (cursiva, negrita, o un gráfico determinado).

Es el lenguaje utilizado en Internet para definir las páginas del World Wide Web (www). Los archivos HTML son archivos de texto puramente ASCII, que pueden ser escritos con cualquier editor básico, tal como Notepad en Windows o vi en Unix. En este archivo de texto se introducen unas marcas o caracteres de control llamadas TAGs (en esto, HTML se parece a los primeros procesadores de texto), que son interpretadas por el navegador. Cuando éste lee un fichero ASCII con extensión *.htm o *.HTML interpreta estos TAGs y formatea el texto de acuerdo con ellas(Biología e ideología 1, s.f.).

2.1.4.1.2. CSS

Cascading Style Sheets (CSS), traducido literalmente al español, es un lenguaje de hojas de estilos creado para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML y XHTML. CSS es la mejor forma de separar los contenidos y su presentación y es imprescindible para crear páginas web complejas.

Separar la definición de los contenidos y la definición de su aspecto presenta numerosas ventajas, ya que obliga a crear documentos HTML/XHTML bien definidos y con significado completo (también llamados "documentos semánticos"). Además, mejora la accesibilidad del documento, reduce la complejidad de su mantenimiento y permite visualizar el mismo documento en infinidad de dispositivos diferentes.

Al crear una página web, se utiliza en primer lugar el lenguaje HTML/XHTML para marcar los contenidos, es decir, para designar la función de cada elemento dentro de la página: párrafo, titular, texto destacado, tabla, lista de elementos, etc.

Una vez creados los contenidos, se utiliza el lenguaje CSS para definir el aspecto de cada elemento: color, tamaño y tipo de letra del texto, separación horizontal y vertical entre elementos, posición de cada elemento dentro de la página, etc.(¿Qué es CSS?, 2019).

2.1.4.1.3. JavaScript

JavaScript es un lenguaje de programación que se utiliza principalmente para crear páginas web dinámicas.

Una página web dinámica es aquella que incorpora efectos como texto que aparece y desaparece, animaciones, acciones que se activan al pulsar botones y ventanas con mensajes de aviso al usuario.

Técnicamente, JavaScript es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos. En otras palabras, los programas

escritos con JavaScript se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios.

A pesar de su nombre, JavaScript no guarda ninguna relación directa con el lenguaje de programación Java. Legalmente, JavaScript es una marca registrada de la empresa Sun Microsystems, como se puede ver en http://www.sun.com/suntrademarks/.

2.1.4.1.4. TypeScript

Es un lenguaje de programación libre y de código abierto desarrollado y mantenido por Microsoft. Es un súper conjunto de JavaScript, que esencialmente añade tipado estático y objetos basados en clases. TypeScript puede ser usado para desarrollar aplicaciones JavaScript que se ejecutarán en el lado del cliente o del servidor (Node.js).

TypeScript extiende la sintaxis de JavaScript, por tanto cualquier código JavaScript existente debería funcionar sin problemas. Está pensado para grandes proyectos, los cuales a través de un compilador de TypeScript se traducen a código JavaScript original.

2.1.4.1.5. Framework

Es un conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular que sirve como referencia, para enfrentar y resolver nuevos problemas de índole similar.

En el desarrollo de software, un entorno de trabajo es una estructura conceptual y tecnológica de asistencia definida, normalmente, con artefactos o módulos concretos de software, que puede servir de base para la organización y desarrollo de software. Típicamente, puede incluir soporte de programas, bibliotecas, y un lenguaje interpretado, entre otras herramientas, para así 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, y provee una estructura y una especial metodología de trabajo, la cual extiende o utiliza las aplicaciones del dominio.

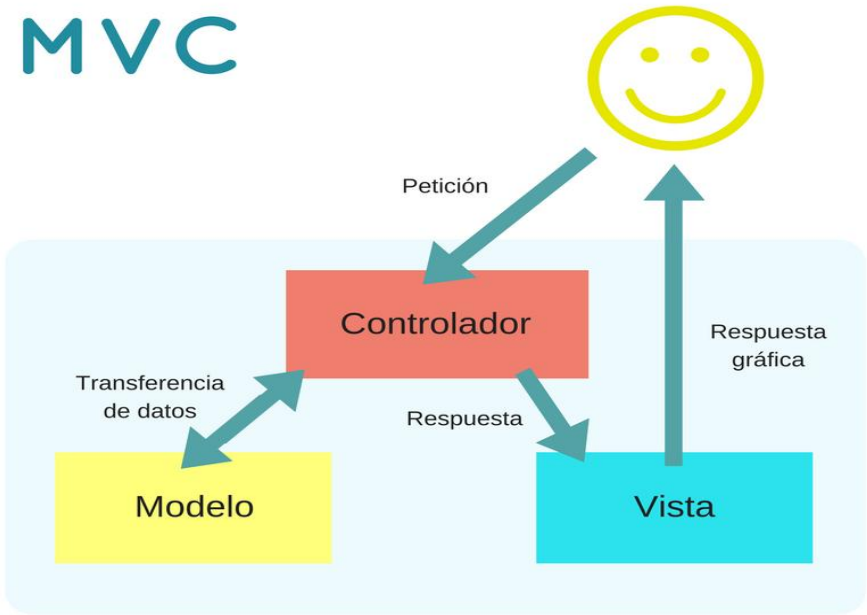
2.1.4.1.6. MVC

El Model-View-Controller (MVC) es un patrón arquitectónico que separa una aplicación en tres componentes lógicos principales: el modelo, la vista y el controlador. Cada uno de estos componentes está diseñado para manejar aspectos específicos de desarrollo de una aplicación. MVC es uno de los marcos de desarrollo web estándar de la industria más utilizados para crear proyectos escalables y extensibles.

- **Modelo:** el componente del modelo corresponde a toda la lógica relacionada con los datos con la que trabaja el usuario. Esto puede representar los datos que se transfieren entre los componentes de View y Controller o cualquier otro dato relacionado con la lógica de negocios. Por ejemplo, un objeto Cliente recuperará la información del cliente de la base de datos, la manipulara y actualizará los datos a la base de datos o la usará para representar los datos.

- **Vista:** El componente Vista se utiliza para toda la lógica de la interfaz de usuario de la aplicación. Por ejemplo, la vista del Cliente incluirá todos los componentes de la IU, como cuadros de texto, menús desplegables, etc., con los que el usuario final interactúa.
- **Controlador**: Los controladores actúan como una interfaz entre los componentes Modelo y Vista para procesar toda la lógica de negocios y las solicitudes entrantes, manipular los datos utilizando el componente Modelo e interactuar con las Vistas para generar el resultado final. Por ejemplo, el controlador del Cliente maneja todas las interacciones y entradas de la Vista del Cliente y actualizará la base de datos utilizando el Modelo del Cliente. El mismo controlador se utilizará para ver los datos del cliente(MVC (Model, View, Controller) Explicad, 2018).

Figura 1. Modelo Vista Controlador



Fuente: Los autores

2.1.4.1.7. SPA

SPA (Single Page Application), un tipo de aplicación web cada vez más usado por la agradable experiencia de usuario que aporta.

En pocas palabras, SPA son las siglas de Single Page Application. Es un tipo de aplicación web donde todas las pantallas las muestra en la misma página, sin recargar el navegador.

Técnicamente, una SPA es un sitio donde existe un único punto de entrada, generalmente el archivo index.html. En la aplicación no hay ningún otro archivo HTML al que se pueda acceder de manera separada y que nos muestre un contenido o parte de la aplicación, toda la acción se produce dentro del mismo index.html.

2.1.4.1.8. Angular

Es un frameworks para aplicaciones web desarrollado en TypeScript, de código abierto, mantenido por Google, que se utiliza para crear y mantener aplicaciones web de una sola página. Su objetivo es aumentar las aplicaciones basadas en navegador con capacidad de Modelo Vista Controlador (MVC), en un esfuerzo para hacer que el desarrollo y las pruebas sean más fáciles.

La biblioteca lee el HTML que contiene atributos de las etiquetas personalizadas adicionales, entonces obedece a las directivas de los atributos personalizados, y une las piezas de entrada o salida de la página a un modelo representado por las variables estándar de JavaScript. Los valores de las variables de JavaScript se pueden configurar

manualmente, o ser recuperados de recursos JSON estáticos o dinámicos(Sandoval, 2018).

2.1.4.1.9. Node.JS

Node.JS es un entorno de ejecución en JavaScript de código abierto y multiplataforma para ejecutar código JavaScript en el servidor. Históricamente, el JavaScript se utiliza principalmente para scripts de navegador, en los que las secuencias de comandos escritas en JavaScript estaban incorporadas en el HTML de una página web, para ser ejecutadas por un motor de JavaScript en el navegador web del usuario. Node.js permite que JavaScript se use para las secuencias de comandos del lado del servidor y ejecute los scripts del lado del servidor para producir contenido dinámico de la página web antes de que la página se envíe al navegador web del usuario.

Node.js tiene una arquitectura basada en eventos capaz de E/S asíncronas. Estas opciones de diseño apuntan a optimizar el rendimiento y la escalabilidad en aplicaciones Web con muchas operaciones de Input/Output, así como para aplicaciones Web en tiempo real.

El proyecto de desarrollo distribuido Node.js, manejado por la Fundación Node.js, es facilitado por el programa de Proyectos Colaborativos de la Fundación Linux(Abernethy, 2011).

2.1.4.1.10. NPM

Es el gestor de paquetes preinstalado para la plataforma del servidor Node.js. Se utiliza para instalar programas Node.js desde el registro npm, organizando la instalación y administración de programas Node.js de terceros. No se utiliza para cargar código; en su lugar, se utiliza para instalar código y administrar dependencias de código desde la línea de comandos.

Tabla 2. Paquetes de node a usar

Paquete	**Versión**
Express js	>= 4.0
MongoDB Node.JS Driver	>=2
shelljs	0.11
bower	1.8.2

Fuente: Los autores

.

2.2. ANTECEDENTES TEÓRICOS

2.2.1. Legales

En Colombia ya existen empresas que tienen aplicaciones de ese tipo, dentro del marco legal Quick Deliver, tiene la política de protección de datos según la Ley 527 de

1999 art. 2 lit. a la cual regula el intercambio de datos electrónicos, todas las actividades que tiene que ver con Quick Deliver están reguladas con conformidad con la ley 527 de 1999 la cual valida todas la acciones que tiene que ver con comercio electrónico además todo el tema legal de los domiciliarios está regulado de acuerdo al código sustantivo del trabajo(Congreso de Colombia, 1999).

2.3.MARCO CONCEPTUAL

2.3.1. Comercio Electrónico

El comercio electrónico o también conocido como e.commerce consiste en la compra y venta de productos y servicios a través de la red o llamada también internet, dentro de las principales ventajas del e-commerce es que se pueden adquirir productos y servicios a cualquier hora del día y desde cualquier lugar del mundo la mayoría de veces, el primer paso del e-commerce fue dado por la compañía Amazon la cual hoy en día es muy grande financieramente gracias al uso del comercio electrónico. En el e-commerce la relación con el cliente consumidor es muy importante por lo cual la información de una aplicación de e-commerce debe tener un objetivo final, el cual es que el consumidor termine comprando o usando un producto o servicio que se ofrezca de manera que esta aplicación debe ser muy intuitiva, se debe acompañar al consumidor en el proceso de compra que debe ser fácil, y segura.

