

ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN DE TECNÓLOGOS

DESARROLLO DE APLICACIÓN MÓVIL PARA “VÍVERES DANIELA” DEL CONJUNTO HABITACIONAL LULUNCOTO

TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE TECNÓLOGO SUPERIOR EN DESARROLLO DE SOFTWARE

ESTEBAN ALEJANDRO RÍOS PÉREZ

esteban.rios@epn.edu.ec

MARVIN ANIBAL ZAMBRANO GILER

marvin.zambrano@epn.edu.ec

DIRECTOR: Ing. Ivonne Fernanda Maldonado Soliz, MSc.

ivonne.maldonadof@epn.edu.ec

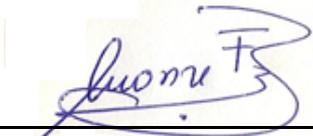
CODIRECTOR: Ing. Luz Marina Vintimilla Jaramillo, MSc.

marina.vintimilla@epn.edu.ec

Quito, abril 2021

CERTIFICACIÓN

Certificamos que el presente trabajo fue desarrollado por los Sres. Esteban Alejandro Ríos Pérez y Marvin Aníbal Zambrano Giler como requerimiento parcial a la obtención del título de TECNÓLOGO SUPERIOR EN DESARROLLO DE SOFTWARE, bajo nuestra supervisión:



Ing. Ivonne Maldonado MSc

DIRECTOR(A) DEL
PROYECTO

Ing. Marina Vintimilla MSc

CODIRECTOR(A) DEL PROYECTO

DECLARACIÓN

Nosotros Ríos Pérez Esteban Alejandro con CI: 1726548108 y Zambrano Giler Marvin Aníbal con CI: 1750804567, declaramos bajo juramento que el trabajo aquí descrito es de nuestra autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y que hemos consultado las referencias bibliográficas que se incluyen en este documento.

Sin perjuicio de los derechos reconocidos en el primer párrafo del artículo 144 del Código Orgánico de la Economía Social de los Conocimientos, Creatividad e Innovación – COESC-, somos titulares de la obra en mención y otorgamos una licencia gratuita, intransferible y no exclusiva de uso con fines académicos a la Escuela Politécnica Nacional.

Entregamos toda la información técnica pertinente, en caso de que hubiese una explotación comercial de la obra por parte de la EPN, se negociará los porcentajes de los beneficios conforme lo establece la normativa nacional vigente.



**Esteban Alejandro Ríos
Pérez**



Marvin Aníbal Zambrano Giler

DEDICATORIA

En primer lugar, quiero dedicar este trabajo de titulación a toda mi familia, principalmente a mis padres que me han brindado todo su apoyo a lo largo de todo este proceso, y por haberme formado como una gran persona con buenos valores. También por siempre estar alentándome para que yo pueda cumplir cada uno de mis objetivos que me he planteado hasta ahora.

A mis amigos incondicionales, quienes confiaron en mí y siempre me han brindado todo su apoyo para que no me dé por vencido, y estuvieron presentes durante este largo camino.

RIOS PEREZ ESTEBAN ALEJANDRO

AGRADECIMIENTO

En primer lugar, agradezco a Dios, por siempre darme la fuerza para nunca rendirme y poder salir adelante, en especial ahora que estamos pasando por momentos difíciles. Agradezco a la Escuela Politécnica Nacional, por darme la oportunidad de estudiar en sus aulas, ya que así pude crecer de manera personal y profesional.

Agradezco a cada uno de los ingenieros que siempre han sabido compartir sus conocimientos, que a la final han sido de mucha ayuda para el desarrollo del proyecto. De manera especial, quiero agradecer a la Ing. Ivonne Maldonado y a la Ing. Luz Marina Vintimilla, por todo el apoyo brindado para poder cumplir con el objetivo de culminar el proyecto de titulación.

A mi compañero de tesis, por el trabajo en conjunto, por no rendirse y por su constante apoyo. Agradezco a mis padres, mi familia y mis amigos por todo su apoyo, su tiempo y sus consejos que han sido un factor importante durante toda mi carrera universitaria.

RIOS PEREZ ESTEBAN ALEJANDRO

DEDICATORIA

En primer lugar, quiero dedicar este trabajo de titulación principalmente a mis padres quienes me apoyaron y confiaron en mí en todo momento, supieron inculcarme buenos valores y que nunca hay que rendirse a pesar de las dificultades. A mis abuelos quienes me brindaron sus consejos y sabiduría, para así convertirme en el muchacho de bien que soy ahora.

También lo quiero dedicar a mis familiares y amigos más cercanos quienes me alentaron, confiaron en mí y siempre me han brindado todo su apoyo incondicional para completar cada una de las metas que me he propuesto.

ZAMBRANO GILER MARVIN ANIBAL

AGRADECIMIENTO

En primer lugar, agradezco a Dios, por siempre darme la fuerza y las ganas para nunca rendirme, dar siempre lo mejor y salir adelante, sobre todo ahora que pasamos por momentos difíciles. Agradezco a la Escuela Politécnica Nacional, por darme la oportunidad de adquirir más conocimiento y experiencia en lo laboral y personal en sus aulas.

Agradezco a cada uno de los ingenieros que compartieron de sus conocimientos y experiencias que han sido de mucha ayuda para el desarrollo del todo el proyecto. De manera especial, quiero agradecer a la Ing. Ivonne Maldonado y a la Ing. Luz Marina Vintimilla, por todo el apoyo y guía brindada para poder culminar el presente proyecto de titulación.

A mi compañero de tesis, por el trabajo en conjunto, por no rendirse, por su constante apoyo y por el empeño puesto. Agradezco a mis padres, abuelos, mis familiares y mis amigos incondicionales por todo su ánimo, tiempo, ayuda, buenas vibras y sus consejos que me han servido para completar cada una de las fases que he pasado hasta este momento.

ZAMBRANO GILER MARVIN ANIBAL

ÍNDICE DE CONTENIDOS

| | | |
|-------|---|----|
| 1 | Introducción..... | 1 |
| 1.1 | Objetivo general..... | 2 |
| 1.2 | Objetivos específicos..... | 2 |
| 1.3 | Alcance | 2 |
| 2 | Metodología..... | 4 |
| 2.1 | Metodología de Desarrollo..... | 4 |
| 2.1.1 | Roles..... | 4 |
| 2.1.2 | Artefactos..... | 5 |
| 2.2 | Diseño de interfaces | 7 |
| 2.2.1 | Marvel App..... | 7 |
| 2.3 | Diseño de la arquitectura..... | 8 |
| 2.3.1 | Patrón arquitectónico Modelo Vista Controlador (MVC)..... | 8 |
| 2.4 | Herramientas de desarrollo | 8 |
| 2.4.1 | Librerías..... | 10 |
| 3 | Resultados y Discusión..... | 11 |
| 3.1 | Sprint 0. Configuración del ambiente de desarrollo..... | 11 |
| 3.1.1 | Diseño y creación de la base de datos..... | 11 |
| 3.1.2 | Creación de proyecto en Ionic..... | 11 |
| 3.1.3 | Creación del proyecto en Laravel..... | 12 |
| 3.1.4 | Usuarios..... | 13 |
| 3.2 | Sprint 1. Autenticación y Módulo Productos (Cliente)..... | 13 |
| 3.2.1 | Inicio de sesión..... | 13 |
| 3.2.2 | Registro de usuarios..... | 15 |
| 3.2.3 | Catálogo de productos..... | 17 |
| 3.3 | Sprint 2. Módulo Compras (Carrito) | 17 |
| 3.3.1 | Carrito de compras y tipo de entrega..... | 17 |
| 3.4 | Sprint 3. Módulo Pedidos (Cliente)..... | 19 |

| | | |
|-------|---|----|
| 3.4.1 | Notificación y visualización de pedido recibido..... | 20 |
| 3.4.2 | Barra de progreso de pedido..... | 20 |
| 3.4.3 | Pedidos por completar. | 21 |
| 3.5 | Sprint 4. Módulo Productos y Módulo Pedidos (Propietario) | 22 |
| 3.5.1 | Registro y actualización de productos. | 22 |
| 3.5.2 | Visualización y notificación de nuevo pedido. | 23 |
| 3.6 | Sprint 5. Módulo Reportes | 24 |
| 3.6.1 | Reporte de ventas por Cliente. | 24 |
| 3.6.2 | Reporte de Productos..... | 24 |
| 3.6.3 | Reporte de Stock..... | 25 |
| 3.7 | Sprint 6. Pruebas y Despliegue | 26 |
| 3.7.1 | Pruebas unitarias..... | 26 |
| 3.7.2 | Pruebas de carga al API | 27 |
| 3.7.3 | Pruebas de compatibilidad | 27 |
| 3.7.4 | Pruebas de aceptación..... | 29 |
| 3.7.5 | Despliegue de la Aplicación Móvil a Producción | 30 |
| 4 | Conclusiones y Recomendaciones..... | 32 |
| 4.1 | Conclusiones..... | 32 |
| 4.2 | Recomendaciones | 33 |
| 5 | Referencias Bibliográficas | 34 |
| 6 | ANEXOS..... | i |
| 6.1 | Manual Técnico..... | i |
| 6.2 | Manual de Usuario..... | i |
| 6.3 | Manual de Instalación..... | i |

ÍNDICE DE FIGURAS

| | |
|---|----|
| Fig. 1: Prototipo de interfaz..... | 7 |
| Fig. 2: Patrón arquitectónico de la Aplicación Móvil | 8 |
| Fig. 3: Proyecto de Ionic | 12 |
| Fig. 4: Proyecto de Laravel..... | 12 |
| Fig. 5: Usuarios asignados para la aplicación móvil | 13 |
| Fig. 6: Módulo de Login | 14 |
| Fig. 7: Mensaje de error en el Login..... | 14 |
| Fig. 8: Módulo de Lista de Productos (cliente)..... | 14 |
| Fig. 9: Validaciones del inicio de sesión..... | 15 |
| Fig. 10: Vista de confirmación de correo..... | 15 |
| Fig. 11: Módulo de Registro de nuevos usuarios..... | 16 |
| Fig. 12: Correo de verificación de cuenta..... | 16 |
| Fig. 13: Validación de campos para el registro | 16 |
| Fig. 14: Catálogo de productos (cliente) | 17 |
| Fig. 15: Carrito de compras | 18 |
| Fig. 16: Estado inicial cantidad del producto..... | 18 |
| Fig. 17: Cambio de cantidad..... | 18 |
| Fig. 18: Eliminación de producto en el carrito | 19 |
| Fig. 19: Restricción por carrito vacío | 19 |
| Fig. 20: Alerta nueva notificación | 20 |
| Fig. 21: Información de notificación..... | 20 |
| Fig. 22: Barra de progreso (Estado del pedido)..... | 21 |
| Fig. 23: Lista de pedidos pendientes..... | 21 |
| Fig. 24: Detalle del pedido | 21 |
| Fig. 25: Módulo Registro de productos | 22 |
| Fig. 26: Módulo Actualizar un producto..... | 22 |
| Fig. 27: Alerta de nuevo pedido..... | 23 |
| Fig. 28: Información de notificación..... | 23 |
| Fig. 29: Lista ordenes nuevas | 23 |
| Fig. 30: Detalle pedido y cambio de estado | 23 |
| Fig. 31: Reportes (Ventas por cliente)..... | 24 |
| Fig. 32: Descarga archivo PDF | 24 |
| Fig. 33: Reportes (Productos más vendidos)..... | 25 |
| Fig. 34: Descarga archivo PDF | 25 |

| | |
|---|----|
| Fig. 35: Reportes (Stock)..... | 25 |
| Fig. 36: Descarga archivo PDF | 25 |
| Fig. 37: Prueba unitaria del componente de actualizar cantidad y precio en el carrito de compras | 27 |
| Fig. 38: Prueba de carga endpoint de productos | 27 |
| Fig. 39: Módulo del carrito de compra en el Huawei PSmart | 28 |
| Fig. 40: Módulo del carrito de compra en el Samsung Galaxy A20s..... | 29 |
| Fig. 41: Módulo de carrito de compras en el Samsung A5..... | 29 |
| Fig. 42: Despliegue de la Aplicación Móvil a producción..... | 31 |

