

ESCUELA MILITAR DE INGENIERÍA  
"MCAL. ANTONIO JOSÉ DE SUCRE"  
BOLIVIA

## PROYECTO FINAL



*"Mcal. Antonio José de Sucre"*  
Prestigio, Disciplina y Mejores Oportunidades

### SISTEMA WEB DE CAFETERÍA "ENTRE NOSTROS"

CRESPO ARRIARAN NATALY NICOLE  
CRUZ SERRANO SHARAI GABRIELA  
GUTIÉRREZ LOZANO ELVIN ANDRES  
SOSSA CHUGAR THIAGO LEONARDO

COCHABAMBA 2025

# ÍNDICE DE CONTENIDO



## ÍNDICE

CONTENIDO	PÁGINAS
1. INTRODUCCIÓN.....	1
2. ANTECEDENTES .....	1
3. PLANTEAMIENTO DEL PROBLEMA .....	2
4. OBJETIVOS .....	2
5. OBJETIVOS GENERAL .....	2
6. OBJETIVOS ESPECIFICOS .....	3
7. JUSTIFICACIÓN .....	3
7.1. JUSTIFICACIÓN TÉCNICA.....	3
7.2. JUSTIFICACIÓN OPERATIVA.....	3
7.3. JUSTIFICACIÓN ECONÓMICA .....	4
8. ALCANCE FUNCIONAL.....	4
9. MARCO TEÓRICO.....	5
9.1. METODOLOGÍAS AGILES parA EL DESARROLLO DE SOFTWARE .....	5
9.1.1. SCRUM .....	6
9.2. FRAMEWORK.....	6
9.2.1. Django .....	7
9.3. BASE DE DATOS.....	7
9.3.1. MariaBD .....	7
10. MARCO PRÁCTICO.....	8
10.1. MODELADO DE NEGOCIO ACTUAL .....	8
10.2. MODELADO DE NEGOCIO ALTERNATIVO .....	9
10.3. ANÁLISIS DE REQUERIMIENTOS.....	10
10.3.1. Requerimientos funcionales .....	10
10.3.2. Requerimientos no funcionales .....	12
10.4. PLANIFICACIÓN DE ITERACIONES.....	12
10.5. 1RA SPRINT - “CLIENTES Y AUTENTICACIÓN” .....	13
10.5.1. Cronograma .....	13

10.5.2.	Diagrama de Casos de Uso del sistema .....	15
10.5.3.	Diagrama de Casos de Uso por actor .....	16
10.5.4.	Diagrama de Colaboración .....	16
10.5.5.	Diagrama de Arquitectura.....	17
10.5.6.	Diagrama de Clases .....	19
10.5.7.	Diagrama de Tablas Relacionales.....	20
10.5.8.	Diccionario de Datos .....	20
10.6.	2DO SPRINT - “ADMINISTRADOR” .....	21
10.6.1.	Cronograma .....	21
10.6.2.	Diagrama de Casos de Uso del sistema .....	24
10.6.3.	Diagrama de Casos de Uso por actor .....	25
10.6.4.	Diagrama de Colaboración .....	25
10.6.5.	Diagrama de Arquitectura.....	26
10.6.6.	Diagrama de Clases .....	28
10.6.7.	Diagrama de Tablas Relacionales.....	29
10.6.8.	Diccionario de Datos .....	29
10.7.	3RA SPRINT - “PEDIDOS, VENTAS Y CAJERO” .....	31
10.7.1.	Cronograma .....	31
10.7.2.	Diagrama de Casos de Uso del sistema .....	33
10.7.3.	Diagrama de Casos de Uso por actor .....	34
10.7.4.	Diagrama de Colaboración .....	35
10.7.5.	Diagrama de Arquitectura.....	35
10.7.6.	Diagrama de Clases .....	37
10.7.7.	Diagrama de Tablas Relacionales.....	37

10.7.8.	Diccionario de Datos .....	38
11.	CONCLUSIONES Y RECOMENDACIONES. ....	41
11.1.	CONCLUSIONES.....	41
11.2.	RECOMENDACIONES. ....	42
12.	Bibliografía .....	43

## CUERPO DEL PROYECTO



## **1. INTRODUCCIÓN**

La cafetería “ENTRE NOSOTROS”, ubicada en la ciudad de Cochabamba, es un emprendimiento local que ofrece un espacio acogedor y cálido donde las personas pueden compartir momentos especiales. Su propuesta combina productos de calidad con una atención cercana y personalizada, buscando crear un ambiente de conexión y bienestar.

Entre los principales servicios que brinda se encuentran: desayunos variados, jugos frutales naturales, una selecta línea de panadería y pastelería artesanal, así como café en grano preparado en casa, cuidadosamente seleccionado para garantizar frescura y sabor. Recientemente, la cafetería ha comenzado a ofrecer ventas online, permitiendo a los clientes realizar pedidos para recoger o recibir a domicilio.

La empresa se guía por una filosofía centrada en la calidad del café, el trato humano y la experiencia del cliente. Su misión se enfoca en ofrecer productos elaborados con dedicación, destacando ingredientes locales y frescos, mientras que su visión apunta a consolidarse como un referente cultural y social en la ciudad, promoviendo relaciones auténticas y un ambiente inspirador.

A pesar de su crecimiento y consolidación en el mercado local, la cafetería aún no cuenta con un sistema informático que respalde su operación diaria. Esta carencia limita su capacidad para optimizar procesos, expandir su alcance digital y sostener un crecimiento ordenado.

## **2. ANTECEDENTES**

La cafetería "ENTRE NOSOTROS" ha operado desde sus inicios con métodos manuales para la gestión de pedidos, inventario y ventas. A medida que el negocio creció, se implementaron herramientas básicas como hojas de cálculo y registros en papel, lo que resultó en una gestión ineficiente, falta de trazabilidad en los procesos y dificultades para escalar operaciones. La incorporación reciente del servicio de ventas en línea evidenció aún más la necesidad de un sistema integrado que permitiera

manejar tanto las operaciones presenciales como digitales de manera centralizada, optimizando tiempos, reduciendo errores humanos y mejorando la experiencia del cliente.

### **3. PLANTEAMIENTO DEL PROBLEMA**

La cafetería “ENTRE NOSOTROS” enfrenta actualmente un problema estructural, la ausencia de un sistema digital para gestionar sus operaciones diarias. Este problema se manifiesta en varios niveles. A nivel operativo, se observa un manejo ineficiente del inventario, retrasos en los procesos internos, y dificultades para generar reportes de ventas o estadísticas relevantes. A nivel comercial, la imposibilidad de ofrecer pedidos anticipados, mostrar un catálogo actualizado o gestionar promociones reduce la competitividad y afecta la percepción del cliente.

Los pedidos se toman de manera manual, lo cual genera errores en la preparación o cobros incorrectos. Además, el seguimiento del inventario es ineficiente, lo que provoca faltantes o excesos de productos. La administración de ventas y reportes también se realiza de forma desorganizada, lo que complica la toma de decisiones.

Por otro lado, los clientes no cuentan con una plataforma para revisar el menú o realizar pedidos de forma anticipada, lo cual limita la experiencia de usuario y reduce el potencial de crecimiento del negocio.

### **4. OBJETIVOS**

#### **5. OBJETIVOS GENERAL**

Diseñar e implementar un sistema web integral para la cafetería "ENTRE NOSOTROS" que centralice la gestión de pedidos, inventario, ventas y clientes, optimizando los procesos operativos y mejorando la experiencia del usuario tanto presencial como en línea.



## **6. OBJETIVOS ESPECIFICOS**

- Desarrollar un módulo de gestión de pedidos que permita registrar, actualizar y rastrear órdenes en tiempo real, tanto para servicio presencial como en línea.
- Implementar un sistema de control de inventario que actualice automáticamente los niveles de stock y genere alertas de reposición.
- Diseñar un panel administrativo para la gestión de empleados, roles, promociones y reportes de ventas.
- Crear una interfaz de usuario intuitiva y responsive que permita a los clientes visualizar el menú, realizar pedidos y gestionar su perfil.

## **7. JUSTIFICACIÓN**

### **7.1. JUSTIFICACIÓN TÉCNICA**

La implementación del sistema web se justifica desde una perspectiva técnica por la necesidad de establecer una plataforma digital robusta y moderna que reemplace la gestión manual actual. Se utilizará un sistema de gestión de bases de datos centralizado para garantizar la integridad, consistencia y disponibilidad de toda la información transaccional. Además, se priorizará un diseño responsive que permita el acceso y la funcionalidad completa del sistema desde cualquier dispositivo, asegurando una experiencia de usuario óptima tanto para el personal interno como para los clientes que realicen pedidos en línea. Esta elección tecnológica permitirá que el sistema escale junto con el crecimiento de "ENTRE NOSOTROS" sin requerir una reescritura costosa.

### **7.2. JUSTIFICACIÓN OPERATIVA**

Operativamente, el sistema es indispensable para eliminar las ineficiencias, retrasos y errores generados por los métodos en papel y hojas de cálculo. La implementación optimizará el flujo de trabajo al automatizar el registro de pedidos en tiempo real, desde el servicio en mesa hasta la venta en línea, proporcionando una trazabilidad clara y reduciendo los errores de preparación y cobro. Un punto crucial es el control de

inventario, donde el sistema actualizará automáticamente los niveles de *stock* con cada venta, generando alertas de reposición que asegurarán la disponibilidad constante de ingredientes frescos y productos terminados, minimizando el desperdicio. Esta automatización liberará al personal de tareas administrativas repetitivas, permitiéndoles enfocarse en brindar la atención cercana y personalizada que distingue a la cafetería, elevando así la calidad general del servicio y la satisfacción del cliente.

### **7.3. JUSTIFICACIÓN ECONÓMICA**

Desde una perspectiva económica, la inversión en el sistema web se justifica por su capacidad para incrementar la rentabilidad y el potencial de crecimiento. En primer lugar, la reducción de errores operativos y de inventario se traducirá directamente en una disminución de las pérdidas por desperdicio de productos y por errores de facturación. En segundo lugar, al formalizar y mejorar el canal de ventas en línea, la cafetería podrá ampliar su alcance de mercado y capturar una mayor cuota de clientes que prefieren el servicio a domicilio o la recogida anticipada, lo que se reflejará en un aumento directo de los ingresos. Finalmente, el sistema proporcionará a la gerencia reportes y estadísticas de ventas precisas, permitiendo una toma de decisiones informada sobre promociones, compras y gestión de personal, asegurando que los recursos se utilicen de la manera más eficiente y rentable posible para sostener el crecimiento ordenado y la consolidación de "ENTRE NOSOTROS" en Cochabamba.

## **8. ALCANCE FUNCIONAL**

El alcance funcional del sistema web se centra en centralizar y automatizar los procesos clave del negocio a través de una plataforma integral. El sistema se dividirá en cuatro módulos principales. El Módulo de Gestión de Pedidos y Ventas permitirá registrar y rastrear órdenes de tres canales (local, *online* y telefónico) y manejar la facturación y el cobro de manera eficiente. El Módulo de Control de Inventario y Productos administrará el catálogo del menú y, de manera crucial, actualizará automáticamente los niveles de *stock* de insumos y productos terminados con cada venta, generando alertas de reposición para evitar desabastecimientos. El Módulo de

Administración y Reportes servirá como el *back-end* gerencial, permitiendo la gestión de personal, la configuración de promociones y, vitalmente, la generación de reportes detallados de ventas y desempeño para la toma de decisiones. Finalmente, la Interfaz de Cliente ofrecerá una experiencia *responsive* y amigable, permitiendo a los clientes visualizar el menú actualizado en tiempo real, realizar pedidos anticipados y gestionar su perfil de usuario. Es importante destacar que el alcance excluye integraciones complejas con sistemas contables externos o el desarrollo de una aplicación móvil nativa en esta fase inicial, manteniendo el foco en la optimización operativa web.

## **9. MARCO TEÓRICO**

### **9.1. METODOLOGÍAS ÁGILES PARA EL DESARROLLO DE SOFTWARE**

En la década de los noventa surgieron metodologías de desarrollo de software ligeras, más adelante nombradas como metodologías ágiles, que buscaban reducir la probabilidad de fracaso por subestimación de costos, tiempos y funcionalidades en los proyectos de desarrollo de software. Estas metodologías nacieron como reacción a las metodologías existentes con el propósito de disminuir la burocracia que implica la aplicación de las metodologías tradicionales en los proyectos de pequeña y mediana escala. Las metodologías ágiles tienen dos diferencias fundamentales con las metodologías tradicionales; la primera es que los métodos ágiles son adaptativos –no predictivos-. La segunda diferencia es que las metodologías ágiles son orientadas a las personas –no orientadas a los procesos. Las metodologías ágiles son adaptativas. Este hecho es de gran importancia ya que contrasta con la predictibilidad buscada por las metodologías tradicionales. Con el enfoque de las metodologías ágiles los cambios son eventos esperados que generan valor para el cliente. Los proyectos ágiles se subdividen en proyectos más pequeños mediante una lista ordenada de características. Cada proyecto es tratado de manera independiente y desarrolla un subconjunto de características durante un periodo de tiempo corto, de entre dos y seis semanas. La comunicación con el cliente es constante al punto de requerir un representante de él durante el desarrollo. (Martínez, 2013)

### **9.1.1. SCRUM**

La metodología Scrum para el desarrollo ágil de software es un marco de trabajo diseñado para lograr la colaboración eficaz de equipos en proyectos, que emplea un conjunto de reglas y artefactos y define roles que generan la estructura necesaria para su correcto funcionamiento. Scrum utiliza un enfoque incremental que tiene como fundamento la teoría de control empírico de procesos. Esta teoría se fundamenta en transparencia, inspección y adaptación; la transparencia, que garantiza la visibilidad en el proceso de las cosas que pueden afectar el resultado; la inspección, que ayuda a detectar variaciones indeseables en el proceso; y la adaptación, que realiza los ajustes pertinentes para minimizar el impacto de estas.

