



PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR

Unidad Académica de Formación Técnica y Tecnológica – PUCE TEC

**PAGINA WEB Y SISTEMA DE PEDIDOS Y RESERVAS DE PRODUCTOS
DEL MINI MARKET F.J**

**Proyecto de titulación previo a la obtención del título de:
Tecnología Superior En Desarrollo De Software**

Autor:

Fernando Daniel Vaca Buitrón

Tutor:

Christian Roberto Tapia Gaibor

Quito, Ecuador

2025

DEDICATORIA

Quiero dedicar esta tesis tanto a mi familia como a mis profesores, quienes siempre confiaron en mí. Gracias a esa confianza he logrado llegar a este punto de mi vida, después de un camino lleno de esfuerzo y sacrificios que, al final, dan sus frutos en todos los conocimientos que he adquirido.

TABLA DE CONTENIDOS

Dedicatoria.....	2
Lista de tablas	5
Lista de figuras	6
Introducción.....	9
Capítulo I.....	16
Levantamiento de Requisitos y Diseño del Sistema	16
1.1 Antecedentes	16
1.2 Metodología de Desarrollo.....	16
1.3 Etapas del Proceso.....	17
1.4 Viabilidad del Proyecto	34
Capítulo II.....	36
Construcción del Sistema.....	36
2.1 Alcance.....	36
2.2 Análisis de diseño.....	37
2.3 Mapa de navegación.....	37
2.4 Lógica de negocio	38
2.5 Modelo de Base de Datos.....	39
2.6 Requisitos Funcionales.....	41
2.7 Requisitos No Funcionales.....	42
2.8. Diseño de la Arquitectura del Sistema	43
Capítulo III	46

Pruebas y Estabilización	46
3.1 Tipos de pruebas realizadas.....	46
3.2 Herramientas utilizadas:.....	55
3.3 Casos de prueba:.....	55
3.4 Estabilización del sistema	56
Conclusiones.....	58
Recomendaciones	59
Referencias bibliográficas	60
Anexos	62

LISTA DE TABLAS

Tabla 1: Historias de Usuarios.....	18
Tabla 2: Casos de Uso	21
Tabla 3: Tabla de Funcionalidades	22
Tabla 4: Interacción 1 Frontend.....	22
Tabla 5: Interacción 2 Frontend.....	25
Tabla 6: Interacción 3 Frontend.....	26
Tabla 7: Interacción 1 Backend	26
Tabla 8: Interacción 2 Backend	27
Tabla 9: Interacción 3 Backend	28
Tabla 10: Hardware Requerido.....	33
Tabla 11: Análisis de Diseño.....	37
Tabla 12: Requisitos Funcionales.....	41
Tabla 13: Requisitos no Funcionales	42
Tabla 14: Casos de Prueba.....	55

LISTA DE FIGURAS

Figura 1: Flujo del Trabajo de GitHub	32
Figura 2: Arquitectura del Proyecto	38
Figura 3: Modelo de Base de Datos.....	39
Figura 4: Arquitectura del Sistema.....	43
Figura 5: Página principal.....	46
Figura 6: Carrito de Compra.....	47
Figura 7: Método de Pago	47
Figura 8: Inicio de Administración del Propietario	48
Figura 9: Panel del Administrador.....	48
Figura 10: Gestión de Usuarios Administrativos	49
Figura 11: Gestión de Administración de Productos.....	49
Figura 12: Gestión de Pedidos y reservas.....	50
Figura 13: Pruebas en POSTMAN	50
Figura 14: Pruebas de Interfaz.....	51
Figura 15: Pruebas en Rendimiento	52
Figura 16: Protección de los datos.....	53
Figura 17: Inicio de sesión con JWT	53

DECLARACIÓN y AUTORIZACIÓN

Yo, **Fernando Daniel Vaca Buitrón** con C.I. 1753066404 autor(a) del trabajo de Titulación intitulado: **“Página Web y Sistema de Pedidos y Reservas de Productos del Mini Market F.J”**, previa a la obtención del título de Tecnólogo Superior en Desarrollo de Software en la Unidad Académica de Formación Técnica y Tecnológica PUCE TEC:

1.- Declaro tener pleno conocimiento de la obligación que tiene la Pontificia Universidad Católica del Ecuador, de conformidad con el artículo 144 de la Ley Orgánica de Educación Superior, de entregar a la SENESCYT en formato digital una copia del referido trabajo de graduación para que sea integrado al Sistema Nacional de Información de la Educación Superior del Ecuador para su difusión pública respetando los derechos de autor.

2.- Autorizo a la Pontificia Universidad Católica del Ecuador a difundir a través de sitio web de la Biblioteca de la PUCE el referido trabajo de titulación, respetando las políticas de propiedad intelectual de Universidad.

Quito, 15 de febrero 2026



Fernando Daniel Vaca Buitrón

C.I. 1753066404

Agradecimientos

Mi agradecimiento hacia mi familia, pilar fundamental que durante mi carrera universitaria ha estado junto a mi brindándome todo su apoyo. De manera especial este agradecimiento a mi papá Daniel Vaca, por ser mi guía en cada etapa de crecimiento en mi vida. A mi mamá Kerly Buitrón, quien con su cariño y apoyo siempre me ha motivado a no rendirme y a esforzarme por alcanzar cada una de las metas que me propongo, y a mi abuelita Glorita Cúñez por todo su apoyo en este proceso. Gracias a su amor y a su apoyo incondicional, he podido llegar a este importante punto de mi vida.

INTRODUCCIÓN

La transformación digital significa “superar el temor al cambio, que en efecto las personas están en una zona de confort donde cambiar la forma en que hacen las cosas representa un reto importante, pero no imposible” (Sanabria, 2021). Es por ello, que contar con una página web y un sistema de pedidos y reservas hoy en día, es una necesidad para todo tipo de negocio, que ayuda a optimizar la comunicación entre sus clientes, ampliar su negocio, aumentar la rentabilidad y la competitividad.

El proyecto que a continuación se describe, tiene como objetivo el diseño, desarrollo e implementación de una página web para el MINI MARKET F.J., orientado tanto a la visualización informativa de productos como a la incorporación de un módulo funcional de pedidos y reservas. Este establecimiento comercial ofrece una gran variedad de productos de consumo diario y de primera necesidad, como alimentos, bebidas, artículos de limpieza y cuidado personal, entre otros.

El objetivo de la página web es proporcionar a los clientes información clara y accesible sobre los productos disponibles con la opción de poder seleccionarlos, añadirlos a un carrito de compras, visualizar el monto total y enviar una solicitud de reserva para retirar posteriormente en tienda.

Para asegurar la calidad, funcionalidad y eficacia de la página web y el sistema de pedidos y reservas, se aplicarán principios fundamentales de usabilidad, seguridad de la información, diseño responsive y arquitectura tecnológica. Esto permitirá que la página web y el módulo funcional sea accesible desde diversos dispositivos como computadoras, tabletas y teléfonos móviles, adaptándose de manera fluida a cada uno de ellos, lo cual maximiza su alcance y utilidad para los distintos clientes.

Problema Técnico/Tecnológico

Actualmente, el MINI MARKET F.J. no cuenta con una página web y un sistema de pedidos y reservas, que le permita gestionar de manera digital, la interrelación entre sus clientes, poder promocionar sus productos y ampliar su alcance comercial. El no disponer de una página web y un sistema de pedidos y reservas, restringe la posibilidad de atender a sus clientes fuera del entorno físico del local.

La gestión de pedidos y reservas de productos del MINI MARKET F.J. en la actualidad el propietario lo viene ejecutando manualmente, lo que ocasiona discrepancias, demoras en la atención, e impide la organización interna. Además, dificulta tener una programación anticipada de compras por parte del cliente y una adecuada reserva de productos. Los clientes, al no contar con una plataforma digital que les permita consultar la disponibilidad de productos, seleccionar los artículos deseados o generar solicitudes de retiro anticipado, recurren a otros establecimientos.

Objetivo General:

Diseñar una página web para el MINI MARKET F.J. que incluya un módulo funcional de pedidos y reservas de productos, utilizando React para el desarrollo del frontend y Nest.js para el backend, con el objetivo de optimizar la comunicación con los clientes, fortalecer la presencia digital del negocio y mejorar su gestión operativa.

Objetivos Específicos:

- Desarrollar una interfaz web responsive y de fácil navegación que permita a los clientes acceder a información de la variedad de productos de consumo diario y de primera necesidad ofrecidos por el MINI MARKET F.J.
- Implementar un módulo funcional de pedidos y reservas en línea, que permita a los clientes seleccionar productos, agregarlos a un carrito virtual y enviar

solicitudes de reserva, optimizando el proceso de atención y reduciendo errores operativos.

- Fortalecer la presencia del MINI MARKET F.J. en el mercado local, mediante el desarrollo de una página web y el sistema de pedidos y reservas, que permitan ampliar su alcance, mejorar su posicionamiento y atraer nuevos clientes.

Marco Conceptual

A continuación, se describen las principales tecnologías que se implementaron y su aporte al proyecto:

HTML (Hypertext Markup Language)

HTML se define como “un lenguaje de marcado, lo que significa que utiliza etiquetas para definir la estructura y el formato del contenido” (Martínez, 2025). Este lenguaje será utilizado para estructurar el contenido de la página web, como encabezados, párrafos, formularios, botones; entre otros.

CSS (Cascading Style Sheets)

CSS se define como “lenguaje de estilos utilizado para describir la presentación de documentos escritos en HTML o XML. Su propósito es definir como se debe mostrar los elementos en pantalla, papel o en otros medios. Permite un control preciso sobre la apariencia de las páginas web” (Martínez, 2025). Se utilizará para que sea amigable de manera visual el sitio web, aspectos como colores, estilos y distribución de los elementos en la pantalla.

JavaScript

JavaScript se presenta “como un lenguaje de desarrollo de aplicaciones cliente/servidor a través de Internet” (Maza, 2012). Se empleará JavaScript para el

desarrollo del backend, permitiendo gestionar el carrito de compras, actualizar de manera automática el total de productos seleccionados y procesar las solicitudes de reserva. Su implementación facilitará una experiencia de usuario más ágil y eficiente.

React (Frontend)

El React se define como “Una biblioteca desarrollada por Facebook que facilita la creación de interfaces de usuario mediante componentes reutilizables y una gestión eficiente del estado “ (Martínez, 2025). Esta herramienta permitirá organizar los distintos elementos de la página web y garantizar que se presenten de manera adecuada en la interfaz para los usuarios.

Nest.js (Backend)

Se define como “desarrollo web basado en Node.js que utiliza TypeScript para proporcionar una estructura de programación sólida y altamente escalable.” (Itequia, 2023). El backend ayudará con la arquitectura modular y escalable para la construcción de la API del sistema. Nest.js permitirá gestionar de forma eficiente las operaciones relacionadas con la información de productos, pedidos y reservas.

PostgreSQL (Base de datos)

“PostgreSQL utiliza un modelo cliente/servidor para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando” (Molina, 2017). Este sistema es para gestionar bases de datos, que servirán para el almacenamiento y organización de la información que se coloque en la página web, lo que va a permitir almacenar datos de manera estructurada, como la información de productos, pedidos y reservas.

Visual Studio Code (Entorno de desarrollo)

Se define como “un editor de código fuente gratuito y de código abierto desarrollado por Microsoft. Es uno de los más populares entre los desarrolladores debido a su capacidad para soportar múltiples lenguajes de programación y sistemas operativos” (Gamarra, 2023). El desarrollo de la página web y su módulo, se realizará utilizando el entorno de desarrollo integrado Visual Studio Code (VS Code); misma que permitirá escribir, depurar y mantener el código del frontend y del backend de manera eficiente.

Git

Git es un “sistema de control de versiones distribuido que se ha convertido en una herramienta esencial para la mayoría de los desarrolladores de software” (Moure, 2024). El cual se utilizará como sistema de control de versiones.

GitHub (Control de versiones y repositorio)

GitHub se define como “una plataforma que ofrece la posibilidad de almacenar sus proyectos en forma de repositorios remotos en los servidores de la empresa. Este servicio ofrece una funcionalidad para trabajar en equipo y colaborar eficazmente entre desarrolladores.” (Samuel, 2018).

Vercel (Despliegue Frontend)

La herramienta de Vercel “es una plataforma de desarrollo en la nube diseñada para crear, desplegar y escalar aplicaciones web de alto rendimiento de manera rápida y automatizada.” (Vercel, 2026). Con esta herramienta se publicará la página web y se pondrá a disposición de los clientes a través del servidor en línea, lo que nos facilitará visualizar los avances del proyecto, como también en realizar pruebas de optimización.

Además, con Vercel se aprovechará del sistema de control de versiones de GitHub para llevar un registro de los cambios realizados en el código, asegurando que cada actualización quede documentada. Esto servirá para contar con un historial ordenado desde el desarrollo inicial hasta la implementación del sistema.

Render (Despliegue Backend)

Render es considerada una plataforma que se aloja en la nube y permite cargar proyectos para realizar el despliegue. En el desarrollo de este proyecto, se utilizó Render para hacer el despliegue del backend, implementando las herramientas que la plataforma ofrece para la gestión y el mantenimiento del sistema. De manera similar a Vercel, Render se aprovecha de su integración nativa con GitHub, lo que facilita la automatización de las actualizaciones de la lógica del sistema de forma inmediata.

Supabase (Despliegue de la Base de datos)

Supabase es “una plataforma de "Backend como Servicio" (BaaS) de código abierto, diseñada como una alternativa potente a Firebase” (Supabase, 2026). Esta plataforma se usó para alojar la base de datos hecha en PostgreSQL.

Metodología de Desarrollo en Cascada

“Un proceso de desarrollo en cascada consiste en la realización de una serie de etapas consecutivas de trabajo (en este caso diseño y desarrollo) hasta la finalización del proyecto que culmina con la entrega completa de la aplicación según los requerimientos definidos” (Carraro, 2015). Esta metodología es adecuada para el desarrollo del proyecto ya que se desarrolla mediante etapas secuenciales, lo que ayudará a llevar un control ordenado de cada etapa. Se inicia con la recolección y análisis de requisitos, diseño del sistema, la implementación de la página web y sistema de pedidos y reservas, pruebas del sistema, despliegue y la documentación y mantenimiento. Esta metodología

es amigable ya que cada etapa debe estar terminada antes de iniciar la siguiente, lo que permite reducir errores.

