

**UNIVERSIDAD POLITÉCNICA SALESIANA  
SEDE QUITO**

**CARRERA:  
INGENIERÍA DE SISTEMAS**

**Trabajo de titulación previo a la obtención del título:  
INGENIERO E INGENIERA DE SISTEMAS**

**TEMA:  
DESARROLLO DE UNA APP MÓVIL EN ANDROID PARA LA  
COMERCIALIZACIÓN DE PRODUCTOS AGRÍCOLAS DE LAS COMUNIDADES  
CAMPELINAS DEL ECUADOR.**

**AUTOR Y AUTORA  
JONATHAN VLADIMIR FLORES HEREDIA  
LUISA ANDREA RIOFRÍO GARCÍA**

**TUTORA:  
LINA PATRICIA ZAPATA MOLINA**

**Quito, septiembre del 2021**

## CESIÓN DE DERECHOS DE AUTOR

Nosotros Jonathan Vladimir Flores Heredia y Luisa Andrea Riofrío García, con documentos de identificación N° 1725358442 y N° 2200224000, manifestamos nuestra voluntad y cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del trabajo de titulación intitulado: DESARROLLO DE UNA APP MÓVIL EN ANDROID PARA LA COMERCIALIZACIÓN DE PRODUCTOS AGRÍCOLAS DE LAS COMUNIDADES CAMPESINAS DEL ECUADOR., mismo que ha sido desarrollado para optar por el título de: Ingenieros de Sistema, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En aplicación a lo determinado en la Ley de Propiedad Intelectual, en mi condición de autor me reservo los derechos morales de la obra antes citada. En concordancia, suscribo este documento en el momento que hago entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.

Quito, septiembre del 2021.



.....  
Jonathan Vladimir Flores Heredia

CI: 1725358442



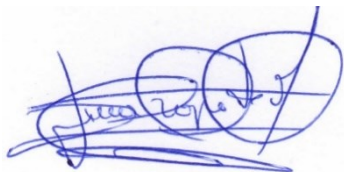
.....  
Luisa Andrea Riofrío García

CI: 2200224000

## DECLARATORIA DE COAUTORÍA DEL DOCENTE TUTOR/A

Yo declaro que bajo mi dirección y asesoría fue desarrollado el Proyecto Técnico, DESARROLLO DE UNA APP MÓVIL EN ANDROID PARA LA COMERCIALIZACIÓN DE PRODUCTOS AGRÍCOLAS DE LAS COMUNIDADES CAMPESINAS DEL ECUADOR., realizado por Jonathan Vladimir Flores Heredia y Luisa Andrea Riofrío García, obteniendo un producto que cumple con todos los requisitos estipulados por la Universidad Politécnica Salesiana, para ser considerados como trabajo final de titulación.

Quito, septiembre del 2021.

A handwritten signature in blue ink, appearing to read 'Lina Patricia Zapata Molina', with a horizontal dashed line below it.

Firma

LINA PATRICIA ZAPATA MOLINA

CI: 0501877278

## **DEDICATORIA**

Dedico mi trabajo de titulación a Dios, a mi querido padre Wilson Flores y a mi amada madre Carlota Heredia, pilares fundamentales de mi vida, a quienes estoy muy agradecido por su sacrificio, esfuerzo, cariño y amor que me han brindado desde el día en que nací, y gracias a ellos he podido cumplir con mi meta porque siempre han estado conmigo incondicionalmente a los que amare eternamente, también va dirigido a mi hermana Alejandra Flores quien me apoyo y estrecho su mano en momentos difíciles, a mi sobrino Ian Puente que es como mi hijo, hermano y a su vez mi mejor amigo a quien adoro con toda mi alma, a mi tía María Sandoval que siempre ha estado apoyándome, a mi primo Wilson Rubén Flores que es como un hermano mayor, a mis abuelitos Segundo Peña y Amelia Segovia a los que les agradeceré siempre por darme unos padres maravillosos.

Este trabajo fue forjado en conjunto con toda mi familia Flores, Heredia, que con cada consejo de aliento supieron motivarme para así poder cumplir con mi objetivo.

Jonathan Vladimir Flores Heredia

## **AGRADECIMIENTO**

Gracias a Dios por colmar mi vida llena de bendiciones, conocimiento y salud, a mis padres por su eterno amor, trabajo y dedicación que me han permitido cumplir con mi meta, a mi hermana y sobrino por estar siempre a mi lado apoyándome.

Agradezco a la Universidad Politécnica Salesiana, a la facultad de Ingeniería en Sistemas y a todos los docentes de esta noble institución, por formar parte de mi formación académica.

Mis más sinceros agradecimientos a la Ing. Lina Patricia Zapata por ser parte fundamental en todo este proceso, al guiarnos, apoyarnos y brindar sus conocimientos para realizar el desarrollo del proyecto.

Jonathan Vladimir Flores Heredia

## **DEDICATORIA**

Dedico este trabajo de titulación a Dios, a mis padres por toda la dedicación, amor y comprensión que me han brindado durante toda mi vida, mis hermanas por ser parte fundamental en mi vida con quienes he aprendido y puesto en práctica todos los valores, y principios para poder conseguir todas las metas propuestas.

Luisa Andrea Riofrío García

## **AGRADECIMIENTO**

Agradezco a la Universidad Politécnica Salesiana, que ha contribuido en mi formación académica.

Mis más sinceros agradecimientos a nuestra tutora Ing. Lina Patricia Zapata por habernos guiado, apoyarnos y brindado sus conocimientos para realizar el desarrollo del proyecto.

Luisa Andrea Riofrío García

## ÍNDICE GENERAL

INTRODUCCIÓN	1
Temas relacionados	1
Problema	2
Justificación	3
Objetivos	3
General	3
Específicos	4
Metodología	4
CAPÍTULO I	5
MARCO TEÓRICO	5
1.1. Arquitectura cliente - servidor	5
1.2. Modelo Vista Controlador (MVC)	5
1.3. Dispositivos móviles	5
1.4. Aplicaciones móviles	7
1.5. Android	8
1.6. Herramientas de Desarrollo	10
1.7. METODOLOGÍA XP	15
1.7.1. FASES DE LA METODOLOGÍA	17
CAPÍTULO II	18
ANÁLISIS Y DISEÑO	18
2.1. ESPECIFICACIONES DEL USUARIO	19
Requerimientos del usuario	19
Requerimientos	19
2.1.1. REQUERIMIENTOS FUNCIONALES	19
2.1.2. REQUERIMIENTOS NO FUNCIONALES	25
2.2. Diseño y Modelado del Sistema	26
2.2.1. Diagramas de caso de uso (CU)	26
Caso de uso: crear nueva cuenta	26
Caso de uso: productor	27
Caso de uso: iniciar sesión	30
Caso de uso: recuperar contraseña	32
Caso de uso: realizar compra	33
2.2.2. Diagramas de secuencia	35
Diagrama de secuencia: crear nueva cuenta	35
Diagrama de secuencia: iniciar sesión	35
Diagrama de secuencia: recuperar contraseña	36



Diagrama de secuencia: realizar compra	37
2.2.3. Menú de navegación	37
Menú: autenticación	37
Menú: gestión de información del producto	38
Menú: realizar compra	39
2.2.4. Diagramas de actividades	40
Diagrama de clases	42
2.2.5. Diagrama de clases	43
2.3. Diseño de Interfases	45
Prototipo de interfaces del sistema	45
2.4. Estructura de la Base de datos	54
Diseño lógico	54
Diseño físico	55
Diccionario de datos	56
CAPÍTULO III	63
CONSTRUCCION Y PRUEBAS	63
3.1. Diagrama de despliegue	63
3.2. Herramientas	64
3.3. Funcionalidad y Codificación de la aplicación móvil	65
3.4. Pruebas	76
3.4.1. Prueba de caja negra	77
3.4.2. Prueba de carga	90
3.4.3. Prueba de estrés	97
3.5. Pruebas de uso de los recursos del dispositivo móvil	99
CONCLUSIONES	103
RECOMENDACIONES	104
LISTA DE REFERENCIAS	105
Anexo 1	108
Manual de usabilidad de la aplicación	108
Anexo 2	108
Implementación de la metodología	108
Anexo 3	110
Pruebas de funcionalidad de la aplicación móvil	110

## ÍNDICE DE TABLAS

Tabla 1 <i>Métodos de petición de Servicios Restful</i>	12
Tabla 2 <i>Códigos de estado de respuesta</i>	13
Tabla 3 <i>Requerimiento funcional 1: Registro de usuario</i>	19
Tabla 4 <i>Requerimiento funcional 2: Iniciar Sesión</i>	20
Tabla 5 <i>Requerimientos funcionales 3: Recuperar contraseña</i>	20
Tabla 6 <i>Requerimiento funcional 4: Ingresar datos del producto</i>	21
Tabla 7 <i>Requerimiento funcional 5: Modificar y eliminar producto</i>	22
Tabla 8 <i>Requerimiento funcional 6: Agregar producto al carrito de compras</i>	22
Tabla 9 <i>Requerimiento funcional 7: Comprar producto</i>	23
Tabla 10 <i>Requerimiento funcional 8: Actualización de información del usuario</i>	23
Tabla 11 <i>Requerimiento funcional 9: Ventas realizadas</i>	24
Tabla 12 <i>Requerimiento funcional 10: Compras realizadas</i>	24
Tabla 13 <i>Requisitos no funcionales</i>	25
Tabla 14 <i>Escenario de caso de uso: Crear nueva cuenta</i>	27
Tabla 15 <i>Escenario de caso de uso: gestión del producto</i>	28
Tabla 16 <i>Escenario de caso de uso: control de stock y caducidad del producto</i>	29
Tabla 17 <i>Escenario de caso de uso: ofertas</i>	30
Tabla 18 <i>Escenario de caso de uso: iniciar sesión</i>	31
Tabla 19 <i>Escenario de caso de uso: recuperar contraseña</i>	32
Tabla 20 <i>Escenario de caso de uso: realizar compras</i>	33
Tabla 21 <i>Descripción de las clases de la aplicación móvil</i>	43
Tabla 22 <i>Descripción de la tabla PERSONA</i>	56
Tabla 23 <i>Descripción de la tabla ROL</i>	57
Tabla 24 <i>Descripción de la tabla DIRECCION</i>	57
Tabla 25 <i>Descripción de la tabla PRODUCTO</i>	57
Tabla 26 <i>Descripción de la tabla CATEGORIA</i>	58
Tabla 27 <i>Descripción de la tabla PRESENTACION</i>	59
Tabla 28 <i>Descripción de la tabla MEDIDA</i>	59
Tabla 29 <i>Descripción de la tabla OFERTA</i>	60
Tabla 30 <i>Descripción de la tabla VENTA</i>	60
Tabla 31 <i>Descripción de la tabla TIPO_PAGO</i>	61
Tabla 32 <i>Descripción de la tabla DETALLE_VENTA</i>	61
Tabla 33 <i>Descripción de la tabla NOTIFICACION</i>	62
Tabla 34 <i>Particiones de equivalencia para pruebas de caja negra</i>	77
Tabla 35 <i>Particiones de equivalencia para el registro de usuarios</i>	77
Tabla 36 <i>Prueba de caja negra registrar usuario</i>	79
Tabla 37 <i>Particiones de equivalencia para iniciar sesión</i>	82
Tabla 38 <i>Prueba de caja negra iniciar sesión</i>	82
Tabla 39 <i>Particiones de equivalencia recuperar contraseña</i>	83
Tabla 40 <i>Prueba de caja negra recuperar contraseña</i>	84
Tabla 41 <i>Particiones de equivalencia ingresar productos</i>	85
Tabla 42 <i>Prueba de caja negra ingresar producto</i>	87

Tabla 43 <i>Características del hardware del dispositivo móvil</i>	90
Tabla 44 <i>Peticiones y métodos HTTP para pruebas de carga</i>	90

## ÍNDICE DE FIGURAS

Figura 1 <i>Arquitectura de Android</i>	9
Figura 2 <i>Roles de la metodología XP</i>	16
Figura 3 <i>Fases de metodología XP</i>	17
Figura 4 <i>CU: Crear cuenta</i>	26
Figura 5 <i>CU: Productor</i>	28
Figura 6 <i>CU: Iniciar sesión</i>	31
Figura 7 <i>CU: Recuperar contraseña</i>	32
Figura 8 <i>CU: Realizar una compra</i>	33
Figura 9 <i>DS: Crear nueva cuenta</i>	35
Figura 10 <i>DS: Iniciar sesión</i>	36
Figura 11 <i>DS: Recuperar contraseña</i>	36
Figura 12 <i>DS: Realizar compras</i>	37
Figura 13 <i>Esquema de navegación ingreso al sistema</i>	38
Figura 14 <i>Esquema de gestión de información del producto</i>	38
Figura 15 <i>Esquema de navegación realizar una compra</i>	39
Figura 16 <i>Flujograma crear nueva cuenta</i>	40
Figura 17 <i>Flujograma ingresar, modificar y eliminar producto</i>	41
Figura 18 <i>Flujograma realizar compra</i>	42
Figura 19 <i>Diagrama de clases</i>	43
Figura 20 <i>Prototipo de interfaz de ingreso al sistema</i>	46
Figura 21 <i>Interfaz crear nueva cuenta</i>	46
Figura 22 <i>Interfaz inicio de sesión</i>	47
Figura 23 <i>Interfaz recuperar contraseña</i>	47
Figura 24 <i>Interfaz menú del sistema</i>	48
Figura 25 <i>Interfaz perfil de usuario</i>	48
Figura 26 <i>Interfaz ingresar producto</i>	49
Figura 27 <i>Interfaz vista principal de productos</i>	49
Figura 28 <i>Lista de productos del productor</i>	50
Figura 29 <i>Interfaz agregar producto al carrito de compras</i>	50
Figura 30 <i>Interfaz vista detalle de producto</i>	51
Figura 31 <i>Interfaz finalizar compra</i>	51
Figura 32 <i>Ofertas de productos</i>	52
Figura 33 <i>Interfaz detalle compra</i>	52
Figura 34 <i>Historial de compras realizadas</i>	53
Figura 35 <i>Historial de ventas realizadas</i>	53
Figura 36 <i>Diseño lógico de la base de datos</i>	54
Figura 37 <i>Diseño físico de la base de datos</i>	55
Figura 38 <i>Diagrama de despliegue</i>	63
Figura 39 <i>Permisos de acceso a las aplicaciones del dispositivo móvil</i>	65
Figura 40 <i>Conexión a la base de datos</i>	66
Figura 41 <i>Pantalla de inicio de la App móvil</i>	67
Figura 42 <i>Pantalla de registro de usuarios de la App móvil</i>	67

Figura 43 Método registro de usuarios	68
Figura 44 Método cargarSpinnerDireccion	69
Figura 45 Pantalla iniciar sesión de la App móvil	69
Figura 46 Método utilizado para el inicio de sesión	70
Figura 47 Pantalla recuperar contraseña de la App móvil	71
Figura 48 Validación del correo electrónico	71
Figura 49 Validación de contraseña	72
Figura 50 Pantalla ingresar producto	72
Figura 51 Método para el autocompletado del nombre del producto	73
Figura 52 Método seleccionar imagen de galería y usar la cámara	73
Figura 53 Pantalla vista principal productos de la App móvil	74
Figura 54 Pantalla vista detalle del producto de la App móvil	74
Figura 55 Método agregar producto al carrito de compras	75
Figura 56 Pantalla detalle de compra del producto de la App móvil	76
Figura 57 Herramienta JMeter	91
Figura 58 Configuración de usuarios en JMeter	92
Figura 59 Configuración de la petición HTTP	93
Figura 60 Configuración de la ruta de la petición	93
Figura 61 Resultado de la prueba de carga registro de usuarios	94
Figura 62 Resultado de la prueba de carga de login	94
Figura 63 Resultado de la prueba de carga crear producto	95
Figura 64 Resultado de la prueba de carga listar producto	95
Figura 65 Resultado de la prueba de carga realizar compra	96
Figura 66 Gráfica de medio, mínimo y máximo de carga de usuarios	96
Figura 67 Gráfica de resultados generales de las pruebas de carga	97
Figura 68 Resultados de la prueba de estrés	98
Figura 69 Gráfica de los resultados de la prueba de estrés	98
Figura 70 Consumo de recursos de la pantalla principal	99
Figura 71 Consumo de recursos de la pantalla registrar usuarios	100
Figura 72 Consumo de recursos del login	100
Figura 73 Consumo de recursos de la pantalla ingresar producto	101
Figura 74 Consumo de recursos al realizar una compra	101

## **RESUMEN**

Nuestro trabajo consiste en el diseño y desarrollo de una aplicación para dispositivos móviles, compatibles con Android, enfocada en la compra y venta de productos agrícolas. Esta aplicación busca ayudar a los pequeños y medianos productores a gestionar el proceso de comercialización de sus productos de manera online. Esta necesidad surgió a raíz de la pandemia mundial, donde se evidenció la falta de un software que apoye a los emprendedores de las provincias del Ecuador en la comercialización de productos agrícolas.

El uso de los recursos tecnológicos actuales junto a la metodología XP permitieron que la aplicación móvil sea desarrollada con éxito, de tal manera que el emprendedor pueda utilizar la aplicación sin ningún problema.

Como resultado de este proyecto, se obtendrá una aplicación para dispositivos móviles compatibles con Android, que se enfoca en la compra y venta de productos agrícolas lo que ayudará a los productores a gestionar sus procesos de comercialización de manera online.

## **ABSTRACT**

Our work consists of the design and development of an application for mobile devices, compatible with Android, focused on the purchase and sale of agricultural products. This application seeks to help small and medium producers to manage the process of marketing their products online. This need arose as a result of the global pandemic, where the lack of software to support entrepreneurs in the provinces of Ecuador in the marketing of agricultural products became evident.

The use of current technological resources together with the XP methodology allowed the mobile application to be developed successfully, so that the entrepreneur can use the application without any problem.

As a result of this project, an application for Android compatible mobile devices will be obtained, which focuses on the purchase and sale of agricultural products that will help producers to manage their marketing processes online.

## **INTRODUCCIÓN**

La productividad agrícola del país contempla un factor alimenticio y económico primordial para la población. Durante muchos años, la comercialización de productos agrícolas se ha realizado a través de la venta de productos, en la cual los intermediarios realizan el proceso de compra y venta al por mayor. La incapacidad de comunicarse directamente con los consumidores afecta a los productores en el pago injusto de los productos cultivados.

En la actualidad, debido a la pandemia provocada por el COVID-19, el proceso de comercialización de productos agrícolas ha generado cambios en los negocios, lo que afecta a gran parte de la población del Ecuador, como es el caso de los productores de las provincias de Pichincha y Cotopaxi, que aparte de comercializar sus productos por medio de intermediarios, no cuentan con herramientas tecnológicas para llevar a cabo el proceso de comercialización ni espacios para promocionar sus productos (Gobierno de la Republica del Ecuador, 2020).

El documento consta de tres capítulos, el primer capítulo es el marco teórico, que contiene todas las definiciones de los requisitos utilizados en el desarrollo del proyecto, el segundo capítulo: análisis y diseño, correspondiente a los diagramas de casos de uso, secuencia, navegación, diseño lógico y físico de la base de datos, finalmente el tercer capítulo: construcción y pruebas, que define como se realizó la construcción de la aplican móvil y sus pruebas correspondientes para medir su óptimo rendimiento.

### **TEMAS RELACIONADOS**

El problema de comercialización de productos agrícolas es evidente a nivel mundial, por lo que existen sistemas similares al proyecto propuesto en este trabajo, se citará y comparará proyectos para tomar decisiones que ayudaran al desarrollo del proyecto planteado.

En la Escuela Superior Politécnica del Litoral en Guayaquil, se propuso el tema: “DESARROLLO DE UNA APLICACIÓN PARA COMERCIALIZAR LOS PRODUCTOS AGRÍCOLAS SIN INTERMEDIARIOS.”



En la Universidad Nacional José maría Arguedas en Perú, se planteó el tema: **DESARROLLO DE UNA APLICACIÓN MOVIL PARA LA COMERCIALIZACION DE PRODUCTOS EN EL SECTOR AGRICOLA DE ANDAHUAYLAS**, el propósito del desarrollo de la aplicación móvil se debe a que los productores desconocen el precio de los productos y a causa de eso venden a precios bajos (Alarcón, 2019).

Se pudo observar que existen varios sistemas de comercialización, entre ellos algunos elaborados para poblaciones extensas en colaboración con los productores. La diferencia entre este proyecto y los anteriormente mencionados, es que se encuentra integrado con la página web que fue desarrollado por otro grupo de estudiantes, conectados a una base de datos en común con la que los emprendedores tienen la opción de usar tanto la página web como la aplicación móvil. También se implementó un proceso de oferta que ayudara a los pequeños agricultores a promocionar sus productos agrícolas antes de que cumplan con su tiempo de caducidad o no se logren vender.

El proceso de desarrollo de los proyectos fue especificado de acuerdo con las características establecidas por el cliente y así construir una aplicación que permita a los emprendedores a generar recursos económicos estables en la venta de productos. La aplicación móvil planteada está dirigido solo para el sector agrícola de las provincias del Ecuador el cual permitirá realizar compras de productos de manera online.

## **PROBLEMA**

El proceso de comercialización de los productos agrícolas, antes de la pandemia del Covid-19, se lo efectuaba en mercados locales y en ferias libres donde el principal comprador eran los comerciantes (intermediario). Esto ocasionaba que los pequeños agricultores (emprendedores) reciba un valor bastante inferior al real. Una manera de recibir un precio más justo por el producto es que el agricultor logre vender directo al consumidor sin intermediarios (comerciantes). Una forma para que el agricultor pueda difundir sus productos y vender

directamente al consumidor, es mediante del uso de aplicaciones tecnológicas como es el caso de una aplicación móvil donde pueda publicar fotos o imágenes de sus productos y los consumidores puedan realizar compras en línea.

Esta app les ayudará a los pequeños productores a desarrollar su negocio de manera rápida y efectiva al promover las ventas de sus productos con la ayuda de teléfonos móviles.

## **JUSTIFICACIÓN**

Ante la problemática antes mencionada, surge la necesidad de disponer de herramientas tecnológicas, es por ello que proponemos desarrollar una APP para las comunidades campesina en especial. La aplicación móvil será intuitiva, confiable y segura donde los pequeños agricultores podrán registrar sus productos y comercializarlos en línea en forma directa al consumidor. Por tratarse de personas de comunidades campesinas que no tienen acceso a una computadora, el uso de App móviles es apropiada por la facilidad de uso, ubicación geográfica y por ser más práctico y económico disponer de un teléfono celular, frente a las aplicaciones pensadas para que funcionen sobre una computadora.

La plataforma móvil tendrá un impacto económico y social positivo, a futuro reemplazará el proceso de compra-venta de productos agrícolas en ferias libres o mercados por compras en línea donde se reducirá el tiempo de comercialización y así recibir precios más justos por los productos. Los consumidores podrán comprar productos de calidad a precios razonables, y también podrán conocer la información y el estado de cosecha de los productos a comprar.

## **OBJETIVOS**

### ***General***

Desarrollar una aplicación móvil en Android para la comercialización de productos agrícolas, que permita a los pequeños agricultores y consumidores la compra y venta de productos en forma directa.

### *Específicos*

Construir una aplicación móvil que facilite la gestión y comercialización de productos agrícolas de las comunidades campesinas.

Implementar un proceso de ofertas o descuentos que permita promocionar ciertos productos agrícolas.

Implementar pruebas al software para medir el cumplimiento de los requerimientos funcionales de la App móvil.

### **METODOLOGÍA**

Obtener información sobre la comercialización de productos agrícolas de las comunidades campesinas de ciertas provincias como Pichincha y Cotopaxi, a fin de determinar las herramientas tecnológicas a utilizar para desarrollar la APP móvil.

Analizar las necesidades del proyecto para determinar los diferentes módulos y tiempos en el desarrollo de la aplicación móvil.

Implementar pruebas de integración y funcionamiento de los módulos, mediante el uso de métricas que ayudan a controlar el estado de la aplicación.

Para implementar la APP móvil, la metodología más adecuada para este proyecto es el método XP (Extreme Programming) que permite trabajar en parejas e interactuar con el cliente, es un enfoque orientado a proyectos de corto plazo que permite entregar el software al cliente cuando sea necesario (Coello, 2017).

También se aplicarán buenas prácticas de otros métodos, como el desarrollo de software iterativo e incremental de la metodología SCRUM, que puede dar resultados en un período de tiempo llamado Sprint, el proceso suele durar de dos a cuatro semanas, tiempo durante el cual reflejará una nueva versión de software con nuevas funciones para optimizar y controlar los riesgos.

# CAPÍTULO I

## MARCO TEÓRICO

### ***1.1. Arquitectura cliente - servidor***

Es una arquitectura que tiene como objetivo principal atender las peticiones solicitadas por el cliente o clientes al servidor.

En la arquitectura cliente – servidor, el cliente envía un mensaje solicitando un recurso al servidor que pueden ser un conjunto de ordenadores que brindan un servicio por medio de la red, envía una respuesta a la petición solicitada por el cliente siempre y cuando tenga los permisos para acceder a esa información, es decir el servidor hace la validación de que el cliente posea los permisos necesarios para entregar una respuesta.

### ***1.2. Modelo Vista Controlador (MVC)***

MVC es una de las arquitecturas más implementadas al momento de desarrollar software, permite dividir a la aplicación en tres componentes, para analizar cada parte por separado, mantener una organización al desarrollar el programa y obtener un software de calidad, se divide en:

- Modelo: la capa donde se trabaja con los datos es el encargado de gestionar, manipular y actualizar la información.
- Vista: muestra el diseño de la interfaz del programa con la que el usuario va a interactuar, diseñada de acuerdo con la base de datos definida.
- Controlador: trabaja como intermediario entre la vista y el modelo para presentar información a los usuarios (Silva, 2017).

### ***1.3. Dispositivos móviles***

A escala mundial, los dispositivos móviles han evolucionado para ayudar a las personas mediante el uso de tecnologías que facilitan el trabajo de los dispositivos electrónicos.

La gran mayoría de los dispositivos móviles pueden conectarse a Internet y realizar sus funciones. Ciertos dispositivos electrónicos (como teléfonos, relojes inteligentes, impresoras, radios, etc.) se han incluido en la categoría de dispositivos móviles porque pueden conectarse a otros dispositivos conectados a través del Internet, Bluetooth o Wifi (Protección de datos, 2019).

***Tipos de dispositivos móviles.*** A continuación, se presenta los dispositivos móviles más comunes en la actualidad:

### **Teléfonos móviles**

Actualmente, los teléfonos móviles son dispositivos inalámbricos que se caracterizan por tener una conexión a Internet accediendo a una red de telefonía móvil o celular que brinda cobertura dentro de un cierto ángulo o rango determinado que permite realizar funciones básicas como hacer y recibir llamadas y mensajes de texto (Tapia & García, 2016).

La portabilidad y usabilidad de los dispositivos móviles son unas de sus principales características porque son fáciles de transportar y utilizar, lo que se debe a su tamaño, que es muy fácil de trasladar de un lugar a otro. La comunicación también es una particularidad del dispositivo, ya que permite comunicarse entre el teléfono y la red de telefonía celular utilizando frecuencias transmitidas a través del espectro electromagnético (Tapia & García, 2016).

### **Smartphones (teléfonos inteligentes)**

Son dispositivos inteligentes con características similares a los teléfonos móviles, pero con la capacidad de realizar funciones similares a las computadoras personales, y son compactos y fáciles de usar. Las características de los teléfonos inteligentes dependen del hardware y el software, porque estos componentes están desarrollados para completar tareas que requieren más memoria y potencia de procesamiento (Tapia & García, 2016).

Los componentes del teléfono inteligente (como software y hardware) cumplen con las características definidas mediante el uso del sistema operativo, el cual es responsable de brindar

seguridad, administrar recursos, optimizar tareas y brindar acceso a Internet mediante la conexión a redes tecnológicas como la telefonía móvil que brinda una cobertura de acceso a datos de internet y otras comunicaciones inalámbricas (por ejemplo: Wifi, GPRS, Bluetooth, GPS), que son herramientas con funciones informáticas (Tapia & García, 2016).

### **Tablet**

Las propiedades son semejantes a las de un smartphone con la diferencia de su pantalla que posibilita visualizar de gran tamaño sus aplicaciones es un dispositivo inteligente localizado en la mitad de un computador personal y un teléfono inteligente. Mediante la interfaz de la pantalla táctil permite ingresar texto sin necesidad de usar un teclado o un mouse (Tapia & García, 2016).

