

MARCO CONCEPTUAL



TECNOLÓGICO
NACIONAL DE MÉXICO



EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO NACIONAL DE MÉXICO
INSTITUTO TECNOLÓGICO DE OAXACA
INGENIERÍA EN SISTEMAS COMPUTACIONALES

MATERIA:

Ingeniería de Software

TÍTULO:

Pastelería- DulceSys

CATEDRÁTICO:

Espinoza Perez Jacob

PRESENTA:

Cervantes Medina Carlos David

Fernández López Jazmín

Mendoza Sánchez Yhudiel

Santiago Vasquez Tania

Grupo:

6SU

Equipo:

3

Oaxaca de Juárez, Oaxaca, a 13 de noviembre del 2025

Índice general

| | |
|--|-----------|
| Introducción..... | 3 |
| MARCO CONCEPTUAL..... | 4 |
| 1.1 Nombre del proyecto..... | 4 |
| 1.2 Giro de la empresa..... | 4 |
| 1.3 Misión..... | 4 |
| 1.4 Visión..... | 5 |
| 1.5 Croquis de ubicación..... | 5 |
| 1.6 Objetivo..... | 5 |
| 1.6.1 Objetivo general..... | 5 |
| 1.6.2 Objetivos específicos..... | 5 |
| 1.7 Planteamiento del problema..... | 6 |
| 1.8 Procesos de la organización..... | 7 |
| Control de Inventario..... | 7 |
| Generación de Cotizaciones..... | 7 |
| Gestión de Pedidos..... | 7 |
| Fidelización de Clientes..... | 7 |
| 1.9 Presentación del producto de software..... | 8 |
| 1.10 Viabilidad..... | 8 |
| 1.10.1 Viabilidad técnica..... | 8 |
| 1.10.2 Viabilidad económica..... | 8 |
| 1.10.3 Viabilidad operacional..... | 8 |
| 1.11 Requerimientos..... | 12 |
| 1.11.1 Requerimientos funcionales..... | 12 |
| 1.11.2 Requerimientos no funcionales..... | 14 |
| 1.11.3 Justificación..... | 16 |
| 2.1 Diagrama UML..... | 17 |
| 2.3 Diagramas de casos de uso..... | 18 |
| 2.3.1 Iniciar Sesión..... | 19 |
| 2.3.2 Cerrar sesión..... | 20 |
| 2.3.3 Registrar Cliente..... | 21 |
| 2.3.4 Consultar Menú de Producto..... | 22 |
| 2.3.5 Realizar Pedido | 23 |
| 2.3.6 Consultar Historial de Pedidos..... | 24 |
| 2.3.7 Consultar Recompensas..... | 25 |
| 2.3.8 Ver Perfil..... | 26 |
| 2.3.9 Actualizar Perfil de Cliente..... | 27 |
| 2.3.10 Gestionar Estado de Pedidos..... | 28 |
| 2.3.11 Consultar Detalle de Pedido..... | 29 |
| 2.3.12 Procesar Pago de Pedido..... | 30 |
| 2.3.13 Imprimir Ticket | |

| | |
|--|-----------|
| | 31 |
| 2.3.14 Gestionar Productos..... | 32 |
| 2.3.15 Gestionar Empleados..... | 33 |
| 2.3.16 Gestionar Clientes..... | 34 |
| 2.3.17 Gestionar Proveedores..... | 35 |
| 2.3.18 Gestionar Recompensas..... | 36 |
| 2.3.19 Visualizar estadísticas..... | 37 |
| 2.3.20 Gestionar Inventario de Ingredientes..... | 38 |
| 2.3.21 Realizar Corte de Caja..... | 39 |
| 2.4 Diagramas de secuencia..... | 40 |
| 2.4.1 Cliente..... | 41 |
| 2.4.2 Administrador..... | 42 |
| 2.4.3 Cajero..... | 43 |
| 2.5 Diagramas de interacciones..... | 44 |
| 2.5.1 Cliente..... | 45 |
| 2.5.2 Administrador..... | 46 |
| 2.5.3 Cajero..... | 47 |
| HISTORIAS DE USUARIOS..... | 48 |
| Cliente..... | 49 |
| ADMINISTRADOR..... | 50 |
| CAJERO..... | 51 |
| BASE DE DATOS..... | 52 |
| DISEÑO Y VISTAS..... | 53 |
| MANUAL DE CONSULTAS..... | 54 |
| DEVELOPER MANUAL..... | 55. |

Introducción

El presente documento describe el análisis y diseño del sistema de gestión DulceSys, desarrollado para optimizar las operaciones de la pastelería "La Casa del Pastel". Esta solución surge como respuesta a las problemáticas identificadas en los procesos actuales de la empresa, donde la gestión manual de inventarios, cotizaciones y pedidos genera ineficiencias operativas, costos adicionales y una experiencia de cliente inconsistente.

Presenta el modelado completo del sistema describiendo su estructura, comportamiento y funcionamiento. A través de diversos diagramas UML se ilustran los componentes del sistema, las interacciones entre actores y procesos, así como los flujos de información necesarios para cumplir con los objetivos del negocio.

El análisis y diseño aquí presentados servirán como base para la implementación técnica del sistema, garantizando que la solución desarrollada responda efectivamente a las necesidades identificadas en "La Casa del Pastel".

MARCO CONCEPTUAL

Este documento tiene como finalidad establecer el marco conceptual para el desarrollo de una aplicación, se enfoca en resolver los desafíos operativos que afectan la eficiencia y la satisfacción del cliente.

1.1 Nombre del proyecto

DulceSys

1.2 Giro de la empresa

Industrial – Alimenticio / Repostería



figura 1 .Logotipo de la empresa

1.3 Misión

La casa del pastel brinda pasteles y productos de repostería de alta calidad, con frescura. Buscamos brindar una experiencia dulce y memorable a través de un excelente servicio, garantizando la satisfacción y confianza de nuestros consumidores.

1.4 Visión

Ser una pastelería líder en Oaxaca por su calidad, innovación y atención al cliente.

1.5 Croquis de ubicación

"La Casa del Pastel" está ubicada en la ciudad de Oaxaca de Juárez, de tamaño medio, con una creciente demanda de productos de repostería de alta calidad. Su principal área de influencia abarca la ciudad y sus alrededores, atendiendo tanto a clientes individuales como a empresas que requieren pedidos personalizados para eventos especiales. La empresa opera a través de su tienda física. Además, está evaluando la posibilidad de expandir sus servicios a otras regiones mediante alianzas estratégicas o plataformas de venta en línea.

Ubicación: Orquídeas 110, Jardines de la Primavera, 68285 Oaxaca de Juárez, Oax.



Figura 2. Croquis de ubicación, casa del pastel

1.6 Objetivo

1.6.1 Objetivo general

Optimizar la administración de pedidos, el control de inventario y la generación de cotizaciones para mejorar la eficiencia operativa y la satisfacción del cliente.

1.6.2 Objetivos específicos

- Controlar el inventario automáticamente
- Alertar cuando falten ingredientes
- Calcular cotizaciones rápidamente

- Registrar pedidos digitalmente
- Guardar información de clientes
- Avisar de productos por vencer
- Crear programa de puntos para clientes
- Eliminar registros manuales en libretas
- Facilitar la comunicación con producción

1.7 Planteamiento del problema

La pastelería "La Casa del Pastel" enfrenta actualmente varios desafíos operativos que afectan directamente su eficiencia y la satisfacción de sus clientes. Los problemas más críticos identificados son:

Descontrol del inventario: Los ingredientes se almacenan sin un sistema formal de control de stock, dependiendo únicamente de verificaciones visuales.

Demoras en la generación de cotizaciones: El cálculo manual de precios para pedidos personalizados consume tiempo considerable. Según encuestas aplicadas, el 75% de los clientes considera que las cotizaciones tardan demasiado, especialmente en productos con decoraciones personalizadas.

Ausencia de estrategias de fidelización: No existe un registro formal del historial de compras de los clientes.

Gestión manual de pedidos: La información se registra manualmente en libretas y se transmite verbalmente o por notas al equipo de producción, lo que genera inconsistencias y posibles errores en la comunicación.

Estos problemas se traducen en altos costos operativos, pérdida de oportunidades de negocio y una experiencia de cliente inconsistente, evidenciando la necesidad urgente de implementar un sistema digital que optimice estos procesos.

1.8 Procesos de la organización

Control de Inventario

- Los ingredientes y materiales de repostería se almacenan en la bodega sin un sistema formal de control de stock
- Los encargados de producción verifican visualmente la disponibilidad de insumos antes de comenzar la producción

Generación de Cotizaciones

- Un cliente solicita información sobre precios de pasteles o productos personalizados
- El empleado registra los siguientes puntos, considerando ingredientes, tamaño, tipo de decoración y tiempo de producción

Gestión de Pedidos

- El cliente visita la tienda física o se comunica por teléfono para realizar un pedido
- Un empleado anota manualmente la información del pedido en una libreta o en una hoja de registro

Fidelización de Clientes

- Los clientes compran productos sin un registro formal de su historial de compras

1.9 Presentación del producto de software

DulceSys es una solución digital integral diseñada específicamente para optimizar la operación de pastelerías sin complicaciones. Facilita el registro, seguimiento y administración de pedidos de forma digital, eliminando los procesos manuales y mejorando la comunicación con el equipo de producción. Genera cotizaciones automáticas considerando ingredientes, tamaño, decoración y tiempo de producción, reduciendo significativamente los tiempos de espera para los clientes. Mantiene un monitoreo constante de los niveles de stock, genera alertas de productos por agotarse o próximos a caducar, y previene compras urgentes innecesarias. Implementa estrategias para fortalecer la relación con los clientes mediante el registro de historial de compras, preferencias y programas de lealtad.

1.10 Viabilidad

1.10.1 Viabilidad técnica

1.10.2 Viabilidad económica

1.10.3 Viabilidad operacional

| Concepto | Detalle | Costo |
|-----------------------|------------------------------|-------------|
| Programadores | 4 programadores | - |
| Sueldo quincenal | \$45,000 por programador | \$45,000 |
| Período de desarrollo | 3 meses = 6 quincenas | - |
| TOTAL DESARROLLO | $4 \times \$45,000 \times 6$ | \$1,080,000 |

Software Requerido

| Componente | Especificación | Costo Mensual (MXN) |
|----------------------------|------------------|---------------------|
| Instancia (CPU y RAM) | 2 vCPU, 8 GB RAM | \$600 - \$1,000 |
| Almacenamiento SSD | 100 GB | \$340 |
| Copias de Seguridad | 50 GB (estimado) | \$80 |
| TOTAL MENSUAL GOOGLE CLOUD | | \$1,020 - \$1,420 |

| Servicio | Descripción | Costo Mensual (MXN) |
|----------------|--------------------------------|---------------------|
| Salesforce CRM | Empresa Centrada en el Cliente | \$500 |

Costo de Implementación (Hardware). Computadoras para Encargados de Gestión

Requisitos recomendados:

- Procesador: Intel Core i5 o superior
- RAM: 8 GB mínimo, 16 GB ideal
- Almacenamiento: SSD de 256 GB o más
- Sistema Operativo: Windows 10/11 o Ubuntu Linux
- Monitor: 21" o más

Costos de Manutención

Por Programador (Mensual)

| Concepto | Costo por Persona |
|---|-------------------|
| Alimentación | \$3,000 |
| Servicios básicos (Internet, Luz, Agua) | \$1,000 |
| Renta | \$1,000 |
| Manutención mensual | \$5,000 |

| Categoría | Concepto | Cálculo | Costo (MXN) |
|--------------------------------|-----------------------------|--|--------------------|
| Técnica | Sueldos programadores | $4 \times \$45,000 \times 6$ quincenas | \$1,080,000 |
| | Google Cloud | $\$1,020 \times 3$ meses | \$3060 |
| | Salesforce CRM | $\$500 \times 3$ meses | \$1,500 |
| Económica | Computadoras | $4 \times \$3,000$ | \$12,000 |
| | Equipo adicional | $4 \times \$30,000$ | \$120,000 |
| | Manutención | $4 \times \$5,000 \times 3$ meses | \$60,000 |
| Operacional | Recursos técnicos y humanos | – | Viable |
| Total general (3 meses) | | | \$1,276,556 |

1.11 Requerimientos

1.11.1 Requerimientos funcionales

| Nº | Requerimiento Funcional | Descripción |
|-----|--|---|
| RF1 | Gestión de inventario | El sistema deberá permitir registrar, modificar y eliminar ingredientes o insumos, así como mostrar alertas cuando un producto esté próximo a agotarse o a caducar. |
| RF2 | Control de stock en tiempo real | El sistema deberá actualizar automáticamente las existencias del inventario cada vez que se registre un pedido o se realice una compra de insumos. |
| RF3 | Gestión de pedidos | El sistema permitirá registrar nuevos pedidos, editar información y consultar su estado (pendiente, en producción, entregado, cancelado). |
| RF4 | Generación de cotizaciones automáticas | El sistema calculará precios basados en los ingredientes, tamaño, tipo de pastel, decoración y tiempo estimado de producción, generando una cotización inmediata. |
| RF5 | Registro de clientes | El sistema permitirá crear y gestionar perfiles de clientes, almacenando su historial de compras y pedidos anteriores. |
| RF6 | Fidelización de clientes | Se deberá implementar un módulo que registre puntos, descuentos o promociones para clientes frecuentes. |

| | | |
|------|-----------------------------|--|
| RF7 | Gestión de usuarios y roles | El sistema deberá permitir la creación de usuarios con distintos roles (Administrador, Vendedor, Encargado de producción). |
| RF8 | Reportes administrativos | El sistema generará reportes de ventas, pedidos y consumo de ingredientes, exportables en formato PDF o Excel. |
| RF9 | Notificaciones y alertas | Se enviarán alertas internas sobre inventarios bajos, pedidos urgentes o entregas próximas. |
| RF10 | Historial de actividades | El sistema guardará un registro de las acciones realizadas por los usuarios (por ejemplo, creación de pedido, modificación de cotización). |

1.11.2 Requerimientos no funcionales

| Nº | Requerimiento Funcional | No | Descripción |
|------|-------------------------|----|---|
| RNF1 | Rendimiento | | El sistema deberá responder a las solicitudes de los usuarios en menos de 3 segundos bajo carga normal. |
| RNF2 | Disponibilidad | | El sistema estará disponible el 99% del tiempo durante el horario laboral (8:00 a 20:00). |
| RNF3 | Usabilidad | | La interfaz deberá ser intuitiva, con menús claros y botones accesibles para usuarios con conocimientos básicos en computación. |
| RNF4 | Seguridad de datos | | Las contraseñas deberán almacenarse cifradas y las conexiones protegidas mediante HTTPS o equivalentes. |
| RNF5 | Escalabilidad | | El sistema deberá poder adaptarse para manejar más productos, clientes o pedidos sin modificar su arquitectura principal. |
| RNF6 | Mantenibilidad | | El código deberá estar documentado y seguir estándares de desarrollo que faciliten futuras actualizaciones. |

| | | |
|-------|----------------------------------|--|
| RNF7 | Portabilidad | El sistema deberá ejecutarse tanto en Windows como en Linux, compatible con navegadores modernos si se despliega en web. |
| RNF8 | Compatibilidad con base de datos | El sistema deberá integrarse con PostgreSQL, según lo especificado en la viabilidad técnica. |
| RNF9 | Respaldo de información | Deberán realizarse copias de seguridad automáticas diarias de la base de datos. |
| RNF10 | Confiabilidad | En caso de error, el sistema deberá permitir la recuperación segura de los datos y registrar logs de fallos. |

1.11.3 Justificación

La implementación del sistema DulceSys surge como respuesta a las problemáticas operativas detectadas en la pastelería “*La Casa del Pastel*”, donde actualmente los procesos de control de inventario, generación de cotizaciones y gestión de pedidos se realizan de manera manual, lo que provoca ineficiencias, errores humanos y pérdida de oportunidades comerciales. El tiempo excesivo en la elaboración de cotizaciones y la falta de control sobre los insumos afectan directamente la calidad del servicio y la satisfacción del cliente.

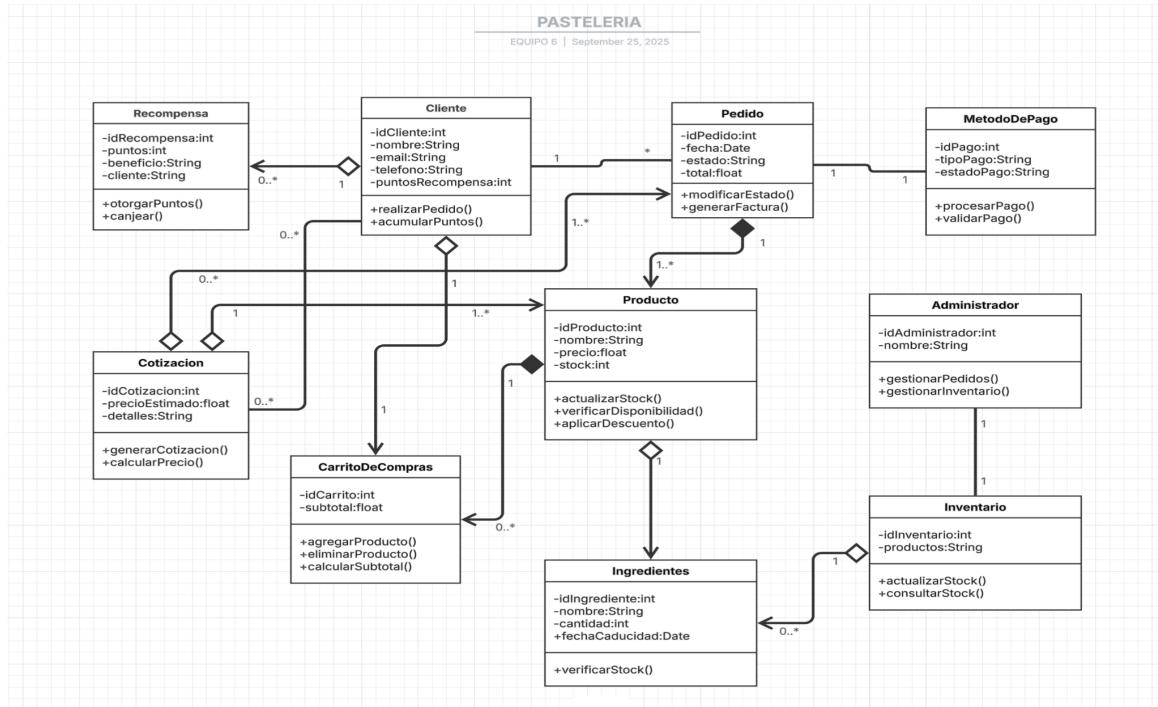
La automatización de estos procesos permitirá optimizar los recursos disponibles y mejorar la eficiencia operativa del negocio. Con el sistema propuesto, la pastelería podrá gestionar en tiempo real su inventario, evitando pérdidas por caducidad o escasez de ingredientes, además de agilizar la generación de cotizaciones mediante cálculos automáticos basados en los insumos y características del pedido. Esto no solo reducirá los tiempos de atención al cliente, sino que también aumentará la precisión en los precios y presupuestos.

Asimismo, DulceSys busca fortalecer la fidelización de los clientes, implementando un registro digital de compras que permita ofrecer promociones, descuentos o recompensas personalizadas, incentivando la recompra y el crecimiento sostenido de la clientela. Desde el punto de vista tecnológico, la propuesta es totalmente viable, ya que se apoya en herramientas modernas como Node.js para el desarrollo del sistema, PostgreSQL para la base de datos y servicios en la nube que garantizan la disponibilidad y seguridad de la información. Estas tecnologías aseguran un sistema confiable, escalable y adaptable a las necesidades futuras de la empresa.

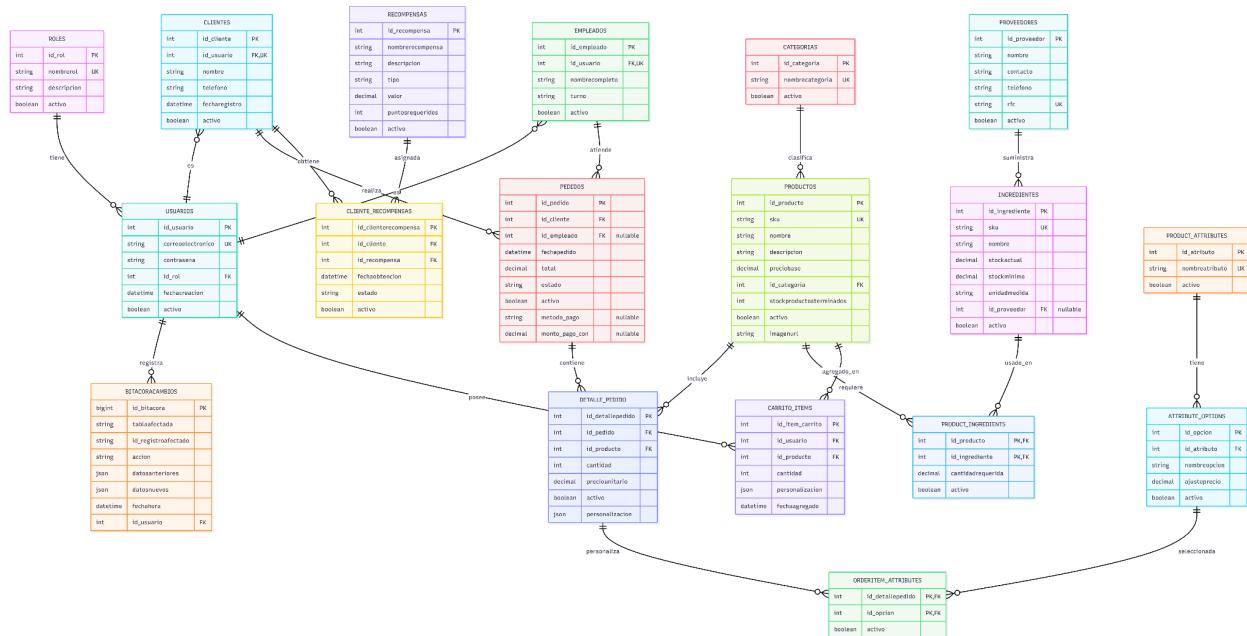
DIAGRAMAS

DIAGRAMAS

2.1 Diagrama de clases.

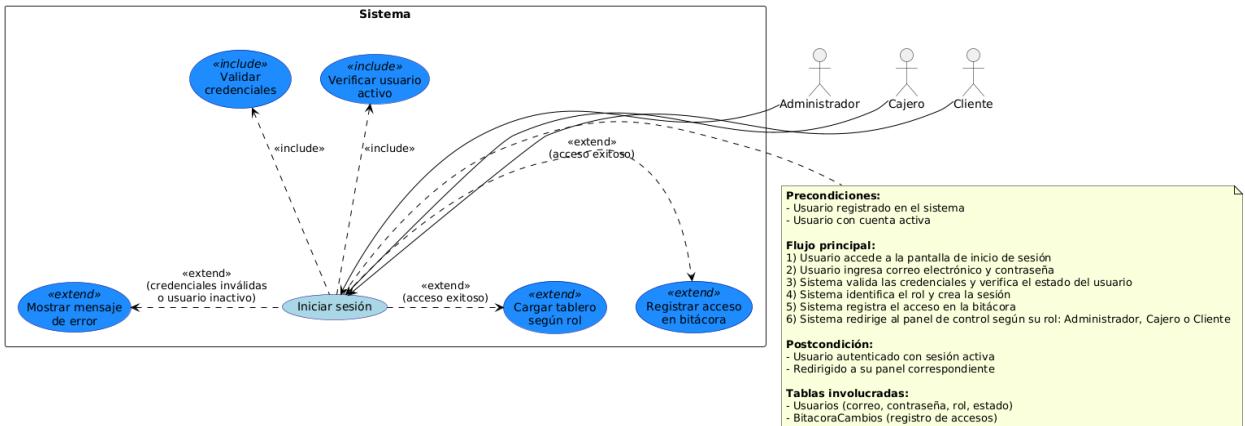


2.2 diagrama relacional

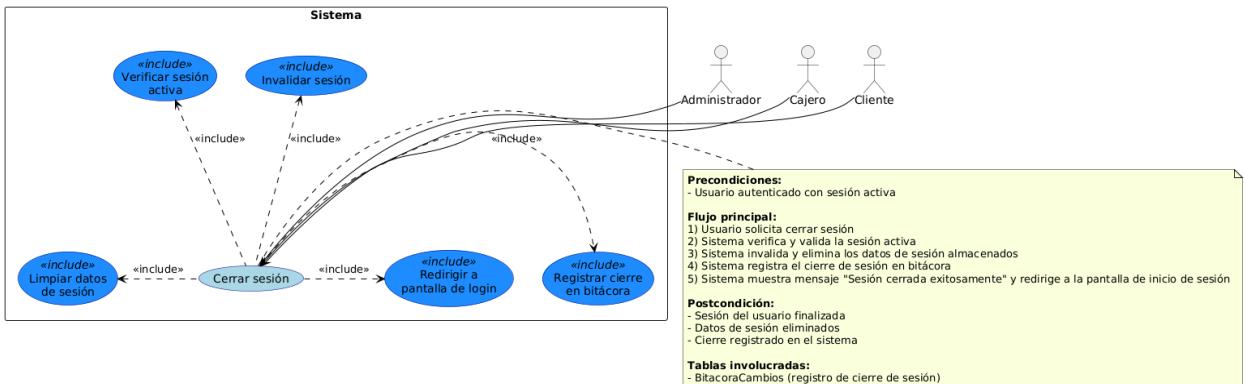


2.3 Diagramas de casos de uso.

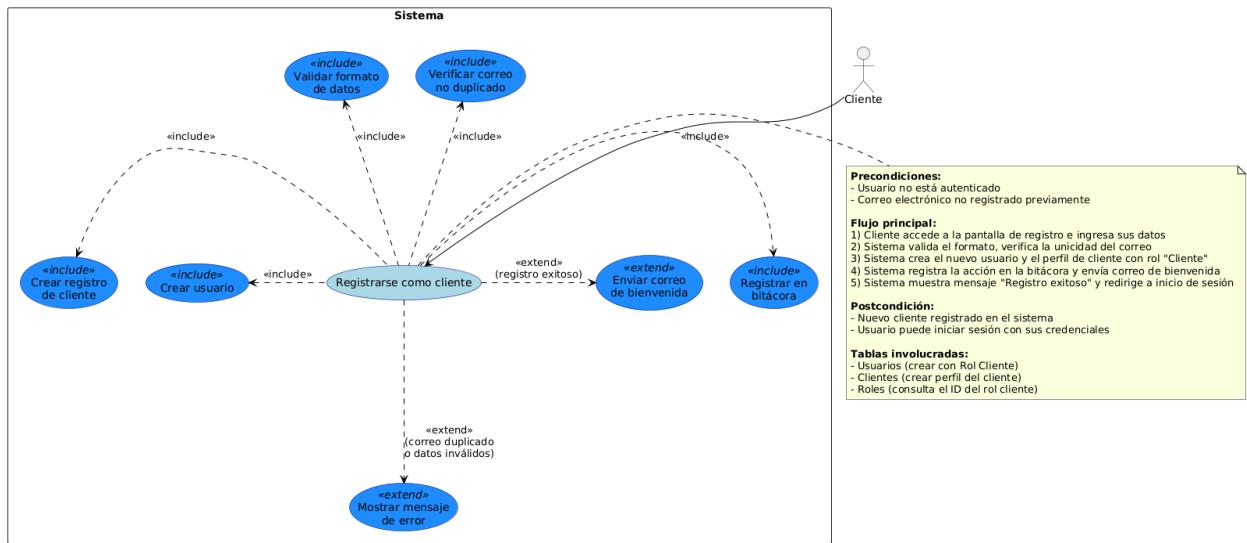
2.3.1 Iniciar Sesión.



