SISTEMA DE INFORMACIÓN MULTIPLATAFORMA PARA OPTIMIZAR LOS PROCESOS DE RECEPCIÓN Y COMERCIALIZACIÓN DE AGUACATE HASS EN LA ASOCIACIÓN AGROBILBAO.

Trabajo de grado para optar el título de Ingeniero de Sistemas

JOHN EDUARD BOLAÑOS CAMACHO
NILSON GIOVANNI AMAYA SANTANA

UNIVERSIDAD DE CUNDINAMARCA EXTENSIÓN CHIA
PROGRAMA DE INGENIERÍA DE SISTEMAS
FACULTAD DE INGENIERÍA
2019

SISTEMA DE INFORMACIÓN MULTIPLATAFORMA PARA OPTIMIZAR LOS PROCESOS DE RECEPCIÓN Y COMERCIALIZACIÓN DE AGUACATE HASS EN LA ASOCIACIÓN AGROBILBAO.

JOHN EDUARD BOLAÑOS CAMACHO
561213115
NILSON GIOVANNI AMAYA SANTANA
561214205

DIRECTOR
PhD. ARLES PRIETO MORENO

UNIVERSIDAD DE CUNDINAMARCA EXTENSIÓN CHIA
PROGRAMA DE INGENIERÍA DE SISTEMAS
FACULTAD DE INGENIERÍA
2019

AGRADECIMIENTOS.

En primer lugar, queremos agradecer a cada una de nuestras familias, ya que ellos nos han brindado todo su apoyo y comprensión a lo largo de la realización y culminación de nuestra carrera.

También agradecemos a la Universidad de Cundinamarca por abrirnos sus puertas para así poder construir conocimiento y formación integral de la mano de nuestros excelentes maestros y compañeros Udecinos.

Agradecemos el apoyo por parte de nuestro director de proyecto de grado, el Ing. Arles Prieto Moreno, quien estuvo siempre dispuesto a colaborarnos durante en la realización del proyecto.

Agradecemos a la asociación AGROBILBAO por permitirnos realizar este proyecto y poder contribuir al desarrollo agrícola de nuestro país.

RESUMEN.

En los últimos años la producción y exportación del aguacate Hass ha ido en aumento en el país, por lo cual su constante crecimiento demanda soluciones que optimicen los procesos de recepción y comercialización de dicho producto. Por lo anterior, se presentó un sistema de información multiplataforma para la asociación tolimense AGROBILBAO el cual, luego de su debida recolección de información e identificación de las principales necesidades, se desarrolló una aplicación web y móvil las cuales permitieron facilitar los procesos administrativos, incrementar la comunicación y contar con un adecuado manejo de procesos entre los integrantes de la asociación. Esto fue posible por medio del uso de una metodología de desarrollo de software del modelo espiral alcanzando el desarrollo de una aplicación enfocada a los usuarios asociado y usuarios clientes.

PALABRAS CLAVE: Aplicaciones móviles, Backend, Base de datos, Frontend, Sistema de información, Webservices.

ABSTRACT.

In recent years the production and export of Hass avocado has been increasing in the country, so its constant growth demands solutions that optimize the processes of reception and commercialization of said product. Therefore, a multiplatform information system for the AGROBILBAO tolimense association was presented, which, after its proper collection of information and identification of the main needs, developed a web and mobile application which allowed to facilitate administrative processes, increase the communication and have adequate process management among the members of the association. This was possible through the use of a software development methodology of the spiral model, reaching the development of an application focused on the associated users and client users.

KEYWORDS: Backend, Data bases, Fronted, Information system, Mobile Apps, Web services.

TABLA DE CONTENIDO

CONTENIDO **Pág.**

LISTA DE FIGURAS.

LISTA DE TABLAS.

LISTA DE ANEXOS.

CAPITULO 1

INTRODUCCIÓN.

La tecnología ha hecho posible que empresas puedan mejorar procesos internos, permitiendo localizar falencias y errores de comunicación que causan pérdidas en cualquier proyecto. La producción de aguacate Hass viene en aumento en los últimos años, la disolución de la guerra entre gobierno y grupos al margen de la ley, han permitido que campesinos vuelvan a sus tierras, emprendan nuevamente sus cultivos e inicie una cadena de producción nuevamente, una de las asociaciones creadas en pro de levantar el ejercicio agrícola es AgroBilbao.

AGROBILBAO hace parte de una parte de pequeños productores, ubicados en el departamento del Tolima en el municipio de Planadas. Actualmente este tipo de organizaciones tienen un manejo convencional de la información, se usan libros de Excel muy básicos, utilizan registros de recepción y ventas en hojas físicas; por ello se hace difícil el archivo y almacenamiento de la información. La información no es organizada, no se tienen datos estadísticos que promuevan a generar acciones de respuesta a los diferentes problemas internos, en el acopio y recepción del producto se evidencian falencias, la falta de organización y comunicación efectiva entre los miembros participes activamente en la organización. Lo que acarrea perdidas de coste para la asociación y sus afiliados, las problemáticas detectadas en la asociación objeto de estudio y por su falta de organización interna, al no contar con una herramienta administrativa, donde se puedan tener registradas todas las actividades relevantes que permiten hacer una gestión eficiente.

Con lo anteriormente dicho, la asociación necesita estar a la vanguardia para lograr una distribución y exportación impecable, empezando desde el interior. Un sistema de información representa una herramienta tecnológica adecuada para optimizar los procesos internos de la asociación, permitiendo mitigar los problemas generados por las falencias en organización y falta de comunicación efectiva dentro de la asociación. Por medio del cual los usuarios podrán llevar a cabo algunas labores en un menor tiempo posible, ya que estarán conectados en tiempo real por medio de las aplicaciones web y móvil, donde podrán acceder a los datos y también generar estadísticas para la toma de decisiones claves en la Asociación.

Varios proyectos se han desarrollado con sistemas de información para el área agrícola con el fin de mejorar la comunicación entre campesinos, empresarios y escuelas; en este caso presentamos un sistema de información enfocado a resolver las falencias puntuales de la asociación. Para llevar a cabo un sistema de información que mitigue las falencias que presenta actualmente la asociación AGROBILBAO fue necesario escoger herramientas que permitan desarrollar los aplicativos tanto web como móvil. Tecnologías de motor de base de datos, frameworks de desarrollo que permitan el desacoplamiento de las partes que integran el sistema, desarrollo backend enfocado a la escalabilidad, la eficiencia y la seguridad en la transferencia de información, desarrollo frontend con tecnologías que permitan la creación de interfaces que generen una gran experiencia de usuario.

La utilización de un patrón de diseño enfocado en la reutilización de código, el desarrollo de la interfaz de programación de aplicaciones API (Application Programming Interface) por sus siglas en inglés del lado del backend, lo que permite ser consumida tanto por el aplicativo web como por la aplicación móvil, son algunos de los aspectos reflejados en el cuerpo de este documento, como muestra de la construcción de un producto enfocado a resolver las necesidades de la asociación, dejando abierta la posibilidad de escalar un producto que puede tener más funcionalidades para los asociados.

1. PROBLEMA.

1.1 Planteamiento del problema.

La Asociación de Productores Agropecuarios del Corregimiento de Bilbao (AGROBILBAO) es una entidad sin ánimo de lucro, ubicada en el Municipio de Planadas Tolima, dedicada al cultivo de aguacate Hass tipo exportación, que agrupa aproximadamente 150 familias campesinas de escasos recursos económicos, quienes ven en este cultivo, una oportunidad para mejorar la calidad de vida.

Una de las necesidades puntuales que se presenta en esta asociación, es organizar y centralizar la información de todos los agricultores, para optimizar los procesos internos como la recolección y comercialización del aguacate en el sector y de esta forma, evitar que se realicen gastos innecesarios, que les afecta directamente su economía, pero que además, representa una de las principales falencias para ser competitivos, en un mercado tan exigente como el de las frutas exóticas tipo exportación, porque se debe hacer una coordinación en tiempo real de todas las actividades, especialmente en la recolección de la fruta para que sea llevada a los centros de acopio y de ahí, la administración conocer cuánta producción se tiene, para sí mismo hacerse la coordinación logística especialmente en la contratación de camiones para extraer la fruta hacia los puertos de embarque, como Buenaventura.

El éxito de la producción y comercialización, radica en la comunicación con cada uno de los productores, para que se tenga una buena coordinación de cada una de las actividades que se desarrollarán durante la semana, así como el poderse transmitir los lineamientos dados por los exportadores, en aras de mejorar la calidad para ser más competitivos y así, garantizar la continuidad del negocio. Actualmente por esta falencia, no se ha logrado el impacto que se espera, a raíz de que no se tiene un punto óptimo en cuanto a la recolección y embalaje de la fruta al momento de ser transportado desde los cultivos hasta el centro de acopio, y desde allí al puerto marítimo. Un buena comunicación es sinónimo de una excelente producción, mejora de tiempos de entrega, conocer con anterioridad cuántos kilos se tienen para abriles mercados, etc.

Actualmente por no contarse con un medio de comunicación eficiente entre los productores y el personal administrativo, se han presentado imprecisiones y demoras en las entregas de la producción, acarreando costos innecesarios especialmente en la logística, debido a que en algunas

veces se les queda fruta sin ser evacuada o no sale la carga completa, generándose sobrecostos que se pueden evitar y de paso pérdidas económicas.

Con base a lo anterior, se hace necesario llevar un registro en tiempo real de la cantidad de producto recolectado por cada miembro de la asociación, para de esta forma, obtener un buen control en todo el proceso, que le permita a la administración hacer una mejor gestión para cumplir oportunamente todas las obligaciones adquiridas, especialmente aquellos relacionados con el factor económico, que de ser solucionadas a tiempo, garantizan un proceso eficiente, con un beneficio colectivo para cada uno de los asociados, así como para el sector, porque se genera progreso y por ende mejora la calidad de vida de sus habitantes.

1.2 Formulación del problema.

¿Cómo optimizar los procesos de recepción y comercialización de aguacate hass en la Asociación AGROBILBAO, por medio del desarrollo de un sistema de información multiplataforma que permita mitigar las falencias en organización y centralización de los datos?

2. OBJETIVOS.

2.1 Objetivo general.

Desarrollar un sistema de información multiplataforma para optimizar los procesos de recepción y comercialización de aguacate hass en la Asociación AGROBILBAO.

2.2 Objetivos específicos.

- Recolectar la información fundamental de la actividad económica que realiza la Asociación AGROBILBAO, con el fin de identificar las principales necesidades que ésta tiene en el sector organizacional.

- Desarrollar una aplicación Web, que facilite los procesos con agilidad y transparencia en pro de incrementar la comunicación y productividad de los campesinos pertenecientes a la Asociación.

- Desarrollar la aplicación Móvil con sistema operativo Android, que sirva de complemento facilitador en modo de consulta, para el adecuado manejo de los procesos por parte de los integrantes de la asociación.

- Realizar las pruebas que permitan verificar el correcto funcionamiento de los aplicativos de acuerdo a los requerimientos establecidos para el adecuado manejo de la herramienta.

3. ALCANCES Y LIMITACIONES.

3.1 Alcances.

- Sistematizar el área de administración de procesos en cuanto a recepción, acopio, despacho y comercialización en la asociación AGROBILBAO en el municipio de Planadas Tolima Colombia.

- Brindar información útil al personal administrativo y campesinos integrantes de la Asociación, para que se haga una excelente coordinación de cada una de las actividades realizadas.

- Desarrollar un sistema de información para toma de pre-órdenes y pedidos que se realizan a la Asociación de una cantidad estimada y transporte del producto.

- Estructurar herramientas de captura de datos estadísticos para los futuros usuarios miembros de la asociación AGROBILBAO personal administrativo y campesinos, en cuanto a ventas y producciones realizadas en distintos intervalos de tiempo.

3.2 Limitaciones.

- Teniendo en cuenta que la Asociación AGROBILBAO se encuentra ubicada en otro Departamento y Municipio, respecto a nuestra localización actual, se dificultarán los desplazamientos para la interacción y los trabajos de campo con la comunidad, para la obtención de información que aporte al desarrollo del proyecto.

- Un número muy significativo de los usuarios finales, no cuentan con amplios conocimientos en el uso y manejo de este tipo de herramientas informáticas, lo que genera una barrera y por ende ciertas dificultades en la adaptación y aceptación de la tecnología a implementar.

- Los integrantes de la Asociación en su mayoría, no cuenta con los recursos tecnológicos básicos como un computador o teléfono inteligente, para acceder y hacer uso de la herramienta apropiadamente.

4. JUSTIFICACIÓN.

Los sistemas de información se han convertido en una herramienta de suma importancia en las empresas, debido que, a través de ellos, se pueden gestionar todos los procesos administrativos que serán vitales para la toma de decisiones de una organización y de esta forma, identificar aquellos que se salen del proceso de control en el manejo y uso de la información. La asociación de Productores Agropecuarios del Corregimiento de Bilbao AGROBILBAO, tiene como domicilio principal de su actividad el corregimiento Bilbao en el Municipio de Planadas Tolima.

Esta fue constituida como entidad sin ánimo para comercializar su propia producción, así como poder tener ayuda y asesoría por parte del Estado, a través de diferentes organizaciones gubernamentales. Actualmente este tipo de organizaciones tienen un manejo convencional de la información, donde los registros de la información se llevan en libros foliados, para hacer el registro de la recepción y ventas de la producción; convirtiéndose esta actividad un poco compleja para quienes hacen la contabilidad, porque en la mayoría de los casos la información registrada allí no es clara o no coincide con la real; además no se tienen datos estadísticos que promuevan a generar acciones de respuesta a los diferentes problemas internos.

En el acopio y recepción del producto se evidencian falencias como la falta de organización y comunicación efectiva entre los miembros que participan activamente en la Asociación, lo que produce pérdidas económicas para los afiliados. En este momento por las problemáticas detectadas en la documentación y por la falta de organización interna, al no contarse con una herramienta administrativa como la que se propone, la Asociación no puede rendir un informe en tiempo real, lo que le genera inconvenientes con los organismos de control, así como sus potenciales compradores, quienes en la mayoría de los casos les exigen cifras estadísticas de la producción.

Teniendo en cuenta lo anterior, la implementación de un sistema de información representaría una herramienta tecnológica adecuada para optimizar los procesos internos de la Asociación AGROBILBAO, permitiendo mitigar los problemas generados por las falencias en

organización y falta de comunicación efectiva dentro de la asociación. Por medio del cual los usuarios podrán llevar a cabo las principales tareas y labores en un menor tiempo posible, ya que estarán conectados en tiempo real por medio de las aplicaciones web y móvil, donde podrán acceder a los datos y también, generar reportes para la toma de decisiones claves en la Asociación.

Los sistemas de información son utilizados para la solución de problemáticas de recolección, almacenamiento, procesamiento y distribución de la información, demostrando así que las empresas que los implementan, tienen mejores resultados en sus procesos internos y externos, ya que reducen costos y tiempos en la realización de las tareas operativas de cada organización. Según la Federación Colombiana de la Industria de Software y TI

> *"El sector de Software y Tecnologías de la Información en Colombia cuenta ahora con un completo informe sobre aspectos de importancia y tiene un mapa guía sobre su potencial, obstáculos y retos que debe capitalizar y encarar para convertirse en un motor cada vez más potente de crecimiento y desarrollo para Colombia, en un contexto global en el que ya se habla de la cuarta revolución industrial*." (Fedesoft, 2016)

La eficiencia de un sistema de información se mide en el mejoramiento de los procesos de una empresa respecto a su implementación, influyendo directamente en las actividades a mejorar, en pro de aumentar la calidad del trabajo realizado, disminuir los tiempos de ejecución, que en ocasiones se hace repetitiva para el ser humano y pueden generar errores. Teniendo en cuenta lo anterior, se puede determinar que un sistema de información sería la mejor solución para las problemáticas de la Asociación AGROBILBAO, ya que esta no cuenta con una herramienta que le permita operar y administrar los datos de una manera óptima. Ayudando a mitigar tareas operativas que conllevan demasiado tiempo de ejecución, mejorando los procesos de recepción y comercialización de sus productos.

Por medio de la implementación de un sistema de información web junto con una aplicación móvil a modo de consulta se podrán mantener los datos de manera remota, centralizados y organizados, este sistema contará con dos tipos de usuarios, el asociado y el cliente. Por el lado de los usuarios productores, contarán con la aplicación web y móvil por las cuales podrán confirmar la cantidad del producto para su posterior recolección, consultar los pagos realizados por la Asociación y así mismo, los valores pendientes por pago. También se sabe que en este tipo de

usuarios habrán personas a las que no se les facilite el uso de estas plataformas, para subsanar ello, se contará con un módulo de gestión donde una funcionaria de la Asociación podrá diligenciar esta información, el usuario productor tan solo tendrá que realizar una llamada y suministrar la información requerida; este usuario podrá encontrar también las estadísticas de su producción y realizar sus propios cálculos con base a la producción de sus cultivos. Por otro lado, el usuario cliente podrá ver la cantidad del producto disponible por la asociación en sus bodegas en tiempo real, lista para la comercialización, así mismo realizar y confirmar el pedido según la cantidad deseada mediante la aplicación web o móvil.

5. LÍNEA(s) DE INVESTIGACIÓN.

SOFTWARE, SISTEMAS EMERGENETS y NUEVAS TECNOLOGIAS, conjunto de programas, subprogramas subrutinas y menús que se elaboran a manera de aplicaciones y/o paquetes para cumplir con un fin específico. (Acuerdo No 007 de mayo 29 de 2003).

CAPITULO 2

6. MARCO TEÓRICO.

Actualmente existen en el mercado diferentes herramientas que le permiten a los agricultores perfeccionar, monitorear y optimizar procesos sobre los diferentes tipos de cultivos en sus fincas, tal es el caso que estos programas de software que circulan en dominios públicos y en tiendas de aplicaciones móviles, para los diferentes sistemas operativos que se encuentran disponibles para los teléfonos móviles. Desde la perspectiva más general se presentan como punto de referencia, algunas de estas herramientas mencionadas y asimismo, se dará a conocer al lector aquellos conceptos propios que están relacionados con la realización y ejecución de este proyecto. También se mencionarán temas directamente relacionados con el desarrollo del proyecto de acuerdo a la ingeniería:

6.1 Marco referencial.

Agronet:

Sistema de información desarrollado por el Ministerio de Agricultura y Desarrollo Rural del gobierno colombiano, y con el apoyo de la organización de las Naciones Unidas para la Agri cultura y la Alimentación FAO, para mejorar la comunicación de los campesinos agropecuarios, empresarios y escuelas, ofreciendo un foro para la discusión y para que la comunidad pueda expresar sus ideas e inquietudes. Donde también se realizan capacitaciones a los campesinos sobre el manejo de sus cultivos y herramientas tecnológicas. Dada la importancia de la información para la toma de decisiones, Agronet fue concebida a través del Proyecto TCP/COL/2902 del año 2005, e inicia su funcionamiento con el portal.

El propósito era conformar una red de comunicación integrada y descentralizada que pudiera proveer información estratégica oportuna y sintética a los responsables de la toma de decisiones políticas del sector, brindar a los diversos actores con especial énfasis en los productores, información agropecuaria regional y nacional que fortaleciera sus procesos productivos y de comercialización.

Agronet a su vez ofrece el servicio Celuagronet, a través del cual pueden los usuarios obtener información en tipo real sobre: clima, precios, productividad etc. Con notificaciones sobre capacitaciones y eventos. En el año 2011 se desarrolla CELUAGRONET, servicio de mensajería de texto gratuito, con el que se envía información relevante del sector a los usuarios registrados, pensando principalmente en los productores agropecuarios.

Para el año 2013, Agronet inicia el desarrollo de aplicativos móviles, con el fin que los usuarios puedan acceder desde sus dispositivos móviles a otras herramientas tecnológicas, que les permitan mantenerse actualizado, informado y a la vez ser difusores y generadores de información. (Ministerio de Agricultura, 2018)

Figura 1. Logo Agronet.

Fuente: Agronet (2019) Logo de Agronet. [Imagen]. Recuperado el 12 de febrero de 2019, en: https://www.agronet.gov.co/Paginas/inicio.aspx

BeSoftware:

Es una compañía europea específicamente de España. Su misión es proporcionar soluciones de gestión empresarial a diferentes tipos de compañías. Ellos cuentan con gran experiencia en la implementación de software en el sector agrícola, cárnico y distribución.

Este sistema de información no solo se basa en los procesos operativos, sino que también, suple otras necesidades no tan comunes, como lo son: clasificación de productos, reporte de la trazabilidad y costes de un ciclo completo. La solución óptima estará en el término medio que contiene un modelo de artículos rígidos de alto nivel, que permita analizar la información, pero flexible a bajo nivel para que las operaciones sean ágiles. Idealmente, estará combinado con un

procedimiento de empresa para detectar excepciones y evolucionarlo periódicamente. (Joyanes, 2014)

Sistema de Información Agrícola y Agroindustrial (Agromovil):

Agromovil es un proyecto liderado por la Fundación para la Innovación Tecnológica Agropecuaria (FIAGRO) una organización privada sin fines de lucro, originaria de El Salvador en América Central. Plantean un sistema de información implementado la telefonía móvil, mediante la cual los usuarios puedan consultar, los precios de los productos, información climatológica, bolsa virtual de ofertas y demandas, alertas y obtener asesoría para las técnicas de producción y manejo de los productos a través de una línea telefónica o por medio de un servicio en línea.

De esta manera, el objetivo específico es desarrollar una plataforma de servicios que permita a los productores y las microempresas, acceder a información para mejorar las actividades de producción y comercialización de productos agropecuarios. Este proyecto es una de las primeras iniciativas en El Salvador, quienes utilizaron las nuevas Tecnologías de Información y Comunicación, cómo la telefonía y mensajería móvil para disminuir la brecha que separa a los productores y las pequeñas y medianas empresas, del acceso a la información y el conocimiento. (AgroMovil, 2014)

Figura 2. Imagen de Lanzamiento de Agromovil.

Fuente: Agromovil (2019) Imagen de Lanzamiento de Agromovil. [Imagen]. Recuperado el 12 de febrero de 2019, en: https://www.youtube.com/watch?v=UvOl28nRVjE

AgroWin Sistema de Gestión total para el agro:

Es una empresa dedicada al diseño y desarrollo de aplicaciones de gestión administrativa y contable para las pequeñas y medianas empresas del sector agrícola. AgroWin es un sistema especialmente diseñado para ayudarle al empresario agrícola en la gestión, planeación y seguimiento de la empresa y sus recursos. Adicional a esto, le permite la disminución de los costos, el mejoramiento de los ingresos, el aumento de las utilidades y llevar la contabilidad agropecuaria de manera automática. Cuenta con ingenieros y personal profesional altamente calificado que trabaja día a día en la satisfacción de las necesidades de sus clientes. AgroWin está presente en más de 10 países a nivel de Latinoamérica y Europa, presentando un software que optimiza los procesos de administración, facturación y reportes de acuerdo a las necesidades de la organización. (AgroWin, 2016)

Las ventajas que expone AgroWin con su modelo de negocio son las siguientes:

- Disminuya costos, incremente ingresos, mejore las utilidades y lleve la contabilidad agropecuaria de manera automática.
- Conozca la trazabilidad de su producción
- Realice un control total de la maquinaria y los equipos agrícolas a través del costeo de estos
- Lleve el control total de los inventarios (materias e insumos) de su empresa agrícola
- Obtenga indicadores, delimite las áreas de su empresa agrícola y maneje múltiples empresas con ilimitadas hectáreas a través del sistema de contabilidad agropecuaria líder en Latinoamérica
- Conozca los costos y la producción por cultivo, por lote, por hectárea y por sitio
- Obtenga una administración y control total de la empresa agrícola, la mano de obra, la maquinaria y los insumos a través del mejor sistema de contabilidad agropecuaria
- Realice el costeo especifico de sus inversiones
- Realice un control total de sus clientes y sus proveedores
- Consultas basadas en mapas
- Único programa especializado en cultivos perennes y transitorios

Farmsoft:

Es un software desarrollado para ayudar a controlar procesos agrícolas de trazabilidad post cosecha – embalaje permitiendo planear y gestionar caja, inventarios y pedidos. El software es tan completo que crea alertas al granjero de cuando regar, pulverizar, recolectar y realizar el resto de tareas, y de esta forma, se garantiza que todos hagan el trabajo correcto en el momento preciso, siempre siguiendo los lineamientos correctos de producción. Permite igual ver todos los datos de los cultivos al granjero desde su computador, teléfono o Tablet; manteniendo la afinidad entre el campo y la oficina, dando respuesta por secciones costos, proyecciones y resultados que llevan a darle una visión importante del producto que más beneficie.

Software de postcosecha para disminuir perdidas, mejorar trazabilidad, control de calidad y la eficiencia de embalaje. Farmsoft software de postcosecha hace fácil, inventario, procesamiento, selección, etiquetado, seguimiento de los empleados, ventas, envíos y control de exportación. La solución postcosecha de Farmsoft cuenta con rastreabilidad, procesamiento, empaque, ventas y distribución al por mayor de frutas, verduras, frutos secos, hierbas, especies, lúpulos, flores, frutas y verduras deshidratados, jugos de frutas, aves de corral, y alimentos básicos manufacturados.