Scrum define un evento principal o Sprint (figura 1) que corresponde a una ventana de tiempo donde se crea una versión utilizable del producto (incremento). Cada Sprint, como en el rugby, es considerado como un proyecto independiente. Su duración máxima es de un mes. Un Sprint se compone de los siguientes elementos: reunión de planeación del Sprint, Daily Scrum, trabajo de desarrollo, revisión del Sprint y retrospectiva del Sprint. (Martínez, 2013)

### **9.2. FRAMEWORK**

En informática y programación, un marco proporciona una estructura sobre la que se pueden crear nuevos programas y aplicaciones de software. Un marco puede referirse a un conjunto de funciones de un sistema y su interrelación; a las capas de un sistema operativo o un subsistema de aplicación; o a cómo debe estandarizarse la comunicación en algún nivel de una red. Los frameworks más fiables y populares incluyen diversas herramientas para el desarrollo, la prueba y la depuración de código. Muchos frameworks también proporcionan plantillas que pueden reutilizarse y modificarse según los requisitos de la aplicación. Estos elementos prediseñados y modificables permiten a los programadores crear nuevos programas sin tener que empezar desde cero. Con el framework como base, pueden añadir funcionalidades de

alto nivel para crear un producto de software de alta calidad con mayor rapidez y menos errores. (Raul Awati, Ben Lutkevich, 2024)

### **9.2.1. Django**

Django es un framework eficiente para la mayoría de las aplicaciones web y móviles , pero puede presentar dificultades al gestionar múltiples solicitudes simultáneas. Normalmente, los desarrolladores solucionan esta limitación utilizando servidores de aplicaciones como Gunicorn o uWSGI para ejecutar múltiples trabajadores que puedan analizar estas solicitudes. Además, al ser un framework monolítico, las diversas funciones de Django pueden resultar excesivas para proyectos pequeños. Los principiantes que implementan el framework podrían encontrar confuso el número de componentes disponibles. (Doyle, 2024)

## **9.3. BASE DE DATOS**

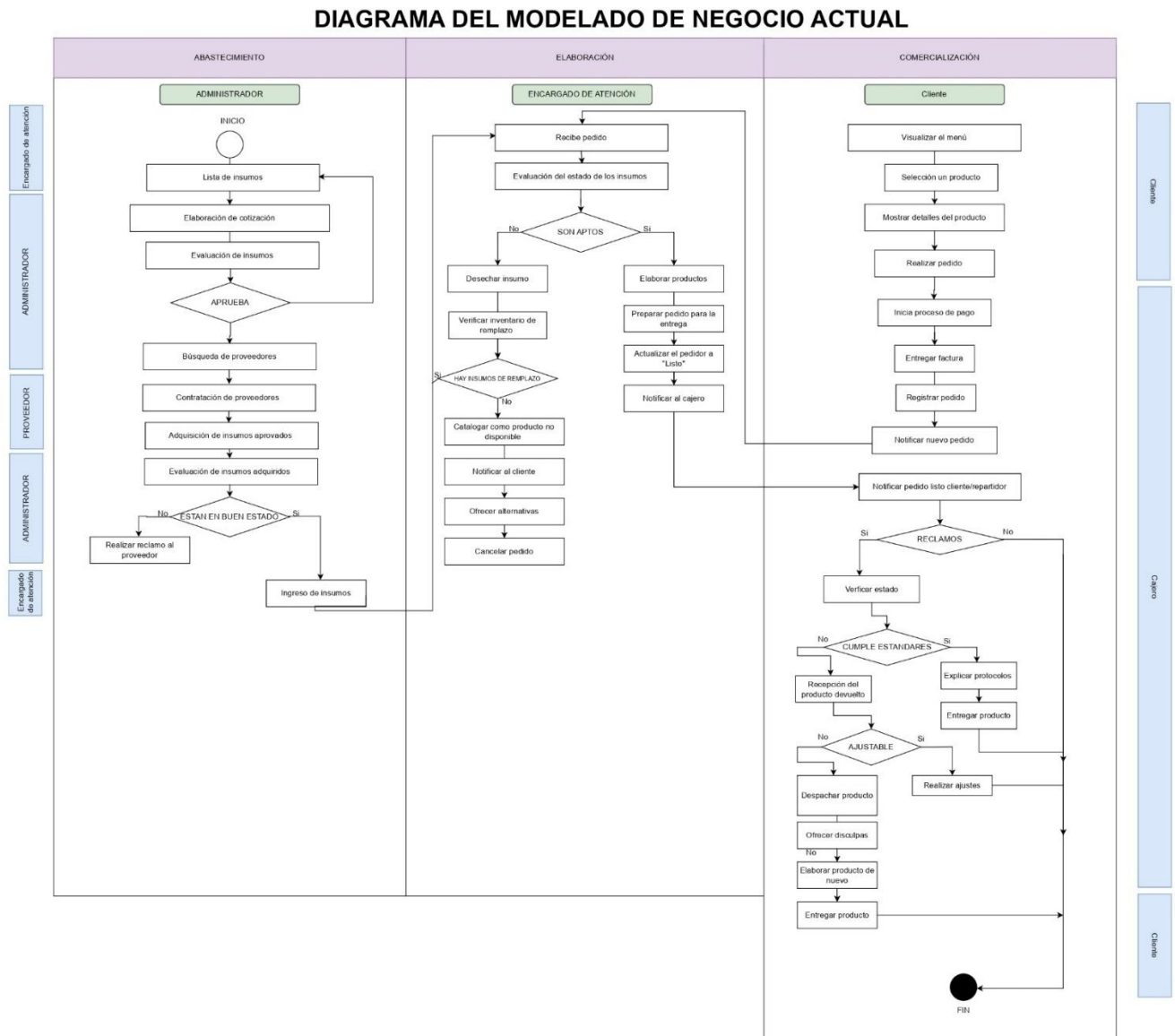
Según (Ballester, 2007) una base de datos es un conjunto de datos que pertenecen al mismo contexto, almacenados sistemáticamente para su posterior uso, es una colección de datos estructurados según un modelo que refleje las relaciones y restricciones existentes en el mundo real. Los datos que han de ser compartidos por diferentes usuarios y aplicaciones deben mantenerse independientes de éstas, y su definición y descripción han de ser únicas estando almacenadas junto a los mismos.

### **9.3.1. MariaBD**

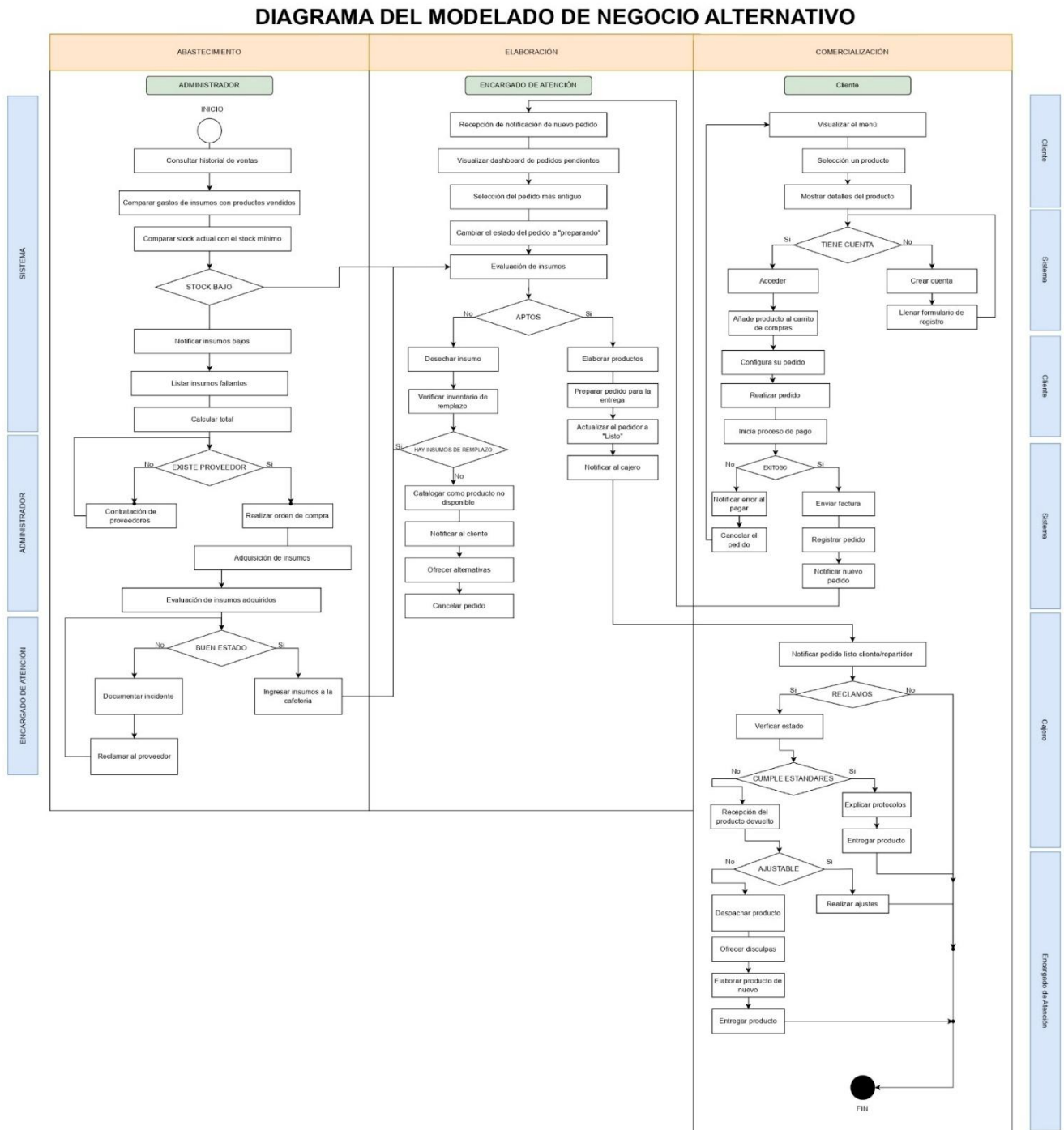
MariaDB Server es un sistema de gestión de bases de datos relacionales de código abierto. Es uno de los servidores de bases de datos más populares del mundo, con usuarios notables como Wikipedia, WordPress.com y Google. MariaDB Server se publica bajo la licencia de código abierto GPLv2 y se garantiza entonces que seguirá siendo abierto. Se puede utilizar para datos de transacciones de alta disponibilidad, análisis de datos, como servidor integrado, y una amplia gama de herramientas y aplicaciones soportan MariaDB Server. (MariaDB, 2009)

## 10. MARCO PRÁCTICO

### 10.1. MODELADO DE NEGOCIO ACTUAL



## 10.2. MODELADO DE NEGOCIO ALTERNATIVO



### 10.3. ANÁLISIS DE REQUERIMIENTOS

#### 10.3.1. Requerimientos funcionales

<b><u>REQUERIMIENTO FUNCIONAL</u></b>	<b><u>DESCRIPCION</u></b>
<b>RF01-CREAR CUENTA DE CLIENTE</b>	El cliente crea su propia cuenta en el sistema al acceder al sistema por primera vez.
<b>RF02-INICIAR SESION</b>	El usuario (cliente, cajero, administrador) usara sus credenciales personales ingresadas anteriormente para acceder al sistema.
<b>RF03-MOSTRAR EL CATALOGO DE PRODUCTOS</b>	El sistema desplegara el menú de la cafetería.
<b>RF04-MOSTRAR EL CATALOGO DE PROMOCIONES</b>	El sistema muestra un catálogo de las promociones vigentes en la cafetería.
<b>RF05 – AÑADIR PRODUCTOS AL CARRITO</b>	El cliente añade productos del catálogo al carrito de compras para después procesar la compra.
<b>RF06 – MODIFICAR DETALLES DE UN PEDIDO</b>	El cliente después de elegir los productos a consumir, puede que tome alguna decisión de ultimo momento y quiera cambiar algún pedido.
<b>RF07 – GENERAR LA LISTA DE PEDIDOS</b>	El sistema genera una lista donde nos muestra todos los pedidos que llevamos en el carrito.
<b>RF08 – CONFIRMAR PRODUCTOS</b>	El cliente y/o el cajero realizaran la confirmación del pedido el cliente vía el sistema y el cajero vía física en la cafetería.