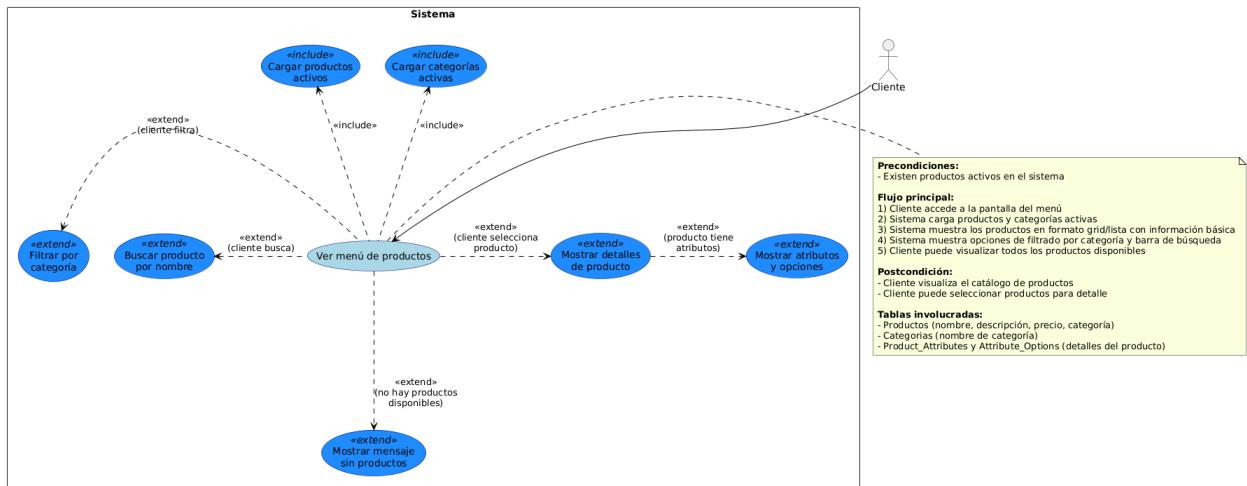
2.3.2 Cerrar sesión



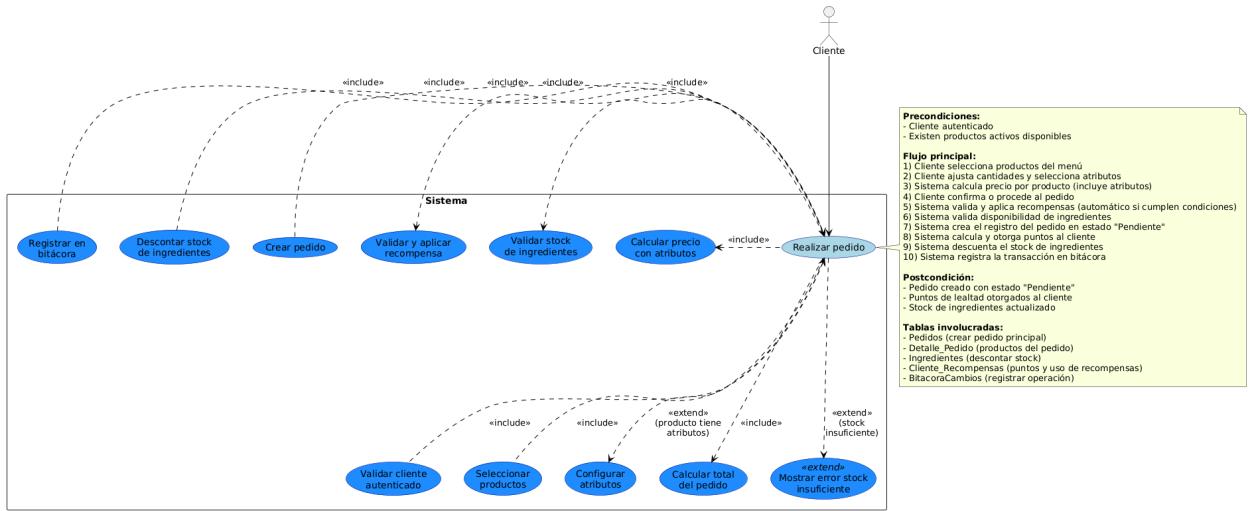
2.3.3 Registrar Cliente



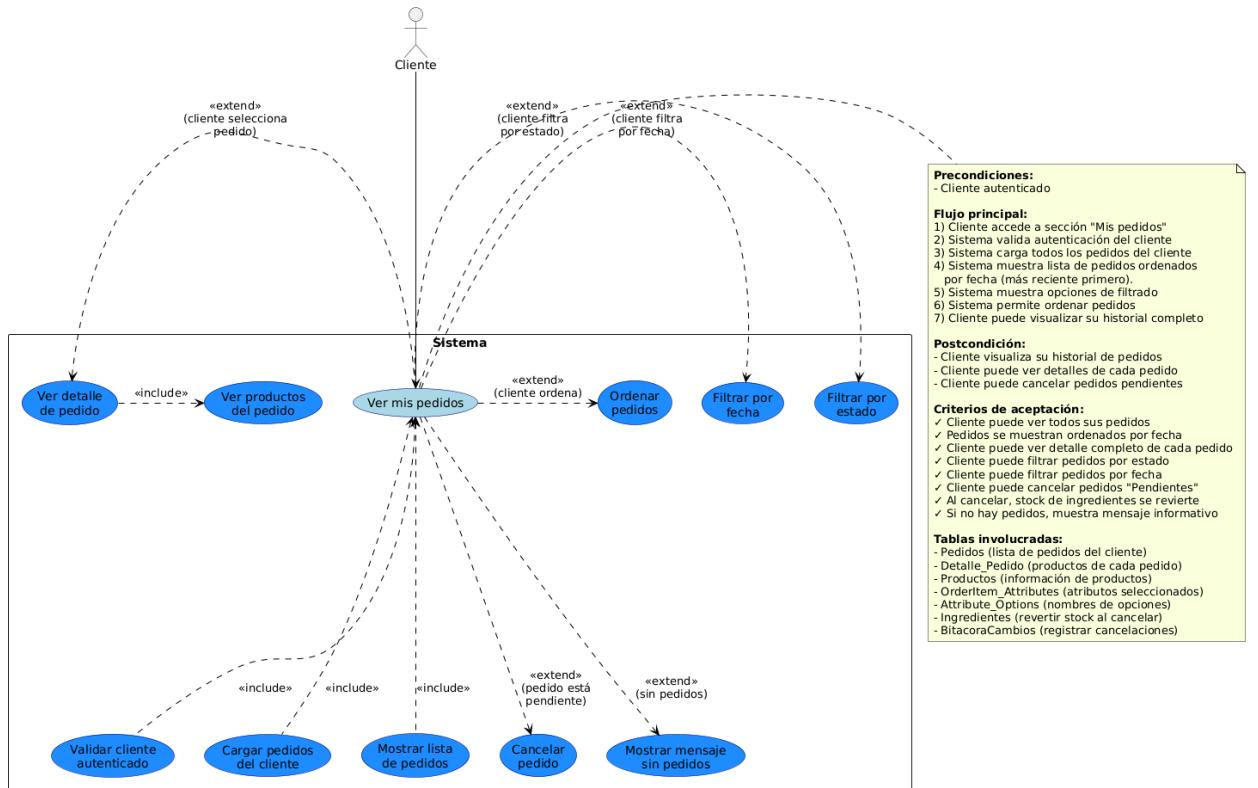
2.3.4 Consultar Menú de Producto



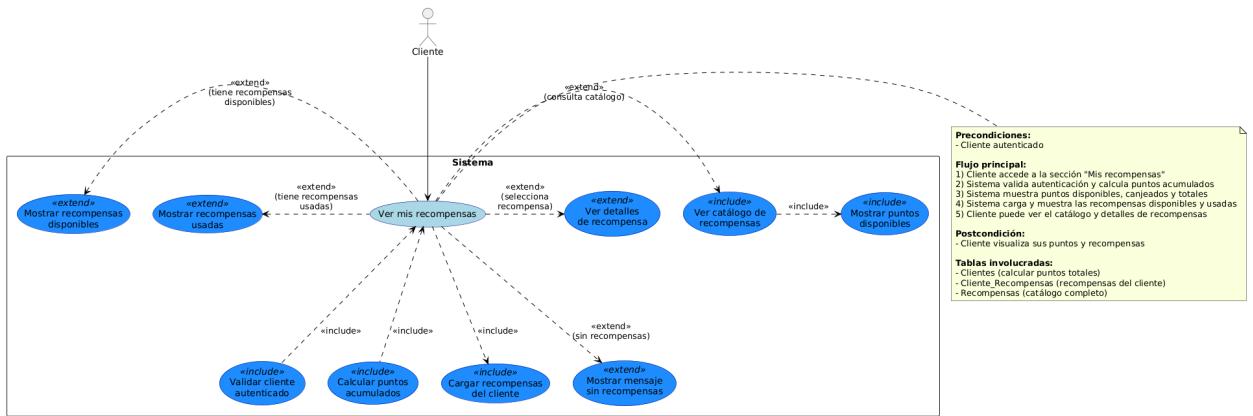
2.3.5 Realizar Pedido



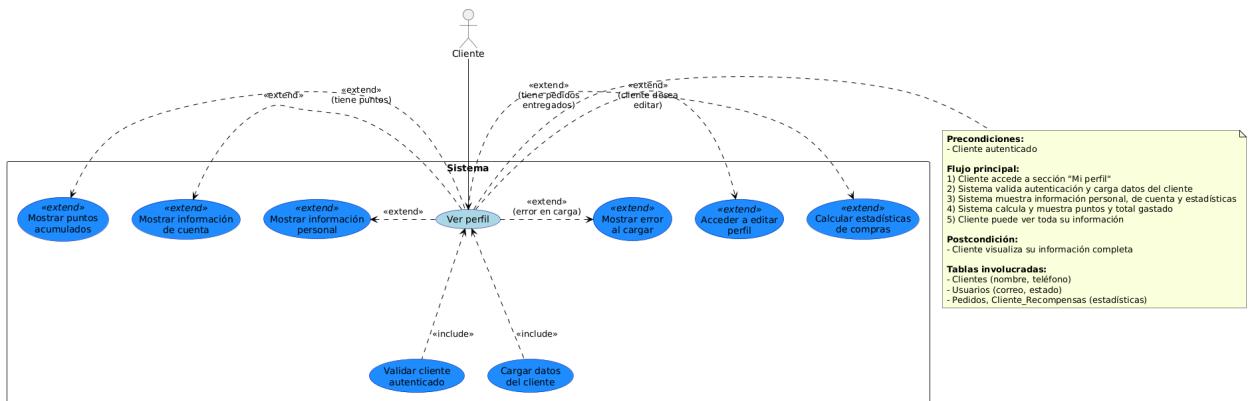
2.3.6 Consultar Historial de Pedidos



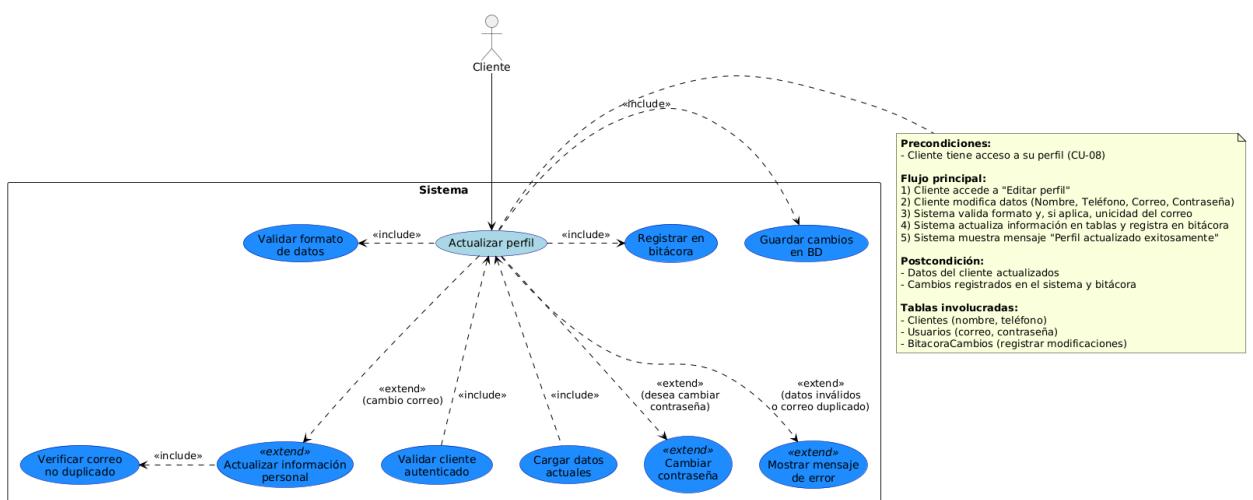
2.3.7 Consultar Recompensas



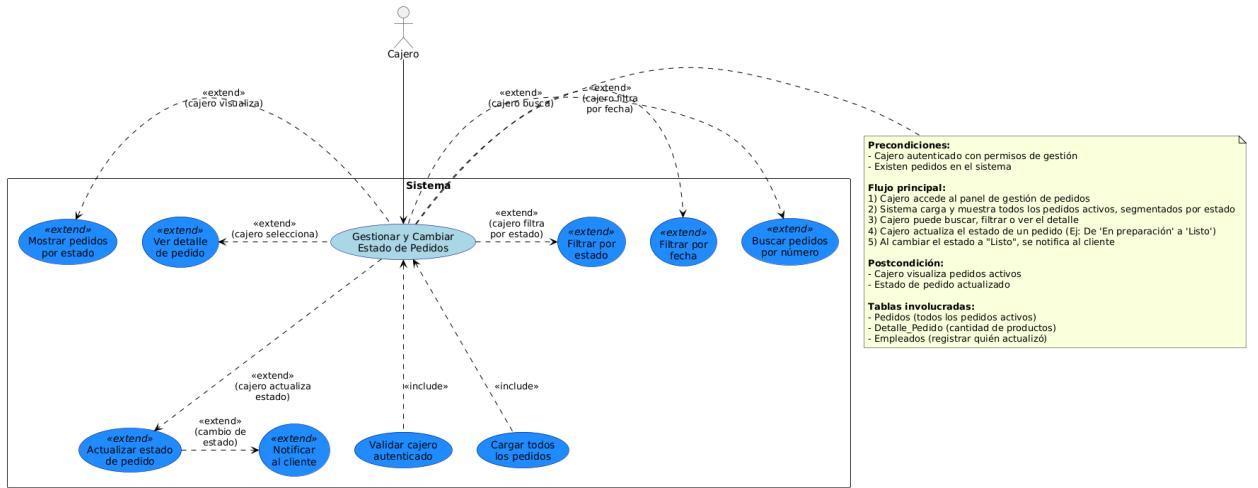
2.3.8 Ver Perfil



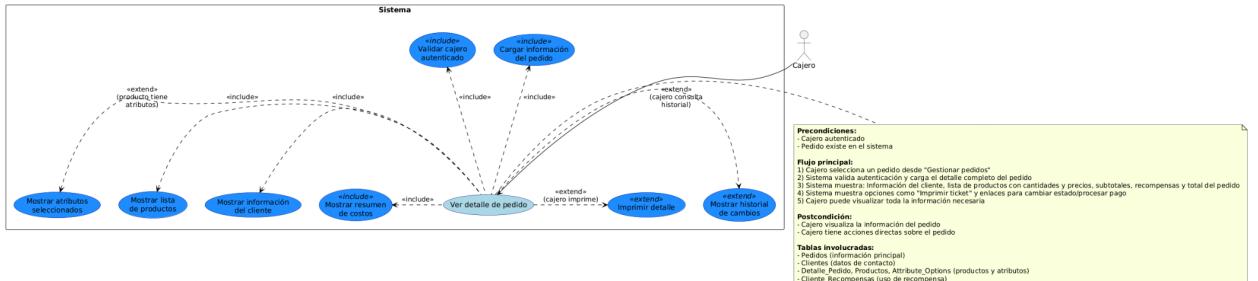
2.3.9 Actualizar Perfil de Cliente



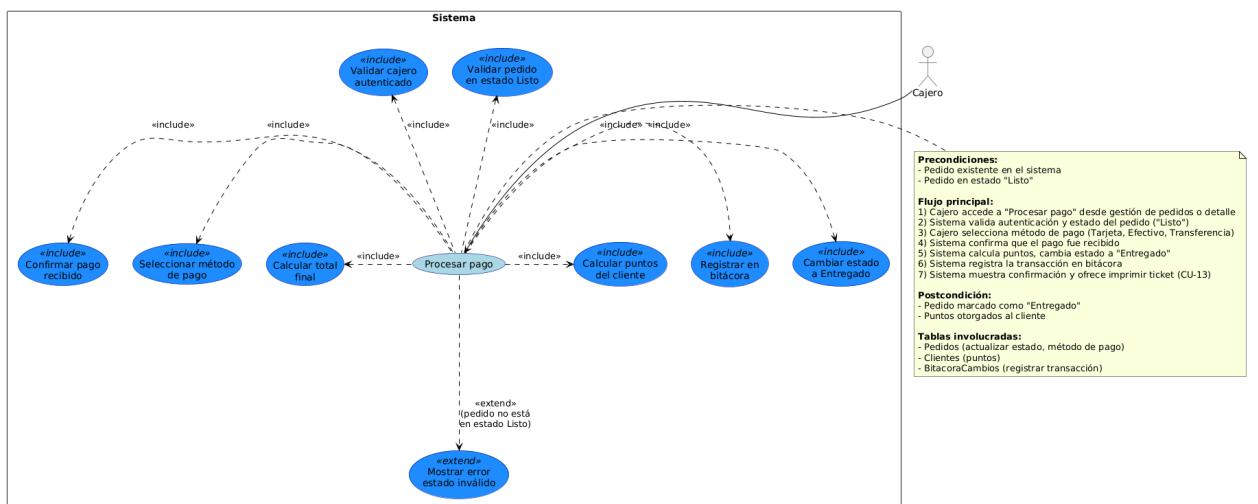
2.3.10 Gestionar Estado de Pedidos



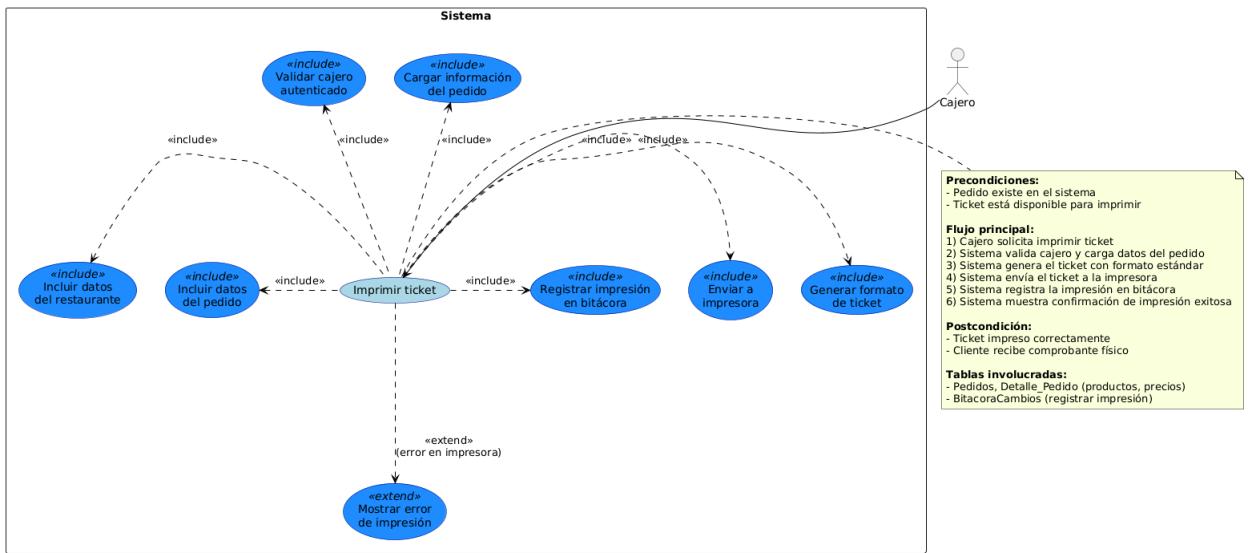
2.3.11 Consultar Detalle de Pedido



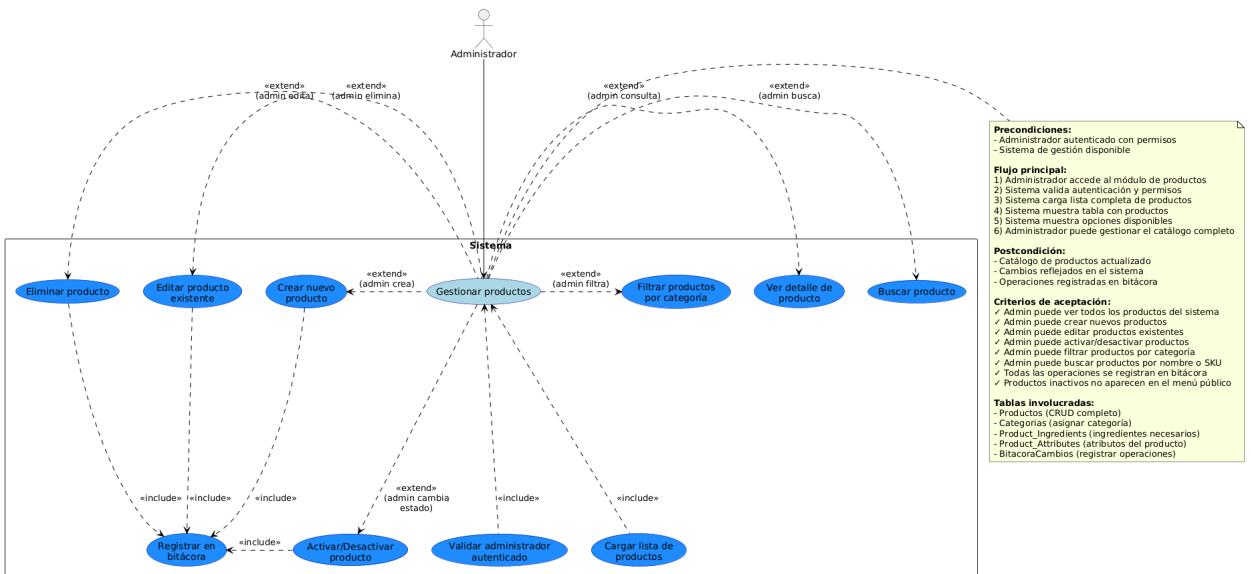
2.3.12 Procesar Pago de Pedido



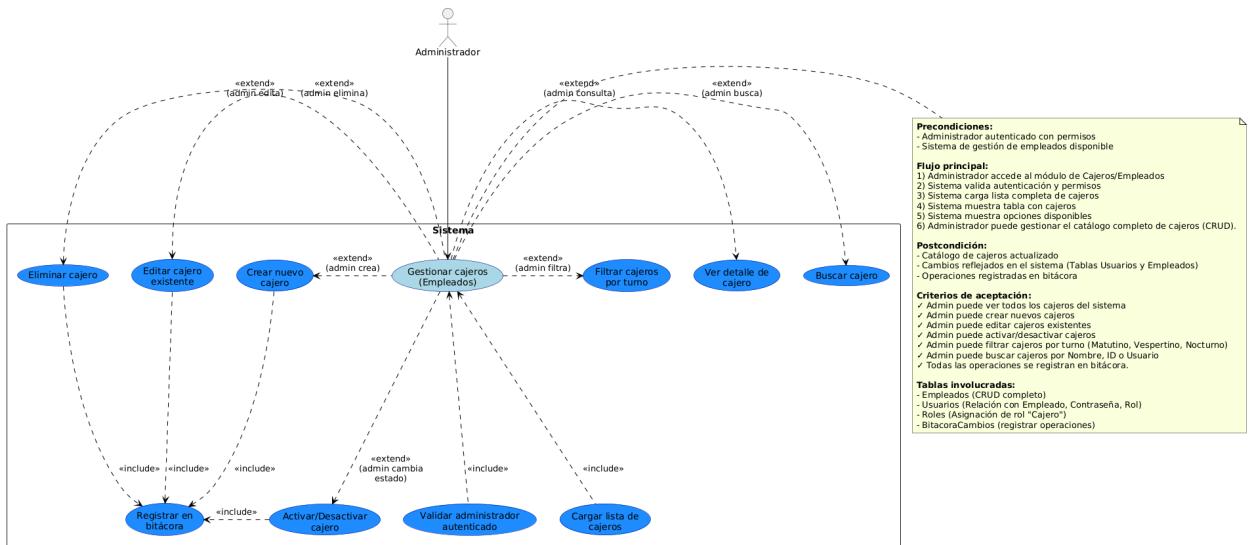
2.3.13 Imprimir Ticket



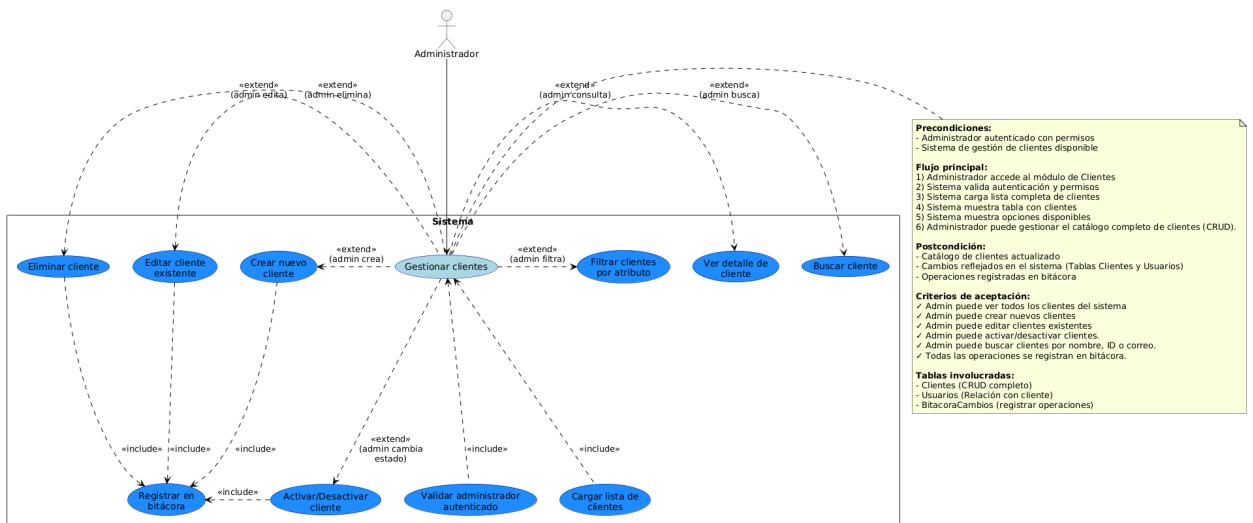
2.3.14 Gestionar Productos



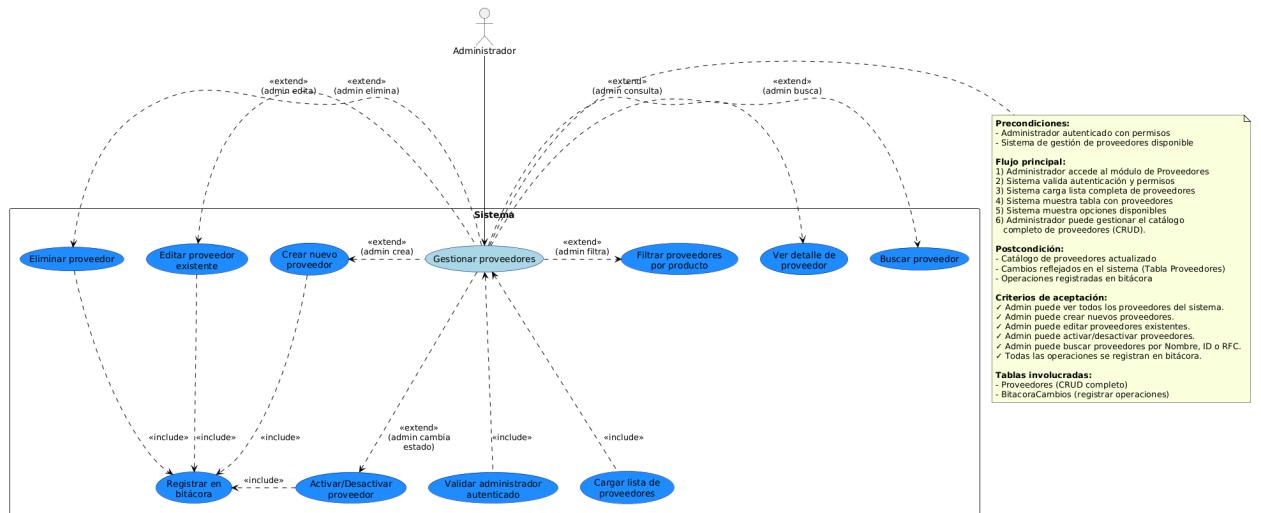
2.3.15 Gestionar Empleados



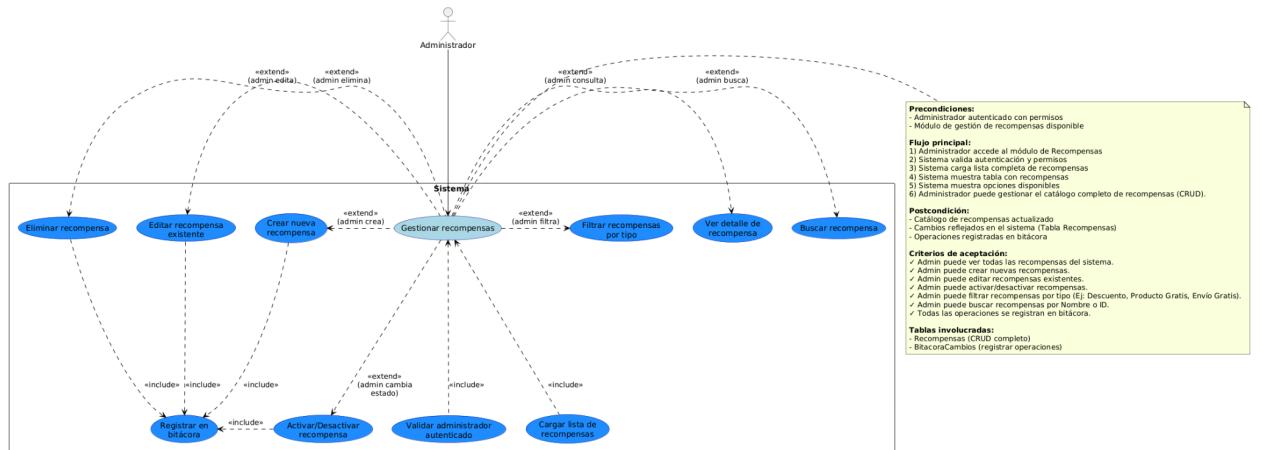
2.3.16 Gestionar Clientes



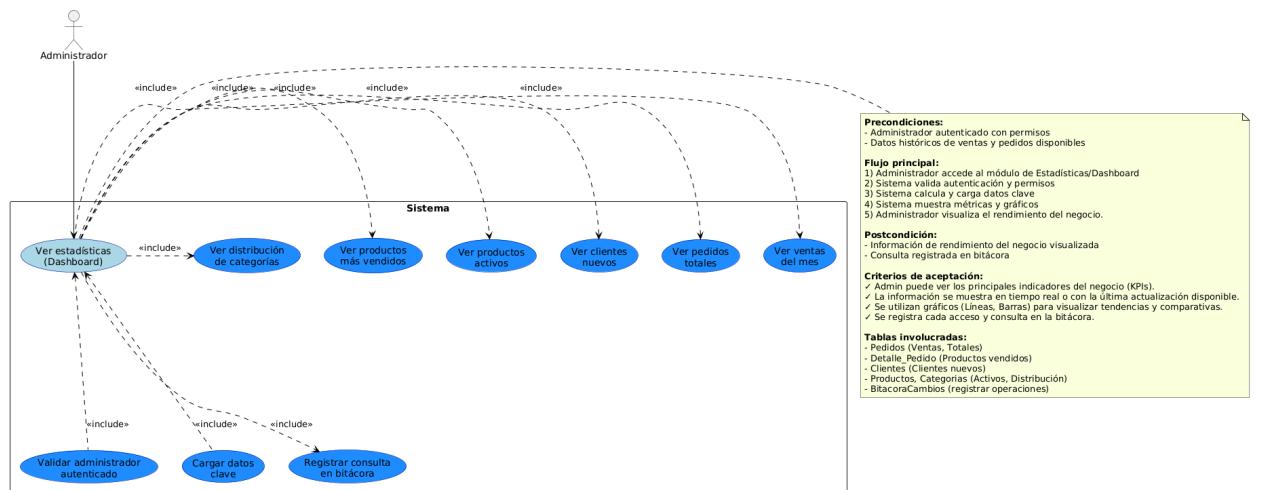
2.3.17 Gestionar Proveedores



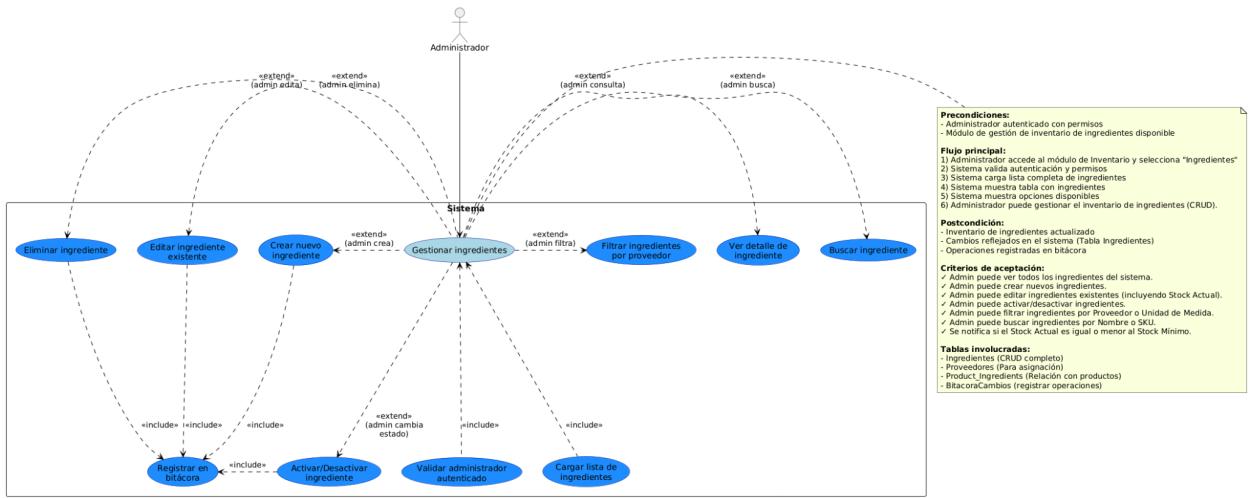
2.3.18 Gestionar Recompensas



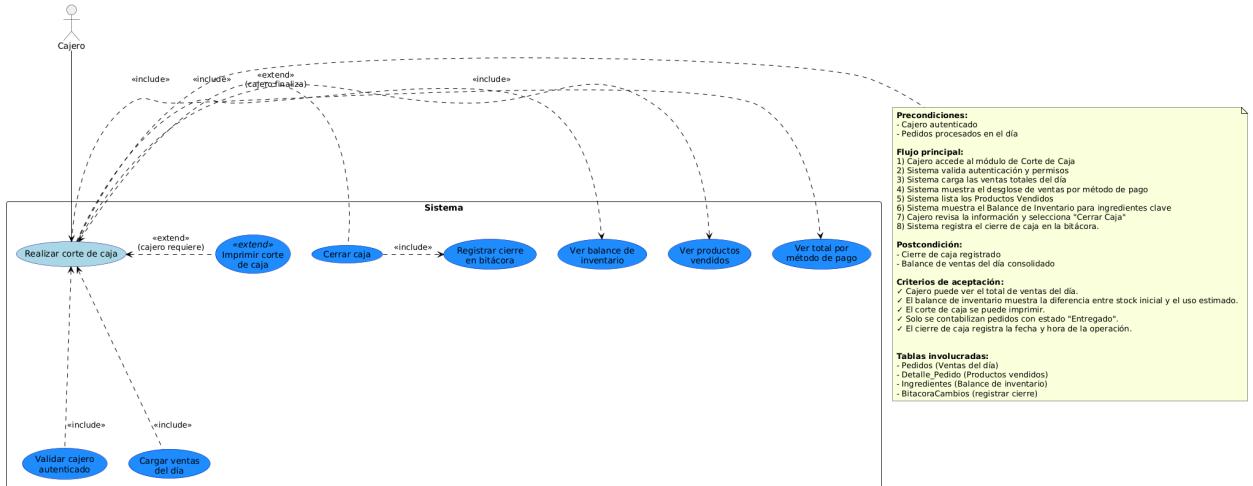
2.3.19 Visualizar estadísticas



2.3.20 Gestionar Inventory de Ingredientes



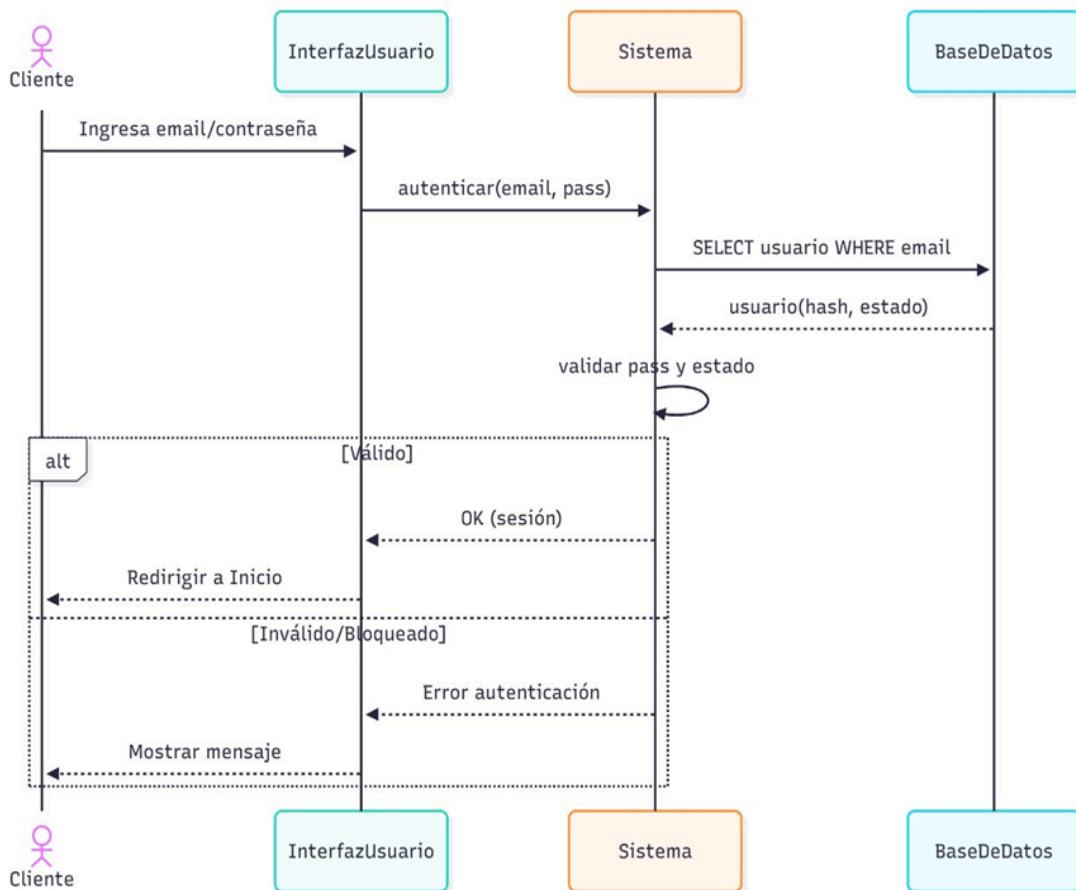
2.3.21 Realizar Corte de Caja



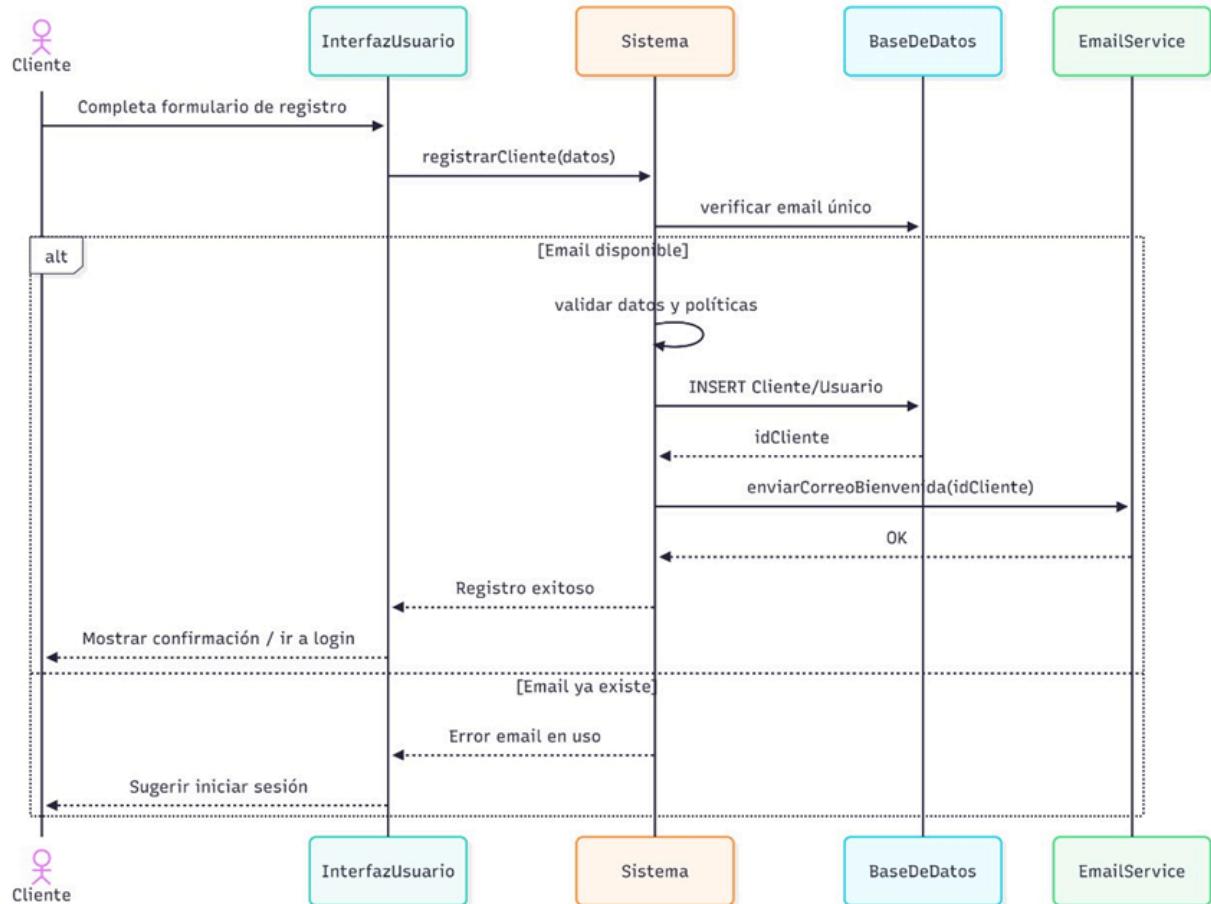
2.4 Diagramas de secuencia

2.4.1 Cliente

LOGIN:

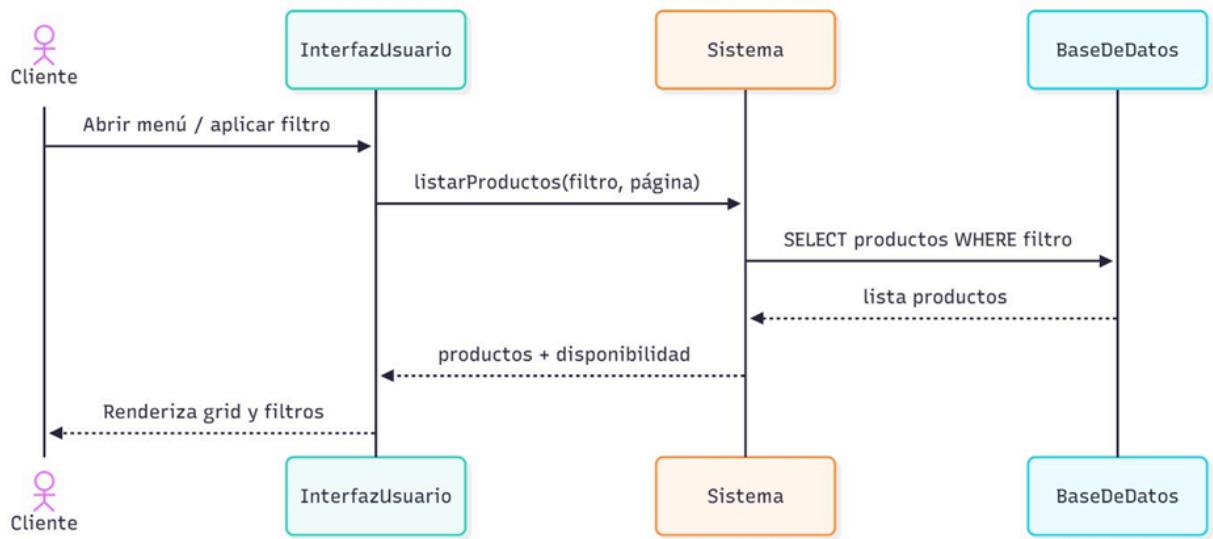


REGISTRO:

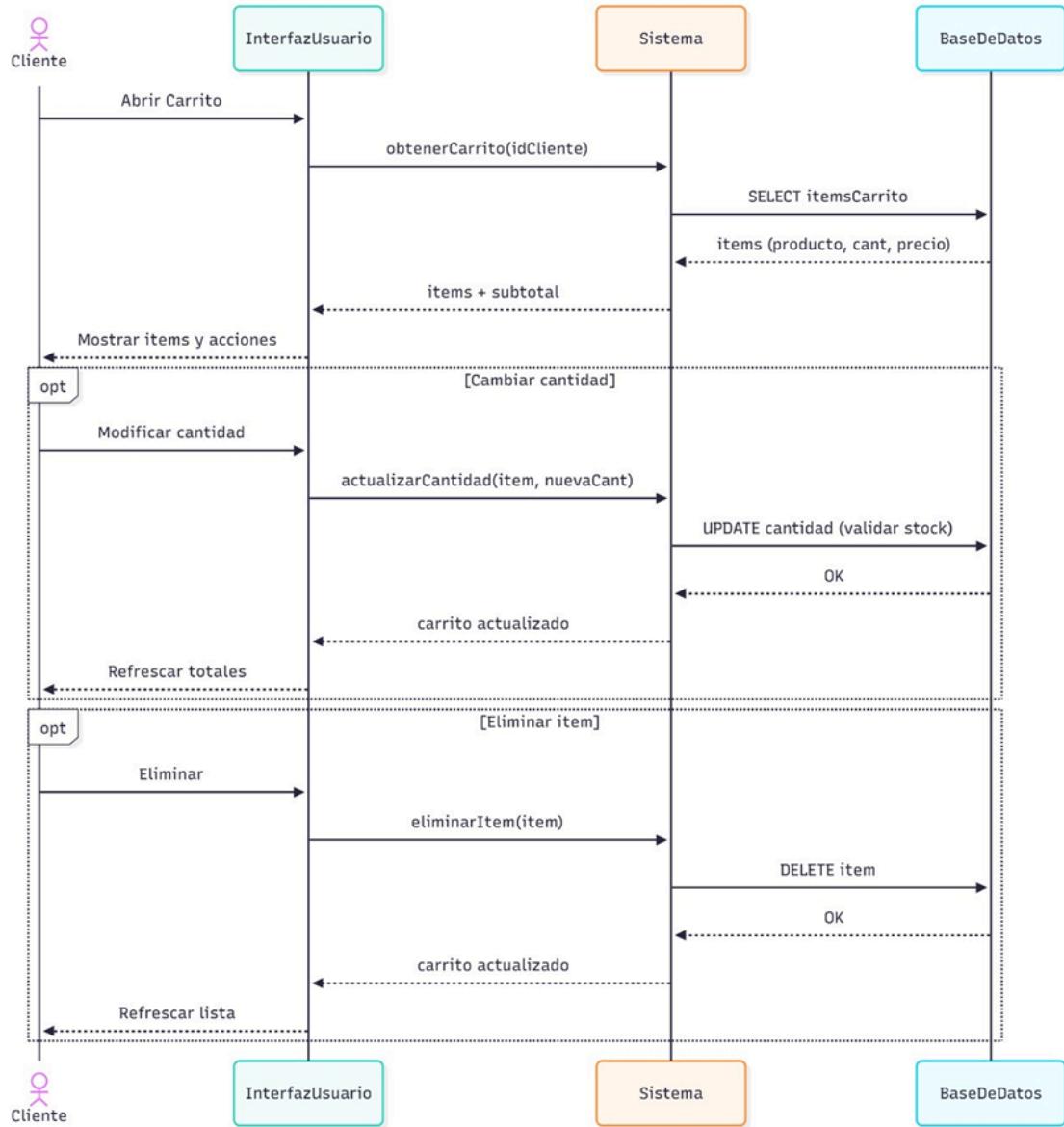


CLIENTE

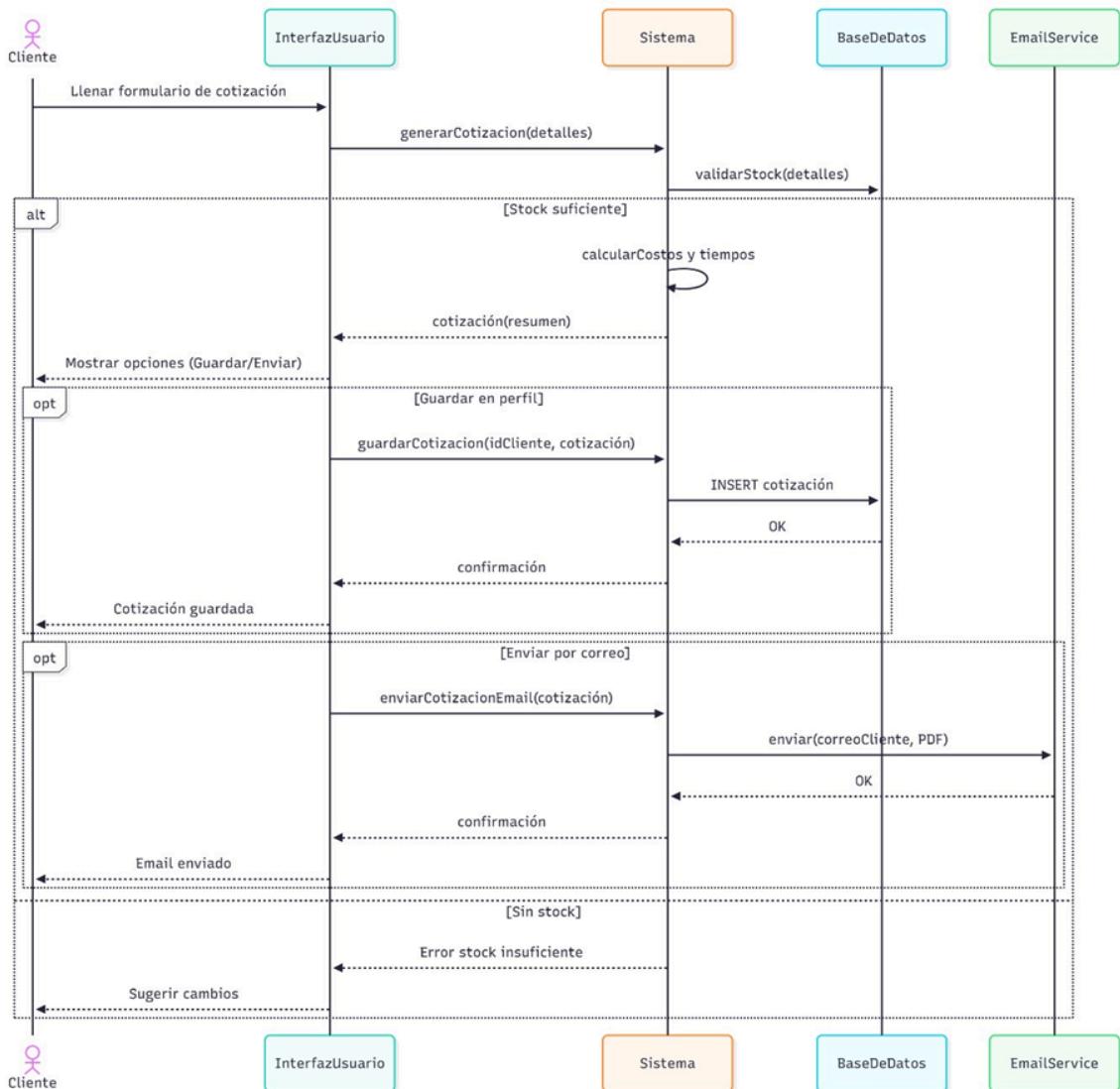
Ver inicio:



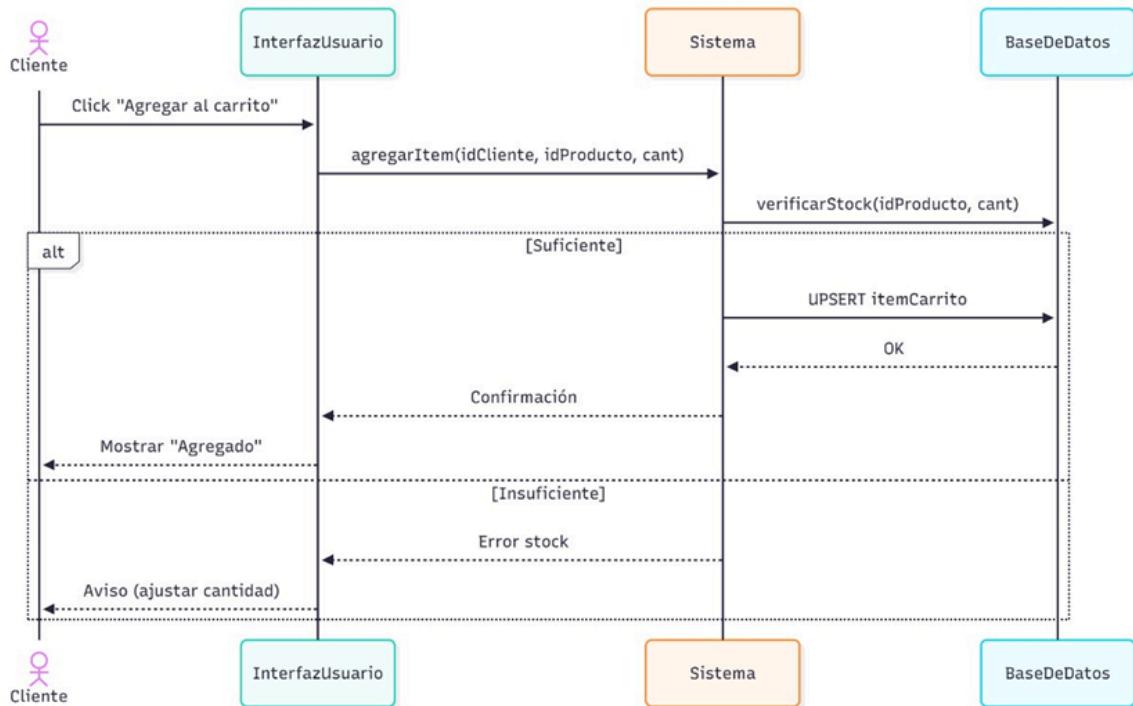
Ver carrito:



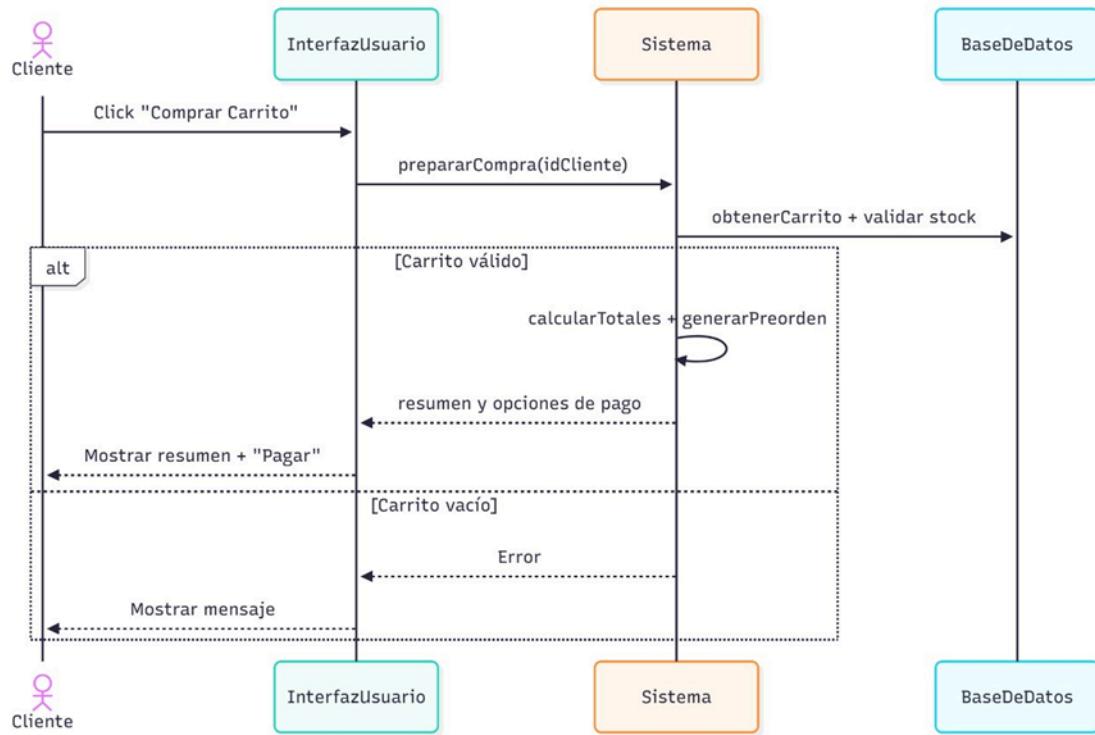
Cotización:



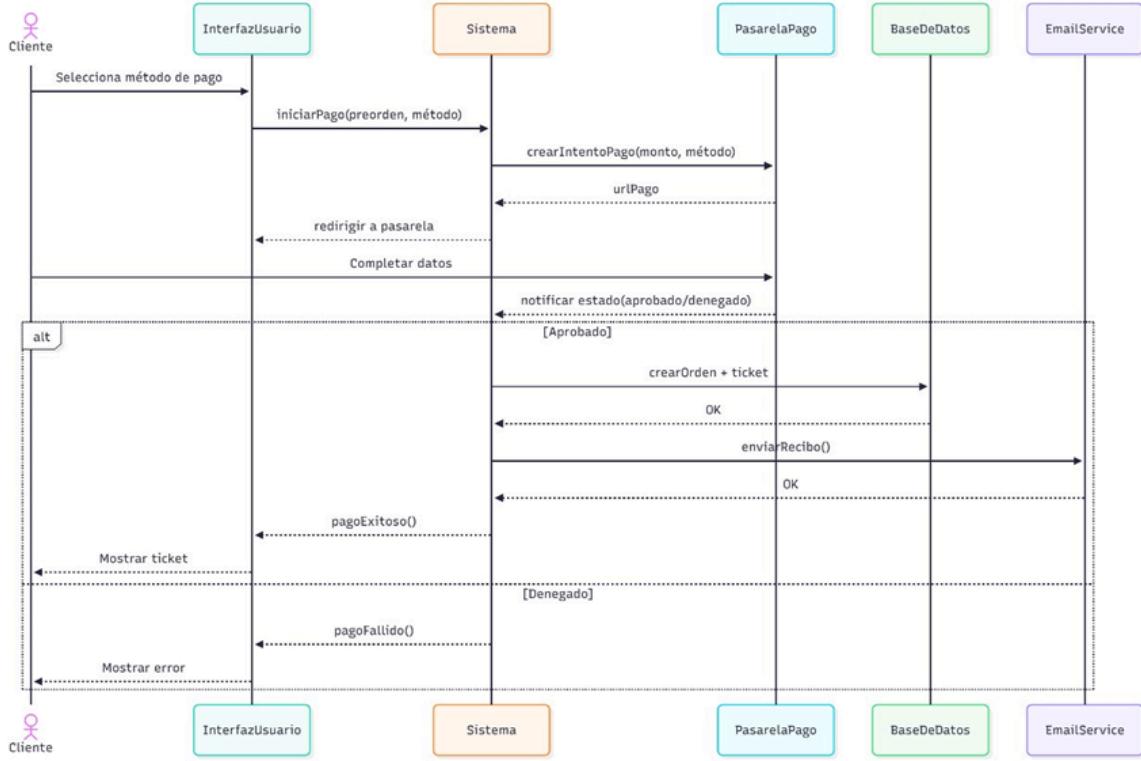
Agregar al carrito:



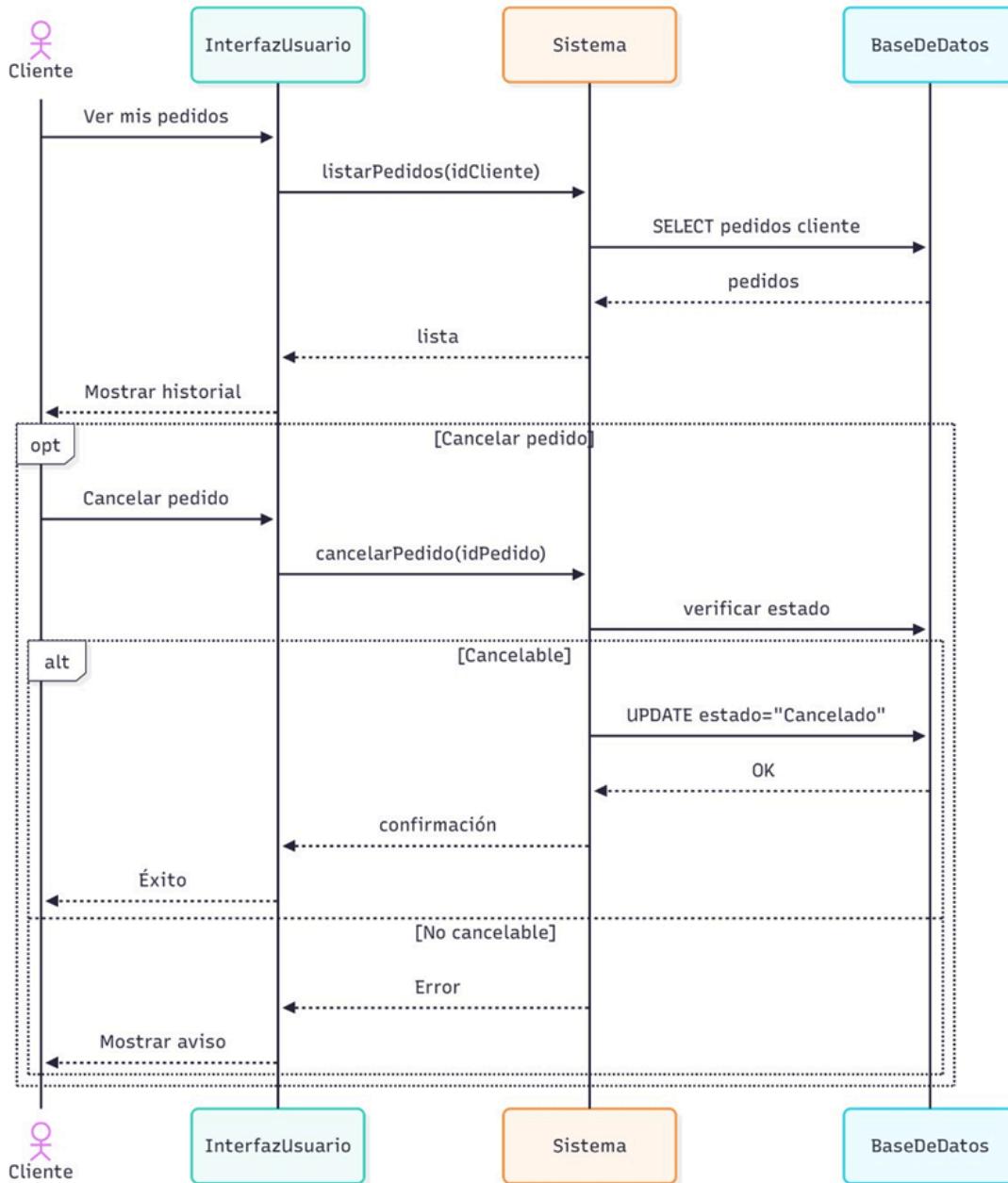
Comprar carrito:



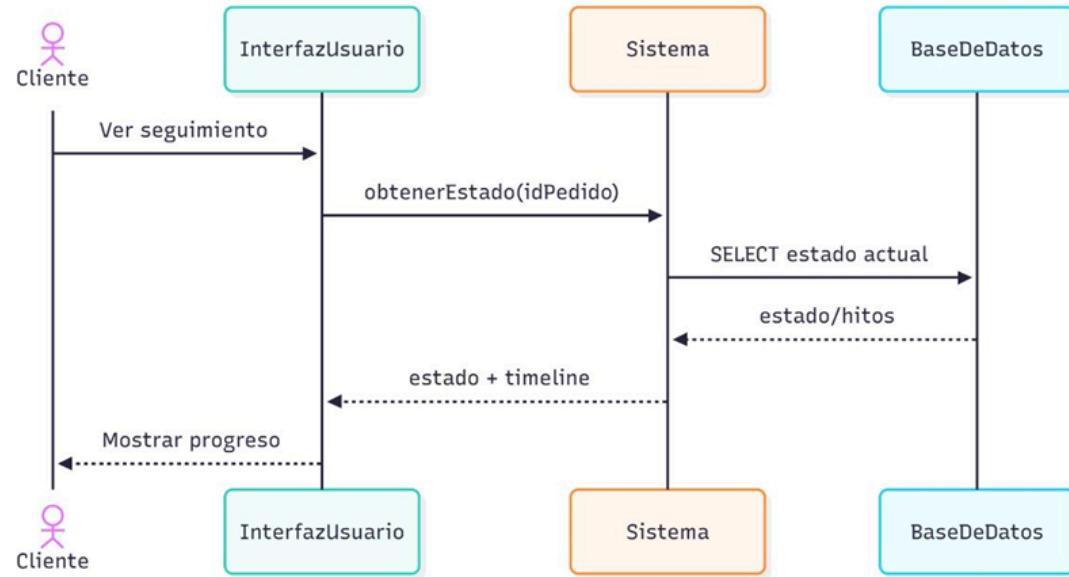
Pago en línea:



Mis pedidos:

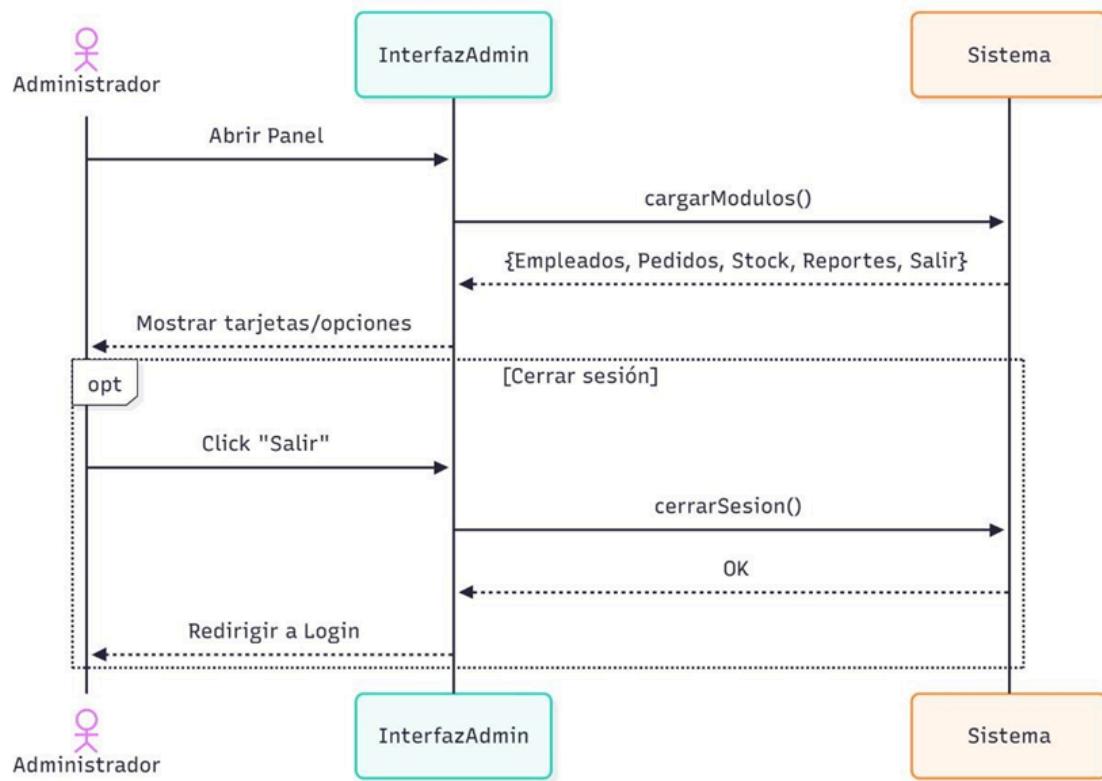


Seguimiento del pedido:

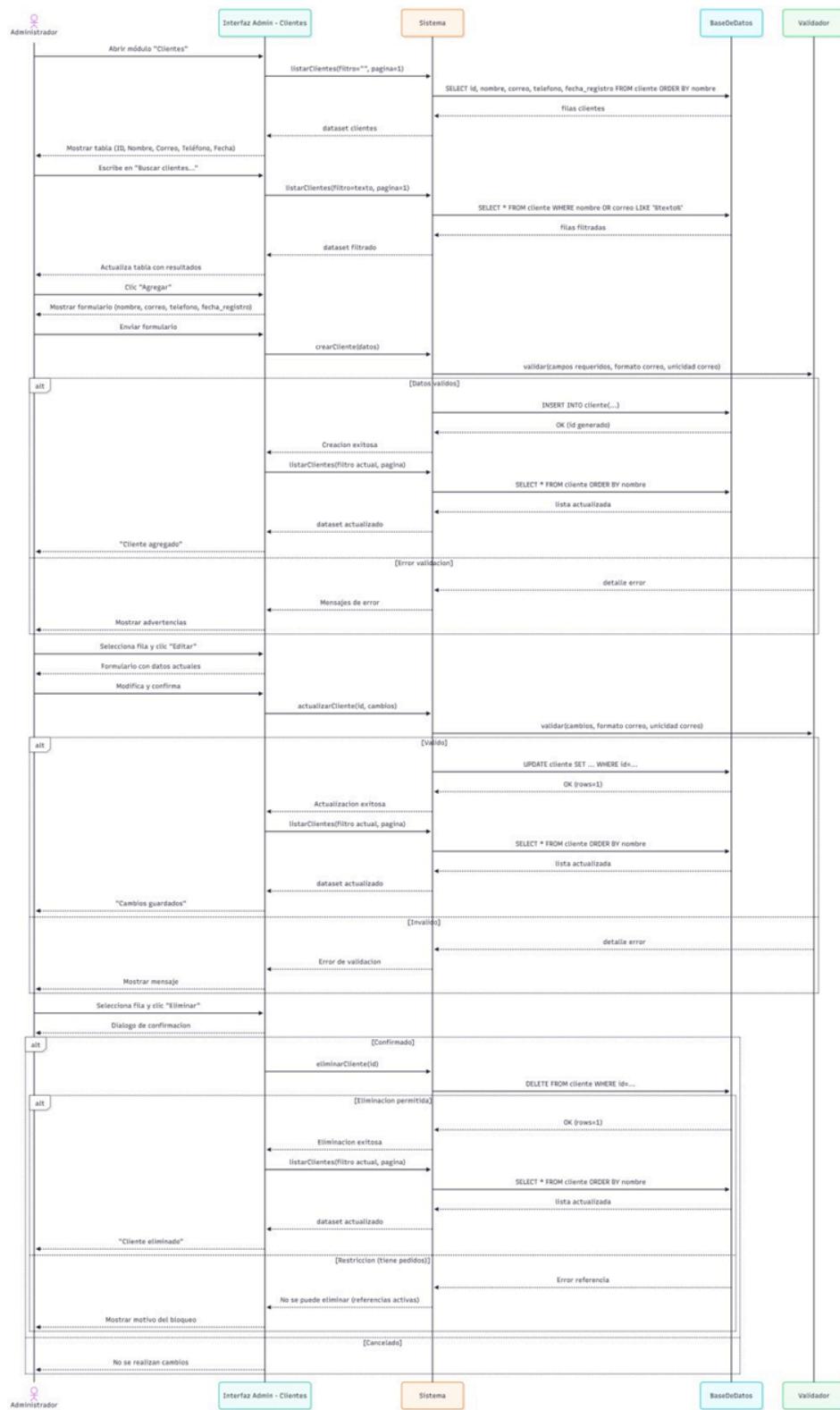


2.4.2 Administrador

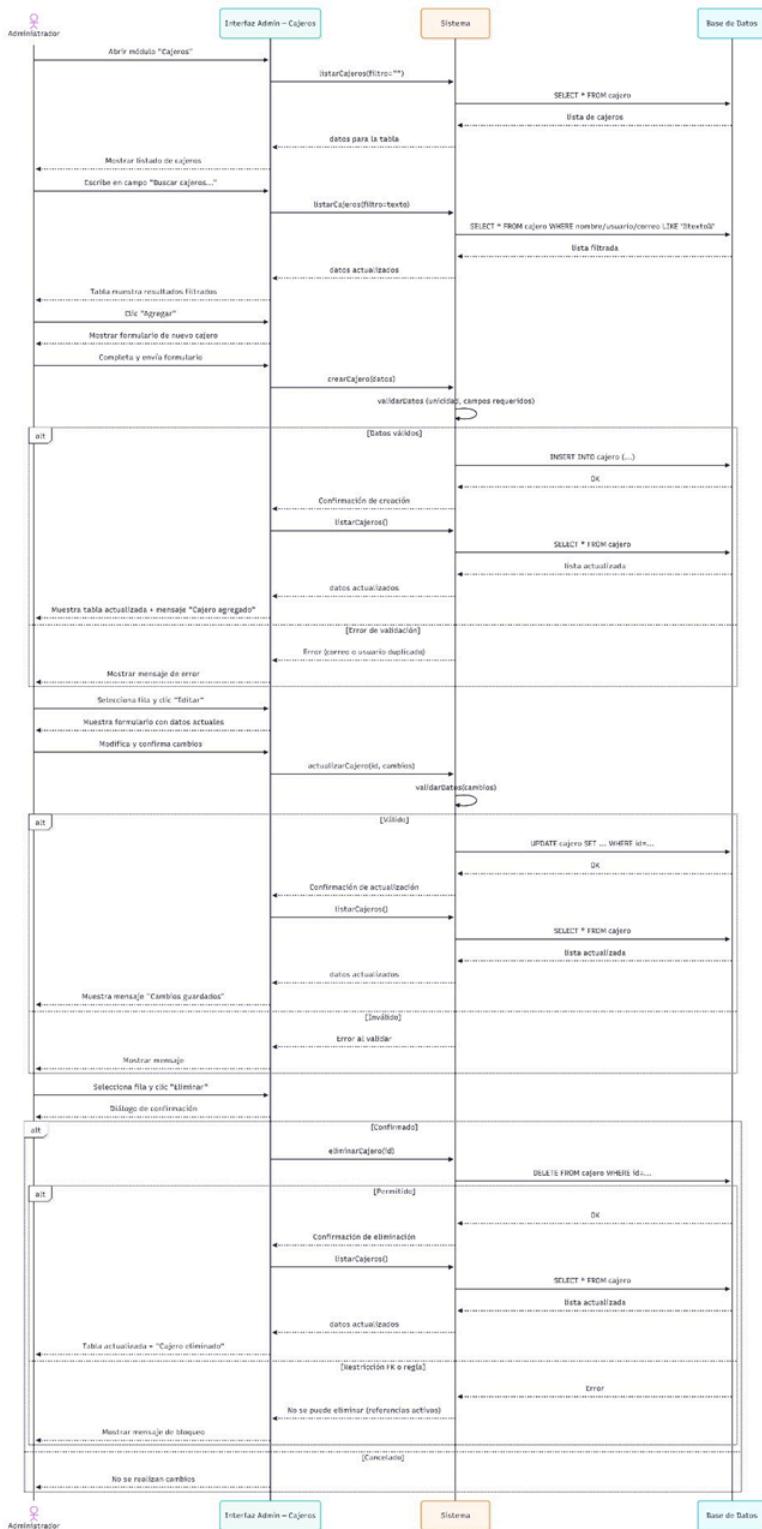
Panel de administrador



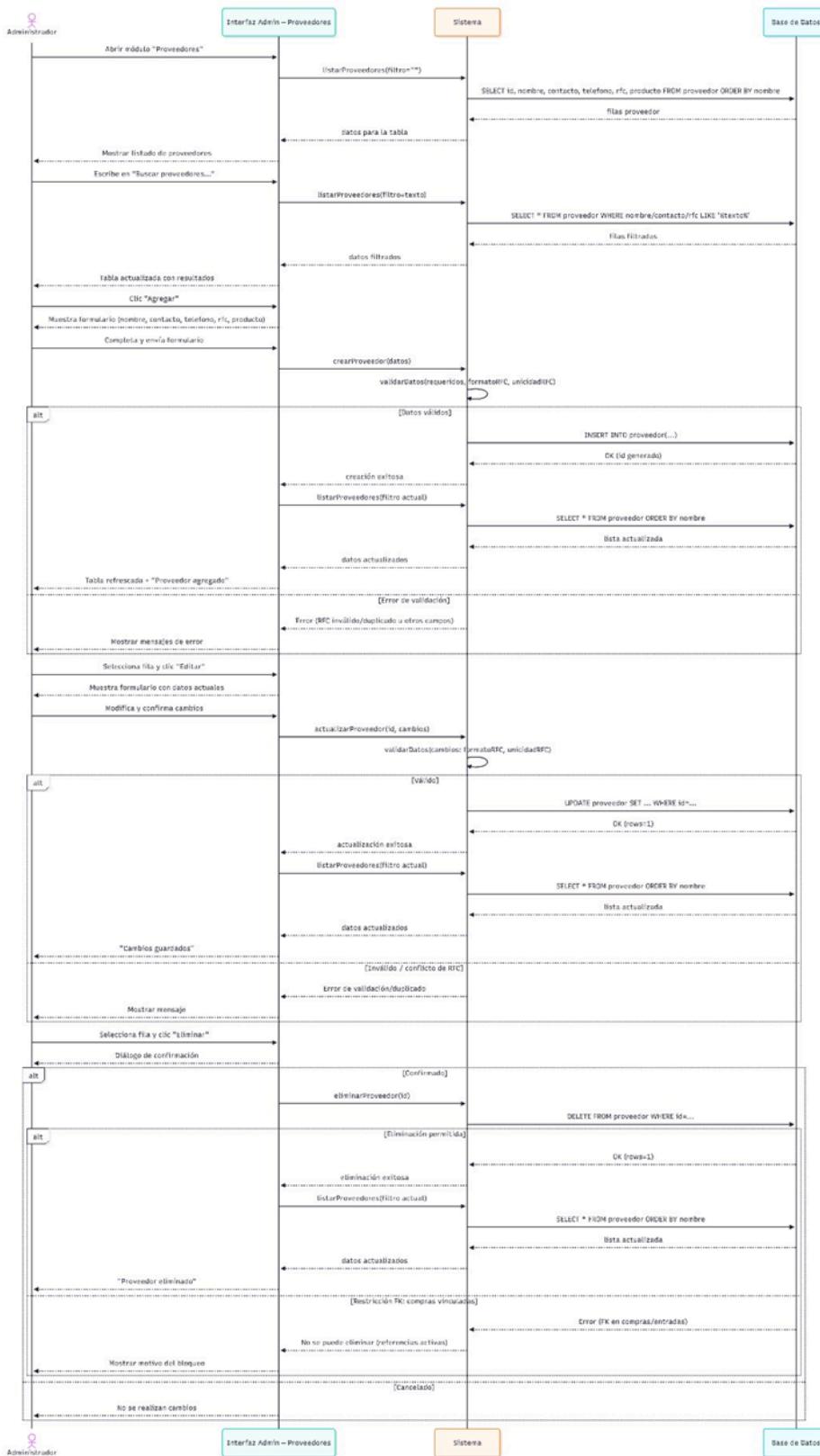
Gestionar clientes



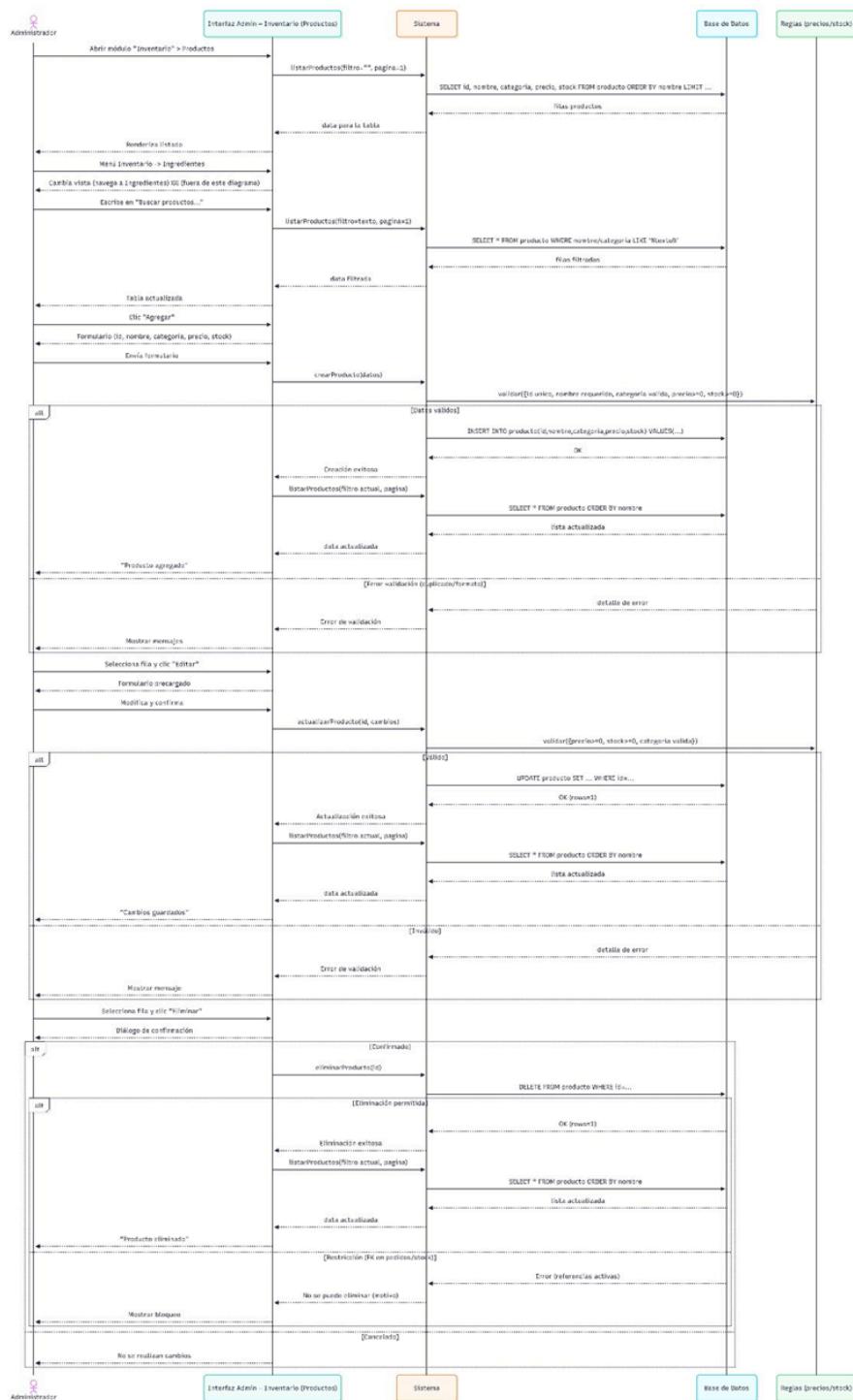
Gestionar cajeros



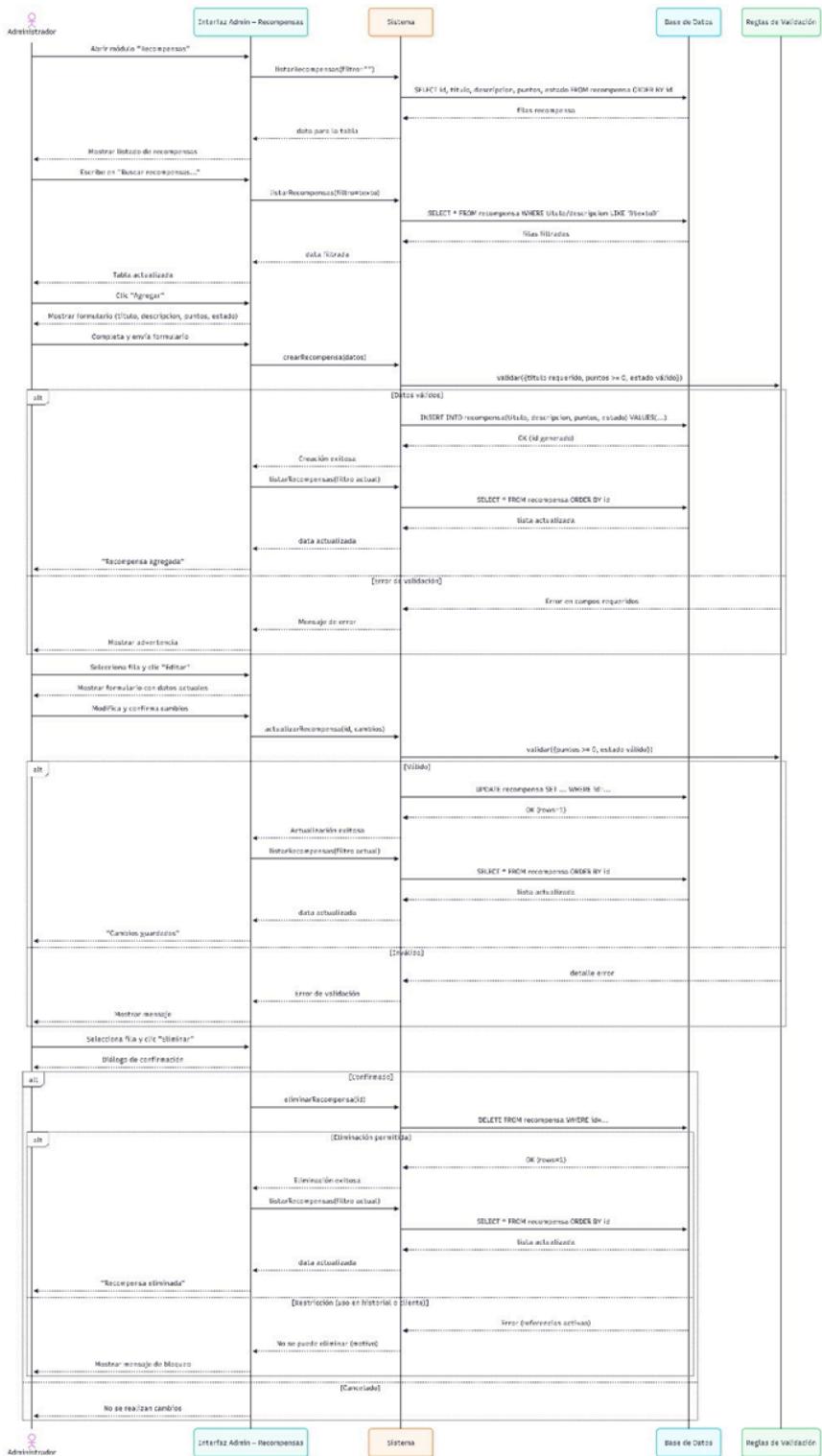
Gestionar Proveedores



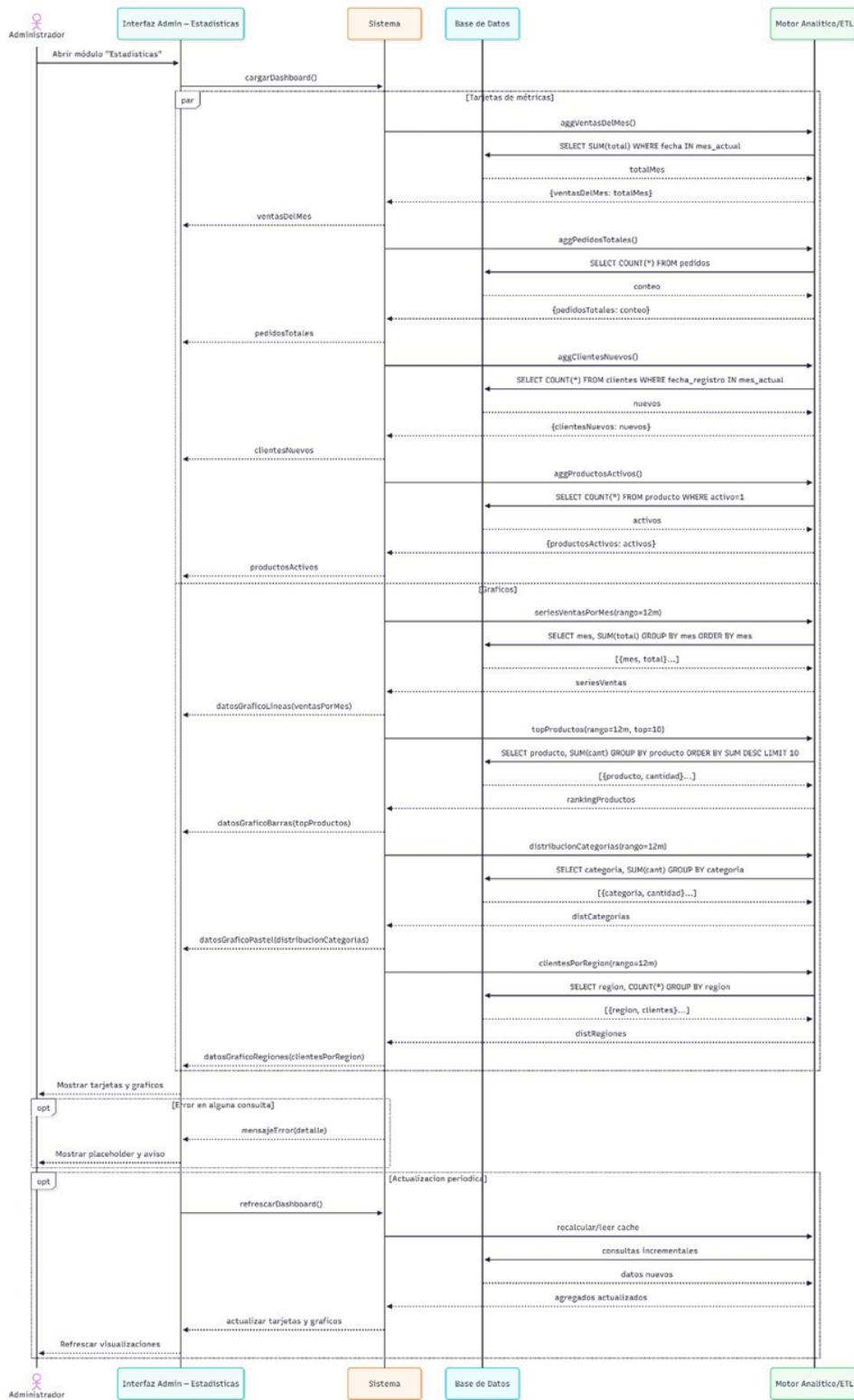
Gestionar inventario:



Gestionar Recompensas

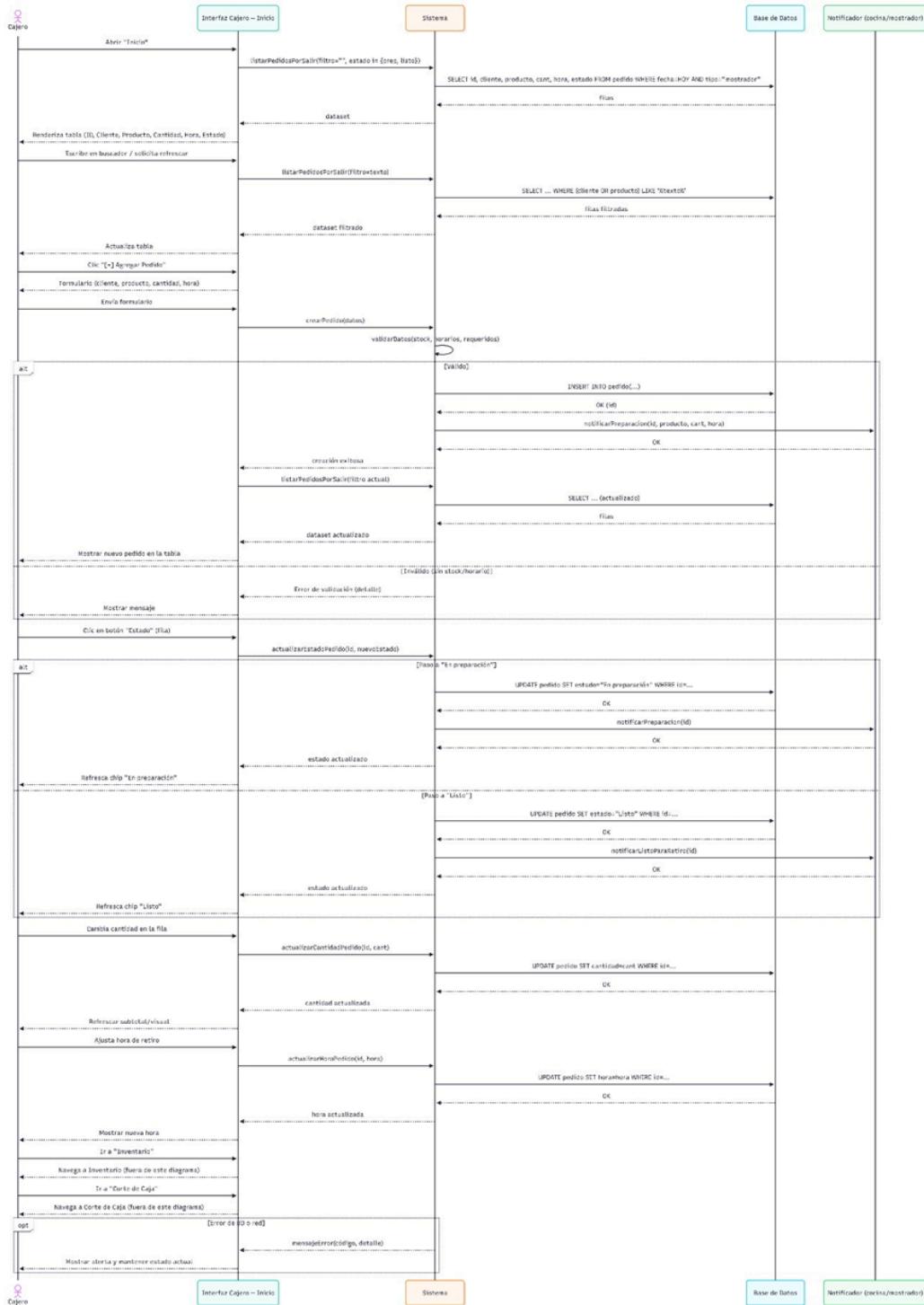


Gestionar estadísticas:

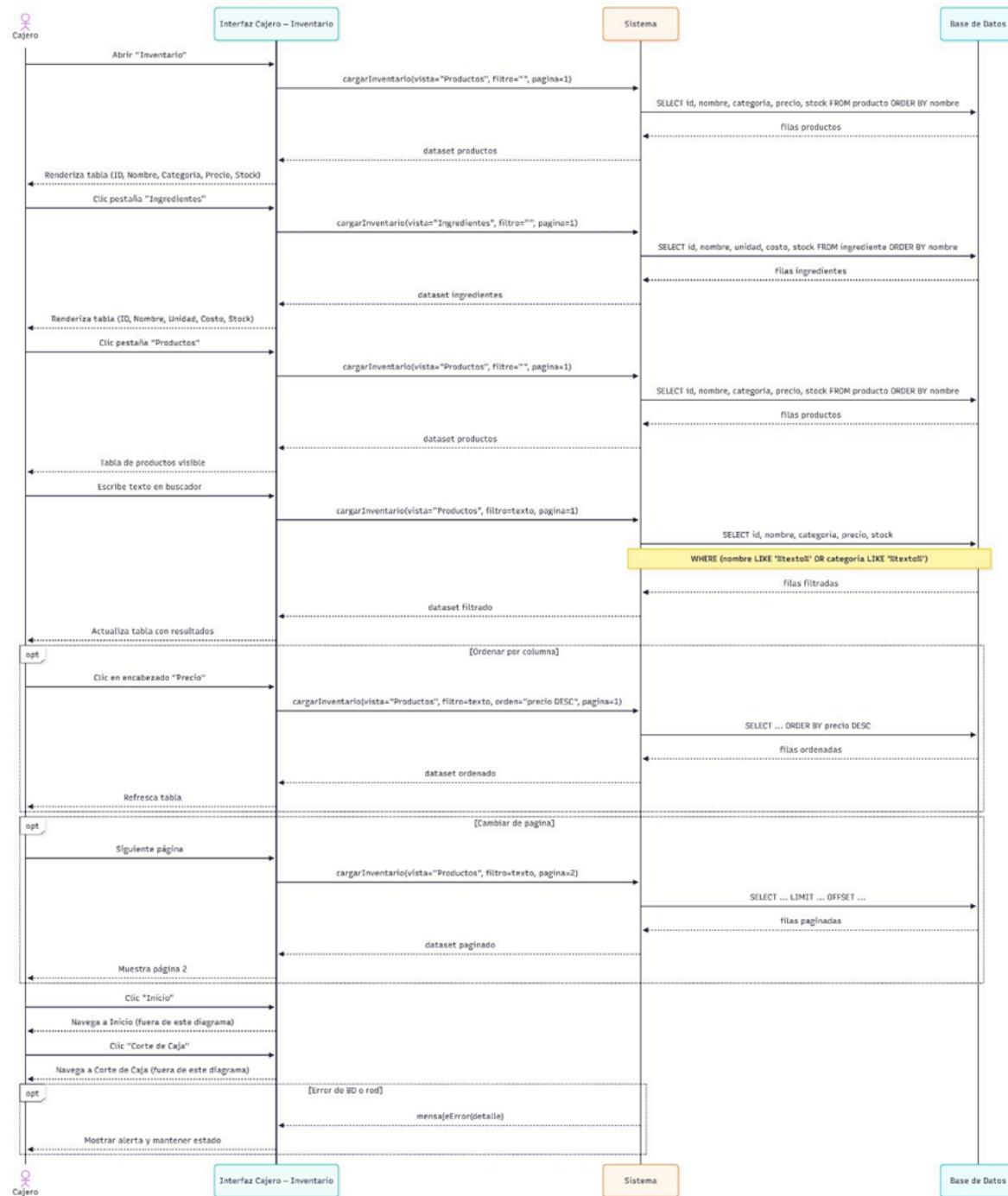


2.4.3 Cajero

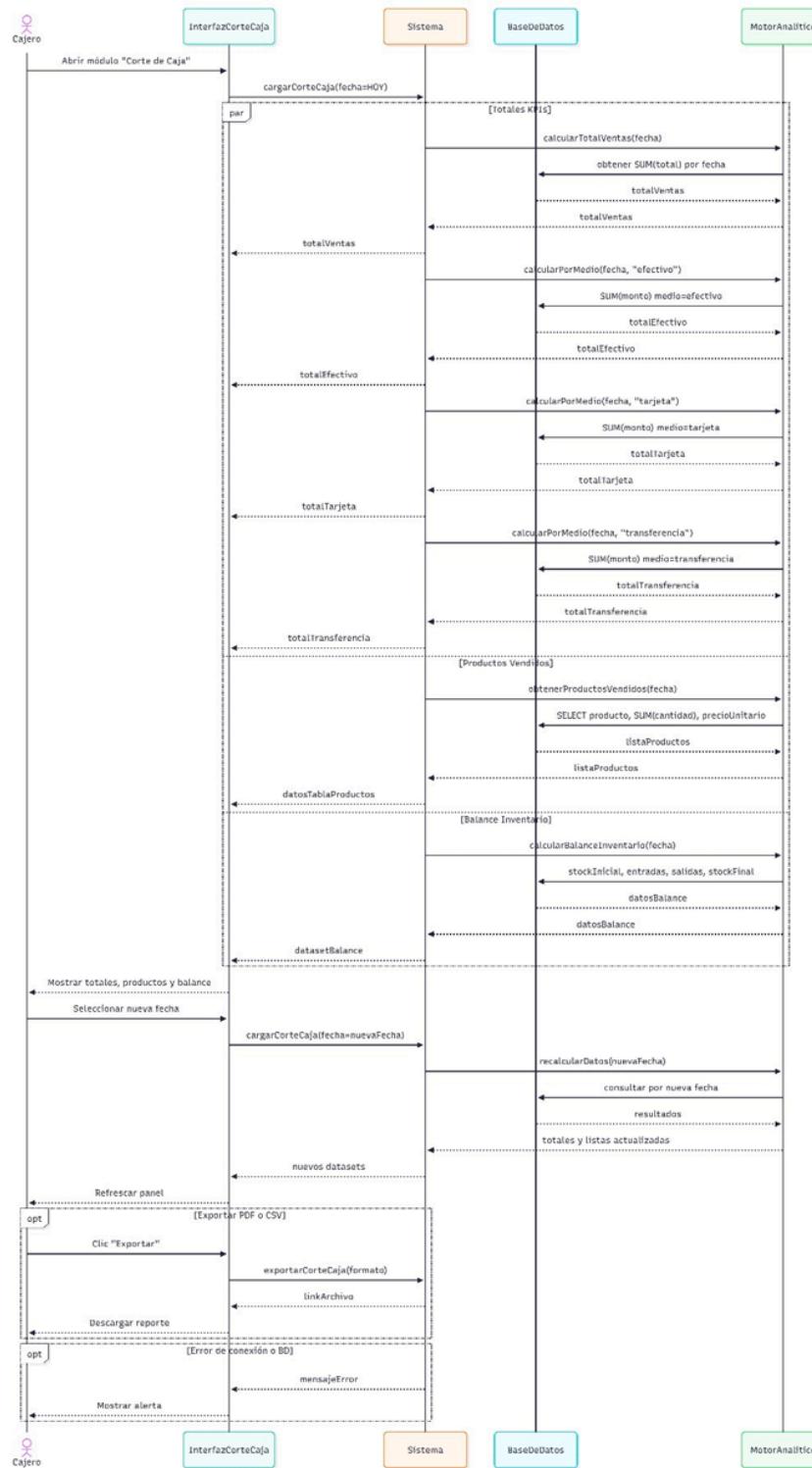
Inicio:



Inventario:

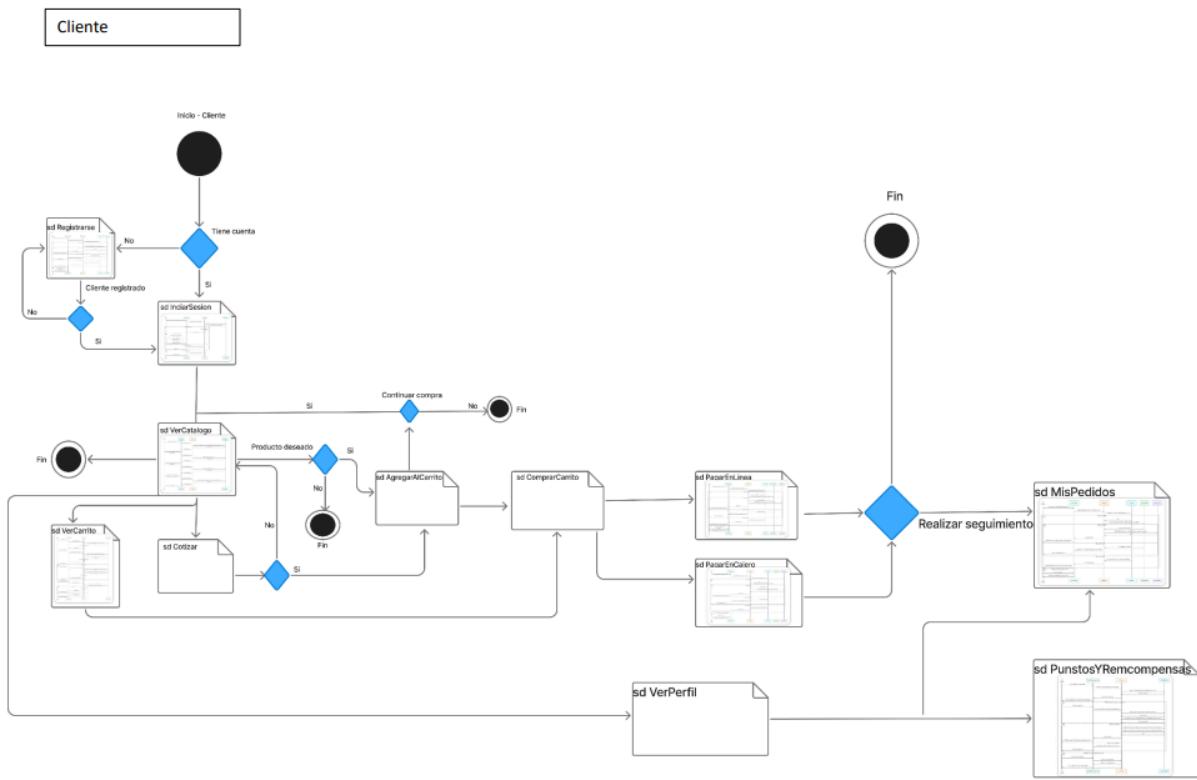


Corte de caja:

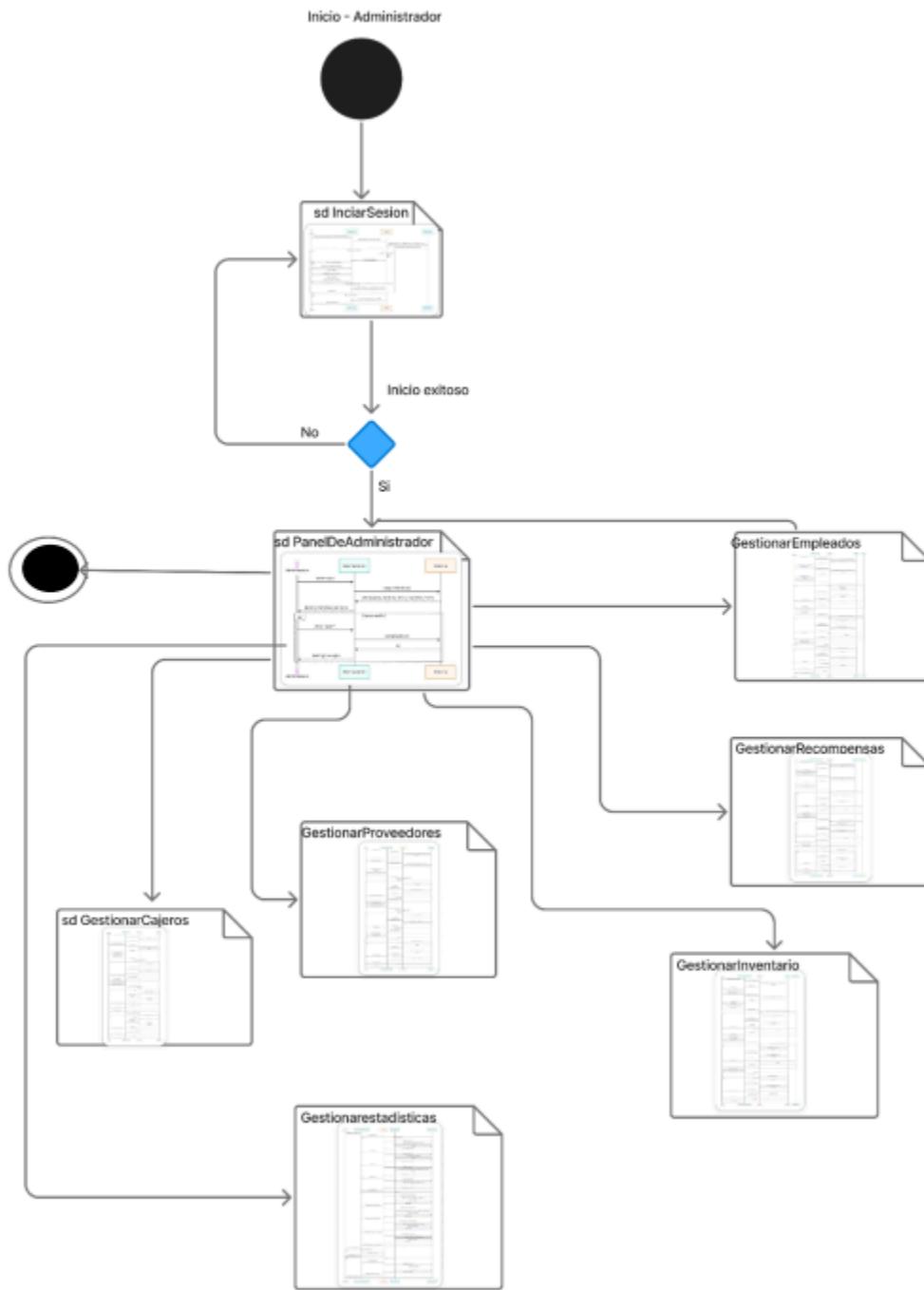


2.5 Diagramas de interacciones

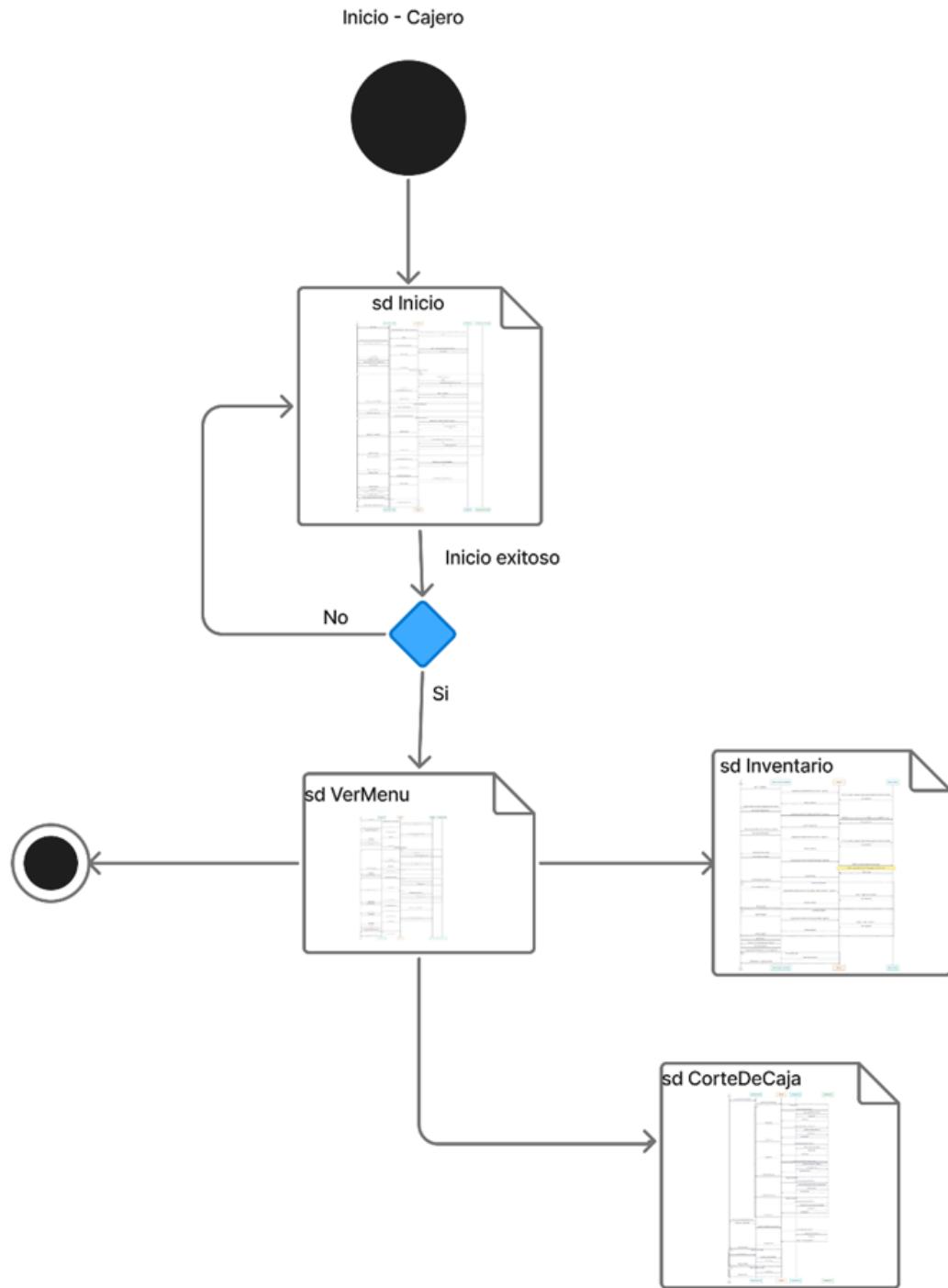
2.5.1 Cliente



2.5.2 Administrador



2.5.3 Cajero



HISTORIAS_USUARIOS

HISTORIAS DE USUARIOS

Cliente

| Historia de Usuario | |
|---|-------------------------------------|
| Número: US01 | Usuario: Usuario Nuevo |
| Nombre historia: Registro de Usuario | |
| Prioridad en negocio: Alta | Riesgo en desarrollo: Bajo |
| Puntos estimados: 3 | Iteración asignada: Sprint 1 |
| Programador responsable: David | |
| Descripción: Como usuario nuevo, quiero registrarme con mi correo para crear una cuenta y acceder a todas las funcionalidades de la aplicación. | |
| Criterios de aceptación: <ul style="list-style-type: none">• El usuario puede ingresar correo electrónico y contraseña• Se valida que el correo no esté registrado previamente• Se envía un email de confirmación• Tras confirmar, se crea la cuenta y se redirige al login• Se muestran mensajes de error claros si hay problemas | |
| Validación: <ul style="list-style-type: none">• Registro exitoso con correo válido• Validación de correo duplicado• Email de confirmación recibido | |