ÍNDICE DE TABLAS

| | |
|--|----|
| TABLA I Roles del Proyecto | 5 |
| TABLA II: Historia de usuario Nro. 7: Asignar cantidad de productos | 6 |
| TABLA III: Herramientas para el desarrollo de la Aplicación Móvil..... | 9 |
| TABLA IV: Librerías usadas en el desarrollo de la Aplicación Móvil..... | 10 |
| TABLA V Especificaciones de los dispositivos para las pruebas de compatibilidad ... | 28 |
| TABLA VI: Prueba de Aceptación Nro. 2: Iniciar sesión..... | 30 |

RESUMEN

El confinamiento y la emergencia sanitaria por el Covid-19, ha cambiado la forma de consumir bienes y servicios, situación que probablemente sea permanente. Medidas como guardar distancia, protección facial, limitar las salidas de casa o mantenerse en ella y hacer uso de aplicaciones *delivery*, son algunas de las alternativas de seguridad para evitar o reducir contagios.

La tecnología se ha convertido en una opción que facilita la apertura de las actividades económicas, ejemplo de esto es la tendencia por el uso de las aplicaciones móviles como herramienta principal en realizar pedidos a domicilio.

Por esta razón se ha desarrollado una Aplicación Móvil que permite a las personas solicitar víveres dentro del Conjunto Habitacional Luluncoto, evitando de esta manera la propagación del Covid-19 por posibles aglomeraciones.

Este trabajo está implementado bajo un entorno de desarrollo con Ionic y PostgreSQL como sistema gestor de base de datos (SGBD). El desarrollo está basado en Scrum, metodología ágil que ha aportado beneficios como: adecuada recopilación de requerimientos, flexibilidad, roles bien definidos, interacción directa con el usuario, entre otros. Permitiendo que el desarrollo se lo realice con calidad y en los tiempos establecidos. Además, se presenta los resultados que se han obtenido en las tareas realizadas en cada Sprint y el despliegue a producción de la Aplicación Móvil en Google Play Store. Finalmente, se mencionan conclusiones y recomendaciones que se han obtenido a lo largo del desarrollo del presente proyecto.

PALABRAS CLAVE: *Scrum, Ionic*, conjunto habitacional, Luluncoto aplicación móvil.

ABSTRACT

The confinement and health emergency by Covid-19 has changed the way we consume goods and services, a situation that is likely to be permanent. Measures such as saving distance, protecting facials, limiting, or staying at home and making use of *delivery apps* are some of the safety alternatives to prevent or reduce contagion.

Technology has become an option that facilitates the opening of economic activities; an example of this is the trend for the use of mobile applications as the main tool in placing home orders.

For this reason, a Mobile Application has been developed that allows requesting groceries within the “Conjunto Habitacional Luluncoto”, thus avoiding the spread of Covid-19 due to possible crowds.

This work is implemented under a development environment with Ionic and MySQL as a database manager system (SGBD). The development is based on Scrum, agile methodology that has brought benefits such as: adequate collection of requirements, flexibility, well-defined roles, direct interaction with the user, among others. Allowing development to be done with quality and at set times. In addition, the results that have been obtained in the tasks performed in each Sprint and the deployment to production of the Mobile Application in the Google Play Store are presented. Finally, there are conclusions and recommendations that have been obtained throughout the development of this project.

KEYWORDS: *Scrum, Ionic*, housing complex, Luluncoto, App Mobile.

1 INTRODUCCIÓN

El COVID-19 no solo ha afectado el bienestar y la cotidianidad de las personas, sino que también ha repercutido en el desarrollo productivo de las empresas [1] debido a que la mayoría de los negocios no estaban preparados para responder a una crisis como la que actualmente se está atravesando [2].

En Quito se han venido realizando capacitaciones para prevenir contagios de Covid-19 desde principios de marzo del 2020 [3], pero conforme el problema fue creciendo se implementaron una serie de resoluciones emitidas por el Comité de Operaciones de Emergencia Nacional (COE Nacional) donde varias actividades tuvieron que suspenderse, entre ellas los negocios y tiendas [4], afectando principalmente las tiendas que se encuentran dentro de los conjuntos habitacionales donde la principal fuente de ingresos es el consumo de los mismos moradores.

Con el avance de la pandemia las medidas impuestas se han flexibilizado, permitiendo a los propietarios de los negocios realizar su trabajo; sin embargo, no todos los negocios tienen las facilidades y presupuesto para implementar medidas de seguridad como: evitar que los clientes manipulen los productos, implementar medios electrónicos para realizar pagos, implementación de barreras y toma de temperatura, entre otras medidas [5] que ayuden a reducir el riesgo de contagio. Es aquí, donde la tecnología y la conectividad se convierte en un aliado clave para controlar la propagación o para mitigar y reducir impactos, sumado a una coordinación pública - privada que ayude a recuperar el desarrollo económico [6]. En este sentido, el uso de las aplicaciones móviles de reparto de comida, redes sociales, salud y GPS se han vuelto fundamentales [7] y con gran demanda, provocando que las plataformas digitales sean de uso diario por personas que no pueden o no desean salir de casa; sobre todo las más vulnerables y propensas al contagio como las personas de la tercera edad [8].

En ese contexto, se ha desarrollado una Aplicación Móvil que permite a los moradores del conjunto habitacional Luluncoto solicitar desde la comodidad de su hogar productos de la tienda "Viveres Daniela".

1.1 Objetivo general

Desarrollar una aplicación móvil para “Viveres Daniela” del conjunto habitacional Luluncoto.

1.2 Objetivos específicos

- Determinar los requerimientos de la aplicación móvil.
- Diseñar el prototipo de interfaz y la arquitectura de la aplicación móvil.
- Codificar los módulos de la aplicación móvil.
- Probar el funcionamiento de la aplicación móvil.

1.3 Alcance

El auge de las aplicaciones de tipo *delivery* en tiempos de pandemia, ha traído beneficios para los negocios y consumidores [9]. Generando ganancias para el negocio, mientras que por parte de los consumidores obtiene beneficios como la optimización de tiempo, evitando largas filas, comprar desde la comodidad del hogar, variedad de productos, comodidad de precios, entre otros [10].

Además, el Ministerio de Salud Pública (MSP) recomienda no salir de casa con frecuencia y usar aplicaciones que ayuden con la adquisición de productos de gran necesidad y medicamentos, de tal forma que se evite un contagio masivo y la propagación de la enfermedad [11].

Es por ello, que este proyecto integrador ha implementado una Aplicación Móvil que funciona como canal entre “Viveres Daniela” y sus clientes, permitiendo por un lado al negocio continuar operando y sobrevivir durante la crisis sanitaria, y por otro lado siendo una solución para los clientes que no pueden o no desean salir de sus domicilios por peligros de contagio. Por otra parte, esta propuesta garantiza la integridad de los datos, consistencia y seguridad, a través de un sistema de autenticación y de varios usuarios descritos a continuación:

El perfil cliente en la Aplicación Móvil permite:

- Registro de un usuario a través de un formulario.
- Inicio de sesión.
- Visualización de un catálogo de productos.
- Realizar un pedido.
- Notificación del pedido recibido.
- Visualización del estado del pedido.

El perfil propietario en la Aplicación Móvil permite:

- Inicio de sesión.
- Registro y modificación de productos.
- Recibir notificación de pedidos nuevos.
- Notificar pedido recibido.
- Generar reporte de ventas.

2 METODOLOGÍA

Las metodologías ágiles permiten el desarrollo de productos y servicios de calidad que satisfacen las diferentes necesidades y problemáticas, beneficiando a los equipos en el análisis y mejora del producto a lo largo de su desarrollo, obteniendo un producto competitivo durante un largo período de tiempo en el mercado [12].

Por ello el desarrollo del presente proyecto se lo ha implementado bajo *Scrum*, una metodología ágil que permite evidenciar el desarrollo del proyecto a través de *Sprints* o iteraciones que se planifican para completar cada objetivo de forma exitosa.

2.1 Metodología de Desarrollo

La metodología ágil *Scrum* plantea realizar un proceso de desarrollo de software en pequeños ciclos iterativos que pueden ir de una semana hasta un mes, período que se conoce como *Sprint* [13], agilizando el proceso de desarrollo por medio de pequeñas tareas que complementan el proyecto general, siendo una metodología útil para implementarla en grupos de trabajo pequeños. El presente proyecto ha sido implementado bajo esta metodología debido a que se adapta al contexto de la solución planteada siendo flexible, incremental y generando valor al usuario final.

2.1.1 Roles

En *Scrum* los roles tienen una participación indispensable en el proyecto, ya que definen las responsabilidades que cada miembro tiene que cumplir para asegurar que la información brindada sea correcta y que la comunicación con el cliente sea eficiente para la mejora del proyecto. Además, están comprometidos y son los responsables del éxito de cada *Sprint* [14].

Product Owner

Es el propietario del software, toma las decisiones, conoce el negocio y la visión del producto. Tiene la responsabilidad de incrementar el valor del producto y mantener el control de los entregables [15]. Este rol está representado por el Sr. Pablo Ríos propietario de “Viveres Daniela”, quien otorga la información sobre cómo se lleva a cabo el proceso de compra y venta de los productos.

Scrum Master

Es la persona encargada de que se sigan las prácticas y valores descritos en la metodología *Scrum*, guiando y orientando al equipo de trabajo hacia el éxito durante el desarrollo del proyecto [15]. Este rol lo representa la Ing. Ivonne Maldonado quien supervisa al equipo de desarrollo para que cada fase sea implementada adecuadamente.

Development Team

Son los profesionales que entregan un incremento potencial del producto al terminar cada *Sprint*. Solo los integrantes del *development team* realizan la creación de un incremento a partir del *Product Backlog* [16]. Este rol es representado por Esteban Ríos y Marvin Zambrano.