<b>RF09- REALIZAR PAGO</b>	El cliente realiza el pago de su pedido por diferentes vías (QR, TARJETA, CASH).
<b>RF10- GENERAR FACTURA</b>	El sistema genera factura de la compra realizada por el cliente
<b>RF11-IMPRIMIR FACTURA</b>	El sistema da la opción de poder imprimir la factura al usuario que lo requiera.
<b>RF12-VER ESTADO DEL PEDIDO</b>	El sistema permite al cliente ver en todo momento el estado en el que se encuentra su pedido.
<b>RF13 – NOTIFICAR PEDIDO LISTO VIA WIDNESS PAGING SYSTEM</b>	El cajero le entrega al cliente una maquina de notificación por sonido y vibración (WIDNESS PAGING SYSTEM), para que el cliente recoja su pedido desde caja.
<b>RF014 - GENERAR REPORTES</b>	El sistema genera los reportes de ventas, insumos y productos de toda la cafetería

### 10.3.2. Requerimientos no funcionales

<b><u>REQUERIMIENTOS NO FUNCIONALES</u></b>	<b><u>DESCRIPCION</u></b>
<b>RNF1 - Gestión de roles y permisos</b>	El acceso a las funcionalidades del sistema debe estar limitado por el rol del usuario (cliente, cajero, administrador), mostrando únicamente las opciones permitidas para cada perfil.
<b>RNF2 - Confirmación de acciones sensibles</b>	El sistema debe solicitar confirmación antes de ejecutar acciones críticas como eliminar un pedido o cerrar sesión, para evitar errores involuntarios.
<b>RNF3 - Visualización adaptable</b>	La interfaz debe ajustarse correctamente a distintos tamaños de pantalla (PC, tablet y celular), conservando la legibilidad y funcionalidad de cada módulo.
<b>RNF4 - Exportación de reportes</b>	Los reportes generados deben poder exportarse en formatos comunes como PDF o Excel, manteniendo el formato legible y ordenado.
<b>RNF5 - Cierre automático de sesión por inactividad</b>	El sistema debe cerrar automáticamente la sesión del usuario tras 10 minutos de inactividad para proteger los datos y recursos del sistema.

### 10.4. PLANIFICACIÓN DE ITERACIONES

Para el desarrollo del sistema web integral, se adoptará la metodología ágil SCRUM, la cual ofrece la flexibilidad necesaria para adaptarse a los requerimientos cambiantes y garantiza la entrega incremental de valor funcional a la cafetería "ENTRE NOSOTROS". La implementación se estructurará mediante Sprints de duración definida (típicamente de dos a tres semanas), donde la priorización y división del trabajo se realizará a partir de las interfaces de usuario y los módulos funcionales clave. Este enfoque garantiza que cada ciclo de desarrollo se centre en entregar un componente usable para un rol específico, facilitando la validación temprana con los usuarios finales (personal y clientes).

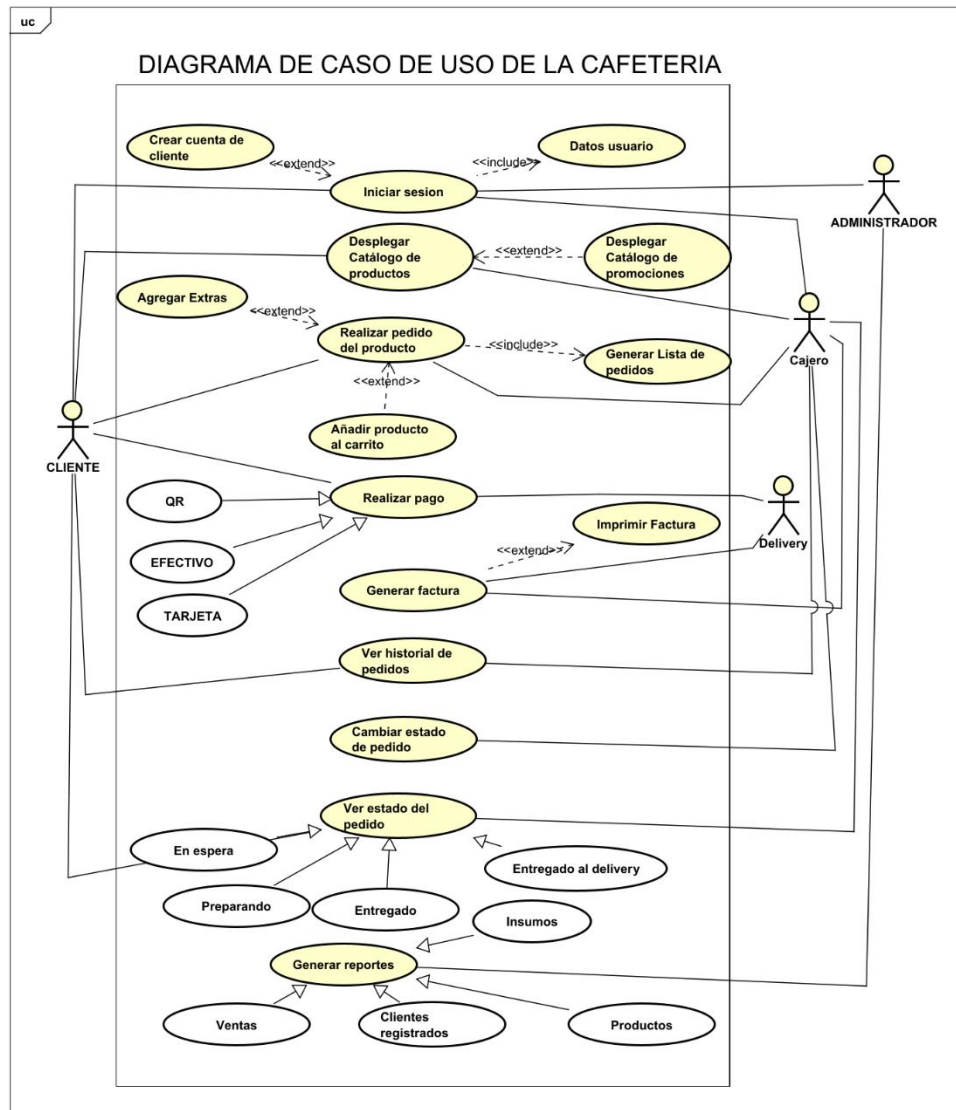
## 10.5. 1RA SPRINT - “CLIENTES Y AUTENTICACIÓN”

### 10.5.1. Cronograma

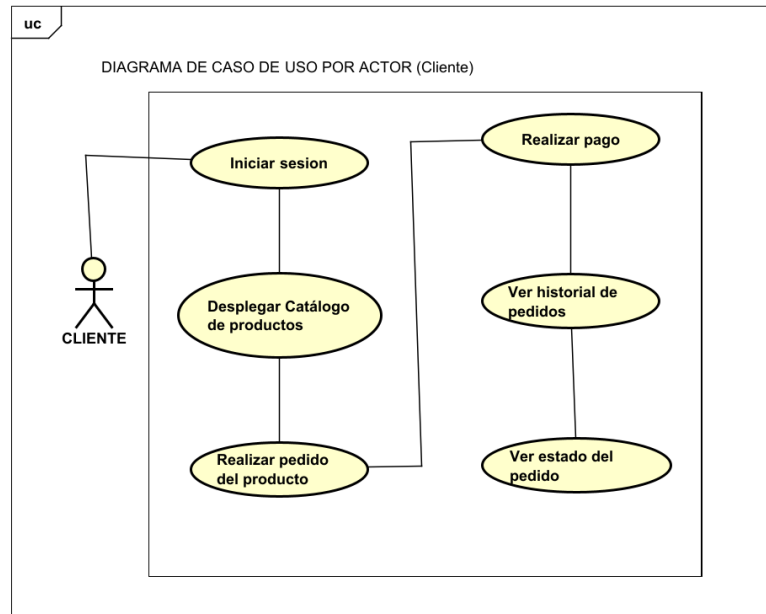
Fase SCRUM	Actividades / Tareas	Duración estimada	Responsables	Entregables / Resultados
<b>Product Backlog (Previo)</b>	Identificar requerimientos del módulo (campos, seguridad, flujo de usuario).	1 día	Product Owner + Equipo	Historias de usuario del cliente y autenticación.
<b>Sprint Planning</b>	Definir objetivos, tareas y criterios de aceptación.	1 día	Scrum Master + Dev Team	Sprint backlog con tareas detalladas.
<b>Desarrollo</b>	Diseño de interfaz “Login” y “Registro”.	2 días	Frontend Dev	Prototipos funcionales UI.
	Creación de API de autenticación.	3 días	Backend Dev	Endpoints: /login, /register, /logout.
	Integración con MariaDB (tabla usuarios).	1 día	DBA	Conexión estable y validada.

<b>Daily Scrum</b>	Reuniones de seguimiento.	Diario (15 min)	Todo el equipo	Identificación temprana de bloqueos.
<b>Pruebas (QA)</b>	Validar flujos de inicio y cierre de sesión.	1 día	QA Tester	Resultados de pruebas unitarias.
<b>Sprint Review</b>	Presentación del módulo funcional al Product Owner.	1 día	Todo el equipo	Retroalimentación y validación.
<b>Sprint Retrospective</b>	Evaluar proceso, tiempos y comunicación.	1 día	Scrum Master + Equipo	Registro de mejoras.
<b>Incremento</b>	Entrega funcional del módulo de autenticación.	Final de la semana 2	Equipo Dev	Incremento integrado al sistema.

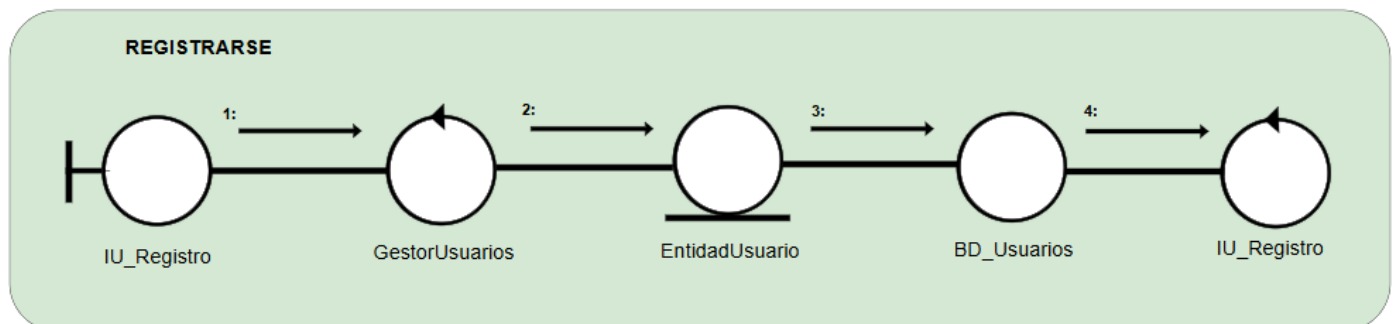
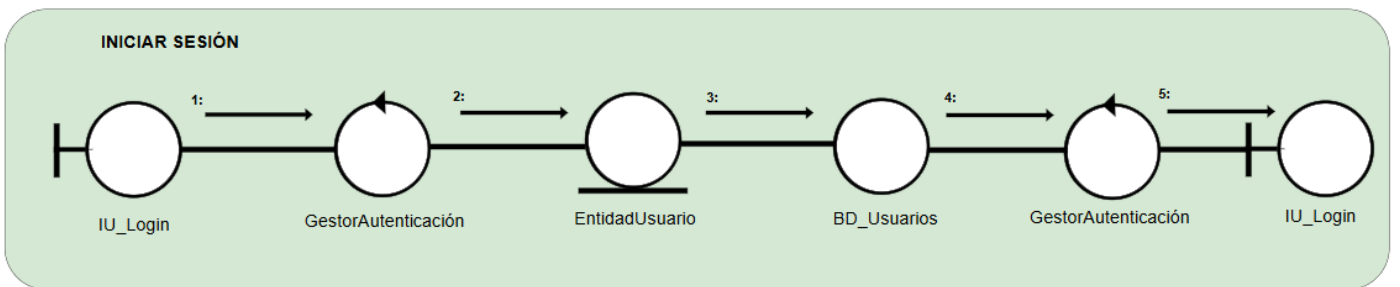
## 10.5.2. Diagrama de Casos de Uso del sistema