Por otra parte, es necesario recalcar que este tipo de software no es un paquete financiero; sino que además, es posible exportar sus facturas, realizar órdenes, listas de asistencia, etc. a su solución financiera, pero no brinda una solución contable como lo haría un paquete financiero especializado. (Farmsoft, 2018)

Geocampo:

Es una empresa de tecnología experta en soluciones de agricultura de precisión para cultivos tropicales. Están enfocados en que la agricultura de precisión es en esencia la toma de decisiones inteligentes, y esta se basa en el acceso oportuno a los datos pertinentes. Los agrónomos entienden que la información precisa y en tiempo real son fundamentales para lograr sus objetivos de optimizar los insumos, maximizar los rendimientos y mejorar la sostenibilidad. Con nuestras herramientas tecnológicas de georreferenciación y teledetección, los agrónomos y agricultores están en capacidad de monitorear sus cultivos, detectar de manera inmediata sus problemas, tomar

oportunamente las acciones correctivas y asegurase que éstas hayan sido efectivamente ejecutadas. (Geocampo, 2017)

Dentro de los servicios que Geocampo como solución tecnológica ofrece están los siguientes:

- Herramienta de manejo integral de plagas (MPI) para cítricos.
- Herramienta de manejo integral de plagas (MPI) para aguacate.
- Herramienta de manejo integral de plagas (MPI) para café.
- Herramienta de manejo integral de plagas (MPI) para cacao.
- Herramienta de manejo integral de plagas (MPI) para guayaba.
- Herramienta de control de labores en el campo.
- Herramienta financiera para gestión de clientes y proveedores.
- Herramienta rastreo vehicular y ruteo.
- Servicio de monitoreo de plagas y enfermedades.
- Servicio de teledetección de problemas.

Comproagro:

Es una plataforma digital para lograr la comercialización de productos del agro, eliminando intermediarios y entregando productos al cliente final (grandes cadenas). De esta forma se mejoran los ingresos del productor agrícola sin afectar el precio de venta al consumidor final. La plataforma le permite registrar el producto que ofrece el campesino, junto con la ubicación, el precio, una breve descripción de su producto y los datos para que un potencial cliente lo pueda contactar.

Nace con el fin de ayudar a eliminar parte de la cadena de intermediación y de esta forma permitir a los agricultores colombianos contactarse con compradores directamente, mejorando sus ingresos y su calidad de vida, cuentan también con una sede en el municipio de Toca Boyacá, en donde trabajan con más de 30 madres cabeza de familia, dando un valor agregado a los productos de la región, con esto se está logrando mejorar la economía de la región y aportar desarrollo y empoderamiento a la mujer rural. (Comproagro, 2016)

6.2 Marco conceptual.

6.2.1 Generalidades.

Colombia experimenta un cambio trascendental desde la firma de los acuerdos de paz, el sector agrícola sin duda alguna es uno de los gremios más golpeados por el conflicto interno que se desarrolló por más de 50 años en todo el territorio nacional. "En Colombia hay 8.376.463 víctimas del conflicto armado, razones suficientes para trabajar por un país en paz" aseguró Juan Manuel Santos en ejercicio de su mandato presidencial en abril de 2017 y para hacer referencia al campo colombiano, Según el Registro Único de Víctimas (RUV), 7.134.646 son casos de desplazamiento, en su gran mayoría campesinos que dejaron atrás el ejercicio agrícola, huyendo de los combates y del hostigamiento armado en busca de nuevas oportunidades en las grandes ciudades. (Portafolio, 2017)

El desplazamiento forzado de estos habitantes campesinos trajo como consecuencia el debilitamiento del ejercicio agrícola en el país, por esta causa el fin del conflicto armado representa un gran salto en la reparación del agro colombiano. La oportunidad de que miles de familias regresen al campo a retomar sus labores, es sin duda un paso firme para el desarrollo y crecimiento de la nación, pero estas familias necesitan del apoyo del gobierno y de iniciativas que incentiven la comercialización de sus productos.

La conformación de cooperativas o asociaciones de pequeños productores agrícolas asentados en una misma región, resulta ser de gran importancia para la comercialización de los productos cosechados, al reunir y organizar varios productores se hacen más llamativo para poder atraer clientes de mayor envergadura o incluso tener la posibilidad de exportar; es así como nace la Asociación de Productores Agropecuarios del Corregimiento De Bilbao AgroBilbao y otras en la región, que presentan objetos sociales muy similares. Para estas organizaciones es fundamental encontrar la manera de mejorar los procesos internos junto con el manejo y la centralización de su información.

6.2.2 Sistema de Información.

Los SI (Sistemas de Información) son relacionados en la mayoría de las veces con conceptos de la informática; las personas tienden a confundir y sesgar el concepto hacia un enfoque

tecnológico, pero en realidad es algo mucho más global. Un SI es un conjunto de elementos que trabajan de manera coordinada para un fin en común.

> *"Podemos plantear la definición técnica de un sistema de información como un conjunto de co1mponentes interrelacionados que recolectan (o recuperan), procesan, almacenan y distribuyen información para apoyar los procesos de toma de decisiones y de control en una organización. Además de apoyar la toma de decisiones, la coordinación y el control, los sistemas de información también pueden ayudar a los gerentes y trabajadores del conocimiento a analizar problemas, visualizar temas complejos y crear nuevos productos."* (Laudon, 2012, p.15)

Existen diversos autores que exponen las funciones que debe cumplir un sistema de información, para que sea totalmente funcional y beneficioso para la organización; en primer lugar se encuentra la alimentación del SI donde se encuentran los mecanismos necesarios para la captura de los datos de interés, en segundo lugar hay un procesamiento de estos datos recolectados haciéndolos más perceptibles para la organización en contexto y por último, existe una etapa de resultado o producto donde estos datos ya procesados, se convierten en información útil para la toma de decisiones.

6.2.3 Bases de Datos.

Un sistema de bases de datos permite recopilar una cantidad aleatoria de datos y organizarlos de manera estructurada, de tal forma que facilite su comprensión y manipulación para los intereses propios del propietario o de quien esté a cargo de la información. Según el libro de Lenguaje SQL con MySQL "una base de datos está constituida por un conjunto de información relevante para una empresa o entidad, junto con los procedimientos para almacenar, controlar, gestionar y administrar esa información" (Lenguaje SQL con MySQL, 2002).

En principio los sistemas de bases de datos tradicionales eran libros, papel o los llamados archivos, donde las organizaciones almacenaban su información de manera física, hoy en día junto con el desarrollo tecnológico y el crecimiento de las ciencias de la informática, la mayoría de las bases de datos evolucionan a formatos digitales convirtiéndose en una gran solución a los inconvenientes de organización y centralización de los datos.

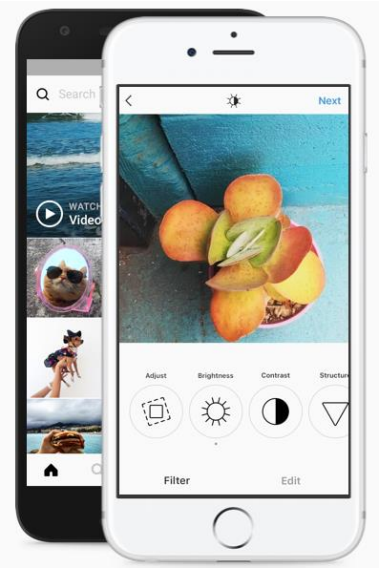
Con la evolución del tratamiento de la información por parte de las organizaciones, surge un nuevo concepto en la aplicación de las mismas; los sistemas de gestión de bases de datos (SGBD), de acuerdo con algunos autores, un SGBD es una herramienta de propósito general, útil para estructurar, almacenar y controlar los datos, ofreciendo interfaces de acceso a la base de datos. Las tareas fundamentales que desempeñan estos sistemas, hacen referencia a la seguridad de acceso a los documentos, al mantenimiento de la integridad de la información, a mecanismos de recuperación debidos a fallos físicos y lógicos, al control de concurrencia en el momento de acceder a los datos y a la eficiencia del sistema evaluada, generalmente, en términos del tiempo de respuesta a las consultas de los usuarios. (McLeod Dennis, 1981)

6.2.4 Apps.

Una App es el acrónimo de la palabra aplicación dentro del contexto de la informática; es una aplicación de software que se instala en dispositivos móviles tales como tabletas o celulares. La finalidad de las Apps o aplicaciones móviles es facilitar la realización de tareas del día a día de las personas y de brindar asistencia, en diversos casos, depende de la finalidad de la aplicación sean de carácter profesional, de ocio, de acceso a servicios, educativa o de cualquier índole. Desde el lanzamiento del App Store en Julio de 2008 existen aplicaciones para todo tipo de actividad desde WhatsApp para mensajería, redes sociales, ventas, juegos, servicios o para vender un producto o hasta para llevar la dieta de una mascota y lo que buscan, es ayudar en esos procesos agilizando los trabajos y optimizando el tiempo de las personas.

> "*Las aplicaciones —también llamadas apps— están presentes en los teléfonos desde hace tiempo; de hecho, ya estaban incluidas en los sistemas operativos de Nokia o BlackBerry años atrás. Los móviles de esa época, contaban con pantallas reducidas y muchas veces no táctiles, y son los que ahora llamamos fe ature phones, en contraposición a los smartphones, más actuales. En esencia, una aplicación no deja de ser un software. Para entender un poco mejor el concepto, podemos decir que las aplicaciones son para los móviles lo que los programas son para los ordenadores de escritorio".* (Javier Cuello, 2014)

Figura 3. Instagram es una de las aplicaciones más famosas para tomar fotos y videos.



Fuente: Instagram (2019) Ilustración de Instagram [Imagen]. Recuperado el 20 de marzo del 2019, en: https://www.instagram.com/?hl=es-la

6.2.5 Backend.

Backend es la capa de acceso a datos de un software o cualquier dispositivo, que no es directamente accesible por los usuarios, el desarrollo backend está del lado servidor, además contiene la lógica de la aplicación que maneja esos datos, en respuesta a las distintas peticiones hechas por el usuario o el sistema. El servidor, que es una aplicación especializada que entiende, la forma como el navegador hace peticiones y generar las respuestas a ellas.

Existen diversos lenguajes de programación y frameworks para estar al lado del servidor como programador, dependerá de la aplicación solicitada o de la tecnología utilizada para distintos tipos de proyectos. Algunos de los más conocidos son Python, PHP, Ruby, C#, Nodejs y Java etc. Cada uno de los anteriores, tiene diferentes frameworks que te permiten trabajar mejor según el proyecto que se está desarrollando.

Pero además de dominar un lenguaje de programación y su framework de preferencia, es necesario estar familiarizado a los sistemas de bases de datos; pues todo buen proyecto de aplicación de software y un buen desarrollador backend, emplea una base de datos dentro de su lógica. Los más comunes motores de bases de datos existentes son SQL Server, MySQL,

PostgreSQL, Oracle, MongoDB y de la misma manera que el lenguaje de programación de preferencia, el desarrollador se inclinara por el motor de base de datos de favoritismo.

6.2.6 Frontend.

Frontend es la parte de un programa, aplicación o software especializado a la que un usuario puede acceder directamente. Abarca la parte visual en la que se mueve el usuario final, son todos aquellos componentes externos que visualiza el cliente final. Son todas las tecnologías de diseño y desarrollo web que corren en el navegador y que se encargan de la interactividad con los usuarios.

HTML, CSS y JavaScript son los lenguajes principales del Frontend, los cuales un desarrollador frontend debe dominar y de los cuales se desprenden una cantidad de frameworks y librerías que expanden sus capacidades para crear cualquier tipo de interfaces de usuarios. React, Redux, Angular, Bootstrap, Foundation, LESS, Sass, Stylus y PostCSS son algunos de ellos.

6.2.7 Web Service.

Un servicio web es una tecnología que usa distintos tipos de protocolos y pautas que permiten la interpretación de código de diferentes lenguajes de programación; esta tecnología es utilizada para intercambiar datos entre aplicaciones de software. Algunos lo interpretan como la posibilidad de la comunicación entre dos máquinas.

> *"Un web service es una vía de intercomunicación e interoperabilidad entre máquinas conectadas en Red. En el mundo de Internet se han popularizado enormemente, ya se trate de web services públicos o privados. Generalmente, la interacción se basa en el envío de solicitudes y respuestas entre un cliente y un servidor, que incluyen datos. El cliente solicita información, enviando a veces datos al servidor para que pueda procesar su solicitud. El servidor genera una respuesta que envía de vuelta al cliente, adjuntando otra serie de datos que forman parte de esa respuesta. Por tanto, podemos entender un servicio web como un tráfico de mensajes entre dos máquinas."* (Arsys, 2015)

6.2.8 Interoperabilidad.

La interoperabilidad es la habilidad de intercambiar información entre dos o más sistemas y tener un adecuado aprovechamiento de esa información. Para entrar en contexto, en Colombia existe el Marco para la Interoperabilidad del Gobierno en línea, el cual brinda una definición citada a continuación: ejercicio de colaboración entre organizaciones para intercambiar información y

conocimiento en el marco de sus procesos de negocio, con el propósito de facilitar la entrega de servicios en línea a ciudadanos, empresas y a otras entidades. (MinTic, 2010)

Mientras que la comisión europea por medio del Marco Europeo de Interoperabilidad (EIF) la define como:

> *"A efectos del EIF, la interoperabilidad es la capacidad de que las organizaciones interactúen con vistas a alcanzar objetivos comunes que sean mutuamente beneficiosos y que hayan sido acordados previa y conjuntamente, recurriendo a la puesta en común de información y conocimientos entre las organizaciones, a través de los procesos empresariales a los que apoyan, mediante el intercambio de datos entre sus sistemas de TIC respectivos".* (Comisión Europea, 2017)

6.2.9 Frameworks.

Un framework es un entorno o marco de trabajo para programadores, con un conjunto de herramientas específicas también llamadas librerías, utilizadas especialmente en el desarrollo de software, lo cual permite que sea más fácil y rápido, garantizando de esta forma, la estructura de los productos que se están diseñando. Un entorno de trabajo de este tipo, brinda la posibilidad de ejercer control en el desarrollo de software, aplicando técnicas de patrones de diseño o metodológicas con distintos elementos. Es importante tener en cuenta que cada lenguaje de programación cuenta con sus propios frameworks y librerías, por ejemplo para el lenguaje de programación JavaScript, este tiene frameworks llamado Angular que ayuda a optimizar el tiempo de trabajo simplificando la creación de diversas rutinas.

La siguiente es una definición sencilla y fácil de comprender de lo que es un framework adaptada a un tipo de trabajo específico tal como el desarrollo web. En este caso se tiene que:

> *"Un framework para aplicaciones web es un software o conjunto de librerías, que está diseñado para dar soporte al desarrollo de sitios y en general a la construcción de cualquier aplicación web. Entonces un framework trata de facilitar aquellas actividades comunes realizadas durante el desarrollo de la aplicación, como, por ejemplo: acceso a la base de datos, uso de plantillas, manejo de sesiones, separación de aspectos de programación; además de promover la re utilización de código*." (Vázquez, 2011)

6.2.10 Librerías.

Las librerías son las herramientas presentes en un framework para resolver un problema específico, se pueden comprender como fragmentos de código que ayudan a resolver problemas y con un propósito determinado. El propósito de ellas es optimizar tiempo y agilizar el desarrollo. Para un lenguaje de programación como JavaScript, existen librerías como React.js, que ayuda a facilitar la creación de interfaces de usuario en el proyecto a realizar.

Para dejar claro el concepto, se puede pensar en un framework como la receta para preparar una comida rápida, en este caso una hamburguesa, en donde se tienen todos los ingredientes y pasos a seguir. Una librería para este caso, sería la salsa de tomate, el cual no exige que se cree las salsas desde cero, porque a través de él, se realizan las mimas funciones y simplemente se toma y se utiliza.

6.2.11 IDE

Integrated Development Enviroment, en español significa (Entorno de Desarrollo Integrado), es una herramienta realizada por desarrolladores para desarrolladores de software, la cual es utilizada para la creación de código, que integran a su vez diferentes "plugins" (grupos de funciones o características), pero además añaden funcionalidades para la realización de diversas tareas específicas, como lo puede ser la ejecución de un servidor web, la descarga de dependencias o librerías para un proyecto, hasta la compilación del proyecto, la conexión a un repositorio del código, etc.

El IDE es el conjunto de herramientas unidas para la realización de un determinado proyecto o aplicación; entre los IDE más populares se encuentra a Eclipse, Intelligen IDEA y Netbeans para el desarrollo en Java, Visual Studio para el desarrollo en .NET y con el lenguaje de programación C#. También hay muchos otros IDE, los cuales se comportan de diferentes maneras y sí lo que se quiere es escoger uno solo, basta con explorarlo y establecer cuál es el que más se ajusta a las necesidades.

6.3 Marco Ingenieril.

6.3.1. PostgreSQL

PostgreSQL es un motor de base de datos muy robusto, esta herramienta es de código abierto (Open Source), cuenta con una amplia comunidad de colaboradores a nivel global, también llamada PGDG (PostgreSQL Global Development Group), es muy usado por distintas compañías a nivel mundial, como un motor de base de datos relacional, aunque ya cuenta con soporte para ser una base de datos NoSQL o no relacional, con tipo de dato nativo JSON, además de mejorar la sintaxis para este tipo de consultas. Actualmente PostgreSQL se encuentra en la versión 11.

> *"PostgreSQL es un potente sistema de base de datos relacional de objetos de código abierto que utiliza y amplía el lenguaje SQL combinado con muchas características que almacenan y escalan de forma segura las cargas de trabajo de datos más complicadas. Los orígenes de PostgreSQL se remontan a 1986 como parte del proyecto POSTGRES en la Universidad de California en Berkeley y tiene más de 30 años de desarrollo activo en la plataforma central*." (PostgreSQL, 2019)

Dentro de las principales características de este sistema de gestión de base de datos están:

- Orientado a Objetos: esto es porque todos los elementos de la base de datos pueden ser tratados como objetos en algunos lenguajes de programación.
- Multisistema: funciona en diversos sistemas operativos como Microsoft Windows, Mac OS, Linux y muchos otros sistemas operativos.
- Extensible: se pueden añadir funciones y herramientas que no sean originales o no suministradas de serie.
- Escalable: maneja bases de datos con millones de datos.
- Bajo licencia libre: donde se puede usar para cualquier proyecto a implementar.

Son grandes las ventajas de PostgreSQL, está dotado de muchas características que pretenden ayudar al desarrollador a crear aplicaciones, múltiples prestaciones, la integridad de los datos, la tolerancia a fallas. En su versión 11 lanzada en octubre de 2018 PostgreSQL cumple 160 de las 179 funciones obligatorias para el estándar SQL (Estructurad Query Language) con el fin de

cumplir con el mismo y permanecer en la vanguardia de los sistemas de gestión de bases de datos como uno de los más usados a nivel mundial.

A continuación, se muestra una lista de las diversas ventajas y características que se encuentran en PostgreSQL, tomadas directamente de su sitio web oficial.

- **Tipos de datos**
  - Primitivas: entero, numérico, cadena, booleano
  - Estructurado: fecha / hora, matriz, rango, UUID
  - Documento: JSON / JSONB, XML, valor-clave (Hstore)
  - Geometría: Punto, Línea, Círculo, Polígono
  - Personalizaciones: Compuestas, Tipos Personalizados.
- **Integridad de los datos**
  - ÚNICO, NO NULO
  - Llaves primarias
  - Llaves extranjeras
  - Restricciones de exclusión
  - Cerraduras explícitas, cerraduras consultivas
- **Concurrencia, rendimiento**
  - Indexación: B-tree, Multicolumn, Expresiones, Parcial
  - Indexación avanzada: GiST, SP-Gist, KNN Gist, GIN, BRIN, índices de cobertura, filtros Bloom
  - Planificador / optimizador de consultas sofisticado, análisis de solo índice, estadísticas de varias columnas
  - Transacciones, transacciones anidadas (a través de puntos guardados)
  - Control de concurrencia multi-versión (MVCC)
  - Paralelización de consultas de lectura y creación de índices de árbol B
  - Particionamiento de tablas
  - Todos los niveles de aislamiento de transacciones definidos en el estándar SQL, incluido Serializable
  - Recopilación de expresiones Just-in-time (JIT)

- **Confiabilidad, Recuperación de Desastres**
  - Registro de escritura anticipada (WAL)
  - Replicación: asíncrona, síncrona, lógica
  - Recuperación de punto en el tiempo (PITR), recursos activos
  - Espacios de tabla
- **Seguridad**
  - Autenticación: GSSAPI, SSPI, LDAP, SCRAM-SHA-256, certificado y más
  - Sistema robusto de control de acceso
  - Seguridad de columnas y filas
- **Extensibilidad**
  - Funciones y procedimientos almacenados.
  - Lenguajes de procedimiento: PL / PGSQL, Perl, Python (y muchos más)
  - Contenedores de datos externos: conéctese a otras bases de datos o flujos con una interfaz SQL estándar
  - Muchas extensiones que proporcionan funcionalidad adicional, incluyendo PostGIS
- **Internacionalización, Búsqueda de texto**
  - Soporte para conjuntos de caracteres internacionales, por ejemplo, a través de colaciones de UCI
  - Búsqueda de texto completo

Las ventajas del sistema de gestión de bases de datos se pueden profundizar en la documentación presente en su sitio web oficial, es código abierto y la información puede estar a la mano de cualquier interesado. (PostgreSQL, 2019)

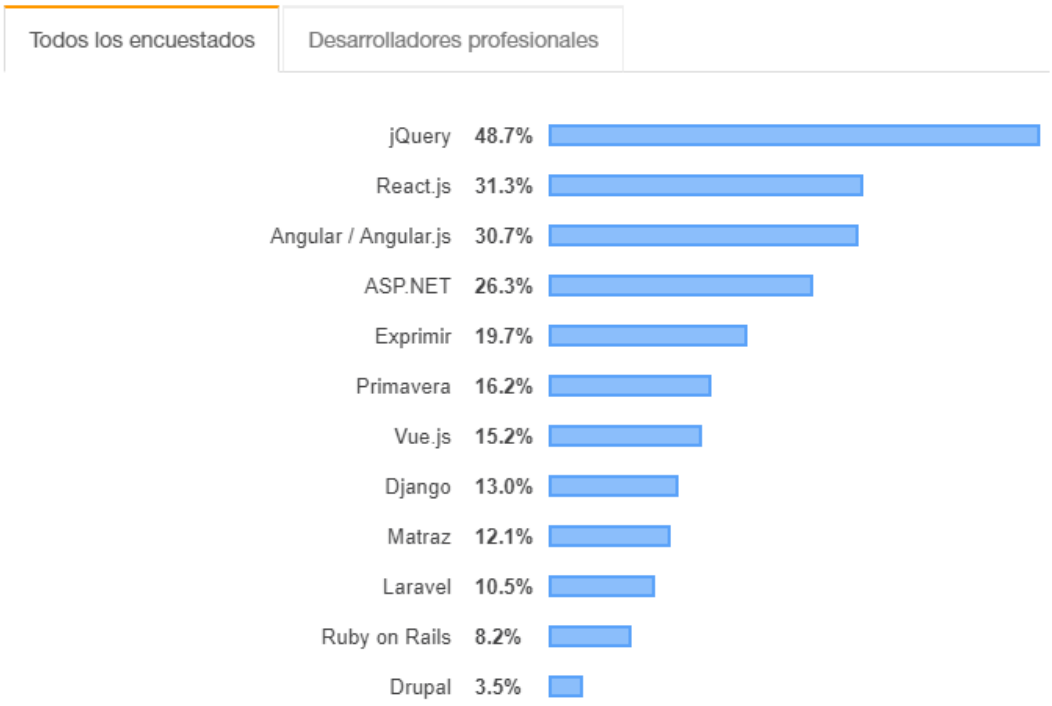
En este proyecto se utilizó a PostgreSQL como el sistema de gestión de bases de datos por varias razones: una de las más importantes es por ser de código abierto, las prestaciones que tiene en cuanto a la transaccionalidad e integración con diversas tecnologías, la organización de la información, es multiplataforma.

6.3.2. ReactJS

ReactJS es una librería basada en JavaScript, para el desarrollo de aplicaciones web, esta librería es desarrollada y soportada por el equipo de desarrollo de Facebook, se ha convertido en una de las librerías más usadas según informe estadístico de Stack Overflow, una comunidad online de desarrolladores informáticos, en donde estos pueden encontrar soluciones a diversos problemas durante el ejercicio de la programación, en diferentes lenguajes.

Cada año Stack Overflow realiza una encuesta, con la cual le preguntan a la comunidad diversos temas en el ejercicio de su profesión. En el 2019 casi 90,000 desarrolladores dijeron cómo aprenden y suben de nivel, qué herramientas están usando, las tecnologías de preferencia y qué quieren. (StackOverflow, 2019)

Figura 4. Resultados de Framework y librería de preferencia.



Fuente: Stack Overflow (2019) Resultados de Framework y librería de preferencia. [Imagen]. Recuperado el 17 de abril de 2019 del 2019, en: https://insights.stackoverflow.com/survey/2019#overview

Ante la pregunta sobre los marcos o entornos de desarrollo de preferencia, ReactJS se posiciona en un segundo lugar, lo que confirma la aceptación por parte de los programadores hacia

esta tecnología. Esta librería se caracteriza por tener una developer experience elevada, su fácil implementación y sintaxis hacen de ReactJS la favorita para ser utilizada por el desarrollador para este tipo de aplicaciones y que los desarrolladores que la usan una vez, quieran volver a usarla para sus futuros desarrollos. Los países desarrollados están optando por usar ReactJS para sus aplicaciones, por su ideología de componentes "reactivos", haciendo que librerías como jQuery, Ajax y hasta el mismo Angular, queden en desventaja si son comparadas con esta. Actualmente ReactJS se encuentra en la versión 16.8.2.

Platzi es una plataforma de educación online de formación profesional en tecnología, con gran surgimiento y crecimiento en la actualidad para Latinoamérica y el mundo; en un artículo publicado en su sitio web (Platzi, 2018), exponen las principales características de la tecnología de ReactJS, las cuales se tienen a continuación:

¿Cuáles son las características de ReactJS?

- Utiliza el DOM virtual en lugar del DOM real.
- Utiliza el renderizado del lado del servidor.
- Sigue un flujo de datos unidireccional.