La **TABLA I** presenta la manera en la que está conformado el equipo de trabajo en el presente proyecto de titulación.

TABLA I Roles del Proyecto

| Rol | Integrante |
|------------------|----------------------------|
| Product Owner | Sr. Pablo Ríos |
| Scrum Master | Ing. Ivonne Maldonado MSc. |
| Development Team | Sr. Esteban Ríos |
| | Sr. Marvin Zambrano |

2.1.2 Artefactos

Son herramientas o elementos que garantizan el funcionamiento, transparencias y registro de la información clave dentro del proceso *Scrum*. Son los recursos necesarios para obtener calidad y productividad alta en cualquier proyecto [17].

Recopilación de Requerimientos

El levantamiento o recopilación de requerimientos hace referencia a la identificación y documentación de requerimientos de un proyecto, los cuales provienen de los usuarios, cliente y/o interesados (*stakeholders*) [18].

El equipo de desarrollo se reúne con el propietario de la tienda “Viveres Daniela” y los moradores del conjunto habitacional Luluncoto, permitiendo entender el problema a resolver, identificar y definir los requerimientos del proyecto, que se encuentran

detallados en el Manual Técnico – Sección Recopilación de Requerimientos (págs. 1 – 2)

Historias de Usuario

Es una explicación general de un requerimiento o función del proyecto, escrita desde la perspectiva del cliente o usuario final, con el propósito de articular el trabajo a realizar por el *development team* como valor específico para el producto final [19].

En las Historias de Usuario se establecen todas las funcionalidades a implementar en la aplicación móvil. La **TABLA II** presenta un ejemplo de modelo de Historias de Usuario que se han utilizado. Las 19 Historias de Usuario restantes se ubican en el Manual Técnico – Sección Historias de Usuario (págs. 3 – 9)

TABLA II: Historia de usuario Nro. 7: Asignar cantidad de productos

| Historia de Usuario | |
|--|--|
| Identificador: HU07 | Usuario: Cliente |
| Nombre Historia: Asignar cantidad de productos | |
| Prioridad en Negocio (Alta/Medio/Baja): Alta | Riesgo en Desarrollo (Alta/Medio/Baja): Alta |
| Iteración asignada: 2 | |
| Responsable (es): Esteban Ríos, Marvin Zambrano | |
| Descripción: El usuario cliente podrá elegir la cantidad de productos que va a adquirir. | |
| Observación: El usuario debe estar previamente registrado y con su sesión activa. El usuario debe haber seleccionado un producto previamente. | |

Product Backlog

Es una lista ordenada que avanza a medida que crece el producto y es la única fuente de requisitos para realizar cualquier cambio. Esta lista prioriza los requisitos y representa las expectativas del cliente con respecto a las entregas del producto final. [20]. El *Product Backlog* definido para el desarrollo de este proyecto se detalla en el Manual Técnico – Sección *Product Backlog* (pág. 10).

Sprint Backlog

Es una lista con los elementos funcionales y las diferentes tareas a ser desarrolladas día a día en cada *sprint*. Cada *sprint* cumplido es un entregable e incremento dentro del proyecto [20]. El *Sprint Backlog* definido para el desarrollo de este proyecto se detalla en el Manual Técnico – Sección *Sprint Backlog* (págs. 11 – 13).

2.2 Diseño de interfaces

Establecidos los requerimientos se inicia la construcción del diseño de las interfaces (*mockups*) de la Aplicación Móvil.

2.2.1 Marvel App

Es una herramienta que permite crear prototipos interactivos de plataformas digitales de manera rápida y sencilla, donde los desarrolladores trabajan de forma colaborativa mediante un grupo. Además, permite cargar imágenes de todo tipo y agregar la funcionalidad que se desee [21], de manera que el usuario tiene una idea de las interacciones de la implementación final. Otra ventaja de esta herramienta es que permite generar bibliotecas de diseños base para usarlas en toda el diseño de la aplicación.

Para el prototipo de la Aplicación Móvil se han establecido 21 pantallas, bajo el diseño y formato que se ilustra en la **Fig. 1**. El diseño completo definido para el desarrollo de este proyecto se encuentra en Manual Técnico – Sección Diseño de Interfaces (págs. 14 – 19).



Fig. 1: Prototipo de interfaz

2.3 Diseño de la arquitectura

La elección del patrón de arquitectura permite dar una solución eficaz a un determinado problema planteado. A continuación, se describe el modelo arquitectónico que se utilizó en el desarrollo de este proyecto.

2.3.1 Patrón arquitectónico Modelo Vista Controlador (MVC)

El patrón MVC separa la interfaz de usuario, la lógica de control y los datos en tres componentes de una aplicación en los siguientes componentes: [22]

Modelo: contiene los datos que maneja y usa la aplicación, la lógica del negocio y la persistencia.

Vista: conocida como la interfaz de usuario, es aquella que muestra la información que se envía al cliente y los métodos de interacción que tiene con este.

Controlador: es el intermediario entre el Modelo y la Vista, controla y administra el flujo de información entre ambos. Además, transforma los datos según las necesidades de cada uno.

La **Fig. 2** muestra el flujo del MVC del proyecto, en función a las herramientas utilizadas, permitiendo que el proyecto este organizado y con un desarrollo mejor estructurado.

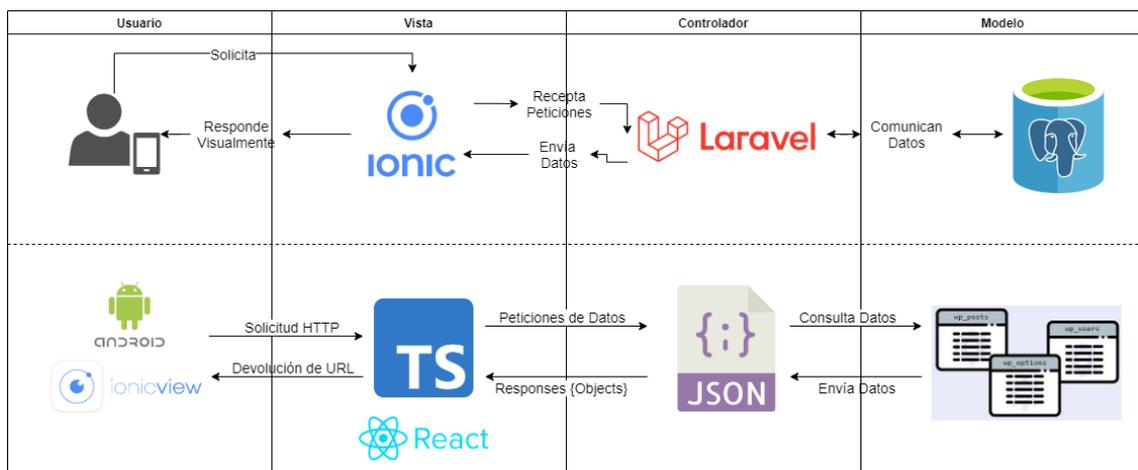


Fig. 2: Patrón arquitectónico de la Aplicación Móvil

2.4 Herramientas de desarrollo

Las herramientas de desarrollo que se usan son acordes a la arquitectura definida para la Aplicación Móvil. La **TABLA III** muestra las herramientas indispensables para avanzar en el desarrollo del proyecto.

TABLA III: Herramientas para el desarrollo de la Aplicación Móvil

| | Herramienta | Justificación |
|-------------------|--|---|
| Ionic | Marco de código abierto para el desarrollo de aplicaciones híbridas, combinando Html5, CSS y JavaScript. Ofrece una biblioteca de componente de interfaz de usuario y herramientas optimizadas para dispositivos móviles [23]. | Su implementación permite mantener una estructura ordenada a lo largo del proceso de desarrollo, como desarrollo híbrido y separar las capas en MVC. |
| Laravel | Marco de trabajo de código abierto para el desarrollo aplicaciones y servicios web con PHP 5 y PHP 7. Sus bases se centran en desarrollar código PHP de forma simple, evitando el "código espagueti". El uso de Eloquent los que permite trabajar con base de datos relacionales y el manejo de objetos directamente sin el uso de consultas SQL [24]. | Su implementación permite generar APIs de manera sencilla, obteniendo los datos del lado del servidor y presentarlos al cliente de forma eficiente. También el manejo de los datos de las bases SQL a objetos de tipo JSON. |
| PostgreSQL | Es un sistema de gestión de bases de datos relacionales, de código abierto, es uno de los gestores de datos más potente del mercado con gran accesibilidad y multiplataforma [25]. | Es fácil de usar, además de permitir el almacenamiento y la consulta de los datos de forma simple a través de relaciones entre tablas. |
| Composer | Es un gestor de dependencias de PHP que permite descargar, declarar y mantener actualizados los paquetes de software para un proyecto que se basado en PHP [26]. | Es la herramienta necesaria para mantener actualizado ciertas dependencias necesarias en el <i>back-end</i> y utilizadas por la herramienta principal Laravel |
| Postman | Herramienta que se utiliza para realizar pruebas a una API REST, | Su uso permite dos partes importantes, la primera es hacer |

| | | |
|---------------|--|--|
| | además permite monitorizarlas, escribir pruebas automatizadas, etc. [27]. | <i>testings</i> a los <i>endpoints</i> como el de autenticación y de ingreso de datos a la API y la segunda es ver como llegan o se reciben los datos al realizar una petición al API. |
| XAMPP | Es una distribución que contiene varios softwares libres como lo son el servidor web Apache, el sistema de administración de BBDD MariaDB y MySQL y los lenguajes de programación Pearl y PHP [28]. | Su uso permite levantar un servidor local que ayuda a conectar herramientas como la base de datos y la API, además de permitir mantener levantada la API para la conexión al <i>front-end</i> y manejar los datos. |
| NodeJS | Es un entorno de tiempo real con todas las dependencias necesarias para ejecutar el código escrito en JavaScript y el resultado únicamente se lo puede visualizar en un navegador como si fuera una aplicación independiente [29]. | Su uso permite instalar los paquetes necesarios para IONIC y obtener librerías específicas para el desarrollo de <i>front-end</i> . |

2.4.1 Librerías

La **TABLA IV** muestra un conjunto de bibliotecas a utilizar que complementan el desarrollo de la Aplicación Móvil para el presente proyecto de titulación.

TABLA IV: Librerías usadas en el desarrollo de la Aplicación Móvil

| | Librería | Justificación |
|--------------|--|---|
| React | Es una biblioteca de JS que permite crear interfaces de usuario de forma sencilla y con un alto nivel de interacción, se basa en componentes los cuales se actualizan y se renderizan al cambiar los datos [30]. | Es la biblioteca en la que se basa el proyecto por ser rápido y ligero, permitiendo usar componentes web en el tema desarrollo móvil. |

3 RESULTADOS Y DISCUSIÓN

En esta sección, se detalla de forma resumida la implementación y resultados obtenidos en cada una de las actividades de los 6 *Sprint*. Además, se presenta el despliegue de la Aplicación Móvil a producción.