CAPÍTULO I

Levantamiento de Requisitos y Diseño del Sistema

1.1 Antecedentes

El MINI MARKET F.J es un establecimiento dedicado a abastecer a sus clientes con productos. Sin embargo, la falta de integración con nuevas tecnologías limita su capacidad para ofrecer atención fuera del entorno físico del local, lo que afecta la eficiencia en sus procesos internos, tales como el control de inventario, el registro de ventas, la administración de productos y la atención al cliente.

Ante esta problemática, se propone el desarrollo de una página web con un sistema de pedidos y reservas, que permite optimizar estos procesos. De esta manera, se busca reducir errores en los registros y proporcionar una herramienta confiable que permita al minimarket mantener un control completo, organizado y eficiente de sus operaciones diarias.

1.2 Metodología de Desarrollo

Para llevar a cabo el proyecto se ha utilizado la metodología en cascada, la cual se inicia con la recolección y análisis de requisitos, diseño del sistema, la implementación de la página web y sistema de pedidos y reservas, pruebas del sistema, despliegue y la documentación y mantenimiento.

Para cumplir con estas actividades, primero se ha identificado los actores principales para el desarrollo del proyecto:

- **Jefe de Proyecto:** Tapia Gaibor Christian Roberto - Responsable de la planificación, supervisión y coordinación de las fases del proyecto, verificando que cada fase del proyecto se cumple en los plazos establecidos.

- **Stakeholder:** Propietario del MINI MARKET F.J. Jerez Jerez Freddy Marcelo
– Responsable de definir los requerimientos necesarios para el funcionamiento de la página web y el sistema de pedidos y reservas. Además, se considerará como responsable de las gestiones administrativas (gestión de usuarios administrativos, gestión de productos, gestión de pedidos)
- **Desarrollador Full-Stack:** Vaca Buitrón Fernando Daniel - Encargado de la integración del frontend y backend, implementando la lógica del sistema y realizando las conexiones del backend hacia la base de datos. Además, se encarga de realizar las pruebas unitarias para asegurar un correcto funcionamiento de la página web y el sistema de pedidos y reservas.
- **Cliente:** Persona que accede a los servicios de la página web, con la opción de realiza pedidos y reservas.

1.3 Etapas del Proceso

1.3.1 Recolección y análisis de requisitos

Se realizó el levantamiento de información sobre el funcionamiento del MINI MARKET F.J., para lo cual se ejecutó una encuesta de manera física al propietario (**ver anexo 1**), abordando temas relacionados con la gestión de productos (inventario) y el manejo de pedidos y reservas; obteniendo los siguientes resultados positivos: se requiere utilizar un sistema informático para realizar pedidos y reservas, se requiere la creación de una página web, los beneficios al contar con una página web y sistema de pedidos y reservas como: mejor atención al cliente, incremento en ventas y mejor control de inventario.

Adicional, se realizaron encuestas presenciales (**ver anexo 2**) a cinco (5) clientes, para conocer si les parece atractivo implementar una página web y sistema de pedidos y reservas del MINI MARKET F.J; obteniendo los siguientes resultados

positivos: están de acuerdo en la creación de la página web y sistema de pedidos y reservas, ya que esto ayudaría a buscar una mejor disponibilidad en los productos, ahorro de tiempo y comodidad.

Requisitos del Sistema:

Durante el desarrollo de la etapa de recolección y análisis de requisitos se identificó las necesidades y expectativas de los clientes y del propietario; para lo cual, el implementar la metodología en cascada ayudó a analizar las historias de usuarios y definir los requisitos funcionales (Prioridad Alta) y no funcionales (Prioridad media) de la página web y el sistema de pedidos y reservas, como se detalla a continuación:

Tabla 1: Historias de Usuarios

ID	Historia de Usuario	Rol	Prioridad
US01	Como propietario, quiero iniciar sesión en el sistema, para acceder a las funcionalidades administrativas de forma segura.	Propietario	Alta
US02	Como propietario, quiero gestionar los productos (crear, editar y eliminar), para mantener actualizado el inventario del MINI MARKET F.J.	Propietario	Alta
US03	Como propietario, quiero visualizar los pedidos y reservas realizados por los clientes, para llevar un control adecuado de las solicitudes.	Propietario	Alta
US04	Como propietario, quiero actualizar el estado de los pedidos y reservas, para reflejar su avance como “En proceso”, “Completado” o “Cancelado”.	Propietario	Alta
US05	Como propietario, quiero gestionar a los usuarios administrativos (crear, editar y eliminar), para administrar el acceso al	Propietario	Alta

	sistema administrativo de forma segura y organizada.		
US06	Como cliente, quiero visualizar los productos disponibles con sus precios, para seleccionar los productos que deseo comprar o reservar.	Ciente	Alta
US07	Como cliente, quiero realizar un pedido de productos, para retirarlos posteriormente en el MINI MARKET F.J.	Ciente	Alta
US08	Como cliente, quiero realizar reservas de productos, para asegurar su disponibilidad.	Ciente	Alta
US09	Como cliente, quiero buscar productos disponibles en la página web, para facilitar la realización de mis pedidos o reservas.	Ciente	Media
US10	Como cliente, quiero realizar mis pedidos y reservas desde mi dispositivo, para agilizar el proceso de compra.	Ciente	Media
US11	Como cliente, quiero seleccionar el método de pago (efectivo o De Una), para completar mis pedidos de manera conveniente y segura.	Ciente	Media

Casos de usos y Flujos de las Historias:

- **US01: Autenticación de propietario:**

El proceso inicia cuando el propietario ingresa sus credenciales de acceso. Estas credenciales son verificadas en la base de datos y permite el acceso al panel administrativo en caso de ser correcta.

- **US02: Gestión de productos:**

El propietario accede al módulo de productos, el cual permite realizar una correcta gestión integral (CRUD) de los productos que se ofrecen en el MINI

MARKET, incluyendo un registro de nuevos productos, la actualización de datos existentes y la eliminación del producto seleccionado en el sistema.

- **US03: Visualización de pedidos y reservas:**

El sistema muestra el listado de los pedidos y reservas realizado, permitiendo al propietario revisar la información correspondiente a cada solicitud.

- **US04: Actualización del estado de pedidos y reservas:**

El propietario selecciona un pedido o reserva y modifica su estado; para lo cual, el sistema muestra al propietario las dos opciones de cambio de estado y guarda estos cambios para proceder con la actualización de la visualización.

- **US05: Gestión de usuarios administrativos:**

El propietario ingresa al módulo de usuarios donde puede registrar nuevos administradores, modificar datos existentes o desactivar cuentas.

- **US06: Visualización de productos:**

El cliente accede a la página principal donde el sistema muestra el catálogo de productos disponibles con su respectivo precio.

- **US07: Seleccionar productos y registro de pedido:**

El cliente selecciona los productos, confirma el pedido y el sistema registra la solicitud.

- **US08: Reserva de productos:**

El cliente selecciona el producto a reservar y confirma la solicitud. El sistema guarda la reserva y la muestra al propietario.

- **US09: Búsqueda de productos:**

El cliente utiliza la barra de búsqueda ingresando un nombre. El sistema filtra y muestra los productos que tengan esa coincidencia.

- **US10: Pedidos desde dispositivos:**

El sistema permite al cliente acceder desde cualquier dispositivo y realizar pedidos o reservas de manera optimizada.

- **US11: Selección del método de pago:**

Al confirmar el pedido o reserva, el sistema mostrará los métodos de pago disponibles, permitiendo al cliente seleccionar y registrar la opción de su preferencia.

Los casos de uso se definieron con los requisitos funcionales y no funcionales, estos casos describen de manera clara y estructurada las principales funcionalidades que el sistema del Mini Market F.J. ofrece a sus actores.

Tabla 2: Casos de Uso

Código	Nombre del Caso de Uso	Actores	Descripción
CU01	Autenticar Usuario	Propietario	Valida las credenciales del usuario para permitir el acceso al panel administrativo.
CU02	Gestionar Inventario	Propietario	Permite gestionar el inventario mediante CRUD para poder visualizar, crear, actualizar y eliminar los productos que están en el catálogo.
CU03	Controlar Pedidos y Reservas	Propietario	Permite visualizar la lista global de solicitudes y actualizar sus estados (En proceso, Completado, etc.).
CU04	Gestionar Usuarios	Propietario	Administración de cuentas con permisos administrativos para el sistema.
CU05	Consultar Catálogo	Cliente	Permite al cliente visualizar, buscar y filtrar los productos disponibles con sus precios.
CU06	Registrar Pedido/Reserva	Cliente	El cliente selecciona productos, elige el método de pago (Efectivo/De Una) y confirma su solicitud.

Creación de la lista de funcionalidades:

La lista de las funcionalidades se desarrolló pensando en priorizar las funciones críticas de la página web y el sistema de pedidos y reservas, optimizando la distribución de las tareas asignadas.

Tabla 3: Tabla de Funcionalidades

ID	Funcionalidad	Prioridad
F01	Autenticación de usuarios administrativos	Alta
F02	Gestionar productos	Alta
F03	Registro de pedidos y reservas	Alta
F04	Visualización y control de pedidos	Alta
F05	Búsqueda de productos	Media
F06	Selección del método de pago	Media

Planificación de las Iteraciones:

Durante la planificación de las iteraciones, el proyecto se dividió en dos apartados principales: el frontend y el backend. Cada uno con sus respectivas tareas y sus iteraciones, esto permitió una mejor organización al momento de desarrollar la página web y el sistema de pedidos y reservas.

Frontend:

- **Interacción 1:**

Tabla 4: Interacción 1 Frontend

ID	Nombre de la tarea	Prioridad	Estimación	Tarea
T1	Diseñar estructura base	Alta	1 días	https://github.com/Ferchov15/minimarket-fj-frontend/issues/1

	del proyecto Next.js			
T2	Creación de la página estática de la página principal	Alta	1 días	https://github.com/Ferchov15/minimarket-fj-frontend/issues/2
T3	Creación de la página estática del carrito de compra	Alta	1 días	https://github.com/Ferchov15/minimarket-fj-frontend/issues/3
T4	Creación de la página estática de la página administrativa	Alta	1 días	https://github.com/Ferchov15/minimarket-fj-frontend/issues/4
T5	Implementación de componentes y unión de paginas	Alta	3 días	https://github.com/Ferchov15/minimarket-fj-frontend/issues/5
T6	Establecer conexión con el backend (Página principal)	Alta	2 días	https://github.com/Ferchov15/minimarket-fj-frontend/issues/6
T7	Establecer conexión con el backend (Carrito de compra)	Alta	2 días	https://github.com/Ferchov15/minimarket-fj-frontend/issues/7
T8	Establecer conexión con el backend (Pagina administrativa)	Alta	1 días	https://github.com/Ferchov15/minimarket-fj-frontend/issues/8

T9	Crear formulario de inicio de sesión	Alta	1 días	https://github.com/Ferchov15/minimarket-fj-frontend/issues/9
T10	Creación de página de inicio de sesión	Alta	1 días	https://github.com/Ferchov15/minimarket-fj-frontend/issues/10
T11	Persistencia de sesión (localStorage)	Alta	2 días	https://github.com/Ferchov15/minimarket-fj-frontend/issues/11
T12	Creación de una página de bienvenida	Alta	1 días	https://github.com/Ferchov15/minimarket-fj-frontend/issues/12
T13	Creación de página administrativa (CRUD productos).	Alta	3 días	https://github.com/Ferchov15/minimarket-fj-frontend/issues/13
T14	Creación de página administrativa (CRUD usuarios administrativos)	Alta	2 días	https://github.com/Ferchov15/minimarket-fj-frontend/issues/14
T15	Creación de las páginas administrativas	Alta	4 días	https://github.com/Ferchov15/minimarket-fj-frontend/issues/15
T16	Corrección del error de stock productos	Media	1 días	https://github.com/Ferchov15/minimarket-fj-frontend/issues/25
T16	Mejorar la edición de productos y usuarios	Media	2 días	https://github.com/Ferchov15/minimarket-fj-frontend/issues/26

- **Interacción 2:**

Tabla 5: Interacción 2 Frontend

ID	Nombre de la tarea	Prioridad	Estimación	Tarea
T1	Filtro por categorías para los productos	Media	1 días	https://github.com/Ferchov15/minimarket-fj-frontend/issues/16
T2	Creación de las páginas con el detalle de los productos por categoría.	Media	1 días	https://github.com/Ferchov15/minimarket-fj-frontend/issues/17
T3	Proceso de confirmación de pedidos	Alta	2 días	https://github.com/Ferchov15/minimarket-fj-frontend/issues/18
T4	Realizar Pruebas al sistema completo	Alta	2 días	https://github.com/Ferchov15/minimarket-fj-frontend/issues/19
T5	Corrección de errores descubiertos durante las pruebas	Alta	2 día	https://github.com/Ferchov15/minimarket-fj-frontend/issues/20
T6	Implementación de método de pago	Alta	1 día	https://github.com/Ferchov15/minimarket-fj-frontend/issues/27

- **Interacción 3:**

Tabla 6: Interacción 3 Frontend

ID	Nombre de la tarea	Prioridad	Estimación	Tarea
T1	Despliegue a la web de la página funcional (Vercel)	Alta	2 días	https://github.com/Ferchov15/minimarket-fj-frontend/issues/21
T2	Realizar pruebas de usuarios reales	Media	3 días	https://github.com/Ferchov15/minimarket-fj-frontend/issues/22
T3	Corrección de errores encontrados en las pruebas de usuarios	Alta	2 días	https://github.com/Ferchov15/minimarket-fj-frontend/issues/23
T4	Finalización del proyecto en Frontend	Alta	1 días	https://github.com/Ferchov15/minimarket-fj-frontend/issues/24

Backend:

- **Interacción 1:**

Tabla 7: Interacción 1 Backend

ID	Nombre de la tarea	Prioridad	Estimación	Tarea
T1	Creación de la base del backend	Alta	1 días	https://github.com/Ferchov15/minimarket-fj-backend/issues/1
T2	Configuración básica del servidor	Alta	2 días	https://github.com/Ferchov15/minimarket-fj-backend/issues/2