- | |
|--|
| <ul style="list-style-type: none"> • Acceso a la aplicación tras registro |
|--|

| Historia de Usuario | |
|---|--|
| Número: US02 | Usuario: Usuario/Operador/Administrador |
| Nombre historia: Iniciar Sesión | |
| Prioridad en negocio: Crítica | Riesgo en desarrollo: Bajo |
| Puntos estimados: 5 | Iteración asignada: Sprint 1 |
| Programador responsable: Jazmin | |
| Descripción: Como usuario registrado, quiero iniciar sesión con mis credenciales para acceder a las funcionalidades según mi rol (usuario/operador/administrador). | |
| Criterios de aceptación: <ul style="list-style-type: none"> • El usuario ingresa correo y contraseña • Se valida la identidad del usuario • Se redirige según el rol asignado • Se mantiene una sesión activa • Opción de "recordarme" disponible | |
| Validación: <ul style="list-style-type: none"> • Login exitoso con credenciales válidas • Rechazo de credenciales inválidas • Redirección según rol • Sesión mantenida correctamente | |

| Historia de Usuario | |
|---|-------------------------------------|
| Número: US03 | Usuario: Cliente registrado |
| Nombre historia: Ver inicio | |
| Prioridad en negocio: Crítica | Riesgo en desarrollo: Medio |
| Puntos estimados: 5 | Iteración asignada: Sprint 1 |
| Programador responsable: Jazmin | |
| Descripción: Los clientes pueden ver la lista de productos disponibles con imágenes, descripciones y precios, facilitando su proceso de compra. | |
| Criterios de aceptación: <ul style="list-style-type: none"> ❖ Al cargar la página, los productos deben mostrarse con su información completa. ❖ Si se ingresa un término en la barra de búsqueda, sólo deben aparecer los productos coincidentes. ❖ Los filtros deben actualizar los resultados sin recargar la página. ❖ Si no hay productos que coincidan con la búsqueda, se muestra un mensaje adecuado. | |
| Validación: <ul style="list-style-type: none"> ● Login exitoso con credenciales válidas ● Rechazo de credenciales inválidas ● Redirección según rol ● Sesión mantenida correctamente | |

| Historia de Usuario | |
|---|-------------------------------------|
| Número: US04 | Usuario: Cliente registrado |
| Nombre historia: Ver carrito | |
| Prioridad en negocio: bajo | Riesgo en desarrollo: Bajo |
| Puntos estimados: 5 | Iteración asignada: Sprint 2 |
| Programador responsable: Yhudiel | |
| Descripción: Los clientes podrán ver y agregar productos a su carrito de compras, modificar cantidades y eliminarlos antes de realizar un pedido. | |
| Criterios de aceptación: | |
| <ul style="list-style-type: none"> ❖ Implementar un carrito que almacene productos en la sesión del usuario. ❖ Permitir agregar múltiples unidades de un mismo producto. ❖ Implementar un botón para eliminar productos del carrito. ❖ Mostrar el subtotal del pedido en tiempo real. | |
| Validación: | |
| <ul style="list-style-type: none"> ● Si se agrega un producto, debe aparecer en el carrito con su cantidad y precio. ● Al aumentar o disminuir la cantidad de un producto, el subtotal debe actualizarse. ● Si el usuario elimina un producto, este debe desaparecer del carrito | |

Historia de Usuario

Número: US05

Usuario: Cliente registrado

Nombre historia: Pagar en línea

Prioridad en negocio: bajo

Riesgo en desarrollo: Bajo

Puntos estimados: 3

Iteración asignada: Sprint 3

Programador responsable: Yhudiel

Descripción: Los clientes podrán realizar pagos de forma segura utilizando tarjetas de crédito/débito o PayPal, asegurando que el pedido quede confirmado solo si el pago se procesa correctamente.

Criterios de aceptación:

- ❖ Pantalla de pago donde el usuario pueda elegir entre tarjeta o PayPal.
- ❖ Implementar una integración con una API de pagos que permita recibir transacciones y validarlas.
- ❖ Capturar los datos de la tarjeta (número, CVV, fecha de vencimiento) y enviarlos a la API de pagos de forma segura (sin almacenarlos).
- ❖ Para pagos con PayPal, redirigir al usuario a la plataforma de PayPal y recibir la confirmación del pago.
- ❖ Cambiar el estado del pedido a "pagado" si la transacción es exitosa.
- ❖ Enviar un recibo de pago al correo del cliente.
- ❖ Si el pago falla, mostrar un mensaje de error con la opción de reintentar.

Validación:

- Si se introduce un número de tarjeta incorrecto, la API debe rechazar el pago y mostrar un mensaje de error.
- En caso de éxito, el cliente debe recibir un correo con la confirmación del pago.
- Si el usuario abandona la página antes de completar el pago, el pedido debe permanecer en estado "pendiente de pago".

| Historia de Usuario | |
|---|-------------------------------------|
| Número: US06 | Usuario: Cliente registrado |
| Nombre historia: Pago en cajero | |
| Prioridad en negocio: Media | Riesgo en desarrollo: Medio |
| Puntos estimados: 8 | Iteración asignada: Sprint 2 |
| Programador responsable: Jazmin | |
| Descripción: Como cliente registrado, quiero tener la opción de realizar el pago de mis pedidos en un cajero autorizado, para poder completar mi compra. | |
| Criterios de aceptación: <ul style="list-style-type: none"> ❖ El sistema deberá generar un comprobante o referencia de pago única para cada pedido. ❖ El cliente podrá seleccionar “Pago en cajero” como método de pago durante el proceso de compra. ❖ Una vez realizado el pago, el sistema deberá actualizar automáticamente el estado del pedido a “Pagado”. ❖ Si el pago no se confirma en un tiempo determinado (por ejemplo, 24 horas), el pedido deberá cambiar su estado a “Cancelado por falta de pago”. <ul style="list-style-type: none"> ● El cliente debe poder visualizar el comprobante o recibo del pago dentro de su historial de pedidos. ● Si ocurre un error durante la generación de la referencia o la actualización del estado, el sistema debe mostrar un mensaje de error: “No se pudo generar la referencia, intenta nuevamente más tarde.” | |

| Historia de Usuario | |
|---|-------------------------------------|
| Número: US07 | Usuario: Cliente registrado |
| Nombre historia: Mis pedidos | |
| Prioridad en negocio: bajo | Riesgo en desarrollo: Medio |
| Puntos estimados: 5 | Iteración asignada: Sprint 2 |
| Programador responsable: Tania | |
| <p>Descripción: Como cliente registrado, quiero poder visualizar el historial de mis pedidos realizados dentro del sistema, para poder consultar el estado, la fecha y los detalles de cada compra que he hecho.</p> | |
| Criterios de aceptación: <ul style="list-style-type: none"> ❖ El cliente podrá acceder a la sección “Mis pedidos” desde su perfil. ❖ El sistema mostrará una lista con todos los pedidos realizados por el cliente. ❖ Cada pedido deberá incluir información como: número de pedido, fecha, estado, total y productos adquiridos. ❖ El cliente podrá filtrar o buscar pedidos por fecha o estado. ❖ Si el cliente no tiene pedidos registrados, el sistema mostrará un mensaje informativo (“No tienes pedidos registrados aún”). | |

Validación:

- El cliente registrado selecciona la opción “Mis pedidos” desde su perfil y el sistema abre la página con la lista de pedidos correspondientes.
- Cuando el cliente selecciona un pedido específico, el sistema muestra los detalles completos de dicho pedido, incluyendo productos, precios y estado.
- Si ocurre una falla en la carga de datos, el sistema muestra el mensaje “No se pudieron cargar los pedidos, intenta más tarde.”

| Historia de Usuario | |
|---|-------------------------------------|
| Número: US08 | Usuario: Cliente registrado |
| Nombre historia: Cotizar un producto | |
| Prioridad en negocio: Alto | Riesgo en desarrollo: Medio |
| Puntos estimados: 3 | Iteración asignada: Sprint 1 |
| Programador responsable: Tania | |
| Descripción: Los clientes podrán personalizar su pastel eligiendo sabores, tamaños, decoraciones y rellenos, obteniendo un precio estimado antes de confirmar su compra. | |
| Criterios de aceptación: <ul style="list-style-type: none"> ❖ Formulario dinámico donde el cliente pueda elegir opciones personalizadas. ❖ Implementar un sistema que calcule automáticamente el precio en función de las opciones seleccionadas. ❖ Mostrar el desglose del precio antes de que el cliente confirme la personalización. ❖ Enviar la cotización al correo del cliente si lo solicita. | |
| Validación: <ul style="list-style-type: none"> ● Al modificar una opción del pastel, el precio estimado debe actualizarse de inmediato. ● Si se intentan elegir combinaciones inválidas, el sistema debe bloquearlas o advertir al usuario. ● La cotización generada debe reflejar con precisión los parámetros ingresados por el cliente. | |

| Historia de Usuario | |
|--|-------------------------------------|
| Número: US09 | Usuario: Cliente registrado |
| Nombre historia: Comprar el carrito | |
| Prioridad en negocio: Alto | Riesgo en desarrollo: Alto |
| Puntos estimados: 3 | Iteración asignada: Sprint 2 |
| Programador responsable: David | |
| Descripción: Los clientes podrán finalizar su compra confirmando el pedido de los productos en su carrito. | |
| Criterios de aceptación: <ul style="list-style-type: none"> ❖ Implementar un botón "Realizar pedido" en la pantalla del carrito. ❖ Verificar que el carrito tenga al menos un producto antes de permitir la confirmación. ❖ Generar un número de pedido único para cada transacción. ❖ Almacenar el pedido en la base de datos con el estado "pendiente". ❖ Enviar notificación del pedido al cliente y administrador. | |
| Validación: <ul style="list-style-type: none"> ● El carrito está vacío, el botón de "Realizar pedido" debe estar deshabilitado. ● Al confirmar un pedido, este debe registrarse en la base de datos con su número único. ● El cliente debe recibir un correo de confirmación con los detalles del pedido. ● El administrador debe ver el nuevo pedido reflejado en su panel de control. | |

ADMINISTRADOR

| Historia de Usuario | |
|---|-------------------------------------|
| Número: US010 | Usuario: Administrador |
| Nombre historia: Iniciar sesión | |
| Prioridad en negocio: Alta | Riesgo en desarrollo: Medio |
| Puntos estimados: 3 | Iteración asignada: Sprint 1 |
| Programador responsable: Jazmín | |
| Descripción: El administrador podrá acceder al sistema mediante un formulario de inicio de sesión, proporcionando sus credenciales (correo y contraseña). El sistema deberá validar la información ingresada y, en caso de ser correcta, permitir el acceso al panel principal de administración. Si las credenciales son incorrectas o el campo está vacío, se mostrará un mensaje de error sin permitir el ingreso. | |
| Criterios de aceptación: | |
| <ul style="list-style-type: none">❖ El sistema debe mostrar una pantalla de inicio de sesión con los campos “Correo” y “Contraseña”.❖ El botón “Iniciar sesión” debe permanecer deshabilitado hasta que ambos campos sean completados.❖ Al presionar el botón, el sistema debe validar las credenciales en la base de datos.❖ Si las credenciales son válidas, el sistema debe redirigir al panel de administrador.❖ Si las credenciales son incorrectas, se debe mostrar un mensaje de error claro y descriptivo.❖ La contraseña debe ser encriptada y protegida durante la verificación. | |

Validación:

- Si los campos están vacíos, el sistema debe mostrar un mensaje de advertencia y no permitir el inicio de sesión.
- Si el correo no existe o la contraseña es incorrecta, el sistema debe indicar: “Credenciales inválidas. Intente nuevamente.”
- El inicio de sesión exitoso debe redirigir al administrador a la pantalla del panel principal.
- El sistema debe registrar la fecha, hora y dirección IP del inicio de sesión en la base de datos.
- Tras un inicio de sesión exitoso, el administrador podrá acceder a los módulos de Empleados, Pedidos, Stock, Reportes y Salir.

| | |
|--|-------------------------------------|
| Número: US011 | Usuario: Administrador |
| Nombre historia: Acceder y gestionar el panel de administración | |
| Prioridad en negocio: Alta | Riesgo en desarrollo: Medio |
| Puntos estimados: 3 | Iteración asignada: Sprint 2 |
| Programador responsable: David | |
| <p>Descripción: El administrador podrá acceder al panel principal del sistema para visualizar y gestionar los diferentes módulos disponibles, como empleados, pedidos, stock y reportes. Además, tendrá la posibilidad de cerrar sesión de forma segura desde la interfaz principal, siendo redirigido automáticamente a la pantalla de inicio de sesión.</p> | |
| <p>Criterios de aceptación:</p> <ul style="list-style-type: none"> ❖ El sistema debe permitir que el administrador abra el panel principal tras iniciar sesión correctamente. ❖ Al ingresar, el sistema debe cargar los módulos disponibles: Empleados, Pedidos, Stock, Reportes y Salir. ❖ El administrador podrá interactuar con las tarjetas/opciones visibles en el panel. ❖ Al seleccionar la opción “Salir”, el sistema debe cerrar la sesión actual y redirigir al inicio de sesión. ❖ El cierre de sesión debe invalidar cualquier sesión activa anterior para garantizar la seguridad. | |

Validación:

- Si el administrador no tiene una sesión activa, no podrá acceder al panel (será redirigido a la pantalla de login).
- Al abrir el panel, los módulos deben visualizarse correctamente en la interfaz.
- El botón “Salir” debe ejecutar el proceso cerrarSesion() y mostrar un mensaje de confirmación o redirección al login.
- Despues de cerrar sesión, el administrador no debe poder regresar al panel utilizando el botón “Atrás” del navegador.
- El sistema debe registrar el evento de cierre de sesión en el log del sistema o bitácora de seguridad.

| | |
|---|-------------------------------------|
| Número: US012 | Usuario: Administrador |
| Nombre historia: Gestionar clientes | |
| Prioridad en negocio: Alta | Riesgo en desarrollo: Alto |
| Puntos estimados: 5 | Iteración asignada: Sprint 3 |
| Programador responsable: Tania | |
| <p>Descripción: El administrador podrá gestionar la información de los clientes registrados en el sistema desde el panel administrativo.</p> <p>Esta funcionalidad permitirá registrar nuevos clientes, actualizar datos existentes (como nombre, dirección, correo, teléfono, etc.) y eliminar registros obsoletos, asegurando que toda la información sea consistente, validada y almacenada correctamente en la base de datos.</p> | |
| <p>Criterios de aceptación:</p> <ul style="list-style-type: none"> ❖ El sistema debe mostrar un listado de clientes registrados con sus datos básicos. ❖ El administrador podrá agregar nuevos clientes mediante un formulario que solicite la información requerida (nombre, correo, dirección, teléfono). ❖ Antes de registrar un cliente, el sistema debe verificar que los campos obligatorios no estén vacíos. ❖ Al confirmar el registro, el sistema debe insertar los datos en la base de datos (INSERT INTO clientes) y mostrar un mensaje de confirmación. ❖ El administrador podrá editar los datos de un cliente existente, los cuales se actualizarán en la base de datos- ❖ Si el administrador selecciona “Eliminar”, el sistema deberá solicitar confirmación previa antes de ejecutar la eliminación. | |

- ❖ Cada operación debe mostrar un mensaje claro indicando si la acción fue exitosa o si hubo error.
- ❖ La interfaz debe actualizar la lista de clientes automáticamente después de cada cambio (alta, modificación o eliminación).

Validación:

- Si algún campo obligatorio está vacío al registrar o editar un cliente, el sistema debe mostrar una advertencia y evitar la operación.
- Al registrar un nuevo cliente, debe aparecer en la lista actualizada y guardarse correctamente en la base de datos.
- Si se modifica la información de un cliente, los cambios deben visualizarse inmediatamente tras la confirmación.
- Al eliminar un cliente, este debe desaparecer del listado y eliminarse de la base de datos sin afectar otros registros.
- El sistema debe evitar registros duplicados mediante validación del correo electrónico o identificador único.
- Todas las operaciones deben registrarse en la bitácora del sistema indicando la acción

| Historia de Usuario | |
|--|-------------------------------------|
| Número: US013 | Usuario: Administrador |
| Nombre historia: Gestionar cajeros | |
| Prioridad en negocio: Alta | Riesgo en desarrollo: Alto |
| Puntos estimados: 5 | Iteración asignada: Sprint 3 |
| Programador responsable: Yhudiel | |
| <p>Descripción: El administrador podrá gestionar la información de los cajeros del sistema desde la interfaz administrativa.</p> <p>Esto incluye crear nuevos registros, actualizar datos existentes y eliminar cajeros de forma controlada, garantizando que las operaciones se reflejen correctamente en la base de datos y se valide la integridad de los datos ingresados.</p> | |
| <p>Criterios de aceptación:</p> <ul style="list-style-type: none"> ❖ El sistema debe mostrar una tabla o listado con los cajeros existentes y un botón “Agregar cajero”. ❖ Al presionar “Agregar cajero”, debe abrirse un formulario que permita registrar los datos requeridos (nombre, usuario, contraseña, correo, etc.). ❖ El sistema debe validar que los campos obligatorios estén completos antes de permitir el registro. ❖ Una vez confirmada la acción, el sistema debe insertar el nuevo cajero en la base de datos y mostrar un mensaje de éxito. ❖ El administrador podrá editar los datos de un cajero existente, lo que deberá reflejarse en la base de datos mediante una actualización (UPDATE). | |

- ❖ Al seleccionar la opción “Eliminar”, el sistema debe solicitar confirmación previa antes de ejecutar la acción (DELETE).
- ❖ Todas las operaciones deben generar mensajes claros de confirmación o error, según el resultado.

Validación:

- Si se intenta registrar un cajero sin llenar todos los campos requeridos, el sistema mostrará un mensaje de advertencia.
- Al guardar un nuevo cajero, este deberá aparecer automáticamente en la lista con sus datos correspondientes.

- En caso de editar un registro, los cambios deberán visualizarse de inmediato tras

| Historia de Usuario | |
|--|-------------------------------------|
| Número: US014 | Usuario: Administrador |
| Nombre historia: Gestionar Proveedores | |
| Prioridad en negocio: Alta | Riesgo en desarrollo: Alto |
| Puntos estimados: 5 | Iteración asignada: Sprint 3 |
| Programador responsable: Jazmín | |
| <p>Descripción: El administrador podrá gestionar la información de los proveedores desde el panel administrativo del sistema.</p> <p>Esta funcionalidad permitirá registrar nuevos proveedores, modificar sus datos (como nombre, empresa, teléfono, correo o dirección) y eliminarlos cuando ya no estén activos, asegurando que los registros se mantengan actualizados y sincronizados con la base de datos.</p> | |
| <p>Criterios de aceptación:</p> <ul style="list-style-type: none"> ❖ El sistema debe mostrar una lista de proveedores existentes junto con un botón “Agregar proveedor”. ❖ Al presionar “Agregar proveedor”, se debe abrir un formulario con los campos requeridos (nombre, empresa, contacto, correo electrónico, teléfono, dirección, etc.). ❖ Antes de guardar, el sistema debe validar que los campos obligatorios estén completos y que no exista un proveedor duplicado. ❖ Al confirmar, el sistema debe insertar el nuevo proveedor en la base de datos y mostrar un mensaje de confirmación. ❖ El administrador podrá editar la información de un proveedor existente, y los cambios deberán reflejarse inmediatamente en la base de datos. | |

- ❖ Si se selecciona “Eliminar”, el sistema debe solicitar confirmación previa antes de ejecutar la eliminación.
- ❖ La interfaz debe actualizar automáticamente la lista de proveedores después de agregar, modificar o eliminar un registro.
- ❖ Todos los mensajes deben ser claros, indicando el éxito o error de la operación realizada.

| Historia de Usuario | |
|--|-------------------------------------|
| Número: US015 | Usuario: Administrador |
| Nombre historia: Gestionar Inventario | |
| Prioridad en negocio: Alta | Riesgo en desarrollo: Alto |
| Puntos estimados: 5 | Iteración asignada: Sprint 4 |
| Programador responsable: Tania | |
| <p>Descripción: El administrador podrá gestionar los productos del inventario desde el panel administrativo.</p> <p>Esta funcionalidad permitirá registrar nuevos productos, modificar sus datos (nombre, categoría, precio, cantidad, proveedor, entre otros) y eliminarlos cuando ya no estén disponibles.</p> <p>Además, el sistema garantizará la integridad de la información, actualizando automáticamente la base de datos y mostrando mensajes claros de confirmación o error para cada operación realizada.</p> | |
| <p>Criterios de aceptación:</p> <ul style="list-style-type: none"> ❖ El sistema debe mostrar un listado de productos existentes con sus datos principales (nombre, categoría, cantidad, precio, proveedor). ❖ El administrador podrá agregar un nuevo producto mediante un formulario que solicite la información requerida. ❖ Antes de guardar, el sistema debe validar que todos los campos obligatorios estén completos y que el producto no exista previamente. ❖ Al confirmar el registro, el sistema deberá insertar el nuevo producto en la base de | |

datos y mostrar un mensaje de confirmación.

- ❖ El administrador podrá editar los datos de un producto existente; los cambios deben reflejarse de inmediato en la base de datos.
- ❖ Si se selecciona la opción “Eliminar”, el sistema deberá solicitar confirmación previa antes de ejecutar la eliminación.
- ❖ La interfaz deberá actualizar automáticamente la lista de productos tras cada operación (alta, modificación o baja).
- ❖ Todas las operaciones deberán mostrar mensajes claros sobre su resultado (éxito o error)

Validación:

- Si se intenta registrar un producto con campos vacíos o duplicados, el sistema mostrará un mensaje de advertencia.
- Al guardar un nuevo producto, este deberá aparecer en la lista actualizada y almacenarse correctamente en la base de datos.
- Si se modifica la información de un producto, los cambios deberán reflejarse de forma inmediata.
- Al eliminar un producto, este debe desaparecer del listado y eliminarse definitivamente de la base de datos.
- En caso de error (por ejemplo, datos inválidos, duplicados o fallo de conexión), el sistema deberá mostrar un mensaje descriptivo.
- El sistema deberá registrar todas las operaciones (alta, modificación y baja) en la bitácora administrativa con fecha, hora y usuario responsable.

| Historia de Usuario | |
|--|-------------------------------------|
| Número: US016 | Usuario: Administrador |
| Nombre historia: Gestionar Recompensas | |
| Prioridad en negocio: Alta | Riesgo en desarrollo: Medio |
| Puntos estimados: 5 | Iteración asignada: Sprint 3 |
| Programador responsable: David | |
| <p>Descripción: El administrador podrá gestionar las recompensas disponibles para los clientes dentro del sistema.</p> <p>Esta funcionalidad permitirá agregar nuevas recompensas, modificar su información (nombre, puntos requeridos, descripción, stock, estado, etc.) y eliminar recompensas que ya no estén disponibles.</p> <p>El sistema debe asegurar la validación de datos, la correcta comunicación con la base de datos y la actualización en tiempo real de la interfaz.</p> | |

Criterios de aceptación:

- ❖ El sistema debe mostrar un listado de recompensas registradas con sus atributos principales (nombre, puntos requeridos, disponibilidad).
- ❖ El administrador podrá agregar una nueva recompensa a través de un formulario con campos obligatorios (nombre, descripción, puntos requeridos, stock).
- ❖ El sistema debe verificar que todos los campos estén completos antes de guardar la nueva recompensa.
- ❖ Al confirmar la acción, se debe insertar la recompensa en la base de datos y mostrar un mensaje de confirmación.
- ❖ El administrador podrá editar una recompensa existente, actualizando los datos directamente en la base de datos.
- ❖ Si se selecciona “Eliminar”, el sistema deberá pedir confirmación antes de realizar la acción.
- ❖ El listado de recompensas deberá actualizarse automáticamente después de agregar, modificar o eliminar un registro.
- ❖ El sistema deberá mostrar mensajes claros de éxito o error según el resultado de la operación.

Validación:

- Si un campo obligatorio está vacío, el sistema debe mostrar un mensaje de advertencia y bloquear la acción.
- Al agregar una recompensa correctamente, esta debe visualizarse en la lista y guardarse en la base de datos.
- Los cambios realizados en una recompensa deberán reflejarse de inmediato en la interfaz después de confirmar la edición.
- Si se elimina una recompensa, debe desaparecer de la lista y eliminarse correctamente de la base de datos.
- En caso de error (duplicado, conexión fallida, o falta de validación), el sistema deberá notificarlo adecuadamente.
- Todas las operaciones (alta, modificación, baja) deben registrarse en la bitácora del sistema, incluyendo la fecha, hora y el administrador responsable.

| Historia de Usuario | |
|----------------------------|-------------------------------|
| Número: US017 | Usuario: Administrador |

| | |
|---|-------------------------------------|
| Nombre historia: Gestionar Estadísticas | |
| Prioridad en negocio: Alta | Riesgo en desarrollo: Medio |
| Puntos estimados: 5 | Iteración asignada: Sprint 3 |
| Programador responsable: Yhudiel | |
| <p>Descripción: El administrador podrá gestionar las recompensas dentro del sistema desde el panel administrativo.</p> <p>Esta funcionalidad permitirá crear nuevas recompensas, editar información existente (nombre, puntos requeridos, descripción, stock o estado) y eliminar recompensas que ya no estén vigentes.</p> <p>El sistema deberá garantizar la validación de datos, la actualización en la base de datos y la retroalimentación visual de cada acción ejecutada.</p> | |

Criterios de aceptación:

- ❖ El sistema debe mostrar una lista de recompensas registradas con información relevante (nombre, puntos requeridos, disponibilidad, descripción).
- ❖ El administrador podrá agregar nuevas recompensas mediante un formulario que contenga los campos necesarios.
- ❖ Antes de guardar una nueva recompensa, el sistema deberá verificar que todos los campos obligatorios estén llenos.
- ❖ Al confirmar la creación, se deberá insertar el registro en la base de datos y mostrar un mensaje de éxito.
- ❖ El administrador podrá modificar datos de una recompensa existente, actualizando su información en la base de datos.
- ❖ Si se selecciona la opción “Eliminar”, el sistema deberá solicitar confirmación previa antes de ejecutar la acción.
- ❖ La interfaz deberá actualizarse automáticamente al agregar, modificar o eliminar una recompensa.
- ❖ Los mensajes del sistema deben ser claros, diferenciando entre operaciones exitosas y errores.

Validación:

- Si algún campo requerido está vacío, el sistema deberá mostrar una advertencia y bloquear la acción.
- Al registrar una recompensa correctamente, esta deberá visualizarse inmediatamente en la lista de recompensas.
- Los cambios aplicados a una recompensa deberán reflejarse de manera inmediata en la interfaz.

- Si una recompensa es eliminada, deberá desaparecer del listado y eliminarse del registro en la base de datos.
- En caso de error (duplicidad, conexión fallida o campo inválido), el sistema mostrará

CAJERO

| Historia de Usuario | |
|---|-------------------------------------|
| Número: US018 | Usuario: Cajero |
| Nombre historia: Inicio | |
| Prioridad en negocio: Alta | Riesgo en desarrollo: Bajo |
| Puntos estimados: 5 | Iteración asignada: Sprint 3 |
| Programador responsable: Jazmin | |
| Descripción: Como cajero, quiero poder acceder al sistema mediante una pantalla de inicio de sesión, para validar mis credenciales y comenzar mis labores diarias dentro del sistema de ventas y control de pedidos. | |
| Criterios de aceptación: | |
| <ul style="list-style-type: none">❖ El sistema deberá mostrar una interfaz de inicio con campos para ingresar usuario y contraseña.❖ Una vez que el cajero inicie sesión correctamente, el sistema deberá redirigirlo al panel principal de operaciones.❖ El sistema deberá permitir cerrar sesión de forma segura.❖ En caso de inactividad prolongada, el sistema deberá cerrar sesión automáticamente por seguridad. | |

Validación:

- El cajero ingresa usuario y contraseña válidos, y el sistema permite el acceso al panel principal.
- Si el cajero ingresa credenciales incorrectas, el sistema muestra el mensaje “Usuario o contraseña incorrectos”.
- Al cerrar sesión, el sistema debe regresar a la pantalla de inicio y terminar la sesión actual.

Historia de Usuario

Número: US019

Usuario: Cajero

Nombre historia: Inventario

Prioridad en negocio: Alta

Riesgo en desarrollo: Medio

Puntos estimados: 8

Iteración asignada: Sprint 3

Programador responsable: David

Descripción: Como cajero, quiero poder consultar y actualizar el inventario de productos desde el sistema, para asegurarse de que haya disponibilidad suficiente al momento de registrar ventas y evitar errores por falta de existencias.

Criterios de aceptación:

- ❖ El sistema deberá permitir al cajero visualizar el inventario general de productos con su cantidad disponible.
- ❖ El cajero podrá buscar productos por nombre, código o categoría.
- ❖ El sistema deberá permitir actualizar las existencias de un producto (aumentar o disminuir la cantidad).

Validación:

- Si un usuario sin permisos intenta modificar el inventario, el sistema deberá denegar la acción y mostrar el mensaje: “Acceso no autorizado”.
- Toda actualización de inventario debe registrarse en un historial con fecha, hora y usuario responsable.

Historia de Usuario

Número: US020

Usuario: Cajero

Nombre historia: Corte de caja

Prioridad en negocio: Alta

Riesgo en desarrollo: Media

Puntos estimados: 8

Iteración asignada: Sprint 4

Programador responsable: Tania

Descripción: Como cajero, quiero poder realizar el corte de caja al finalizar turno, para revisar el total de ventas realizadas, los pagos recibidos y asegurar que el monto en efectivo coincida con el registrado en el sistema.

Criterios de aceptación:

- ❖ El sistema deberá permitir al cajero generar un reporte de corte de caja al final del turno.
- ❖ El reporte deberá incluir: total de ventas, número de transacciones, método de pago (efectivo, tarjeta, etc.) y monto total recaudado.
- ❖ El cajero podrá registrar el monto en efectivo existente y compararlo con el total esperado.
- ❖ Si existe una diferencia entre el monto registrado y el total esperado, el sistema deberá mostrar una alerta con la diferencia.

Validación:

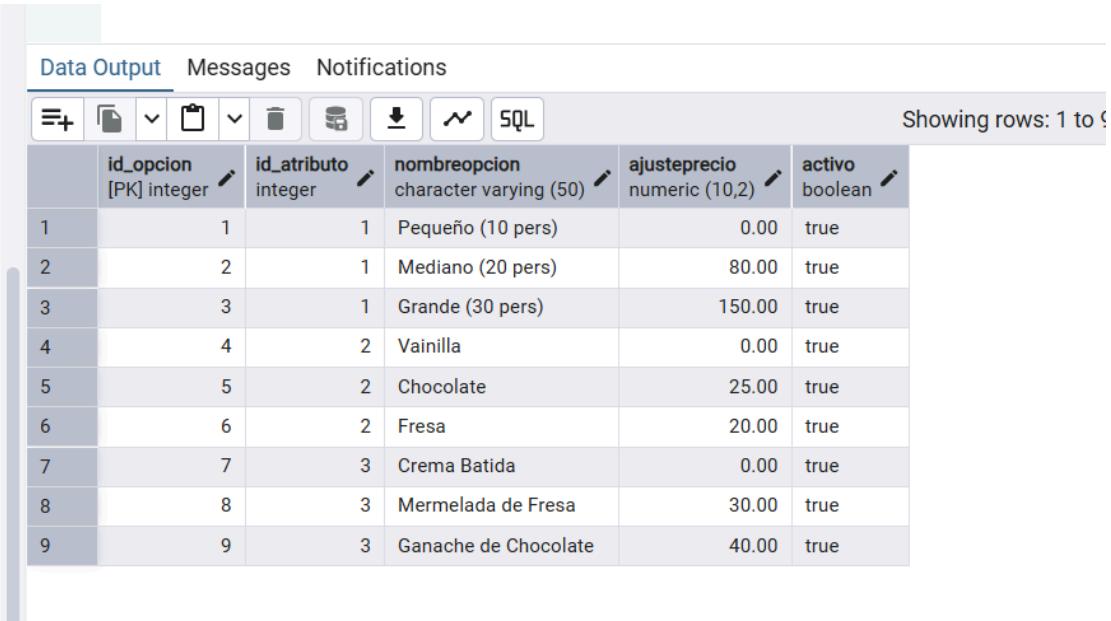
- El sistema guarda el reporte de corte de caja con la fecha, hora y usuario que lo generó.
- Si un usuario sin permisos intenta realizar el corte, el sistema muestra el mensaje: “Acceso no autorizado”.
- En caso de error al generar o guardar el reporte, el sistema muestra el mensaje:

BD

BASE DE DATOS

attribute_options

Esta tabla almacena las variaciones disponibles para personalizar productos, como tamaños, sabores o rellenos. Cada opción está relacionada con un atributo específico y puede modificar el precio base del producto. Permite activar o desactivar opciones sin eliminarlas del sistema.