3.1 Sprint 0. Configuración del ambiente de desarrollo

Basándose en la planificación del *Sprint Backlog*, en el *Sprint 0*, se definen las actividades para la configuración del ambiente sobre el cual se desarrolla la Aplicación Móvil.

Como resultado de este *Sprint* se tiene:

- Diseño y creación de la base de datos.
- Creación del proyecto en Ionic.
- Creación del proyecto en Laravel.
- Usuarios

3.1.1 Diseño y creación de la base de datos.

Para la creación de la Base de Datos se ha utilizado el SGBD MySQL, siguiendo los parámetros establecidos por el marco de trabajo para una rápida integración y comunicación. La Base de Datos permite administrar y almacenar toda la transaccionalidad de la información de los usuarios, productos, compras, entre otros. El diseño completo (claves primarias, foráneas y relaciones) se muestran de forma detallada en el Manual Técnico – Sección Diseño de la Base de Datos (pág. 20).

3.1.2 Creación de proyecto en Ionic.

Para crear el entorno móvil en Ionic, se ha instalado las herramientas *NodeJS* y *Visual Studio Code*. En consiguiente, se crea el proyecto con la característica esencial de trabajar con *React*, además que su estructura de directorios y archivos están bajo el patrón de arquitectura seleccionado. La **¡Error! No se encuentra el origen de la referencia.** muestra la estructura del proyecto.

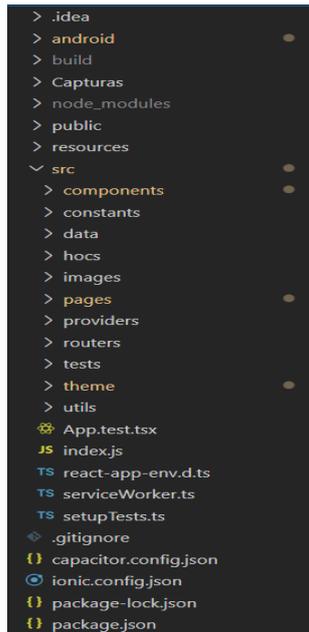


Fig. 3: Proyecto de Ionic

3.1.3 Creación del proyecto en Laravel.

Para la respectiva configuración de la API y los componentes para su funcionamiento, se procede a instalar previamente *PHP Storm*, *Composer*, *Postman* y *XAMPP*. Además, siguiendo los parámetros establecidos por Laravel y el patrón de arquitectura se procede a la creación del proyecto como se ilustra en la **Fig. 4**.

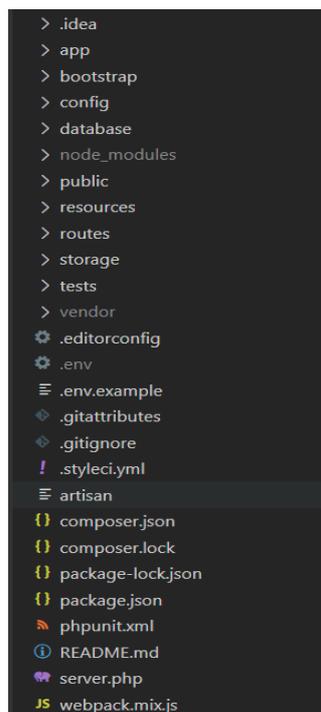


Fig. 4: Proyecto de Laravel

3.1.4 Usuarios

En la **Fig. 5** muestra los usuarios que pueden interactuar con la Aplicación Móvil, cada uno según el rol y permisos a los diferentes módulos asignados una vez realizada la respectiva autenticación.



Fig. 5: Usuarios asignados para la aplicación móvil

3.2 Sprint 1. Autenticación y Módulo Productos (Cliente)

En base a la planificación del *Sprint Backlog*, en el *Sprint 1*, se definen las actividades para la autenticación de usuarios y el módulo de productos en lo referente a la parte del usuario con rol Cliente.

Como resultado de este *Sprint* se tiene:

- Inicio de sesión.
- Registro de usuarios.
- Catálogo de productos

3.2.1 Inicio de sesión.

La **Fig. 6** muestra el *Login* de la Aplicación móvil, específicamente el formulario para iniciar sesión, para ello es necesario ingresar el correo y contraseña registrados, si alguno de los datos ingresados no es o no son correctos se emite un mensaje señalando que los datos son incorrectos como se puede observar en la **Fig. 7**. Por el contrario, si los datos son correctos se redirige al módulo correspondiente según

rol del usuario como se ilustra en la **Fig. 8**. Por otro lado, existen validaciones pertinentes de los campos que se llenan al momento de hacer un inicio de sesión como se ilustra en la **Fig. 9**. Al pasar el inicio de sesión, si el usuario no ha verificado el email se redirige a la página de notificación como se puede observar en la **Fig. 10**. En el Manual de Usuario se aprecia en detalle todo el procedimiento para el inicio de sesión. Cabe mencionar que esta funcionalidad es la misma para todos los roles de usuario.



Fig. 6: Módulo de Login

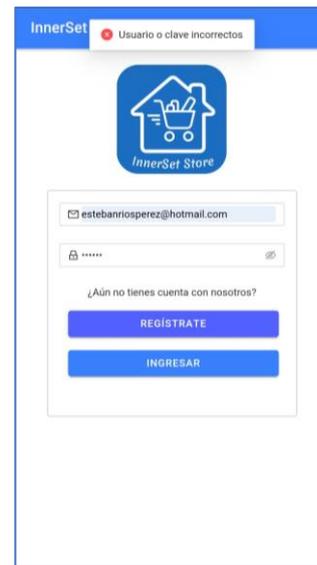


Fig. 7: Mensaje de error en el Login

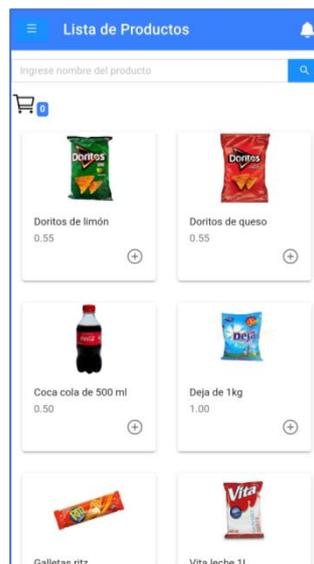


Fig. 8: Módulo de Lista de Productos (cliente)



Fig. 9: Validaciones del inicio de sesión

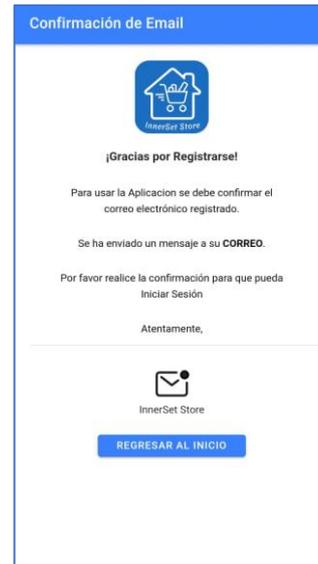


Fig. 10: Vista de confirmación de correo

3.2.2 Registro de usuarios.

La **Fig. 11**, muestra el formulario para el registro de datos personales de un nuevo usuario: nombre, apellido, número de casa, correo electrónico y contraseña. Una vez que los datos son validados y enviados, es indispensable activar la cuenta, para ello se requiere la verificación del correo electrónico registrado, el usuario recibirá un correo de verificación como se puede observar en la **Fig. 12**. Por otro lado, existen validaciones respectivas de los campos para que un usuario nuevo se registre en la aplicación, todos los campos son obligatorios como se muestra en la **Fig. 13**. En el Manual de Usuario se aprecia detalladamente el procedimiento, interfaces y validaciones respectivas para este módulo.

Registro

InnerSet Store

Nombre Completo

Número de Casa

Correo

Contraseña

Confirma tu contraseña

REGISTRAR

Fig. 11: Módulo de Registro de nuevos usuarios

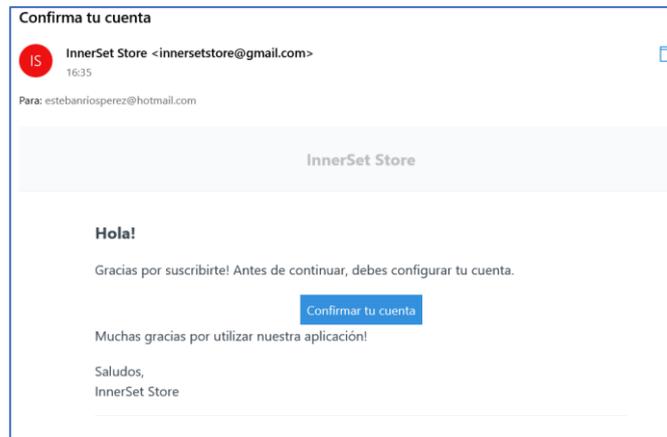


Fig. 12: Correo de verificación de cuenta

Registro

InnerSet Store

Nombre Completo
Ingrese su nombre

Número de Casa
Ingrese su número de casa (1-60)

Correo
Ingrese su correo

Contraseña
Ingrese su contraseña

Confirma tu contraseña
Confirme su contraseña

REGISTRAR

Fig. 13: Validación de campos para el registro

3.2.3 Catálogo de productos.

La Fig. 14, muestra el catálogo de productos, que contiene una lista de todos los productos que el usuario propietario ha ingresado. Además, cuenta con una sección de búsqueda que facilita el encontrar un producto en específico. En el Manual de Usuario se puede apreciar detalladamente el procedimiento, interfaces y validaciones respectivas para este módulo.

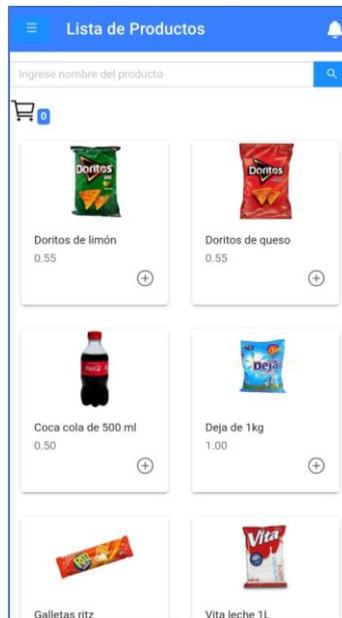


Fig. 14: Catálogo de productos (cliente)

3.3 Sprint 2. Módulo Compras (Carrito)

En base a la planificación del *Sprint Backlog*, en el *Sprint 2*, se definen las tareas para el módulo compras es decir lo referente al carrito de compras.

Como resultado de este *Sprint* se tiene:

- Carrito de compras y tipo de entrega.