En el caso de Quick Deliver se tiene una plataforma virtual que usa el modelo de e-commerce para ofrecer su servicio de domiciliario a todas aquellas empresas que

puedan necesitarlo a la hora de entregar sus productos sin preocuparse por la alta demanda o la ubicación geográfica del cliente.

2.3.2. Problema

El problema a solucionar con este prototipo es que la empresa Quick Deliver en este momento no tiene forma de administrar sus domiciliarios y no tiene bastantes clientes como para crecer como es su deseo, la comunicación con sus clientes es ardua, actualmente deben solicitar el servicio de domiciliario con un día de antelación y este domiciliario es contratado por día sin saber si las ventas van a generarse de forma activa o no, simplemente puede ocurrir que el domiciliario no pueda realizar la entrega entre los tiempos establecidos a todos los consumidores o también puede suceder que el domiciliario simplemente no esté siendo utilizado si no para unos pocos pedidos en el día. Otros problemas a los que las empresas que venden comida u otros productos puede ser la credibilidad de sus clientes ya que al demorar tantos sus pedidos la percepción del buen servicio se pierde y con ello también hay pérdidas graduales de clientes y muchas veces se prefiere comprar directamente por el cliente.

3. DESARROLLO DEL PROYECTO

3.1.METODOLOGÍAS DE DESARROLLO

Un Sistema de Versionado de Código (SVC) en abstracto es lo que nos permite compartir el código fuente de nuestros desarrollos y a la vez mantener un registro de los cambios por los que va pasando.

En general va a ser la herramienta más importante y fundamental dentro del desarrollo. Después veremos que también vamos a tener otras herramientas para publicar releases, documentación, para integración y control de calidad, pero todo va a depender de la herramienta SVC.

Si se pierden los releases, más allá de ser un inconveniente, se puede salvar en base al SVC.

En cambio sí se corrompe, borra o "pierde" el SVC estamos sonados.

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

A lo largo del tiempo, una gran cantidad de métodos han sido desarrollados diferenciándose por su fortaleza y debilidad.

El framework para metodología de desarrollo de software consiste en:

Una filosofía de desarrollo de programas de computación con el enfoque del proceso de desarrollo de software.

Herramientas, modelos y métodos para asistir al proceso de desarrollo de software.

Estos frameworks son a menudo vinculados a algún tipo de organización, que además desarrolla, apoya el uso y promueve la metodología.

3.1.1. GIT

Git fue creado pensando en la eficiencia y la confiabilidad del mantenimiento de versiones de aplicaciones cuando éstas tienen un gran número de archivos de código fuente, es decir Git nos proporciona las herramientas para desarrollar un trabajo en equipo de manera inteligente y rápida y por trabajo nos referimos a algún software o página que implique código el cual necesitemos hacerlo con un grupo de personas(Sistemas de control de versiones, s.f.).

Algunas de las características más importantes de Git son:

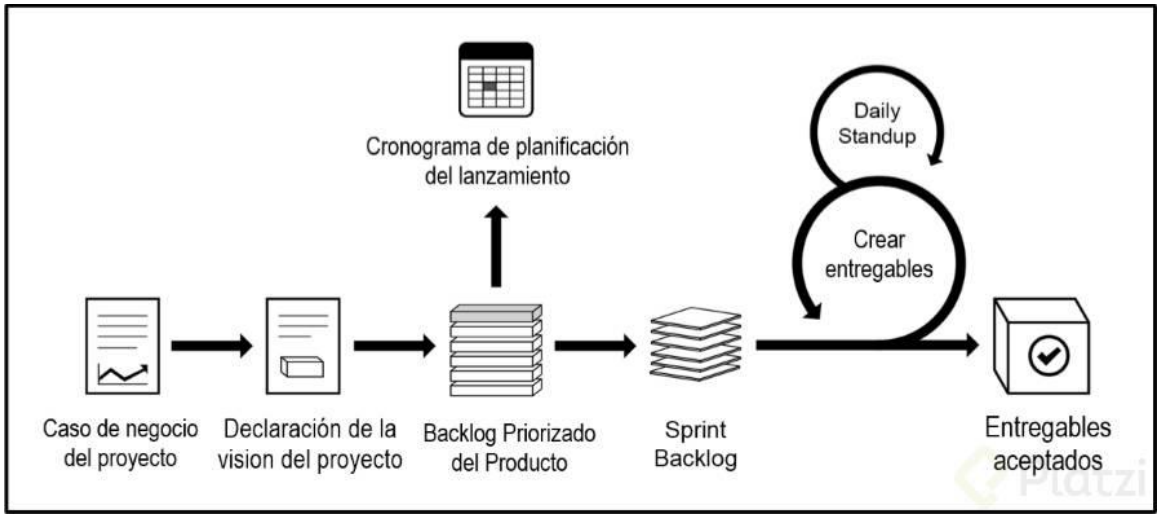
- Rapidez en la gestión de ramas, debido a que Git nos dice que un cambio será fusionado mucho más frecuentemente de lo que se escribe originalmente.
- Gestión distribuida; Los cambios se importan como ramas adicionales y pueden ser fusionados de la misma manera como se hace en la rama local.
- Gestión eficiente de proyectos grandes.
- Re almacenamiento periódico en paquetes.

3.1.2. SCRUM

Scrum es un framework ágil muy completo para el desarrollo de proyectos. En Scrum la palabra producto hace referencia a un producto o servicio o cualquier otro resultado que esté de acuerdo con definición de la visión del proyecto, es decir que puede aplicarse a TODO tipo de proyectos, pero no todos los proyectos requieren el uso de Scrum.

Scrum es un framework adaptable, iterativo, rápido, flexible y eficaz que está diseñado para entregar valor al cliente durante todo el desarrollo del proyecto. El objetivo primordial es satisfacer las necesidades del cliente a través de un entorno de transparencia en la comunicación, responsabilidad colectiva y progreso continuo.

Figura 2. Procesos en Scrum



Fuente: Los autores

4. LEVANTAMIENTO DE REQUERIMIENTOS

Planteado en la descripción del problema, dentro de los principales requerimientos para Quick Deliver el prototipo a diseñar a la compañía Grupo Cosechar, están que sea una aplicación virtual móvil y web para solicitud, administración, seguimiento y despacho de domiciliarios que trabajan para la empresa Grupo Cosechar.

4.1. REQUERIMIENTOS DEL SISTEMA

4.1.1. Requerimientos Funcionales

Tabla 3. Requerimiento Funcional RF01

Identificación del requerimiento:	RF01.
Nombre del Requerimiento:	Solicitud de domiciliario
Características:	El prototipo debe tener un módulo de solicitud de domiciliario en donde se deben registrar la ubicación geográfica y contactos del cliente final.
Descripción del requerimiento:	En este módulo se podrá hacer la solicitud de un domiciliario registrando dirección domiciliaria asociado teléfonos y formas de pago.
Requerimiento NO funcional:	• RF01
Prioridad del requerimiento: Alta	

Fuente: Los autores

Tabla 4. Requerimiento Funcional RF02

Identificación del requerimiento:	RF02.
Nombre del Requerimiento:	Autenticar Usuario por NIT
Características:	Los usuarios que serán las empresas deberán identificarse para poder acceder al sistema por medio del NIT.
Descripción del requerimiento:	El sistema podrá ser consultado por la empresa solicitante, por el administrador y por el domiciliario.
Requerimiento NO funcional:	• RF02
Prioridad del requerimiento: Alta	

Fuente: Los autores

Tabla 5. Requerimiento Funcional RF03

Identificación del requerimiento:	RF03
Nombre del Requerimiento:	Cerrar pedido
Características:	El prototipo debe tener un módulo de seguimiento del pedido.
Descripción del requerimiento:	En este módulo se podrá hacer seguimiento al pedido debe tener un estado, en camino, pendiente asignar domiciliario o cerrar pedido para finalizar el proceso.
Requerimiento NO funcional:	• RF03
Prioridad del requerimiento: Alta	

Fuente: Los autores

Tabla 6. Requerimiento Funcional RF04

Identificación del requerimiento:	RF04
Nombre del Requerimiento:	Consultar Información.
Características:	El sistema debe permitir al administrador al domiciliario y al cliente que solicita el servicio de domicilios consultar el estado de entrega del pedido.
Descripción del requerimiento:	Mostrará información pertinente al servicio solicitado como cerrar pedido, en camino pendiente asignación.
Requerimiento NO funcional:	• RF04
Prioridad del requerimiento: Alta	

Fuente: Los autores

Tabla 7. Requerimiento Funcional RF05

Identificación del requerimiento:	RF05
Nombre del Requerimiento:	Mostrar dashboard
Características:	El cliente debe poder visualizar el estado de sus pendidos por medio de un dash.
Descripción del requerimiento:	El sistema mostrará por medio de un dash la toda la información referente al estado de sus pedidos.
Requerimiento NO funcional:	RF05
Prioridad del requerimiento: Alta	

Fuente: Los autores

Tabla 8. Requerimiento Funcional RF06

Identificación del requerimiento:	RF06.
Nombre del Requerimiento:	Gestionar reportes Administrativos.
Características:	El administrador podrá generar reportes del sistema.
Descripción del requerimiento:	Permitirá al administrador generar diferentes reportes de la aplicación.
Requerimiento NO funcional:	• RF06
Prioridad del requerimiento: Alta	

Fuente: Los autores

4.1.2. Requerimientos no Funcionales

Los requerimientos identificados en esta parte del documento son aplicables al prototipo en general. Para la implementación de la plataforma web es necesario que cuente con unas especificaciones tecnológicas las cuales se mencionan a continuación:

Tabla 9. Requerimiento No Funcional RNF01

Identificación del requerimiento:	RNF01.
Nombre del Requerimiento:	Interfaz del sistema.
Características:	El sistema presentará una interfaz de usuario sencilla para que sea de fácil manejo a los usuarios del sistema.
Descripción del requerimiento:	El sistema debe tener una interfaz de uso intuitiva y sencilla.
Prioridad del requerimiento: Alta	

Fuente: Los autores

Tabla 10. Requerimiento No Funcional RNF02

Identificación del requerimiento:	RNF02.
Nombre del Requerimiento:	Ayuda en el uso del sistema.
Características:	La interfaz del usuario deberá de presentar un sistema de ayuda para que los mismos usuarios del sistema se les faciliten el trabajo en cuanto al manejo del sistema.
Descripción del requerimiento:	La interfaz debe estar complementada con un buen sistema de ayuda (la administración puede recaer en personal con poca experiencia en el uso de aplicaciones informáticas).
Prioridad del requerimiento: Alta	

Fuente: Los autores

Tabla 11. Requerimiento No Funcional RNF03

Identificación del requerimiento:	RNF03.
Nombre del Requerimiento:	Desempeño.
Características:	El sistema garantizará a los usuarios un desempeño en cuanto a los datos almacenado en el sistema ofreciéndole una confiabilidad a esta misma.
Descripción del requerimiento:	Garantizar el desempeño del sistema informático a los diferentes usuarios. En este sentido la información almacenada o registros realizados podrán ser consultados y actualizados permanente y simultáneamente, sin que se afecte el tiempo de respuesta.
Prioridad del requerimiento: Alta	

Fuente: Los autores

Tabla 12. Requerimiento No Funcional RNF04

Identificación del requerimiento:	RNF04.
Nombre del Requerimiento:	Confiabilidad continúa del sistema.
Características:	El sistema tendrá que estar en funcionamiento las 24 horas los 7 días de la semana.
Descripción del requerimiento:	La disponibilidad del sistema debe ser continua con un nivel de servicio para los usuarios de 7 días por 24 horas.
Prioridad del requerimiento: Alta	