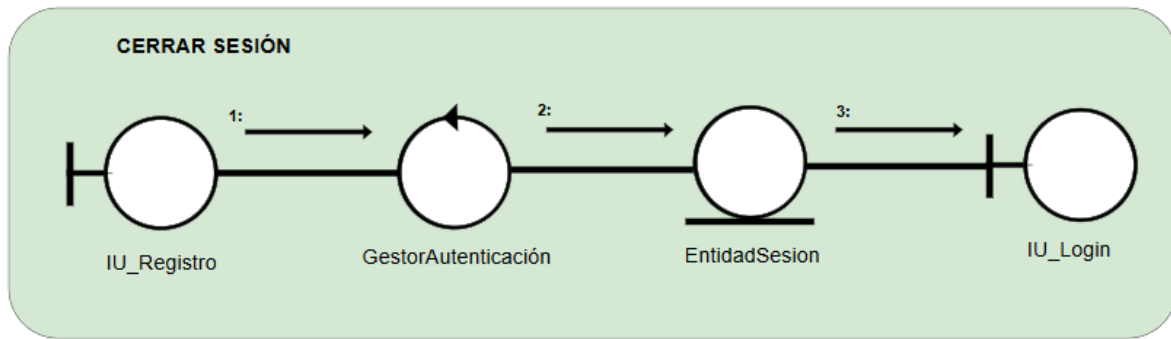


### 10.5.3. Diagrama de Casos de Uso por actor



### 10.5.4. Diagrama de Colaboración





### 10.5.5. Diagrama de Arquitectura

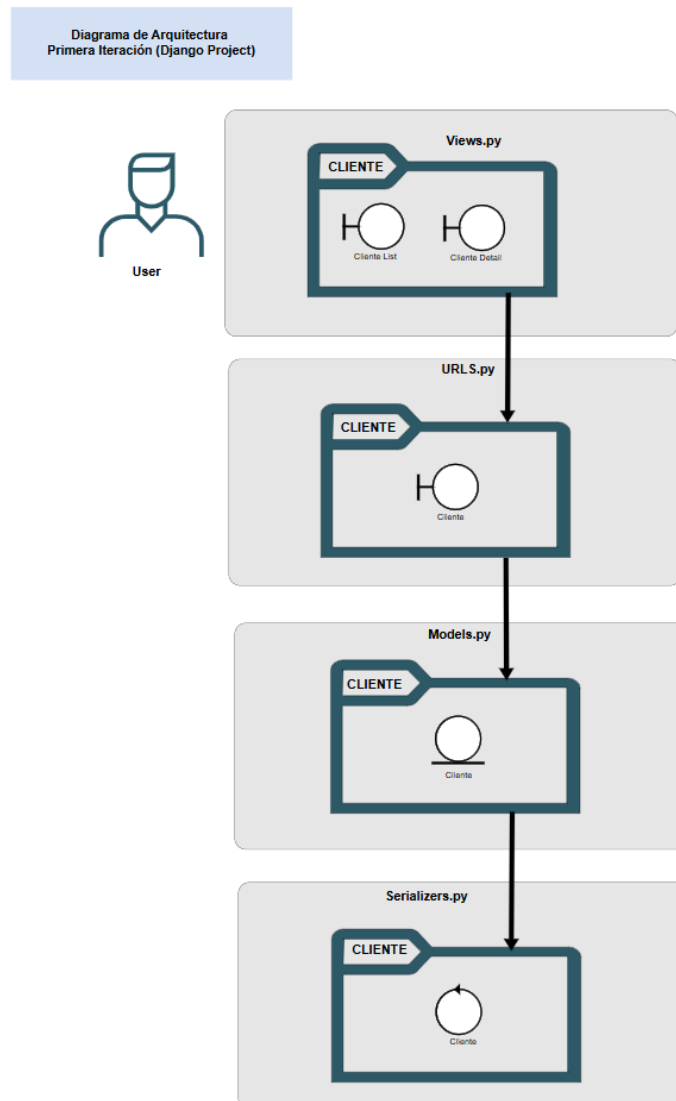
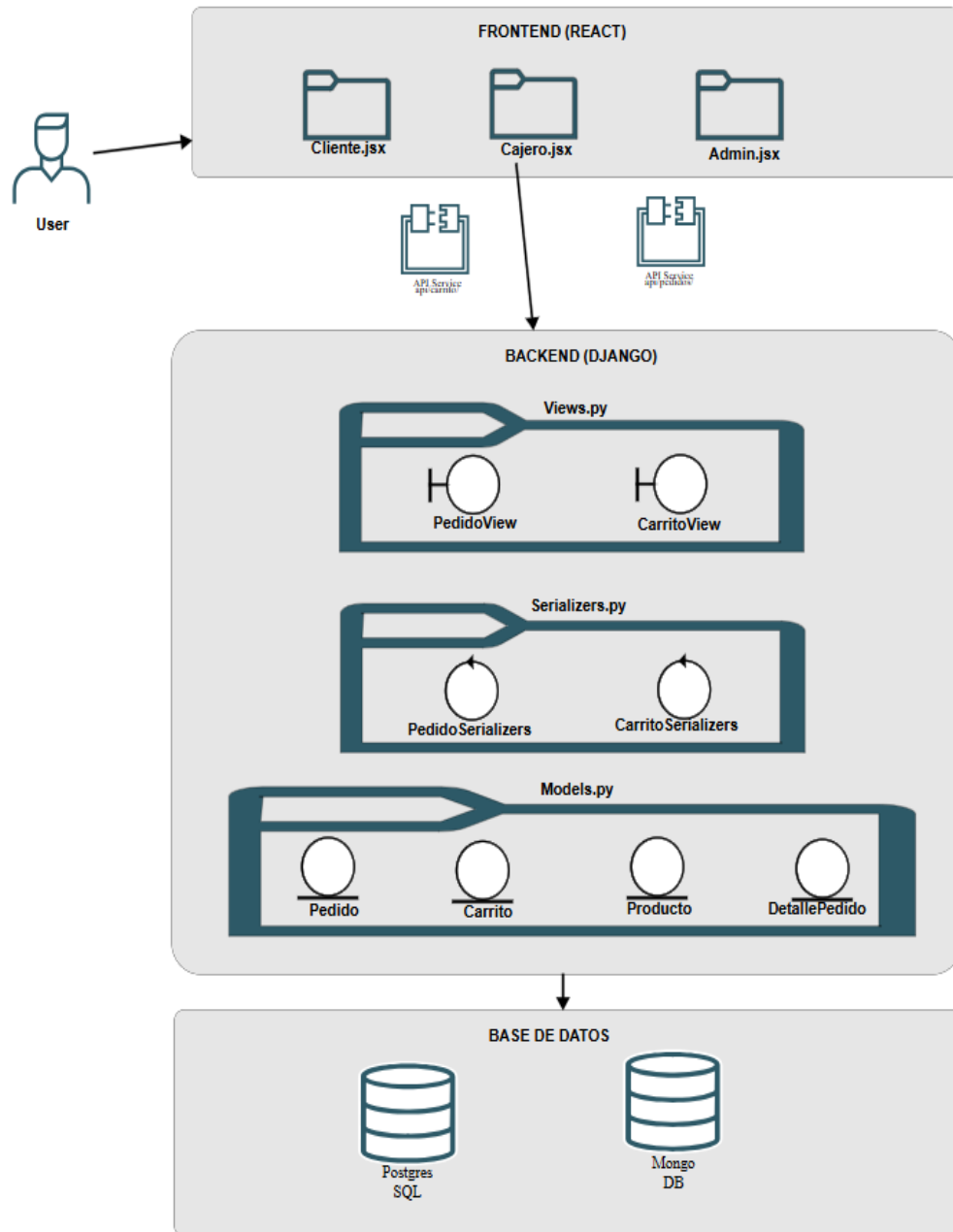
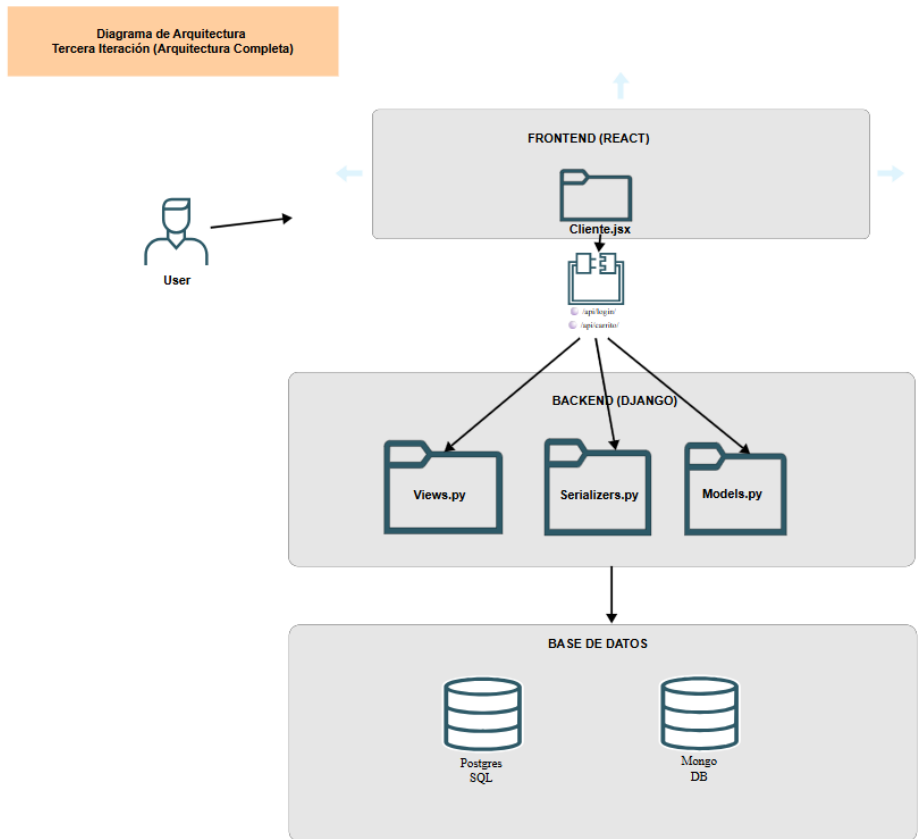


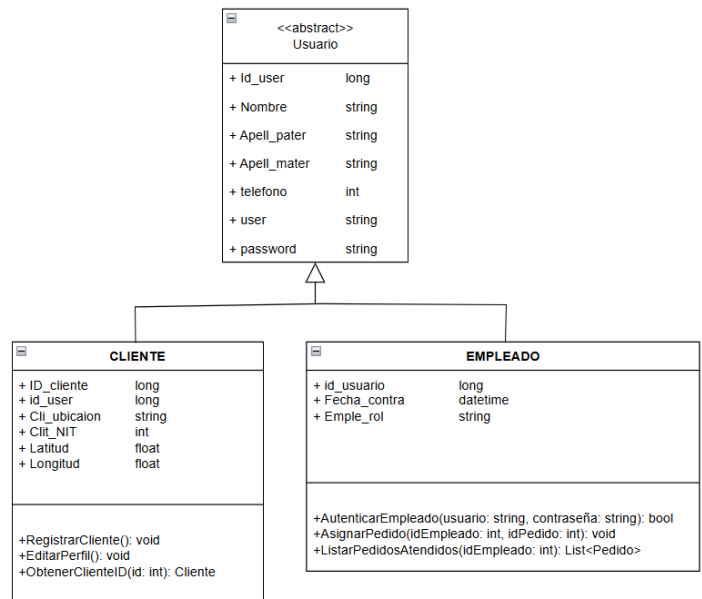
Diagrama de Arquitectura  
Segunda Iteración (Integración con react Frontend + Backend)







## 10.5.6. Diagrama de Clases



### 10.5.7. Diagrama de Tablas Relacionales

CLIENTE		
PK	ID_cliente	long
	Cli_ubicacion	string
	Cliit_NIT	int
	Latitud	float
	Longitud	float
	Nombre	string
	Apellido_pater	string
	Apellido_mater	string
	telefono	int
	user	string
	password	string

EMPLEADO		
PK	id_usuario	long
	Fecha_contra	datetime
	Emple_rol	string
	Nombre	string
	Apellido_pater	string
	Apellido_mater	string
	telefono	int
	user	string
	password	string

### 10.5.8. Diccionario de Datos

#### 10.5.8.1. Diccionario de datos de la tabla “CLIENTE”

CAMPO	DESCRIPCION	TIPO DE DATO	INDICACION	VALIDACION
Id_cliente	Clave del cliente	long	numero	No es nulo
Cli_ubicacion	Indica la ubicación del cliente	String	texto	No es nulo
Cli_nit	Indica el NIT del cliente	int	numero	No es nulo
Latitud	Indica la latitud de la ubicacion	float	numero	No es nulo
Longitud	Indica la longitud de la ubicacion	flota	numero	No es nulo

Nombre	Indica el nombre del cliente	string	texto	No es nulo
Apell_pater	Indica el apellido paterno del cliente	string	texto	No es nulo
Apell_mater	Indica el apellido materno del cliente	string	texto	No es nulo
Telefono	Indica el numero telefónico del cliente	int	numero	No es nulo
user	Indica el usuario del cliente	string	texto	No es nulo
password	Indica la contraseña del cliente	string	texto	No es nulo

