



# UTMACH

FACULTAD DE INGENIERÍA CIVIL

CARRERA DE INGENIERÍA DE SISTEMAS

DESARROLLO DE UNA APLICACION MOVIL DE GESTION DE  
PEDIDOS PARA LA EMPRESA POLLO BROSTER DLAZONA  
UTILIZANDO TECNOLOGIA HIBRIDA

CAMPOVERDE VEGA JULIO ADRIAN  
INGENIERO DE SISTEMAS

MACHALA  
2022



# UTMACH

FACULTAD DE INGENIERÍA CIVIL

CARRERA DE INGENIERÍA DE SISTEMAS

DESARROLLO DE UNA APLICACION MOVIL DE GESTION DE  
PEDIDOS PARA LA EMPRESA POLLO BROSTER DLAZONA  
UTILIZANDO TECNOLOGIA HIBRIDA

CAMPOVERDE VEGA JULIO ADRIAN  
INGENIERO DE SISTEMAS

MACHALA  
2022



# UTMACH

FACULTAD DE INGENIERÍA CIVIL

CARRERA DE INGENIERÍA DE SISTEMAS

TRABAJO TITULACIÓN  
PROPUESTAS TECNOLÓGICAS

DESARROLLO DE UNA APLICACION MOVIL DE GESTION DE PEDIDOS PARA  
LA EMPRESA POLLO BROSTER DLAZONA UTILIZANDO TECNOLOGIA HIBRIDA

CAMPOVERDE VEGA JULIO ADRIAN  
INGENIERO DE SISTEMAS

CARTUCHE CALVA JOFFRE JEORWIN

MACHALA, 20 DE SEPTIEMBRE DE 2022

MACHALA  
2022

# tesis

## INFORME DE ORIGINALIDAD

6%

INDICE DE SIMILITUD

5%

FUENTES DE INTERNET

1%

PUBLICACIONES

3%

TRABAJS DEL  
ESTUDIANTE

## FUENTES PRIMARIAS

1

Submitted to Universidad Técnica de Machala

Trabajo del estudiante

2%

2

oa.upm.es

Fuente de Internet

<1%

3

hdl.handle.net

Fuente de Internet

<1%

4

repositorio.umsa.bo

Fuente de Internet

<1%

5

Submitted to Consorcio CIXUG

Trabajo del estudiante

<1%

6

uisrael.edu.ec

Fuente de Internet

<1%

7

dokumen.pub

Fuente de Internet

<1%

8

repositorio.unp.edu.pe

Fuente de Internet

<1%

9

aws.amazon.com

Fuente de Internet

<1%

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

El que suscribe, CAMPOVERDE VEGA JULIO ADRIAN, en calidad de autor del siguiente trabajo escrito titulado DESARROLLO DE UNA APLICACION MOVIL DE GESTION DE PEDIDOS PARA LA EMPRESA POLLO BROSTER DLAZONA UTILIZANDO TECNOLOGIA HIBRIDA, 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, 20 de septiembre de 2022



CAMPOVERDE VEGA JULIO ADRIAN  
0706619541

## **DEDICATORIA**

Este trabajo está dedicado principalmente a Dios y a mi Familia, por el amor y los consejos que me ha brindado en los momentos más difíciles de mi vida, por su apoyo incondicional y oraciones que han realizado en mi nombre para alcanzar las metas que me he propuesto, ya que sin ellos y todo el cariño y amor que me brindan no hubiera podido cumplir este objetivo

Sr. Campoverde Vega Julio Adrián

## **AGRADECIMIENTO**

Principalmente Agradezco a Dios por haberme otorgado una maravillosa familia, quienes me han brindado todo su apoyo lo cual me ha permitido alcanzar esta meta, a mis padres por ser un ejemplo a seguir de perseverancia y esfuerzo, por su amor incondicional y sus sabios consejos. También me gustaría agradecer a todos los docentes que conforman la Carrera de Ingeniería de Sistemas quienes compartieron conmigo sus conocimientos y experiencias, preparándome de la mejor manera para la vida profesional.

Sr. Campoverde Vega Julio Adrián

## RESUMEN

Con el pasar del tiempo, los grandes y pequeños negocios buscan incrementar el alcance de los clientes, implementando la tecnología como una herramienta vital dentro del negocio. Hoy en día, el uso de dispositivos móviles está creciendo de manera exponencial, convirtiendo a las aplicaciones móviles en una parte fundamental de la diaria rutina de cada persona, sea adolescente o adulto. Al igual que el uso de la tecnología para entretenimiento o aplicaciones diarias, el ecommerce ha llegado a ser considerada como herramienta funcional para pequeños y grandes negocios. De esta manera el negocio expande su alcance, y los clientes tienen mayor facilidad y acceso a los productos y promociones que estos ofertan día a día, desde la comodidad del hogar.

Los procesos de pedidos a domicilio de la empresa Pollo Broster D La Zona, se gestionan de manera similar a los pedidos en local. Sin embargo, muchas de las actividades inmersas en este proceso consumen más recursos de lo normal, un gran ejemplo es la comunicación de los comensales con el negocio para conocer el menú, los productos y los precios, y posteriormente hacer un pedido. El tiempo invertido en esos procesos se puede reducir empleando una tienda virtual, en donde el cliente pueda conocer los productos ofertados y entablar solo la comunicación necesaria para finalizar el proceso de compra.

Por ello, se propone optimizar los procesos de negocio mediante una app de gestión de pedidos a domicilio. La propuesta consta en dos aplicativos móviles, el primero orientado al administrador, es decir el negocio, de manera que se puedan administrar los productos y ordenes de pedido, reduciendo el tiempo y recursos invertidos en el proceso de venta a domicilio. El segundo está orientado al cliente, de manera que pueda visualizar el menú y proceder a realizar la compra de manera rápida y sencilla, permitiendo gestionar sus propios datos de facturación y las direcciones de envío.

Un punto clave dentro de la propuesta es la multiplataforma, es por ello que se ha considerado desarrollar las funcionalidades de las apps mediante React Native y Next, y una base de datos en la Nube mediante Firebase. La tecnología en la Nube se ha seleccionado debido a su facilidad de



comunicación y disponibilidad, sin mencionar, los diversos servicios que ofrece Firebase para el desarrollo de apps móviles, como por ejemplo Authentication, Firestore, Storage, Functions y Messaging. Cada una empleada para cumplir un requerimiento del sistema. De esta manera se facilita el desarrollo de una app que pueda ser ejecutada tanto en Android como en iOS. Finalmente, para garantizar un desarrollo eficiente y productivo se consideran algunas de las fases de la metodología de MMS, gracias a su enfoque ágil en todo su ciclo de vida. Gracias al enfoque ágil se logró mantener una comunicación constante con el cliente y sobre todo con los interesados. Se utiliza también la normativa ISO/IEC 25010 para garantizar la calidad de las apps, evaluando que cumplan respectivamente con las métricas de mayor relevancia para aplicaciones móviles, como son la usabilidad, funcionalidad, y seguridad.

**Palabras clave:** MMS metodología, Aplicaciones móviles, e-commerce, gestión de pedidos, React Native.

## **ABSTRACT**

As time goes by, large and small businesses seek to increase the range of customers, implementing technology as a vital tool within the business. Today, the use of mobile devices has increased exponentially, and with it the use of mobile applications has become part of daily's routine. As well as the use of technology for entertainment or daily applications, e-commerce has become a functional tool used from small to big enterprises. In this way the business expands its reach, and customers have greater access to products and promotions that they offer every day, from the comfort of home.

Pollo Broster D La Zona's delivery order processes are managed in a similar way to local orders. However, many of the activities involved in this process consume more resources than normal; a great example is the communication of diners with the business to learn about the menu, products or prices, and then place an order. The time invested in these processes can be reduced by using a virtual store, where the customer can learn about the products offered and set up only the necessary communication to complete the purchase process.

Therefore, this project proposes to optimize business processes through a home order management app. The proposal consists of two mobile applications, the first one oriented to the administrator, i.e. the business, so that products and orders can be managed, reducing the time and resources invested in the process of home sales. The second one is oriented to the customer, so that he can visualize the menu and proceed to make the purchase in a quick and easy way, allowing to manage his own billing data and shipping addresses.

A key point within the proposal is the multiplatform, that is why it has been considered to develop the functionality of the apps using React Native and Next, and the database using Firebase in the Cloud. Cloud technology has been selected due to its ease of communication and availability, not to mention the various services offered by Firebase for the development of mobile apps, such as Authentication, Firestore, Storage, Functions and Messaging. Each one is used to fulfill a requirement of the system. This facilitates the development of applications that can run on both Android and iOS. Finally, to ensure an efficient

and productive development, some of the phases of the MMS methodology are considered, thanks to its agile approach throughout its life cycle. Thanks to the agile approach it was possible to maintain constant communication with the customer and especially with the stakeholders. The ISO/IEC 25010 standard is also used to guarantee the quality of the apps, evaluating that they comply with the most relevant metrics for mobile applications, such as usability, functionality and security.

**Keywords:** MMS methodology, mobile applications, e-commerce, order management, React Native.

## ÍNDICE GENERAL

DEDICATORIA .....	1
AGRADECIMIENTO .....	2
RESUMEN .....	3
ABSTRACT .....	5
INTRODUCCIÓN .....	11
1 CAPÍTULO I. DIAGNÓSTICO DE NECESIDADES Y REQUERIMIENTOS. ....	13
1.1 ÁMBITO DE APLICACIÓN: DESCRIPCIÓN DEL CONTEXTO Y HECHOS DE INTERÉS .....	13
1.2 ESTABLECIMIENTOS DE REQUERIMIENTOS .....	14
1.3 JUSTIFICACIÓN DE REQUERIMIENTOS A SATISFACER .....	17
2 CAPÍTULO II. DESARROLLO DEL PROTOTIPO .....	19
2.1 DEFINICIÓN DEL PROTOTIPO TECNOLÓGICO. ....	19
2.2 FUNDAMENTACIÓN TEÓRICA DEL PROTOTIPO .....	23
2.3 OBJETIVOS DEL PROTOTIPO. ....	32
2.4 DISEÑO DEL PROTOTIPO .....	33
2.5 EJECUCIÓN Y/O ENSAMBLAJE DEL PROTOTIPO .....	66
3 CAPÍTULO III: EVALUACIÓN DEL PROTOTIPO .....	78
3.1 PLAN DE EVALUACIÓN. ....	79
3.2 RESULTADOS DE EVALUACIÓN .....	80
3.3 CONCLUSIONES. ....	83
3.4 RECOMENDACIONES .....	84
BIBLIOGRAFÍA .....	85
ANEXOS .....	89
ANEXO A: DESGLOSE DE REQUERIMIENTOS .....	89
ANEXO B: DEFINICIÓN DE HISTORIAS DE USUARIO .....	97
ANEXO C: ESTRUCTURA DE DESGLOSE DE TRABAJO .....	104

## ÍNDICE DE TABLAS

Tabla 1 Matriz de Stakeholders.....	38
Tabla 2 Usuario 1 del sistema.....	39
Tabla 3 Usuario 2 del sistema.....	40
Tabla 4 Matriz general de requerimientos funcionales .....	42
Tabla 5 Matriz general de requerimientos no funcionales .....	43
Tabla 6 Esquema general de historias de usuarios.....	45
Tabla 7 Sprint Backlog .....	47
Tabla 8 Sprint Backlog .....	49
Tabla 9: Módulos del sistema.....	55
Tabla 10: Revisión de entregables por sprint .....	78
Tabla 11: plan de evaluación 25010.....	79
Tabla 12: Escala de Likert en evaluación 25010 .....	80
Tabla 13: plan de evaluación 25010.....	81
Tabla 14: Especificación de RF-1.....	89
Tabla 15: Especificación de RF-2.....	90
Tabla 16: Especificación de RF-3.....	90
Tabla 17: Especificación de RF-4.....	91
Tabla 18: Especificación de RF-5.....	92
Tabla 19: Especificación de RF-6.....	94
Tabla 20: Especificación de RF-7.....	94
Tabla 21: Especificación de RF-8.....	95
Tabla 22: Historia de usuario No1 .....	97
Tabla 23: Historia de usuario No2 .....	97
Tabla 24: Historia de usuario No3 .....	98
Tabla 25: Historia de usuario No4 .....	98
Tabla 26: Historia de usuario No5 .....	99
Tabla 27: Historia de usuario No6 .....	99
Tabla 28: Historia de usuario No7 .....	100
Tabla 29: Historia de usuario No8 .....	100
Tabla 30: Historia de usuario No9 .....	101
Tabla 31: Historia de usuario No10 .....	101
Tabla 32: Historia de usuario No11 .....	102

Tabla 33: Historia de usuario No12 .....	102
Tabla 34: Historia de usuario No13 .....	103

## ÍNDICE DE ILUSTRACIONES

Figura 1 Arquitectura del software.....	20
Figura 2 Arquitectura del hardware: cliente-servidor .....	21
Figura 3 Estructura de fundamentación teórica del prototipo.....	23
Figura 4 MMS: Ciclo de vida .....	25
Figura 5 Tecnologías Backend.....	28
Figura 6 Base de datos – Cloud Firestore .....	31
Figura 7 Tecnologías Backend.....	31
Figura 8 Modelo de negocio actual .....	35
Figura 9 Modelo de negocio optimizado.....	36
Figura 10 Cronograma de actividades .....	47
Figura 11 Esquema NoSql empleado para la app administrador.....	53
Figura 12 Esquema NoSql empleado para la app cliente.....	54
Figura 13 Prototipo de interfaz cliente: inicio de sesión .....	56
Figura 14 Prototipo de interfaz cliente: pantalla de inicio.....	57
Figura 15 Prototipo de interfaz cliente: perfil de usuario.....	57
Figura 16 Prototipo de interfaz cliente: datos de facturación .....	58
Figura 17 Prototipo de interfaz cliente: datos de facturación .....	59
Figura 18 Prototipo de interfaz cliente: ayuda e información .....	59
Figura 19 Prototipo de interfaz cliente: menú y productos .....	60
Figura 20 Prototipo de interfaz cliente: agregar al carrito .....	61
Figura 21 Prototipo de interfaz cliente: visualización de pedido .....	61
Figura 22 Prototipo de interfaz cliente: envío y método de pago .....	62
Figura 23 Prototipo de interfaz cliente: visualizar pedidos .....	62
Figura 24 Prototipo de interfaz administrador: Gestión de secciones .....	63
Figura 25 Prototipo de interfaz administrador: Gestión de productos .....	64
Figura 26 Prototipo de interfaz administrador: pedidos .....	65
Figura 27 Prototipo de interfaz administrador: panel informativo (dashboard) .....	66
Figura 28 Prototipo de interfaz administrador: ayuda e información .....	66
Figura 29 Implementación del prototipo: inicio de sesión .....	68

Figura 30 Implementación del prototipo: registro.....	69
Figura 31 Implementación del prototipo: perfil de usuario .....	69
Figura 32 Implementación del prototipo: datos de facturacion.....	70
Figura 33 Implementación del prototipo: direcciones .....	71
Figura 34 Implementación del prototipo: proceso de compra .....	72
Figura 35 Implementación del prototipo: pedidos .....	73
Figura 36 Implementación del prototipo: .....	73
Figura 37 Implementación del prototipo: gestión de usuarios .....	74
Figura 38 Implementación del prototipo: gestión de secciones .....	75
Figura 39 Implementación del prototipo: gestión de productos.....	75
Figura 40 Implementación del prototipo: gestión de pedido .....	76
Figura 41 Implementación del prototipo: gestión de pedido aceptado .....	77
Figura 42 Implementación del prototipo: gestión de pedido en camino .....	77
Figura 43 EDT del proyecto .....	104

## INTRODUCCIÓN

En la actualidad, la implementación y optimización de procesos de negocio mediante aplicaciones móviles ha llegado a ser una de las herramientas fundamentales para potenciar el negocio de cualquier empresa o negocio, ya sea esta grande, mediana o pequeña. Con el avance de la tecnología y la era de la digitalización la implementación de aplicaciones móviles ha llegado a ser tendencia no solo entre los desarrolladores de software, sino también entre las PYMES y grandes empresas, principalmente para la optimización de procesos contables. De igual manera, ha incrementado el uso de aplicaciones móviles por parte de los usuarios [1] para actividades cotidianas, siendo la más frecuente las compras en línea [2].

El comercio electrónico (ecommerce) es una de las tendencias tecnológicas con rápido crecimiento a nivel mundial, como uno de los efectos de las restricciones durante la pandemia de COVID-19. En Ecuador, durante el periodo mencionado, hubo un incremento notable de aproximadamente el 42% para pagos mediante transacciones electrónicas en restaurantes [3], llegando a ser considerado el e-commerce como unas de las principales herramientas tecnológicas para incrementar los ingresos en restaurantes, puestos de comidas o emprendimientos.

Sin embargo, la necesidad de un aplicativo móvil para la gestión de operaciones del comercio electrónico dentro de un negocio, viene acompañado con la multiplataformidad, es decir para cualquier tipo de plataforma [4] (Android, iOS, y demás). Por consiguiente, el usuario final podrá tener facilidad acceso e instalación de la aplicación sin restricción de sistema operativo, llegando así a una mayor cantidad de usuarios.

Muchos restaurantes, principalmente los pequeños negocios, no cuentan con un sistema que permita gestionar y organizar los pedidos y entregas diarias, empleando como recurso principal aplicaciones de mensajería secundarias y registros manuales. Este último requiere una mayor inversión de tiempo y recursos humanos, destinados a la organización, registro y despacho de comida, según la demanda existente.



En base a la problemática mencionada, se propone el diseño y desarrollo de una app móvil para los procesos de gestión y administración de pedidos en la empresa Pollo Broster D la Zona, siendo el objetivo principal la optimización en procesos de pedidos y entregas del restaurante Pollo Broster de la Zona y sus sucursales. Obteniendo como principales resultados la disminución en el tiempo de espera para efectuar el pedido, y sobre todo el incremento en los montos de ventas diarias.

Para llevar a cabo el desarrollo de la app se ha considerado las actividades con mayor relevancia dentro de la metodología de diseño y desarrollo de aplicaciones móviles, MMS. Por otro lado, para la implementación se consideró un modelo de arquitectura cliente-servidor, empleando Node JS y Firebase para el desarrollo del back-end, y los frameworks de React Native y Expo para el front-end. Las tecnologías seleccionadas permiten el control y administración de datos en tiempo real, facilitando la recepción de pedidos y actualización del estado de los mismo, así como la ejecución multiplataforma.

El resto del documento está dividido según la siguiente estructura:

**Capítulo I:** Se enfoca en el análisis respectivo al objeto de estudio, mediante la descripción del ámbito de aplicación y entorno a implementar la propuesta, así como la elicitación, análisis y justificación de requerimientos a satisfacer mediante la aplicación móvil.

**Capítulo II:** Se detalla el proceso de diseño y desarrollo del prototipo, incluyendo la descripción teórica de la arquitectura, tecnologías, y herramientas a utilizar durante dicho proceso. Adicionalmente, se definen los objetivos y se desarrollan las fases de la metodología de desarrollo MMS, incluyendo el desarrollo del prototipo.

**Capítulo III:** El último capítulo se enfoca en la evaluación de la aplicación móvil desarrollada, considerando las actividades correspondientes para el proyecto según la fase de prueba de la metodología MMS. De igual manera, las conclusiones y recomendaciones obtenidas durante el proceso de desarrollo son detalladas respectivamente.

## **1 CAPÍTULO I. DIAGNÓSTICO DE NECESIDADES Y REQUERIMIENTOS.**

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

“Pollos Broster D la Zona”, es un restaurante que ofrece servicio de comida tanto para consumir en el local o llevar a domicilio. Al igual que muchos restaurantes, el proceso de recepción de pedidos se realizan mediante llamadas telefónicas o mediante mensajería en redes sociales, lo cual limita al cliente en el conocimiento del menú actual, y requiere de mayor esfuerzo por parte del cliente y del empleado.

En la actualidad, los pedidos a domicilio, principalmente en el ámbito de comidas, han incrementado debido a varios factores como lo son el clima, comodidad, o disponibilidad de los clientes. Los clientes buscan la forma de agilizar la compra de productos, evitando emplear su tiempo en largas filas, o en la movilidad hacia el local para adquirir la comida. Como solución, varios negocios e inclusive pequeños emprendimientos, han optado por la adquisición de aplicaciones móviles, las cuales optimicen el proceso de compra y venta entre el cliente y el restaurante, dando facilidad al comensal de seleccionar y comprar el platillo que desee mediante un clic.

Pollo Broster D la Zona, emplea las redes sociales tanto como medio publicitario, como gestor de pedidos, sin embargo, el esfuerzo y tiempo requerido para los mismos son inmensos. En muchos casos, el cliente no conoce las opciones del menú, ni precios ofrecidos por el restaurante, por lo que regularmente la mayor parte de los mensajes, e inclusive llamadas telefónicas son destinadas a la explicación del menú, platillos y promociones disponibles. Debido a que el restaurante cuenta con algunas sucursales, la información debe ser detallada acorde a la ubicación del comensal.

En este punto, es importante considerar algunos escenarios, como son: (1) El cliente desea adquirir un platillo que no se encuentra en la sucursal correspondiente al número que llamó, por lo que es necesario que se contacte con el número de otra sucursal para conocer la disponibilidad del mismo. En dicho escenario, el cliente emplea una gran cantidad de tiempo, y la empresa dependiendo del cliente puede perder un pedido.

(2) El cliente solicita información del menú mediante redes sociales o vía telefónica, luego del vendedor a cargo detallar los precios y promociones, el cliente opta por no comprar ningún platillo, o adquirirlo otro día. Bajo el escenario propuesto, el restaurante invierte una gran cantidad de tiempo atendiendo al cliente, y posiblemente perdiendo alguna posible venta durante ese lapso de tiempo. Por otro lado, el cliente puede no recordar todos los platos y precios descritos, por lo que será necesaria una nueva comunicación con el vendedor.

(3) El cliente visualiza el menú y precios mediante publicaciones en redes sociales, sin embargo, las publicaciones son de fechas anteriores, por lo que los precios pueden estar desactualizados o el producto ya no se encuentra disponible. En este escenario, el cliente se encontrará decepcionado debido a que el precio es mayor, además de perder la posibilidad de seleccionar una opción acorde a sus necesidades.

Según lo mencionado, se propone el diseño de una aplicación móvil que permita la gestión y administración de pedidos en la compra y venta de comida del restaurante Pollos D la Zona, administrando de igual manera la información relacionada al menú, precios y promociones actualizadas. La propuesta contará con dos aplicaciones móviles, la principal será destinada al cliente, mediante la cual el comensal tendrá facilidad para visualizar el menú y las promociones actualizadas aún vigentes, y efectuar la orden del pedido desde su domicilio. La segunda será destinada al negocio, mediante la cual será posible la gestión de platillos al menú, gestión de promociones, y el principal la gestión de pedidos.

## **1.2 ESTABLECIMIENTOS DE REQUERIMIENTOS**

Para lograr diseñar una aplicación móvil eficiente para la gestión y administración de pedidos a domicilios, es fundamental identificar los requerimientos que se deben satisfacer, tanto propios de la empresa dentro del proceso de negocio, así como las funcionalidades y requisitos mínimos que los negocios comúnmente necesitan para ofrecer venta en línea, y envío a domicilio de manera eficiente.

La elicitación de requerimientos es llevada a cabo mediante una entrevista directa con los propietarios y/o encargados del negocio, así como la observación y análisis del proceso empleado para la gestión de pedidos, y ventas a domicilio (Anexo A). Los requerimientos identificados son detallados a profundidad en el Anexo B, identificando los que son para la app del cliente y del negocio respectivamente.

Debido a las necesidades dentro del negocio, los requerimientos principales inmersos en la app del negocio son los siguientes:

RF-01: Autenticación mediante usuario y contraseña.

- La app debe contar con un sistema de autenticación para que se muestren las funcionalidades correspondientes al rol del usuario que ingresó al sistema. Además, de ser validado mediante usuario (número de cédula) y contraseña.

RF-02: Gestión de usuarios.

- La aplicación contará con los siguientes roles para la administración efectiva de la app: (1) Administrador, que es el encargado de agregar, modificar y eliminar los registros en los diferentes módulos, así como realizar las demás funcionalidades del sistema (2) Vendedor, que es el encargado de receptar y validar los pedidos, así como encargado de actualizar el estado de los pedidos y enviarlos según corresponda.
- El administrador será encargado de gestionar los usuarios del sistema, crear uno nuevo, modificar la información personal del mismo, asignar rol, o eliminar un usuario existente.
- Los permisos establecidos para cada rol son visualizados por el usuario logeado, los roles y permisos no se pueden eliminar, modificar ni agregar.

RF-03: Gestión de menú.

- Se permitirá crear y eliminar secciones o categorías.

- Se permitirá agregar productos a las secciones o categorías creadas.

#### RF-04: Gestión de productos y promociones.

- El administrador podrá ingresar un nuevo producto, modificar o eliminar uno existente.
- Cada producto tiene un campo de visibilidad y disponibilidad, mediante el cual se indica si el producto se muestra u oculta en el menú, o si se encuentra disponible o agotado, respectivamente.
- Se podrán crear, modificar, eliminar, o cambiar el estado de las promociones, a las cuales se les puede agregar diferentes productos (combos).

#### RF-05: Gestión de pedidos.

- La aplicación emitirá una notificación para un nuevo pedido, el cual será aceptado o rechazado por el vendedor (tiempo límite para aceptación 15 min).
- Los pedidos mantienen un estado con el cual se indican si se encuentran pendientes, atendidos, enviados o entregados.
- El pedido debe detallar la información detallada de la orden, al igual que los datos del cliente, y el valor total a cancelar.
- El pedido debe contar con un campo de pagado o pendiente de pago, el cual puede ser actualizado por el vendedor. Los pagos se realizan en efectivo al momento de la entrega.

#### RF-06: Gestión de entrega.

- El vendedor realiza el envío del pedido actualizando el estado de atendido a enviado.
- La aplicación contará con un listado de pedidos pendientes de entrega.

