



UTMACH

FACULTAD DE INGENIERÍA CIVIL

CARRERA DE INGENIERÍA DE SISTEMAS

DESARROLLO DE UNA APLICACIÓN MÓVIL PARA EL CONTROL DE
PEDIDOS DELIVERY CRUZALUD EN LA CIUDAD DE MACHALA
UTILIZANDO REACT NATIVE.

BRITO AGUILAR JONATHAN JAVIER
INGENIERO DE SISTEMAS

MACHALA
2023



UTMACH

FACULTAD DE INGENIERÍA CIVIL

CARRERA DE INGENIERÍA DE SISTEMAS

DESARROLLO DE UNA APLICACIÓN MÓVIL PARA EL
CONTROL DE PEDIDOS DELIVERY CRUZALUD EN LA CIUDAD
DE MACHALA UTILIZANDO REACT NATIVE.

BRITO AGUILAR JONATHAN JAVIER
INGENIERO DE SISTEMAS

MACHALA
2023



UTMACH

FACULTAD DE INGENIERÍA CIVIL

CARRERA DE INGENIERÍA DE SISTEMAS

TRABAJO TITULACIÓN
PROPUESTAS TECNOLÓGICAS

DESARROLLO DE UNA APLICACIÓN MÓVIL PARA EL CONTROL DE PEDIDOS
DELIVERY CRUZALUD EN LA CIUDAD DE MACHALA UTILIZANDO REACT
NATIVE.

BRITO AGUILAR JONATHAN JAVIER
INGENIERO DE SISTEMAS

CARTUCHE CALVA JOFFRE JEORWIN

MACHALA, 14 DE MARZO DE 2023

MACHALA
2023

Tesis

INFORME DE ORIGINALIDAD

8%

INDICE DE SIMILITUD

8%

FUENTES DE INTERNET

1%

PUBLICACIONES

4%

TRABAJOS DEL
ESTUDIANTE

FUENTES PRIMARIAS

1	Submitted to Consorcio CIXUG Trabajo del estudiante	1%
2	dspace.ups.edu.ec Fuente de Internet	1%
3	www.coursehero.com Fuente de Internet	1%
4	dialnet.unirioja.es Fuente de Internet	1%
5	www.slideshare.net Fuente de Internet	<1%
6	repositorio.utn.edu.ec Fuente de Internet	<1%
7	hdl.handle.net Fuente de Internet	<1%
8	Submitted to Universidad Distrital FJDC Trabajo del estudiante	<1%
9	guiadev.com Fuente de Internet	<1%

CLÁUSULA DE CESIÓN DE DERECHO DE PUBLICACIÓN EN EL REPOSITORIO DIGITAL INSTITUCIONAL

El que suscribe, BRITO AGUILAR JONATHAN JAVIER, en calidad de autor del siguiente trabajo escrito titulado DESARROLLO DE UNA APLICACIÓN MÓVIL PARA EL CONTROL DE PEDIDOS DELIVERY CRUZALUD EN LA CIUDAD DE MACHALA UTILIZANDO REACT NATIVE., otorga a la Universidad Técnica de Machala, de forma gratuita y no exclusiva, los derechos de reproducción, distribución y comunicación pública de la obra, que constituye un trabajo de autoría propia, sobre la cual tiene potestad para otorgar los derechos contenidos en esta licencia.

El autor declara que el contenido que se publicará es de carácter académico y se enmarca en las disposiciones definidas por la Universidad Técnica de Machala.

Se autoriza a transformar la obra, únicamente cuando sea necesario, y a realizar las adaptaciones pertinentes para permitir su preservación, distribución y publicación en el Repositorio Digital Institucional de la Universidad Técnica de Machala.

El autor como garante de la autoría de la obra y en relación a la misma, declara que la universidad se encuentra libre de todo tipo de responsabilidad sobre el contenido de la obra y que asume la responsabilidad frente a cualquier reclamo o demanda por parte de terceros de manera exclusiva.

Aceptando esta licencia, se cede a la Universidad Técnica de Machala el derecho exclusivo de archivar, reproducir, convertir, comunicar y/o distribuir la obra mundialmente en formato electrónico y digital a través de su Repositorio Digital Institucional, siempre y cuando no se lo haga para obtener beneficio económico.

Machala, 14 de marzo de 2023



BRITO AGUILAR JONATHAN JAVIER
0940301856

DEDICATORIA

Dedico el presente trabajo a mis padres y hermanas que gracias a su confianza y enseñanza al largo de mi vida me han enseñado una serie de valores que me han ayudado a formar como estudiante, hijo y ser humano.

También a mis amigos y docentes de la universidad quienes me ayudaron durante mi formación con su apoyo y consejos.

Sr. Brito Aguilar Jonathan Javier

AGRADECIMIENTO

Agradezco a Dios por permitirme estar con vida y por todo lo bueno que ha hecho por mí, dándome fuerzas para seguir adelante sin importar la situación.

También agradezco a mis padres por el sacrificio que han hecho por mí, estar pendiente para que pueda llevar a cabo mi formación de ser profesional.

También agradezco a mis amigos Davis, Juan y Freddy que me han brindado su apoyo y confianza desde el inicio sin importar la situación.

A mi enamorada Maite Coronel que ha estado conmigo apoyándome en estos últimos momentos alentándome a seguir adelante.

Sr. Brito Aguilar Jonathan Javier

RESUMEN

En consecuencia, a la pandemia provocada por el virus Covid-19 en estos últimos los negocios se vieron afectados al no poder brindar sus servicios de venta de productos con normalidad debido a las medidas de bioseguridad y restricciones establecidas por las autoridades competentes, por tal razón muchos negocios comenzaron a automatizar sus negocios ofertando el servicio con entrega a domicilio a través de redes sociales.

Sin embargo, aplicar esta modalidad no brindaba las herramientas adecuadas para que realicen el servicio de entrega a domicilio, impidiendo a los clientes ver detalles de los productos a adquirir, además de buscar en diferentes redes sociales hasta encontrar lo que estaba necesitando.

Frente a esta problemática se propone el desarrollo de una aplicación móvil para la toma de pedidos del negocio CRUZ SALUD, el desarrollo del trabajo abarca las necesidades y recomendaciones del negocio, para ofrecer a sus clientes habituales este servicio, así como a nuevos clientes.

Para el desarrollo del proyecto de investigación se utilizó la metodología SWIRL que se caracteriza por ofrecer una mejor flexibilidad con respecto a los cambios y que permite integrarse ante los diferentes procesos inmerso en el desarrollo de software, además con el apoyo de Visual Code, Expo y React Native el cual es un Framework JavaScript el cual es para el desarrollo de aplicaciones nativas, se obtuvo una aplicación con una interfaz amigables para facilitar el ingreso y consulta de los productos que requiere el cliente.

La aplicación contara con un registro para los clientes a través de su correo electrónico para poder acceder a las opciones del sistema, en el menú principal mostrara al usuario el listado de los productos disponibles con su respectivo precio, además proporcionara un buscador para agilizar la compra de los productos.

El cliente podrá ver el listado de todos los productos agregados a su carrito de compras, para agregar su dirección domiciliaria y esta información se notificará al administrador para que realice el respectivo envío a través del repartidor.

Una vez finalizada la aplicación se realizó una prueba de calidad aplicando la norma de calidad ISO/IEC 9126, que proporciona métricas que califican la funcionalidad, confiabilidad, usabilidad, eficiencia, mantenibilidad y portabilidad, todas estas características permiten determinar si la aplicación es de calidad o no, con la aplicación de esta norma se pudo determinar que el software posee un alto grado de cumplimiento de los requisitos del usuario.

Como resultado, se lograron excelentes niveles de calidad, según la escala de Likert; en funciones sobre funciones para 4 criterios, una calificación de 4/4 es excelente; también en carácter confiabilidad, 2/3 excelente y 1/3 buena 3 criterios; facilidad de uso continua, puntuado 4/4 excelente en los 4 criterios; la características de desempeño, 2 criterios evaluados excelente 1/2, bueno 1/2; con funciones de mantenimiento, los resultados fueron 1/2 excelente y 1/2 tan bueno en sus 2 criterios, y finalmente obtiene 2 criterios de portabilidad, 1/2 excelente tan malo como 1/2, se puede decir que el software cumple en gran medida con los requisitos y detalles del usuario, planificación y diseño.

Palabras clave: Aplicación móvil, Swirl, Node.JS, React Native, Expo

ABSTRACT

Consequently, due to the pandemic caused by the Covid-19 virus, in the latter, businesses were affected by not being able to provide their product sales services normally due to biosecurity measures and restrictions established by the competent authorities, for this reason. many businesses began to automate their businesses by offering the service with home delivery through social networks.

However, applying this modality did not provide the appropriate tools to carry out the home delivery service, preventing customers from seeing details of the products to be purchased, in addition to searching on different social networks until they found what they were needing.

Faced with this problem, the development of a mobile application for taking orders from the CRUZ SALUD business is proposed, the development of the work covers the needs and recommendations of the business, to offer this service to its regular customers, as well as to new customers.

For the development of the research project, the SWIRL methodology was used, which is characterized by offering better flexibility with respect to changes and that allows integration with the different processes immersed in software development, in addition to the support of Visual Code, Expo and React Native which is a JavaScript Framework which is for the development of native applications, an application with a friendly interface was obtained to facilitate the entry and consultation of the products that the client requires.

The application will have a registration for customers through their email to access the system options, in the main menu it will show the user the list of available products with their respective prices, it will also provide a search engine to expedite the purchase of the products.

The customer will be able to see the list of all the products added to his shopping cart, to add his home address and this information will be notified to the administrator so that he can make the respective shipment through the delivery man.

Once the application was finished, a quality test was carried out applying the ISO/IEC 9126 quality standard, which provides metrics that qualify the

functionality, reliability, usability, efficiency, maintainability and portability, all these characteristics allow us to determine if the application is of quality. or not, with the application of this standard it can be determined that the software has a high degree of compliance with the user's requirements.

As a result, excellent quality levels were achieved, according to the Likert scale; in features on features for 4 criteria, a rating of 4/4 is excellent; also in reliability, 2/3 excellent and 1/3 good 3 criteria; continued ease of use, rated 4/4 excellent on all 4 criteria; the performance characteristics, 2 criteria evaluated excellent 1/2, good 1/2; with maintenance features, the results were 1/2 excellent and 1/2 as good on your 2 criteria, and finally you get 2 portability criteria, 1/2 excellent as bad as 1/2, you can say that the software complies on largely with user requirements and details, planning and design.

Keywords: Mobile app, Swirl, Node.JS, React Native, Expo

CONTENIDO

DEDICATORIA.....	1
AGRADECIMIENTO	2
RESUMEN.....	3
ABSTRACT.....	5
INTRODUCCIÓN.....	10
1. CAPÍTULO I. DIAGNOSTICO DE NECESIDADES Y REQUERIMIENTOS.....	12
1.1. ÁMBITO DE APLICACIÓN: DESCRIPCIÓN DEL CONTEXTO Y HECHOS DE INTERÉS.....	12
1.2. ESTABLECIMIENTO DE REQUERIMIENTOS	13
1.3. JUSTIFICACIÓN DE REQUERIMIENTO	15
2. CAPÍTULO II. DESARROLLO DEL PROTOTIPO	16
2.1. DEFINICIÓN DEL PROTOTIPO TECNOLÓGICO	16
2.1.1. Enfoque del prototipo.....	16
2.2. FUNDAMENTACIÓN TEÓRICA DEL PROTOTIPO	16
2.2.1. Arquitectura de software	17
2.2.2. PostgreSQL.....	18
2.2.3. Herramientas Case	20
2.2.5. JavaScript.....	21
2.2.6. Node.Js	22
2.2.7. React Native	22
2.2.8. Metodología de desarrollo de software.....	23
2.3. OBJETIVOS DEL PROTOTIPO.....	26
2.3.1. OBJETIVO GENERAL	26
2.3.2. OBJETIVOS ESPECÍFICOS.....	26
2.4. DISEÑO DEL PROTOTIPO	26
2.4.1. FASE DE ANÁLISIS	26
2.4.2. FASE DE PLANIFICACIÓN.....	32
2.4.3. FASE DE MODELADO.....	36
2.5. EJECUCIÓN Y/O ENSAMBLAJE DEL PROTOTIPO	54
2.5.1. Módulo principal	54
2.5.2. Módulo productos disponibles	55
2.5.3. Módulo general de pedidos.....	56
2.5.4. Módulo administración de datos de facturación	56
2.5.5. Módulo perfil del cliente.....	57
3. CAPÍTULO III. EVALUACIÓN DEL PROTOTIPO.....	59

3.1. PLAN DE EVALUACIÓN	59
3.2. RESULTADOS DE LA EVALUACIÓN	59
3.3. CONCLUSIONES	63
3.4. RECOMENDACIONES	64
REFERENCIAS BIBLIOGRAFICAS	65
ANEXOS	68
Anexo 1: Evaluación utilizando la herramienta MobSF de análisis dinámico	68

ÍNDICE DE TABLAS

Tabla 1: Características de Postgresql.....	19
Tabla 2: Matriz de interesados	27
Tabla 3: Funcionalidades del sistema	27
Tabla 4: Limitaciones del sistema	28
Tabla 5: Factibilidad Operativa.....	29
Tabla 6: Factibilidad Técnica	29
Tabla 7: Factibilidad Económica	30
Tabla 8: Requerimientos Funcionales	31
Tabla 9: Requerimientos no funcionales	32
Tabla 10: Historia de usuario Inicio de sesión.....	32
Tabla 11: Historia de usuario registro de usuario	33
Tabla 12: Historia de usuario Administración de datos de Facturación.....	33
Tabla 13: Historia de usuario perfil del negocio	34
Tabla 14: Historia de usuario Visualización de los productos	34
Tabla 15: Historia de usuario Gestión de productos	34
Tabla 16: Historia de usuario Gestión de Pedidos.....	35
Tabla 17: Resultados de la evaluación.....	59
Tabla 18: Resumen resultado de evaluación	61

ÍNDICE DE FIGURAS

Figura 1 : Fundamentación teórica del prototipo.....	17
Figura 2: Arquitectura Cliente-servidor [10]	17
Figura 3: Modelado de tres capas.....	18
Figura 4 : Representación React Native [33].....	23
Figura 5 :Fases del ciclo de vida de la metodología Swirl [40]	25
Figura 6: Gestión de cronograma.....	36
Figura 7: Diseño de base de datos no relacional.....	37
Figura 8: Diseño navegacional	37
Figura 9 Diseño: Menú principal.....	38
Figura 10 Diseño: Registro de usuario	39
Figura 11 Diseño: Menú de opciones de la aplicación.....	40
Figura 12 Diseño: Perfil del puesto de trabajo	41

Figura 13 Diseño: Registro de pedido	42
Figura 14 Diseño: Registro oferta de trabajo.....	43
Figura 15: Caso de uso Inicio de sesión	44
Figura 16: Casos de uso Registro.....	45
Figura 17: Casos de uso Visualización de productos.....	46
Figura 18: Caso de uso Registro de pedido	46
Figura 19: Caso de uso Gestión de perfil.....	47
Figura 20: Diagrama de actividades Registro de cliente.....	48
Figura 21: Diagrama de actividades Inicio de Sesión	48
Figura 22: Diagrama de actividades selección de productos	49
Figura 23: Diagrama de actividades visualizar producto	49
Figura 24: Diagrama de secuencia registro de usuario	50
Figura 25: Diagrama de secuencia Inicio de sesión.....	51
Figura 26: Diagrama de secuencia registro de pedido.....	52
Figura 27: Diagrama de secuencia buscar producto	53
Figura 28: Módulo principal Inicio de Sesión	54
Figura 29: Módulo productos disponibles	55
Figura 30: Módulo general de pedido.....	56
Figura 31: Módulo administración de datos de facturación	57
Figura 32: Módulo perfil del cliente	58
Figura 34: Resumen de los resultados de evaluación	62

INTRODUCCIÓN

A consecuencia del inicio del Covid-19 los negocios han tenido que adaptarse a las nuevas tecnologías para poder ofertar sus productos, como son las aplicaciones de pedido de comida a domicilio; Francesca Ferrero señala que en tiempos pre pandemia el servicio de delivery solo constituía hasta un 15% de las ventas, pero esta tasa alcanzó el 50% en mayo del 2021 [1].

La velocidad con la que el virus se ha propagado en el país y el resto del mundo ha desestabilizado las condiciones del mercado, afectando a sectores económicos y por ende a los hogares consumidores, ya que nadie estaba preparado para un evento de esta magnitud. A pesar del sombrío panorama económico, algunas empresas se han aventurado en las herramientas digitales para sobrevivir y continuar generando ingresos, incluidas las aplicaciones digitales, o "Apps", programas informáticos diseñados para realizar tareas complejas y mejorar la experiencia informática [2].

Actualmente vivimos en un mundo de mucho crecimiento en donde la aplicación de la tecnología ha revolucionado las actividades de la vida cotidiana a través de los dispositivos móviles como teléfonos inteligentes o smartphone. Tu herramienta de teléfono inteligente más importante viene con una aplicación móvil, por lo que todas las actividades como el trabajo, la educación, el entretenimiento, la salud, el comercio y los negocios de alimentos no son una excepción.

Hoy en día, la automatización de los negocios es una de las claves para lograr el éxito, y cada vez es más importante imprescindible en tiempos de crisis, porque quien no es capaz de crear y gestión eficaz de las herramientas de innovación relacionadas con la tecnología aplicadas al producto, la estrategia o la gestión en general es claramente peor si lo hacen, tendrá graves consecuencias para sus dueños.

Las empresas de comida y ventas de artículos aprovechan las diversas funciones que tienen de las aplicaciones móviles para hacer conocer sus servicios y así tener mayores ventas.

Ante esta situación se plantea el desarrollo de una aplicación móvil para la gestión de compra de productos con entrega a domicilio, donde se muestren todos los productos que oferta el negocio, con la finalidad de que los usuarios tengan una mejor experiencia de compra, mediante la interacción con menús digitales y el seguimiento de los pedidos realizados.

El proyecto requiere de una serie de herramientas para que sea eficiente y escalable por lo que se han considerado tecnologías como Node JS, y React Native como framework para el funcionamiento multiplataforma de la App.

A continuación, se detalla la estructura del documento de la siguiente forma:

Capítulo 1: Dentro de este capítulo se analizarán los requerimientos del prototipo además de detallar el desarrollo de la problemática y justificación del proyecto.