## 10.6. 2DO SPRINT - “ADMINISTRADOR”

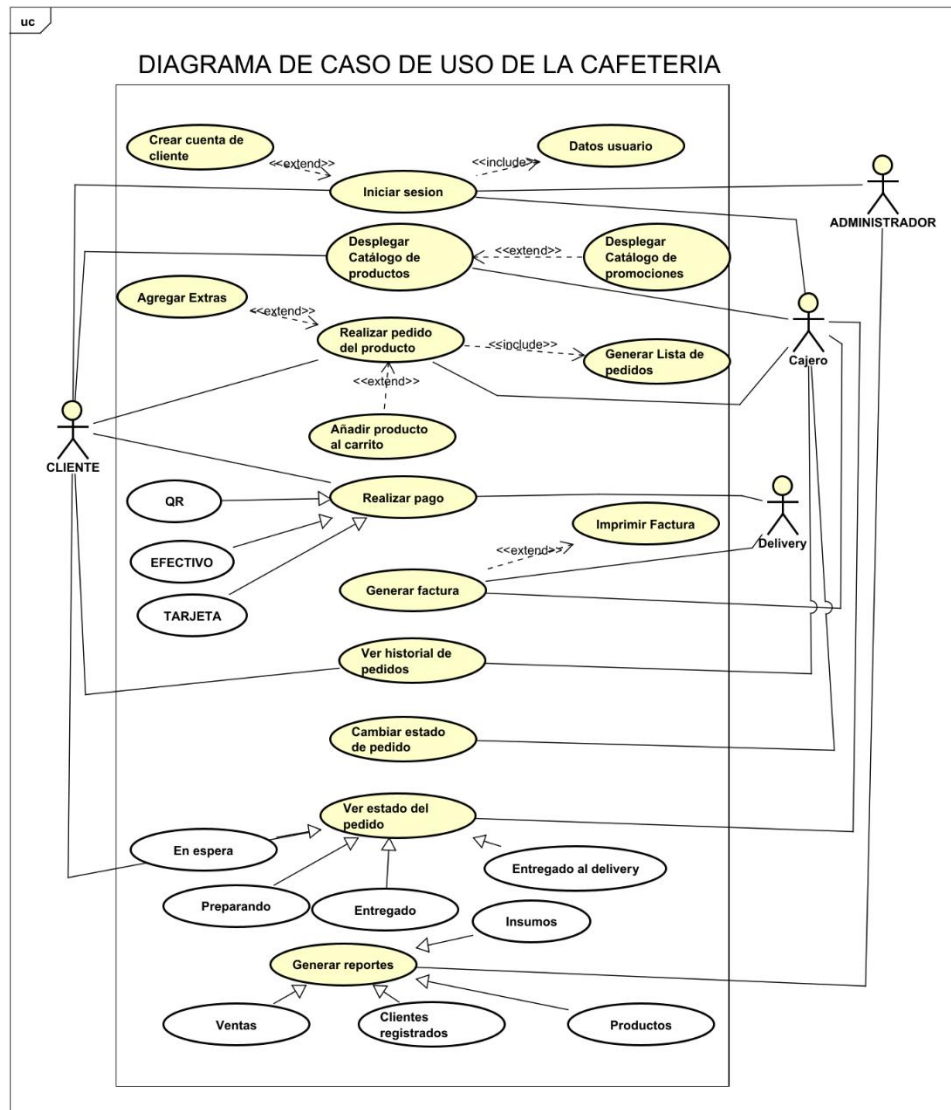
### 10.6.1. Cronograma

Fase SCRUM	Actividades / Tareas	Duración estimada	Responsables	Entregables / Resultados
<b>Product Backlog (Previo)</b>	Definir requerimientos del administrador (CRUDs, roles, vistas).	1 día	Product Owner + Equipo	Historias de usuario “Administrador”.
<b>Sprint Planning</b>	Planificar tareas,	1 día	Scrum Master + Equipo	Sprint backlog del módulo admin.

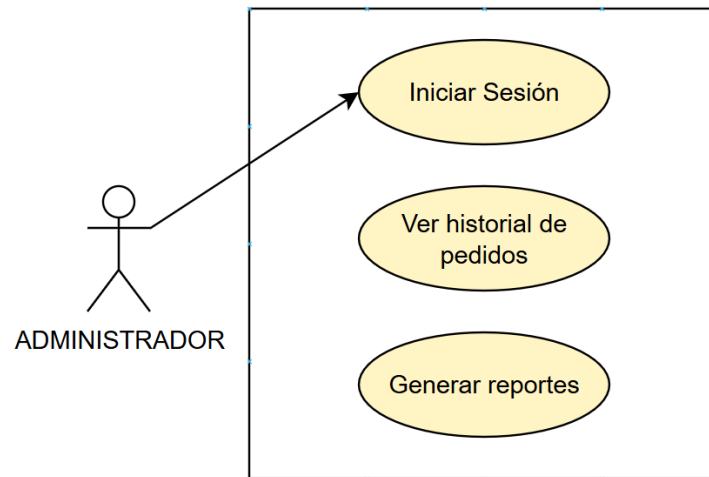
	dependencias y backlog.			
<b>Desarrollo</b>	Crear UI del panel administrativo.	3 días	Frontend Dev	Dashboard funcional con navegación.
	Implementar CRUD de productos (crear, editar, eliminar, listar).	3 días	Backend Dev	API y vistas integradas.
	Implementar CRUD de usuarios con roles.	3 días	Backend Dev + DBA	Control de roles y autenticación extendida.
	Módulo de reportes (ventas, productos, usuarios).	2 días	Backend Dev	Reportes PDF o tabla dinámica.
<b>Daily Scrum</b>	Seguimiento de avances y bloqueos.	Diario	Todo el equipo	Monitoreo continuo del progreso.
<b>Pruebas (QA)</b>	Validación de CRUDs y reportes.	2 días	QA Tester	Módulo validado funcionalmente.

<b>Sprint Review</b>	Presentar al Product Owner el panel administrativo.	1 día	Todo el equipo	Retroalimentación sobre reportes y CRUDs.
<b>Sprint Retrospective</b>	Evaluar desempeño del equipo, tiempo y complejidad.	1 día	Scrum Master	Plan de mejoras.
<b>Incremento</b>	Módulo administrativo funcional integrado.	Fin de la semana 3	Equipo Dev	Incremento al ERP.

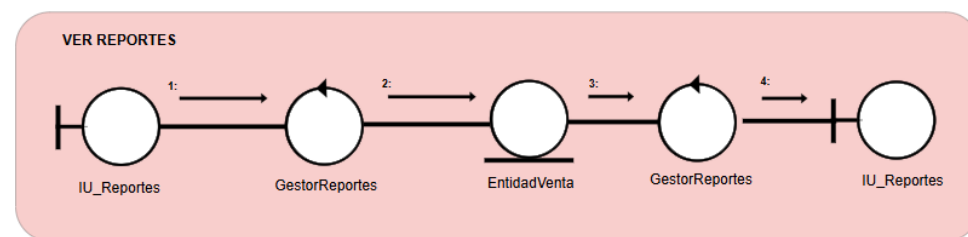
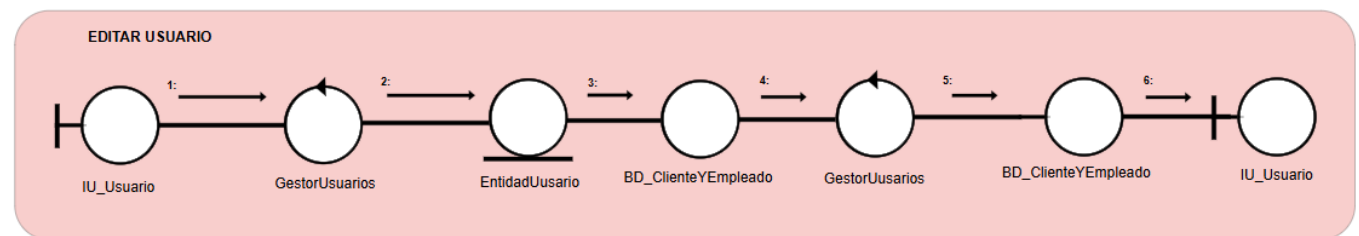
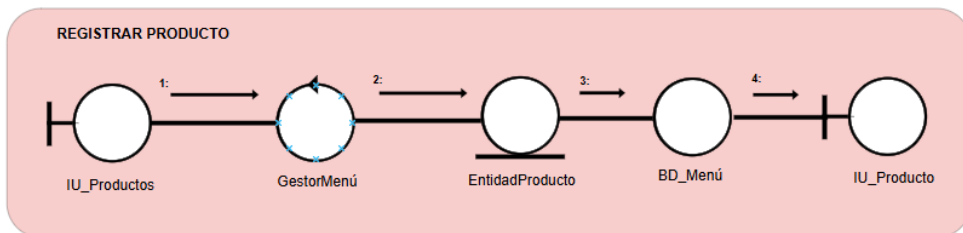
## 10.6.2. Diagrama de Casos de Uso del sistema



### 10.6.3. Diagrama de Casos de Uso por actor



### 10.6.4. Diagrama de Colaboración



### 10.6.5. Diagrama de Arquitectura

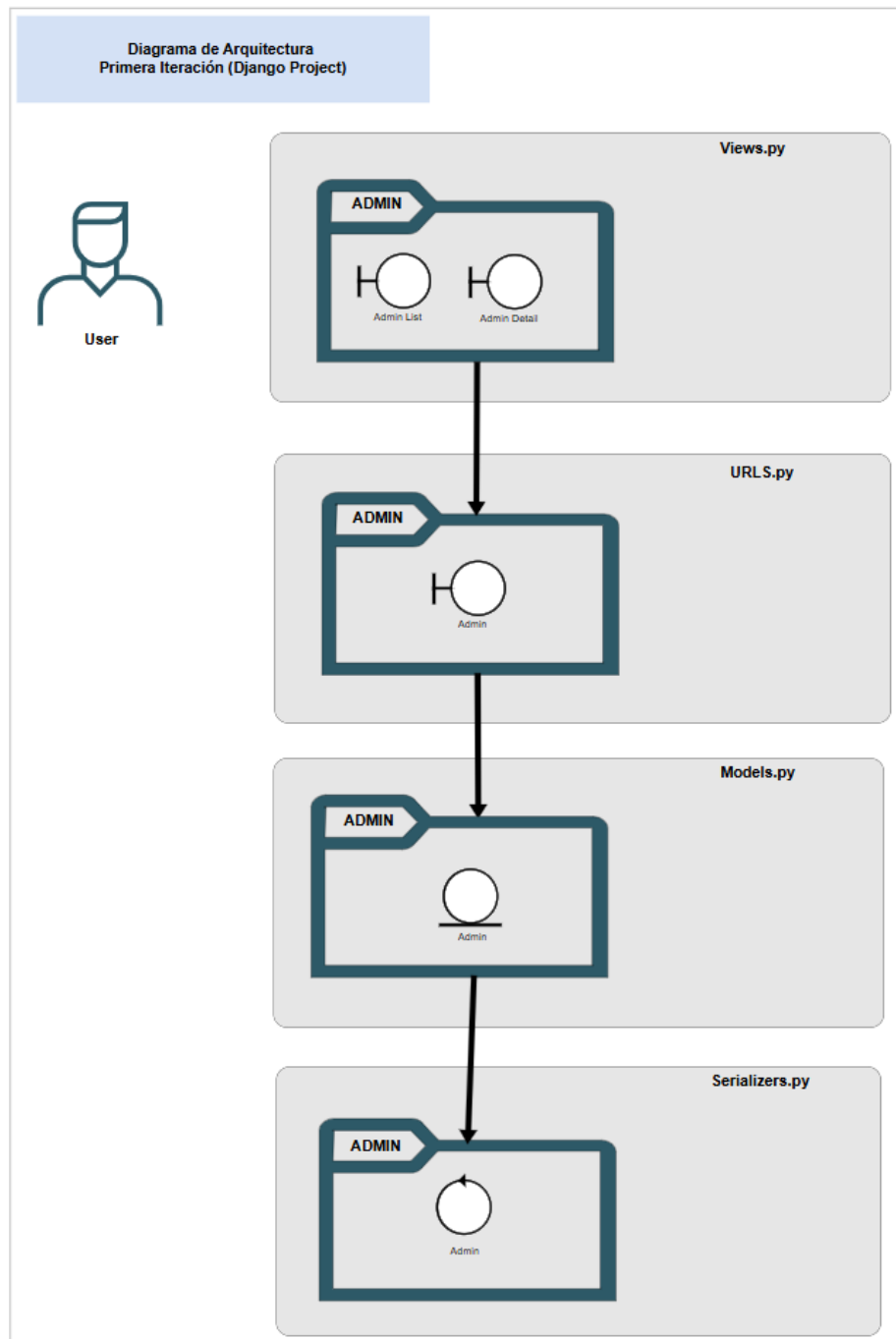
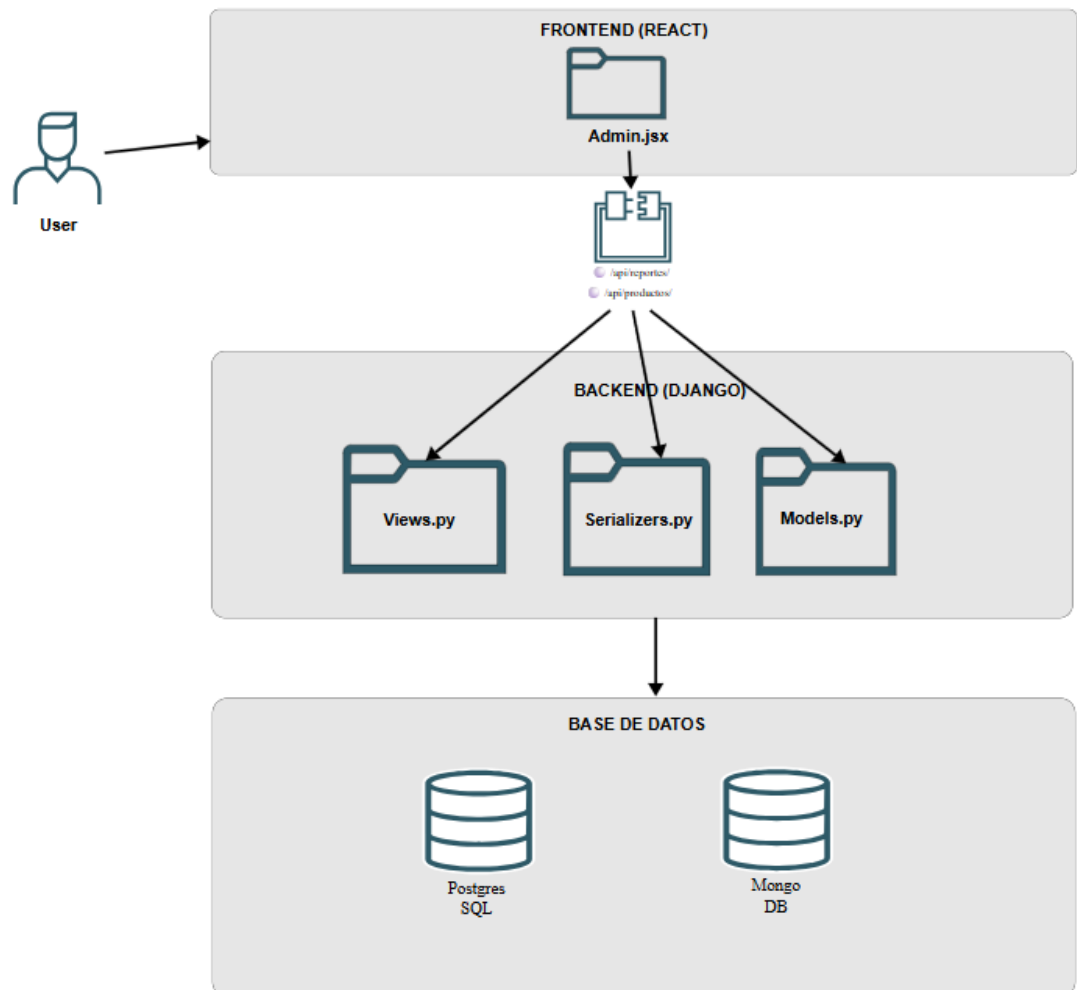
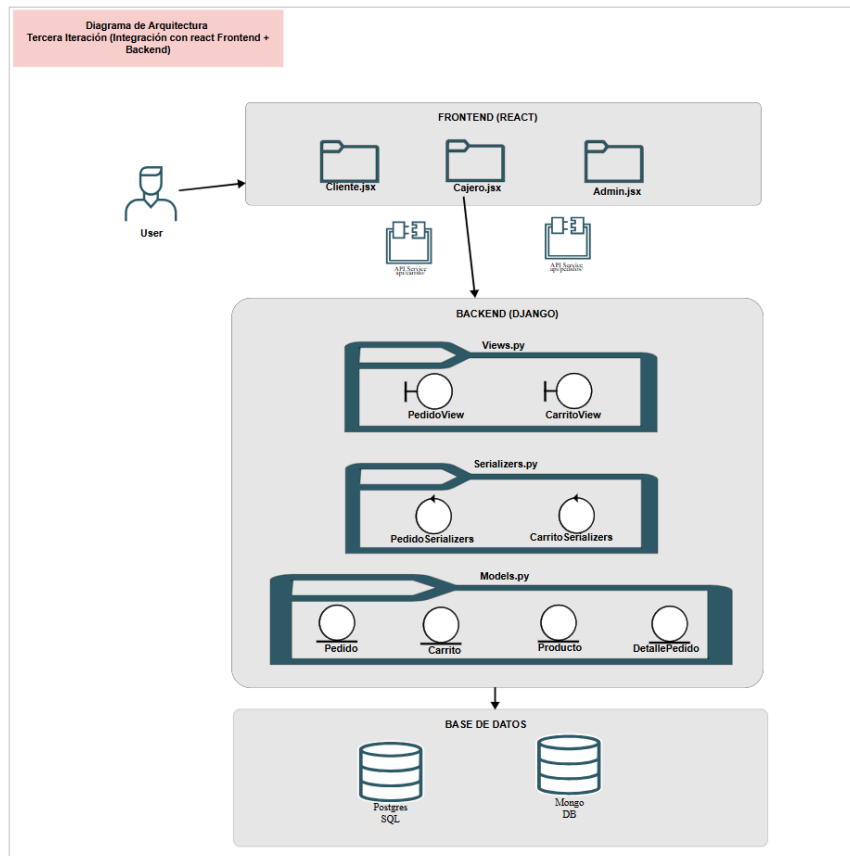