3.3.1 Carrito de compras y tipo de entrega

Una vez que el cliente haya seleccionado los productos para su pedido, en el módulo “Carrito de Compra” se puede observar el resumen del pedido a realizar, aquí existe la posibilidad de aumentar o disminuir la cantidad de los productos seleccionados, ver el subtotal de la compra, así como seleccionar el “Tipo de entrega” que desea, con lo que la aplicación incluye o no el valor por entrega a domicilio. Como se puede observar en la Fig. 15.

La Fig. 16 y Fig. 17 muestra cómo se efectúa la actualización de los precios y la cantidad de cada uno de los productos que ha seleccionado el cliente. Además, La Fig. 18 muestra la eliminación respectiva del producto seleccionado dentro del carrito de compras. Por otra parte, la Fig. 19. Indica la validación o restricción de no enviar el pedido mientras el carrito este vacío. En el Manual de Usuario se puede apreciar detalladamente el procedimiento, interfaces y validaciones respectivas para este módulo.

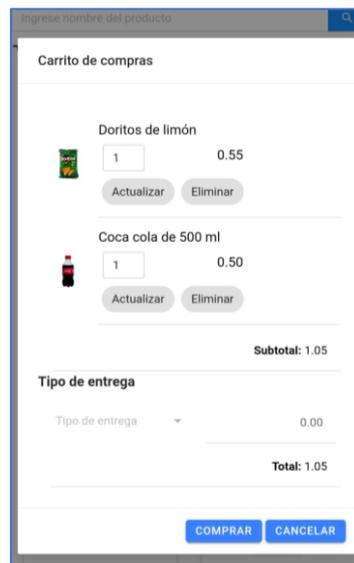


Fig. 15: Carrito de compras

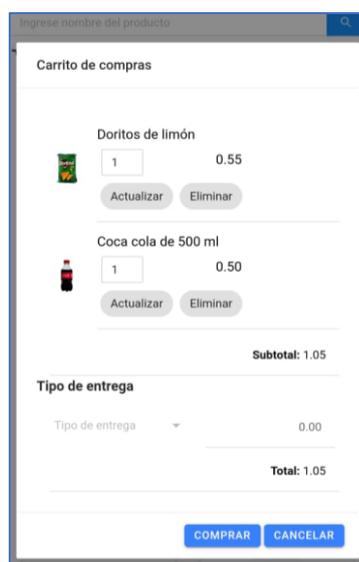


Fig. 16: Estado inicial cantidad del producto

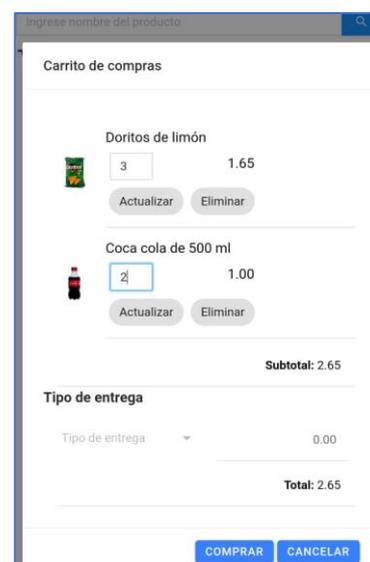


Fig. 17: Cambio de cantidad

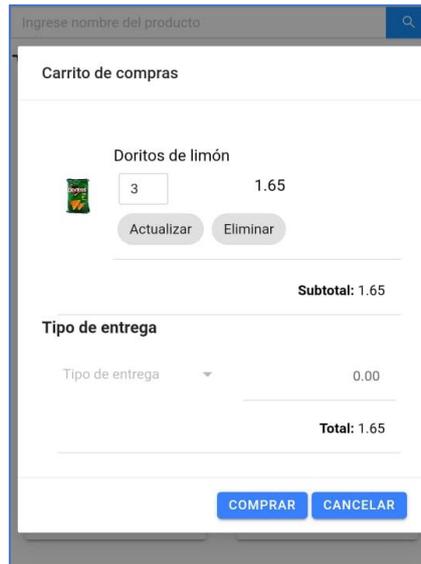


Fig. 18: Eliminación de producto en el carrito

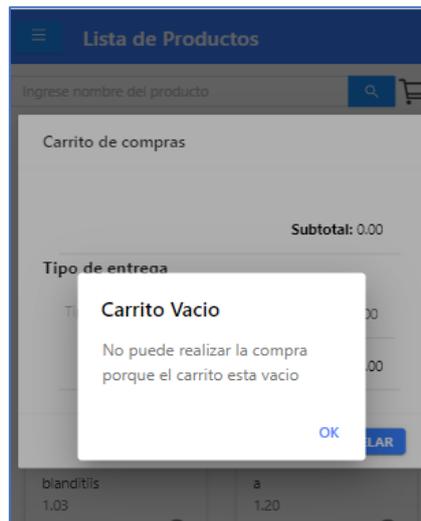


Fig. 19: Restricción por carrito vacío

3.4 Sprint 3. Módulo Pedidos (Cliente)

En base a la planificación del *Sprint Backlog*, en el *Sprint 3*, se definen las tareas para el módulo pedidos en lo referente a la parte del usuario con rol Cliente.

Como resultado de este *Sprint* se tiene:

- Notificación y visualización de pedido recibido.
- Barra de progreso de pedido.
- Pedidos por completar.

3.4.1 Notificación y visualización de pedido recibido.

Cuando el propietario marca un pedido como recibido, el cliente recibe la notificación por medio de una “campanita” tal como se puede observar en la **Fig. 20**. Gracias a este icono el cliente puede visualizar que su pedido ha sido recibido, **Fig. 21**, lo que significa que lo están preparando para ser enviado. En el Manual de Usuario se puede apreciar detalladamente el procedimiento, interfaces y validaciones respectivas para este módulo.

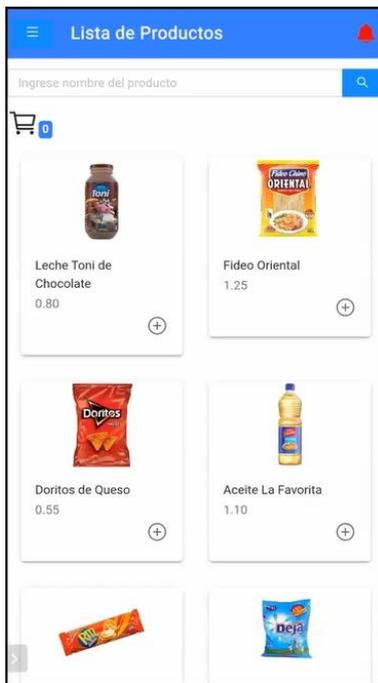


Fig. 20: Alerta nueva notificación

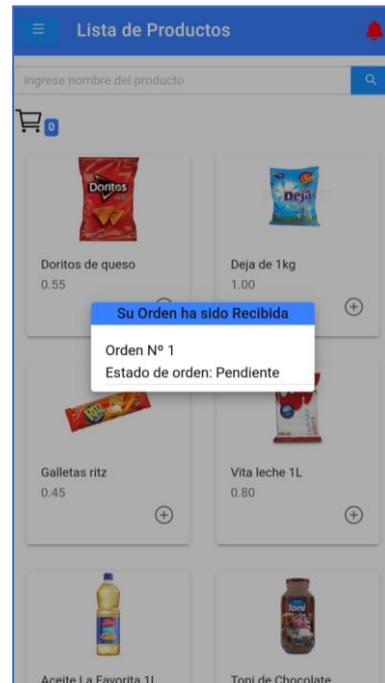


Fig. 21: Información de notificación

3.4.2 Barra de progreso de pedido.

El cliente puede observar el progreso de su pedido mediante una barra de progreso, tal como se muestra en la **Fig. 22**. Esta barra va cambiando en función a como se encuentra el pedido, es decir: pendiente, realizado, en camino (si es a domicilio) y entregado. En el Manual de Usuario se puede apreciar detalladamente el procedimiento, interfaces y validaciones respectivas para este módulo.



Fig. 22: Barra de progreso (Estado del pedido)

3.4.3 Pedidos por completar.

La **Fig. 23** muestra el listado de pedidos que aún se encuentran en estado pendiente del cliente autenticado, en cada pedido pendiente de entrega se puede visualizar la cantidad, el producto, su precio, el subtotal de la compra, el costo de entrega a domicilio y el total como se muestra en la **Fig. 24**. En el Manual de Usuario se puede apreciar detalladamente el procedimiento, interfaces y validaciones respectivas para este módulo.

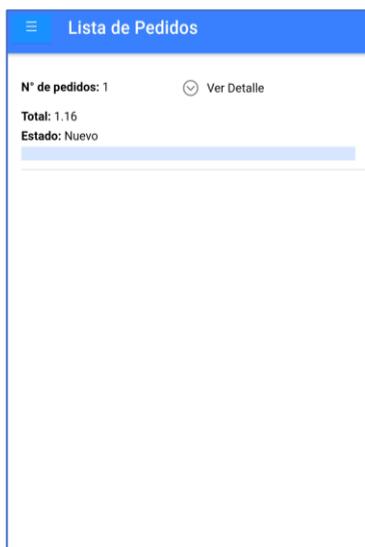


Fig. 23: Lista de pedidos pendientes



Fig. 24: Detalle del pedido

3.5 Sprint 4. Módulo Productos y Módulo Pedidos (Propietario)

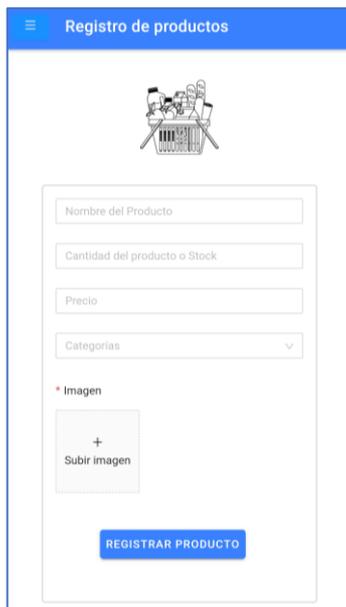
En base a la planificación del *Sprint Backlog*, en el *Sprint 4*, se definen las tareas para los módulos productos y pedidos en lo referente a la parte del usuario con rol Propietario.

Como resultado de este *Sprint* se tiene:

- Registro y actualización de Productos
- Visualización y Notificación de pedido recibido

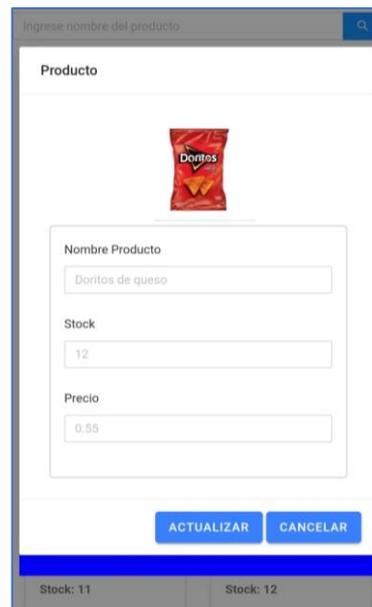
3.5.1 Registro y actualización de productos.