Principales ventajas de ReactJS.

- Aumenta el rendimiento de la aplicación
- Se puede utilizar cómodamente tanto en el lado del cliente como del servidor.
- Debido a JSX, la legibilidad del código aumenta
- React es fácil de integrar con frameworks como Meteor, Angular, etc.
- Usando React, escribir casos de prueba de UI se vuelve extremadamente fácil.

¿Cuáles son las limitaciones de ReactJS?

- React es sólo una biblioteca, no un framework completo.
- Su biblioteca es muy grande y lleva tiempo comprenderla.

- Puede ser poco difícil para los programadores novatos entender
- La codificación se vuelve compleja a medida que usa plantillas en línea y JSX

6.3.3 JavaScript

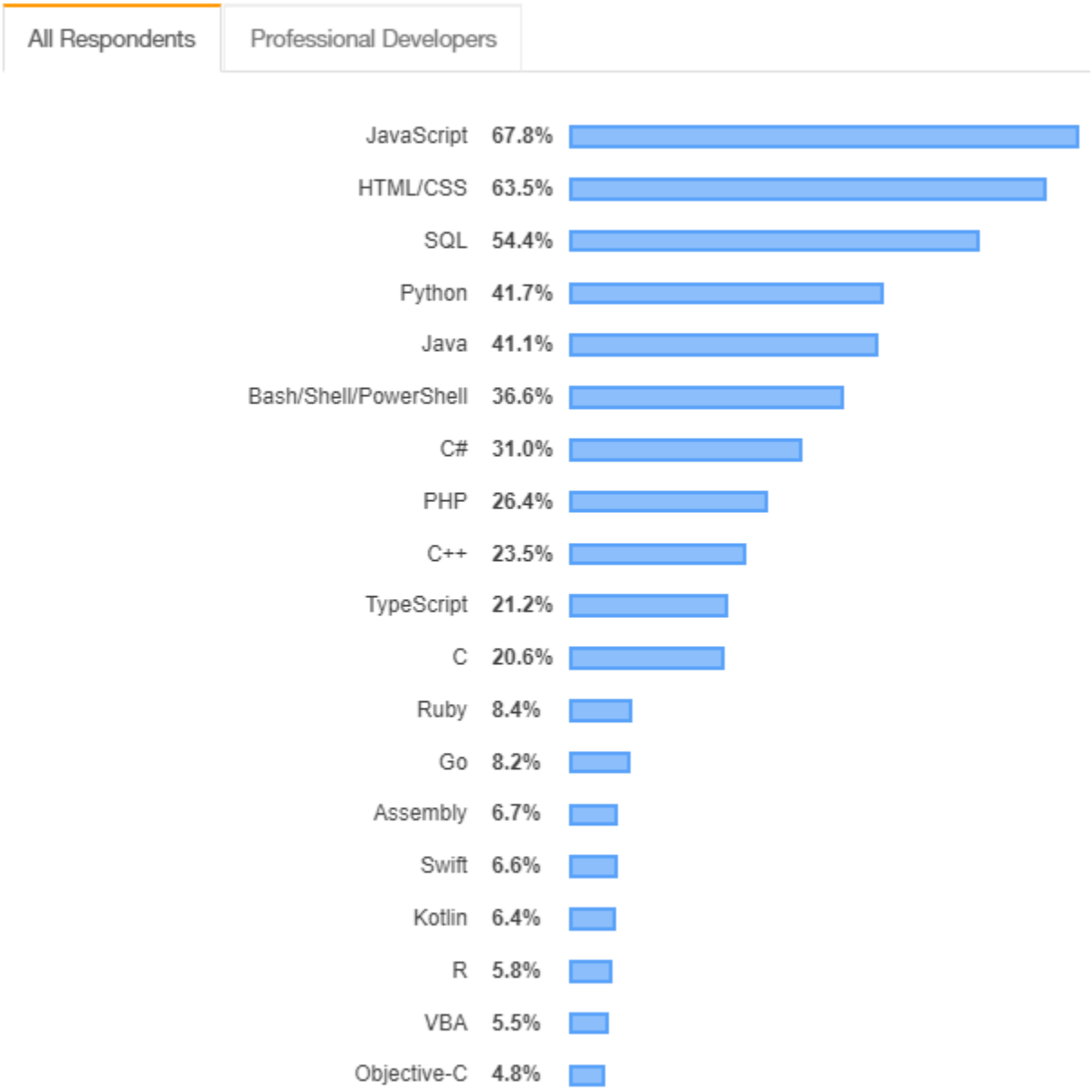
JavaScript también conocido con su abreviación como JS, es un lenguaje de programación interpretado; esto quiere decir que no requiere de compilación sino que es analizado por otro programa, que en este caso específico son los navegadores web, expresado de otra manera, se concluye que JavaScript está diseñado para ser ejecutado en los navegadores (Internet Explorer, Mozilla Firefox, Opera 10, Google Chrome, Apple Safari); es apropiado resaltar que este lenguaje es utilizado para ejecutar acciones del lado del cliente, lo que no significa que no se pueden utilizar aplicaciones de JavaScript del lado del servidor, incluso se hacen este tipo de soluciones.

> *"JavaScript es un lenguaje de los denominados lenguajes de scripting. Los scripts (script se traduce como guion, literalmente) son archivos de órdenes, programas por lo general simples. Es por esto que no podemos definir JavaScript como un lenguaje de programación en un sentido estricto, pero sin embargo sí nos permite crear páginas dinámicas, con algunos efectos realmente interesantes y que mejoren considerablemente su aspecto. Nos permite tener cierta interacción con el usuario de nuestras páginas, reconocer determinados eventos que se puedan producir y responder a éstos adecuadamente. Podemos, por ejemplo, añadir elementos con movimiento que recuerdan a las animaciones Flash. Incluso podemos crear algunos programas más complejos que manejen estructuras de datos."* (Alba, 2011)

La principal función del lenguaje es agregar interactividad a las páginas web, la cual consiste en que el usuario responde a múltiples eventos y estos proveen de información y de instrucciones al sistema de la página web usada. JavaScript está definido como un lenguaje de programación orientada a objetos (POO) lo que lo hace una excelente herramienta para los distintos tipos de proyectos a realizar.

Este lenguaje de programación se consolida como uno de los más usados en el mundo, es sobre lo que está implementada la web moderna, se ejecuta tanto el frontend como el backend y la tendencia es a ser más utilizado en los próximos años. En la encuesta ya mencionada del sitio Stack Overflow, se puede comprobar el impacto y la aceptación que tiene el lenguaje. A continuación en la figura 5, se ilustran los resultados ante la pregunta de las tecnologías más usadas:

Figura 5. Lenguaje de programación más utilizado



Fuente: Stack Overflow (2019) Resultados de Framework y librería de preferencia. [Imagen]. Recuperado el 18 de abril de 2019 del 2019, en: https://insights.stackoverflow.com/survey/2019#technology

El informe indica que, por séptimo año consecutivo, JavaScript es el lenguaje de programación más utilizado, razón por la cual, muchos programadores y profesionales en tecnología, se interesan en la utilización de estas herramientas, por ser uno de los más populares y esta condición facilita la presencia de contenido y de soporte en línea, los cuales ayudan a los desarrolladores a resolver sus dudas con más facilidad. (StackOverflow, 2019)

6.3.4 NodeJS

NodeJS es un entorno de ejecución multiplataforma para aquellos proyectos realizados en JavaScript, que ha adquirido una acogida bastante importante en el mundo del desarrollo de software, siendo la más utilizada al momento de realizar un proyecto tanto web como móvil. Concebido como un entorno de ejecución de JavaScript orientado a eventos asíncronos, está diseñado para construir aplicaciones en red escalables.

Machine de Ruby o Twisted de Python y Node lleva el modelo de eventos un poco más allá, este presenta un bucle que es utilizado como un entorno en vez de una librería. En otros sistemas siempre existe una llamada que bloquea para iniciar un bucle o un ciclo, de manera síncrona, teniendo que realizar o ejecutar las acciones con el orden de codificación, en cambio, NodeJS funciona de manera asíncrona y orientado a eventos; en este caso la ejecución de una tarea no va a bloquear o a retrasar la de otra, lo que lo hace muy eficiente. (NodeJS, 2019)

Actualmente NodeJS se encuentra en la versión 11. Están en constante desarrollo de nuevas tecnologías en su entorno; la versión 11 fue lanzada el 23 de octubre del 2018. En NodeJS tratan de establecer un cronograma de control de versiones, establecen las fechas de sus lanzamientos, las fechas de mantenimiento y el fin de la versión que ya cumple un ciclo. A continuación, la imagen nos ilustra las versiones actuales de NodeJS extraídas desde su sitio web:

Figura 6. Versiones de NodeJS.

Versión	Estado	Nombre Clave	Lanzamiento Inicial	Inicio LTS Activo	Inicio Mantenimiento LTS	Fin-de-vida
v6	Mantenimiento LTS	Boron	2016-04-26	2016-10-18	2018-04-30	2019-04-30
v8	Mantenimiento LTS	Carbon	2017-05-30	2017-10-31	2019-01-01	2019-12-31
v10	LTS Activo	Dubnium	2018-04-24	2018-10-30	2020-04-01	2021-04-01
v11	Actual		2018-10-23		2019-04-22	2019-06-01
v12	Actual		2019-04-23	2019-10-22	2021-04-01	2022-04-01
v13	Pendiente		2019-10-22		2020-04-20	2020-06-01
v14	Pendiente		2020-04-21	2020-10-20	2022-04-01	2023-04-01

Fuente: NodeJS (2018) Versiones de NodeJS. [Imagen]. Recuperado el 13 de marzo del 2019, En: https://nodejs.org/es/about/releases/

6.3.5 Ionic Framework

Es un framework para el desarrollo de aplicaciones móviles, aplicando la sintaxis de las tecnologías web, se pueden crear increíbles aplicaciones móviles con una alta interoperabilidad, y lo más importante, es que se pueden crear aplicaciones para los diferentes sistemas operativos de móviles (Android y IOS). Actualmente se encuentra en la versión 4.

Ionic Framework es un kit de herramientas de interfaz de usuario de código abierto para crear aplicaciones móviles y de escritorio de alta calidad y de alto rendimiento utilizando tecnologías web (HTML, CSS y JavaScript). (ionicframework, 2019)

A continuación, se hace referencia de algunas de las características principales de 5 Ionic Framework:

- **Es multiplataforma** - Se trata de un marco para la construcción de aplicaciones móviles iOS nativo, Android, web o las aplicaciones web progresivas.
- **Se basa en estándares web** – Se compone de las conocidas tecnologías web (CSS, HTML, JavaScript).
- **Diseño** -Dispone de modelos y componentes prediseñados que le permiten al desarrollador implementarlos y generar una experiencia de usuario (UX) de alta calidad en los proyectos que realice.
- **Sencillez** – En Ionic se han interesado por tener un framework sencillo y con una curva de aprendizaje ágilmente variable para el desarrollador que posee habilidades de desarrollo web.
- **Licencia** -Es un proyecto de código abierto (open source) y quien requiera de su utilización para sus proyectos puede usar esta gran herramienta sin costo alguno.

- **Soporte** -Para un ágil desarrollo de las interfaces Ionic desde un comienzo soportó Angular como herramienta para el desarrollo, pero actualmente se encuentran versiones disponibles que trabajan con librerías como Vue y React.

6.3.6 Python

Es un lenguaje de programación con gran popularidad en la actualidad; la historia de este lenguaje inicia con su creador Guido Van Rossum, quien a finales de la década de los años 80 y comienzos de los 90 inició a desarrollar un lenguaje, que en principio lo tomó como pasatiempo pero que con el pasar de los años se convirtió en un proyecto de grandes dimensiones. La primera versión publicada fue la 0.9.0 en 1991 en esta versión ya se manejaban conceptos como clase, herencia, manejo de excepciones; después Python siguió su crecimiento como un proyecto colaborativo bajo la licencia Open Source y en la actualidad la última versión estable es la 3.7.4 liberada el 8 de julio de 2019.

> *"Python es un lenguaje de programación el cual se usa para muchas aplicaciones diferentes. Se usa en algunas escuelas secundarias y universidades como un lenguaje de programación introductorio porque Python es fácil de aprender, pero también es utilizado por desarrolladores de software profesionales en lugares como Google, NASA y Lucasfilm Ltd."* (Foundation., 2019)

6.3.7 Django Resto Framework

El desarrollo web ha evolucionado y cambiado con los últimos años; con estas actualizaciones se han implementado frameworks de desarrollo que facilitan el trabajo del programador. Con estas iniciativas nacen alternativas como Django, una de las herramientas de desarrollo más poderosas para construir aplicaciones web basadas en Python.

> *"Django es un marco web Python de alto nivel que fomenta el desarrollo rápido y el diseño limpio y pragmático. Creado por desarrolladores experimentados, se ocupa de gran parte de la molestia del desarrollo web, por lo que puede concentrarse en escribir su aplicación sin necesidad de reinventar la rueda. Es gratis y de código abierto."* (Foundation, 2019)

Django se crea en 2004 pensado para desarrollar grandes aplicaciones. Algunas de las características que hacen a este framework muy poderoso son el manejo formulario, rutas, métodos

de autenticación, la capacidad de ser extensible, escalable y adicionalmente es open source; pero una de las más importantes y poderosas características es el manejo del ORM (Object-Relational mapping) un plus que se encarga del manejo y la interacción con la base de datos en el manejo de la información que se necesite. Dicha funcionalidad es compatible con los principales y más conocidos motores de base de datos (PostgreSQL, MySQL, Oracle, SQLServer).

6.3.8 HTML 5

Hyper Text Markup Language traduce esta sigla es decir lenguaje de marcas de hipertexto. Es el lenguaje con el que es construido Internet, muchas personas lo definen como un lenguaje de programación, pero no lo es, se utiliza en realidad para modelar la forma en que se estructura un sitio web.

> *"HTML usa un lenguaje de etiquetas para construir páginas web. Estas etiquetas HTML son palabras clave y atributos rodeados de los signos mayor y menor (por ejemplo,). En este caso, HTML es la palabra clave y lang es el atributo con el valor. La mayoría de las etiquetas HTML se utilizan en pares, una etiqueta de apertura y una de cierre, y el contenido se declara entre ellas."* (Gauchat, 2012)

HTML tiene como raíz la etiqueta <HTML> la cual se debe crear como etiqueta de apertura y luego de cierre. Luego se divide en dos partes; la primera es la cabecera, la etiqueta <head>que es donde se colocan todas las etiquetas especiales, pero que no se ven en el navegador, por ejemplo, el título, la descripción o la conexión con css. Luego la segunda parte es el body o cuerpo, la etiqueta <body> es donde está lo que se ve en el navegador, por ejemplo, párrafos de texto o imágenes. Lo anterior se muestra graficamente en la siguiente figura (Fig. 7):

Figura 7. Estructura Básica de HTML

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>Document</title>
</head>
<body>