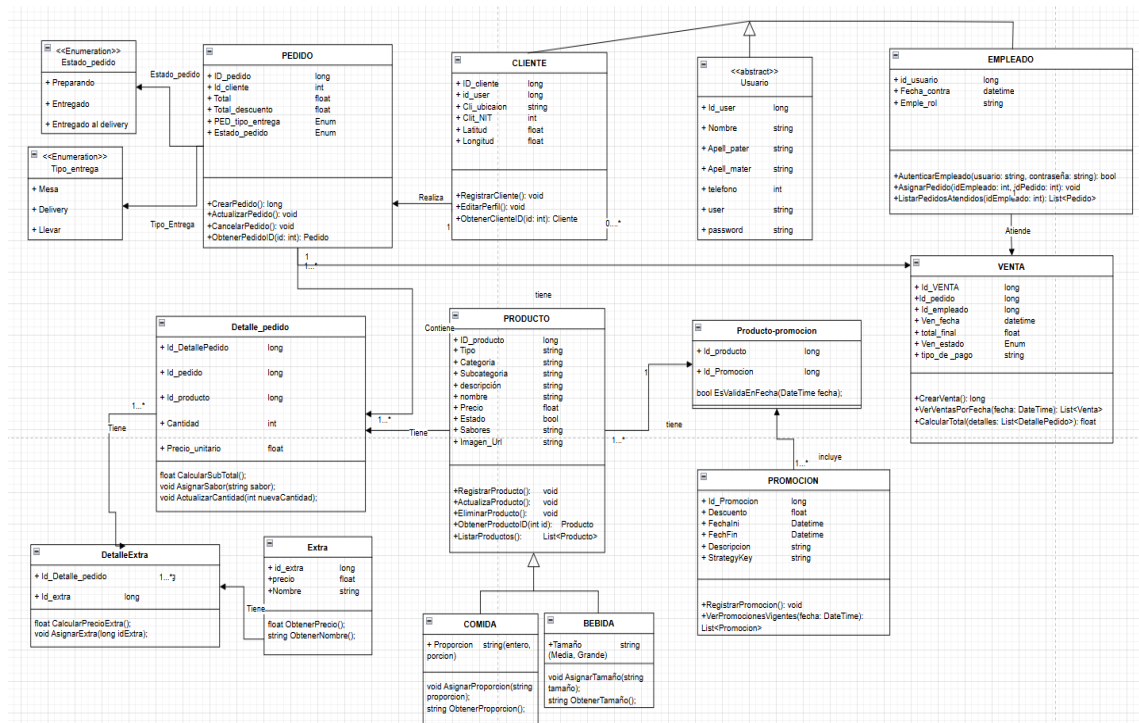


Diagrama de Arquitectura  
Cuarta Iteración (Arquitectura Completa)

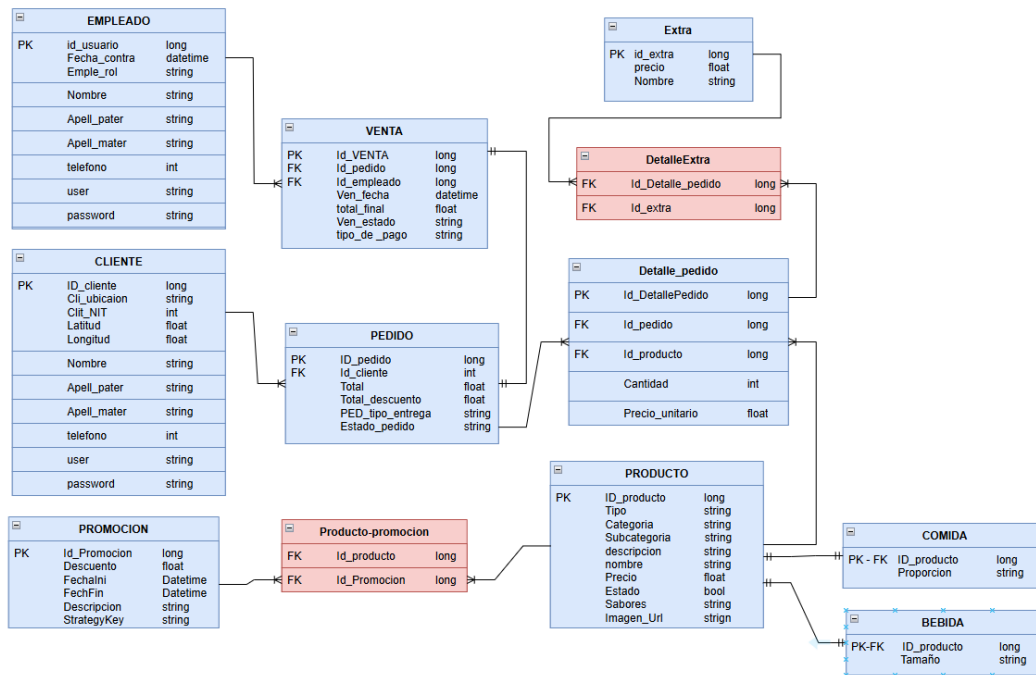




## 10.6.6. Diagrama de Clases



### 10.6.7. Diagrama de Tablas Relacionales



### 10.6.8. Diccionario de Datos

#### 10.6.8.1. Diccionario de datos de la tabla “PRODUCTO”

CAMPO	DESCRIPCION	TIPO DE DATO	INDICACION	VALIDACION
Id_producto	Clave del producto	long	numero	No es nulo
Tipo	Describe el tipo de producto	string	Texto	No es nulo
Categoria	Indica a que categoría pertenece el producto	String	texto	No es nulo
Subcategoria	Indica a que subcategoria pertenece	string	texto	No es nulo

descripcion	Describe el producto y que es lo que contiene	string	texto	No es nulo
nombre	Es el nombre del producto	string	texto	No es nulo
Precio	Indica el precio del producto	float	numero	No es nulo
Estado	Indica en que estado se encuentra	bool	True or false	No es nulo
Sabores	Indica los sabores del producto	string	texto	No es nulo
Imagen_url	Es la imagen demostrativa del producto	string	texto	No es nulo

#### 10.6.8.1.1. Diccionario de datos de la tabla “PROMOCION”

CAMPO	DESCRIPCION	TIPO DE DATO	INDICACION	VALIDACION
Id_Promocion	Clave de la promocion	long	numero	No es nulo
Descuento	Es el descuento que habrá en la promocion	float	numero	No es nulo
FechaIni	Indica cuando empieza la promocion	DateTime	fecha	No es nulo
FechaFin	Indica cuando termina la promocion	DateTime	fecha	No es nulo
Descripcion	Describe en que consiste la promocion	string	texto	No es nulo

#### 10.6.8.2. Diccionario de datos de la tabla “PRODUCTO-PROMOCION”

CAMPO	DESCRIPCION	TIPO DE DATO	INDICACION	VALIDACION
Id_producto	Clave foránea de la tabla producto	long	numero	No es nulo
Id_Promocion	Clave foránea de la tabla Promocion	long	numero	No es nulo