La **Fig. 25** se puede observar la interfaz con el formulario para registrar un nuevo producto en particular, cada dato a ser ingresado se encuentra validado como se muestra en la **Fig. 26**. El usuario propietario puede modificar los productos ya registrados. En el Manual de Usuario se puede apreciar detalladamente el procedimiento, interfaces y validaciones respectivas para este módulo.



The screenshot shows a mobile application interface titled "Registro de productos". At the top, there is a blue header with a menu icon and the title. Below the header is a central illustration of a shopping cart. The main area contains a form with the following fields: "Nombre del Producto", "Cantidad del producto o Stock", "Precio", and "Categorías" (a dropdown menu). Below these is a section for "Imagen" with a plus sign and the text "Subir imagen". At the bottom of the form is a blue button labeled "REGISTRAR PRODUCTO".

Fig. 25: Módulo Registro de productos



The screenshot shows a mobile application interface titled "Ingrese nombre del producto". At the top, there is a search bar. Below it is a section titled "Producto" with a small image of a Doritos bag. The main area contains a form with the following fields: "Nombre Producto" (with "Doritos de queso" entered), "Stock" (with "12" entered), and "Precio" (with "0.55" entered). At the bottom of the form are two blue buttons: "ACTUALIZAR" and "CANCELAR". Below the form, there are two small boxes showing "Stock: 11" and "Stock: 12".

Fig. 26: Módulo Actualizar un producto

3.5.2 Visualización y notificación de nuevo pedido.

Cuando un cliente realiza un pedido, el propietario recibe la notificación del mismo por medio de una “campanita” tal como se puede observar en la **Fig. 27**. Gracias a este icono el propietario puede visualizar todos los pedidos nuevos y estos se notifican a los clientes cuando cambie de estado a pendiente, tal como se muestra en la **Fig. 28**. Por otro lado, la **Fig. 29** muestra la visualización de los pedidos que llegan al propietario y los datos de quien pertenece dicho pedido en especial el estado en el que se encuentra. Finalmente, la **Fig. 30** muestra la pantalla donde el propietario puede cambiar el estado del pedido. En el Manual de Usuario se puede apreciar detalladamente el procedimiento, interfaces y validaciones respectivas para este módulo.

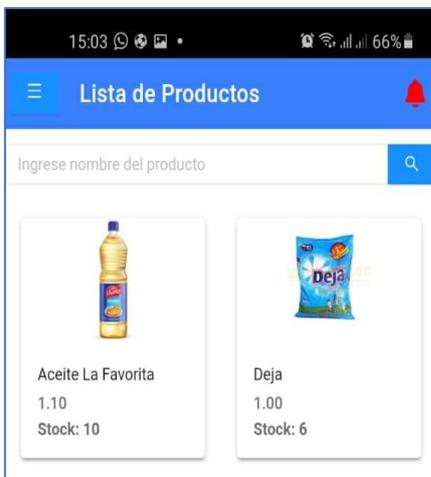


Fig. 27: Alerta de nuevo pedido

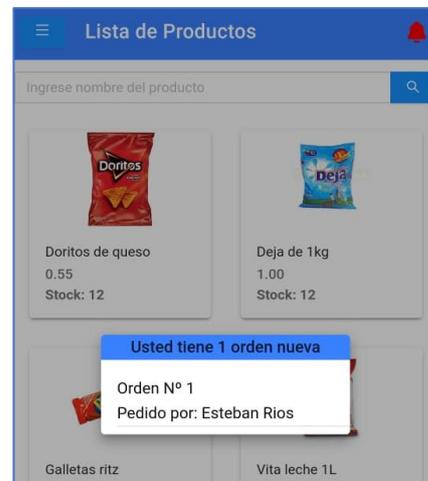


Fig. 28: Información de notificación

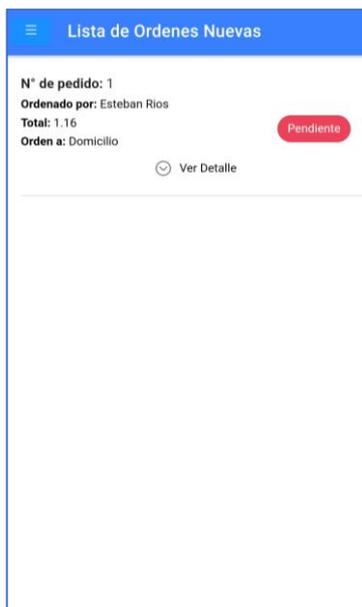


Fig. 29: Lista ordenes nuevas

Fig. 30: Detalle pedido y cambio de estado

3.6 Sprint 5. Módulo Reportes

En base a la planificación del *Sprint Backlog*, en el *Sprint 5*, se definen las tareas que permiten la realización de reportes.

Como resultado de este *Sprint* se tiene:

- Reporte de ventas por cliente.
- Reporte de productos vendidos.
- Reporte de stock.

3.6.1 Reporte de ventas por Cliente.

El propietario puede obtener un reporte de ventas por cliente en un periodo de fechas, tal como se indica en la

Fig. 31. Se genera un reporte en formato pdf y automáticamente se descarga, tal como se muestra en la **Fig. 32.** En el Manual de Usuario se puede apreciar detalladamente el procedimiento, interfaces y validaciones respectivas para este módulo.

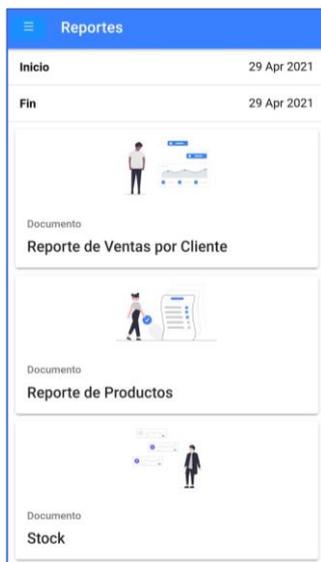


Fig. 31: Reportes (Ventas por cliente)

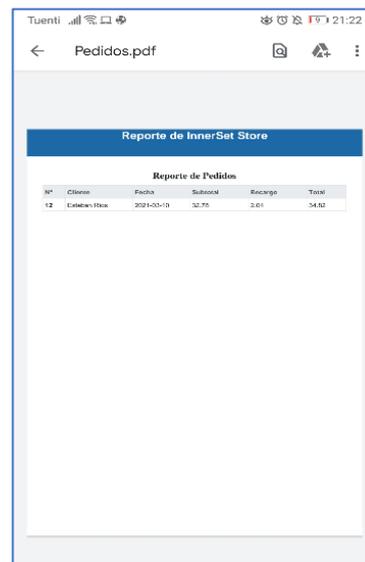


Fig. 32: Descarga archivo PDF

3.6.2 Reporte de Productos.

El propietario puede obtener un reporte donde puede visualizar los productos más vendidos en función a un periodo de fechas, tal como se indica en la **Fig. 33.** Se genera un reporte en formato pdf y automáticamente se descarga, tal como se muestra en la

Fig. 34. En el Manual de Usuario se puede apreciar detalladamente el procedimiento, interfaces y validaciones respectivas para este módulo.



Fig. 33: Reportes (Productos más vendidos)



Fig. 34: Descarga archivo PDF

3.6.3 Reporte de Stock.

El propietario puede obtener un reporte del stock de los productos, tal como se muestra en la **Fig. 35**. Se genera un reporte en formato pdf y automáticamente se descarga, tal como se muestra en la **Fig. 36**. En el Manual de Usuario se puede apreciar detalladamente el procedimiento, interfaces y validaciones respectivas para este módulo.

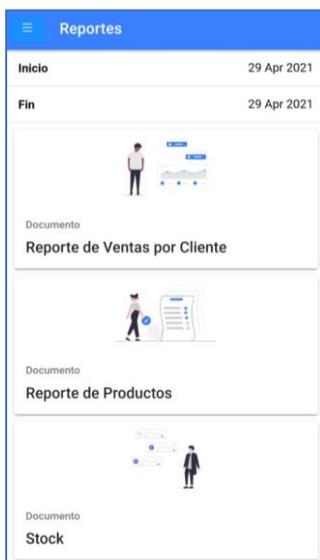


Fig. 35: Reportes (Stock)

Fig. 36: Descarga archivo PDF

3.7 Sprint 6. Pruebas y Despliegue

En base a la planificación del *Sprint Backlog*, en el *Sprint 6*, se definen las tareas que permiten la realización de las diferentes pruebas a la Aplicación Móvil y el despliegue a producción.

Como resultado de este *Sprint* se tiene:

- Pruebas unitarias.
- Pruebas de carga al API.
- Pruebas de compatibilidad.
- Pruebas de aceptación.
- Despliegue de la Aplicación Móvil a producción.

3.7.1 Pruebas unitarias.

Las pruebas unitarias son parte de las actividades que se deben llevar a cabo al hacer uso de metodologías ágiles ya que permiten detectar errores a tiempo, de forma que se los corrijan sin necesidad de tener que volver al principio, estas consisten en comprobar que trozos de código estén funcionando como se espera, específicamente para cubrir todos los requisitos y verificar sus resultados desde las primeras etapas [31].

La **Fig. 37** muestra la estructura de una prueba unitaria para la respectiva ejecución y validación en la actualización de la cantidad, precio de cada producto y el total del pedido en el carrito de compras. Las pruebas unitarias restantes se encuentran en el Manual Técnico – Sección Pruebas Unitarias (págs. 21 – 23)

```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

✓ Actualizar la cantidad y precio del carrito (15ms)

console.log src/tests/App.test.js:10
  nuevaCantidad 3

console.log src/tests/App.test.js:18
  cart [
    { cartId: 0, cartPrice: 9, cartQuantity: 3 },
    { cartId: 1, cartPrice: 12, cartQuantity: 3 },
    { cartId: 2, cartPrice: 21, cartQuantity: 3 },
    { cartId: 3, cartPrice: 3, cartQuantity: 3 }
  ]

Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
Snapshots:  0 total
Time:        36.996s, estimated 112s
Ran all test suites related to changed files.

```

Fig. 37: Prueba unitaria del componente de actualizar cantidad y precio en el carrito de compras

3.7.2 Pruebas de carga al API

El proyecto tiene dos partes, una que es el API de donde se hace el uso de peticiones para obtener los datos y la aplicación móvil que envía las peticiones del usuario, y las muestran en la interfaz. De todo este proceso es importante saber cuánta carga resiste o soportar hasta alcanzar el máximo sin afectar el rendimiento de ahí la importancia de realizar pruebas de carga al API [32].

En la **Fig. 38** se muestra el resultado de realizar la prueba de carga respectiva a *endpoint* que devuelve los datos de los productos, se establecen 100 conexiones y todas son aceptadas. Las restantes pruebas de cargas realizadas se encuentran en el Manual Técnico – Sección Pruebas de Carga (pág. 23).