#### RF-07: Gestión de sucursales.

- El administrador podrá ingresar, modificar y eliminar las sucursales.

- A cada producto, menú o promoción se le puede asignar o eliminar la sucursal a la que pertenece.
- Se mostrará un análisis de las ventas vendidas durante el día, y durante los últimos 7 días, por sucursales.

Por otro lado, la app del cliente debe satisfacer los siguientes requerimientos:

#### RF-02: Gestión de usuarios

- El cliente debe poder registrarse mediante una cuenta con usuario y contraseña, modificar sus datos personales, y eliminar su cuenta.

#### RF-08: Gestión de orden de compra.

- El cliente podrá agregar el producto(s), menú o promoción que desee al carrito de compras.
- El cliente generará la compra mediante la validación de detalle del pedido, y selección de la ubicación de entrega.
- El cliente podrá revisar el estado de su pedido, así como la información respectiva del mismo.

#### RF-9: Gestión de ubicaciones.

- El usuario podrá agregar, modificar, o eliminar una o varias ubicaciones de envío.

#### RF-10: Multiplataforma

- Las aplicaciones deben ser accesibles para cualquier plataforma móvil, sin variación de contenido o funcionalidades.

### **1.3 JUSTIFICACIÓN DE REQUERIMIENTOS A SATISFACER.**

La optimización del proceso del negocio, y la reducción del tiempo empleado por el cliente mediante las ventas en línea, ha incrementado exponencialmente debido al uso constante de dispositivos móviles y la facilidad que conlleva la virtualización del proceso de pedido de comidas a domicilio. Es por ello, que el diseño de una aplicación móvil es un recurso fundamental para la empresa Pollo Broster D La Zona, de manera que los pedidos y órdenes puedan ser

organizados y gestionados de manera rápida y precisa generando mayores ingresos diarios.

Sin embargo, el diseño de la aplicación conlleva la inclusión de diversos requerimientos vinculados, como son la gestión de usuarios, clientes, y productos.

El proceso de venta en línea conlleva dos procesos, un proceso realizado por el restaurante (venta y gestión del pedido), y otro proceso realizado por el cliente (compra y orden de compra). Si ambos procesos son diseñados en la misma aplicación, los recursos que empleará dicha aplicación serán desventajosos. En base a lo analizado, se requiere el desarrollo de dos aplicaciones móviles, una designado al cliente para la optimización del proceso de ordenar o comprar un pedido, y otro designado al restaurante para que administre los productos, y gestione los pedidos respectivamente.

Como todo sistema informático, la gestión de usuarios es fundamental, incluyendo los roles y permisos que cada uno posee en empresas jerarquizadas o con áreas asignadas, el cual es el caso del restaurante Pollo Broster D La Zona. Es por ello, que se designan los permisos respectivos según el rol que cumpla el empleado dentro de la organización.

Los pedidos (órdenes de compra) deben ser flexibles, es decir, el estado de los mismos debe poder ser actualizado, de manera que se informe al cliente del proceso en que se encuentra su pedido, y el tiempo de entrega del mismo. De igual manera, brinda mayor organización al restaurante permitiendo conocer los pedidos pendientes y los ya entregados.

Uno de los módulos con los que debe contar el sistema es la gestión de direcciones, en la cual el cliente establecerá una lista de direcciones en donde será la entrega, esta funcionalidad reduce el tiempo del cliente al momento de la compra, así como brinda flexibilidad al cliente para realizar el pedido desde la ubicación en donde se encuentre.

El pago dentro de la (in-app purchase) ha tenido un gran auge y acogida por la mayoría de desarrolladores, debido a ello, también ha incrementado el número de víctimas de estafas o posibles hackeos, un gran ejemplo de ello es el ataque

de aplicaciones de iOS [5]. Por ello, el cliente, la empresa Pollo Broster D La Zona, ha solicitado que los pagos se realicen mediante efectivo, es decir cancelar al momento de la entrega. Este requerimiento brinda mayor facilidad al usuario, ya que no siempre cuentan con una cuenta bancaria, o un método de pago online. De esta manera, el rango de clientes es amplio y variado, y brinda seguridad al cliente, de cancelar al momento de recibirlo.

Un enfoque principal del presente trabajo de titulación, es la necesidad del modelo multiplataforma para desarrollo de aplicaciones de e-commerce. La fragmentación [6], es uno de los problemas más comunes que presenta el desarrollo de aplicaciones [7], debido a que los usuarios no utilizan una plataforma heterogénea, ocasionando en algunos procesos problemas con la portabilidad o usabilidad. Pese a que no se ha definido una clasificación de apps que requieran un modelo multifuncional, existen una diversidad de aplicaciones para las cuales se debería considerar fundamental, principalmente en aplicaciones educativas [8], de la salud [7], y de comercio [9], ya que no cuentan con usuarios finales con plataforma heterogénea.

## **2 CAPÍTULO II. DESARROLLO DEL PROTOTIPO**

### **2.1 DEFINICIÓN DEL PROTOTIPO TECNOLÓGICO.**

Se centra en la definición de la estructura empleada para diseñar y desarrollar la propuesta tecnológica, identificando las tecnologías y herramientas a emplear en el lado del cliente y el servidor, así como la arquitectura de la aplicación.

#### **2.1.1 Arquitectura del Software**

La arquitectura de software es un implemento utilizado para simplificar el diseño y el entendimiento de la construcción del sistema, empleado herramientas, vistas y relaciones [10], es por ello que se lo puede considerar como una guía de desarrollo. La arquitectura de software empleada para desarrollar la presente propuesta es la Arquitectura de 3 capas como se muestra en la Figura 1, centrando el diseño del mismo en la escalabilidad y la reusabilidad [11].



La arquitectura tres capas, consiste en la división de la estructura del software en tres partes, la capa de presentación o interfaces de usuario, la capa lógica o de servicios, y finalmente, la capa de datos [12]. Es de gran importancia que cada nivel cuente con las tecnologías correspondientes para lograr un gran funcionamiento de la aplicación, así como la independencia de funcionamiento, es decir poder incrementar módulos de desarrollo sin afectar el funcionamiento de los ya desarrollados.

El primer nivel, la **capa de presentación**, se encuentra del lado del cliente, y es visible para los usuarios finales a través de un navegador web. Dentro de esta capa se almacenan todas las interfaces de usuario, y se establecen las peticiones requeridas para cada funcionalidad mediante el uso de librerías y componentes de React Native.

El segundo nivel es la **capa lógica**, que es la encargada de recibir las peticiones de los usuarios según las funciones que deseen realizar en las interfaces. Dentro de ella se codifican las peticiones GET y SET para la base de datos respectivamente. Para lograr dicha estructuración se emplea Node.js. Finalmente, tenemos la **capa de datos**, la cual consiste en la conexión y peticiones a la base de datos en la nube, empleando Firebase como gestor de base de datos.

*Figura 1 Arquitectura del software*



**Fuente:** Elaboración Propia

### 2.1.2 Arquitectura del hardware

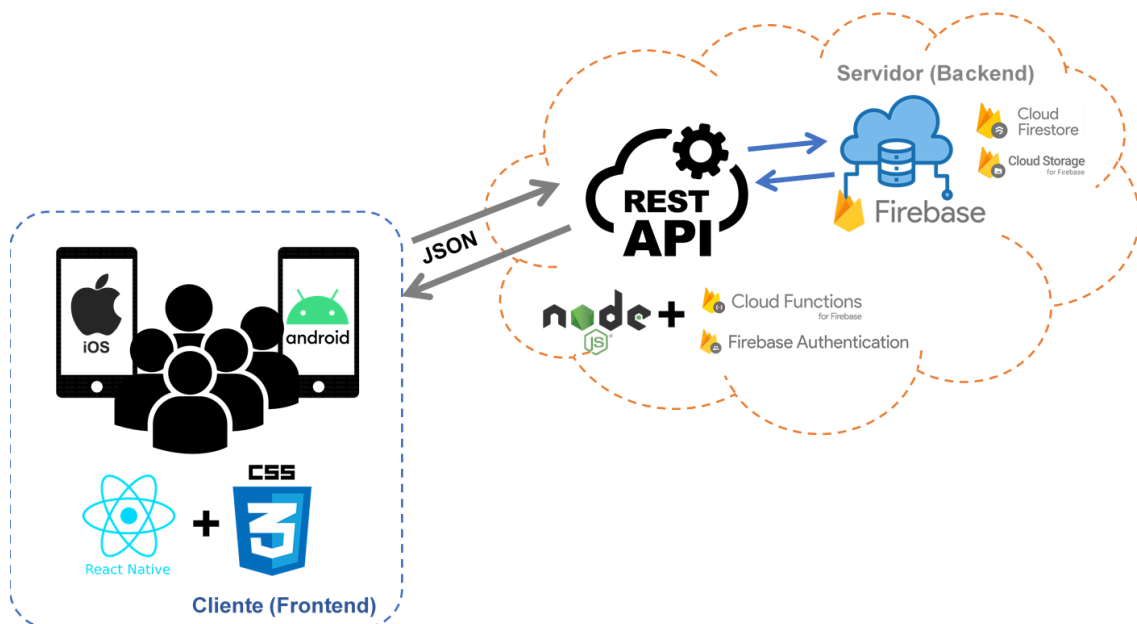
Por otro lado, la arquitectura del hardware consiste en las infraestructuras internas para que el sistema a desarrollar funcione de la manera esperada. Para ello se deben considerar varias características, entre las cuales se están el rendimiento de la aplicación, las funcionalidades que poseerá el sistema, el

tipo y tamaño de datos a almacenar, y por último la capacidad de procesamiento que manejará. En base a las consideraciones mencionadas, y según las necesidades de la aplicación propuesta, se determina que la arquitectura cliente-servidor es la más conveniente para un mejor funcionamiento de la app.

Esta arquitectura, es comúnmente empleada para desarrollo móvil [13] y web, debido a divide la carga de trabajo en dos partes y facilita la interacción entre ellos mediante un REST API, distribuyendo la programación en los conocidos frontend y backend [14].

Las tecnologías inmersas dentro del lado del cliente y del lado del servidor se distribuyen según las herramientas empleadas en la arquitectura de tres capas, considerando en el lado del cliente a la capa de presentación, y en el lado del servidor a la capa lógica, y de datos, como se muestra en la Figura 2.

*Figura 2 Arquitectura del hardware: cliente-servidor*



**Fuente:** Elaboración Propia

**Lado del cliente:** En el lado del cliente se codifica todos los recursos inmersos en la capa de presentación. Dentro de esta se encuentran las interfaces de usuario, componentes de IU (Interfaz de Usuario), y las librerías empleadas para mejorar la usabilidad y visibilidad de las funcionalidades de la app. En vista de que uno de los objetivos de la presente propuesta es la tecnología multiplataforma, se emplea React Native como lenguaje de programación para

las interfaces e interacciones con el usuario final, de manera que la app pueda ser lanzada e instalada por dispositivos Android e iOS. Para el diseño de la IU también se emplea la tecnología css.

**Lado del servidor:** Dentro de este se establece un REST-API, en el cual se encuentran almacenadas las peticiones a las bases de datos, así como los procesos internos a efectuar con los datos para la respuesta a las peticiones del cliente. Al emplear una bae de datos en la nube, la API solo conlleva sentencias para gestionar o recuperar los registros en la base de datos, y métodos o funciones (según corresponda) para procesar la información según la funcionalidad que sea solicitada.

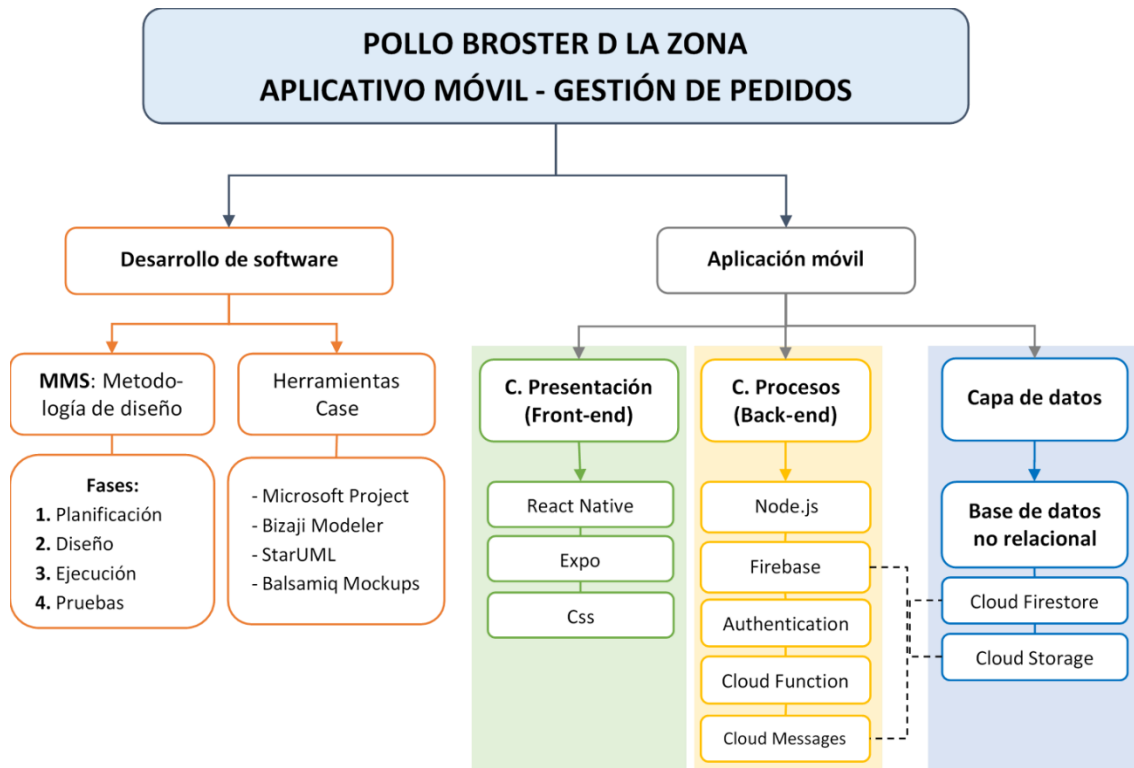
La protección y seguridad de los datos es un tema de suma importancia, y más aún cuando se tratan de datos sensibles como la información personal del cliente. Es por ello que se emplea el cifrado de datos y seguridad del lado del servidor, empleando Firebase Authentication, previniendo los posibles ciberataques para el robo de datos.

El almacenamiento de los datos será empleando Firebase como DBMS (database manager system), y almacenados en la nube utilizando la tecnología de Cloud Firestore y Cloud Storage.

## 2.2 FUNDAMENTACIÓN TEÓRICA DEL PROTOTIPO.

El siguiente apartado define teóricamente la metodología, las herramientas y tecnologías que se emplearán durante el proceso de desarrollo, y las cuales se desglosan en la Figura 3.

Figura 3 Estructura de fundamentación teórica del prototipo



Fuente: Elaboración Propia

### 2.2.1 Aplicaciones móviles.

Las aplicaciones móviles en la actualidad presentan una gran demanda y reto tanto para los clientes como para los empresarios y microempresarios. Desde la pandemia del COVID-19, la implementación de estrategias digitales en los negocios y micro negocios se ha incrementado a gran escala [15].

Pese que las restricciones de movilidad y aforo en locales a disminuido, los hábitos en los consumidores y los beneficios que la tecnología ha dejado a los negocios no han cambiado, principalmente en los negocios de comida. Es por ello que la necesidad de implementar aplicaciones móviles para gestionar la compra y venta de productos se ha vuelto imprescindible día a día.

Con ello, la complejidad requerimientos para el desarrollo de las aplicaciones móviles ha incrementado [16], llegando a considerarse un nuevo reto para los desarrolladores. Como resultado, el uso de metodologías de desarrollo tanto ágiles [17] como híbridas es un punto clave al momento de desarrollar las aplicaciones.

### **2.2.2 Metodología de desarrollo de software.**

Gracias al incremento del e-commerce en la gestión de compra y venta, el desarrollo de software móvil ha presenciado cambios. Entre ellos el uso de modelos ágiles de desarrollo, o híbridos según la disponibilidad y requerimientos del proyecto.

Para lograr un desarrollo de funcionalidades eficientes, y evitar que existan fallas durante la programación o puesta en marcha de la aplicación [18]. Para el proyecto se emplea la metodología MMS, la cual involucra actividades de un modelo ágil, junto a actividades de metodologías nativas de desarrollo móvil.

Una ventaja principal de las metodologías ágiles es la gestión del tiempo y procesos de manera más efectiva, para ello los procesos ágiles enfatizan mayormente en la identificación de tareas, establecimiento de equipos y reducción de tiempo para cada entregable, en vez de una planificación completa del proyecto [19].

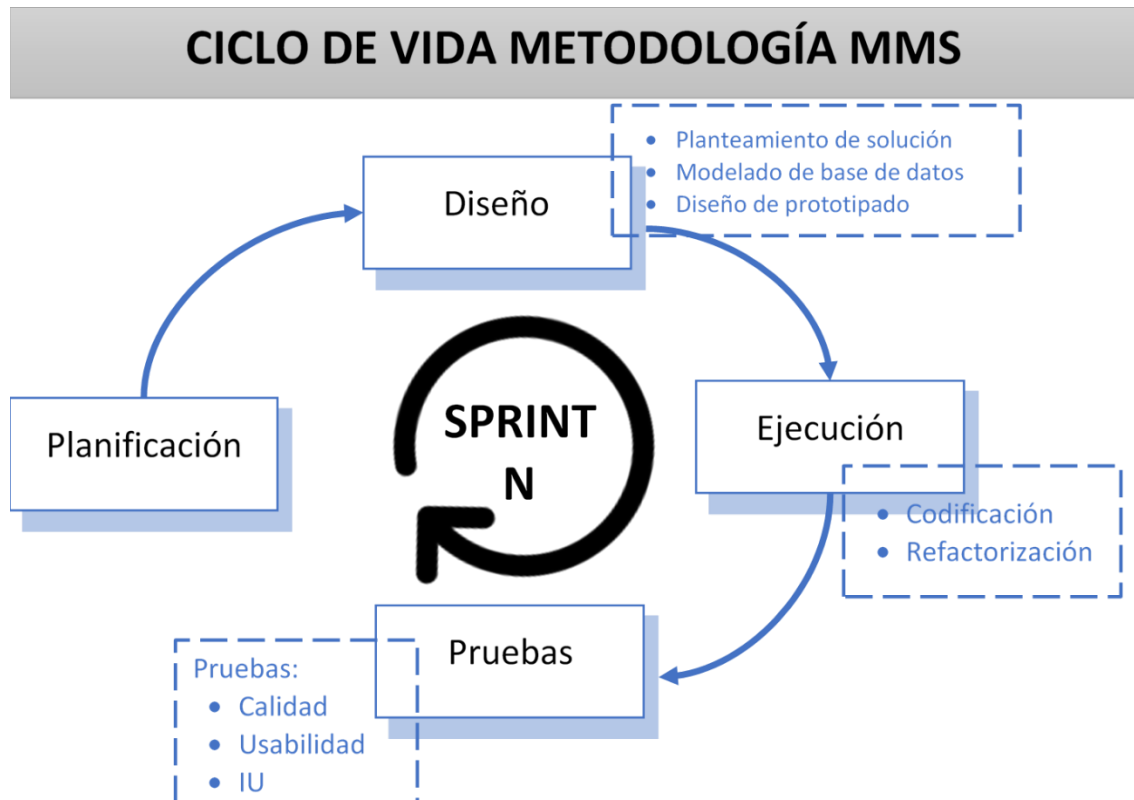
Se emplean algunas actividades de metodologías ágiles gracias a la facilidad y adaptabilidad ante el cambio de requerimientos del sistema, mediante la combinación del trabajo en equipo y la efectividad [20].

#### **2.2.2.1 Metodología MMS.**

La metodología “Modelo Mobile Sprint”, que por sus siglas es MMS, implementa cinco fases en su proceso de desarrollo. Implementa una serie de Sprints, en cortos ciclos de aproximadamente 2 a 4 semanas [21]. Este modelo de desarrollo enfatiza el uso de los sprints, y la definición de entregables durante cada ciclo, según se haya establecido en el cronograma de

actividades. El ciclo de vida de la metodología MMS y los involucrados se representan en la Figura 4.

Figura 4 MMS: Ciclo de vida



Fuente: Elaboración Propia

- **Fase de planificación.**

Esta primera fase, propone identificar las dimensiones del proyecto, el entorno y el modelo de negocio el cual se va a optimizar con la aplicación. Dentro de este punto se deben gestionar y planificar el uso de los recursos, el cronograma con las actividades, y el establecimiento de las actividades a cada miembro del equipo según sea necesario.

Dentro del proyecto se enfatizará en la gestión de requerimiento, stakeholders, alcance, modelado de negocio, y gestión de calidad.

- **Fase de diseño.**

Dentro de esta fase se implementan el análisis y los requerimientos del cliente, para la construcción y diseño de las interfaces y funcionalidades del sistema. Se deben establecer los módulos a diseñar, y las interfaces necesarias para cada una de ellas.

Dentro del proyecto se enfatiza el análisis de requerimientos, modelado de base de datos, y definición de IU de forma modular.

- **Fase de ejecución**

La fase de ejecución, consiste principalmente en la implementación de los prototipados diseñados en la fase anterior, involucra la etapa de codificación, diseño de pruebas y en caso de ser necesario la refactorización.

Dentro del presente proyecto se enfatiza la etapa de codificación y refactorización, sin embargo, dentro del documento principal solo se detallan los resultados de las codificaciones de interfaces pertinentes.

- **Fase de pruebas**

La última fase a implementar dentro del proyecto consiste en la ejecución de pruebas para analizar si los objetivos se han cumplido dentro de cada uno de los módulos. En esta fase se debe planificar y reacomodar los sprint según los resultados de cada prueba realizada.

Para la evaluación del prototipo se realizan pruebas de usabilidad y calidad según las métricas establecidas.

### **2.2.3 Herramientas CASE.**

Las herramientas CASE se emplean con la finalidad de facilitar el diseño y modelado de diagramas, de manera que el entendimiento entre los miembros del equipo y los diagramas y funcionalidades del aplicativo sea fluido y comprensible.

#### **2.2.3.1 Microsoft Project**

Microsoft Project es una herramienta muy útil dentro del proceso de planificación, sobre todo para el mundo empresarial. Esta herramienta brinda los recursos y funcionalidades necesarias para planificar y gestionar tanto los recursos como el tiempo, según las actividades y miembros de trabajo [22]. El uso de esta herramienta ayuda principalmente en la etapa de planificación, gracias a la cual se puede organizar, planificar y crear un cronograma de actividades según los requerimientos del proyecto.

### **2.2.3.2 Bizagi Modeler**

Bizagi Modeler es una herramienta de diagramación la cual provee herramientas y recursos necesarios para realizar procesos de negocios [23]. La simbología empleada y que es facilitada mediante la herramienta, como las flechas, rectángulos, pasarelas, rombos, entre otros, facilita el entendimiento de los procesos a los diferentes stakeholders del proyecto.

### **2.2.3.3 StarUML**

Es un software de diagramación y modelado, empleado principalmente para el diseño de diagramas UML, como son los casos de uso, diagrama de procesos, diagrama entidad-relación, entre otros.

La herramienta gratuita, cuenta con herramientas para la identificación de clases, relaciones, y otras semánticas que facilitan el entendimiento de los diagramas entre el grupo de trabajo [24]. Se empleará esta herramienta para realizar los diagramas de caso de uso dentro del proyecto.

### **2.2.3.4 Balsamiq Mockups**

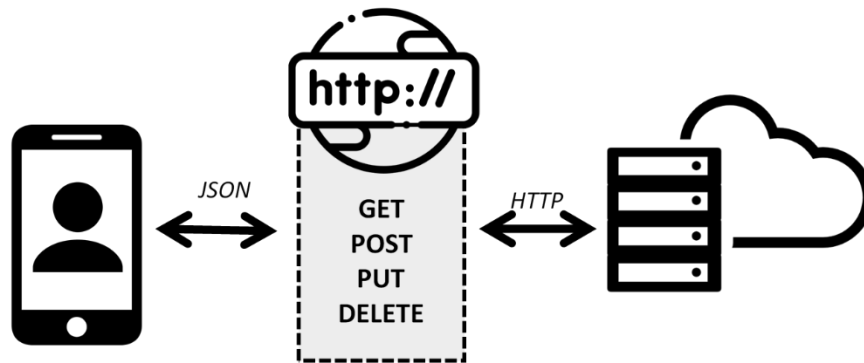
Permite el diseño de prototipos de interfaz de usuario, de manera que se bosqueje la idea principal de cada una de las interfaces de los módulos, necesarios para la fase de ejecución. La herramienta cuenta con una gran cantidad de componentes, brindándole al usuario una sensación similar a bosquejar con lápiz y papel, pero manteniendo un entendimiento del diseño entre los desarrolladores y analistas [25], [26].

## **2.2.4 Tecnologías Back-end.**

El Back-end es considerado como el conjunto de funcionalidades y procesos que son invisibles para el usuario, pero son imprescindibles para mantener el funcionamiento adecuado del sistema. Usualmente, se establecen en servidores remotos, sin embargo, debido a los requerimientos de la aplicación y fácil acceso a información, se empleará un backend almacenado en la nube, también conocido como Backend as a Service (BaaS) [27].



Figura 5 Tecnologías Backend



Fuente: Elaboración Propia

#### 2.2.4.1 Cloud Computing

La computación en la nube, o también conocida como Cloud Computing, es un tipo de tecnología versátil que implementa una estructura flexible y segura, en la cual el usuario paga por el servicio según la demanda y las necesidades [28]. Mediante esta tecnología, se logra describir cómo se logra almacenar, administrar y procesar los datos en servidores remotos (en la nube), en lugar de construir servidores físicos (computadoras) [29].

Para el desarrollo de los aplicativos se emplea la computación en la nube debido a sus numerosos beneficios, entre los cuales se incluyen la disminución de costos de dispositivos de almacenamiento, y facilidad en la elasticidad de la infraestructura de la base de datos [30].