T3	Crear conexión a PostgreSQL	Alta	2 días	https://github.com/Ferchov15/minimarket-fj-backend/issues/3
T4	Creación de principales modelos para el proyecto	Alta	2 días	https://github.com/Ferchov15/minimarket-fj-backend/issues/4
T5	Creación de los Endpoints CRUD básicos	Alta	3 días	https://github.com/Ferchov15/minimarket-fj-backend/issues/5
T6	Test de conexión con frontend	Alta	2 días	https://github.com/Ferchov15/minimarket-fj-backend/issues/6

- **Interacción 2:**

Tabla 8: Interacción 2 Backend

ID	Nombre de la tarea	Prioridad	Estimación	Tarea
T1	Creación del CRUD de usuarios administrativos.	Alta	2 días	https://github.com/Ferchov15/minimarket-fj-backend/issues/7
T2	Realizar pruebas de inicio de sesión	Alta	2 días	https://github.com/Ferchov15/minimarket-fj-backend/issues/8
T3	Actualizar stock automáticamente según el pedido	Alta	2 días	https://github.com/Ferchov15/minimarket-fj-backend/issues/9

- **Interacción 3:**

Tabla 9: Interacción 3 Backend

ID	Nombre de la tarea	Prioridad	Estimación	Tarea
T1	Despliegue del backend funcional a la web (Render)	Alta	3 días	https://github.com/Ferchov15/minimarket-fj-backend/issues/10
T2	Realizar pruebas de usuarios reales	Alta	4 días	https://github.com/Ferchov15/minimarket-fj-backend/issues/11
T3	Corrección de errores encontrados en las pruebas de usuarios	Alta	3 días	https://github.com/Ferchov15/minimarket-fj-backend/issues/12
T4	Finalización del proyecto en Backend	Alta	1 días	https://github.com/Ferchov15/minimarket-fj-backend/issues/13

Criterios de finalización:

Para el proyecto, se utilizaron los criterios de finalización, para asegurar que cada funcionalidad o las tareas del sistema cumplan con sus requisitos antes de ser consideradas finalizadas. Estos criterios fueron definidos por el desarrollador Full-Stack y el jefe del proyecto, para una correcta finalización de las funcionalidades, algunos de estos criterios son:

- Revisión de los criterios y de los requisitos funcionales, para tener una culminación del 100%.
- La correcta implementación del código, para evitar errores críticos al momento de hacer la ejecución e impedir fallos visibles en la interfaz.

- Verificación de la conexión entre los módulos del sistema (frontend, backend y la base de datos).
- Validación del funcionamiento por parte del desarrollador Full-Stack y revisión por el jefe del proyecto.
- Para mejorar el rendimiento del sistema se realizará una adecuada depuración del código
- Ejecutar las pruebas de funcionalidad, verificando que el sistema este cumpliendo con los requisitos establecidos.

1.3.2 Diseño del sistema

Con los requisitos del sistema implementados, se procede a realizar la implementación del diseño de la arquitectura del sistema, abarcando desde la capa de presentación (frontend), posterior la lógica de negocio (backend) y finalmente la base de datos. Para eso se incorporó los siguientes módulos:

- **Administrador:** Registro e inicio de sesión de usuarios administrativos.
- **Administrador:** incorporación, actualización y eliminación de los productos del MINI MARKET F.J.
- **Clientes:** la creación y registro de pedidos y reservas
- **Administrador:** permite la visualización, gestión y la actualización del estado de cada pedido.

El desarrollo técnico de la página web y el sistema de pedidos y reservas se distribuyó de la siguiente manera:

Backend: Se ha implementado NestJS que es un framework basado en Node.js y construido sobre Express, utilizando controladores, middleware de autenticación de los

usuarios mediante tokens y utilizando Sequelize ORM, lo que ha servido para la gestión de la base de datos PostgreSQL.

Frontend: Se ha implementado el framework de Next.js, utilizando componentes reutilizables y aprovechando sus capacidades de renderizado y estructura por páginas. Para la construcción de una interfaz clara y responsiva se utilizaron librerías de estilos y componentes visuales.

Control de versiones: Se implementó la herramienta de GitHub para tener un control de las versiones de la página web y el sistema de pedidos y reservas.

Gestión de tareas: Se implementó la herramienta GitHub Projects con el tablero tipo Kanban para organizar las funcionalidades de la página web y el sistema de pedidos y reservas.

1.3.3 Implementación

En esta etapa, se desarrolló la página web y sistema de pedidos y reservas siguiendo lo definido en la etapa de diseño.

El desarrollo de ambos componentes se ejecutó dentro del entorno de Visual Studio Code, el cual facilitó la edición y depuración de los cambios realizados a la página web. Además, el trabajo está versionado y administrado mediante GitHub, lo que permitirá mantener un control ordenado de los cambios realizados para la página web y el sistema de pedidos y reservas.

1.3.4 Pruebas del sistema

En esta etapa, se llevó a cabo pruebas interactivas para asegurar el correcto funcionamiento de la página web y el sistema de pedidos y reservas, entre las principales pruebas ejecutadas se incluyen las siguientes:

Pruebas del Backend (API): Validación del correcto uso de los endpoints descritos en las rutas y sus respectivas respuestas, con errores que fueron solucionados.

Pruebas funcionales: Verificación del correcto funcionamiento de los módulos de inicio de sesión, gestión de usuarios administrativos, gestión de productos, así como la visualización y actualización del estado de los pedidos y reservas, con varios errores de visualización, datos dañados, entre otros, mismos que fueron solventados.

Pruebas de interfaz: Validación del comportamiento de la página web frente a los clientes, así como su funcionamiento en distintos dispositivos, con errores que fueron solucionados.

Pruebas de seguridad: Se verificó la correcta generación de los tokens mediante JWT y su adecuada protección de las rutas administrativas, con errores de implementación de cookies y del localstore, que fueron cambiados por el uso de tokens.

1.3.5 Despliegue

Una vez superada todas las pruebas, se realiza el despliegue utilizando las herramientas de Vercel, Render y supabase respectivamente.

Frontend: Cada vez que se incorporan cambios nuevos en la rama principal de GitHub, la plataforma los detecta y realiza automáticamente la actualización de la página web.

Backend: Se usó la plataforma Render, para que el sistema realice una actualización automática.

Base de datos: se usó la plataforma de Supabase, para la correcta administración y almacenamiento de la información relacionada con los usuarios, productos, pedidos y reservas.

1.3.6 Herramientas y software Utilizados

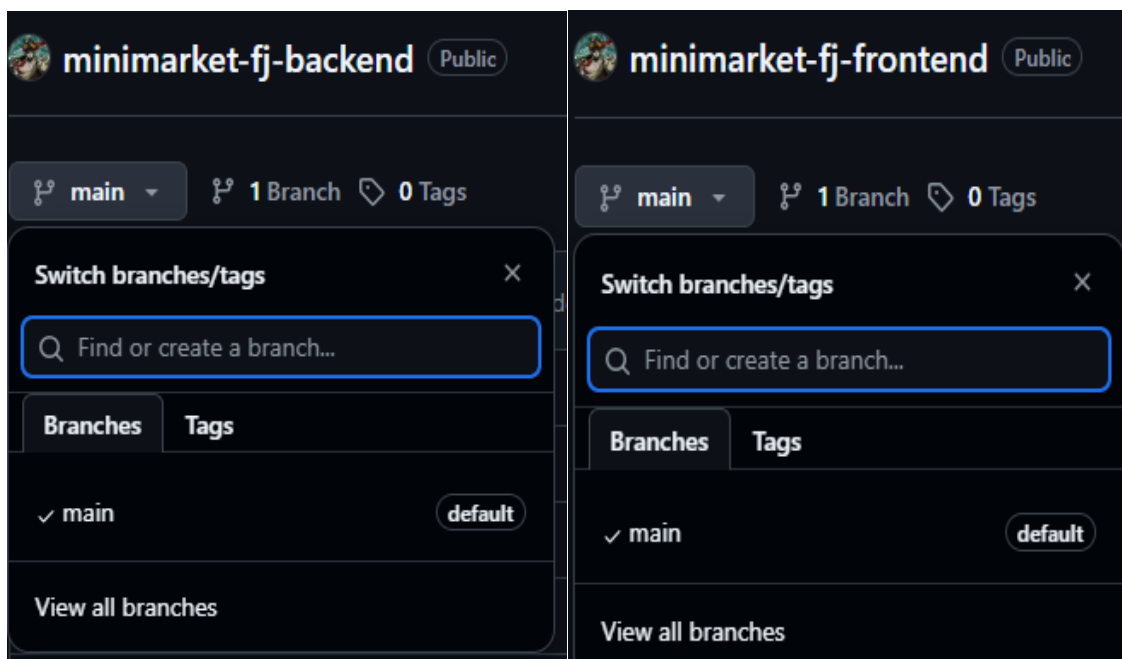
A continuación, se describen las principales tecnologías que se implementaron y su aporte al proyecto:

- HTML (Hypertext Markup Language).

- CSS (Cascading Style Sheets).
- Bootstrap (Librería CSS).
- JavaScript.
- Postman (Pruebas de API)
- Next.js (Frontend)
- Nest.js (Backend)
- PostgreSQL (Base de datos).
- Visual Studio Code (Entorno de desarrollo).
- Git.
- GitHub (Control de versiones y repositorio).
- Vercel (Despliegue Frontend).
- Render (Despliegue Backend).
- SupaBase (Despliegue Base de datos).

Flujo de trabajo con Git:

Figura 1: Flujo del Trabajo de GitHub



Debido a que el desarrollo del proyecto fue realizado por un único desarrollador, se trabajó principalmente en la rama main en los repositorios tanto del frontend como del backend. Cada avance fue documentado mediante el uso de commits estructurados y descriptivos para garantizar la trazabilidad del código.

No obstante cabe aclarar que para el uso de buenas prácticas del control de versiones, es recomendable el uso del Git-flow, para mantener una organización clara y permitir la escalabilidad del proyecto en caso de futuras ampliaciones del equipo.

1.3.7 Hardware Requerido

Tabla 10: Hardware Requerido

COMPONENTE	CLIENTE WEB	SERVIDOR BACKEND
PROCESADOR	Intel i3 o Ryzen 3	Intel i5 o Ryzen 5
MEMORIA RAM	4 GB	8 GB
ALMACENAMIENTO	120 GB libres	120 GB libres (SSD recomendado)
SISTEMA OPERATIVO	Windows, Linux o macOS	Linux o Windows Server
NAVEGADOR	Google Chrome, Edge o Firefox	No es necesario un navegador
SOFTWARE REQUERIDO	Navegador web	Node.js, Nest.js, PostgreSQL

1.3.8 Documentación y mantenimiento

Se realizó una capacitación al dueño del MINI MARKET F.J. en el uso de la página web y el sistema de pedidos y reservas y se entregó el manual de usuario (**Ver anexo 3**). Cabe indicar que se brindará el mantenimiento y actualización del sistema por un periodo de 3 meses de forma gratuita; el propietario ha indicado que a partir de los 3 meses contratará un técnico y se procederá a entregar la documentación técnica que

sustenta el desarrollo de la página web y el sistema de pedidos y reservas. Esto con la finalidad de brindar la garantía necesaria para su mantenimiento, actualización y escalabilidad futura.

Documentación técnica: Se entregará la estructura básica de la base de datos (Tablas y relaciones), así como el catálogo de endpoints de la API y el acceso directo a los repositorios de frontend y del backend alojados en la plataforma de GitHub.

1.4 Viabilidad del Proyecto

La viabilidad del proyecto es alta desde tres aspectos: técnico, económico y operativo. Este enfoque asegura que el desarrollo de la página web y sistema de pedidos y reservas, resulte ser factible, eficiente y sostenible en el tiempo.

1.4.1 Viabilidad Técnica

El proyecto es viable técnicamente gracias al uso de herramientas modernas y ampliamente comprobadas. Para el desarrollo de la página web y del sistema de pedidos y reservas del Mini Market F.J., se utilizó tecnologías React, Next.js, Node.js, NestJS y PostgreSQL. Además, se incorporaron herramientas de apoyo como Visual Studio Code, PgAdmin y Postman, que simplificaron las tareas de codificación, pruebas y administración de la base de datos.

1.4.2 Viabilidad Económica

El proyecto es viable económicamente gracias al uso de software libre, lo cual elimina los costos para la adquisición de licencias.

Recursos: no hay costos por adquisición de equipos de computación, ya que se utilizaron recursos tecnológicos previamente disponibles.

En conclusión, la ejecución del proyecto resulta viable económicamente, ya que su desarrollo no generó gastos y se maximizó la eficiencia de los recursos disponibles.

No obstante, se recomienda al propietario una vez culminado el tiempo de tres meses gratuitos destinado al mantenimiento y actualización del sistema, se contemple un costo mensual estimado entre \$5,00 y \$10,00 dólares correspondiente a la suscripción de un hosting básico. Esta inversión contribuirá a mejorar el ancho de banda y el rendimiento del servidor, garantizar la disponibilidad permanente del sistema 24/7, ofrecer mayor capacidad de almacenamiento de la información y permitir la utilización de un dominio propio (www.minimarketfj.com).

1.4.3 Viabilidad Operativa

En cuanto a la viabilidad operativa, el propietario del MINI MARKET F.J. participó en las fases de pruebas, validación de funcionalidades, en la capacitación del manejo de la página web, lo que garantizó que el sistema sea amigable para los clientes y se encuentre en estado óptimo para que sea operativo y sea sostenible en el tiempo.

CAPÍTULO II

Construcción del Sistema

2.1 Alcance

La página web y el sistema de pedidos y reservas tiene como propósito proporcionar una plataforma web que permita al MINI MARKET F.J. gestionar sus productos y ofrecer a los clientes un sistema de pedidos y reservas en línea de manera eficiente y organizada. Sin embargo, es importante mencionar que el sistema no contará con las siguientes acciones:

- pago con tarjeta de crédito y débito.
- módulo de marketing digital para promocionar marcas específicas del mercado.
- detalle de ofertas, cupones, promociones.
- servicios de delivery a domicilio.

Objetivos específicos del alcance:

- Mostrar el catálogo de los productos disponibles a los clientes, incluyendo su información y precio.
- Permitir al propietario gestionar los productos del minimarket (crear, editar y eliminar).
- Habilitar el sistema de pedidos y reservas en línea, para que los clientes ejecuten el sistema de forma remota y segura.
- Garantizar la visualización y gestión de pedidos y reservas por parte del propietario.
- Incorporar la selección de métodos de pago al momento de realizar la confirmación de los pedidos o reservas.
- Permitir el acceso al sistema administrativo mediante autenticación segura.

