



**PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR**

**Unidad Académica de Formación Técnica y Tecnológica – PUCE TEC**

**SISTEMA DE GESTIÓN DE CLIENTES Y PROMOCIONES EN  
BAR/DISCOTECA BOUNTY**

**Proyecto de titulación previo a la obtención del título de Tecnólogo Superior En Desarrollo  
de Software**

**Autor:**

Nelson Allauca

**Tutor:**

Ing. Chiliquinga Tutachá Andrés Mauricio

**Quito, Ecuador 2026**

## **Dedicatoria**

Dedico este trabajo a mis padres y personas que me apoyaron a esforzarme cada día para lograr completar la meta de poder llegar a la graduación teniendo en cuenta los obstáculos junto a los retos que se presentaron cada día.

## Índice

Introducción .....	1
Antecedentes .....	1
Planteamiento del Problema.....	2
Justificación.....	2
Objetivo General .....	3
Objetivos Específicos.....	3
Metodología Desarrollo de Software Scrum.....	4
Capítulo I .....	5
1.1 Marco Teórico .....	5
1.2 Procesos Actuales.....	6
1.3 Infraestructura Actual.....	8
1.4 Marco de Trabajo Scrum.....	9
1.5 Sprints.....	10
Capitulo II .....	13
2.1 Análisis y Diseño .....	13
2.2 Mapa de Navegación.....	13
2.3 Diseño de arquitectura.....	15
2.4 Viabilidad Técnica .....	17
2.5 Viabilidad Económica.....	18
2.6 Casos de Uso .....	21
Capitulo III.....	37
3.1 Desarrollo.....	37
3.2 Requerimientos del software.....	37
3.3 Cronograma.....	39
3.4 Diagrama de clases.....	44
3.5 Diagrama de secuencia.....	48
3.6 Código fuente .....	53
3.7 Modelo Base de datos .....	61
Capitulo IV.....	71
4.1 Pruebas y resultado.....	71
4.2 Tablas de iteración de cada prueba con sus versiones .....	76

Conclusiones .....	78
Recomendaciones.....	79
Bibliografía .....	80

## Lista de Tablas

Tabla 1.....	8
Tabla 2.....	10
Tabla 3.....	11
Tabla 4.....	17
Tabla 5.....	18
Tabla 6.....	19
Tabla 7.....	20
Tabla 8.....	21
Tabla 9.....	23
Tabla 10.....	25
Tabla 11.....	27
Tabla 12.....	29
Tabla 13.....	31
Tabla 14.....	33
Tabla 15.....	35
Tabla 16.....	40
Tabla 17.....	41
Tabla 18.....	42
Tabla 19.....	43
Tabla 20.....	63
Tabla 21.....	64
Tabla 22.....	64

Tabla 23.....	65
Tabla 24.....	65
Tabla 25.....	66
Tabla 26.....	66
Tabla 27.....	67
Tabla 28.....	68
Tabla 29.....	69
Tabla 30.....	69
Tabla 31.....	76

## Lista de Figuras

Figura 1	7
Figura 2	14
Figura 3	16
Figura 4	21
Figura 5	23
Figura 6	25
Figura 7	27
Figura 8	29
Figura 9	31
Figura 10	33
Figura 11	35
Figura 12	38
Figura 13	39
Figura 14	45
Figura 15	46
Figura 16	47
Figura 17	48
Figura 18	49
Figura 19	50
Figura 20	51
Figura 21	52
Figura 22	53

Figura 25	56
Figura 26	56
Figura 27	57
Figura 28	57
Figura 29	58
Figura 30	58
Figura 31	59
Figura 32	60
Figura 33	61
Figura 34	62
Figura 35	71
Figura 36	72
Figura 37	72
Figura 38	73
Figura 39	73
Figura 40	74
Figura 41	75
Figura 42	75
Figura 43	76

## **Declaración Y Autorización**

Yo, NELSON GABRIEL ALLAUCA GUILCAPI con C.I. 0605419522 autor(a) del trabajo de titulación intitulado: “Sistema De Gestión De Clientes Y Promociones En Bar/Discoteca Bounty”, 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, 14 de febrero 2026



Nelson Gabriel Allauca Guilcapi

C.I. 0605419522

## **Agradecimientos**

Agradezco a la vida, valores y conocimientos impartidos por cada ingeniero dentro de la carrera técnica que fueron el detonante de mi mejora continua, siendo su esfuerzo diario lo que me hizo mejorar mentalmente, entender la realidad de trabajar en la vida, aplicar los conocimientos prácticos y teóricos en mi vida laboral y entender que en la vida puedo seguir avanzando para superar nuevos límites.

## **Introducción**

Dentro del mercado local de bares/discotecas en Quito, uno de los desafíos más comunes es la falta de una estrategia eficaz para mantener y atraer nuevos clientes. En este caso, el Bar Bounty depende de la captación de nuevos clientes con uso de personal y promoción informal, poco adaptados a mecanismos estructurados para promover la lealtad y aumentar la frecuencia de visitas de los clientes recurrentes.

Añadido que se excluye una gestión adecuada de la relación con los clientes, perdiendo oportunidades de maximizar su potencial de ingresos, sin un sistema definido que contemple una forma clara de registrar la información de los clientes, promover los servicios actuales del Bar o tener un Dashboard para el área administrativa que permita gestionar los procesos relacionados al negocio.

Por tanto, este sistema web de gestión de clientes y promociones será construido para implementar herramientas al área administrativa de forma viable, escalable y estratégica para resolver los desafíos actuales del negocio. Al transformar datos en acciones concretas y automatizar la relación con el cliente

## **Antecedentes**

Al plantearse enfoques tecnológicos para mejorar los servicios de locales comerciales y promover la visibilidad de la marca, en muchos establecimientos conocidos para conectar con el cliente se ha generado webs interactivas para la gestión de sus procesos y de clientes.

A este punto se suma que la digitalización es un sector de inversión actualmente valorado dentro de Ecuador pues demuestra viabilidad y la falta de esa inversión en el local Bounty ha hecho que objetivamente su operabilidad sufra costos operativos ante realización de procesos manuales, poco fiables y basados en errores.

Por consiguiente, la evidencia demuestra que herramientas tecnológicas modernas responden mejor a las necesidades reales del sector de entretenimiento, pues históricamente ha permanecido al margen de la transformación digital y la situación de la cual el negocio se basa no es la excepción.

### **Planteamiento del Problema**

Actualmente, el Bar Bounty carece de un sistema centralizado para almacenar y consultar los datos de los clientes. Esto se traduce en una poca segmentación de los clientes, lo que limita la capacidad de lanzar promociones personalizadas que respondan a los intereses específicos de cada grupo de clientes.

A la vez, la poca relación con sus clientes de manera eficiente no permite premiar su lealtad, sobre todo, crear experiencias personalizadas que fortalezcan el vínculo con el cliente, incentivando su regreso y mejorando la rentabilidad del negocio. En consecuencia, no existe un sistema que permita ver datos para la gerencia centrados en sus clientes, para tomar decisiones de maximización de impacto en la marca como lugar de diversión nocturna.

### **Justificación**

El desarrollo del Sistema de Gestión de Clientes y Promociones se fundamenta en dar un paso junto a la tecnología mediante la construcción de software que motive busque el captar el interés de los clientes, en contraposición al enfoque transaccional tradicional que prioriza ventas únicas sin considerar el valor del cliente a lo largo del tiempo.

El sistema propuesto materializa el incentivo a la recurrencia de visitas y fortalece el vínculo emocional entre el cliente y el establecimiento, desde una perspectiva tecnológica. Según la teoría de Sistemas de Información, Laudon y Laudon (2020) definen: “los sistemas de información como conjuntos de componentes interrelacionados que recolectan, procesan,

almacenan y distribuyen información para apoyar la toma de decisiones, la coordinación y el control organizacional” (p.12).

El Sistema de Gestión de Clientes y Promociones implementa esta definición al capturar datos de comportamiento del consumidor, procesarlos mediante promociones y valorar su participación al consumir, almacenarlos en bases de datos relacionales y distribuirlos a través de dashboards analíticos que fundamentan decisiones estratégicas del negocio

### **Objetivo General**

Crear un sistema web que convierta el comportamiento e información de los clientes del bar en datos estratégicos, con la implementación de puntos, beneficios o descuentos a los clientes nuevos o frecuentes dentro de las funcionalidades del sistema, para aumentar el flujo constante de visitantes y generar decisiones comerciales basadas en datos reales mejorando la marca Bounty dentro de la zona comercial Foch.

### **Objetivos Específicos**

- Implementar un módulo de notificaciones personalizadas que informe a los clientes sobre nuevas promociones, eventos y beneficios exclusivos según sus preferencias y comportamiento de consumo.
- Construir una experiencia de usuario que permita a los clientes consultar sus puntos, acceder a promociones y conocer novedades, así como a los administradores gestionar los datos y campañas fácilmente.
- Desarrollar código modular utilizando patrones de diseño y buenas prácticas responsive que se adapte a diferentes dispositivos, junto a la implementación de frameworks y Apis, de la página web que actualmente tienen en producción.

## **Metodología Desarrollo de Software Scrum**

La selección de Scrum como metodología de desarrollo se justifica por su capacidad de gestionar la incertidumbre inherente a proyectos de software mediante entregas incrementales que permiten validación temprana con el cliente.

Los Sprints de dos semanas garantizan que el sistema para el negocio del presente proyecto reciba funcionalidades operativas en ciclos cortos, reduciendo el riesgo de desarrollar características que no satisfagan las necesidades reales del negocio.

La transparencia del proceso mediante ceremonias definidas (Planning, Daily, Review, Retrospective) facilita la comunicación entre el equipo de desarrollo y los stakeholders, mientras que la priorización del Product Backlog asegura que los esfuerzos se concentren en las funcionalidades de mayor valor para el establecimiento.

## Capítulo I

### 1.1 Marco Teórico

El desarrollo de sistemas digitales centrados al cliente mediante promociones representa una evolución significativa en la gestión de establecimientos de entretenimiento nocturno. Este marco teórico integra cinco áreas fundamentales que sustentan técnica y conceptualmente el proyecto propuesto afirmando que el sistema web abarca componentes clave como la interfaz de usuario (UI), el desarrollo Backend, la gestión de bases de datos, el uso de frameworks y la comunicación mediante APIs (Das, 2024).

Siendo así, que la construcción de la interfaz de usuario se buscó modelos adaptados a Mobile First en repositorios de búsqueda UX/UI siendo Dribbble la plataforma para destacar diseños inspirados, debido a que, dentro del público objetivo se consolida con la presencia de dispositivos móviles para las transacciones y otros procesos, se usó las librerías de React Icons por ser de licencia libre para la construcción del Frontend personalizado a cada módulo.

Por otro lado, para la persistencia de datos se utilizó PostgreSQL, debido a la gran documentación existente junto a ser de código abierto para desarrollo activo. Según IBM (s.f.), las consultas (queries) en SQL, también denominadas sentencias o sentencias SQL, facilitan gestionar datos en una base de datos relacional, abarcando desde la definición de estructuras mediante DDL (Data Definition Language) hasta la manipulación y recuperación de registros. Teniendo en cuenta la definición anterior, para este proyecto las consultas no se escriben directamente en la consola de PostgreSQL, en su lugar se integran en el desarrollo mediante Node.js más Express, facilitando enviar instrucciones a la base de datos desde el backend y proveer los datos a través de APIs RESTful.

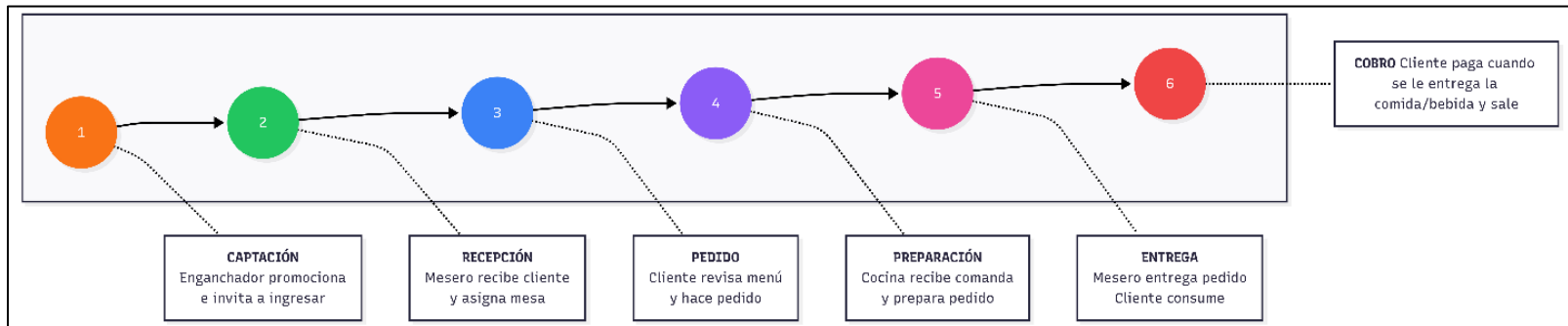
De este modo, el producto final se diseñó teniendo los parámetros anteriores, junto al objetivo general y requisitos recopilados denotados en cada módulo funcional del sistema desplegado en web, el sistema es funcional, escalable y centrado al usuario.

## **1.2 Procesos Actuales**

Dentro de proceso actual, debemos tener en cuenta este flujo de procesos general:

**Figura 1**

*Proceso Actual de Atención*



Basándome en el proceso actual de la Figura 1 los problemas objetivos se basan en tiempo con el cuello de botella de atención para el cliente, seguidos por pedidos mal interpretados al ser de carácter humano las comandas u olvidos del personal hacen cálculos errores afectando a los ingresos por cuenta de usuarios del establecimiento, generando una terrible experiencia e insatisfacción.

### 1.3 Infraestructura Actual

Para entender la situación operacional de la empresa resalto el uso de sus equipos:

**Tabla 1**

*Infraestructura Tecnológica Actual*

N°	Equipo/Dispositivo	Modelo/Marca	Estado	Función	Características
1	Laptop central	Lenovo	Buen Estado	Sistema de facturación	Intel Core i7, 8GB RAM, 512GB SSD, Windows 11
2	Impresora Térmica	Hp	Regular	Impresión de facturas/tickets	80mm, USB, velocidad 200mm/s, corte automático
3	Terminal de cobro (Datafast/POS)	Dell	Buen estado	Pagos con tarjeta crédito/débito	Pantalla táctil 10", lector chip/NFC, conexión WiFi/Ethernet
4	Dispositivo Movil	Xiaomi	Buen estado	Toma de pedidos	Android 12+, 4GB RAM, 64GB almacenamiento, pantalla 6.5"

*Nota.* Los equipos son propiedad de Bounty Bar para operaciones comerciales.

Cabe resaltar que la situación futura según la base operativa por parte del Gerente va a ser la adquisición de nuevos equipos para escalar operaciones comerciales, pero a corte de presentación del presente proyecto la Tabla 1 es la representación tecnológica que se encuentran en ejecución.

#### **1.4 Marco de Trabajo Scrum**

Este trabajo se basa en la participación activa de diferentes roles que permiten ajustar prioridades en cada sprint mediante el feedback recibido por cada usuario. Pero en este proyecto los usuarios prioritarios participan activamente siendo divididos en:

El Product Owner es asumido por el administrador del Bar Bounty, enfocándose en definir el ¿por qué? junto con el ¿cómo? de cada necesidad operativa del negocio pasa a ser un elemento del Product Backlog lo que reduce el riesgo de construir funcionalidades que no aporten valor, simultáneamente, es con quien se valida los criterios de aceptación para asegurar que cada entrega cumpla con coherencia entre lo solicitado y lo entregado.

Si los criterios de aceptación son validados, el producto Backlog recibe un incremento al tener funcionalidades para desarrollo, encargado del Developer Team, pero tengamos en cuenta que la parte digital es un complemento de estrategias, arquitectura y definiciones claras del modelo de negocio para llegar al código, por lo que el ultimo participante activo, siendo el Scrum Master dentro del contexto del proyecto promueve la mejora continua del proceso de desarrollo.

Así, bajo este concepto “Con Scrum, los equipos de desarrollo pueden enfocarse en la entrega continua de valor. Además, permite una mejor gestión del tiempo y los recursos, lo que resulta en una mayor eficiencia y en la reducción de costos” (UNIR México, 2025). El valor final

del proyecto es materializar la visión del cliente de forma funcional respecto a la plataforma digital.

## 1.5 Sprints

El desarrollo del sistema se organizará en 6 Sprints de 2 semanas cada uno, para poder realizar cada actividad dentro de los plazos establecidos incluyendo en algunos casos fines de semana junto a días feriados, debido a que, el compromiso es poder obtener resultados con una versión operativa.