#### 2.2.4.2 Firebase

Firebase es una de las plataformas de desarrollo de acceso abierto que brinda facilidad a los desarrolladores para diseñar y programar aplicaciones para plataformas IOS, Android y web [31]. Una de sus principales cualidades es que la base de datos está alojada en la nube (cloud computing) sin SQL y en tiempo real (real-time), facilitando la rápida recepción y recuperación de datos entre las aplicaciones [32].

Firebase es de muy fácil implementación, además de que posee una gran comunidad en línea, mediante la cual se puede desarrollar las funcionalidades de las aplicaciones de manera sencilla. Además, la

plataforma brinda una serie de servicios [33] útiles para los desarrolladores, de los cuales se utilizarán los siguientes:

- **Cloud Functions.**

Este servicio proporcionado por Firebase, permite ejecutar directamente en la nube el código back-end escrito en código JavaScript o TypeScript, de forma que se evita el uso y administración de servidores propios [34].

- **Authentication.**

Este servicio facilita la autenticación de los diversos usuarios mediante estándares, como son OAuth 2.0, numeros telefónicos, e inclusive mediante redes sociales [35].

- **Storage.**

Principalmente es un servicio orientado al almacenamiento multimedia, rápido y seguro para la carga y descarga de archivos de usuarios [34].

- **Cloud Messaging.**

Esta funcionalidad facilita el envío de mensajes o notificaciones dentro de múltiples plataformas de manera gratuita y segura a cada uno de los usuarios [36].

#### **2.2.4.3 Node.js**

Node.js permite la ejecución de código JavaScript multiplataforma, destinado principalmente para la programación del lado del servidor [37], mediante la ejecución asíncrona y basada en eventos. Gracias a las funcionalidades mencionadas, se logra obtener un código escalable y con múltiples conexiones simultáneamente [38].

Para el presente proyecto se diseñan las funcionalidades del servidor back-end mediante Node.js, y mediante el servicio Functions (de Firebase) se implementarán las funcionalidades en la nube, creando así una API REST que facilite esta comunicación.

#### **2.2.4.4 API REST**

Es un conjunto de restricciones que facilitan el envío y recepción de solicitudes HTTP, permitiendo de esta manera una comunicación segura entre el servidor, y varios clientes [39]. Se puede considerar a los API REST como una tecnología para lograr desarrollar aplicaciones multiplataforma. Las comunicaciones entre los clientes y el servidor se ejecutan mediante solicitudes HTTP, entre las cuales se emplean: GET, POST, PUT, DELETE [40].

#### **2.2.5 Base de Datos NoSQL.**

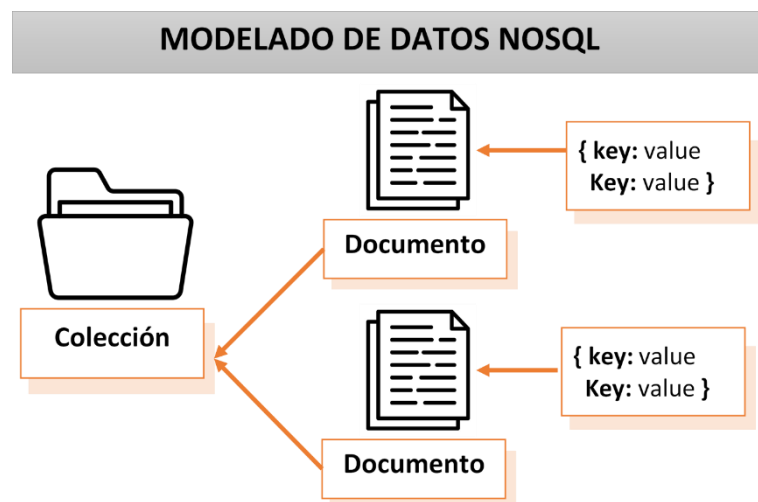
Las bases de datos NoSQL, son gestores de bases de datos no relacionales, empleadas principalmente para almacenar un gran volumen de información no estructura [41]. Una de las ventajas principales de emplear este modelo para el almacenamiento es que permite escalabilidad horizontal, y no mantiene restricciones ni relaciones estrictas entre tablas [42]. Cloud Firestore es una base de datos en la nube que implementa NoSQL para la administración de sus datos almacenados.

##### **2.2.5.1 Cloud Firestore.**

Cloud Firestore es la base de datos NoSQL para la gestión de grandes datos, principalmente orientada a documentos, organizados en colecciones [43]. Los datos en las bases de datos NoSQL son almacenados con un id único, y por conjunto de clave-valor, que contiene la información y que no debe ser igual para cada uno de los registros como se presenta en la Figura 6.

Gracias a la facilidad de información no estructurada que permite almacenar la base de datos, se logra implementar una rápida búsqueda y almacenamiento de los registros de los clientes, ordenes, y demás funcionalidades del sistema.

Figura 6 Base de datos – Cloud Firestore

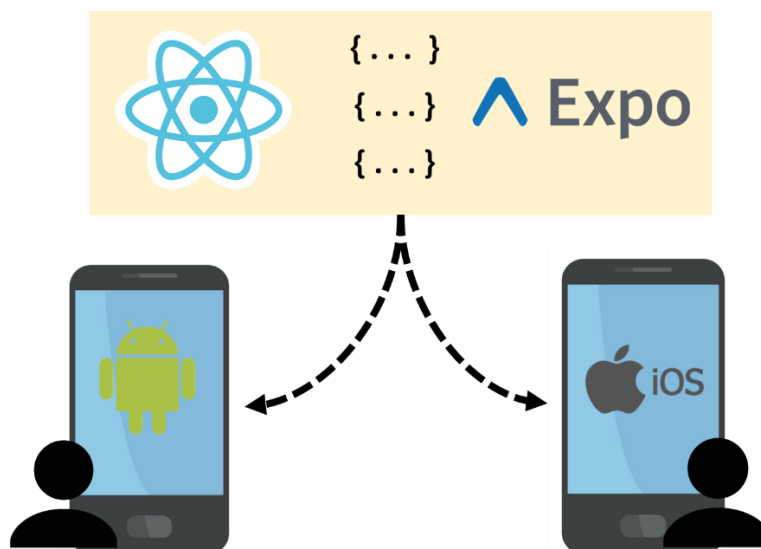


Fuente: Elaboración Propia

## 2.2.6 Tecnologías Front-end.

El front-end es considerado como la parte que se comunica con el cliente, es decir las interfaces de usuario que los clientes van a visualizar, y mediante las cuales realizarán las solicitudes para cada una de las funcionalidades de la app. Las interfaces a realizar se deben clasificar tanto para el lado del cliente, como para el lado de la compañía (vendedor), sin embargo, se emplean las mismas tecnologías para su programación como son React Native, y Expo.

Figura 7 Tecnologías Backend



Fuente: Elaboración Propia

### **2.2.6.1 React Native**

Es un framework de JavaScript, orientado netamente al desarrollo de aplicativos móviles, considerando las tecnologías web para su funcionamiento y ejecución en Android e iOS [44].

Mediante el uso de React Native, el desarrollador tiene facilidad de tiempo y desarrollo puesto que se puede codificar con conocimiento básico de JavaScript y React.js [45]. Además, de permitir el desarrollo multiplataforma [46], facilitando el desarrollo del proyecto propuesta en la presente investigación.

### **2.2.6.2 Expo**

Expo, es un framework específicamente empleado para el desarrollo de aplicaciones diseñadas con React Native. Gracias al framework se logra diseñar un despliegue de aplicaciones multiplataforma, tanto para Android como para iOS, sin la necesidad de codificar diferentes códigos fuentes [47].

## **2.3 OBJETIVOS DEL PROTOTIPO.**

### **2.3.1 Objetivo General**

Desarrollar una aplicación móvil para la gestión y administración de pedidos a domicilios para el restaurante Pollo Broster D La Zona mediante el uso de la metodología MMS y tecnologías de desarrollo híbrida.

### **2.3.2 Objetivos Específicos**

- Investigar las etapas de la metodología de diseño y desarrollo de aplicaciones móviles “MMS”, mediante la recopilación de información en fuente bibliográfica y casos de estudio.
- Analizar el proceso de gestión de pedidos y entrega a domicilio del restaurante Pollo Broster D La Zona para la elicitación de los requerimientos a satisfacer por el sistema.

- Diseñar los prototipos de interfaces de usuario necesarios en los procesos del cliente y del restaurante, mediante el uso de herramientas CASE para la satisfacción de los requerimientos del sistema.
- Desarrollar las aplicaciones móviles mediante el uso de tecnología híbrida, para su funcionamiento multiplataforma.

## **2.4 DISEÑO DEL PROTOTIPO.**

El proceso del diseño del prototipo se realiza en base a las actividades propuestas por la metodología MMS, considerando para ello las dos primeras fases, la fase de planificación y la fase de diseño. Dentro de ellas se definen los requerimientos e historias de usuarios, funcionalidades y módulos del sistema, así como los modelados y prototipado de las interfaces gráficas de usuario y base de datos.

### **2.4.1 FASE DE PLANIFICACIÓN.**

Dentro de esta fase se realiza el análisis tanto de los requerimientos como de los recursos del proyecto. Esta es la primera fase, y la más imprescindible antes de comenzar la ejecución o diseño de algún proyecto. Al involucrar procesos ágiles dentro del desarrollo de software, buscamos asegurar la escalabilidad y facilidad de cambios dentro del proyecto, es por ello, que en esta primera fase se analiza en entorno o situación actual de la empresa, el modelo de negocio bajo el cual trabajan, y los resultados esperados mediante el uso de la propuesta de optimización. Ya que se deben gestionar los recursos y actividades que se posee, es necesario analizar y planificar la gestión de stakeholders, alcance, cronograma (tiempo) y gestión de calidad.

#### **2.4.1.1 Entorno de la empresa.**

La empresa Pollo Broster de la Zona, es un negocio de venta de comida rápida, principalmente de papas fritas con pollo brosterizado. La empresa cuenta con algunas sucursales, en las cuales el menú y precios se mantienen iguales, sin embargo, la disponibilidad de los pedidos puede variar dependiendo del stock de materia y productos. Gracias al incremento

de servicios de encomienda, motorizados o delivery, las ventas a domicilio han incrementado de manera exponencial, siendo crucial para el buen funcionamiento del negocio la gestión de compras tanto en el local como a domicilio. Gestionar las ordenes mediante una aplicación, facilita el orden de los pedidos de cada cliente, generando un comprobante para el cliente y una orden nueva a cumplir para el negocio, así como un eficiente manejo de envíos y direcciones.

Los pedidos actualmente en la empresa se realizan mediante llamada telefónica, mensaje de texto, o mediante redes sociales, las cuales consumen el tiempo y recursos de la empresa, ya que para cada mensaje se debe brindar atención al cliente, incluyendo menú, precios, combos, o promociones del día. Además, que el proceso de finalización de una compra puede demorar demasiado o en ocasiones puede no llegar a darse.

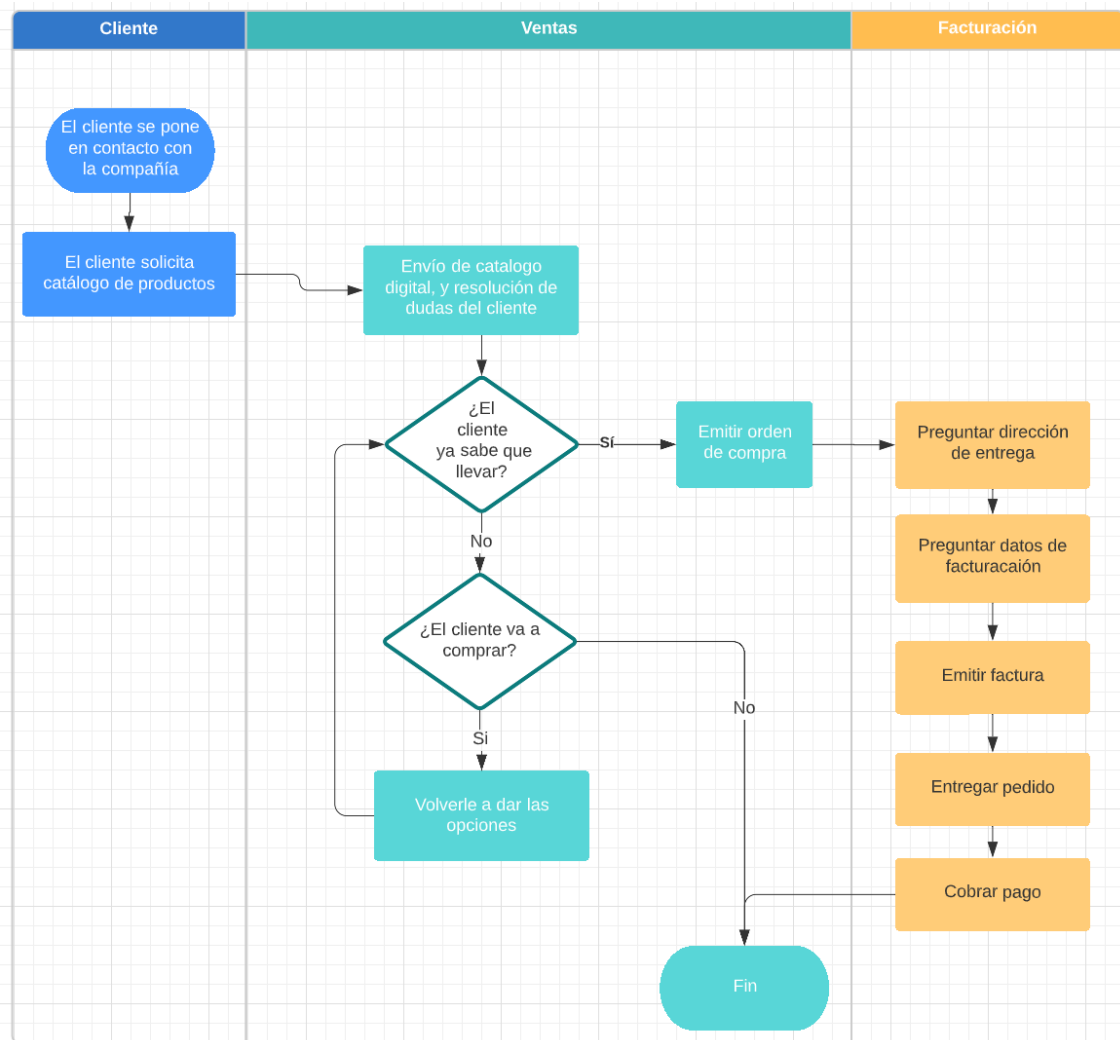
Es por ello que se propone como optimización la realización de un aplicativo móvil que permita gestionar los pedidos a domicilios de los clientes. Para ello es necesario implementar dos apps, la primera para la parte del cliente, mediante la cual se podrán visualizar los menús, precios, disponibilidad de platillos, e información básica sobre los diversos procesos de compra y adquisición a domicilio. Por otro lado, la app del negocio, el cual permite administrar el contenido, el menú y el precio de cada una de las comidas, para cada una de las sucursales.

#### **2.4.1.2 Modelo de Negocio.**

Un modelo de negocio se emplea para describir las actividades que presenten un gran valor en la empresa, y que intervengan de manera directa o indirecta dentro del proceso que se desea optimizar. Un punto clave del diseño y análisis del proceso de negocio es la verificación de alineación de las actividades desarrolladas con los objetivos de la empresa [48].

Para analizar el modelo de negocio, es necesario plasmar las actividades que realiza la empresa durante la gestión de pedidos, enfatizando los procesos de pedidos a domicilios, como se muestra en la Figura 8

**Figura 8** Modelo de negocio actual

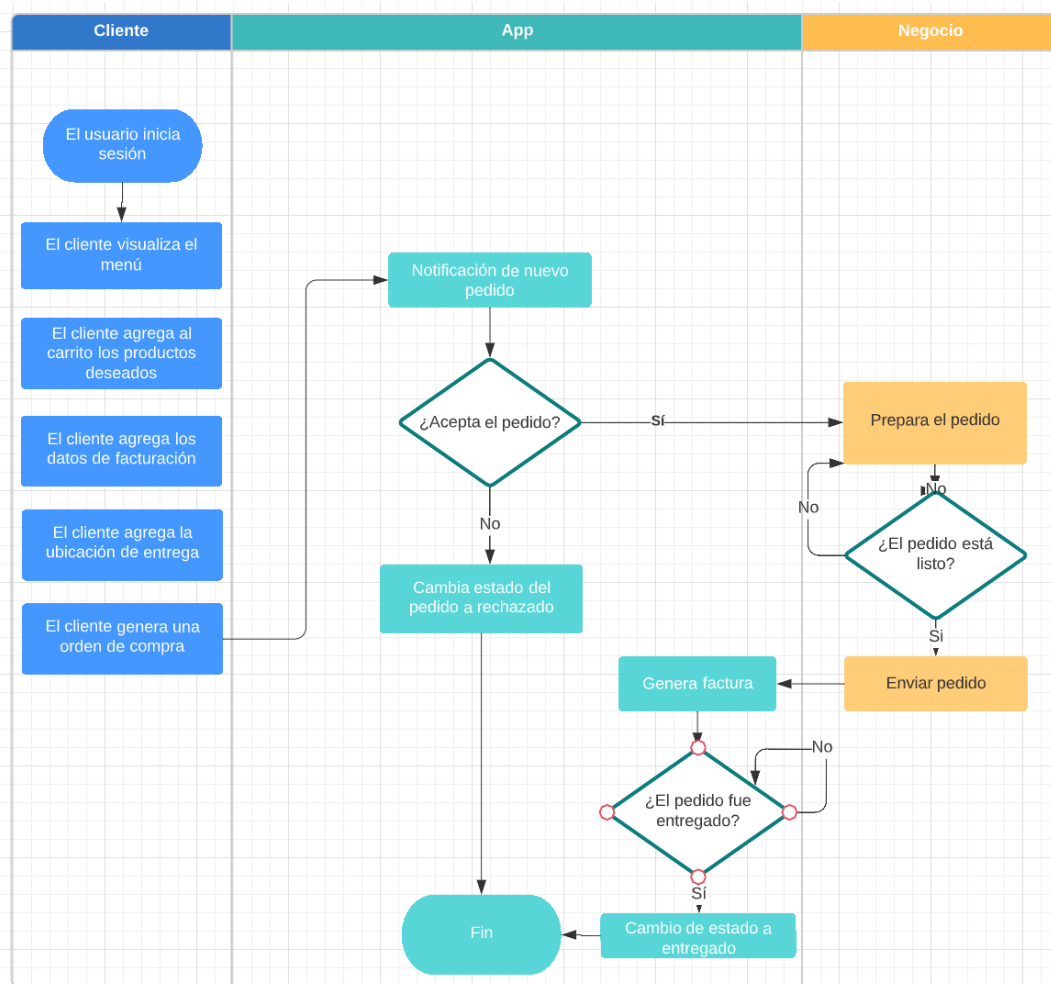


**Fuente:** Elaboración Propia

La descripción del proceso actual, comienza mediante la comunican del cliente con el vendedor, luego el cliente pasa por un proceso de visualización y selección del pedido (visualización del menú y opciones disponibles), una vez seleccionadas las opciones el vendedor de turno realiza la orden de pedido (recopilación personal del cliente), y envía el pedido para su preparación correspondiente. El pedido puede ser cancelado mediante transferencia, pago electrónico, o cancelar al momento de la entrega. Una vez que el pedido se encuentra listo se coordina el lugar de entrega, y se envía al cliente y ubicación respectiva.



**Figura 9** Modelo de negocio optimizado



**Fuente:** Elaboración Propia

En base al análisis, se propone un modelado de procesos de negocio optimizado como se muestra en la Figura 9, en el cual se interviene la comunicación del cliente con el vendedor por las funcionalidades de la aplicación. Este cambio disminuye el tiempo de ejecución del proceso además de ser eficiente y rápido para que cada cliente tenga conocimiento pleno, y se agilice el proceso de compra. Entre las opciones de la aplicación por el lado del cliente se encuentra la visualización del menú, visualización de pedidos, comprar, agregar ubicación, y la información personal. Entre las funcionalidades con mayor impacto se encuentra la compra, realizar pedido, y agregar ubicación.

## **Reglas de negocio.**

Como en todo proyecto y negocio, es fundamental que se especifiquen las reglas del negocio de manera que se agreguen detalles y restricciones correspondientes según las reglas que la empresa considera dentro del proceso. En base a lo mencionado, el presente proyecto considera las siguientes reglas de negocio.

- Los usuarios pueden realizar un compra y pedido de una comida, siempre y cuando hayan iniciado sesión correctamente.
- La orden de compra generada por el usuario, es enviada mediante mensaje (Whatsapp) al vendedor, para que pueda ser procesada según como corresponda.
- El cliente (usuario) y el repartidor (delivery) deben mantener la ubicación activada para realizar un rastreo y envío adecuado.

### **2.4.1.3 Resultados esperados.**

Es imprescindible que se establezcan y delimiten cuales son los resultados esperados dentro de un proyecto de software, y más aún cuando se optimiza el modelo de negocio. Los resultados deben estar vinculados estrechamente a los objetivos del proyecto, de esta manera se logran establecer los resultados, y los beneficios (tangibles e intangibles).

Entre los resultados visibles esperados dentro del proceso de negocio, se encuentra la generación de órdenes de pedidos a domicilio de forma ordenada y rápida, incrementando tanto el número de pedidos, como el monto de ingresos en ventas. Además, se espera que el cliente tenga mayor satisfacción por el servicio brindado, además de poseer pleno conocimiento de los precios y platillos disponibles, dando la sensación de estar en el local. Por otro lado, se espera que con la implementación disminuya el tiempo de realización de un pedido a domicilio, así como el uso del personal en actividades más necesarias.

Algunos de los **beneficios tangibles** que se espera obtener con la propuesta son:

- Incremento de ventas diarias.
- Gestión eficiente del proceso de ventas y entregas a domicilio.

Algunos de los **beneficios intangibles** que se espera obtener con la propuesta son:

- Disminución de uso de recursos tanto de tiempo como de personal en procesos de pedidos a domicilio.
- Facilidad y satisfacción por parte del usuario.

#### 2.4.1.4 Gestión de Stakeholders.

Los stakeholders dentro de los procesos ágiles tienen mayor importancia durante todo el proceso de desarrollo. Ya que cumplen con el rol de decidir los requerimientos y aceptar los entregables de cada uno de los sprints. Los interesados dentro del proyecto se detallan en la Tabla 1. Aunque dentro de la metodología MMS la formación de equipos es una de las actividades inmersas en esta etapa, no se la realizará dentro del proyecto. Como el grupo de desarrollo (y del proyecto en general) se encuentra conformado por una sola persona (el autor), este será el encargado de la realización de todas las actividades.

**Tabla 1** Matriz de Stakeholders

MATRIZ DE STAKEHOLDERS		
Nombres	Rol	Responsabilidad
Jaime Guamán	Gerente	Definir los requerimientos de la aplicación. Revisión y análisis de entregables.
Julio Adrián Campoverde	Director de proyecto. Desarrollador	Desarrollo del proyecto. Diseño e implementación del software. Pruebas y lanzamiento. Revisión de entregables.

**Fuente:** Elaboración Propia

De igual forma que se identifican los stakeholders, es imprescindible que se identifiquen los tipos de usuarios que emplearán las aplicaciones móviles. Debido a los requerimientos y funcionalidades del proyecto, las aplicaciones

son destinadas a operarse según los siguientes usuarios del sistema: Administrador del negocio, y clientes.

Los administradores del negocio tienen acceso a la aplicación del lado del negocio, permitiendo realizar diversas actividades para gestionar los platillos y precios, como se visualiza en la Tabla 2. Para poder obtener acceso a las diversas funcionalidades que posee la aplicación administrativa, el usuario debe ingresar y validar los campos mediante usuario y contraseña, y según los roles que le fueron asignados, tendrá acceso a las funcionalidades del mismo.

**Tabla 2** Usuario 1 del sistema

<b>Usuario:</b>	Administrador
<b>Tipo de usuario:</b>	Principal
<b>Nivel:</b>	Primer nivel.
<b>Tipo de interacción:</b>	Directa – Total
<b>Actividades:</b>	<ul style="list-style-type: none"> <li>• Registro de productos</li> <li>• Registro de precios</li> <li>• Gestión de secciones y productos del menú.</li> <li>• Gestión de pedidos</li> <li>• Dashboard informativo</li> </ul>
<b>Contenido de interacción:</b>	Aplicación móvil del vendedor
<b>Habilidades:</b>	Uso de tecnologías móviles

**Fuente:** Elaboración Propia

Por otro lado, los usuarios tipos *Clientes* son los encargados de acceder a la aplicación por parte del comprador, permitiendo log-in, visualización, agregar al carrito y compra, como se detalla en la

Tabla 3. Cada cliente para poder acceder a las funcionalidades de la aplicación deberá iniciar sesión en caso de contar con un usuario; en caso de no contar con las credenciales necesarias, el cliente deberá crear un nuevo usuario. Una vez validado el cliente mediante usuario y contraseña, podrá acceder a todas las funcionalidades que ofrece la app, como son visualización de menú, agregar al carrito, edición de perfil de usuario y ubicaciones de entrega, y comprar.

**Tabla 3** Usuario 2 del sistema

<b>Usuario:</b>	Cliente - comprador
<b>Tipo de usuario:</b>	Principal
<b>Nivel:</b>	Primer nivel.
<b>Tipo de interacción:</b>	Directa – Total
<b>Actividades:</b>	<ul style="list-style-type: none"><li>• Registrarse</li><li>• Login</li><li>• Visualización del menú</li><li>• Selección o agregar al carrito</li><li>• Compra</li><li>• Gestión de información personal</li><li>• Agregar ubicaciones.</li></ul>
<b>Contenido de interacción:</b>	Aplicación móvil del cliente
<b>Habilidades:</b>	Uso de tecnologías móviles

**Fuente:** Elaboración Propia

#### **2.4.1.5 Gestión de Alcance y Requerimientos.**

La gestión de alcance y requerimiento se centra en el análisis de las funcionalidades que la propuesta (apps) desempeñaran, incluyendo no solo funciones que puede realizar, sino también las limitaciones y restricciones que el sistema tendrá. El alcance definido debe estar enfocado a los objetivos planteados, así como en los requerimientos funcionales y no funcionales definidos por el cliente. Un punto clave dentro de este proceso es el análisis de requerimientos, dentro del presente proyecto se considerarán los procesos de elicitación de requerimientos funcionales y no funcionales, y la definición de historias de usuario, las cuales serán empleadas durante el establecimiento de las actividades para cada interesado.