The screenshot shows a database interface with a toolbar at the top containing icons for Data Output, Messages, Notifications, and various database operations like insert, update, delete, and export. Below the toolbar is a table with the following data:

| | id_opcion [PK] integer | id_atributo integer | nombreopcion character varying (50) | ajusteprecio numeric (10,2) | activo boolean |
|---|----------------------------------|-------------------------------|---|---------------------------------------|--------------------------|
| 1 | 1 | 1 | Pequeño (10 pers) | 0.00 | true |
| 2 | 2 | 1 | Mediano (20 pers) | 80.00 | true |
| 3 | 3 | 1 | Grande (30 pers) | 150.00 | true |
| 4 | 4 | 2 | Vainilla | 0.00 | true |
| 5 | 5 | 2 | Chocolate | 25.00 | true |
| 6 | 6 | 2 | Fresa | 20.00 | true |
| 7 | 7 | 3 | Crema Batida | 0.00 | true |
| 8 | 8 | 3 | Mermelada de Fresa | 30.00 | true |
| 9 | 9 | 3 | Ganache de Chocolate | 40.00 | true |

bitacoracambios

Registra cada cambio importante realizado en cualquier tabla del sistema para fines de auditoría.

Guarda quién hizo la acción, qué cambió y el estado anterior y nuevo del registro. Garantiza trazabilidad y control sobre acciones administrativas.

The screenshot shows a database management system interface with a toolbar at the top labeled "Data Output", "Messages", and "Notifications". Below the toolbar is a menu bar with icons for file operations like New, Open, Save, Print, and a SQL button. The main area displays the schema of the "bitacoracambios" table:

| <code>id_bitacora</code> [PK] bigint | <code>tablaafectada</code> character varying (100) | <code>id_registroafectado</code> character varying (50) | <code>accion</code> character varying (10) | <code>datosanteriores</code> jsonb | <code>datosnuevos</code> jsonb | <code>fechahora</code> timestamp with time zone | <code>id_usuario</code> integer |
|---|---|--|---|---------------------------------------|-----------------------------------|--|------------------------------------|
|---|---|--|---|---------------------------------------|-----------------------------------|--|------------------------------------|

carrito_items

Contiene los productos que cada usuario agrega temporalmente a su carrito de compras. Incluye cantidad, personalización seleccionada y la fecha de agregado. Sirve como estructura previa antes de generar un pedido formal.

The screenshot shows a database management system interface with a toolbar at the top labeled "Data Output", "Messages", and "Notifications". Below the toolbar is a menu bar with icons for file operations like New, Open, Save, Print, and a SQL button. The main area displays the schema of the "carrito_items" table and two rows of data:

| <code>id_item_carrito</code> [PK] integer | <code>id_usuario</code> integer | <code>id_producto</code> integer | <code>cantidad</code> integer | <code>personalizacion</code> jsonb | <code>fechaagregado</code> timestamp with time zone |
|--|------------------------------------|-------------------------------------|----------------------------------|---------------------------------------|---|
| 1 | 9 | 1 | 3 | 1 | {"Relleno": "Crema Batida", "Tamaño": "Pequeño (10 pers)", "Sabor de Pan": "Vainilla"} 2025-11-11 20:21:09.5 |
| 2 | 10 | 14 | 4 | 1 | {"Relleno": "Mermelada de Fresa", "Tamaño": "Pequeño (10 pers)", "Sabor de Pan": "Chocolate"} 2025-11-13 11:47:42.6 |

categorías

Agrupa los productos dentro de categorías generales como pasteles, cupcakes o bebidas. Facilita la organización y filtrado de productos dentro del catálogo. Permite activar o desactivar categorías sin perder información histórica.

The screenshot shows a database management interface with a toolbar at the top. The toolbar includes icons for Data Output (highlighted in blue), Messages, Notifications, and SQL. Below the toolbar is a table with the following data:

| | id_categoria [PK] integer | nombreCategoria character varying (100) | activo boolean |
|---|-------------------------------------|---|--------------------------|
| 1 | 1 | Pasteles | true |
| 2 | 2 | Galletas | true |
| 3 | 3 | Cupcakes | true |
| 4 | 4 | Bebidas | true |

cliente_recompensas

Registra las recompensas obtenidas por los clientes dentro del sistema de fidelización. Almacena la fecha en que se otorgó la recompensa y su estado de uso. Permite activar, cancelar o administrar recompensas individualmente.

The screenshot shows a database interface with a toolbar at the top containing icons for Data Output, Messages, Notifications, and various file operations. Below the toolbar is a SQL input field. The main area displays the structure of the `cliente_recompensas` table:

| | <code>id_cliente_recompensa</code> [PK] integer | <code>id_cliente</code> integer | <code>id_recompensa</code> integer | <code>fechaobtencion</code> timestamp with time zone | <code>estado</code> character varying (50) | <code>activo</code> boolean |
|--|--|------------------------------------|---------------------------------------|---|---|--------------------------------|
|--|--|------------------------------------|---------------------------------------|---|---|--------------------------------|

clientes

Guarda la información de los clientes registrados, vinculados directamente a un usuario del sistema. Incluye datos de contacto y la fecha de registro. Controla si el cliente está activo para compras y promociones.

The screenshot shows a database interface with a toolbar at the top containing icons for Data Output, Messages, Notifications, and various file operations. Below the toolbar is a SQL input field. The main area displays the data in the `clientes` table:

| | <code>id_cliente</code> [PK] integer | <code>id_usuario</code> integer | <code>nombre</code> character varying (100) | <code>telefono</code> character varying (20) | <code>fecharegistro</code> timestamp with time zone | <code>activo</code> boolean |
|---|---|------------------------------------|--|---|--|--------------------------------|
| 1 | 1 | 5 | Juan Pérez | 555-0001 | 2025-10-14 20:51:23.384671... | true |
| 2 | 2 | 6 | Maria García | 555-0002 | 2025-10-14 20:51:23.384671... | true |
| 3 | 3 | 7 | Carlos López | 555-0003 | 2025-10-14 20:51:23.384671... | true |
| 4 | 4 | 8 | Ana Martínez | 555-0004 | 2025-10-14 20:51:23.384671... | true |
| 5 | 5 | 9 | Luis Rodríguez | 555-0005 | 2025-10-14 20:51:23.384671... | true |
| 6 | 6 | 14 | Tania Santiago | 9513583708 | 2025-11-13 11:46:54.237-06 | true |

detalle_pedido

Representa cada producto incluido en un pedido, indicando su cantidad y precio unitario. Puede almacenar personalizaciones específicas por producto. Es esencial para el cálculo del total del pedido y para el historial de ventas.

The screenshot shows a database interface with a toolbar at the top containing various icons for file operations, a search bar, and a SQL button. Below the toolbar is a table titled "detalle_pedido". The table has the following columns and data:

| | id_detallepedido [PK] integer | id_pedido integer | id_producto integer | cantidad integer | preciounitario numeric (10,2) | activo boolean | personalizacion jsonb |
|---|----------------------------------|----------------------|------------------------|---------------------|----------------------------------|-------------------|--------------------------|
| 1 | | 1 | 3 | 4 | 1 | 0.00 | true |
| 2 | | 2 | 3 | 3 | 1 | 0.00 | true |
| 3 | | 3 | 4 | 1 | 1 | 350.00 | true |
| 4 | | 4 | 5 | 3 | 1 | 300.00 | true |
| 5 | | 5 | 6 | 2 | 1 | 380.00 | true |
| 6 | | 6 | 7 | 4 | 1 | 350.00 | true |

empleados

Registra la información del personal que trabaja en la empresa o punto de venta. Incluye el turno asignado y su vínculo con un usuario del sistema. Controla qué empleados están activos para operaciones internas.

The screenshot shows a database interface with a toolbar at the top containing various icons for file operations, a search bar, and a SQL button. Below the toolbar is a table titled "empleados". The table has the following columns and data:

| | id_empleado [PK] integer | id_usuario integer | nombrecompleto character varying (100) | turno character varying (50) | activo boolean |
|---|-----------------------------|-----------------------|---|---------------------------------|-------------------|
| 1 | | 2 | 1 Admin General | Completo | true |
| 2 | | 9 | 2 Roberto Gómez | Matutino | true |
| 3 | | 10 | 3 Patricia Silva | Vespertino | true |
| 4 | | 11 | 4 Elena Morales | Nocturno | true |

ingredientes

Contiene los insumos utilizados para fabricar los productos, junto con su cantidad disponible y mínima. Permite controlar inventarios por unidad de medida y asociar cada ingrediente con un proveedor. Sirve para alertas de abastecimiento.

The screenshot shows a database interface with a toolbar at the top containing various icons for data manipulation. Below the toolbar is a header row with column names and their data types. The main area displays a grid of data rows, each representing an ingredient with its ID, SKU, name, current stock, minimum stock, unit of measure, supplier ID, and active status.

| Data Output Messages Notifications | | | | | | | | |
|------------------------------------|---------------------------------------|--------------------------------------|--|--------------------------------------|--------------------------------------|---|--------------------------------|--------------------------|
| | id_ingrediente [PK] integer | sku character varying (50) | nombre character varying (100) | stockactual numeric (10,2) | stockminimo numeric (10,2) | unidadmedida character varying (20) | id_proveedor integer | activo boolean |
| 1 | 2 | AZU2 | Azúcar | 50.00 | 10.00 | kg | 2 | true |
| 2 | 3 | LEC3 | Leche | 20.00 | 5.00 | lt | 1 | true |
| 3 | 4 | HUE4 | Huevo | 100.00 | 24.00 | pza | 1 | true |
| 4 | 5 | CH05 | Chocolate Amargo | 15.00 | 3.00 | kg | 1 | true |
| 5 | 6 | FRE6 | Fresa | 10.00 | 2.00 | kg | 3 | true |
| 6 | 7 | HAR001 | Harina de Trigo | 50.00 | 10.00 | kg | 2 | false |
| 7 | 1 | HAR01 | Harina | 50.00 | 10.00 | kg | 2 | true |

orderitem_attributes

Relaciona qué opciones de personalización se asignaron a un producto dentro de un pedido. Permite identificar las variantes seleccionadas por el cliente para cada artículo. Su activación controla si la opción sigue siendo válida en el pedido.

The screenshot shows a database interface with a toolbar at the top containing various icons for data manipulation. Below the toolbar is a header row with column names and their data types. The main area displays a grid of data rows, each representing an attribute assignment with its detail ID, option ID, and active status.

| Data Output Messages Notifications | | | |
|------------------------------------|---|----------------------------------|--------------------------|
| | id_detallepedido [PK] integer | id_opcion [PK] integer | activo boolean |
| | | | |

pedidos

Registra los pedidos realizados por los clientes, incluyendo fecha, total a pagar y estado del proceso. Almacena datos de pago como método y monto cubierto. Controla la fase del pedido: preparación, listo, completado, etc.

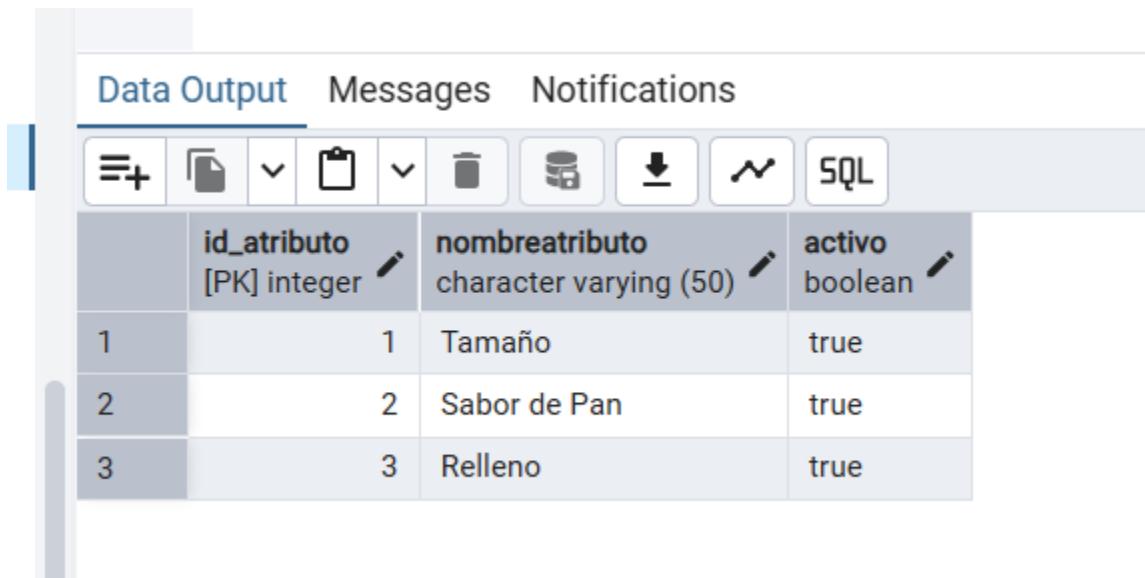


The screenshot shows a database interface with a toolbar at the top containing icons for Data Output, Messages, Notifications, and various file operations. Below the toolbar is a table with the following columns: id_pedido [PK] integer, id_cliente integer, id_empleado integer, fechapedido timestamp with time zone, total numeric (10,2), estado character varying (50), activo boolean, metodo_pago character varying (50), and monto_pago_con numeric (10,2). The table displays 7 rows of data:

| | id_pedido [PK] integer | id_cliente integer | id_empleado integer | fechapedido timestamp with time zone | total numeric (10,2) | estado character varying (50) | activo boolean | metodo_pago character varying (50) | monto_pago_con numeric (10,2) |
|---|---------------------------|-----------------------|------------------------|---|-------------------------|----------------------------------|-------------------|---------------------------------------|----------------------------------|
| 1 | 2 | 4 | 9 | 2025-11-07 20:49:36.384754... | 1200.00 | En preparación | true | [null] | [null] |
| 2 | 6 | 1 | 9 | 2025-11-09 21:41:00.328-06 | 380.00 | Completado | true | Efectivo | 500.00 |
| 3 | 7 | 1 | 9 | 2025-11-09 21:48:58.519-06 | 350.00 | Completado | true | Efectivo | 500.00 |
| 4 | 5 | 4 | [null] | 2025-11-09 21:39:03.607-06 | 300.00 | Listo | true | Transferencia | [null] |
| 5 | 1 | 4 | 9 | 2025-11-06 20:49:36.384754... | 900.00 | Terminado | true | [null] | [null] |
| 6 | 3 | 4 | [null] | 2025-11-09 19:59:40.264-06 | 650.00 | Completado | true | Efectivo | 1000.00 |
| 7 | 4 | 4 | [null] | 2025-11-09 21:26:27.09-06 | 350.00 | Listo | true | Tarjeta | [null] |

product_attributes

Define los tipos de atributos que pueden tener los productos, como tamaño, sabor de pan o relleno. Es la base para agrupar opciones y construir personalizaciones. Puede activarse o desactivarse sin afectar registros previos.



The screenshot shows a database interface with a toolbar at the top containing icons for Data Output, Messages, Notifications, and various file operations. Below the toolbar is a table with the following columns: id_atributo [PK] integer, nombreatributo character varying (50), and activo boolean. The table displays 3 rows of data:

| | id_atributo [PK] integer | nombreatributo character varying (50) | activo boolean |
|---|-----------------------------|--|-------------------|
| 1 | 1 | Tamaño | true |
| 2 | 2 | Sabor de Pan | true |
| 3 | 3 | Relleno | true |

product_ingredients

Indica qué ingredientes utiliza cada producto y en qué cantidad. Permite calcular costos, descontar inventario y controlar la producción. Su activación indica si la receta sigue vigente.

The screenshot shows a database management interface with a toolbar at the top containing icons for Data Output, Messages, Notifications, and various file operations. Below the toolbar is a table with four columns: id_atributo [PK] integer, nombreatributo character varying (50), and activo boolean. The table has three rows with data: 1 (Tamaño, true), 2 (Sabor de Pan, true), and 3 (Relleno, true).

| | id_atributo [PK] integer | nombreatributo character varying (50) | activo boolean |
|---|-----------------------------|--|-------------------|
| 1 | 1 | Tamaño | true |
| 2 | 2 | Sabor de Pan | true |
| 3 | 3 | Relleno | true |

productos

Almacena los productos que se venden, incluyendo nombre, SKU, descripción y precio base. Se vincula con una categoría y puede incluir información de inventario del producto terminado. Contiene la URL de la imagen para mostrar en la tienda.

The screenshot shows a database management interface with a toolbar at the top containing icons for Data Output, Messages, Notifications, and various file operations. Below the toolbar is a table with eight columns: lucto integer, sku character varying (50), nombre character varying (100), descripcion text, preciobase numeric (10,2), id_categoria integer, stockproductosterminalados integer, activo boolean, and imagenurl character varying (255). The table has four rows with data: 1 (PASC, Pastel de Chocolate, Clásico pastel de chocolate, 350.00, 1, 4, true, /uploads/PastelChocolate.jpg), 2 (PASO, Pastel de oreo, Pastel de 3 leches, hecho con galletas ore..., 300.00, 1, 19, true, /uploads/PastelOreo.jpg), 3 (PASF, Pastel de Fresa y Crema, Pastel de Fresas con crema, 3 leches..., 380.00, 1, 9, true, /uploads/PastelFresasCrema.j...), and 4 (PASV, Pastell de vainilla, Pastel de Vainilla, clásico..., 350.00, 1, 14, true, /uploads/PastelVainilla.jpg).

| | lucto integer | sku character varying (50) | nombre character varying (100) | descripcion text | preciobase numeric (10,2) | id_categoria integer | stockproductosterminalados integer | activo boolean | imagenurl character varying (255) | |
|---|------------------|-------------------------------|-----------------------------------|---|------------------------------|-------------------------|---------------------------------------|-------------------|--------------------------------------|---------------------------------|
| 1 | 1 | PASC | Pastel de Chocolate | Clásico pastel de chocolate | 350.00 | 1 | | 4 | true | /uploads/PastelChocolate.jpg |
| 2 | 3 | PASO | Pastel de oreo | Pastel de 3 leches, hecho con galletas ore... | 300.00 | 1 | | 19 | true | /uploads/PastelOreo.jpg |
| 3 | 2 | PASF | Pastel de Fresa y Crema | Pastel de Fresas con crema, 3 leches... | 380.00 | 1 | | 9 | true | /uploads/PastelFresasCrema.j... |
| 4 | 4 | PASV | Pastell de vainilla | Pastel de Vainilla, clásico... | 350.00 | 1 | | 14 | true | /uploads/PastelVainilla.jpg |

proveedores

Registra los proveedores que suministran ingredientes o insumos. Incluye información fiscal, datos de contacto y estado del proveedor. Permite administrar relaciones comerciales y compras.

The screenshot shows a database management interface with a toolbar at the top containing icons for Data Output, Messages, Notifications, and various file operations. Below the toolbar is a table structure with columns: id_proveedor [PK] integer, nombre character varying (100), contacto character varying (100), telefono character varying (20), rfc character varying (13), and activo boolean. The table contains three rows of data:

| | id_proveedor [PK] integer | nombre character varying (100) | contacto character varying (100) | telefono character varying (20) | rfc character varying (13) | activo boolean |
|---|------------------------------|-----------------------------------|-------------------------------------|------------------------------------|-------------------------------|-------------------|
| 1 | 1 | Lácteos del Sur | Ernesto Paz | 555-1001 | LDS120101AA1 | true |
| 2 | 2 | Harinas La Oaxaqueña | María López | 555-1002 | HAO120101BB2 | true |
| 3 | 3 | Frutas Frescas de la Sie... | Carlos Ruiz | 555-1003 | FFS120101CC3 | true |

recompensas

Define las recompensas disponibles dentro del sistema de puntos para clientes. Establece el tipo de beneficio, su valor y cuántos puntos requiere. Controla cuáles recompensas están activas para ser otorgadas.

The screenshot shows a database management interface with a toolbar at the top containing icons for Data Output, Messages, Notifications, and various file operations. Below the toolbar is a table structure with columns: id_recompensa [PK] integer, nombre_recompensa character varying (100), descripcion text, tipo character varying (50), valor numeric (10,2), puntos_requeridos integer, and activo boolean. The table contains one row of data:

| | id_recompensa [PK] integer | nombre_recompensa character varying (100) | descripcion text | tipo character varying (50) | valor numeric (10,2) | puntos_requeridos integer | activo boolean |
|---|-------------------------------|--|--|--------------------------------|-------------------------|------------------------------|-------------------|
| 1 | 1 | Promoción Apertura | Recompensa por hacer la primera com... | PORCENTAJE_DESCUEN... | 15.00 | 200 | true |

roles

Administra los roles del sistema, como administrador, cajero o cliente. Define los permisos y el alcance de acceso de cada rol. Controla qué roles están habilitados para uso.

The screenshot shows a database interface with a toolbar at the top containing icons for Data Output, Messages, Notifications, and various file operations. Below the toolbar is a table titled 'roles' with the following columns: id_rol [PK] integer, nombrerol character varying (50), descripcion text, and activo boolean. The table contains three rows:

| | id_rol [PK] integer | nombrerol character varying (50) | descripcion text | activo boolean |
|---|------------------------|-------------------------------------|--|-------------------|
| 1 | 1 | Administrador | Acceso total al sistema para configuración y gestión. | true |
| 2 | 2 | Cajero | Acceso al punto de venta, gestión de pedidos e inventario. | true |
| 3 | 3 | Cliente | Acceso a la tienda para realizar compras y ver historial. | true |

usuarios

Guarda las cuentas de acceso al sistema con su correo, contraseña cifrada y rol asignado. Registra la fecha de creación y controla si la cuenta está activa. Es la base de autenticación y permisos del sistema.

The screenshot shows a database interface with a toolbar at the top containing icons for Data Output, Messages, Notifications, and various file operations. Below the toolbar is a table titled 'usuarios' with the following columns: id_usuario [PK] integer, correoelectronico character varying (100), contrasena character varying (255), id_rol integer, fechacreacion timestamp with time zone, and activo boolean. The table contains ten rows:

| | id_usuario [PK] integer | correoelectronico character varying (100) | contrasena character varying (255) | id_rol integer | fechacreacion timestamp with time zone | activo boolean |
|----|----------------------------|--|--|-------------------|---|-------------------|
| 1 | 1 | admin@dulsys.com | \$2b\$10\$F8edSnU7qXF.zKe7CY0J.3MK6z2T4Gsxdh0EDqxJtDCly/0cT... | 1 | 2025-10-14 20:48:12.801982... | true |
| 2 | 2 | roberto.gomez@dulsys.... | \$2b\$10\$Ngh2PQkRiZZx3GEHTFk2iuhU6XB3j8A1od4/KreypcdjkG04U... | 2 | 2025-10-14 20:49:36.384754... | true |
| 3 | 3 | patricia.silva@dulsys.c... | \$2b\$10\$Ngh2PQkRiZZx3GEHTFk2iuhU6XB3j8A1od4/KreypcdjkG04U... | 2 | 2025-10-14 20:49:36.384754... | true |
| 4 | 4 | elena.morales@dulsys.... | \$2b\$10\$Ngh2PQkRiZZx3GEHTFk2iuhU6XB3j8A1od4/KreypcdjkG04U... | 2 | 2025-10-14 20:49:36.384754... | true |
| 5 | 5 | juan.perez@email.com | \$2b\$10\$Ngh2PQkRiZZx3GEHTFk2iuhU6XB3j8A1od4/KreypcdjkG04U... | 3 | 2025-10-14 20:50:41.304722... | true |
| 6 | 6 | maria.garcia@email.com | \$2b\$10\$Ngh2PQkRiZZx3GEHTFk2iuhU6XB3j8A1od4/KreypcdjkG04U... | 3 | 2025-10-14 20:50:41.304722... | true |
| 7 | 7 | carlos.lopez@email.com | \$2b\$10\$Ngh2PQkRiZZx3GEHTFk2iuhU6XB3j8A1od4/KreypcdjkG04U... | 3 | 2025-10-14 20:50:41.304722... | true |
| 8 | 8 | ana.martinez@email.co... | \$2b\$10\$Ngh2PQkRiZZx3GEHTFk2iuhU6XB3j8A1od4/KreypcdjkG04U... | 3 | 2025-10-14 20:50:41.304722... | true |
| 9 | 9 | luis.rodriguez@email.c... | \$2b\$10\$Ngh2PQkRiZZx3GEHTFk2iuhU6XB3j8A1od4/KreypcdjkG04U... | 3 | 2025-10-14 20:50:41.304722... | true |
| 10 | 14 | santiagovtania@gmail.... | \$2b\$10\$6RQT6t55NY7XIRsdrXAEMuZATafS..boobSONHk6fA49vcy... | 3 | 2025-11-13 11:46:54.205-06 | true |

DISEÑO Y VISTAS

DISEÑO Y VISTAS

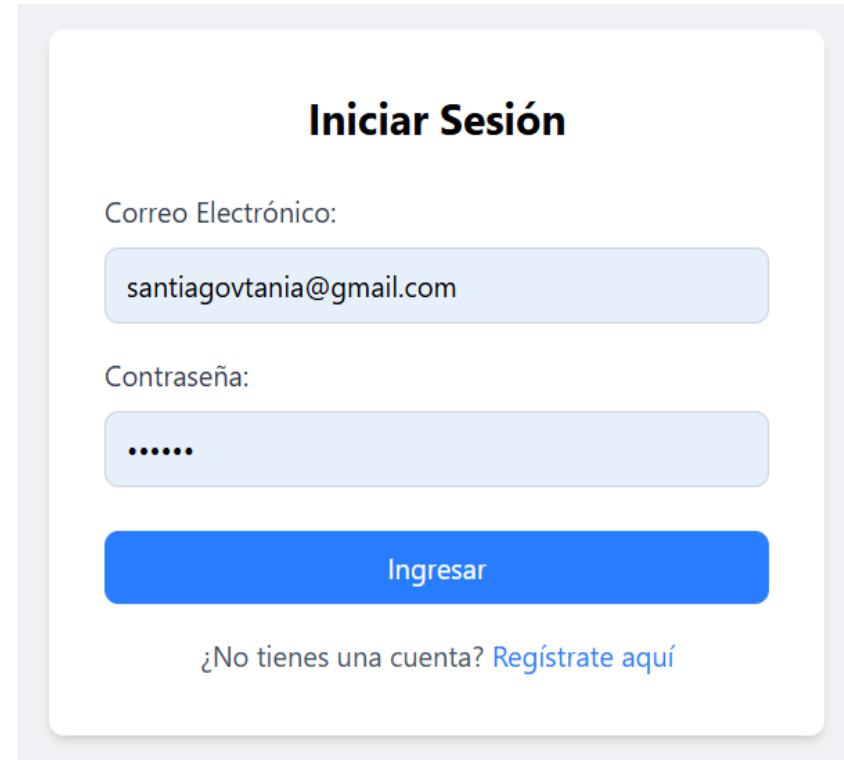
- INICIAR SESIÓN :



The original login interface is a white rectangular form titled "Login". It contains fields for "Usuario" (with a user icon) and "Contraseña" (with a lock icon). Below these is a "Tipo de Usuario" section with three radio button options: "Cliente", "Administrador" (which is selected), and "Cajero". At the bottom is a blue rectangular button labeled "[→] Ingresar".

El administrador selecciona el tipo de usuario ,

y se cambiaron los colores de los botones.



The new login interface is a white rounded rectangle titled "Iniciar Sesión". It features a light blue input field for "Correo Electrónico" containing "santiagovtania@gmail.com". Below it is another light blue input field for "Contraseña" with five dots indicating the password. A large blue rectangular button at the bottom right is labeled "Ingresar". At the bottom left, there is a link "¿No tienes una cuenta? [Regístrate aquí](#)".

Diseño actual del sistema

Registro:

Registro de Cliente

Correo Electrónico

[✉] correo@ejemplo.com

Usuario

[phone] Telefono

Contraseña

[🔒] Contraseña

[✓] Registrar [✗] Cancelar

Los cambios realizados son mínimos, solo se reorganizó el orden al actual, y se agregó el ¿Ya tienes cuenta? inicia sesión

Crear una Cuenta

Nombre Completo:

Tania Santiago

Teléfono:

9513583708

Correo Electrónico:

santiagovtania@gmail.com

Contraseña:

.....

Registrarse

¿Ya tienes una cuenta? [Inicia sesión](#)

Diseño actual

- **CLIENTE**



Inicio:

El diseño previo planteaba un carrusel horizontal de productos, con flechas laterales y tarjetas muy básicas sin imágenes reales. Los botones superiores del menú eran cuadrados y minimalistas.

En la versión final, el carrusel fue reemplazado por una cuadrícula de productos más realista con imágenes de alta calidad. El menú superior se estilizó con un diseño moderno, alineación derecha y un botón rojo para cerrar sesión. La búsqueda se conservó, pero ahora el contenido tiene mayor aire visual y las tarjetas muestran precios resaltados y descripciones más atractivas, logrando una experiencia más comercial y profesional.

Buscar productos...

Nuestros Productos



Pastel de Chocolate

Clásico pastel de chocolate

\$350



Pastel de oreo

Pastel de 3 leches, hecho con gall...

\$300



Pastel de Fresa y Crema

Pastel de Fresas con crema, 3 lech...

\$380

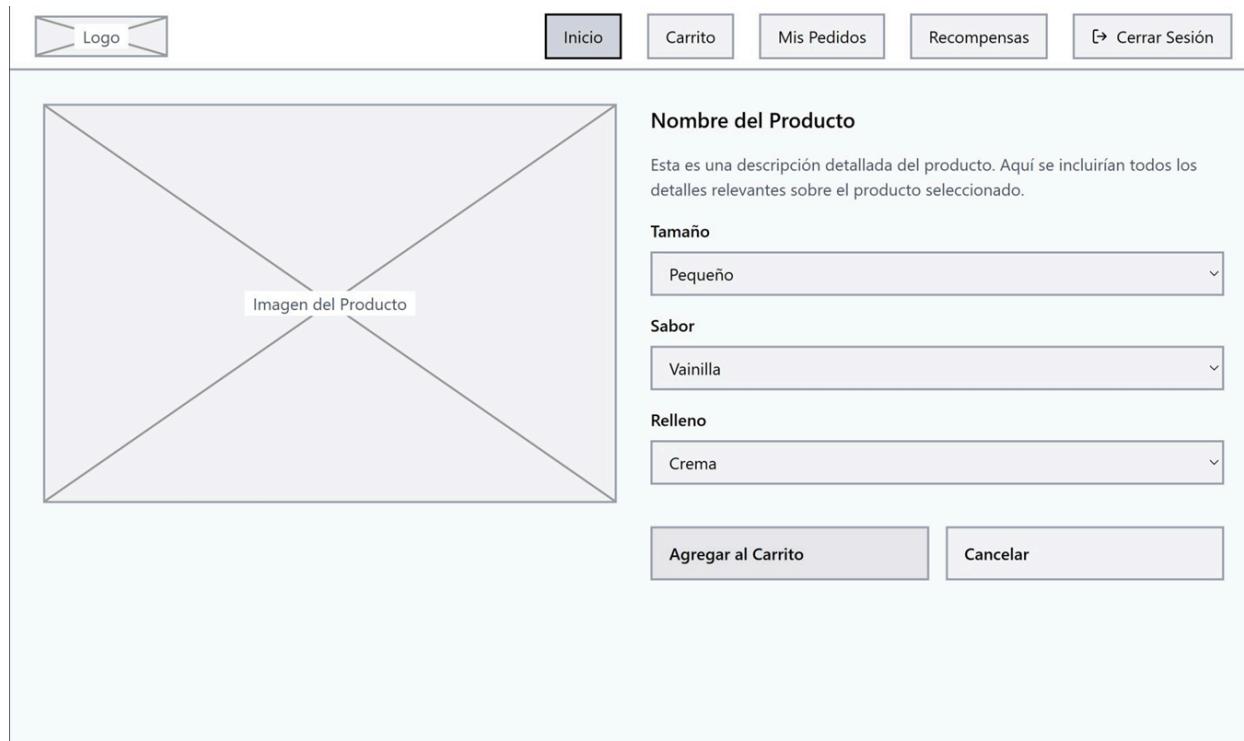


Pastell de vainilla

Pastel de Vainilla, clásico...

\$350

El diseño previo mostraba una estructura simple: imagen del producto, nombre genérico, una breve descripción y tres listas desplegables para tamaño, sabor y relleno, además de botones de agregar y cancelar.



En la versión final, el diseño se enriqueció con una imagen real del producto, una descripción más elaborada, opciones claramente identificadas con sus variaciones de precio y la indicación del stock disponible. El contenedor final luce más limpio y profesional, con botones estilizados, un layout mejor distribuido y una visual más agradable que comunica confianza al comprador.

DulceSys

Inicio Carrito Mis Pedidos Hola, Cliente Cerrar Sesión



Pastel de oreo
Pastel de 3 leches, hecho con galletas oreo...

Tamaño
Pequeño (10 pers) (+\$0.00)

Sabor de Pan
Vainilla (+\$0.00)

Relleno
Crema Batida (+\$0.00)

Disponibles: 17 pzas.

Agregar al Carrito **Cancelar**

Carrito

En el diseño inicial, el carrito era una tabla clásica con columnas para producto, cantidad, precio unitario, subtotal y un botón de eliminación por cada fila. La vista era más parecida a un panel administrativo que a una tienda moderna.

The screenshot shows a user interface for a shopping cart. At the top, there is a navigation bar with a logo icon and the word "Logo" on the left, and five buttons: "Inicio", "Carrito" (which is highlighted with a dark gray background), "Mis Pedidos", "Recompensas", and "Cerrar Sesión" (Logout) on the right. Below the navigation bar is a table representing the shopping cart's contents. The table has columns for "Producto" (Product), "Cantidad" (Quantity), "Precio Unitario" (Unit Price), "Subtotal", and "Acción" (Action). There are four rows in the table, each representing a product: "Producto 1" (2 units, \$15.00 unit price, \$30.00 subtotal, "Eliminar" button), "Producto 2" (1 unit, \$25.00 unit price, \$25.00 subtotal, "Eliminar" button), "Producto 3" (3 units, \$10.00 unit price, \$30.00 subtotal, "Eliminar" button), and "Producto 4" (1 unit, \$35.00 unit price, \$35.00 subtotal, "Eliminar" button). At the bottom of the cart area, there is a summary: "Total: \$120.00" and a "Comprar Carrito" (Buy Cart) button.

| Producto | Cantidad | Precio Unitario | Subtotal | Acción |
|------------|----------|-----------------|----------|---------------------------|
| Producto 1 | 2 | \$15.00 | \$30.00 | <button>Eliminar</button> |
| Producto 2 | 1 | \$25.00 | \$25.00 | <button>Eliminar</button> |
| Producto 3 | 3 | \$10.00 | \$30.00 | <button>Eliminar</button> |
| Producto 4 | 1 | \$35.00 | \$35.00 | <button>Eliminar</button> |

Total: \$120.00 Comprar Carrito

En la versión final, el carrito adoptó un estilo más visual y amigable, con una tarjeta central, texto reforzado y botones llamativos. Se mantiene la información clave, pero ahora se presenta con mejor estética y con un botón “Comprar Carrito” destacado en color. El diseño final se adapta más al enfoque de e-commerce, priorizando la claridad y la experiencia del usuario.