Capítulo 2: Se muestra la documentación del prototipo, la fundamentación teórica, además de la metodología que se usa, los objetivos y los diseños de la aplicación.

Capítulo 3: Se procede a realizar la evaluación del prototipo, la funcionalidad, se analizarán los resultados para determinar las correspondiente conclusiones y recomendaciones

1. CAPÍTULO I. DIAGNOSTICO DE NECESIDADES Y REQUERIMIENTOS

1.1. ÁMBITO DE APLICACIÓN: DESCRIPCIÓN DEL CONTEXTO Y HECHOS DE INTERÉS

El uso de dispositivos móviles hoy en día se considera un elemento básico en el desarrollo de la comunicación, las tecnologías de la información y la comunicación son el resultado de desarrollos científicos que permiten a las personas interactuar en un entorno digital con acceso en tiempo real a grandes cantidades de datos, productos y servicios [3].

Hoy en día, existen todo tipo de aplicaciones que permiten satisfacer las necesidades de las personas. Los juegos, escuchar música, leer libros, ver videos, redes sociales, etc. son cada vez más populares y la demanda también aumenta.

Debido a la pandemia por el COVID-19 en el año 2020 uno de los mayores retos de los negocios fue el alto nivel de bajo de ventas, la crisis del Covid-19 está obligando a las empresas y negocios a buscar oportunidades digitales e identificar formas de entregar productos y servicios con un contacto físico mínimo y seguro. [4].

Actualmente existen múltiples plataformas y aplicaciones móviles que brindan diferentes servicios para permitir cubrir las necesidades de las personas como es pedir transporte, pago de servicios básicos, pedido de comidas rápidas en locales comerciales, hasta algo simple que es pedir productos de primera necesidad de un local a través de aplicaciones móviles.

En la actualidad, muchos negocios utilizaban anuncios en los periódicos para promocionar su negocio y vacantes existentes, pero debido al confinamiento se redujo su venta y en consecuencia la pérdida de empleo, haciendo del uso de redes sociales y aplicaciones móviles un mayor incremento, lo cual obligó a muchos negocios a utilizar estos medios y así evitar aglomeraciones en los diferentes establecimientos

Desde el 2020 la creación de aplicaciones móviles para el servicio de Delivery se incrementó para satisfacer las necesidades de los negocios, tanto así que cada ciudad se adapta a las necesidades de sus clientes para crear aplicaciones amigables para el usuario y fácil de usar, pero en Ecuador el sector comercial se

ve afectado por el proceso de compra, estamos pasando de comprar en tiendas físicas tradicionales a comprar en línea a través de redes sociales, páginas web, apps y otras herramientas de compra virtual, aunque esta forma de comprar parece más fácil, es también un proceso complejo, ya que se han encontrado problemas como retrasos en los plazos de entrega, problemas de pago, falta de selección de productos, problemas de ubicación, etc.; como resultado, los clientes no procesan las compras ni evitan las aplicaciones, lo que genera una experiencia de usuario más deficiente.

Al poder ver estos tipos de inconvenientes que se presentan dentro de la sociedad, se puede conocer que para lograr el éxito y la experiencia sea satisfactoria se debe desarrollar servicios rápidos y que garanticen la entrega a tiempo del servicio a ofrecer.

1.2. ESTABLECIMIENTO DE REQUERIMIENTOS

La presente aplicación móvil se denomina PideAhora la cual tiene como finalidad brindar a las personas que puedan comprar los productos con el servicio de entrega a domicilio.

Por lo tanto, se plantea desarrollar una aplicación móvil donde se agrupe todo lo antes mencionado y para eso se obtienen los siguientes requerimientos:

Requerimientos funcionales

- La aplicación permitirá el registro de los clientes por medio de correo electrónico.
- El registro del cliente debe realizarse desde la aplicación donde deberá registrar el usuario información relevante del mismo.
- Los productos registrados tendrán un perfil donde se mostrará su información básica: código, nombre, categoría, precio, stock disponible.
- La orden de pedido podrá ser visualizada y eliminada en cualquier momento por el dueño del negocio que tenga acceso a la cuenta.
- El dueño del negocio y el cliente tendrán un perfil donde se mostrará su información: Nombre del propietario, correo, nombre de la empresa, ciudad, dirección y teléfono.

- La información del negocio y propietario podrá ser actualizada y eliminada en cualquier momento.
- La aplicación contará con un panel general donde se visualizará todos los pedidos realizados.
- La aplicación poseerá un buscador y un filtro para los usuarios puedan encontrar los productos con mayor facilidad.
- Los usuarios podrán observar todas sus facturas de los pedidos realizados.
- En cuanto al método de pago, solo se manejará con dinero en efectivo, el usuario podrá avisar si el repartidor debe llevar cambio o no.

Requerimientos no funcionales

- **Requisitos de rendimiento**

El tiempo de respuesta para las consultas o toma de pedidos deben ser rápidas, garantizando que no afecte el tiempo de búsqueda dentro de la base de datos.

- **Seguridad**

Garantizar la seguridad del sistema con respecto a la información y datos de todos sus clientes, manteniendo la confiabilidad de estos.

- **Fiabilidad**

La interfaz de la aplicación debe ajustarse a las diferentes dimensiones de los distintos dispositivos móviles.

- **Disponibilidad**

Para realizar los pedidos solo lo podrá hacer en el horario establecido por el negocio.

- **Mantenibilidad**

La aplicación debe ser fácil de utilizar para que sea administrada por los usuarios así no tengan experiencia en el manejo de aplicaciones móviles.

- **Portabilidad**

La aplicación será implantada bajo la plataforma Android e IOS

Requisitos comunes de las interfaces

- **Interfaces de hardware**

Es necesario poseer dispositivos móviles funcionales

- **Interfaces de software**

En dispositivos Android debe ser superior a la versión 9.0

En dispositivos IOS debe ser superior a la versión 14.0

- **Interfaces de comunicación**

El servidor, cliente y aplicación se comunicarán entre sí mediante protocolos y estándares en internet.

1.3. JUSTIFICACIÓN DE REQUERIMIENTO

Desde la llegada de las actuales tecnologías y la creación de las diferentes plataformas, las grandes, medianas, medianas y pequeñas empresas e incluso pequeños emprendimientos han optado por una estrategia muy eficaz y útil como son las aplicaciones móviles de servicio de delivery, el uso de estas tecnologías han permitido que los clientes interactúen en tiempo real con el negocio [5], el motivo del desarrollo de esta aplicación móvil es para mejorar la interacción entre el negocio con las personas que deseen adquirir un producto que se ofrece en el neológico. La aplicación lo podrán utilizar los usuarios desde un smartphone con el sistema operativo Android o iOS.

Hoy en día existen una gran cantidad de aplicaciones móviles que ofrecen entregas a domicilio, su inicio radica en buscar la mejor opción para el problema latente en la sociedad en base a estudios realizados se detectó que los usuarios buscan ahorrar tiempo, dinero y que las entregas sean rápidas y seguras con la finalidad que el cliente evite salir de su casa y no correr riesgos ni pérdida de tiempo.[6]

Para facilitar agilizar y facilitar los procesos de venta del negocio se considera la creación de la aplicación móvil denominada PideAhora, para que los clientes tengan la facilidad de poder comprar sus productos preferidos desde la comodidad de su domicilio y así eviten trasladarse hasta el negocio y evitar alguno inconveniente al momento de realizar un pedido en el negocio.

Esta aplicación también contará con un panel general con todos los productos disponibles del negocio con las principales características y precio para facilitar la búsqueda del cliente.

2. CAPÍTULO II. DESARROLLO DEL PROTOTIPO

2.1. DEFINICIÓN DEL PROTOTIPO TECNOLÓGICO

2.1.1. Enfoque del prototipo

El enfoque de este proyecto tiene como objetivo desarrollar una aplicación móvil que permita realizar las compras del negocio Cruz Salud con un diseño intuitivo y fácil de usar para los usuarios finales, para su desarrollo se usará Node.js y el IDE Android Studio, con un backend basado en la nube a través de los servicios que ofrece PostgreSQL. El proyecto se desarrolló con el objetivo de aumentar las ventas del negocio y facilitar al cliente realizar pedidos desde la comodidad de su hogar.

2.2. FUNDAMENTACIÓN TEÓRICA DEL PROTOTIPO

Para llevar a cabo un software de calidad se debe implementar herramientas significativas para su desarrollo, por lo tanto, para este proyecto se ha implementado tecnologías modernas, centradas al desarrollo de aplicaciones multiplataformas y elaboración de proyectos. Como se aprecia en la figura 1 se detalla cada una de estas.

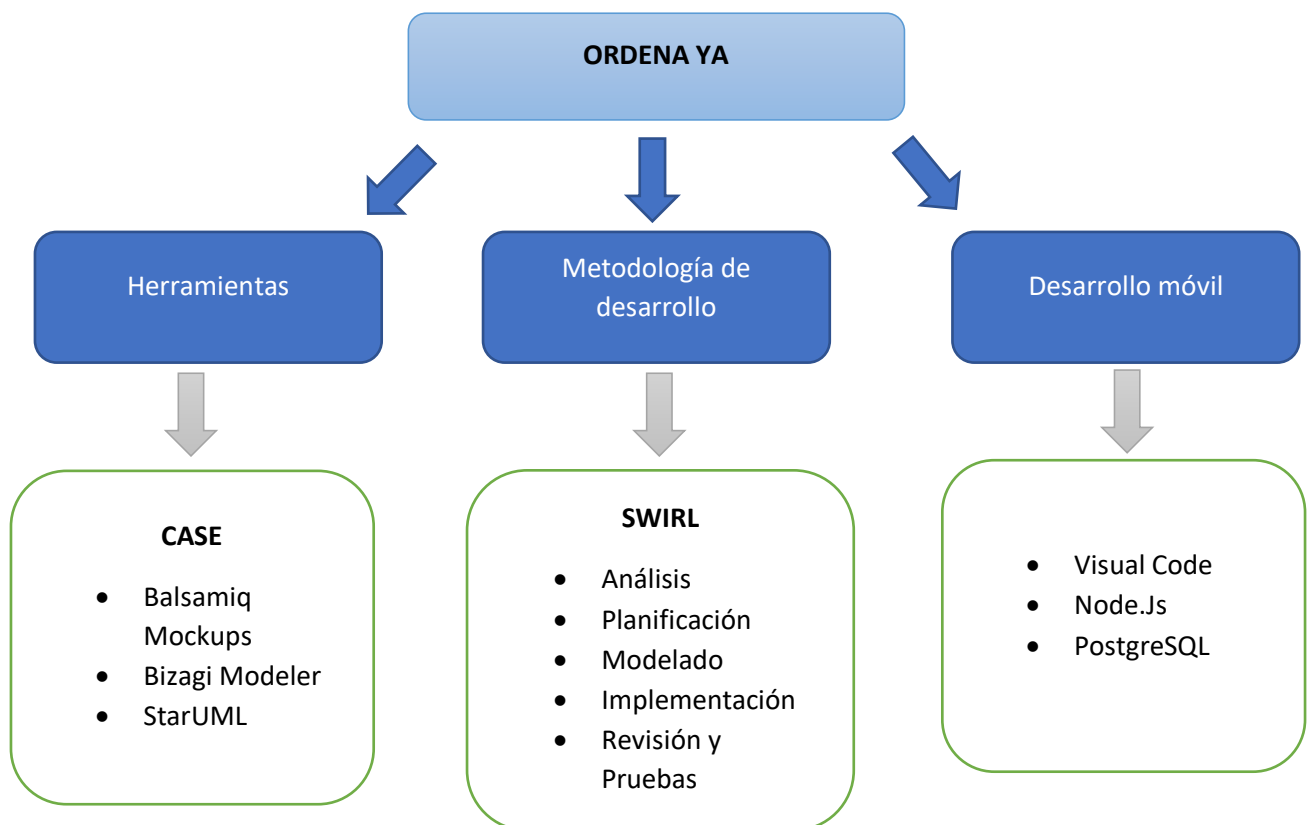


Figura 1 : Fundamentación teórica del prototipo

2.2.1. Arquitectura de software

Un modelo de arquitectura de software es la presentación actual de los componentes que corresponden a la estructura general del sistema en sí, con el objetivo de que se cumplan los requerimientos establecidos [7]

Los modelos de arquitectura de software tienen como finalidad soportar múltiples tareas de desempeño como son las operaciones del sistema, el tiempo de respuesta de las peticiones y el rendimiento [8]

2.2.1.1. Arquitectura Cliente/servidor

Este tipo de arquitectura es adecuada para el desarrollo de aplicaciones móviles porque se dividen en procesos independientes que cooperan entre sí para intercambiar información, servicios o recursos [9].

Para demostrar el modelo cliente-servidor, visualice la figura 1.



Figura 2: Arquitectura Cliente-servidor [10]

Arquitectura implementada

Para asegurar la eficiencia en la implementación de la aplicación, la arquitectura de software que se ha escogido es la arquitectura cliente-servidor, donde el usuario mediante el dispositivo final que use pueda realizar solicitudes a los servidores donde está alojada la información necesaria que el cliente lo requiera y que pueda interpretarla en su pantalla.

2.2.1.2. Modelado de tres capas

Para garantizar la escalabilidad del sistema es importante implementar una arquitectura basada en capas, de esta forma permite de manera eficaz el buen

funcionamiento de las tareas del sistema de forma independiente sin implicar otras capas y que el rendimiento sea el más óptimo [11].

En la figura 2 se detalla el modelado tres capas.

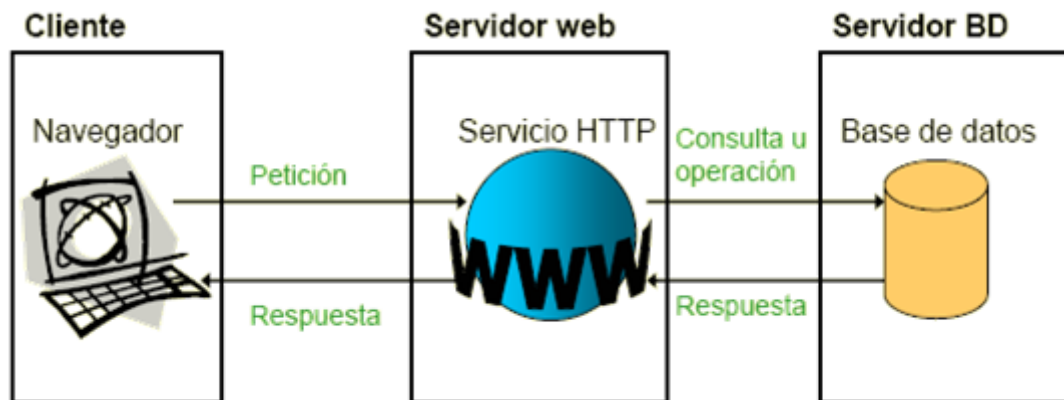


Figura 3: Modelado de tres capas

Capa de datos

La capa de datos es la que se encarga en el almacenamiento y gestión que posea el sistema que puede ser tanto relacional como no relacional y que mantiene una conexión constante con la capa lógica [12].

Capa de presentación

Conocida como capa de diseño es donde se visualizan todas las interfaces que el usuario podrá hacer uso del sistema. En esta capa se obtiene todas las peticiones que el usuario hará y luego enviar los datos adquiridos a la siguiente capa lógica [10].

Capa lógica

Esta capa se encarga de abarcar toda la lógica en cada fase del sistema, la cual permite una interacción entre la capa de datos y presentación para que ambas se puedan comunicar internamente cuando el usuario vaya a realizar una petición [13].

2.2.2. PostgreSQL

Se considera altamente calificado por su estabilidad, funcionalidad, robustez y facilidad de administración e implementación. Además, utiliza un sistema cliente-

servidor que utiliza subprocesos para procesar correctamente las consultas en la base de datos [14].

PostgreSQL mantiene una relación entre el volumen de datos y el tiempo de respuesta, es decir, aumenta proporcionalmente, mientras que en las bases de datos columnares Cassandra y MongoDB, el tiempo de ejecución no difiere significativamente entre el volumen de datos y el tiempo [15].

Como se observa en la tabla 1 se detallan las características de Postgresql

Tabla 1: Características de Postgresql

Servicios de Postgresql	
Descripción	RDBMS de código abierto muy utilizado [16]
Modelo de base de datos principal	SGBD relacional
Modelos de bases de datos secundarias	Almacén de documentos Spatial DBMS
Sistemas operativos de servidor	FreeBSD HP-UX Linux NetBSD OpenBSD OS X Solaris Unix Windows
API y otros métodos de acceso	ADO.NET JDBC biblioteca C nativa API de transmisión ODBC

2.2.3. Herramientas Case

Estas herramientas se utilizan en la ingeniería de software porque sirven de soporte a los SI, cada vez más enfocadas al concepto de colaboración, como herramientas para mejorar el proceso de desarrollo, integradas en el entorno WEB y móvil, con todas sus implicaciones a nivel técnico [18].

Se puede decir que las herramientas CASE son un conjunto de reglas, utilidades, técnicas y ayudas que brindan una ayuda a los diseñadores y desarrolladores de sistemas aplicados en todos los pasos del ciclo de vida del desarrollo de un software.

Para el desarrollo de la aplicación es necesario usar las siguientes herramientas.

2.2.3.1. Enterprise Architect

Enterprise Architect es una herramienta que tiene como objetivo de diseñar diagramas en UML en la etapa de diseño con los requisitos del sistema desde la recolección [19].

EA se define como la organización básica de un sistema, que incluye sus componentes, las relaciones entre los componentes y el entorno, y los principios de su diseño y desarrollo [20].

2.2.3.2. Balsamiq Mockups

Es de uso fácil y rápido para trabajar con visualizaciones, demostraciones y pruebas la cual permite la modificación de diseños de cómo se vería el software que se está trabajando para poder llegar al producto del proyecto final. Tiene una interfaz accesible que se puede usar en Linux también en Mac OS X [21].

2.2.3.3. StarUML

Esta herramienta permite trabajar en el moldeamiento de los proyectos debido a que es orientada al desarrollo de diagramas UML como son los de clase, de estado, de secuencia, de caso y de relación entidad. Es un software gratis que se puede trabajar con C#, java y C++, también permite producir plugin y genera información en diferentes [22].