#### **2.4.1.5.1 Alcance del proyecto**

En base al estudio realizado para el diseño de la aplicación se establecen de forma general las siguientes funcionalidades con las que contará la aplicación, en orden a sus prioridades y restricciones. Las mismas serán analizadas y posteriormente detalladas como requerimientos funcionales y no funcionales respectivamente.

En cuanto a las funcionalidades del sistema se especifican las siguientes:

- Registro de nuevos usuarios directamente desde la aplicación, solicitando la información básica para el perfil de usuario.
- Ingreso y validación de usuarios, mediante sistema de autenticación por credenciales.
- La gestión del menú permitirá las funcionalidades básicas de administración, crear nuevo, modificar, u eliminar categorías o secciones.
- Visualización y edición de perfil de usuario, sin restricción por el tipo de usuario o rol al que pertenezca. Se podrá visualizar y modificar la información respectiva.
- Para lograr registrar un producto (plato), se deben completar los campos obligatorios como foto, descripción, y precio.
- Los datos referentes a cada producto pueden ser eliminada, o en su defecto modificada.
- La disponibilidad de cada producto, se encontrará vinculado con un estado para cada sucursal.
- Cada sucursal puede mantener un menú independiente, o en su defecto mantener un estado de disponibilidad independiente.
- Notificaciones en cuanto a la llegada de un nuevo pedido para ser aceptado o rechazado respectivamente.
- Clasificación de pedidos mediante 4 categorías, pendientes, atendidos, enviados y entregados.

- La orden de pedido presentará información relevante del cliente, del pedido y la ubicación de entrega.
- Compra o pedido de una orden mediante un usuario de tipo cliente, el cual puede realizar una vez que haya registrado sus datos y su ubicación de manera correcta.

De igual forma, es necesario establecer las limitaciones que el sistema poseerá analizando los requerimientos iniciales, las funcionalidades a cumplir y los objetivos del sistema. En base a lo mencionado, las aplicaciones poseen las siguientes limitaciones:

- No existe restricción en la gestión de sucursales, siendo posible la administración del mismo por un mismo empleado.
- No es posible la carga de videos o archivos multimedia diferentes a las imágenes.
- Las aplicaciones no emiten reporte de ventas ni de ninguna índole.
- La aplicación no restringe la visualización de las sucursales según la ubicación actual del cliente, ni al momento del envío.

#### **2.4.1.5.2 Requerimientos Funcionales**

El proceso de elicitación de requerimientos involucra la interacción directa con los interesados, siendo el cliente uno de los stakeholders con mayor relevancia dentro el proyecto y de la metodología MMS. Es por ello que se empleó como técnica principal de elicitación la entrevista con el cliente, definiendo las necesidades y expectativas del sistema mediante el dialogo con las personas responsables del funcionamiento y procesos de venta del local.

Una vez analizado los resultados de la entrevista se procede a detallar los requerimientos funcionales, como se muestra en la Tabla 4.

**Tabla 4** Matriz general de requerimientos funcionales

REQUERIMIENTOS FUNCIONALES		
Código	Descripción de Requerimiento	App Designada
RF-1	Autenticación y recuperación de	Cliente,

REQUERIMIENTOS FUNCIONALES		
Código	Descripción de Requerimiento	App Designada
	contraseñas	administrador
RF-2	Gestión de usuarios	Cliente, administrador
RF-3	Gestión de menú y categorías	Administrador, Cliente
RF-4	Gestión de productos y promociones	Administración
RF-5	Gestión de pedidos	Administrador, Cliente
RF-6	Gestión de entrega	Administrador
RF-7	Gestión de sucursales	Administrador
RF-8	Gestión de ubicaciones	Cliente

**Fuente:** Elaboración Propia

Como se logra visualizar en la tabla anterior, la aplicación cuenta con diversas funcionalidades generales, que pueden llegar a denominarse módulos, sin embargo, cada uno conlleva un proceso único y diferente, implementando actividades dependiendo del módulo para el cual fueron creados. Es por ello que se realiza el respectivo desglose de requerimientos en el ANEXO A, detallando la especificación de RF (requerimientos funcionales) del sistema.

#### **2.4.1.5.3 Requerimientos No Funcionales**

Los requerimientos no funcionales (RNF), a diferencia de los requerimientos Funcionales, se centran en las especificaciones del producto, y características relacionadas con la ejecución, calidad, y seguridad, de manera que la ejecución de la app tenga resultados satisfactorios.

En la, se detallan los requerimientos no funcionales del sistema.

**Tabla 5** Matriz general de requerimientos no funcionales



REQUERIMIENTOS NO FUNCIONALES		
Código	Descripción de Requerimiento	App Designada
RNF-1	En caso de poseer roles específicos, las funcionalidades solo serán visibles según los roles y permisos asignados.	Administrador
RNF-2	Debe existir un usuario administrador como mínimo.	Administrador
RNF-3	La base de datos debe ser escalable, y admitir archivos multimedia (jpg o png)	Administrador
RNF-4	Los datos sensibles emplearan una encriptación para su almacenamiento.	Cliente, administrador
RNF-5	La interfaz de usuario debe ser amigable y fácil de usar	Cliente, administrador
RNF-6	Debe ser multiplataforma, disponible para dispositivos Android y iOS.	Administrador, cliente
RNF-7	Las interfaces deben ser adaptables a los diferentes tamaños de pantallas.	Administrador, cliente

**Fuente:** Elaboración Propia

#### **2.4.1.5.4 Historias de usuarios.**

Como herramienta para el análisis de los requerimientos funcionales se emplean las historias de usuarios. Mediante esta herramienta se logra analizar el requerimiento mediante las actividades que el usuario debe realizar. Algunos beneficios de realizar historias de usuario es el conocimiento específico de los pasos necesarios para efectuar dicho proceso, de manera que se analiza la complejidad, facilidad o coherencia de las mismas.

En la Tabla 6 se detallan los números de historia de usuario respectivos para cada requerimiento funcional. Cada una de las historias de usuario se detallan a profundidad en el ANEXO B.

**Tabla 6** Esquema general de historias de usuarios

<b>ESQUEMA DE HISTORIAS DE USUARIOS</b>	
<b>Requerimiento (RF)</b>	<b>Historia de usuario</b>
RF-1	HU-1: Inicio de sesión (Autenticación) HU-2: Recuperación de contraseñas
RF-2	HU-3: Registro de usuario HU-4: Gestión de perfil de usuario
RF-3	HU-5: Gestión de secciones HU-6: Gestión de menú
RF-4	HU-7: Gestión de productos y promociones
RF-5	HU-8: Notificaciones de pedidos HU-9: Realización de compra HU-10: Visualización de orden de compra
RF-6	HU-11: Gestión de entrega
RF-7	HU-12: Gestión de sucursales
RF-8:	HU-13: Gestión de ubicaciones

**Fuente:** Elaboración Propia

#### **2.4.1.6 Elaboración del EDT.**

La Estructura de Desglose de trabajo, por sus siglas EDT, ayuda a organizar e identificar las actividades de manera ordenada, respetando una jerarquía de niveles [49].

El ANEXO C presenta una estructura de desglose del trabajo para el presente proyecto, considerando las fases que involucra la metodología MMS. Se seleccionan las fases de la metodología con mayor relevancia dentro del proyecto, además de definir actividades y entregables inmersas en cada una de ellas según se consideran para el proyecto.

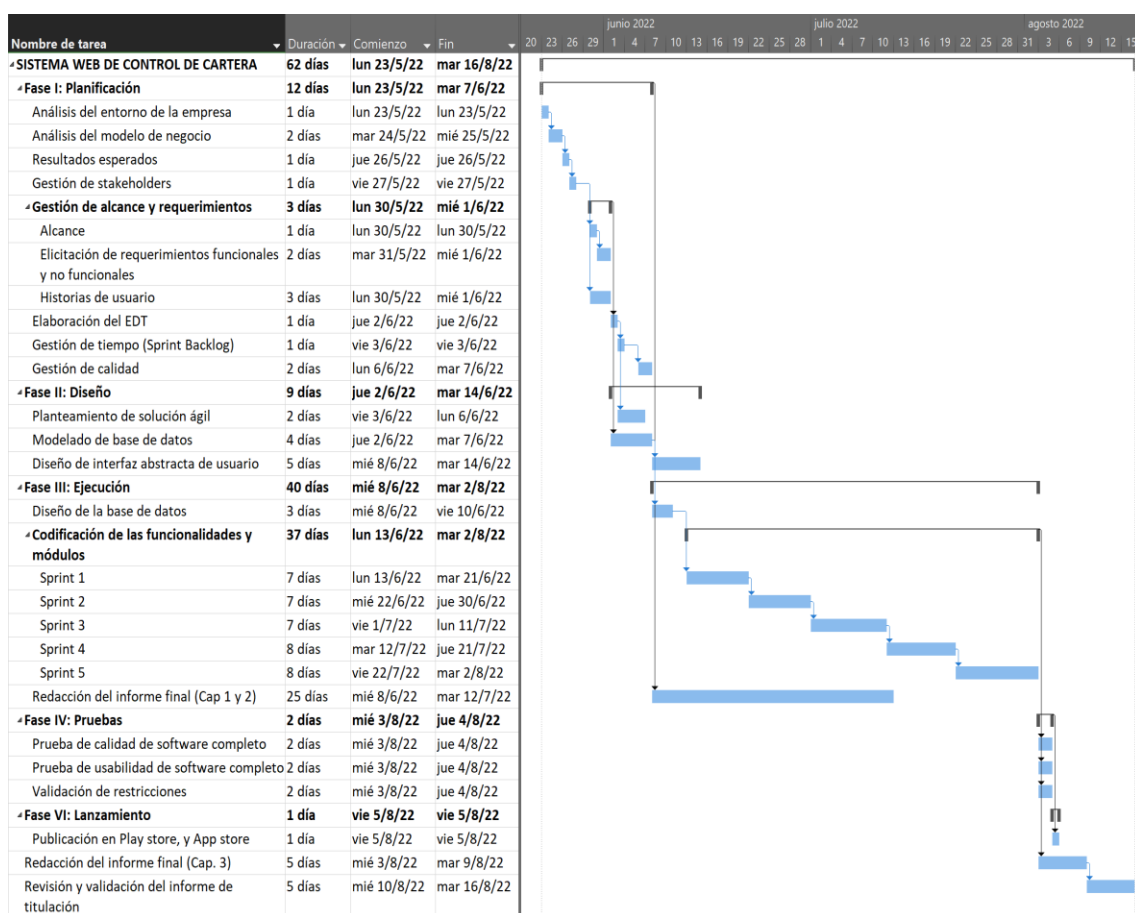
- Planificación
- Diseño
- Ejecución
- Pruebas

#### **2.4.1.7 Gestión del tiempo.**

Uno de los requerimientos dentro de todo proyecto es la gestión del tiempo del proyecto, de manera que todas las actividades se puedan realizar de manera continua, y con prevención de fallas. La metodología MMS, al contemplar el enfoque ágil dentro de su proceso, considera de suma importancia la gestión de tiempo. El proceso de cada actividad es denominado un sprint, y la planificación de actividades dentro de cada uno es vital para un óptimo proceso. La gestión de tiempo se dividirá en dos entregables, el primero consta de un cronograma de actividades, y el segundo del sprint Backlog. En este último se detallarán los sprints, junto con las actividades y entregables en cada uno de ellos.

Para el diseño del cronograma se emplea la herramienta Microsoft Project dando como resultado el calendario de actividades presentado en la Figura 10.

**Figura 10** Cronograma de actividades



Fuente: Elaboración Propia

- **Sprint Backlog.**

El Sprint Backlog detalla de forma clara cada uno de los sprint del sistema, así como las actividades que se deben realizar, los módulos a desarrollar, y lo más importante el tiempo que durará cada Spring, como se detalla en la Tabla 7. Dentro de cada Split se pondrá cual será el entregable a realizar, el cual será analizado con el cliente en la fecha asignada para la reunión.

**Tabla 7** Sprint Backlog

Sprint No	Tema	Actividad o historia de usuario	Fecha (Inicio- fin)	Fecha de reunión	Entregable
1	Gestión de usuarios	Diseño de interfaces de usuario HU-1, HU-3, HU-2, HU-4	22 - 24/6/22	21/06/22	Interfaces respectivas de la app

Sprint No	Tema	Actividad o historia de usuario	Fecha (Inicio- fin)	Fecha de reunión	Entregable
		Codificación de funcionalidades			administrador, y app cliente.
		Validación y pruebas de funcionalidades.			
		Modificación y corrección de errores			
2	Gestión de menú, productos y sucursales ( <i>app admin</i> )	Diseño de interfaces de usuario HU-5, HU-6, HU-7, HU-12	22/6/2022 – 30/6/2022	30/6/2022	Avance app administrador.
		Codificación de funcionalidades.			
		Validación y pruebas de funcionalidades.			
		Modificación y corrección de errores			
3	Gestión de compra. ( <i>app cliente</i> )	Diseño de interfaces de usuario HU-6, HU-7, HU-9, HU-10, HU-13.	1/7/2022 - 11/7/2022	11/7/22	Avance app cliente
		Codificación de funcionalidades.			
		Validación y pruebas de funcionalidades.			
		Modificación y corrección de errores			
4	Notificaciones y gestión de entrega ( <i>app admin, y app cliente</i> )	Diseño de interfaces de usuario HU-11, HU-13	12/7/2022 - 21/7/2022	21/7/2022	Avance app administrador y cliente
		Codificación de funcionalidades			
		Validación y pruebas de funcionalidades.			
		Modificación y corrección de errores			
5	Dashboard y demás funcionalidades	Diseño de interfaces de usuario para dashboard.	22/7/2022 – 2/8/2022	2/8/2022	Informe de prueba de funcionalidad. Interfaces de usuario.
		Codificación de funcionalidades			
		Validación y pruebas de todo el sistema.			

Sprint No	Tema	Actividad o historia de usuario	Fecha (Inicio- fin)	Fecha de reunión	Entregable
		Modificación y corrección de errores			

**Fuente:** Elaboración Propia

El Spring Backlog presentado en la tabla anterior, considerará los siguientes campos. El número del sprint, el tema con el cual será descrito, la actividad e historias de usuario que se desarrollarán dentro de ese periodo, la fecha de inicio y la fecha de finalización, la fecha de reunión (que es el último día del sprint), y finalmente el entregable que se socializará con el cliente.

#### 2.4.1.8 Gestión de calidad.

La fase de gestión de calidad dentro de un proyecto se emplea para prevenir errores en el proceso, además de desarrollar un producto de calidad, que logré satisfacer al cliente y sus necesidades. En las aplicaciones móviles, la calidad es un factor clave ya que tanto el cliente (administrador) como el usuario (comensales) deben percibir una buena experiencia al hacer uso de la app, permitiendo así que el uso de las misma incremente con el tiempo.

Bajo lo anterior descrito, se emplean los estándares de calidad de la métrica ISO/IEC 25010, la cual permite evaluar si el producto desarrollado cumple con las expectativas y requerimientos definidos. La métrica menciona propone evaluar 8 atributos del software, cada uno con sus respectivas métricas. Para evaluar la aplicación móvil a diseñar se definen en la Tabla 8 los atributos y métricas.

**Tabla 8** Sprint Backlog

EVALUACIÓN CALIDAD ISO/IEC 25010	
Atributos	Métricas

Adecuación funcional	<ul style="list-style-type: none"> <li>- Completitud</li> <li>- Corrección</li> <li>- Pertinencia</li> </ul>
Eficiencia del desempeño	<ul style="list-style-type: none"> <li>- Comportamiento</li> <li>- Uso de recursos</li> <li>- Capacidad</li> </ul>
Compatibilidad	<ul style="list-style-type: none"> <li>- Coexistencia</li> <li>- Inter-operabilidad</li> </ul>
Usabilidad	<ul style="list-style-type: none"> <li>- Aprendizaje fácil</li> <li>- Operabilidad</li> <li>- Aviso ante errores del usuario</li> <li>- Accesibilidad</li> <li>- Satisfacción</li> </ul>
Seguridad	<ul style="list-style-type: none"> <li>- Confidencialidad</li> <li>- Autenticidad</li> <li>- Integridad de datos</li> </ul>
Mantenibilidad	<ul style="list-style-type: none"> <li>- Modularidad</li> <li>- Reusabilidad</li> </ul>
Portabilidad	<ul style="list-style-type: none"> <li>- Adaptabilidad</li> <li>- Fácil de instalar</li> <li>- Capacidad de reemplazo</li> </ul>

**Fuente:** Elaboración Propia

## **2.4.2 FASE DE DISEÑO.**

El diseño se centra en la transformación y bosquejo de las funcionalidades, análisis, y requerimientos planteados en la etapa de planificación. Adicionalmente, se encuentra el diseño de la solución ágil, el modelado de base de datos e interfaces de usuario, tanto para la app del administrador (pollo de la zona) como la app del cliente (comensales).

### **2.4.2.1 Planteamiento de solución ágil.**

Para lograr mejorar el proceso de negocio actual de la empresa se han definido dos tipos de usuario, (1) el administrador, quien es el encargado de gestionar la información respectiva del negocio; y (2) el cliente, quienes

serán los comensales. Es por ello que se desarrollaran dos aplicaciones móviles, disminuyendo de esta manera el peso y velocidad de carga de la misma, y evitando que la seguridad de datos sea vulnerada, ya que ninguna funcionalidad administrativa estará presente en la aplicación del cliente.

En base a lo mencionado se procede a describir el proceso a efectuar en cada una de las aplicaciones, las cuales serán desarrolladas en 5 sprints de 7 a 8 días cada uno.

**App cliente:** Dentro de la app del cliente, el usuario podrá visualizar el menú y realizar la compra de los productos que desee, así como visualizar un listado de todos los pedidos efectuados. Dentro de la misma se administran los datos para la facturación, y la(s) dirección(es) para la entrega.

- El usuario podrá registrarse desde la pantalla principal de la app, en caso de no contar con las credenciales necesarias.
- El usuario podrá iniciar sesión, o recuperar su contraseña mediante correo electrónico en caso de no recordarla.
- El usuario visualizará una lista con el menú y opciones disponibles, así como podrá agregar al carrito los productos deseados para posteriormente comprarlos.
- Los datos de facturación y las ubicaciones se pueden agregar o modificar desde el menú principal. O en caso de no tener registrada ninguna, al momento de la compra se pueden completar los datos.
- El proceso de compra es sencillo e intuitivo, una vez verificados todos los datos, para poder efectuar la compra el cliente enviará la orden de pedido como mensaje al WhatsApp del restaurante el cuál procederá a aceptar la compra y realizar el envío pertinente.
- Existirá una sección de ayuda en la cual se podrá responder algunas preguntas frecuentes, así como la información sobre el local.

**App administrador:** Dentro de la app del administrador se podrá gestionar toda la información referente del lado de la empresa, la



modificación del menú, los productos y gestionar el proceso de pedido de manera rápida y eficiente.

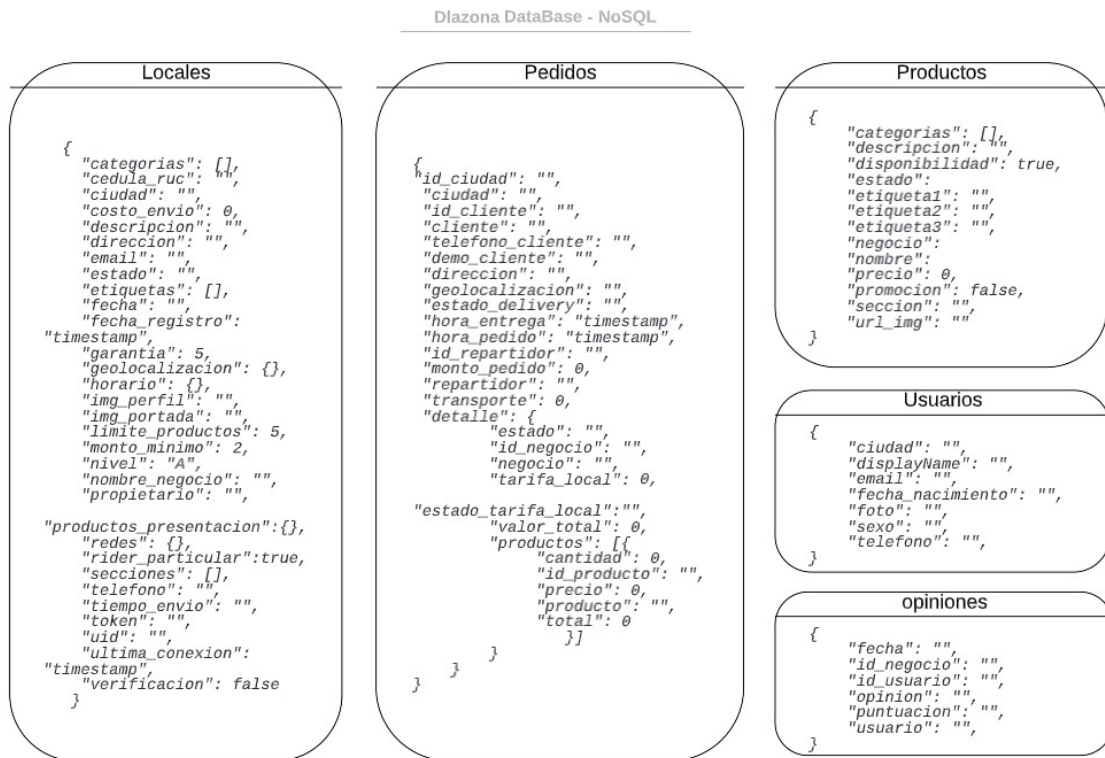
- Al igual que en la aplicación del cliente, el administrador podrá iniciar sesión o recordar contraseña.
- El administrador podrá gestionar los productos, teniendo la opción de marcarlos como promoción o productos normales. La disponibilidad de cada uno puede ser modificada mediante un cambio de estado. Dentro del menú el administrador podrá gestionar tanto productos como secciones.
- El administrador podrá visualizar la lista de los pedidos clasificados por 4 estados, Pendiente, preparación, enviados, y entregados. El listado incluye los detalles del pedido, así como la información de contacto del cliente. Una vez que se recibe el pedido se el administrador debe Aceptar o Rechazarlo.
- Una vez que se el administrador acepte el pedido, el estado del mismo pasará a preparación. Cuando el pedido se encuentre listo, se seleccionará la opción “Listo”, procediendo a asignar al motorizado encargado para enviar el pedido. Posteriormente el estado cambiara a enviado, para que al momento de ser recibido el estado se convierta a Entregado.
- Finalmente, el administrador podrá visualizar un Dashboard con las ganancias del negocio, incluyendo las ventas del día, y las estadísticas del último mes.

#### **2.4.2.2 Modelado de la base de datos.**

El almacenamiento de los datos se realiza en una base de datos NoSql, debido al tipo de datos que se desea almacenar (archivos multimedia y registros sin estructura definida). Como gestor de base de datos se seleccionó Firebase, creando documentos y registros adecuados para almacenar los datos necesarios. El esquema NoSql mostrado en la Figura 11 es el esquema empleado para la app del administrador, detallando los campos, y en caso de ser necesario documentos adicionales. Las relaciones

entre ellos se ven implícitas dentro del proceso o mediante identificadores similares.

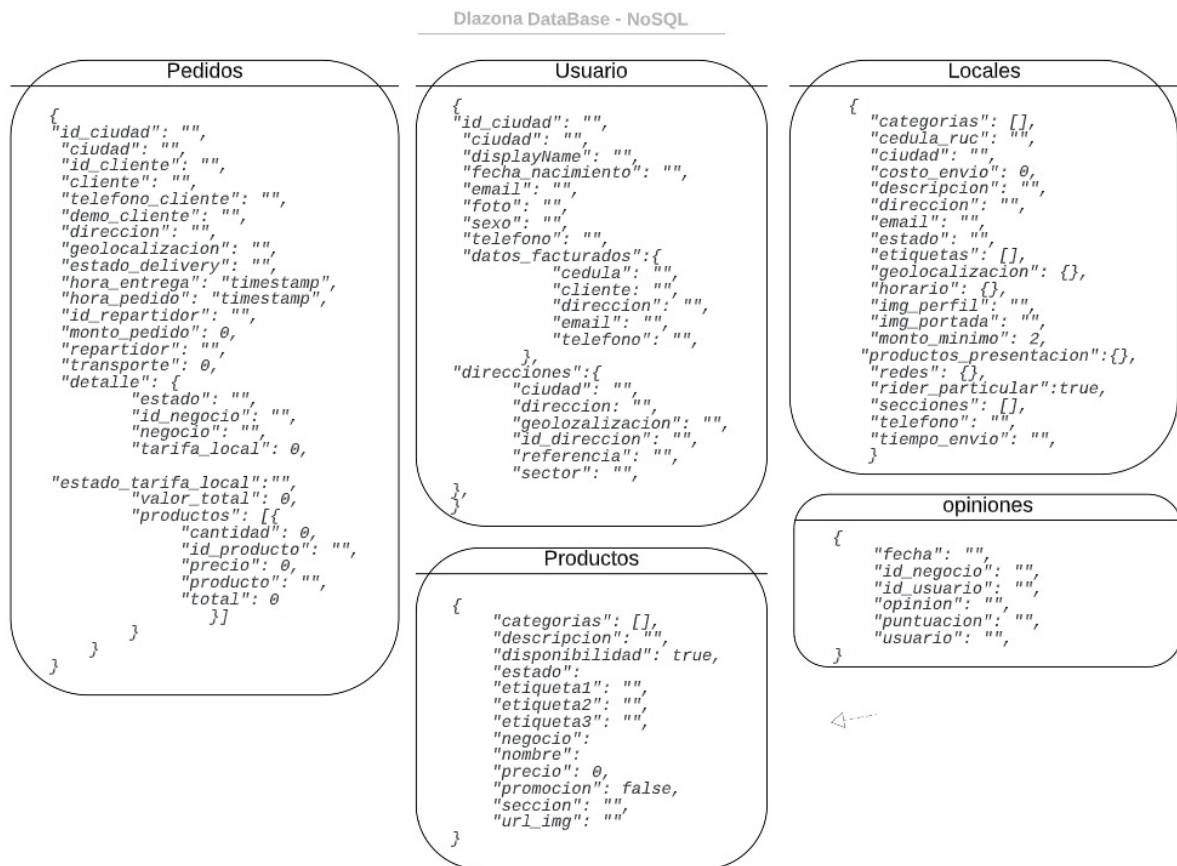
**Figura 11** Esquema NoSql empleado para la app administrador



**Fuente:** Elaboración Propia

Por otro lado, en la Figura 12 se puede visualizar el esquema de datos de la app del cliente, detallando los campos necesarios para el correcto funcionamiento. Cabe recalcar que, aunque se presenten por separado, la base de datos es una sola.