Cada sprint tiene objetivos específicos y entregables definidos, pero al ser el primer avance, el módulo para Sprints queda definido a:

**Tabla 2**

*Sprints 1- Configuración Base del Sistema*

Tarea	Tipo	Responsable	Inicio	Fin	Días	Estado
Sprint 1 - Configuración y diseño base			15/10/2025	28/10/2025	14	Finalizado
Investigación y selección de tecnologías	Desarrollo	Nelson Allauca	15/10/2025	17/10/2025	3	Finalizado
Configuración del entorno de desarrollo	Configuración	Nelson Allauca	18/10/2025	19/10/2025	2	Finalizado

Configuración de repositorio GitHub	Configuración	Nelson Allauca	20/10/2025	20/10/2025	1	Finalizado
Diseño de estructura de base de datos	Diseño	Nelson Allauca	21/10/2025	22/10/2025	2	Finalizado
Creación de mockups de interfaz	Diseño UI/UX	Nelson Allauca	23/10/2025	26/10/2025	4	Finalizado
Revisión y aprobación de prototipos	Validación	Daniel Chamorro	27/10/2025	28/10/2025	2	Finalizado

**Tabla 3**

*Sprints 2 – Primeros Avances Modulares*

Tarea	Tipo	Responsable	Inicio	Fin	Días	Estado
Sprint 2 - módulo de gestión de clientes			29/10/2025	12/11/2025	14	Finalizado
Desarrollo de interfaz de registro	Desarrollo Frontend	Nelson Allauca	29/10/2025	31/10/2025	3	Finalizado

Implementación de autenticación Jwt	Desarrollo	Nelson	01/11/2025	03/11/2025	2	Finalizado
	Backend	Allauca				
Validación de datos de usuario	Desarrollo	Nelson	04/11/2025	05/11/2025	2	Finalizado
	Backend	Allauca				
Encriptación de información sensible	Seguridad	Nelson	06/11/2025	07/11/2025	2	Finalizado
		Allauca				
Pruebas del módulo de clientes	QA	Andrés	08/11/2025	08/11/2025	1	Finalizado
		Chilliquinga				
Crud de perfiles de cliente	Desarrollo	Nelson	09/11/2025	12/11/2025	4	Finalizado
	Full Stack	Allauca				

Según la representación de la Tabla 2 y 3 las necesidades por establecer servirán como primeros elementos que responden a la etapa temprana de la construcción del proyecto.

## Capítulo II

### 2.1 Análisis y Diseño

Las justificaciones previas permiten visualizar el problema actual tecnológico, ahora parte la pregunta central para avanzar en la coordinación de la información para crear el sistema, parte de la pregunta: ¿Que se puede mejorar?

En respuesta, se identificaron tres áreas de oportunidad:

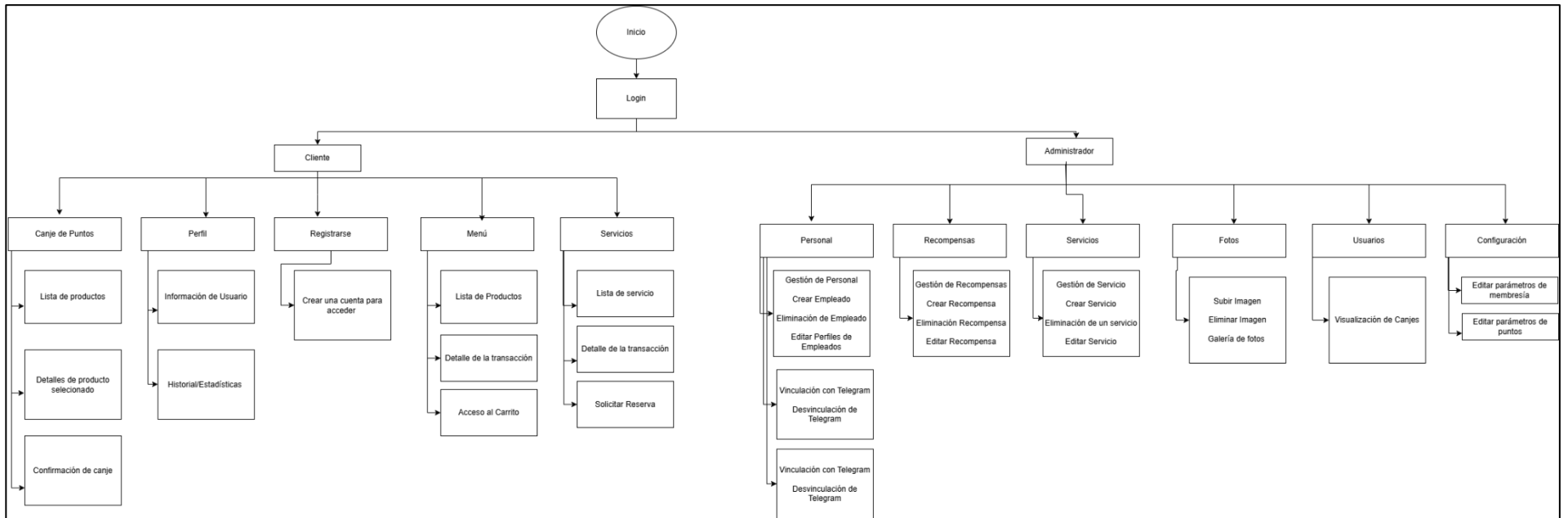
- Digitalización de Servicios: Estructurar la información de los servicios y promociones del local para generar visibilidad.
- Mejora de procesos: Incorporar herramientas que agilicen la atención y reduzcan la carga operativa del personal.
- Control administrativo: Proveer herramientas de configuración, monitoreo y control operativo.

### 2.2 Mapa de Navegación

Tomando como referencia los Sprints realizados durante el Capítulo I, se organiza el mapa de navegación para interacción de los usuarios con el sistema, según Linares Sánchez (2004) afirma que “Los sistemas de navegación [...] permiten ubicarnos en cada módulo y entender la estructura de la web como está organizada” (p.9). Esto quiere decir, que el mapa de navegación de este proyecto objetivamente mejora la comprensión del sistema para responder a donde estamos, que funciones tengo para mí usabilidad y a dónde puedo ir.

**Figura 2**

*Mapa de navegación Sistema Web Bounty Bar*



## 2.3 Diseño de arquitectura

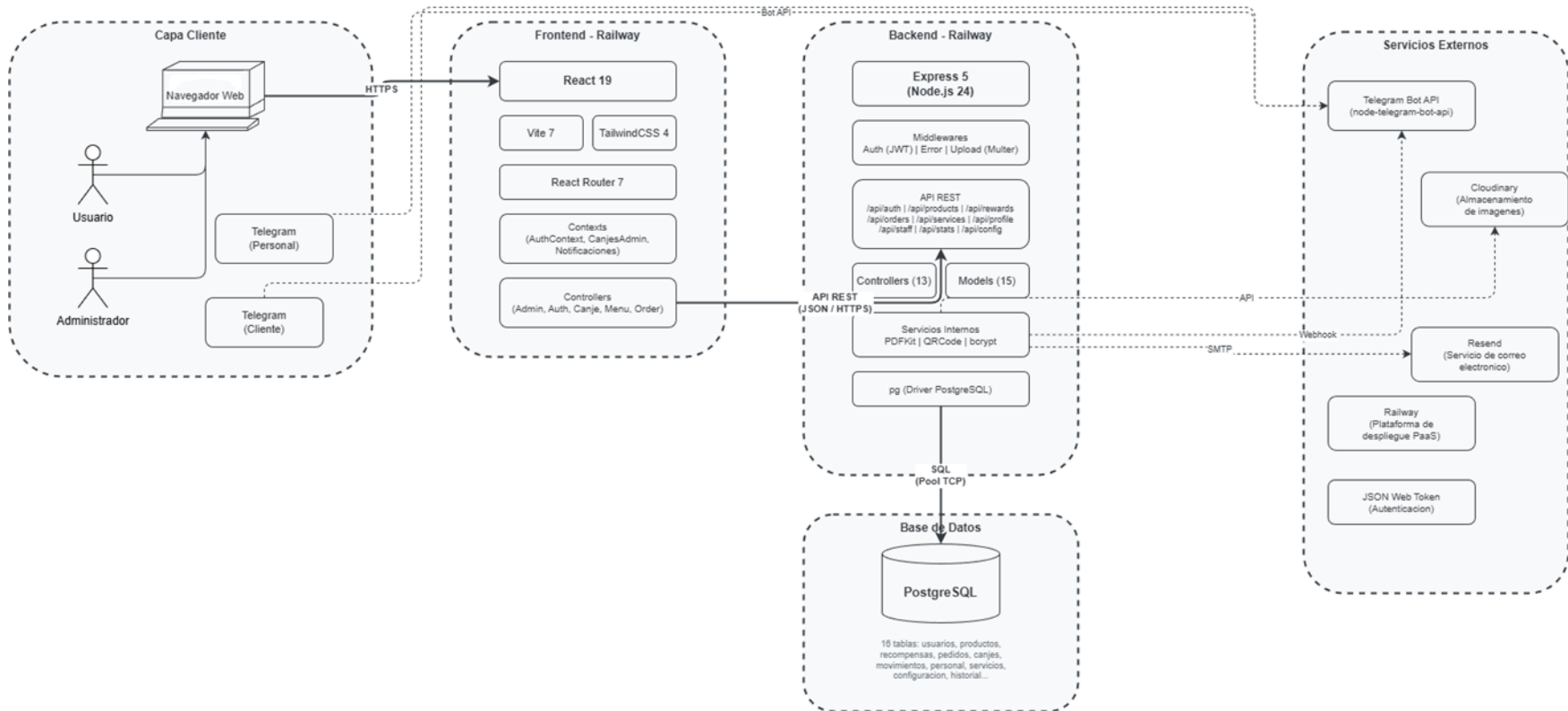
Para el presente proyecto, la idea central en el diseño de la arquitectura es tener un sistema con responsabilidades definidas, lo que reduce el acoplamiento entre componentes y facilita la detección de errores en cada nivel de forma aislada. Desde la capa de clientes, la interacción se diferencia entre los módulos de usuarios y administradores, permitiendo que cada rol opere dentro de su propio alcance.

A nivel del backend, los endpoints se estructuran según la funcionalidad que resuelven, lo que facilita tanto el mantenimiento como la incorporación de nuevos módulos sin comprometer los existentes. La comunicación entre el frontend y el backend se realiza a través de API REST mediante JSON sobre HTTPS, asegurando un intercambio de datos estandarizado y seguro. De igual forma, la capa de seguridad y el manejo de contexto trabajan de manera integrada dentro del backend, desplegado en una plataforma como servicio en la nube.

Adicionalmente, el sistema incorpora servicios externos que, si bien se acoplan al backend, no forman parte de su lógica central. Su función se limita a cubrir necesidades específicas como el almacenamiento de archivos, el envío de notificaciones y la comunicación por correo electrónico, de modo que, el diagrama de arquitectura se representaría de la siguiente manera:

**Figura 3**

*Diagrama de arquitectura del sistema Bounty*



## 2.4 Viabilidad Técnica

La empresa fundamentalmente no incluye dentro de sus inversiones infraestructura local para montar servidores propios, por lo que, se optó usar servicios basados en nube (cloud), según Algo Coding Experts (2026) afirma que “permite a las empresas ejecutar aplicaciones y servicios en un entorno virtualizado, flexible, escalable y altamente disponible” (párr.1). Bajo esta perspectiva, la visión actual del negocio es poder desplegar en nube las tecnologías descritas en la Tabla 4 para operar sus servicios.

**Tabla 4**

*Distribución de Componentes del Sistema en Plataformas Cloud*

Componente	Tecnología	Justificación Técnica
Frontend	React + Vercel	Se lo escoge por su red de contenido (CDN) lo que minimiza los tiempos de respuesta para una conexión segura.
Backend	Node.js + Express	El desarrollo se lo hizo con JavaScript y sus librerías, por su arquitectura y documentación en el modelo de operaciones se integra fácilmente a través de librerías de NPM.
Base de datos	PostgreSQL + Railway	El sistema relacional para construcción de esquemas y transacciones ACID, junto a ser Open Source mejora la compatibilidad para el proyecto.

Notificaciones	Telegram Bot API	API gratuita y estable, que permite comunicación activa a través de internet y no por servicios de pago.
----------------	------------------	--

*Nota.* Elaboración propia basada en la documentación oficial de cada plataforma.

Por consiguiente, este modelo permite llevar a cabo el proyecto para alojar el frontend y backend en la nube eliminando la necesidad de adquisición de equipos físicos, centrándose en la accesibilidad y disponibilidad del servicio para cualquier usuario desde cualquier dispositivo con internet, por lo que las condiciones técnicas son viables para el desarrollo.

## 2.5 Viabilidad Económica

Para poder ejecutar el proyecto se toma en cuenta que todos los gastos serán asumidos por el estudiante para poder desarrollarlo de forma propia, para obtener mejores resultados para la viabilidad económica se justifica en la demanda por servicio:

- Modelo Gratuito
- Modelo de Pago Mensual

En los dos casos anteriores la premisa principal es saber que ambos modelos tienen limitaciones como beneficios para la viabilidad del servicio y el poder garantizar sostenibilidad de las operaciones del sistema

### 2.5.1 Tablas de valores

#### *Costos de Desarrollo e Implementación*

#### **Tabla 5**

#### *Costos por Desarrollo*

Categoría	Descripción	Costo (USD)
-----------	-------------	-------------

Materiales Y Suministros	Licencias De Software	\$0.00
	Herramientas De Diseño (Canva Gratuito)	\$0.00
	Impresiones De Documentación	\$13.00
Transporte	Movilización A Bounty Club (Metro, Ecovía, Bus)	\$20.00
Imprevistos	Gastos por pruebas	\$50.00
	Espacio Para Backups En La Nube	\$50.00
Total Desarrollo		\$133.00

Nota. *Costos Únicos Durante La Fase De Desarrollo Del Proyecto.*

### ***Costos de Operación***

**Tabla 6**

*Escenario Plan Pago Mensual*

Componente	Tecnología	Mensual	Anual
Frontend	Vercel Pro	\$20.00	\$240.00
Backend	Railway Starter	\$5.00	\$60.00
Base De Datos	Railway	\$5.00	\$60.00
	PostgreSQL		
Notificaciones	Telegram Bot	\$0.00	\$0.00
	API		
Dominio	Dominio. Com	\$1.25	\$15.00
Total Operación		\$31.25	\$375.00

Nota. *Costos Referenciales Basados en Cada Plataforma.*

### ***Resumen De Costos***

**Tabla 7***Resumen por Costos*

Concepto	Plan Gratuito	Plan de Pago
Costos de Desarrollo e Implementación	\$133.00	\$133.00
Costos de Operación Anual	\$0.00	\$375.00
Costos de Mantenimiento Anual	\$0.00	\$0.00
<b>Total Primer Año</b>	<b>\$133.00</b>	<b>\$508.00</b>

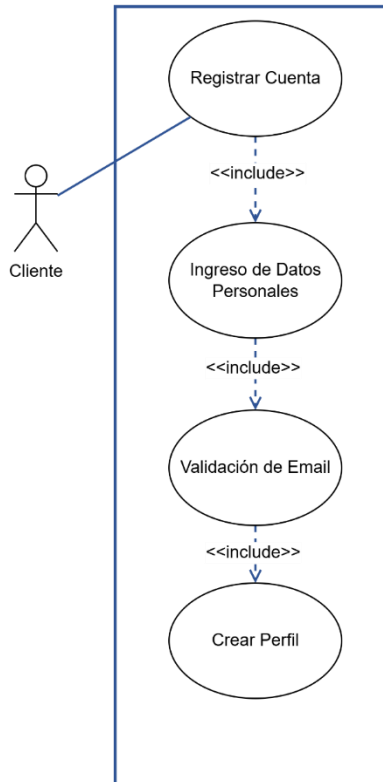
*Nota.* Comparativas de precios en planes de implementación a Bounty Bar Club

Por ende, haciendo referencia al costo anual, durante el primer año la inversión asciende a \$508.00 dólares americanos, considerando los gastos de desarrollo, operación e implementación. A partir del segundo año, el costo no supera los \$375.00 dólares americanos, lo que representa una inversión accesible que permite consolidar la opción de pago, garantizando a su vez la sostenibilidad de los gastos operativos del sistema a futuro.

## 2.6 Casos de Uso

**Figura 4**

*Creación de Email*



**Tabla 8**

*Creación de un perfil*

<b>CU-01</b>	Registrar Usuario
<b>Rol</b>	Usuario
<b>Objetivos</b>	OBJ-01: Permitir el registro de nuevos clientes en el sistema.
<b>Asociados</b>	

<b>Requisitos Asociados</b>	RF-01: Gestión de usuarios.
<b>Descripción</b>	El sistema deberá comportarse como se describe en el siguiente caso de uso cuando un nuevo cliente desee registrarse en la aplicación.
<b>Precondición</b>	El usuario no debe tener una cuenta registrada en el sistema.
<b>Secuencia Normal</b>	<ol style="list-style-type: none"> <li>1. El usuario accede a la pantalla de registro.</li> <li>2. El sistema muestra el formulario de registro solicitando: nombre, correo electrónico, teléfono y contraseña.</li> <li>3. El usuario completa los campos requeridos.</li> <li>4. El sistema valida que el correo electrónico no esté registrado.</li> <li>5. El sistema valida el formato de los datos ingresados.</li> <li>6. El sistema crea la cuenta y registra al usuario en la base de datos.</li> <li>7. El sistema redirige al usuario a la pantalla de inicio de sesión.</li> </ol>
<b>Excepciones</b>	<ol style="list-style-type: none"> <li>4. Si el correo electrónico ya está registrado, el sistema muestra un mensaje de error. Ir al paso 2.</li> <li>5. Si los datos no cumplen el formato requerido, el sistema muestra un mensaje indicando los errores. Ir al paso 2.</li> </ol>
<b>Postcondición</b>	El usuario queda registrado en el sistema y puede iniciar sesión.