### 10.7. 3RA SPRINT - “PEDIDOS, VENTAS Y CAJERO”

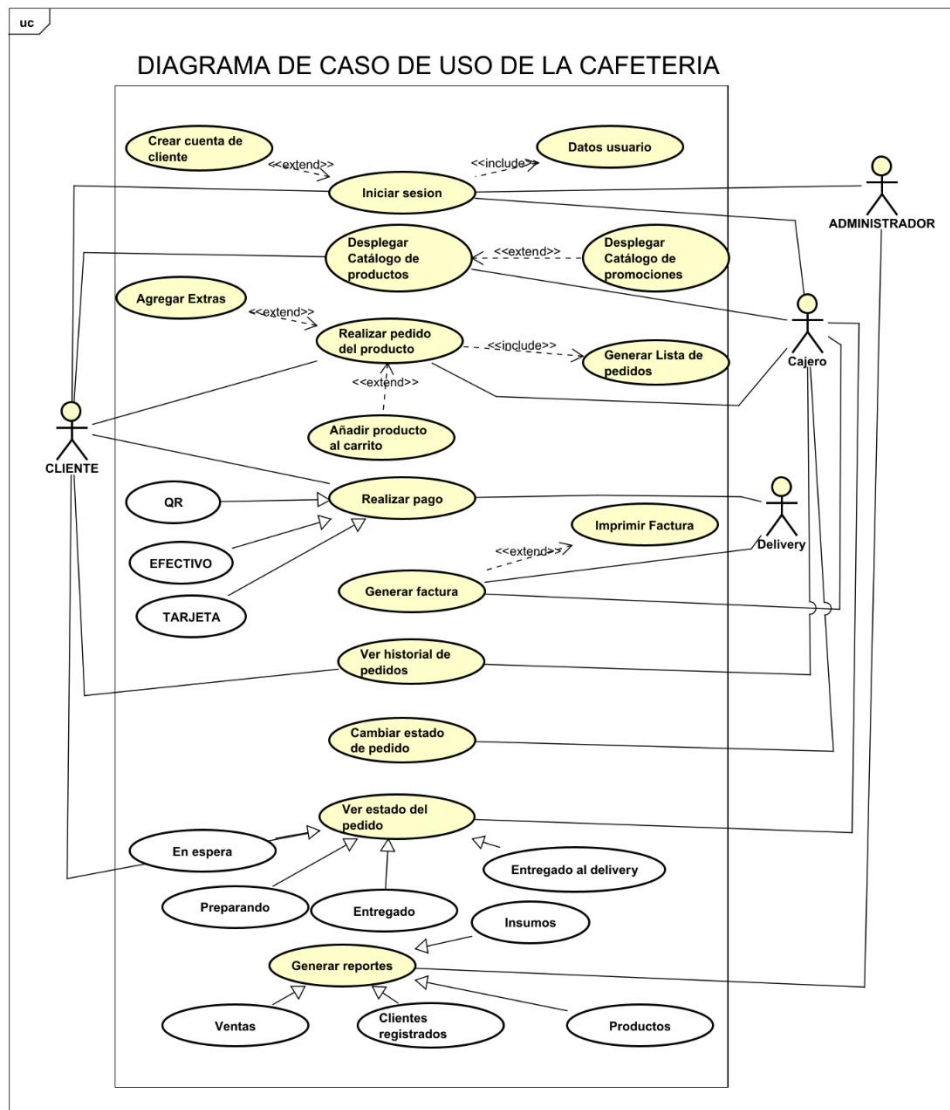
#### 10.7.1. Cronograma

Fase SCRUM	Actividades / Tareas	Duración estimada	Responsables	Entregables / Resultados
<b>Product Backlog (Previo)</b>	Definir requerimientos de ventas y stock.	1 día	Product Owner + Equipo	Historias de usuario de ventas y stock.
<b>Sprint Planning</b>	Definir tareas, roles y dependencias.	1 día	Scrum Master + Dev Team	Sprint backlog de ventas.
<b>Desarrollo</b>	Crear interfaz de “Punto de venta” (cajero).	3 días	Frontend Dev	UI para registro rápido de ventas.

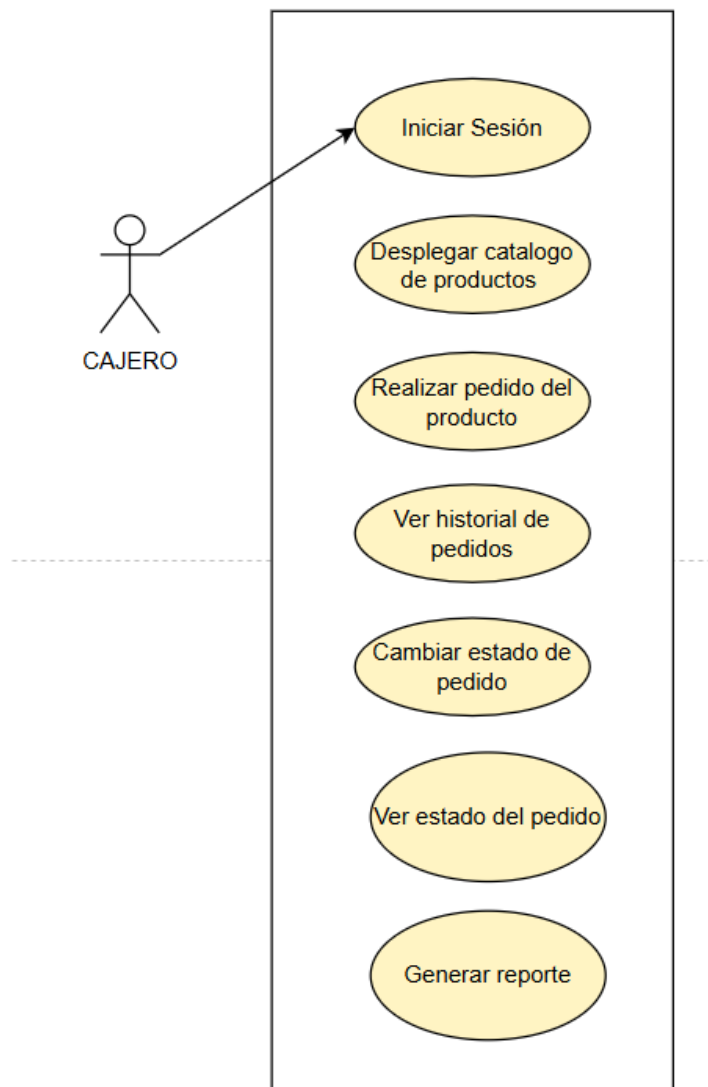
	Backend de registro de pedido y cobro.	4 días	Backend Dev	API funcional /pedido /venta.
	Actualización automática de stock tras la venta.	2 días	Backend Dev + DBA	Trigger o lógica en BD.
	Validación de flujos completos (cliente → venta → stock).	2 días	QA Tester	Flujo probado extremo a extremo.
<b>Daily Scrum</b>	Seguimiento diario.	Diario	Todo el equipo	Control de avances.
<b>Pruebas (QA)</b>	Validación de flujo POS, cobro y stock.	1 día	QA Tester	Módulo estable y funcional.
<b>Sprint Review</b>	Demostración del flujo completo al Product Owner.	1 día	Todo el equipo	Retroalimentación final.
<b>Sprint Retrospective</b>	Evaluación general del proyecto.	1 día	Scrum Master + Equipo	Documentación de aprendizajes.

Incremento	Sistema ERP completo con integración de módulos.	Fin de la semana 3	Equipo Dev	Entrega final e informe.
------------	--	--------------------	------------	--------------------------

## 10.7.2. Diagrama de Casos de Uso del sistema

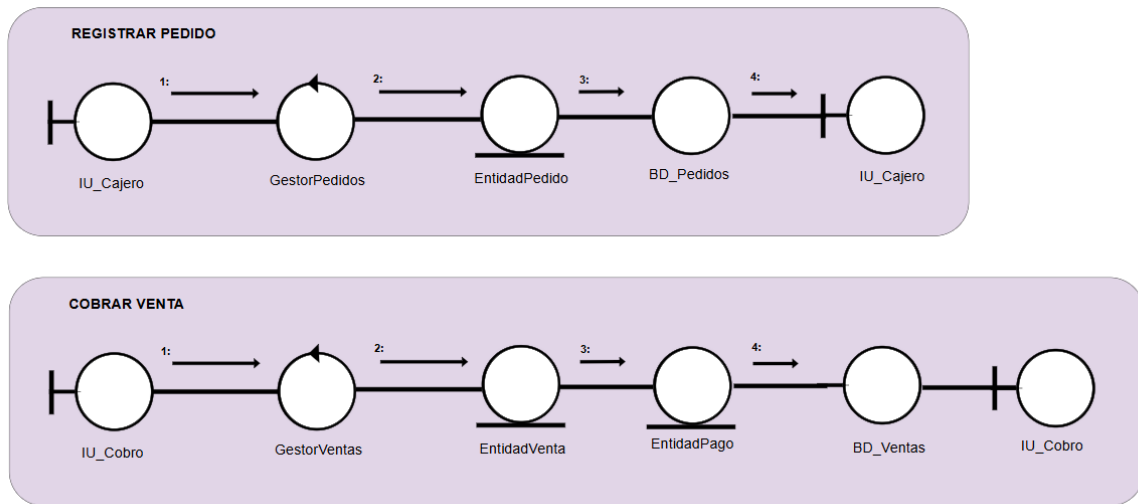


### 10.7.3. Diagrama de Casos de Uso por actor





#### 10.7.4. Diagrama de Colaboración



#### 10.7.5. Diagrama de Arquitectura

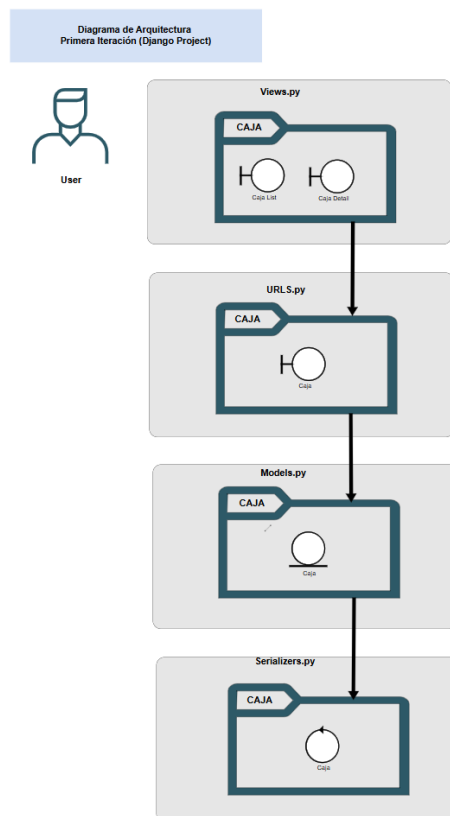


Diagrama de Arquitectura  
Segunda Iteración (Integración con react Frontend + Backend)

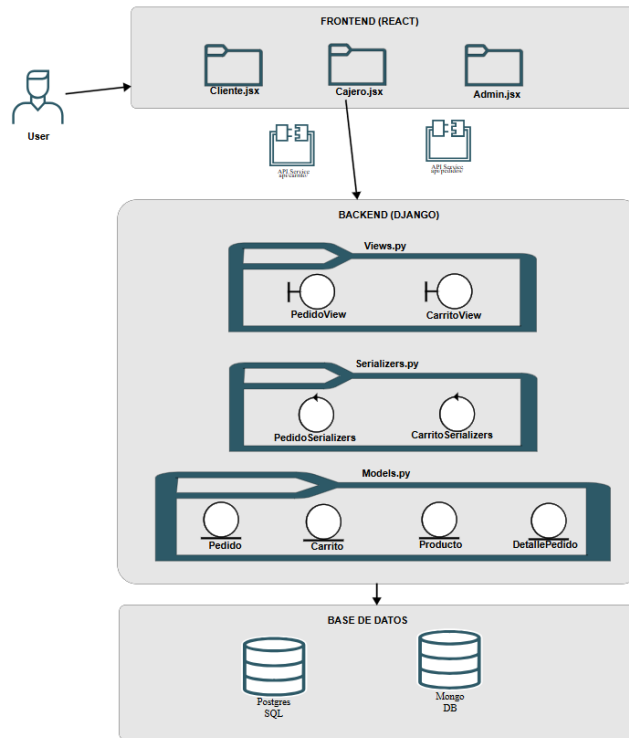
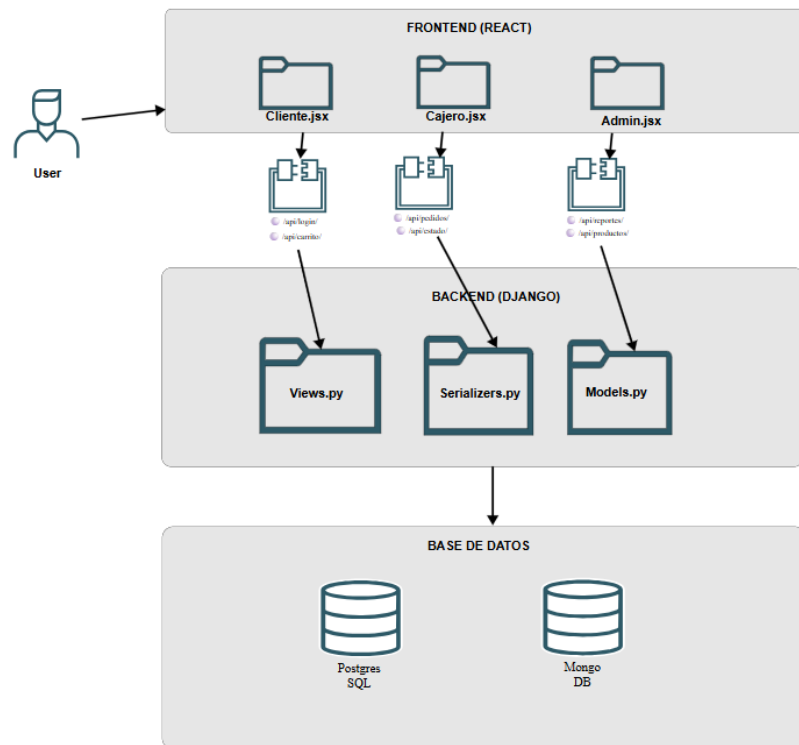
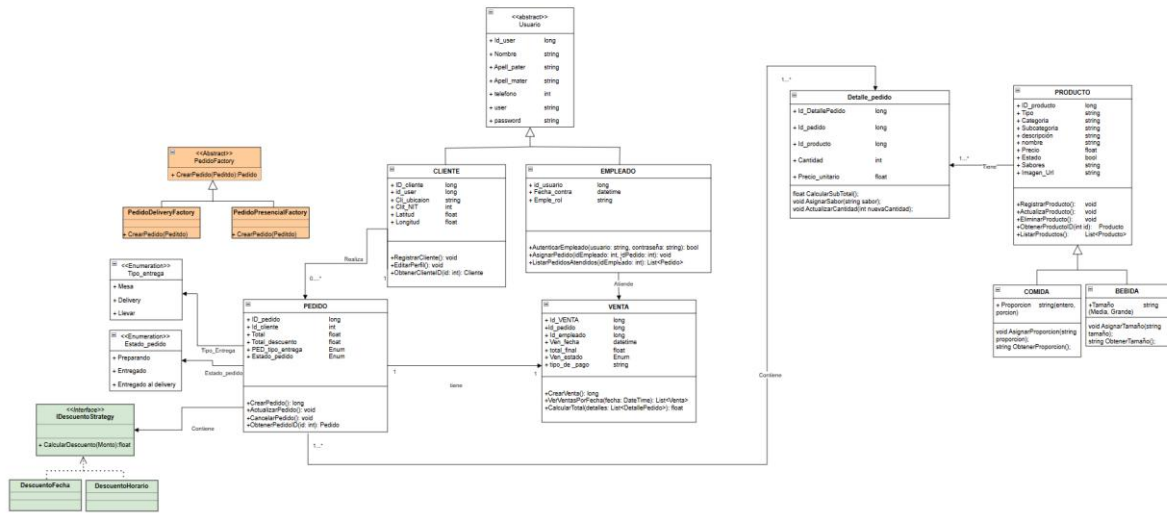