The screenshot shows a web browser window with the URL "localhost:5173/cart". The page title is "DulceSys". The main content area is titled "Tu Carrito de Compras" (Your Shopping Cart). It displays a single item: "Pastell de vainilla (Mermelada de Fresa, Pequeño (10 pers), Chocolate)". The table includes columns for PRODUCTO, CANTIDAD, PRECIO UNITARIO, and SUBTOTAL. The total amount is \$350.00. A pink button labeled "Comprar Carrito" is prominently displayed at the bottom right. The top navigation bar includes links for Inicio, Carrito, Mis Pedidos, Hola, Cliente, and Cerrar Sesión, along with various browser icons.

| PRODUCTO | CANTIDAD | PRECIO UNITARIO | SUBTOTAL |
|--|----------|-----------------|----------|
| Pastell de vainilla (Mermelada de Fresa, Pequeño (10 pers), Chocolate) | 1 | \$350.00 | \$350.00 |

Total: \$350.00 **Comprar Carrito**

Método de pago

El previo diseño es mucho más básico, presentando únicamente los campos de tarjeta, fecha y CVV con botones planos, sin la estructura visual refinada ni las opciones de pago detalladas que se integraron en la versión final.

Pago

Número de Tarjeta

0000 0000 0000 0000

Fecha de Vencimiento

MM/AA

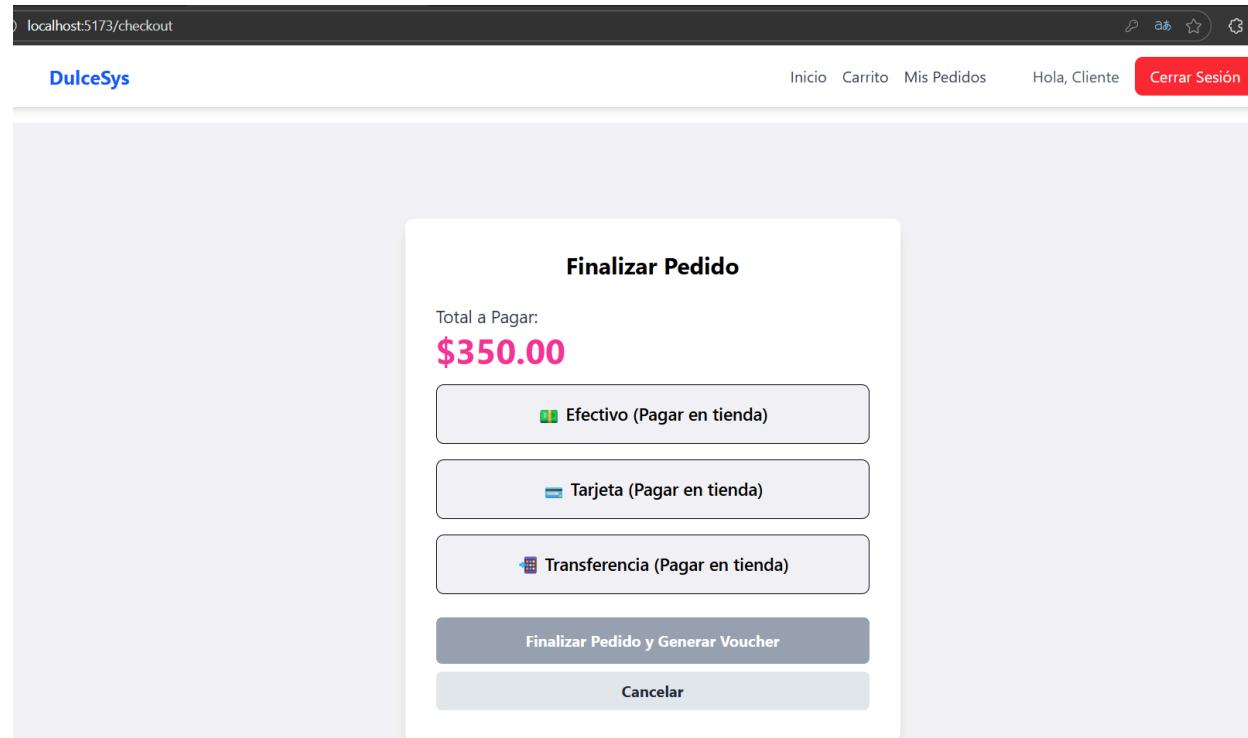
CVV

000

Pagar

Cancelar

El diseño final ofrece una interfaz más moderna y completa, con botones visualmente destacados, opciones de pago claras y un campo adicional para indicar el monto con el que se pagará, logrando una experiencia más intuitiva y profesional.



Finalizar Pedido

Total a Pagar:

\$350.00

 Efectivo (Pagar en tienda)

 Tarjeta (Pagar en tienda)

 Transferencia (Pagar en tienda)

¿Con cuánto vas a pagar?

350



Finalizar Pedido y Generar Voucher

Cancelar

Mis pedidos

El previo diseño utiliza una tabla más completa pero con un estilo simple y menos estilizado, mostrando los pedidos en un formato tradicional sin la jerarquía visual ni la estética refinada del diseño final.

The screenshot displays a user interface for managing orders. At the top, there is a navigation bar with a logo icon and the word "Logo". Below the logo are five buttons: "Inicio", "Carrito", "Mis Pedidos" (which is highlighted with a dark border), "Recompensas", and "Cerrar Sesión". A search bar labeled "Buscar pedidos..." is positioned below the navigation. The main content area contains a table with the following data:

| ID Pedido | Fecha | Total | Estado |
|-----------|------------|----------|------------|
| 001 | 2025-09-25 | \$85.00 | Completado |
| 002 | 2025-09-26 | \$45.00 | En proceso |
| 003 | 2025-09-27 | \$120.00 | Pendiente |
| 004 | 2025-09-28 | \$65.00 | Completado |
| 005 | 2025-09-29 | \$95.00 | En proceso |

El diseño final muestra una interfaz limpia y moderna, con una tarjeta central que presenta el estado de los pedidos de forma clara y minimalista, dando una experiencia más ordenada y visualmente atractiva.



Mis Pedidos

Aún no has realizado ningún pedido.

Recompensas disponibles

Cambiamos las recompensas a las vistas del administrador, ya que para clientes no hacía mucho sentido, así que ahora el admin asigna directamente las recompensas en la compra de los clientes

The screenshot shows a user interface for managing rewards. At the top, there is a navigation bar with a logo icon and the word "Logo", followed by links for "Inicio", "Carrito", "Mis Pedidos", "Recompensas" (which is highlighted with a dark border), and "Cerrar Sesión". Below the navigation, the title "Recompensas Disponibles" is displayed. Six reward options are listed in a grid:

| Recompensa | Detalles |
|------------------|---|
| 10% de Descuento | Descuento del 10% en tu próxima compra |
| Envío Gratis | Envío gratuito en pedidos mayores a \$50 |
| Producto Gratis | Un producto gratis al acumular 100 puntos |
| 20% de Descuento | Descuento del 20% en tu cumpleaños |
| 2x1 en Pasteles | Compra un pastel y lleva otro gratis |
| Café Gratis | Café gratis con cualquier compra |

● ADMINISTRADOR

El diseño final integra una interfaz moderna con productos visualmente atractivos, tarjetas estilizadas y una navegación clara, ofreciendo una presentación mucho más profesional y llamativa.

DulceSys Inicio Cajeros Proveedores Inventario Recompensas Estadísticas Gestión de Pedidos Cierre de Caja Hola, Administrador Cerrar Sesión

Buscar productos...

Nuestros Productos

| | | | |
|--|---|--|---|
| Pastel de Chocolate Clásico pastel de chocolate \$350 | Pastel de oreo Pastel de 3 leches, hecho con gall... \$300 | Pastel de Fresa y Crema Pastel de Fresas con crema, 3 lech... \$380 | Pastell de vainilla Pastel de Vainilla, clásico... \$350 |
|--|---|--|---|

Cajeros

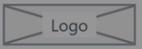
El previo diseño muestra formularios y tablas más simples y planos, con botones básicos y una estructura funcional pero menos estética, sin la jerarquía visual ni el estilo refinado del diseño final.

The screenshot shows a user interface for managing cashier data. At the top, there is a navigation bar with a logo icon and the text "Logo". Below the logo are several menu items: "Cajeros" (which is highlighted in a darker shade), "Clientes", "Proveedores", "Inventario", "Recompensas", "Estadísticas", and "Cerrar Sesión".

Below the navigation bar is a search bar labeled "Buscar cajeros...". To the right of the search bar are three buttons: "Agregar", "Editar", and "Eliminar".

The main content area is a table listing four cashier records:

| ID Cajero | Nombre Completo | Usuario | Correo | Teléfono | Turno |
|-----------|-----------------|----------|--------------------------------|----------|------------|
| CAJ001 | Roberto Gómez | rgomez | roberto.gomez@pasteleria.com | 555-2001 | Matutino |
| CAJ002 | Patricia Silva | psilva | patricia.silva@pasteleria.com | 555-2002 | Vespertino |
| CAJ003 | Fernando Castro | fcastro | fernando.castro@pasteleria.com | 555-2003 | Matutino |
| CAJ004 | Elena Morales | emorales | elena.morales@pasteleria.com | 555-2004 | Nocturno |



Logo

Cajeros

Cajeros

Clientes

Proveedores

Inventario

Recomper

Buscar cajeros...

| ID Cajero | Nombre Completo | Usuario | Correo | Teléfono |
|-----------|-----------------|----------|--------------------------------|-------------|
| CAJ001 | Roberto Gómez | rgomez | roberto.gomez@pasteleria.com | 555-1234567 |
| CAJ002 | Patricia Silva | psilva | patricia.silva@pasteleria.com | 555-1234567 |
| CAJ003 | Fernando Castro | fcastro | fernando.castro@pasteleria.com | 555-1234567 |
| CAJ004 | Elena Morales | emorales | elena.morales@pasteleria.com | 555-1234567 |

Agregar Cajero X

Nombre Completo

Usuario

Correo Electrónico

Contraseña

Teléfono

Turno
 ▾

The screenshot displays a software application interface. At the top, there is a navigation bar with tabs: Logo, Clientes, Cajeros (highlighted in blue), Proveedores, Inventario, and Recomper. Below the navigation bar is a search bar labeled "Buscar cajeros...". A table lists four employees with columns: ID Cajero, Nombre Completo, Usuario, Correo, and Teléfono. The table rows are CAJ001 (Roberto Gómez), CAJ002 (Patricia Silva), CAJ003 (Fernando Castro), and CAJ004 (Elena Morales). On the right side, a modal window titled "Editar Cajero" is open, containing fields for Nombre Completo, Usuario, Correo Electrónico, Contraseña, Teléfono, and Turno (with "Matutino" selected). At the bottom of the modal are "Guardar" and "Cancelar" buttons.

| ID Cajero | Nombre Completo | Usuario | Correo | Teléfono |
|-----------|-----------------|----------|--------------------------------|----------|
| CAJ001 | Roberto Gómez | rgomez | roberto.gomez@pasteleria.com | 555-1234 |
| CAJ002 | Patricia Silva | psilva | patricia.silva@pasteleria.com | 555-1234 |
| CAJ003 | Fernando Castro | fcastro | fernando.castro@pasteleria.com | 555-1234 |
| CAJ004 | Elena Morales | emorales | elena.morales@pasteleria.com | 555-1234 |

El diseño final ofrece una interfaz más moderna y organizada, con tablas estilizadas, acciones resaltadas y un botón de agregar proveedor más visible, mejorando la claridad y la experiencia de administración.

Gestión de Cajeros

[Agregar Cajero](#)

| IDENTIFICACIÓN | NOMBRE COMPLETO | CORREO (USUARIO) | TURNO | ACCIONES |
|----------------|-----------------|---------------------------|------------|---|
| 9 | Roberto Gómez | roberto.gomez@dulsys.com | Matutino | Editar Eliminar |
| 10 | Patricia Silva | patricia.silva@dulsys.com | Vespertino | Editar Eliminar |
| 11 | Elena Morales | elena.morales@dulsys.com | Nocturno | Editar Eliminar |

Proveedores:

El previo diseño muestra formularios y tablas más simples y planos, con botones básicos y una estructura funcional pero menos estética, sin la jerarquía visual ni el estilo refinado del diseño final.

| ID Proveedor | Nombre | Contacto | Teléfono | RFC | Producto Principal |
|--------------|-------------------|-------------|----------|--------------|--------------------|
| P001 | Lácteos SA | Pedro Díaz | 555-1001 | LAC840101AB1 | Leche y Crema |
| P002 | Harinas del Norte | Laura Ruiz | 555-1002 | HNO900215CD2 | Harina |
| P003 | Frutas Frescas | Miguel Soto | 555-1003 | FFR950330EF3 | Frutas |
| P004 | Azúcar y Más | Carmen Vera | 555-1004 | AZU880512GH4 | Azúcar |



Clientes

Cajeros

Proveedores

Inventario

Recomper

Agregar Proveedor

X

Buscar proveedores...

| ID Proveedor | Nombre | Contacto | Teléfono | RFC |
|--------------|-------------------|-------------|----------|--------|
| P001 | Lácteos SA | Pedro Díaz | 555-1001 | LA001 |
| P002 | Harinas del Norte | Laura Ruiz | 555-1002 | HN002 |
| P003 | Frutas Frescas | Miguel Soto | 555-1003 | FFF003 |
| P004 | Azúcar y Más | Carmen Vera | 555-1004 | AZ004 |

Nombre

Contacto

Teléfono

RFC

Producto Principal

Guardar

Cancelar

Logo

[Clientes](#)

[Cajeros](#)

[Proveedores](#)

[Inventario](#)

[Recomper](#)

| ID Proveedor | Nombre | Contacto | Teléfono | RFC |
|--------------|-------------------|-------------|----------|--------|
| P001 | Lácteos SA | Pedro Díaz | 555-1001 | LA001 |
| P002 | Harinas del Norte | Laura Ruiz | 555-1002 | HN002 |
| P003 | Frutas Frescas | Miguel Soto | 555-1003 | FFF003 |
| P004 | Azúcar y Más | Carmen Vera | 555-1004 | AZ004 |

Editar Proveedor

X

Nombre

Contacto

Teléfono

RFC

Producto Principal

Guardar

Cancelar

El diseño final ofrece una interfaz más moderna y organizada, con tablas estilizadas, acciones resaltadas y un botón de agregar proveedor más visible, mejorando la claridad y la experiencia de administración.

Gestión de Proveedores

Buscar por nombre o RFC...

Agregar Proveedor

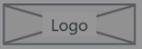
| IDENTIFICACIÓN | NOMBRE | CONTACTO | TELÉFONO | RFC | ACCIONES |
|----------------|------------------------------------|-------------|----------|--------------|---|
| 3 | Frutas Frescas de la Sierra | Carlos Ruiz | 555-1003 | FFS120101CC3 | Editar Eliminar |
| 2 | Harinas La Oaxaqueña | María López | 555-1002 | HAO120101BB2 | Editar Eliminar |
| 1 | Lácteos del Sur | Ernesto Paz | 555-1001 | LDS120101AA1 | Editar Eliminar |

Inventario

El previo diseño muestra una tabla sencilla con formularios planos para agregar o editar productos, manteniendo una estructura funcional pero sin la estética refinada ni la presentación visual detallada del diseño final.

The wireframe illustrates a user interface for managing inventory. At the top, there is a navigation bar with links for Logo, Clientes, Cajeros, Proveedores, Inventario (which is the active tab), Recompensas, Estadísticas, and Cerrar Sesión (Logout). Below the navigation bar is a search bar labeled "Buscar productos...". To the right of the search bar is a vertical menu with two items: "Productos" (selected) and "Ingredientes". Further to the right are three buttons: Agregar (Add), Editar (Edit), and Eliminar (Delete). The main content area displays a table of products with the following columns: ID Producto, Nombre, Categoría, Precio, and Stock. The table contains five rows of data:

| ID Producto | Nombre | Categoría | Precio | Stock |
|-------------|---------------------|-----------|---------|-------|
| PR001 | Pastel de Chocolate | Pasteles | \$35.00 | 15 |
| PR002 | Tarta de Fresa | Tartas | \$28.00 | 20 |
| PR003 | Cupcakes Vainilla | Cupcakes | \$12.00 | 50 |
| PR004 | Pan Francés | Pan | \$8.00 | 30 |
| PR005 | Galletas de Avena | Galletas | \$6.00 | 40 |

 Logo

[Clientes](#)

[Cajeros](#)

[Proveedores](#)

[Inventario](#)

[Recomper](#)

| ID Producto | Nombre | Categoría | Precio |
|-------------|---------------------|-----------|---------|
| PR001 | Pastel de Chocolate | Pasteles | \$35.00 |
| PR002 | Tarta de Fresa | Tartas | \$28.00 |
| PR003 | Cupcakes Vainilla | Cupcakes | \$12.00 |
| PR004 | Pan Francés | Pan | \$8.00 |
| PR005 | Galletas de Avena | Galletas | \$6.00 |

Agregar Producto

Nombre

Categoría

Precio

Stock

The screenshot displays a user interface for managing a bakery's inventory. At the top, there is a navigation bar with icons for Logo, Clientes, Cajeros, Proveedores, Inventario, and Recomper. Below the navigation is a search bar labeled "Buscar productos...". A modal window titled "Editar Producto" is open on the right side, containing fields for Nombre (Name), Categoría (Category), Precio (Price), and Stock. The main content area shows a table of products with columns for ID Producto, Nombre, Categoría, and Precio. The products listed are:

| ID Producto | Nombre | Categoría | Precio |
|-------------|---------------------|-----------|---------|
| PR001 | Pastel de Chocolate | Pasteles | \$35.00 |
| PR002 | Tarta de Fresa | Tartas | \$28.00 |
| PR003 | Cupcakes Vainilla | Cupcakes | \$12.00 |
| PR004 | Pan Francés | Pan | \$8.00 |
| PR005 | Galletas de Avena | Galletas | \$6.00 |

At the bottom of the modal, there are "Guardar" (Save) and "Cancelar" (Cancel) buttons.

El diseño final presenta una interfaz moderna y visualmente clara, con imágenes de productos, acciones bien identificadas y un estilo más profesional que facilita la gestión del inventario.



Gestión de Inventario

Productos Ingredientes

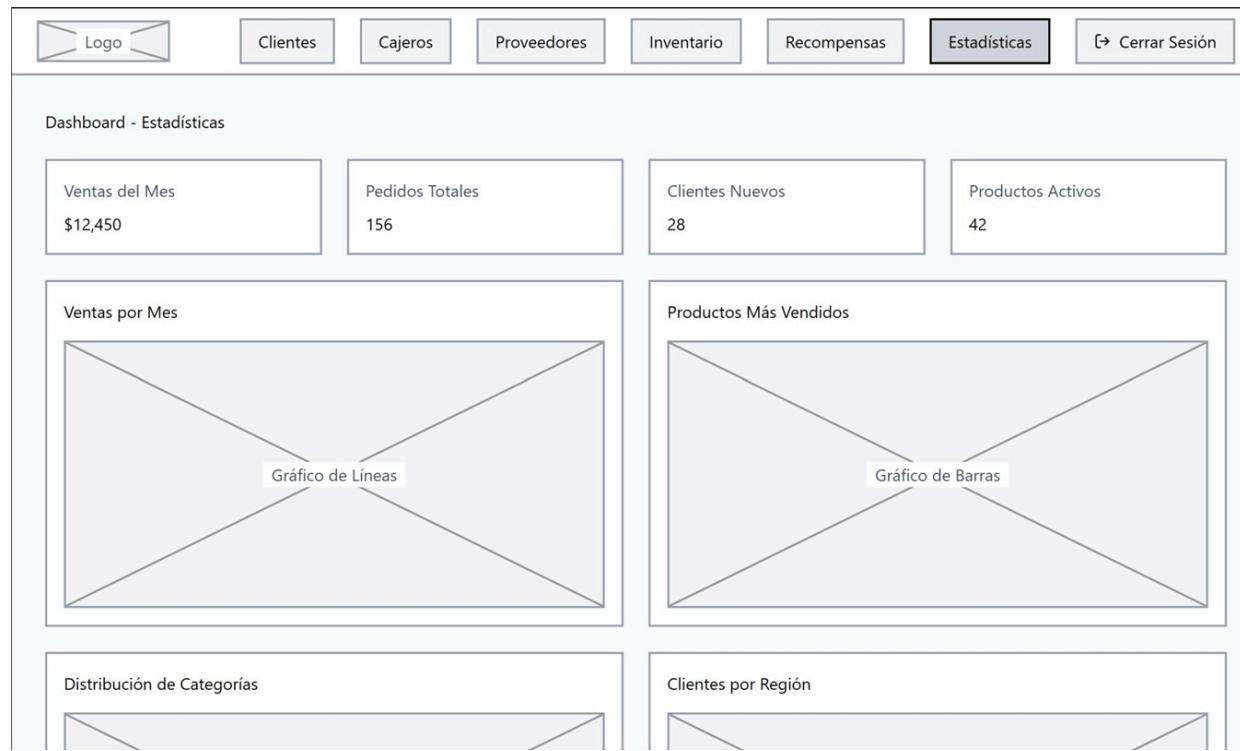
Buscar productos por nombre o SKU...

Agregar Producto

| SKU | IMAGEN | NOMBRE | PRECIO | ACCIÓN | ACCIONES |
|------|--------|-------------------------|----------|--------|---|
| PASC | | Pastel de Chocolate | \$350.00 | 4 | Editar Eliminar |
| PASO | | Pastel de oreo | \$300.00 | 19 | Editar Eliminar |
| PASF | | Pastel de Fresa y Crema | \$380.00 | 9 | Editar Eliminar |
| | | | | | |

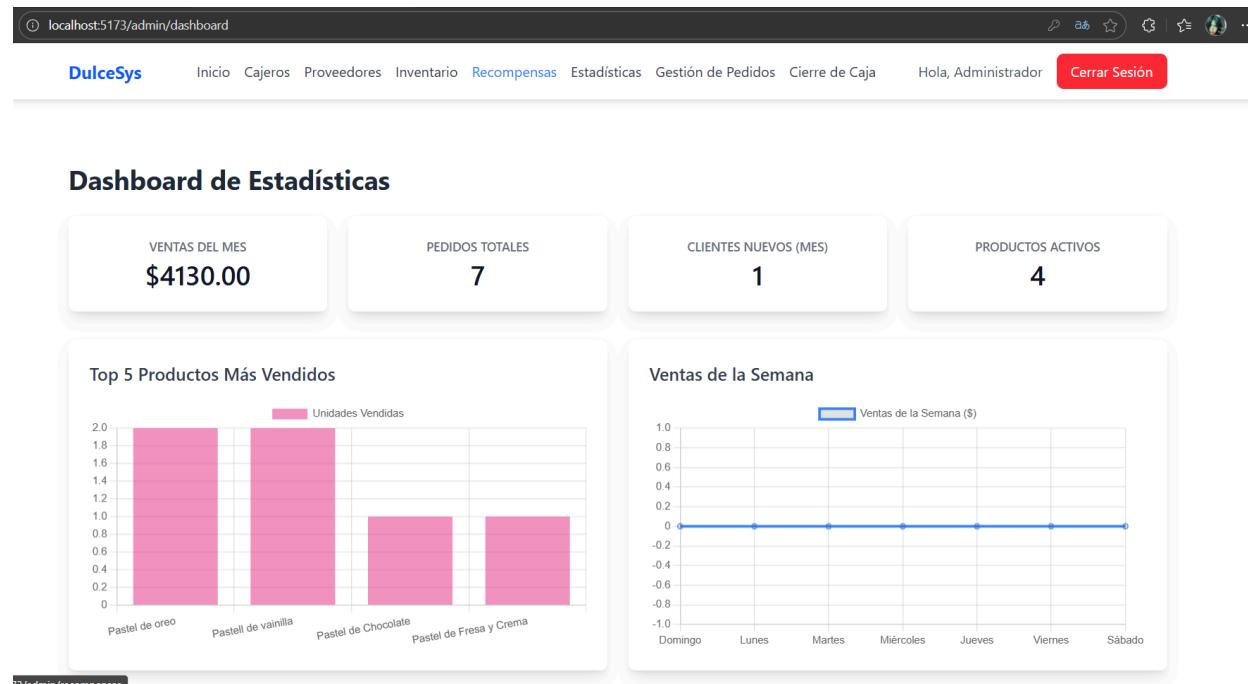
Estadísticas

El diseño previo del dashboard de estadísticas se mostraba únicamente como un conjunto de recuadros genéricos y cajas vacías destinadas para gráficos, sin ningún tipo de indicador visual real. Todo estaba representado mediante placeholders sin funcionalidad.



En la versión final, el dashboard se implementó con datos reales y diseño profesional, incluyendo tarjetas con valores importantes como ventas del mes, pedidos totales, clientes nuevos y productos activos. Los gráficos se muestran con estilos modernos, colores diferenciados y métricas claras. La distribución también mejora, permitiendo una visualización más ordenada de los gráficos más

vendidos, ventas por semana y distribución de categorías. El resultado final es un dashboard completo y visualmente atractivo que aporta información útil al administrador del sistema.



• CAJERO

Inicio:

En el diseño inicial, la vista del cajero consistía únicamente en una tabla sencilla de pedidos por salir, con clientes, productos, cantidades y estados. La vista era completamente plana y sin estilo visual distintivo.

The screenshot shows a user interface for a cashier system. At the top, there is a navigation bar with a logo icon and four buttons: 'Inicio' (highlighted in black), 'Inventario', 'Corte de Caja', and 'Cerrar Sesión'. Below the navigation bar is a section titled '[?] Pedidos por Salir' containing a table of pending orders. The table has columns: '#', 'ID', 'Cliente', 'Producto', 'Cantidad', 'Hora', and 'Estado'. The data in the table is as follows:

| # | ID | Cliente | Producto | Cantidad | Hora | Estado |
|-------|-------------------|-------------------------|----------|----------------|--------------------|--------|
| [#] 1 | [Φ] Juan Pérez | [▼] Pastel de Chocolate | [#] 1 | ⌚ [🕒] 10:30 AM | [●] En preparación | |
| [#] 2 | [Φ] María García | [▼] Cupcakes x12 | [#] 2 | ⌚ [🕒] 10:45 AM | [●] Listo | |
| [#] 3 | [Φ] Carlos López | [▼] Pastel de Vainilla | [#] 1 | ⌚ [🕒] 11:00 AM | [●] En preparación | |
| [#] 4 | [Φ] Ana Martínez | [▼] Galletas x24 | [#] 3 | ⌚ [🕒] 11:15 AM | [●] Listo | |
| [#] 5 | [Φ] Pedro Sánchez | [▼] Pastel Red Velvet | [#] 1 | ⌚ [🕒] 11:30 AM | [●] En preparación | |
| [#] 6 | [Φ] Laura Torres | [▼] Brownies x6 | [#] 2 | ⌚ [🕒] 11:45 AM | [●] Listo | |

At the top right of the 'Pedidos por Salir' section is a button '+ [+ Agregar Pedido]'. The entire interface has a plain, functional design.

En la versión final, esta sección se vuelve más moderna y coherente con el resto del sistema, integrando una cabecera clara, un menú mejor organizado y tarjetas de productos al fondo para facilitar la navegación del cajero. La interfaz ahora muestra un entorno más funcional y agradable, con botones estilizados y una organización más limpia, permitiendo al cajero interactuar más fácilmente con el sistema y controlar los pedidos en tiempo real.

Buscar productos...

Nuestros Productos



Pastel de Chocolate

Clásico pastel de chocolate

\$350



Pastel de oreo

Pastel de 3 leches, hecho con gall...

\$300



Pastel de Fresa y Crema

Pastel de Fresas con crema, 3 lech...

\$380



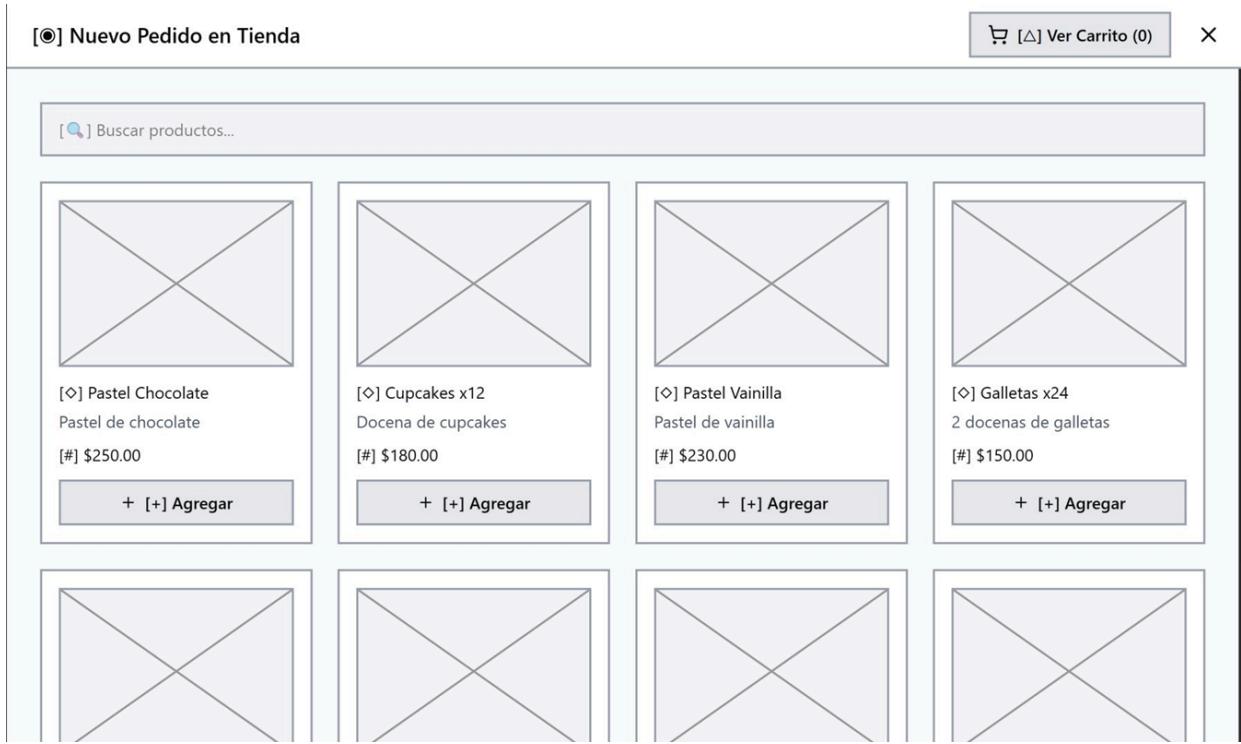
Pastell de vainilla

Pastel de Vainilla, clásico...

\$350

Pedido:

En el diseño previo, la gestión de pedidos estaba planteada como un catálogo con productos en tarjetas y un carrito lateral para crear pedidos en tienda. La interfaz mostraba solo la parte visual de los productos sin ninguna tabla de seguimiento.



En la versión final, el sistema evoluciona hacia una vista más administrativa y funcional, donde se presentan listas de pedidos activos e históricos, con columnas para cliente, hora, método de pago, total y estado. El diseño final está claramente orientado a la gestión real de pedidos, mostrando botones como “Procesar pago” y estados como “En preparación” o “Listo”. Esto convierte la vista en una herramienta operativa completa para cajeros y administradores.

Gestión de Pedidos

[Pedidos Activos](#) [Historial \(Terminados\)](#)

Buscar por ID de pedido o nombre de cliente...

| IDENTIFICACIÓN | CLIENTE | HORA | MÉTODO PAGO | TOTAL | ESTADO | ACCIÓN |
|----------------|--------------|-------------|---------------|-----------|----------------|-------------------------------|
| #2 | Ana Martínez | 20:49 horas | | \$1200.00 | En preparación | Procesar Pago |
| #4 | Ana Martínez | 21:26 horas | Tarjeta | \$350.00 | Listo | Procesar Pago |
| #5 | Ana Martínez | 21:39 horas | Transferencia | \$300.00 | Listo | Procesar Pago |

Inventario:

En el diseño previo, el carrito de compras se planteaba como un panel lateral que se desplegaba junto con la vista del catálogo, permitiendo ver simultáneamente los productos disponibles y el listado de artículos agregados. La interfaz mostraba elementos muy esquemáticos, con tarjetas simples, íconos genéricos y botones básicos sin jerarquía visual. El subtotal y el botón para procesar el pago se colocaban dentro del mismo panel, siguiendo una estructura más parecida a un prototipo conceptual.

Este prototipo conceptual muestra la visualización de inventario. En la parte superior, hay un menú horizontal con un logo, botones para 'Inicio', 'Inventario' (destacado en azul), 'Corte de Caja' y 'Cerrar Sesión'. La sección principal titulada '[?] Visualización de Inventario' incluye filtros para 'Productos' y 'Ingredientes', y un campo de búsqueda '[?] Buscar...'. Una lista detallada de cinco productos muestra columnas para ID, Nombre, Categoría, Precio y Stock.

| [#] ID | [◊] Nombre | [▼] Categoría | [#] Precio | [#] Stock |
|--------|-------------------------|---------------|--------------|-----------|
| [#] 1 | [◊] Pastel de Chocolate | [▼] Pasteles | [#] \$450.00 | [#] 12 |
| [#] 2 | [◊] Cupcakes | [▼] Pequeños | [#] \$35.00 | [#] 48 |
| [#] 3 | [◊] Galletas | [▼] Pequeños | [#] \$25.00 | [#] 120 |
| [#] 4 | [◊] Pastel de Vainilla | [▼] Pasteles | [#] \$420.00 | [#] 8 |
| [#] 5 | [◊] Brownies | [▼] Pequeños | [#] \$40.00 | [#] 36 |

En la versión final, el carrito se convirtió en una página dedicada, con un diseño más limpio, centrado y enfocado en facilitar la lectura de la información. La interfaz utiliza tarjetas modernas, texto estilizado y un botón de acción destacado en color, además de

mensajes claros como “Tu carrito está vacío” cuando no hay elementos. Todo el diseño se enfoca en ofrecer una experiencia más clara, minimalista y fácil de entender para el cliente.

Gestión de Inventario

| Productos | Ingredientes |
|--------------------------------------|---|
| Buscar productos por nombre o SKU... | |
| SKU | IMAGEN |
| PASC |  |
| | Pastel de Chocolate |
| | \$350.00 |
| | 4 |
| PASO |  |
| | Pastel de oreo |
| | \$300.00 |
| | 19 |
| PASF |  |
| | Pastel de Fresa y Crema |
| | \$380.00 |
| | 9 |
| PASV |  |
| | Pastell de vainilla |
| | \$350.00 |
| | 14 |

Corte de caja:

El diseño inicial del corte de caja mostraba un panel técnico con totales de ventas, efectivo, tarjeta y transferencia en cajas simples, seguido de una tabla detallada con los productos vendidos y cantidades. Esta versión tenía un estilo rígido, parecido a un sistema administrativo interno.