**Figura 5**

*Inicio de sesión*



**Tabla 9**

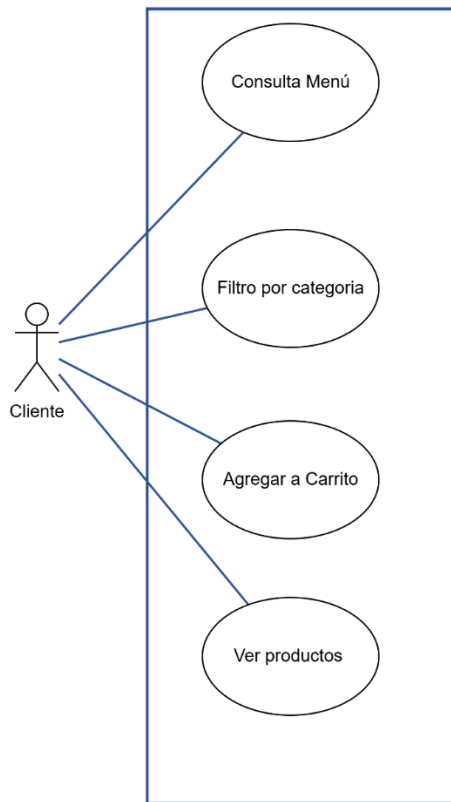
*Logueo para el sistema web bounty*

<b>CU-02</b>	Iniciar Sesión
<b>Rol</b>	Usuario
<b>Objetivos</b>	OBJ-02: Acceso controlado a la aplicación.
<b>Asociados</b>	

<b>Requisitos Asociados</b>	RF-02: Autenticación de usuarios.
<b>Descripción</b>	El sistema deberá comportarse como se describe en el siguiente caso de uso cuando un usuario registrado desee acceder a la aplicación.
<b>Precondición</b>	El usuario debe tener una cuenta registrada y credenciales válidas.
<b>Secuencia Normal</b>	<ol style="list-style-type: none"> <li>1. El usuario accede a la pantalla de inicio de sesión.</li> <li>2. El sistema solicita correo electrónico y contraseña.</li> <li>3. El usuario ingresa sus credenciales.</li> <li>4. El sistema verifica las credenciales en la base de datos.</li> <li>5. El sistema genera un token de autenticación JWT.</li> <li>6. El sistema redirige al usuario a la página principal de la aplicación.</li> </ol>
<b>Excepciones</b>	<p>4. Si el correo electrónico no existe, el sistema muestra un mensaje de error. Ir al paso 2.</p> <p>4. Si la contraseña es incorrecta, el sistema muestra un mensaje de error. Ir al paso 2.</p>
<b>Postcondición</b>	El usuario accede a la página principal con su sesión activa.

**Figura 6**

*Ingreso a menú*



**Tabla 10**

*Proceso al elegir un producto*

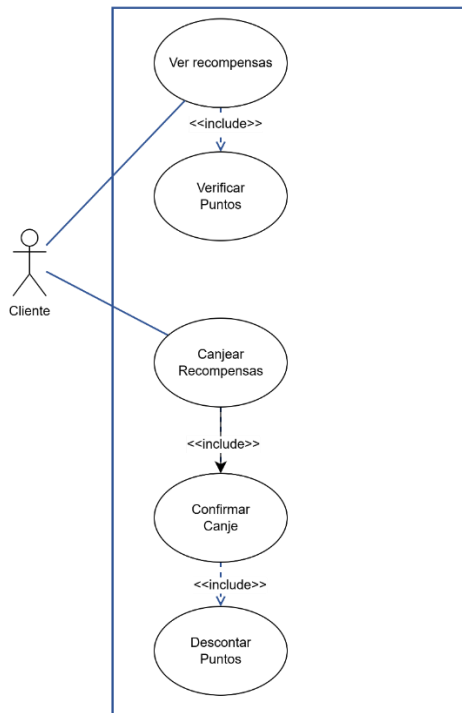
<b>CU-03</b>	Realizar Pedido
<b>Rol</b>	Cliente
<b>Objetivos</b>	OBJ-03: Permitir a los clientes realizar pedidos de productos.
<b>Asociados</b>	

<b>Requisitos Asociados</b>	RF-03: Gestión de pedidos y acumulación de puntos.
<b>Descripción</b>	El sistema deberá comportarse como se describe en el siguiente caso de uso cuando un cliente desee realizar un pedido de productos.
<b>Precondición</b>	El usuario debe estar autenticado y tener menos productos agregados al carrito.
<b>Secuencia Normal</b>	<ol style="list-style-type: none"> <li>1. El usuario accede al menú de productos.</li> <li>2. El sistema muestra la lista de productos disponibles con precios y puntos a ganar.</li> <li>3. El usuario selecciona productos y cantidades, agregándolos al carrito.</li> <li>4. El usuario accede al carrito y revisa su pedido.</li> <li>5. El usuario ingresa el número de mesa y notas opcionales.</li> <li>6. El usuario confirma el pedido.</li> <li>7. El sistema calcula el total y los puntos a acumular.</li> <li>8. El sistema genera un pdf del pedido.</li> <li>9. El sistema envía una notificación acerca de producto comprado</li> </ol>
<b>Excepciones</b>	<ol style="list-style-type: none"> <li>3. Si un producto no está disponible, el sistema muestra un mensaje de advertencia.</li> <li>6. Si el usuario tiene 3 o más pedidos activos, el sistema muestra un mensaje de error.</li> </ol>

<b>Postcondición</b>	El pedido queda registrado con estado pendiente y el cliente puede dar seguimiento mediante el código QR.
----------------------	---

**Figura 7**

*Canje de puntos por cada cliente*



**Tabla 11**

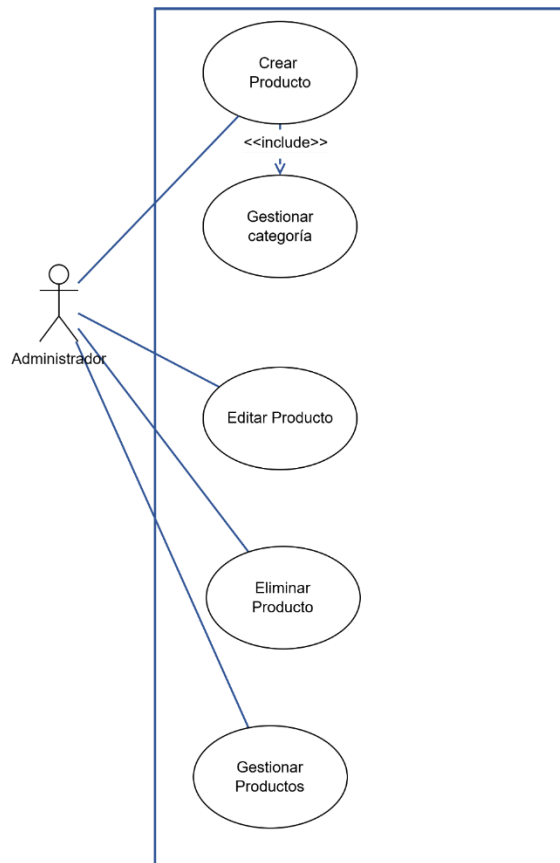
*Proceso de canje por puntos*

<b>CU-04</b>	Canjear Recompensa
<b>Rol</b>	Cliente
<b>Objetivos</b>	OBJ-04: Permitir el canje de puntos por recompensas.
<b>Asociados</b>	

<b>Requisitos Asociados</b>	RF-04: Sistema de recompensas y canjes.
<b>Descripción</b>	El sistema deberá poder a un cliente que desee canjear sus puntos acumulados por una recompensa.
<b>Precondición</b>	El usuario debe estar autenticado y tener puntos suficientes para el canje.
<b>Secuencia Normal</b>	<ol style="list-style-type: none"> <li>1. El usuario accede a la sección de recompensas.</li> <li>2. El sistema muestra las recompensas disponibles con su costo en puntos.</li> <li>3. El usuario selecciona una recompensa.</li> <li>4. El sistema muestra el detalle de la recompensa y solicita confirmación.</li> <li>5. El usuario confirma el canje.</li> <li>6. El sistema verifica que el usuario tenga puntos suficientes.</li> <li>7. El sistema descuenta los puntos del usuario.</li> <li>8. El sistema genera un código de canje único.</li> <li>9. El sistema muestra el código de canje al usuario.</li> </ol>
<b>Excepciones</b>	<p>6. Si el usuario no tiene puntos suficientes, el sistema muestra un mensaje indicando los puntos requeridos y disponibles. Ir al paso 2.</p> <p>6. Si la recompensa no tiene stock disponible, el sistema muestra un mensaje de error. Ir al paso 2.</p>
<b>Postcondición</b>	Los puntos son descontados y el usuario obtiene un código de canje para reclamar su recompensa.

**Figura 8**

*Panel de gestión de productos*



**Tabla 12**

*Gestión de productos por Administrador*

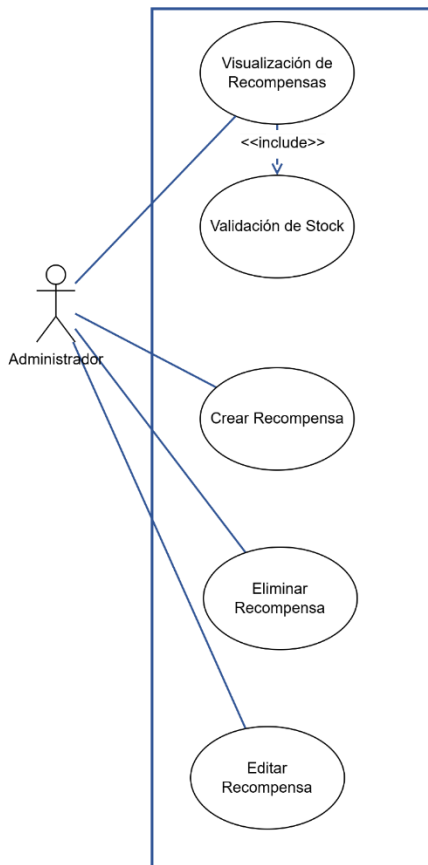
<b>CU-05</b>	Gestionar Productos
<b>Rol</b>	Administrador
<b>Objetivos</b>	OBJ-05: Administrar el catálogo de productos del establecimiento.
<b>Asociados</b>	

<b>Requisitos Asociados</b>	RF-05: CRUD de productos.
<b>Descripción</b>	El sistema deberá comportarse como se describe en el siguiente caso de uso cuando el administrador desee gestionar los productos del menú.
<b>Precondición</b>	El usuario debe estar autenticado con rol de administrador.
<b>Secuencia Normal</b>	<ol style="list-style-type: none"> <li>1. El administrador accede al panel de administración.</li> <li>2. El administrador selecciona la sección de productos.</li> <li>3. El sistema muestra la lista de productos existentes.</li> <li>4. El administrador selecciona una acción: crear, editar o eliminar.</li> <li>5. Para crear/editar: el sistema muestra el formulario con campos (nombre, descripción, precio, categoría, puntos, disponibilidad, imagen).</li> <li>6. El administrador completa o modifica los datos.</li> <li>7. El sistema valida los datos ingresados.</li> <li>8. El sistema guarda los cambios en la base de datos.</li> <li>9. El sistema actualiza la lista de productos.</li> </ol>
<b>Excepciones</b>	<ol style="list-style-type: none"> <li>7. Si los datos no son válidos, el sistema muestra los errores. Ir al paso 5.</li> <li>4. Si se intenta eliminar un producto con pedidos activos, el sistema solicita confirmación.</li> </ol>
<b>Postcondición</b>	El catálogo de productos queda actualizado según la acción realizada.

*Nota.* Este caso de uso de basa exclusivamente con los permisos de administración para gestionar cada producto dentro del local a través de una interfaz gráfica

**Figura 9**

*Opciones de creación de recompensas*



**Tabla 13**

*Creación de recompensas canjeables para usuarios*

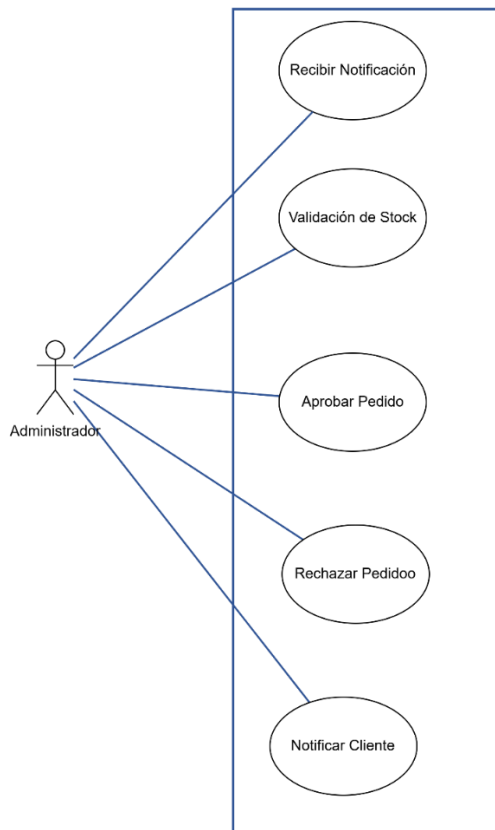
<b>CU-06</b>	Gestionar Recompensas
<b>Rol</b>	Administrador
<b>Objetivos</b>	OBJ-06: Administrar el catálogo de recompensas canjeables.
<b>Asociados</b>	

<b>Requisitos Asociados</b>	RF-06: CRUD de recompensas.
<b>Descripción</b>	El sistema deberá comportarse como se describe en el siguiente caso de uso cuando el administrador desee gestionar las recompensas disponibles para canje.
<b>Precondición</b>	El usuario debe estar autenticado con rol de administrador.
<b>Secuencia Normal</b>	<ol style="list-style-type: none"> <li>1. El administrador accede al panel de administración.</li> <li>2. El administrador selecciona la sección de recompensas.</li> <li>3. El sistema muestra la lista de recompensas existentes.</li> <li>4. El administrador selecciona una acción: crear, editar o eliminar.</li> <li>5. Para crear/editar: el sistema muestra el formulario con campos (nombre, descripción, costo en puntos, categoría, stock, imagen, disponibilidad).</li> <li>6. El administrador completa o modifica los datos.</li> <li>7. El sistema valida los datos ingresados.</li> <li>8. El sistema guarda los cambios en la base de datos.</li> <li>9. El sistema actualiza la lista de recompensas.</li> </ol>
<b>Excepciones</b>	<ol style="list-style-type: none"> <li>7.1. Si los datos no son válidos, el sistema muestra los errores. Ir al paso 5.</li> <li>4.1. Si se intenta eliminar una recompensa con canjes pendientes, el sistema muestra una advertencia.</li> </ol>

**Postcondición**

El catálogo de recompensas queda actualizado según la acción realizada.