</body>
</html>
```

Fuente: Elaboración propia.

6.3.9 Css3

CSS en sus siglas en inglés traducen (Cascading Style Sheets), lo que español es traducido como “Hojas de estilo en cascada", es un lenguaje de diseño gráfico que se usa para realizar el diseño visual de aplicaciones web. Según (Gauchat, 2012) CSS es un lenguaje que trabaja junto con HTML para proveer estilos visuales a los elementos del documento, como tamaño, color, fondo, bordes, etc.

CSS es una tecnología que apoya el desarrollo de páginas web, haciéndolo más exacto y dinámico. El contenido de títulos, párrafos, enlaces, listas, tablas y formularios es totalmente independiente de su presentación, es decir, colores, bordes, márgenes, fondos, layouts, tamaños, fuente, alineación de texto entre otros. Actualmente CSS se encuentra en la versión 3, la cual optimizo varias de sus características de las versiones anteriores y agregó algunas nuevas. Se puede incorporar código CSS a un documento HTML y existen varias formas de lograrlo, así como diferentes librerías y frameworks que ayudan y facilitan la codificación de estilos, ejemplo boostrap.

6.3.10 Boostrap4

Es un framework de CSS con el cual se pueden maquetar estilos de algún sitio web en muy cortos periodos de tiempo, según la definición publicada en su sitio web oficial (Bootstrap, 2019) Bootstrap es un conjunto de herramientas de código abierto para desarrollar con HTML, CSS y JS.

Realice rápidamente un prototipo de sus ideas o construya su aplicación completa con nuestras variables y mixins de Sass, sistema de cuadrícula sensible, extensos componentes previamente compilados y potentes complementos creados en jQuery.

6.3.11 Visual Studio Code.

Dentro de las herramientas más importantes para un desarrollador se encuentran los editores de código, que permite la edición de manera óptima del código, Visual Studio es un conjunto completo que permite crear aplicaciones web y de escritorio de alto rendimiento, que pueden ser utilizadas como componentes de Visual Studio y otras tecnologías para simplificar el diseño, el desarrollo y la implementación de soluciones empresariales basadas en el equipo. Este editor de código maneja también la posibilidad de instalar extensiones que ayudan a implementar diversas funcionalidades como la opción de manejar la versión de código con herramientas como Git Lab.

CAPITULO 3

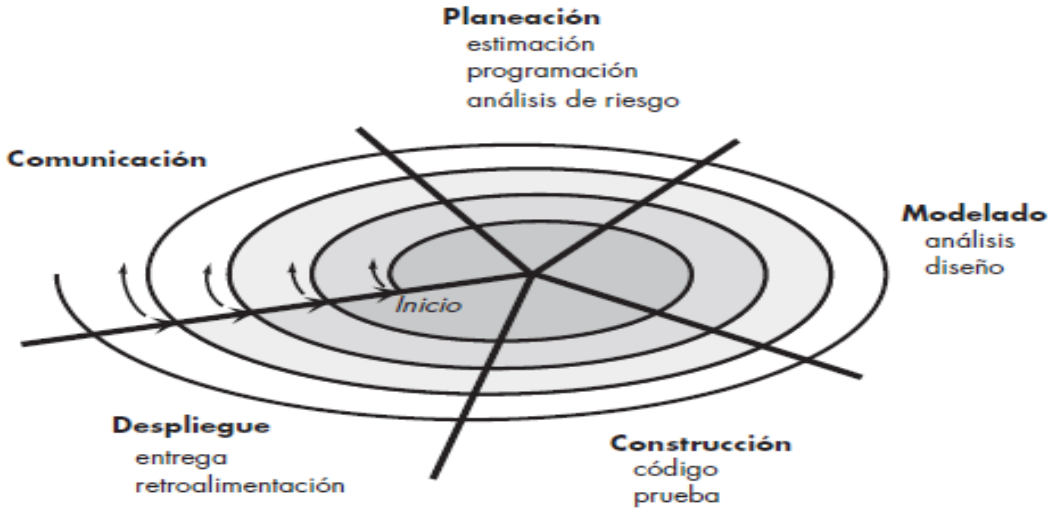
7. METODOLOGÍA.

Con la ejecución de proyectos de ingeniería que requieren del desarrollo de software para hacer una buena planeación y la programación de las actividades, la construcción o codificación del mismo, se hace necesario implementar metodologías para la realización de este tipo de proyectos. En las últimas décadas han surgido autores que han propuesto diferentes tipos de metodologías para intervenir proyectos relacionados con el desarrollo de software.

Uno de los tipos de metodologías más acondicionados a la realidad y con el que mejor se adaptan los partícipes del proyecto, son aquellos basados en modelos de procesos evolutivos. Para abarcar este desarrollo, se escogió una metodología que posee esta cualidad; esto debido a que el software por ser un producto que varía con el tiempo, necesita de metodologías que se adapten al cambio. Es complejo trazar una línea recta con cada una de las fases en que se debe abordar el proyecto, sin tener la necesidad de retroceder y reajustar algunos detalles que no se contemplaron; para ello se ha seleccionado la metodología del modelo espiral, por cumplir con las exigencias del proyecto.

> *"El modelo de desarrollo espiral es un generador de modelo de proceso impulsado por el riesgo, que se usa para guiar la ingeniería concurrente con participantes múltiples de sistemas intensivos en software. Tiene dos características que los distintivas principales. La primera es el enfoque cíclico para crecimiento incremental del grado de definición de un sistema y su implementación, mientras que disminuye su grado de riesgo. La otra es un conjunto de puntos de referencia de anclaje puntual para asegurar el compromiso del participante con soluciones factibles y mutuamente satisfactorias."* (Pressman, 2010)

Figura 8. Modelo de espiral



Fuente: Pressman, R. S. (2010) Modelo de espiral [Imagen]. Recuperado en abril del 2019 del libro *Ingeniería del Software. Un Enfoque Practico.*

En la imagen se pueden observar las etapas del modelo en espiral, junto con el flujo y la dinámica de esta metodología. El conjunto de actividades que definen las fases aplicadas con la metodología de este modelo, las cuales estarán relacionadas y definidas a continuación:

ETAPA 1. COMUNICACIÓN

Se desarrolla con la interacción con el cliente y los usuarios finales, en donde se dan a conocer las necesidades o requisitos que se exigen satisfacer, con el desarrollo del software en cuestión, y así lograr establecer los objetivos que se van a cubrir en el proceso.

Aspectos importantes a tener en cuenta en esta fase de la metodología:

- Establecer un canal de comunicación acorde entre los usuarios finales (cliente) y los desarrolladores o los ingenieros a cargo del proyecto.
- Se debe llegar a un acuerdo de requisitos para el sistema: es importante tener claro este punto entre las dos partes, más adelante será muy complejo incluir un nuevo requerimiento, pues puede afectar el avance de las fases previas.
- Establecer la función o los objetivos a los que se quiere llegar.

- Hacer un levantamiento de información referente al proyecto a realizar.
- Análisis de viabilidad del proyecto.

ETAPA 2. PLANEACIÓN

El objetivo a desarrollar en esta fase de la metodología, abarca todo lo relacionado con la planificación, estimación y análisis de riesgos que se pueden tener en el pleno desarrollo del proyecto. Lo ideal es generar cronogramas de actividades, estimar tiempos, también recursos que sean necesarios para las tareas contempladas y analizar cada contingencia que pueda presentarse.

ETAPA 3. DISEÑO Y MODELADO

En esta parte la metodología se enfoca en plasmar aquellos aspectos de una manera más técnica y sencilla de comprender. Es importante tener en cuenta la elaboración de los siguientes aspectos:

- Modelo de datos.
- Flujo de Navegación, estructura del sistema.
- Modelo de la base de datos.
- Casos de uso.
- Caracterización de la posible interfaz del aplicativo web y de la aplicación.

ETAPA 4. CONSTRUCCIÓN

El objetivo de esta etapa, luego de la obtención de las características más importantes descrita en la fase anterior, es realizar la codificación de los aplicativos utilizando los lenguajes de programación y las herramientas descritas en el marco ingenieril. En esta etapa del proyecto también se contemplan las pruebas de caja blanca y las de caja negra; las pruebas realizadas durante la codificación de las distintas unidades, corresponden a las de caja blanca y las de caja negra son aquellas realizadas cuando se tiene un producto final o un ejecutable correspondiente. Es importante resaltar que, para un óptimo funcionamiento de las aplicaciones, los problemas se dividen en unidades y las pruebas de caja blanca se vuelven importantes, convirtiendo en uno solo que debe ser segmentado en más pequeños y más fáciles de manipular y resolver.

En el desarrollo de la construcción del código se tuvieron en cuenta los siguientes aspectos:

- Construcción de la base de datos.
- Construcción del API (backend).
- Construcción del frontend.
- Construcción de aplicación móvil.

ETAPA 5. DESPLIEGUE

En esta etapa el usuario final podrá disponer de los aplicativos web y móvil para su utilización, se contará con los servicios presentes en la web y con la aplicación disponible en el play store.

8. DESARROLLO DEL PROYECTO.

8.1 DESARROLLO DE LA METODOLOGÍA.

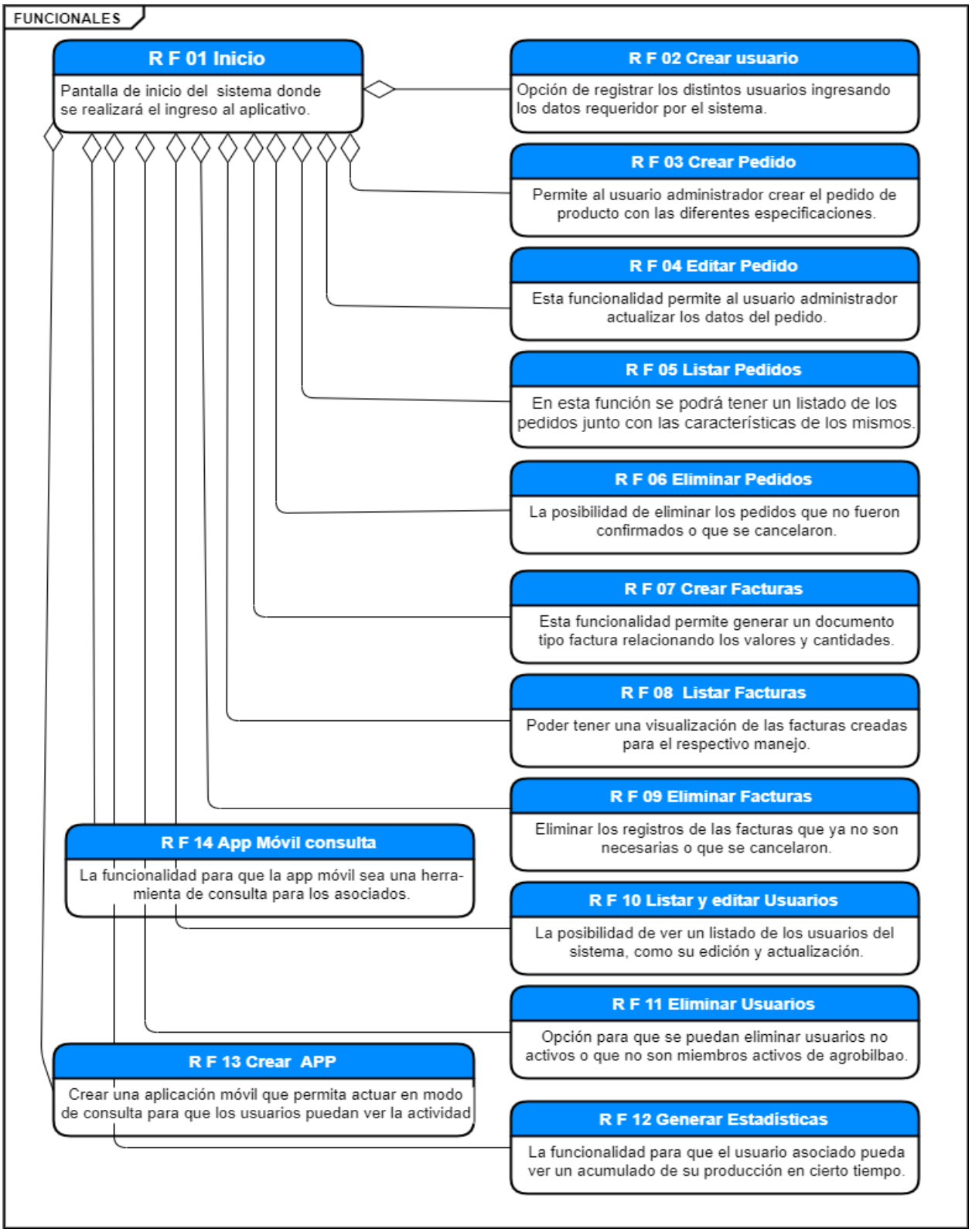
8.1.1 ETAPA 1. COMUNICACIÓN

En la fase de comunicación, se realizó el levantamiento de información necesario para el desarrollo del sistema, con la interacción de personal de la asociación y miembros de la misma. Se identificaron las necesidades y falencias que presentan y a continuación, se relacionan los requisitos funcionales y no funcionales:

Requisitos funcionales y no funcionales:

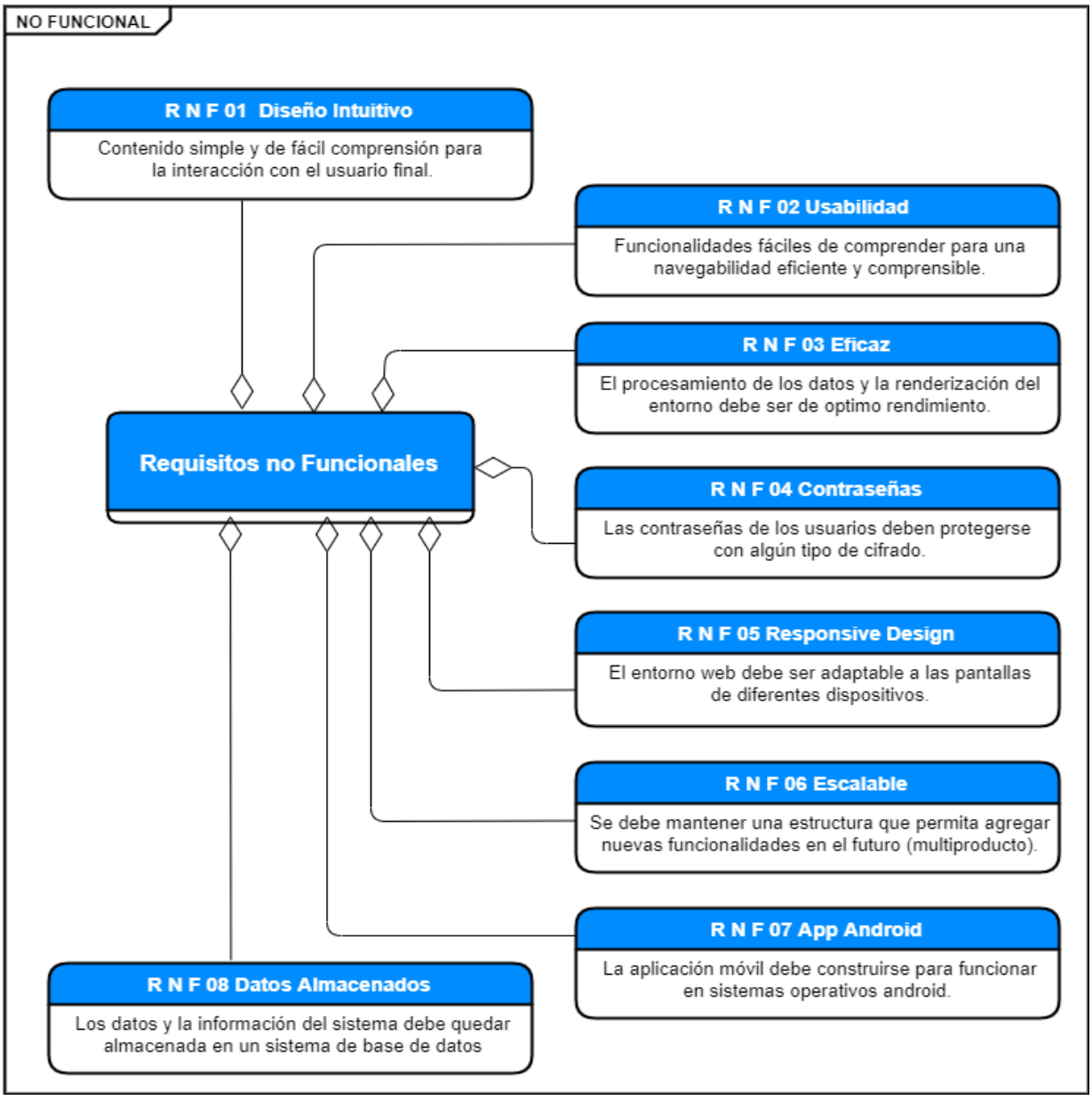
En la Figura 9 se listan las diferentes actividades que aplican como requisitos funcionales, con las necesidades y especificaciones que se requieren para satisfacer las exigencias del cliente. Posteriormente en la Figura 10, se pueden observar aquellos requisitos que no afectan directamente el funcionamiento del sistema, pero que son requeridos por el cliente para un mejor uso, así como los usuarios finales para el manejo de datos.

Figura 9. Diagrama de requisitos funcionales.



Fuente: Elaboración propia.

Figura 10. Diagrama de requisitos no funcionales.



Fuente: Elaboración propia.

8.1.2 ETAPA 2. PLANEACIÓN

Durante la etapa de comunicación donde se realizó el levantamiento de información, fue necesario hacer una encuesta, cuya información y resultados se pueden consultar en los anexos. En este ejercicio se evidenciaron algunos puntos importantes y variables que permitieron tomar decisiones, para así planificar la manera más óptima el cómo abordar el desarrollo del software

para este proyecto. La población completa de miembros en la asociación AgroBilbao es de aproximadamente 150 personas, para el ejercicio de recolección de información se tomó una muestra poblacional de 34, que representa el 22% de personal apto para el ejercicio.

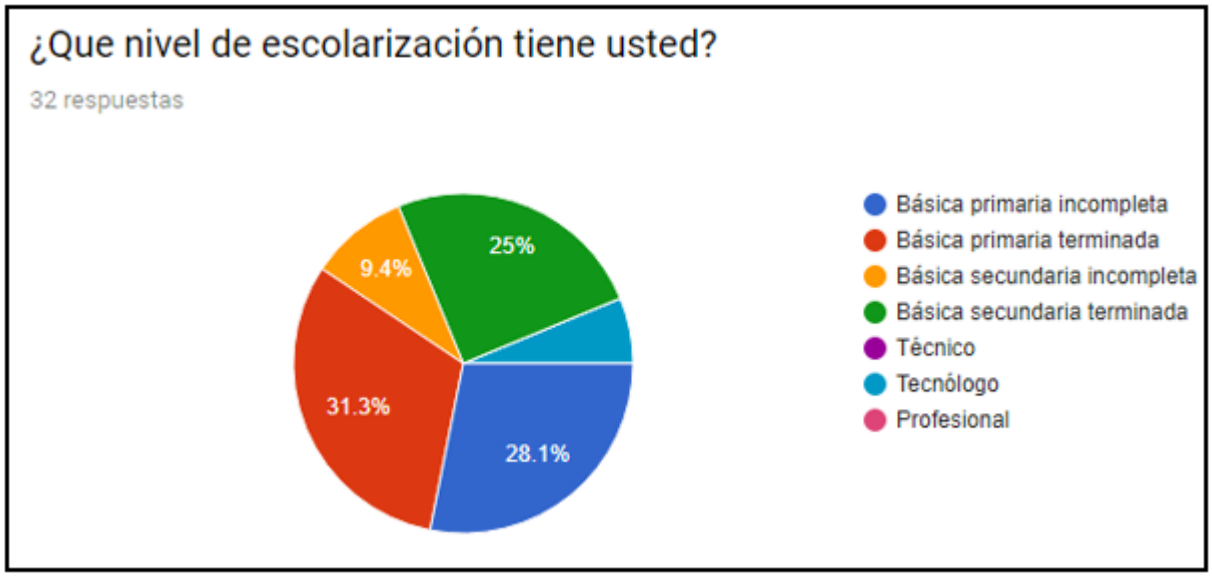
Con los resultados obtenidos en la encuesta, se pudo hacer el análisis y la planificación de algunas actividades para el desarrollo del proyecto, donde se contempló como importantes los siguientes puntos:

- Ante la pregunta ¿Posee y utiliza computador ya sea portátil o de escritorio?, se obtuvo una respuesta negativa en un 81.8%, lo cual permitió decidir el crear una aplicación web con característica adaptables a dispositivos móviles (Responsive Web Design) y que sirva para que el usuario administrador lo pueda trabajar desde un equipo de cómputo.

- Se decidió también crear una aplicación móvil (App) en Android; pues según los resultados el 73,5% de los encuestados poseen un teléfono inteligente (smartphone), y el 100% de esos dispositivos, trabajan sobre este sistema operativo. Esta App no tendrá las mismas funcionalidades que posee la aplicación web, y por el momento, aportará características de consulta de la información para que los datos se encuentren disponible en este entorno.

- El diseño de las aplicaciones tanto web como móvil debe ser muy intuitivo y de fácil funcionamiento y navegabilidad, puesto que la población final a la que va dirigido en un 68.8% según los resultados de la encuesta tiene un nivel de escolaridad entre ningún estudio y básica secundaria incompleta, lo que hizo que se construyera un sistema de fácil interpretación por el usuario. En el siguiente grafico se observa los resultados del nivel de escolarización de los encuestados:

Figura 11. Diagrama de resultados para el nivel de escolaridad.



Fuente: Elaboración propia.

8.1.3 ETAPA 3. DISEÑO Y MODELADO

MODELO DE DATOS

En la investigación para el desarrollo y codificación del software se tuvo el dilema de escoger una arquitectura de software que permita tener una organización adecuada del código. Dentro de los clásicos patrones de diseño utilizados, se encuentra el MVC Modelo-Vista-Controlador; en este marco la primera parte el modelo es el encargado de representar los datos de la aplicación y el acceso a ellos, la vista es la encargada de presentar al usuario los datos entregados por el modelo y el controlador contiene la lógica de negocio (lo que debe hacer el programa).

Inicialmente los desarrollos de aplicaciones web trazaban un modelo donde la mayoría de funciones de la vista, el modelo y el controlador recaían en el servidor. La eficiencia de estos sistemas se percibía como lenta por parte de los clientes y con poca dinámica; la evolución de esos enfoques llevó a irle asignando cada vez más funciones al navegador y liberar la carga al servidor. El modelo MVC permitió optimizar estos procesos, pero se sigue viendo el crecimiento en el tema.

La evolución del desarrollo de software trae un nuevo patrón de diseño el SPA (Single-Page Applications). Las aplicaciones de una sola página (SPA) son aplicaciones web que cargan

una sola página HTML y la actualizan dinámicamente a medida que el usuario interactúa con la aplicación; este modelo es escogido para el desarrollo de este proyecto. Los SPA utilizan AJAX (ahora en la evolución de JavaScript fetch que cumple la misma funcionalidad de Ajax) y HTML5 para crear aplicaciones web fluidas y receptivas, sin recargas constantes de página. Sin embargo, esto significa que gran parte del trabajo ocurre en el lado del cliente, en JavaScript. Para el desarrollador tradicional, puede ser difícil dar el salto. Afortunadamente, hay muchos marcos de JavaScript de código abierto que facilitan la creación de SPA. En este proyecto se utilizaron marcos de desarrollo como React JS.

EL patrón SPA desacopla el desarrollo web (HTML, CSS y JavaScript) conocido en la programación como front-end de la parte lógica y de interacción con la base de datos conocida como back-end, el cual se encarga de gestionar los datos y la lógica de negocio. Se ejecuta en el servidor y expone al front una API (Application Programming Interface). El front-end constituye la interfaz de usuario; es la encargada de todo el dinamismo y la interacción, ejecutándose típicamente en el navegador. Se comunica con la API mediante el intercambio de ficheros ligeros de datos (habitualmente JSON). (Microsoft, 2013).

Algunas de las ventajas al aplicar SPA y desacoplar sus partes son las siguientes:

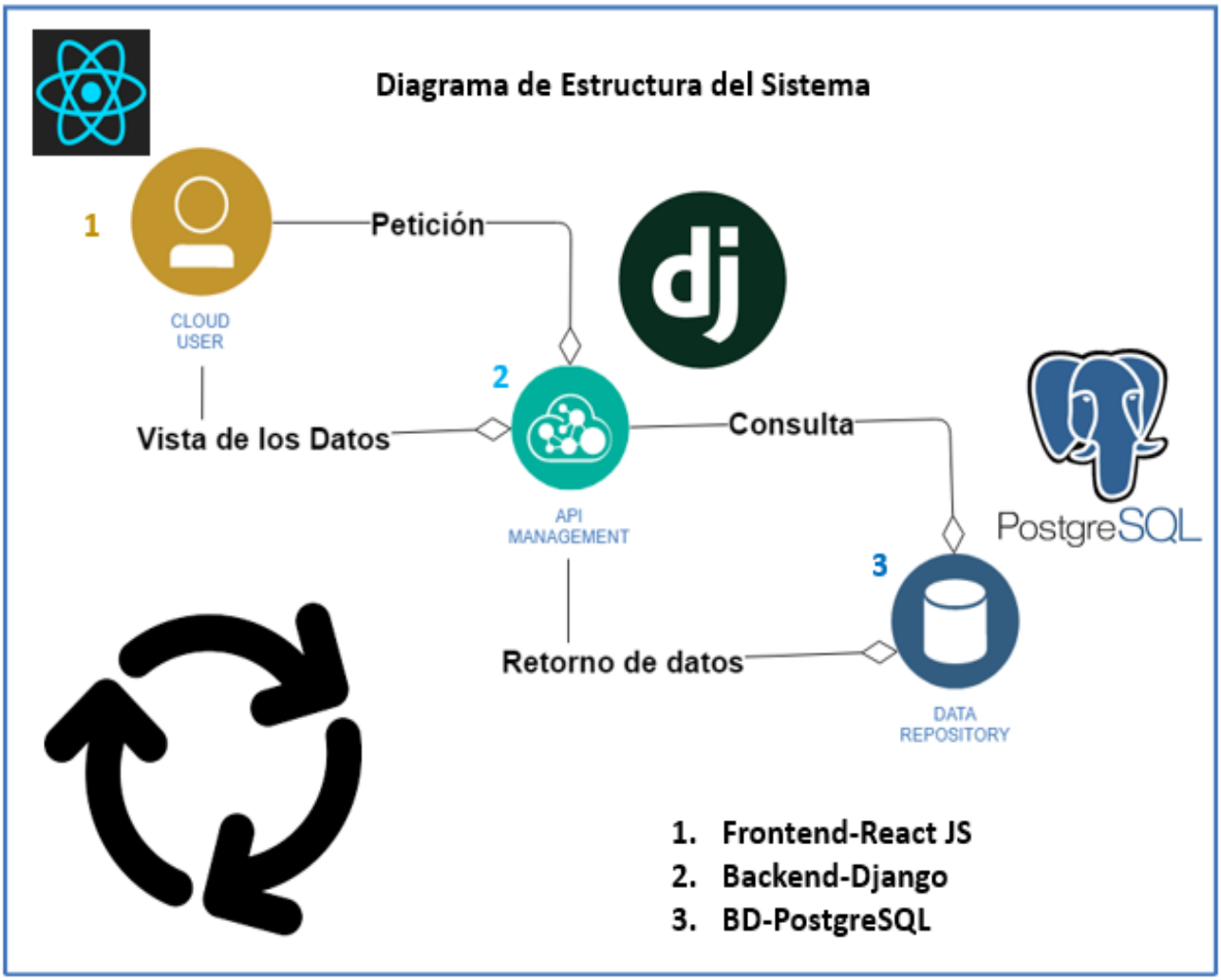
- Tener mayor eficiencia o rapidez que mejore la interactividad entre las interfaces y el procesamiento de la información presente.
- El back-end que para el caso de SPA donde se almacena toda la lógica del negocio puede ser utilizada por distintas interfaces de usuario. Los datos se exponen como una API y brinda sus servicios a múltiples usuarios.
- La interacción de información se hace con el formato JSON (JavaScript-Object-Notation) lo que lo hace muy ligero para las distintas peticiones y mejora los tiempos de respuesta entre cliente y servidor.

- Permiten el almacenamiento de datos en memoria cache del navegador, a los que se hacen peticiones previas y luego no tener que recargar nuevamente el sitio y poder acceder a esta información.

FLUJO DE NAVEGACIÓN Y ESTRUCTURA

A continuación, el diagrama de estructura del sistema:

Figura 12. Diagrama de estructura del sistema.

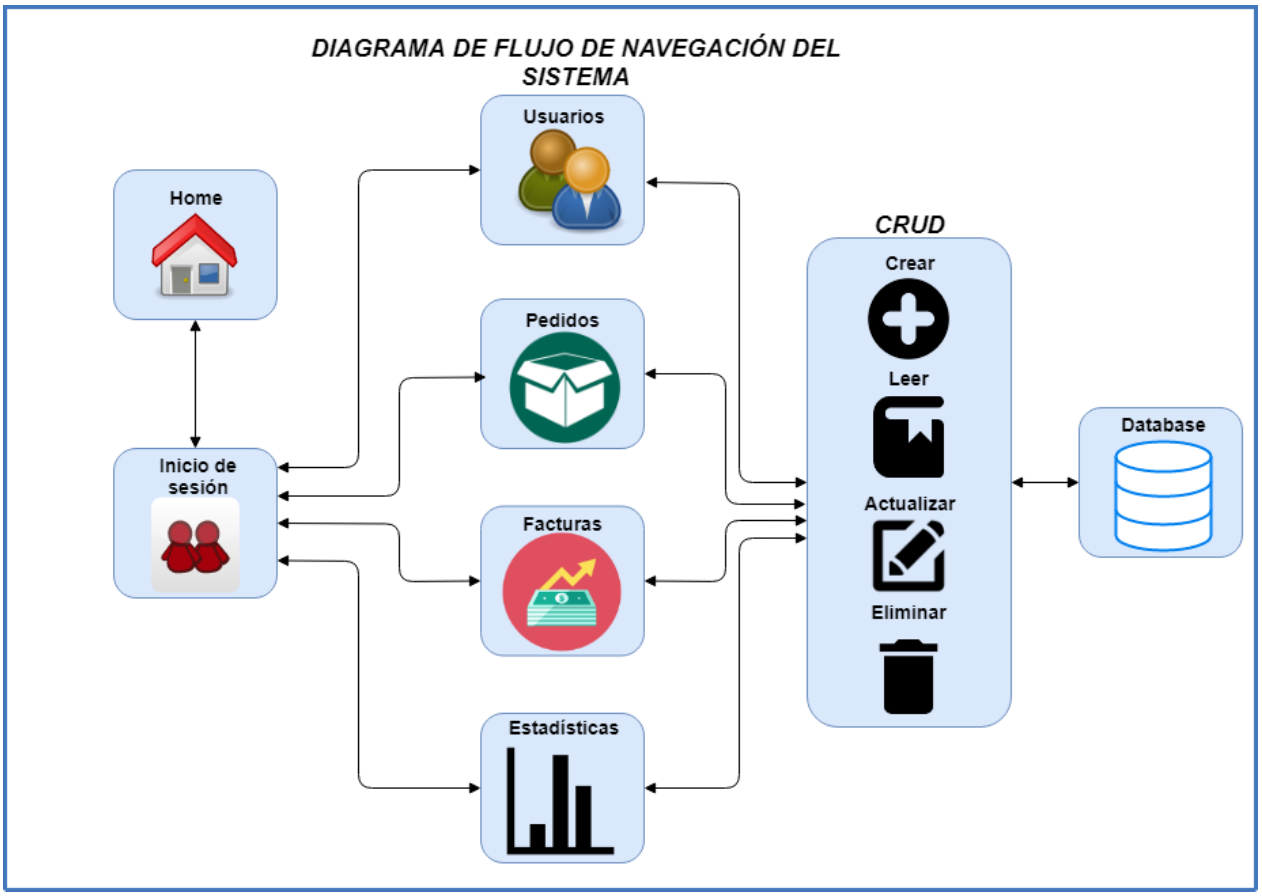


Fuente: Elaboración propia.

En la Figura 12 se observa la interacción del sistema, iniciando de una petición por parte del usuario desde nuestra capa presentación el frontend desarrollado en React JS, dicha petición es recibida por el backend construido con el framework Django adoptando el comportamiento de una API encargada de generar el intercambio informacional con la base de datos, este comportamiento con esa estructura conforma la naturaleza de una Single-Page Application (SPA).

En la figura 13 se puede apreciar el flujo de navegación del sistema y la interacción de cada uno de los módulos, iniciando con el menú principal, posteriormente generando un inicio de sesión para tener acceso a los distintos módulos y en ellos poder ejecutar acciones como crear, consultar, actualizar y eliminar, también conocidas como métodos CRUD que son almacenados en la base datos.

Figura 13. Diagrama de flujo de navegación del sistema



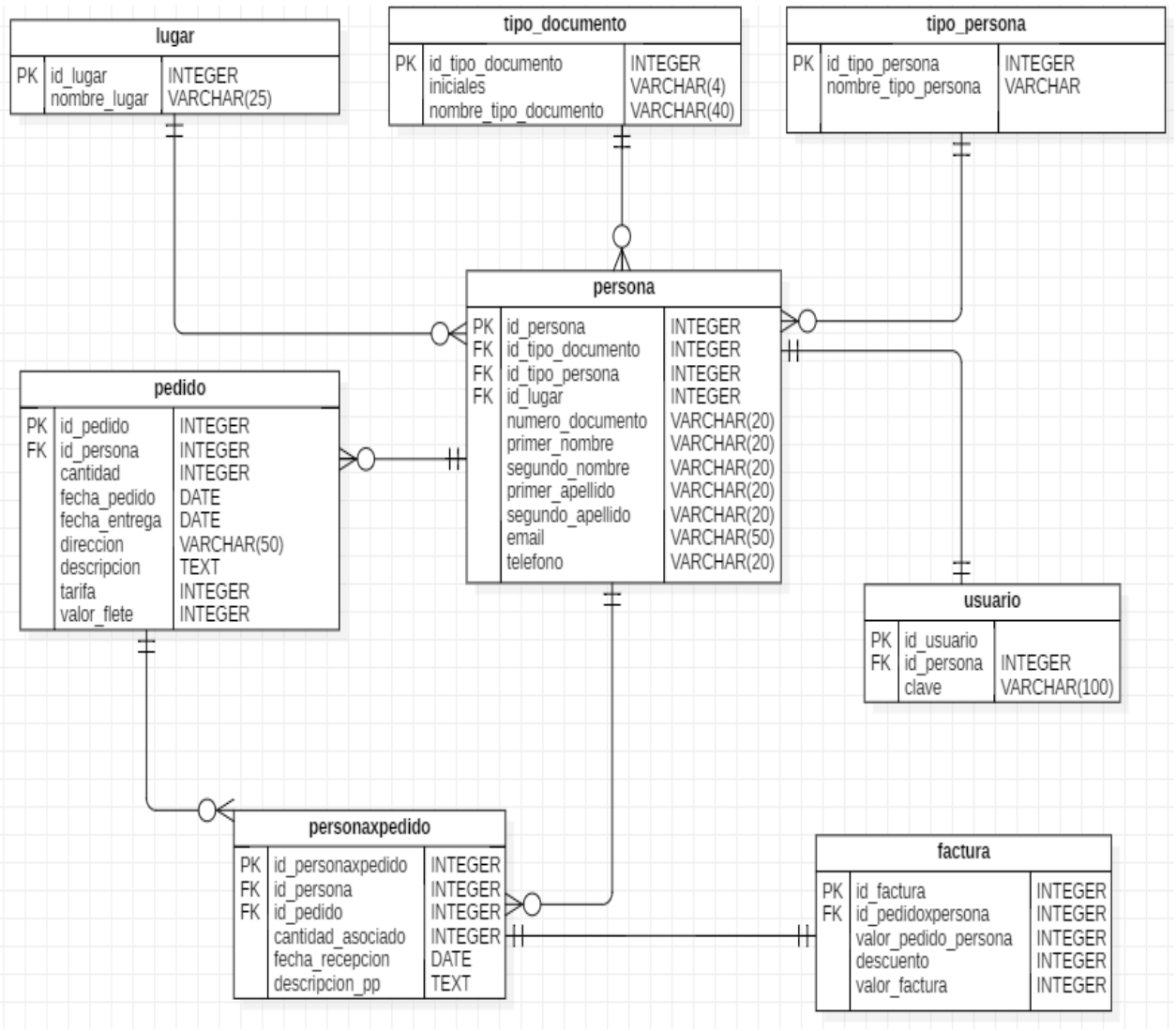
Fuente: Elaboración propia.

MODELO DE LA BASE DE DATOS

La construcción del modelo de la base de datos requirió un proceso que se describe en cuatro pasos. Primero contar con unos requerimientos claros, los cuales permitieron el segundo paso de tener las entidades y sus respectivas relaciones claras, tercero realizar una tabulación del diseño lógico para la normalización de la base de datos, en cuarto y último lugar se definen los tipos de datos y la decisión de utilizar PostgreSQL como motor de base de datos.

A continuación, el diagrama entidad relación resultante:

Figura 14. Modelo entidad relación de la base de datos.



Fuente: Elaboración propia.

CASOS DE USO

Los casos de uso se realizaron teniendo en cuenta los requisitos funcionales influyentes y relevantes para el aplicativo. Se realizó la construcción en el formato extendido para la descripción de los diferentes casos de uso; en las tablas se muestra el nombre del caso de uso, los actores que intervienen, se hace una descripción del caso de uso, la información que sirve como entrada y salida, las condiciones previas o posteriores para cada caso y finalmente los casos de uso próximos que usan o extienden el actual.

A continuación, se muestran los casos de uso elaborados:

Tabla 1. Caso de uso de Inicio de sesión.

Nombre Caso de Uso: Inicio de Sesión CU-01	
Actor (es):	Usuario, Sistema, Base de datos.
Descripción:	Es el formulario de inicio del sistema, en el cual se presenta la opción de inicio de sesión para usuarios.
Entradas:	Datos para inicio de sesión (correo electrónico y contraseña).
Salidas:	La confirmación e ingreso de sesión exitoso para el usuario.
Precondiciones:	El usuario debe estar creado en la base de datos y debe existir el formulario de inicio.
Postcondiciones:	El usuario tendrá la siguiente interfaz de su perfil.
Precedentes:	Ninguno.
Usan o extienden:	Crear Usuario CU-02, Mostrar usuario, modificar usuario.

Fuente: Elaboración propia.

Tabla 2. Caso de uso crear usuario

Nombre Caso de Uso: Crear Usuario CU-02	
Actor (es):	Usuario administrador, Sistema, Base de datos.
Descripción:	En esta opción se pueden crear o registrar los distintos usuarios digitando los datos requeridos por el sistema.
Entradas:	Datos personales del nuevo usuario.
Salidas:	Usuario creado satisfactoriamente.
Precondiciones:	Tipo persona, tipo persona, lugar previamente creados y Base de datos funcionando.
Postcondiciones:	Registro de usuario quede correctamente almacenado en la base de datos.
Precedentes:	Caso de uso inicio de sesión. Crear usuario Administrador. Base de datos funcionando.
Usan o extienden:	Inicio de Sesión CU-01 Mostrar usuario, Consultar, Modificar o actualizar usuario.

Fuente: Elaboración propia.

Tabla 3. Caso de uso crear lugar

Nombre Caso de Uso: Crear Lugar CU-03	
Actor (es):	Usuario administrador, Sistema, Base de datos.
Descripción:	Permite al usuario administrador del sistema crear los distintos lugares, para posteriormente asignarlos a los usuarios comunes.
Entradas:	Código para el lugar y nombre del lugar.
Salidas:	Lugar registrado en Base de Datos.
Precondiciones:	Base de Datos en funcionamiento, el lugar a crear no debe existir.
Postcondiciones:	El registro del lugar queda correctamente almacenado en la base de datos.
Precedentes:	Base de Datos creada. Crear Usuario CU-02
Usan o extienden:	Crear Usuario CU-02, Modificar, eliminar o listar lugar.

Fuente: Elaboración propia.

Tabla 4. Caso de uso crear pedido. Fuente elaboración propia

Nombre Caso de Uso: Crear Pedido CU-04	
Actor (es):	Usuario administrador, Sistema, Base de datos.
Descripción:	Permite al usuario administrador crear el pedido realizado junto con sus características para posterior confirmación de disponibilidad de producto del usuario asociado o agricultor
Entradas:	datos del usuario creador, cantidad, fechas, dirección, descripción, tarifa y los valores del flete.
Salidas:	Pedido creado y listo para editarse, listarse o eliminarse.
Precondiciones:	Base de datos en funcionamiento, el pedido no debe existir en base de datos, debe tener un id de persona asociado.
Postcondiciones:	Pedido almacenado en base de datos, debe poder buscarse y listarse por parte de los usuarios.
Precedentes:	Base de Datos creada. Crear Usuario CU-02
Usan o extienden:	Actualización de Pedido CU-05

Fuente: Elaboración propia.

Tabla 5. Caso de uso actualización de pedido

Nombre Caso de Uso: Actualización de Pedido CU-05	
Actor (es):	Usuario, Sistema, Base de datos.
Descripción:	Permite a los usuarios tanto administrador como asociado ver el estado del pedido y colocar la cantidad de fruta aportada. El usuario administrador podrá cambiar las características del pedido y actualizarlo en la base de datos
Entradas:	Datos del pedido a editar.
Salidas:	Pedido actualizado con datos nuevos y listo para volver a editarse, listarse o eliminarse.
Precondiciones:	Pedido creado y almacenado.
Postcondiciones:	Pedido correctamente almacenado en base de datos.
Precedentes:	Base de Datos creada. Crear Usuario CU-02. Crear Pedido CU-04
Usan o extienden:	Crear Persona a Pedido CU-6, Crear Pedido CU-04

Fuente: Elaboración propia.

Tabla 6. Caso de uso crear persona a pedido

Nombre Caso de Uso: Crear Persona a Pedido CU-6	
Actor (es):	Usuario administrador, Sistema, Base de datos.
Descripción:	Permite al usuario administrador asignar la cantidad aportada por cada asociado y gestionar qué integrantes participaran en la despachada del pedido creado
Entradas:	Id del asociado a aportar, Id del pedido y los datos del aporte de producto, fecha y su descripción.
Salidas:	Cantidad asociada de fruta a cada integrante en el pedido creado.
Precondiciones:	El pedido esté creado, el usuario asociado esté creado.
Postcondiciones:	Registro exitoso en base de datos del cuerpo del pedido con las cantidades aportadas de cada agricultor.
Precedentes:	Base de Datos creada. Pedidos y asociados creados.
Usan o extienden:	Crear Usuario CU-02. Crear Pedido CU-04

Fuente: Elaboración propia.

Tabla 7. Caso de uso actualización persona a pedido

Nombre Caso de Uso: Actualización Persona a Pedido CU-07	
Actor (es):	Usuario administrador, Sistema, Base de datos.
Descripción:	Permite al usuario administrador editar la cantidad aportada de producto a cada asociado, ya sea para aumentarla o disminuirla en pro de completar con la cantidad requerida en el pedido
Entradas:	Id del asociado a editar Id del pedido a editar y los datos de actualización del aporte de producto, fecha y su descripción.
Salidas:	Pedido asociado a personas actualizado.
Precondiciones:	Persona a pedido previamente creada y almacenado en base de datos.
Postcondiciones:	Registro actualizado del pedido asociado a cada contribuidor.
Precedentes:	El pedido creado y asociado a cada contribuidor junto con los datos requeridos para ello.
Usan o extienden:	Crear Persona a Pedido CU-6, Crear Pedido CU-04, Eliminación Persona a Pedido CU-08

Fuente: Elaboración propia.

Tabla 8. Caso de uso eliminar persona a pedido.

Nombre Caso de Uso: Eliminación Persona a Pedido CU-08	
Actor (es):	Usuario administrador, Sistema, Base de datos.
Descripción:	La funcionalidad para que el usuario administrador tenga la capacidad de cancelar un participante asociado en el pedido a contribuir y poder eliminar la cantidad de fruta que inicialmente iba a aportar.
Entradas:	Id del asociado a eliminar, Id del pedido a editar.
Salidas:	Registro del pedido por persona eliminado de forma exitosa.
Precondiciones:	El pedido creado previamente y asociado al contribuidor a cancelar.
Postcondiciones:	Registro en base de datos actualizado, disponible para ser editado nuevamente.
Precedentes:	Estar creado previamente el pedido asociado por persona
Usan o extienden:	Actualización Persona a Pedido CU-07, Crear Persona a Pedido CU-6.

Fuente: Elaboración propia.

Tabla 9. Caso de uso crear factura.

Nombre Caso de Uso: Crear Factura CU-09	
Actor (es):	Usuario administrador, Sistema, Base de datos.
Descripción:	El sistema a medida que se ingresan los datos de los pedidos y de la contribución de fruta que cada asociado aportará irá realizando un registro tipo factura que servirá para procesos contables más adelante. Permitiéndole al usuario ver sus estados de cuenta.
Entradas:	Id de persona por pedido, valores, descuentos y valor final de factura.
Salidas:	Registro de la factura, disponible para su consulta.
Precondiciones:	Debe existir un pedido creado, junto con las contribuciones de cada asociado en dicho pedido.
Postcondiciones:	Factura creada y almacenada en base de datos disponible para su consulta y visualización.
Precedentes:	Pedido creado y asociación de pedido por persona realizada.
Usan o extienden:	Eliminación Persona a Pedido CU-08, Actualización Persona a Pedido CU-07, Crear Persona a Pedido CU-6, Crear Pedido CU-04.

Fuente: Elaboración propia.

Tabla 10. Caso de uso generación de estadísticas.

Nombre Caso de Uso: Generación de Estadísticas CU-10	
Actor (es):	Usuario, Sistema, Base de datos.
Descripción:	El sistema le permitirá al usuario poder generar un reporte estadístico de a través de gráficos de los aportes de fruta realizados en intervalos de tiempo.
Entradas:	Datos del usuario asociado. Intervalos de tiempo.
Salidas:	Reporte estadístico de aportes a la asociación.
Precondiciones:	Usuario creado con pedidos asociados y contribuciones hechas.
Postcondiciones:	Visualización de información para el usuario.
Precedentes:	Pedidos en los que contribuyó y facturas asociadas.
Usan o extienden:	Crear Persona a Pedido CU-6, Crear Pedido CU-04, Inicio de Sesión CU-01, Crear Usuario CU-02, Crear Factura CU-09.

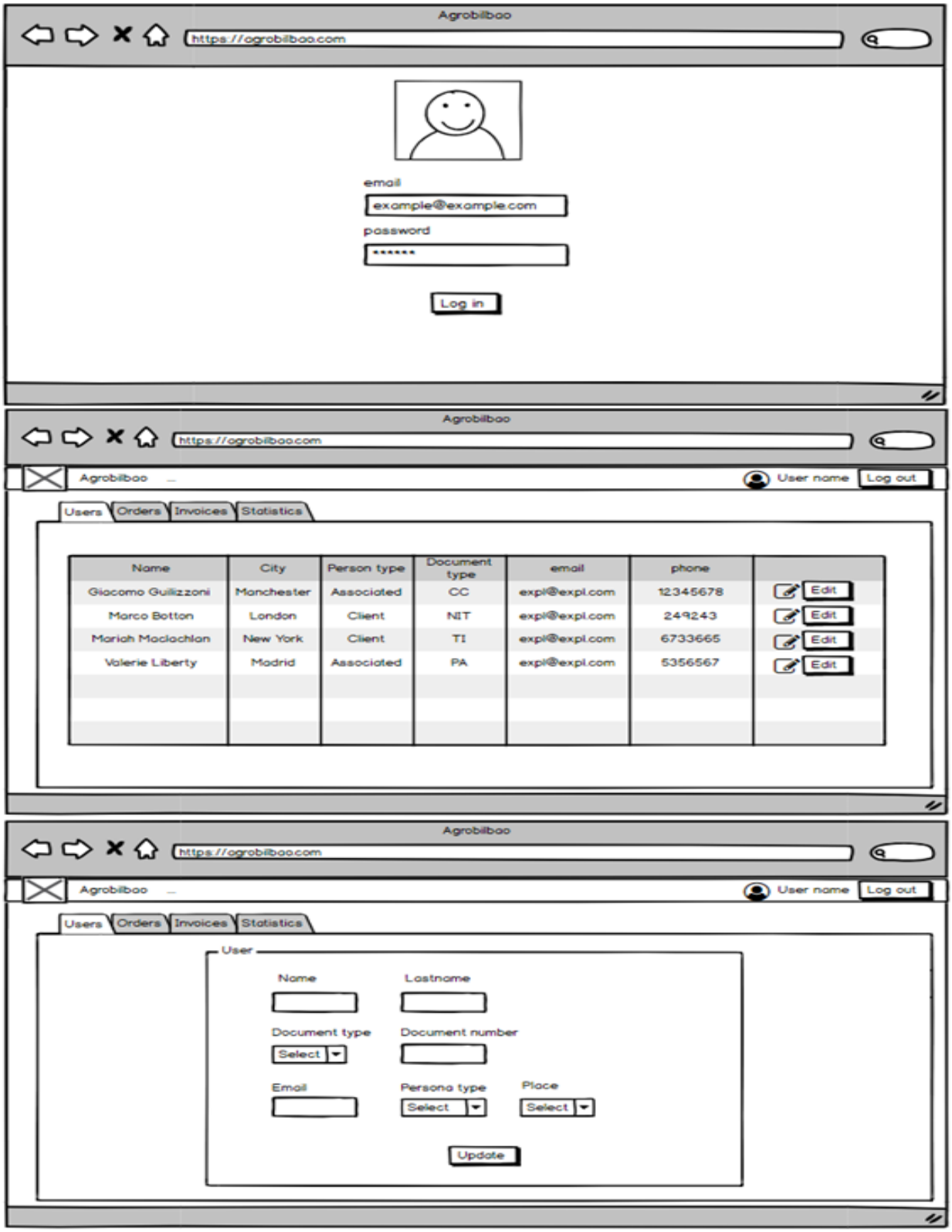
Fuente: Elaboración propia.

MODELO INTERFAZ DEL APLICATIVO WEB Y DE LA APLICACIÓN MÓVIL.

Para la construcción y diseños de las vistas de la aplicación web y App móvil se elaboraron unos bocetos conocidos en el área de diseño como mockups, los cuales consisten en plasmar un modelo del posible diseño de la interfaz gráfica de usuario que dispondrá el cliente. Estos diseños pueden estar sujetos a cambios en el resultado final puesto que son una apreciación inicial de la posible visualización del sistema en su culminación.

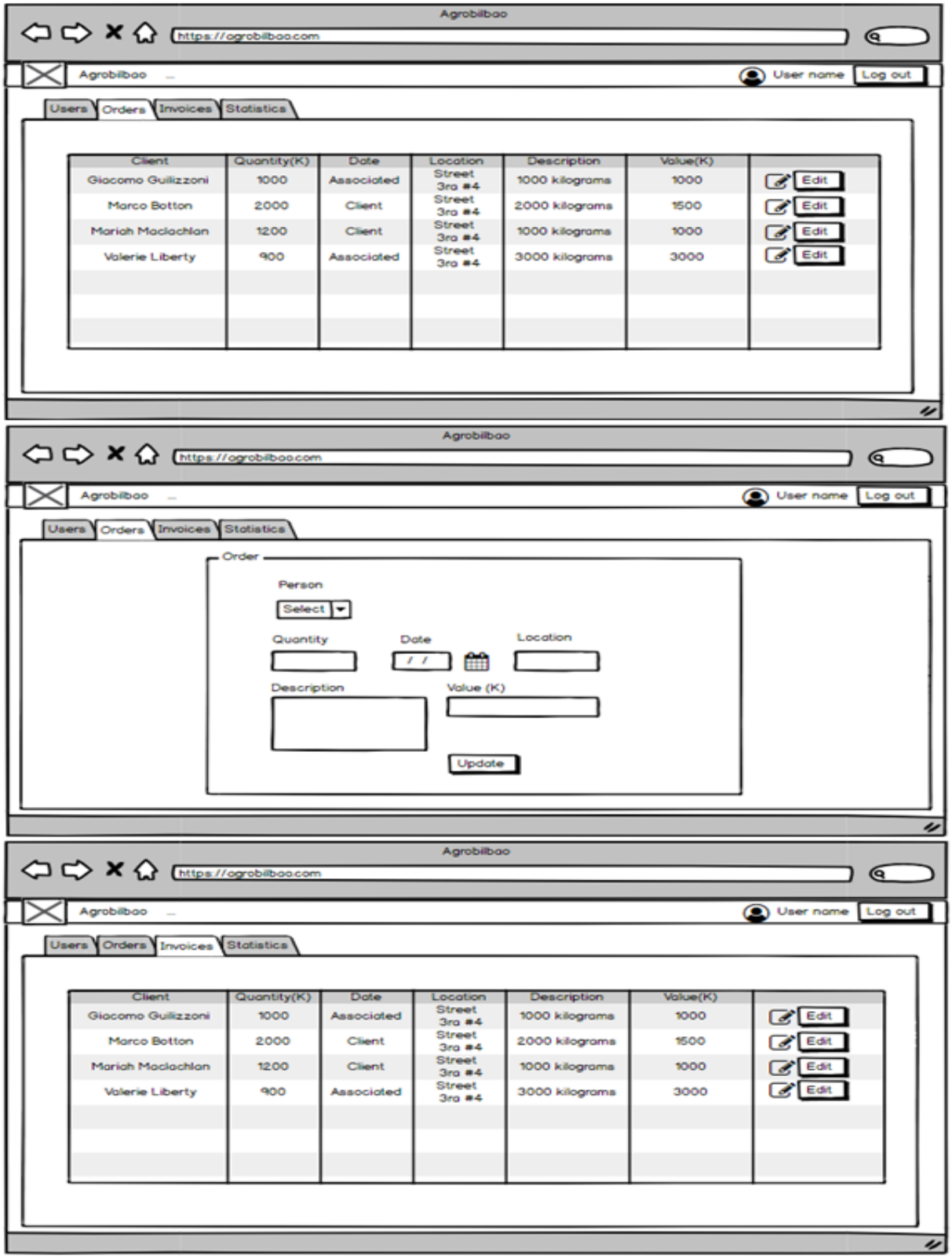
A continuación, en las figuras 15, 16 y 17 se evidencian los diseños efectuados para las interfaces de la aplicación web:

Figura 15. Modelo de interfaz de inicio de sesión, usuarios y editar usuarios.



Fuente: Elaboración propia.

Figura 16. Modelo de interfaz de pedidos, editar pedidos y facturas.



Fuente: Elaboración propia.

Figura 17. Modelo de interfaz de editar facturas y de las estadísticas.

Fuente: Elaboración propia.

A continuación, en las figuras 18 y 19 se muestran los diseños realizados para las interfaces de usuario de la aplicación móvil del proyecto:

Figura 18. Modelo de interfaz Inicio de sesión, usuarios y pedidos de la app.

Fuente: Elaboración propia.

Figura 19. Modelo de interfaz Facturas y estadísticas de la app.

Fuente: Elaboración propia.

8.1.4 ETAPA 4. CONSTRUCCIÓN

Como ya anteriormente se había mencionado se utilizó Single-Page Application (SPA) como el patrón de diseño de software para este proyecto; el cual permite una mayor seguridad y escalabilidad del proyecto. Para la descripción del desarrollo y codificación del software se tienen cuatro secciones importantes, Frontend, Backend, Base de Datos y la App móvil las cuales serán relacionadas a continuación:

FRONTEND

La parte gráfica, vista o frontend del proyecto como se conoce en el desarrollo de software, incluye toda la lógica y construcción de los componentes para el despliegue de la información de manera gráfica, junto con la interacción hacia el cliente generando una agradable experiencia de usuario. Esta capa es ejecutada por el navegador web ya sea Google Chrome, Mozilla, Firefox, Opera, Safari u otro, y para su realización se utilizó como herramienta React JS. La filosofía de React JS está basada en componentes y en su reutilización; como ejemplo hay que resaltar la construcción de nuestro módulo de registro de usuarios, cuyo componente se utiliza en el registro del usuario de manera independiente por asociado o ya sea que el usuario administrador lo registre.

En la figura 20, se puede observar la codificación del método (saveData) encargado de registrar los usuarios, en el cual se captura la información que el usuario está digitando en el formulario, se asigna a una variable de tipo JSON (JavaScript Object Notation) y posteriormente se realiza una petición de tipo POST al backend de nuestra aplicación para culminar el proceso.

En la figura 21, se tiene el método render del formulario de registro, el cual es el encargado de mostrar (renderizar) en el navegador la interfaz construida, y posteriormente en las demás figuras se puede observar algunos ejemplos del resultado de esas interfaces:

Figura 20. Método para registrar usuario.