### **Reloj inteligente (Smartwatch)**

Son relojes de pulsera se utilizan para realizar diversas funciones de los teléfonos inteligentes por medio de la interfaz de un teléfono, cuenta con un procesador que permite almacenar información. La conectividad entre un smartwatch y un smartphone es mediante Bluetooth el cual admite la comunicación de los dispositivos.

## ***1.4. Aplicaciones móviles***

Las aplicaciones móviles están transformando la tecnología mundial, mejorando sus capacidades técnicas, para que las computadoras se adapten a las plataformas móviles, que pueden acumular grandes cantidades de información (Banda, 2015).

Son programas que permiten a los usuarios realizar tareas individuales, acceder a servicios, comunicarse continuamente entre usuarios y actualizarse con información diaria, están diseñados para ejecutarse en teléfonos inteligentes, tables e incluso algunos relojes inteligentes.

Características de las aplicaciones móviles:

- Interfaz Simple: diseño amigable, sencillo y entendible de la App para que cualquier usuario pueda interactuar con el sistema sin problemas y sin la necesidad y una capacitación.
- Seguridad APPs: permiten mantener seguros los datos y privacidad de los usuarios.
- Funcionamiento Offline (Fuera de línea): la disponibilidad de internet no tiene que influir en la App, la cual tiene que cumplir con su funcionalidad para la que fue desarrollada.
- Actualizaciones periódicas: para tener una buena funcionalidad de las aplicaciones es necesario actualizarlas cada cierto tiempo, lo que permite corregir errores y establecer nuevas versiones para cumplir con las necesidades de los usuarios (Tubón, 2020).

### ***1.5. Android***

Es un sistema operativo móvil que fue desarrollado por Android In. hasta el año 2005 en donde Google realizo la adquisición de Android, para ello se basó en el kernel de Linux y también en otro software de código abierto. Este sistema operativo fue diseñado para funcionar en dispositivos móviles con pantalla táctil, por ejemplo, teléfonos inteligentes, tablets, relojes inteligentes automóviles y también Smart TV (Urriolabeytia, 2020, pág. 6).

Características del sistema operativo Android

- Es un sistema de código abierto.
- Presenta un núcleo basado en el Kernel de Linux.
- Utiliza SQLite para almacenar los datos.
- Se adapta a varias pantallas y resoluciones.
- Ofrece diversas formas de mensajería.

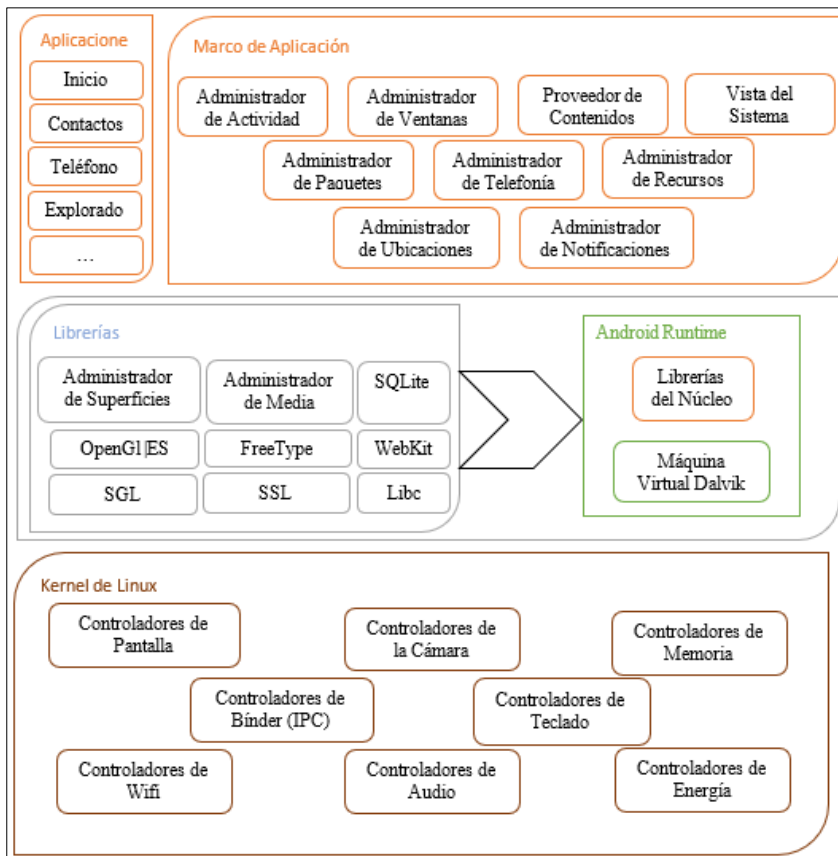
- Navegador Web basado en WebKit incluido (Basterra, Berteza, Borello, Castillo, & Venturi, 2017, pág. 4).

## Arquitectura de Android

La arquitectura Android está conformada por cuatro capas que se relacionan entre sí, una característica importante es que todas las capas están basadas en software libre, a continuación, se observará la representación de cada nivel de la arquitectura.

**Figura 1**

*Arquitectura de Android*



**Nota.** Capas de la arquitectura de Android. Elaborado por: Flores, Riofrío.

En la figura 1, se observa cómo está estructurada cada capa de la arquitectura de Android, a continuación, se describe las definiciones de las capas:



- Capa del Kernel de Linux: se define como la parte central del sistema operativo, donde se puede acceder a las funciones de los diferentes controladores que permite que el software y el hardware del dispositivo puedan trabajar juntos.
- Capa de librerías: permite incorporar funciones especiales en el desarrollo de aplicaciones Android.
- Capa de marco de aplicaciones: administra la gestión de los recursos de los procesos de desarrollo de las aplicaciones.
- Capa de aplicación: definida por defecto y aquellas capas de aplicación que los usuarios pueden agregar a sus propias aplicaciones.

## ***1.6.Herramientas de Desarrollo***

### **Android Studio**

Android Studio se anunció en la conferencia de Google el 16 de mayo de 2013, definiéndolo como un entorno de desarrollo compuesto para la plataforma Android, es un IDE para la implementación de aplicaciones móviles, que sustituyó a Eclipse IDE.

En diciembre de 2014, IntelliJ IDEA de JetBrains, fue anunciado como la primera versión, publicada de forma gratuita bajo la licencia Apache 2.0. La misma que estará disponible para plataformas Microsoft Windows, Mac OS X y GNU/Linux (Rincón & Ceballos, 2016).

Para implementar aplicaciones móviles, Android Studio admite lenguajes de programación como: Java, Kotlin y C++ para realizar el desarrollo.

### **JAVA**

Es un lenguajes de programación orientado a objetos, para el diseño y desarrollo de diversas aplicaciones, publicado por la empresa Sun Microsystem en el año de 1995, multiplataforma que permite que los programas puedan ejecutarse en diferentes plataformas,

permitiendo a los desarrolladores desarrollar páginas web para ser ejecutados en cualquier dispositivo (Montalvo, 2018).

#### Principales características de Java

- Simple: es un lenguaje de programación sencillo, fácil de entender y de usar al momento de desarrollar aplicaciones.
- Orientado a objetos: es un paradigma de la programación que permite entender el modelo de la programación mediante pautas para desarrollar software en él, permite organizar y reutilizar código para mantener una programación estable fácil de entender.
- Distribuido: elabora aplicaciones con diferentes componentes que son ejecutados en ambientes separados por medio de compartición de clases y redes tecnológicas.
- Robusto: es robusto porque al ejecutar las aplicaciones asegura la funcionalidad del sistema al permitir la multi-verificación.
- Seguro: al contener varias capas de seguridad en sus entornos, hace que Java elabore aplicaciones seguras.
- Compatible: la ventaja de java es ser compatible con sistemas operativos comunes como puede ser Unix, Linux, Mac, Windows y hasta con sistemas operativos móviles como es el caso de Android (Montalvo, 2018).

#### **Web Service**

Son herramientas tecnológicas que permiten interactuar con otras aplicaciones intercambiando información entre sí, para proporcionar servicios de comunicación web. Componente al que se puede acceder por medio de protocolos de transporte, utilizando un conjunto de reglas de codificación para el intercambio de información (García, 2017).

#### Definiciones:

- HTTP (Protocolo de Transferencia de Hipertexto): protocolo de comunicación que pertenece a la capa de aplicación del modelo OSI, establecido para realizar transferencia de archivos.
- JSON (JavaScript Object Notation): permite realizar el intercambio de información, definido como un formato de texto ligero.
- XML (Extensible Markup Lenguaje): es un meta-lenguaje que se caracteriza por diseñar etiquetas personalizadas.

A continuación, se definen tecnologías que incorporan las Web Service:

- SOAP (Simple Object Access Protocol): es un protocolo que permite intercambiar información a través de XML para establecer comunicación entre diferentes procesos.
- REST (Representational State Transfer): es una interfaz que permite la comunicación a través del protocolo HTTP, al establecer verbos (GET, POST, PUT y DELETE) para realizar el intercambio de información, las peticiones generadas por los verbos producen un código numérico de tres dígitos (1xx, 2xx, 3xx, 4xx, 5xx), que se establece para indicar el estado de la solicitud (Castro, 2020).

**Tabla 1**

*Métodos de petición de Servicios Restful*

Verbo	Descripción
<b>GET (obtener)</b>	Método para obtener un recurso
<b>POST (crear)</b>	Método para enviar o crear nuevos recursos
<b>PUT (modificar)</b>	Método para modificar un recurso

<b>DELETE (eliminar)</b>	Método para eliminar un recurso
--------------------------	---------------------------------

*Nota.* Métodos para realizar peticiones entre la App móvil y el servidor. Elaborado por: Flores, Riofrío.

## Tabla 2

### *Códigos de estado de respuesta*

Código	Descripción
<b>100</b>	Código informativo especificado en HTTP
<b>200</b>	Indica que la solicitud generada se realizó satisfactoriamente
<b>300</b>	Visualiza un recurso que se movió de manera inesperada a otra ubicación
<b>400</b>	Error por parte del cliente (ingreso de datos incorrectos o faltantes)
<b>500</b>	Error generado por el servidor

*Nota.* Respuestas que genera las peticiones HTTP. Elaborado por: Flores, Riofrío.

## PHP

Es un lenguaje de programación de código abierto creado por Lerdorf, para desarrollar aplicaciones web, el cual puede ser incrustado en HTML. Cuando se envía una solicitud, se ejecuta en el lado del servidor y devuelve una respuesta, que permite visualizar al cliente lo solicitado.

## Bases de Datos

Una base de datos es una colección de archivos de datos organizado y estructurado que permite almacenar electrónicamente información, usando una variedad de operaciones sobre los archivos se pueden acceder, administrar, modificar, actualizar y controlar los datos fácilmente.

Para cumplir con el concepto de bases de datos se necesita realizar y considerar siguientes puntos:

- Realizar una metodología.
- El modelado de los datos.
- Una estructura formal (tablas, columnas y filas)

### **Normalización de base de datos**

Es el proceso que permite organizar y permite aplicar un conjunto de reglas para tener una estructura de datos ordenados, el objetivo es garantizar un almacenamiento optimo, la eliminación de datos repetidos, eliminar errores lógicos y la asignación de valor correcto a los datos.

### **Modelo Entidad Relación**

Es un método que te permite realizar el bosquejo de la estructura física y lógica de la base de datos de un proyecto, establecidos por los siguientes componentes: objetos, entidades, atributos y relaciones.

Principales componentes del Modelo Entidad-Relación:

- Entidad: Es una representación de un objeto, algo tangible en la vida real. Ejemplos estudiantes, clientes, productos.
- Atributos: las características principales que permiten describir los tipos de la entidad. Ejemplo de nombre, apellido, sexo, categoría.
- Relaciones: permite a las entidades establecer conexiones, al compartir atributos. Ejemplo, cada alumno tendrá un nombre, apellido y género.

### **XAMPP**

Es un entorno de código libre que permite evaluar aplicaciones. La instalación de Apache se realiza de forma sencilla en el ordenador, sin importar el sistema operativo (Linux,

Windows, MAC o Solaris), su uso es gratuito. Incluye además servidores de bases de datos como MySQL, SQL Lite entre otros (Haro & Inca, 2016).

Paquetes de XAMPP:

- Apache: representa el servidor web HTTP (Protocolo de Transferencia de Hipertexto) de código abierto que permite enviar páginas web de forma estáticas y dinámicas.
- PHP, Perl: Lenguajes de programación que permiten desarrollar páginas web, web Service, etc.
- MySQL: Gestor de bases de datos de código abierto que almacena información en forma de tablas.
- ProFTPD: servidor FTP (Protocolo de transferencia de archivos), basados en la arquitectura cliente servidor es un conjunto de normas que permite que los dispositivos establezcan una comunicación entre sí.

### **PhpMyAdmin**

Es una herramienta que permite organizar y administrar la información de MySQL mediante un navegador web, es un software gratuito desarrollado en PHP. A través de la interfaz el usuario puede administrar las bases de datos, relaciones, permisos y también permite ejecutar consultas SQL.

## **1.7. METODOLOGÍA XP**

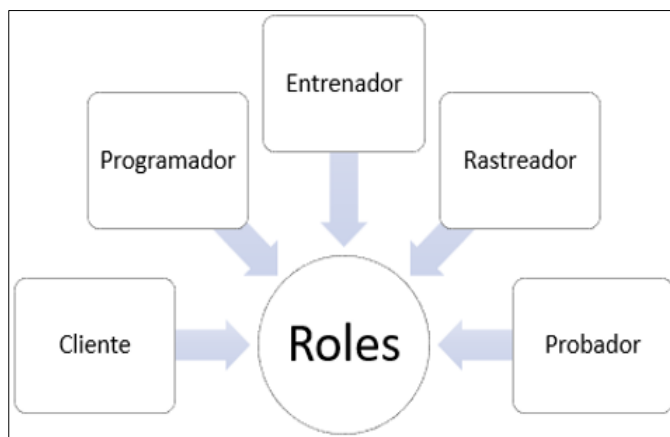
XP (Extreme Programming) es un método que permite la interacción con el cliente, establece iteraciones cada dos semanas, se caracteriza por ser el código la única documentación que existe en el proyecto. La intervención continua del cliente en el sistema permite realizar cambios en el desarrollo de software, teniendo en cuenta la disponibilidad planteada en el cronograma de actividades (Coello, 2017).

Al momento de desarrollar un proyecto de software la metodología XP permite establecer soluciones simples en la implementación de la aplicación, mediante la prontitud y disciplina aplicada en sus fases de progreso. XP permite la intervención del cliente desde el comienzo del desarrollo de la aplicación hasta finalmente obtener un programa de calidad.

Los roles que implica la metodología XP permiten realizar un desarrollo óptimo de las aplicaciones, los cuales se pueden evidenciar en la figura 2.

## Figura 2

*Roles de la metodología XP*



**Nota.** Roles que especifican los diferentes procesos que realizar el desarrollador. Elaborado por: Flores, Riofrío.

Cliente (Customer): especifica los requerimientos y las funcionalidades para la elaboración del sistema.

Programador (Developer): es el encargado de diseñar, desarrollar y realizar las respectivas pruebas de funcionamiento tomando en cuenta las disposiciones técnicas.

Entrenador (Coach): es el encargado de enseñar, vigilar y tomar decisiones para la implementación de los procesos.

Rastreador (Tracker): persona encargada de controlar el diseño y desarrollo de los procesos del software al identificar problemas en la aplicación.

Probador (Tester): miembro del equipo que trabaja con el cliente para probar la correcta funcionalidad del software.

### Ciclo de Vida

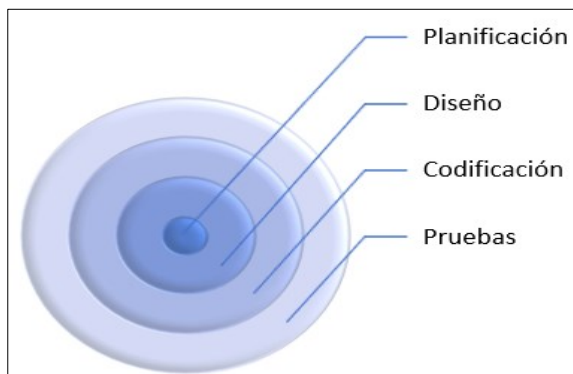
Es el tiempo de ejecución del software que está determinado por la iteración que se presentan a corto plazo, entre más breve sea el periodo de tiempo de entrega de la aplicación, se podrá realizar cambios en el diseño o funcionalidad si fuera necesario, y así obtener un producto de calidad.

#### 1.7.1. FASES DE LA METODOLOGÍA

La metodología XP está compuesta por cuatro fases para la implementación de software, que se describen a continuación:

#### Figura 3

*Fases de metodología XP*



**Nota.** Roles que especifican los diferentes procesos que realiza el desarrollador. Elaborado por: Flores, Riofrío.

En la figura 3, se puede visualizar las fases que la metodología XP, las mismas que serán descritas a continuación:

#### **Planificación**



Elaborar la planificación mediante los requisitos solicitados por el cliente, desarrollo de cronogramas de actividades para estimar el tiempo de duración del software.

### **Diseño**

El diseño se realiza mediante el bosquejo que es elaborado por el desarrollador y en gran parte del boceto el cliente tiende a interactuar especificando algunas partes del sistema.

### **Codificación**

El proceso de desarrollo de la aplicación se realizará en parejas, donde los miembros del equipo de trabajo se organizan para realizar la codificación de la aplicación asegurando que el código sea estable y entendible para futuras actualizaciones.

### **Pruebas**

Al realizar las pruebas funcionales, se pueden detectar fallas en la ejecución del sistema, por lo cual se deben realizar pruebas de manera continua para rastrear posibles fallos en la aplicación. El desarrollador y cliente son los responsables de realizar las pruebas necesarias, el programador es el encargado desarrollar a la app y realizar los diferentes testing para validar que el software es de calidad, por otro lado, el cliente se encarga de revisar que el sistema cumpla con las funcionalidades especificadas.

## **CAPÍTULO II**

### **ANÁLISIS Y DISEÑO**

El capítulo dos se analizará las especificaciones del usuario y requerimientos del sistema para establecer el funcionamiento de la aplicación y también establecer un bosquejo del diseño de la App móvil.

## 2.1. ESPECIFICACIONES DEL USUARIO

### *Requerimientos del usuario*

A través del uso de la tecnología se proporcionará a los agricultores una aplicación móvil que les permita comercializar en línea sus productos en forma directa con los consumidores a fin de que reciban un precio justo por su producto.

Cabe mencionar que el diseño de la base de datos de la aplicación móvil será utilizado también para el uso de la página web la cual será desarrollada por otro grupo de estudiantes en la que se implementaría el módulo de administración para las dos aplicaciones.

### *Requerimientos*

A continuación, se detallan los requerimientos funcionales y requerimientos no funcionales, especificados por el usuario y así poder cumplir con las necesidades planteadas por el cliente.

#### 2.1.1. REQUERIMIENTOS FUNCIONALES

A través de los requerimientos funcionales se va a describir las actividades que realiza el sistema siempre y cuando cumpla con las precondiciones establecidos en cada caso.

Como se puede observar en las tablas a continuación el usuario para poder acceder y realizar una actividad en el sistema debe cumplir con unos parámetros que le permitirán cumplir los procesos de ejecución de la aplicación.

En tabla 3, se puede visualizar el requerimiento funcional crear una cuenta en la que el usuario debe ingresar sus datos personales, como se trata del registro por primera vez no cuenta con una precondición.

**Tabla 3**

*Requerimiento funcional 1: Registro de usuario*

Descripción	Crear nueva cuenta
Precondición	Ninguna

<b>Entrada</b>	Nombre, teléfono, dirección, correo, contraseña, imagen
<b>Proceso</b>	Registrar los datos del usuario a la base de datos
<b>Salida</b>	Mensaje de correcto o error

**Nota.** Requisito funcional: crear nueva cuenta, el usuario ingresa sus credenciales para su registro. Elaborado por: Flores, Riofrío.

En tabla 4, se puede visualizar el requerimiento funcional iniciar sesión en la que el usuario debe ingresar su correo electrónico y contraseña, el proceso será completado siempre y cuando el usuario se encuentre registrado en el sistema.

**Tabla 4**

*Requerimiento funcional 2: Iniciar Sesión*

Descripción	Ingreso al sistema
<b>Precondición</b>	Usuario registrado.
<b>Entrada</b>	Correo electrónico y contraseña.
<b>Proceso</b>	Validación de datos ingresados sean correctos.
<b>Salida</b>	Mensaje de correcto o error.

**Nota.** Requisito funcional: iniciar sesión, el usuario ingresa al sistema por medio del correo y contraseña. Elaborado por: Flores, Riofrío.

En tabla 5, se puede visualizar el requerimiento funcional recuperar contraseña en la que el usuario debe ingresar el correo electrónico, siempre y cuando se encuentre registrado en el sistema.

**Tabla 5**

*Requerimientos funcionales 3: Recuperar contraseña*

Descripción	Recuperar contraseña
<b>Precondición</b>	Usuario registrado.

<b>Entrada</b>	Correo electrónico.
<b>Proceso</b>	Validación de correo electrónico sean correctos.
<b>Salida</b>	Mensaje de correcto o error.

**Nota.** Requisito funcional: recuperar contraseña, para poder recuperar la contraseña el usuario debe estar registrado. Elaborado por: Flores, Riofrío.

En tabla 6, se puede visualizar el requerimiento funcional ingresar información del producto en el sistema, para poder realizar el proceso primero debe autenticarse.

**Tabla 6**

*Requerimiento funcional 4: Ingresar datos del producto*

Descripción	Agregar información del producto
<b>Precondición</b>	Autenticación de usuario.
<b>Entrada</b>	Imagen, presentación, categoría, medida, nombre, medida, cantidad, precio, fecha cosecha, fecha expiración, descripción.
<b>Proceso</b>	Autenticación de usuario.  Ingresar al menú a mis productos.  Lista de productos.  Agregar producto.  Ingresar datos del producto.
<b>Salida</b>	Mensaje de correcto o error.

**Nota.** Requisito funcional: ingresar datos del producto, el usuario por medio un formulario ingresara los datos del producto. Elaborado por: Flores, Riofrío.

En tabla 7, se puede visualizar el requerimiento funcional modificar y eliminar producto, para realizar el proceso primero debe autenticarse.

**Tabla 7***Requerimiento funcional 5: Modificar y eliminar producto*

Descripción	Eliminar - modificar producto
<b>Precondición</b>	Autenticación de usuario.
<b>Entrada</b>	Seleccionar producto
<b>Proceso</b>	Autenticación de usuario. Ingresar al menú, mis productos. Lista de productos. Eliminar o modificar datos del producto.
<b>Salida</b>	Mensaje de correcto o error.

*Nota.* Requisito funcional: eliminar y modificar datos del producto, permite borrar y actualizar los datos del producto al usuario con permisos. Elaborado por: Flores, Riofrío.

En tabla 8, se puede visualizar el requerimiento funcional agregar el producto al carrito de compras, para realizar el proceso primero debe autenticarse.

**Tabla 8***Requerimiento funcional 6: Agregar producto al carrito de compras*

Descripción	Agregar producto al carrito de compras
<b>Precondición</b>	Autenticación de usuario.
<b>Entrada</b>	Seleccionar producto.
<b>Proceso</b>	Autenticación de usuario. Seleccionar producto. Agregar producto al carrito de compras.
<b>Salida</b>	Mensaje de correcto o error.

*Nota.* Requisito funcional: agregar producto al carrito, para poder agregar un producto al carrito primero tiene que autenticarse. Elaborado por: Flores, Riofrío.

En tabla 9, se puede visualizar el requerimiento funcional comprar un producto, para poder realizar la compra el usuario debe autenticarse.

**Tabla 9**

*Requerimiento funcional 7: Comprar producto*

Descripción	Comprar producto
<b>Precondición</b>	Autenticación de usuario.
<b>Entrada</b>	Ingresar al carrito de compras.
<b>Proceso</b>	Seleccionar forma de pago. Seleccionar botón finalizar compra. Aceptar compra.
<b>Salida</b>	Mensaje de correcto o error.

*Nota.* El requisito funcional: comprar producto, se cumplirá siempre y cuando se autentifique en el sistema. Elaborado por: Flores, Riofrío.

En tabla 10, se puede visualizar el requerimiento funcional actualizar información del usuario y cambiar contraseña, el cual se debe cumplir con la precondición autenticación.

**Tabla 10**

*Requerimiento funcional 8: Actualización de información del usuario*

Descripción	Actualización de datos del usuario y cambiar contraseña.
<b>Precondición</b>	Autenticación de usuario.
<b>Entrada</b>	Mi perfil.
<b>Proceso</b>	Autenticación de usuario. Ingresar al menú, ir a mi perfil. Modificar datos del usuario. Cambiar contraseña.
<b>Salida</b>	Mensaje de correcto o error.

**Nota.** Requisito funcional: actualizar información de usuario, permite modificar los datos del usuario. Elaborado por: Flores, Riofrío.

En tabla 11, se puede visualizar el requerimiento funcional ventas realizadas, para poder visualizar las ventas el usuario debe autenticarse en el sistema.

**Tabla 11**

*Requerimiento funcional 9: Ventas realizadas*

Descripción	Lista ventas realizadas.
<b>Precondición</b>	Autenticación de usuario.
<b>Entrada</b>	Mis ventas.
<b>Proceso</b>	Autenticación de usuario. Ingresar al menú, ir a mis ventas.
<b>Salida</b>	Ninguno

**Nota.** Requisito funcional: ventas realizadas por el usuario. Elaborado por: Flores, Riofrío.

En tabla 12, se puede visualizar el requerimiento funcional compras realizadas, para poder visualizar las compras el usuario debe autenticarse en el sistema.

**Tabla 12**

*Requerimiento funcional 10: Compras realizadas*

Descripción	Lista compras realizadas.
<b>Precondición</b>	Autenticación de usuario.
<b>Entrada</b>	Mis compras.
<b>Proceso</b>	Autenticación de usuario. Ingresar al menú, ir a mis compras.
<b>Salida</b>	Ninguno

*Nota.* Requisito funcional: compras realizadas, compras realizadas por el usuario. Elaborado por: Flores, Riofrío.

## 2.1.2. REQUERIMIENTOS NO FUNCIONALES

**Tabla 13**

*Requisitos no funcionales*

<b>REQUISITOS ORGANIZATIVOS</b>	
Restricciones Técnicas	
<b>Identificador</b>	<b>Descripción</b>
<b>RNF1</b>	El sistema utilizara la herramienta PhpMyAdmin para la administración de la base de datos.
<b>RNF2</b>	El sistema será desarrollado mediante el lenguaje de programación Java utilizando el entorno de desarrollo Android Studio versión 2.1.
<b>RNF3</b>	El sistema será compatible con versiones superiores a API 26 (versión 8 de Android).
<b>REQUISITOS DEL PRODUCTO</b>	
<b>RENDIMIENTO</b>	
<b>RNF4</b>	La aplicación debe permitir el acceso por lo menos a 100 usuarios simultáneamente.
<b>SEGURIDAD</b>	
<b>RNF5</b>	La encriptación implementada para generar las contraseñas será mediante el algoritmo de encriptación AES.
<b>FIABILIDAD</b>	
<b>RNF6</b>	El sistema deberá proporcionar un funcionamiento estable durante las 24 horas del día.
<b>USABILIDAD</b>	



<b>RNF7</b>	Proporcionar manual de usabilidad para una mejor comprensión al momento de acceder a la aplicación.
<b>RNF8</b>	El sistema mostrará mensajes que ayuden al usuario a identificar cada proceso que se realizó.
<b>RNF9</b>	El sistema debe visualizar un diseño amigable y sencillo con el cual el usuario pueda manipular la aplicación sin ningún problema.
<b>PORTABILIDAD</b>	
<b>RNF10</b>	La App será alojada temporalmente en un OneDrive en formato APK para que los usuarios puedan descargarse y realizar la compra-venta de productos.

*Nota.* Propiedades de los requisitos no funcionales del sistema. Elaborado por: Flores, Riofrío.