Fuente: Los autores

Tabla 13. Requerimiento No Funcional RNF05

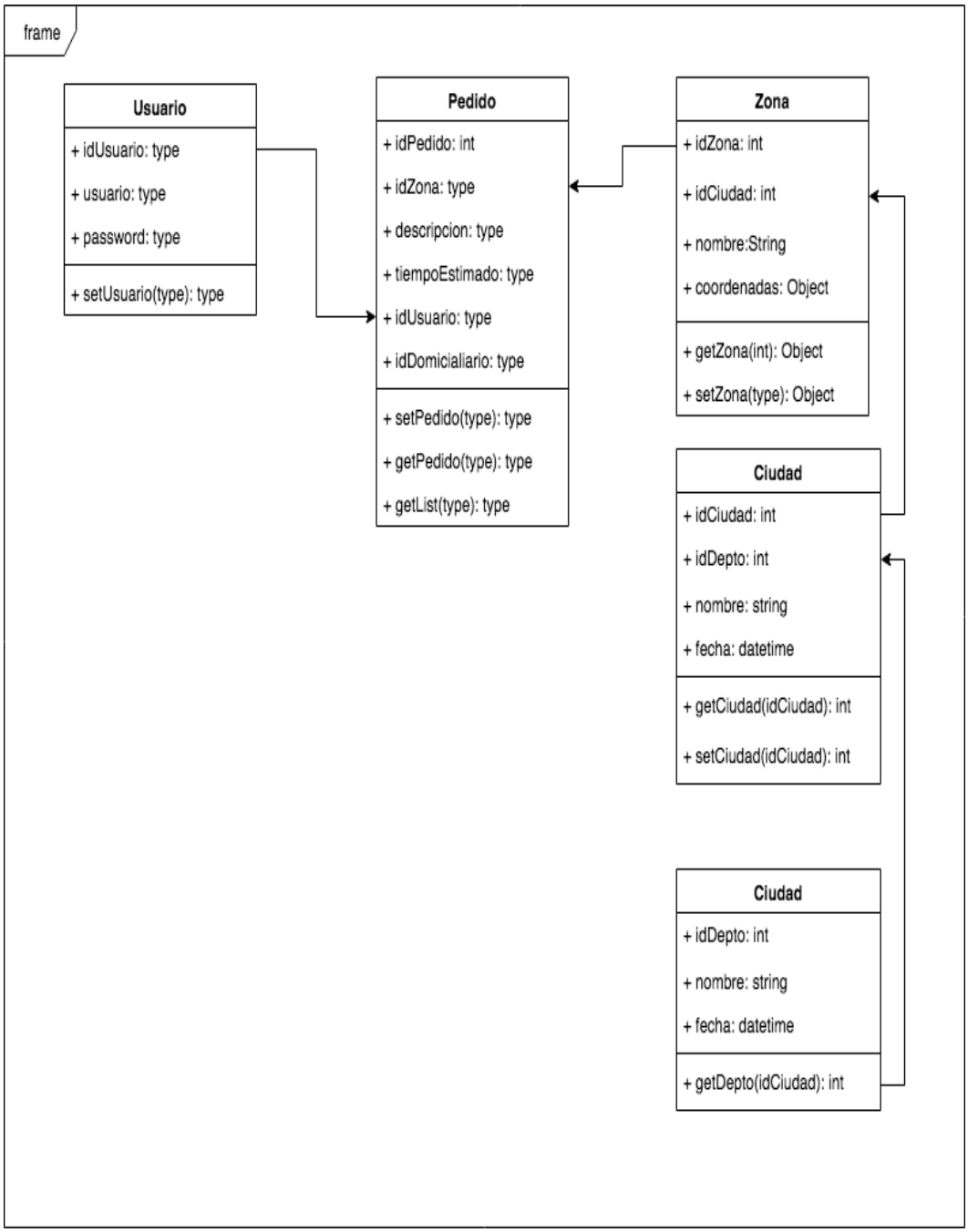
Identificación del requerimiento:	RNF05.
Nombre del Requerimiento:	Seguridad en información.
Características:	El sistema garantizará a los usuarios una seguridad en cuanto a la información que se procede en el sistema.
Descripción del requerimiento:	Garantizar la seguridad del sistema con respecto a la información y datos que se manejan tales sean documentos, archivos y contraseñas.
Prioridad del requerimiento: Alta	