2.2 Análisis de diseño

Para el levantamiento de requisitos se usaron las historias de usuario.

En la etapa de recolección y análisis de requisitos, mediante las historias de usuario, se obtuvo la siguiente información:

Tabla 11: Análisis de Diseño

Código	Descripción
HU01	Permitir al propietario iniciar sesión en el sistema administrativo de forma segura.
HU02	Gestionar productos del minimarket (crear, editar y eliminar).
HU03	Visualizar y actualizar el estado de pedidos y reservas.
HU04	Gestionar usuarios administrativos del sistema.
HU05	Visualizar y buscar productos disponibles para clientes.
HU06	Permitir a los clientes realizar pedidos y reservas desde la página web
HU07	Seleccionar el método de pago al confirmar un pedido o reserva

2.3 Mapa de navegación

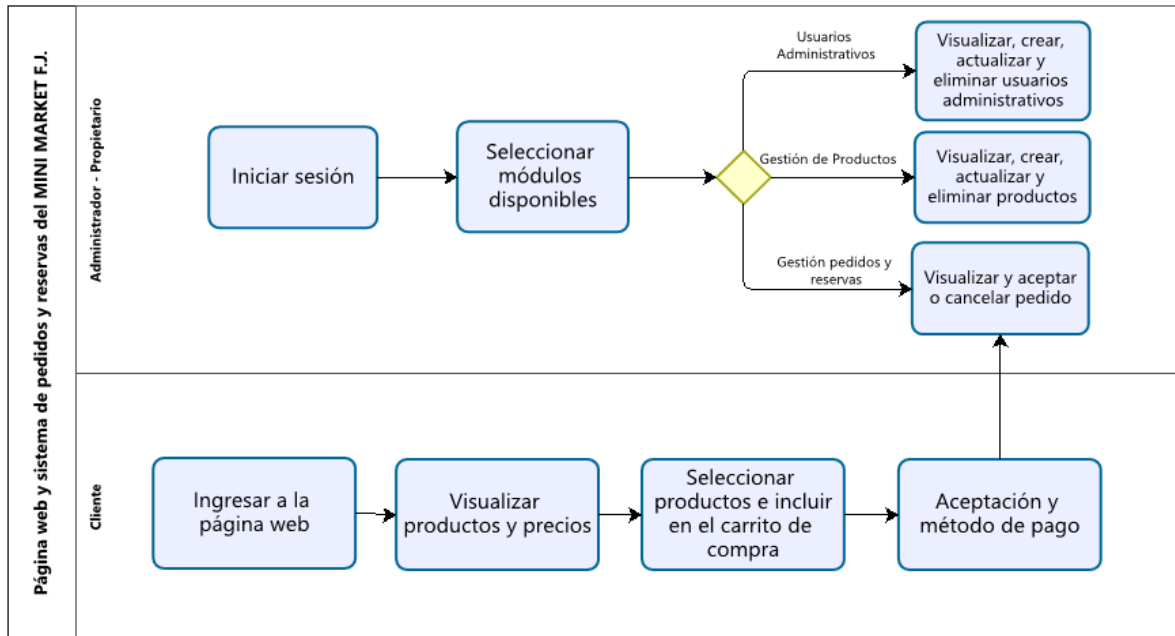
La página web posee una navegación simple y estructurada que permite a los usuarios desplazarse de manera intuitiva:

- **Inicio:** Mostrar los productos disponibles.
- **Carrito:** Visualizar los productos seleccionados por el cliente y también el método de pago.
- **Dashboard Administrativo:** Explorar los distintos módulos que posee el panel administrativo:
 - Usuarios Administrativos
 - Gestión de Productos

- Gestión de Pedidos

2.4 Lógica de negocio

Figura 2: Arquitectura del Proyecto



Como se muestra en la Figura 2, la arquitectura del proyecto muestra las actividades que realizan los dos roles del sistema.

Gestión de Módulos:

El sistema otorga al propietario del MINI MARKET F.J, la facultad de registrar los productos disponibles, incluyendo información como nombre, descripción, precio, stock y categoría. De la misma forma, podrá editar o eliminar productos según las necesidades del negocio, garantizando que el inventario se mantenga actualizado de forma permanente.

Gestión de pedidos y reservas

Los clientes podrán seleccionar productos desde la página principal y realizar pedidos o reservas a través de la página web. Cada pedido o reserva quedará registrado en el sistema con un estado inicial “En proceso”, el cual podrá ser actualizado por el

propietario conforme avance el proceso, permitiendo un control claro y ordenado de las solicitudes.

Gestión de usuarios y autenticación

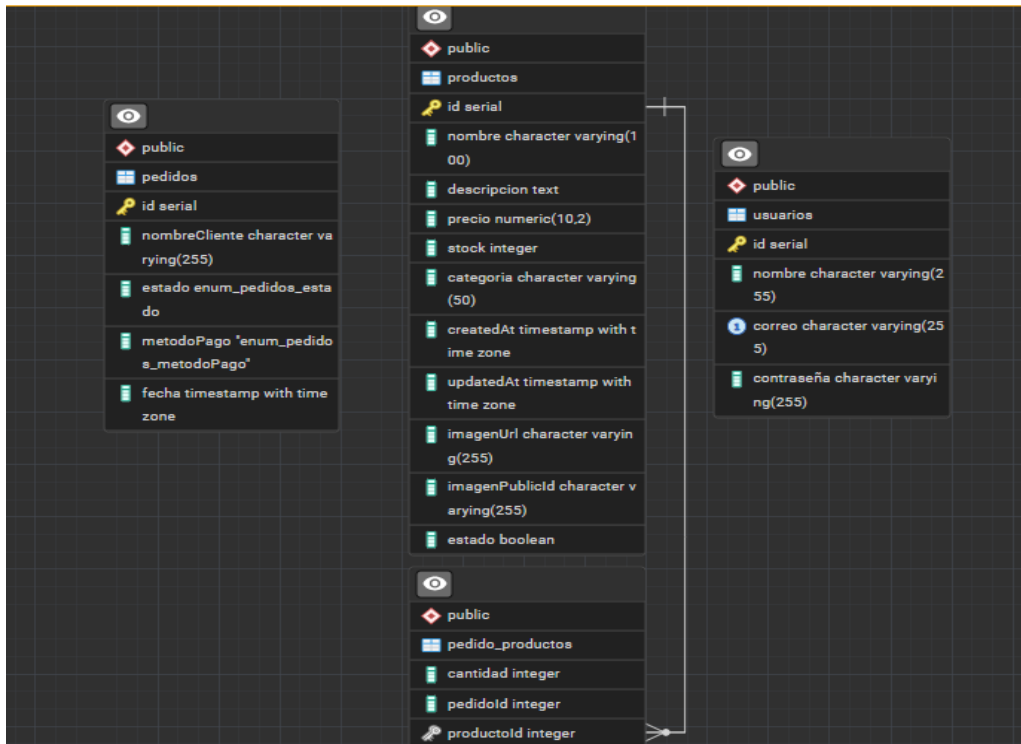
El sistema contará con un módulo de autenticación que permitirá al propietario y usuarios administrativos acceder de manera segura a las funcionalidades del sistema. El acceso estará restringido únicamente a usuarios autorizados, protegiendo la información del negocio y sus operaciones.

Selección de método de pago

Al momento de realizar la confirmación de un pedido o reserva, el cliente debe seleccionar el método de pago disponible, el cual quedará registrado en el sistema para su control y posterior verificación por parte del propietario.

2.5 Modelo de Base de Datos

Figura 3: Modelo de Base de Datos



Usuarios:

La tabla almacena la información de los usuarios administrativos del sistema, incluyendo datos como nombre, correo electrónico y contraseña.

Productos:

Guarda la información de los productos disponibles en el Mini Market, como nombre, descripción, precio, stock, categoría, estado y referencias a imágenes (Cloudinary). Esta tabla es fundamental para el control del inventario y la visualización del catálogo por parte de los clientes.

Pedidos:

Contiene los registros de pedidos y reservas realizados por los clientes, incluyendo el nombre del cliente, el estado del pedido, el método de pago seleccionado y la fecha de creación.

pedido_productos:

Es una tabla de relaciones que registra el detalle de los productos asociados a cada pedido, incluyendo la cantidad solicitada.

Para este modelo de base de datos se identificó 3 relaciones entre las tablas:

1. Relación entre pedidos y productos:

- **Tipo:** Muchos a muchos (N:M)
- **Clave foránea:** pedidoId
- **Descripción:** Un pedido puede contener varios productos, y un mismo producto puede formar parte de múltiples pedidos.

2. Relación entre pedido_productos y pedidos:

- **Tipo:** Muchos a uno (N:1)
- **Clave foránea:** pedidoId

- **Descripción:** Los registros en pedido_productos están asociado a un único pedido, mientras tanto que un pedido puede contener varios registros de productos.

3. Relación entre pedido_productos y productos

- **Tipo:** Muchos a uno (N:1)
- **Clave foránea:** productoId
- **Descripción:** Cada detalle de pedido corresponde a un solo producto, mientras que un producto puede estar asociado a múltiples pedidos.

2.6 Requisitos Funcionales

Tabla 12: Requisitos Funcionales

Código	Detalle
RF01	El sistema debe permitir la autenticación de usuarios administrativos mediante JWT.
RF02	El sistema debe permitir la navegación en la página web de manera segura y con paneles fáciles de usar
RF03	El sistema debe permitir al propietario iniciar sesión para acceder a las funcionalidades administrativas.
RF04	El sistema debe permitir en el módulo “Usuarios Administrativos” visualizar, crear, actualizar y eliminar usuarios administrativos.
RF05	El sistema debe permitir en el módulo “Gestión de Productos” visualizar, crear, actualizar y eliminar productos.
RF06	El sistema debe permitir en el módulo “Gestión de Pedidos y reservas” visualizar y aceptar o cancelar pedido.
RF07	El sistema debe permitir al cliente ingresar a la página web, mediante el uso de dispositivos móviles y de escritorio.
RF08	El sistema debe permitir al cliente visualizar el catálogo de productos disponibles con precios e imágenes.
RF09	El sistema debe permitir seleccionar productos e incluir en el carrito de compra y seleccionar el método de pago.

RF10	Contar con la Documentación técnica completa (estructura básica de la base de datos (Tablas y relaciones), así como el catálogo de endpoints de la API y el acceso directo a los repositorios de frontend y del backend alojados en la plataforma de GitHub)
RF11	El sistema debe permitir el cifrado de contraseñas con algoritmos seguros (bcrypt)

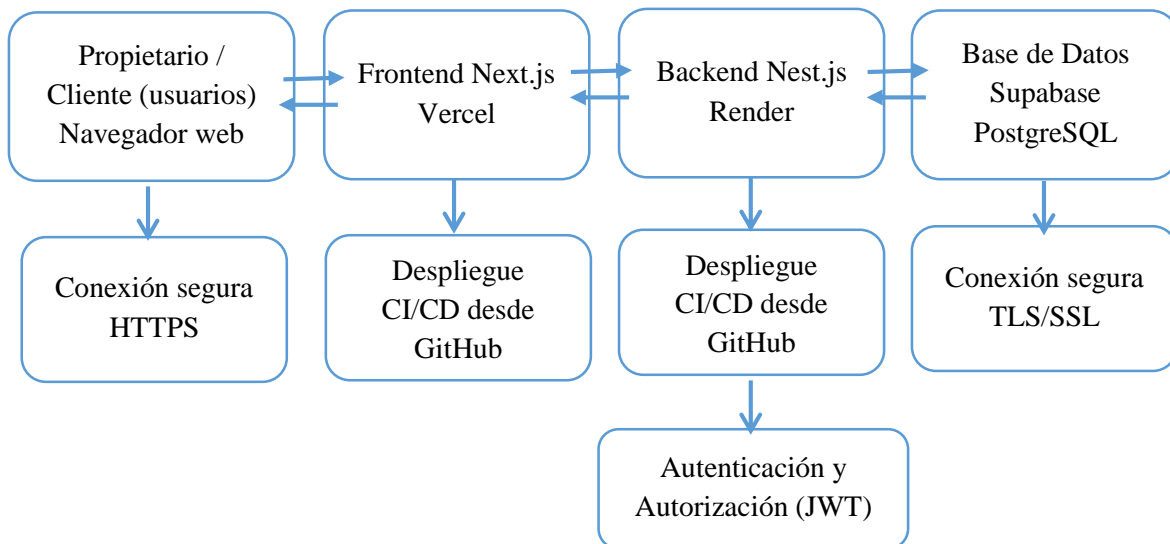
2.7 Requisitos No Funcionales

Tabla 13: Requisitos no Funcionales

Código	Detalle
RNF01	El sistema debe permitir la búsqueda de productos disponibles en la página web.
RNF02	La información debe almacenarse de forma segura y confiable en una base de datos PostgreSQL.
RNF03	La aplicación debe mantener una interfaz clara e intuitiva para los usuarios.
RNF04	El sistema debe permitir su mantenimiento y escalabilidad para futuras ampliaciones.
RNF05	El sistema debe asegurar la integridad y consistencia de los datos registrados.
RNF06	El sistema debe permitir a futuro agregar nuevas funcionalidades como que los clientes puedan contar con una membresía de pago y tener acceso a ofertas, descuentos especiales, atención prioritaria
RNF07	El sistema debe permitir el crecimiento progresivo de productos del MINI MARKET F.J.
RNF08	El sistema debe tener compatibilidad con navegadores como Google Chrome, Mozilla Firefox y Microsoft Edge.

2.8. Diseño de la Arquitectura del Sistema

Figura 4: Arquitectura del Sistema



Como se observa en la Figura N°4 la Arquitectura del Sistema se encuentra implementada por cuatro capas: capa del propietario/cliente (usuario), capa de presentación (Frontend), capa de lógica del negocio (Backend) y la capa de datos; esta interconexión se la realiza por rutas seguras que están autenticados, lo que permite la separación correcta entre las responsabilidades y la escalabilidad el sistema.

Capa del propietario/cliente (usuario): está compuesta por los usuarios, mismos que deben acceder de manera segura a través de un navegador web. Con esta capa no es necesario instalar software adicional, lo que permite el acceso desde distintos dispositivos. El navegador es un intermediario entre los usuarios y el sistema.