## 2.2.DISEÑO Y MODELADO DEL SISTEMA

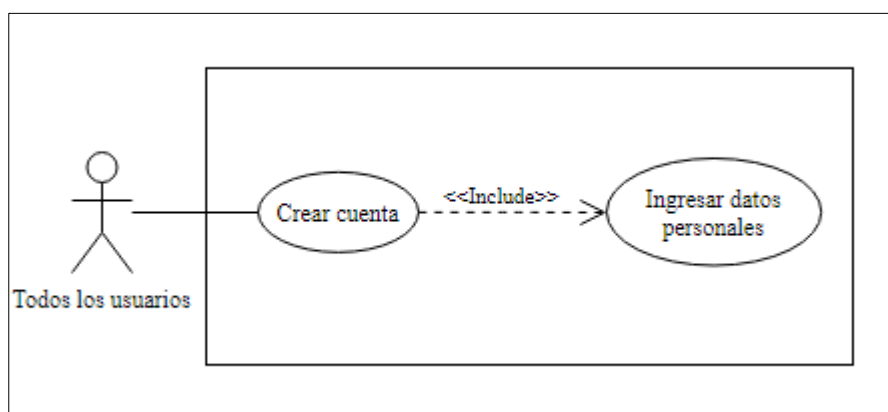
### 2.2.1. Diagramas de caso de uso (CU)

#### *Caso de uso: crear nueva cuenta*

En la figura 4, se puede observar el diagrama de casos de uso para la creación de una nueva cuenta para el usuario, representado como actor todos los usuarios, de esta manera se describe las actividades que se realiza para crear una nueva cuneta en el sistema.

**Figura 4**

*CU: Crear cuenta*



*Nota.* Actividad que se realiza para crear una nueva cuenta. Elaborado por: Flores, Riofrío.

**Tabla 14**

*Escenario de caso de uso: Crear nueva cuenta*

Nombre: <b>Crear nueva cuenta</b>	
<b>Actor/es</b>	Productor/Cliente
<b>Descripción:</b>	Permite al usuario crear una nueva cuenta para acceder al sistema
<b>Flujo Normal</b>	<ol style="list-style-type: none"> <li>1. El usuario selecciona la opción de crear cuenta.</li> <li>2. El sistema muestra en pantalla un formulario de registro.</li> <li>3. El usuario ingresa sus datos personales y envía a guardar sus credenciales.</li> <li>4. El sistema envía un mail de confirmación de cuenta.</li> <li>5. El usuario activa la cuenta mediante el correo electrónico.</li> <li>6. El usuario puede ingresar al sistema.</li> </ol>
<b>Flujo Alternativo</b>	<ol style="list-style-type: none"> <li>3. El sistema identifica campos obligatorios en blanco y campos con datos incorrectos e informa los campos obligatorios y erróneos.</li> <li>4. El sistema detecta que el usuario ya existe o es inválido e informa el problema al usuario.</li> <li>4. Si el usuario no confirma la cuenta no podrá acceder al sistema.</li> </ol>

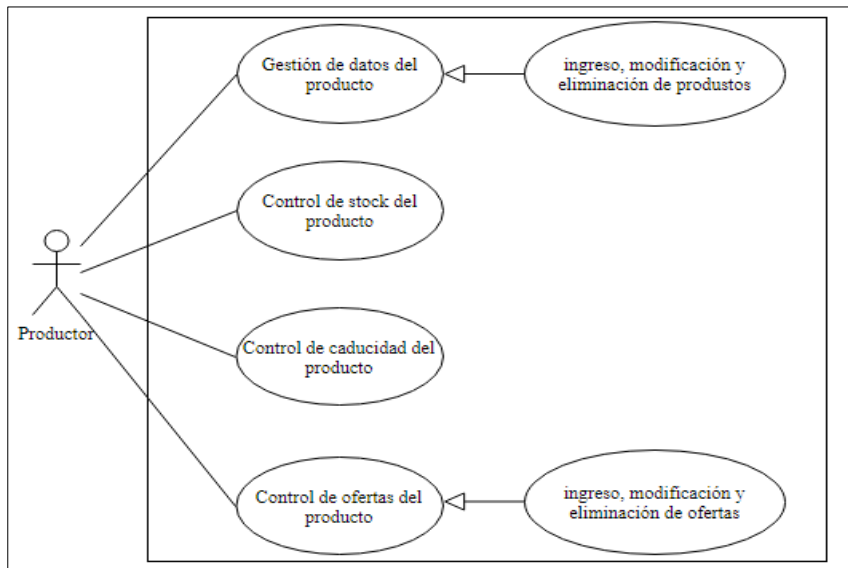
*Nota.* Descripción de los procesos para crear una nueva cuenta. Elaborado por: Flores, Riofrío.

***Caso de uso: productor***

En la figura 5, se observa el diagrama de caso de uso para la gestión de información del producto, la cual describe las acciones que debe realizar el productor que va a vender sus productos en por medio del sistema.

**Figura 5**

*CU: Productor*



**Nota.** Actividad que se realiza para la gestión del producto. Elaborado por: Flores, Riofrío.

**Tabla 15**

*Escenario de caso de uso: gestión del producto*

Nombre: <b>Gestión de Información del Producto</b>	
<b>Actor/es</b>	Productor
<b>Descripción:</b>	Permite al productor gestionar los datos del producto.
<b>Flujo Normal</b>	<ol style="list-style-type: none"> <li>1. El usuario previamente registrado en el sistema selecciona la opción de ingresar productos.</li> <li>2. El sistema muestra en pantalla un formulario de ingreso de datos del producto.</li> <li>3. El usuario ingresa la información del producto y envía a guardar.</li> <li>4. El sistema notifica que el producto fue registrado.</li> <li>5. El sistema publica el producto en la pantalla principal donde se encuentran todos los productos en venta.</li> </ol>

	6. El sistema genera un control de ventas, para que el usuario pueda tener información de ventas realizadas.
<b>Flujo Alternativo</b>	3. El sistema identifica campos obligatorios en blanco e informa los campos erróneos.

*Nota.* Descripción de los procesos que se realiza para ingresar, modificar y eliminar. Elaborado por: Flores, Riofrío.

**Tabla 16**

*Escenario de caso de uso: control de stock y caducidad del producto*

Nombre: <b>Control de Stock y Caducidad del Producto</b>	
<b>Actor/es</b>	Productor
<b>Descripción:</b>	El productor tendrá un control de stock y caducidad de sus productos.
<b>Flujo Normal</b>	<ol style="list-style-type: none"> <li>1. El usuario ingresa al sistema.</li> <li>2. El usuario ingresa a la opción de notificaciones o a su vez le llegara un mensaje al correo electrónico.</li> <li>3. El sistema informa al usuario si el stock del producto está por terminarse.</li> <li>4. El sistema informara al usuario si el producto ya caduco.</li> </ol>
<b>Flujo Alternativo</b>	<ol style="list-style-type: none"> <li>3. El sistema detecta que el producto es <math>\leq</math> a 2 notifica al usuario el inconveniente.</li> <li>4. El sistema detecta que el producto se va a caducar entonces notifica al usuario 5 días antes de su caducidad.</li> </ol>

*Nota.* Proceso que controla el stock y la caducidad del producto. Elaborado por: Flores, Riofrío.

**Tabla 17***Escenario de caso de uso: ofertas*

Nombre: Control de Ofertas del Producto	
<b>Actor/es</b>	Productor
<b>Descripción:</b>	El productor tendrá un control de ofertas de sus productos.
<b>Flujo Normal</b>	<ol style="list-style-type: none"> <li>1. El usuario ingresa al sistema.</li> <li>2. El usuario ingresa a la opción de ingresar producto.</li> <li>3. El sistema muestra un formulario de registro de producto.</li> <li>4. El usuario ingresa los datos del producto especificando en el campo oferta su descripción.</li> <li>5. La oferta del producto se publica en la interfaz de ofertas.</li> </ol>
<b>Flujo Alternativo</b>	<ol style="list-style-type: none"> <li>4.. El sistema identifica campos obligatorios en blanco e informa los campos erróneos.</li> <li>5. Si el usuario no ingresa una descripción en el campo oferta el sistema registrara un producto normal.</li> </ol>

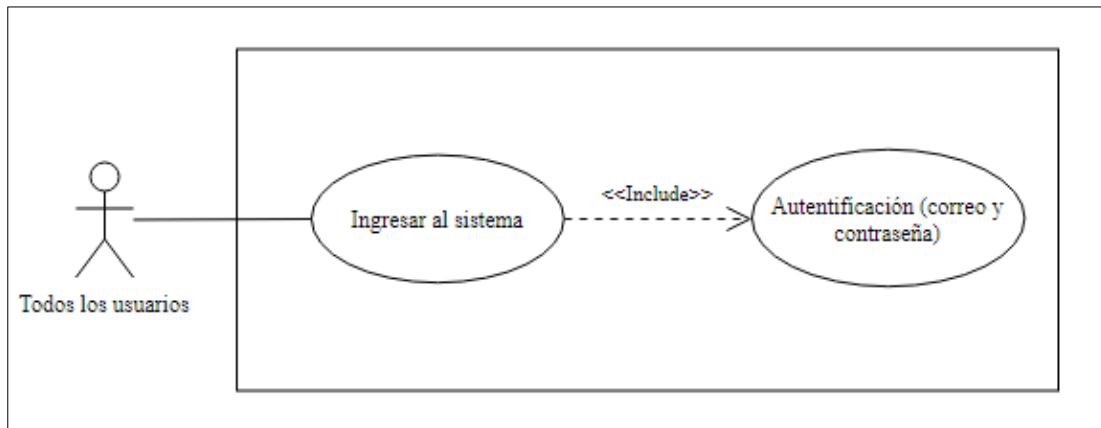
*Nota.* Descripción del caso de uso para ingresar una oferta del producto. Elaborado por: Flores, Riofrío.

#### ***Caso de uso: iniciar sesión***

En la figura 6, se puede observar mediante un diagrama de caso de uso como se realiza el proceso de inicio de sesión de los usuarios, especifica las acciones que debe realizar el usuario para ingresar al sistema.

**Figura 6**

CU: Iniciar sesión



**Nota.** Actividad que se realiza para ingresar al sistema. Elaborado por: Flores, Riofrío.

**Tabla 18**

Escenario de caso de uso: iniciar sesión

Nombre: Iniciar Sesión	
<b>Actor/es</b>	Productor/Cliente
<b>Descripción:</b>	Permite al usuario iniciar sesión para acceder al sistema.
<b>Flujo Normal</b>	<ol style="list-style-type: none"><li>1. El usuario selecciona la opción de iniciar sesión.</li><li>2. El sistema muestra la interfaz de inicio de sesión.</li><li>3. El usuario ingresa correo y contraseña.</li><li>4. El usuario ingreso con éxito al sistema.</li></ol>
<b>Flujo Alternativo</b>	<ol style="list-style-type: none"><li>3. El sistema identifica campos obligatorios en blanco y campos con datos incorrectos e informa los inconvenientes.</li></ol>

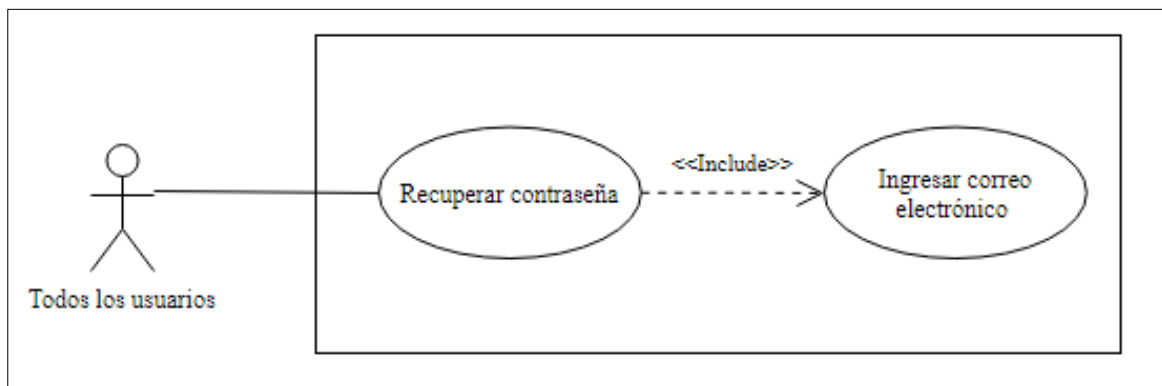
**Nota.** Descripción para ingresar al sistema mediante la autenticación. Elaborado por: Flores, Riofrío.

**Caso de uso: recuperar contraseña**

En la figura 7, se evidencia el caso de uso para realizar el proceso de recuperación de contraseña en caso de que el usuario no la recuerde su clave, para realizar la transacción el usuario debe estar registrado en el sistema y así ingresar su correo para obtener una clave temporal.

**Figura 7**

*CU: Recuperar contraseña*



**Nota.** Actividad que se realiza para recuperar una contraseña. Elaborado por: Flores, Riofrío.

**Tabla 19**

*Escenario de caso de uso: recuperar contraseña*

Nombre: Recuperar de contraseña	
<b>Actor/es</b>	Productor/Cliente
<b>Descripción:</b>	El usuario en caso de olvidar su contraseña podrá generar una nueva contraseña.
<b>Flujo Normal</b>	<ol style="list-style-type: none"> <li>1. El usuario ingresa a la opción iniciar sesión.</li> <li>2. El usuario selecciona la opción recuperar contraseña.</li> <li>3. El usuario ingresa su correo.</li> <li>4. El sistema envía un correo indicando la nueva contraseña.</li> </ol>
<b>Flujo Alternativo</b>	<ol style="list-style-type: none"> <li>3. El sistema detecta que el correo es incorrecto e informa al usuario del mail invalido.</li> </ol>

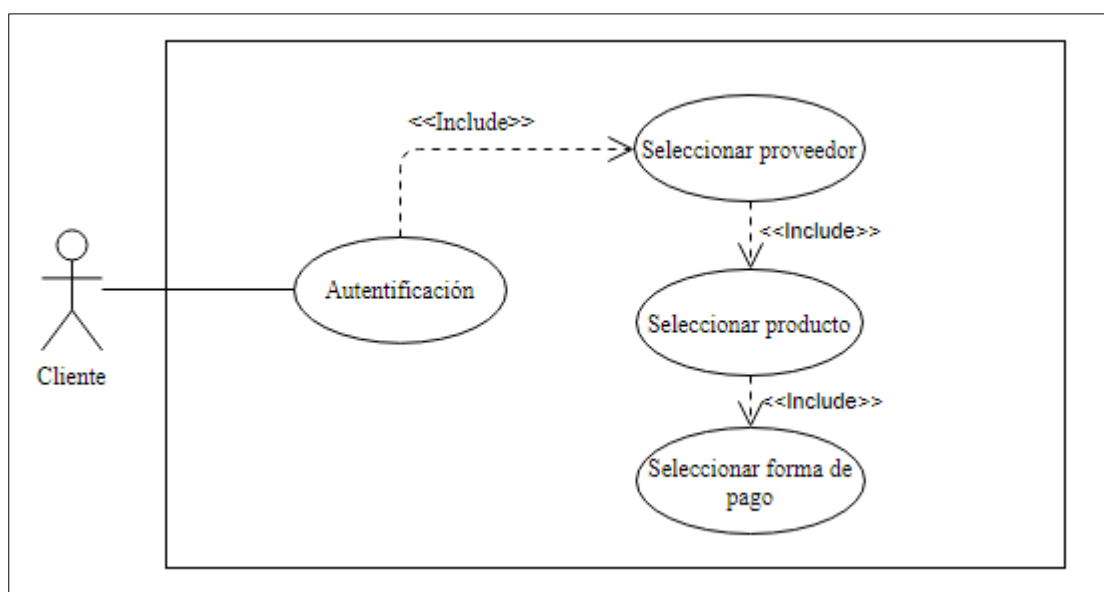
**Nota.** Describe el proceso para recuperar una contraseña. Elaborado por: Flores, Riofrío.

**Caso de uso: realizar compra**

En la figura 8, se puede visualizar el caso de uso para realizar el proceso de una compra mediante la aplicación móvil, en la que el usuario primero debe autenticarse, seleccionar un producto, agregar al carrito de compras, seleccionar forma de pago y por último finalizar la compra.

**Figura 8**

*CU: Realizar una compra*



**Nota.** Actividad que se realiza para generar una compra del producto. Elaborado por: Flores, Riofrío.

**Tabla 20**

*Escenario de caso de uso: realizar compras*

Nombre: Realizar Compras	
<b>Actor/es</b>	Cliente
<b>Descripción:</b>	Permite al usuario ingresar al sistema para realizar una compra.
<b>Flujo Normal</b>	1. El usuario ingresa a la App móvil.



	<ol style="list-style-type: none"> <li>2. El sistema carga una pantalla de inicio de sesión y omitir inicio de sesión.</li> <li>3. Si el usuario selecciona la opción de inicio de sesión. <ol style="list-style-type: none"> <li>3.1. El sistema carga en pantalla un formulario de ingreso a la App móvil.</li> <li>3.2. El usuario ingresa su correo y contraseña.</li> <li>3.3. El sistema verifica la autenticación e ingresa exitosamente.</li> <li>3.4. El usuario selecciona un producto.</li> <li>3.5. El usuario realiza la compra.</li> <li>3.6. El usuario selecciona un método de pago.</li> <li>3.7. El sistema genera un detalle de compra que es enviada al correo.</li> </ol> </li> <li>4. Si el usuario selecciona la opción omitir inicio de sesión. <ol style="list-style-type: none"> <li>4.1. El sistema se dirige a la pantalla principal de productos.</li> <li>4.2. El usuario selecciona el producto.</li> <li>4.3. El usuario desea realizar la compra.</li> </ol> </li> </ol>
<p><b>Flujo</b></p> <p><b>Alternativo</b></p>	<ol style="list-style-type: none"> <li>1.3.1. El sistema detecta que existen campos vacíos e informa la existencia de campos en blanco.</li> <li>1.3.2. El sistema detecta que el correo y/o contraseña son incorrectos e informa al usuario.</li> <li>3.6.1 El sistema identifica que no selecciona forma de pago e informa al usuario.</li> <li>4.3 El sistema identifica que no inicio sesión e informa al usuario.</li> <li>4.3.1 El sistema redirige al usuario a la pantalla de inicio de sesión.</li> </ol>

**Nota.** Describe el proceso para realizar una compra. Elaborado por: Flores, Riofrío.

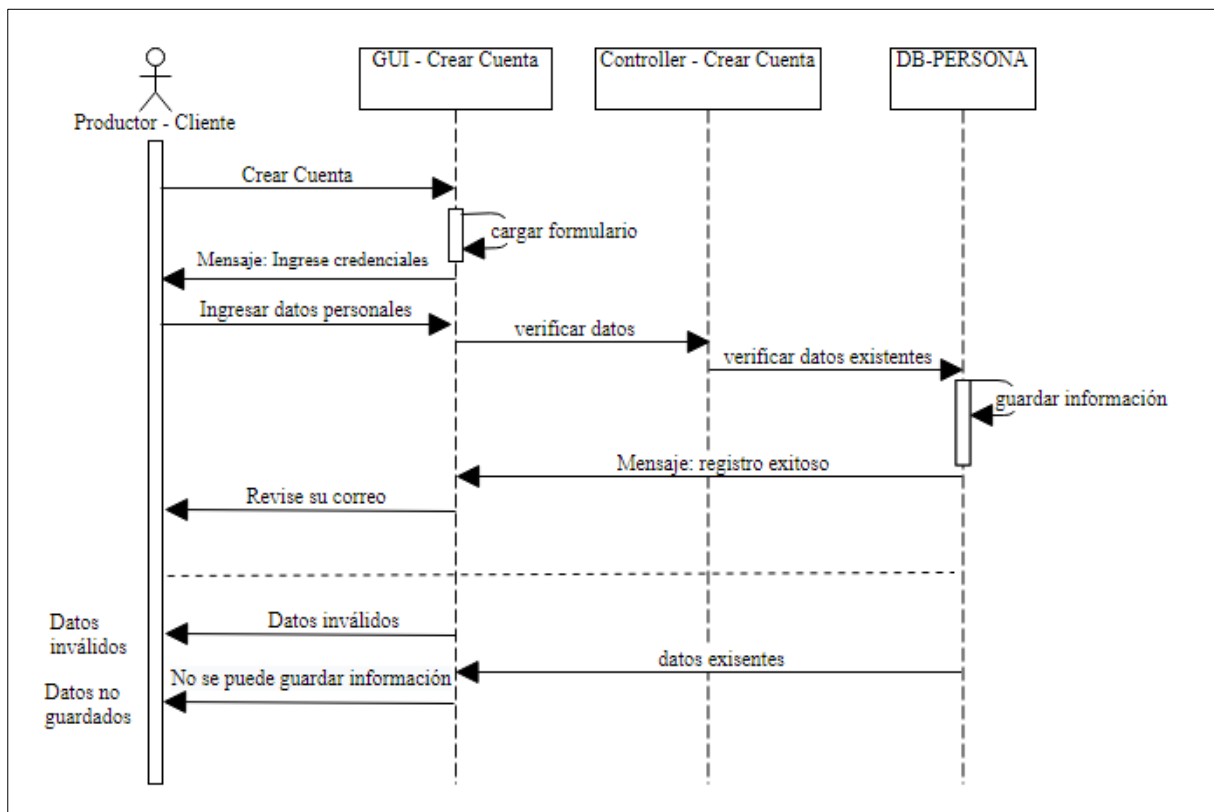
## 2.2.2 Diagramas de secuencia

### Diagrama de secuencia: crear nueva cuenta

En la figura 9, se puede visualizar las iteraciones entre objetos que se realiza en la aplicación móvil para crear una nueva cuenta donde cada usuario que se registre tendrá ingresar sus datos en un formulario que se encuentra en la App móvil para poder ingresar al sistema.

**Figura 9**

*DS: Crear nueva cuenta*



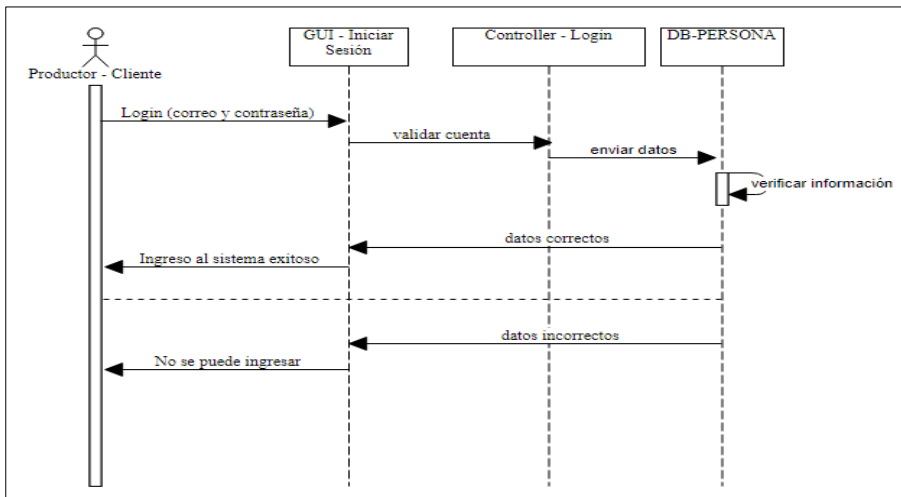
**Nota.** Describe las iteraciones realizadas en el proceso crear nueva cuenta. Elaborado por: Flores, Riofrío.

### Diagrama de secuencia: iniciar sesión

En la figura 10, se puede visualizar las iteraciones entre objetos que se realiza en la aplicación móvil para que el usuario pueda ingresar al sistema por medio del correo electrónico y contraseña registrados.

**Figura 10**

*DS: Iniciar sesión*



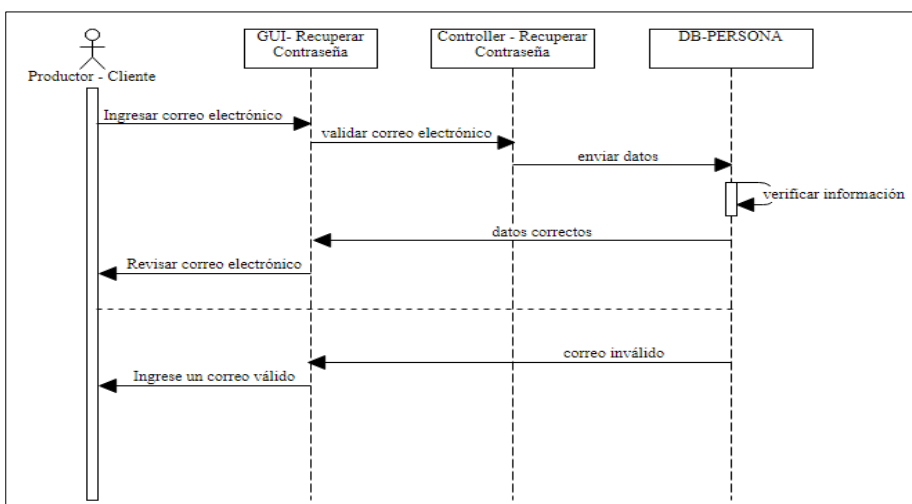
**Nota.** Describe las iteraciones realizadas para ingresar al sistema. Elaborado por: Flores, Riofrío.

**Diagrama de secuencia: recuperar contraseña**

En la figura 11, se puede visualizar las iteraciones entre objetos que se realiza en la aplicación móvil para recuperar una contraseña, el proceso se llevara a cabo siempre y cuando el usuario se encuentre registrado en el sistema.

**Figura 11**

*DS: Recuperar contraseña*



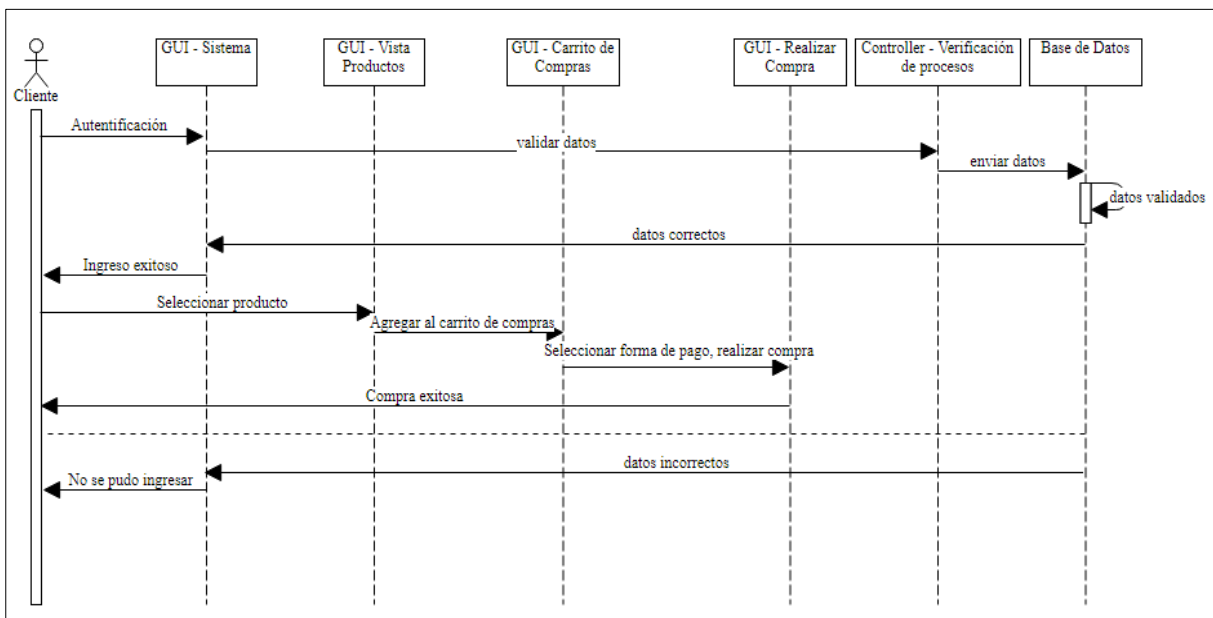
**Nota.** Describe las iteraciones para recuperar contraseña. Elaborado por: Flores, Riofrío.

### ***Diagrama de secuencia: realizar compra***

En la figura 12, se puede visualizar las iteraciones entre objetos que se realiza en la aplicación móvil para generar una orden de compra en la que interactúa el usuario con el sistema.