Fuente: Los autores

4.2.DISEÑO DE APLICACIÓN

4.2.1. Diagrama De Clases

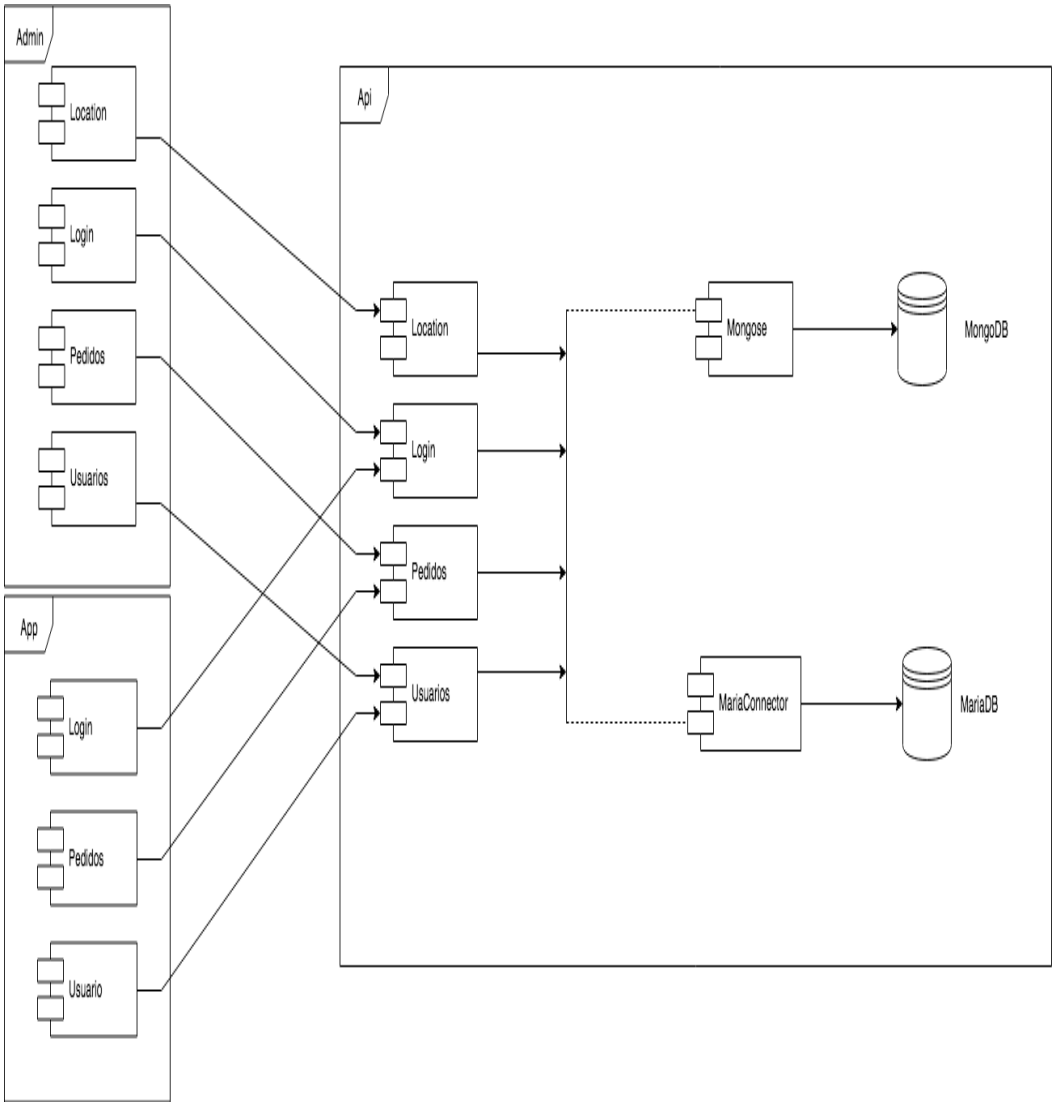
Figura 3. Diagrama de clases



Fuente: Los autores

4.2.2. Diagrama De Componentes

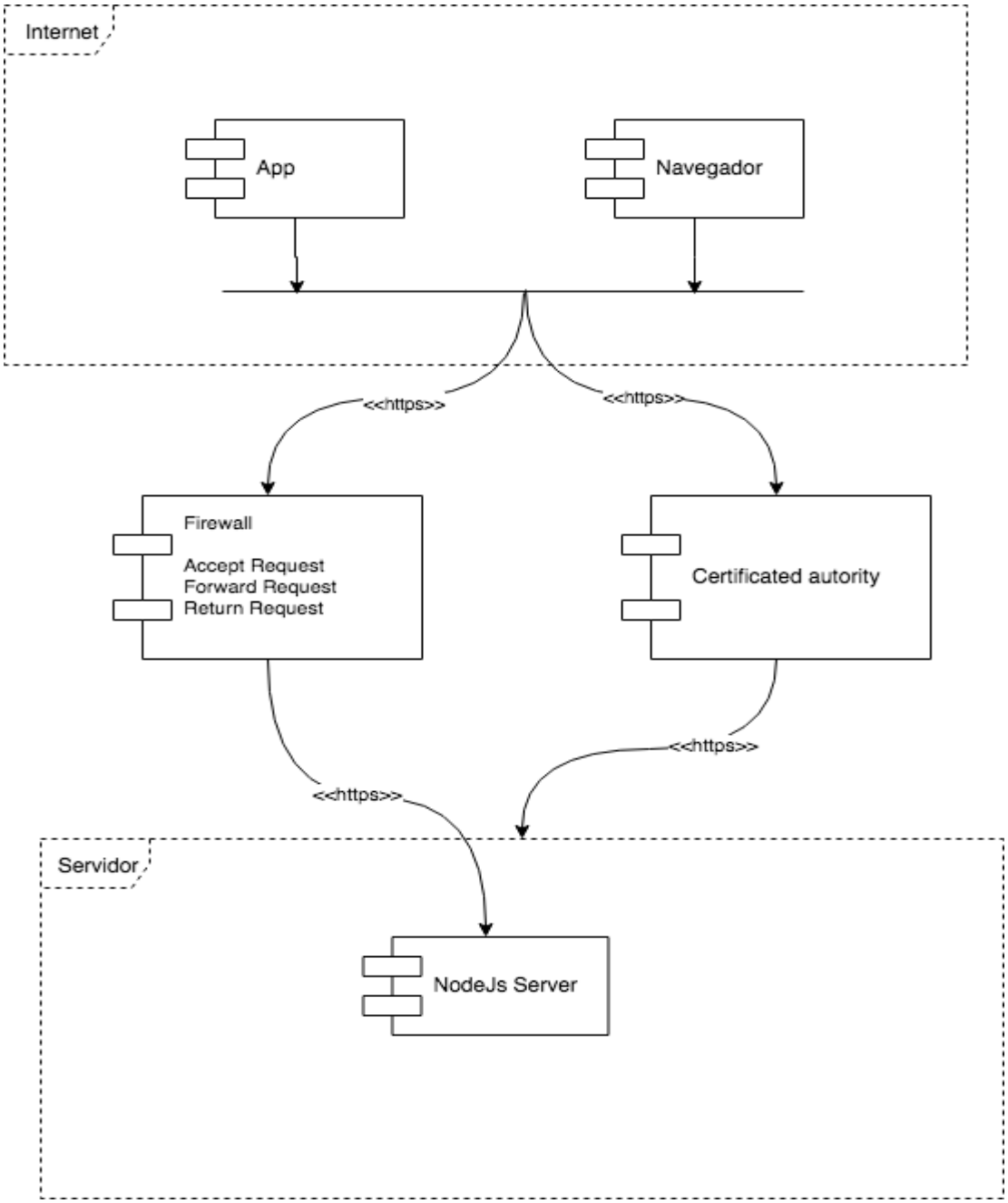
Figura 4. Diagrama de componentes



Fuente: Los autores

4.2.3. Diagrama De Implementación

Figura 5. Diagrama de implementación

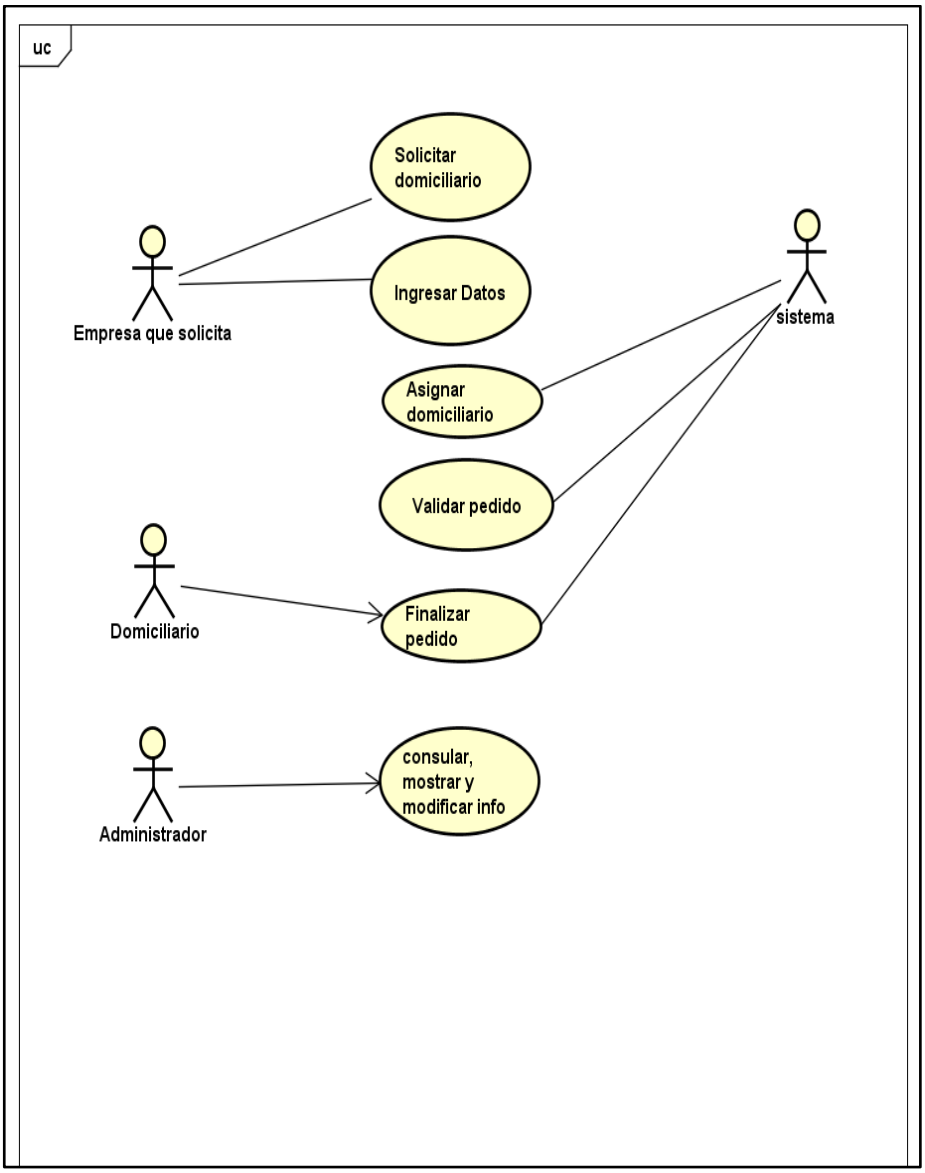


Fuente: Los autores

4.2.4. Diagrama Uml Comportamiento

4.2.5. Diagrama Casos De Uso

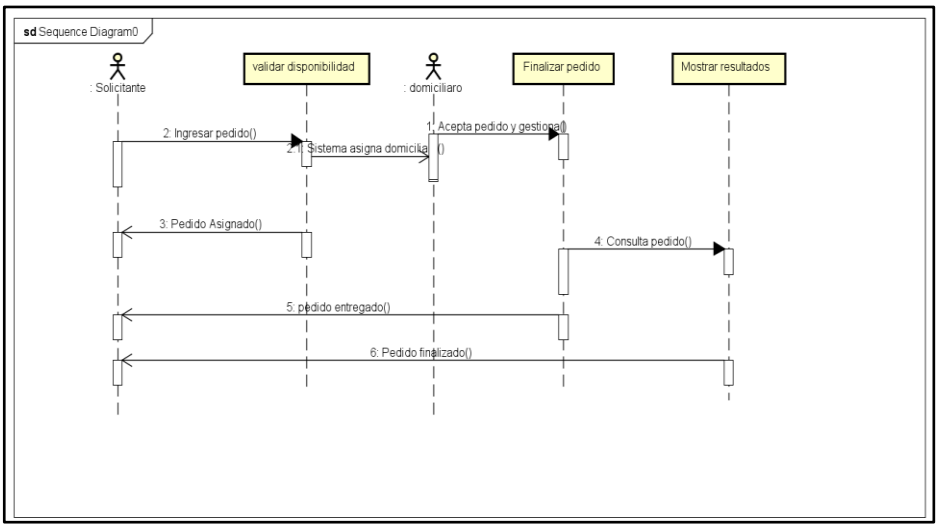
Figura 6. Diagrama de casos de uso



Fuente: los autores

4.2.6. Diagrama De Secuencia

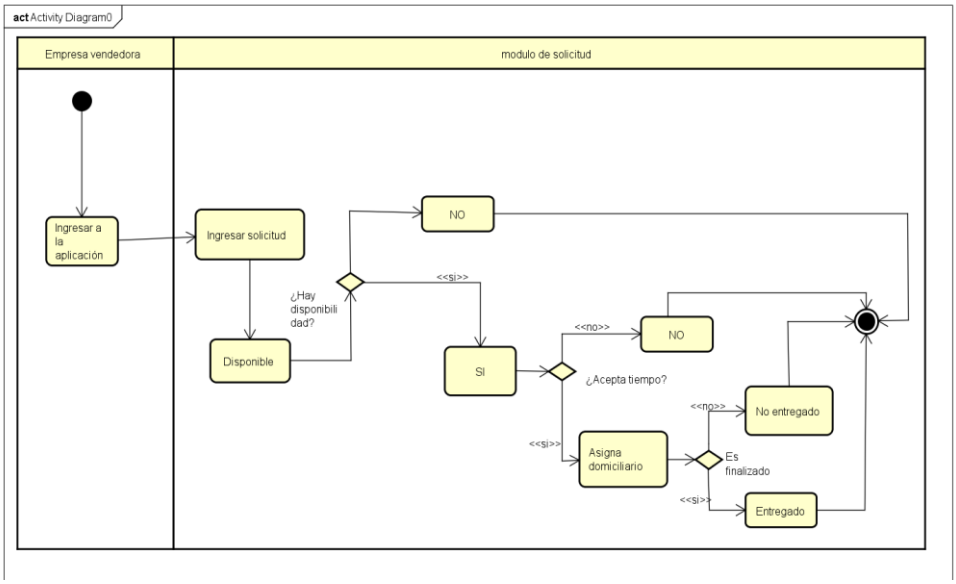
Figura 7. Diagrama de secuencia



Fuente: los autores

4.2.7. Diagrama De Actividades

Figura 8. Diagrama de actividades



Fuente: los autores

4.3.IMPLEMENTACIÓN

De ser necesaria una implementación del prototipo es necesario disponer de equipos de cómputos en perfecto estado con las siguientes características:

- Adaptadores de red.
- Procesador de 1.66GHz o superior.
- Memoria mínima de 256Mb

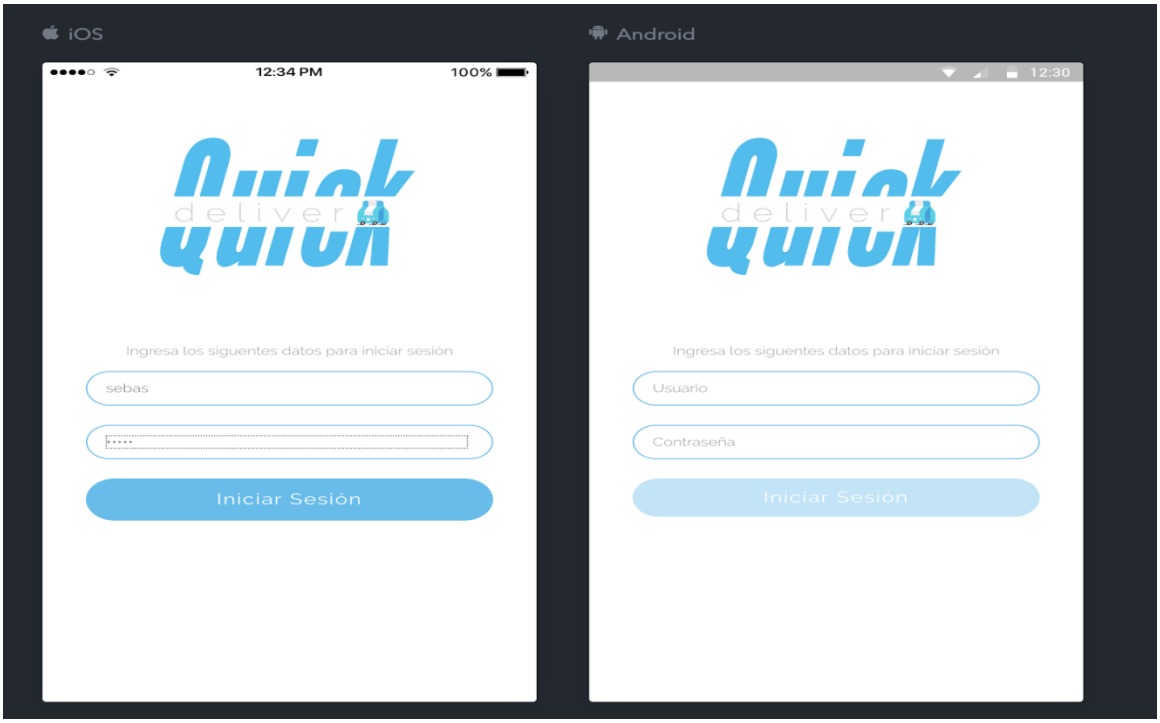
Para la implementación del prototipo móvil es necesario tener un Smartphone (Celular) con la aplicación instalada sistema Android o IOS y conexión a internet.