**Figura 12** Esquema NoSql empleado para la app cliente



**Fuente:** Elaboración Propia

Dentro de la metodología MMS el diccionario de datos es clave para conseguir un coherente diseño, y posteriormente implementación de la base de datos. Sin embargo, el esquema NoSql diseñado, incluye tipos de datos, descripciones dentro de la bd, y restricciones en caso de que existieran. Por lo que, el diccionario de datos se simplifica empleando los esquemas previamente diseñados.

### 2.4.2.3 Definición de interfaz.

#### 2.4.2.3.1 Diseño modular.

El sistema al ser modular, es necesario definir cada uno de los módulos a diseñar mediante las aplicaciones. Cabe recalcar que algunos módulos se emplean tanto en la app del cliente, como en la app del

administrador. Es por ello que la Tabla 9 describe los módulos y actividades inmersas en cada uno de ellos diferenciando entre la app cliente, y la app administrador.

Tabla 9: Módulos del sistema

#	Módulo	Actividades
1	Gestión de usuarios	Todos los usuarios, tanto de la <i>app cliente</i> como de la <i>app admin</i> , puede registrar un nuevo usuario, modificar datos personales del mismo, y recuperar contraseña en caso de olvidarla.  Existen solamente dos roles, el administrador y el cliente respectivamente.
2	Gestión de menú y categorías	En la <i>app admin</i> el usuario puede ingresar, modificar o eliminar una sección (categoría) dentro del menú.  El usuario en la <i>app cliente</i> puede visualizar el menú y contenido del mismo desde la página de inicio, así como los productos enlazados al menú.
3	Gestión de productos y promociones	El usuario en la <i>app admin</i> puede crear un nuevo producto ingresando los campos correspondientes. También tiene la opción de modificar la información, o eliminar dicho producto. Para cambiar el estado de disponibilidad no es necesario eliminar el producto, solo se debe modificar manualmente.
4	Gestión de pedidos	El usuario en la <i>app admin</i> puede ver un listado de los pedidos clasificados en 4 categorías, Pendientes, preparados, enviados y entregados. Para que el estado de cada pedido se modifique, el usuario debe cambiarlo manualmente mediante un botón.  El usuario en la <i>app cliente</i> puede ver una lista con todos los pedidos realizados.  Así como modificar los datos de facturación.
5	Gestión de entrega	El usuario en la <i>app admin</i> puede aceptar el pedido o rechazarlo, una vez que se encuentre listo, asigna al repartidor correspondiente y procede a realizar el envío.
6	Gestión de sucursales	El usuario en la <i>app admin</i> puede asignar un producto a cada sucursal, dependiendo de su disponibilidad.
7	Gestión de ubicaciones	El usuario en la <i>app cliente</i> puede modificar, agregar o eliminar diversas ubicaciones de manera que puedan ser seleccionadas al momento de la entrega.

**Fuente:** Elaboración propia.

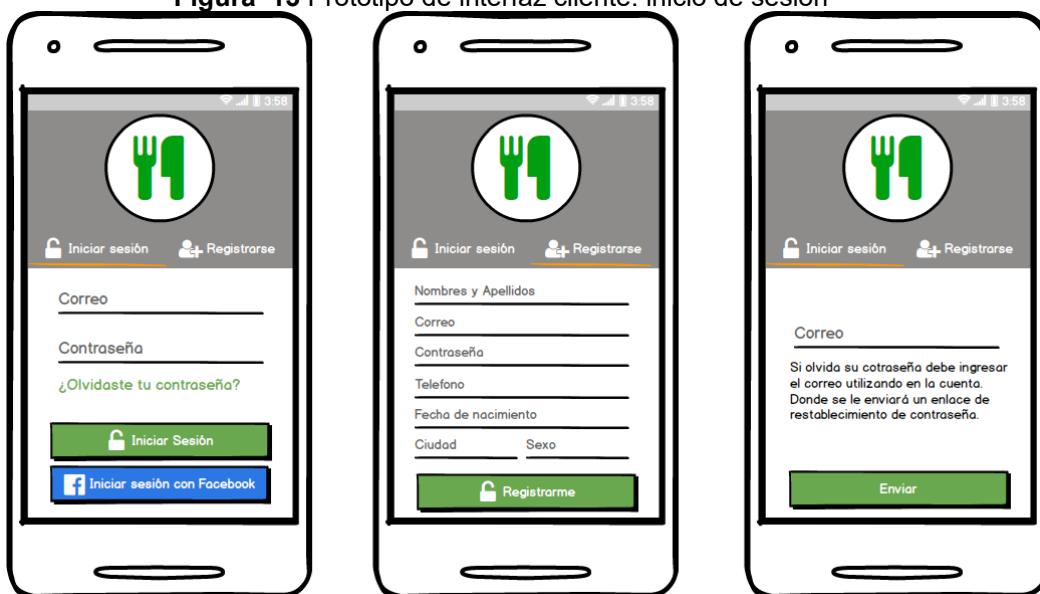
#### 2.4.2.3.2 Prototipado de interfaces de usuario.

Debido que la solución propuesta implementa una aplicación para la gestión del contenido del restaurante (app administrador), y otra aplicación para las funcionalidades de los usuarios comensales (app cliente), se realiza el prototipado de interfaces dividido para cada aplicación, sin embargo, las funcionalidades similares en ambos, se las describirá una sola vez dentro de la app de usuario.

#### › APP CLIENTE

Para simplificar y facilitar el proceso de compra de uno o más platillos, se implementó un sistema de rápido aprendizaje y sobre todo con diseño simple y de fácil entendimiento. La primera vez que un usuario nuevo abre la aplicación tiene dos opciones, registrarse o iniciar sesión. Al momento de registrarse un nuevo usuario debe completar los campos básicos con la información personal. Una vez que el usuario se ha registrado se dirige al inicio de sesión, en donde deberá validar los datos de usuario y contraseñas recién creados. En caso que el usuario olvide la contraseña, la app permite recuperar la contraseña mediante correo electrónico, el cual se visualizara en la pantalla de recuperación, así como en la Figura 13.

**Figura 13** Prototipo de interfaz cliente: inicio de sesión



Fuente: Elaboración Propia

**Pantalla de inicio:** Si los datos del usuario fueron verificados de manera exitosa, se redireccionará a la pantalla de inicio, en donde tendrá en la barra de tareas las opciones de *home*, *perfil*, *mis pedidos*, *mis direcciones*, *mis datos de facturación*, *ayuda*, y *acerca de*, como está en la Figura 14.

**Figura 14** Prototipo de interfaz cliente: pantalla de inicio



**Fuente:** Elaboración Propia

El usuario puede visualizar la información personal, así como editar los campos permitidos, como en la Figura 15.

**Figura 15** Prototipo de interfaz cliente: perfil de usuario



**Fuente:** Elaboración Propia

**Datos de facturación:** Se puede acceder a los datos de facturación desde el menú lateral. Al dar clic se muestra la información por defecto del usuario registrado, en caso de querer modificarla puede hacerlo. Este proceso no modifica la información personal del usuario, solo guarda los nuevos datos de facturación relacionados con ese cliente como se muestra en la Figura 16.

**Figura 16** Prototipo de interfaz cliente: datos de facturación

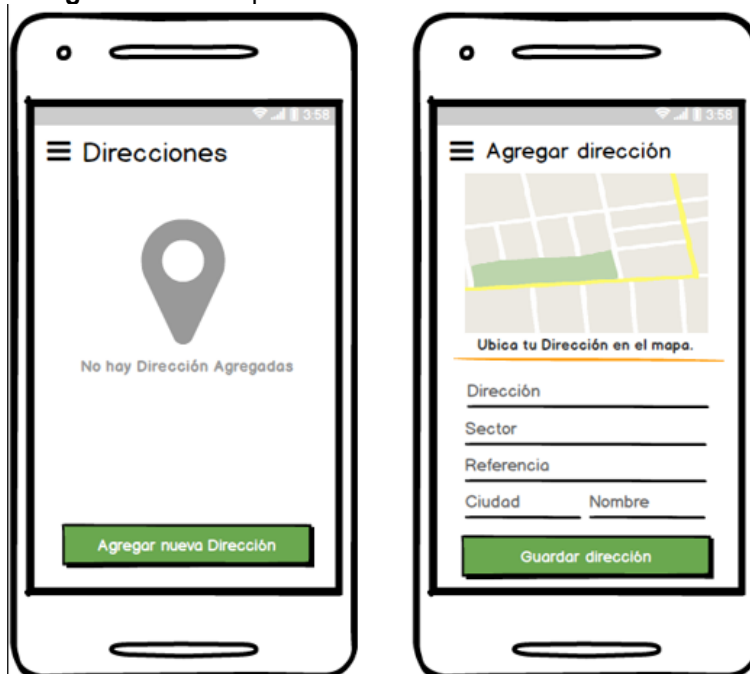
The image shows two mobile app prototypes side-by-side, both titled 'Datos de Facturación'. Each screen features a list of input fields: 'Cédula o R.U.C.', 'Nombres y Apellidos', 'Dirección', 'Nombres y Apellidos', and 'Dirección'. The left prototype has a green 'Actualizar Información' button at the bottom. The right prototype has a green 'Guardar' button and a red 'Cancelar' button at the bottom. Both screens include a hamburger menu icon in the top left corner and status bar icons (signal, Wi-Fi, battery, time) at the top.

**Fuente:** Elaboración Propia

**Direcciones:** Las direcciones pueden ser gestionadas mediante dos formas. La primera consiste en ingresar a la opción “*Mis direcciones*” ubicado en la barra lateral. Una vez ingresado se procede a agregar una nueva dirección, completando los campos respectivos.

La segunda forma es ingresar la dirección como nueva al momento de efectuar la compra. Al dar clic, en “*Agregar nueva dirección*”, se abre la ventana para completar la información necesaria, como se muestra en la Figura 17.

**Figura 17** Prototipo de interfaz cliente: datos de facturación



Fuente: Elaboración Propia

**Ayuda e información:** La sección de ayuda para el usuario puede ser accedida desde el menú lateral de la pantalla principal. Se diseñan dos pantallas, la primera consiste en preguntas frecuentes sobre cómo usar la app, y la segunda sobre información de la empresa, como se muestra en la Figura 18.

**Figura 18** Prototipo de interfaz cliente: ayuda e información



Fuente: Elaboración Propia



**Menú y productos:** El menú se mostrará en la pantalla principal de la app del cliente. En caso de que el menú tenga secciones, en la parte superior se mostrarán las diferentes categorías, como se muestra en Figura 19. Al seleccionarlás se puede visualizar los productos que posee cada una.

**Figura 19** Prototipo de interfaz cliente: menú y productos



**Fuente:** Elaboración Propia

**Comprar:** Para realizar una compra el usuario debe seleccionar uno o más productos del menú mostrados en la pantalla principal, y seleccionar la cantidad que desea, y luego dar clic en “Anadir”. Cada vez que se aumente algo al carrito el valor monetario en la parte superior incrementará.

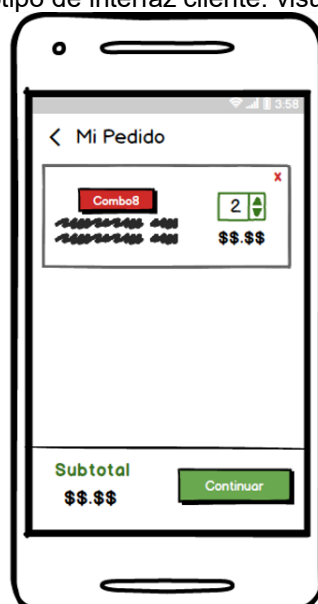
**Figura 20** Prototipo de interfaz cliente: agregar al carrito



**Fuente:** Elaboración Propia

Una vez que se ha añadido al carrito, se procede a la visualización del pedido que se desea comprar. En el cual se muestra una lista con los productos y el subtotal del mismo.

**Figura 21** Prototipo de interfaz cliente: visualización de pedido



**Fuente:** Elaboración Propia

Una vez que se ha confirmado que la lista es correcta, se procede a seleccionar la dirección de entrega, luego el método de pago conjuntamente con el detalle de la factura. Al realizar el pedido, se enviará la orden de pago como mensaje por WhatsApp a respectivo número de Pollo D La Zona.

**Figura 22** Prototipo de interfaz cliente: envío y método de pago



Fuente: Elaboración Propia

**Visualizar Pedidos:** La lista de los pedidos realizados, se pueden visualizar en la sección “*Mis pedidos*”, En donde se encontrará un resumen de los datos importantes del pedido, y el estado en que se encuentra. También es posible visualizar todo el detalle del pedido al dar clic en “*Ver más*”.

**Figura 23** Prototipo de interfaz cliente: visualizar pedidos



Fuente: Elaboración Propia

## › APP ADMINISTRADOR

Dentro de la app administrador existen algunas interfaces similares a las de los usuarios, como son la autenticación e inicio de sesión, el registro y la recuperación de contraseña. Esta app se centra principalmente en administrar y gestionar la información y productos que la empresa ofrece.

**Gestión de productos:** Dentro de la gestión de productos se puede agregar, eliminar y modificar un producto creado. Primero el usuario debe tener como mínimo una sección, en caso de querer agregar una nueva debe dar click en “+ Sección”, con el cual le aparecerá una nueva ventana con los datos necesarios.

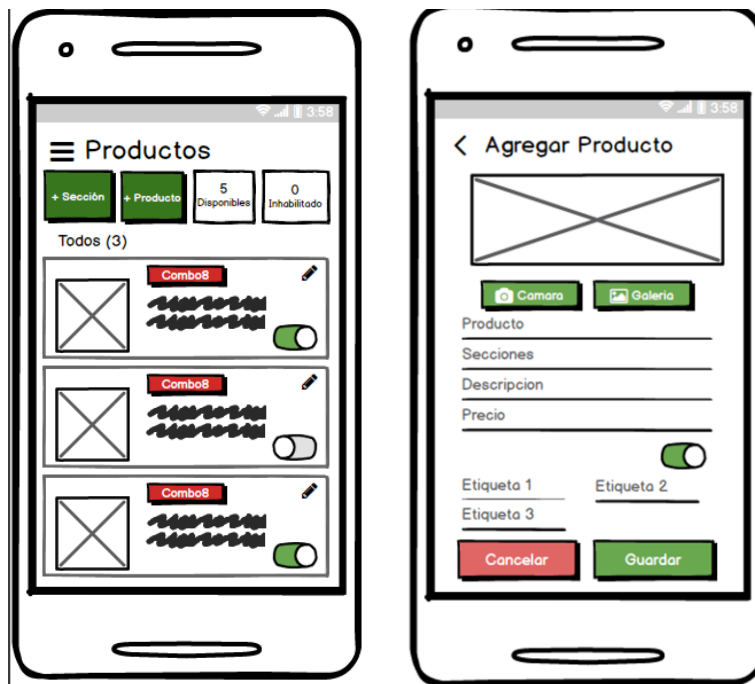
**Figura 24** Prototipo de interfaz administrador: Gestión de secciones



Fuente: Elaboración Propia

Para crear un producto el usuario debe dar click en “+ Producto”, con el cual le aparecerá una nueva ventana con los datos necesarios para que el nuevo producto sea registrado. Una parte dentro del proceso es que el usuario puede asignar si el producto se encuentra en *Promoción*, o si existe disponibilidad inmediata. Para editar el producto se debe tocar el símbolo de lápiz.

**Figura 25** Prototipo de interfaz administrador: Gestión de productos



Fuente: Elaboración Propia

**Pedidos y Envíos:** Dentro de la sección de Pedidos, el usuario visualizará el listado de los pedidos organizado en 4 estados, “*Pendiente, Preparación, Enviados, Entregados*”. El listado incluye los detalles del pedido, así como la información de contacto del cliente. Una vez que se recibe el pedido se el administrador debe Aceptar o Rechazarlo.

Una vez que se el administrador acepte el pedido, se muestra un detalle de la compra, y luego el estado del mismo pasará a preparación. Cuando el pedido se encuentre listo, se seleccionará la opción “Listo”, procediendo a asignar al motorizado encargado para enviar el pedido. Posteriormente el estado cambiara a enviado, para que al momento de ser recibido el estado se convierta a Entregado.

**Figura 26** Prototipo de interfaz administrador: pedidos



**Fuente:** Elaboración Propia

**Panel informativo:** Dentro del panel informativo se presentan algunos indicativos, como son los ingresos del día, el número de ventas realizadas. Y de manera general las ganancias del último mes.

**Figura 27** Prototipo de interfaz administrador: panel informativo (dashboard)



Fuente: Elaboración Propia

**Ayuda e información:** De igual forma como en la app del cliente; la app del administrador cuenta con una página de información en la cual se presentan soluciones a dudas frecuentes o a los procesos con mayor relevancia dentro de la app.

**Figura 28** Prototipo de interfaz administrador: ayuda e información



Fuente: Elaboración Propia

## 2.5 EJECUCIÓN Y/O ENSAMBLAJE DEL PROTOTIPO.

La fase dentro de la metodología MMS consiste en la codificación de cada uno de los módulos, interfaces y base de datos respectivamente. La codificación se

realiza mediante el lenguaje de programación React Native, quien además de ser un lenguaje propio para el diseño móvil, cuenta con beneficios para la reutilización de componentes.

Un tema clave dentro de la fase de ensamblaje es que se mantuvo el mismo diseño, orientación y colores en todas las interfaces de usuario; así como identificadores claros según lo que se desee definir. Para esta etapa se emplea las distintas actividades Conversiones para el código, codificación, y refactorización de los módulos respectivamente.

### **2.5.1 Refactorización del código.**

La refactorización del código consiste en la reestructuración del código fuente, en cuanto a reglas de codificación, o mediante conversiones. Una vez codificados cada uno de los módulos se realizó la refactorización, entre los problemas o partes del código que se modificaron (en estructura) fueron las siguientes:

- Omisión de uso innecesario de controles o sentencias anidadas.
- Cambio de nombre de funciones o métodos destinadas al mismo proceso. Si son funciones en cascada se definen como el nombreDelProceso\_1, sin embargo, si son funciones independientes, se modifica el nombre por uno más significativo y fácil de entender. Si la misma función o método se debe definir en varias clases, se emplea el mismo nombre.
- Simplificación de fórmulas condicionadas y simplificación de las solicitudes a métodos.
- Para funciones largas y complejas se emplean comentarios.

### **2.5.2 Codificación de módulos.**

Una vez identificados los puntos clave de refactorización, se procede a codificar cada una de las funcionalidades y módulos diseñadas mediante el prototipado en la fase anterior. A continuación, se muestran las páginas diseñadas para cada proceso descrito en la propuesta.

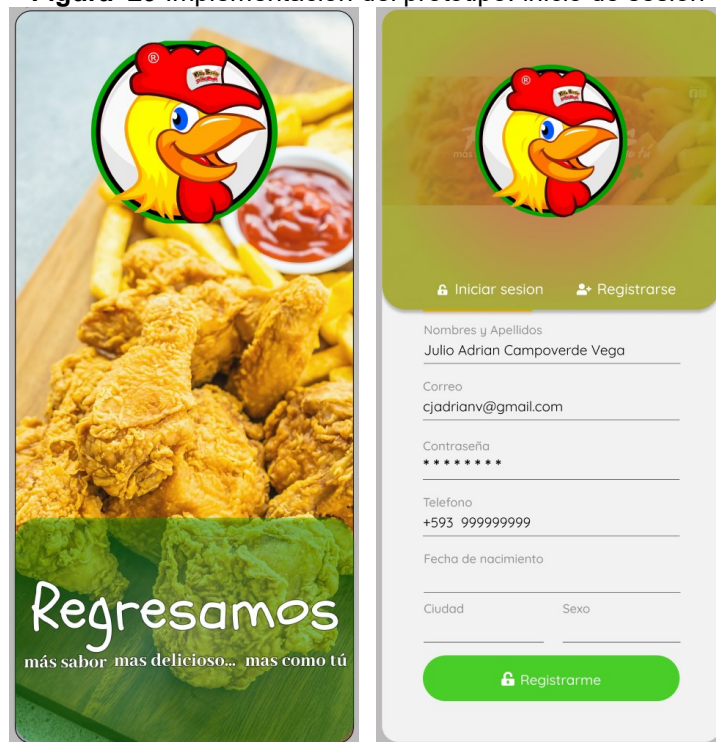


### 2.5.2.1 Páginas App Cliente.

#### **Gestión de usuarios.**

Al iniciar la app el usuario visualizará una pantalla de espera con el logo de la empresa, y luego de cargar los recursos necesarios se visualiza la pantalla de inicio de sesión o registro. Si el usuario no posee una cuenta, debe proceder con el registro de usuario. Para ello es necesario completar los campos requeridos, ingresar un correo electrónico en uso, y posteriormente dar clic en “*Registrarme*”, como se muestra en la Figura 29.

**Figura 29** Implementación del prototipo: inicio de sesión



**Fuente:** Elaboración Propia

Si ya posea una cuenta se ingresa el correo y contraseña para ser validado. En caso de olvidar la contraseña, puede dar clic en “*¿Olvidaste tu contraseña?*”, y se enviará un enlace de restablecimiento de contraseña al correo electrónico.

**Figura 30** Implementación del prototipo: registro

Correo  
cjadrianv@gmail.com

Contraseña  
\*\*\*\*\*

¿Olvidastes tu contraseña?

Iniciar Sesión

Iniciar sesión con Facebook

Correo  
cjadrianv@gmail.com

Si olvidó su contraseña debe ingresar el correo utilizado en la cuenta. Donde se le enviara un enlace de restablecimiento de contraseña.

Enviar

Fuente: Elaboración Propia

Validado el usuario, se redirige a la pantalla de inicio en la cual se visualiza el menú con los productos disponibles. La pantalla principal posee una barra de opciones laterales, en la cual se encontrarán las opciones de **inicio**, **perfil**, **mis pedidos**, **mis direcciones**, **mis datos de facturación**, **ayuda**, y **acerca de**. En perfil, el usuario podrá visualizar su información, así como editar los campos permitidos.

**Figura 31** Implementación del prototipo: perfil de usuario

Julio Adrian Campoverde Vega  
cjadrianv@gmail.com

Inicio

Perfil

Mis Pedidos

Mis Direcciones

Mis Datos de Facturación

Ayuda

Acerca de Dlazona

Cerrar Sesión

Perfil

Personal details

Adrian Campoverde  
cjadrianv@gmail.com  
0939706637  
Machala, El Oro, Ecuador

Editar Información

Editar

Nombres y Apellidos  
Julio Adrian Campoverde Vega

Telefono  
+593 999999999

Fecha de nacimiento  
03 06 1997

Ciudad Machala Sexo Masculino

Guardar

Fuente: Elaboración Propia

Otro campo importante relacionado al usuario son los datos de facturación. Dentro de las opciones del menú lateral se pueden seleccionar Mis Datos de Facturación, los cuales por defecto serán los mismos datos que el cliente registro como datos personales. Si desea actualizar los datos de facturación, puede dar clic en “*Actualizar Información*”, y modificar los datos a los cuales se realizarán todas las facturas. Sin embargo, el modificar los datos, no cambia la información personal del perfil del usuario, ya que los datos de facturación son campos separados.

**Figura 32** Implementación del prototipo: datos de facturación

The figure displays two mobile application screens for managing billing data. Both screens feature a header with a hamburger menu icon and the title 'Datos de Facturación'. The form fields are identical on both screens and include: 'Cédula o RuC' (0706619541 001), 'Nombres y Apellidos' (Julio Adrian Campoverde Vega), 'Dirección' (Ciudadela la Primavera), 'Teléfono' (+593 939706637), and 'Email' (cjadrianv@gmail.com). The left screen has a green 'Actualizar Información' button at the bottom. The right screen has a green 'Guardar' button and a grey 'Cancelar' button at the bottom.

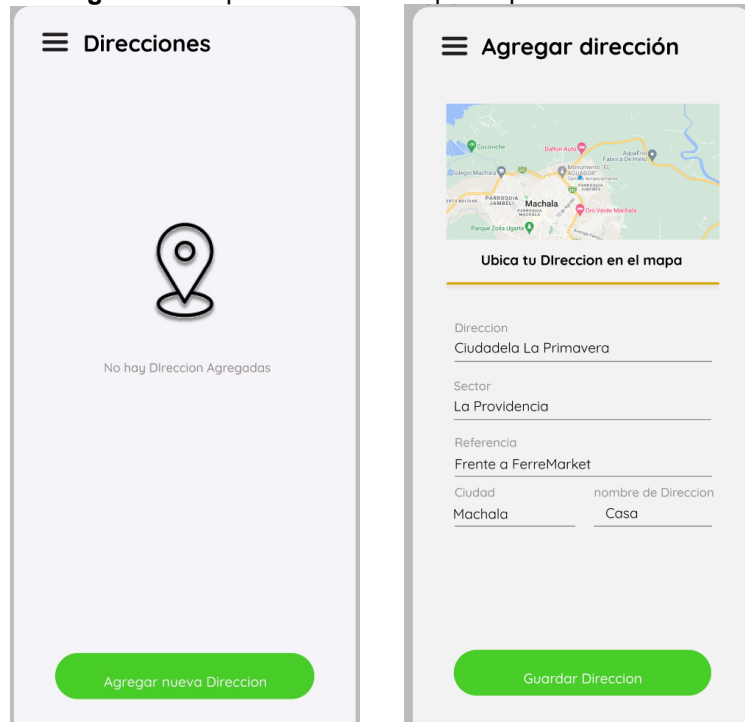
**Fuente:** Elaboración Propia

### ***Gestión de ubicaciones.***

Para la gestión de ubicaciones, se debe realizar un proceso similar a los datos de facturación.