**Figura 12**

*DS: Realizar compras*



**Nota.** Describe las iteraciones realizadas en el proceso para generar una compra. Elaborado por: Flores, Riofrío.

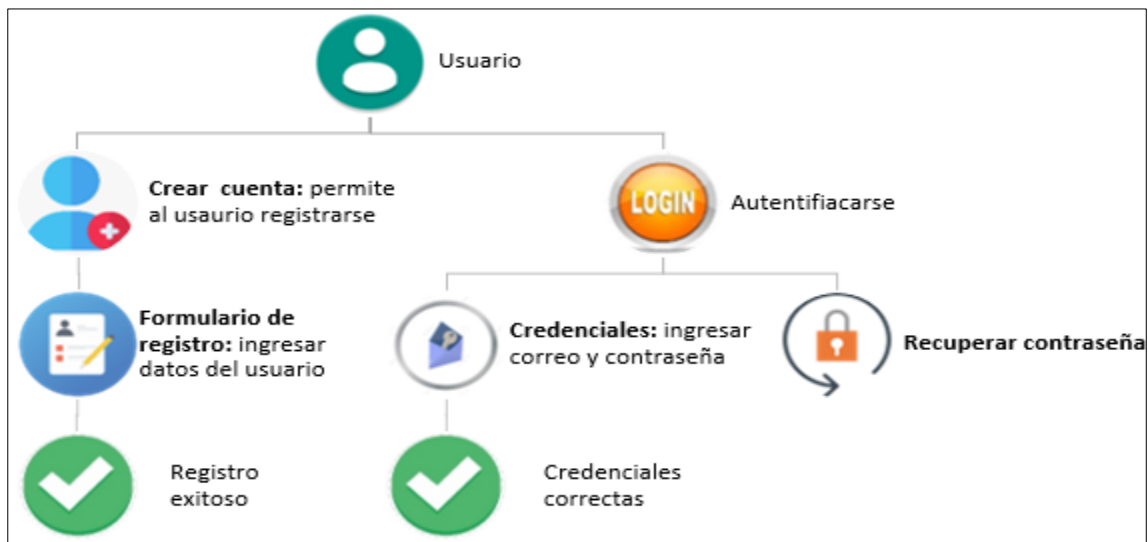
### ***2.2.3. Menú de navegación***

#### ***Menú: autenticación***

En la figura 13, se puede observar una representación gráfica de navegación del usuario al ingresar al sistema.

**Figura 13**

*Esquema de navegación ingreso al sistema*



**Nota.** Mediante un esquema visualiza los procesos para ingresar al sistema. Elaborado por: Flores, Riofrío.

***Menú: gestión de información del producto***

En la figura 14, se puede observar una representación gráfica de navegación del usuario al gestionar la información del producto mediante el ingreso, modificación y eliminación de datos.

**Figura 14**

*Esquema de gestión de información del producto*



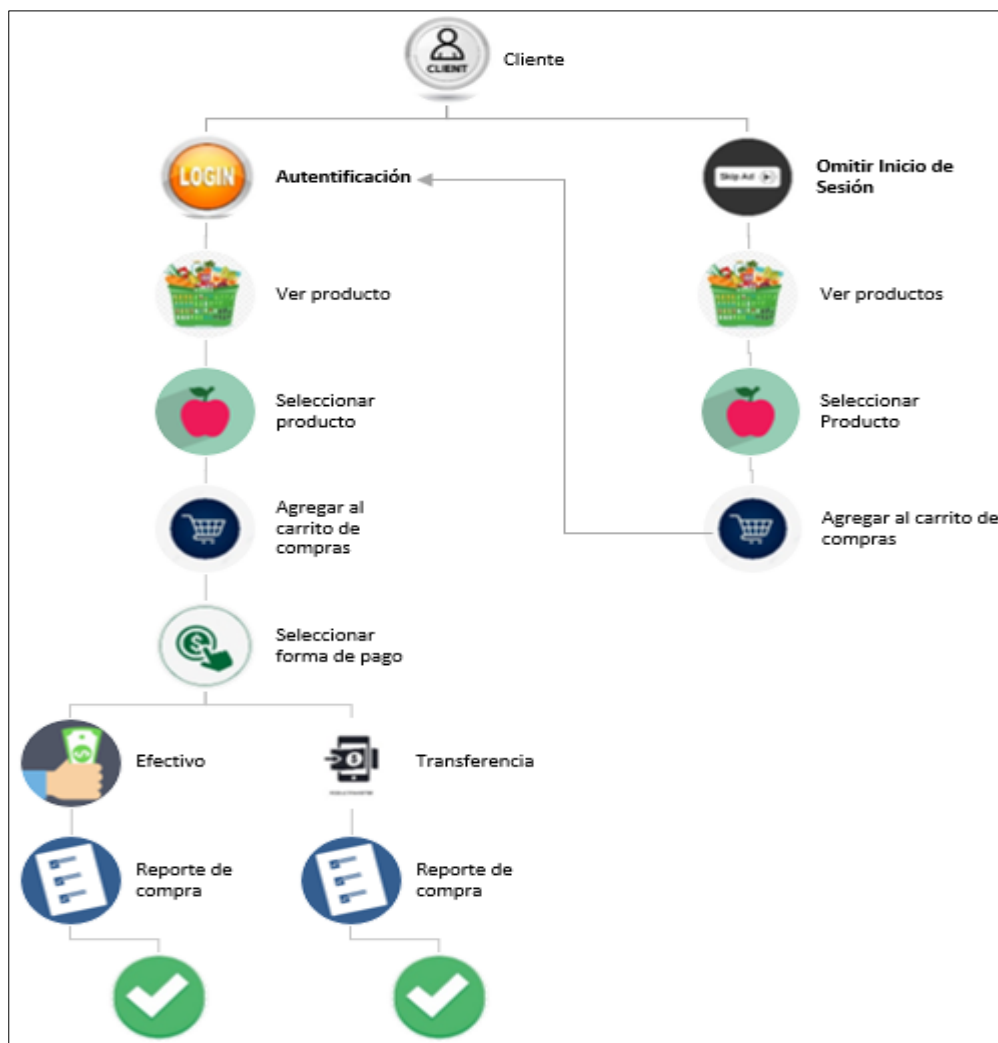
*Nota.* Mediante un esquema visualiza los procesos para ingresar, modificar, controlar el stock y caducidad del producto. Elaborado por: Flores, Riofrío.

**Menú: realizar compra**

En la figura 15, se puede observar una representación gráfica de navegación del usuario al realizar una compra. El usuario podrá conocer los pasos que se efectúa para seleccionar un producto, agregarlo al carrito de compras y finalmente ejecutará la transacción de compras.

**Figura 15**

*Esquema de navegación realizar una compra*

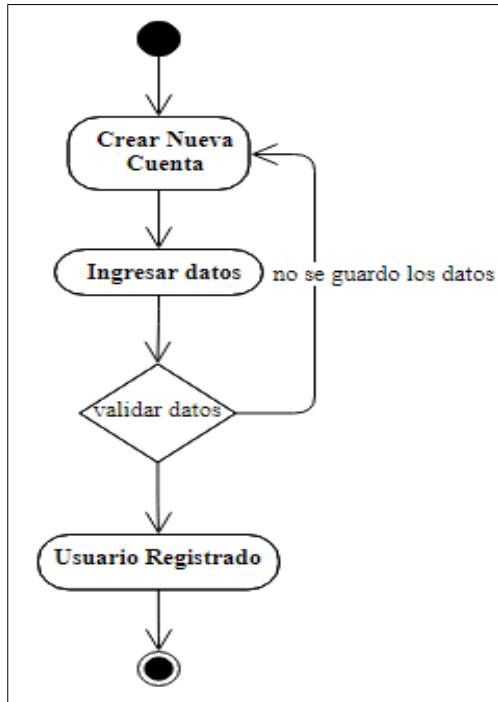


*Nota.* Mediante un esquema visualiza los procesos para realizar una compra. Elaborado por: Flores, Riofrío.

#### 2.2.4. Diagramas de actividades

Figura 16

Flujograma crear nueva cuenta

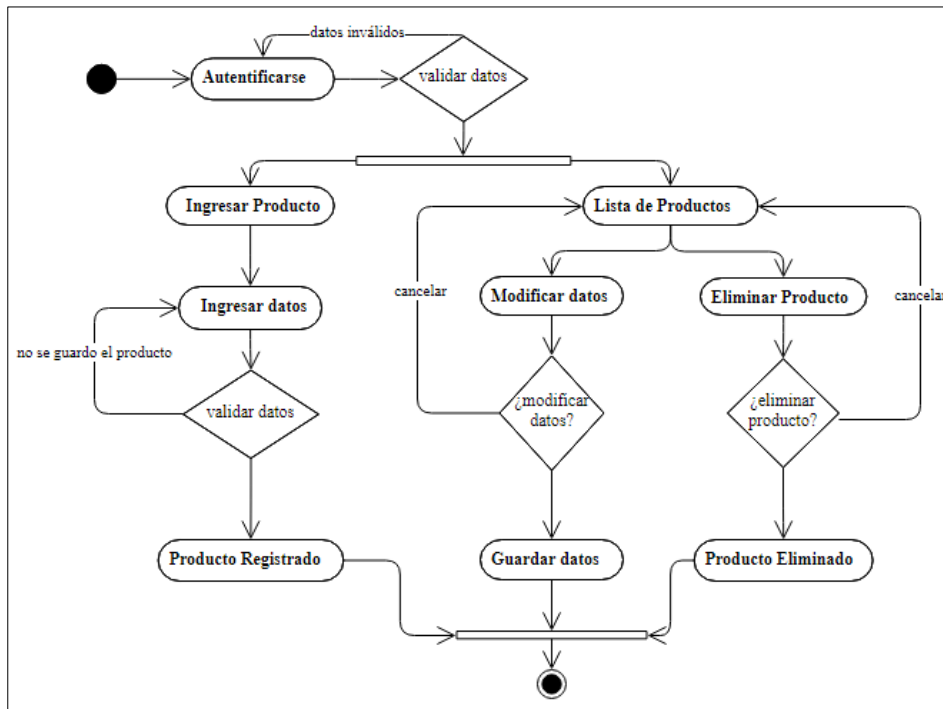


**Nota.** Flujograma que representa el algoritmo de crear nueva cuenta. Elaborado por: Flores, Riofrío.

En la figura 16, se observa el diagrama de flujo de actividades para crear una nueva cuenta, proceso por el cual el usuario ingresa sus datos personales para registrarse los cuales serán validados antes de guardarlos, si algún campo está vacío o el usuario ingresa mal un dato se le notificara por medio de un mensaje que no se guardó o existe un error en la información ingresada.

**Figura 17**

*Flujograma ingresar, modificar y eliminar producto*



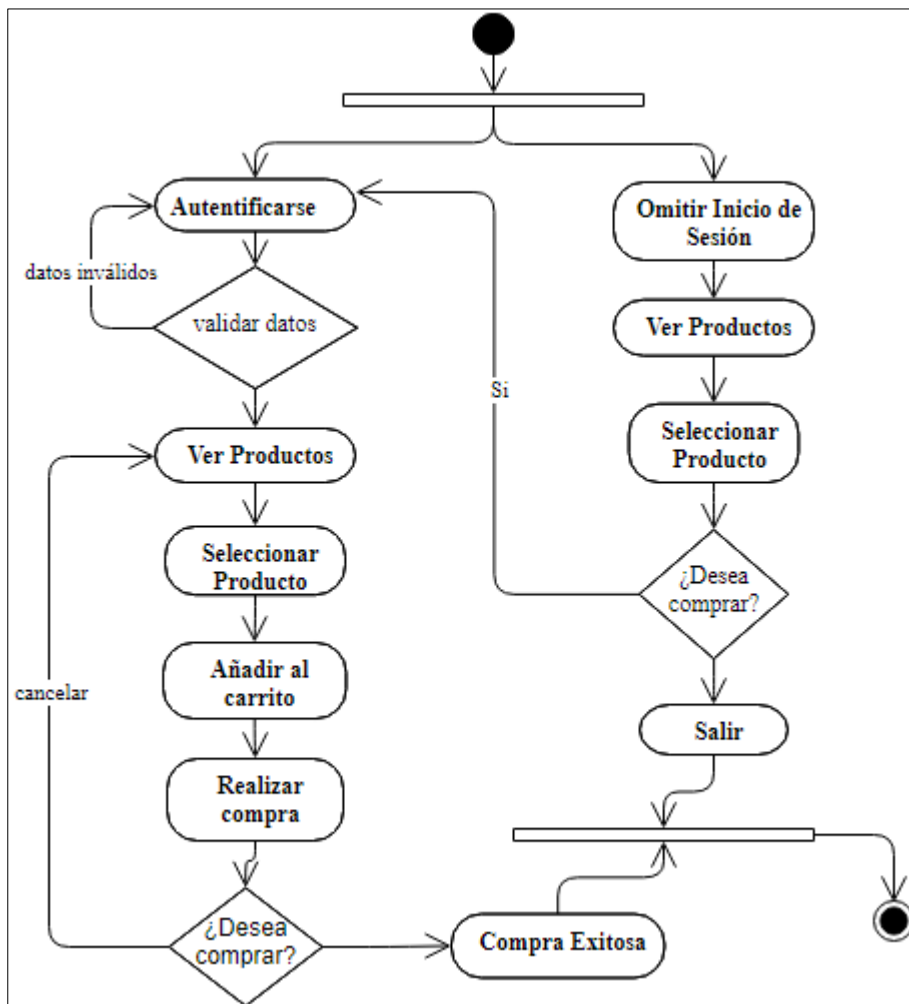
**Nota.** Flujograma que representa el algoritmo que permite realizar la gestión de información de los productos. Elaborado por: Flores, Riofrío.

En la figura 17, se observa el diagrama de flujo de actividades para gestionar información del producto, proceso que debe llevar a cabo el productor al momento de ingresar modificar y eliminar un producto.



**Figura 18**

*Flujograma realizar compra*



**Nota.** Flujograma que representa el algoritmo que permite ingresar al sistema. Elaborado por: Flores, Riofrío.

En la figura 18, se observa el diagrama de flujo de actividades para realizar una compra, proceso que permite seleccionar, ver detalle, añadir al carrito y comprar un producto

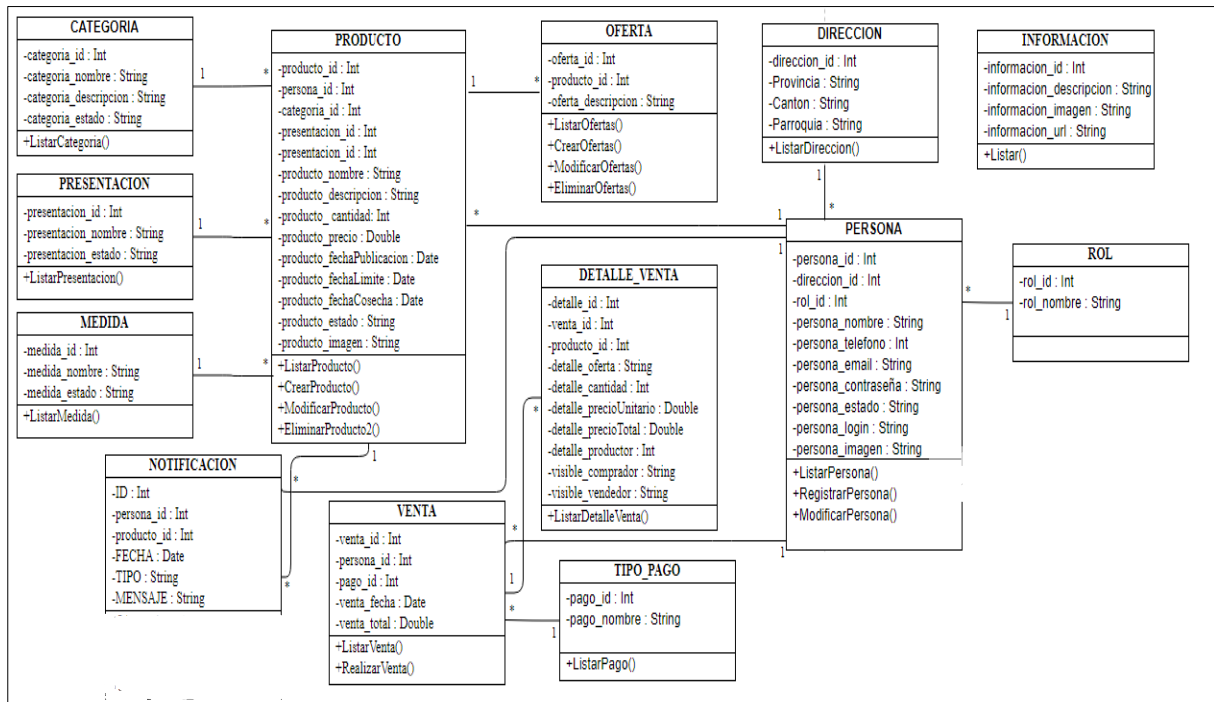
### ***Diagrama de clases***

En la figura 19, se puede observar el diseño del diagrama de clases, donde se visualiza las relaciones existentes de las clases con sus respectivos atributos y métodos.

## 2.2.5. Diagrama de clases

Figura 19

Diagrama de clases



**Nota.** Diagrama de clases de la aplicación móvil que describe sus atributos y métodos.

Elaborado por: Flores, Riofrío.

En la figura 19, se observa la definición de las clases y relaciones que existe en la construcción del sistema las cuales son representadas gráficamente en una colección de nodos en la que se puede evidenciar las clases, métodos, atributos y relaciones.

A continuación, en la tabla 21, se describe las clases más importantes utilizadas en la App móvil.

Tabla 21

Descripción de las clases de la aplicación móvil

Clase	Descripción	Función
<b>PERSONA</b>		
<b>RegistrarPersona()</b>	Registrarse en la App móvil	Permite realizar el registro de usuarios en el sistema.

<b>ListarPersona()</b>	Lista datos del usuario	Permite visualizar el perfil del usuario.
<b>ModificarPersona()</b>	Actualiza datos del usuario	Permite modificar información del usuario.
<b>DIRECCIÓN</b>		
<b>ListarDirecciones</b>	Lista direcciones	Permite listar las direcciones (Provincia, Cantón y Parroquia)
<b>PRODUCTO</b>		
<b>CrearProducto()</b>	Crea productos	Permite ingresar información del producto.
<b>ListarProducto()</b>	Lista productos	Permite listar los productos ingresados.
<b>ModificarProducto()</b>	Modifica productos	Modifica los productos ingresados.
<b>EliminarProducto()</b>	Elimina productos	Elimina productos no deseados.
<b>CATEGORÍA</b>		
<b>ListarCategoria()</b>	Lista categorías del producto	Permite listar las categorías de los diferentes productos a ingresar.
<b>PRESENTACIÓN</b>		
<b>ListarPresentacion()</b>	Lista presentación del producto	Permite listar la presentación en la que se va a entregar el producto.
<b>MEDIDA</b>		
<b>ListarMedida()</b>	Lista la medida del producto	Lista la medida en la que se va a pesar o adquirir el producto.
<b>VENTA</b>		
<b>ListarVenta()</b>	Lista ventas realizadas	Permite visualizar ventas procesadas.
<b>RealizarVenta()</b>	Realiza la venta	Realiza el proceso de venta.

<b>DETALLE_VENTA</b>		
<b>ListarDetalleVenta()</b>	Lista el detalle de la venta	Permite visualizar los datos de la venta realizada.
<b>TIPO_PAGO</b>		
<b>ListarPago()</b>	Lista tipo de pagos	Lista las diferentes formas de pago.
<b>OFERTA</b>		
<b>CrearOferta()</b>	Crea ofertas de productos	Permite ingresar información de la oferta del producto.
<b>ListarOferta()</b>	Lista ofertas de productos	Permite listar los productos con ofertas.
<b>ModificarOferta()</b>	Modifica ofertas de productos	Modifica los productos con ofertas.
<b>EliminarOferta()</b>	Elimina ofertas de productos	Elimina productos con ofertas.
<b>Clase</b>	<b>Descripción</b>	<b>Función</b>

Nota. Descripción de las clases utilizadas en la aplicación móvil. Elaborado por: Flores, Riofrío.

### 2.3. DISEÑO DE INTERFASES

La propuesta y el prototipo de diseño de las interfaces se la realizo en Balsamiq Wireframes, que es una herramienta que permite realizar el diseño de las maquetas de interfaz para páginas web y aplicaciones móviles.

#### *Prototipo de interfaces del sistema*

En la figura 20, se puede apreciar un diseño de un prototipo de interfaz para la pantalla de ingreso al sistema, en el que se puede identificar tres botones: iniciar sesión, crear cuenta y omitir inicio de sesión.

## Figura 20

*Prototipo de interfaz de ingreso al sistema*



**Nota.** Permite visualizar el prototipo de la interfaz pantalla principal del sistema. Elaborado por: Flores, Riofrío.

El diseño del prototipo de interfaz para la pantalla crear nueva cuenta, como se puede ver en la figura 21, se presenta mediante un formulario en la que los usuarios deben ingresar sus datos personales.

## Figura 21

*Interfaz crear nueva cuenta*



**Nota.** Permite visualizar el prototipo de la interfaz para crear una nueva cuenta. Elaborado por: Flores, Riofrío.

En la figura 22, se puede observar el prototipo de interfaz de inicio de sesión, donde el usuario para poder acceder al sistema deberá ingresar su correo electrónico y contraseña.

### **Figura 22**

#### *Interfaz inicio de sesión*



**Nota.** Permite visualizar el prototipo de la interfaz iniciar sesión cuenta. Elaborado por: Flores, Riofrío.

En la figura 23, el diseño del prototipo de interfaz recuperar contraseña se puede observar que se realiza mediante el ingreso de un correo electrónico.

### **Figura 23**

#### *Interfaz recuperar contraseña*

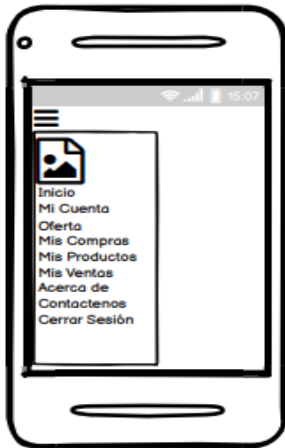


**Nota.** Permite visualizar el prototipo de la interfaz recuperar contraseña. Elaborado por: Flores, Riofrío.

En la figura 24, se puede observar el diseño del prototipo de interfaz: menú del sistema, el cual consta de los siguientes ítems: inicio, mi cuenta, oferta, mis compras, etc.

### Figura 24

*Interfaz menú del sistema*

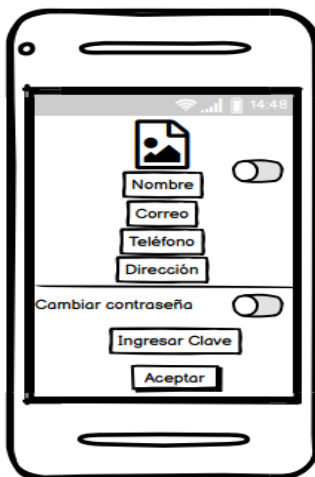


**Nota.** Permite visualizar el prototipo de la interfaz menú del sistema. Elaborado por: Flores, Riofrío.

En la figura 25, se puede observar el diseño del prototipo de interfaz: perfil del usuario, que refleja la información personal y también donde se podrá cambiar la clave.

### Figura 25

*Interfaz perfil de usuario*

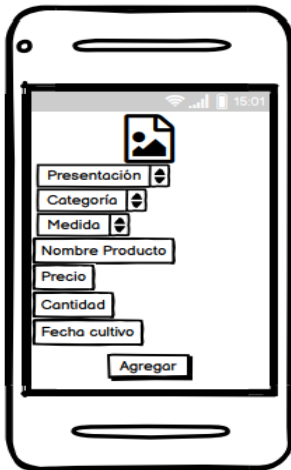


**Nota.** Permite visualizar el prototipo de la interfaz perfil de usuario. Elaborado por: Flores, Riofrío.

En la figura 26, se puede observar el diseño del prototipo de interfaz: ingresar un producto, el cual consta de un formulario para el ingreso de los datos del producto.

**Figura 26**

*Interfaz ingresar producto*

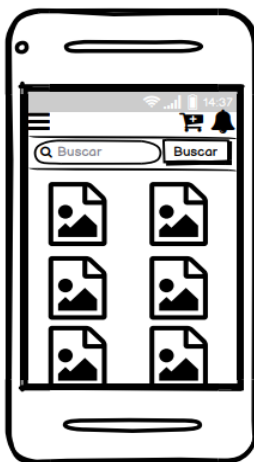


**Nota.** Permite visualizar el prototipo de la interfaz ingreso de productos. Elaborado por: Flores, Riofrío.

En la figura 27, se puede observar el diseño del prototipo de interfaz: vista principal de producto, en donde aparecerán todos los productos ingresados por los productores.

**Figura 27**

*Interfaz vista principal de productos*



**Nota.** Permite visualizar el prototipo de la interfaz vista principal de productos. Elaborado por: Flores, Riofrío.



En la figura 28, se puede observar el diseño del prototipo de interfaz: lista de productos del productor, que muestra todos los productos ingresados por el emprendedor.

### Figura 28

*Lista de productos del productor*



**Nota.** Permite visualizar el prototipo de la interfaz lista de productos ingresados por el productor. Elaborado por: Flores, Riofrío.

En la figura 29, se puede observar el diseño del prototipo de interfaz: agregar producto al carrito de compras, en que el usuario agregara un producto para realizar una compra.

### Figura 29

*Interfaz agregar producto al carrito de compras*

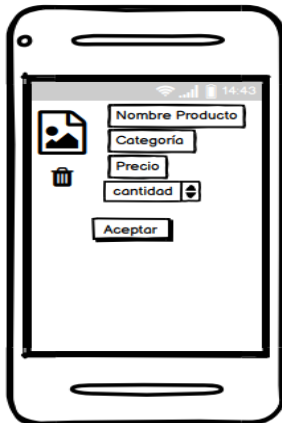


**Nota.** Permite visualizar el prototipo de la interfaz agregar al carrito de compras. Elaborado por: Flores, Riofrío.

En la figura 30, se puede observar el diseño del prototipo de interfaz: vista detalle del producto, el cual muestra los datos de cada producto registrado en el sistema.

**Figura 30**

*Interfaz vista detalle de producto*



**Nota.** Permite visualizar el prototipo de la interfaz que permite ver el detalle del producto a comprar. Elaborado por: Flores, Riofrío.

En la figura 31, se puede observar el diseño del prototipo de interfaz: finalizar compra, que muestra el detalle del producto y tres botones: finalizar compra, seguir comprando y vaciar carrito.

**Figura 31**

*Interfaz finalizar compra*



**Nota.** Permite visualizar el prototipo de la interfaz que permite finalizar la compra de los productos. Elaborado por: Flores, Riofrío.

En la figura 32, se puede observar el diseño del prototipo de interfaz: ofertas, en la que se desplegara información de productos que contengan ofertas.

### Figura 32

*Ofertas de productos*



**Nota.** Permite visualizar el prototipo de la interfaz ofertas de productos. Elaborado por: Flores, Riofrío.

En la figura 33, se puede observar el diseño del prototipo de interfaz: detalle compra, como parte final del proceso de compras se reflejará información del producto, forma de pago y del emprendedor.

### Figura 33

*Interfaz detalle compra*

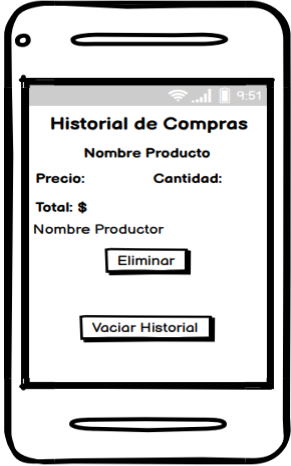


**Nota.** Permite visualizar el prototipo de la interfaz detalle de una compra. Elaborado por: Flores, Riofrío.

En la figura 34, se puede observar el diseño del prototipo de interfaz: historial de compras, en donde se almacenará las compras realizadas por los usuarios.

**Figura 34**

*Historial de compras realizadas*

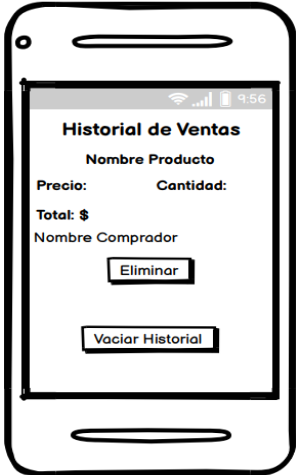


*Nota.* Permite visualizar el prototipo de la interfaz detalle de una compra. Elaborado por: Flores, Riofrío.

En la figura 35, se puede observar el diseño del prototipo de interfaz: historial de ventas, en donde se almacenará las ventas realizadas por los usuarios.