Esta comunicación se realiza mediante la conexión segura HTTPS, con el uso de certificados TLS/SSL, lo que permite que los datos personales viajen de forma cifrada y protegida.

Capa de presentación (Frontend): esta capa está desarrollada con la herramienta de Next.js, un framework moderno basado en React, utilizar esta herramienta permite generar páginas estáticas y manejo adecuado de rutas, lo que permite mejor la velocidad de carga de conexión. El despliegue del frontend se realizó en la plataforma Vercel, con un despliegue CI/CD con GitHub como repositorio de código fuente, así como soporte automático para HTTPS; permitiendo que la capa de presentación se encuentra siempre disponible y mantenga estándares de seguridad.

Capa de lógica del negocio (Backend): corresponde al núcleo de sistema, debido a que se ejecuta la lógica del negocio como validar solicitudes y coordinar la comunicación con la base de datos, utilizando la herramienta Nest.js que busca el uso de una arquitectura y principios de diseño visualizados hacia la mantenibilidad.

El backend se encuentra desplegado en la plataforma Render, el backend actúa como intermediario entre el frontend y la base de datos, asegurando que todas las operaciones se realicen bajo reglas de negocio definidas y mecanismos de seguridad.

Capa de Datos: la capa de base de datos se gestiona mediante supabase, donde se almacena de manera estructura datos de usuarios, productos, pedidos, pedidos_productos, cuenta con conexión segura TLS/SSL. La base de datos se creó a través del lenguaje PostgreSQL.

Bajo este contexto, la información empieza a procesar cuando el usuario interactúa con la aplicación del frontend. Captando las solicitudes que se obtengan para después enviarlas a la API, la cual se encuentra alojada en Render, lo que permite procesar las peticiones, aplicar las reglas de negocio, validar la información recibida y devuelve la información obtenida de la base de datos hacia el cliente. La base de datos realiza un proceso de autenticación y autorización, garantizando que los usuarios

administrativos tengan acceso al CRUD de usuarios administrativos, CRUD de productos y a la gestión de pedidos y reservas.

Gracias a la integración continua (CI/CD) conectado con GitHub, permite que los cambios realizados en la rama main, se documenten de forma estructurada y realiza una actualización automática tanto en Vercel (Interfaz) como en Render (API). Esta arquitectura asegura un equilibrio adecuado entre seguridad, rendimiento y mantenibilidad.

CAPÍTULO III

Pruebas y Estabilización

En la etapa de despliegue se realizó los procesos de verificación, validación y estabilización sobre la página web y el sistema de pedidos y reservas, avalando el comportamiento de las funcionalidades del sistema, además se incluyó una etapa de estabilización para poder corregir los errores presentados, optimizar el rendimiento de la página y asegurar que la página y el sistema funcionen correctamente antes de su despliegue final.

3.1 Tipos de pruebas realizadas

3.1.1 Pruebas funcionales

Se realizaron pruebas para comprobar el correcto funcionamiento de los distintos módulos y que cumplan con los requisitos definidos.

Figura 5: Página principal



Figura 6: Carrito de Compra

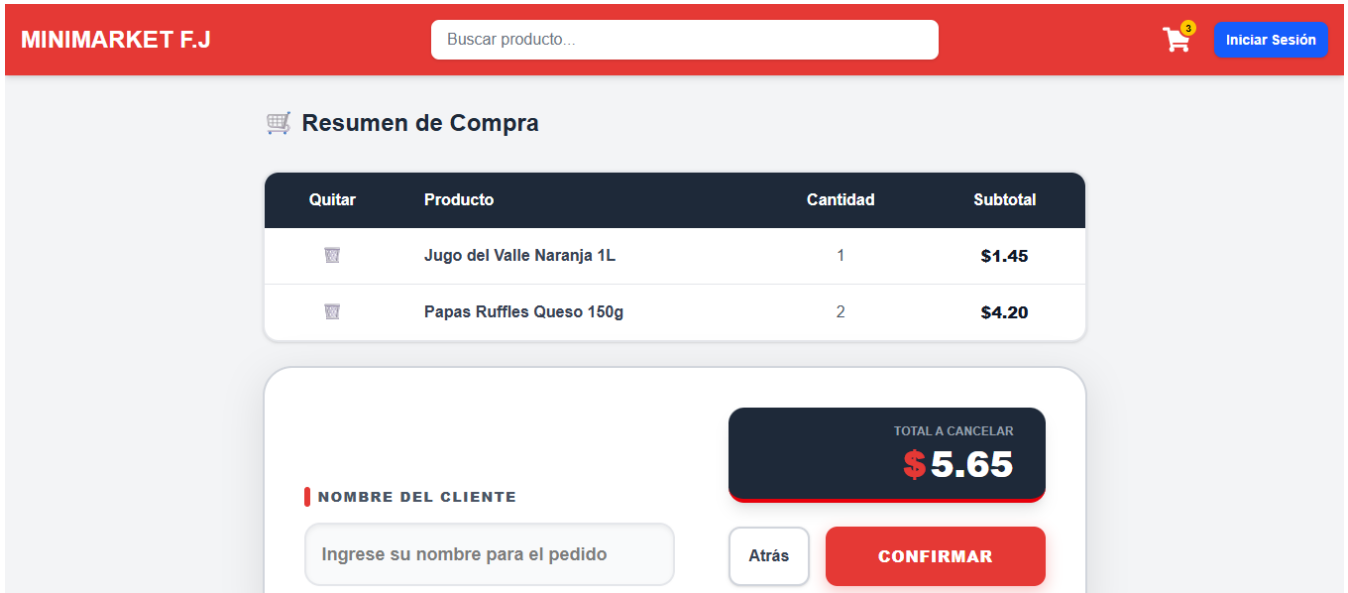


Figura 7: Método de Pago

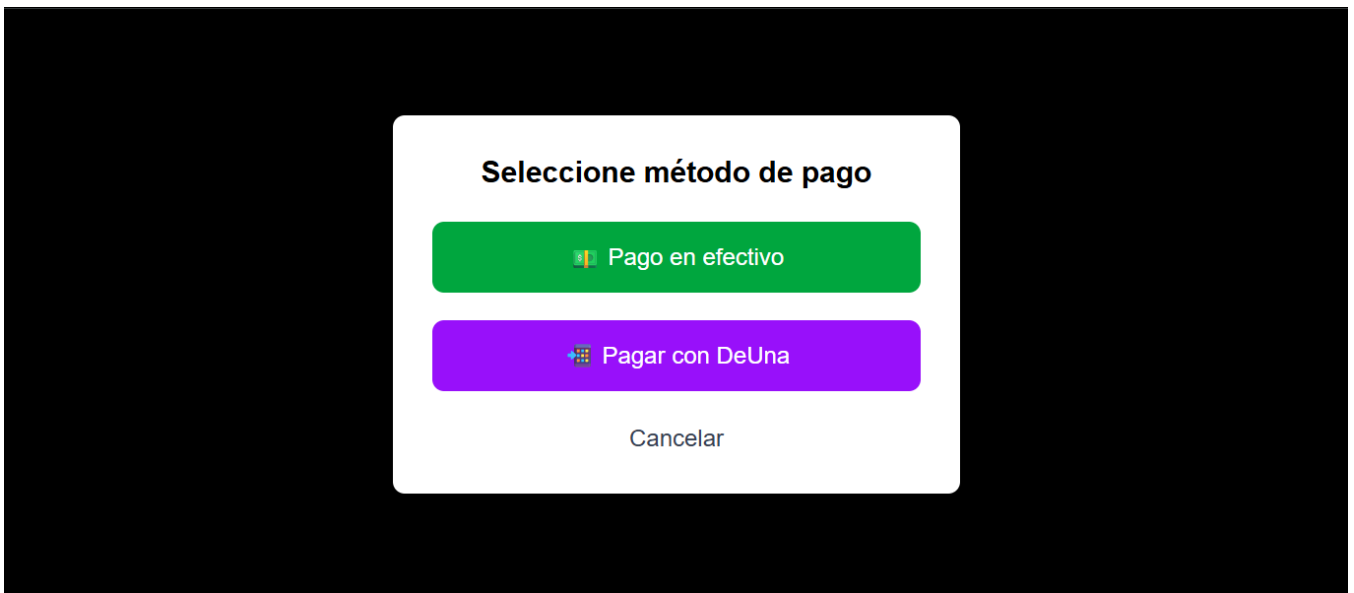


Figura 8: Inicio de Administración del Propietario

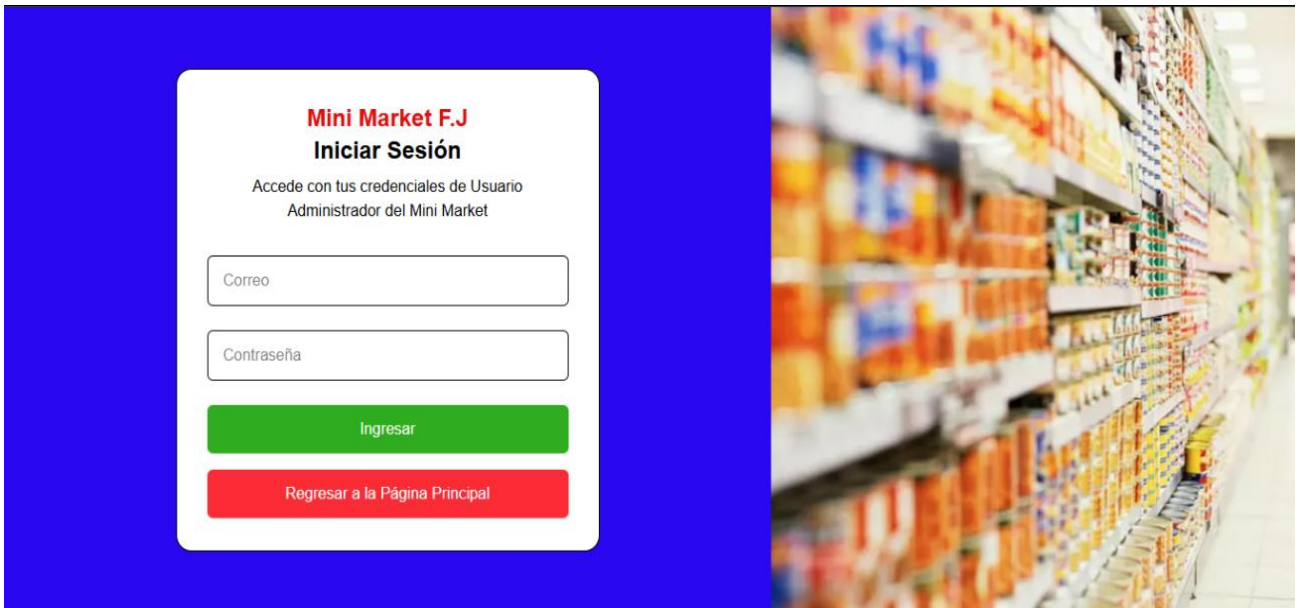


Figura 9: Panel del Administrador



Figura 10: Gestión de Usuarios Administrativos

Administración de Usuarios

+ Registrar Usuario

Fernando Vaca
fdvaca@puce.edu.ec

Editar Eliminar

Regresar al Panel

Figura 11: Gestión de Administración de Productos

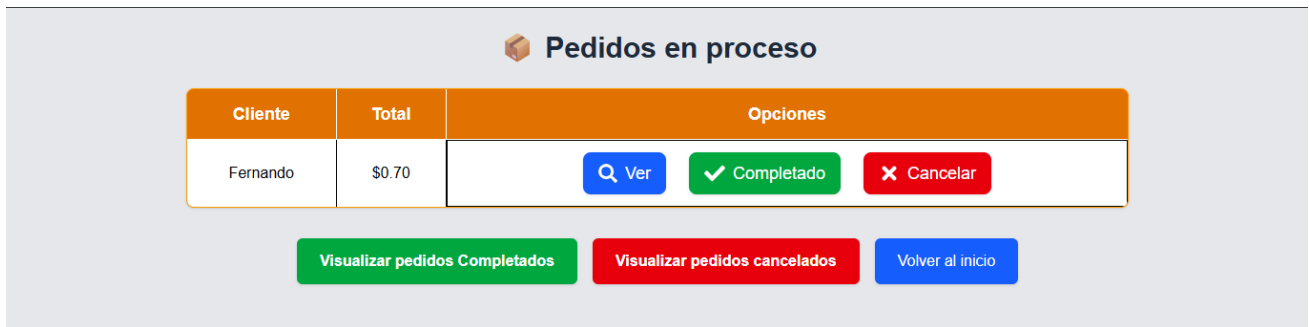
Administración de Productos

+ Nuevo Producto

Más recientes ▼ Todas las categorías ▼

ID	Nombre	Precio	Stock	Categoría	Acciones
8	Aceite La Favorita 1L	\$3.85	40	Abarrotes	Editar Eliminar
7	Caramelos Halls Menta	\$0.65	100	Confitería	Editar Eliminar
6	Papas Ruffles Queso 150g	\$2.10	45	Snacks	Editar Eliminar
5	Jugo del Valle Naranja 1L	\$1.45	60	Bebidas no alcohólica	Editar Eliminar
4	Coca-Cola 2 Litros	\$2.25	80	Bebidas no alcohólica	Editar Eliminar
3	Vino Tinto Gato Negro 750ml	\$8.50	35	Bebidas alcohólica	Editar Eliminar

Figura 12: Gestión de Pedidos y reservas

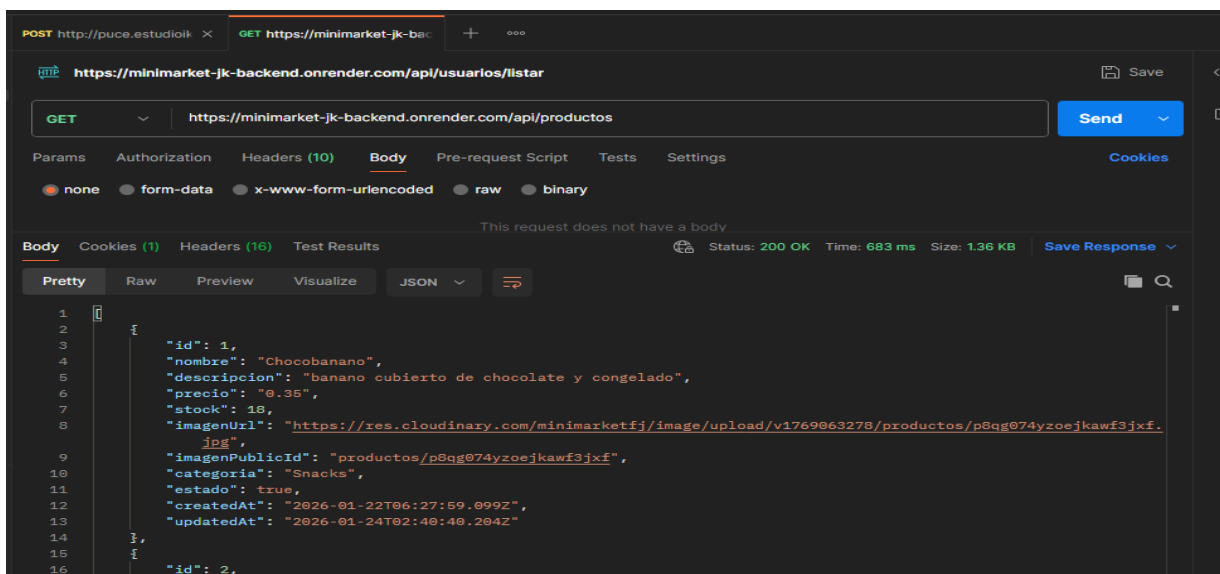


Como se muestra en las figuras presentadas, durante las pruebas iniciales de funcionalidad se encontraron errores al momento de implementar el backend. Estas incidencias fueron solventadas en etapas posteriores de prueba, logrando al final el correcto despliegue del sistema.