*Nota.* Este caso de uso de basa exclusivamente con los permisos de administración para gestionar cada recompensa a los clientes dentro del local a través de un módulo de gestión

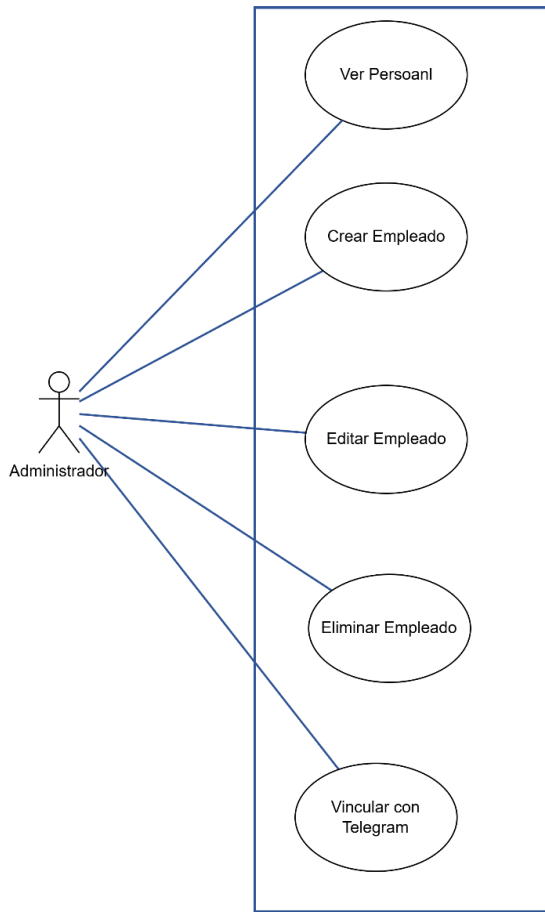
**Figura 10***Opciones dentro de telegram configuradas***Tabla 14***Aprobación de un pedido vía telegram*

<b>CU-07</b>	Aprobar Pedido
<b>Rol</b>	Administrador

<b>Objetivos Asociados</b>	OBJ-07: Gestionar el flujo de aprobación de pedidos.
<b>Requisitos Asociados</b>	RF-07: Gestión de estados de pedidos.
<b>Descripción</b>	El sistema deberá comportarse como se describe en el siguiente caso de uso cuando el administrador desee aprobar o rechazar un pedido pendiente.
<b>Precondición</b>	El usuario debe estar autenticado con rol de administrador y debe existir al menos un pedido pendiente.
<b>Secuencia Normal</b>	<ol style="list-style-type: none"> <li>1. El administrador recibe notificación de nuevo pedido vía Telegram.</li> <li>2. El administrador elige aprobar o rechazar el pedido</li> <li>3. El sistema actualiza el estado del pedido a aprobado.</li> <li>7. El sistema notifica al cliente sobre la aprobación.</li> <li>8. Cuando el pedido se completa, el sistema acredita los puntos al cliente.</li> </ol>
<b>Excepciones</b>	<ol style="list-style-type: none"> <li>2.1. Si el administrador rechaza el pedido, el sistema actualiza el estado a rechazado y notifica al cliente.</li> <li>8. Si ocurre un error al acreditar puntos, el sistema registra el error para revisión manual.</li> </ol>
<b>Postcondición</b>	El pedido cambia de estado y el cliente es notificado. Si se completa, los puntos son acreditados.

**Figura 11**

*Opciones en panel administrador para operaciones*



**Tabla 15**

*Vinculación de personal mediante telegram*

<b>CU-08</b>	Gestionar Personal
<b>Rol</b>	Administrador
<b>Objetivos</b>	OBJ-08: Administrar el personal del establecimiento.
<b>Asociados</b>	

<b>Requisitos Asociados</b>	RF-08: CRUD de personal y vinculación con Telegram.
<b>Descripción</b>	El sistema deberá poder gestionar el personal del establecimiento.
<b>Precondición</b>	El empleado debe tener el Bot de telegram en su teléfono móvil
<b>Secuencia Normal</b>	<ol style="list-style-type: none"> <li>1. El administrador accede al panel de administración.</li> <li>2. El administrador selecciona la sección de personal.</li> <li>3. El sistema muestra la lista de empleados registrados.</li> <li>4. El administrador selecciona una acción: crear, editar o eliminar.</li> <li>5. Para crear/editar: el sistema muestra el formulario con campos (nombre, cargo, teléfono, ID de Telegram, permisos).</li> <li>6. El administrador completa o modifica los datos.</li> <li>7. El sistema valida los datos ingresados.</li> <li>8. El sistema guarda los cambios en la base de datos.</li> <li>9. El sistema actualiza la lista de personal.</li> </ol>
<b>Excepciones</b>	<ol style="list-style-type: none"> <li>7. Si los datos no son válidos, el sistema muestra los errores. Ir al paso 5.</li> <li>5. Si el ID de Telegram ya está vinculado a otro empleado, el sistema muestra un error.</li> </ol>
<b>Postcondición</b>	El registro de personal queda actualizado según la acción realizada.

*Nota.* Este caso de uso de basa exclusivamente con los permisos de administración para gestionar al personal del local para atención al cliente.

## Capítulo III

### 3.1 Desarrollo

La presentación con cada caso de uso del Capítulo II son la base de las funcionalidades para cada módulo, por expandir “una descripción corta y simple de una característica que se debe incluir en el sistema contada desde la perspectiva de la persona que desea la nueva capacidad.” (Sánchez & Reyes, 2021, p. 8). A partir de lo señalado, cada historia de usuario establece los criterios funcionales que orientan el desarrollo de los módulos, complementado por los diagramas de secuencia y casos de uso que detallan el comportamiento esperado del sistema, junto a uso de:

- Arquitectura en Backend/Fronted
- Principio Solid
- Clean Architecture

Esto garantiza que, el sistema adopta una arquitectura fundamentada en separar sus responsabilidades manteniendo bajo acoplamiento y alta cohesión pues la separación de responsabilidades se adopta dentro del Backend con una estructura por capas derivada del patrón MVC para adaptarla a las Apis RESTful, junto al Frontend basado en componentes de React siendo ejecutado del lado del cliente.

### 3.2 Requerimientos del software

Los requerimientos funcionales del sistema se derivan del análisis establecido en la identificación inicial de necesidades como se menciona “El documento de especificación de requerimientos establece las funciones y capacidades que el sistema debe brindar” (Rueda,2015). En este sentido, los requerimientos serian:

**Figura 12***Requerimiento Funcionales del Sistema*

ID	Módulo	Descripción
RF-01	Gestión de Clientes	Registrar nuevos usuarios mediante formularios con validación de datos obligatorios y formato correcto.
RF-02	Gestión de Clientes	Autenticar usuarios de forma segura mediante credenciales encriptadas.
RF-03	Puntos y Recompensas	Acumular puntos automáticamente según las transacciones realizadas por el cliente.
RF-04	Puntos y Recompensas	Presentar un catálogo configurable de recompensas canjeables para el usuario.
RF-05	Puntos y Recompensas	Procesar canjes de recompensas con validación de saldo de puntos disponible.
RF-06	Puntos y Recompensas	Mostrar el historial de movimientos de puntos (acumulaciones y canjes) al usuario.
RF-07	Gestión de Promociones	Crear publicidades visibles con parámetros configurables: vigencia, tipo, valor y condiciones de aplicación.
RF-08	Dashboard Administrativo	Gestionar módulos de empleados desde el panel administrativo.
RF-09	Dashboard Administrativo	Generar y administrar publicidad desde el panel administrativo.
RF-10	Dashboard Administrativo	Visualizar el menú con los servicios disponibles para consumo.

### Figura 13

#### *Requerimientos No funcionales*

ID	Categoría	Descripción
RNF-01	Rendimiento	El sistema debe responder a solicitudes del usuario en un tiempo máximo de 10 segundos, tanto para el rol administrativo como para el de cliente.
RNF-02	Usabilidad	La interfaz debe ser responsiva y adaptable a diferentes dispositivos (laptops, tabletas y dispositivos móviles).
RNF-03	Seguridad	Almacenar contraseñas mediante algoritmos de hash seguros.
RNF-04	Seguridad	Implementar autenticación basada en tokens JWT para el manejo seguro de sesiones.
RNF-05	Seguridad	Proteger datos personales conforme a la Ley Orgánica de Protección de Datos Personales del Ecuador.
RNF-06	Portabilidad	El sistema debe ser accesible desde navegadores web sin requerir instalación de software adicional en los equipos.

### 3.3 Cronograma

Para tomar en cuenta los Sprints junto a considerar las actividades por cada módulo para el desarrollo del sistema, se implementó como parte integral del desarrollo desde la semana 3:

**Tabla 16***Sprint 3 – Modulo de puntos*

SPRINT 3						
Sistema de Puntos y Recompensas	Actividad	Encargado	12/11/2025	27/11/2025	14	Finalizado
Lógica de acumulación de puntos	Desarrollo Backend	Nelson Allauca	12/11/2025	14/11/2025	3	Finalizado
Catálogo de recompensas	Desarrollo Full Stack	Nelson Allauca	15/11/2025	17/11/2025	2	Finalizado
Proceso de canje de puntos	Desarrollo Backend	Nelson Allauca	18/11/2025	19/11/2025	2	Finalizado
Interfaz de saldo de puntos	Desarrollo Frontend	Nelson Allauca	20/11/2025	21/11/2025	2	Finalizado
Pruebas del sistema de puntos	QA	Andrés Chilliquinga	22/11/2025	22/11/2025	1	Finalizado
Historial de transacciones	Desarrollo Full Stack	Nelson Allauca	24/11/2025	27/11/2025	4	Finalizado

**Tabla 17***Sprint 4 – Diseño de Módulos para Promociones*

SPRINT 4 - Módulo de Promociones	Actividad	Encargado	29/11/2025	12/12/2025	14	Finalizado
Añadir logo	Desarrollo Fronted	Nelson Allauca	29/11/2025	01/12/2025	2	Finalizado
Programación de publicidad	Desarrollo FullStack	Nelson Allauca	02/12/2025	03/12/2025	2	Finalizado
Desarrollo en backend seguimiento de pedidos	Desarrollo Backend	Nelson Allauca	04/12/2025	05/12/2025	2	Finalizado
Pruebas generales	QA	Andrés Chilliquinga	06/12/2025	06/12/2025	1	Finalizado
Interfaz de seguimiento de pedidos del cliente	Desarrollo Fronted	Nelson Allauca	08/12/2025	10/12/2025	4	Finalizado
Personalización de mensajes	Desarrollo Frontend	Nelson Allauca	10/12/2025	12/12/2025	3	Finalizado

**Tabla 18***Sprint 5 – Diseño de Módulo Administrativo*

SPRINT 5 - Dashboard Administrativo y Bot Telegram	Actividad	Encargado	14/12/2025	19/01/2026	14	Finalizado
Panel de control	Desarrollo Frontend	Nelson Allauca	14/12/2026	21/12/2025	8	Finalizado
Corrección de bugs	Desarrollo Backend	Nelson Allauca	08/01/2026	08/01/2026	1	Finalizado
Pruebas con Personal del módulo administrativo	Desarrollo Backend	Nelson Allauca	09/01/2026	09/01/2026	1	Finalizado
Configuración Bot Telegram API	Integración	Nelson Allauca	15/01/2026	16/01/2026	2	Finalizado
Pruebas Dashboard y Bot	QA	Andrés Chilliquinga	17/01/2026	17/01/2026	1	Finalizado
Comandos y notificaciones Bot	Desarrollo Backend	Nelson Allauca	19/01/2026	19/01/2026	1	Finalizado

**Tabla 19***Sprint 6 – Integración Funcional con Pruebas*

SPRINT 6 - Integración, Pruebas y Despliegue	Actividad	Encargado	20/01/2026	02/02/2026	14	Pendiente
Ajustes finales del panel administrativo	Desarrollo	Nelson Allauca	20/01/2026	23/01/2026	4	Pendiente
Pruebas integrales del sistema	QA	Andrés Chilliquinga	24/01/2026	24/01/2026	1	Pendiente
Corrección de errores críticos	Desarrollo	Nelson Allauca	26/01/2026	28/01/2026	3	Pendiente
Despliegue en producción final	Despliegue	Nelson Allauca	29/01/2026	29/01/2026	1	Pendiente
Capacitación al personal	Capacitación	Nelson Allauca	30/01/2026	31/01/2026	2	Pendiente
Pruebas de aceptación final	QA	Andrés Chilliquinga	31/01/2026	31/01/2026	1	Pendiente

Documentación	Validación	Daniel Chamorro	01/02/2026	02/02/2026	2	Pendiente
---------------	------------	-----------------	------------	------------	---	-----------

*Nota.* Al cierre de este documento el sprint 6 se encuentra en desarrollo activo y serán finalizados para la defensa del proyecto.

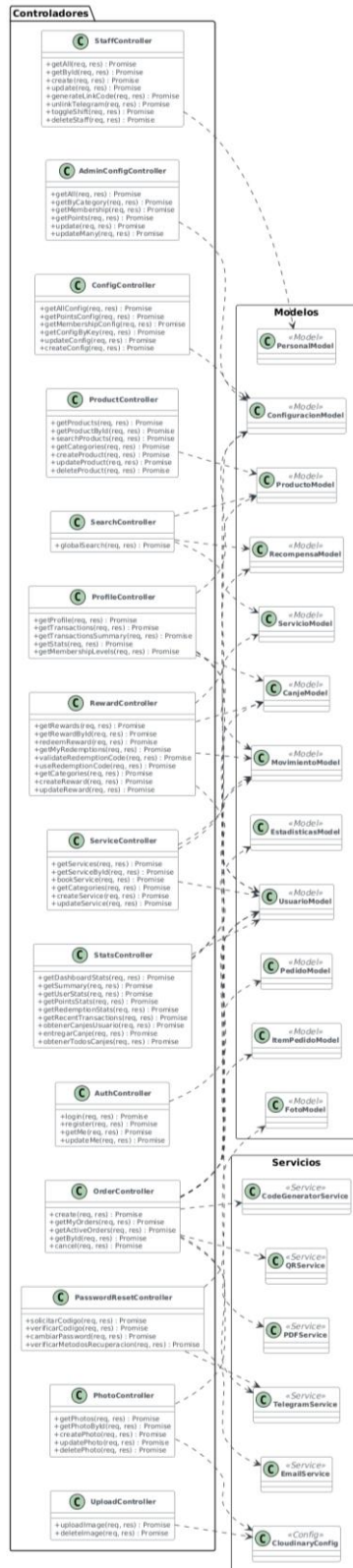
### 3.4 Diagrama de clases

La representación estructural del sistema se implementa mediante el patrón de diseño Modelo-Vista-Controlador (MVC). Para una mejor comprensión del sistema, el diagrama de clases se dividió en tres componentes: Modelo, Controlador, Componentes Transversales



Figura 15

Diagrama de clases de los controladores



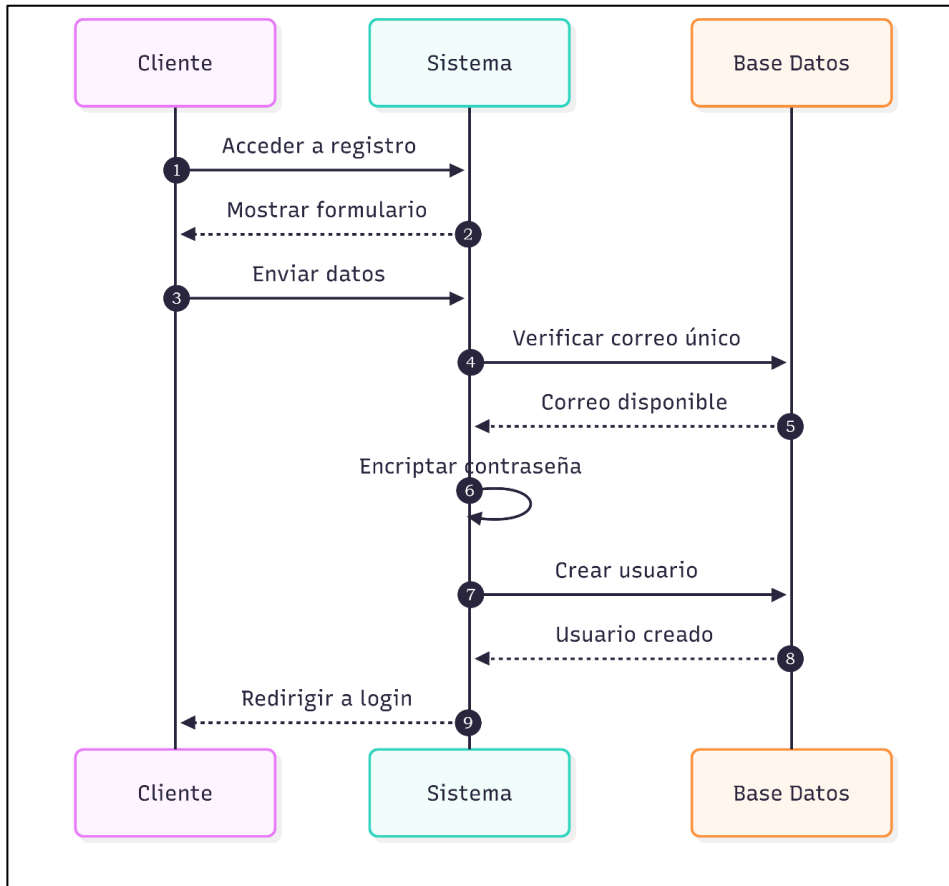


### 3.5 Diagrama de secuencia

Para hablar de secuencia vamos a dividir cada módulo para detallar su uso:

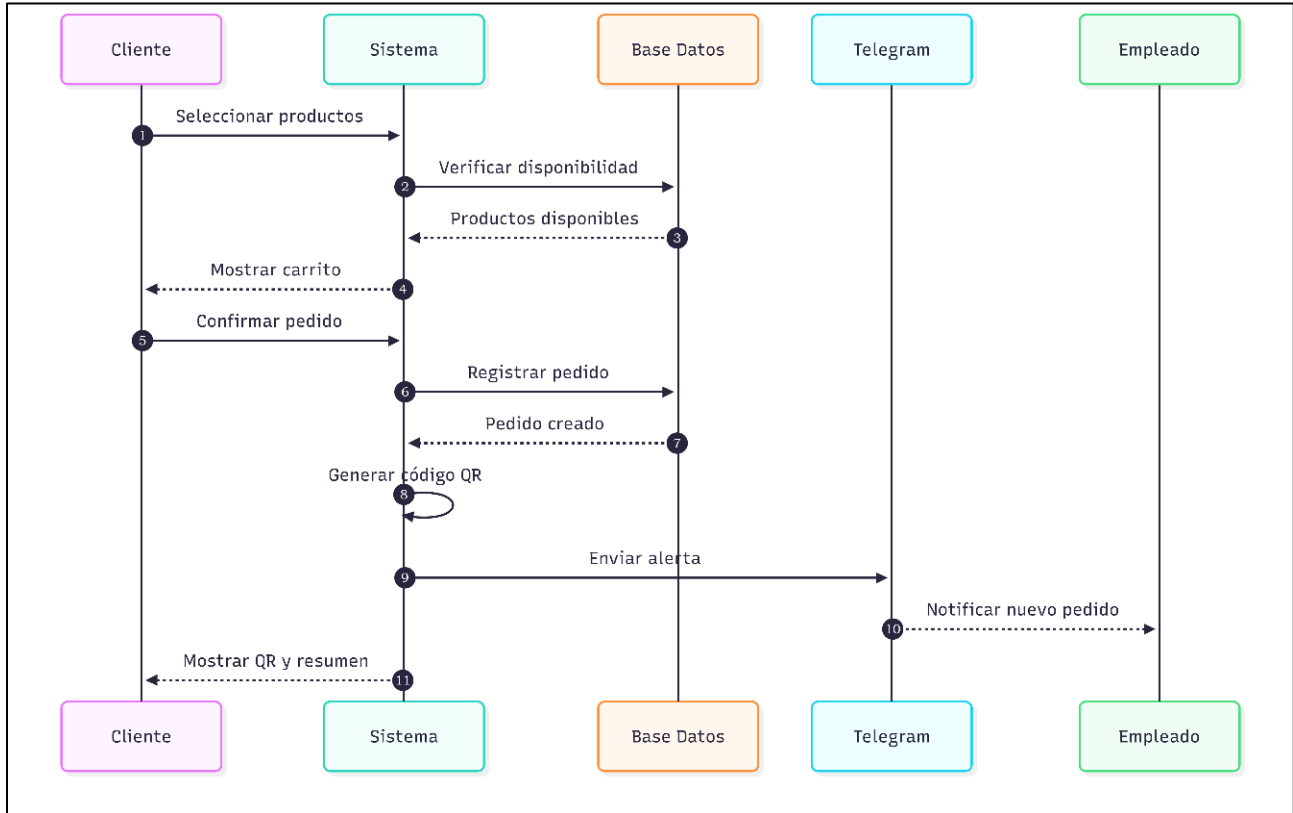
**Figura 17**

*Secuencia de registro usuario*



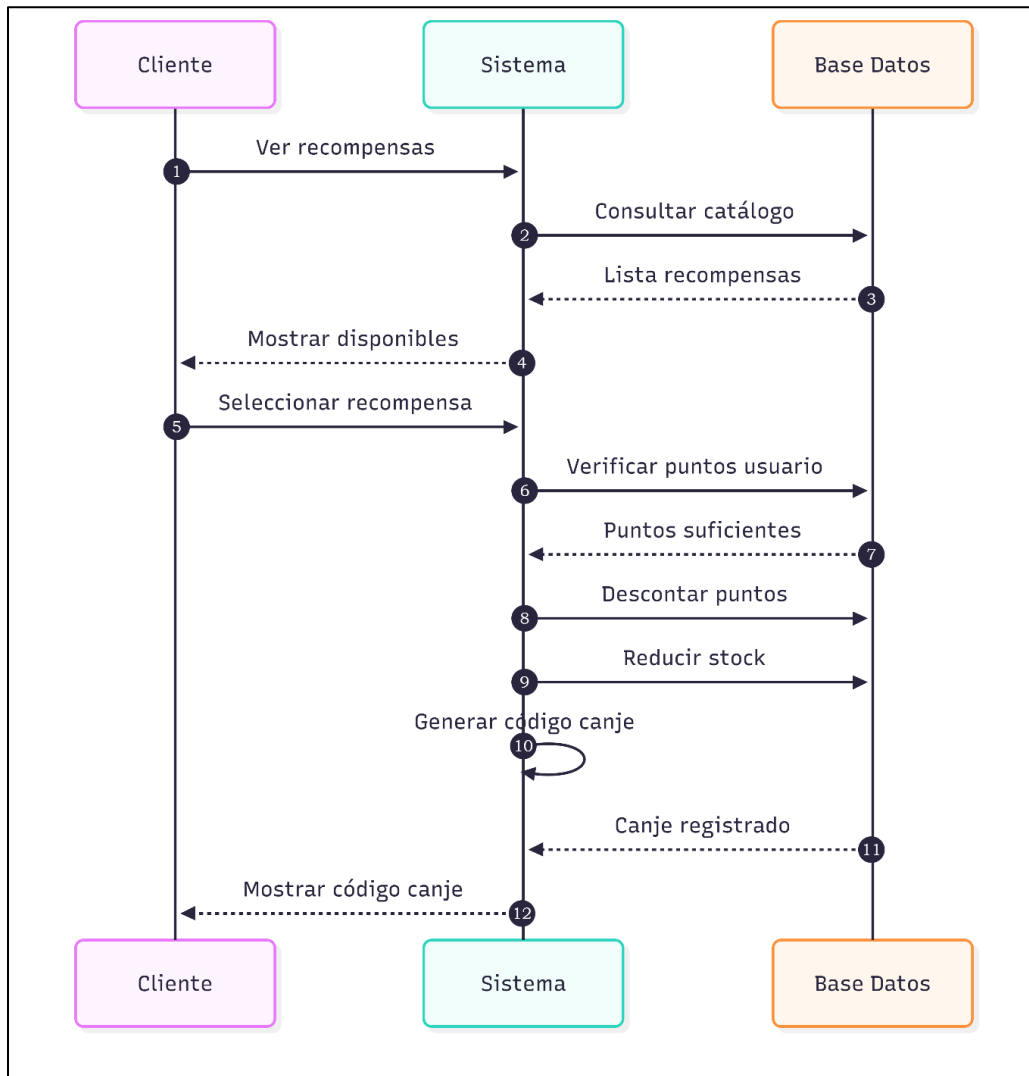
**Figura 18**

*Selección de productos de bar*



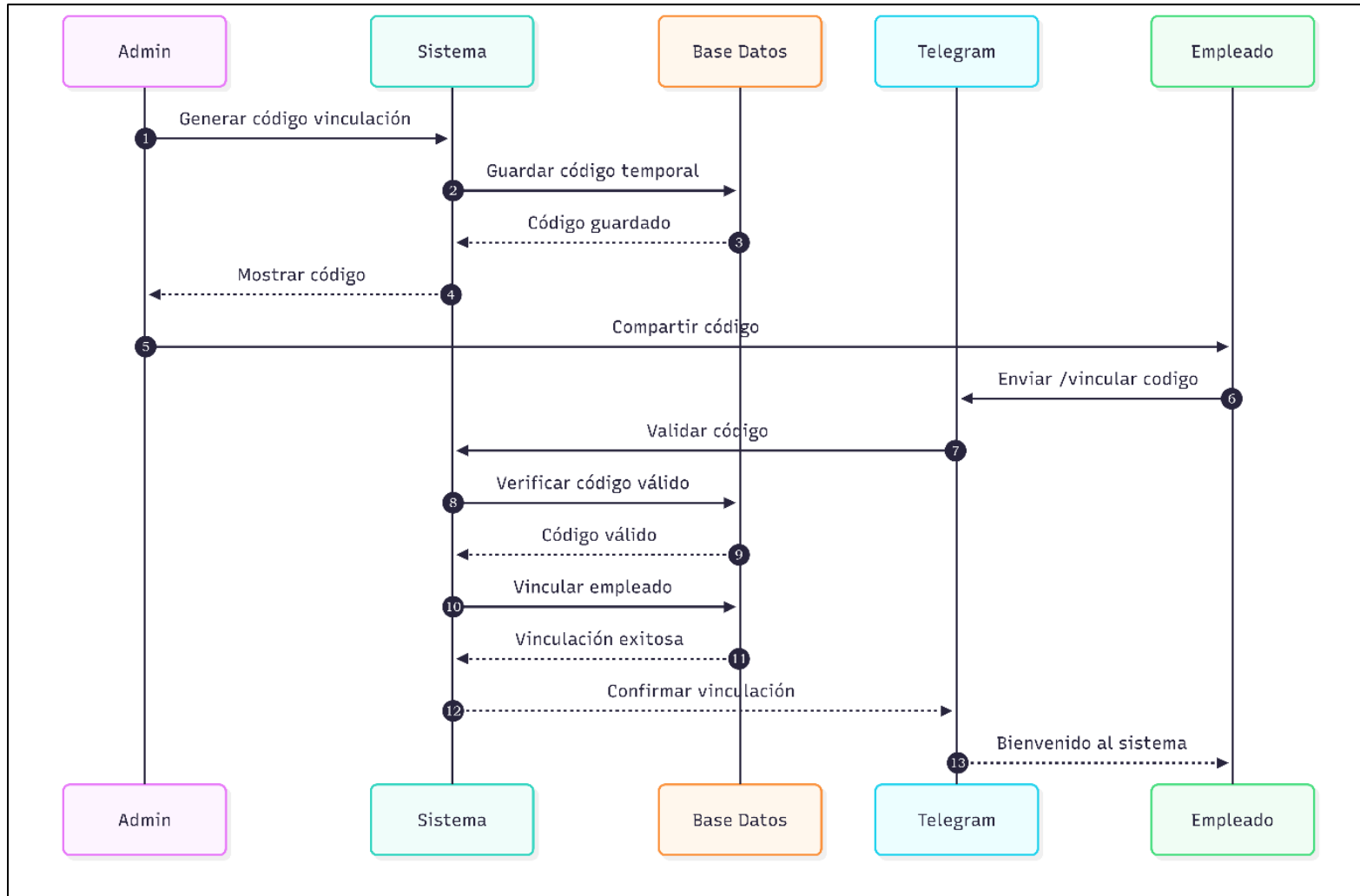
**Figura 19**

*Canje de recompensas*



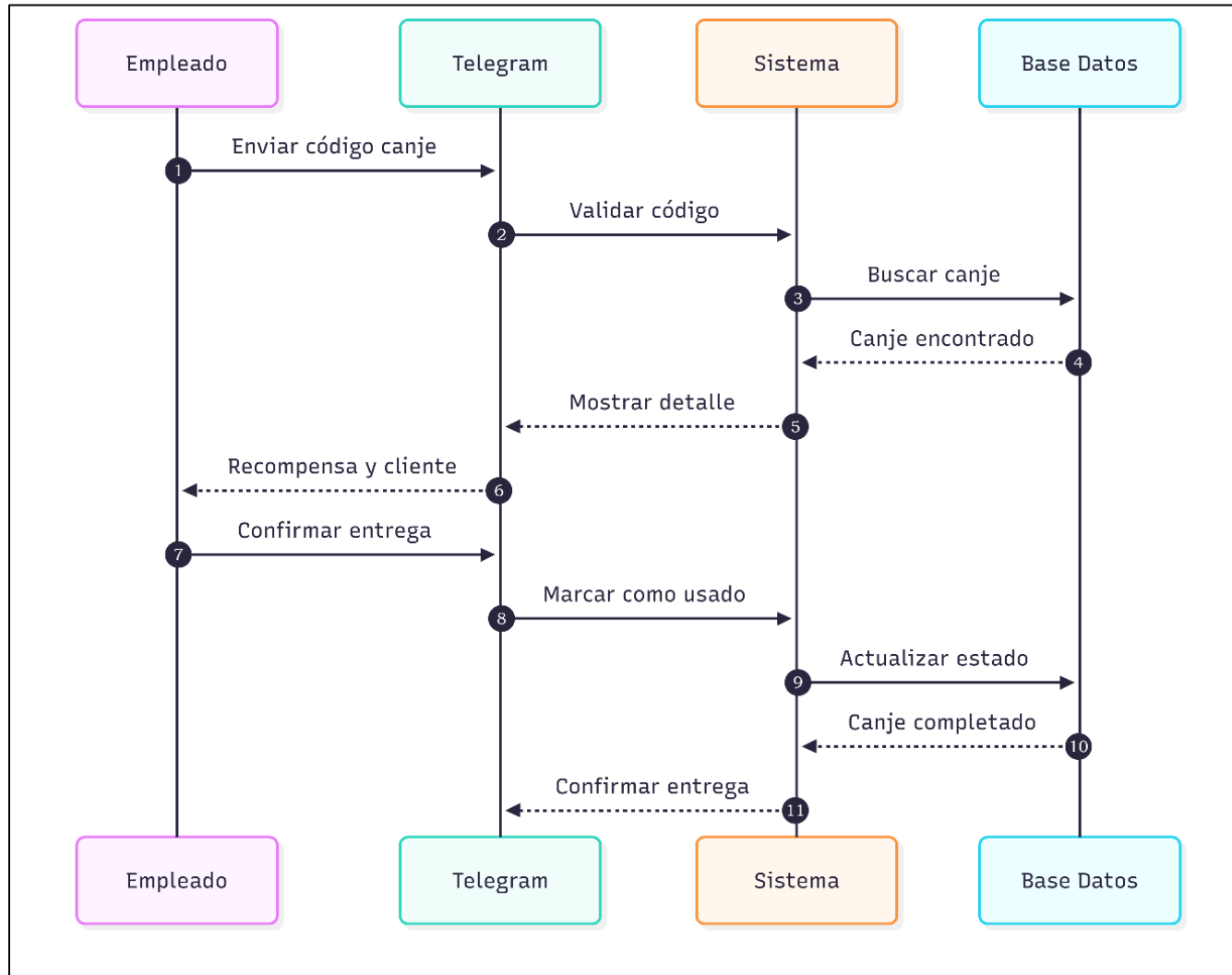
**Figura 20**

*Vinculación de Personal Bounty*



**Figura 21**

*Canje de puntos en producto*



### 3.6 Código fuente

Para analizar mejor el proceso de módulos se maneja el login como método de entrada para los usuarios:

**Figura 22**

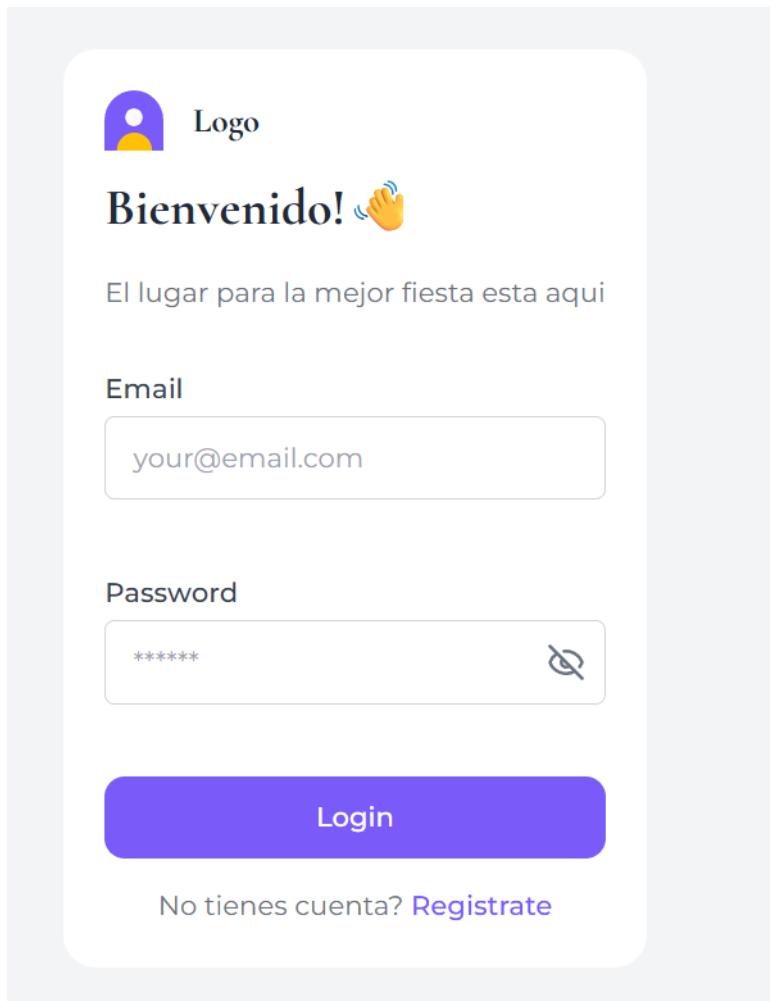
*Inputs de Login*

```
    { /* Email */ }
    <Input
      label="Email"
      type="email"
      name="email"
      placeholder="your@email.com"
      value={formData.email}
      onChange={onInputChange}
      required
    />

    { /* Password */ }
    <PasswordInput
      label="Password"
      name="password"
      value={formData.password}
      onChange={onInputChange}
      onTogglePassword={onTogglePassword}
      required
    />
```