2.2.3.4. Bizagi Modeler

Es una aplicación que le permite modelar, documentar, simular, publicar y compartir visualmente procesos de negocios de acuerdo con estándares de la industria como BPMN [23].

Bizagi se caracteriza por ser un modelador gratuito de procesos de negocios amigable con el usuario. Esta herramienta permite el desarrollo de diagramas de actividades, diagramas de flujo y diagramas de actividades que pueda tener un servicio en la nube para poder acceder cuando se lo requiera [24].

2.2.4. Visual Studio Code

Dada la creciente popularidad de los IDE ligeros, así como nuestra observación de las limitaciones de soporte de colaboración en tiempo real para ellos, propusimos, diseñamos e implementamos un novedoso entorno de programación colaborativo para Visual Studio Code, uno de los más populares y ampliamente IDE ligeros usados [25]. El entorno propuesto cumplió con tres objetivos de diseño:

- preservar la plena compatibilidad con el IDE ligero existente
- apoyar la programación colaborativa en tiempo real con conciencia intuitiva
- lograr una colaboración en tiempo real sin restricciones y flexible para
- programadores con roles iguales.

Tiene soporte incorporado para la programación lenguajes JavaScript y TypeScript y además habilita el soporte para otros lenguajes a través de un rico ecosistema de extensiones. VS Code usa un sistema de carpetas o espacio de trabajo para interactuar con un proyecto y un sistema de documentos para manejar los archivos de código fuente en el proyecto [26].

2.2.5. JavaScript

JavaScript es el lenguaje que proporcionó un sitio web dinámico que se comunica activamente con los usuarios. JavaScript se utiliza en las aplicaciones web actuales como lenguaje de secuencias de comandos del cliente y del lado del servidor [27].

JavaScript es relativamente fácil de utilizar y se la utiliza para realizar:

- Juegos basados en el navegador web, incluidos los juegos 2D y 3D

- Desarrollo de aplicaciones para móviles
- Presentaciones web [28]

2.2.6. Node.Js

Node es un entorno de ejecución de JavaScript asíncrono basado en eventos diseñado para crear aplicaciones de red escalables, como la característica principal que tenemos Node.js, le permite crear una aplicación web asíncrona y escalable basada en eventos. Node.js permanece inactivo hasta que una conexión activa una llamada de servicio, todo lo posible gracias al uso de JavaScript del lado del servidor [29].

Node.js (más específicamente, JavaScript) no es un lenguaje complicado. sigue la base principios que siguen lenguajes de secuencias de comandos similares (como Ruby, Python y PHP), pero con un toque (como todos lo hacen). Node.js es lo suficientemente simple para que cualquier desarrollador lo recoja y comience a codificar en poco tiempo y, sin embargo, es lo suficientemente potente como para lograr casi cualquier cosa los desarrolladores pueden decidir [30].

Node.js es una opción particularmente sólida para desarrollar API, como está a punto de ver.

2.2.7. React Native

Permite crear aplicaciones nativas mediante JavaScript ya que permite ingresar a las APIs y Views de cada dispositivo, de esta forma se inicia con crear los elementos visuales para que corran directamente en las plataformas móviles [31].

React Native es React, pero en el que JavaScript se procesa en un dispositivo móvil en el proceso de fondo en un formato comprensible para el sistema operativo del dispositivo móvil.

Esto le permite utilizar las interfaces nativas del sistema operativo del dispositivo móvil, como la cámara y la gestión de la memoria. Además, se permite el uso de componentes nativos de la interfaz de usuario, como el teclado y los botones [32].

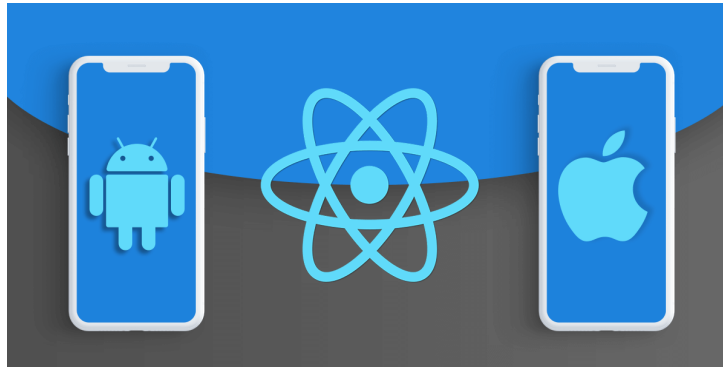


Figura 4 : Representación React Native [33]

2.2.8. Metodología de desarrollo de software

Es un enfoque estructurado para el desarrollo de software que incluye modelos de sistemas, notaciones, reglas, consejos de diseño y guías de procesos. Con la ayuda de directrices, el proceso de desarrollo de software se vuelve más organizado y ágil, satisfaciendo así las necesidades y requisitos deseados de los usuarios [34].

Estas metodologías también se denominan como un conjunto de métodos organizados que ayudan a corregir los problemas de software existentes, siempre y cuando se utilice la metodología correcta en los proyectos que se esté trabajando [34].

2.2.8.1. Metodologías ágiles

Esta metodología permite entregar al cliente un producto de software de calidad con menos tiempo y costo, debido a que es una metodología flexible y moldeable lo que permite que el proyecto se adapte a las circunstancias, además tiene la ventaja de ser una metodología predecible en la respuesta y demanda del cliente [35].

Dentro de las características que destacan esta metodología se encuentra que es iterativo lo cual permite el desarrollo en menos tiempo y costo. Por lo tanto, se enfoca menos en la documentación y se enfoca más en la codificación del proceso de su desarrollo [36].

Una de las diferencias que posee con la metodología tradicional es que la metodología ágil está abierta a que sucedan errores debido a que durante su desarrollo permite que los errores que sucedan se vayan encontrando y pueda ser solucionado en el transcurso que se avanza el proyecto [37].

2.2.8.2. Metodologías híbridas

Hoy en día es importante de forma rápida y ágil se lleve a cabo el desarrollo de software pero también hay que tomar en cuenta que si se requiere modificar o actualizar es indispensable que haiga una guía donde se refleje lo que se haya avanzado, como consecuencia nace la metodología híbrida que es una combinación en sus procesos entre las metodologías tradicionales con las ágiles [38].

En la ingeniería de software se ha comenzado a desarrollar metodologías híbridas como una innovación en la mejora de los procedimientos del desarrollo de software, creando así una combinación entre dos o más metodologías ágiles dando como resultado un método más fuerte y flexible [39].

2.2.8.3. Metodología SWIRL

En este proyecto la metodología que se escogió para la elaboración de la ampliación móvil es la SWIRL debido a que posee un marco de desarrollo con una guía que facilitará en el desarrollo de aplicaciones móviles.

La metodología SWIRL es un modelo que permite la iteración considerando cinco criterios tales como: costo, tiempo, calidad, volumen y comunicación que El modelo se hace de acuerdo a las necesidades de cada cliente, es decir Adaptar siempre los cambios necesarios para cada proyecto y cuando se considera el alcance.

Cada actividad en sus fases aporta agilidad al proceso de desarrollo, además de mejorar la comunicación del equipo de desarrollo con los clientes, que está incluida en la mayoría de los procesos. Este último beneficia tanto al producto como al proceso en el proyecto. En términos de producción, se garantiza que una aplicación web, incluido el cliente, cumplirá con todos los requisitos que ha establecido al inicio, así como también cumplirá con los requisitos de los proyectos orientados a la web [40].

Esta metodología posee cinco fases para el ciclo de vida sobre el proyecto de software, a continuación, en la figura 5 se detalla las fases de la metodología:

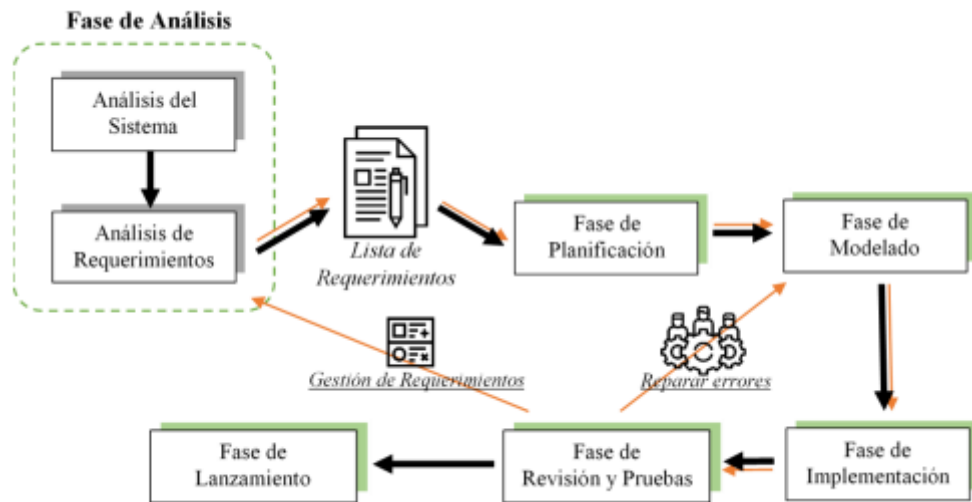


Figura 5 :Fases del ciclo de vida de la metodología Swirl [40]

2.3. OBJETIVOS DEL PROTOTIPO

2.3.1. OBJETIVO GENERAL

Desarrollar una aplicación móvil en React Native para la gestión de pedidos con servicio de delivery de CRUZSALUD utilizando la metodología Swirl, JavaScript y PostgreSQL.

2.3.2. OBJETIVOS ESPECÍFICOS

- Recolectar los requerimientos para el desarrollo de la aplicación mediante la implementación de historias de usuario.
- Investigar mediante fuentes bibliográficas las tecnologías a implementarse en la aplicación.
- Diseñar prototipos mediante la herramienta Balsamiq Mockups para las interfaces correspondientes de la aplicación.
- Evaluar la aplicación móvil aplicando la normativa ISO/IEC9126 para garantizar el funcionamiento y desarrollo de un sistema de calidad.

2.4. DISEÑO DEL PROTOTIPO

2.4.1. FASE DE ANÁLISIS

2.4.1.1. Reconocimiento general del sistema

Como es de conocimiento la problemática surgida desde la pandemia del COVID-19, las personas tuvieron la necesidad de optar por nuevas modalidades en base a las compras, siendo el servicio de delivery la opción más utilizada en estos últimos tiempos.

La compra de víveres de primera necesidad a domicilio es un servicio que muy pocos negocios ofertan a sus clientes, sin embargo existe la necesidad que los negocios ofrezcan esta modalidad de compra para que sus clientes tengan la comodidad de conocer los productos y pedirlos a domicilio.

2.4.1.2. Stakeholders

Al mencionar a las partes interesadas o las personas que intervienen dentro de la aplicación hacemos énfasis a los stakeholders los mismos que durante el proceso ayudan a recoger los requerimientos de software, estos son analizados tanto para el éxito o fracaso de proyectos de software, de la misma forma se

busca ir de la mano de una metodología la que ayudará a la búsqueda de datos y así caracterizar y obtener los mejores stakeholders. [41] [42]

Para el desarrollo de este proyecto, se detalla en la tabla 2 los stakeholders.

Tabla 2:Matriz de interesados

MATRIZ GENERAL DE STAKEHOLDERS			
Nombre	Roles	Profesión	Responsabilidad
Ing. Joffre Cartuche Calva	Docente/ Gerente de proyecto	Ingeniero de Sistemas	Limitaciones del sistema y revisión de entregables.
Jonathan Brito Aguilar	Desarrollador	Estudiante	Planificación. Análisis. Diseño. Desarrollo. Implementación. Pruebas y Lanzamiento.
Ing. Alex Andres Ramon Brito	Usuario/Cliente	Ingeniero de Sistemas	Requerimientos

2.4.1.3. Alcance del proyecto

En base a la creación de la aplicación se busca permitir al cliente ahorrar tiempo al momento de realizar los pedidos de sus productos dentro del negocio CRUZ SALUD, mediante la visualización de la información detallada de los productos a adquirir de tal manera que el proceso de compra sea eficiente y seguro.

De acuerdo con la tabla 3 se detalla las funcionalidades del sistema.

Tabla 3: Funcionalidades del sistema

Funcionalidades de PIDEAHORA
Permite la autenticación mediante email para que el usuario pueda registrar e iniciar sesión.

El usuario podrá agregar los datos de facturación en caso de solicitar la factura en su pedido.
El usuario podrá conocer el perfil del negocio como su dirección, ubicación, productos, teléfono, horario de atención, etc.
El usuario tendrá la opción de buscar el producto que necesite mediante el buscador que se encuentra en la pantalla principal.
El usuario antes de agregar un producto al pedido se podrá escribir alguna indicación para que el negocio lo tenga presente.
Al abrir una categoría se puede visualizar diferentes secciones como: enlatados, granos, frutas, verduras, bebidas, etc.
El usuario podrá editar su perfil en caso lo requiera.
El usuario podrá visualizar sus pedidos realizados.
El usuario podrá conocer el subtotal de la compra y el costo de envió.
El usuario podrá notificar la denominación del billete con el que cancelará el pedido en caso de ser necesario para que el repartidor pueda llevar el cambio correspondiente.

Las limitaciones del sistema se detallan en la tabla 4.

Tabla 4: Limitaciones del sistema

Limitaciones de PIDEAHORA
La aplicación únicamente aceptará pagos en efectivo
El usuario no podrá escribir al repartidos desde la aplicación
El usuario no puede realizar pagos con billetes de denominaciones de 50 y 100 dólares.
El usuario no podrá realizar pedidos desde otra ciudad.

2.4.1.4. Análisis de la factibilidad del sistema

La factibilidad del sistema son todos los recursos que se tiene a disposición para poder llevar a cabo los objetivos establecidos, por tal razón se deben analizar tanto la factibilidad operativa, técnica y económica [43].

2.4.1.4.1. Factibilidad Operativa

La factibilidad operativa abarca todos los recursos que el proyecto necesita tales como recursos humanos o tecnológicos, la cual se debe garantizar la viabilidad de estos para poder llevar a cabo el software [44].

De acuerdo con la tabla 5 se detalla la factibilidad operativa del proyecto.

Tabla 5: Factibilidad Operativa

Factibilidad Operativa		
No.	Actividad	Prioridad
1	Se tiene los componentes informáticos precisos para el desarrollo de la aplicación	Alta
2	Poseer conexión a internet para su buen funcionamiento	Alta
3	Tener conocimientos técnicos sobre el desarrollo de aplicaciones móviles	Alta
4	Compra de las cuentas de desarrollador de las principales tiendas de Play Store y App Store	Alta

2.4.1.4.2. Factibilidad Técnica

En esta fase de este proyecto se usarán herramientas tecnológicas que permitirán el desarrollo de la aplicación que funcionará en base a dispositivos como Android, espacio en la nube etc [45].

La factibilidad técnica del proyecto se encuentra detallada en la tabla 6.

Tabla 6: Factibilidad Técnica

Hardware		
Cantidad	Descripción	Costo (\$)
1	Laptop Intel Core i7 6ta generación, 8 Gb memoria RAM, 512 SSD, 1 TB HDD	\$1100.00

1	Teléfono móvil Iphone 11	\$600
Software		
Cantidad	Descripción	Costo (\$)
1	Node.Js	\$0.00
1	Visual Code Studio	\$0.00
1	Suite Microsoft Office	\$0.00
1	Bizagi Modeler	\$0.00
1	StartUML	\$0.00
1	Balsamiq Mockups	\$0.00
1	React Native	\$0.00
1	Appetize.io	\$0.00
Total de Costos		\$1700.00

2.4.1.4.3. Factibilidad Económica

En esta fase se enfoca todos los recursos económicos con los que posee el equipo de trabajo para lograr el desarrollo de la aplicación y que pueda funcionar.

Según la tabla 7 se detalla la factibilidad económica del proyecto [46]

Tabla 7: Factibilidad Económica

Tipo Recurso	Tipo Elemento	Descripción	Unidad Medida	Cantidad	P. Unitario	Costo
HUMANO	Ing. Sistemas	Jonathan Brito	Horas	560	7.50	4,200.00
TECNOLOGICO	Software	Node.Js	Unidad	1	0	0
TECNOLOGICO	Software	React Native	Unidad	1	0	0
TECNOLOGICO	Software	Bizagi Modeler	Unidad	1	0	0
TECNOLOGICO	Software	StartUML	Unidad	1	0	0
TECNOLOGICO	Software	Balsamiq Mockups	Unidad	1	0	0
TECNOLOGICO	Software	Visual Studio Code	Unidad	1	0	0
TECNOLOGICO	Software	Appetize.io	Unidad	1	0	0
TECNOLOGICO	Hardware	Laptop Core I7, 512 SSD, 8 RAM	Unidad	1	1,100.00	1,100.00

TECNOLOGICO	Hardware	Dispositivo IOS	Unidad	1	600.00	600.00
TECNOLOGICO	Servicios	App Store	Unidad	1	100.00	100.00
TECNOLOGICO	Servicios	Play Store	Unidad	1	25.00	25.00
TECNOLOGICO	Servicios	Internet	Meses	3	30.00	90.00
MATERIALES	Bienes	Escritorio	Unidad	1	80.00	80.00
MATERIALES	Bienes	Sillas	Unidad	1	60.00	60.00
MATERIALES	Bienes	Oficina	Meses	3	100.00	300.00
COSTO					\$	6,555.00
IMPREVISTOS					10%	655.50
PRESUPUESTO TOTAL					\$	7,210.00

2.4.1.5. Requerimientos funcionales

Estos requerimientos son los que permiten el comportamiento del software por lo tanto esto influye en si la solución es aceptada o rechazada por parte de los usuarios finales, se corre los riesgos de que el proyecto falle en caso de que estos requerimientos sean deficientes.

De acuerdo con la tabla 8 se encuentra detallado los requerimientos funcionales del sistema.

Tabla 8: Requerimientos Funcionales

REQUERIMIENTOS FUNCIONALES	
Código	Descripción
RF – 01	Permite el ingreso al sistema mediante autenticación a los usuarios
RF – 02	Permite al usuario registrarse a través de su correo electrónico
RF – 03	Administración de datos de facturación
RF – 04	El usuario podrá visualizar el perfil del negocio
RF – 05	El usuario podrá observar los productos disponibles organizados por categorías
RF – 06	Permite al dueño del negocio la gestión de los productos (Editar, agregar y eliminar)
RF – 07	El usuario podrá realizar su pedido seleccionando los productos y la cantidad deseada.