3.1.2 Pruebas de integración

Mediante el uso de las pruebas funcionales, se verifica la correcta comunicación entre los elementos del frontend con los datos y la lógica de negocio del backend. También se examina si la conexión realizada mediante sequelize con la base de datos esté funcionando.

Figura 13: Pruebas en POSTMAN

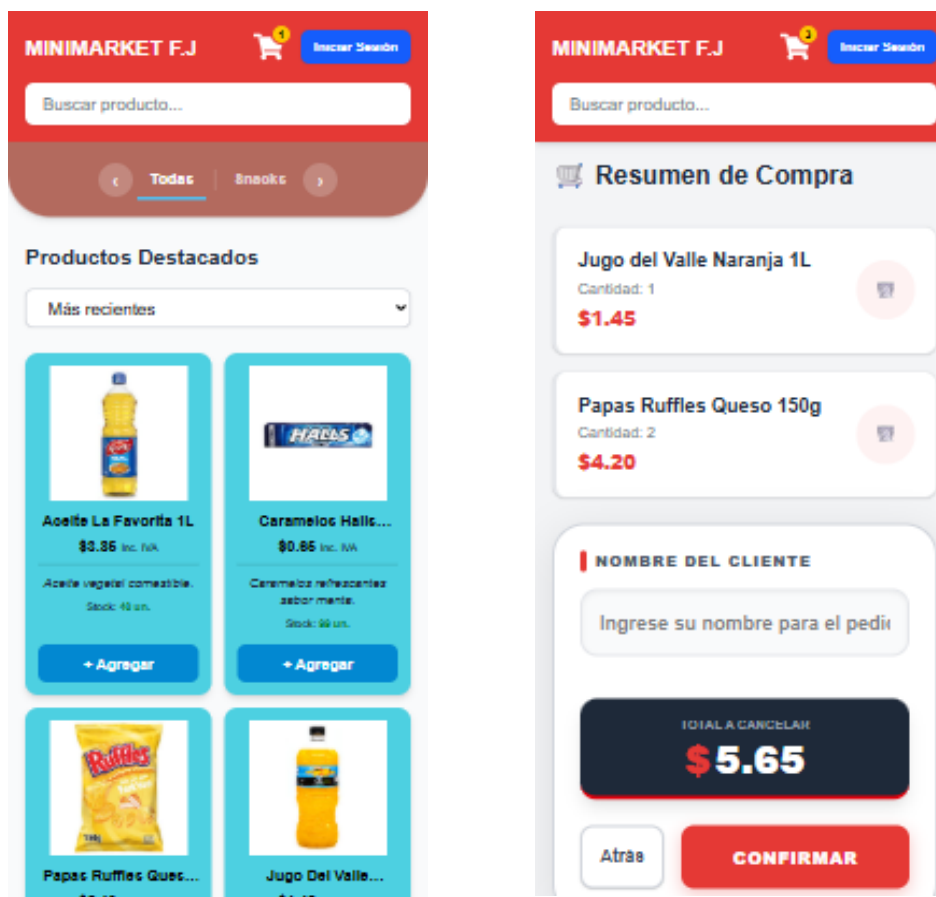


Al momento de conectar el backend con la base de datos, se identificaron problemas en cuanto al despliegue con la obtención de estos datos, mismo que fueron solventados y existió una correcta conexión entre backend y la base de datos.

3.1.3 Pruebas de interfaz (UI/UX)

Se realizaron pruebas para evaluar el diseño responsivo en dispositivos móviles y de escritorio, con el propósito de identificar posibles inconvenientes relacionados con la adaptación de la interfaz, la usabilidad y la correcta visualización de los elementos en diferentes tamaños de pantalla.

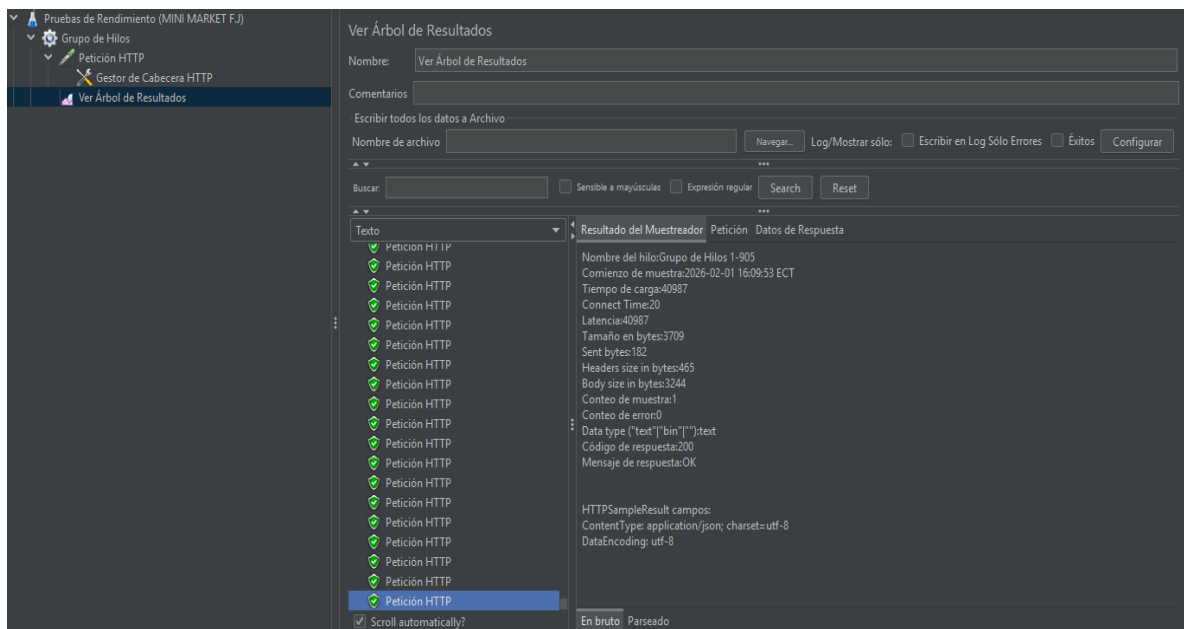
Figura 14: Pruebas de Interfaz



3.1.4 Pruebas de rendimiento

Se midió el tiempo de la respuesta del backend al momento de recibir peticiones, como se usó la plataforma de Render con un plan gratuito, el backend tiende a congelarse cuando no recibe peticiones hasta que el sistema despierte, adicional con la herramienta de Apache JMeter se realizaron pruebas de carga básica simulando múltiples clientes ingresando a la página web, de lo cual se pudo obtener que la pagina web aguanta hasta 2000 clientes conectados.

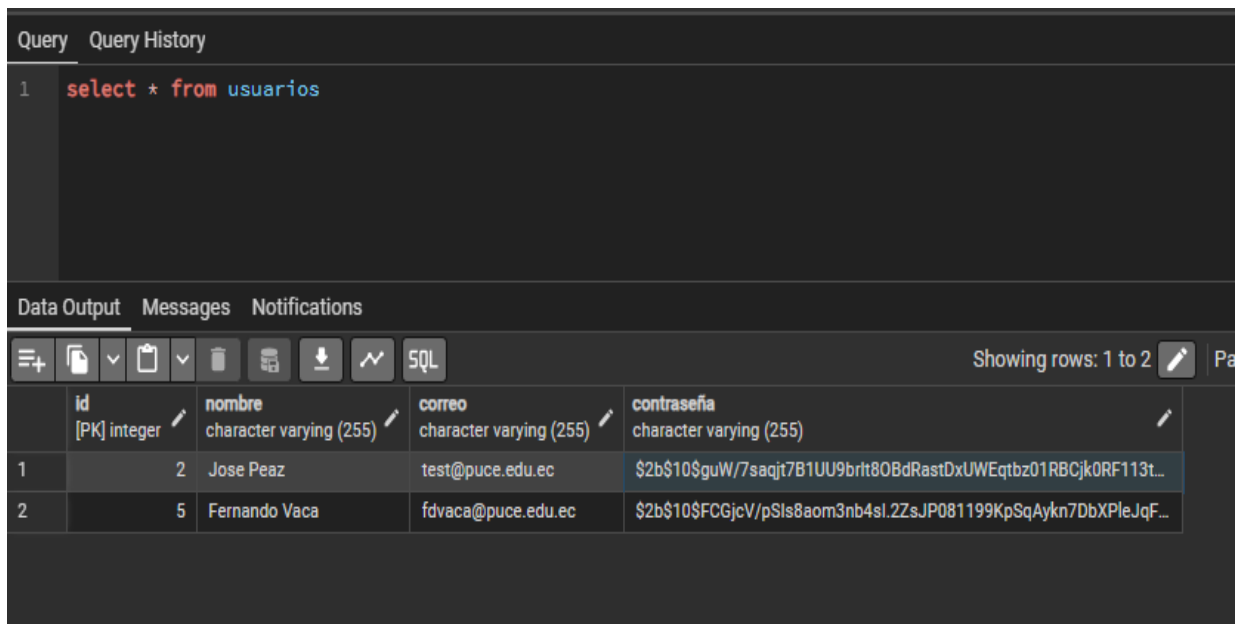
Figura 15: Pruebas en Rendimiento



3.1.5 Pruebas de seguridad

En la ejecución de esta prueba se verificó el cifrado del campo “contraseña” con la seguridad bcrypt.

Figura 16: Protección de los datos

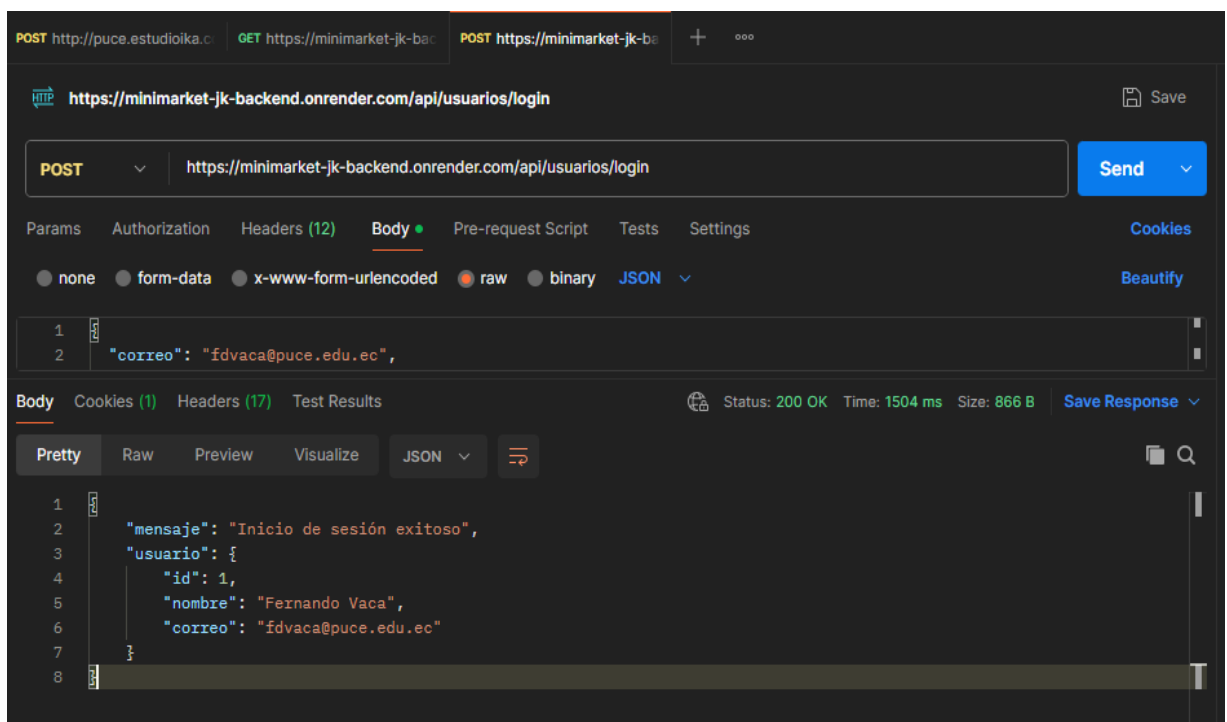


The screenshot shows a database query interface. At the top, there is a 'Query' tab with the SQL query: `select * from usuarios`. Below the query, there are tabs for 'Data Output', 'Messages', and 'Notifications'. The 'Data Output' tab is active, displaying a table with the following columns: 'id' (integer, PK), 'nombre' (character varying (255)), 'correo' (character varying (255)), and 'contraseña' (character varying (255)). The table contains two rows of data:

id	nombre	correo	contraseña
2	Jose Peaz	test@puce.edu.ec	\$2b\$10\$guW/7saqjt7B1UU9brlt80BdRastDxUWEqtbz01RBCjk0RF113t...
5	Fernando Vaca	fdvaca@puce.edu.ec	\$2b\$10\$FCGjcV/pSls8aom3nb4sl.2Zs.JP081199KpSqAykn7DbXPlE.JqF...