```
    saveData() {
        if (this.form.current.reportValidity()) {
            this.setState({
                loading: true,
            });

            let postData = {
                numero_documento: this.form.current.numero_documento.value,
                primer_nombre: this.form.current.primer_nombre.value,
                segundo_nombre: this.form.current.segundo_nombre.value,
                primer_apellido: this.form.current.primer_apellido.value,
                segundo_apellido: this.form.current.segundo_apellido.value,
                email: this.form.current.email.value,
                telefono: this.form.current.telefono.value,
                id_tipo_documento: this.form.current.id_tipo_documento.value,
                id_tipo_persona: this.form.current.id_tipo_persona.value,
                id_lugar: this.form.current.id_lugar.value,
            };

            console.log(JSON.stringify(postData));

            axios.post(URL_API + "/person", postData, this.headers
            ).then(res => {
                if (res.status === 201) {
                    // console.log("res: " + JSON.stringify(res.data));
                    this.setState({
                        loading: false,
                    });
                    this.props.onDataSaved();
                }
            }).catch(error => {
                console.log(error);
            });
        }
    }
```

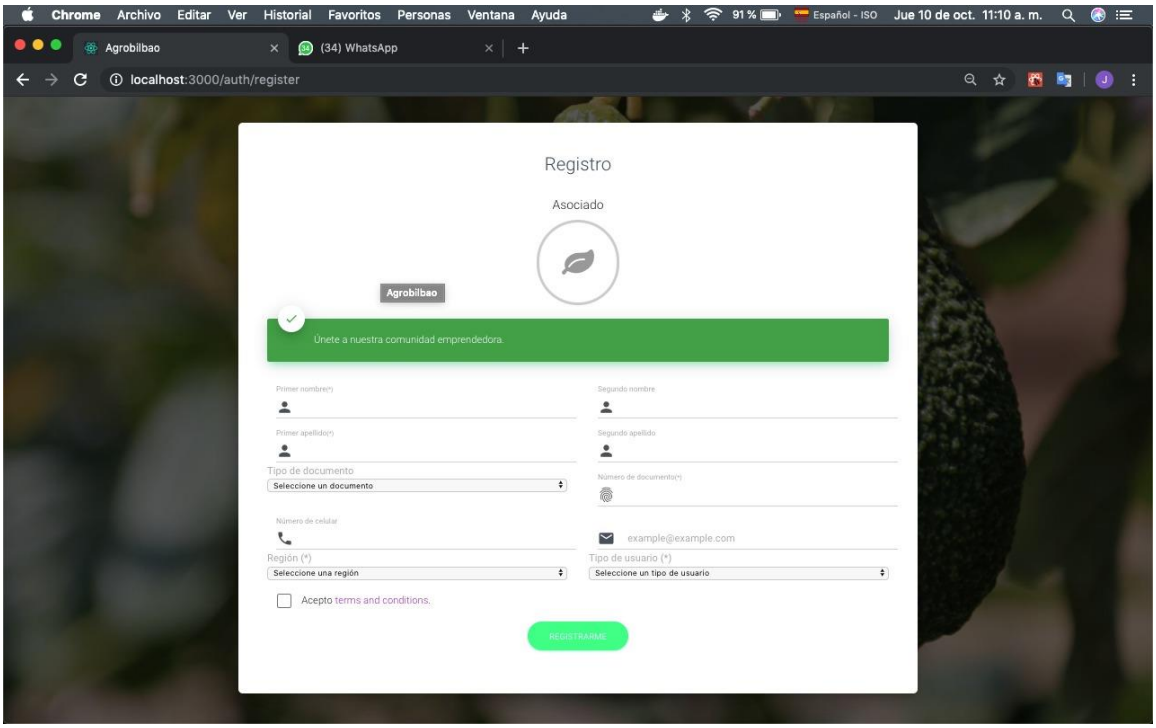
Fuente: Elaboración propia.

Figura 21. Método render del componente de registro de usuario.