2.4.1.6. Requerimientos no funcionales

Estos requerimientos son todos aquellos requisitos indirectos del sistema pero que no son importantes ni indispensables y que son utilizados para juzgar las funciones del sistema [47].

Tabla 9: Requerimientos no funcionales

REQUERIMIENTOS NO FUNCIONALES	
Código	Descripción
RNF – 01	La aplicación debe ser intuitiva para el usuario.
RNF – 02	La aplicación de ajustar a las diversas dimensiones de los dispositivos móviles
RNF – 03	El tiempo de carga y búsqueda deben ser rápidos

2.4.2. FASE DE PLANIFICACIÓN

2.4.2.1. Historias de usuario

Las historias de usuario sirven para mostrar la presentación de las funciones del software de forma más explícita y general e informal, tiene como objetivo proporcionar valor al cliente de acuerdo a las funciones [48].

Tabla 10: Historia de usuario Inicio de sesión

Historia de usuario	
Numero: 1	Usuario: Usuario Final
Nombre de la historia: Inicio de sesión en la aplicación	
Prioridad en negocio: Alta	Riesgo en desarrollo: Baja
Requerimiento funcional: RF- 01	Iteración asignada: 1
Programador responsable: Jonathan Brito	
Descripción: La aplicación móvil debe contar con un inicio de sesión a través de su correo y contraseña para los clientes pueda ingresar y poder realizar sus pedidos	
Observaciones: Mostrar un mensaje de error cuando los datos ingresados al inicio de sesión no se encuentren en la base de datos. La aplicación permitirá el ingreso a los usuarios que quieran realizar un pedido del negocio CRUZ SALUD y podrá visualizar todos los productos disponibles.	

Tabla 11: Historia de usuario registro de usuario

Historia de usuario	
Numero: 2	Usuario: Usuario final
Nombre de la historia: Registro de usuario	
Prioridad en negocio: Alta	Riesgo en desarrollo: Baja
Requerimiento funcional: RF-02	Iteración asignada: 1
Programador responsable: Jonathan Brito	
Descripción: En lo que respecta al registro de usuario tiene que ingresar los Nombres y Apellidos de la persona, además del correo electrónico su respectiva contraseña, también su número telefónico, edad y sexo.	
Observaciones: Los usuarios que ingresen datos erróneos se presentara un mensaje para que pueda corregir los datos. La aplicación permitirá recuperar la contraseña a través del correo electrónico registrado.	

Tabla 12: Historia de usuario Administración de datos de Facturación

Historia de usuario	
Numero: 3	Usuario: Usuario Final
Nombre de la historia: Administración de datos de facturación	
Prioridad en negocio: Alta	Riesgo en desarrollo: Baja
Requerimiento funcional: RF-03	Iteración asignada: 1
Programador responsable: Jonathan Brito	
Descripción: El usuario tiene la opción de solicitar una factura al momento de solicitar el pedido para ello debe registrar sus datos como cedula o RUC, nombres y apellidos completos, dirección, correo electrónico y teléfono.	
Observaciones: La aplicación permitirá realizar cualquier cambio una vez que haya registrado los datos.	

Tabla 13: Historia de usuario perfil del negocio

Historia de usuario	
Numero: 4	Usuario: Usuario Final
Nombre de la historia: Perfil de Negocio	
Prioridad en negocio: Alta	Riesgo en desarrollo: Baja
Requerimiento funcional: RF-04	Iteración asignada: 1
Programador responsable: Jonathan Brito	
Descripción: El empleador podrá visualizar los datos relevantes del negocio tales como: nombre, dirección, horario de atención, teléfono, servicio que oferta.	
Observaciones: La aplicación permitirá visualizar los datos generales del negocio.	

Tabla 14: Historia de usuario Visualización de los productos

Historia de usuario	
Numero: 5	Usuario: Usuario Final
Nombre de la historia: Visualización de los puestos de trabajo	
Prioridad en negocio: Alta	Riesgo en desarrollo: Baja
Requerimiento funcional: RF-05	Iteración asignada: 1
Programador responsable: Jonathan Brito	
Descripción: En el apartado de productos el cliente podrá visualizar toda la variedad de productos a través de las categorías como son: enlatados, granos, frutas, verduras, gaseosas, etc.	
Observaciones: Se visualizará toda la información del producto seleccionado, a demás existe un buscador de productos.	

Tabla 15: Historia de usuario Gestión de productos

Historia de usuario	
Numero: 6	Usuario: Dueño del negocio
Nombre de la historia: Gestión de puestos de trabajo.	

Prioridad en negocio: Alta	Riesgo en desarrollo: Baja
Requerimiento funcional: RF-06	Iteración asignada: 1
Programador responsable: Jonathan Brito	
Descripción: Como dueño tendrá habilitado todas las funciones como agregar, editar y eliminar los productos de cada categoría.	
Observaciones: La aplicación permitirá la edición de información y visualizar los productos.	

Tabla 16: Historia de usuario Gestión de Pedidos

Historia de usuario	
Numero: 7	Usuario: Usuario Final
Nombre de la historia: Gestión de pedidos	
Prioridad en negocio: Alta	Riesgo en desarrollo: Baja
Requerimiento funcional: RF-07	Iteración asignada: 1
Programador responsable: Jonathan Brito	
Descripción: En esta sección el usuario podrá agregar productos que necesita y conocerá el subtotal a pagar, así como el listado de los productos agregados. Luego deberá seleccionar la dirección de la entrega, una vez registrada la dirección el usuario podrá conocer el valor total a pagar.	
Observaciones: El usuario deberá seleccionar el método de pago, por el momento solo estará disponible la opción de efectivo; en caso de que el usuario necesite pagar con un billete puede escribir la denominación del mismo para que el repartidor lleve el cambio necesario y así evitar contratiempos.	

2.4.2.2. Gestión de cronograma

La gestión de cronograma del proyecto es importante para conocer las actividades que se sigue en la gestión del proyecto, para lo cual se detallan las

actividades y el tiempo estimado para su ejecución además de definir el tiempo para la presentación del proyecto, lo cual se la visualiza en la figura 6.

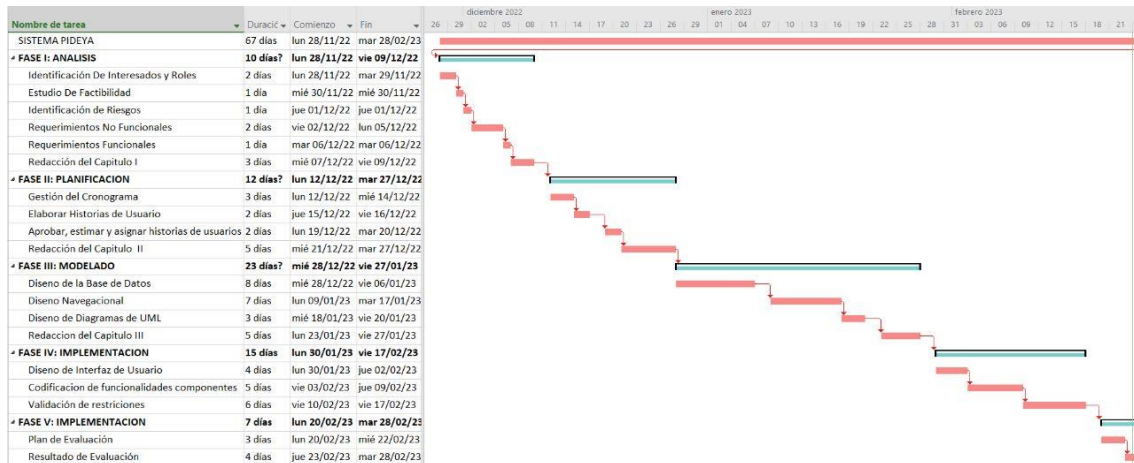


Figura 6: Gestión de cronograma

2.4.3. FASE DE MODELADO

Dentro de esta fase en la metodología SWIRL se encuentra el diseño de los diagramas conceptuales, navegaciones y UML que el software incorporará que permitirán tener una vista previa de la aplicación a desarrollar.

2.4.3.1. Diseño de base de datos no relacional

De acuerdo con la figura 6 se detalla el diseño de base de datos.

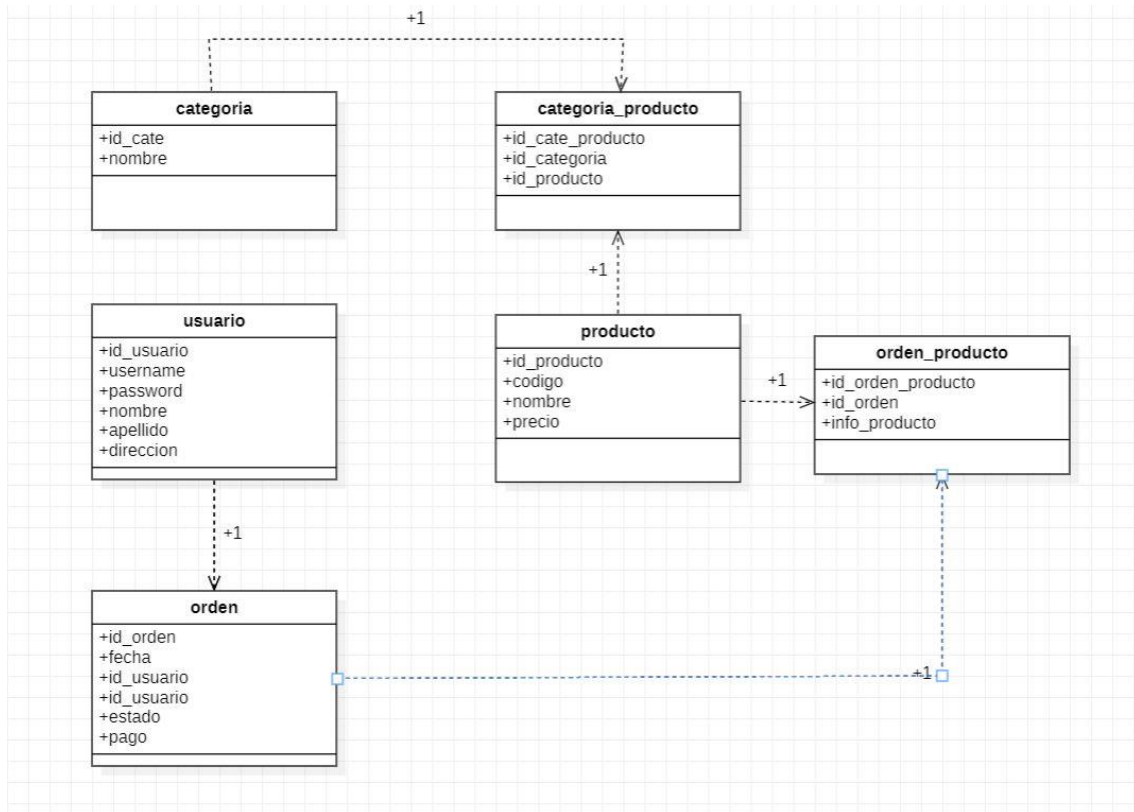


Figura 7: Diseño de base de datos no relacional

2.4.3.2. Diseño navegacional

Dentro de la metodología SWIRL el diseño navegacional permite desarrollar una topología que detalla todas las funcionalidades que el usuario podrá acceder con las tareas unificadas que contará la aplicación, en la figura 8 se detalla la estructura del diseño.

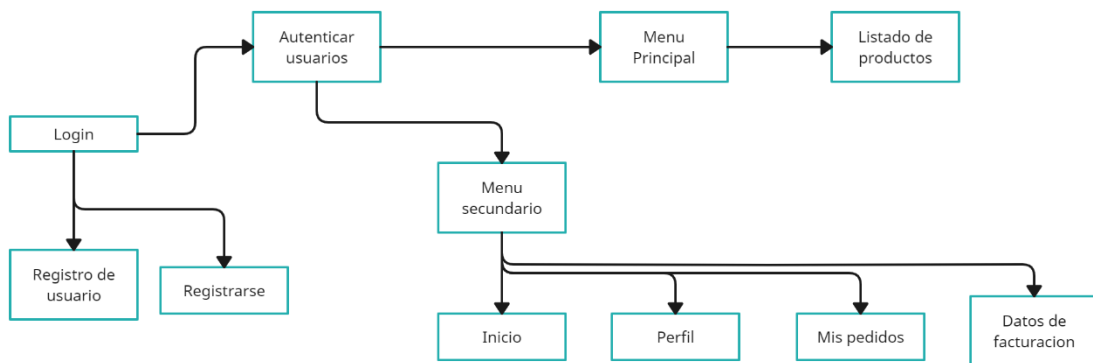


Figura 8: Diseño navegacional

En la figura 8 se muestran las diferentes funcionalidades de las aplicaciones a las cuales el cliente final tendrá acceso, para lo cual en primer lugar deberá autenticarse o registrarse para poder disfrutar de todas las opciones de la aplicación.

2.4.3.3. Diseño de interfaces

Para el desarrollo de interfaces se utilizó la herramienta Balsamiq Mockups la cual nos permite crear modelos similares a cómo será la representación final de la aplicación móvil en los dispositivos móviles, a continuación, se mostrarán los prototipos diseñados.

Módulo 1 – Menú principal



Figura 9 Diseño: Menú principal

La figura 9 nos indica que todo usuario al ingresar a la aplicación deberá ingresar su correo electrónico y su contraseña, en caso de no tener una cuenta deberá registrarse para poder ingresar a la aplicación.

Módulo 2 – Registro de usuarios



Figura 10 Diseño: Registro de usuario

La figura 10 nos indica el proceso para registrar un nuevo usuario por cual se le solicitará información básica de la misma que servirá para que pueda ingresar y hacer uso a la aplicación.

Módulo 3 – Menú de navegación de la aplicación

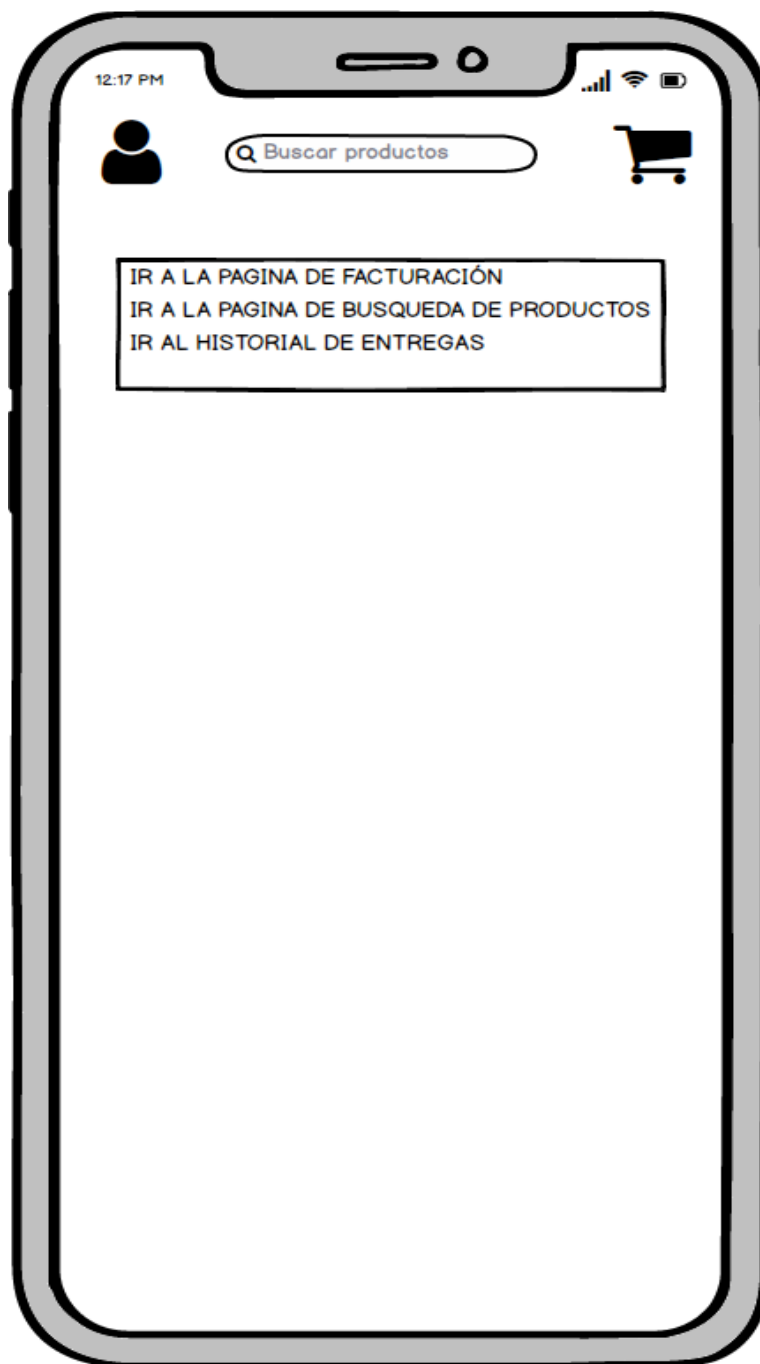
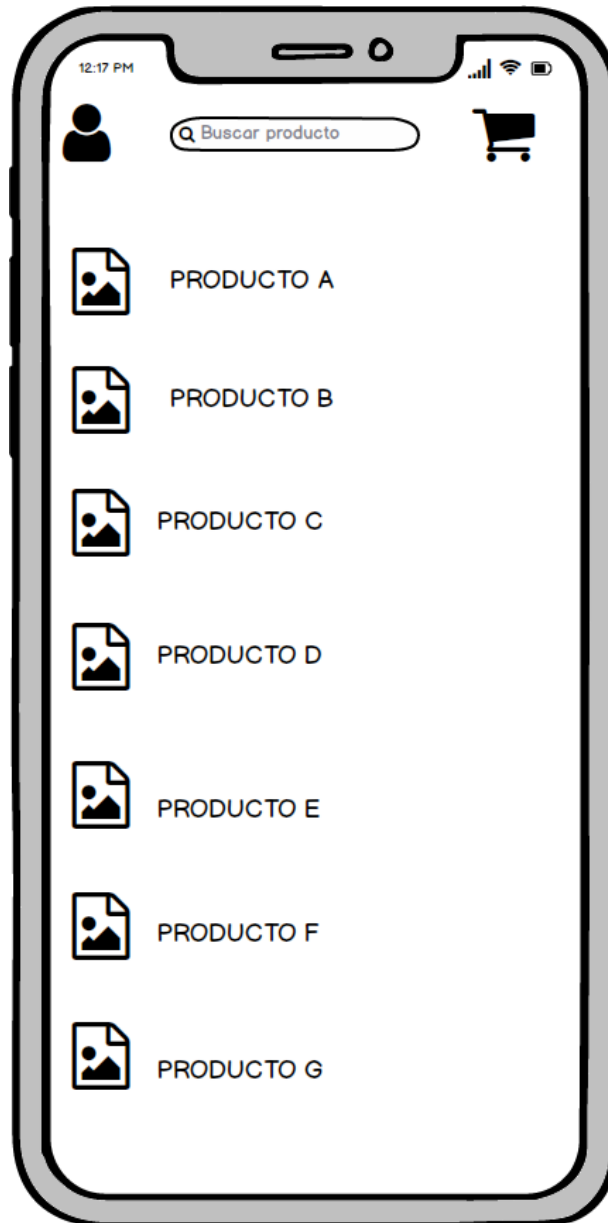


Figura 11 Diseño: Menú de opciones de la aplicación

La figura 11 nos despliega la ventana principal donde se mostrará todas las opciones disponibles y podrá ir a las diferentes opciones que le brinda la aplicación.