**Figura 35**

*Historial de ventas realizadas*



*Nota.* Permite visualizar el prototipo de la interfaz detalle de una compra. Elaborado por: Flores, Riofrío.

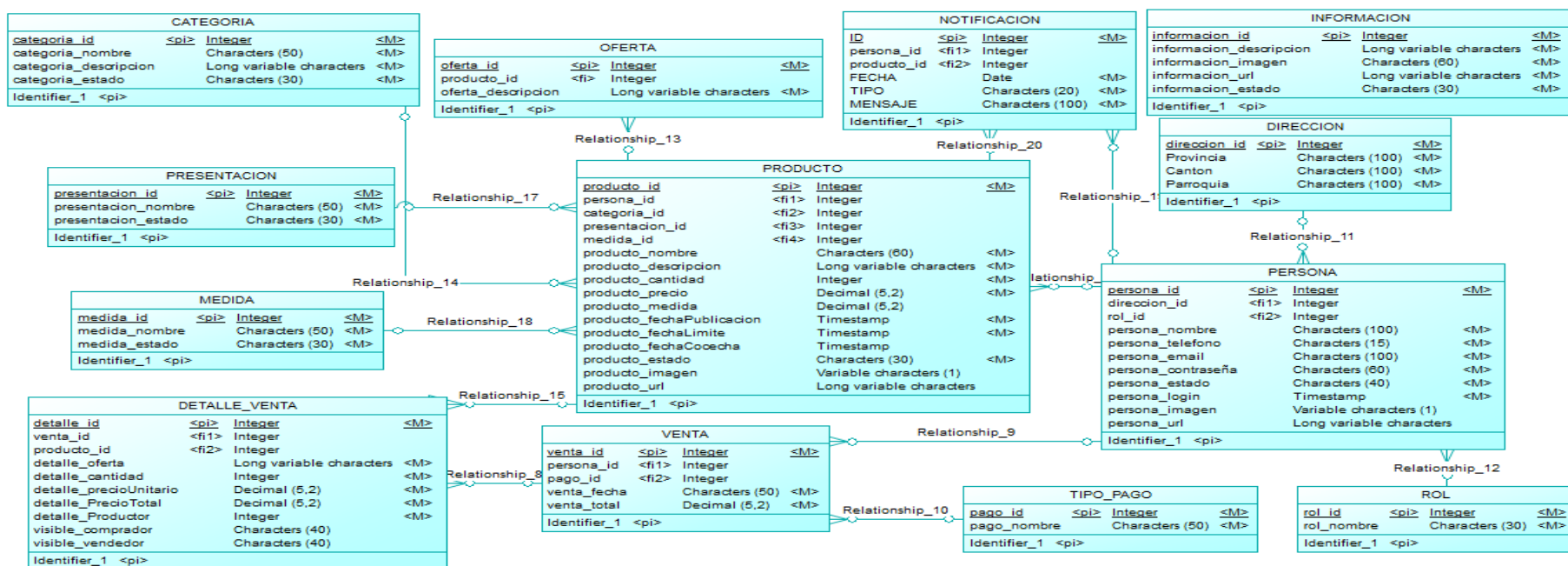
## 2.4. ESTRUCTURA DE LA BASE DE DATOS

### *Diseño lógico*

En el diseño de la base de datos se puede visualizar las tablas, campos y relaciones que se establecen para la implementación del sistema.

**Figura 36**

*Diseño lógico de la base de datos*

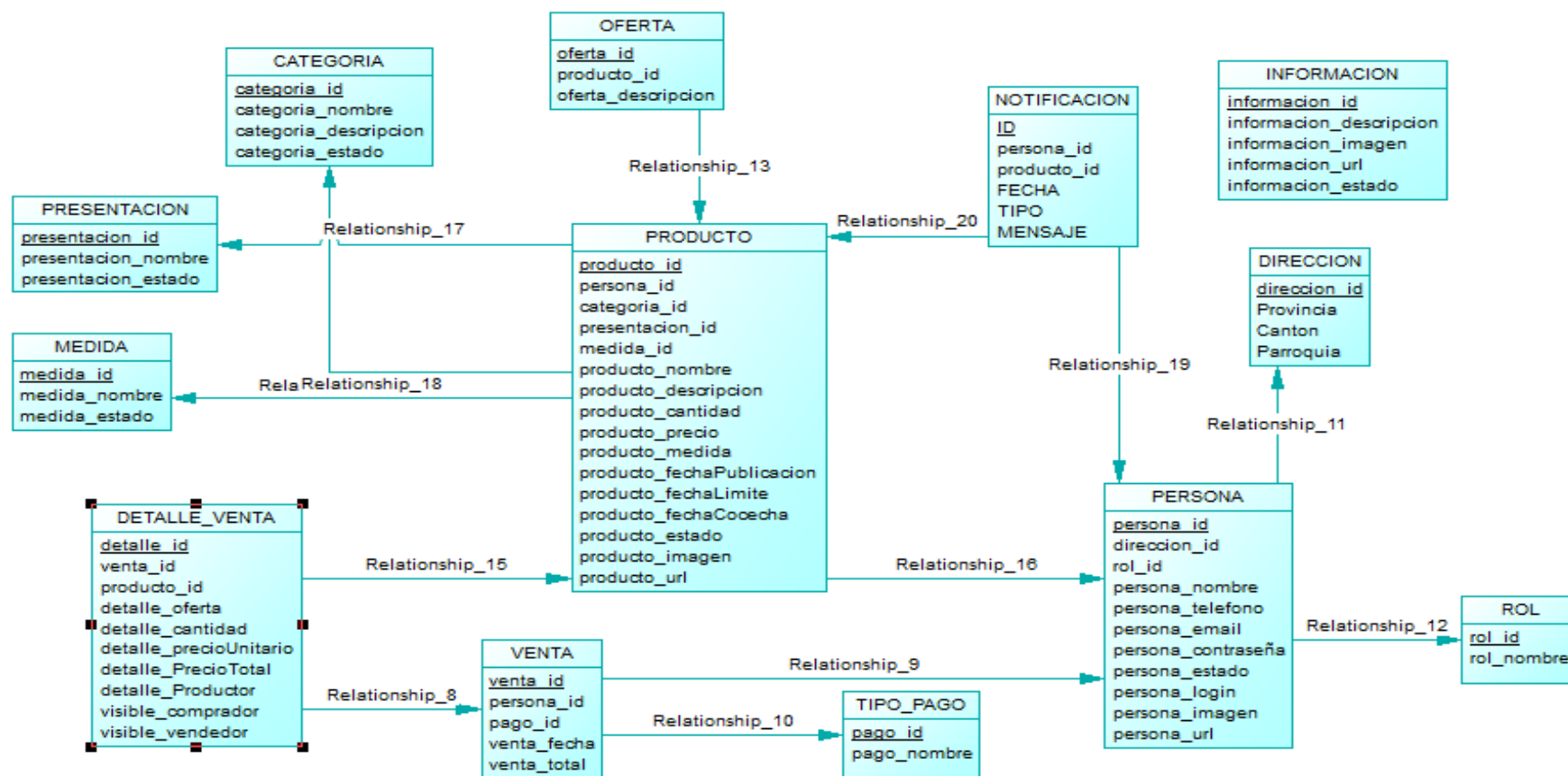


**Nota.** Modelo lógico de base de datos para la implementación de la aplicación. Elaborado por: Flores, Riofrío.

## Diseño físico

Figura 37

Diseño físico de la base de datos



**Nota.** Modelo físico de base de datos para la implementación de la aplicación. Elaborado por: Flores, Riofrío.

### *Diccionario de datos*

Como se puede visualizar el diseño lógico de la base de datos en la figura 34, a continuación, se describirá cada tabla establecida en el diseño, dando a conocer que la base de datos tiene un uso en común con la página web desarrollada por otro grupo de estudiantes y la aplicación móvil.

Cabe mencionar que la tabla información que se encuentra en el diseño de la base de datos será utilizada por la página web.

#### **Tabla 22**

##### *Descripción de la tabla PERSONA*

Nombre de la tabla: PERSONA					
Campo	Tipo de dato	Tamaño	PK	FK	Descripción
persona_id	Int	Valor por defecto	X		Identificador del usuario
persona_nombre	Characters	100			Nombre del usuario
persona_telefono	Number	15			Teléfono del usuario
persona_email	Characters	100			Correo electrónico del usuario
persona_contraseña	Characters	20			Contraseña del usuario
persona_estado	Characters	20			Visualiza el estado de la cuenta del usuario
persona_login	Timestamp	Valor por defecto			Fecha y hora del registro del usuario
persona_imagen	Variable characters	Valor por defecto			Imagen del usuario
<b>Relaciones entre tablas</b>					
direccion_id	Int	Valor por defecto		X	Relación con la tabla dirección
rol_id	Int	Valor por defecto		X	Relación con la tabla rol

*Nota.* Contiene la descripción de cada campo que conforma la tabla “PERSONA”. Elaborado por: Flores, Riofrío.

**Tabla 23***Descripción de la tabla ROL*

Nombre de la tabla: ROL					
Campo	Tipo de dato	Tamaño	PK	FK	Descripción
rol_id	Int	Valor por defecto	X		Identificador del rol del usuario
rol_nombre	Characters	30			Nombre del rol que identificara al usuario

*Nota.* Contiene la descripción de cada campo que conforma la tabla “ROL”. Elaborado por: Flores, Riofrío.

**Tabla 24***Descripción de la tabla DIRECCION*

Nombre de la tabla: DIRECCION					
Campo	Tipo de dato	Tamaño	PK	FK	Descripción
direccion_id	Int	Valor por defecto	X		Identificador de dirección
Provincia	Characters	100			Nombre de la provincia
Canton	Characters	100			Nombre del cantón
Parroquia	Characters	100			Nombre de la parroquia

*Nota.* Contiene la descripción de cada campo que conforma la tabla “DIRECCION”. Elaborado por: Flores, Riofrío.

**Tabla 25***Descripción de la tabla PRODUCTO*

Nombre de la tabla: PRODUCTO					
Campo	Tipo de dato	Tamaño	PK	FK	Descripción
producto_id	Int	Valor por defecto	X		Identificador del producto
producto_nombre	Characters	60			Nombre del producto
producto_descripcion	Long variable characters	Valor por defecto			Descripción del producto



<b>producto_cantidad</b>	Int	Valor por defecto			Cantidad del producto
<b>producto_precio</b>	Decimal	5,2			Precio del producto
<b>producto_medida</b>	Decimal	5,2			Mediada del producto, esta puede ser: onza, libra, etc.
<b>producto_fechaPublicacion</b>	Timestamp	Valor por defecto			Fecha de publicación del producto
<b>producto_fechaLimite</b>	Timestamp	Valor por defecto			Fecha de expiración del producto
<b>producto_fechaCosecha</b>	Timestamp	Valor por defecto			Fecha de cosecha del producto
<b>producto_estado</b>	Characters	30			Estado del producto
<b>producto_imagen</b>	Variable Characters	Valor por defecto			Imagen del producto
<b>producto_url</b>	Long variable characters	Valor por defecto			Este campo es usado por la página web.
<b>Relaciones entre tablas</b>					
<b>persona_id</b>	Int	Valor por defecto		X	Relación con la tabla persona
<b>categoria_id</b>	Int	Valor por defecto		X	Relación con la tabla categoría
<b>presentación_id</b>	Int	Valor por defecto		X	Relación con la tabla presentación
<b>medida_id</b>	Int	Valor por defecto		X	Relación con la tabla medida

**Nota.** Contiene la descripción de cada campo que conforma la tabla “PRODUCTO”. Elaborado por: Flores, Riofrío.

## Tabla 26

*Descripción de la tabla CATEGORIA*

Nombre de la tabla: CATEGORIA					
Campo	Tipo de dato	Tamaño	PK	FK	Descripción

<b>categoria_id</b>	Int	Valor por defecto	X		Identificador de la categoría del producto
<b>categoria_nombre</b>	Characters	50			Nombre de la categoría del producto
<b>categoria_descripcion</b>	Long variable characters	Valor por defecto			Descripción de la categoría del producto
<b>categoria_estado</b>	Characters	30			Estado de la categoría del producto

*Nota.* Contiene la descripción de cada campo que conforma la tabla “CATEGORIA”.

Elaborado por: Flores, Riofrío.

**Tabla 27**

*Descripción de la tabla PRESENTACION*

Nombre de la tabla: PRESENTACION					
Campo	Tipo de dato	Tamaño	PK	FK	Descripción
<b>presentacion_id</b>	Int	Valor por defecto	X		Identificador de la presentación del producto
<b>presentacion_nombre</b>	Characters	50			Nombre de la presentación del producto
<b>presentacion_estado</b>	Characters	50			Estado de la presentación del producto

*Nota.* Contiene la descripción de cada campo que conforma la tabla “PRESENTACION”.

Elaborado por: Flores, Riofrío.

**Tabla 28**

*Descripción de la tabla MEDIDA*

Nombre de la tabla: MEDIDA					
Campo	Tipo de dato	Tamaño	PK	FK	Descripción
<b>medida_id</b>		Valor por defecto	X		Identificador de la medida del producto
<b>medida_nombre</b>	Characters	50			Nombre de la medida del producto

<b>mediad_estado</b>	Characters	50			Estado de la mediad del producto
----------------------	------------	----	--	--	----------------------------------

**Nota.** Contiene la descripción de cada campo que conforma la tabla “MEDIDA”. Elaborado por: Flores, Riofrío.

**Tabla 29**

*Descripción de la tabla OFERTA*

Nombre de la tabla: OFERTA				
Campo	Tipo de dato	PK	FK	Descripción
<b>oferta id</b>	Int	X		Identificador de la oferta del producto
<b>oferta_descripcion</b>	Long variable characters			Descripción de la oferta del producto
Relación entre tablas				
<b>producto_id</b>	Int		X	Relación con la tabla producto

**Nota.** Contiene la descripción de cada campo que conforma la tabla “OFERTA”. Elaborado por: Flores, Riofrío.

**Tabla 30**

*Descripción de la tabla VENTA*

Nombre de la tabla: VENTA				
Campo	Tipo de dato	PK	FK	Descripción
<b>venta_id</b>	Int	X		Identificador de la venta
<b>venta_fecha</b>	Characters			Fecha de la venta realizada
<b>Venta_total</b>	Decimal			Total de la venta
Relación entre tablas				
<b>persona_id</b>	Int		X	Relación con la tabla persona
<b>pago_id</b>	Int		X	Relación con la tabla TIPO_PAGO

**Nota.** Contiene la descripción de cada campo que conforma la tabla “VENTA”. Elaborado por: Flores, Riofrío.

**Tabla 31***Descripción de la tabla TIPO\_PAGO*

Nombre de la tabla: TIPO_PAGO					
Campo	Tipo de dato	Tamaño	PK	FK	Descripción
<b>pago_id</b>	Int	Valor por defecto	X		Identificador del tipo de pago
<b>pago_nombre</b>	Characters	50			Nombre del tipo de pago

*Nota.* Contiene la descripción de cada campo que conforma la tabla “TIPO\_PAGO”. Elaborado por: Flores, Riofrío.

**Tabla 32***Descripción de la tabla DETALLE\_VENTA*

Nombre de la tabla: DETALLE_VENTA					
Campo	Tipo de dato	Tamaño	PK	FK	Descripción
<b>detalle_id</b>	Int	Valor por defecto	X		Identificador del detalle de la venta
<b>detalle_oferta</b>	Long variable characters	Valor por defecto			Detalle de la oferta del producto
<b>detalle_cantidad</b>	Int	Valor por defecto			Detalle de la cantidad del producto
<b>detalle_precioUnitario</b>	Decimal	5,2			Detalle del precio del producto
<b>detalle_PrecioTotal</b>	Decimal	5,2			Detalle del precio total del producto
<b>detalle_Productor</b>	Int	Valor por defecto			Detalle del productor
<b>visible_comprador</b>	Characters	40			Datos comprador
<b>visible_vendedor</b>	Characters	40			Datos del vendedor
<b>Relaciones entre tablas</b>					

<b>venta_id</b>	Int	Valor por defecto		X	Relación con la tabla venta
<b>producto_id</b>	Int	Valor por defecto		X	Relación con la tabla producto

**Nota.** Contiene la descripción de cada campo que conforma la tabla “DETALLE\_VENTA”.

Elaborado por: Flores, Riofrío.

### Tabla 33

*Descripción de la tabla NOTIFICACION*

Nombre de la tabla: NOTIFICACION					
Campo	Tipo de dato	Tamaño	PK	FK	Descripción
<b>ID</b>	Int	Valor por defecto	X		Identificador de la notificación
<b>FECHA</b>	Date	Valor por defecto			Fecha de la notificación
<b>TIPO</b>	Characters	20			Tipo de notificación
<b>MENSAJE</b>	Characters	20			Mensaje de la notificación
Relación entre tablas					
<b>persona_id</b>	Int	Valor por defecto		X	Relación con la tabla persona
<b>producto_id</b>	Int	Valor por defecto		X	Relación con la tabla producto

**Nota.** Contiene la descripción de cada campo que conforma la tabla “NOTIFICACION”.

Elaborado por: Flores, Riofrío.

## CAPÍTULO III

### CONSTRUCCION Y PRUEBAS

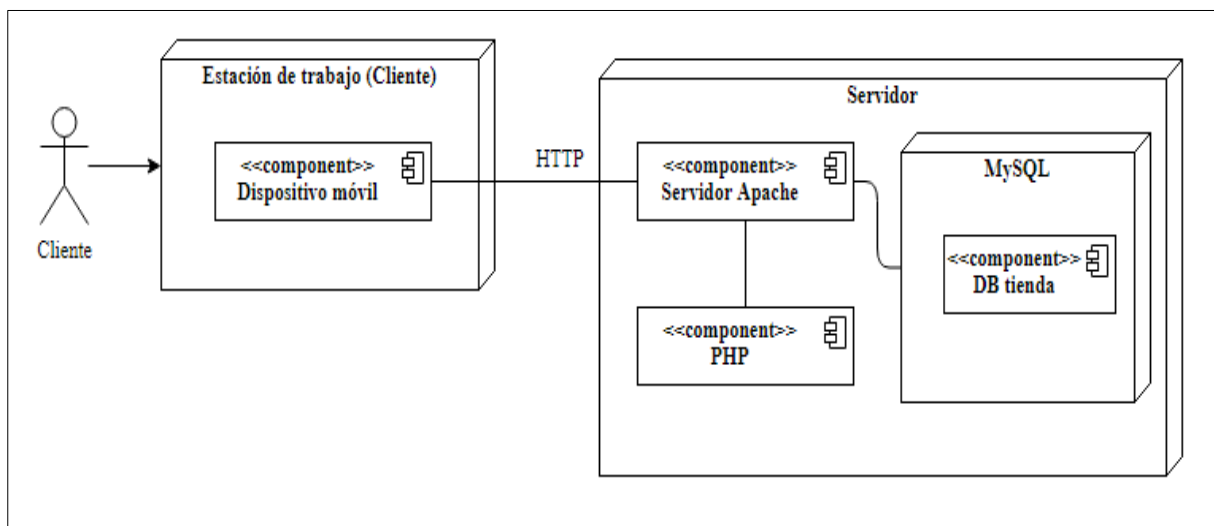
En el capítulo tres se describe el proceso de construcción de la App móvil, la implementación de la metodología aplicada y las herramientas utilizadas para la codificación de la aplicación móvil, tomando en cuenta que la elaboración del software se va a realizar para personas que no tienen mucha experiencia con la tecnología por lo que se planteó que su funcionamiento sea fácil y sencilla para el usuario.

#### 3.1. Diagrama de despliegue

En la figura 38, se puede observar el diagrama de despliegue que es utilizado para visualizar la arquitectura de ejecución del sistema. Es decir, mostrar las relaciones físicas de cada nodo de los componentes de la aplicación. Un nodo está representado por un recurso computacional de software o hardware.

**Figura 38**

*Diagrama de despliegue*



**Nota.** Diagrama de despliegue que representa los componentes de la aplicación móvil.

Elaborado por: Flores, Riofrío.

### **3.2.Herramientas**

A continuación, se describe las herramientas implementadas para el desarrollo de la aplicación móvil, las mismas que fueron seleccionadas según las especificaciones del software y por su facilidad de uso.

#### **Android Studio**

Es un entorno de desarrollo que permite crear aplicaciones móviles para dispositivos compatibles con sistemas operativos Android.

#### **Lenguaje de programación**

El desarrollo de la aplicación móvil fue elaborado en el lenguaje de programación Java, en el cual se estableció la codificación de clases, constructor y métodos.

#### **Librerías**

Volley: es una librería que permite establecer peticiones HTTP entre el servidor y las aplicaciones (Developers, 2020).

Glide: es una librería introducida por Google para Android, que permite guardar las imágenes en memoria cache (Mendoza, 2016).

#### **Web Service**

La implementación de la web service en la aplicación permitió establecer un medio de comunicación, ya que es una tecnología que utiliza un conjunto de protocolos que permiten intercambiar información entre aplicaciones informáticas.

#### **Visual Studio Code**

Para el desarrollo de la web services se utilizó Visual Studio Code que es un editor de código fuente desarrollado por Microsoft para Windows, Linux y macOS, compatible para varios lenguajes de programación con una interfaz amigable para el usuario.

## Base de Datos

Se utilizó la herramienta PhpMyAdmin que es de código abierto el cual permitió organizar y administrar la base de datos para el almacenamiento de información de los usuarios. Las peticiones de la App móvil se llevaron a cabo por medio de API REST codificadas en PHP que estableciendo la conexión enviada por el protocolo HTTP para el intercambio de información.

### 3.3. Funcionalidad y Codificación de la aplicación móvil

A continuación, se visualizará la funcionalidad y el código más relevante de algunos métodos del proceso de ejecución de la aplicación móvil.

Para utilizar varios recursos del dispositivo móvil en el desarrollo de la aplicación se necesita establecer los permisos de acceso, como es el caso de acceso al internet, cámara y memoria de almacenamiento como se puede observar en la figura # 39.

#### Figura 39

*Permisos de acceso a las aplicaciones del dispositivo móvil*

```
<!-- Permisos de acceso a las aplicaciones del dispositivo -->  
<uses-permission android:name="android.permission.INTERNET" />  
<uses-permission android:name="android.permission.CAMERA" />  
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />  
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
```

**Nota.** Permisos de acceso para el uso de recursos del dispositivo móvil. Elaborado por: Flores, Riofrío

La figura 40, visualiza la conexión PDO, que es una extensión que permite acceder a diferentes bases de datos por medio de sistemas de bases de datos como MySQL, ORACLE, SQLite.



## Figura 40

Conexión a la base de datos

```
class Conectar{
    public static function conexion(){
        try{
            $conexion = new PDO('mysql:host=localhost; dbname=data_base','root','');
            $conexion -> setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
        }catch(Exception $e){
            die("Error " . $e->getMessage());
            echo "Linea del error " . $e->getLine();
        }

        return $conexion;
    }
}
```

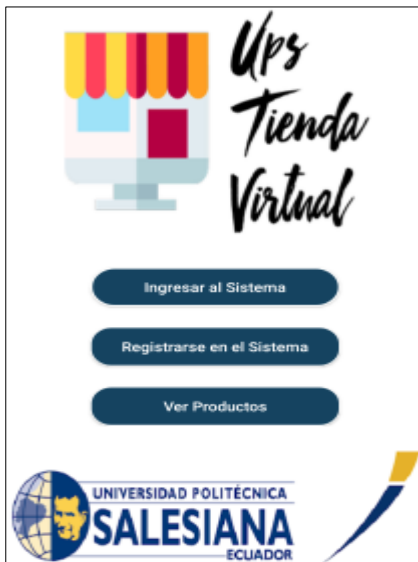
**Nota.** Conexión a la base de datos. Elaborado por: Flores, Riofrío

La conexión PDO especifica el tipo de base de base datos, que en este caso es “MYSQL” también visualiza el host, nombre de la base de datos, usuario y contraseña. El parámetro `setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION)`, permite detectar los posibles errores en la ejecución y por último la línea que identifica e imprimir el error.

En la figura 41, se puede visualizar la pantalla principal de la aplicación móvil que van a visualizar los usuarios, comprendida por tres botones: ingresar al sistema, registrarse en el sistema y ver productos.

**Figura 41**

*Pantalla de inicio de la App móvil*



**Nota.** Vista de pantalla de inicio de la aplicación móvil. Elaborado por: Flores, Riofrío

En la figura 42, se puede visualizar la pantalla de la App móvil para crear una nueva cuenta por medio del registro de información mediante un formulario.

**Figura 42**

*Pantalla de registro de usuarios de la App móvil*



**Nota.** Vista de pantalla de inicio de la aplicación móvil. Elaborado por: Flores, Riofrío

En la figura 43, se observa el método Registrar() que permite realizar el registro de información del usuario, utilizando la librería Volley que permite realizar las peticiones HTTP. En caso de que no se realice la petición o exista algún error la clase Response.ErrorListener informara sobre el inconveniente, por último, ingresamos el método getParams el cual contendrá los parámetros que el sitio solicita para poder tener una respuesta.

**Figura 43**

*Método registro de usuarios*

```
private void Registrar() {
    progreso = new ProgressDialog(this);
    progreso.setMessage("Iniciando...");
    progreso.show();
    String url = ConstantUtils.URL_REGISTRAR;

    stringRequest = new StringRequest(Request.Method.POST, url, new Response.Listener<String>() {
        @Override
        public void onResponse(String response) {
            Log.i("RESPUESTA JSON: ", "" + response);
            try {
                Gson gson = new Gson();
                JSONObject obj = new JSONObject(response);
                Type token = new TypeToken<GenericResponse<PersonaDTO>>() {
                }.getType();
                GenericResponse<PersonaDTO> loginResponsePersona = gson.fromJson(obj.toString(), token);
                if (loginResponsePersona.code.equals(ConstantUtils.CODE_200)) {
                    Functions.mostrarToast("Registro Correcto, porfavor intente logearse", getApplicationContext());
                    goToHome();
                } else {
                    Functions.mostrarToast("Registro Error:" + loginResponsePersona.message, getApplicationContext());
                }
            } catch (Exception e) {
                e.printStackTrace();
            }
            progreso.dismiss();
        }
    }, new Response.ErrorListener() {
        @Override
        public void onErrorResponse(VolleyError error) {
            Toast.makeText(getApplicationContext(), "No se ha podido conectar", Toast.LENGTH_SHORT).show();
            Log.i("ERROR: ", "" + error.toString());
            progreso.dismiss();
        }
    }) {
        @Override
        protected Map<String, String> getParams() throws AuthFailureError { //para enviar los datos mediante POST
            String sEmail = email.getText().toString();
            String sPassword = password.getText().toString();
            String sNombre = nombre.getText().toString();
            String sImagePhoto = Functions.convert(bitmap);
            String telefono = txt_telefonoregistro.getText().toString();
            String direccionId = listDireccionesDTO.get(spinnerDireccionesRegistro.getSelectedItemPosition()).getDIRECCION_ID();

            Map<String, String> parametros = new HashMap<>();
            parametros.put("PERSONA_EMAIL", sEmail);
            parametros.put("PERSONA_NOMBRE", sNombre);
            parametros.put("PERSONA_CONTRASENA", sPassword);
            parametros.put("PERSONA_IMAGEN", sImagePhoto);
            parametros.put("PERSONA_TELEFONO", telefono);
            parametros.put("DIRECCION_ID", direccionId);
            //estas parametros son enviados a nuestro web service
            Log.i(ConstantUtils.TAG_NAME_APP, "Usuario a crear: " + Functions.toJsonString(parametros));
            return parametros;
        }
    };
    //Agregar timeout
    stringRequest.setRetryPolicy(new DefaultRetryPolicy(
        ConstantUtils.MY_SOCKET_TIMEOUT_MS,
        0,
        DefaultRetryPolicy.DEFAULT_BACKOFF_MULT));
    requestsQueue.add(stringRequest);
}
```

**Nota.** En la figura se puede observar el método para registro de usuarios. Elaborado por: Flores,

Riofrío

En la figura 44, se observa el método que permite cargar las direcciones que se encuentran en la base de datos en un Spinner, el cual permitirá al usuario desplegar la información para luego seleccionarla.