5. MANUALES

5.1.MANUAL DE USUARIO

- Ingreso usuario

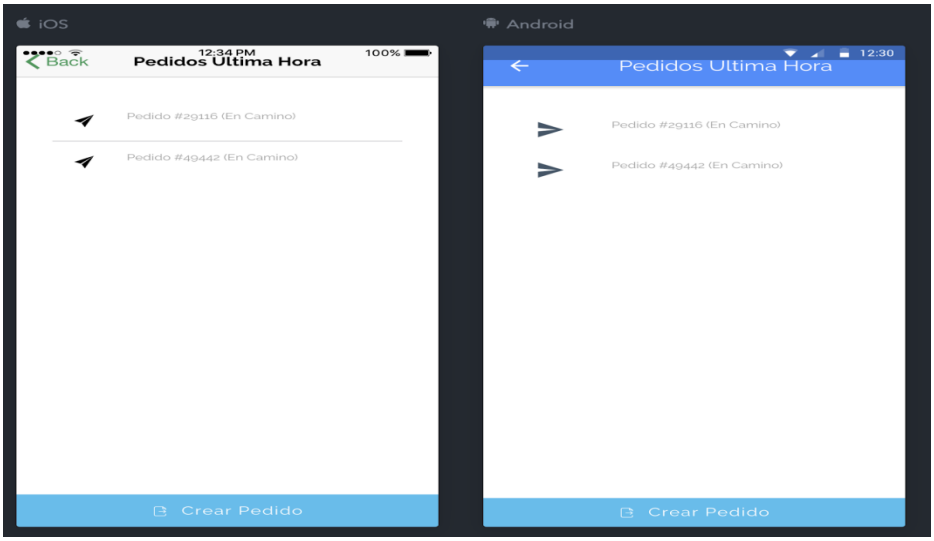
Imagen 1. Ingreso a la aplicación



Fuente: los autores

- Listado de pedidos

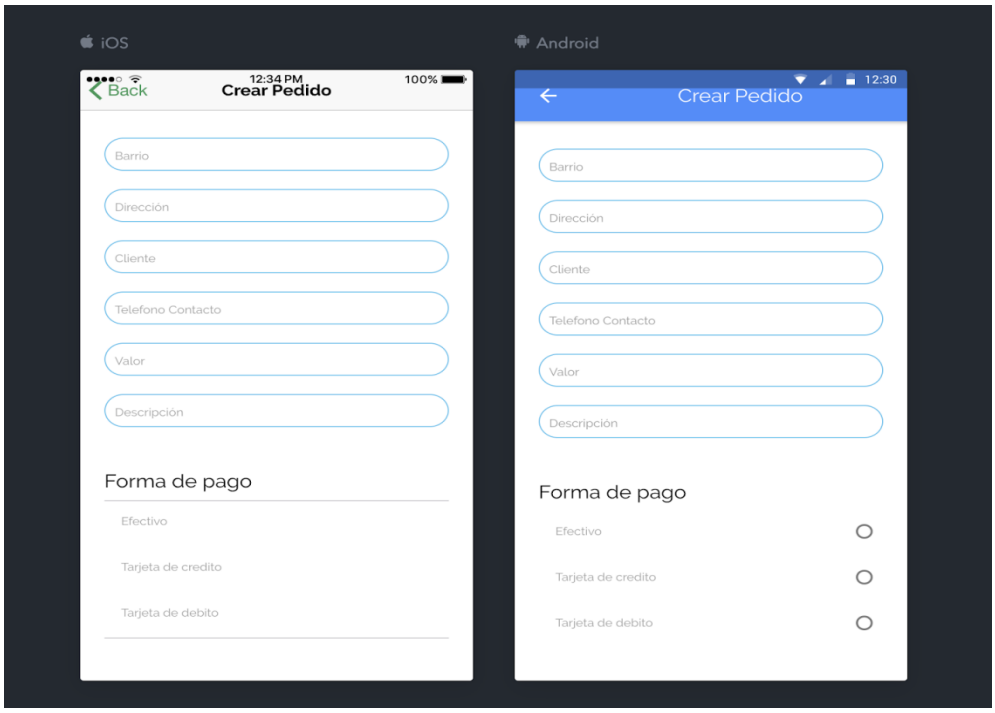
Imagen 2. Listado de pedidos



Fuente: Los autores

- Creación del pedido

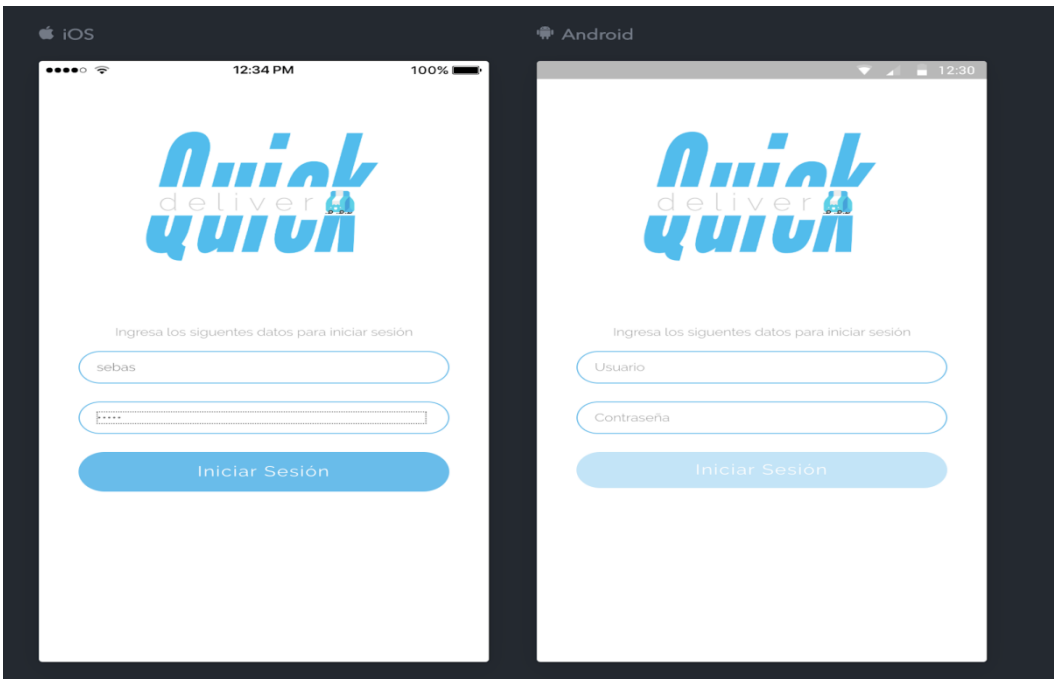
Imagen 3. Creación de los pedidos



Fuente: los autores

- Ingreso domiciliario

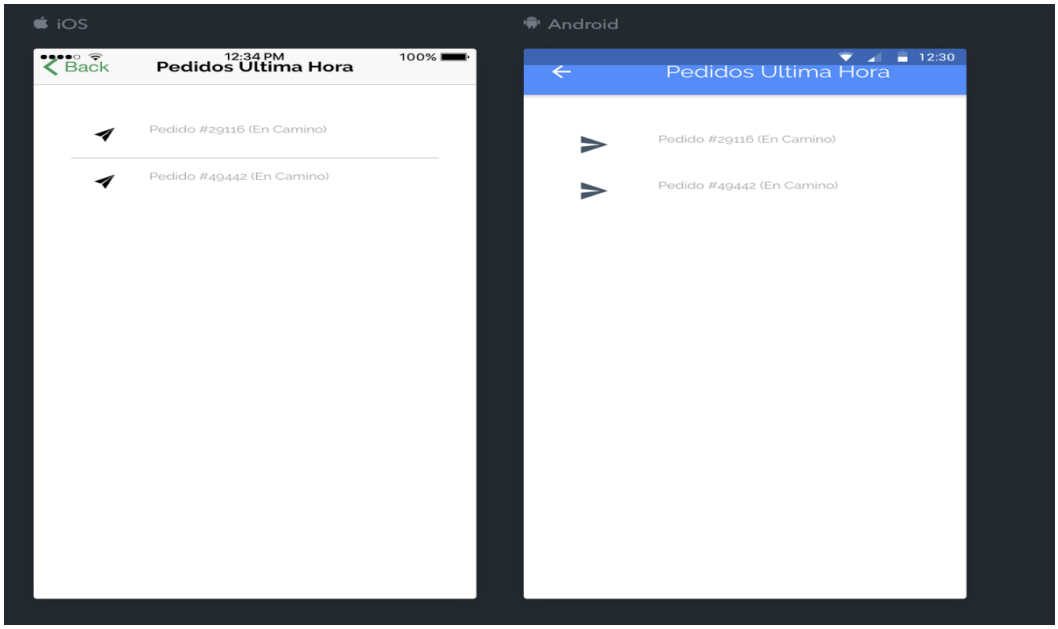
Imagen 4. Ingreso del domiciliario



Fuente: los autores

- Listado de pedidos asignados

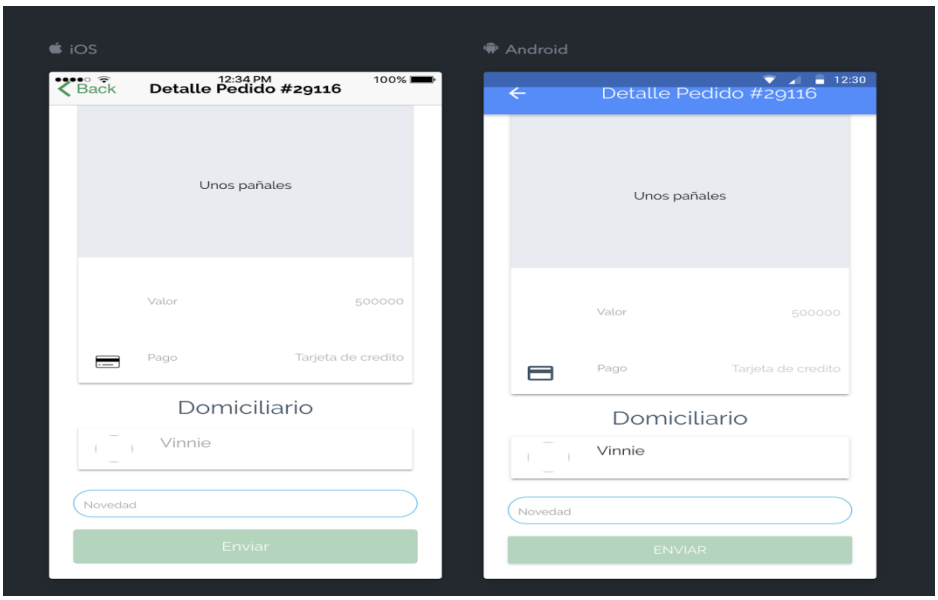
Imagen 5. Listado de pedidos asignados



Fuente: los autores