Módulo 4 – Menú de productos

Figura 12 Diseño: Perfil del puesto de trabajo



La figura 12 muestra la interfaz que el usuario visualizará al momento de ver los productos disponibles del negocio para lo cual estará dividido por secciones y por un buscador que permitirá encontrar un producto mas rápido.

Módulo 5 – Registro de pedido

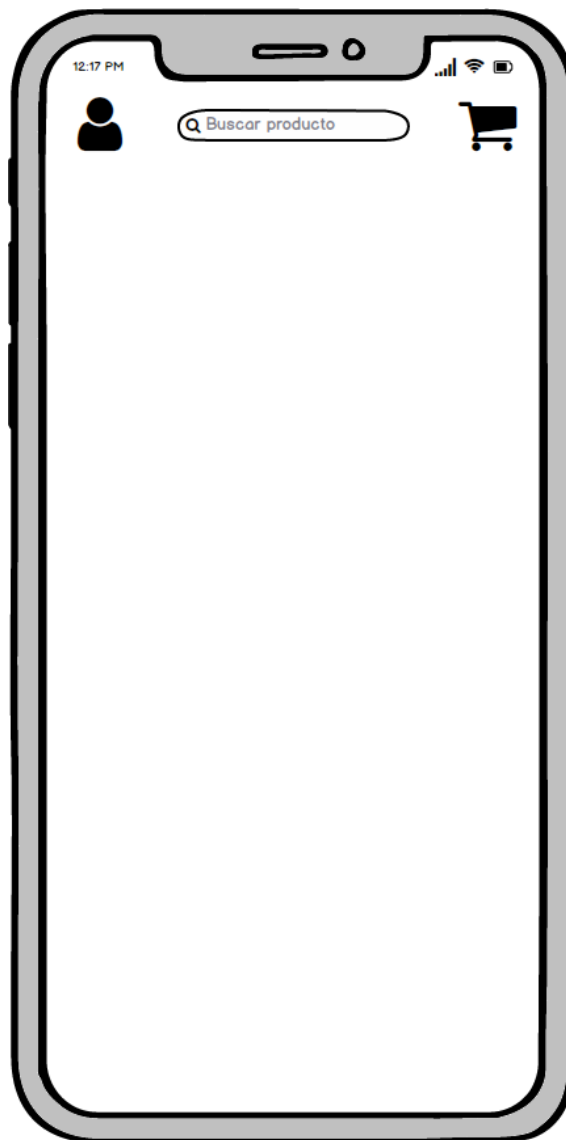


Figura 13 Diseño: Registro de pedido

La ilustración 13 se observa la interfaz que permite al cliente realizar su pedido y observar su valor a pagar con el servicio de entrega a domicilio.

Módulo 6 – Perfil del cliente



Figura 14 Diseño: Registro oferta de trabajo

La figura 14 se observa la interfaz donde el cliente podrá observar sus datos personales, además ingresar datos para facturación y los pedidos que ha realizado.

2.4.3.4. Diseño de diagramas UML

Los diagramas UML también llamado lenguaje de modelado unificado permite diseñar un sistema de software y documentarlo, son de gran utilidad debido a que permiten poseer una mejor vista de cómo será el funcionamiento de la aplicación.

Casos de uso

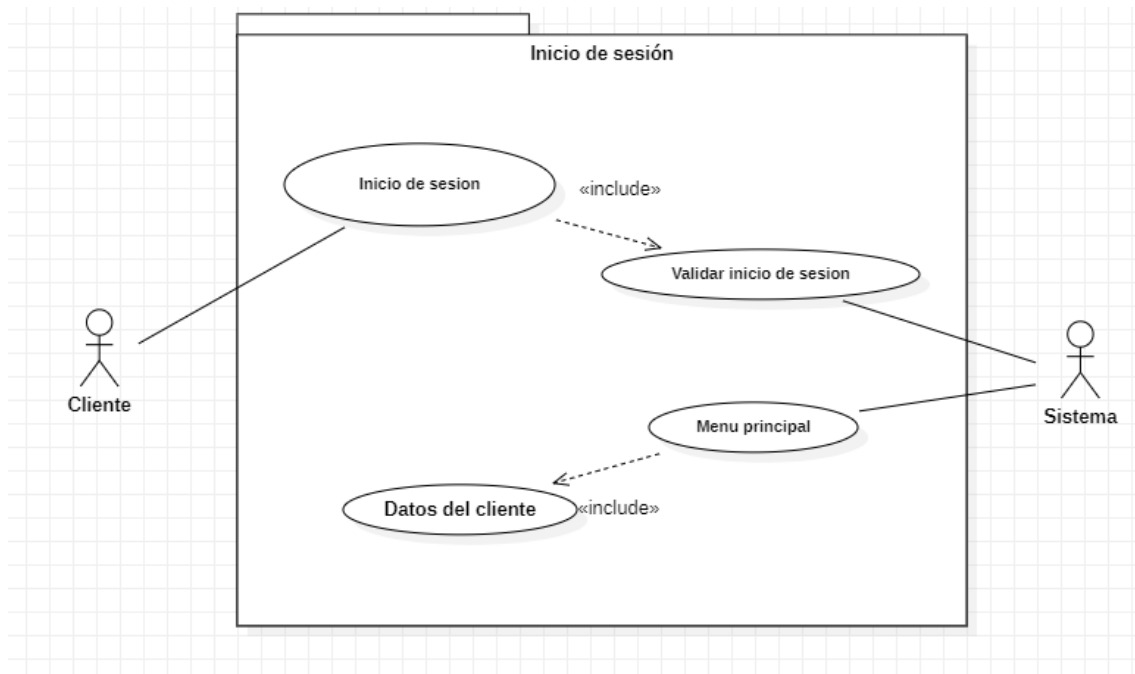


Figura 15: Caso de uso Inicio de sesión

En la figura 15 se muestra que el cliente podrá ingresar a la aplicación mediante la comprobación de sus datos como es el correo y la contraseña, las mismas que el sistema valida de forma interna, por lo tanto, permitirá de forma correcta ingresar para tener acceso a todas las funcionalidades de la aplicación.

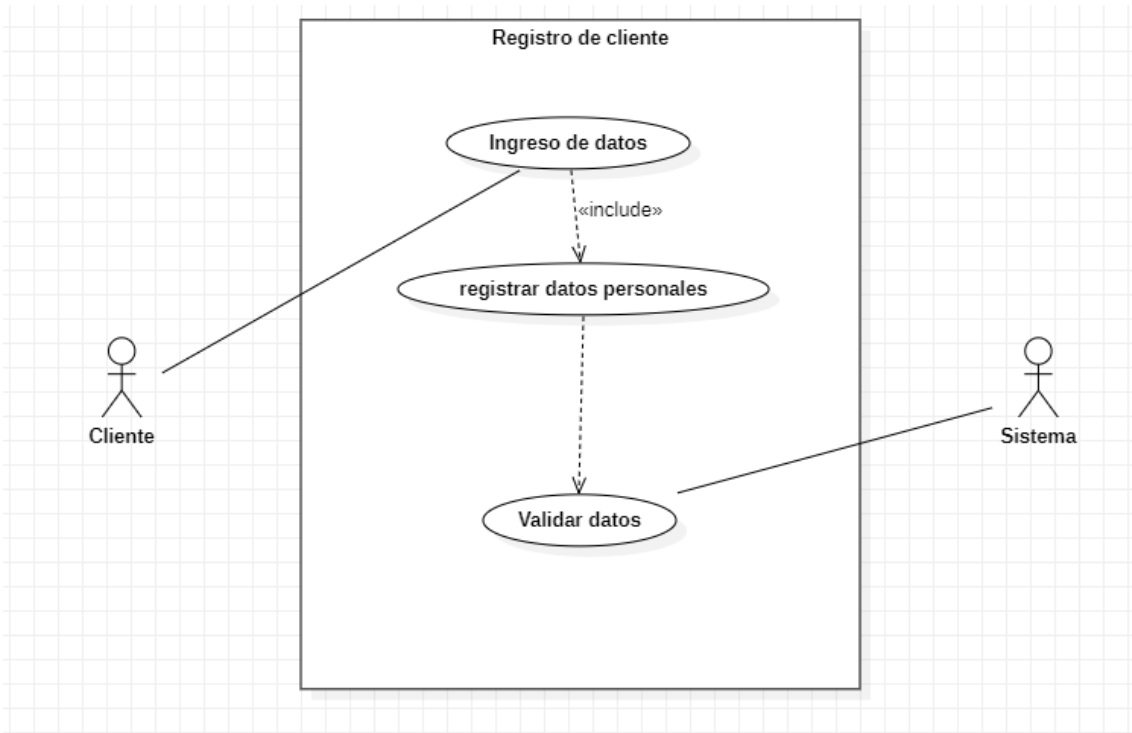


Figura 16: Casos de uso Registro

En la figura 16 se detalla el módulo de registrarse el cual solo los clientes podrán registrarse para poder iniciar sesión en la aplicación y poder visualizar los productos disponibles.

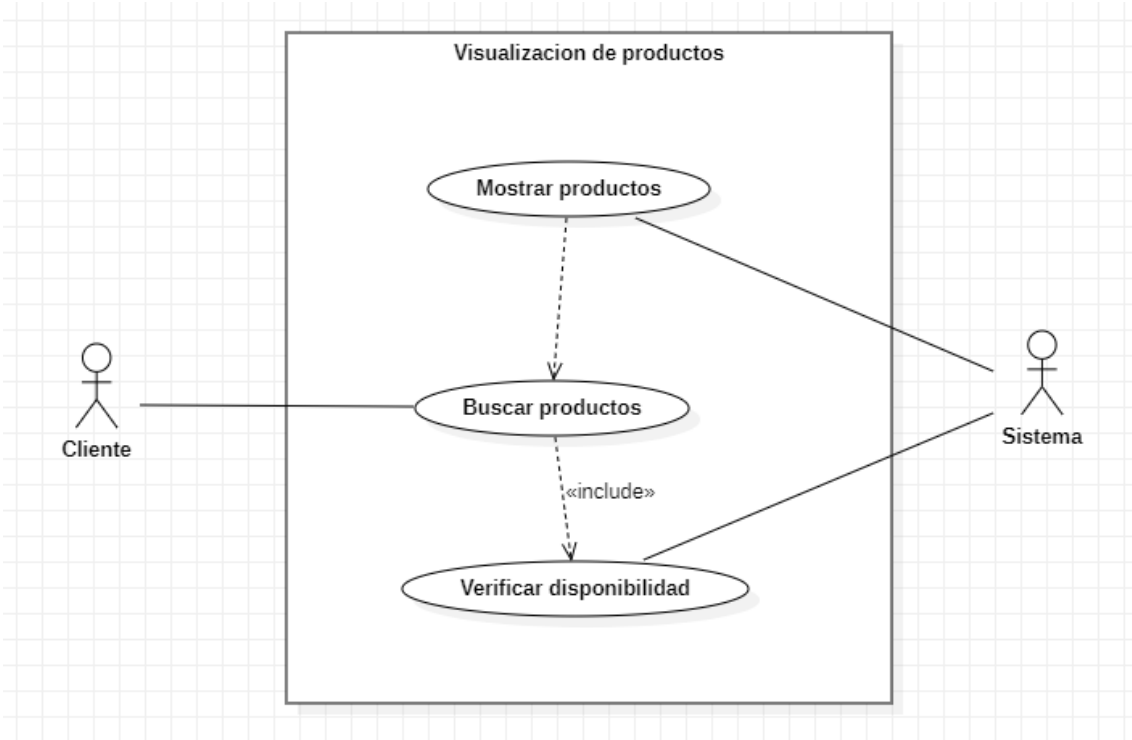


Figura 17: Casos de uso Visualización de productos

En la figura 17 nos muestra de forma general todos los productos que se encuentran disponibles que el usuario podrá observar, también contará con una opción de búsqueda y un filtro de categoría la misma que será verificada por el sistema antes de visualizarla al usuario.

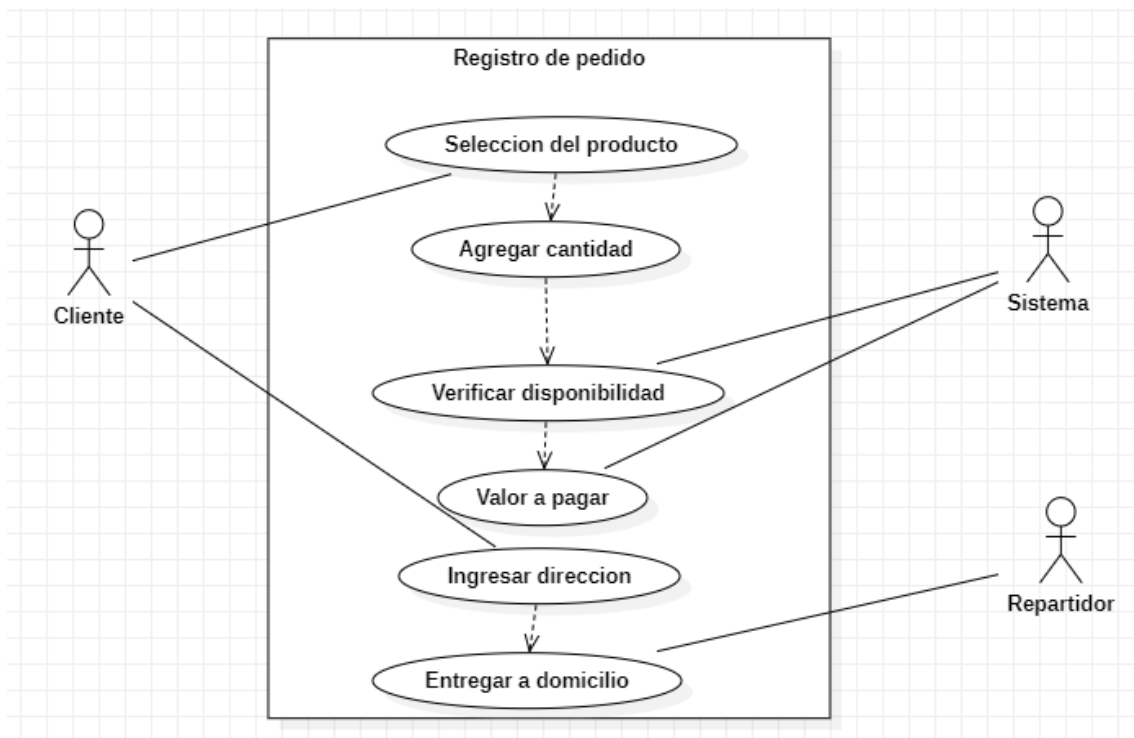


Figura 18: Caso de uso Registro de pedido

En la figura 18 se visualiza las opciones que tendrá el cliente al momento de agregar productos para realizar su pedido, para lo cuales debe ingresar la dirección para que el repartidor haga la entrega.

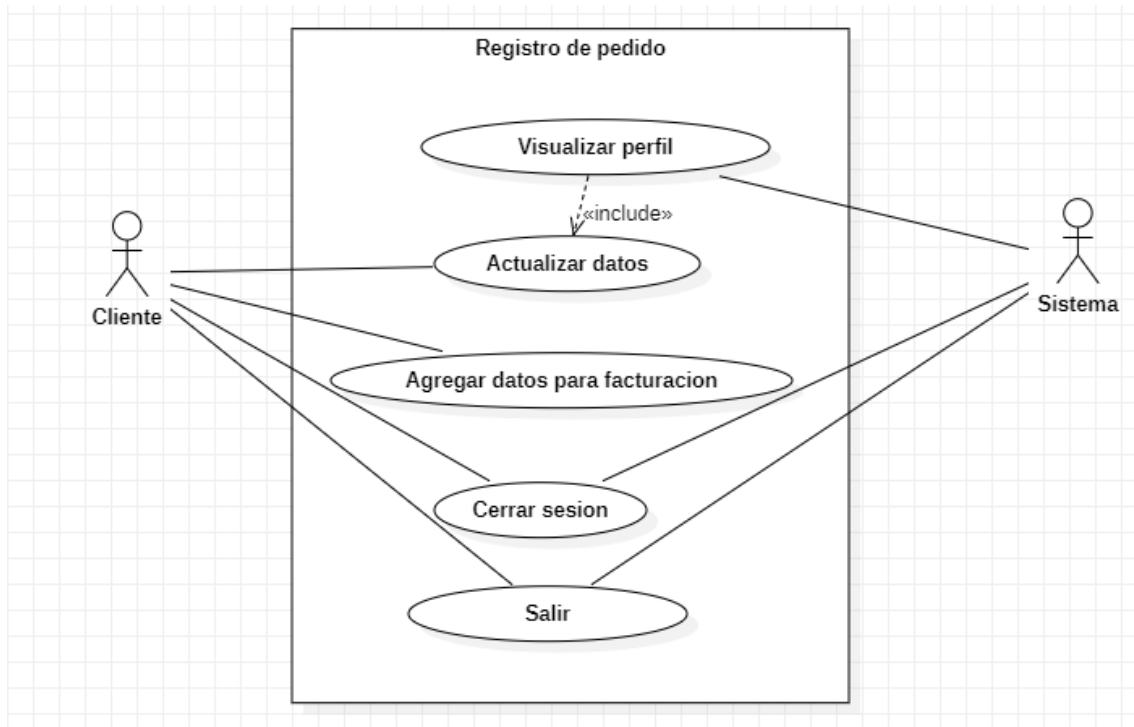


Figura 19: Caso de uso Gestión de perfil

En la figura 19 se detalla la opción de gestionar el perfil del cliente, la cual podrá actualizar sus datos, así como también agregar datos para facturación, cierre de la misma y cierre de sesión.