En la sección de direcciones se muestra una lista con direcciones guardadas, en caso de no contar con ninguna se puede agregar una nueva dirección. Se pueden agregar varias direcciones, siempre y cuando se identifiquen con un nombre diferente.

**Figura 33** Implementación del prototipo: direcciones

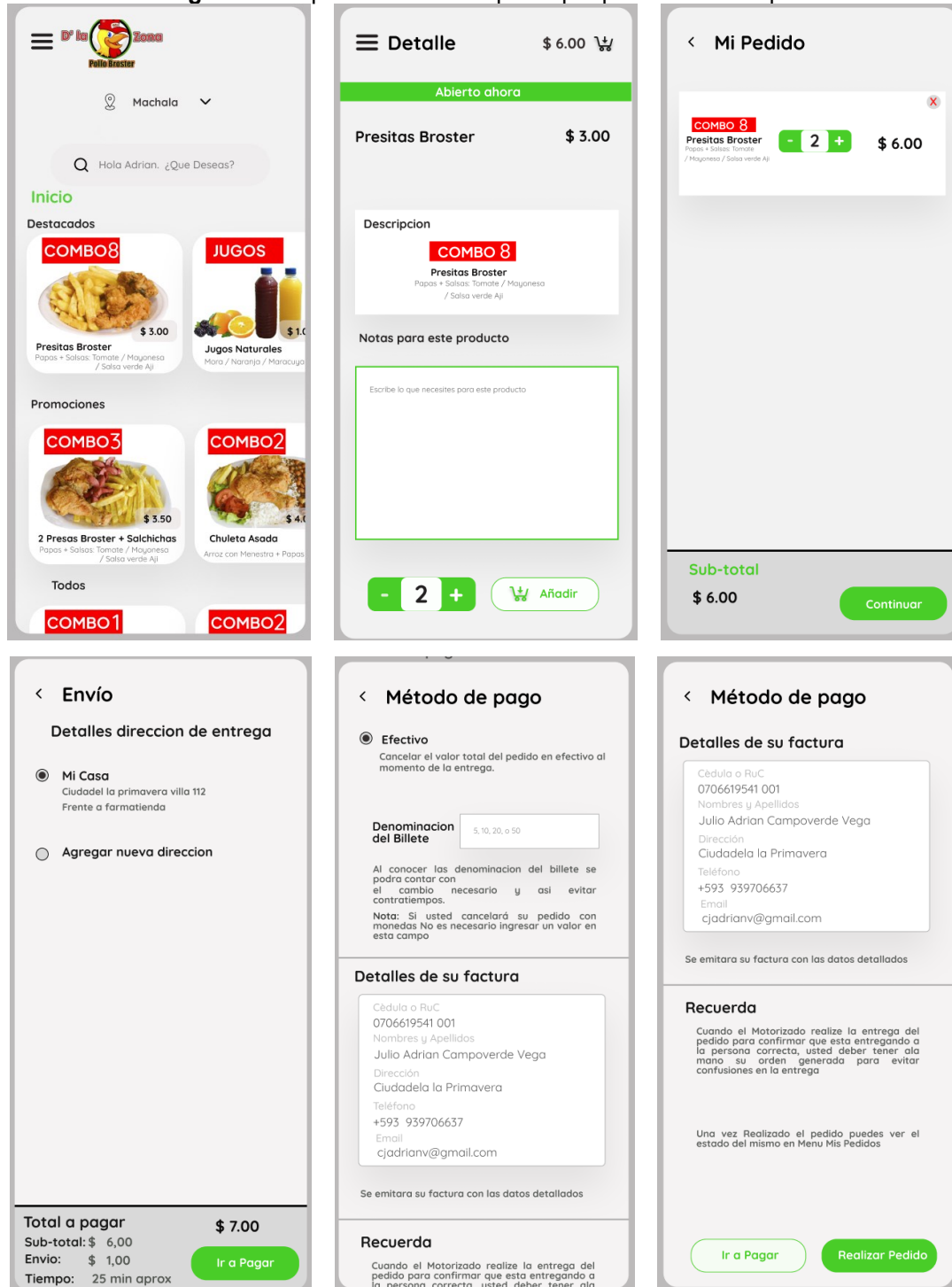


**Fuente:** Elaboración Propia

### ***Gestión de compra***

Para realizar la compra, se debe seleccionar un producto del menú, y automáticamente aparecerá el detalle del producto y la cantidad que desea. Una vez seleccionado debe dar clic a agregar al carrito. Una vez seleccionados todos los productos deseados, en el carrito de compras (Mi pedido) se visualiza una lista en donde el usuario podrá agregar o disminuir la cantidad, así como eliminar el producto si ya no lo desea. Al dar clic en continuar, se procede a seleccionar la dirección de entrega (puede ser de entre las direcciones guardadas o agregar una nueva), luego se procede a seleccionar el método de pago (*efectivo*). Cuando se desea finalizar el pedido, y se da clic en *“Realizar pedido”* y automáticamente abre una conversación en WhatsApp para enviar la orden de pedido al número registrado del restaurante.

**Figura 34** Implementación del prototipo: proceso de compra

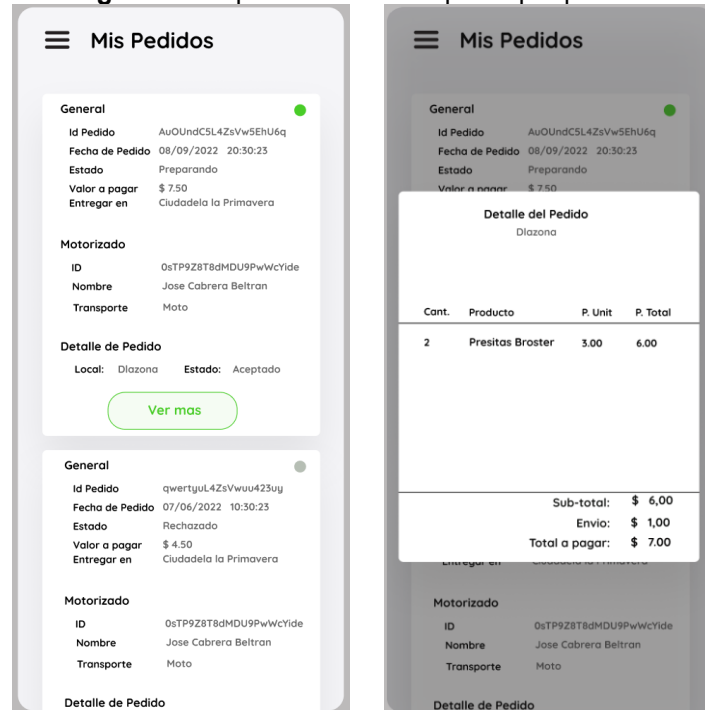


Fuente: Elaboración Propia

### **Gestión de pedidos.**

El cliente puede visualizar un listado con todos los pedidos que ha realizado, así como una breve descripción y el estado de los mismos. Si se desea visualizar todos los detalles del pedido se debe dar clic en “Ver más”.

**Figura 35** Implementación del prototipo: pedidos

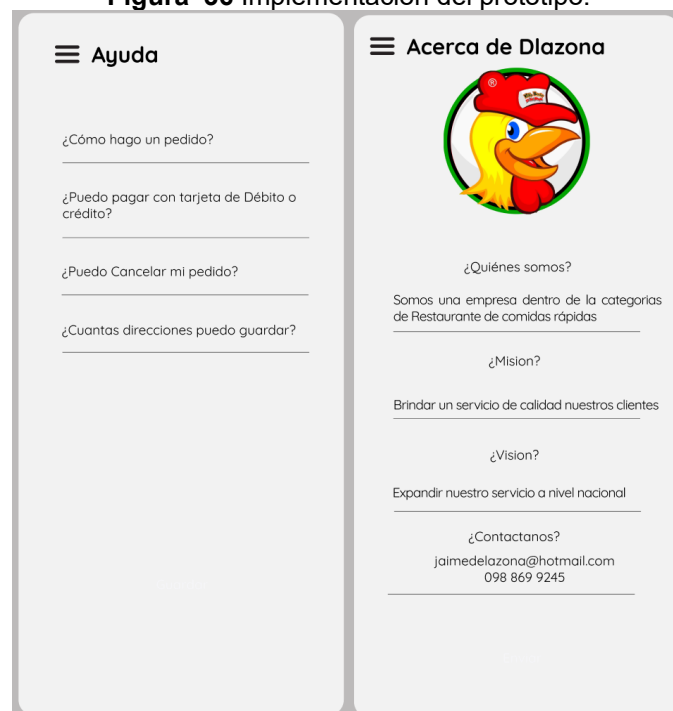


Fuente: Elaboración Propia

## Sección de Ayuda y Acerca de.

Las páginas que se encuentran tanto en el cliente como en el administrador son la de ayuda, y acerca de. En estas se establecen preguntas frecuentes que el usuario puede tener al momento de utilizar la aplicación.

**Figura 36** Implementación del prototipo:



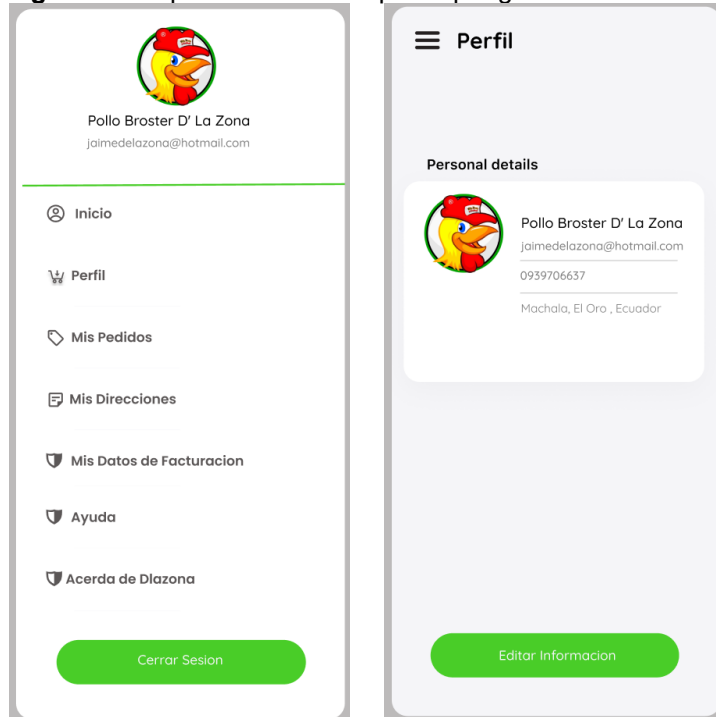
Fuente: Elaboración Propia

### 2.5.2.2 Páginas App Administrador.

#### ***Gestión de usuarios.***

Al igual que la app del cliente, la app del administrador posee el mismo log in, el mismo registro y el mismo proceso de recuperación de contraseña. La página de inicio al validar el correo y contraseña se muestra en la Figura 37.

**Figura 37** Implementación del prototipo: gestión de usuarios

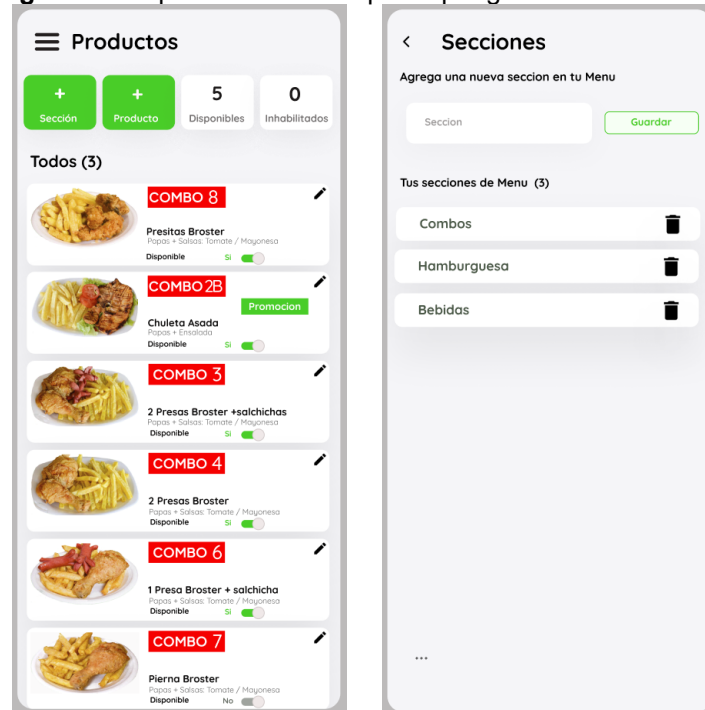


**Fuente:** Elaboración Propia

#### ***Gestión de secciones***

Para gestionar de manera adecuada las secciones del menú, en el apartado de productos se puede ingresar una nueva sección. Aquí se visualizará un listado de las secciones existentes y un label para agregar nuevas secciones.

**Figura 38** Implementación del prototipo: gestión de secciones

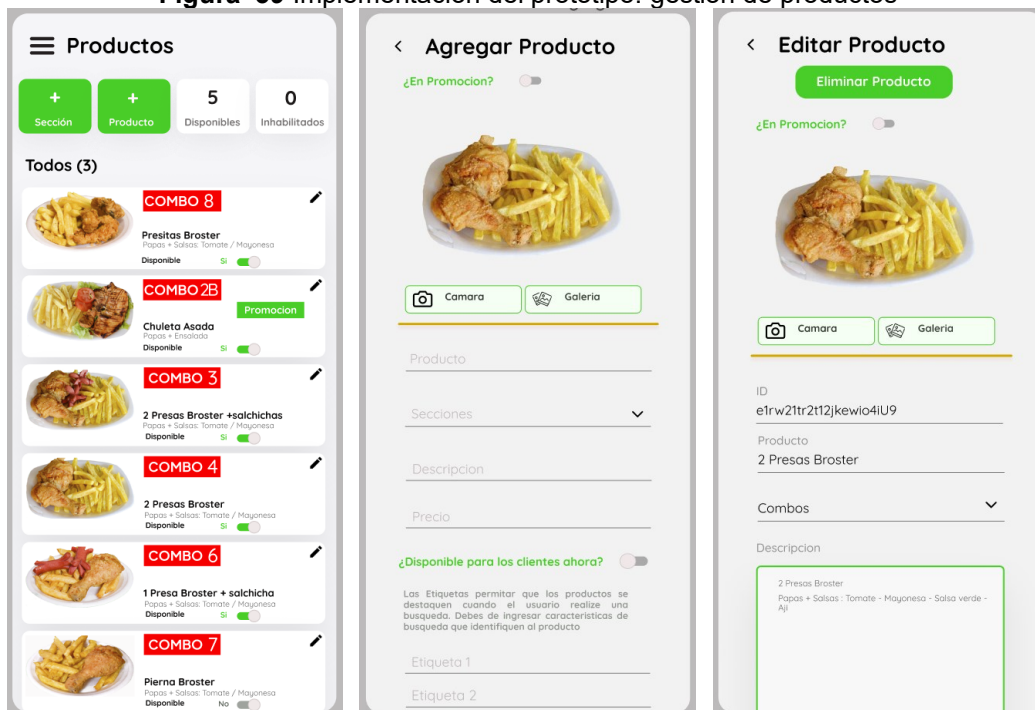


Fuente: Elaboración Propia

### **Gestión de productos.**

De igual forma se puede agregar un nuevo producto rellenando los campos requeridos, incluyendo una foto del producto (png o jpg).

**Figura 39** Implementación del prototipo: gestión de productos



Fuente: Elaboración Propia



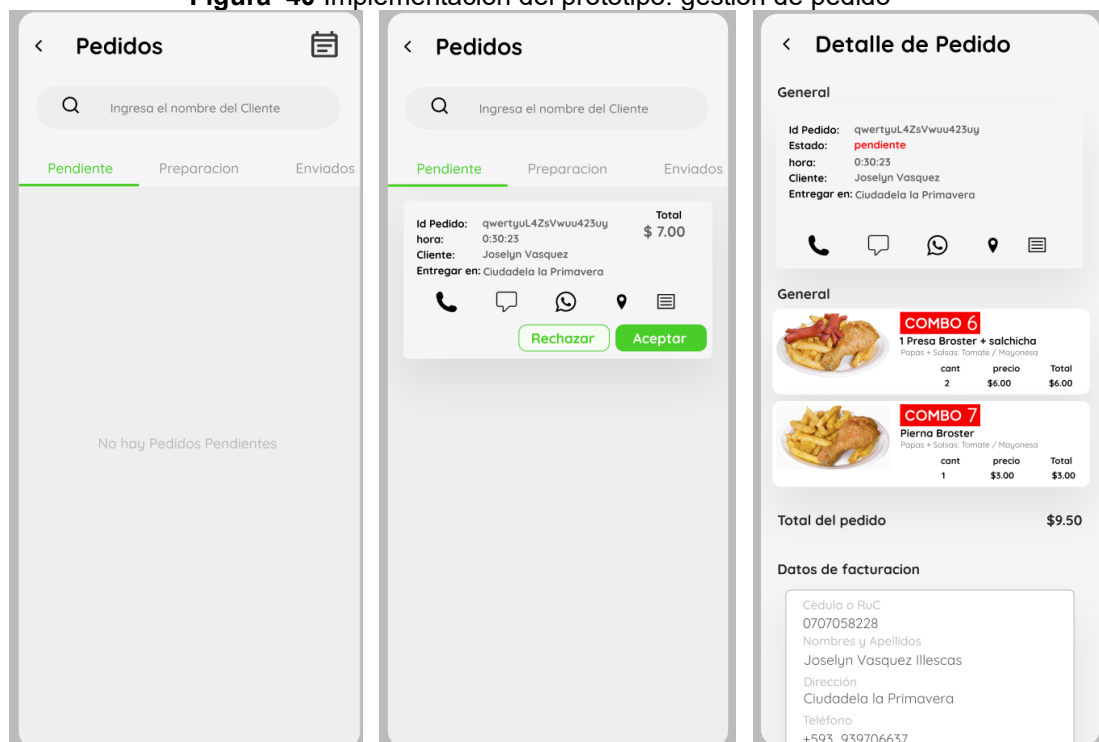
Para editar un producto se debe dar clic en el lápiz de la esquina superior de cada producto. La disponibilidad de cada producto puede ser cambiada manualmente desde la lista de productos, sin necesidad de eliminar el producto.

### ***Gestión de pedidos.***

Dentro de la sección de Pedidos, el usuario visualizará el listado de los pedidos organizado en 4 estados, “*Pendiente, Preparación, Enviados, Entregados*”. El listado incluye los detalles del pedido, así como la información de contacto del cliente. Una vez que se recibe el pedido se el administrador debe Aceptar o Rechazarlo, como se muestra en la Figura 40.

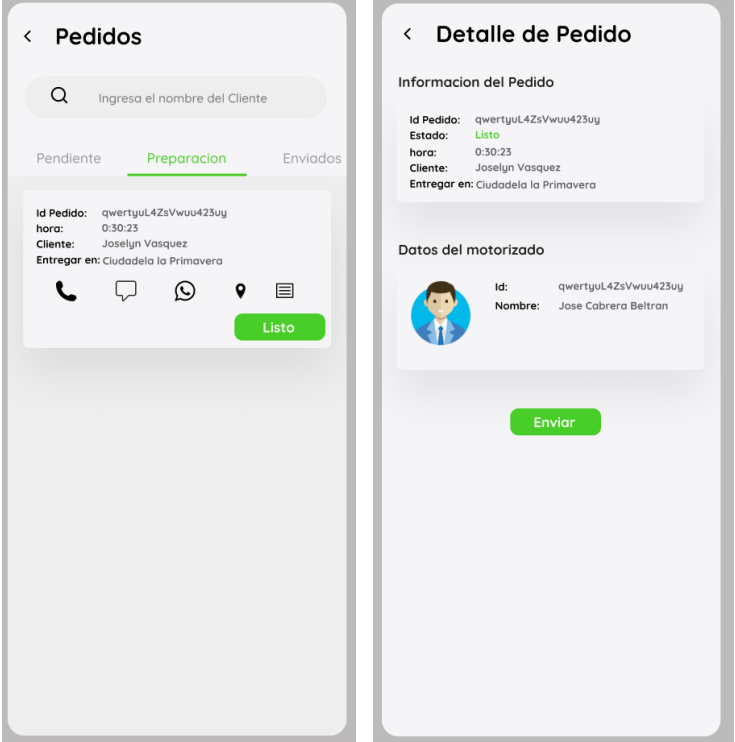
Una vez que se el administrador acepte el pedido, se muestra un detalle de la compra, y luego el estado del mismo pasará a preparación. Cuando el pedido se encuentre listo, se seleccionará la opción “Listo”, procediendo a asignar al motorizado encargado para enviar el pedido como se muestra en la Figura 41. Posteriormente el estado cambiara a enviado, para que al momento de ser recibido el estado se convierta a Entregado, como se muestra en la Figura 42.

**Figura 40** Implementación del prototipo: gestión de pedido



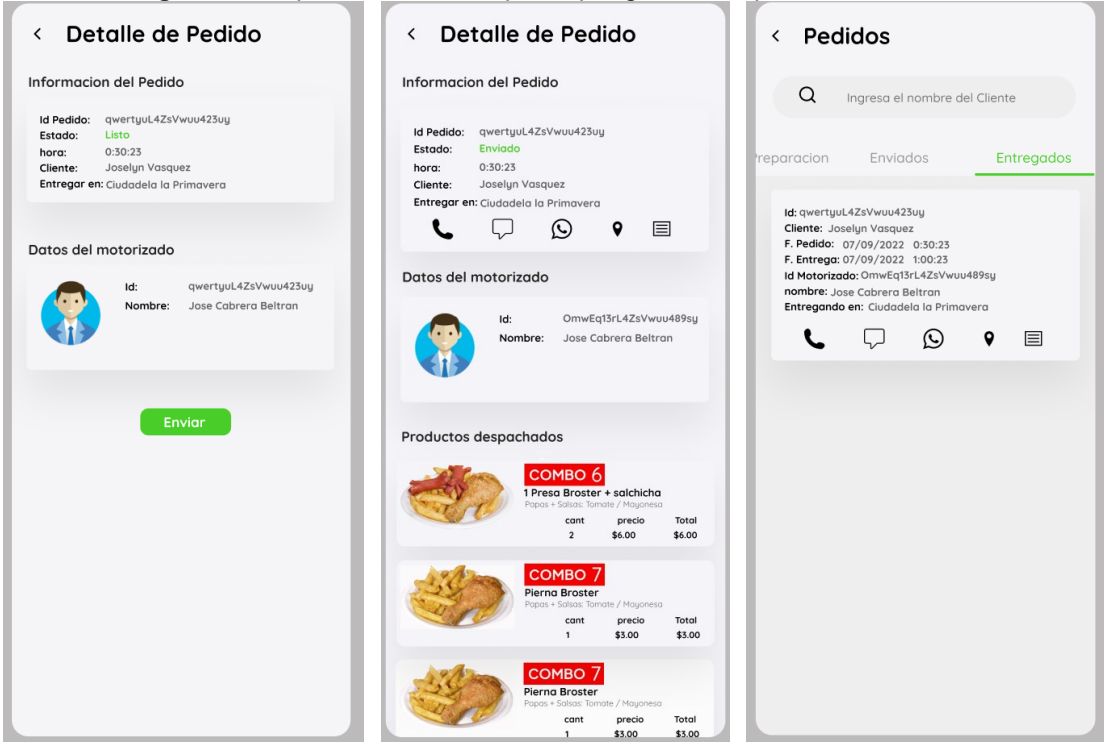
**Fuente:** Elaboración Propia

Figura 41 Implementación del prototipo: gestión de pedido aceptado



Fuente: Elaboración Propia

Figura 42 Implementación del prototipo: gestión de pedido en camino



Fuente: Elaboración Propia

### 3 CAPÍTULO III: EVALUACIÓN DEL PROTOTIPO

La metodología MMS al tener enfoque ágil, propone una revisión y evaluación de los entregables al final de cada sprint, de esta manera se establece un porcentaje de cumplimiento de las actividades de desarrollo de los módulos del proyecto. En la Tabla 10 se detalla un resumen de revisiones de cada iteración.

Tabla 10: Revisión de entregables por sprint

# Sprint	Fecha de revisión	Actividades Detalle Actividad					
			0%	25%	50%	75%	100%
1	21/06/22	Diseño de interfaces de usuario HU-1, HU-3, HU-2, HU-4					X
		Codificación de funcionalidades					X
		Validación y pruebas de funcionalidades.					X
		Modificación y corrección de errores					X
2	30/6/2022	Diseño de interfaces de usuario HU-5, HU-6, HU-7, HU-12					X
		Codificación de funcionalidades.					X
		Validación y pruebas de funcionalidades.					X
		Modificación y corrección de errores					X
3	11/7/22	Diseño de interfaces de usuario HU-6, HU-7, HU-9, HU-10, HU-13.					X
		Codificación de funcionalidades.					X
		Validación y pruebas de funcionalidades.					X
		Modificación y corrección de errores					X
4	21/7/2022	Diseño de interfaces de usuario HU-11, HU-13					X
		Codificación de funcionalidades					X
		Validación y pruebas de funcionalidades.					X
		Modificación y corrección de errores					X
5	2/8/2022	Diseño de interfaces de usuario para dashboard.					X
		Codificación de funcionalidades					X
		Validación y pruebas de todo el sistema.					X
		Modificación y corrección de errores					X

Fuente: Elaboración propia.