- Notificación de novedades

Imagen 6. Notificación de novedades

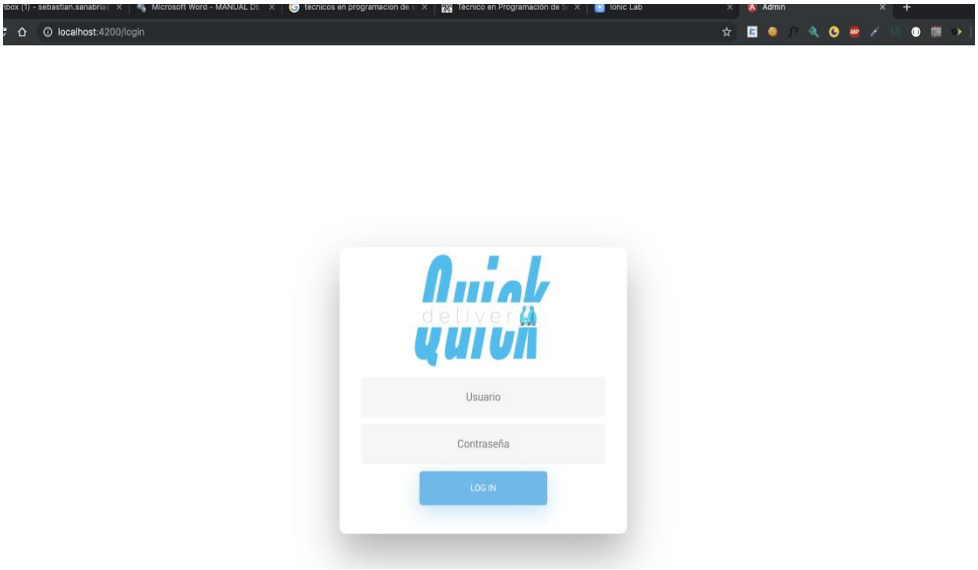


Fuente: los autores

5.2.MANUAL USUARIO ADMIN

- Ingreso administrador

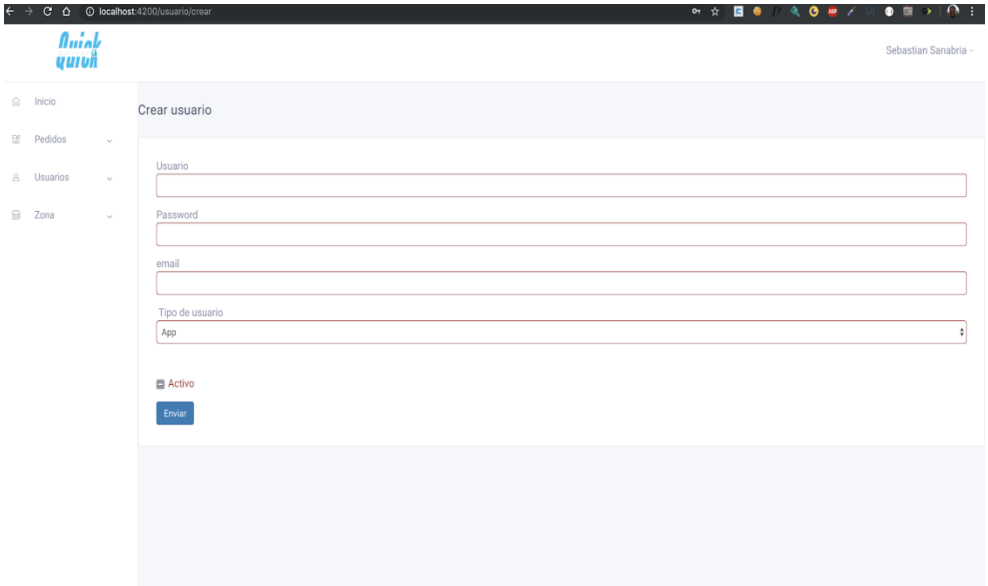
Imagen 7. Ingreso administrador de la cuenta



Fuente: los autores

- Creación de usuarios

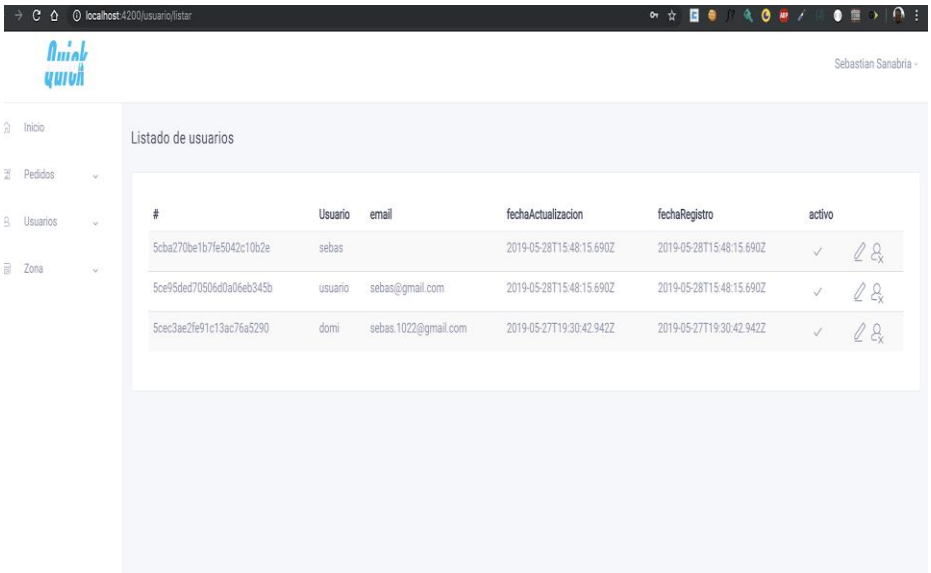
Imagen 8. Creación de usuarios



Fuente: los autores

- Listado de usuarios agregados al sistema

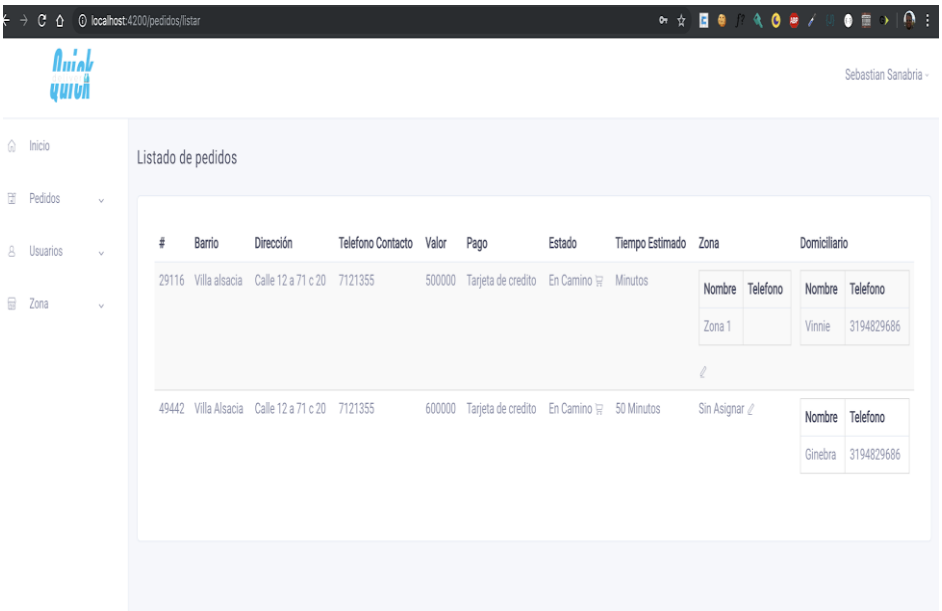
Imagen 9. Listado de usuarios agregados



Fuente: los autores

- Listado de pedidos solicitados

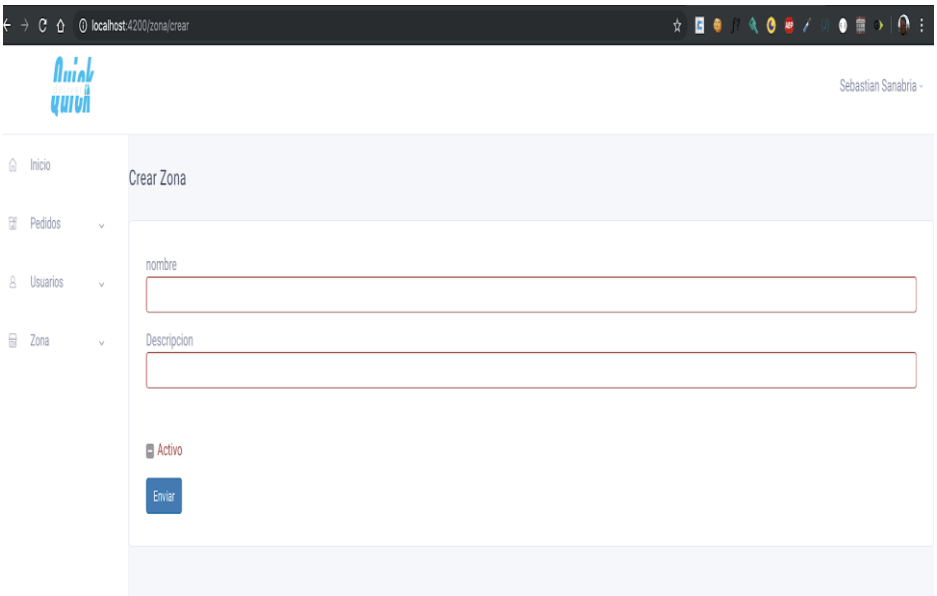
Imagen 10. Listado de pedidos solicitados