Actividades

Estos diagramas permiten representar todos los flujos que poseen los procesos dentro del software con el objetivo de analizar cómo se comportan dentro del sistema.

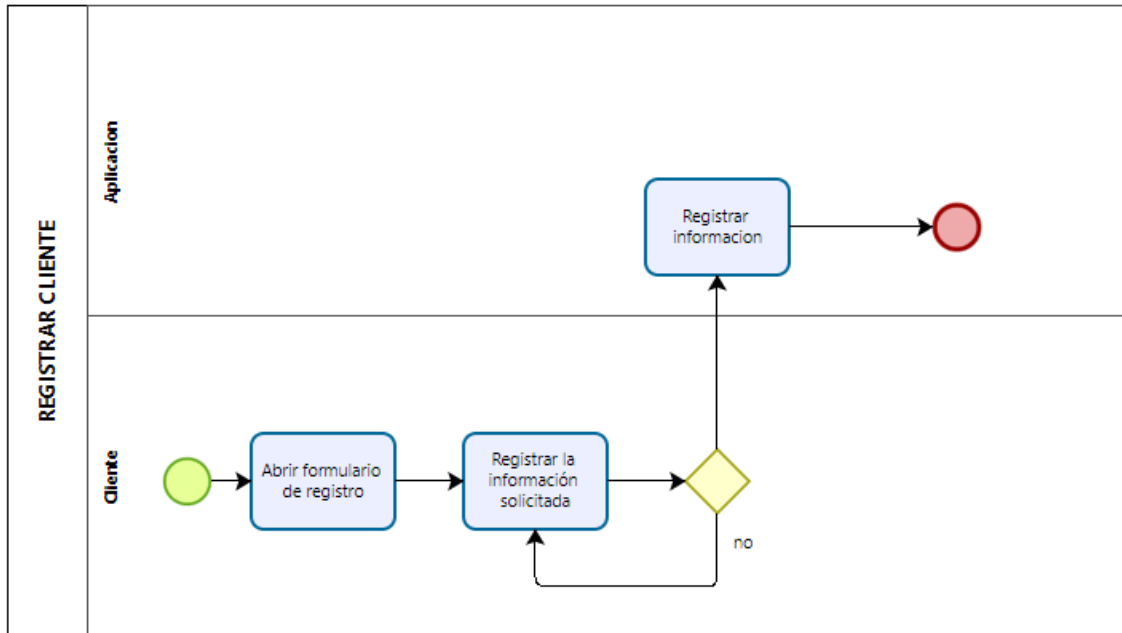


Figura 20: Diagrama de actividades Registro de cliente

La figura 22 representa el desarrollo que se realizará el software al momento que el cliente se registra, la cual abrirá un formulario donde el cliente deberá registrar la información que se pide, por lo tanto, si no llena todos los campos el sistema no podrá proceder con el registro ni ingresar al sistema.

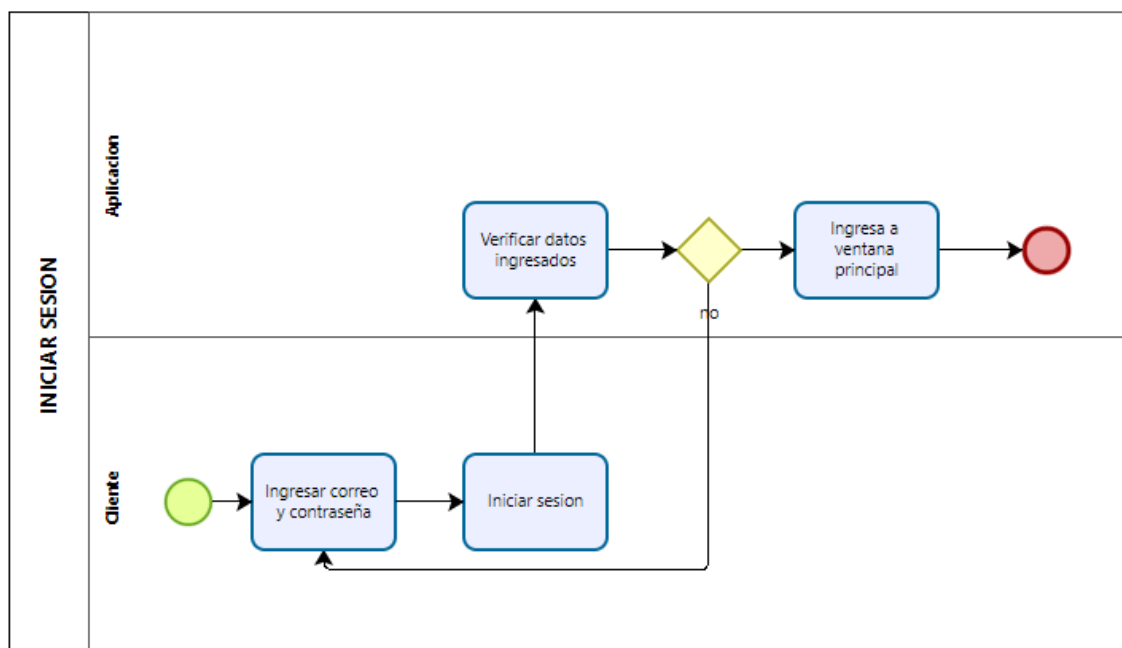


Figura 21: Diagrama de actividades Inicio de Sesión

La figura 23 nos detalla el proceso que se realiza para ingresar a la aplicación por medio del inicio de sesión, donde el sistema pedirá el correo y la contraseña,

las cuales estas serán autenticadas y en caso que el sistema no las valide el cliente no podrá acceder a la aplicación.

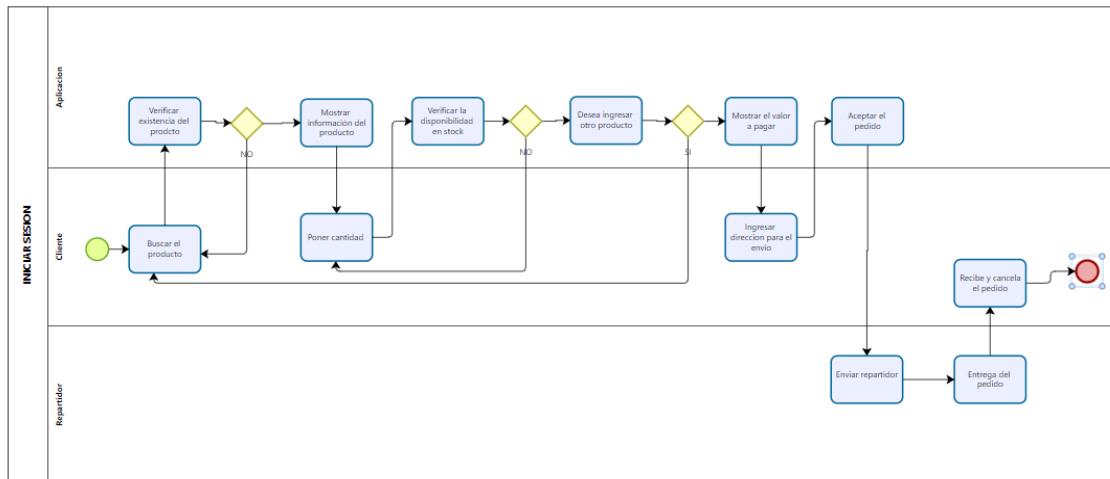


Figura 22: Diagrama de actividades selección de productos

La figura 22 nos muestra el proceso para llevar a cabo el registro de los productos para comprar por parte del cliente, donde en primer lugar deberá iniciar sesión y escoger los productos que desea adquirir para luego proceder al pago correspondiente.

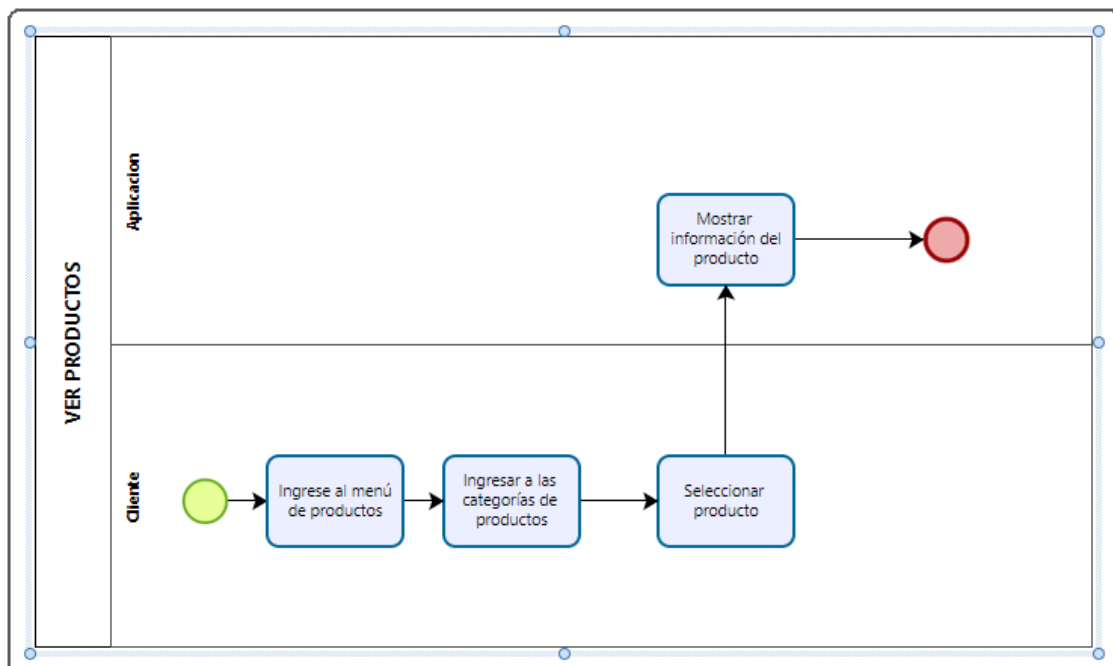


Figura 23: Diagrama de actividades visualizar producto

La figura 23 nos detalla el proceso para buscar los productos disponibles, el usuario deberá acceder a la aplicación y se procederá a visualizar los productos

disponibles y podrá navegar por las diferentes categorías, una vez que el usuario haya seleccionado el producto en específico elegirá la cantidad y se mostrara el valor a cancelar.

Diagramas de secuencia

Estos diagramas ayudan a verificar los diferentes roles de los objetos dentro del software mediante una secuencia de mensajes que describen la interacción entre ellos.

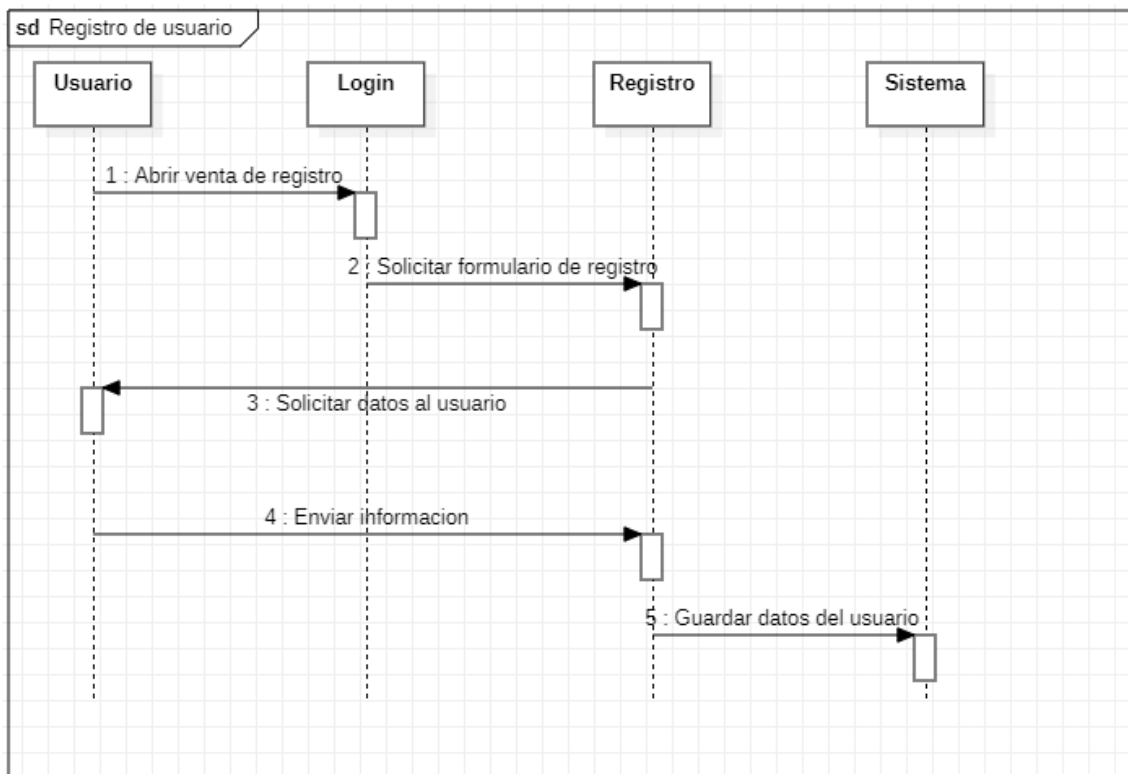


Figura 24: Diagrama de secuencia registro de usuario

La figura 24 permite conocer el proceso para un cliente nuevo, para lo cual se mostrará en primer lugar la interfaz del login, donde podrá acceder al formulario para ingresar sus datos y luego proceder a guardar la información en la base de datos para poder ingresar al sistema.

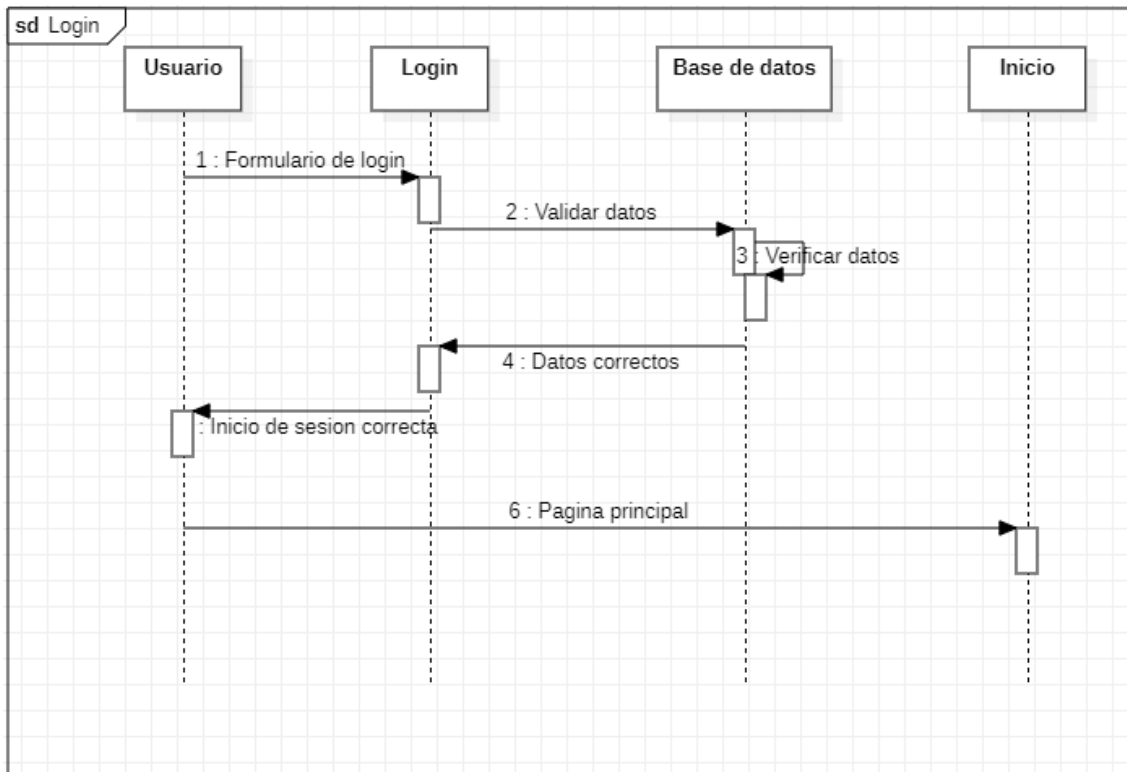


Figura 25: Diagrama de secuencia Inicio de sesión

La figura 25 nos permite conocer cómo será el proceso para el inicio de sesión, mediante el ingreso del correo y contraseña se procede a buscar dichos datos dentro de la base de datos, si el usuario existe el usuario podrá navegar por toda la aplicación y hacer uso de sus funciones.