## Figura 23

### *Login de sistema Bounty*




Logo

**Bienvenido!** 🖐️

El lugar para la mejor fiesta esta aqui

Email

Password

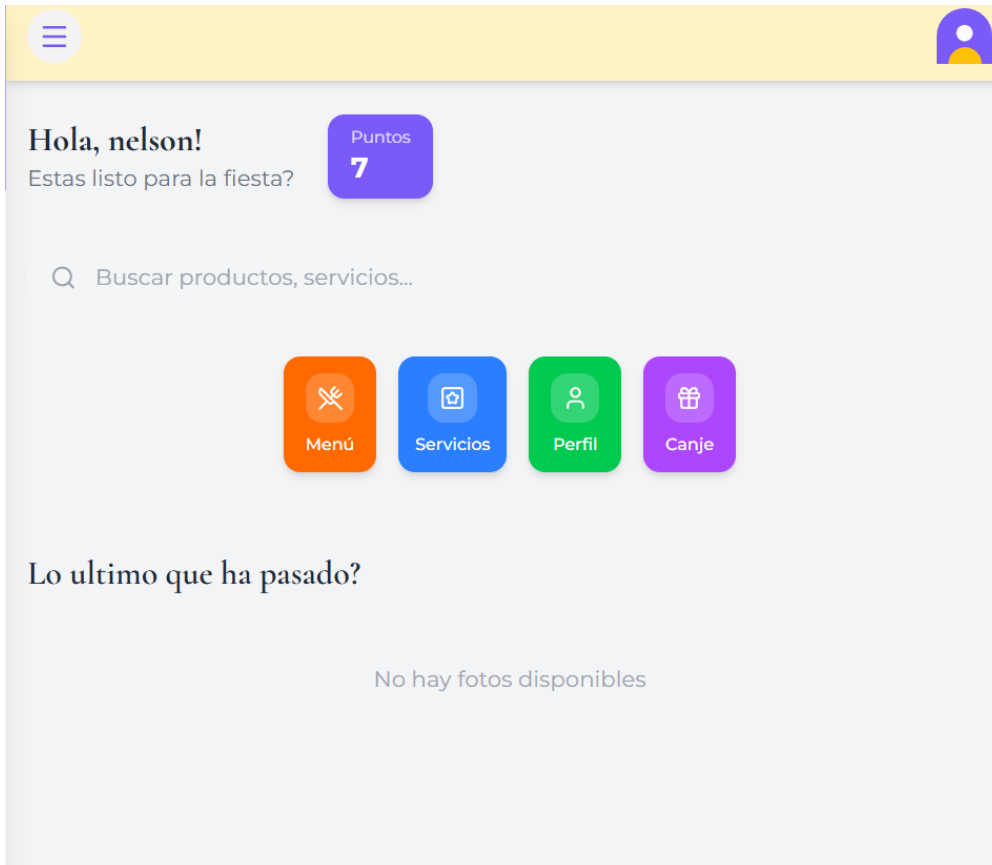
**Login**

No tienes cuenta? [Registrate](#)

Dentro del sistema la página principal después del logeo exitoso se puede acceder a las funcionalidades del sistema dependiendo del tipo de privilegio, es decir, puede ser cliente o administrador:

## Figura 24

### Home Bounty Bar Web



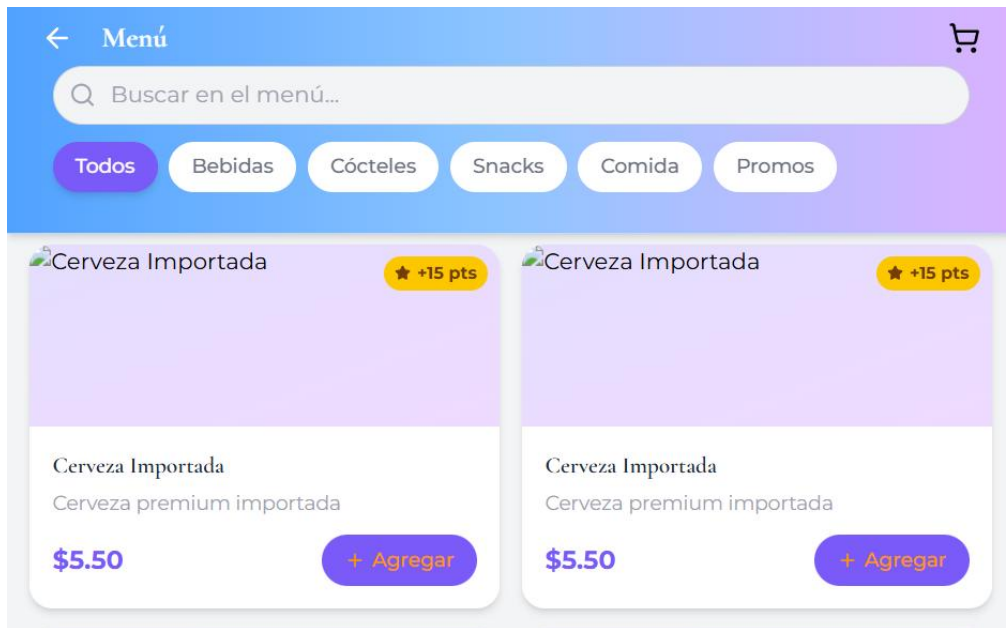
**Figura 25**

*Código de página principal*

```
/* ===== CONTENIDO PRINCIPAL ===== */
<main className="px-4 py-6">
  <div className="mb-8 flex items-center gap-6">
    <div>
      <h1 className="text-2xl font-bold text-text-primary">
        Hola{usuarioActual?.name ? ', ${usuarioActual.name.split(' ')[0]}': ''}!
      </h1>
      <p className="text-text-secondary">
        Estas listo para la fiesta?
      </p>
    </div>
    <div className="rounded-xl bg-primary px-4 py-2 text-white shadow-md">
      <p className="text-xs opacity-80">Puntos</p>
      <p className="text-xl font-bold">
        {usuarioActual?.points?.current?.toLocaleString() || '0'}
      </p>
    </div>
  </div>
  <div className="relative mb-8">
    <div className="relative">
      <input
        type="text"
        placeholder="Buscar productos, servicios..."
        value={textoBusqueda}
        onChange={manejarCambioBusqueda}
        onFocus={() => textoBusqueda.length >= 2 && setMostrarResultados(true)}
        onBlur={() => setTimeout(cerrarResultados, 200)}
        className="w-full rounded-full border border-white/30 bg-white/10 backdrop-blur-sm py-3 pl-12 pr-4 text-text-primary"
      />
    </div>
  </div>
</main>
```

**Figura 26**

*Menú de servicios Bounty*



**Figura 27**

*Código estructurado para menú*

```
return (  
  <div  
    key={item.id}  
    className={ `overflow-hidden rounded-2xl █ bg-surface-primary shadow-md transition-all hover:shadow-lg ${  
      !item.available ? 'opacity-60' : ''  
    }` }  
  >  
    { /* Imagen que debe venir de admin */ }  
    <div className="relative h-32 bg-gradient-to-br █ from-primary/20 █ to-purple-500/20 overflow-hidden">  
      {item.imageUrl ? (  
        <img  
          src={item.imageUrl}  
          alt={item.name}  
          className="h-full w-full object-cover"  
        />  
      ) : (  
        <div className="absolute inset-0 flex items-center justify-center text-4xl">  
          {getIconByCategory(item.category)}  
        </div>  
      ) }  
    <div className="absolute right-2 top-2 flex items-center gap-1 rounded-full █ bg-yellow-400 px-2 py-1 text-xs font-bo  
    <star size={12} fill="currentColor" />  
    +{item.points} pts  
  </div>  
  
  {item.available && (  
    <div className="absolute inset-0 flex items-center justify-center █ bg-black/40">  
      <span className="rounded-full █ bg-red-500 px-3 py-1 text-sm font-bold █ text-white">  
        No disponible  
      </span>  
    </div>  
  ) }  
</div>  
)
```

**Figura 28**

*Servicios de Bounty bar*

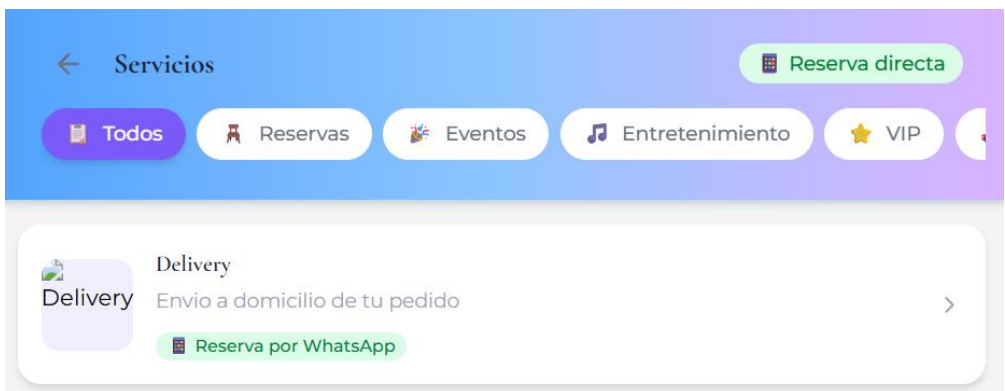


Figura 29

Código de módulo de servicios

```
import { X, MessageCircle } from 'lucide-react'

// Número de WhatsApp
const WHATSAPP_NUMBER = '593958988509'

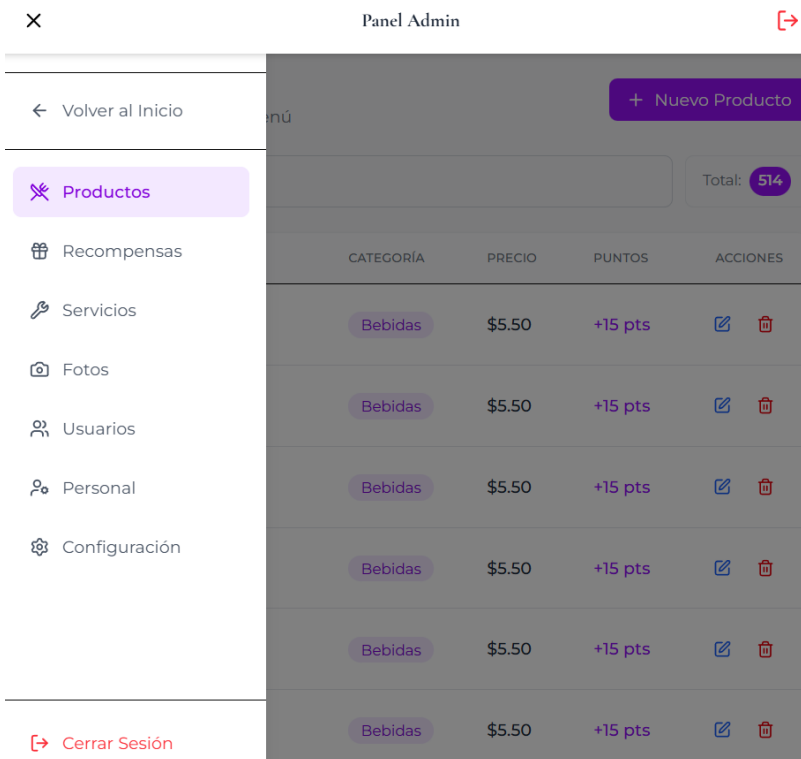
const ServicioModal = ({ service, onClose }) => {

  const contactarWhatsApp = () => {
    const mensaje = encodeURIComponent(
      `Hola! 🍷 Me interesa el servicio: *${service.name}*\n\n¿Quisiera más información para reservar.`
    )
    const url = `https://wa.me/${WHATSAPP_NUMBER}?text=${mensaje}`
    window.open(url, '_blank')
  }

  return (
    <div className="fixed inset-0 z-50 flex items-end justify-center □bg-black/50 sm:items-center">
      <div className="w-full max-w-md rounded-t-3xl □bg-white p-6 shadow-xl sm:rounded-3xl">
        <div className="flex items-start justify-between">
          <div className="flex items-center gap-4">
            {service.imageUrl ? (
              <img
                src={service.imageUrl}
                alt={service.name}
                className="h-16 w-16 rounded-xl object-cover"
              />
            ) : (
              <div className="flex h-16 w-16 items-center justify-center rounded-xl □bg-primary/10 text-4xl">
                {service.icon || '🍷'}
              </div>
            )}
          </div>
          <div>
            <h2 className="text-xl font-bold □text-gray-800">
              {service.name}
            </h2>
          </div>
        </div>
      </div>
    </div>
  )
}
```

Figura 30

Panel Administrativo



## Figura 31

### Código de modulo Administrador

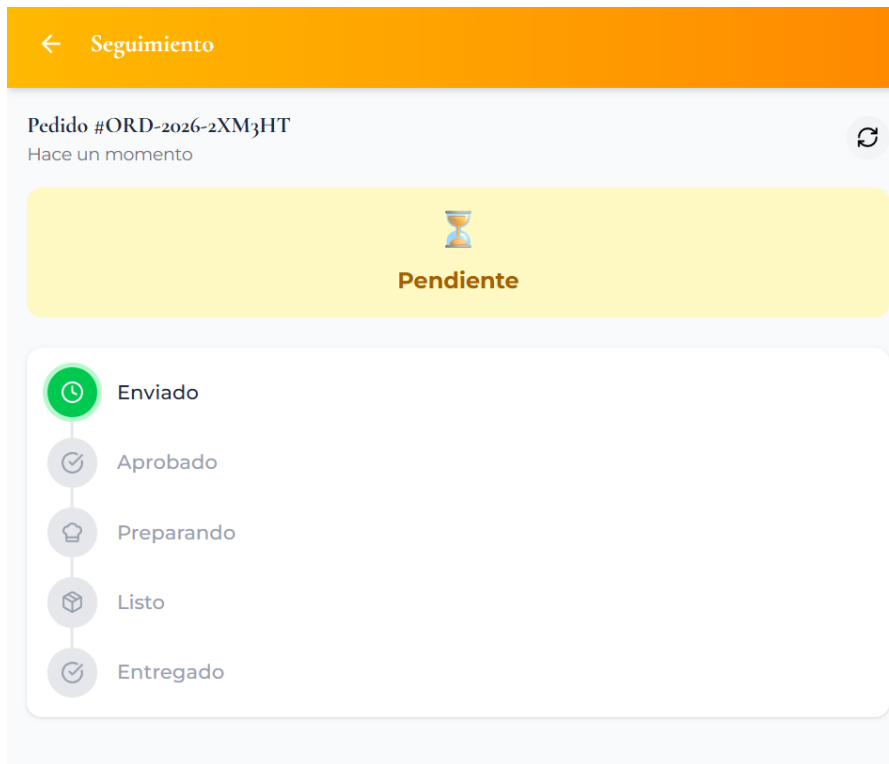
```
const AdminPage = () => {
  <Logout size={24} />
  </button>
  </div>
  </header>

  <div className="flex">
    <aside
      className={
        fixed lg:sticky top-0 left-0 z-40 h-screen w-64 bg-white shadow-lg
        transform transition-transform duration-300 ease-in-out
        ${sidebarOpen ? 'translate-x-0' : '-translate-x-full'}
        lg:translate-x-0
      }
    >
    <div className="flex items-center gap-3 px-6 py-5 border-b">
      <div className="w-10 h-10 bg-purple-600 rounded-lg flex items-center justify-center">
        <LayoutDashboard className="text-white" size={24} />
      </div>
      <div>
        <h2 className="font-bold text-gray-800">Admin Panel</h2>
        <p className="text-xs text-gray-500">Sistema</p>
      </div>
    </div>

    { /* Botón Volver */ }
    <div className="p-4 border-b">
      <button
        onClick={handleBack}
        className="w-full flex items-center gap-3 px-4 py-2 rounded-lg
          text-gray-600 hover:bg-gray-100 transition-colors"
      >
        <ArrowLeft size={20} />
        <span>Volver al Inicio</span>
      </button>
    </div>
  </div>
}
```