Fuente: los autores

- Creación de las zonas

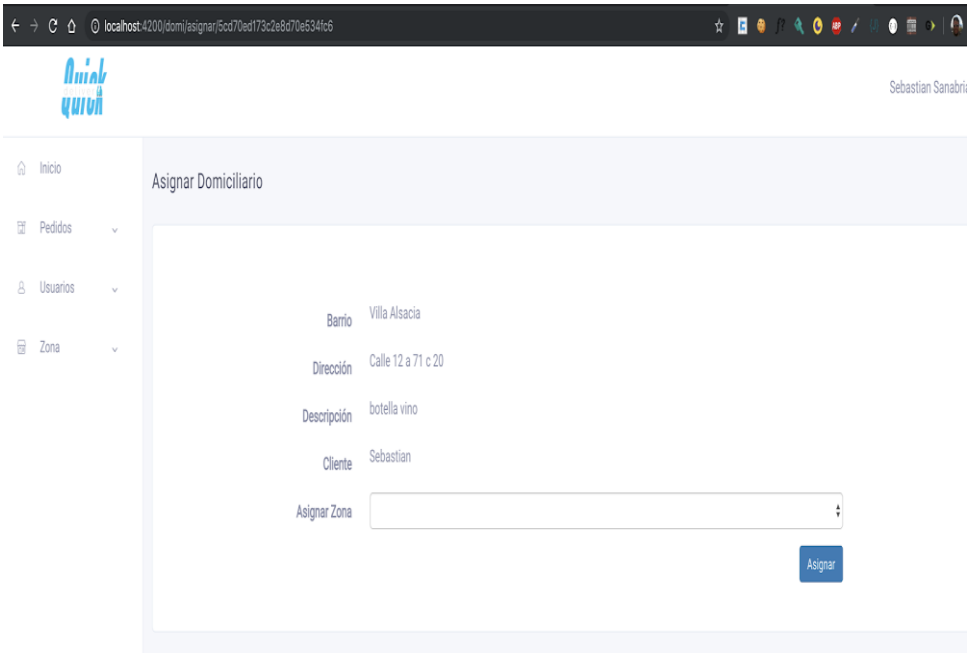
Imagen 11. Creación de las zonas



Fuente: los autores

- Asignación de la zona

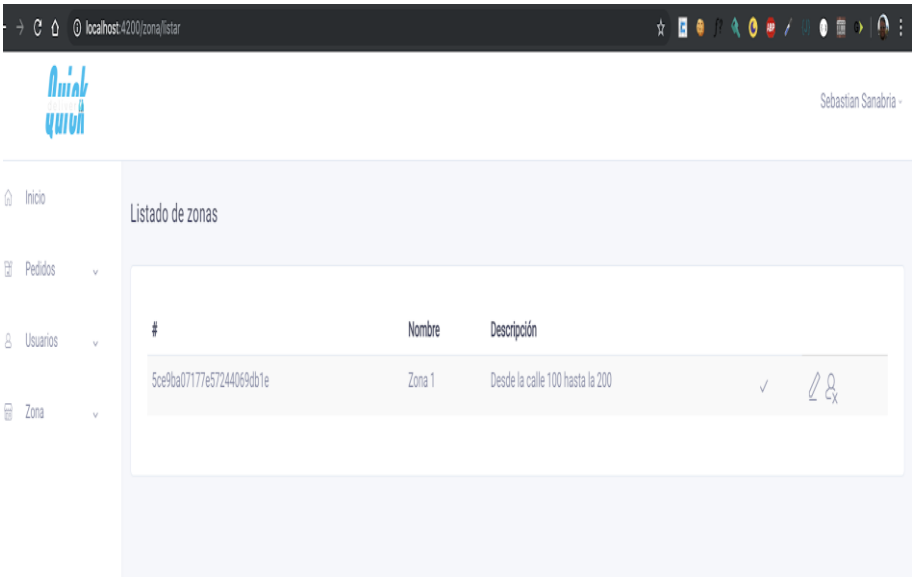
Imagen 12. Asignación de zonas



Fuente: los autores

- Listado de las zonas

Imagen 13. Listado de zonas



Fuente: los autores

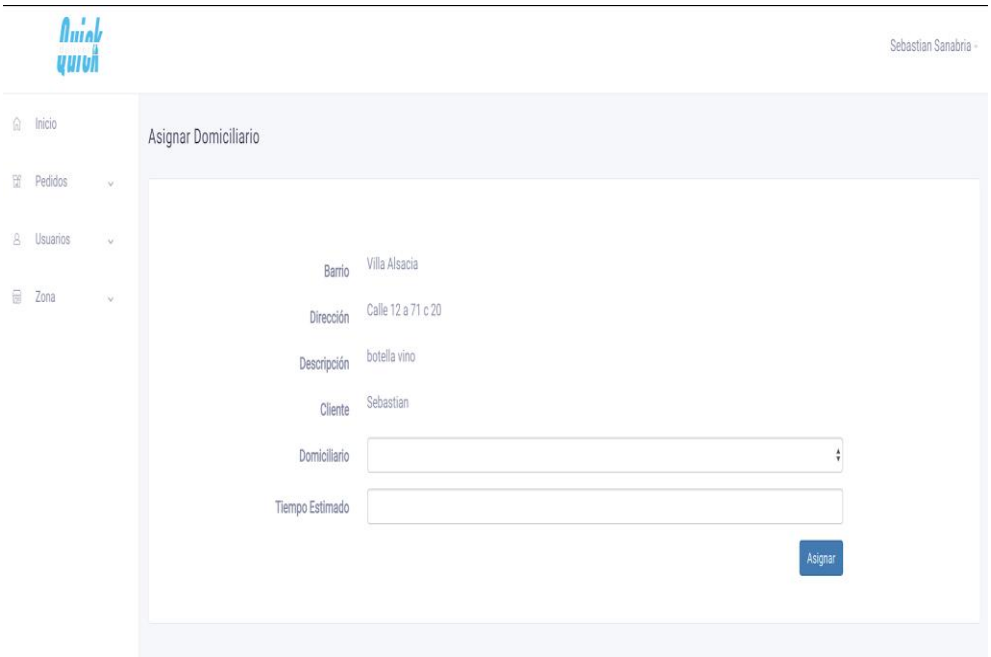
- Ingreso a la zona creada

Imagen 14. Ingreso a la zona

Fuente: los autores

- Asignación de pedido por zona al domiciliario

Imagen 15. Asignación de la zona



Fuente: los autores

5.3.MANUAL TÉCNICO

El control de versiones del prototipo se realizó con Git por lo cual es necesario tener instalado Git en la máquina que se va ejecutar.

Adicional la elaboración está realizada con Node JS por lo cual se debe hacer la descarga de este en su sitio web, https://nodejs.org/en/download/ , Una vez descargado el archivo procedemos con la instalación.

Primero vamos a verificar que Node.js se ha instalado correctamente para esto abrimos una consola cmd y ejecutamos el siguiente comando:

```
> node -v
```

Este comando retornará la versión actual de Node.js.

Lo siguiente que vamos hacer es verificar la versión del gestor de paquetes NPM instalada, para esto usando la consola abierta ejecutamos el comando:

```
> npm -v
```

Este comando retornará la versión actual del gestor de paquetes.

Seguido a esto procedemos a instalar el framework de Ionic para el prototipo, para esto usando la consola abierta ejecutamos el comando:

```
> npm install -g Ionic
```

Luego de esto vamos a descargar e instalar las dependencias de las 3 partes del proyecto

Para el programa prototipo

```
> git clone git@github.com:sebas1022/quick-deliver-app.git
```

Ingresamos a la carpeta

```
> cd quick-deliver-app/
```

Instalamos las dependencias

```
> npm install
```

Corremos el servidor en ambiente local

```
> npm run lab
```

Para web api

```
> git clone git@github.com:sebas1022/quick-deliver-api.git
```

Ingresamos a la carpeta

```
> cd quick-deliver-api/
```

Instalamos las dependencias

```
> npm install
```

Corremos el servidor en ambiente local

```
> npm run start
```

Para web api

```
> git clone git@github.com:sebas1022/quick-deliver-admin.git
```

Ingresamos a la carpeta

```
> cd quick-deliver-admin/
```

Instalamos las dependencias

```
> npm install
```

Corremos el servidor en ambiente local

```
> ng serve
```

5.4.MANUAL MONTAJE SOFTWARE

Para el despliegue de estas 3 instancias los pasos son los siguientes

Para el prototipo

Primero se debe generar una versión de lanzamiento de nuestro prototipo, dirigida a cada plataforma en la que se desee elaborar. Antes de dar inicio al diseño, debemos

tener cuidado de ajustar los complementos necesarios durante la elaboración que no deberían estar en modo de producción. Por ejemplo, probablemente no queremos que el complemento de la consola de depuración esté habilitado, por lo que deberíamos eliminarlo antes de generar las compilaciones de lanzamiento:

```
>Ionic cordova plugin rm cordova-plugin-console
```

Androide Publishing

Para generar una versión de lanzamiento para el sistema Androide, podemos usar el siguiente comando de cordova cli:

```
>ionic cordova build --release android
```

Esto generará una versión de lanzamiento basada en la configuración de su config.xml. Ionic tendrá valores predeterminados en este archivo, pero si necesita personalizar cómo se crea, se puede editar este archivo para que se ajuste a las preferencias.

A continuación, se puede encontrar el archivo APK sin firmar en platforms/android/build/outputs/apk. En el ejemplo, el archivo será platforms/android/build/outputs/apk/Hello.apk. Ahora, se debe firmar el APK y ejecutar una utilidad de alineación para optimizarlo y prepararlo para la tienda de aplicaciones. Si ya se tiene una clave de firma, omitir estos pasos y usar esa en su lugar.

Generando la clave privada se debe usar el comando keytool que viene con el JDK. Si no encuentra esta herramienta, consulte la guía de instalación:

```
>keytool -genkey-v-keystore my-release-key.keystore -alias alias_name -keyalg RSA -keysize 2048 -validity 10000
```

Primero solicitará que se cree una contraseña para el almacén de claves. Luego, responder al resto de las preguntas de las herramientas útiles y cuando se haya terminado, debe tener un archivo llamado my-release-key.keystore creado en el directorio actual.

Para firmar el APK sin firmar, se debe ejecutar la herramienta jarsigner que también se incluye en el JDK:

```
>jarsigner -verbose-sigalg SHA1withRSA -digestalg SHA1 -keystore my-release-key.keystore HelloWorld-release-unsigned.apk alias_name
```

Esto firma el apk en su lugar. Finalmente, ejecutar la herramienta zip align para optimizar el APK. La herramienta zipalign se puede encontrar en /path/to/Androide/sdk/build-tools/VERSION/zipalign.

```
>zipalign -v 4 HelloWorld-release-unsigned.apk HelloWorld.apk
```

¡Ahora se tendrá el binario de lanzamiento final llamado HelloWorld.apk y se lanzará en la plataforma de Google Play Store.

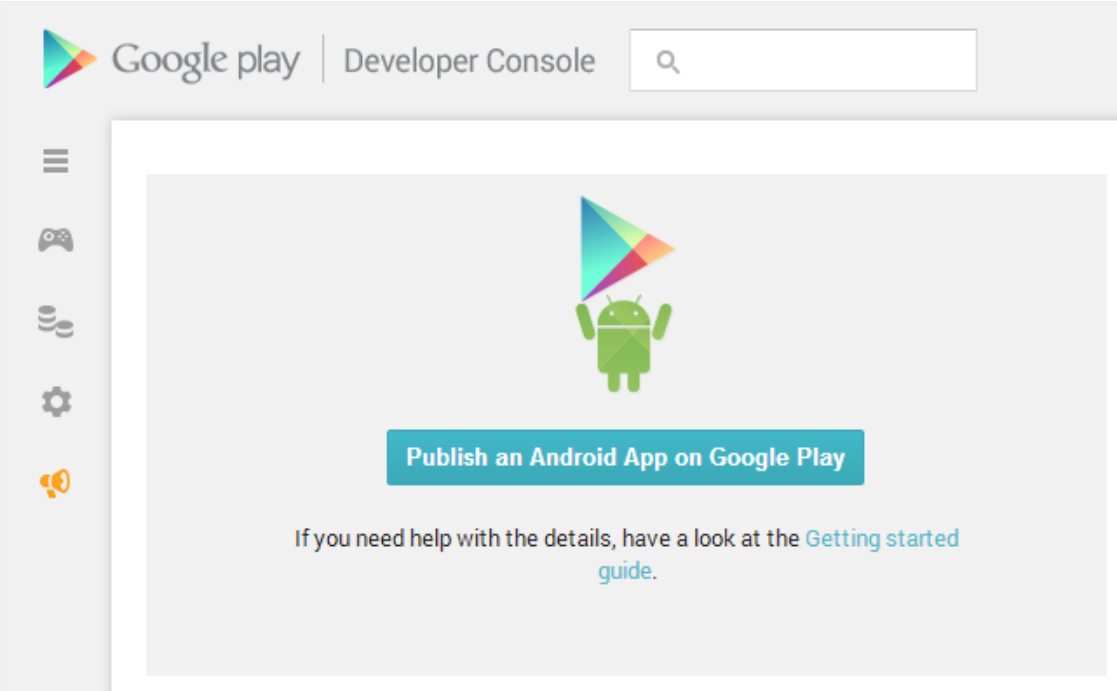
Google Play Store

En el momento que se tenga APK de lanzamiento listo para Google Play Store, se creará una lista de Play Store para cargar el APK.

Para empezar, visitar la consola de desarrollador de Google Play Store y crear una nueva cuenta de desarrollador. Desafortunadamente, esto no es gratis. Más sin embargo, el costo a la fecha mayo 2019 en Android $ 25 comparado con los $ 99 de Apple.