### 3.1 PLAN DE EVALUACIÓN.

El plan de evaluación de un sistema de información beneficia directamente al proceso de desarrollo, logrando al final un producto de calidad, y que satisfaga las necesidades del usuario en su totalidad. Las evaluaciones de las aplicaciones móviles aun no poseen metodologías estrictas o gran variedad de herramientas online que faciliten su evaluación. En vista que las necesidades de las apps móviles se centran en la usabilidad, accesibilidad [50], calidad, entendimiento, y rápido aprendizaje, se emplea la métrica de calidad ISO/IEC 25010 para evaluar los diferentes parámetros de la misma. La evaluación se realiza por un tercero, quien responderá una encuesta con preguntas orientadas al conocimiento de las diferentes métricas, especificadas en la Tabla 11.

Tabla 11: plan de evaluación 25010

ATRIBUTO	MÉTRICA	PREGUNTA
<b>Adecuación funcional</b>	<b>Complejidad</b>	1. ¿Las funcionalidades del sistema resuelve el problema para el que fueron diseñadas? 2. ¿Todas las funcionalidades pueden ser efectuadas hasta el final, de manera rápida y correcta?
	<b>Corrección</b>	3. El sistema emite mensajes en caso del ingreso de información o tipo de datos errónea en los campos?
	<b>Pertinencia</b>	4. La app brinda información específica según el tipo de usuario que la emplea?
<b>Eficiencia del desempeño</b>	<b>Comportamiento</b>	5. ¿Las apps procesan las funcionalidades de manera rápida? 6. ¿Los botones realizan la funciones para las que están creadas?
	<b>Uso de recursos</b>	7. ¿La app consume solo los recursos necesarios del móvil? 8. ¿Los recursos o permisos requeridos son empleados adecuadamente?
	<b>Capacidad</b>	9. ¿Los procesos funcionan normalmente aun en los límites máximos?
<b>Compatibilidad</b>	<b>Coexistencia</b>	10. ¿Las apps no interrumpen el buen funcionamiento de otras aplicaciones? 11. ¿Las apps no son interrumpidas por otras aplicaciones instaladas?

ATRIBUTO	MÉTRICA	PREGUNTA
	<b>Inter-operabilidad</b>	12. ¿La funcionalidad de mensaje por WhatsApp funciona adecuadamente? 13. ¿Es posible llamar sin problema y de forma rápida a un cliente dentro de la aplicación?
<b>Usabilidad</b>	<b>Fácil aprendizaje</b>	14. ¿El uso de la aplicación es intuitivo? 15. ¿La interfaz de usuario es clara y amigable?
	<b>Operabilidad</b>	16. ¿Cada módulo es manejado de manera independiente por actividades independientes?
	<b>Aviso ante errores del usuario</b>	17. ¿La app ofrece mensaje en caso de que el usuario ingrese un campo erróneo?
	<b>Accesibilidad</b>	18. ¿La app puede ser usada por cualquier tipo de persona?
<b>Seguridad</b>	<b>Autenticidad</b>	19. ¿La app deniega el acceso a personas no autorizadas? 20. ¿La app tiene autenticación y verificación de datos?
	<b>Integridad de datos</b>	21. ¿La aplicación solo permite modificar datos autorizados y con personas con credenciales?
<b>Portabilidad</b>	<b>Adaptabilidad</b>	22. ¿La visualización de las pantallas de las apps es ajustable a cualquier tipo de dispositivo móvil?
	<b>Fácil de instalar</b>	23. ¿La app es fácil de instalar?

Fuente: Elaboración propia.

### 3.2 RESULTADOS DE EVALUACIÓN.

Dentro de los resultados de la evaluación se detallan los resultados de la encuesta de la métrica 25010, evaluando su grado de aceptación mediante escala de Likert, detallada en la Tabla 12.

Tabla 12: Escala de Likert en evaluación 25010

Descripción	Valor
-------------	-------

Descripción	Valor
1	Muy Malo
2	Malo
3	Regular
4	Bueno
5	Excelente

Fuente: Elaboración propia.

En la Tabla 13 se muestra la valoración de la evaluación realizada, mediante las preguntas y métricas de la norma ISO/IEC 25010.

Tabla 13: plan de evaluación 25010

ATRIBUTO	MÉTRICA	PREGUNTA	VALOR
<b>Adecuación funcional</b> <b>Eficiencia del desempeño</b>	<b>Complejidad</b>	1. ¿Las funcionalidades del sistema resuelve el problema para el que fueron diseñadas?	5
		2. ¿Todas las funcionalidades pueden ser efectuadas hasta el final, de manera rápida y correcta?	5
	<b>Corrección</b>	3. El sistema emite mensajes en caso del ingreso de información o tipo de datos errónea en los campos?	4
	<b>Pertinencia</b>	4. La app brinda información específica según el tipo de usuario que la emplea?	5
<b>Compatibilidad</b> <b>Usabilidad</b>	<b>Comportamiento</b>	5. ¿Las apps procesan las funcionalidades de manera rápida?	4
		6. ¿Los botones realizan la funciones para las que están creadas?	5
	<b>Uso de recursos</b>	7. ¿La app consume solo los recursos necesarios del móvil?	5
		8. ¿Los recursos o permisos requeridos son empleados adecuadamente?	5
	<b>Capacidad</b>	9. ¿Los procesos funcionan normalmente aun en los límites máximos?	5
<b>Seguridad</b>	<b>Coexistencia</b>	10. ¿Las apps no interrumpen el buen funcionamiento de otras aplicaciones?	5

		11. ¿Las apps no son interrumpidas por otras aplicaciones instaladas?	5
		12. ¿La funcionalidad de mensaje por WhatsApp funciona adecuadamente?	5
<b>Portabilidad Adecuación funcional</b>	<b>Inter-operabilidad</b>	13. ¿Es posible llamar sin problema y de forma rápida a un cliente dentro de la aplicación?	5
		14. ¿El uso de la aplicación es intuitivo?	5
	<b>Fácil aprendizaje</b>	15. ¿La interfaz de usuario es clara y amigable?	5
	<b>Operabilidad</b>	16. ¿Cada módulo es manejado por actividades independientes?	5
	<b>Aviso ante errores del usuario</b>	17. ¿La app ofrece mensaje en caso de que el usuario ingrese un campo erróneo?	5
<b>Eficiencia del desempeño</b>	<b>Accesibilidad</b>	18. ¿La app puede ser usada por cualquier tipo de persona?	5
		19. ¿La app deniega el acceso a personas no autorizadas?	5
	<b>Autenticidad</b>	20. ¿La app tiene autenticación y verificación de datos?	4
		21. ¿La aplicación solo permite modificar datos autorizados y con personas con credenciales?	5
<b>Compatibilidad</b>	<b>Integridad de datos</b>	22. ¿La visualización de las pantallas de las apps es ajustable a cualquier tipo de dispositivo móvil?	5
	<b>Adaptabilidad</b>	23. ¿La app es fácil de instalar?	5
<b>TOTAL(115: 100%)</b>			112: <b>97%</b>

Fuente: Elaboración propia.

### 3.3 CONCLUSIONES.

Una vez finalizado el proceso de desarrollo, y las demás etapas inmersas dentro de la metodología MMS para la optimización de los procesos de gestión de pedidos se logra concluir lo siguiente:

- El uso de MMS como metodología de desarrollo permitió desarrollar un proyecto eficiente, y de calidad en el periodo de tiempo establecido. Esto se debe al enfoque ágil sobre el cuál se basa la metodología, distribuyendo los procesos y actividades en sprints e incluyendo la comunicación con los interesados como herramienta principal para la evaluación y valoración del sistema final.
- Mediante las herramientas de elicitación de requerimiento y las actividades en la fase de planificación de la metodología MMS, se logró identificar y analizar los requerimientos tanto funcionales como no funcionales necesarios para lograr optimizar el proceso de gestión de pedidos del restaurante pollo Broster D La Zona. De igual forma, se logró determinar las funcionalidades básicas que cada aplicación (para administrador y para cliente) debía contemplar para satisfacer las necesidades de los comensales y del restaurante.
- El uso de las herramientas CASE, y principalmente de Balzamiq Mockups, permitió analizar los procesos de negocio de compra-venta, y los requerimientos y funcionalidades del sistema. Además, de diseñar los prototipos de interfaces de usuarios, los cuales sirvieron de guía en el proceso de desarrollo, y durante las reuniones con los clientes.
- La implementación de la tecnología backend en la nube, y los frameworks respectivos de React Native y Expo permitieron desarrollar apps móviles funcionales y sobre todo multiplataforma, simplificando el proceso de desarrollo. De manera que la app pueda ser empleada en dispositivos Android y iOS sin necesidad de codificar dos sistemas. Finalmente, el uso de Firebase y sus servicios, permitió almacenar los datos garantizando la seguridad, integridad y escalabilidad de los mismos.



### **3.4 RECOMENDACIONES.**

Luego de desarrollar la propuesta tecnológica se plantean las recomendaciones siguientes:

- Identificar, y analizar el proceso de negocio de la empresa antes de la elicitación y análisis de requerimientos, ya que de esta manera tanto el cliente y el encargado de la elicitación tendrán conocimiento del proceso que se está hablando, además de responder las dudas o plantear escenarios bajo los cuales son necesarios requerimientos específicos, o restricciones dentro de los mismos.
- Analizar los requerimientos y recursos que posee la empresa al momento de implementarlo, para lograr identificar el lenguaje de programación y los frameworks adecuados para el desarrollo.
- Establecer un blacklog o calendario que logre desarrollar el sistema de forma modular y ordena. Además de proponer las reuniones después de actividades específicas y de valor.
- Establecer los medios y formas de comunicación con las partes interesadas, de manera que, ante una emergencia o retraso en el cronograma, todos los miembros e interesados puedan mantenerse informados.

## BIBLIOGRAFÍA

- [1] S. M. Velasquez, D. E. M. Sossa, M. E. Zapata, M. E. G. Adasme, y J. P. Ríos, «Pruebas a aplicaciones móviles: avances y retos», *Lámpsakos*, n.º 21, pp. 39-50, 2019.
- [2] H. Liu y F. J. Sese, «The Impact of Mobile App Adoption on Cross-buying: The Moderating Roles of Product Category Characteristics and Adoption Timing», *Journal of Retailing*, feb. 2021, doi: 10.1016/j.jretai.2021.02.002.
- [3] Commerce Institute, «¡Ecuador vive un gran crecimiento en eCommerce!» <https://ecommerceday.org/2021/07/06/ecuador-vive-un-gran-crecimiento-en-ecommerce/> (accedido 8 de junio de 2022).
- [4] C. Rieger y T. A. Majchrzak, «Towards the definitive evaluation framework for cross-platform app development approaches», *Journal of Systems and Software*, vol. 153, pp. 175-199, jul. 2019, doi: 10.1016/j.jss.2019.04.001.
- [5] Y. Zhou, T. Yue, X. Liu, C. Shen, L. Tong, y Z. Ding, «Payment-Guard: Detecting fraudulent in-app purchases in iOS system», *Neurocomputing*, vol. 422, pp. 263-276, ene. 2021, doi: 10.1016/j.neucom.2020.10.007.
- [6] A. Ahmad, K. Li, C. Feng, S. M. Asim, A. Yousif, y S. Ge, «An Empirical Study of Investigating Mobile Applications Development Challenges», *IEEE Access*, vol. 6, pp. 17711-17728, 2018, doi: 10.1109/ACCESS.2018.2818724.
- [7] 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», *Journal of Biomedical Informatics*, vol. 105, p. 103420, may 2020, doi: 10.1016/j.jbi.2020.103420.
- [8] J. L. Escobar-Reynel, R. Baena-Navarro, B. Giraldo-Tobón, M. Macea-Anaya, y S. Castaño-Rivera, «Modelo de desarrollo para la construcción de aplicaciones móviles educativas», *TecnoLógicas*, vol. 24, n.º 52, 2021, Accedido: 12 de agosto de 2022. [En línea]. Disponible en: <https://www.redalyc.org/journal/3442/344268257008/>
- [9] K. J. McKibbin y M. Shabani, «Building a better mobile app marketplace: A legal and governance toolkit for app mediated genomics research», *Computer Law & Security Review*, vol. 46, p. 105707, sep. 2022, doi: 10.1016/j.clsr.2022.105707.
- [10] O. A. Varela, F. R. Sanabria, y S. V. Rodríguez, «Evaluación de una Arquitectura de Software», *Prospectiva*, vol. 19, n.º 2, p. 10, 2021.
- [11] G. Gatica, C. Contreras-Bolton, N. Venegas, O. Opazo, R. Linfati, y J. W. Escobar, «Una aplicación web, para asignación y ruteo de vehículos en caso de desastres», *Iteckne*, vol. 14, n.º 1, pp. 62-69, jun. 2017.
- [12] Z. M. Rodríguez, L. D. P. Rodríguez, y J. C. G. Suarez, «Arquitectura basada en Microservicios y DevOps para una ingeniería de software continua», *Industrial Data*, vol. 23, n.º 2, pp. 141-149, 2020.
- [13] R. E. Smith Colás, R. Cobo Alea, C. R. Vázquez Seisdedos, R. E. Smith Colás, R. Cobo Alea, y C. R. Vázquez Seisdedos, «Diseño de un sistema inalámbrico de monitorización electrocardiográfica para dispositivos Android», *Ingeniería Electrónica, Automática y Comunicaciones*, vol. 41, n.º 2, pp. 63-79, ago. 2020.
- [14] J. Calvo, J. Gracia, y E. Bayo, «Robust design to optimize client-server bi-directional communication for structural analysis web applications or

- services», *Advances in Engineering Software*, vol. 112, pp. 136-146, oct. 2017, doi: 10.1016/j.advengsoft.2017.04.010.
- [15] E. A. Lagunas, «Determinantes del gasto en e-commerce debido a la COVID-19: un análisis empírico para los micronegocios en Monterrey, México», *Paradigma económico. Revista de economía regional y sectorial*, vol. 13, n.º 1, pp. 119-145, 2021.
- [16] J. P. L. Bocanegra y L. F. B. Cardona, «Hábitos de consumo en plataformas e-commerce en adultos jóvenes de la ciudad de Bogotá», *Universidad & Empresa*, vol. 22, n.º 38, pp. 211-236.
- [17] M. O. G. González, J. M. S. Castaño, y L. M. M. Suárez, «Metodologías Ágiles de Desarrollo, un Caso de Aplicación en Medellín, Colombia 2016 - 2017», *Lámpsakos*, n.º 18, pp. 58-63, 2017.
- [18] F. J. L. Fernández y J. C. F. Rodríguez, «La metodología Lean Startup: desarrollo y aplicación para el emprendimiento», *Revista EAN*, n.º 84, pp. 79-95, 2018.
- [19] J. L. Gallego, S. L. Marín, C. L. Romero, J. D. V. Montoya, L. Á. E. Castrillón, y S. R. Pareja, «Desarrollo de un aplicativo móvil y web que calcule la huella de carbono en el sector educativo y transporte», *Lámpsakos*, n.º 23, pp. 45-55, 2020.
- [20] H. H. de G. Viola y I. de A. Braglia, «LEMMEX — an agile approach to service design consultancies», *ModaPalavra e-periódico*, vol. 14, n.º 31, pp. 222-243, 2021.
- [21] J. Ríos *et al.*, “MMS”, *Metodología para el Diseño y Desarrollo de Aplicaciones Móviles*. 2021. doi: 10.17993/IngyTec.2021.77.
- [22] C. Raibulet y F. Arcelli Fontana, «Collaborative and teamwork software development in an undergraduate software engineering course», *Journal of Systems and Software*, vol. 144, pp. 409-422, oct. 2018, doi: 10.1016/j.jss.2018.07.010.
- [23] A. Nunes, N. Freitas, M. Avila, J. Baldissera, M. Cruz, y K. Arceno, «GESTÃO POR PROCESSOS NO SERVIÇO DE HEMATOLOGIA E HEMOTERAPIA: UMA METODOLOGIA REDESENHADA PARA A BUSCA DE MAIOR EFICIÊNCIA E EFICÁCIA ORGANIZACIONAL», *Hematology, Transfusion and Cell Therapy*, vol. 43, p. S375, oct. 2021, doi: 10.1016/j.htct.2021.10.639.
- [24] F. Chen, L. Zhang, X. Lian, y N. Niu, «Automatically recognizing the semantic elements from UML class diagram images», *Journal of Systems and Software*, vol. 193, p. 111431, nov. 2022, doi: 10.1016/j.jss.2022.111431.
- [25] T. Tang, M. E. Lim, E. Mansfield, A. McLachlan, y S. D. Quan, «Clinician user involvement in the real world: Designing an electronic tool to improve interprofessional communication and collaboration in a hospital setting», *International Journal of Medical Informatics*, vol. 110, pp. 90-97, feb. 2018, doi: 10.1016/j.ijmedinf.2017.11.011.
- [26] E. Kurilovas y S. Kubilinskiene, «Lithuanian case study on evaluating suitability, acceptance and use of IT tools by students – An example of applying Technology Enhanced Learning Research methods in Higher Education», *Computers in Human Behavior*, vol. 107, p. 106274, jun. 2020, doi: 10.1016/j.chb.2020.106274.

- [27] F. Desai *et al.*, «HealthCloud: A system for monitoring health status of heart patients using machine learning and cloud computing», *Internet of Things*, vol. 17, p. 100485, mar. 2022, doi: 10.1016/j.iot.2021.100485.
- [28] S. A. Bello *et al.*, «Cloud computing in construction industry: Use cases, benefits and challenges», *Automation in Construction*, vol. 122, p. 103441, feb. 2021, doi: 10.1016/j.autcon.2020.103441.
- [29] M. Javaid, A. Haleem, R. P. Singh, S. Rab, R. Suman, y I. H. Khan, «Evolutionary trends in progressive cloud computing based healthcare: Ideas, enablers, and barriers», *International Journal of Cognitive Computing in Engineering*, vol. 3, pp. 124-135, jun. 2022, doi: 10.1016/j.ijcce.2022.06.001.
- [30] S. A. Murad, A. J. M. Muzahid, Z. R. M. Azmi, M. I. Hoque, y M. Kowsher, «A review on job scheduling technique in cloud computing and priority rule based intelligent framework», *Journal of King Saud University - Computer and Information Sciences*, vol. 34, n.º 6, Part A, pp. 2309-2331, jun. 2022, doi: 10.1016/j.jksuci.2022.03.027.
- [31] S. C. Olisa, C. N. Asiegbu, J. E. Olisa, B. O. Ekengwu, A. A. Shittu, y M. C. Eze, «Smart two-tank water quality and level detection system via IoT», *Heliyon*, vol. 7, n.º 8, p. e07651, ago. 2021, doi: 10.1016/j.heliyon.2021.e07651.
- [32] V. Chukowry, G. Nanuck, y R. K. Sungkur, «The future of continuous learning–Digital badge and microcredential system using blockchain», *Global Transitions Proceedings*, vol. 2, n.º 2, pp. 355-361, nov. 2021, doi: 10.1016/j.gltp.2021.08.026.
- [33] A. Olaya, H. D. Bohórquez, y A. R. Barrios, «CardioResyncApp: Un aplicativo móvil para recolectar datos de investigación en Cardiología», *Revista Colombiana de Cardiología*, vol. 27, n.º 4, pp. 270-275, jul. 2020, doi: 10.1016/j.rccar.2020.01.010.
- [34] S. Harous, M. El Menshawy, M. A. Serhani, y A. Benharref, «Mobile health architecture for obesity management using sensory and social data», *Informatics in Medicine Unlocked*, vol. 10, pp. 27-44, ene. 2018, doi: 10.1016/j.imu.2017.12.005.
- [35] O. Ivanochko, M. Gregus, M. Szalek, J. Roliński, y B. Stoliński, «City Tourism Services with Mobile Geolocation Sharing», *Procedia Computer Science*, vol. 191, pp. 49-56, ene. 2021, doi: 10.1016/j.procs.2021.07.010.
- [36] G. Albertengo, F. G. Debele, W. Hassan, y D. Stramandino, «On the performance of web services, google cloud messaging and firebase cloud messaging», *Digital Communications and Networks*, vol. 6, n.º 1, pp. 31-37, feb. 2020, doi: 10.1016/j.dcan.2019.02.002.
- [37] F. Kaimer y P. Brune, «Return of the JS: Towards a Node.js-Based Software Architecture for Combined CMS/CRM Applications», *Procedia Computer Science*, vol. 141, pp. 454-459, ene. 2018, doi: 10.1016/j.procs.2018.10.143.
- [38] K. Tserpes, M. Pateraki, y I. Varlamis, «Strand: scalable triliteration with Node.js», *Journal of Cloud Computing*, vol. 8, n.º 1, p. 16, nov. 2019, doi: 10.1186/s13677-019-0142-y.
- [39] N. Harrand, A. Benelallam, C. Soto-Valero, F. Bettega, O. Barais, y B. Baudry, «API beauty is in the eye of the clients: 2.2 million Maven dependencies reveal the spectrum of client-API usages», *Journal of*

- Systems and Software*, vol. 184, p. 111134, feb. 2022, doi: 10.1016/j.jss.2021.111134.
- [40] H. Brabra *et al.*, «On semantic detection of cloud API (anti)patterns», *Information and Software Technology*, vol. 107, pp. 65-82, mar. 2019, doi: 10.1016/j.infsof.2018.10.012.
- [41] D. Ganesh Chandra, «BASE analysis of NoSQL database», *Future Generation Computer Systems*, vol. 52, pp. 13-21, nov. 2015, doi: 10.1016/j.future.2015.05.003.
- [42] A. Gueidi, H. Gharsellaoui, y S. B. Ahmed, «Towards Unified Modeling for NoSQL Solution Based on Mapping Approach», *Procedia Computer Science*, vol. 192, pp. 3637-3646, ene. 2021, doi: 10.1016/j.procs.2021.09.137.
- [43] Y. Sukmana y Y. Rosmansyah, «The Use of Cloud Firestore For Handling Real-time Data Updates: An Empirical Study of Gamified Online Quiz», en *2021 2nd International Conference on Electronics, Communications and Information Technology (CECIT)*, dic. 2021, pp. 1239-1244. doi: 10.1109/CECIT53797.2021.00220.
- [44] T. Dorfer, L. Demetz, y S. Huber, «Impact of mobile cross-platform development on CPU, memory and battery of mobile devices when using common mobile app features», *Procedia Computer Science*, vol. 175, pp. 189-196, ene. 2020, doi: 10.1016/j.procs.2020.07.029.
- [45] T. A. Majchrzak, A. Bjørn-Hansen, y T.-M. Grønli, «Comprehensive Analysis of Innovative Cross-Platform App Development Frameworks», presentado en Mobile App Development Minitrack, ene. 2017. doi: 10.24251/HICSS.2017.745.
- [46] V. Kaushik, K. Gupta, y D. Gupta, «React Native Application Development», 2018, Accedido: 5 de septiembre de 2022. [En línea]. Disponible en: <https://papers.ssrn.com/abstract=3330011>
- [47] Shubham y S. Ali, «App Development using React Native, Expo and AWS», *International Journal of Trend in Scientific Research and Development*, vol. Volume-4, n.º Issue-4, jun. 2020, Accedido: 5 de septiembre de 2022. [En línea]. Disponible en: <https://www.ijtsrd.com/computer-science/other/31360/app-development-using-react-native-expo-and-aws/shiraz-ali>
- [48] X. Li, H. Qiao, y S. Wang, «Exploring evolution and emerging trends in business model study: a co-citation analysis», *Scientometrics*, vol. 111, n.º 2, pp. 869-887, may 2017, doi: 10.1007/s11192-017-2266-5.
- [49] M. Al-Kasasbeh, O. Abudayyeh, y H. Liu, «An integrated decision support system for building asset management based on BIM and Work Breakdown Structure», *Journal of Building Engineering*, vol. 34, p. 101959, feb. 2021, doi: 10.1016/j.jobbe.2020.101959.
- [50] A. P. L, D. B. R, y C. Mera, «Un método para la evaluación de la accesibilidad y la usabilidad en aplicaciones móviles», *TecnoLógicas*, vol. 23, n.º 48, pp. 99-117, 2020.

## ANEXOS

### ANEXO A: DESGLOSE DE REQUERIMIENTOS

La especificación de requerimientos funcionales se emplea con la finalidad de detallar las características, categorías, y actividades que se encuentran inmersas dentro de cada uno de los requerimientos funcionales generales.

A continuación, se detallan las especificaciones necesarias para cada requerimiento funcional (RF). Algunos de los requerimientos son accesibles tanto para el lado del cliente (clientes o comensales) como para el lado del administrador (empleados), es por ello que se especificarán las observaciones y características para cada app designada.