## Figura 32

### Seguimiento del Pedido



**Figura 33**

*Código de módulo de seguimiento*

```
const OrderTracking = ({ order, onRefresh, isLoading }) => {
  const isCancelled = order.status === 'cancelled'
  const isCompleted = order.status === 'completed' || order.status === 'delivered'
  const tieneTelegram = usuarioActual?.telegram_chat_id

  // las notificaciones vienen despues de cada accion
  useEffect(() => {
    const estadoPrevio = estadoAnterior.current

    if (order.status === 'completed' && estadoPrevio !== 'completed') {
      if (!pedidoYaNotificado(order.id)) {
        agregarNotificacionPedido(order, false)
      }
    }

    // Pedido cancelado o rechazado
    if ((order.status === 'cancelled' || order.status === 'rejected') &&
      estadoPrevio !== 'cancelled' && estadoPrevio !== 'rejected') {
      agregarNotificacionCancelado(order)
    }

    estadoAnterior.current = order.status
  }, [order, agregarNotificacionPedido, agregarNotificacionCancelado, pedidoYaNotificado])

  const irAMain = () => navigate('/main')

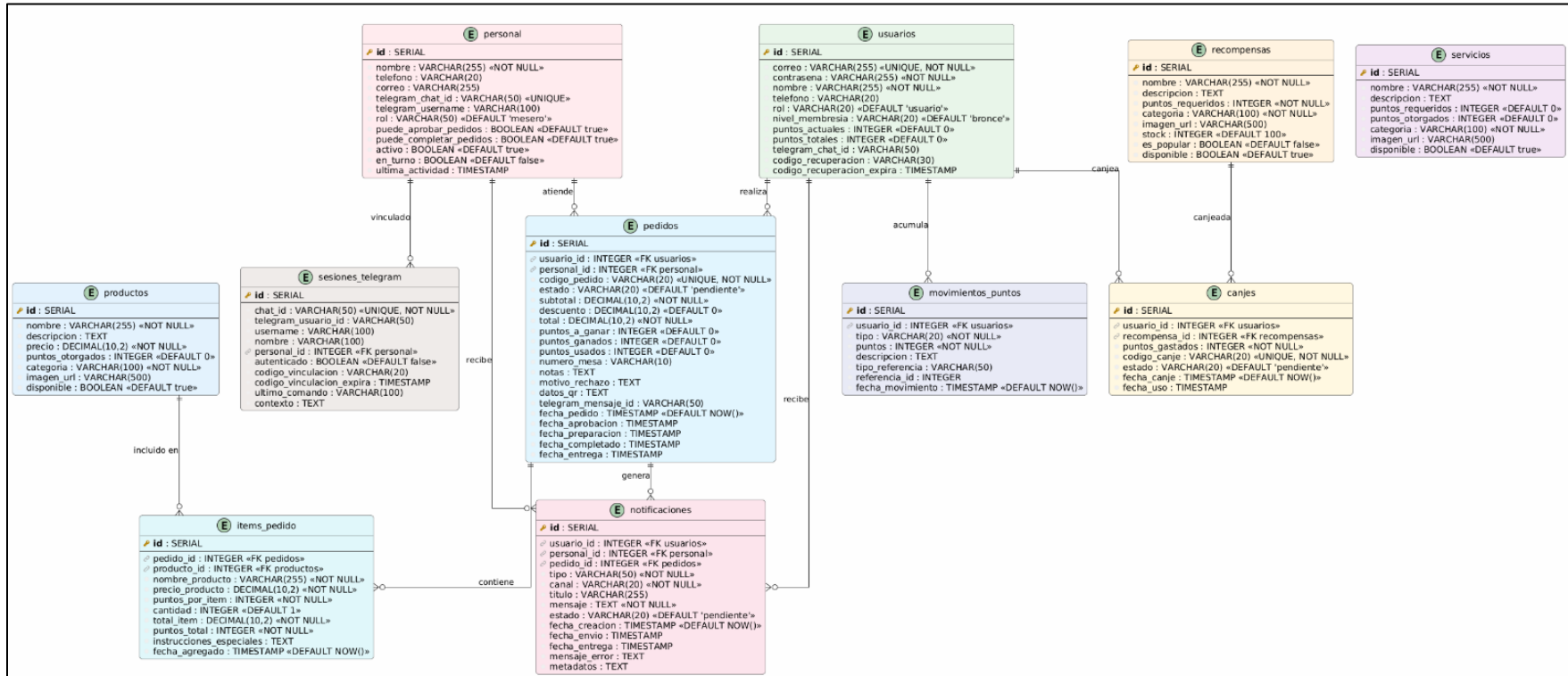
  const descargarComprobante = async () => {
    try {
      setDescargando(true)
      await orderServices.downloadPDF(order.id, order.order_code)
    } catch (error) {
      console.error('Error al descargar:', error)
    } finally {
      setDescargando(false)
    }
  }
}
```

### 3.7 Modelo Base de datos

#### 3.7.1 MER

Figura 34

Mer de sistema backend



### 3.7.2 Diccionario de datos

La información para presentar cada dato es basada en la utilización de datos guardados en PostgreSQL, detallando:

**Tabla 20**

*Estructura de la tabla usuarios del sistema*

Columna	Tipo de Dato	Condiciones	Detalle
id	SERIAL	LLAVE PRIMARIA	Identificador único del usuario
correo	VARCHAR(255)	UNIQUE NOT NULL	Correo electrónico del usuario
contrasena	VARCHAR(255)	NOT NULL	Contraseña encriptada con bcrypt
nombre	VARCHAR(255)	NOT NULL	Nombre completo del usuario
teléfono	VARCHAR(20)	NULLABLE	Número de teléfono de contacto
rol	VARCHAR(20)	DEFAULT 'usuario'	Rol del usuario (usuario/admin)
nivel_membresia	VARCHAR(20)	DEFAULT 'bronce'	Nivel de membresía (bronce/plata/oro/platino)
puntos_actuales	INTEGER	DEFAULT 0	Puntos disponibles para canjear
puntos_totales	INTEGER	DEFAULT 0	Puntos acumulados históricamente
telegram_chat_id	VARCHAR(50)	NULLABLE	ID de chat de Telegram vinculado
codigo_recuperacion	VARCHAR(30)	NULLABLE	Código temporal para recuperar contraseña
codigo_recuperacion_expira	TIMESTAMP	NULLABLE	Fecha de expiración del código

*Nota.* Tabla principal que almacena los datos de clientes registrados

**Tabla 21***Estructura de la tabla productos del establecimiento*

Columna	Tipo de Dato	Condiciones	Detalle
id	SERIAL	LLAVE PRIMARIA	Identificador único del producto
nombre	VARCHAR(255)	NOT NULL	Nombre del producto
descripción	TEXT	NULLABLE	Descripción detallada del producto
precio	DECIMAL(10,2)	NOT NULL	Precio de venta en dólares
puntos_otorgados	INTEGER	DEFAULT 0	Puntos que otorga al comprar
categoria	VARCHAR(100)	NOT NULL	Categoría del producto
imagen_url	VARCHAR(500)	NULLABLE	URL de la imagen del producto
disponible	BOOLEAN	DEFAULT true	Indica si está disponible para venta

*Nota.* Catálogo de productos disponibles para la venta en el establecimiento.

**Tabla 22***Estructura de la tabla recompensas canjeables por puntos*

Columna	Tipo de Dato	Condiciones	Detalle
id	SERIAL	LLAVE PRIMARIA	Identificador único de la recompensa
nombre	VARCHAR(255)	NOT NULL	Nombre de la recompensa
descripcion	TEXT	NULLABLE	Descripción de la recompensa
puntos_requeridos	INTEGER	NOT NULL	Puntos necesarios para canjear
categoria	VARCHAR(100)	NOT NULL	Categoría de la recompensa
imagen_url	VARCHAR(500)	NULLABLE	URL de la imagen
stock	INTEGER	DEFAULT 100	Cantidad disponible en inventario
es_popular	BOOLEAN	DEFAULT false	Indica si es una recompensa destacada

disponible	BOOLEAN	DEFAULT true	Indica si está disponible para canje
------------	---------	--------------	--------------------------------------

*Nota.* Catálogo de recompensas disponibles para canje con puntos de fidelización.

**Tabla 23**

*Estructura de la tabla servicios ofrecidos por el establecimiento*

Columna	Tipo de Dato	Condiciones	Detalle
id	SERIAL	LLAVE PRIMARIA	Identificador único del servicio
nombre	VARCHAR(255)	NOT NULL	Nombre del servicio
descripcion	TEXT	NULLABLE	Descripción del servicio
puntos_requeridos	INTEGER	DEFAULT 0	Puntos requeridos para acceder
puntos_otorgados	INTEGER	DEFAULT 0	Puntos que otorga al usar
categoria	VARCHAR(100)	NOT NULL	Categoría del servicio
imagen_url	VARCHAR(500)	NULLABLE	URL de la imagen
disponible	BOOLEAN	DEFAULT true	Indica si está disponible

*Nota.* Catálogo de servicios adicionales que ofrece el establecimiento.

**Tabla 24**

*Estructura de la tabla movimientos de puntos por consumo*

Columna	Tipo de Dato	Condiciones	Detalle
id	SERIAL	LLAVE PRIMARIA	Identificador único del movimiento
usuario_id	INTEGER	FK → usuarios	Usuario que realizó el movimiento
tipo	VARCHAR(20)	NOT NULL	Tipo de movimiento (ganado/canjeado)
puntos	INTEGER	NOT NULL	Cantidad de puntos del movimiento
descripcion	TEXT	NULLABLE	Descripción del movimiento
tipo_referencia	VARCHAR(50)	NULLABLE	Tipo de entidad relacionada

referencia_id	INTEGER	NULLABLE	ID de la entidad relacionada
fecha_movimiento	TIMESTAMP	DEFAULT NOW()	Fecha y hora del movimiento

*Nota.* Registro histórico de todas las transacciones de puntos del sistema.

**Tabla 25**

*Estructura de la tabla canjes de recompensas realizados*

Columna	Tipo de Dato	Condiciones	Detalle
id	SERIAL	LLAVE PRIMARIA	Identificador único del canje
usuario_id	INTEGER	FK → usuarios	Usuario que realizó el canje
recompensa_id	INTEGER	FK → recompensas	Recompensa canjeada
puntos_gastados	INTEGER	NOT NULL	Puntos utilizados en el canje
codigo_canje	VARCHAR(20)	UNIQUE NOT NULL	Código único de verificación
estado	VARCHAR(20)	DEFAULT 'pendiente'	Estado del canje (pendiente/usado)
fecha_canje	TIMESTAMP	DEFAULT NOW()	Fecha de realización del canje
fecha_uso	TIMESTAMP	NULLABLE	Fecha de uso de la recompensa

*Nota.* Registro de canjes de recompensas realizados por los usuarios.

**Tabla 26**

*Estructura de la tabla personal vinculado al establecimiento*

Columna	Tipo de Dato	Condiciones	Detalle
id	SERIAL	LLAVE PRIMARIA	Identificador único del empleado
nombre	VARCHAR(255)	NOT NULL	Nombre completo del empleado
telefono	VARCHAR(20)	NULLABLE	Teléfono de contacto

correo	VARCHAR(255)	NULLABLE	Correo electrónico
telegram_chat_id	VARCHAR(50)	UNIQUE	ID de Telegram para notificaciones
telegram_username	VARCHAR(100)	NULLABLE	Nombre de usuario en Telegram
rol	VARCHAR(50)	DEFAULT 'mesero'	Rol del empleado
puede_aprobar_pedidos	BOOLEAN	DEFAULT true	Permiso para aprobar pedidos
puede_completar_pedidos	BOOLEAN	DEFAULT true	Permiso para completar pedidos
activo	BOOLEAN	DEFAULT true	Indica si el empleado está activo
en_turno	BOOLEAN	DEFAULT false	Indica si está en turno actualmente
ultima_actividad	TIMESTAMP	NULLABLE	Última actividad registrada

*Nota.* Registro de empleados del establecimiento con acceso al sistema.

**Tabla 27**

*Estructura de la tabla pedidos generados por clientes*

Columna	Tipo de Dato	Condiciones	Detalle
id	SERIAL	LLAVE PRIMARIA	Identificador único del pedido
usuario_id	INTEGER	FK → usuarios	Cliente que realizó el pedido
personal_id	INTEGER	FK → personal	Empleado que atendió el pedido
codigo_pedido	VARCHAR(20)	UNIQUE NOT NULL	Código único del pedido
estado	VARCHAR(20)	DEFAULT 'pendiente'	Estado actual del pedido
subtotal	DECIMAL(10,2)	NOT NULL	Subtotal antes de descuentos
descuento	DECIMAL(10,2)	DEFAULT 0	Descuento aplicado
total	DECIMAL(10,2)	NOT NULL	Total a pagar
puntos_a_ganar	INTEGER	DEFAULT 0	Puntos a ganar al completar

puntos_ganados	INTEGER	DEFAULT 0	Puntos efectivamente ganados
numero_mesa	VARCHAR(10)	NULLABLE	Número de mesa del cliente
notas	TEXT	NULLABLE	Notas adicionales del pedido
motivo_rechazo	TEXT	NULLABLE	Motivo si fue rechazado
datos_qr	TEXT	NULLABLE	Datos para generar código QR
telegram_mensaje_id	VARCHAR(50)	NULLABLE	ID del mensaje en Telegram
fecha_pedido	TIMESTAMP	DEFAULT NOW()	Fecha de creación del pedido
fecha_aprobacion	TIMESTAMP	NULLABLE	Fecha de aprobación
fecha_preparacion	TIMESTAMP	NULLABLE	Fecha inicio preparación
fecha_completado	TIMESTAMP	NULLABLE	Fecha de completado
fecha_entrega	TIMESTAMP	NULLABLE	Fecha de entrega al cliente

*Nota.* Registro de pedidos realizados por los clientes en el establecimiento.

**Tabla 28**

*Estructura de la tabla detalle de productos por pedido*

Columna	Tipo de Dato	Condiciones	Detalle
id	SERIAL	LLAVE PRIMARIA	Identificador único del ítem
pedido_id	INTEGER	FK → pedidos	Pedido al que pertenece
producto_id	INTEGER	FK → productos	Producto seleccionado
nombre_producto	VARCHAR(255)	NOT NULL	Nombre del producto (snapshot)
precio_producto	DECIMAL(10,2)	NOT NULL	Precio unitario (snapshot)
puntos_por_item	INTEGER	NOT NULL	Puntos por unidad
cantidad	INTEGER	DEFAULT 1	Cantidad solicitada
total_item	DECIMAL(10,2)	NOT NULL	Total del ítem (precio × cantidad)
puntos_total	INTEGER	NOT NULL	Puntos totales del ítem
instrucciones_especiales	TEXT	NULLABLE	Instrucciones del cliente
fecha_agregado	TIMESTAMP	DEFAULT NOW()	Fecha de agregado al pedido

**Tabla 29***Estructura de la tabla notificaciones enviadas al usuario*

Columna	Tipo de Dato	Condiciones	Detalle
id	SERIAL	LLAVE PRIMARIA	Identificador único
usuario_id	INTEGER	FK → usuarios	Usuario destinatario
personal_id	INTEGER	FK → personal	Empleado destinatario
pedido_id	INTEGER	FK → pedidos	Pedido relacionado
tipo	VARCHAR(50)	NOT NULL	Tipo de notificación
canal	VARCHAR(20)	NOT NULL	Canal de envío (telegram/email)
titulo	VARCHAR(255)	NULLABLE	Título de la notificación
mensaje	TEXT	NOT NULL	Contenido del mensaje
estado	VARCHAR(20)	DEFAULT 'pendiente'	Estado de envío
fecha_creacion	TIMESTAMP	DEFAULT NOW()	Fecha de creación
fecha_envio	TIMESTAMP	NULLABLE	Fecha de envío efectivo
fecha_entrega	TIMESTAMP	NULLABLE	Fecha de entrega confirmada
mensaje_error	TEXT	NULLABLE	Error si falló el envío
metadatos	TEXT	NULLABLE	Datos adicionales en JSON

*Nota.* Registro de notificaciones enviadas a usuarios y personal del sistema.

**Tabla 30***Estructura de la tabla sesiones activas del bot de Telegram*

Columna	Tipo de Dato	Condiciones	Detalle
id	SERIAL	LLAVE PRIMARIA	Identificador único de sesión
chat_id	VARCHAR(50)	UNIQUE NOT NULL	ID del chat de Telegram
telegram_usuario_id	VARCHAR(50)	NULLABLE	ID del usuario en Telegram

username	VARCHAR(100)	NULLABLE	Nombre de usuario Telegram
nombre	VARCHAR(100)	NULLABLE	Nombre del usuario
personal_id	INTEGER	FK → personal	Empleado vinculado
autenticado	BOOLEAN	DEFAULT false	Estado de autenticación
codigo_vinculacion	VARCHAR(20)	NULLABLE	Código para vincular cuenta
codigo_vinculacion_expira	TIMESTAMP	NULLABLE	Expiración del código
ultimo_comando	VARCHAR(100)	NULLABLE	Último comando ejecutado

*Nota.* Gestión de sesiones activas del bot de Telegram para personal

## Capítulo IV

### 4.1 Pruebas y resultado

Con el objetivo de verificar el correcto funcionamiento y el rendimiento del sistema bajo diferentes perspectivas adaptadas al público de un bar, es necesario evaluar cómo se comporta ante un volumen elevado de peticiones simultáneas las condiciones más realistas, por esta razón se tomó en cuenta tres tipos de pruebas para este proyecto:

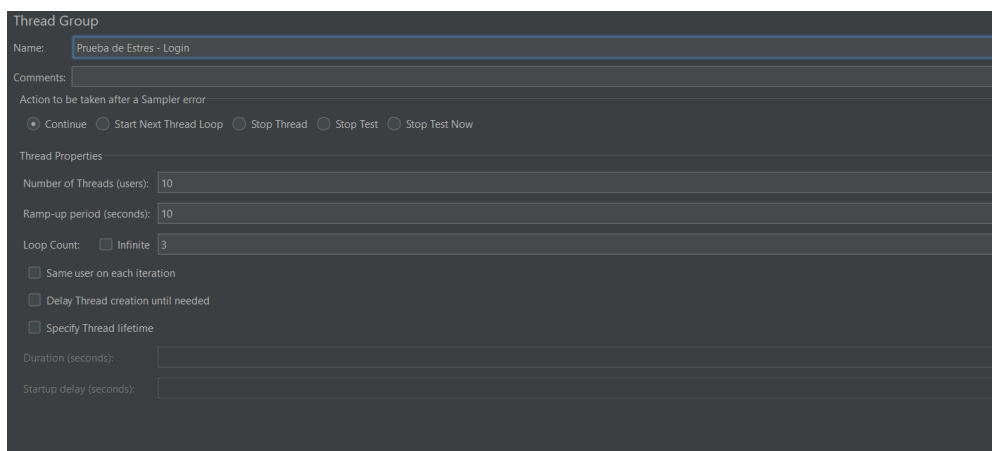
- Pruebas de Estrés
- Pruebas de Funcionalidad
- Pruebas de Concurrencia

Por lo que, se tomó a Apache Jmeter como herramienta para ejecutar operaciones al sistema sobre el backend desplegado en Railway.