**Figura 44**

*Método cargarSpinnerDireccion*

```
public void cargarSpinnerDireccion(List<DireccionDTO> listaCategorias) {  
    List<String> spinnerArray = new ArrayList<>();  
    for (DireccionDTO direccionDTO : listaCategorias) {  
        spinnerArray.add(direccionDTO.getPROVINCIA() + "-" + direccionDTO.getCANTON() + "-" + direccionDTO.getPARROQUIA())  
    }  
    ArrayAdapter<String> adapter = new ArrayAdapter<>(  
        getApplicationContext(), android.R.layout.simple_spinner_item, spinnerArray);  
  
    adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);  
    spinnerDireccionesRegistro.setAdapter(adapter);  
}
```

**Nota.** Método que permite cargar direcciones en un Spinner. Elaborado por: Flores, Riofrío.

En la figura 45, se visualiza la pantalla de inicio de sesión para acceder al sistema, mediante el ingreso del correo electrónico y contraseña.

**Figura 45**

*Pantalla iniciar sesión de la App móvil*



**Nota.** Vista de pantalla de iniciar sesión de la aplicación móvil. Elaborado por: Flores, Riofrío.

En la figura

En la figura 46, se observa el método iniciar() que permite realizar el ingreso al sistema, utilizando la librería Volley que permite realizar las peticiones HTTP. En caso de que no se realice la petición o exista algún error la clase Response.ErrorListener informara sobre el inconveniente, por último, ingresamos el método getParams el cual contendrá los parámetros que el sitio solicita para poder tener una respuesta.

**Figura 46**

*Método utilizado para el inicio de sesión*

```
private void iniciar() {
    progreso = new ProgressDialog(this);
    progreso.setMessage("Iniciando...");
    progreso.show();
    String url = ConstantUtils.URL_LOGIN;

    StringRequest = new StringRequest(Request.Method.POST, url, new Response.Listener<String>() {
        @Override
        public void onResponse(String response) {
            Usuario userParcelable = new Usuario();
            Log.d("RESPUESTA JSON: ", "" + response);
            try {
                Gson gson = new Gson();
                JSONObject obj = new JSONObject(response);
                Type token = new TypeToken<GenericResponse<PersonaDTO>>() {
                }.getType();
                GenericResponse<PersonaDTO> loginResponsePersona = gson.fromJson(obj.toString(), token);
                if (loginResponsePersona.code.equals(ConstantUtils.CODE_200)) {
                    Functions.mostrarToast("Login Correcto", getApplicationContext());
                    Usuario usuarioLogeado = new Usuario();
                    if (loginResponsePersona.data.getPERSONA_IMAGEN() != null) {
                        usuarioLogeado.setImage(loginResponsePersona.data.getPERSONA_IMAGEN());
                    }
                    usuarioLogeado.setEmail(loginResponsePersona.data.getPERSONA_EMAIL());
                    usuarioLogeado.setNombre(loginResponsePersona.data.getPERSONA_NOMBRE());
                    usuarioLogeado.setId(Integer.parseInt(loginResponsePersona.data.getPERSONA_ID()));
                    usuarioLogeado.setTelefono(loginResponsePersona.data.getPERSONA_TELEFONO());
                    usuarioLogeado.setDIRECCION_ID(loginResponsePersona.data.getPERSONA_DIRECCION_ID());
                    usuarioLogeado.setImage(loginResponsePersona.data.getPERSONA_IMAGEN());
                    SharedPrefsUtils.guardarUsuarioLogin(usuarioLogeado, getApplicationContext());
                    goToMenuProductos();
                } else {
                    Functions.mostrarToast("Login Error:" + loginResponsePersona.message, getApplicationContext());
                }
            } catch (Exception e) {
                e.printStackTrace();
            }

            progreso.dismiss();
        }
    }, new Response.ErrorListener() {
        @Override
        public void onErrorResponse(VolleyError error) {
            Toast.makeText(getApplicationContext(), "No se ha podido conectar", Toast.LENGTH_SHORT).show();
            progreso.dismiss();
            error.printStackTrace();
        }
    });
    @Override
    protected Map<String, String> getParams() throws AuthFailureError {
        String sEmail = correo.getText().toString();
        String sPassword = contrasefai.getText().toString();

        Map<String, String> parametros = new HashMap<>();
        parametros.put("PERSONA_EMAIL", sEmail);
        parametros.put("PERSONA_CONTRASENA", sPassword);
        //estos parametros son enviados a nuestro web service

        return parametros;
    }
};

//Agregar timeout
stringRequest.setRetryPolicy(new DefaultRetryPolicy(
    MY_SOCKET_TIMEOUT_MS,
    0,
    DefaultRetryPolicy.DEFAULT_BACKOFF_MULT));

requestQueue.add(stringRequest);
}
```

**Nota.** En la figura se puede observar el método para ingresar al sistema. Elaborado por: Flores,

Riofrío

En la figura 47, se visualiza la pantalla para recuperar la contraseña, mediante el ingreso del correo electrónico del usuario que debe estar registrado en el sistema.

**Figura 47**

*Pantalla recuperar contraseña de la App móvil*



**Nota.** Vista de pantalla recuperar contraseña de la aplicación móvil. Elaborado por: Flores, Riofrío.

En la figura 48, se muestra la validación que se realizó para el ingreso del correo electrónico el cual esta validado para que el usuario no coloque un correo invalido.

**Figura 48**

*Validación del correo electrónico*

```
public static boolean validarCorreo(String email) {  
    String emailREGEX = "[a-zA-Z0-9_+&*~]+(?:\\.([a-zA-Z0-9_+&*~]+)*@(?:[a-zA-Z0-9-]+\\.|)+[a-zA-Z]{2,7})$";  
    Pattern pattern = Pattern.compile(emailREGEX );  
    if (email == null){  
        return false;  
    }  
    return pattern .matcher(email).matches();  
}
```

**Nota.** Código de validación del correo electrónico. Elaborado por: Flores, Riofrío.

En la figura 49, se muestra la validación que se realizó para el ingreso de la contraseña el cual esta validado por medio de una expresión regular para que el usuario coloque una contraseña segura al ingresar una letra mayúscula, un número y un carácter especial.

## Figura 49

### Validación de contraseña

```
private static final Pattern PASSWORD_PATTERN =
    Pattern.compile("^ +
        "(?=.*[0-9])" + //números
        "(?=.*[a-z])" + //letra minúscula
        "(?=.*[A-Z])" + //letra mayúscula
        "(?=.*[a-zA-Z])" + //letra
        "(?=.*[@#$%^&+=])" + //Un caracterer especial
        "(?=\\S+$)" + //sin espacios
        ".{4,}" + //mayor a 4
        "$");
```

**Nota.** Expresión regular para realizar la validación de la contraseña. Elaborado por: Flores, Riofrío.

En la figura 50, se puede observar la pantalla de ingreso del producto en la cual el usuario tendrá que ingresar la información del producto, también se encuentra la opción de oferta en la que si desea ofertar su producto deberá llenar el cuadro de texto que pertenece a la oferta y así podría promocionar sus productos.

## Figura 50

### Pantalla ingresar producto



**Nota.** Vista de pantalla de ingreso de los productos. Elaborado por: Flores, Riofrío.

En la figura 51, se observa el método que permite autocompletar el nombre del producto, para que el usuario no ingrese un dato inválido que contenga faltas ortográficas, para ello solo tendrá que escribir la primera letra del nombre y aparecerá una lista de nombres que podrá seleccionar, para ello se codifico una instancia del array que contendrá la lista de los nombres.

### Figura 51

*Método para el autocompletado del nombre del producto*

```
public void cargarAutocompleteComponent() {
    List<String> nombresPorductosArray = new ArrayList<>();
    for (ProductoDTO item : listaNombresProductos) {
        nombresPorductosArray.add(item.PRODUCTO_NOMBRE);
    }
    //Crear la instancia del arrayAdapter que contiene el nombre de las listas
    ArrayAdapter<String> adapter = new ArrayAdapter<>
        ( context: this, android.R.layout.select_dialog_item, nombresPorductosArray);
    //obtener la instancia del autocompleteTextView del producto
    txtProductoNombre.setThreshold(1); //empezara a el proceso de autocompletado desde el primer caracter
    txtProductoNombre.setAdapter(adapter); //establecer los datos del adaptador autocomplete
}
```

**Nota.** Método para autocompletar el nombre del producto. Elaborado por: Flores, Riofrío.

En la figura 52, se muestra el método que permite seleccionar una imagen en la galería y para tomar fotografías de los productos utilizando el dispositivo móvil y así poder subir la información a la App móvil.

### Figura 52

*Método seleccionar imagen de galería y usar la cámara*

```
//Escoger fotos de la galería
private void mostrarfotogaleria() {
    Intent galeriaIntent = new Intent(Intent.ACTION_PICK, android.provider.MediaStore.Images.Media.EXTERNAL_CONTENT_URI);
    startActivityForResult(galeriaIntent, GALLERY);
}

//Tomar fotos desde la cámara
private void tomarfotoscamara() {
    Intent intent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
    startActivityForResult(intent, CAMERA);
}
```

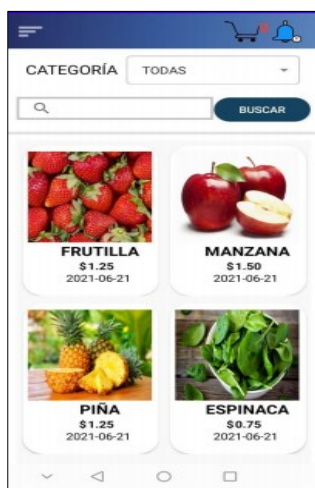
**Nota.** Método para seleccionar imágenes y tomar fotografías desde el dispositivo móvil. Elaborado por: Flores, Riofrío.



En la figura 53, se puede observar la pantalla de la vista principal de los productos, donde se encuentran el producto que ingresan los agricultores en el sistema para promocionarlos y poder venderlos.

### Figura 53

*Pantalla vista principal productos de la App móvil*



**Nota.** Vista principal de productos de la aplicación móvil. Elaborado por: Flores, Riofrío.

En la figura 54, se puede observar la vista de detalle de los productos que son seleccionados por los usuarios, en la que el comprador puede ver la información del producto y también seleccionar la cantidad del producto a comprar.

### Figura 54

*Pantalla vista detalle del producto de la App móvil*



**Nota.** Vista detalle de productos de la aplicación móvil. Elaborado por: Flores, Riofrío.

En la figura 55, se puede observar el método agregarProductoCarroCompras, el que se valida por medio de un if, si el PRODUCTO\_PERSONA\_ID es el del usuario logeado, se le notificara que no puede comprar sus productos. De igual manera se verificará si el producto ya se encuentra seleccionado solo se actualizará la cantidad.

### Figura 55

*Método agregar producto al carrito de compras*

```
//Agregar producto al carrito de compras
public void agregarProductoCarroCompras(View view){
    //Validar que no sea su mismo producto
    if(PRODUCTO_PERSONA_ID.equals(ConstantUtils.getUserLogeado(getApplicationContext()))) {
        Toast.makeText(getApplicationContext(),"No puede comprar sus productos.",Toast.LENGTH_LONG).show();
        return;
    }
    try{
        ProductoDTO productoDTO = new ProductoDTO();
        productoDTO.PRODUCTO_ID = PRODUCTO_ID;
        cantidadProducto = Integer.parseInt(spinnerCantidadProductoDetalle.getSelectedItem().toString());
        productoDTO.PRODUCTO_CANTIDAD = ""+cantidadProducto;
        productoDTO.PRODUCTO_CANTIDAD_MAXIMA = ""+cantidadProductoMaxima;
        productoDTO.PRODUCTO_PRECIO = ""+precioProducto;
        productoDTO.PRODUCTO_TOTAL = cantidadProducto*precioProducto;
        productoDTO.PRODUCTO_NOMBRE = textViewNombreProducto.getText().toString();
        productoDTO.PRODUCTO_PRECIO = textViewPrecio.getText().toString();
        productoDTO.CATEGORIA_NOMBRE = textViewCategoria.getText().toString();
        productoDTO.PRODUCTO_IMAGEN = IMAGEN_PRODUCTO_BASE64;
        productoDTO.PERSONA_NOMBRE = PERSONA_NOMBRE_EMPRENDEDOR;
        productoDTO.PERSONA_TELEFONO = PERSONA_TELEFONO_EMPRENDEDOR;
        productoDTO.PERSONA_ID = PRODUCTO_PERSONA_ID;

        CarritoComprasDTO carritoComprasDTO = SharedPrefencesUtils.obtenerCarrito(getApplicationContext());
        if(carritoComprasDTO == null){
            carritoComprasDTO = new CarritoComprasDTO();
        }
        List<ProductoDTO> listaProductos = carritoComprasDTO.getListaProductos();
        //Verificar si ya existe en la lista y modificar
        boolean existeProductoEnCarrito= false;
        for(int i =0; i<listaProductos.size(); i++){
            if(listaProductos.get(i).PRODUCTO_ID.equals(productoDTO.PRODUCTO_ID)){
                existeProductoEnCarrito = true;
                //Actualizar en la lista la cantidaddel producto
                listaProductos.get(i).PRODUCTO_CANTIDAD = productoDTO.PRODUCTO_CANTIDAD;
                listaProductos.get(i).PRODUCTO_CANTIDAD_MAXIMA = productoDTO.PRODUCTO_CANTIDAD_MAXIMA;
                break;
            }
        }
        if (!existeProductoEnCarrito){
            listaProductos.add(productoDTO);
        }

        carritoComprasDTO.setListaProductos(listaProductos);
        //Actualizar compras
        SharedPrefencesUtils.guardarCarrito(carritoComprasDTO, getApplicationContext());
        Toast.makeText(getApplicationContext(),"Se agrego producto al carrito Num
items(""+listaProductos.size()+"")",Toast.LENGTH_LONG).show();
    }catch (Exception e){
        Toast.makeText(getApplicationContext(),"No se pudo agregar el producto al carrito",Toast.LENGTH_LONG).show();
        e.printStackTrace();
    }
}
```

**Nota.** Método para cargar producto en el carrito de compras. Elaborado por: Flores, Riofrío.

En la figura 56 visualiza el detalle de la compra que se muestra al finalizar la compra el cual esta detallado por fecha, forma de pago, nombre del producto, precio, cantidad, total y los datos (nombre y teléfono) del vendedor.

### Figura 56

*Pantalla detalle de compra del producto de la App móvil*



**Nota.** Vista detalle de compra del producto de la aplicación móvil. Elaborado por: Flores, Riofrío.

### 3.4.PRUEBAS

El proceso final que permite medir el desempeño del programa se realiza a través de pruebas que determinan el mejor funcionamiento de la aplicación, verificando así que la App móvil cumple con los requisitos establecidos para obtener un software de calidad.

Es la parte principal del ciclo de vida del desarrollo de software, el testing es una actividad que se puede realizar de forma manual y automática para identificar posibles fallas en el sistema.

Para el funcionamiento de la App móvil hay que tomar en cuenta los requisitos mínimos para ejecutarlo en un dispositivo móvil.

Requerimientos del software:

**Sistema operativo:** el dispositivo en el que se realizó las pruebas tiene el sistema operativo Android versión 10.

**Servidor:** las pruebas se realizaron en un servidor local.

### 3.4.1. Prueba de caja negra

Son pruebas que excluyen la codificación y estructura del software y se enfocan solo en escenarios, entradas y salidas del sistema, con el fin de evaluar los diferentes ingresos y resultados de cada módulo de la aplicación.

Para el proceso de prueba de caja negra, se implementa la partición de equivalencias, que permite identificar los valores válidos e inválidos del sistema, especificando así cada acción de la aplicación móvil.

**Tabla 34**

*Particiones de equivalencia para pruebas de caja negra*

<b>PARTICIÓN DE EQUIVALENCIAS</b>	
<b>ATRIBUTO</b>	<b>SIGNIFICADO</b>
Condición de entrada	Ingresar datos al sistema
Tipo	Tipos de datos a ingresar en el sistema que pueden ser: conjunto, rango, valor y lógicas
Equivalencia valida	Datos correctos ingresados en el sistema
Equivalencia invalida	Datos incorrectos ingresados en el sistema

**Nota.** Identifica los atributos de entrada y salida de la aplicación. Elaborado por: Flores, Riofrío.

**Tabla 35**

*Particiones de equivalencia para el registro de usuarios*

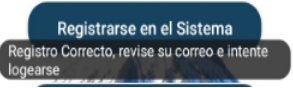
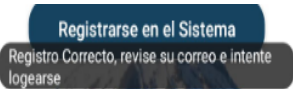

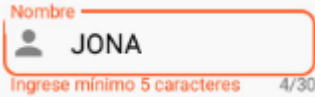

<b>Partición de equivalencia registro de usuario</b>
--

<b>Condición de Entrada</b>	<b>Tipo</b>	<b>Equivalencia valida (EV)</b>	<b>Equivalencia no valida (ENV)</b>
<b>Nombre</b>	Valor	EV1: 20 caracteres alfabéticos cada uno	ENV1.1: campo en blanco ENV1.2: dato alfabético menor a 5
<b>Teléfono</b>	Valor	EV2: 10 caracteres numéricos	ENV2.1: campo en blanco ENV2.2: valor numérico menor a 10 ENV2.2: valor numérico que no empiece con 09 y 08
<b>Correo</b>	Valor	EV3: 20 Caracteres alfanuméricos y caracteres especiales cada uno	ENV3.1: campo en blanco ENV3.2: campo sin valor alfanumérico y sin caracteres especiales ENV3.3: valor al principio o final con punto ENV3.4: solo un carácter @ es permitido
<b>Contraseña</b>	Valor	EV4: 10 caracteres alfanuméricos cada uno	ENV4.1: campo en blanco ENV4.2: campo sin valor alfanumérico y campo sin caracteres especiales ENV4.3: campo sin carácter en mayúscula ENV4.4: mínimo 5 caracteres

**Nota.** Registra las entradas de datos válidos e inválidos de la App móvil. Elaborado por: Flores, Riofrío.

**Tabla 36**

*Prueba de caja negra registrar usuario*

PERFIL REGISTRAR USUARIO							
DATOS VALIDOS							
#	Equivalencia	Nombre	Teléfono	Dirección	Correo Electrónico	Contraseña	Resultado
1	EV1 EV2 EV3 EV4	JONATHAN	0989987571	COTOPAXI- LATACUNGA- LATACUNGA	<a href="mailto:jona_flores13@gmail.com">jona_flores13@gmail.com</a>	Jona13#	
	EV1 EV2 EV3 EV4	JONATHAN	0889987571	COTOPAXI- LATACUNGA- LATACUNGA	<a href="mailto:jona_flores13@gmail.com">jona_flores13@gmail.com</a>	Jona13#	
DATOS NO VALIDOS							
#	Equivalencia	Nombre	Teléfono	Dirección	Correo Electrónico	Contraseña	Resultado
2	EV1		0989987571	COTOPAXI- LATACUNGA- LATACUNGA	<a href="mailto:jona_flores13@gmail.com">jona_flores13@gmail.com</a>	Jona13#	
3	EV1	JONA	0989987571	COTOPAXI- LATACUNGA- LATACUNGA	<a href="mailto:jona_flores13@gmail.com">jona_flores13@gmail.com</a>	Jona13#	
4	EV2	JONATHAN		COTOPAXI- LATACUNGA- LATACUNGA	<a href="mailto:jona_flores13@gmail.com">jona_flores13@gmail.com</a>	Jona13#	

5	EV2	JONATHAN	098998757	COTOPAXI- LATACUNGA- LATACUNGA	<a href="mailto:jona_flores13@gmail.com">jona_flores13@gmail.com</a>	Jona13#	
6	EV2	JONATHAN	0189987577 0289987577 0389987577 0489987577 0589987577 0689987577 0789987577	COTOPAXI- LATACUNGA- LATACUNGA	<a href="mailto:jona_flores13@gmail.com">jona_flores13@gmail.com</a>	Jona13#	
7	EV3	JONATHAN	0989987571	COTOPAXI- LATACUNGA- LATACUNGA		Jona13#	
8	EV3	JONATHAN	0989987571	COTOPAXI- LATACUNGA- LATACUNGA	<a href="mailto:jona@_flores13@gmail.com">jona@_flores13@gmail.com</a> <a href="mailto:jona_flores13@gmail.com">jona_flores13@gmail.com</a> <a href="mailto:jona_flores13@gmail.com">jona_flores13@gmail.com</a> <a href="mailto:jona_flores13@gmail.com">jona_flores13@gmail.com</a>	Jona13#	

9	EV4	JONATHAN	0989987571	COTOPAXI- LATACUNGA- LATACUNGA	<a href="mailto:jona_flores13@gmail.com">jona_flores13@gmail.com</a>		
10	EV4	JONATHAN	0989987571	COTOPAXI- LATACUNGA- LATACUNGA	<a href="mailto:jona_flores13@gmail.com">jona_flores13@gmail.com</a>	jona jona13 jona#	

**Nota.** Ingreso de datos validos e inválidos en la App móvil. Elaborado por: Flores, Riofrío.



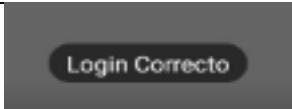
**Tabla 37***Particiones de equivalencia para iniciar sesión*



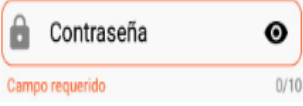
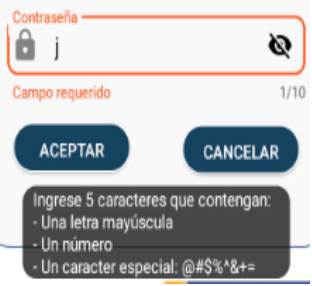
<b>Partición de equivalencia iniciar sesión</b>			
<b>Condición de Entrada</b>	<b>Tipo</b>	<b>Equivalencia valida (EV)</b>	<b>Equivalencia no valida (ENV)</b>
Correo	Valor	EV1: 20 caracteres alfanuméricos y caracteres especiales cada uno	ENV1.1: campo en blanco ENV1.2: campo sin valor alfanumérico y sin caracteres especiales ENV1.3: valor al principio o final con punto ENV1.4: solo permite un @ ENV1.4: letras con tildes
Contraseña	Valor	EV2: 10 caracteres alfanuméricos cada uno	ENV2.1: campo en blanco ENV2.2: campo sin valor alfanumérico y campo sin caracteres especiales ENV2.3: campo sin carácter en mayúscula ENV2.3: mínimo 5 caracteres

**Nota.** Registra las entradas de datos válidos e inválidos al iniciar sesión en la App móvil.

Elaborado por: Flores, Riofrío.

**Tabla 38***Prueba de caja negra iniciar sesión*

<b>PERFIL INICIAR SESIÓN</b>				
<b>DATOS VALIDOS</b>				
<b>#</b>	<b>Equivalencia</b>	<b>Correo Electrónico</b>	<b>Contraseña</b>	<b>Resultado</b>
1	EV1 EV2	<a href="mailto:jona_flores13@gmail.com">jona_flores13@gmail.com</a>	Jona13#	
<b>DATOS NO VALIDOS</b>				

#	Equivalencia	Correo Electrónico	Contraseña	Resultado
2	EV1		Jona13#	
3	EV1	<a href="mailto:jona@flores13@gmail.com">jona@_flores13@gmail.com</a> <a href="mailto:jona_flores13@gmail.com">.jona_flores13@gmail.com.</a> <a href="mailto:jona_flores13@gmail.com">.jona_flores13@gmail.com.</a> <a href="mailto:jona_flores13@gmail.com">joná_flores13@gmail.com</a>	Jona13#	
4	EV2	<a href="mailto:jona_flores13@gmail.com">jona_flores13@gmail.com</a>		
5	EV2	<a href="mailto:jona_flores13@gmail.com">jona_flores13@gmail.com</a>	jona jona13 jona#	

**Nota.** Identificar el ingreso de datos validos e inválidos al iniciar sesión. Elaborado por: Flores, Riofrío.

**Tabla 39**

*Particiones de equivalencia recuperar contraseña*

Partición de equivalencia recuperar contraseña			
Condición de Entrada	Tipo	Equivalencia valida (EV)	Equivalencia no valida (ENV)
Correo	Valor	EV1: 20 caracteres alfanuméricos y	ENV1.1: campo en blanco

		caracteres especiales cada uno	ENV1.2: campo sin valor alfanumérico y sin caracteres especiales ENV1.3: valor al principio o final con punto ENV1.4: solo permite un @ ENV1.4: letras con tildes
--	--	-----------------------------------	--

**Nota.** Registra las entradas de datos válidos e inválidos al recuperar contraseña. Elaborado por: Flores, Riofrío.

**Tabla 40**

*Prueba de caja negra recuperar contraseña*

Perfil recuperar contraseña			
DATOS VALIDOS			
#	Equivalencia	Correo Electrónico	Resultado
1	EV1	<a href="mailto:jona_flores13@gmail.com">jona_flores13@gmail.com</a>	
DATOS NO VALIDOS			
#	Equivalencia	Correo Electrónico	Resultado
2	EV1		
3	EV1	<a href="mailto:jona@_flores13@gmail.com">jona@_flores13@gmail.com</a> <a href="mailto:.jona_flores13@gmail.com">.jona_flores13@gmail.com</a> <a href="mailto:.jona_flores13@gmail.com">.jona_flores13@gmail.com</a> <a href="mailto:joná_flores13@gmail.com">joná_flores13@gmail.com</a>	   

*Nota.* Identificar el ingreso de datos validos e inválidos al recuperar contraseña. Elaborado por:  
Flores, Riofrío.

**Tabla 41**

*Particiones de equivalencia ingresar productos*

<b>Partición de equivalencia ingresar producto</b>			
<b>Condición de Entrada</b>	<b>Tipo</b>	<b>Equivalencia valida (EV)</b>	<b>Equivalencia no valida (ENV)</b>
Imagen	Valor	EV1: formato JPG, PNG	ENV1: ningún valor
Categoría	Valor	EV2: 20 caracteres alfabéticos cada uno	ENV2: ningún valor
Presentación	Valor	EV3: 20 caracteres alfabéticos cada uno	ENV3: ningún valor
Medida	Valor	EV4: valor numérico	ENV4: campo en blanco
Nombre del producto	Valor	EV5: 20 caracteres alfabéticos cada uno	ENV5: campo en blanco
Valor medida	Valor	EV6: valor numérico	ENV6.1: campo en blanco ENV6.2: valor numérico menor a 1
Cantidad disponible	Number	EV7: valor numérico	ENV7.1: campo en blanco ENV7.2: valor numérico menor a 1
Precio unitario	Number	EV8: valor numérico	ENV8.1: campo en blanco ENV8.2: valor numérico menor a 1
Fecha cosecha	Valor	EV9: valor tipo fecha formato de fecha: yyyy-mm-dd	ENV9.1: campo en blanco
Fecha expiración	Valor	EV10: valor tipo fecha formato de fecha: yyyy-mm-dd	ENV10.1: campo en blanco
Descripción producto	Valor	EV11: 30 caracteres alfanuméricos cada uno	ENV11.1: campo en blanco

			EVN11.2: valores menores a 10 caracteres alfanuméricos
Descripción oferta	Valor	EV12: 30 caracteres alfanuméricos cada uno	EVN12.1: campo en blanco EVN12.2: valores menores a 10 caracteres alfanuméricos

**Nota.** Registra las entradas de datos válidos e inválidos para el ingreso de productos. Elaborado por: Flores, Riofrío.