```
    render() {
        const { classes } = this.props;
        return (
            <div>
                {this.state.alert}
                <form className={classes.form} ref={this.form} onSubmit={e => e.preventDefault()}>
                    <GridContainer>···
                    <Checkbox
                        checkedIcon={<Check className={classes.checkedIcon} />}
                        icon={<Check className={classes.uncheckedIcon} />}
                        classes={{
                            checked: classes.checked,
                            root: classes.checkRoot
                        }}
                        inputProps={{
                            name: "checkTerms",
                            required: true
                        }}
                    />
                    <span>Acepto <a href="#pablo">terms and conditions</a>.</span>
                    <div className={classes.center}>···
                </form>
            </div>
        );
    }
}

export default withStyles(registerPageStyle)(RPlayer);
```

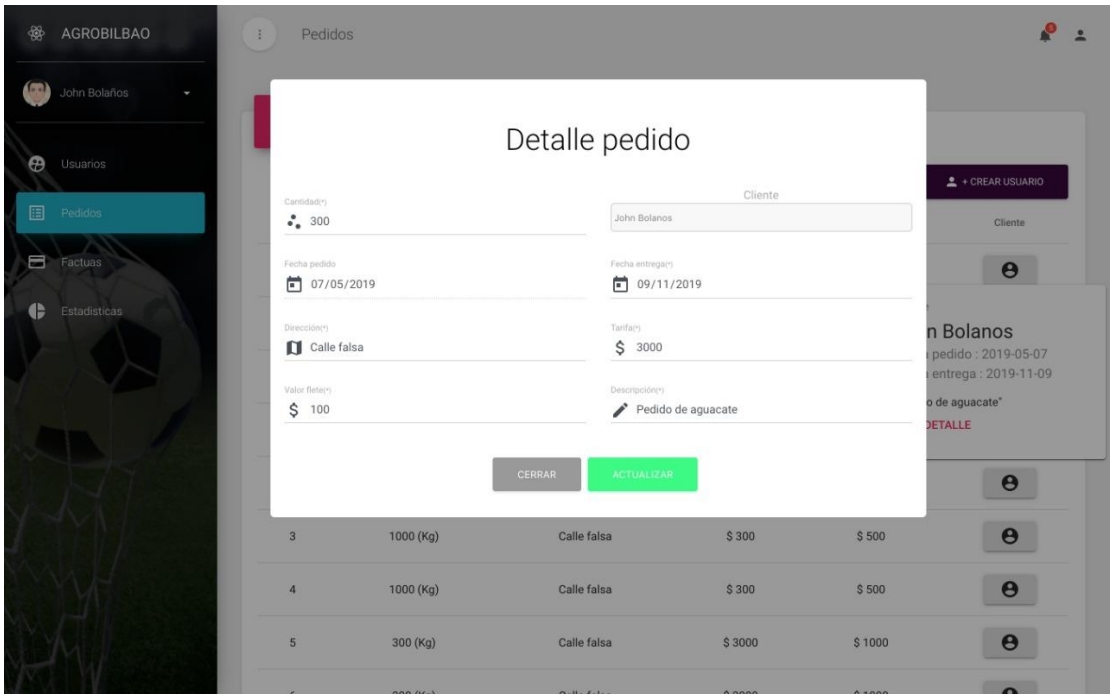
Fuente: Elaboración propia.

Figura 22. Interfaz de registro de usuarios.



Fuente: Elaboración propia.

Figura 23. Interfaz de lista y detalle de pedidos.



Fuente: Elaboración propia.

BACKEND

Para la capa del backend, se trabajó en el desarrollo de un API (Application Programming Interface), definida como una aplicación del lado del backend que va a tener una serie de rutas configuradas, métodos configurados y lógica de control que permite hacer una interacción con la base de datos y que podrá ser consumida con el protocolo HTTP por cualquier tipo de cliente ya sea un cliente presente en un navegador web o desde una aplicación móvil, es importante hacer énfasis en esta característica ya que para aplicación móvil permite consumir este mismo servicio y destacar lo útil de utilizar la lógica separada entre el backend y el frontend propuesta en el patrón de diseño SPA

Esta mencionada API está desarrollada con Django Rest Framework, el cual permite una codificación eficiente de todos los métodos presentes con el protocolo HTTP. Los elementos importantes para el desarrollo de la API, son rutas vistas y modelos que a continuación, se hace un ejemplo de cada uno de estos elementos:

Rutas: en esta sección se tiene la declaración de las rutas o también conocidos como endpoints, los cuales son las URL que serán consumidas por el frontend. Estos endpoints serán el enlace de entrada de los datos para los distintos tipos de métodos HTTP a realizar, y luego posteriormente validados para hacer las conexiones con la base de datos y su respectiva acción. En la imagen 24 se puede observar un ejemplo de las rutas declaradas para los lugares.

Figura 24. Ejemplo de construcción de endpoints del proyecto.

```
16  from django.contrib import admin
17  from django.urls import path
18  from agro.queries import query
19  from agro.views import PlaceList, PlaceSave, PlaceUpdate, PlaceDelete, PlaceDetails, \...
26
27  urlpatterns = [
28      path('admin/', admin.site.urls),
29      # place
30      path('api/places', PlaceList.as_view(),name='place_list'),
31      path('api/place', PlaceSave.as_view(),name='place_save'),
32      path('api/place/<pk>/update', PlaceUpdate.as_view(),name='place_update'),
33      path('api/place/<pk>/delete', PlaceDelete.as_view(),name='place_delete'),
34      path('api/place/<pk>', PlaceDetails.as_view(),name='place_details'),
35      # TipoDocumento
36      path('api/documenttypes', TipoDocumentoList.as_view(),name='documenttype_list'),
```

Fuente: Elaboración propia.

Vistas: para este apartado, las vistas ayudan a describir cada uno de los métodos respecto al serializador descrito, para ello se importan los serializadores creados por cada entidad involucrada. Lo que se hace es obtener el serializador y describir cada uno de los métodos para la interacción con la base de datos, utilizando el protocolo HTTP con sus métodos, en caso de consulta se utiliza GET, si es una inserción se utiliza POST, si se va a realizar una actualización se usa PUT y cuando se realice una eliminación se utiliza DELETE.

A continuación, se puede observar un ejemplo de vistas para la entidad de lugares.

Figura 25 Ejemplo de construcción de vistas del proyecto.

```python
from django.shortcuts import render
from rest_framework import generics

from .models import Lugar, TipoDocumento, TipoPersona, Persona, Pedido, Personaxpedido,
                    Factura, Usuario
from .serializers import LugarSerializer, TipoDocumentoSerializer, TipoPersonaSerializer
                         PedidoSerializer, PersonaxpedidoSerializer, FacturaSerializer, U
# Create your views here.

# Lugar
class PlaceList(generics.ListCreateAPIView):
    queryset = Lugar.objects.all()
    serializer_class = LugarSerializer

class PlaceSave(generics.CreateAPIView):
    serializer_class = LugarSerializer

class PlaceUpdate(generics.UpdateAPIView):
    queryset = Lugar.objects.all()
    serializer_class = LugarSerializer

class PlaceDelete(generics.DestroyAPIView):
    queryset = Lugar.objects.all()
    serializer_class = LugarSerializer

class PlaceDetails(generics.RetrieveAPIView):
    queryset = Lugar.objects.all()
    serializer_class = LugarSerializer
```

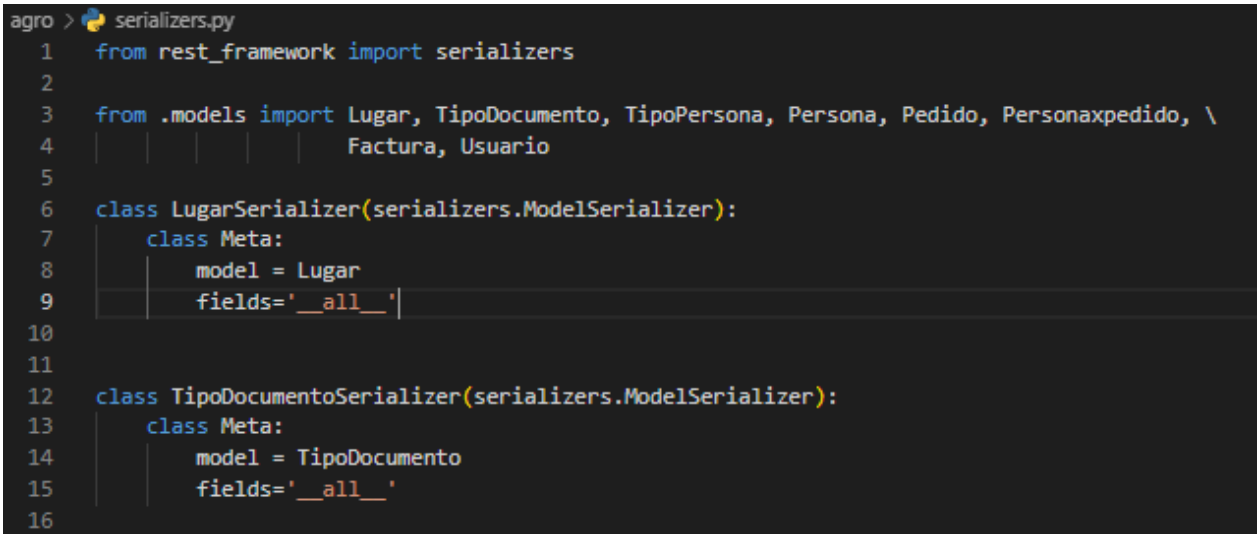
Fuente: Elaboración propia.

Serializadores: son muy útiles porque permiten manejar los datos complejos como consultas o instancias que luego son convertidos para ser fácilmente transmitidos en formatos JSOM o XML, en este desarrollo se utiliza el formato JSON puesto que su manejo es más sencillo,

también evitan la reescritura de código, simplemente se le indica que va a ser un serializador del modelo y se envía el modelo agregándole todos los métodos a efectuarse en ese modelo.

A continuación, en la figura 26 un ejemplo de declaración de serializador para tipo de documento y para lugares:

Figura 26. Ejemplo de construcción de serializador del proyecto.



Fuente: Elaboración propia.

Modelos: en los modelos se declaran las clases con todas las tablas presentes en nuestro modelo entidad relación de la base de datos, lo que se hace es declarar cada uno de los campos a los cuales se le incluye los distintos tipos de datos, al campo definido como llave primaria (primary key) se le asigna la característica para que sea auto incremental, también se declaran los campos que sean recibidos como llaves foráneas (foreign key) en donde se referencia la tabla y el campo a recibir; adicionalmente a los campos que sea necesario agregarles una configuración de extensión como por ejemplo a un tipo caracter para una dirección.

Django construye el ORM (Object-Relational mapping) a partir de la relación de las secciones construidas entre los modelos, los serializadores y las vistas. Esta técnica es la que permite transformar de una fila de la base de datos relacional, a un objeto JSON; en donde cada una de las columnas se convierte en una propiedad del objeto y viceversa.

A continuación, en la figura un ejemplo de la construcción del modelo de la entidad de pedidos de nuestro proyecto:

Figura 27 . Ejemplo de construcción de modelos del proyecto.

agro > models.py

```python
# This is an auto-generated Django model module.
# You'll have to do the following manually to clean this up:
#   * Remove `managed = False` lines if you wish to allow Django to create, modify, and delete the table
# Feel free to rename the models, but don't rename db_table values or field names.
from django.db import models

class Pedido(models.Model):
    id_pedido = models.AutoField(primary_key=True)
    id_persona = models.ForeignKey('Persona', models.DO_NOTHING, db_column='id_persona', blank=True, null=True)
    cantidad = models.IntegerField(blank=True, null=True)
    fecha_pedido = models.DateField(blank=True, null=True)
    fecha_entrega = models.DateField(blank=True, null=True)
    direccion = models.CharField(max_length=50, blank=True, null=True)
    descripcion = models.TextField(blank=True, null=True)
    tarifa = models.IntegerField(blank=True, null=True)
    valor_flete = models.IntegerField(blank=True, null=True)

    class Meta:
        managed = False
        db_table = 'pedido'

```

Fuente: Elaboración propia.

Se utilizó en la codificación otra funcionalidad fuera del ORM, un query que realiza una inserción de tipo join en nuestra base de datos; lo cual cada vez que se crea una persona se inserta automáticamente a la tabla usuario y en esta tabla están las contraseñas codificadas en md5. Cuando un usuario va a realizar el login y escribe su contraseña existe una librería que transforma esos datos en md5 y eso es lo que recibe el backend para realizar la consulta en base de datos de las contraseñas codificadas, en la imagen se puede ver la declaración del query que hace la inserción automática:

Figura 28. Query de inserción

```python
@csrf_exempt
def query(request):
    body_unicode = request.body.decode('utf-8')
    body = json.loads(body_unicode)

    try:
        user = Usuario.objects.get(id_persona__email='{}'.format(body['email']), clave='{}'.format(body['password']))
        # print(user.id_persona.id_persona)
        persona = Persona.objects.get(pk=user.id_persona.id_persona)
        serializer = PersonaSerializer(persona)
        # print(q_json)
        data = {
            "status": 200,
            "data": serializer.data
        }
    except Usuario.DoesNotExist:
        data = {
            "status": 404,
            "message": "User not found"
        }
    return JsonResponse(data, safe=False)

class PersonaSerializer(serial.ModelSerializer):
    class Meta:
        model=Persona
        fields='__all__'
```

Fuente: Elaboración propia.

BASE DE DATOS

Desde Django, se realizan las configuraciones la declaración de los parámetros necesarios para la conexión a la base de datos, en la siguiente figura se puede observar dicha configuración:

Figura 29. Parámetros de conexión a la base de datos.

```python
# Database
# https://docs.djangoproject.com/en/2.2/ref/settings/#databases

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.postgresql_psycopg2',
        'OPTIONS' : {
                'options': '-c search_path=produccion'
            },
        'NAME': 'agrobilbao',
        'HOST': '127.0.0.1',
        'USER':  'postgres',
        'PASSWORD': '*******',
        'PORT': 5432
    }
}
```

Fuente: Elaboración propia.

Triggers: para el buen funcionamiento de la aplicación web y móvil, se implementó la utilización de disparadores (triggers) los cuales se pueden definir como procesos que se ejecutan inmediatamente después que ocurre un evento en alguna tabla de la base de datos ya sea una inserción, eliminación o actualización.

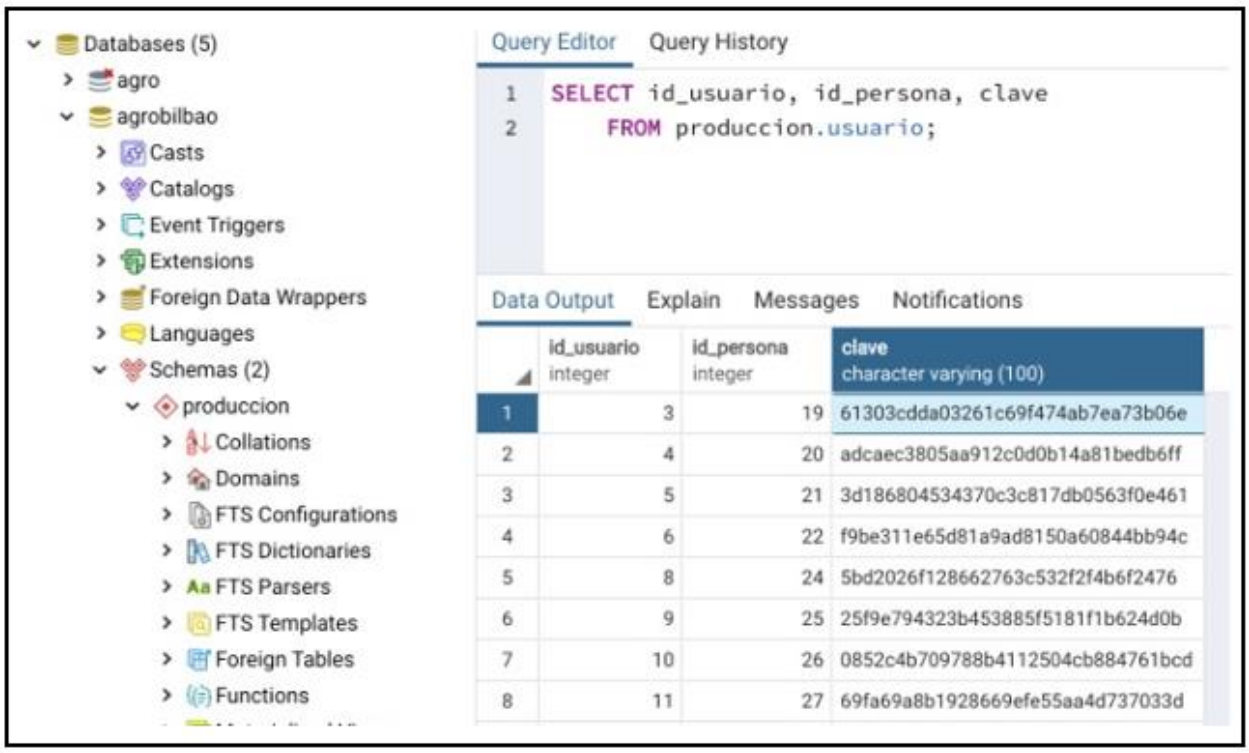
El primer trigger implementado tiene asociado el evento de actualización o inserción en la tabla persona; cada vez que estos eventos sucedan, se hace una actualización o una inserción dependiendo del evento en la tabla usuario, dejando allí el ID y la contraseña convertida en MD5. En la figura 30 se puede observar la declaración del trigger para realizar el proceso mencionado y en la figura 31 se observa la representación en la base de datos de la tabla mencionada:

Figura 30. Declaración del trigger para actualizar usuario.

```
-- FUNCTION: produccion.actualizar_usuario()
-- DROP FUNCTION produccion.actualizar_usuario();
CREATE FUNCTION produccion.actualizar_usuario()
    RETURNS trigger
    LANGUAGE 'plpgsql'
    COST 100
    VOLATILE NOT LEAKPROOF
AS $BODY$
  DECLARE
  BEGIN

    IF tg_op='INSERT' THEN

        INSERT INTO produccion.usuario (id_persona, clave)
            VALUES (NEW.id_persona,md5(NEW.numero_documento));

    ELSIF tg_op='UPDATE' THEN
        UPDATE produccion.usuario SET clave = md5(NEW.numero_documento) WHERE id_persona = OLD.id_persona;
    END IF;

   RETURN NEW;
  END;
$BODY$;

ALTER FUNCTION produccion.actualizar_usuario()
    OWNER TO postgres;
```

Fuente: Elaboración propia.

Figura 31. Visualización de la tabla usuarios en pgAdmin 4.



Fuente: Elaboración propia.

El segundo trigger implementado funciona en la tabla factura y ocurre en el evento de la inserción o actualización por parte del asociado de los kilogramos de aguacate aportados, el trigger se encarga de multiplicar esa cantidad por el valor de la tarifa que ha puesto el cliente para pagar en el pedido, dando como resultado el saldo a cancelar para el asociado.

A continuación, en la figura 32 se puede observar la declaración del trigger que involucra el evento mencionado:

Figura 32. Declaración del trigger para calcular valor de pedido.

```
-- FUNCTION: produccion.actualizar_factura()
-- DROP FUNCTION produccion.actualizar_factura();
CREATE FUNCTION produccion.actualizar_factura()
    RETURNS trigger
    LANGUAGE 'plpgsql'
    COST 100
    VOLATILE NOT LEAKPROOF
AS $BODY$
  DECLARE
    pedido RECORD;
    valor_pedido integer;
  BEGIN

    select * from produccion.pedido where id_pedido = NEW.id_pedido into pedido;
    valor_pedido :=  NEW.cantidad_asociado * pedido.tarifa;

    IF tg_op='INSERT' THEN

    INSERT INTO produccion.factura (id_personaxpedido, valor_pedido_persona)
        VALUES (NEW.id_personaxpedido,valor_pedido);

    ELSIF tg_op='UPDATE' THEN
        UPDATE produccion.factura SET valor_pedido_persona = valor_pedido WHERE id_personaxpedido = OLD.id_personaxpedido;
    END IF;

   RETURN NEW;
  END;
$BODY$;