Tabla 14: Especificación de RF-1

ESPECIFICACIÓN DE REQUERIMIENTO FUNCIONAL			
<b>RF-1:</b>	Autenticación y recuperación de contraseñas		
<b>Categoría:</b>	Completo, prioritario		
<b>App Designada:</b>	Cliente, Administrador		
<b>Prioridad/negocio:</b>	Alto	<b>Responsable:</b>	Julio Campoverde
<b>Observación:</b>	<p>Los usuarios registrados, tanto en los empleados del negocio (app administrador) como los posibles clientes (app cliente), podrán acceder a las funcionalidades de la aplicación siempre y cuando autentifiquen sus credenciales (usuario y contraseña) ya creadas anteriormente.</p> <p>En caso del usuario olvidar la contraseña se podrá emitir un enlace para su respectiva recuperación, el mensaje será enviado a su correo electrónico registrado al usuario.</p>		
<b>Actores:</b>	Empleados, y clientes (comensales).		
<b>Pre-Condición:</b>	El usuario debe encontrarse ya registrado.		
<b>Post-Condición:</b>	<p>En caso de ingresar credenciales incorrectas se visualiza un mensaje de aviso.</p> <p>En caso de validar las credenciales correctamente se accede a la página principal.</p>		

**Fuente:** Elaboración propia.

Tabla 15: Especificación de RF-2

ESPECIFICACIÓN DE REQUERIMIENTO FUNCIONAL			
<b>RF-2:</b>	Gestión de usuarios		
<b>Categoría:</b>	Completo, prioritario		
<b>App Designada:</b>	Cliente, Administrador		
<b>Prioridad/negocio:</b>	Alto	<b>Responsable:</b>	Julio Campoverde
<b>Observación:</b>	<p>Los sub-requerimientos dentro de las apps respectivamente son:</p> <p><b>Cliente-Administrador:</b></p> <p><b>RF-2.1:</b> Ingreso de nuevo usuario</p> <p><b>RF-2.2:</b> Visualización del perfil de usuario</p> <p><b>RF-2.3:</b> Modificación de datos vinculados al perfil de usuario.</p> <p>Los permisos y roles se encuentran predeterminados dentro del sistema, contando con dos roles (1) administrador para los usuarios de la app administrador, y (2) cliente para los usuarios de la app cliente es decir para los comensales.</p>		
<b>Actores:</b>	Empleados, y clientes (comensales).		
<b>Pre-Condición:</b>	El usuario no debe estar creado.		
<b>Post-Condición:</b>	<ul style="list-style-type: none"> <li>• Una vez creado un nuevo usuario se muestra la pantalla para iniciar sesión.</li> <li>• Al visualizar la información personal, se muestran los campos ingresados durante el registro.</li> <li>• Se puede seleccionar el botón de modificación, para realizar cualquier cambio.</li> </ul>		

**Fuente:** Elaboración propia.

Tabla 16: Especificación de RF-3

ESPECIFICACIÓN DE REQUERIMIENTO FUNCIONAL			
<b>RF-3:</b>	Gestión de menú y categorías		
<b>Categoría:</b>	Completo, prioritario		
<b>App Designada:</b>	Administrador, Cliente		
<b>Prioridad/negocio:</b>	Alto	<b>Responsable:</b>	Julio Campoverde
<b>Observación:</b>	<p>Un menú va a contar con un mínimo de una sección, la cual se debe mantener siempre creada. A cada sección, hay como agregarle varios productos, para crear el menú:</p>		

	<b>Cliente-Administrador:</b> <b>RF-3.1:</b> Visualización de contenido de menú  <b>Administrador:</b> <b>RF-3.2:</b> Creación de nueva sección <b>RF-3.3:</b> Modificación de sección. <b>RF-3.4:</b> Eliminación de sección. <b>RF-3.5:</b> Agregar/ eliminar productos de una sección
<b>Actores:</b>	Empleados, y clientes (comensales).
<b>Pre-Condición:</b>	Para agregar nuevos productos a una sección del menú deben existir creados uno o más clientes previamente.
<b>Post-Condición:</b>	Una vez creada la sección, se procede a agregar o eliminar los productos a la misma, siempre y cuando mantengan disponibilidad.

**Fuente:** Elaboración propia.

Tabla 17: Especificación de RF-4

ESPECIFICACIÓN DE REQUERIMIENTO FUNCIONAL			
<b>RF-4:</b>	Gestión de productos y promociones		
<b>Categoría:</b>	Completo, prioritario		
<b>App Designada:</b>	Administrador		
<b>Prioridad/negocio:</b>	Alto	<b>Responsable:</b>	Julio Campoverde
<b>Observación:</b>	<p>Para poder crear un producto de manera correcta, se deben crear las secciones o categoría a las cuales va a pertenecer. Posteriormente se crea el producto, el cual debe contar con los siguientes campos:</p> <ul style="list-style-type: none"> <li>- Código</li> <li>- Descripción.</li> <li>- Imagen del producto (en formato png o jpg)</li> <li>- Especificación si es promoción.</li> <li>- Precio.</li> <li>- Disponibilidad, y estado.</li> <li>- Etiquetas (máximo 3).</li> </ul> <p>Por otro lado, las acciones y funcionalidades dentro de este</p>		



	<p>requerimiento se designan en su mayoría a la app del administrador.</p> <p><b>Administrador:</b></p> <p><b>RF-4.1:</b> Creación de nuevo producto</p> <p><b>RF-4.3:</b> Modificación de producto.</p> <p><b>RF-4.4:</b> Eliminación de producto.</p> <p><b>RF-4.5:</b> Modificar y visualizar la disponibilidad del producto.</p> <p>Sin embargo, una funcionalidad estará disponible tanto para la app del <b>cliente</b> como la del administrador.</p> <p><b>RF-4.6:</b> Visualización del detalle y disponibilidad del producto.</p>
<b>Actores:</b>	Empleados, y clientes (comensales).
<b>Pre-Condición:</b>	La sección a la cual pertenece el producto debe estar creada.
<b>Post-Condición:</b>	Una vez creado los productos, se deben agregar al menú para poder ser solicitados (comprados).

**Fuente:** Elaboración propia.

Tabla 18: Especificación de RF-5

ESPECIFICACIÓN DE REQUERIMIENTO FUNCIONAL			
<b>RF-5:</b>	Gestión de pedidos		
<b>Categoría:</b>	Completo, prioritario		
<b>App Designada:</b>	Administrador, cliente		
<b>Prioridad/negocio:</b>	Alto	<b>Responsable:</b>	Julio Campoverde
<b>Observación:</b>	<p>La gestión de pedidos divide sus funcionalidades en las apps de administrador y cliente. En la parte del administrador se presentan las notificaciones siempre que se realice un pedido, con la siguiente información:</p> <ul style="list-style-type: none"> <li>- Nombre del cliente.</li> <li>- Detalles del pedido (cantidad y producto).</li> <li>- Dirección y de que sucursal fue designada.</li> </ul> <p>Además, el usuario o empleado, podrá visualizar los pedidos</p>		

	<p>según varias categorías.</p> <ul style="list-style-type: none"> <li>- Pendientes</li> <li>- Enviados (empleando método FIFO).</li> <li>- Entregados.</li> </ul> <p>El empleado es el encargado de asignar el repartidor correspondiente, mientras que el cliente puede conocer el nombre y hora aproximada de entrega.</p> <p>El administrador podrá conocer de manera general los pedidos que tiene pendientes, sin embargo, en el panel solo se mostrarán los pedidos del día actual, además de permitir la búsqueda por cliente mediante una barra de búsqueda.</p> <p><b>Administrador:</b></p> <p><b>RF-5.1:</b> Notificación de pedido.</p> <p><b>RF-5.3:</b> Visualización de clasificación de pedidos.</p> <p><b>RF-5.4:</b> Visualización de panel de pedidos.</p> <p><b>RF-5.5:</b> Búsqueda de pedidos por clientes y por fechas.</p> <p><b>RF-5.6:</b> Modificación del estado de pedidos.</p> <p><b>Cliente:</b></p> <p><b>RF-5.8:</b> Realizar pedido.</p> <p><b>RF-5.9:</b> Visualizar orden de compra.</p> <p><b>Administrador-Cliente:</b></p> <p><b>RF-5.9:</b> Visualización del detalle del pedido.</p>
<b>Actores:</b>	Empleados, y clientes (comensales).
<b>Pre-Condición:</b>	Los productos y menú deben estar creados y disponibles en la sucursal.
<b>Post-Condición:</b>	<ul style="list-style-type: none"> <li>- Se podrá visualizar el estado del pedido.</li> <li>- Se puede visualizar un resumen de las ventas realizadas durante el día</li> <li>- Una vez entregado el pedido, se procede a modificar el estado del mismo.</li> </ul>

**Fuente:** Elaboración propia.

Tabla 19: Especificación de RF-6

ESPECIFICACIÓN DE REQUERIMIENTO FUNCIONAL			
<b>RF-6:</b>	Gestión de entrega		
<b>Categoría:</b>	Completo, prioritario		
<b>App Designada:</b>	Administrador		
<b>Prioridad/negocio:</b>	Alto	<b>Responsable:</b>	Julio Campoverde
<b>Observación:</b>	<p>El proceso de entrega se realiza netamente en la app administrador, el cual consiste en asignar al repartidor correspondiente a cada pedido, y posteriormente modificar el estado del pedido. Esta funcionalidad se encuentra estrechamente relacionada con la gestión de ubicaciones ya que en cada entrega la ubicación del cliente una vez realizado el pedido no puede ser modificado.</p> <p><b>Administrador:</b></p> <p><b>RF-6.1:</b> Asignación de repartidor.</p> <p><b>RF-6.2:</b> Visualización de información del cliente (nombre, numero de teléfono, y dirección).</p> <p><b>RF-6.3:</b> Modificación del estado de pedidos.</p>		
<b>Actores:</b>	Empleados		
<b>Pre-Condición:</b>	Se debe generar una orden de pedido		
<b>Post-Condición:</b>	Una vez entregado el pedido, se debe modificar manualmente el estado del pedido a “entregado”.		

**Fuente:** Elaboración propia.

Tabla 20: Especificación de RF-7

ESPECIFICACIÓN DE REQUERIMIENTO FUNCIONAL			
<b>RF-7:</b>	Gestión de sucursales		
<b>Categoría:</b>	Completo, prioritario		
<b>App Designada:</b>	Administrador		
<b>Prioridad/negocio:</b>	Alto	<b>Responsable:</b>	Julio Campoverde
<b>Observación:</b>	El negocio puede administrar la información de cada sucursal, así como visualizar el menú y productos disponibles en cada una de ellas.		

	<b>Administrador:</b> <b>RF-5.1:</b> Crear nueva sucursal. <b>RF-5.2:</b> Modificar la información de cada sucursal. <b>RF-5.3:</b> Eliminar sucursales. <b>RF-5.4:</b> Visualización de menú y estado de productos en cada sucursal.
<b>Actores:</b>	Empleados
<b>Pre-Condición:</b>	Deben estar los productos asignados a una sucursal
<b>Post-Condición:</b>	Ninguna.

**Fuente:** Elaboración propia.

Tabla 21: Especificación de RF-8

ESPECIFICACIÓN DE REQUERIMIENTO FUNCIONAL			
<b>RF-8:</b>	Gestión de ubicaciones		
<b>Categoría:</b>	Completo, prioritario		
<b>App Designada:</b>	Cliente		
<b>Prioridad/negocio:</b>	Alto	<b>Responsable:</b>	Julio Campoverde
<b>Observación:</b>	<p>El usuario puede gestionar las ubicaciones que posee, puede crear diferentes ubicaciones, sin embargo, al momento de comprar se debe asignar la ubicación a la cual recibirá el pedido. Para el ingreso de nuevas direcciones se solicitarán los siguientes datos.</p> <ul style="list-style-type: none"> <li>- Dirección</li> <li>- Sector.</li> <li>- Referencia.</li> <li>- Ciudad</li> <li>- Nombre de dirección.</li> <li>- Ubicación de dirección en mapa.</li> </ul> <p><b>Cliente:</b></p> <p><b>RF-9.1:</b> Agregar nueva dirección.</p> <p><b>RF-9.2:</b> Modificar direcciones.</p> <p><b>RF-9.3:</b> Eliminar dirección.</p>		

<b>Actores:</b>	Cliente (Comensal)
<b>Pre-Condición:</b>	El usuario debe estar registrado.
<b>Post-Condición:</b>	El usuario podrá usar las direcciones al momento de realizar la compra.

**Fuente:** Elaboración propia.

## ANEXO B: DEFINICIÓN DE HISTORIAS DE USUARIO

En las siguientes tablas se especifican las historias de usuario según el desglose de requerimientos funcionales.

El módulo de usuarios incluye los procesos de ingreso, registro y gestión de cada uno de los usuarios, como se muestra en las siguientes historias de usuario.

Tabla 22: Historia de usuario No1

Historia de usuario	
<b>Número:</b> 1	<b>App designada:</b> Todos
<b>Nombre de historia:</b> Inicio de sesión (Autenticación)	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Bajo
<b>Requerimiento funcional:</b> RF-1	<b>Iteración asignada:</b>
<b>Programador responsable:</b> Julio Campoverde	
<b>Descripción:</b> El usuario ingresará al sistema y a sus funcionalidades respectivas mediante el inicio de sesión con autenticación de usuario y contraseña.	
<b>Observaciones:</b> El usuario debe estar registrado en el sistema, si no se posee credenciales debe registrarse.	

**Fuente:** Elaboración propia.

Tabla 23: Historia de usuario No2

Historia de usuario	
<b>Número:</b> 2	<b>App designada:</b> Todos
<b>Nombre de historia:</b> Recuperación de contraseñas	
<b>Prioridad en negocio:</b> Baja	<b>Riesgo en desarrollo:</b> Bajo
<b>Requerimiento funcional:</b> RF-1	<b>Iteración asignada:</b>
<b>Programador responsable:</b> Julio Campoverde	
<b>Descripción:</b> En caso de el usuario olvidar su contraseña, podrá recuperarla mediante el envío del mismo al correo electrónico vinculado al cliente.	
<b>Observaciones:</b> Solo se emplea el método de envío a correo electrónico.	

**Fuente:** Elaboración propia.

Tabla 24: Historia de usuario No3

Historia de usuario	
<b>Número:</b> 3	<b>App designada:</b> Todos
<b>Nombre de historia:</b> Registro de usuario	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Baja
<b>Requerimiento funcional:</b> RF-2	<b>Iteración asignada:</b>
<b>Programador responsable:</b> Julio Campoverde	
<p><b>Descripción:</b> El usuario si no cuenta con las credenciales para el inicio de sesión, puede crear un nuevo usuario desde la pantalla principal.</p> <p>Los datos a ingresar para el perfil del usuario en la <b>app administrador:</b> Email, Nombre, Fecha de nacimiento, Foto, Teléfono, Ciudad, Sexo.</p> <p>Los datos a ingresar para el perfil del usuario en la <b>app cliente:</b> Email, ciudad, id ciudad, fecha de nacimiento, email, nombres, foto, sexo, teléfono.</p>	
<b>Observaciones:</b> No se puede registrar un mismo usuario dos veces.	

**Fuente:** Elaboración propia.

Tabla 25: Historia de usuario No4

Historia de usuario	
<b>Número:</b> 4	<b>App designada:</b> Todos
<b>Nombre de historia:</b> Gestión de perfil de usuario	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Baja
<b>Requerimiento funcional:</b> RF-2	<b>Iteración asignada:</b>
<b>Programador responsable:</b> Julio Campoverde	
<p><b>Descripción:</b> El usuario de la app administrador como de la app cliente, debe ser capaz de visualizar la información personal, así como ser capaz de modificar los datos en cualquier momento.</p>	
<b>Observaciones:</b> Solo se modificaran los campos permitidos.	

**Fuente:** Elaboración propia.

El módulo de menú cuenta con la gestión de categorías (secciones), y gestión de productos.

Tabla 26: Historia de usuario No5

Historia de usuario	
<b>Número:</b> 5	<b>App designada:</b> Administrador
<b>Nombre de historia:</b> Gestión de secciones.	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Baja
<b>Requerimiento funcional:</b> RF-3	<b>Iteración asignada:</b>
<b>Programador responsable:</b> Julio Campoverde	
<b>Descripción:</b> El administrador debe ser capaz de crear, modificar, y eliminar secciones para catalogar los diferentes productos, o también denominadas categorías, de manera que la creación del menú sea ordenada y secciones, por ejemplo: “bebidas, comida, aperitivos, etc...”	
<b>Observaciones:</b> Debe existir como mínimo una categoría para el menú. Las categorías deben ser diferentes entre sí.	

**Fuente:** Elaboración propia.

Tabla 27: Historia de usuario No6

Historia de usuario	
<b>Número:</b> 6	<b>App designada:</b> Todos
<b>Nombre de historia:</b> Gestión de menú	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Baja
<b>Requerimiento funcional:</b> RF-3	<b>Iteración asignada:</b>
<b>Programador responsable:</b> Julio Campoverde	
<b>Descripción:</b> El <b>administrador</b> podrá crear un menú mediante la asignación de categorías y productos, presentando principalmente el nombre y foto del producto. Debe poder cambiar la disponibilidad del producto de manera sencilla.  El <b>usuario</b> (comensal) debe ser capaz de visualizar el menú, los productos, y la disponibilidad de los mismos de forma clara.	
<b>Observaciones:</b> Un producto puede pertenecer a varias categorías, y una categoría puede tener varios productos.	

**Fuente:** Elaboración propia.



Tabla 28: Historia de usuario No7

Historia de usuario	
<b>Número:</b> 7	<b>App designada:</b> Todos
<b>Nombre de historia:</b> Gestión de productos y promociones	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Baja
<b>Requerimiento funcional:</b> RF-4	<b>Iteración asignada:</b>
<b>Programador responsable:</b> Julio Campoverde	
<b>Descripción:</b> El <i>administrador</i> podrá ingresar, modificar o eliminar un nuevo producto. Así como asignarle la(s) categoría(s) correspondientes. Al crear un producto se puede marcar si esta es una promoción o no. El <i>usuario</i> debe ser capaz de visualizar el detalle y disponibilidad del producto.	
<b>Observaciones:</b> Un producto puede pertenecer a varias categorías.	

**Fuente:** Elaboración propia.

El módulo de compra incluye las notificaciones y gestión de pedidos, órdenes de compra de los mismos y entrega de los mismos.

Tabla 29: Historia de usuario No8

Historia de usuario	
<b>Número:</b> 8	<b>App designada:</b> Administrador
<b>Nombre de historia:</b> Notificaciones de pedidos	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Baja
<b>Requerimiento funcional:</b> RF-5	<b>Iteración asignada:</b>
<b>Programador responsable:</b> Julio Campoverde	
<b>Descripción:</b> El administrador debe ser capaz de recibir notificaciones siempre que se realice un pedido, con la opción de visualizar los detalles del mismo (nombre del cliente, detalles del pedido, dirección y sucursal a cuál fue designada). Cada pedido posee un estado, el cual puede ser Pendiente, Enviado, o Entregado.	
<b>Observaciones:</b> Los estados del pedido deben ser modificados manualmente.	

**Fuente:** Elaboración propia.

Tabla 30: Historia de usuario No9

Historia de usuario	
<b>Número:</b> 9	<b>App designada:</b> Cliente
<b>Nombre de historia:</b> Realización de compra	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Baja
<b>Requerimiento funcional:</b> RF-5	<b>Iteración asignada:</b>
<b>Programador responsable:</b> Julio Campoverde	
<p><b>Descripción:</b> El cliente puede realizar la compra, la cual generara un pedido. Para ello el cliente debe seleccionar los productos y agregarlos al carrito, luego seleccionar los datos de facturación y la dirección a la cual se realizará el envío. Una vez seleccionado se emite los datos de facturas y se procede a seleccionar el método de pago (solo en efectivo). Al finalizar se emite un resumen de la información del pedido para ser confirmada y efectuar el proceso de compra.</p>	
<p><b>Observaciones:</b> Al momento de seleccionar el método de pago, se debe especificar la denominación del billete, de manera que se pueda realizar un proceso de cobranza rápido y eficiente.</p>	

**Fuente:** Elaboración propia.

Tabla 31: Historia de usuario No10

Historia de usuario	
<b>Número:</b> 10	<b>App designada:</b> Cliente
<b>Nombre de historia:</b> Visualización de orden de compra	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Baja
<b>Requerimiento funcional:</b> RF-5	<b>Iteración asignada:</b>
<b>Programador responsable:</b> Julio Campoverde	
<p><b>Descripción:</b> El usuario al realizar una compra podrá visualizar la orden de compra respectiva, así como un detalle del pedido realizado. La orden de compra es fundamental al momento de la entrega, ya que es el medio de validación del usuario que recibirá el pedido.</p>	
<p><b>Observaciones:</b> El cliente puede ver las ordenes de compras pasadas.</p>	

**Fuente:** Elaboración propia.

Tabla 32: Historia de usuario No11

Historia de usuario	
<b>Número:</b> 11	<b>App designada:</b> Administrador
<b>Nombre de historia:</b> Gestión de entrega	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Baja
<b>Requerimiento funcional:</b> RF-6	<b>Iteración asignada:</b>
<b>Programador responsable:</b> Julio Campoverde	
<b>Descripción:</b> El administrador podrá asignar al repartidor correspondiente a cada pedido, y posteriormente modificar el estado del pedido. Esta funcionalidad se encuentra estrechamente relacionada con la gestión de ubicaciones ya que en cada entrega la ubicación del cliente una vez realizado el pedido no puede ser modificado.	
<b>Observaciones:</b> Para verificar que es la persona correcta a entregar el pedido se debe mostrar la orden de pedido emitida al momento de la compra.	

**Fuente:** Elaboración propia.

El módulo de sucursales cuenta con la siguiente historia de usuario.

Tabla 33: Historia de usuario No12

Historia de usuario	
<b>Número:</b> 12	<b>App designada:</b> Administrador
<b>Nombre de historia:</b> Gestión de sucursales	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Baja
<b>Requerimiento funcional:</b> RF-7	<b>Iteración asignada:</b>
<b>Programador responsable:</b> Julio Campoverde	
<b>Descripción:</b> El administrador podrá crear, modificar o eliminar las sucursales del negocio. Así como visualizar el menú y estado de productos (disponibilidad) en cada sucursal.	
<b>Observaciones:</b> El cambio en la disponibilidad de un producto en una sucursal se realiza en la gestión de productos.	

**Fuente:** Elaboración propia.

Finalmente, la gestión de ubicaciones incluye lo siguiente:

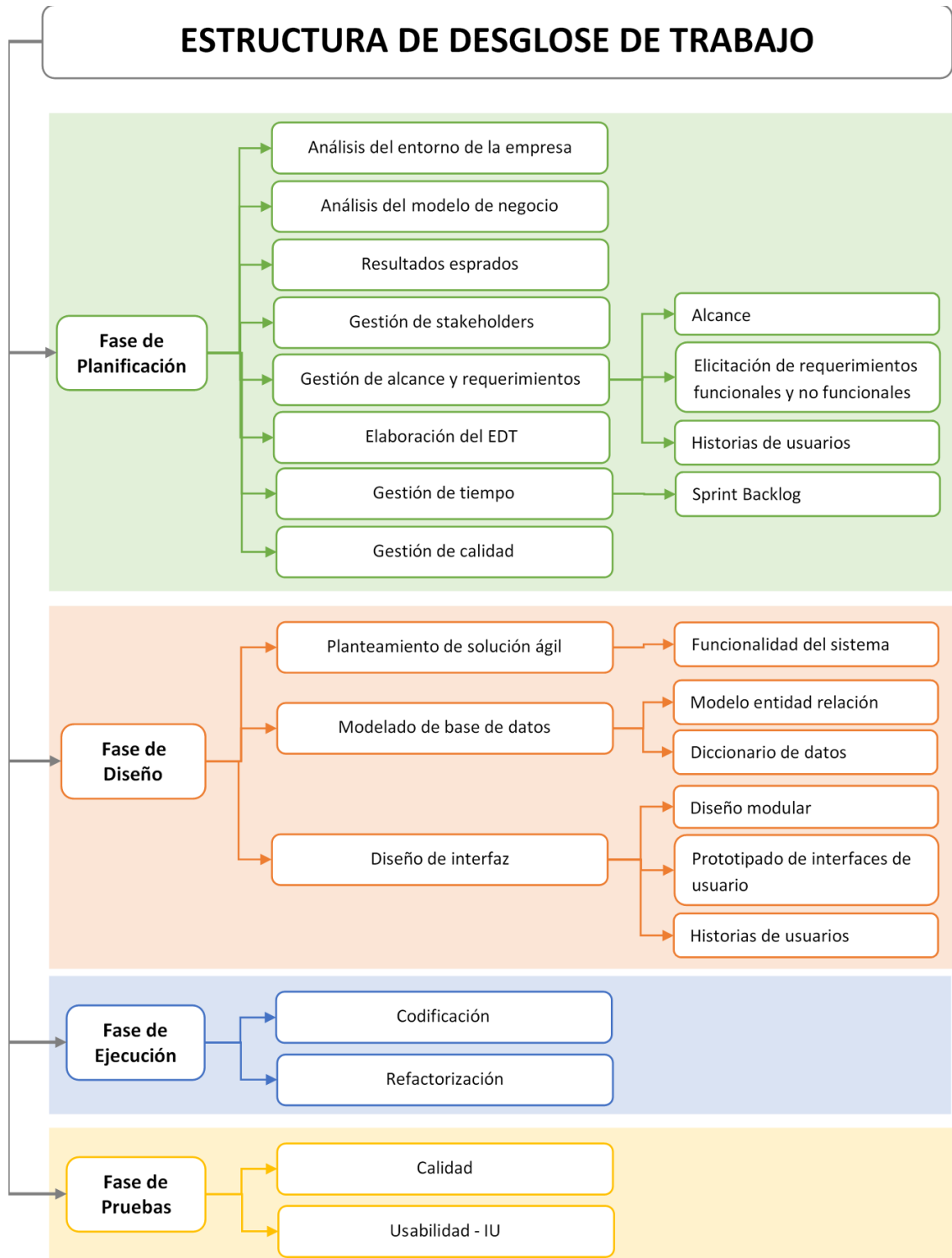
Tabla 34: Historia de usuario No13

Historia de usuario	
<b>Número:</b> 13	<b>App designada:</b> Cliente
<b>Nombre de historia:</b> Gestión de ubicaciones	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Baja
<b>Requerimiento funcional:</b> RF-8	<b>Iteración asignada:</b>
<b>Programador responsable:</b> Julio Campoverde	
<p><b>Descripción:</b> El cliente podrá agregar una nueva ubicación, así como modificar o eliminar una ubicación ya existente. Puede buscar la ubicación que desee mediante un mapa. Adicionalmente debe ingresar los siguientes campos en las ubicaciones que cree: dirección, sector, referencia, ciudad, nombre de dirección (por ejemplo, casa u oficina), y seleccionar la ubicación de dirección en el mapa.</p>	
<p><b>Observaciones:</b> El usuario podrá seleccionar la dirección de entrega al momento de realizar la compra.</p>	

**Fuente:** Elaboración propia.

## ANEXO C: ESTRUCTURA DE DESGLOSE DE TRABAJO

Figura 43 EDT del proyecto



Fuente: Elaboración Propia