**Tabla 42**

*Prueba de caja negra ingresar producto*

PERFIL INGRESAR PRODUCTO																
DATOS VALIDOS																
#	Equivalencia	Imagen	Categoría	Presentación	Medida	Nombre del Producto	Valor Medida	Cantidad Disponible	Precio Unitario	Fecha Cosecha	Fecha Expiración	Descripción Producto	Descripción Oferta	Resultado		
1	EV1		FRUTAS	FUNDA	LIBRA	FRESA	5	10	0.25	2021-06-04	2021-06-20	FRESCA		Se creo correctamente el producto		
2	EV2	JPG, PNG	FRUTAS	FUNDA	LIBRA	FRESA	5	10	0.25	2021-06-04	2021-06-20	FRESCA				
3	EV3	JPG, PNG	FRUTAS	FUNDA	LIBRA	FRESA	5	10	0.25	2021-06-04	2021-06-20	FRESCA	Fresas 2x1			
	EV4															
	EV5															
	EV6															
	EV7															
	EV8															
	EV9															
	EV1															
	0															
	EV1															
	1															
	EV1															
	2															
DATOS NO VALIDOS																

#	Equivalencia	Imagen	Categoría	Presentación	Medida	Nombre del Producto	Valor Medida	Cantidad Disponible	Precio Unitario	Fecha Cosecha	Fecha Expiración	Descripción Producto	Descripción Oferta	Resultado
3	EV1	JPG, PNG	FRUTAS	FUNDA	LIBRA		5	10	0.25	2021-06-04	2021-06-20	FRESCA		Nombre producto <hr/> Campo requerido
4	EV1	JPG, PNG	FRUTAS	FUNDA	LIBRA	FR	5	10	0.25	2021-06-04	2021-06-20	FRESCA		Nombre producto <b>FR</b> <hr/> Ingrese por lo menos 3 caracteres
5	EV2	JPG, PNG	FRUTAS	FUNDA	LIBRA	FRESA		10	0.25	2021-06-04	2021-06-20	FRESCA		Medida <input type="text"/> Campo requerido
6	EV2	JPG, PNG	FRUTAS	FUNDA	LIBRA	FRESA	0	10	0.25	2021-06-04	2021-06-20	FRESCA		Medida <input type="text" value="0"/> Ingrese valores mayores a 0
7	EV3	JPG, PNG	FRUTAS	FUNDA	LIBRA	FRESA	5		0.25	2021-06-04	2021-06-20	FRESCA		Cantidad disponible <input type="text"/> Campo requerido
8	EV3	JPG, PNG	FRUTAS	FUNDA	LIBRA	FRESA	5	0	0.25	2021-06-04	2021-06-20	FRESCA		Cantidad disponible <input type="text" value="0"/> Ingrese valores mayores a 0
9	EV4	JPG, PNG	FRUTAS	FUNDA	LIBRA	FRESA	0	10		2021-06-04	2021-06-20	FRESCA		Precio <input type="text"/> Campo requerido
10	EV4	JPG, PNG	FRUTAS	FUNDA	LIBRA	FRESA	0	10	0	2021-06-04	2021-06-20	FRESCA		Precio <input type="text" value="0"/> Ingrese valores mayores a 0
11	EV5	JPG, PNG	FRUTAS	FUNDA	LIBRA	FRESA	5	10	0.25		2021-06-20	FRESCA		Fecha cosecha <input type="text"/> Campo requerido

12	EV6	JPG, PNG	FRUTAS	FUNDA	LIBRA	FRESA	5	10	0.25	2021-06-04		FRESCA		Fecha expiración Campo requerido
13	EV7	JPG, PNG	FRUTAS	FUNDA	LIBRA	FRESA	5	10	0.25	2021-06-04	2021-06-20			Descripción producto Campo requerido
14	EV7	JPG, PNG	FRUTAS	FUNDA	LIBRA	FRESA	5	10	0.25	2021-06-04	2021-06-20	FR		Descripción producto FR Ingrese mínimo 10 caracteres

**Nota.** Identificar el ingreso de datos validos e inválidos al momento de ingresar producto. Elaborado por: Flores, Riofrío



### 3.4.2. Prueba de carga

La prueba de carga se utiliza para evaluar la cantidad de solicitudes admitidas por el sistema sin reducir el rendimiento del software. Por ejemplo, la cantidad de transacciones ejecutadas y la cantidad de usuarios que ingresan al sistema, se evaluarán de esta manera para determinar si el sistema puede soportar el máximo capacidad de carga.

Las características del hardware utilizado para realizar las pruebas son las siguientes:

**Tabla 43**

*Características del hardware del dispositivo móvil*

Características del Hardware	
Dispositivos	Huawei P20
Sistema Operativo	Android
Versiones	10
Memoria RAM	4

**Nota.** Principales características del smartphone. Elaborado por: Flores, Riofrío.

Petición que serán evaluadas para la prueba de carga:

**Tabla 44**

*Peticiones y métodos HTTP para pruebas de carga*

Peticiones	Métodos de petición HTTP	Puerto
Iniciar sesión	POST	8080
Listar productos	GET	8080
Registro usuario	POST	8080
Crear Producto	POST	8080

**Nota.** Peticiones evaluadas al iniciar sesión, listar los productos, registro de usuarios y crear producto. Elaborado por: Flores, Riofrío.

Las pruebas de carga y estrés se realizaron en la plataforma JMeter, es una herramienta que permite generar pruebas a las aplicaciones al analizar el funcionamiento de los servicios del software, ver figura 39.

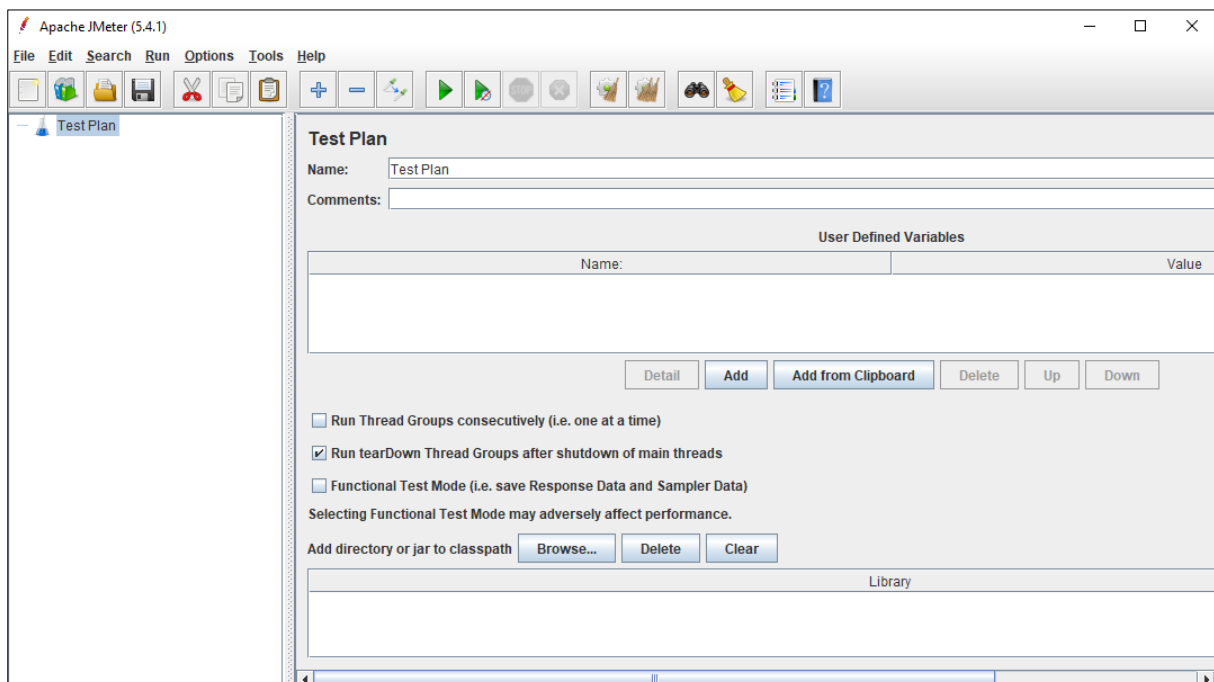
En el caso de la App móvil se llevó a cabo la evaluación de las métricas en el entorno del inicio de sesión, listar productos, registro de usuarios, ingresar producto y realizar una compra por medio del ingreso incremental de 100 peticiones.

A continuación, se presentará la configuración que se realizó en JMeter para realizar las pruebas de funcionamiento de la aplicación móvil.

Como primer paso se puede observar la herramienta JMeter que nos permitirá realizar las pruebas de carga de la aplicación móvil, ver figura 57.

## Figura 57

### Herramienta JMeter

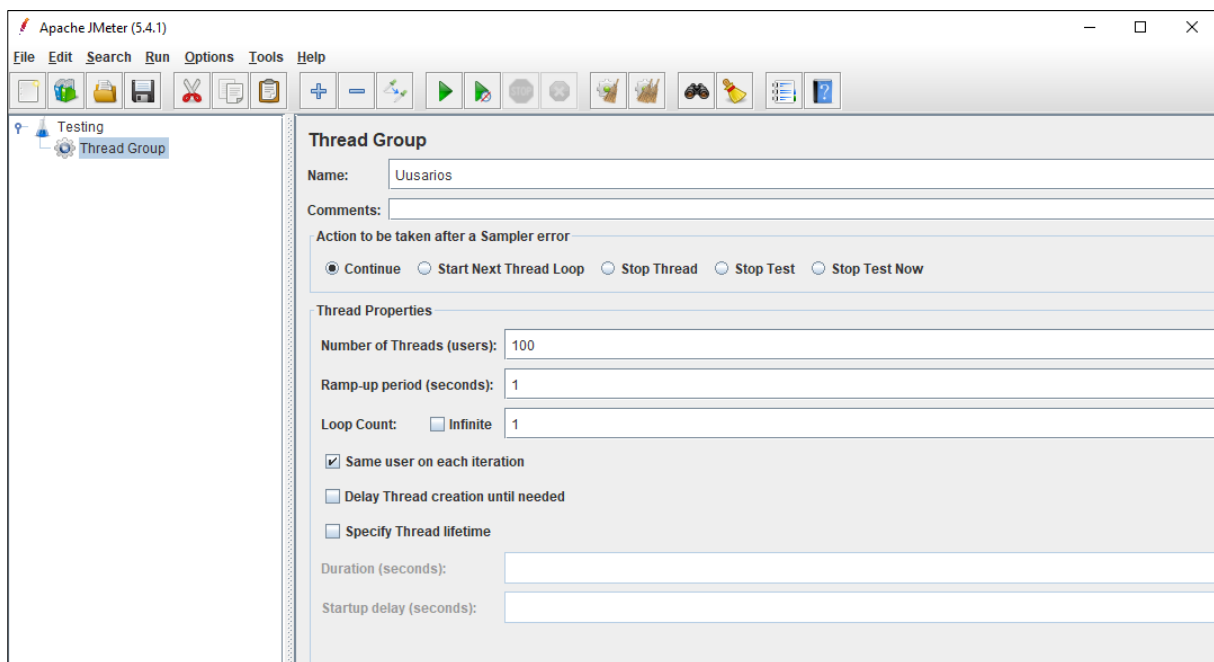


**Nota.** JMeter que permite realizar diferentes tipos de pruebas como por ejemplo pruebas de carga a las aplicaciones móviles. Elaborado por: Flores, Riofrío.

En la figura 58, se observa la configuración realizada para la prueba de carga de 100 usuarios la que se estableció en la opción de grupo de usuarios, en el ítem propiedades del hilo que está establecido por el número de hilos, periodo de subida en segundos y las iteraciones que realiza, es decir que ingresan 100 usuarios a la aplicación móvil en un periodo de tiempo y cuando llegue al usuario final o la herramienta detecte un error se detendrá dando a conocer los resultados de carga de la App.

**Figura 58**

*Configuración de usuarios en JMeter*

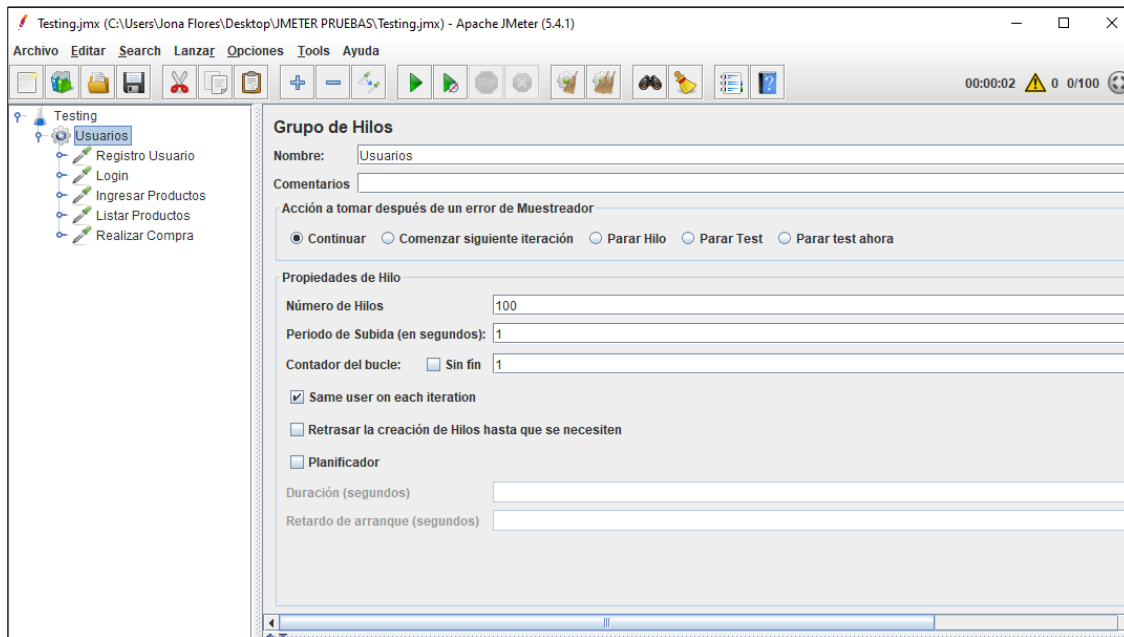


**Nota.** Configuración de usuarios para la prueba de carga. Elaborado por: Flores, Riofrío.

En la figura 59, se muestra la configuración de la peticiones de registro de usuario, login, ingresar producto, listar producto y realizar una compra con las que se va a realizar la prueba de carga con 100 usuarios.

**Figura 59**

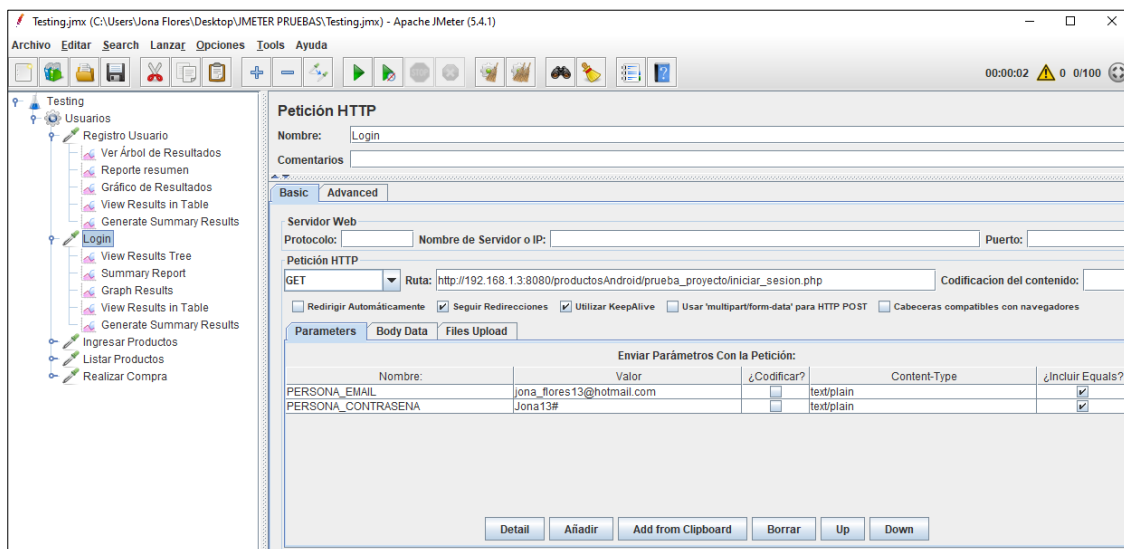
*Configuración de la petición HTTP*



**Nota.** Configuración de peticiones HTTP para registrar usuario, login, ingresar producto, listar productos y realizar una compra. Elaborado por: Flores, Riofrío.

**Figura 60**

*Configuración de la ruta de la petición*



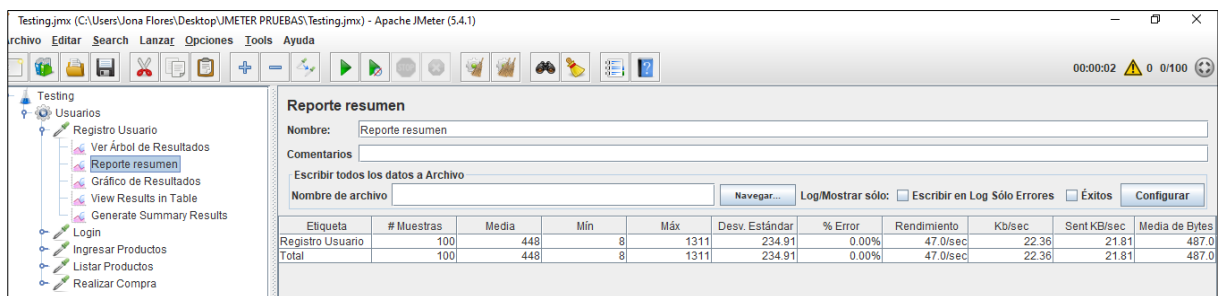
**Nota.** Configuración de la ruta para realizar la petición. Elaborado por: Flores, Riofrío.

En la figura 60, muestra la configuración de la ruta para realizar las peticiones donde se va a ejecutar la prueba, para lo cual se debe ingresar en cada petición HTTP la dirección donde se encuentra las peticiones de registro de usuario, login, crear producto, listar productos y realizar una compra.

En la figura 61, se observa el resultado que se obtuvo al realizar la prueba de carga al momento de registrar 100 usuarios en el sistema.

**Figura 61**

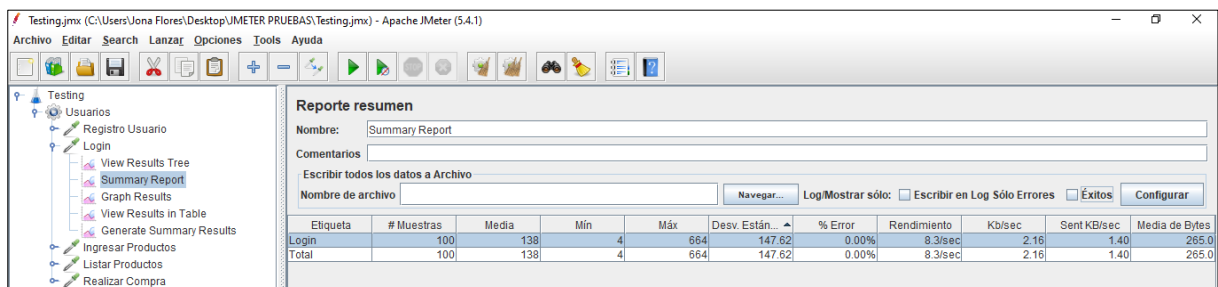
*Resultado de la prueba de carga registro de usuarios*



**Nota.** Resultado al registrar 100 usuarios en el sistema. Elaborado por: Flores, Riofrío.

**Figura 62**

*Resultado de la prueba de carga de login*



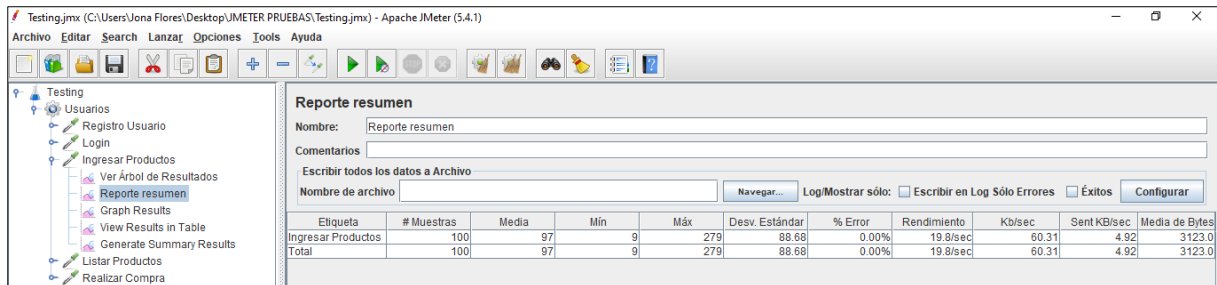
**Nota.** Resultado al autenticar 100 usuarios en el sistema. Elaborado por: Flores, Riofrío.

En la figura 62, se visualiza los resultados obtenidos al momento de realizar la prueba de carga con 100 usuarios para ingresar al sistema.

La prueba que se visualiza en la figura 63, muestra los resultados obtenidos al momento de ingresar productos en el sistema configurada con una carga de 100 usuarios que realizan este proceso.

**Figura 63**

*Resultado de la prueba de carga crear producto*

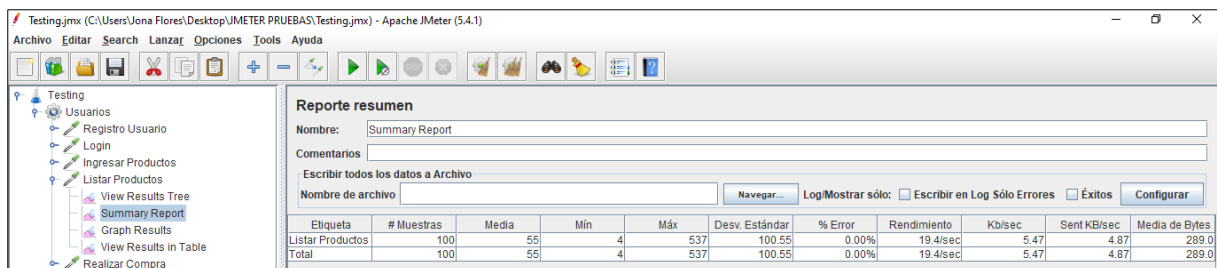


**Nota.** Resultado de la prueba de carga al ingresar productos en el sistema. Elaborado por: Flores, Riofrío.

La prueba que se visualiza en la figura 64, muestra los resultados obtenidos al momento de listar los productos en el sistema para que el usuario pueda visualizar, seleccionar y realizar una compra.

**Figura 64**

*Resultado de la prueba de carga listar producto*

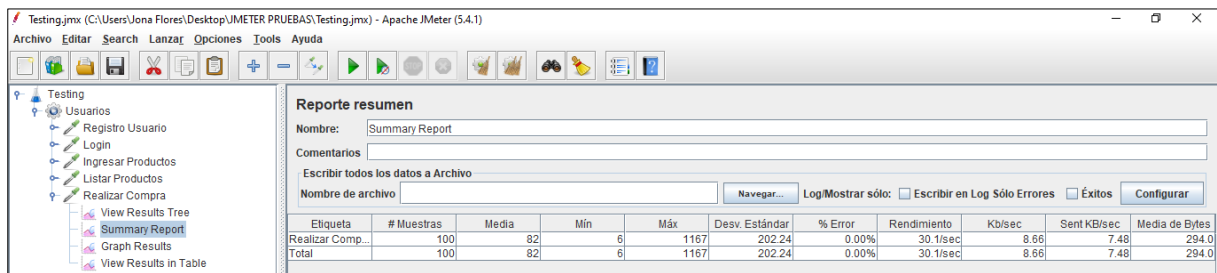


**Nota.** Resultado de la prueba de carga al momento de listar productos en el sistema. Elaborado por: Flores, Riofrío.

La prueba que se visualiza en la figura 65, muestra los resultados obtenidos en el proceso que se realiza para realizar un a compra en el sistema.

**Figura 65**

*Resultado de la prueba de carga realizar compra*



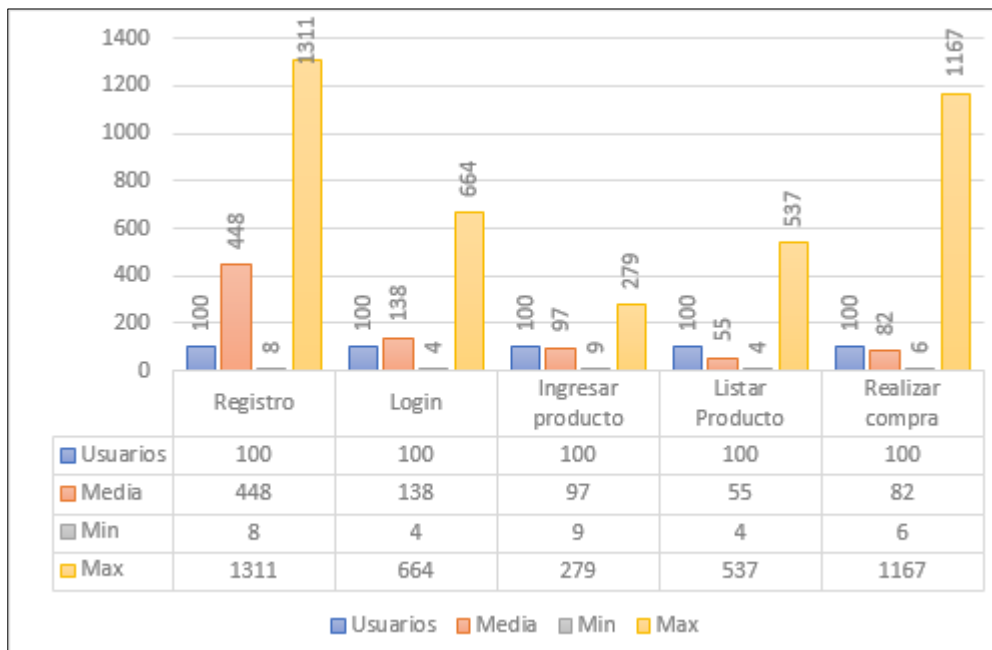
**Nota.** Resultado de la prueba de carga al momento de realizar una compra en el sistema.

Elaborado por: Flores, Riofrío.

Los resultados obtenidos en la figura 66, representan la carga media, mínima, y máxima al ingresar de 100 usuarios al momento de registrar usuarios, iniciar sesión, listar productos y realizar una compra.

**Figura 66**

*Gráfica de medio, mínimo y máximo de carga de usuarios*

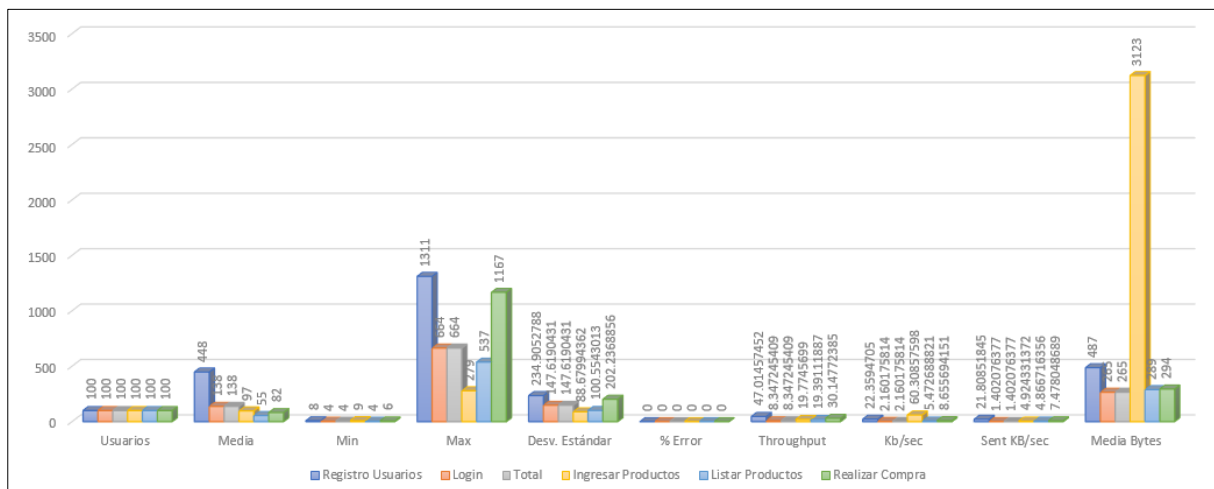


**Nota.** Vista grafica de valores de media, mínimo y máximo, obtenida mediante la carga de usuarios. Elaborado por: Flores, Riofrío.

Como se puede observar en la figura 67, se observa los resultados generales de la prueba de carga realizado con 100 usuarios al momento de registrar usuarios, iniciar sesión, ingresar productos, listar productos y realizar una compra, los cuales se evidencia que todas las peticiones fueron atendidas sin perdidas de conexión.

**Figura 67**

*Gráfica de resultados generales de las pruebas de carga*



**Nota.** Resultados generales de las pruebas de carga, registrar usuario, login, crear producto, listar producto, realizar compra. Elaborado por: Flores, Riofrío

### 3.4.3. Prueba de estrés

Las pruebas de estrés permiten conocer el rendimiento del sistema frente a sobrecargas de peticiones de información, identificando el momento en que la aplicación deja de funcionar. La finalidad de las pruebas de estrés es saturar la funcionalidad del sistema para identificar el fallo o bugs al momento de ejecutarlos.