Además, las rutas administrativas se encuentran protegidas mediante el uso de tokens de JWT, para que solo el usuario administrativo pueda acceder a esas rutas.

Figura 17: Inicio de sesión con JWT



The screenshot shows a REST client interface. The request is a POST to `https://minimarket-jk-backend.onrender.com/api/usuarios/login`. The request body is `"correo": "fdvaca@puce.edu.ec",`. The response status is 200 OK, with a time of 1504 ms and a size of 866 B. The response body is a JSON object:

```
{  "mensaje": "Inicio de sesión exitoso",  "usuario": {    "id": 1,    "nombre": "Fernando Vaca",    "correo": "fdvaca@puce.edu.ec"  }}
```

3.1.6 Pruebas de ejecución

Durante la ejecución de esta prueba, los clientes y el propietario del MINI MARKET F.J participaron activamente en revisar la página web mediante una prueba piloto, para verificar el funcionamiento de la página web con las herramientas de despliegue que se utilizó como Vercel, Render y Supabase. Dando como resultado observaciones tanto de los clientes como del propietario.

En el caso de los clientes, manifestaron que la página web presenta una interfaz amigable y fácil de utilizar, destacando la rapidez en el proceso de visualización de productos y precios, así como la simplicidad del sistema de pedidos y reservas. Adicional, indicaron que la página web les facilita la realización de pedidos de manera más cómoda en comparación con los métodos tradicionales.

El propietario del MINI MARKET F.J, señaló que el sistema brinda una mejor organización de los pedidos, un control más eficiente del inventario y brindar una atención de calidad a sus clientes. Además, destacó que la página web optimiza los procesos operativos del negocio y representa una herramienta tecnológica.

Entre los comentarios señalaron que la página web presentaba un retraso de carga de la información de aproximadamente un minuto; sin embargo, una vez transcurrido ese tiempo, el sistema funcionaba con mayor agilidad. Esto es causado por la plataforma de Render en su versión gratuita, ya que cuenta con una desventaja la cual es que, si el sistema permanece inactivo más de 15 minutos, el sistema entra en un modo de congelamiento hasta que se vuelva a activar nuevamente mediante una nueva solicitud. Esta desventaja se puso en conocimiento del propietario y se le recomendó que puede pagar la suscripción en la plataforma de Render, por el valor de \$ 7 dólares mensuales.

3.2 Herramientas utilizadas:

- **Postman:** Para verificar el correcto uso de los endpoints y sus respectivas respuestas.
- **GitHub Projects e Issues:** Registro de las funcionalidades y su seguimiento.
- **Apache JMeter:** Para realizar las pruebas de carga básica y medición del rendimiento de la API.

3.3 Casos de prueba:

Tabla 14: Casos de Prueba

ID	Caso de Prueba	Datos de Entrada	Resultado Esperado	Resultado Obtenido
CP01	Login con JWT	Correo y contraseña	Creación del token del usuario administrativo y acceso a las rutas administrativas	Validado Exitosamente
CP02	Registro de usuario administrativo	Nombre, correo y contraseña	Registro del usuario administrativo en la base de datos	Validado Exitosamente
CP03	Actualización de usuario administrativo	Nombre, correo y contraseña	Registrar los cambios realizados al usuario administrativo en la base de datos	Validado Exitosamente
CP04	Eliminación de usuario administrativo	N/A	Eliminación del usuario administrativo en la base de datos	Validado Exitosamente
CP05	Registro de Producto	Nombre, descripción, precio, stock, categoría y la imagenUrl	Registro del producto en la base de datos	Validado Exitosamente

CP06	Actualización de Producto	Nombre, descripción, precio, stock, categoría y la imagenUrl	Registrar los cambios realizados al producto en la base de datos	Validado Exitosamente
CP07	Eliminación de Producto	N/A	Eliminación del producto en la base de datos	Validado Exitosamente
CP08	Registro de Pedidos o reservas	Nombre del cliente, estado del pedido o reserva, productos seleccionados y método de pago	Registro del pedido o reserva en la base de datos	Validado Exitosamente
CP07	Actualización de Pedidos o reservas	Estado del pedido o reserva	Registrar los cambios realizados al estado del pedido o reserva en la base de datos	Validado Exitosamente

3.4 Estabilización del sistema

Antes de realizar el despliegue final se hizo la corrección de los errores y las implementaciones y depuraciones que fueron detectados en las pruebas realizadas:

- Se corrigió el error de la página principal que no mostraba los precios de los productos.
- Se corrigió el error del carrito de compra que la tabla creada no eliminaba de forma correcta los productos.
- Se corrigió el error de la gestión de usuarios administrativos que no permitía la actualización del campo “contraseña”.
- Se realizó una depuración en el código del backend, para la eliminar de funciones que ya no se usaban.

- Se implementó la biblioteca de Tailwind CSS, para una interfaz limpia en dispositivos móviles y de escritorio.
- Se corrigió el error del token que no expiraba en el tiempo indicado.
- Se corrigió el error de la implementación de Render por falta de datos esenciales.
- Se corrigió el error en el backend de la conexión con la base de datos por tratar de conectarse por un método que no funcionaba.
- Se corrigió el error del backend (Render) con la conexión del frontend (Vercel) por falta de la url del frontend para realizar la conexión.
- Se depuró la base de datos para eliminar un campo innecesario de la tabla “productos”.

Como resultado de la corrección de los errores identificados, el sistema obtuvo una mejora significativa tanto en su tiempo de carga, como en su interacción con los clientes. Garantizando un despliegue estable y funcional del sistema y ejecutando requisitos técnicos y requisitos de calidad establecidos para la solución desarrollada.

CONCLUSIONES

- La implementación de un diseño responsivo mediante Tailwind CSS, garantizó que la plataforma sea funcional en dispositivos móviles y de escritorio, garantizando adaptabilidad y experiencia a los clientes para visualizar los productos y realizar pedidos y reservas cómodamente.
- El desarrollo del software bajo el stack de JavaScript (Next.js/Node.js) proporciona al establecimiento una herramienta propia que no depende de terceros o de aplicaciones externas, maximizando así el margen de beneficio por cada pedido y reserva, generando independencia tecnológica.
- La arquitectura del sistema presentada otorga una base sólida para el desarrollo de la página web y el sistema de pedidos y reservas del MINI MARKET F.J., convirtiéndose en un sistema moderno, seguro y con visión a realizar nuevas ampliaciones en el sistema, para beneficio del propietario y sus clientes, fortaleciendo la presencia en el mercado local del establecimiento.
- Con la integración de un Backend alojado en Render y el uso de React Context para el carrito de compras, se aseguró una persistencia de datos fluida, eliminando errores comunes en la selección y sumatoria de productos, generando eficiencia en el procesamiento de datos. Además de contar con seguridad y transparencia en los métodos de pagos modernos (como códigos QR de Deuna y efectivo).

RECOMENDACIONES

- Se recomienda mantener de manera actualizada la información de los productos que ofrece el MINI MARKET F.J, utilizadas en el frontend y backend, asegurando compatibilidad con nuevas versiones. Se sugiere migrar a un plan de hosting de mayor capacidad en el futuro para asegurar la estabilidad del servidor ante un incremento masivo de clientes.
- Es fundamental establecer un cronograma de revisiones técnicas para actualizar las librerías de Next.js y Tailwind, garantizando así la seguridad del código frente a posibles vulneraciones, con ello se pretende contar con un mantenimiento programado.

REFERENCIAS BIBLIOGRÁFICAS

- Author, G. (12 de abril de 2020). *Bootstrap: guía para principiantes de qué es, porque y cómo usarlo*. Obtenido de rockcontent: <https://rockcontent.com/es/blog/bootstrap/>
- Brais, M. (2023). *Git y GitHub desde cero*. España: Brais Moure.
- Carraro, J. M. (2015). *Diseño de Experiencia de Usuario*. Buenos Aires: Editorial Autores de Argentina.
- Gamarra, F. (2023). Visual Studio Code - Vol.1. En F. Gamarra, *Visual Studio Code - Vol.1* (pág. 6). RedUSERS.
- Itequia. (26 de enero de 2023). *Itequia*. Obtenido de Descubre Nest.js, el Framework de desarrollo de Backend para Frontend: <https://itequia.com/es/descubre-nestjs/>
- Martínez, J. (2025). *Fundamento de Diseño web, una guía para empezar desde cero*. México: Amazon Digital Services LLC - kdp.
- Maza, M. Á. (2012). Javascript. En M. Á. Maza, *Javascript* (pág. 9). Innovación y Cualificación, S.L.
- Molina, J. R. (2017). Administración de base de datos con PostgreSQL. En R. F. Molina Jimmy, *Administración de base de datos con PostgreSQL* (pág. 12). área de innovación y desarrollo, S.L.
- Moure, B. (2024). Git y Github desde cero. En B. Moure, *Git y Github desde cero* (pág. 19). Brais Moure.
- Render. (2026). *Render*. Obtenido de <https://render.com/>

Samuel, D. (2018). *Git Controle la version de sus versiones (conceptos, utilización y casos practicos)*. Ediciones ENI.

Sanabria, M. (2021). *Transformación digital en las organizaciones*. Bogotá: Editores Académicos.

Supabase. (2026). *Supabase*. Obtenido de <https://supabase.com/>

Vercel. (2026). *Vercel Inc*. Obtenido de <https://vercel.com/>

ANEXOS

Anexo 1: Encuestas al Propietario



PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR
TECNOLOGÍA SUPERIOR DESARROLLO DE SOFTWARE
PROYECTO: "PAGINA WEB Y SISTEMA DE PEDIDOS Y RESERVAS DE PRODUCTOS DEL MINI MARKET F.J"

ENCUESTA A PROPIETARIO:

Nombre Completo: _____

Correo electrónico: _____

Número celular: _____

1. ¿Cuánto tiempo lleva funcionando el MINI MARKET F.J.?

Menos de 1 año

Entre 1 y 3 años

Entre 3 y 5 años

Más de 5 años

2. ¿Qué tipo de productos ofrece en su MINI MARKET F.J.?

3. ¿Actualmente, cómo realiza la atención a sus clientes?

De forma presencial

Llamadas telefónicas

WhatsApp

Otro (especifique) _____

4. ¿Actualmente, cómo realiza el control de inventario de productos?

De manera manual

Utiliza Excel (computadora)

Utiliza un sistema informático

No lleva control de inventario

5. ¿Usted, ha tenido dificultad por la falta de control en su inventario?

SI

NO

6. ¿Actualmente, utiliza algún sistema informático para pedidos y reservas?

SI

NO

7. Usted está de acuerdo en la creación de una página web y sistema de pedidos y reservas, para su MINI MARKET F.J.?

SI

NO

8. ¿Indique cuales serían los beneficios que tendría su MINI MARKET F.J al contar con una página web y sistema de pedidos y reservas?

Mejor atención al cliente

Incremento en ventas

Ahorro de tiempo

Mejor control de inventario

Otro (especifique) _____

9. ¿Cuenta con acceso a internet de forma permanente en su MINI MARKET F.J.?

SI

NO

Firma del propietario

Anexo 2: Encuesta al Cliente



**PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR
TECNOLOGÍA SUPERIOR DESARROLLO DE SOFTWARE
PROYECTO: "PAGINA WEB Y SISTEMA DE PEDIDOS Y RESERVAS DE PRODUCTOS DEL MINI
MARKET F.J"**

ENCUESTA A CLIENTES:

Nombre Completo: _____

1. ¿Indique la frecuencia que Usted realiza compras en el MINI MARKET F.J?

- Diaria
- Tres veces por semana
- Una vez por semana
- Casualmente

2. ¿Califique la atención recibida en el MINI MARKET F.J.?

- Excelente
- Buena
- Regular
- Mala

3. ¿Usted, utiliza páginas web o aplicaciones móviles para realizar pedidos y reservas?

- SI
- NO

4. Si su respuesta es afirmativa. Indique con qué frecuencia lo realiza

- Frecuentemente
- Ocasionalmente
- Nunca

5. ¿Le gustaría que el MINI MARKET F.J. cuente con una página web y un sistema de pedidos y reservas de los productos que ofrece?

- SI
- NO

6. ¿Indique que beneficios considera relevantes al contar con una página web y un sistema de pedidos y reservas?