La prueba de Estrés se llevó a cabo verificando que los endpoints esenciales del sistema respondan correctamente ante las peticiones que realizara un usuario.

#### Figura 35

*Configuración del grupo de hilos para la prueba de estrés*

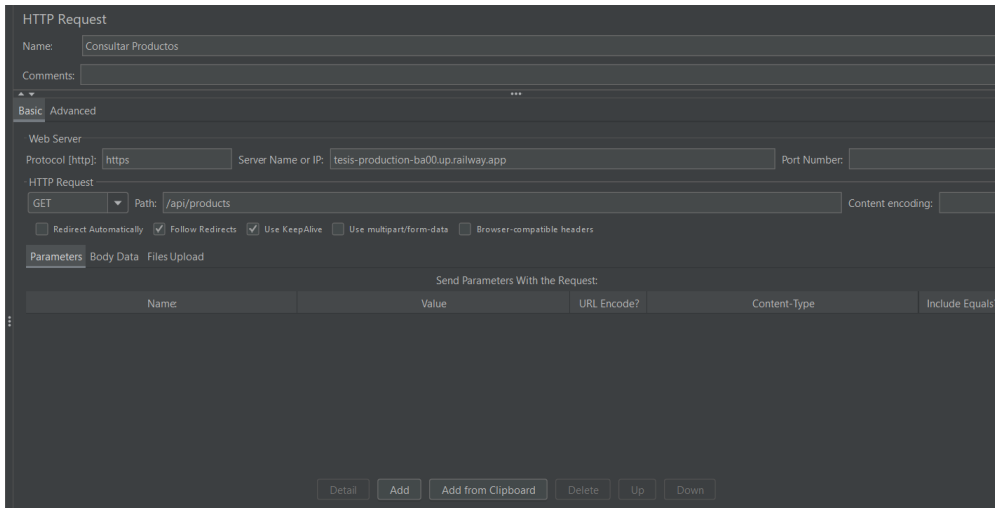




valida que el flujo de inicio de sesión, que involucra la verificación de credenciales, el cifrado mediante bcrypt y la generación del token JWT, se ejecuta en tiempos considerables de respuesta siendo el tiempo promedio de 2,794 ms con un mínimo de 976 ms, cuando múltiples usuarios acceden de forma simultánea.

**Figura 38**

*Configuración para api a productos*



**Figura 39**

*Respuesta de solicitudes a api*

Summary Report

Name:

Comments:

Write results to file / Read from file

Filename:   Log/Display Only:  Errors  Successes

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
Consultar Produ...	30	1203	228	5027	913.84	6.67%	2.5/sec	188.90	0.34	78752.1
TOTAL	30	1203	228	5027	913.84	6.67%	2.5/sec	188.90	0.34	78752.1

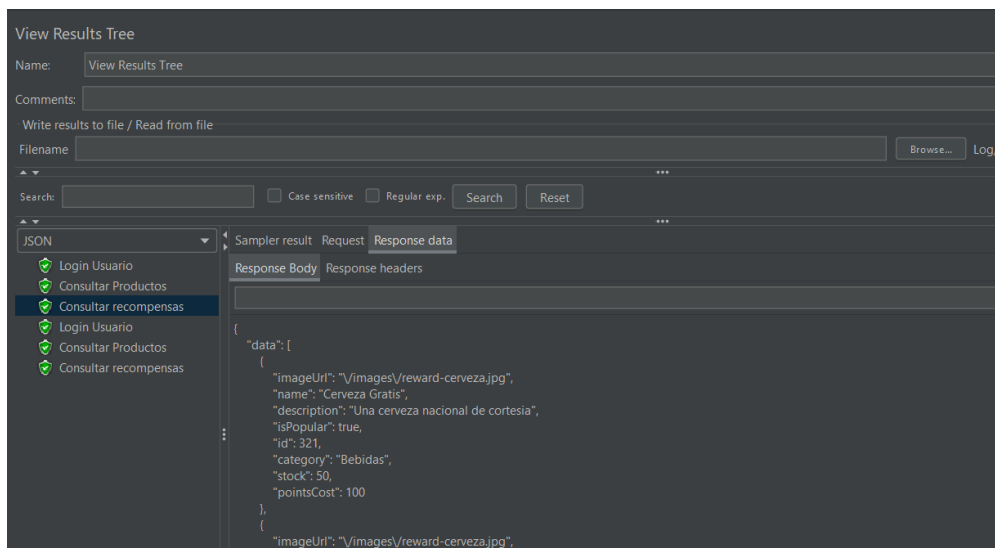
Include group name in label?  Save Table Data  Save Table Header

La siguiente prueba de estrés sobre el endpoint de productos se ejecutó con la misma configuración de la figura 26, en este caso con un tiempo promedio de 1,203 ms y un mínimo de 228 ms, lo que desde la perspectiva del usuario se traduce en una carga del catálogo de productos en aproximadamente 1 segundo, una experiencia fluida al navegar el menú de cualquiera de las marcas. El volumen promedio de respuesta fue de 78.7 KB, lo cual refleja la carga completa del catálogo con imágenes y descripciones.

Avanzando en el tema vamos a realizar las pruebas de funcionalidad al aplicar la secuencia que un usuario dentro del sistema: autenticación, consulta del catálogo de productos y consulta de recompensas disponibles.

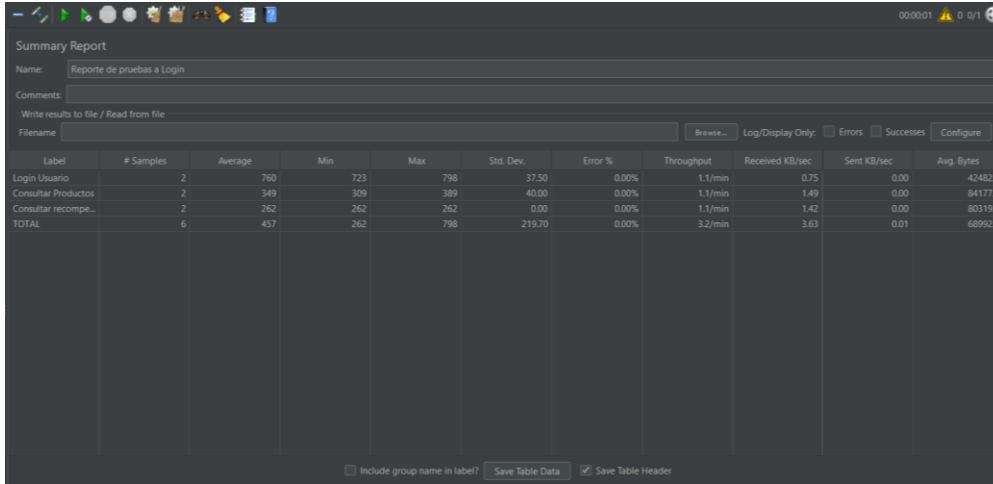
## Figura 40

### *Resultados de pruebas de funcionalidad*



## Figura 41

### Resultados de pruebas de funcionalidad



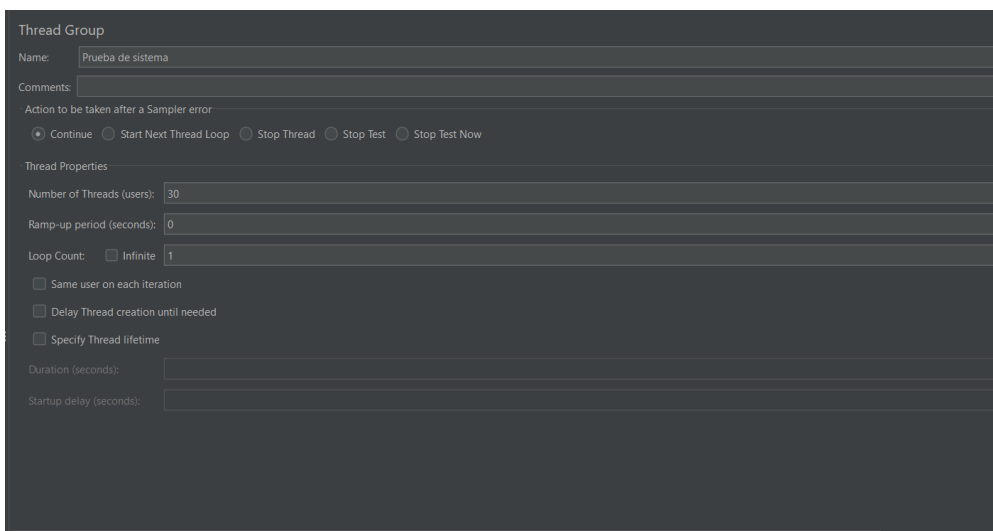
Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
Login Usuario	2	760	723	796	37.50	0.00%	1.1/min	0.75	0.00	42482.0
Consultar Productos	2	349	309	389	40.00	0.00%	1.1/min	1.49	0.00	84177.6
Consultar recompe...	2	262	262	262	0.00	0.00%	1.1/min	1.42	0.00	80319.6
TOTAL	6	457	262	796	219.70	0.00%	3.2/min	3.63	0.01	68992.7

Los tres endpoints respondieron exitosamente con 0.00% de errores según la figura 32, confirmando que la comunicación entre el frontend, el backend y la base de datos PostgreSQL opera de forma íntegra.

Finalmente, las pruebas de concurrencia analizando un escenario donde todos los usuarios llegan al mismo tiempo se configuro a 30 usuarios:

## Figura 42

### Configuración de prueba de concurrencia



Thread Group

Name: Prueba de sistema

Comments:

Action to be taken after a Sampler error

Continue  Start Next Thread Loop  Stop Thread  Stop Test  Stop Test Now

Thread Properties

Number of Threads (users): 30

Ramp-up period (seconds): 0

Loop Count:  Infinite 1

Same user on each iteration

Delay Thread creation until needed

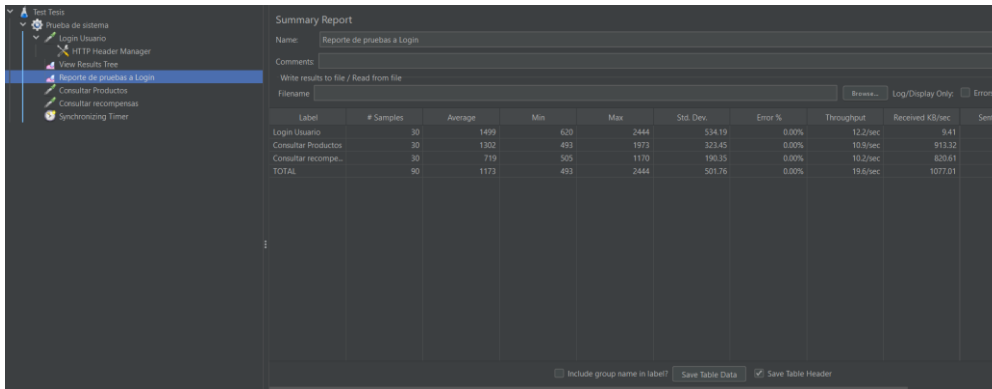
Specify Thread lifetime

Duration (seconds):

Startup delay (seconds):

**Figura 43**

*Resultados de prueba de concurrencia*



Los resultados son favorables al no haber caídas o pérdidas en ningún caso por lo que la experiencia basada en el usuario al usar el sistema está garantizada.

**4.2 Tablas de iteración de cada prueba con sus versiones**

**Tabla 31**

*Iteraciones del Proyecto por Sprint*

Sprint	Versión	Modulo	Funcionalidades	Estado
Sprint 1	0.1.0	Arquitectura Base	Configuración del proyecto, Base de datos PostgreSQL, Estructura de carpetas, Autenticación (Login/Registro)	Completado
Sprint 2	0.2.0	Catálogo de Productos	CRUD de productos, Categorías, Gestión de imágenes, Listado y filtros	Completado
Sprint 3	0.3.0	Sistema de Recompensas	Recompensas, Acumulación de puntos, Movimientos de puntos, Niveles de membresía	Completado
Sprint 4	0.4.0	Gestión de Pedidos	Carrito de compras, Creación de pedidos, Generación de código QR, Estados del pedido	Completado
Sprint 5	0.5.0	Integración Telegram	Bot de Telegram, Notificaciones en tiempo real, Vinculación de	Completado

			cuentas, Gestión de personal y turnos	
Sprint 6	1.0.0	Administración y Entrega	Panel de administración, Validación de canjes, Recuperación de contraseña, Pruebas de integración	Completado

*Nota.* El versionado utiliza el estándar Semantic Versioning

## Conclusiones

El proyecto se encuentra consolidado para fomentar los servicios y productos de Bounty Bar desde un panel administrativo el cual sirve como medio centralizado las operaciones de personal, añadiendo la configuración para beneficios dentro del establecimiento.

A medida que, el cliente va consumiendo puede ver la publicidad ofrecida por el establecimiento dentro del sistema, por lo que, se puede gestionar eventos, promociones o exponer beneficios que se tendrá a dentro de fechas específicas.

La implementación del módulo de perfil permite que cada usuario consulte de manera autónoma su información relevante, como el historial de compras, los puntos acumulados y su nivel de membresía actual. Esto contribuye directamente a la experiencia de usuario, ya que reduce la dependencia de consultas al personal del establecimiento y brinda transparencia sobre los beneficios obtenidos tras cada compra.

Finalmente, la proyección de este proyecto demuestra que la digitalización de los procesos operativos del establecimiento no solo optimiza la gestión actual, sino que sienta las bases para un crecimiento escalable, permitiendo integrar nuevas herramientas y funcionalidades conforme el negocio lo requiera.

## **Recomendaciones**

Al tener el proyecto desplegado, se debe analizar ampliar el alcance del servicio de correo electrónico, el cual actualmente se limita a la recuperación de contraseñas. Integrar notificaciones por email para confirmaciones de pedidos, resúmenes de puntos acumulados o alertas al alcanzar un nuevo nivel de membresía permitiría mantener informado al cliente a través de un canal adicional a Telegram, aumentando el alcance de comunicación del sistema.

Añadiendo que, se podría evaluar la integración a futuro de una pasarela de pagos como PayPhone , compatibles dentro del mercado ecuatoriano, por lo que esta incorporación permitiría cerrar el ciclo completo de la transacción desde la aplicación, agilizando la operación del establecimiento.

Por último, el sistema tiene limitaciones claras con respecto a no ser nativo con el celular, al contrario, el sistema es basado en web y se debe abrir la posibilidad de tener un desarrollo destinado a obtener los mejores beneficios de una versión nativa móvil para una mejor experiencia del usuario.

## Bibliografía

- Das, S. (22 de agosto de 2024). *Web Application Development: Process, Tools, & Examples*. BrowserStack. <https://www.browserstack.com/guide/web-application-development-guide>
- UNIR México. (17 de enero de 2025). *Metodología scrum para la gestión de proyectos tecnológicos*. <https://mexico.unir.net/noticias/ingenieria/metodologia-scrum/>
- Linares Sánchez, V. (2004). *Análisis de sistemas de navegación de sitios Web* [Trabajo de fin de carrera, Universitat Oberta de Catalunya]. O2 Repositori UOC. <https://openaccess.uoc.edu/handle/10609/146553>
- Algo Coding Experts. (2026). *¿Qué es el despliegue en la nube?* <https://algotestingexperts.com/es/que-es-el-despliegue-en-la-nube/>
- Rueda, S. (2015). Clase 9: Documentos de Especificación [Diapositivas de presentación]. Universidad Nacional del Sur. <https://cs.uns.edu.ar/materias/rs/downloads/CLASES%20TEORICAS/Clase%209%20%20Documentos%20de%20Especificacion%20RdS%202015.pdf>
- Humble, J., & Farley, D. (2010). *Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation*. Addison-Wesley Professional.
- IBM. (s.f.). *What Is Structured Query Language (SQL)?* <https://www.ibm.com/think/topics/structured-query-language>