Una vez que tenga una cuenta de desarrollador, puede seguir adelante y hacer clic en "Publicar una aplicación de Androide en Google Play" como se muestra en la siguiente captura de pantalla:

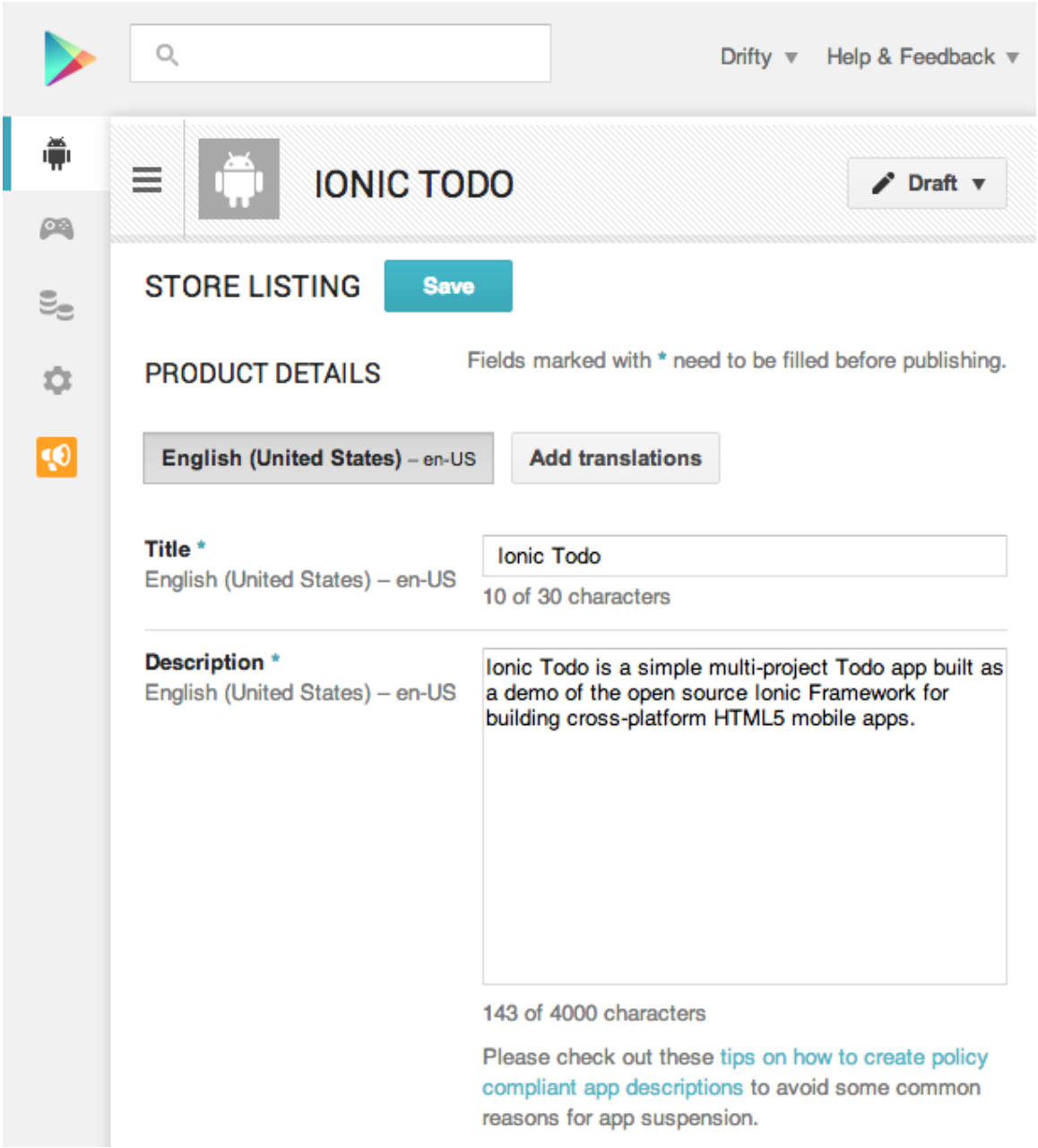
Imagen 16. Pantalla ingreso a Google Play



Fuente: los autores

Luego, se puede continuar y hacer clic en el botón para editar el listado de la tienda (Cargaremos una APK más adelante). Para completar la descripción, aquí hay una pequeña vista previa de cuando completamos el prototipo con la aplicación Ionic:

Imagen 17. Configuración Google Play



Fuente: los autores

Cuando esté listo, cargar el APK para la versión de lanzamiento y publicar la lista. ¡Sea paciente y su trabajo duro debe ser vivir en la naturaleza!

Actualizando el prototipo

A medida que elabore una aplicación se debe tener en cuenta que se actualizará periódicamente.

Para que Google Play Store acepte los APK actualizados, deberá editar el archivo config.xml para incrementar el valor de la versión y luego reconstruir la aplicación para su lanzamiento.

Para el administrador

Para hacer la publicación en un servidor se ejecuta el siguiente comando

```
> ng build
```

Este comando genera en la carpeta del proyecto una carpeta llamada **dist,** la cual es la que se publica en el servidor

6. CONCLUSIONES

- Realizando el proyecto se observa la gran demanda que se tiene y la facilidad en la aceptación de medios tecnológicos.
- La utilización de recurso humano para el domicilio es una gran oportunidad de empleo.
- El crecimiento en alta demanda de servicios solicitudes puede ser de forma progresiva exponencial de forma positiva para la ampliación de infraestructura y personal de la empresa.

LOS LIBERTADORES
FUNDACIÓN UNIVERSITARIA

REFERENCIAS BIBLIOGRÁFICAS

¿Qué es CSS? (2019). Obtenido de Uniwebsidad: https://uniwebsidad.com/libros/css/capitulo-1

Abernethy, M. (14 de Junio de 2011). *¿Simplemente qué es Node.js?* Obtenido de IBM.com: https://www.ibm.com/developerworks/ssa/opensource/library/os-nodejs/index.html

Biología e ideología 1. (s.f.). Obtenido de https://www.ecasals.net/mobile/

Congreso de Colombia. (21 de Agosto de 1999). *Ley 527 de 1999.* Obtenido de http://www.secretariasenado.gov.co/senado/basedoc/ley_0527_1999.html

IDE de programación. (2012). Obtenido de EcuRed: https://www.ecured.cu/IDE_de_Programaci%C3%B3n

MVC (Model, View, Controller) Explicad. (2018). Obtenido de Codigofacilito: https://codigofacilito.com/articulos/mvc-model-view-controller-explicado

Qué es el lenguaje unificado de modelado (UML). (2019). Obtenido de Lucidchart: https://www.lucidchart.com/pages/es/que-es-el-lenguaje-unificado-de-modelado-uml

Sandoval, D. (2018). *Top 5 Frameworks y librerías JavaScript con Pedro Arce.* Obtenido de Next U: https://www.nextu.com/blog/mejores-librerias-frameworks-javascript/

Sistemas de control de versiones. (s.f.). Obtenido de Codigofacilito: https://codigofacilito.com/articulos/que-es-git

ANEXOS

Anexo A. ACTA INICIO DE REQUERIMIENTOS

Bogotá D.C. 08 de Marzo de 2019

Hernán Parraga

Grupo Cosechar

3156402549

REF: Acta de requerimientos

Por medio de la presente se realiza la documentación y consigna de la reunión que se sostuvo el viernes 8 de marzo, en el que se tomaron los requerimientos para el desarrollo de la aplicación solicitada por el Grupo Cosechar de la cual se manejara por 7 zonas (Salitre, Colina, Cedritos, Usaquén, Chico, Macarena y Venecia). La aplicación APP requerida por el grupo cosechar en la que se deberá manejar con los siguientes requerimientos principales:

- Domicilio de comida (Se recibirán las solicitudes por parte de las empresas afiliadas para la entrega de los pedidos de insumos solicitados).
- Domicilio de licores (Negocio propio por la empresa solicitante).
- Mensajería en entrega de paquetes y diligencias.

Durante la reunión se solicita información con relación a precios e imágenes que corresponden a la parte de domicilio de licores para el desarrollo y documentación

de la aplicación, de la misma forma se solicita información sobre el área de la zona que se está cubriendo con datos de dirección y alcance de cada una de las zonas que se manejan.

En el requerimiento de domicilio de comida se generará una página web para la recepción de la información solicitada por la empresa que requiere el servicio y que desde la empresa Grupo Cosechar se tenga el control de los domicilios solicitados por comida.

Esta aplicación se entregará como prueba piloto con los requerimientos solicitados y la funcionalidad requerida por parte del Grupo Cosechar, la cual estará dividida en 3 módulos los cuales se explican a continuación:

1. Módulo de Domicilios (B2B): por medio de este módulo las empresas asociadas podrán pedir al Grupo Cosechar un domiciliario para el cual se dirigirá a la sede indicada, recogerá el domicilio y lo llevara a su destino, este módulo está dividido en 2 partes
    1.1. Aplicación Móvil: La cual tendrán instalada las empresas asociadas y por la cual solicitaron el domiciliario una vez asignado por la central el domiciliario que va a ir , se mostrará la placa, el nombre y el tiempo aproximado de llegada.
    1.2. Administrador Web: Estará dividida en 4

1.2.1. Gestión de domiciliarios. Creación , actualización de domicilios y vehículos

1.2.2. Gestión de domicilios: asignación de domicilio a cada domiciliario.

1.2.3. Informes:

1.2.3.1. Domicilios por domiciliario

1.2.3.2. Domicilios por empresa

1.2.4. Usuarios: En este módulo se crearán los usuarios por cada empresa para el acceso a la aplicación.

Módulo de Licores (P2B): en este módulo un usuario se puede registrar en la aplicación y solicitar cualquier licor que esté disponible en el inventario de la aplicación, deberá ser acompañado por un módulo de manejo de inventario, de un módulo de pagos conectado con pasarela de pagos y módulo para domiciliarios en la aplicación para que el usuario sepa en donde viene su domicilio. Adicionalmente se recomienda los siguientes ítems.

- Pauta
- Desarrollo de marca.
- Actualización de módulo de informes.

- Migración a base de datos a no relacional en la nube la cual pueda manejar las transacciones en tiempo real, así como el tracking de los domiciliarios, por medio de una modelo MQTT

3. Módulos de Mensajería (P2B): en este módulo un usuario se puede registrar en la aplicación y solicitar un mensajero para hacer sus vueltas y/o favores. Que él no pueda hacer por estar ocupado.

Anotación: Como parte del proyecto de grado se entregará la prueba piloto funcional módulo de domicilios, sin el diseño de interfaces este solo será propuesto como parte del documento final.

Licencias y Hosting

Las licencias deben ser asumidas por el grupo cosechar y serán propiedad de este mismo. Para el módulo 1 las licencias que se requieren son las siguientes

Licencia	Costo
Desarrollador Android	25 USD
Desarrollador iOS	100 USD

LOS LIBERTADORES
FUNDACIÓN UNIVERSITARIA

Así mismo se requiere un hosting dedicado para la aplicación con un valor aproximado de $300.000 pesos mensuales el cual se entrega con usuario y contraseña de administración.

Nota: Las licencias requieren de un pago único anual de las cuales debe hacerse cargo el Grupo Cosechar

Se envía acta para su firma y aprobación para el inicio y desarrollo de la aplicación requerida la cual se realizará por estudiantes de la Fundación Universitaria los Libertadores, Sebastián Camilo Sanabria Bocanegra y Hugo Ferney Celis Sotelo como opción de grado autorizado por el Ing. Javier Daza Piragauta. Coordinador de Prácticas y Pasantías.

HERNAN PARRAGA HERRERA
PRESIDENTE
GRUPO EMPRESARIAL "COSECHAR"

NOTA: PENDIENTE ELABORAR LA BASE DE DATOS DE LA APP