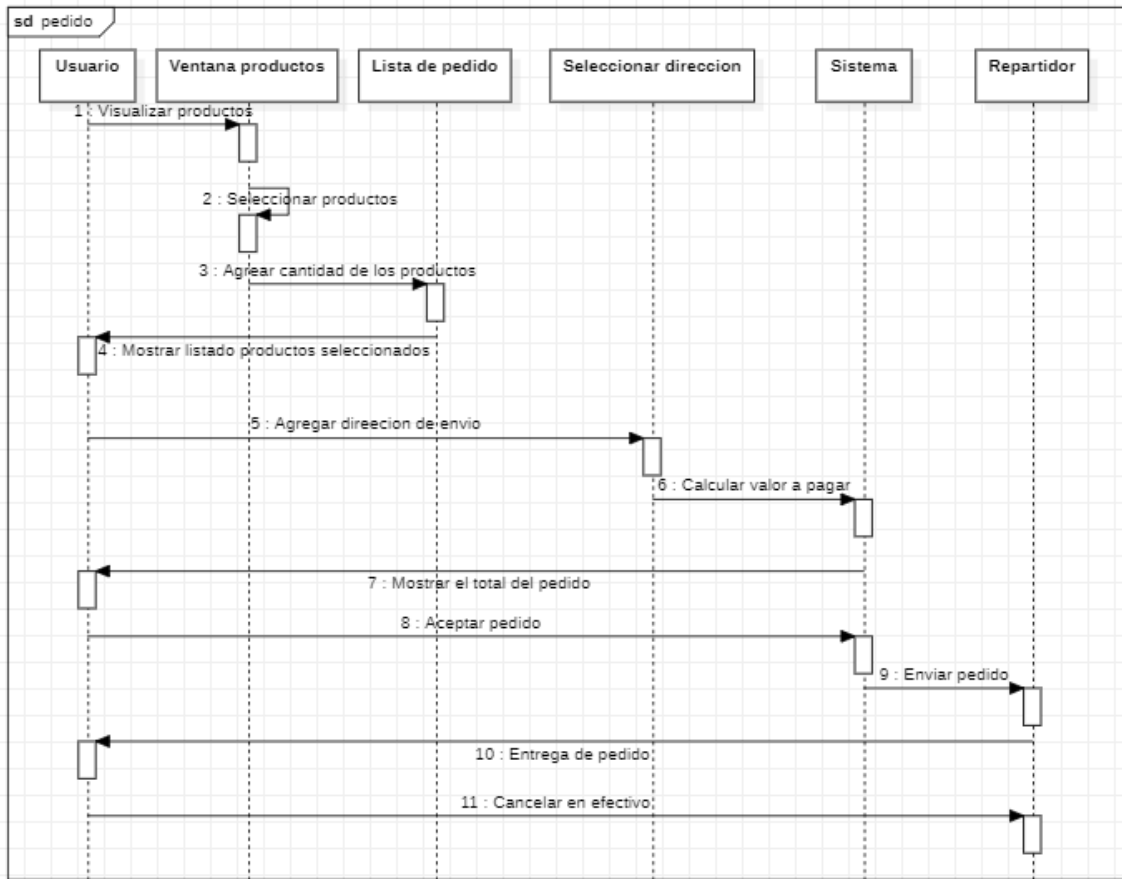


Figura 26: Diagrama de secuencia registro de pedido

La figura 26 nos permite conocer cómo será el procedimiento la selección de productos para realizar el pedido, la cual el cliente deberá seleccionar los productos y las cantidades, una vez seleccionados el sistema calculara el total a pagar, procediendo así en enviar el pedido al cliente a través de un repartidor.

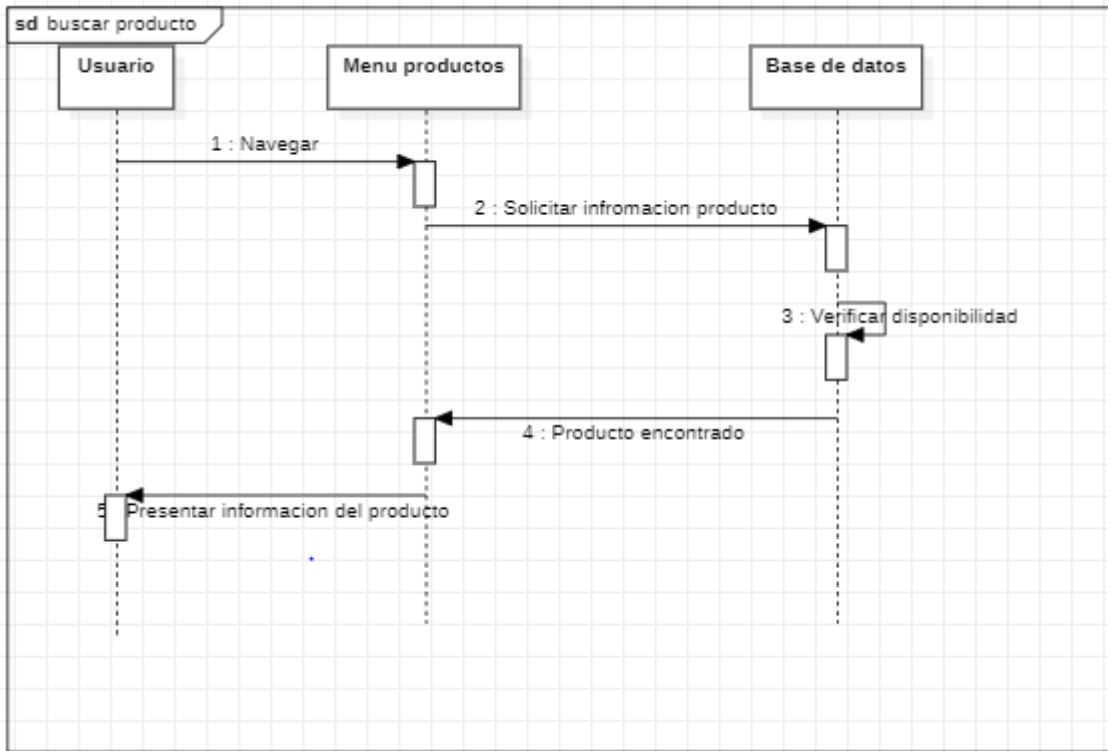


Figura 27: Diagrama de secuencia buscar producto

La figura 27 nos permite conocer los pasos que llevará a cabo el usuario para poder buscar un producto, la cual podrá navegar todas las categorías que están disponibles, al ingresar a un producto se solicitará la información en la base de datos y después presentar al usuario el producto seleccionado con la información de este.

2.5. EJECUCIÓN Y/O ENSAMBLAJE DEL PROTOTIPO

Para el desarrollo de la presente aplicación para realizar pedidos con entrega a domicilio del negocio CRUZ SALUD, se utilizó la metodología Swirl con el objetivo de realizar una aplicación de forma correcta y ágil.

A continuación, se detallarán los módulos funcionales más destacadas en la aplicación.

2.5.1. Módulo principal

Una vez que el usuario descargue la aplicación y ejecute se mostrará una ventana principal donde deberá iniciar sesión o a su vez registrarse a través de su correo electrónico para poder ingresar al sistema.

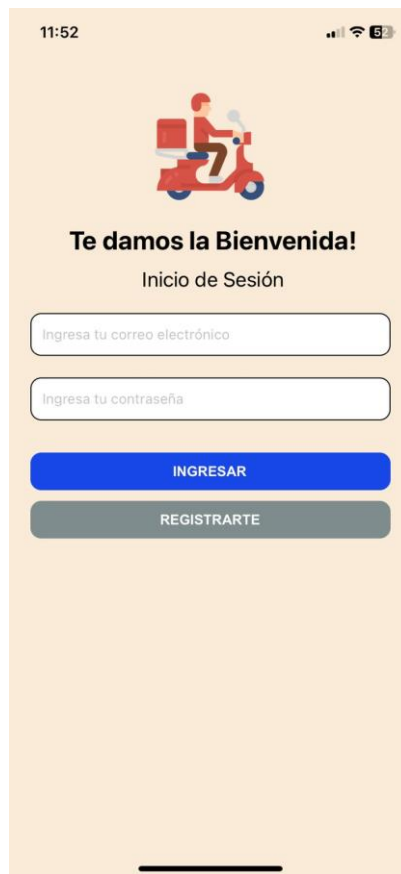


Figura 28: Módulo principal Inicio de Sesión

2.5.2. Módulo productos disponibles

Una vez que el usuario haya ingresado a la aplicación, debe dirigirse al feed de navegación de productos donde el usuario podrá conocer los diversos productos que se encuentran disponibles, esta ventana también incluye un buscador para ayudar al usuario a encontrar de una manera más rápida los productos que necesite, además de contar con una lista de categorías que facilita la organización de los productos.

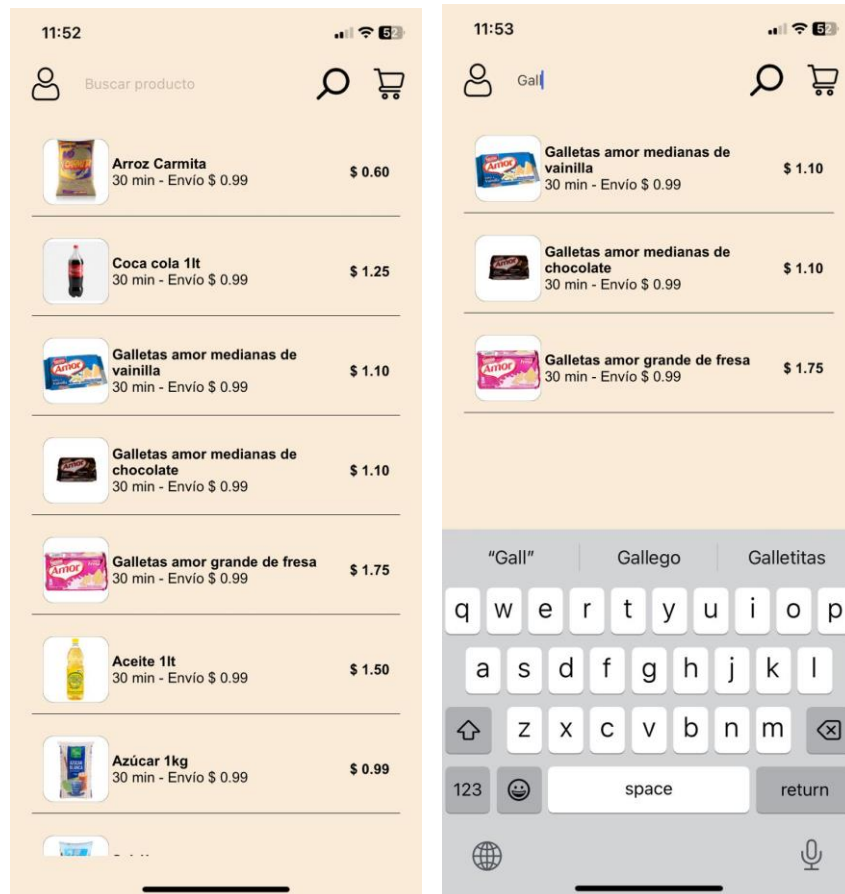


Figura 29: Módulo productos disponibles

2.5.3. Módulo general de pedidos

Una vez que el usuario que estén listos los productos que desea comprar se le mostrará una lista con todos los productos agregados con los respectivos precios y cantidades solicitadas, de esta manera el cliente podrá conocer el valor a cancelar. Como siguiente paso el usuario deberá poner la dirección de envío, finalmente se procederá a realizar el pedido para que el motorizado pueda llevar su pedido hasta su domicilio.

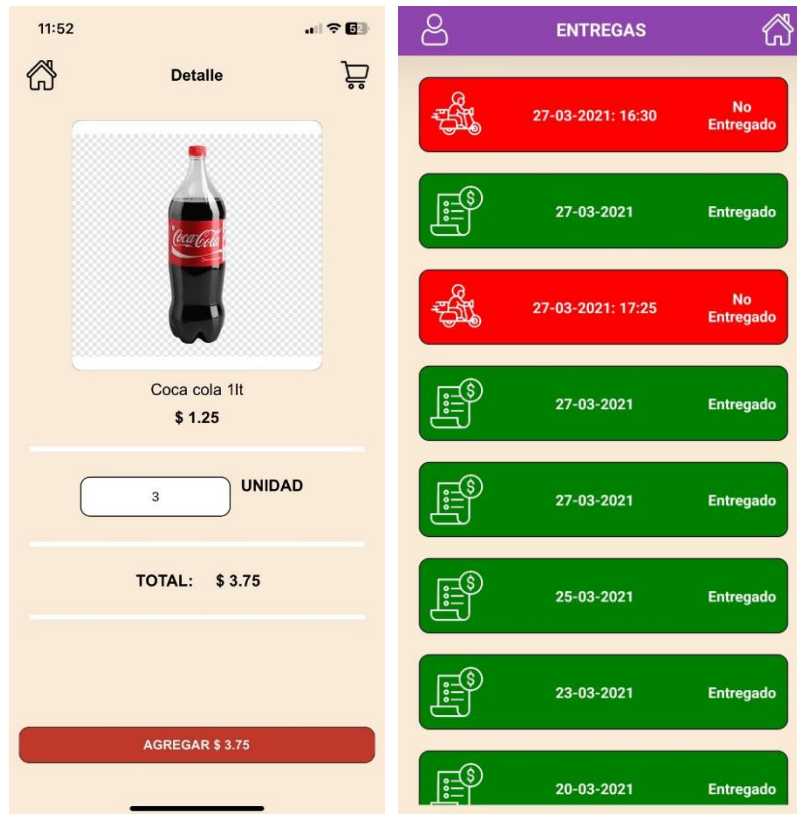


Figura 30: Módulo general de pedido

2.5.4. Módulo administración de datos de facturación

Este módulo permite al cliente proporcionar datos para que se le pueda emitir una factura, estos datos el usuario lo podrá modificar en cualquier momento.



Figura 31: Módulo administración de datos de facturación

2.5.5. Módulo perfil del cliente

En este modulo se puede apreciar el perfil del cliente, estos perfiles tiene la finalidad de dar a conocer el dueño del negocio toda la informacion importante de sus clientes



Figura 32: Módulo perfil del cliente

3. CAPÍTULO III. EVALUACIÓN DEL PROTOTIPO

3.1. PLAN DE EVALUACIÓN

Los criterios y objetivos del estándar ISO/IEC9126 se utilizan para la evaluación de aplicaciones, con el objetivo de proporcionar herramientas de evaluación de la calidad del software con el objetivo de probar la funcionalidad de los sistemas para cumplir con los requisitos de los usuarios.

El estándar evaluará los siguientes indicadores: funcionalidad, confiabilidad, facilidad de uso, eficiencia, mantenibilidad y portabilidad, los cuales serán divididos en subcategorías y lineamientos necesarios para determinar la calidad de la aplicación.

3.2. RESULTADOS DE LA EVALUACIÓN

Mediante los resultados que se mostraron al aplicar la norma IOS/IEC 9126 en la aplicación ha dado a conocer el nivel de calidad del sistema en base a las métricas de funcionalidad, confiabilidad, usabilidad, eficiencia, mantenibilidad y portabilidad, las cuales los resultados fueron evaluados mediante la escala de Likert que cuenta con una puntuación del 1 al 5.

Tabla 17: Resultados de la evaluación

Característica	Sub-características	Criterio	Escala de Likert				
			Excelente	Bueno	Regular	Malo	Muy malo
Funcionalidad	Conveniencia	El software desempeña las tareas requeridas por el usuario.	x				
	Exactitud	El software presenta los resultados sin errores.	x				
	Interoperabilidad	Las funciones del software van acorde al rol del usuario.	x				
	Seguridad	El software prohíbe el	x				

		acceso no autorizado.					
Confiabilidad	Tolerancia a las fallas	El software es tolerante antes las fallas.		x			
	Madurez	El software muestra mensajes de advertencia antes fallos.	x				
	Recuperación	El software puede seguir funcionando antes fallos.		x			
Usabilidad	Comprensión	El software detecta y entiende el rol de cada usuario.	x				
	Capacidad de aprender	El software es fácil a utilizar.	x				
	Operatividad	El software permite adaptarse velozmente a él.	x				
	Atractividad	El software tiene una interfaz amigable.	x				
Eficiencia	Comportamiento en el tiempo	El software tiene tiempos de espera agradables.	x				
	Gasto de recursos	El software utiliza de forma eficiente los recursos del dispositivo.		x			
Mantenibilidad	Facilidad de análisis	El software permite detectar fácilmente los errores.	x				
	Facilidad de pruebas	El software puede implementar un proceso de pruebas.		x			

	Facilidad de cambio	El software puede implementar cambios dentro del sistema rápidamente.	x				
Portabilidad	Capacidad de instalación	El software es fácil de instalar.	x				
	Capacidad de reemplazamiento	El software puede ser sustituido por otro.				x	

Para llevar a cabo los valores obtenidos de la tabla anterior, se presenta un resumen en una tabla para cada sub característica, dándonos a conocer de forma más exacta los resultados obtenidos.

Tabla 18: Resumen resultado de evaluación

Característica	Excelente	Bueno	Regular	Malo	Muy malo	Total
Funcionalidad	4					
Confiabilidad	2	1				
Usabilidad	4					
Eficiencia	1	1				
Mantenibilidad	2	1				
Portabilidad	1			1		

Después de la evaluación de calidad utilizando el estándar ISO/IEC 1926, según la tabla, se puede encontrar que la mayoría de las funciones tienen niveles excelentes y buenos, lo que demuestra que el software es de alta calidad.

Para un mejor entendimiento de los resultados se presenta un gráfico que represente la evaluación de calidad bajo la norma ISO/IEC 1926 a la aplicación móvil para la búsqueda de trabajo, visualice figura 34.