ALTER FUNCTION produccion.actualizar_factura()
    OWNER TO postgres;
```

Fuente: Elaboración propia.

A continuación en la figura 33 se puede observar una consulta hecha en el cliente de base de datos pgAdmin 4 a la tabla de persona por pedido, donde queda su resultado calculado y actualizado automáticamente:

Figura 33. Visualización de la tabla de pedido por persona.

Fuente: Elaboración propia.

APP MÓVIL

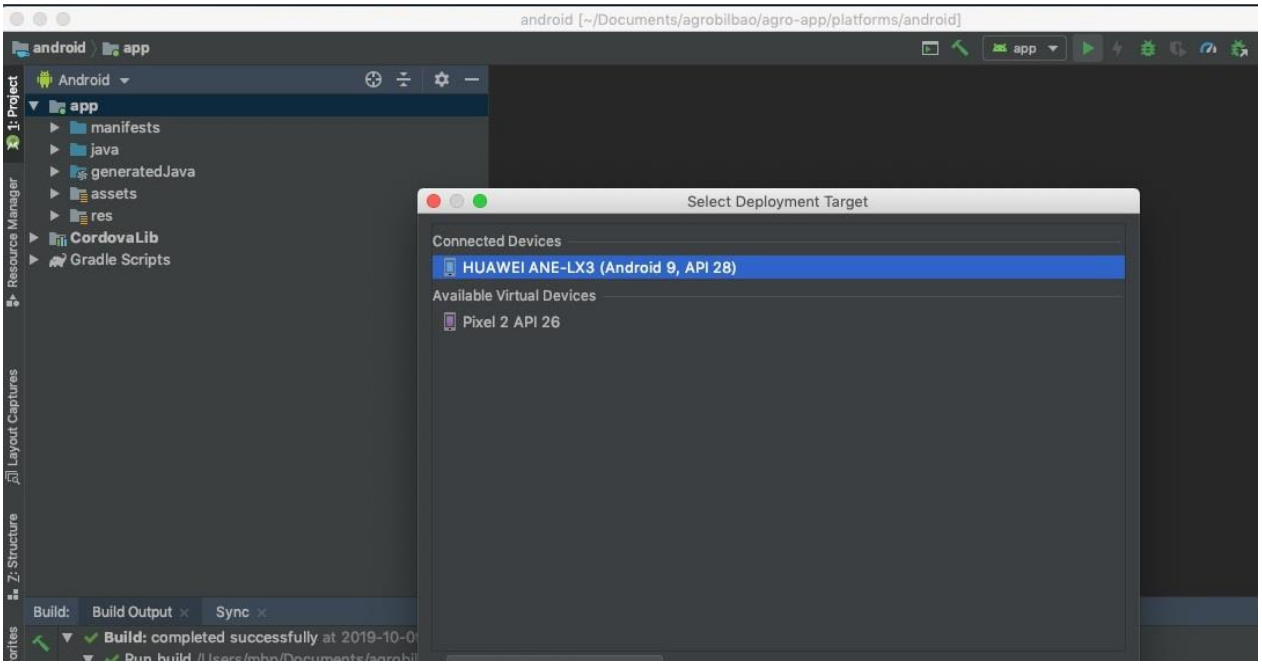
Dentro del proceso de codificación y desarrollo de la aplicación móvil se utilizó el framework ionic, el cual soporta diferentes frameworks o librerías de desarrollo web. TypeScript es el lenguaje usado para la codificación de la aplicación, el cual se transpila o transforma a JavaScript; cuando se va a compilar para algún sistema operativo ionic trabaja junto con la librería cordova para formar un proyecto de Android Studio, generando un archivo; este archivo se abre con Android Studio y ahí permite compilar en un emulador o si se posee un smartphone disponible y conectado al computador. Android Studio permite hacer la firma de las Apps para posteriormente poder subir la aplicación al Play Store.

A continuación, un fragmento del código creado para la realización de la App, se puede observar en el gestor de archivos la carpeta “Android” la cual es utilizada para la compilación del proyecto:

Figura 34. Codificación de la aplicación móvil.

Fuente: Elaboración propia.

Figura 35. Fase donde se compila la app en Android studio

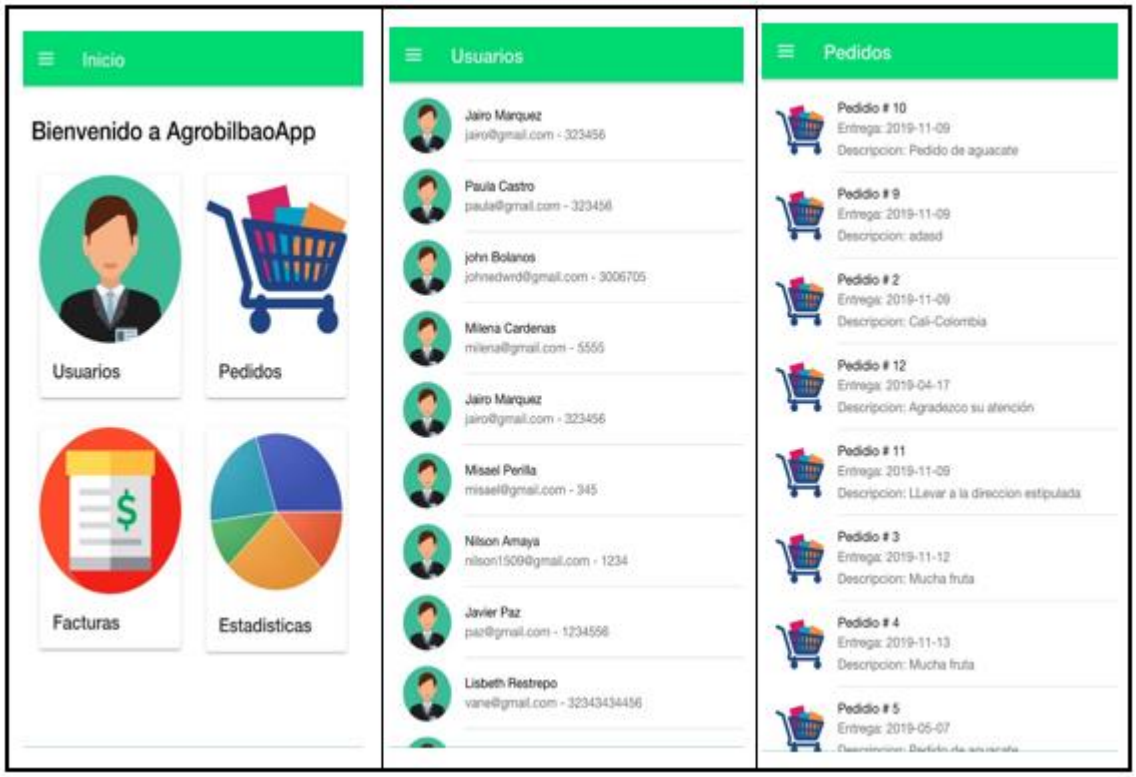


Fuente: Elaboración propia.

Es importante resaltar que esta otra capa de frontend desde un dispositivo móvil, está haciendo las peticiones de consulta al backend desarrollado en Django Rest Framework, lo cual potencia aún más la decisión de separar la lógica del backend y el frontend; es la tendencia del desarrollo actual porque promueve la reutilización de código y mejora las estructuras de los proyectos.

En la siguiente figura se observan tres pantallas de la aplicación móvil, la información aquí expuesta es en tiempo real; todo lo que ingrese a la base de datos se visualizara en la aplicación. Como ejemplo las interfaces del inicio, el módulo de lista de usuario y la lista de los pedidos:

Figura 36. Visualización de la interfaz de la app.



Fuente: Elaboración propia.

8.1.5 ETAPA 5. DESPLIEGUE

Para el despliegue del proyecto, se realizó de manera local en los computadores de trabajo, se deben tener en cuenta los siguientes puntos:

- Del lado del frontend de la App web, el despliegue se hace con NodeJS una herramienta que transforma el código en un archivo .html que el navegador entienda.
- Para el despliegue de lado del backend en Django se debe configurar un entorno virtual para Python el cual permite el despliegue del backend de manera óptima.
- Para la app móvil se hace la compilación desde Android y luego la subida al play store.

8.2 COSTO DEL PROYECTO.

Recursos financieros

Tabla 11. Descripción de los costos económicos para la ejecución del proyecto.

RECURSO	TIPO DE RECURSO	COSTO UNITARIO	CANTIDAD	COSTO TOTAL
PAPELERÍA	INSUMO OFICINA	$ 50.000	1	$ 50.000
TRANSPORTES	MOVILIZACION	$ 240.000	2	$ 480.000
INTERNET	SERVICIO	$ 60.000	5	$ 300.000
TRABAJO DE INGIENERIA	SERVICIO	$ 10.000	350	$ 3.500.000
MOTOROLA MOTO G 5S PLUS	SMARTPHOME	$ 600.000	1	$ 800.000
HUAWEI P20 LITE	SMARTPHOME	$ 570.000	1	$ 570.000
PORTATIL ASUS X550Z	EQUIPO DE COMPUTO	$ 1.500.000	1	$ 1.500.000
MAC BOOK PRO	EQUIPO DE COMPUTO	$ 2.100.000	1	$ 2.100.000
TOTAL		**$ 5.230.000**	**362**	**$ 9.100.000**

Fuente: Elaboración propia.

Recursos técnicos y tecnológicos.

Tabla 12. Especificación de los recursos técnicos y tecnológicos utilizados para el desarrollo del proyecto.

RECURSO	TIPO	ESPECIFICACIONES	COSTO
TELÉFONO CELULAR	MOTOROLA MOTO G 5S PLUS	•ANDROID™ 7.1.1, NOUGAT •MEMORIA RAM 3 GB •REDES: 2G - GSM/GPRS/EDGE (850, 900, 1800, 1900 MHZ) 3G - UMTS/HSPA+ (850, 900, 1900, 2100 MHZ) 4G - LTE (B1, B3, B5, B7, B28) - CAT 6 •ALMACENAMIENTO TOTAL: 32 GB •CÁMARA POSTERIOR CÁMARA DUAL - 13 MP + 13 MP APERTURA F / 2.0 FLASH DUAL DEL LED CON EQUILIBRIO DE COLOR. •CÁMARA FRONTAL 8 MP FLASH DELANTERO LENTE ANGULAR + SELFIE PANORAMIC	$ 600.000
TELÉFONO CELULAR	HUAWEI P20 LITE	•SISTEMA OPERATIVO EMUI 8.0 (BASADO EN ANDROID 8.0) •MEMORIA ANE-LX3: 4 GB + 32 GB (SOPORTE MICROSD DE HASTA 256 GB) •CÁMARA FRONTAL: 16 M F/2.0 •CÁMARA POSTERIOR: 16 M + 2 M F/2.2 •PROCESADOR HUAWEI KIRIN 659 4 X CORTEX-A53 2,36 GHZ + 4 X CORTEX-A53 1,7 GHZ	$ 570.000
EQUIPO DE COMPUTO	PORTATIL ASUS X550Z	•PROCESADOR AMD® FX-7600P DDR3L 1600 MHZ SDRAM, 4 GB. •WINDOWS 8.1. 15.6" 16:9 HD (1366X768)/HD/GL/LED •(1366X768)/FULL HD (1920X1080 •AMD RADEON® R5 M230 + RADEON® R7 •M270 DX DUAL GRAPHICS CON 2GB DDR3 •VRAM. 2.5" 9.5MM SATA 1TB 5400/7200 RPM. •SUPER-MULTI DVD. 2 -EN-1 LECTOR DE TARJETAS (SD/ SDHC/ MMC)	$ 1.500.000

EQUIPO DE COMPUTO	MAC BOOK PRO	•PROCESADOR: CORE I5 •VEL. PROCES: 2.5GHZ (3.0 TURBO BOOST) •MEMORIA: 16GB •DISCO: 1000GB + SOLIDO 120 •GRAFICOS: INTEL HD 4000 •VIDEO 1536MB •PANTALLA: 13.3 LED •PUERTOS: 2USB, SALIDA ENTRADA AUDIO, SDXC, THUNDERBOLT, FIREWIRE	$ 2.100.000
MICROSOFT OFFICE 360 X64	HERRAMIENTA OFIMATICA		$ 350.000
REACT JS Y REACT NATIVE BOOTSTRAP	FRAMEWORKS DE DESARROLLO WEB (FRONTEND)	HERRAMIENTA OPEN SOURCE	$ -
PYTHON NODEJS DJANGO REST	FRAMEWORKS DE DESARROLLO BACKEND	HERRAMIENTA OPEN SOURCE	$ -
POSTGRESQL	MOTOR BASE DE DATOS	HERRAMIENTA OPEN SOURCE	$ -
VISUAL STUDIO CODE	EDITOR DE CODIGO	HERRAMIENTA OPEN SOURCE	$ -
TOTAL			**$ 5.120.000**

Fuente: Elaboración propia.

9. TESTER.

9.1 PRUEBAS DE CAJA BLANCA.

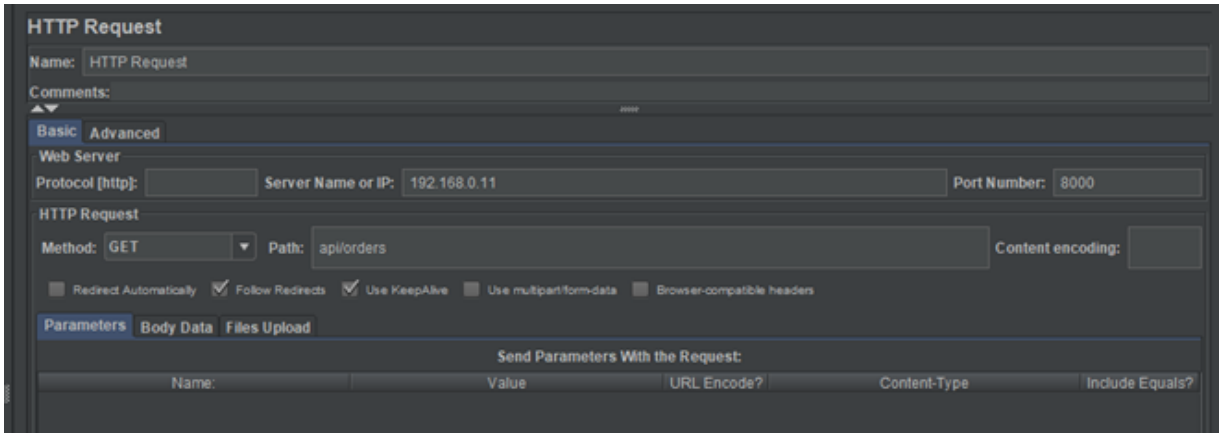
Para este tipo de pruebas se utilizó la herramienta Apache JMeter en su versión 5.1.1 con la cual se hacen muchas peticiones para medir la carga que se esté soportando. A continuación, se van exponer los resultados obtenidos de las pruebas realizadas a la aplicación, las herramientas utilizadas, los métodos o procesos realizados para ello y la información utilizada para realizar las pruebas.

Escenario propuesto: Para el primer test, evaluando las posibles cargas que puede tener el aplicativo, las pruebas serán de consulta de información. Se hace una petición http de tipo GET, y para ello se siguen los siguientes pasos de configuración:

- Se especifica la ruta a la cual se le hace la petición: 192.168.0.11 (en este caso)
- Se selecciona el método, para este caso es de tipo: GET
- En el apartado de “Path” se colocan las extensiones de la ruta y que van a intervenir en la petición: api/orders.

- Se especifica el puerto donde se encuentra desplegado: 8000

Figura 37. Configuración de Jmeter para petición GET.

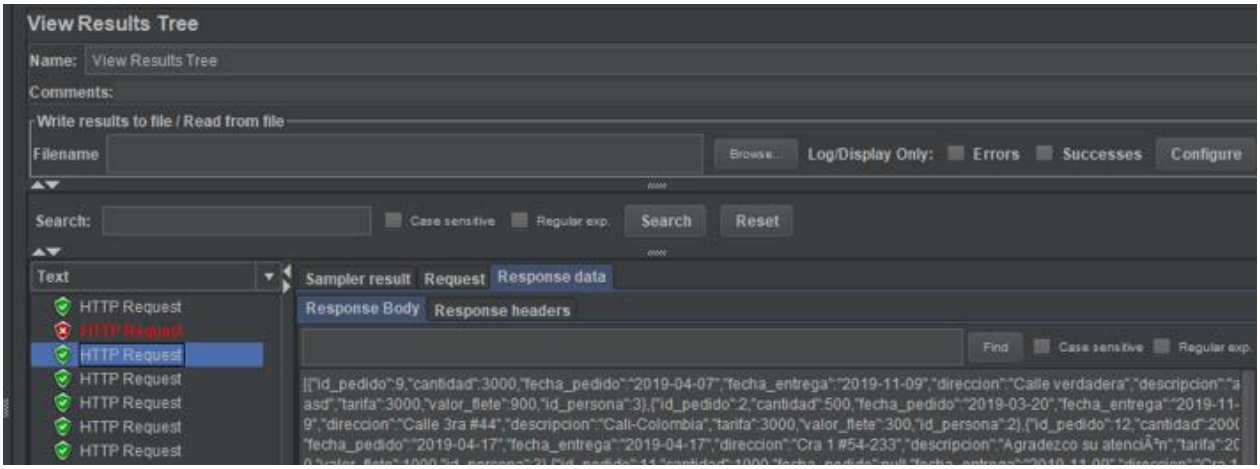


Fuente: Elaboración propia.

Para la prueba se usó un número de 100 usuarios por segundo, una carga de peticiones la cual el tiempo de la simulación duró 5 segundos y se obtuvieron los siguientes resultados:

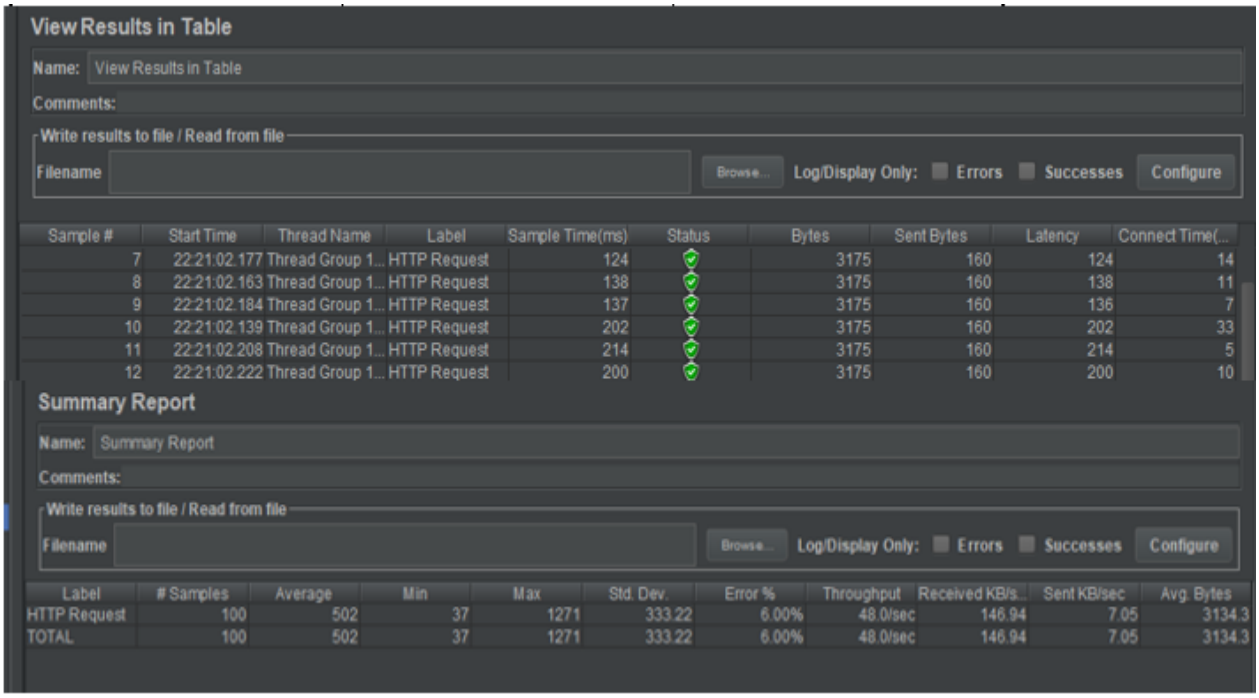
Se obtuvo un resultado de un 94% de efectividad de las peticiones hechas, lo cual es un porcentaje importante para una carga de 100 usuarios en un segundo. En la figura 38 se puede observar el resultado de las peticiones hechas donde se tiene la data de respuesta y en la figura 39 se observa el error obtenido, la latencia, la cantidad de bytes de información obtenida y el tiempo de respuesta.

Figura 38. Árbol de resultados de la prueba.



Fuente: Elaboración propia.

Figura 39. Reporte resumen de la prueba.

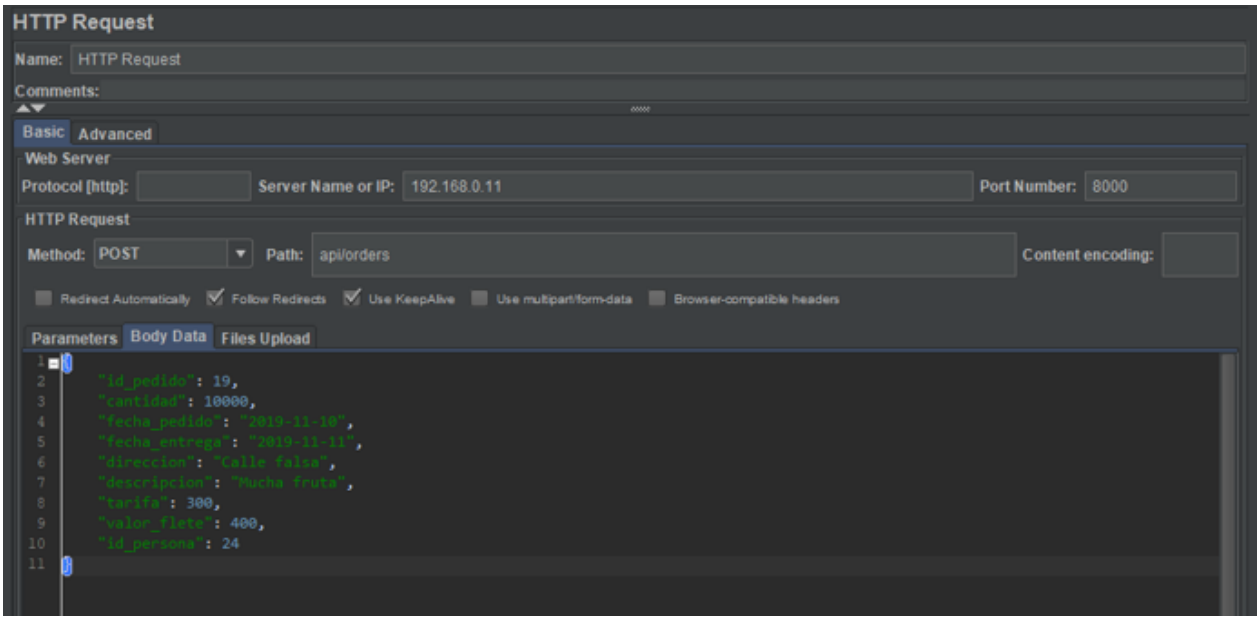


Fuente: Elaboración propia.

Para el segundo test, se hace una petición http de tipo POST, en la que se incluye un paso adicional porque se debe hacer el envió de información y para ello se siguen los siguientes pasos:

- Se especifica la ruta a la cual se le hace la petición: 192.168.0.11 (en este caso)
- Se selecciona el método, para este caso es de tipo: POST
- En el apartado de “Path” se colocan las extensiones de la ruta y que van a intervenir en la petición: api/orders.
- Se especifica el puerto donde se encuentra desplegado: 8000
- Se adiciona un Body Data de tipo Json, en el cual se digita la información con la que se filtró la consulta (figura 40).
- Se incluye una cabecera de a la petición HTTP con un nombre Content-type y un valor de application/json.

Figura 40. Configuración para prueba de inserción.



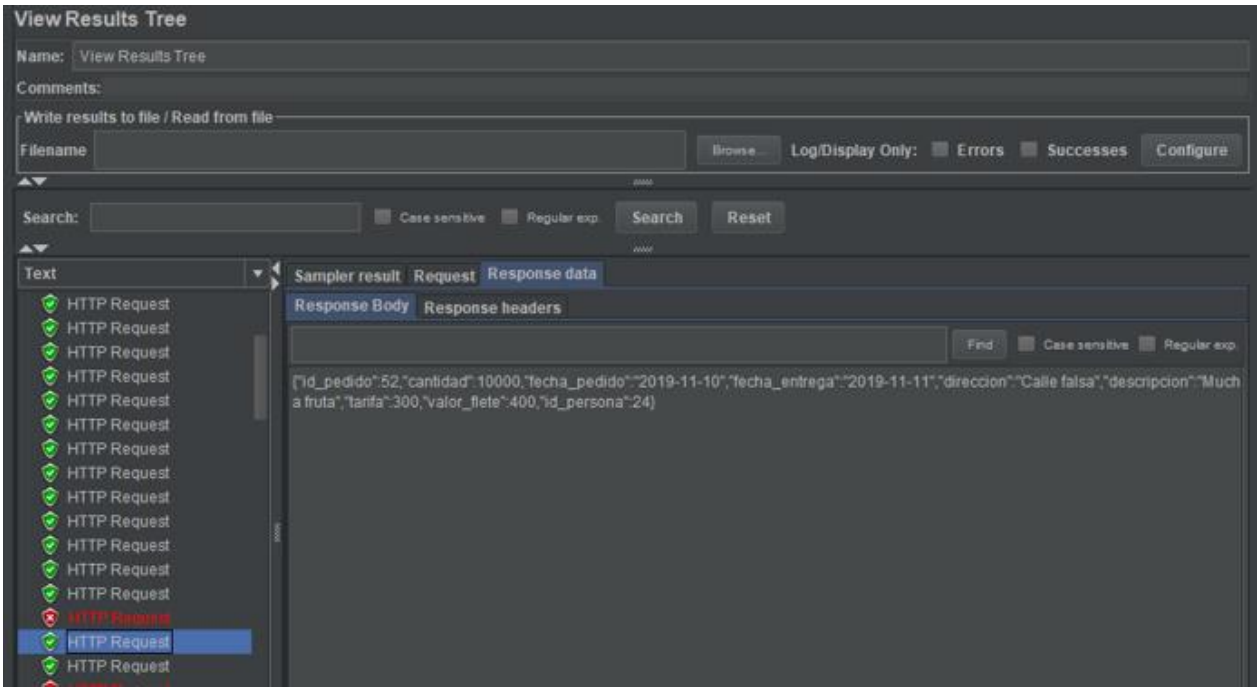
Fuente: Elaboración propia.

En esta prueba se usa el mismo número de 100 usuarios por segundo, una carga de peticiones para insertar el registro en base de datos a través del API, la cual el tiempo de la simulación duró 3 segundos y se obtuvieron los siguientes resultados:

Se obtuvo un resultado de un 93% de efectividad de las inserciones hechas. En la figura 41 se puede observar el resultado de las inserciones en la respuesta se relaciona la información insertada y el retorno también el ID del registro hecho.

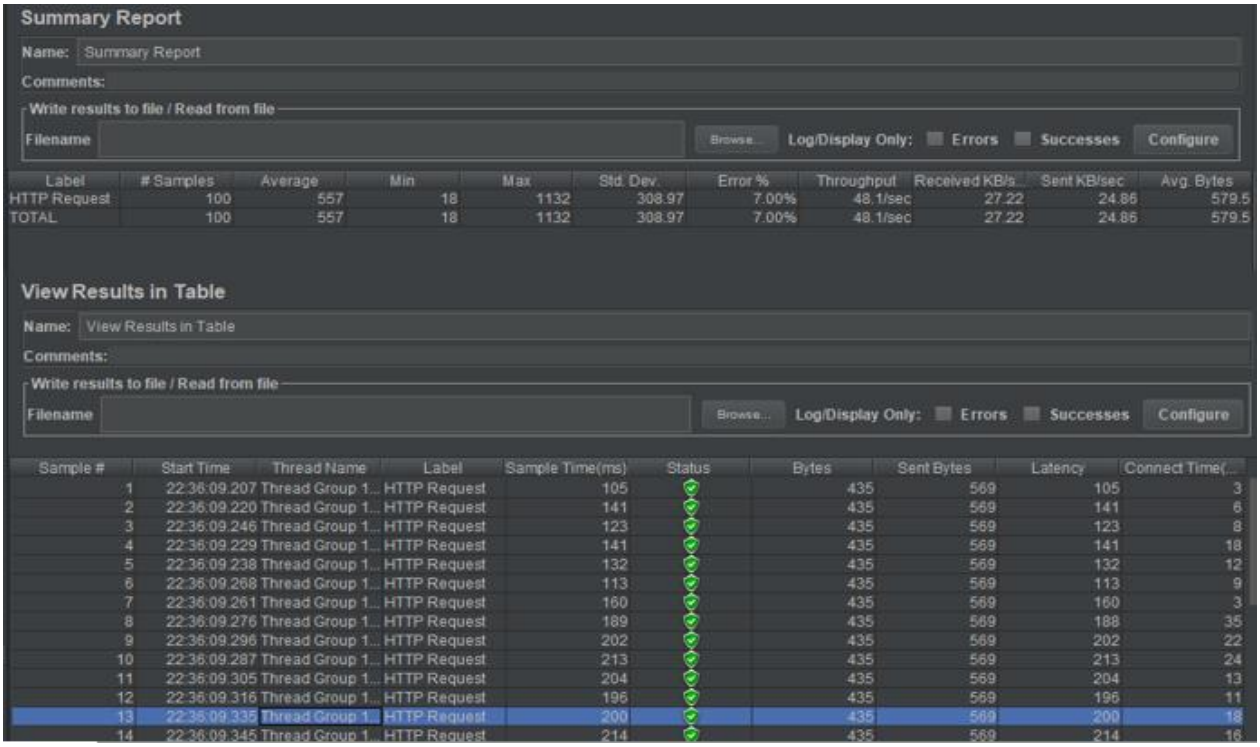
En la figura 42 se observa el error obtenido de 7 %, la latencia, la cantidad de bytes de información registrada (435) y el tiempo de respuesta que en promedio por petición fue de 557 milisegundos:

Figura 41. Árbol de resultados de la inserción.



Fuente: Elaboración propia.

Figura 42. Resumen de resultados.



Fuente: Elaboración propia.

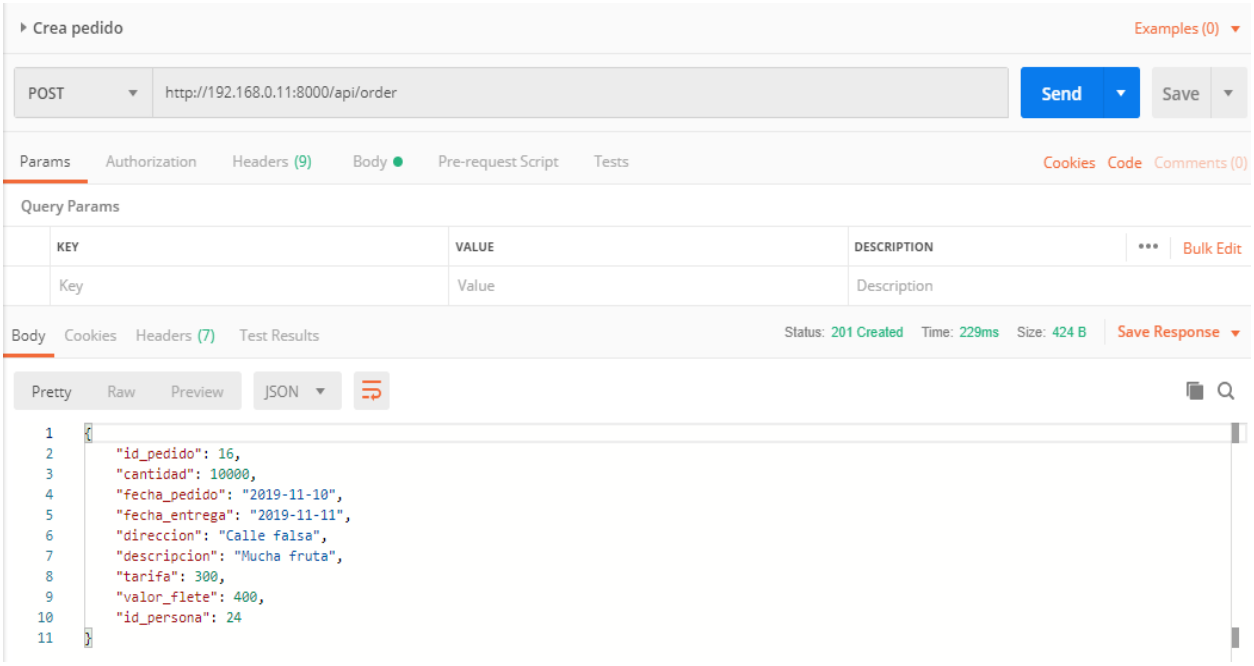
9.2 PRUEBAS DE CAJA NEGRA

Utilizada la aplicación Postman, esta herramienta que principalmente permite crear peticiones sobre API de una manera sencilla y rápida, a medida que se codificaba uno de los métodos de la API se simulaba una petición y se garantiza el óptimo rendimiento del método previamente creado, algo que ayudó a la verificación de cada una de las funcionalidades.

Una petición de tipo POST, en la cual le se envía el objeto a registrar en la base de datos, en este objeto está la información de un pedido de producto con las características del mismo (cantidad, fecha entrega y pedido, dirección, descripción, tarifa, valor flete y un id de persona). la herramienta luego de hacer la petición HTTP, entrega un status 200 lo cual quiere decir que respondió correctamente, en este caso un código 201 que indica que el registro fue creado, verificando en la base de datos el resultado fue exitoso:

A continuación, se puede observar en la figura 43 la simulación:

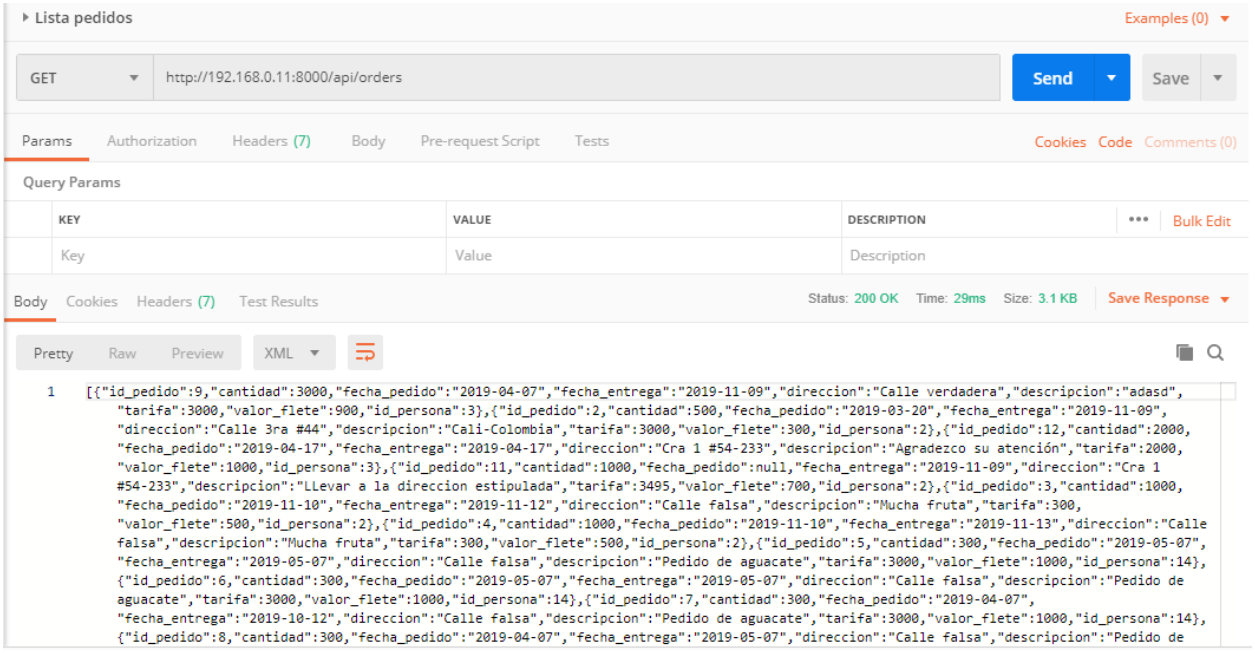
Figura 43. Prueba de caja blanca api POST.



Fuente: Elaboración propia.

En la figura 44 se expone otro caso de prueba, en esta ocasión de tipo GET el cual está haciendo una petición para obtener todas las ordenes guardadas en la tabla en ese momento. Se observa una respuesta de status 200 y un cuerpo con toda la información presente en ese momento:

Figura 44. Prueba de caja blanca api GET



Fuente: Elaboración propia.

CAPITULO 4

10. CONCLUSIONES.

- La implementación de este proyecto, permitió que como estudiantes y desarrolladores de software, se pudieran mejorar las habilidades de investigación, análisis y estudio, para de esta forma, entender lo variable que es el mundo del desarrollo, lo que obliga que se debe estar en constante actualización de los conocimientos, conocer las nuevas tecnologías que surgen para fortalecer en el día a día en el campo profesional.

- Lograr conocer el ejercicio agrícola de una parte del territorio nacional, hizo posible el conocer las diferentes problemáticas que poseen los agricultores, especialmente cuando no se trabaja tecnificadamente, especialmente en cultivos que presentan un gran potencial para ser exportado. El resultado de ello, fue poderles brinda una solución económica y ajustada a las exigencias del mercado actual, que les mejorará la calidad de vida, pero además los beneficiará económicamente.

- Con el análisis de los requerimientos en la etapa inicial, la planeación y posterior modelado, se evidenció que se pudo hacer un desarrollo del aplicativo de forma controlada, con la ventaja de tener una metodología adaptable y que permitió hacer la retroalimentación en cada una de las etapas; así como un buen trabajo en las fases iniciales, al lograr un desarrollo más claro para entregar un producto de calidad y satisfaciendo a las necesidades del usuario final.

- El desacoplamiento del frontend y el backend permitirán a futuro utilizar la lógica del negocio, para usarse en otras interfaces de usuario que agreguen funcionalidades al software; es preciso concluir que en la práctica, el desarrollo de software actual es altamente utilizada contribuyendo a la interoperabilidad y escalabilidad de las aplicaciones.