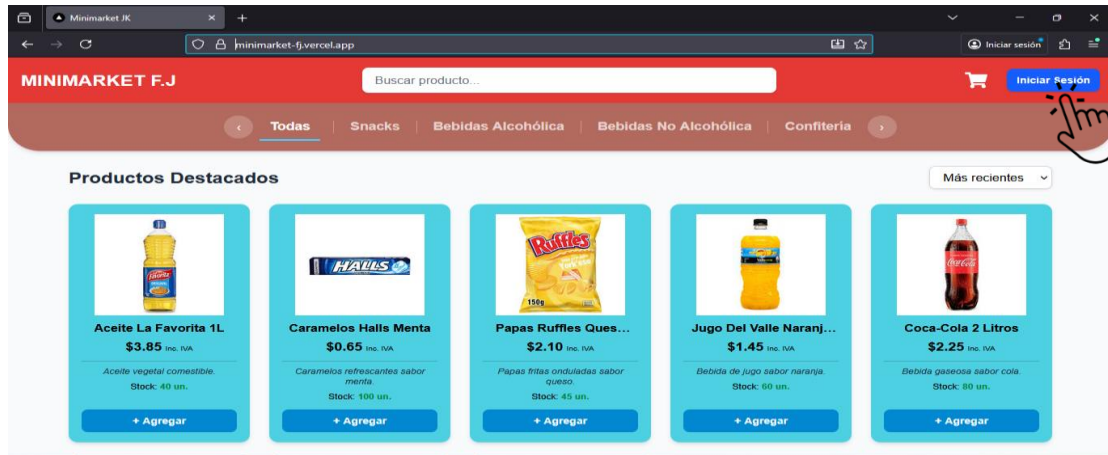
- Ahorro de tiempo
- Disponibilidad de productos
- Comodidad
- Otro (especifique) _____

Anexo 3: Manual de Usuario

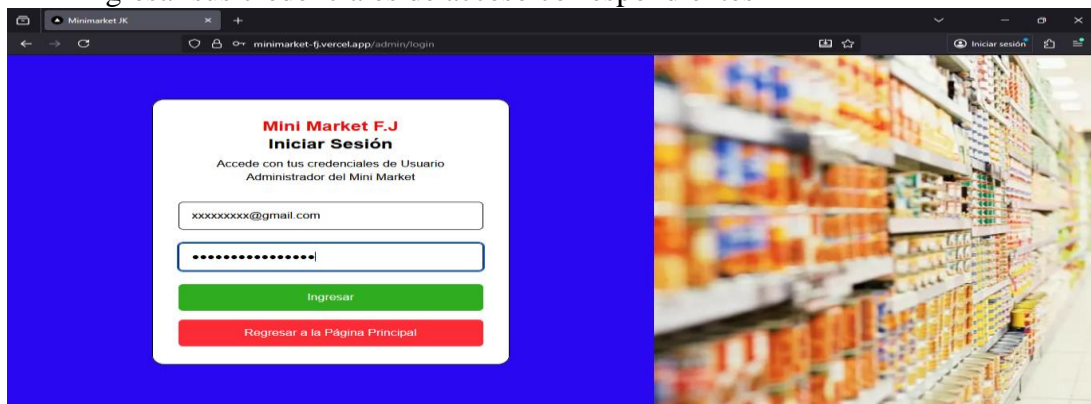
MANUAL DE USUARIO PÁGINA WEB Y SISTEMA DE PEDIDOS Y RESERVAS DEL MINI MARKET F.J

PARA EL PROPIETARIO:

1. Ingresar al sistema en el siguiente link: <https://minimarket-fj.vercel.app/> y proceso a iniciar sesión.



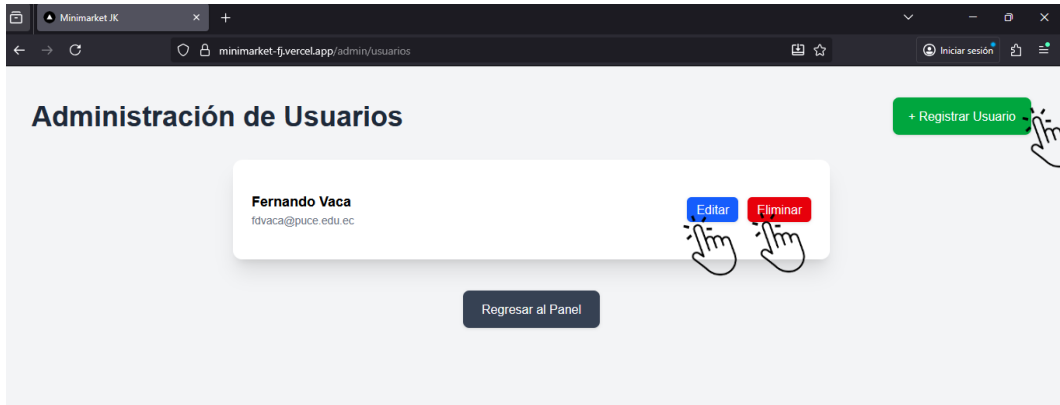
2. Se mostrará la pantalla de inicio de sesión, en la cual el propietario deberá ingresar sus credenciales de acceso correspondientes



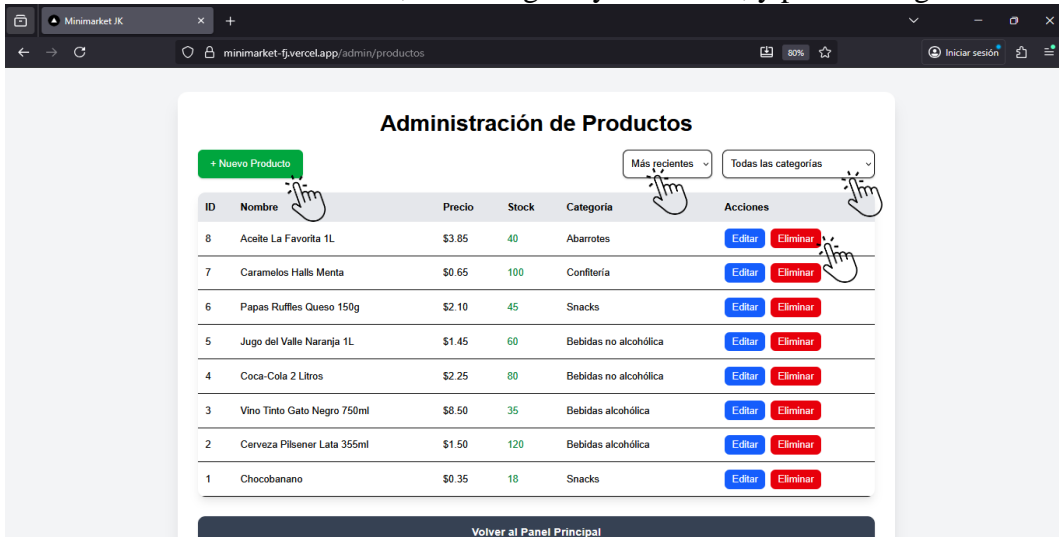
3. Una vez iniciada la sesión, se visualizarán los módulos del sistema administrativo. Debe seleccionar el módulo al cual desea ingresar.



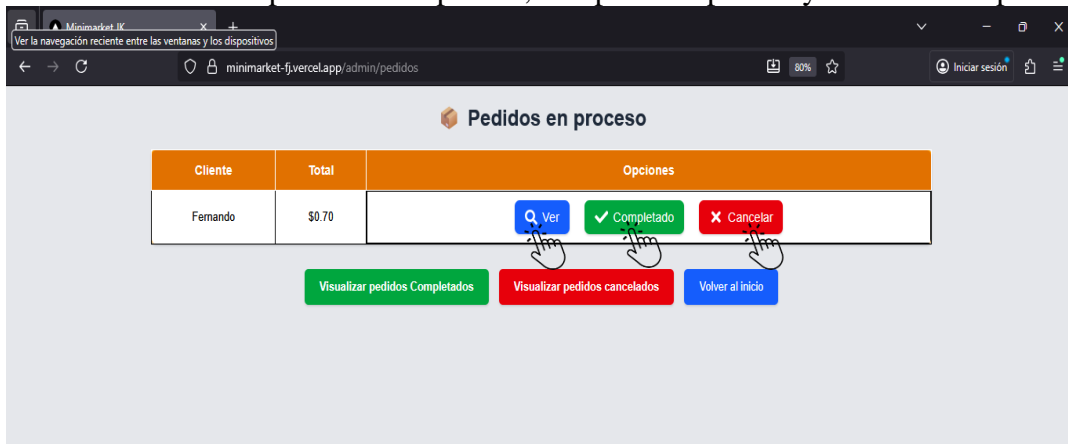
4. Al ingresar al módulo “Administración de Usuarios”, el propietario podrá seleccionar la opción: registrar, editar o eliminar usuarios administrativos, según sea necesario.



5. Al ingresar al módulo “Administración Productos”, el propietario podrá seleccionar la opción de: nuevo producto, editar y/o eliminar producto. Puede filtrar entre: más recientes, más antiguos y stock <10, y por la categoría.

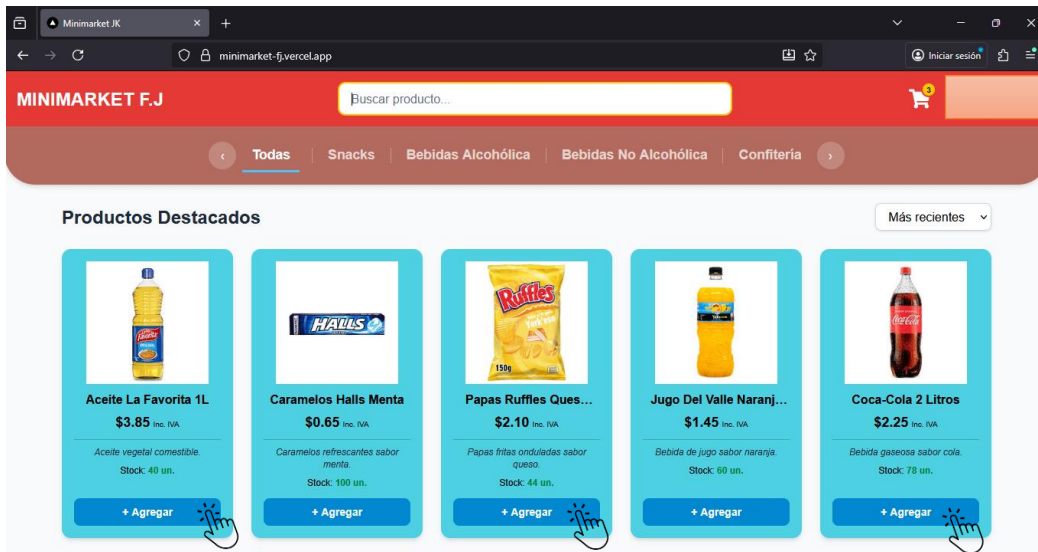


6. Al ingresar al módulo “Administración de Pedidos”, el propietario podrá seleccionar la opción: ver el pedido, completar el pedido y/o cancelar el pedido.

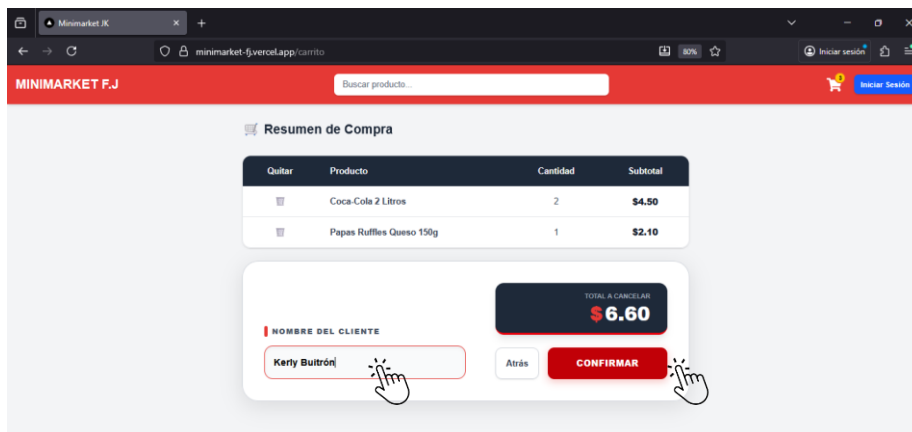


PARA EL CLIENTE:

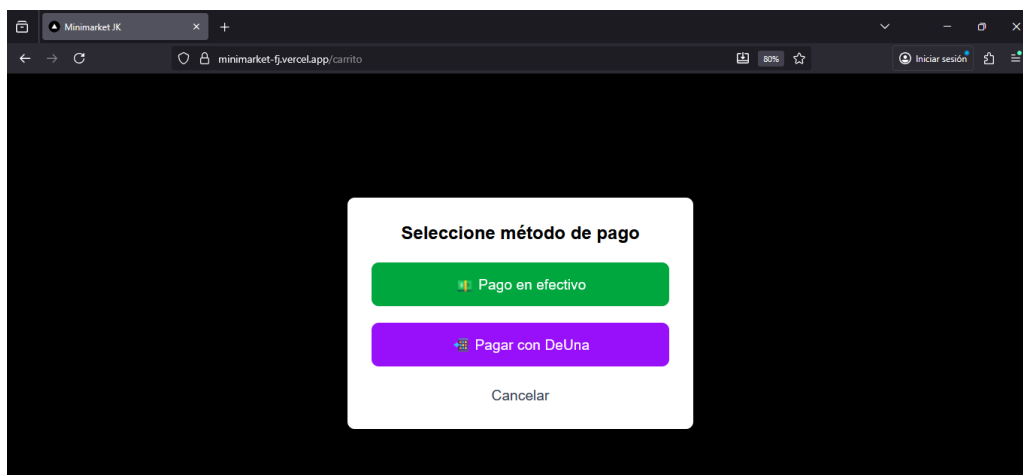
1. Ingresar al sistema en el siguiente link: <https://minimarket-fj.vercel.app/>, buscar el producto que se desea adquirir y seleccionarlo para el carrito de compra.



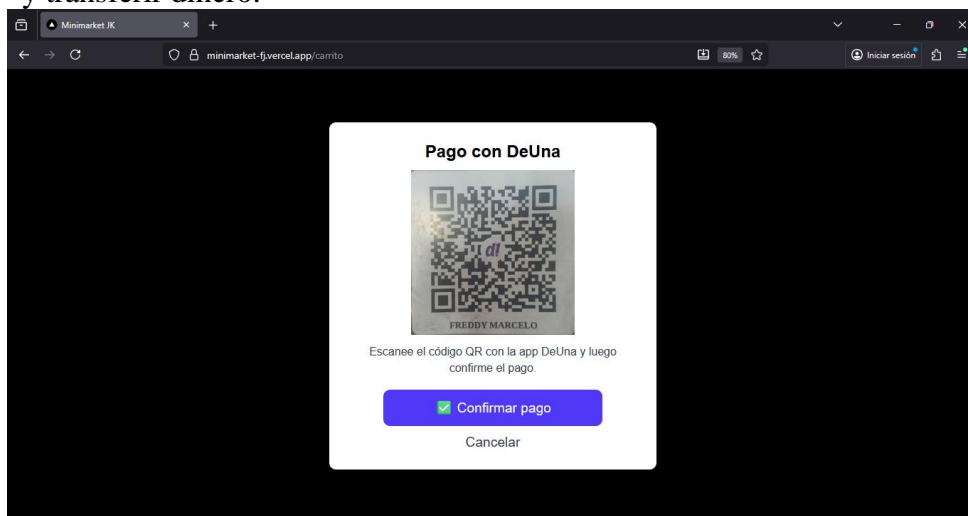
2. El cliente debe llenar la información solicitada y confirmar la compra



3. El cliente una vez confirmado la compra, procede a seleccionar el método de pago y se acerca al establecimiento a retirar su pedido.



4. Si el método de pago es “Pagar con De Una” le aparecerá el código QR, escanear y transferir dinero.



Anexo 4: Entrada del MINI MARKET F.J



Anexo 5: Encuesta al Propietario



Anexo 6: Capacitación al Propietario