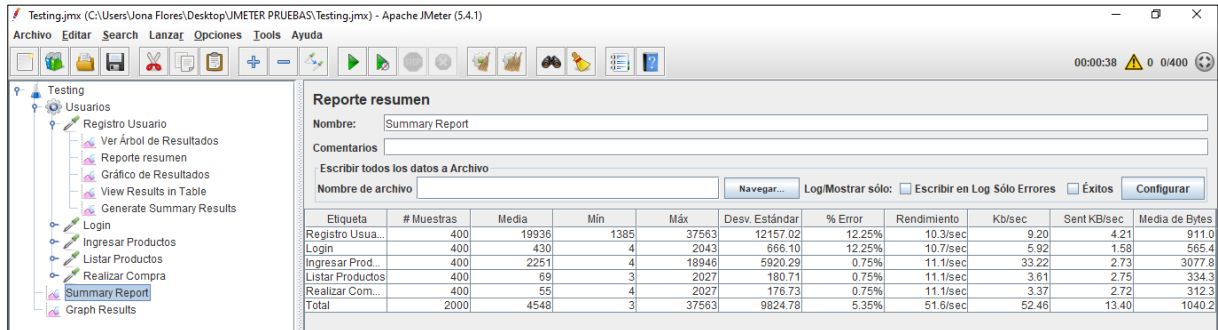
En la figura 68, se puede observar la prueba de estrés que se realizó con las peticiones anteriormente utilizadas en las pruebas de carga en la que se evidencia el margen de error que existe al sobrecargar de información el sistema, dando como resultados un porcentaje de error para el registro de usuarios de 12,25%, iniciar sesión 12,25%, ingresar producto 0,75%, listar productos 0,75%, realizar una compra 0,75% con un total de 5,35%. A partir de 400 usuarios



se visualiza un incremento del porcentaje de error, el incremento es leve lo cual no afecta a los procesos de las más funcionalidades del sistema, ver figura 69.

**Figura 68**

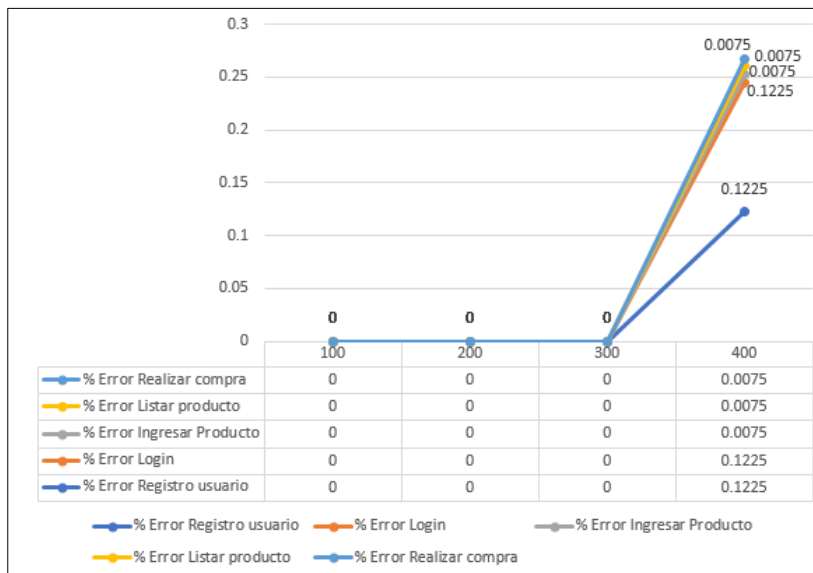
*Resultados de la prueba de estrés*



**Nota.** Sobrecarga de información para conocer el margen de error en la prueba. Elaborado por: Flores, Riofrío.

**Figura 69**

*Gráfica de los resultados de la prueba de estrés*



**Nota.** Gráfica de la sobrecarga de información para conocer el margen de error. Elaborado por: Flores, Riofrío.

### 3.5. Pruebas de uso de los recursos del dispositivo móvil

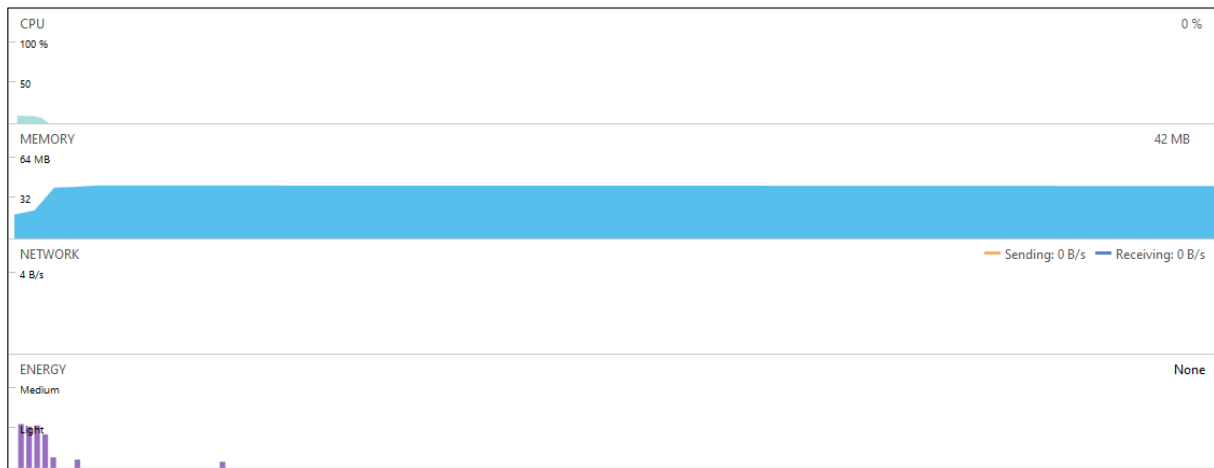
Las pruebas para medir el uso de los recursos del dispositivo móvil con la aplicación son muy importantes ya que nos permite conocer si los procesos que se están ejecutando. Las pruebas se las realizo en entorno de desarrollo Android Studio.

Para realizar las pruebas de consumo de recursos de la aplicación se utilizó la herramienta Android Profiler que se encuentra en Android Studio, la que permite conocer los recursos de CPU, memoria y batería que consume las aplicaciones.

En la figura 70, se observa el consumo de recursos frente a la ejecución de la aplicación iniciando en la pantalla principal de la App móvil la que indica un incremento leve en el uso de memoria y CPU del dispositivo.

#### Figura 70

*Consumo de recursos de la pantalla principal*

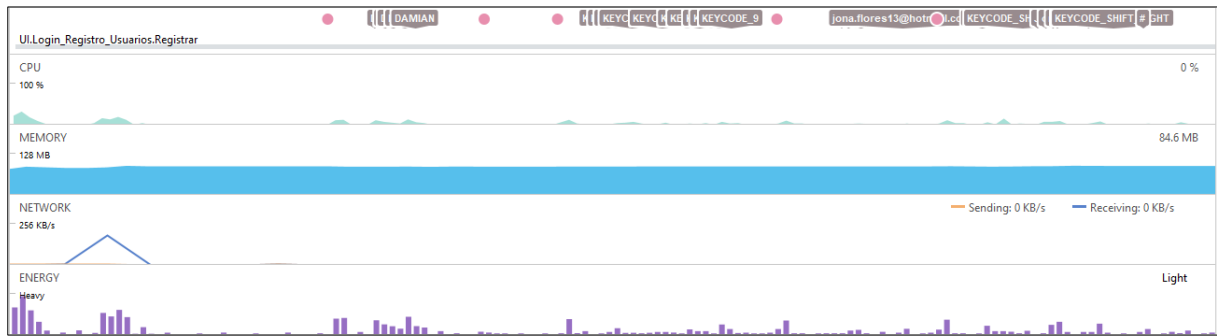


**Nota.** Gráfica uso de recursos de la pantalla principal. Elaborado por: Flores, Riofrío.

El consumo de recursos para la pantalla de registro de usuarios que se muestra en la figura 71, muestra que al momento de realizar el proceso de ingreso de usuarios el consume sube ya que envía varios parámetros.

## Figura 71

*Consumo de recursos de la pantalla registrar usuarios*

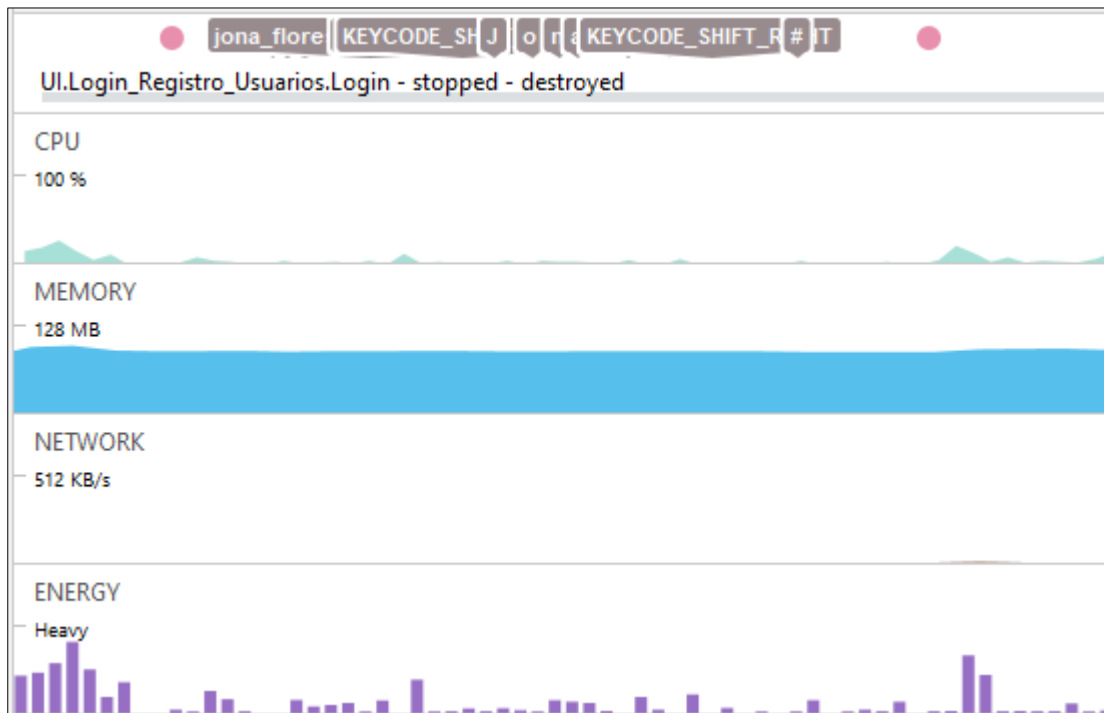


**Nota.** Gráfica consumo de recursos de la pantalla registrar usuario. Elaborado por: Flores, Riofrío.

En la figura 72, se muestra de igual manera un incremento al momento de ingresar datos como correo electrónico y contraseña para iniciar sesión el cual genera la consulta en servidor local al enviar los parámetros.

## Figura 72

*Consumo de recursos del login*

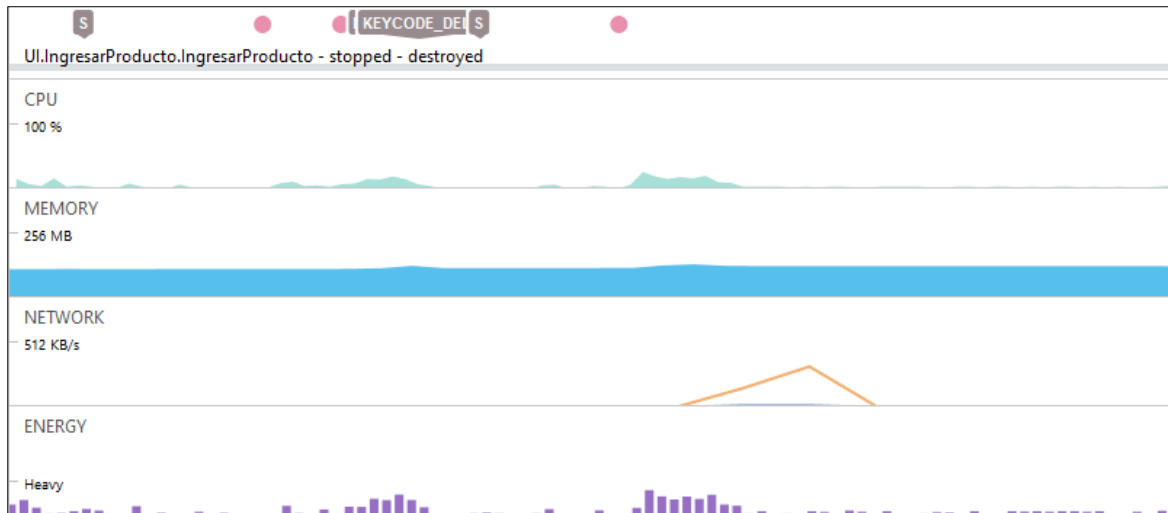


**Nota.** Gráfica consumo de recursos del login. Elaborado por: Flores, Riofrío.

En la figura 73, se observa un incremento constante de CPU y para el consumo de la memoria se visualiza un ligero aumento al ingresar la información del producto, también se puede ver un consumo leve de energía.

### Figura 73

*Consumo de recursos de la pantalla ingresar producto*

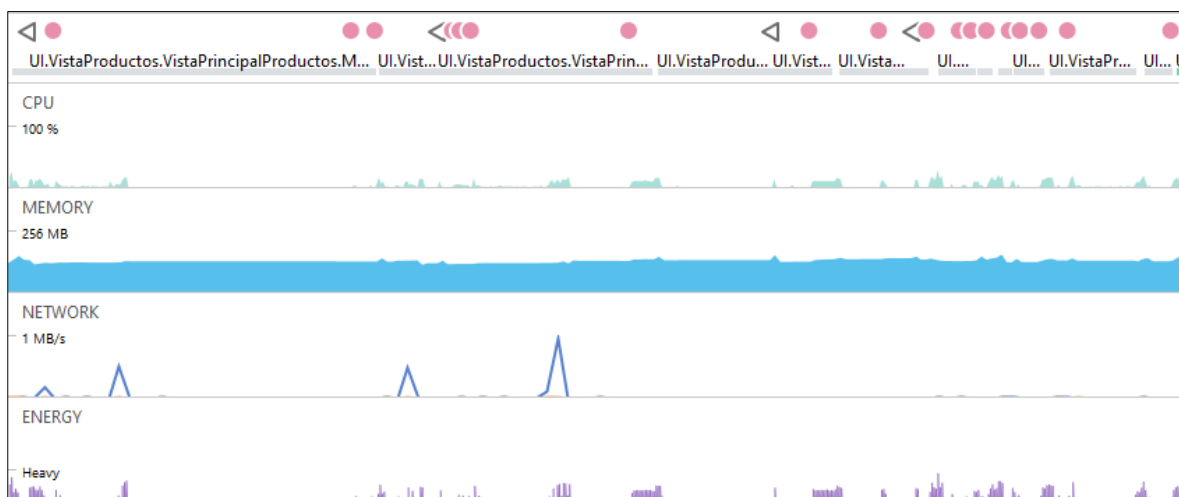


**Nota.** Gráfica consumo de recursos al ingresar un producto. Elaborado por: Flores, Riofrío.

En la figura 74, se puede evidenciar el consumo de recursos al momento de realizar una compra en la aplicación la misma que evidencia un incremento leve al realizar el proceso.

### Figura 74

*Consumo de recursos al realizar una compra*



**Nota.** Gráfica consumo de recursos al realizar una compra. Elaborado por: Flores, Riofrío.

Los resultados obtenidos en las pruebas de caja negra, carga y estrés identifican un funcionamiento óptimo de la aplicación móvil cumpliendo con los requisitos no funcionales lo que indica que el rendimiento del software no presenta fallos al momento de realizar su trabajo, por lo cual cumple con las necesidades establecidas por el usuario.

## CONCLUSIONES

Actualmente, tecnologías como los dispositivos móviles con internet se han convertido en parte del día a día de las personas, por lo que se ha desarrollado una aplicación móvil para el sector agrícola de las provincias del Ecuador, que permitirá a los pequeños productores involucrarse directamente con los consumidores en la comercialización de productos agrícolas.

El desarrollo de la aplicación móvil permite a los agricultores administrar la información de los usuarios además de vender productos agrícolas. También existe un proceso de ofertas para que los pequeños agricultores puedan promocionar sus productos. La aplicación se construirá como un prototipo que cumple con todas las funciones hasta se pueda publicar o entregar a una organización que la necesite.

Cuando se ejecute la aplicación móvil en el dispositivo móvil, el consumo de recurso del celular será leve, solo habrá un ligero incremento al momento de realizar la compra, por lo que el funcionamiento del teléfono móvil no se verá afectado cuando se utilice la aplicación.

Las herramientas tecnológicas disponibles actualmente permitieron el desarrollo de la aplicación móvil, mediante el uso de la plataforma Android Studio, que es un entorno de desarrollado específicamente para teléfonos inteligentes que utilizan el sistema operativo Android.

## **RECOMENDACIONES**

Para la implementación de nuevas versiones de la aplicación móvil es recomendable realizar una investigación de la situación en que se encuentra el sector agrícola en el Ecuador y así poder establecer cambios que ayuden a los agricultores a comercializar sus productos.

Se recomienda para futuras mejoras de la aplicación móvil establecer un módulo de facturación con diferentes métodos de pago online, que permita al usuario pagar de forma directa por medio de la aplicación.

Se recomienda para futuras actualizaciones de la aplicación móvil tener en cuenta las versiones de Android Studio para realizar cambios en la codificación de la App verificar si la versión del sistema operativo de los teléfonos celulares son compatibles con la aplicación.

Es recomendable realizar un mantenimiento preventivo y correctivo de la aplicación móvil para mitigar posibles fallos en el funcionamiento del sistema, el mismo que se recomienda que sea elaborado por el departamento de TICs.

## LISTA DE REFERENCIAS

### Libros

Basterra, Berteia, Borello, Castillo, & Venturi. (2017). *Android OS Documentation*.

<https://readthedocs.org/projects/androidos/downloads/pdf/latest/>

Urriolabeytia, J. (2020). *Android al máximo*. Creative Andina Corp.

[https://books.google.com.ec/books?id=ogXfDwAAQBAJ&printsec=copyright&redir\\_esc=y#v=onepage&q&f=false](https://books.google.com.ec/books?id=ogXfDwAAQBAJ&printsec=copyright&redir_esc=y#v=onepage&q&f=false)

### Páginas Web

Developers. (2020, 23 de junio). Cómo interpretar el ciclo de vida de una actividad.

<https://developer.android.com/guide/components/activities/activity-lifecycle?hl=es>

Gobierno de la Republica del Ecuador. (04 de 2020). *Ministerio de Agricultura y Ganadería*.

Obtenido de Ministros de Agricultura de América unen esfuerzos para afrontar impactos del COVID-19 en sector agroalimentario: <https://www.agricultura.gob.ec/ministros-de-agricultura-de-america-unen-esfuerzos-para-afrontar-impactos-del-covid-19-en-sector-agroalimentario/>

Mendoza, A. (18 de 08 de 2016). Obtenido de Android — Primeros pasos con Glide:

<https://medium.com/@alexmendozaventura>

Protección de datos. (2019, 4 de abril). Tipos de dispositivos móviles.

<https://www.protecciondatos.org/tipos-de-dispositivos-moviles/>

### Tesis

Alarcón, R. (2019). *DESARROLLO DE UNA APLICACIÓN MOVIL PARA LA COMERCIALIZACION DE PRODUCTOS EN EL SECTOR AGRICOLA DE ANDAHUAYLAS 2019* [Tesis de pregrado, Universidad Nacional José María Arguedas del Perú]. Repositorio Institucional Unajma.

<https://repositorio.unajma.edu.pe/handle/123456789/549>



- Alarcón, R. (2019). *DESARROLLO DE UNA APLICACIÓN MOVIL PARA LA COMERCIALIZACION DE PRODUCTOS EN EL SECTOR AGRICOLA DE ANDAHUAYLAS* 2019.  
<http://revistas.unitru.edu.pe/index.php/agroindscience/article/view/114/131>
- Auria, B., & Del Rosario, J. (2017). “*DESARROLLO DE UNA APLICACIÓN PARA COMERCIALIZAR LOS PRODUCTOS AGRÍCOLAS SIN INTERMEDIARIOS.*”  
<https://www.dspace.espol.edu.ec/retrieve/100095/D-106216.pdf>
- Banda, J. (2015). “*Integración de aplicaciones web y móvil, para la gestión de restaurantes y servicio al cliente*”.
- Castro, J. (2020). “*DISEÑO DE UN ECOSISTEMA DE SOFTWARE, PARA LA INTEROPERABILIDAD ENTRE SISTEMAS DE E-COMMERCE Y COURIER MEDIANTE APIS RESTFUL EFICIENTES Y SEGURAS.*”
- Coello, G. (2017). *Diseño de la plataforma de libros digitales de la UCE Trabajo*.  
<http://www.dspace.uce.edu.ec/handle/25000/21351%0Ahttp://www.dspace.uce.edu.ec/handle/25000/20368%0Ahttp://www.dspace.uce.edu.ec/bitstream/25000/12519/1/T-UCE-0015-726.pdf>
- García, A. (2017). *Restful: un caso de uso de gestión de bibliotecas*.  
[http://oa.upm.es/45203/12/TFG\\_ADOLFO\\_RODRIGO\\_GARCIA\\_NUNEZ.pdf](http://oa.upm.es/45203/12/TFG_ADOLFO_RODRIGO_GARCIA_NUNEZ.pdf)
- Haro, P., & Inca, L. (2016). “*DISEÑO E IMPLEMENTACIÓN UN SISTEMA DE TELECOMUNICACIÓN PARA LOS DISPOSITIVOS ELECTRÓNICOS DE PERSONAS CON DISCAPACIDADES DE LA UNIVERSIDAD NACIONAL DE CHIMBORAZO, QUE PERMITA AYUDAR EN LA ORIENTACIÓN, UBICACIÓN E INTERACCIÓN ENTRE LOS USUARIOS Y LA SOCIE.*”  
<http://dspace.unach.edu.ec/handle/51000/677%0Ahttp://dspace.unach.edu.ec/bitstream/5>

1000/1381/1/UNACH-EC-AGR-2016-0002.pdf

- Montalvo, D. (2018). *DESARROLLO DE UN SISTEMA INFORMATICO PARA LA TRAZABILIDAD DE LOS RESULTADOS DE LA MATERIA PRIMA (PLASMAS REACTIVOS Y NO REACTIVOS) EN EL PROGRAMA DE EVALUACION EXTERNO DEL DESEMPEÑO PARA EL CENTRO DE INVESTIGACIÓN PARA LA SALUD EN AMÉRICA LATINA (CISEAL)*.  
[http://repositorio.puce.edu.ec/bitstream/handle/22000/14629/Tesis Diego Paúl Montalvo Páez.pdf?sequence=1&isAllowed=y](http://repositorio.puce.edu.ec/bitstream/handle/22000/14629/Tesis%20Diego%20Pa%20l%20Montalvo%20P%20a%20z.pdf?sequence=1&isAllowed=y)
- Rincón, J., & Ceballos, M. (2016). *IMPLEMENTACIÓN DE UNA PLATAFORMA WEB Y UNA APLICACIÓN ANDROID PARA EL CONOCIMIENTO Y CALIFICACIÓN DE LA GESTIÓN DEL CONGRESO*.
- Silva, H. (2017). *Desarrollo e implementación de un sistema wweb con MVC para el control del mantenimiento preventivo y correctivo de los bienes del Cuerpo de Bomberos del Gobierno Autónomo Descentralizado Municipal de Santo Domingo; periodo 2016 - 2017*.
- Tapia, J., & García, M. (2016). *ANÁLISIS DE LA VULNERABILIDAD EN LAS APLICACIONES ANDROID DE LOS DISPOSITIVOS MÓVILES*.
- Tubón, G. (2020). *Aplicación móvil con Georreferenciación para gestión de pedidos a domicilio de un local de comida*.

## ANEXO 1

### *Manual de usabilidad de la aplicación*

## ANEXO 2

### *Implementación de la metodología*

XP (Programación Extrema) es una metodología que permite el desarrollo de la ingeniería de software, mediante el cual se construyó el proyecto basándonos en las especificaciones del cliente.

Se puede observar la descripción de los diferentes roles que fueron desempeñados tanto por el cliente como los responsables de la aplicación móvil.

### *Roles en el proyecto*

Rol	Responsable	Detalle
<b>Cliente</b>	Ing. Lina Zapata	Define los requerimientos de la aplicación móvil, que marcan las necesidades y prioridades del proyecto.
<b>Coach</b>	Ing. Lina Zapata	Se encarga de asesorar y orientar tanto al equipo de trabajo como al cliente.
<b>Programadores</b>	Jonathan Flores Andrea Riofrío	Equipo de trabajo que se encargará de la decodificación del proyecto.
<b>Tester</b>	Andrea Riofrío	Se encarga de realizar pruebas amplias del funcionamiento de la aplicación con el cliente.
<b>Tracker (Encargado del seguimiento)</b>	Ing. Lina Zapata	Es el encargado de llevar un control en el desarrollo de la aplicación, analizando el proceso que se realiza en cada cosa.

*Nota.* Responsabilidades que cumple cada miembro del proyecto. Elaborado por: Flores, Riofrío.

### **Reuniones**

- Las reuniones para evidenciar avances se quedó en mutuo acuerdo realizarlas cada semana, para ser más específicos cada viernes en horario vespertino, por medio de reuniones establecidas en Zoom.
- Las reuniones del equipo de trabajo se realizaron en horas de la tarde por mutuo acuerdo.

### **Fases de la metodología XP aplicadas en el desarrollo del proyecto**

Para el desarrollo de la aplicación se llevó a cabo las fases de la metodología XP que permitieron realizar la construcción de la aplicación móvil, las que serán descritas a continuación.

#### *Fases del proyecto*

**Planificación:** La planificación se llevó a cabo mediante una reunión por Zoom, en la que por medio de las historias de usuarios el cliente nos especificó los requerimientos del sistema y se quedó de acuerdo que cada semana se evidenciarían avances.

**Diseño:** Con la herramienta Balsamiq Wireframes se diseñó un prototipo de la interfaz del sistema que fueron mostradas en la reunión acordada.

**Programación:** La integración de los dos programadores para realizar la codificación de la aplicación fue primordial ya que mediante observaciones expuestas por cada parte del equipo de trabajo permitieron construir un software con código universal, que pueda ser entendido por otro programador.

**Pruebas:** Las pruebas a la aplicación móvil se realizaron con el cliente en cada reunión establecida, para revisar que el funcionamiento del software funcione correctamente.

También las pruebas nos permitieron identificar errores en la aplicación que fueron corregidos en cada versión presentada.

### ANEXO 3

#### *Pruebas de funcionalidad de la aplicación móvil*

Las pruebas de funcionalidad tienen como finalidad mostrar los resultados de la ejecución de la aplicación móvil, identificando los errores y dando soluciones a los mismos.

Cabe mencionar que para realizar la prueba de funcionalidad se solicitó la ayuda un grupo de 3 personas que forman parte de una fundación de productores agrícolas.

N°	Nombre de la prueba	Pasos	Cumple	No cumple	Observaciones
1	Registrarse en el sistema	1. Ingresar registrarse en el sistema 2. Ingresar datos personales	X		
2	Ingresar al sistema	1. Ingresar correo electrónico 2. Ingresar contraseña	X		
3	Recuperar contraseña	1. Ingresar correo electrónico	X		
4	Ingresar un producto	1. Ingresar al sistema 2. Seleccionar menú 3. Escoger mis productos 4. Agregar producto 5. Ingresar datos del producto	X		

5	Vista detalle del producto	1. Seleccionar un producto	X		
6	Comprar un producto	1. Ingresar al sistema 2. Seleccionar un producto 3. Agregar al carrito 4. Seleccionar forma de pago 5. Finalizar compra	X		
8	Ingresar un producto con oferta	1. Ingresar al sistema 2. Seleccionar menú 3. Escoger mis productos 4. Agregar producto 5. Ingresar datos del producto (oferta descripción)	X		
9	Actualizar, eliminar un producto	1. Ingresar al sistema 2. Seleccionar menú 3. Escoger mis productos 4. Seleccionar producto que se quiera modificar o eliminar	X		

		5. Seleccionar modificar o eliminar según sea el caso			
<b>10</b>	Visualizar compras realizada	1. Ingresar al sistema 2. Seleccionar menú 3. Seleccionar mis compras	X		
<b>11</b>	Visualizar ventas realizada	1. Ingresar al sistema 2. Seleccionar menú 3. Seleccionar mis ventas			