Reporte resumen

Nombre:

Comentarios:

Escribir todos los datos a Archivo

Nombre de archivo: Log/Mostrar sólo: Escribir en Log Solo Errores Éxitos

| Etiqueta | # Muestras | Media | Min | Máx | Desv. Estándar | % Error | Rendimiento | Kb/sec | Sent KB/sec | Media de Byt... |
|----------------|------------|--------|------|--------|----------------|---------|-------------|--------|-------------|-----------------|
| Petición: HTTP | 100 | 121212 | 4252 | 251497 | 73087,07 | 0,00% | 23,7/min | 0,69 | 0,05 | 1792,0 |
| Total | 100 | 121212 | 4252 | 251497 | 73087,07 | 0,00% | 23,7/min | 0,69 | 0,05 | 1792,0 |

Fig. 38: Prueba de carga *endpoint* de productos

3.7.3 Pruebas de compatibilidad

Cuando se desarrolla aplicaciones móviles se debe garantizar su correcto funcionamiento en distintos dispositivos, de ahí la necesidad de realizar pruebas de compatibilidad que permitan evaluar la aplicación sobre diferentes ambientes [33]. Para llevar a cabo este tipo de pruebas se recomienda la estrategia de utilizar dispositivos

reales para lograr definir la versión del sistema operativo que soporta la aplicación, comprobando que se vea bien y funciona sin problemas en diferentes resoluciones y tamaños de pantalla [34].

La **TABLA V** presenta la lista de dispositivos móviles en los cuales se han llevado a cabo este tipo de pruebas, verificando que la información, contenido multimedia y notificaciones se presenten de forma homogénea. Por otra parte, las **Fig. 39**, **Fig. 40** y **Fig. 41** muestran los resultados tras la ejecución de la prueba y sus resultados según cada dispositivo móvil. Las pruebas de compatibilidad restantes se encuentran en el Manual Técnico – Sección Pruebas de Compatibilidad (Págs. 24 – 34)

TABLA V Especificaciones de los dispositivos para las pruebas de compatibilidad

| MARCA | MODELO | VERSION DE ANDROID |
|---------------------|----------|--------------------|
| Huawei PSmart | FIG-LX3 | 9.0.0 |
| Samsung Galaxy A20s | SM-A207M | 10.0.0 |
| Samsung Galaxy A5 | SM-A520F | 8.0.0 |

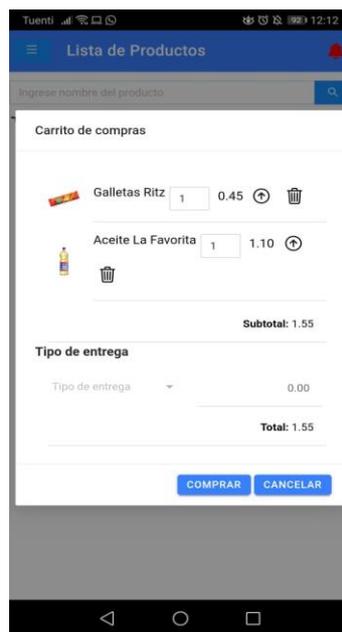


Fig. 39: Módulo del carrito de compra en el *Huawei PSmart*

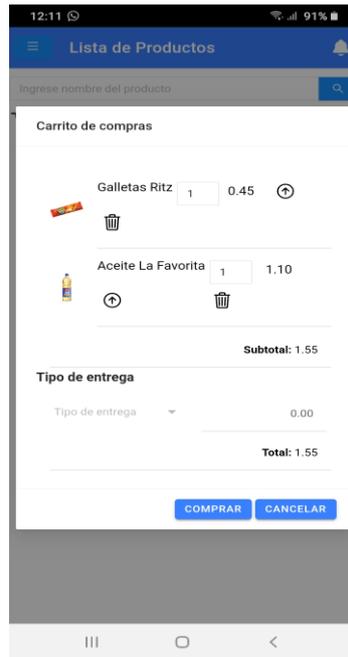


Fig. 40: Módulo del carrito de compra en el *Samsung Galaxy A20s*

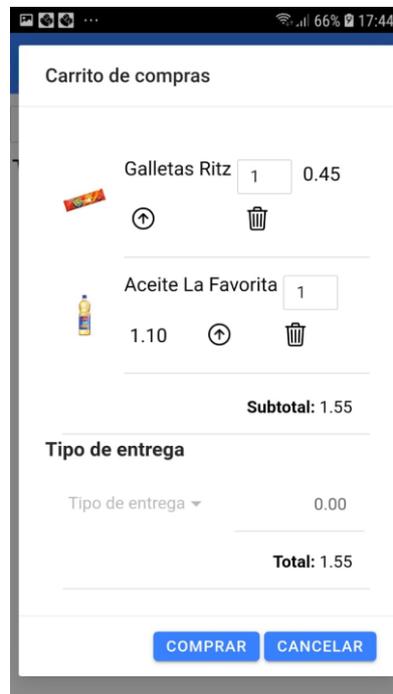


Fig. 41: Módulo de carrito de compras en el *Samsung A5*

3.7.4 Pruebas de aceptación

A diferencia del resto de pruebas que se llevan a cabo en el desarrollo de software, las pruebas de aceptación no tienen el objetivo de encontrar defectos, más bien su misión en validar el cumplimiento del propósito establecido antes de liberar la aplicación al

ambiente de producción. Este tipo de pruebas son realizadas por los usuarios quienes verifican los requerimientos y procesos de negocio para determinar si se debe o no aceptar la aplicación [35].

La **TABLA VI** detalla un modelo de las pruebas de aceptación que se han realizado. Las restantes pruebas de aceptación realizadas se encuentran en el Manual Técnico – Sección Pruebas de Aceptación (págs. 35 – 44).

TABLA VI: Prueba de Aceptación Nro. 2: Iniciar sesión

| Prueba de Aceptación | |
|--|--|
| Identificador: PA02 | Identificador Historia de Usuario: HU02 |
| Nombre de la Prueba de Aceptación: Iniciar sesión | |
| Descripción: El usuario cliente y el usuario propietario pueden iniciar la sesión llenando los campos usuario y contraseña. | |
| Pasos de ejecución: Ingresar a la aplicación Ingresar el correo Ingresar la contraseña Dar click en el botón <i>Ingresar</i> | |
| Resultado deseado: La Aplicación Móvil permite el inicio de sesión de un usuario y lo redirige a los módulos dependiendo de su rol. El usuario cliente una vez iniciada sesión puede visualizar los productos y realizar la respectiva compra. El usuario propietario una vez iniciada sesión puede ofertar productos y recibir pedidos. | |
| Evaluación de la prueba: Resultado exitoso. La Aplicación Móvil concede el inicio de sesión. Aprobación del cliente en un 100%. | |

3.7.5 Despliegue de la Aplicación Móvil a Producción

El proceso para desplegar la Aplicación Móvil dentro de *Google Play Store* se detalla en el Manual de Instalación - Sección Despliegue de la aplicación móvil. La **Fig. 42** ilustra de manera general el procedimiento para desplegar la aplicación en su primera versión.

Producción

Crea y gestiona versiones de producción para que tu aplicación esté disponible para todos los usuarios de los países que elijas. [Más información](#)

Resumen del canal

Activo · Versión 1 (1.0) en revisión · 1 país o región · 0 descargas

Panel de control de la versión **Versiones** Países/regiones

Versiones

1 (1.0)

🕒 En revisión · 1 código de versión

Mostrar resumen ▾

Fig. 42: Despliegue de la Aplicación Móvil a producción

4 CONCLUSIONES Y RECOMENDACIONES

A continuación, se presentan las conclusiones y recomendaciones que se han obtenido a lo largo del desarrollo del presente trabajo de titulación.

4.1 Conclusiones

- La Aplicación Móvil cumple con el alcance y objetivos planteados, permitiendo a los moradores del conjunto habitacional Luluncoto solicitar desde la comodidad de sus hogares los productos de la tienda “Viveres Daniela”; funcionando como una alternativa diferente no solo de compra sino de venta y atención a los clientes.
- Una adecuada recopilación de requerimientos por medio de reuniones con el dueño de “Viveres Daniela” en el inicio del proyecto, marca significativamente el desarrollo de la Aplicación Móvil, debido a que gracias a ello se ha logrado determinar a partir de las necesidades planteadas, objetivos, tiempos y experticia de los desarrolladores las herramientas adecuadas para el proyecto.
- El aplicar la metodología ágil *Scrum* para desarrollar la Aplicación Móvil ha facilitado identificar y ejecutar cada etapa que se involucra en el ciclo de vida de desarrollo, definir de manera clara los interesados y sus roles dentro del proyecto y generar entregables acorde a las necesidades del cliente en periodos cortos; ayudando así, no solo a la detección rápida de errores sino al cumplimiento de los objetivos planteados en los tiempos establecidos.
- La implementación de la Aplicación Móvil mediante el uso del patrón arquitectónico MVC permite fácil mantenimiento y mayor escalabilidad a momento de incluir nuevos módulos o funcionalidades evitando el uso de nuevas configuraciones.
- El uso de MySQL como SGBD para el desarrollo del proyecto mejora el rendimiento, escalabilidad e integración con las diferentes herramientas utilizadas.
- Gracias a una buena selección de los *Frameworks* (Ionic y Laravel) se ha logrado el desarrollo de manera estructurada, con un orden adecuado y con la facilidad de reutilizar componentes y funcionalidades.
- La etapa de pruebas ha permitido validar la Aplicación Móvil de manera que se asegura que se han cubierto todos los requerimientos y expectativas del cliente.

4.2 Recomendaciones

- Se recomienda validar periódicamente la existencia de actualizaciones en los *Frameworks* que se han utilizado para el desarrollo de la Aplicación Móvil, con el objetivo de mantener librerías, interfaces, componentes, archivos de configuración, entre otros, evitando los errores y asegurando que los resultados de retorno sean íntegros.
- Si se deseara realizar una nueva implementación o actualización se recomienda seguir el modelo arquitectónico establecido MVC de manera que se agregue nuevas funcionalidades y/o un mantenimiento rápido y seguro que conserve la distinción de cada una de las capas.