- React JS es una herramienta que facilitó la creación de interfaces de usuario reutilizables, debido a su filosofía de componentes, que además permitió replicar formularios. Con la experiencia de desarrollador que proporciona, debido a que no recarga la aplicación

cuando se realiza un cambio simple, sino que solo lo hace, cuando son detectados en la estructura del código, lo cual hace que se gane tiempo importante durante el desarrollo.

- Django Rest Framework, fue la tecnología utilizada para desarrollar el back-end, porque proporciona un ORM (object relational maping), que facilita la conexión con la base de datos y la realización de métodos CRUD, para luego serializar los datos resultantes. Los end points del API (entradas del API) son una característica importante de Django, ya que estas se utilizan como patrones para su funcionamiento, funcionalidad que posibilitó la creación de varios end points personalizados para hacer las distintas peticiones en la API.

11. RECOMENDACIONES.

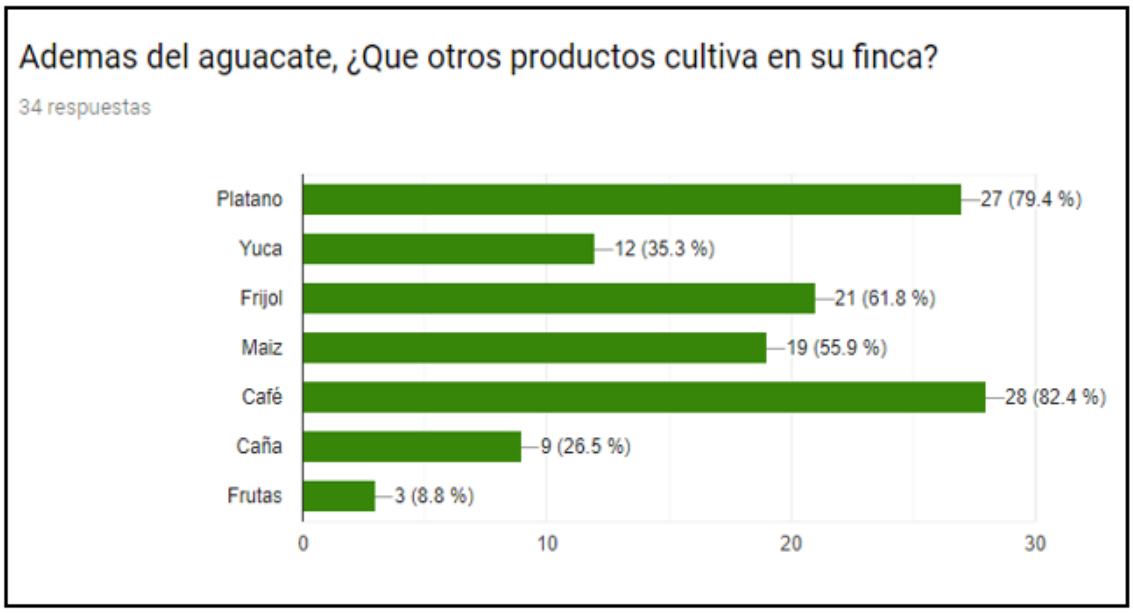
Realizar una buena capacitación de uso de los aplicativos al personal de la asociación tanto en la aplicación web como en la móvil; esto debido a que la población campesina no está muy familiarizada con el uso de las nuevas tecnologías.

Se recomienda tener una buena recepción de señal de Internet para el caso del usuario administrador que dispondrá más actividad en la aplicación web. En el caso de quien haga uso de la aplicación móvil; la encuesta en una escala de 1-5 en calidad de señal, siendo 5 de mejor calidad, un 84.8% de la población se ubican entre 3-5 por lo que se recomienda hacer uso de la herramienta en zonas que puedan garantizar una buena señal para un óptimo funcionamiento.

12. PROYECCIONES.

El desarrollo de este aplicativo tiene como fin, mejorar los procesos de recepción y comercialización de aguacate Hass en el municipio de Planadas Tolima dirigido a la asociación Agrobilbao, pero sería muy positivo aplicarlo a otros productos que se producen no solo en esta región sino en todo el territorio nacional. En la figura a continuación se evidencia la variedad de productos cultivados. Llegar a más agricultores por medio de la implementación de software solventará muchas de las falencias y las dificultades a las que se enfrenta el campesino colombiano.

Figura 45. Diagrama de barras de productos cultivados por los asociados.



Fuente: Elaboración propia.

El software del resultado de este proyecto tiene características que permiten una adecuada escalabilidad; que dan la posibilidad de agregar nuevos módulos o funcionalidades, que mejoren el funcionamiento del sistema y así darle el mejor beneficio posible al campesino, promoviendo la actividad agrícola en la región, lo cual beneficia la economía del entorno.

Incentivar al campesino agricultor para que se adapte al uso del software, promoviendo una cultura de uso de tecnologías que ayuden a mejorar el ejercicio agrícola, por que traerá beneficios como la disminución del desplazamiento hacia las ciudades por falta de oportunidades, el uso de la tecnología puede disminuir la brecha que existe entre el campesino productor y el cliente o consumidor final; eliminando intermediarios para que ellos puedan aumentar su rentabilidad.

REFERENCIAS BIBLIOGRÁFICAS.

Referencias

AgroMovil. (2014). *Innovación Tecnológica* . Obtenido de https://www.redinnovagro.in/docs/agromovil.pdf

AgroWin. (2016). *AgroWin Sistema de Gesttión total para el agro*. Obtenido de http://www.agrowin.com/

Alba, P. R. (2011). *MANUAL DE JAVASCRIPT*. Madrid: EDITORIAL CEP S.L.

Arsys. (03 de Agosto de 2015). *¿Qué son los web services y qué tecnología usan en su desarrollo?* Obtenido de https://www.arsys.es/blog/programacion/diseno-web/web-services-desarrollo/

Bootstrap. (07 de 03 de 2019). *Bootstrap*. Obtenido de https://getbootstrap.com/

Comisión Europea. (2017). *Marco Europeo de Interoperabilidad - Estrategia de aplicación.* Bruselas.

Comproagro. (2016). *¡ Sin Intermediarios !* Obtenido de https://www.comproagro.com/

EcuRed. (12 de 03 de 2017). *EcuRed*. Recuperado el 12 de 03 de 2017, de https://www.ecured.cu/Modelo_en_cascada

Farmsoft. (11 de 10 de 2018). *Software agricola / granjas*. Obtenido de https://farmsoft-es.com/software-para-agricultura.html

Fedesoft. (03 de Junio de 2016). *¿Cómo es la industria de Software y TI colombiana?* Obtenido de https://fedesoft.org/noticias-fedesoft/como-es-la-industria-de-software-y-ti-colombiana/

Foundation, D. S. (2019). *DjangoThe web framework for perfectionists with deadlines.* Obtenido de https://www.djangoproject.com/

Foundation., P. S. (2019). *https://docs.python.org*. Obtenido de https://docs.python.org/3/faq/installed.html

Gauchat, J. D. (2012). *El gran libro de HTML5, CSS3 y Javascript.* Barcelona: MARCOMBO, S.A. 2012.

Geocampo. (2017). *Geocampo Agricultura de Precisión*. Obtenido de http://site.geocampo.co/

ionicframework. (23 de 01 de 2019). *What is Ionic Framework?* Obtenido de https://ionicframework.com/docs/intro

Javier Cuello, J. V. (2014). *Diseñando apps para móviles.* Barcelona: Catalina Duque Giraldo.

Joyanes, J. F. (Junio de 2014). *Sistemas de Información en El Sector Agrícola.* Obtenido de es.scribd.com/document/228070051/Sistemas-de-Informacion-en-El-Sector-Agricola

Laudon, K. C. (2012). *Sistemas de Información gerencial.* Ciudad de Mexico: Pearson.

Lenguaje SQL con MySQL. (14 de Febrero de 2002). *LENGUAJE SQL. GESTION DE DATOS.* Obtenido de http://descargas.pntic.mec.es/mentor/visitas/mysql_I.pdf

McLeod Dennis, S. M. (1981). Abstractions in Databases. SIGMOD Record 11(2).

Microsoft, M. M. (Noviembre de 2013). *Single-Page Applications: Build Modern, Responsive Web Apps* . Obtenido de https://msdn.microsoft.com/en-gb/magazine/dn463786.aspx

Ministerio de Agricultura. (26 de 12 de 2018). *Acerca de Agronet*. Obtenido de http://www.agronet.gov.co/agronet/Paginas/default.aspx

MinTic. (2010). *Marco para la Interoperabilidad.* Bogotá: Marksigma Derechos de autor patrimoniales.

NodeJS. (26 de 03 de 2019). *Acerca de Node.js*. Obtenido de https://nodejs.org/es/about/

Panigrahi, K. (2018). *GraphQL.* Tutorialspoint.

Panigrahi, Kiran. (2018). *React Native.* Tutorials Point.

Platzi. (2018). *Conceptos básicos de React Js que deberías saber*. Obtenido de https://platzi.com/tutoriales/1199-react/1934-conceptos-basicos-de-react-js-que-deberias-saber/

Portafolio. (17 de Abril de 2017). *8.376.463: las víctimas del conflicto armado en Colombia*. Obtenido de https://www.portafolio.co/economia/gobierno/el-numero-de-victimas-del-conflicto-armado-en-colombia-504833

PostgreSQL. (15 de Abril de 2019). *Acerca de postgresql*. Obtenido de https://www.postgresql.org/about/

Pressman, R. S. (2010). *Ingeniería del Software. Un Enfoque Practico.* Connecticut: McGraw-Hill Interamericana Editores S.A.

StackOverflow. (Abril de 2019). *Resultados de la encuesta del desarrollador 2019*. Obtenido de https://insights.stackoverflow.com/survey/2019#overview

Vázquez, I. M. (20 de 06 de 2011). Definición de un Framework para aplicaciones web con navegación sensible a concerns (Tesis de maestría). Universidad Nacional de la Plata, Argentina.

ANEXOS.

Anexo A Tabla de relación de la encuesta realizada con las respuestas obtenidas.

Marca temporal	¿Posee y utiliza computador ya sea portátil o de escritorio?	¿Posee un teléfono inteligente? (Celular)	¿Qué sistema operativo usa su teléfono inteligente?	¿Cuáles son las aplicaciones que más utiliza?	Además del aguacate, ¿Que otros productos cultiva en su finca?	¿Cada cuanto cosecha el aguacate?	¿Cuál considera usted que es el proceso más duro de realizar?	¿Cuenta con un plan de datos para su celular?	¿Qué tan buena es la recepción de señal telefónica en su finca?	¿Cuenta con el servicio de Internet en su vivienda?	¿Qué nivel de escolarización tiene usted?
2/8/2019 11:52:36	No	No	Android	Ninguna	Plátano, Frijol, Maíz	De 2 a 3 meses	La fumigación	No	4	No	Básica primaria terminada
2/8/2019 11:55:27	No	Sí	Android	Facebook, WhatsApp	Plátano, Frutas	-	La fumigación	Otra	5	No	Básica primaria incompleta
2/8/2019 11:58:46	SI	Sí	Android	Facebook, WhatsApp, Gmail	Plátano, Yuca, Frijol, Maíz	De 3 a 4 meses	la fumigación	Otra	4	No	Básica primaria terminada
2/9/2019 14:56:18	SI	Sí	Android	Facebook, WhatsApp, Gmail, Outlook (Hotmail)	Frijol, Maíz, Café, Caña	De 1 a 2 meses	Fertilización	Si	5	Sí	Tecnólogo
2/9/2019 15:04:02	No	No	Android	Ninguna	Plátano, Frijol, Maíz, Café, Caña	-	Injertar	Si	5	No	Básica primaria incompleta
2/9/2019 15:06:03	No	Sí	Android	Facebook, WhatsApp	Plátano, Maíz, Café	De 4 a 5 meses	Siembra	No	3	No	Básica secundaria incompleta
2/9/2019 15:10:01	No	Sí	Android	Facebook, WhatsApp	Plátano, Yuca, Frijol, Maíz, Café, Caña	-	La fumigación	Si	5	Sí	Básica primaria incompleta

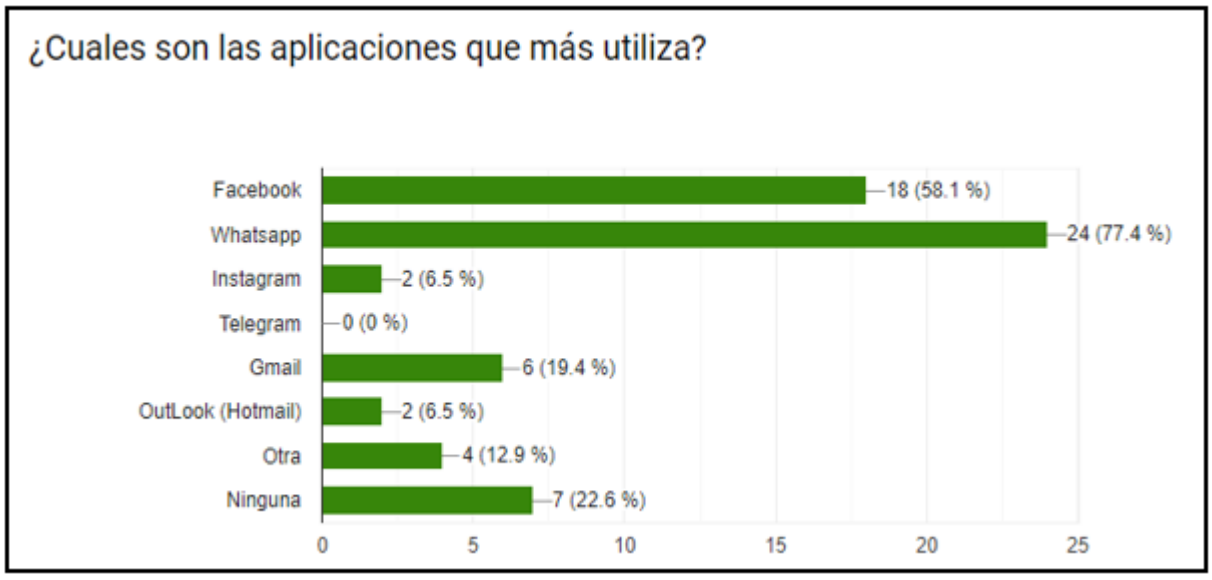
2/9/2019 15:12:07	No	Sí	Android	Facebook, WhatsApp	Plátano, Frijol, Maíz, Café	-	Plantación	Si	4	No	Básica secundaria incompleta
2/9/2019 15:14:17	SI	Sí	Android	Facebook, WhatsApp, Instagram, Gmail	Plátano, Yuca, Frijol, Maíz, Café	-	La fumigación	Si	3	No	Básica secundaria terminada
2/9/2019 15:15:49	SI	Sí	Android	Facebook, WhatsApp, Instagram, Gmail, Outlook (Hotmail)	Plátano, Yuca, Maíz, Café	De 2 a 3 meses	La fumigación	Si	2	No	Básica secundaria terminada
2/9/2019 15:17:43	No	Sí	Android	Facebook, WhatsApp, Gmail	Plátano, Café	De 3 a 4 meses	La fumigación	Si	3	No	Básica primaria terminada
2/9/2019 15:20:15	No	Sí	Android	WhatsApp	Plátano, Café	-	Injertar	No	2	No	Básica primaria incompleta
2/9/2019 15:22:28	No	Sí	Android	WhatsApp	Plátano, Frijol, Café	-	Injertar	Si	3	No	Básica primaria terminada
2/9/2019 15:24:03	No	No	Android	-	Plátano, Café	-	La poda	No	2	No	Básica primaria terminada
2/9/2019 15:26:29	No	Sí	Android	-	Café, Caña	-	Fumigación	Si	3	No	Básica primaria incompleta
2/9/2019 15:28:28	No	Sí	Android	Facebook, WhatsApp	Plátano, Café, Caña	-	Fumigación	Si	3	No	Básica secundaria terminada
2/9/2019 15:31:30	SI	Sí	Android	Facebook, WhatsApp	Plátano, Yuca, Frijol, Maíz, Café, Caña	De 1 a 2 meses, De 2 a 3 meses	Fumigación	Si	3	No	Básica primaria terminada
2/9/2019 15:48:36	SI	Sí	Android	WhatsApp	Plátano, Yuca, Frijol, Maíz, Café, Caña	De 4 a 5 meses	Fumigación	Si	3	No	Básica primaria terminada

2/9/2019 15:55:10	No	Sí	Android	Facebook, WhatsApp	Maíz	De 4 a 5 meses	Fumigación	Si	4	No	Básica secundaria terminada
2/9/2019 16:04:58	No	Sí	Android	Facebook, WhatsApp	Maíz	De 4 a 5 meses	Fumigación	Si	4	No	Básica secundaria terminada
2/9/2019 16:11:47	No	No	Android	Ninguna	Plátano, Yuca, Maíz, Café	-	Fumigación	No	3	No	Básica primaria incompleta
2/9/2019 16:15:06	No	Sí	Android	Facebook, WhatsApp	Plátano, Yuca, Frijol, Maíz, Café	-	Plagas	Si	4	No	Básica primaria terminada
2/10/2019 10:30:03	No	No	Android	Ninguna	Plátano, Café, Caña	De 4 a 5 meses	Asistencia Técnica	Si	4	No	Ninguno
2/10/2019 10:31:21	No	Sí	Android	WhatsApp	Yuca, Frijol, Maíz, Café, Frutas	De 4 a 5 meses	Asistencia Técnica	No	3	Sí	Básica secundaria terminada
2/10/2019 10:33:02	No	Sí	Android	Facebook, WhatsApp, Otra	Yuca, Frijol, Maíz, Café	De 4 a 5 meses	Asistencia Técnica	Si	3	Sí	Básica primaria incompleta
2/10/2019 10:34:42	No	Sí	Android	Facebook, WhatsApp, Otra	Frijol, Café	De 4 a 5 meses	Asistencia Técnica	Si	1	No	Básica secundaria terminada
2/10/2019 10:36:19	No	No	Android	Ninguna	Plátano, Yuca, Frijol, Café	De 4 a 5 meses	Levante	Si	5	No	Ninguno
2/10/2019 10:37:56	No	Sí	Android	Facebook, WhatsApp, Otra	Plátano, Frijol, Café	De 2 a 3 meses	Asistencia Técnica	Si	2	No	Básica secundaria incompleta
2/10/2019 10:39:07	No	No	Android	Ninguna	Plátano, Frijol, Café	De 4 a 5 meses	Asistencia Técnica	No	3	No	Básica secundaria terminada
2/10/2019 10:40:28	No	No	Android	Ninguna	Plátano, Frijol, Maíz, Café	De 4 a 5 meses	Asistencia Técnica	No	3	No	Básica primaria incompleta

2/10/2019 10:42:14	SI	Sí	Android	Facebook, WhatsApp, Gmail, Otra	Plátano, Frijol, Caña, Frutas	De 2 a 3 meses	Asistencia Técnica	Si	5	Sí	Tecnólogo
2/10/2019 10:43:19	No	Sí	Android	WhatsApp	Plátano, Frijol, Maíz, Café	De 2 a 3 meses, De 4 a 5 meses	Asistencia Técnica	No	3	No	Básica primaria terminada
2/10/2019 10:45:38	No	Sí	Android	WhatsApp	Plátano, Yuca, Frijol, Café	De 4 a 5 meses	Asistencia Técnica	Si	5	Sí	Básica primaria terminada
2/10/2019 10:46:59	No	No	Android	Ninguna	Plátano, Café	De 4 a 5 meses	Levante	No	5	No	Básica primaria incompleta

Anexo B Gráficos de resultados.

En el siguiente gráfico se evidencian los resultados en cuanto al tipo de aplicaciones que usan normalmente los encuestados:



En el diagrama se observa el claro dominio del sistema operativo Android en los teléfonos inteligentes de los miembros de la asociación:

Anexo C Registro fotográfico de las visitas a los cultivos y la interacción con las personas de la asociación.