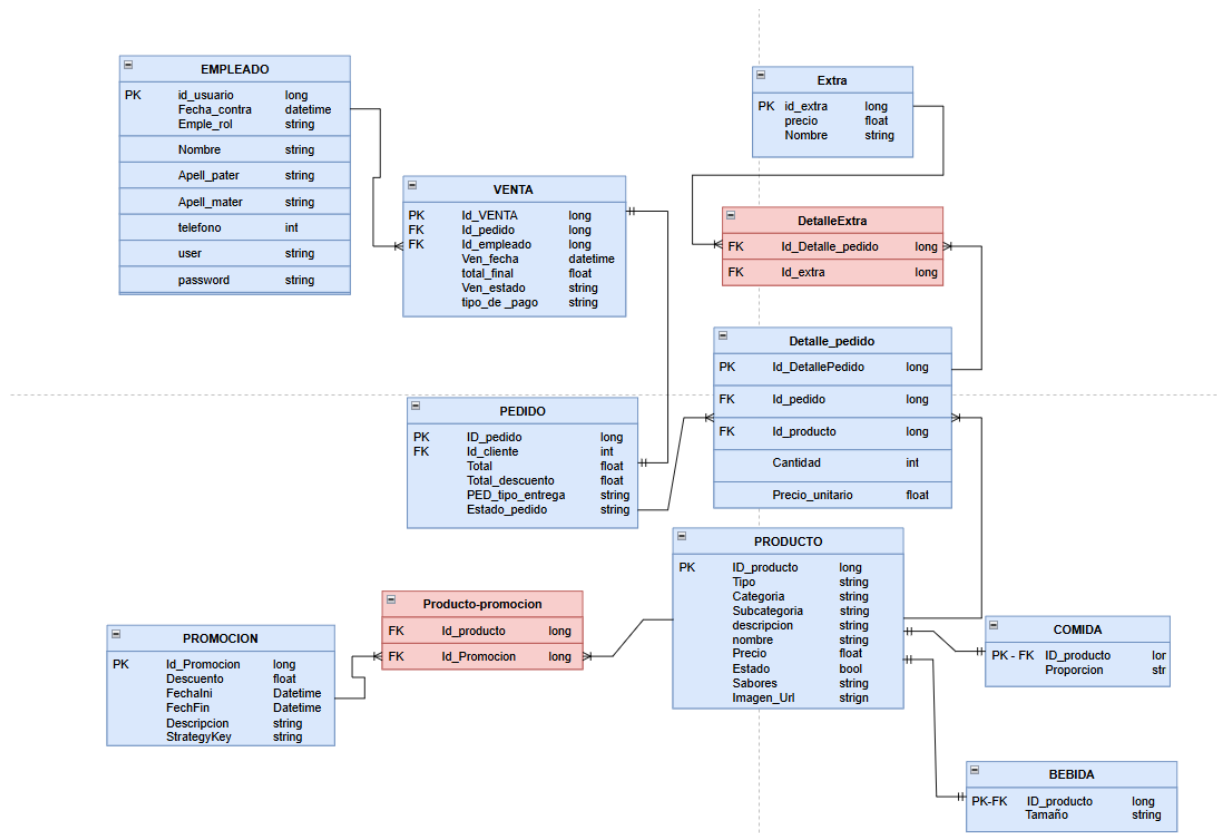
Diagrama de Arquitectura  
Cuarta Iteración (Arquitectura Completa)



### 10.7.6. Diagrama de Clases



### 10.7.7. Diagrama de Tablas Relacionales



### 10.7.8. Diccionario de Datos

#### 10.7.8.1. Diccionario de datos de la tabla “VENTA”

CAMPO	DESCRIPCION	TIPO DE DATO	INDICACION	VALIDACION
Id_venta	Clave de la venta	long	numero	No es nulo
Id_pedido	Clave foranea de la tabla pedido	long	numero	No es nulo
Id_empleado	Clave foránea de la tabla empleado	long	numero	No es nulo
Ven_fecha	Es la fecha en la que se hizo la venta	DateTime	fecha	No es nulo
Total_final	Es el precio total de todos la venta	float	numero	No es nulo
Ven_estado	Es el estado en el que se encuentra la venta	string	texto	No es nulo
Tipo_de_pago	Es el método con el que se hizo el pago (QR,tarjeta,efectivo)	string	texto	No es nulo

#### 10.7.8.2. Diccionario de datos de la tabla “PEDIDO”

CAMPO	DESCRIPCION	TIPO DE DATO	INDICACION	VALIDACION
Id_pedido	Clave del pedido	long	numero	No es nulo
Id_cliente	Clave foranea de la tabla cliente	long	numero	No es nulo
Total	Es el precio total del pedido	float	numero	No es nulo

Total_descuento	Es el precio total tomando en cuenta algun descuento si es que aplica	float	Numero	Puede ser nulo
PED_tipo_entrega	Describe el tipo de entrega que se hará(en sucursal o delivery)	string	texto	No es nulo
Estado_pedido	Indica en qué estado se encuentra el pedido, se pasara 0 si esta en espera, 1 si fue entregado.	string	int	No es nulo

#### 10.7.8.3. Diccionario de datos de la tabla “EMPLEADO”

CAMPO	DESCRIPCION	TIPO DE DATO	INDICACION	VALIDACION
Id_usuario	Clave del usuario	long	numero	No es nulo
Fecha_contra	Fecha en la que se contrato al empleado	dateTime	fecha	No es nulo
Emple_rol	Indica el rol del empleado	string	texto	No es nulo
Nombre	Indica el nombre del empleado	string	texto	No es nulo
Apell_pater	Indica el apellido paterno del empleado	string	texto	No es nulo
Apell_mater	Indica el apellido materno del empleado	string	texto	No es nulo
Telefono	Indica el numero telefónico del empleado	int	numero	No es nulo

user	Indica el usuario del empleado	string	texto	No es nulo
password	Indica la contraseña del empleado	string	texto	No es nulo

#### 10.7.8.4. Diccionario de datos de la tabla “DETALLE-PEDIDO”

CAMPO	DESCRIPCION	TIPO DE DATO	INDICACION	VALIDACION
Id_detalle_pedido	Clave del detalle del pedido	long	numero	No es nulo
Id_pedido	Clave foránea de tabla pedido	Long	Numero	No es nulo
Id_producto	Clave foránea de tabla productos	Long	Numero	No es nulo
cantidad	Indica la cantidad del producto en el pedido	int	Numero	No es nulo
Precio_unitario	Indica el precio de cada articulo en el pedido	float	Numero	No es nulo

#### 10.7.8.5. Diccionario de datos de la tabla “EXTRA”

CAMPO	DESCRIPCION	TIPO DE DATO	INDICACION	VALIDACION
Id_extra	Clave del extra	long	numero	No es nulo
precio	Precio del extra	float	Numero	No es nulo
Nombre	Nombre del extra	string	Texto	No es nulo

#### 10.7.8.6. Diccionario de datos de la tabla “DETALLE-EXTRA”

CAMPO	DESCRIPCION	TIPO DE DATO	INDICACION	VALIDACION
Id_detalle_pedido	Clave foránea del detalle del pedido	long	numero	No es nulo
Id_extra	Clave foránea de la tabla extra	long	Numero	No es nulo

#### 10.7.8.7. Diccionario de datos de la tabla “COMIDA”

CAMPO	DESCRIPCION	TIPO DE DATO	INDICACION	VALIDACION
Id_producto	Clave del producto	long	numero	No es nulo
Proporcion	Tamaño de la porcion	string	Texto	No es nulo

#### 10.7.8.8. Diccionario de datos de la tabla “BEBIDA”

CAMPO	DESCRIPCION	TIPO DE DATO	INDICACION	VALIDACION
Id_producto	Clave de la bebida	long	numero	No es nulo
Tamaño	Tamaño de la bebida	string	Número y unidades [ml,L]	No es nulo

### 11. CONCLUSIONES Y RECOMENDACIONES.

#### 11.1. CONCLUSIONES.

- Se logró diseñar un módulo de gestión de pedidos integral que abarca los tres canales de venta (local, online y telefónico), permitiendo el registro, actualización y seguimiento en tiempo real de cada orden. A través de diagramas de colaboración, casos de uso y un modelo de base de datos

relacional, se aseguró que el sistema ofrezca trazabilidad completa desde que se recibe el pedido hasta su entrega, mejorando la eficiencia operativa y la experiencia del cliente.

- Se definió un modelo de base de datos con tablas como PRODUCTO, DETALLE\_PEDIDO y VENTA, que permiten la actualización automática del inventario con cada transacción. El sistema incluye triggers o lógica de negocio para reducir el stock tras una venta y generar alertas cuando los niveles sean críticos, garantizando una gestión proactiva y evitando desabastecimientos o excesos de productos.
- Se planificó y estructuró un panel administrativo robusto durante el segundo sprint, con capacidad para gestionar usuarios y roles (empleados, administradores), crear y editar promociones, y generar reportes detallados de ventas. Este módulo fue respaldado por un diccionario de datos completo y diagramas de clases que aseguran su funcionalidad y seguridad mediante un control de acceso basado en roles.
- Se desarrollaron prototipos de interfaz de usuario durante el primer sprint, centrados en la experiencia del cliente. La interfaz es responsive, adaptable a distintos dispositivos, y permite a los usuarios registrarse, iniciar sesión, visualizar el catálogo de productos y promociones, gestionar un carrito de compras y realizar pedidos de forma anticipada, cumpliendo con los estándares de usabilidad y accesibilidad definidos en los requerimientos no funcionales.

## **11.2. RECOMENDACIONES.**

- Se recomienda incorporar módulos adicionales como generación de reportes automáticos, sistema de puntos o fidelización de clientes, y soporte para múltiples sucursales o cadenas.
- Realizar pruebas periódicas de rendimiento y seguridad, además de mantener la base de datos optimizada con índices y procedimientos almacenados según sea necesario. Implementar herramientas de monitoreo puede ayudar a detectar cuellos de botella en tiempo real.



- Brindar capacitación básica al personal que utilizará el sistema, especialmente en funciones críticas como registro de pedidos para asegurar un uso eficiente del sistema.
- Para mejorar la disponibilidad y escalabilidad del sistema, se recomienda considerar su despliegue en plataformas cloud como Azure o AWS, lo que facilitaría el mantenimiento, el respaldo automático y el acceso remoto seguro.

## 12. BIBLIOGRAFÍA

Doyle, K. (12 de Diciembre de 2024). *Django*. Obtenido de TecTarget.com: <https://www.techtarget.com/searchapparchitecture/tip/Django-vs-Flask-Comparing-Python-web-frameworks>

Fuentes, G. (2012). *Bases de datos. Gestión*. Obtenido de Notas\_del\_curso\_Bases\_de\_Datos: [http://local.cua.uam.mx/pdfs/conoce/libroselec/Notas\\_del\\_curso\\_Bases\\_de\\_Datos.pdf](http://local.cua.uam.mx/pdfs/conoce/libroselec/Notas_del_curso_Bases_de_Datos.pdf)

MariaDB, F. (2009). *MariaDB Server*. Obtenido de mariadb.org: <https://mariadb.org/es/>

Martínez, F. (2013). Revisión de metodologías ágiles para el desarrollo de software. En A. Navarro Cadavid, *PROSPECTIVA* (págs. 30-39). Colombia: Universidad

Raul Awati, Ben Lutkevich. (22 de Noviembre de 2024). *¿Qué es un framework?* Obtenido de TechTarget.com: <https://www.techtarget.com/whatis/definition/framework>

Rioja, U. I. (20 de Noviembre de 2023). *Que son las librerías en programación y para que sirven*. Obtenido de unir.net: <https://www.unir.net/revista/ingenieria/librerias-programacion/>

s.

# ANEXOS