5 REFERENCIAS BIBLIOGRÁFICAS

- [1] A. Oneto, «COVID-19: Continuidad del negocio, gestión de crisis y gobierno corporativo,» SCIOTECA, 23 Junio 2020. [En línea]. Available: <http://scioteca.caf.com/handle/123456789/1596>. [Último acceso: 16 Julio 2020].
- [2] Y. García, «Plan de Continuidad de Negocio frente a pandemia de COVID-19,» Universidad Piloto de Colombia, 02 Junio 2020. [En línea]. Available: <http://repository.unipiloto.edu.co/handle/20.500.12277/7659>. [Último acceso: 16 Julio 2020].
- [3] M. d. Quito, «Comerciantes de los mercados continúan capacitándose para prevenir el Covid 19,» Quito Informa, 09 Marzo 2020. [En línea]. Available: <http://www.quitoinforma.gob.ec/2020/03/09/comerciantes-de-los-mercados-continuan-capacitandose-para-prevenir-el-covid-19/>. [Último acceso: 15 Julio 2020].
- [4] C. Nacional, «Resoluciones de la sesión del 15 de marzo de 2020,» COE Nacional, 15 Marzo 2020. [En línea]. Available: <https://www.gestionderiesgos.gob.ec/wp-content/uploads/2020/03/Resoluciones-COE-Nacional-15-de-marzo-2020.pdf>. [Último acceso: 15 Julio 2020].
- [5] M. d. Quito, «Guía de normas de bioseguridad para pequeños negocios y comercios.,» Municipio de Quito, [En línea]. Available: https://www.quito.gob.ec/documents/guias/guia_PequeñosNegocios_comercios.pdf. [Último acceso: 15 Julio 2020].
- [6] A. G. Zaballos, «¿Cómo la tecnología y la conectividad pueden ayudar a enfrentar la crisis causada por el coronavirus?,» BID, 25 Marzo 2020. [En línea]. Available: <https://blogs.iadb.org/innovacion/es/tecnologia-y-conectividad-enfrentar-crisis-coronavirus/>. [Último acceso: 16 Julio 2020].
- [7] HOSTELVENDING.COM, «Aplicaciones móviles para frenar la propagación del Covid-19: ¿una oportunidad para las compañías especializadas en software?,» HOSTELVENDING, 12 Mayo 2020. [En línea]. Available:

<https://www.hostelvending.com/noticias-vending/aplicaciones-moviles-para-frenar-la-propagacion-del-covid-19--una-oportunidad-para->. [Último acceso: 16 Julio 2020].

- [8] E. Tapia y V. Silva, El Comercio, 30 Marzo 2020. [En línea]. Available: <https://www.elcomercio.com/actualidad/apps-delivery-demanda-demora-pedidos.html>. [Último acceso: 16 Julio 2020].
- [9] F. d. Rosa, «Este artículo fue hecho gracias al esfuerzo de trabajo de un periodista profesional. Reconocelos compartiendo esta nota mediante este link: <https://www.cronista.com/pyme/negocios/Aplicaciones-moviles-el-aliado-para-vender-con-delivery-20200504-0001.html> C,» El Cronista PYME, 04 Mayo 2020. [En línea]. Available: <https://www.cronista.com/pyme/negocios/Aplicaciones-moviles-el-aliado-para-vender-con-delivery-20200504-0001.html>. [Último acceso: 16 Julio 2020].
- [10] A. Devincenzi, «Coronavirus que medias implemntan los deliverys para las entregas a domicilio,» Cronista Apertura y Negocios, 18 Marzo 2022. [En línea]. Available: <https://www.cronista.com/apertura-negocio/empresas/Coronavirus-que-medidas-implementan-los-delivery-para-realizar-las-entregas-y-evitar-contagios-20200317-0002.html>. [Último acceso: 16 Julio 2020].
- [11] Ambientum, «Tecnología móvil para controlar el COVID-19,» Ambientum - Tecnología, 6 Abril 2020. [En línea]. Available: <https://www.ambientum.com/ambientum/tecnologia/tecnologia-movil-para-controlar-el-covid-19.asp>. [Último acceso: 16 Julio 2020].
- [12] L. Gonçalvez, «Qué es la metodología Ágil,» ADAPT METHodology, 30 octubre 2020. [En línea]. Available: <https://adaptmethodology.com/es/que-es-la-metodologia-agil/>. [Último acceso: 22 Noviembre 2020].
- [13] M. T. Gallego, «Metodología Scrum,» [En línea]. Available: <http://openaccess.uoc.edu/webapps/o2/bitstream/10609/17885/1/mtrigasTFC0612memoria.pdf>. [Último acceso: 20 Agosto 2020].
- [14] Y. L. C. Pineda, «Qué es SCRUM y los roles en SCRUM,» Platzi, 2017. [En línea]. Available: <https://platzi.com/blog/que-es-scrum-y-los-roles-en-scrum/>. [Último acceso: 25 noviembre 2020].

- [15] M. T. Gallego, «Metodología Scrum,» 2012. [En línea]. Available: <http://openaccess.uoc.edu/webapps/o2/bitstream/10609/17885/1/mtrigasTFC0612memoria.pdf>. [Último acceso: 07 Julio 2020].
- [16] J. Roche, «Scrum: roles y responsabilidades,» Deloitte, [En línea]. Available: <https://www2.deloitte.com/es/es/pages/technology/articles/roles-y-responsabilidades-scrum.html>. [Último acceso: 29 noviembre 2020].
- [17] M. Bara, «Tendencias & Innovación Roles, Eventos y Artefactos en la Metodología Scrum,» OBS Business School, [En línea]. Available: <https://obsbusiness.school/es/blog-investigacion/project-management/roles-eventos-y-artefactos-en-la-metodologia-scrum>. [Último acceso: 22 Noviembre 2020].
- [18] PMOinformatica.com, «7 Técnicas de levantamiento de requerimientos software,» PMOinformatica.com - La oficina de proyectos de informática, 3 agosto 2016. [En línea]. Available: <http://www.pmoinformatica.com/2016/08/tecnicas-levantamiento-requerimientos.html>. [Último acceso: 23 Noviembre 2020].
- [19] M. Rehkopf, «Agile - Historias de usuario,» ATlassian Agile Couch, [En línea]. Available: <https://www.atlassian.com/es/agile/project-management/user-stories>. [Último acceso: 25 noviembre 2020].
- [20] EALDE Business School, «Dirección de Proyectos,» EALDE Business School, 27 Agosto 2019. [En línea]. Available: <https://www.ealde.es/product-backlog-sprint-backlog/#:~:text=El%20Product%20Backlog%20o%20pila,y%20el%20entorno%20del%20proyecto..> [Último acceso: 10 Diciembre 2020].
- [21] J. Pérez, «Marvel App en la creación de prototipos,» IDA blog, 04 Mayo 2017. [En línea]. Available: <https://blog.ida.cl/disenio/ventajas-marvel-app-prototipos/>. [Último acceso: 13 Diciembre 2020].
- [22] Universidad de Alicante, «Servicio de Informática ASP:NET MVC 3 Framework,» Universidad de Alicante, 20 Agosto 2012. [En línea]. Available: <https://si.ua.es/es/documentacion/asp-net-mvc-3/1-dia/modelo-vista-controlador-mvc.html>. [Último acceso: 13 Diciembre 2020].

- [23] E. A., «Ionic Framework,» OpenWebinars, 13 Junio 2016. [En línea]. Available: <https://openwebinars.net/blog/ionic-framework-que-es/>. [Último acceso: 22 Diciembre 2020].
- [24] K. Palomares, «Que es Laravel,» Kiko Palomares, 22 Octubre 2019. [En línea]. Available: <https://www.kikopalomares.com/blog/que-es-laravel-y-para-que-sirve-frameworks-de-php>. [Último acceso: 22 Diciembre 2020].
- [25] HostingPedia, «Alogamiento Web - PostgreSQL,» HostingPedia, 07 Febrero 2019. [En línea]. Available: <https://hostingpedia.net/postgresql.html>. [Último acceso: 01 Mayo 2021].
- [26] Desarrolloweb.com, «Manuales - Tutorial de Composer,» desarrolloweb.com, 24 enero 2020. [En línea]. Available: <https://desarrolloweb.com/articulos/composer-gestor-dependencias-para-php.html>. [Último acceso: 25 Febrero 2021].
- [27] A. López, «Desarrollo Web - Que es Postman y para que sirve,» Open Webinars, 03 junio 2019. [En línea]. Available: <https://openwebinars.net/blog/que-es-postman/>. [Último acceso: 25 Febrero 2021].
- [28] IONOS, «Herramientas - Instala tu servidor local XAMPP,» Digital Guide IONOS by 1&1, 03 Septiembre 2019. [En línea]. Available: <https://www.ionos.es/digitalguide/servidores/herramientas/instala-tu-servidor-local-xampp-en-unos-pocos-pasos/>. [Último acceso: 25 Febrero 2021].
- [29] J. Lucas, «Desarrollo web - Qué es NodeJS y para que sirve,» Open Webinars, 04 Septiembre 2019. [En línea]. Available: <https://openwebinars.net/blog/que-es-nodejs/>. [Último acceso: 25 Febrero 2021].
- [30] React Org, «React,» React.org - Facebook Inc., 2021. [En línea]. Available: <https://es.reactjs.org>. [Último acceso: 25 Febrero 2021].
- [31] APIUMHUB, «Beneficios de las Pruebas Unitarias,» TENDENCIAS TECNOLÓGICAS, 24 Agosto 2017. [En línea]. Available: <https://apiumhub.com/es/tech-blog-barcelona/beneficios-de-las-pruebas-unitarias/>. [Último acceso: 05 Marzo 2021].

- [32] S. S. Avalos, «Pruebas de carga vs Pruebas de estrés,» Somos SPNT, 26 Septiembre 2019. [En línea]. Available: <https://sospnt.com/blog/103-load-testing-vs-stress-testing>. [Último acceso: 16 Marzo 2021].
- [33] Green SQA, «Servicio de Pruebas para Aplicaciones Móviles,» Green SQA, [En línea]. Available: <https://greensqa.com/servicio-de-pruebas-para-aplicaciones-moviles/>. [Último acceso: 05 Marzo 2021].
- [34] Tobar Pia., «¿Cómo realizar pruebas de software en una aplicación móvil?,» Abstracta Chile, [En línea]. Available: <https://cl.abstracta.us/blog/pruebas-software-aplicacion-movil/>. [Último acceso: 05 Marzo 2021].
- [35] S. M. Velásquez, D. E. Monsalve Sossa, M. E. Zapata, M. E. Gómez Adasme y J. P. Ríos, «Pruebas a aplicaciones móviles: avances y retos,» *Lámpsakos*, n° 21, pp. 39-50, 2019.
- [36] M. Alcaraz, «Coronavirus: Cómo evitar contagios al hacer la compra,» ABC, 27 Marzo 2020. [En línea]. Available: https://www.abc.es/bienestar/alimentacion/abci-covid-19-como-evitar-contagios-hacer-compra-202003231108_noticia.html?ref=https:%2F%2Fwww.google.com%2F. [Último acceso: 16 Julio 2020].

6 ANEXOS

6.1 Manual Técnico

- Recopilación de requerimientos
- Historias de Usuario
- *Product Backlog*
- *Sprint Backlog*
- Diseño de Interfaces
- Diseño de la Base de Datos
- Pruebas Unitarias
- Pruebas de Carga
- Pruebas de Compatibilidad
- Pruebas de Aceptación

6.2 Manual de Usuario

<https://youtu.be/9SULWEZleGw>

6.3 Manual de Instalación

- Despliegue de la API REST en Digital Ocean
- Despliegue de la Aplicación Móvil en Google Play Store