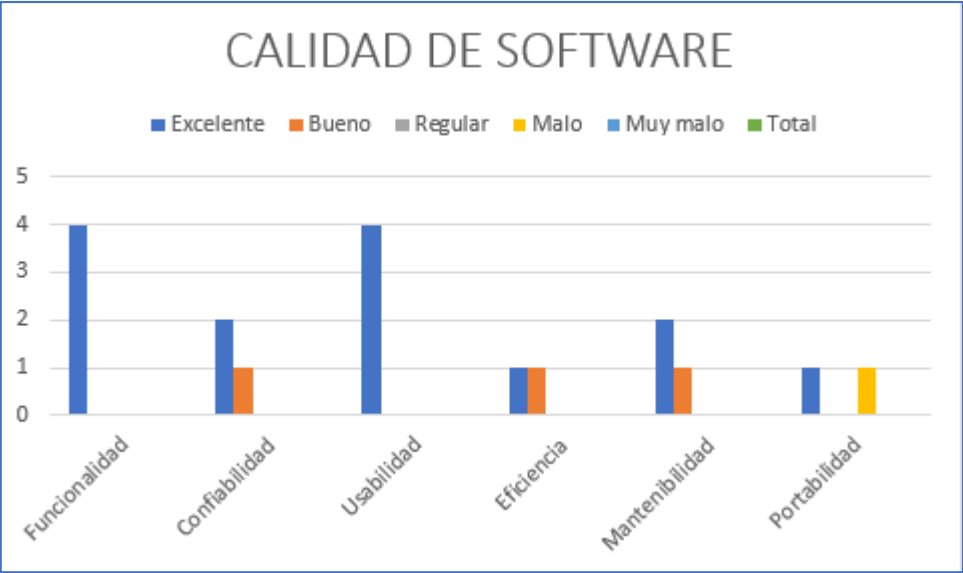


Figura 33: Resumen de los resultados de evaluación

3.3. CONCLUSIONES

Una vez finalizado la creación de la aplicación móvil para ofertas de trabajo aplicando la metodología Swirl, se puede afirmar lo siguiente:

- Se obtuvo los requisitos funcionales y no funcionales del software mediante el desarrollo de historias de usuario con la finalidad de determinar de una forma más clara los procesos de la aplicación.
- Las tecnologías implementadas en el desarrollo de este proyecto tales como: PostgreSQL, Node.js, React Native, permitieron obtener una aplicación multiplataforma con un backend totalmente en la nube lo que facilita una rápida accesibilidad a los datos.
- El desarrollo de los prototipos de la aplicación fue realizado con la herramienta Balsamiq Mockups, debido a que nos ofrece una vista previa de la aplicación que facilita la navegación y orientación del usuario dentro de la aplicación. Mediante las herramientas de modelado se permitió modelar correctamente los diferentes procesos y funciones de la aplicación, así como Bizaji y StarUML, la cual ayudó a optimizar y reducir contratiempos en la fase de implementación.
- Un conjunto de indicadores y lineamientos en el estándar ISO/IEC9126 pueden evaluar el software para cumplir con sus requisitos, logrando así niveles excelentes y buenos, por lo que debido a esta evaluación, es posible un software de calidad en la aplicación, satisfaciendo así las necesidades de los usuarios.

3.4. RECOMENDACIONES

Al concluir este proyecto es necesario reconocer varios puntos que faciliten y ayuden a mejorar el desarrollo y evitar todo tipo de retrasos.

- Para garantizar el cumplimiento de entregables en los tiempos establecidos es recomendable planificar actividades en relación al proyecto utilizando una metodología de software.
- Es indispensable realizar una investigación sobre las tecnologías que se van a implementar para el desarrollo del software con la finalidad de poder cumplir con todos los objetivos planteados y desarrollar una aplicación eficiente.
- Utilizar herramientas de evaluación de aplicaciones para detectar los posibles errores presentados en las diferentes etapas del desarrollo de la aplicación y poder mejorar estos errores sin afectar el funcionamiento.
- Se recomienda para proyectos de software que manejen grandes volúmenes de información el uso de una base de datos “NoSQL” ya que permiten almacenar grandes cantidades de información y evitar una afectación en la integridad de los datos.

REFERENCIAS BIBLIOGRAFICAS

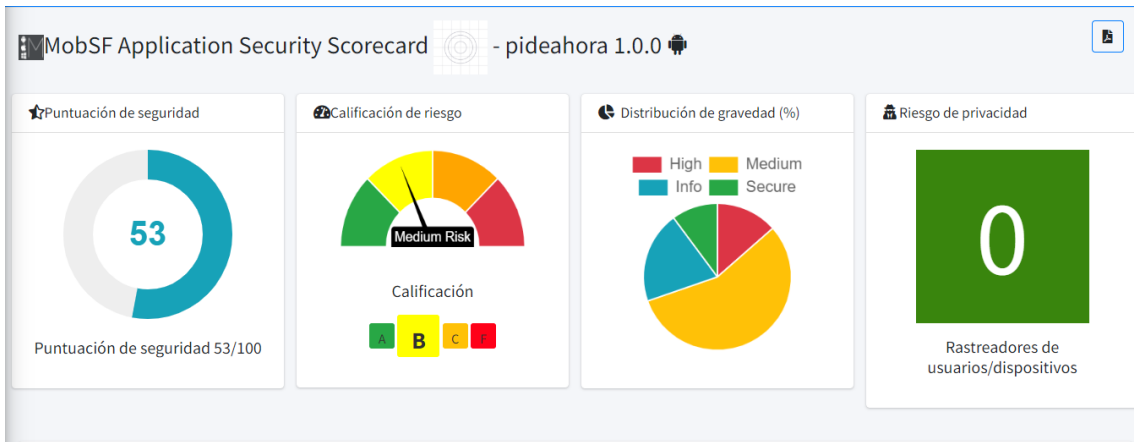
- [1] S. Oñate, «El confinamiento impulsa otra vez al “delivery” en Ecuador», *El Comercio*, 6 de mayo de 2021. <https://www.elcomercio.com/actualidad/negocios/confinamiento-impulsa-delivery-ecuador-pandemia.html> (accedido 27 de febrero de 2023).
- [2] A. V. R. Campuzano, «Aplicaciones digitales de delivery: Incidencia en el consumo de comidas rápidas durante el primer semestre del año 2020 en Guayaquil».
- [3] R. M. Hernandez, «Impacto de las TIC en la educación: Retos y Perspectivas», *Propósitos Represent.*, vol. 5, n.º 1, p. 325, abr. 2017, doi: 10.20511/pyr2017.v5n1.149.
- [4] M.-Á. García-Madurga, A. J. Grilló-Méndez, y T. Morte-Nadal, «La adaptación de las empresas a la realidad COVID: una revisión sistemática», *Retos*, vol. 11, n.º 21, pp. 55-70, mar. 2021, doi: 10.17163/ret.n21.2021.04.
- [5] L. A. Cobo Campo y R. I. Pérez Uribe, «Proyecto Anamnesis – Desarrollo de una aplicación web y móvil para la gestión de una historia clínica unificada de los colombianos», *Rev. Esc. Adm. Negocios*, n.º 80, pp. 91-104, jun. 2016, doi: 10.21158/01208160.n80.2016.1459.
- [6] J. Christino, É. Cardozo, R. Petrin, y L. Pinto, «Factors Influencing the Intent and Usage Behavior of Restaurant Delivery Apps», *Rev. Bus. Manag.*, pp. 21-42, ene. 2021, doi: 10.7819/rbgn.v23i1.4095.
- [7] «rcim06116.pdf».
- [8] INGAR - Instituto de Desarrollo y Diseño, M. J. Blas, H. Leone, INGAR - Instituto de Desarrollo y Diseño, S. Gonnet, y INGAR - Instituto de Desarrollo y Diseño, «Modelado y Verificación de Patrones de Diseño de Arquitectura de Software para Entornos de Computación en la Nube», *RISTI - Rev. Ibérica Sist. E Tecnol. Informação*, n.º 35, pp. 1-17, dic. 2019, doi: 10.17013/risti.35.1-17.
- [9] L. M. Nancy Magaly, M. R. Jimmy Rolando, M. R. Rodrigo Fernando, y P. S. Celmira Marcela, «Mensajería cliente-servidor aplicando sockets en las herramientas GEANY IDE 1.31, PHYTON 3.7 y POSTGRESQL 9.5 en el sistema operativo CENTOS 7: Client-server messaging applying sockets in GEANY IDE 1.31, PHYTON 3.7 and POSTGRESQL 9.5 tools in CENTOS 7 operating system», *Conf. Proc. Machala*, vol. 4, n.º 1, pp. 177-185, ago. 2020, doi: 10.48190/cp.v4n1a11.
- [10] C. F. V. Pérez, J. E. P. Cleves, y L. Pallares, «COMPUTACIÓN EN LA NUBE: UN NUEVO PARADIGMA EN LAS TECNOLOGÍAS DE LA INFORMACIÓN Y LA COMUNICACIÓN».
- [11] Z. E. Mamani Rodríguez, L. Del Pino Rodríguez, y J. C. Gonzales Suarez, «Arquitectura basada en Microservicios y DevOps para una ingeniería de software continua», *Ind. Data*, vol. 23, n.º 2, pp. 141-149, dic. 2020, doi: 10.15381/idata.v23i2.17278.
- [12] E. Salinas, N. Cerpa, y P. Rojas, «Arquitectura orientada a servicios para software de apoyo para el proceso personal de software», *Ingeniare Rev. Chil. Ing.*, vol. 19, n.º 1, pp. 40-52, jun. 2011, doi: 10.4067/S0718-33052011000100005.
- [13] D. Inupakutika, S. Kaghyan, D. Akopian, P. Chalela, y A. G. Ramirez, «Facilitating the development of cross-platform mHealth applications for chronic supportive care and a case study», *J. Biomed. Inform.*, vol. 105, p. 103420, may 2020, doi: 10.1016/j.jbi.2020.103420.
- [14] M. P. Zea Ordóñez, J. R. Molina Ríos, y F. F. Redrován Castillo, *ADMINISTRACIÓN DE BASES DE DATOS CON POSTGRESQL*, 1.ª ed. Editorial Científica 3Ciencias, 2017. doi: 10.17993/IngyTec.2017.18.
- [15] «3082-Article Text-20347-2-10-20190725.htm».
- [16] A. Makris, K. Tserpes, G. Spiliopoulos, y D. Anagnostopoulos, «Performance Evaluation of MongoDB and PostgreSQL for spatio-temporal data».
- [17] T. Gkamas, V. Karaiskos, y S. Kontogiannis, «Performance Evaluation of Distributed Database Strategies Using Docker as a Service for Industrial IoT Data: Application to Industry 4.0», *Information*, vol. 13, n.º 4, p. 190, abr. 2022, doi: 10.3390/info13040190.

- [18] N. Battaglia, C. Neil, M. D. Vincenzi, R. Martínez, y D. González, «10238 UCASE - CL: APRENDIZAJE COLABORATIVO DE LA INGENIERÍA DE SOFTWARE EN ENTORNOS VIRTUALES UBICUOS».
- [19] E. Vacacela, J. Tenecota, J. Torres, y J. Celleri-Pacheco, «Automatización de procesos de Investigación, Vinculación, Prácticas/Pasantías Preprofesionales para universidades ecuatorianas», *Alternativas*, vol. 19, n.º 1, jun. 2019, doi: 10.23878/alternativas.v19i1.186.
- [20] M. Zhang, H. Chen, y A. Luo, «A Systematic Review of Business-IT Alignment Research With Enterprise Architecture», *IEEE Access*, vol. 6, pp. 18933-18944, 2018, doi: 10.1109/ACCESS.2018.2819185.
- [21] Z. Munawar y D. Z. Musadad, «PERANCANGAN INTERFACE APLIKASI PENCATATAN PERSEDIAAN BARANG DI KIOS BUKU PALASARI BANDUNG DENGAN METODE USER CENTERED DESIGN MENGGUNAKAN BALSAMIQ MOCKUPS», *J. Inform.*, vol. 06.
- [22] R. P. Valderrama y I. P. Valderrama, «Aplicación Web con sistema asistencial para el modelado UML usando inteligencia artificial para una infraestructura de software para la Educación 4.0 Web application with assistance system for UML modeling using artificial intelligence for a software infrastructure for Education 4.0», 2020.
- [23] G. D. Salas-Paucar y M. A. Campoverde-Molina, «Proceso de Gestión de Asistencia Técnica. Caso de estudio: Hospital General Isidro Ayora de Loja, Ecuador», *Dominio Las Cienc.*, vol. 5, n.º 3, p. 159, jul. 2019, doi: 10.23857/dc.v5i3.930.
- [24] B. Rajkumarsingh y D. Totah, «Drowsiness Detection using Android Application and Mobile Vision Face API», *RD J.*, vol. 37, 2021, doi: 10.17159/2309-8988/2021/v37a4.
- [25] Fan *et al.*, «CoVSCode: A Novel Real-Time Collaborative Programming Environment for Lightweight IDE», *Appl. Sci.*, vol. 9, n.º 21, p. 4642, oct. 2019, doi: 10.3390/app9214642.
- [26] J. Fitzgerald, T. Oda, y H. D. Macedo, «Proceedings of the 18th International Overture Workshop». arXiv, 19 de enero de 2021. Accedido: 23 de febrero de 2023. [En línea]. Disponible en: <http://arxiv.org/abs/2101.07261>
- [27] S. Delcev y D. Draskovic, «Modern JavaScript frameworks: A Survey Study», en *2018 Zooming Innovation in Consumer Technologies Conference (ZINC)*, may 2018, pp. 106-109. doi: 10.1109/ZINC.2018.8448444.
- [28] R. S. Malik, J. Patra, y M. Pradel, «NL2Type: Inferring JavaScript Function Types from Natural Language Information», en *2019 IEEE/ACM 41st International Conference on Software Engineering (ICSE)*, may 2019, pp. 304-315. doi: 10.1109/ICSE.2019.00045.
- [29] E. Haro, T. Guarda, A. O. Z. Peñaherrera, y N. Quiña, «Desarrollo backend para aplicaciones web, Servicios Web Restful: Node.js vs Spring Boot».
- [30] F. Doglio, *REST API Development with Node.js: Manage and Understand the Full Capabilities of Successful REST Development*. Berkeley, CA: Apress, 2018. doi: 10.1007/978-1-4842-3715-1.
- [31] «tvilaltav.pdf».
- [32] J. Rönkkö, «Koneen häiriöajan seurantatyö- kalun toteutus».
- [33] A. Patel, «The Difference Between iOS and Android App Development Using React Native», 21 de noviembre de 2018. <https://www.mindinventory.com/blog/difference-between-ios-android-app-development-using-react-native/> (accedido 23 de febrero de 2023).
- [34] J. P. Z. Gamboa y C. A. L. Arreaga, «Evolución de las Metodologías y Modelos utilizados en el Desarrollo de Software. Evolution of the Methodologies and Models used in Software Development.», vol. 3, n.º 10, 2018.
- [35] F. S. Blanco, F. G. Sierra, y J. C. R. Rivera, «ESTABLISHMENT OF AGILE TEAMS FOR SOFTWARE DEVELOPMENT: REVIEW OF LITERATUREi», *Dimens. Empres.*.
- [36] C. M. Budoya, M. M. Kissake, y J. S. Mtebe, «Instructional Design enabled Agile Method using ADDIE Model and Feature Driven Development Process».

- [37] C. Tam, E. J. da C. Moura, T. Oliveira, y J. Varajão, «The factors influencing the success of on-going agile software development projects», *Int. J. Proj. Manag.*, vol. 38, n.º 3, pp. 165-176, abr. 2020, doi: 10.1016/j.ijproman.2020.02.001.
- [38] M. A. Santos-Romero y N. E. Escudero-López, «Propuesta de Metodología Híbrida y Base de Documentación para el Desarrollo de Software Actual», 2020.
- [39] H. L. I. Marcelo y C. C. J. Jeorwin, «TRABAJO TITULACIÓN PROPUESTAS TECNOLÓGICAS», 2022.
- [40] «MolinaRios_Jimmy_TD_2021.pdf».
- [41] «administrador,+Gestor_a+de+la+revista,+5909-29238-1-CE.pdf».
- [42] A. A. Salazar, «Identificación De Los Stakeholders Y Conflictos De Interés En Las Cooperativas De Ahorro Y Crédito De Ecuador», *Rev. Científica Visión Futuro*, vol. 22, n.º 2, pp. 73-94, 2018.
- [43] M. P. Abreu, C. R. R. Rodríguez, y P. Y. P. Pérez, «Computación con palabras para el análisis de factibilidad de proyectos de software», vol. 20, n.º 50, 2016.
- [44] «Conjunto habitacional de interés social para el barrio San Antonio en Riobamba. Análisis de factibilidad», *CHAKIÑAN Rev. Cienc. Soc. HUMANIDADES*, n.º 7, pp. 40-57, abr. 2019, doi: 10.37135/chk.002.07.04.
- [45] A. Montenegro, R. Zamora Sánchez, y V. Arias-Villavicencio, «Análisis de factibilidad en la producción de calzado ortopédico para niños en la provincia de Tungurahua», *Retos*, vol. 7, n.º 14, p. 131, sep. 2017, doi: 10.17163/ret.n14.2017.07.
- [46] A. Altamirano, «Identificación de los Stakeholders y conflictos de interés en las Cooperativas de Ahorro y Crédito del Ecuador», *Visión Futuro*, vol. 22, n.º 2, pp. 00-00, jul. 2018.
- [47] K. I. Gómez Sotelo, C. Baron, P. Esteban, C. Y. A. G. Estrada, y L. de J. Laredo Velázquez, «How to find non-functional requirements in system developments», *IFAC-Pap.*, vol. 51, n.º 11, pp. 1573-1578, 2018, doi: 10.1016/j.ifacol.2018.08.272.
- [48] A. J. Gregory, J. P. Atkins, G. Midgley, y A. M. Hodgson, «Stakeholder identification and engagement in problem structuring interventions», *Eur. J. Oper. Res.*, vol. 283, n.º 1, pp. 321-340, may 2020, doi: 10.1016/j.ejor.2019.10.044.

ANEXOS

Anexo 1: Evaluación utilizando la herramienta MobSF de análisis dinámico



Mafia SF

ESCAÑEOS RECIENTES | ANALIZADOR ESTÁTICO | ANALIZADOR DINÁMICO | API REST | DONAR | DOCUMENTOS | ACERCA DE

Analizador estático

- Información
- Opciones de escaneo
- Certificado de firmante
- permisos
- API de Android
- Actividades navegables
- Análisis de seguridad
- Análisis de malware
- Reconocimiento
- Componentes
- Informe PDF

PUNTAJES DE LA APLICACIÓN: 53/100

DETECCIÓN DE RASTREADORES: 0/428

INFORMACIÓN DEL ARCHIVO:

- Nombre del archivo: aplicación-3ffb07fe-6290-45af-ad59-d151f86cb893.apk
- Tamaño: 23,44 MB
- MD5: c99f5cf308ca78233b430293369938c6
- SHA1: 41826e6769fe9e7b83440e99ef23e09facb356e1
- SHA256: 07fd61ffa7ae2c3a6b009ef21ff8d5dca7262c02f119f2d05989ec0b8e4dd04b

INFORMACIÓN DE LA APLICACIÓN:

- Nombre de la aplicación: pideahora
- Nombre del paquete: com.javi21.pideahora
- Actividad principal: com.javi21.pideahora.MainActivity
- SDK de destino: 33 | SDK mínimo: 21 | SDK máximo: 33
- Nombre de la versión de Android: 1.0.0
- Código de versión de Android: 1

Tarjeta de puntuación de MobSF

- ACTIVIDADES: 2
- SERVICIOS: 0
- RECEPTORES: 0
- PROVEEDORES: 2