Logo

Inicio | Inventario | Corte de Caja | Cerrar Sesión

⌚ [●] 02 de octubre de 2025

฿ [●] Corte de Caja

| | | | |
|--------------------------------------|-------------------------------|------------------------------|------------------------------------|
| ▲ [=] Total Ventas [#] \$10630.00 | [■] Efectivo [#] \$4252.00 | [■] Tarjeta [#] \$4783.50 | [■] Transferencia [#] \$1594.50 |
|--------------------------------------|-------------------------------|------------------------------|------------------------------------|

⌚ [▼] Productos Vendidos

| [#] ID | [◊] Producto | [#] Cantidad | [#] Precio Unitario | [=] Total |
|--------|-------------------------|--------------|---------------------|---------------|
| [#] 1 | [◊] Pastel de Chocolate | [#] 5 | [#] \$450.00 | [=] \$2250.00 |
| [#] 2 | [◊] Cupcakes x12 | [#] 8 | [#] \$200.00 | [=] \$1600.00 |
| [#] 3 | [◊] Galletas x24 | [#] 12 | [#] \$150.00 | [=] \$1800.00 |
| [#] 4 | [◊] Pastel de Vainilla | [#] 3 | [#] \$420.00 | [=] \$1260.00 |
| [#] 5 | [◊] Brownies x6 | [#] 10 | [#] \$180.00 | [=] \$1800.00 |

[△] Balance de Inventario

En la versión final, se modernizó por completo la interfaz, incorporando tarjetas de colores para cada tipo de ingreso, lo que facilita la lectura inmediata de los totales del día. Además, el diseño final organiza la información con secciones separadas, gráficos y tablas mejor distribuidas, logrando un aspecto más profesional y alineado con dashboards actuales. El selector de fecha se integró de manera más natural en la cabecera, y el balance de inventario se muestra con una estructura más clara. La apariencia final ofrece una lectura visual mucho más atractiva y una experiencia más intuitiva para el administrador o cajero.

Corte de Caja

13/11/2025



Total de Ventas (13 de noviembre de 2025)

TOTAL VENTAS
\$0.00

Efectivo
\$0.00

TARJETA
\$0.00

TRANSFERENCIA
\$0.00

Productos Vendidos Hoy

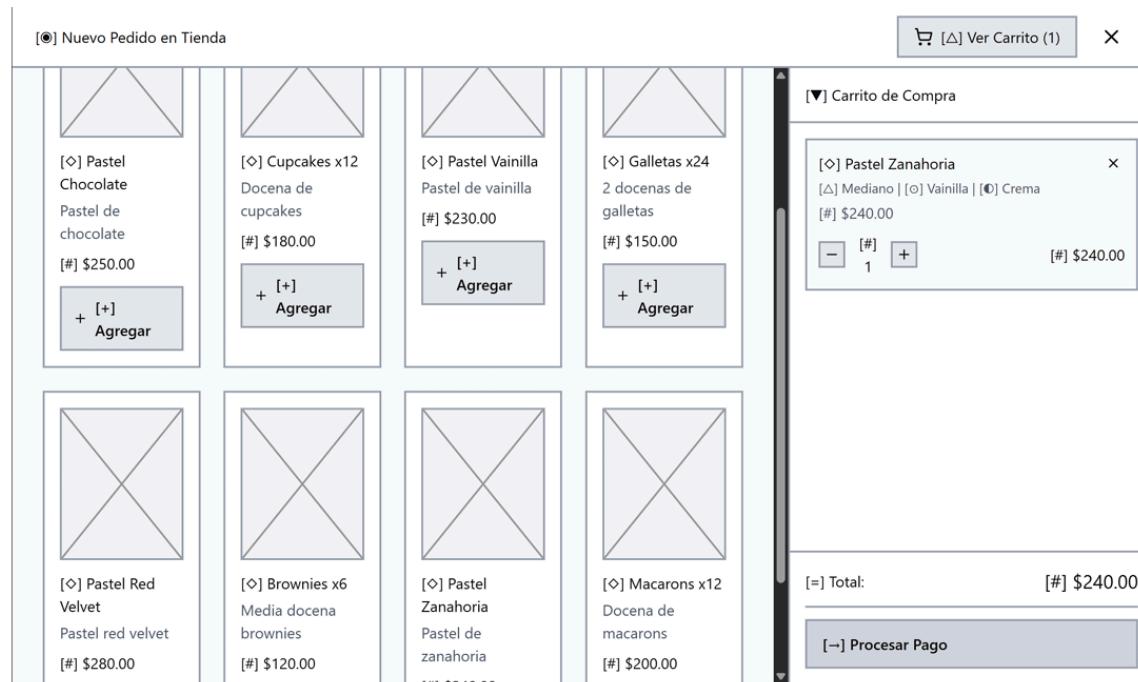
| PRODUCTO | HIPOCRESÍA. | TOTAL |
|----------|-------------|-------|
| | | |

Stock Actual de Ingredientes

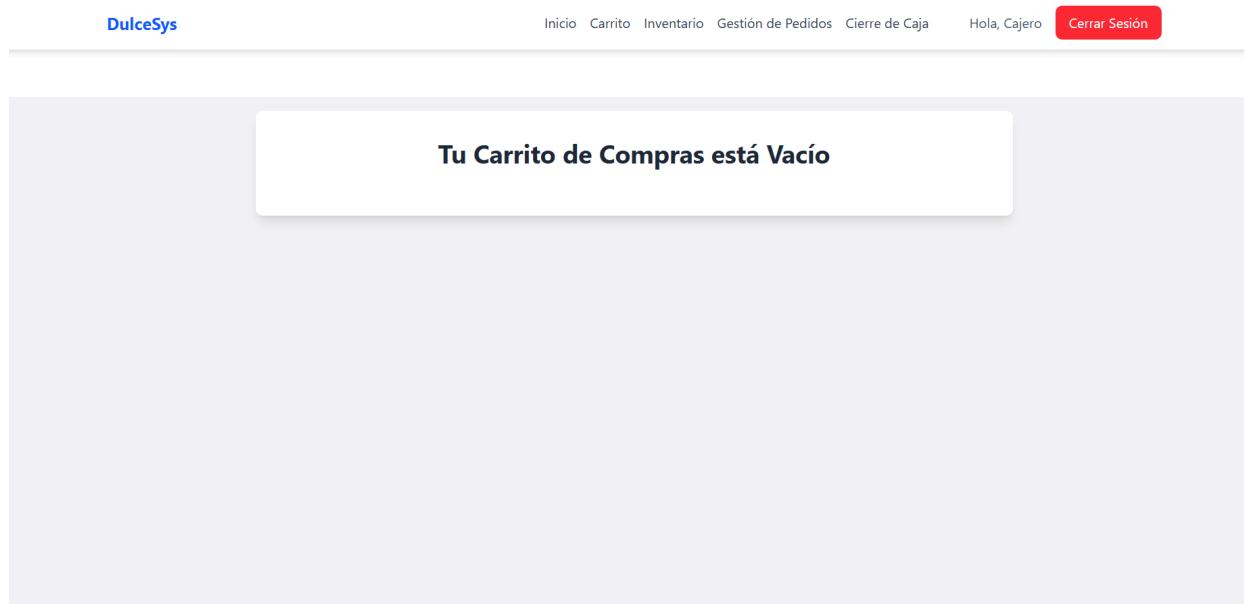
| INGREDIENTE | STOCK REAL | STOCK MÍNIMO |
|------------------|-------------------|--------------|
| Fresa | 10,00 kg | 2,00 kg |
| Chocolate Amargo | 15,00 kg | 3,00 kg |
| Leche | 20,00 lt | 5,00 lt |
| Azúcar | 50,00 kg | 10,00 kg |
| Harina | 50,00 kg | 10,00 kg |
| Huevo | 100,00 pza | 24,00 pza |

Carrito de compras:

En el diseño previo, el carrito de compras se planteaba como un panel lateral que se desplegaba junto con la vista del catálogo, permitiendo ver simultáneamente los productos disponibles y el listado de artículos agregados. La interfaz mostraba elementos muy esquemáticos, con tarjetas simples, íconos genéricos y botones básicos sin jerarquía visual. El subtotal y el botón para procesar el pago se colocaban dentro del mismo panel, siguiendo una estructura más parecida a un prototipo conceptual.



En la versión final, el carrito se convirtió en una página dedicada, con un diseño más limpio, centrado y enfocado en facilitar la lectura de la información. La interfaz utiliza tarjetas modernas, texto estilizado y un botón de acción destacado en color, además de mensajes claros como “Tu carrito está vacío” cuando no hay elementos. Todo el diseño se enfoca en ofrecer una experiencia más clara, minimalista y fácil de entender para el cliente.



MANUAL DE CONSULTAS

MANUAL DE CONSULTAS

1. AUTENTICACIÓN

- LOGIN

Buscar usuario por correo

```
await prisma.usuarios.findUnique({  
  where: { correoelectronico: correo },  
  include: { roles: true }  
});
```

Comparar contraseña

(No es consulta Prisma, pero parte del proceso)
await bcrypt.compare(contrasena, usuario.contrasena);

Buscar rol Cliente

```
await prisma.roles.findUnique({  
  where: { nombrerol: 'Cliente' },  
  select: { id_rol: true }  
});
```

- REGISTRO

Crear usuario + cliente en transacción

```
await prisma.$transaction(async (tx) => {  
  const nuevoUsuario = await tx.usuarios.create({  
    data: {  
      correoelectronico: correo,  
      contrasena: hashedPassword,  
      id_rol: clienteRole.id_rol,  
    }  
  });  
  const nuevoCliente = await tx.clientes.create({  
    data: {  
      id_usuario: nuevoUsuario.id_usuario,  
      nombre: nombre,  
      telefono: telefono  
    }  
  });  
  return nuevoCliente;
```

```
});
```

2. CARRITO

Obtener carrito

```
await prisma.carrito_items.findMany({  
    where: { id_usuario: idUsuario },  
    include: { productos: true }  
});
```

Agregar item al carrito

```
await prisma.carrito_items.create({  
    data: {  
        id_usuario: idUsuario,  
        id_producto: id_producto,  
        cantidad: cantidad,  
        personalizacion: personalizacion  
    }  
});
```

Eliminar item del carrito

```
await prisma.carrito_items.delete({  
    where: { id_item_carrito: parseInt(itemId) }  
});
```

3. PRODUCTOS

Obtener todos los productos

```
await prisma.productos.findMany({  
    where: { activo: true }  
});
```

Obtener un solo producto (detalle)

```
await prisma.productos.findUnique({  
    where: { id_producto: parseInt(id) }  
});
```

Obtener atributos del producto

```
await prisma.product_attributes.findMany({  
    include: { attribute_options: true }  
});
```

```
});
```

Obtener ID de cliente según usuario

```
await prisma.clientes.findUnique({
  where: { id_usuario: idUsuarioLogueado }
});
```

4. PEDIDOS (Historial)

Obtener pedidos del cliente

```
await prisma_pedidos.findMany({
  where: { id_cliente: cliente.id_cliente, activo: true },
  orderBy: { fechapedido: 'desc' }
});
```

Crear pedido (Transacción Completa)

Incluye validar stock, descontar stock, crear pedido, detalle y limpiar carrito:

```
await prisma.$transaction(async (tx) => {
  const pedido = await tx.pedidos.create({
    data: {
      id_cliente: clientePedidold,
      id_empleado: empleadold,
      total: total,
      estado: estado || 'Pendiente',
      metodo_pago: metodoPago,
      monto_pago_con: montoPagoCon
    }
  });
});
```

```
await tx.detalle_pedido.createMany({
  data: itemsParaDetalle.map(item => ({
    ...item,
    id_pedido: pedido.id_pedido
  }))
});
```

```
await tx.carrito_items.deleteMany({
  where: { id_usuario: idUsuario }
});
```

```
return pedido;
});
```

5. GESTIÓN DE PEDIDOS (Admin/Cajero)

Obtener pedidos con filtros dinámicos

```
await prisma_pedidos.findMany({  
  where: {  
    activo: true,  
    estado: statusFilter  
  },  
  include: {  
    clientes: { select: { nombre: true } },  
    detalle_pedido: {  
      take: 1,  
      include: { productos: { select: { nombre: true } } }  
    }  
  },  
  orderBy: orderByFilter  
});
```

Actualizar estado del pedido

```
await prisma_pedidos.update({  
  where: { id_pedido: parseInt(id) },  
  data: { estado }  
});
```

6. INVENTARIO — PRODUCTOS (Admin)

Crear producto

```
await prisma_productos.create({  
  data: {  
    sku,  
    nombre,  
    descripcion,  
    preciobase: parseFloat(precioBase),  
    id_categoria: parseInt(id_categoria),  
    stockproductostterminados: parseInt(stockproductostterminados),  
    imagenurl: imageURL  
  }  
});
```

```
    }  
});
```

Actualizar producto

```
await prisma.productos.update({  
  where: { id_producto: parseInt(id) },  
  data: dataToUpdate  
});  
Desactivar producto (borrado lógico)  
await prisma.productos.update({  
  where: { id_producto: parseInt(id) },  
  data: { activo: false }  
});
```

7. INGREDIENTES (Admin)

Listar ingredientes

```
await prisma.ingredientes.findMany({  
  where: { activo: true }  
});
```

Crear ingrediente

```
await prisma.ingredientes.create({  
  data: {  
    sku,  
    nombre,  
    stockactual: parseFloat(stockactual),  
    stockminimo: parseFloat(stockminimo),  
    unidadmedida,  
    id_proveedor  
  }  
});
```

Actualizar ingrediente

```
await prisma.ingredientes.update({  
  where: { id_ingrediente: parseInt(id) },  
  data: {  
    sku,  
    nombre,  
    stockactual: parseFloat(stockactual),  
  }  
});
```

```
    stockminimo,  
    unidadmedida,  
    id_proveedor  
}  
});
```

Desactivar ingrediente

```
await prisma.ingredientes.update({  
  where: { id_ingrediente: parseInt(id) },  
  data: { activo: false }  
});
```

8. PROVEEDORES (Admin)

Listar proveedores

```
await prisma.proveedores.findMany({  
  where: { activo: true },  
  orderBy: { nombre: 'asc' }  
});
```

Crear proveedor

```
await prisma.proveedores.create({  
  data: { nombre, contacto, telefono, rfc }  
});
```

Actualizar proveedor

```
await prisma.proveedores.update({  
  where: { id_proveedor: parseInt(id) },  
  data: { nombre, contacto, telefono, rfc }  
});
```

Desactivar proveedor

```
await prisma.proveedores.update({  
  where: { id_proveedor: parseInt(id) },  
  data: { activo: false }  
});
```

9. CAJEROS (Admin)

Listar cajeros

```
await prisma.empleados.findMany({  
  where: {  
    activo: true,  
    usuarios: {  
      roles: { nombrerol: 'Cajero' }  
    }  
  },  
  include: {  
    usuarios: { select: { correoElectronico: true } }  
  }  
});
```

Crear cajero (transacción)

```
await prisma.$transaction(async (tx) => {  
  const nuevoUsuario = await tx.usuarios.create({ ... });  
  const nuevoEmpleado = await tx.empleados.create({ ... });  
  return nuevoEmpleado;  
});
```

Actualizar cajero (transacción)

```
await prisma.$transaction(async (tx) => {  
  const empleado = await tx.empleados.update({ ... });  
  await tx.usuarios.update({ ... });  
});
```

Borrado lógico (transacción)

```
await prisma.$transaction(async (tx) => {  
  await tx.empleados.update({ ... });  
  await tx.usuarios.update({ ... });  
});
```

10. RECOMPENSAS (Admin)

Listar recompensas

```
await prisma.recompensas.findMany({  
  where: { activo: true }  
});
```

Crear recompensa

```
await prisma.recompensas.create({  
  data: { nombrerecompensa, descripcion, tipo, valor, puntosrequeridos }  
});
```

Actualizar recompensa

```
await prisma.recompensas.update({  
  where: { id_recompensa: parseInt(id) },  
  data: { nombrerecompensa, descripcion, tipo, valor, puntosrequeridos }  
});
```

Desactivar recompensa

```
await prisma.recompensas.update({  
  where: { id_recompensa: parseInt(id) },  
  data: { activo: false }  
});
```

11. DASHBOARD (Admin)

Ventas del mes

```
await prisma_pedidos.aggregate({  
  _sum: { total: true },
```

```
    where: { fechapedido: { gte: inicioDeMes, lte: finDeMes } }  
});
```

Conteos

```
await prisma.pedidos.count();  
await prisma.clientes.count({ where: { fecharegistro: { gte: inicioDeMes, lte: finDeMes } } });  
await prisma.productos.count({ where: { activo: true } });
```

Top productos

```
await prisma.detalle_pedido.groupBy({  
  by: ['id_producto'],  
  _sum: { cantidad: true },  
  orderBy: { _sum: { cantidad: 'desc' } },  
  take: 5  
});
```

Ventas por día (RAW SQL)

```
await prisma.$queryRaw`  
SELECT EXTRACT(DOW FROM fechapedido) AS "diaSemana", SUM(total) AS  
"totalVentas"  
FROM pedidos  
WHERE fechapedido BETWEEN ${inicioDeSemana} AND ${finDeSemana}  
GROUP BY "diaSemana"  
ORDER BY "diaSemana";  
`;
```

Tamaños más vendidos

```
await prisma.$queryRaw`  
SELECT (personalizacion->>'Tamaño') AS "tamano", SUM(cantidad) AS "totalVendido"  
FROM detalle_pedido  
WHERE personalizacion->>'Tamaño' IS NOT NULL  
GROUP BY "tamano"  
ORDER BY "totalVendido" DESC;  
`;
```

Ventas por cajero

```
await prisma.$queryRaw`  
SELECT e.nombrecompleto, COUNT(p.id_pedido) AS "totalVentas"
```

```
FROM pedidos p
JOIN empleados e ON p.id_empleado = e.id_empleado
WHERE p.id_empleado IS NOT NULL
GROUP BY e.nombrecompleto
ORDER BY "totalVentas" DESC;
';
```

12. CORTE DE CAJA

Ventas por método de pago

```
await prisma_pedidos.groupBy({
  by: ['metodo_pago'],
  _sum: { total: true },
  where: {
    estado: 'Completado',
    fechapedido: { gte: inicioDelDia, lte: finDelDia }
  }
});
```

Productos vendidos hoy

```
await prisma_detalle_pedido.groupBy({
  by: ['id_producto'],
  _sum: { cantidad: true, preciounitario: true },
  where: {
    pedidos: {
      estado: 'Completado',
      fechapedido: { gte: inicioDelDia, lte: finDelDia }
    }
  }
});
```

Inventario actual

```
await prisma_ingredientes.findMany({
  where: { activo: true },
  select: { nombre: true, stockactual: true, stockminimo: true, unidadmedida: true },
  orderBy: { stockactual: 'asc' }));
```

MANUAL DE CONSULTAS EN SQL

1.- Autenticación

LOGIN: Buscar usuario por correo

```
SELECT u.*, r.*  
FROM usuarios u  
INNER JOIN roles r ON u.id_rol = r.id_rol  
WHERE u.correoelectronico = 'correo@ejemplo.com';
```

Buscar rol Cliente

```
SELECT id_rol  
FROM roles  
WHERE nombrerol = 'Cliente';
```

REGISTRO: Crear usuario + cliente en transacción

```
BEGIN;  
INSERT INTO usuarios (correoelectronico, contrasena, id_rol)  
VALUES ('correo@ejemplo.com', 'hashedPassword', 1)  
RETURNING id_usuario;  
  
INSERT INTO clientes (id_usuario, nombre, telefono)  
VALUES (1, 'Juan Pérez', '1234567890')  
RETURNING *;  
COMMIT;
```

2.- CARRITO

Obtener carrito

```
SELECT ci.*, p.*  
FROM carrito_items ci  
INNER JOIN productos p ON ci.id_producto = p.id_producto  
WHERE ci.id_usuario = 1;
```

Agregar item al carrito

```
INSERT INTO carrito_items (id_usuario, id_producto, cantidad, personalizacion)
VALUES (1, 5, 2, '{"Tamaño": "Grande", "Sabor": "Chocolate"}');
```

Eliminar item del carrito

```
DELETE FROM carrito_items
WHERE id_item_carrito = 10;
```

3.- PRODUCTOS

Obtener todos los productos activos

```
SELECT * FROM productos
WHERE activo = true;
```

Obtener un solo producto (detalle)

```
SELECT * FROM productos
WHERE id_producto = 5;
```

Obtener atributos del producto

```
SELECT pa.*, ao.*
FROM product_attributes pa
INNER JOIN attribute_options ao ON pa.id_attribute = ao.id_attribute;
```

Obtener ID de cliente según usuario

```
SELECT * FROM clientes
WHERE id_usuario = 1;
```

4-PEDIDOS

Obtener pedidos del cliente

```
SELECT * FROM pedidos  
WHERE id_cliente = 1 AND activo = true  
ORDER BY fechapedido DESC;
```

Crear pedido (Transacción Completa)

```
BEGIN;  
    Crear pedido  
    INSERT INTO pedidos (id_cliente, id_empleado, total, estado, metodo_pago,  
    monto_pago_con)  
    VALUES (1, 2, 150.00, 'Pendiente', 'Efectivo', 200.00)  
    RETURNING id_pedido;
```

Crear detalle del pedido

```
INSERT INTO detalle_pedido (id_pedido, id_producto, cantidad, preciounitario,  
personalizacion)  
VALUES  
(1, 5, 2, 50.00, '{"Tamaño": "Grande"}'),  
(1, 7, 1, 50.00, '{"Tamaño": "Mediano"}');
```

Actualizar stock de productos

```
UPDATE productos  
SET stockproductosterminados = stockproductosterminados - 2  
WHERE id_producto = 5;
```

Limpiar carrito del usuario

```
DELETE FROM carrito_items  
WHERE id_usuario = 1;  
COMMIT;
```

5.- GESTIÓN DE PEDIDOS

Obtener pedidos con filtros dinámicos

```
SELECT
    p.*,
    c.nombre AS nombre_cliente,
    dp.id_producto,
    pr.nombre AS nombre_producto
FROM pedidos p
INNER JOIN clientes c ON p.id_cliente = c.id_cliente
LEFT JOIN LATERAL (
    SELECT * FROM detalle_pedido
    WHERE id_pedido = p.id_pedido
    LIMIT 1
) dp ON true
LEFT JOIN productos pr ON dp.id_producto = pr.id_producto
WHERE p.activo = true
    AND p.estado = 'Pendiente' -- Filtro opcional
ORDER BY p.fechapedido DESC;
```

Actualizar estado del pedido

```
UPDATE pedidos
SET estado = 'Completado'
WHERE id_pedido = 1;
```

6.- INVENTARIO

Crear producto

```
INSERT INTO productos (sku, nombre, descripcion, preciobase, id_categoria,  
stockproductostterminados, imagenurl)  
VALUES ('SKU001', 'Pastel de Chocolate', 'Delicioso pastel', 50.00, 1, 10, 'url-imagen.jpg');
```

Actualizar producto

UPDATE productos

```
SET nombre = 'Pastel de Chocolate Premium',  
preciobase = 55.00,  
stockproductostterminados = 15  
WHERE id_producto = 5;
```

Desactivar producto

UPDATE productos

```
SET activo = false  
WHERE id_producto = 5;
```

7.- INGREDIENTES

Listar ingredientes activos

```
SELECT * FROM ingredientes  
WHERE activo = true;
```

Crear ingrediente

```
INSERT INTO ingredientes (sku, nombre, stockactual, stockminimo, unidadmedida,  
id_proveedor)  
VALUES ('ING001', 'Harina', 50.5, 10.0, 'kg', 1);
```

Actualizar ingrediente

```
UPDATE ingredientes
SET nombre = 'Harina Integral',
stockactual = 60.0,
stockminimo = 15.0,
unidadmedida = 'kg',
id_proveedor = 2
WHERE id_ingrediente = 1;
```

Desactivar ingrediente

```
UPDATE ingredientes
SET activo = false
WHERE id_ingrediente = 1;
```

8. PROVEEDORES(admin)

Listar proveedores activos

```
SELECT * FROM proveedores
WHERE activo = true
ORDER BY nombre ASC;
```

Crear proveedor

```
INSERT INTO proveedores (nombre, contacto, telefono, rfc)
VALUES ('Proveedor ABC', 'Juan López', '1234567890', 'ABC123456XYZ');
```

Actualizar proveedor

```
UPDATE proveedores
SET nombre = 'Proveedor ABC S.A.',
contacto = 'María García',
telefono = '0987654321',
rfc = 'ABC654321XYZ'
WHERE id_proveedor = 1;
```

Desactivar proveedor

```
UPDATE proveedores  
SET activo = false  
WHERE id_proveedor = 1;
```

9.- CAJERO (admin)

Listar cajeros activos

```
SELECT  
    e.*,  
    u.correoelectronico  
FROM empleados e  
INNER JOIN usuarios u ON e.id_usuario = u.id_usuario  
INNER JOIN roles r ON u.id_rol = r.id_rol  
WHERE e.activo = true  
AND r.nombrerol = 'Cajero';
```

Crear cajero (transacción)

```
BEGIN;  
INSERT INTO usuarios (correoelectronico, contrasena, id_rol)  
VALUES ('cajero@ejemplo.com', 'hashedPassword', 2)  
RETURNING id_usuario;
```

```
INSERT INTO empleados (id_usuario, nombrecompleto, telefono, puesto)  
VALUES (5, 'Pedro Martínez', '5551234567', 'Cajero')  
RETURNING *;  
COMMIT;
```

Actualizar cajero (transacción)

```
BEGIN;  
UPDATE empleados  
SET nombrecompleto = 'Pedro Martínez Gómez',  
    telefono = '5559876543'  
WHERE id_empleado = 1;
```

```
UPDATE usuarios
SET correoelectrónico = 'nuevo.cajero@ejemplo.com'
WHERE id_usuario = (SELECT id_usuario FROM empleados WHERE id_empleado = 1);
COMMIT;
```

Borrado lógico de cajero (transacción)

```
BEGIN;
UPDATE empleados
SET activo = false
WHERE id_empleado = 1;
```

```
UPDATE usuarios
SET activo = false
WHERE id_usuario = (SELECT id_usuario FROM empleados WHERE id_empleado = 1);
COMMIT;
```

10.-RECOMPENSAS

Listar recompensas activas

```
SELECT * FROM recompensas
WHERE activo = true;
```

Crear recompensa

```
INSERT INTO recompensas (nombrerecompensa, descripcion, tipo, valor, puntosrequeridos)
VALUES ('Descuento 10%', 'Descuento del 10% en tu próxima compra', 'Descuento', 10, 100);
```

Actualizar recompensa

```
UPDATE recompensas
SET nombrerecompensa = 'Descuento 15%',
descripcion = 'Descuento del 15% en tu próxima compra',
valor = 15,
puntosrequeridos = 150
WHERE id_recompensa = 1;
```

Desactivar recompensa

```
UPDATE recompensas  
SET activo = false  
WHERE id_recompensa = 1;
```

-11. DASHBOARD (Admin)

Ventas del mes

```
SELECT SUM(total) AS ventas_mes  
FROM pedidos  
WHERE fechapedido >= DATE_TRUNC('month', CURRENT_DATE)  
AND fechapedido <= DATE_TRUNC('month', CURRENT_DATE) + INTERVAL '1 month' -  
INTERVAL '1 day';
```

Conteo de pedidos

```
SELECT COUNT(*) AS total_pedidos  
FROM pedidos;
```

Clientes nuevos del mes

```
SELECT COUNT(*) AS clientes_nuevos  
FROM clientes  
WHERE fecharegistro >= DATE_TRUNC('month', CURRENT_DATE)  
AND fecharegistro <= DATE_TRUNC('month', CURRENT_DATE) + INTERVAL '1 month' -  
INTERVAL '1 day';
```

Productos activos

```
SELECT COUNT(*) AS productos_activos  
FROM productos  
WHERE activo = true;
```

Top 5 productos más vendidos

```
SELECT
dp.id_producto,
p.nombre,
SUM(dp.cantidad) AS cantidad_vendida
FROM detalle_pedido dp
INNER JOIN productos p ON dp.id_producto = p.id_producto
GROUP BY dp.id_producto, p.nombre
ORDER BY cantidad_vendida DESC
LIMIT 5;
```

Ventas por día de la semana

```
SELECT
EXTRACT(DOW FROM fechapedido) AS diaSemana,
SUM(total) AS totalVentas
FROM pedidos
WHERE fechapedido >= DATE_TRUNC('week', CURRENT_DATE)
AND fechapedido <= DATE_TRUNC('week', CURRENT_DATE) + INTERVAL '1 week' -
INTERVAL '1 day'
GROUP BY diaSemana
ORDER BY diaSemana;
```

Tamaños más vendidos

```
SELECT
personalizacion->>'Tamaño' AS tamano,
SUM(cantidad) AS totalVendido
FROM detalle_pedido
WHERE personalizacion->>'Tamaño' IS NOT NULL
GROUP BY tamano
ORDER BY totalVendido DESC;
```

Ventas por cajero

```
SELECT
    e.nombrecompleto,
    COUNT(p.id_pedido) AS totalVentas
FROM pedidos p
INNER JOIN empleados e ON p.id_empleado = e.id_empleado
WHERE p.id_empleado IS NOT NULL
GROUP BY e.nombrecompleto
ORDER BY totalVentas DESC;
```

12.- CORTE DE CAJA

Ventas por método de pago (hoy)

```
SELECT
    metodo_pago,
    SUM(total) AS total_por_metodo
FROM pedidos
WHERE estado = 'Completado'
    AND fechapedido >= CURRENT_DATE
    AND fechapedido < CURRENT_DATE + INTERVAL '1 day'
GROUP BY metodo_pago;
```

Productos vendidos hoy

```
SELECT
    dp.id_producto,
    p.nombre,
    SUM(dp.cantidad) AS cantidad_vendida,
    SUM(dp.preciounitario * dp.cantidad) AS total_ventas
FROM detalle_pedido dp
INNER JOIN pedidos pe ON dp.id_pedido = pe.id_pedido
INNER JOIN productos p ON dp.id_producto = p.id_producto
WHERE pe.estado = 'Completado'
    AND pe.fechapedido >= CURRENT_DATE
    AND pe.fechapedido < CURRENT_DATE + INTERVAL '1 day'
GROUP BY dp.id_producto, p.nombre
ORDER BY cantidad_vendida DESC;
```

Inventario actual (ordenado por stock bajo)

SELECT

 nombre,
 stockactual,
 stockminimo,
 unidadmedida

FROM ingredientes

WHERE activo = true

ORDER BY stockactual ASC;

TOTAL DE CONSULTAS: 47

DEVELOPER MANUAL

DEVELOPER MANUAL

El presente documento constituye el Manual del técnico del sistema DulceSys, una solución tecnológica desarrollada para la gestión integral de procesos en una pastelería o negocio de venta de productos personalizados, tales como pasteles, panes artesanales o postres.

Este manual describe detalladamente el funcionamiento interno del sistema, su arquitectura, estructura del código, lógica de interacción entre componentes y configuración necesaria para su ejecución. Está dirigido a programadores, desarrolladores o personal técnico responsable de mantener, actualizar o ampliar el software.

El sistema está diseñado como un **Monorepo** es decir, son 2 aplicaciones separadas dentro del mismo repositorio, una aplicación corresponde al Backend, y otra al Frontend, la comunicación entre ambas hace posible todas las transacciones y el uso del sistema.

- **frontend/**: Una aplicación de cliente (SPA) construida en React. Se ejecuta en el navegador del usuario.
- **backend/**: Una API REST construida en Node.js/Express. Sirve como el intermediario que se comunica con la base de datos y aplica toda la lógica de negocio.
- **Base de Datos**: Un servidor PostgreSQL que almacena todos los datos de forma persistente.

Para el desarrollo de este proyecto se ocuparon estas herramientas tanto en la parte de backend como de frontend:

| Frontend | Backend |
|---|--|
| React + Vite (<i>Framework</i>) React Router (<i>Enrutamiento</i>) Tailwind CSS (<i>Estilos</i>) AXIOS (<i>Cliente HTTP</i>) React Context API (<i>Estado global</i>) jwt-decode (<i>Leer Tokens</i>) react-chartjs-2 (<i>Gráficos estadísticos</i>) | Node.js (<i>Entorno</i>) Express.js (<i>Framework API</i>) ORM Prisma (<i>Interacción con BD</i>) jsonwebtoken (<i>Autenticación</i>) bcryptjs (<i>Encriptación</i>) multer (<i>Manejo de imágenes/archivos</i>) cors (<i>Comunicación con el frontend</i>) date-fns (<i>Para dashboard y corte de caja</i>) nodemon (<i>Reinicia en automático el backend</i>) react-hot-toast (<i>Manejo de errores/validación</i>) zod (<i>Manejo de “reglas”</i>) |

DESCRIPCIÓN GENERAL DEL SISTEMA.

DulceSys es una plataforma web integral diseñada específicamente para la gestión operativa de la pastelería "La Casa del Pastel". El sistema nace como respuesta a la necesidad crítica de digitalizar procesos manuales que generaban ineficiencias, tales como el descontrol de inventario, la lentitud en la generación de cotizaciones y la falta de herramientas de fidelización de clientes.

El sistema unifica en una sola plataforma dos mundos operativos:

1. **El E-commerce (Cliente Final):** Permitiendo la compra y personalización de productos en línea.
2. **El Punto de Venta y ERP (Administración Interna):** Permitiendo al personal gestionar ventas en mostrador, inventario, producción y finanzas.

A. Módulo de Cliente (E-commerce)

Enfocado en la experiencia de compra remota:

- ❖ Catálogo Interactivo: Visualización de productos con control de disponibilidad basado en stock real menos un stock de seguridad ("pivot").
- ❖ Personalización de Pedidos: Capacidad de configurar atributos dinámicos (tamaño, sabor, relleno) que ajustan el precio final en tiempo real.
- ❖ Carrito de Compras Persistente: Almacenamiento del carrito en base de datos para recuperar la sesión en cualquier dispositivo.
- ❖ Sistema de Pedidos y Voucher: Generación de pedidos con estado "Pendiente" y emisión de un ID de seguimiento (Voucher) para pago en tienda.
- ❖ Historial y Lealtad: Consulta de pedidos anteriores y acumulación automática de recompensas invisibles.

B. Módulo Operativo (Cajero)

Herramientas para la operación diaria en sucursal:

- ❖ Punto de Venta (POS): Interfaz para procesar ventas rápidas en mostrador ("Venta en Tienda"), generando pedidos con estado "Completado" inmediatamente.
- ❖ Gestión de Pedidos Web: Visualización y cobro de pedidos pendientes generados por clientes en línea.
- ❖ Consulta de Inventario: Acceso de solo lectura para verificar existencias de productos e ingredientes en tiempo real.
- ❖ Corte de Caja: Generación de reportes diarios de ingresos desglosados por método de pago (Efectivo, Tarjeta, Transferencia).

C. Módulo Administrativo (Gerencia)

- ❖ Control total del negocio y configuración del sistema:
- ❖ Gestión de Inventario Avanzada:
 - CRUD de Productos (con carga de imágenes) e Ingredientes.
 - Sistema de Recetas y Producción: Funcionalidad crítica que vincula ingredientes con productos. Permite registrar la producción ("Horneamos 5 pasteles") para descontar automáticamente la materia prima y aumentar el stock de venta.
- ❖ Gestión de Usuarios: Administración completa de cuentas de Cajeros y Clientes.
- ❖ Proveedores y Compras: Base de datos de proveedores para reabastecimiento.
- ❖ Reglas de Negocio: Configuración de recompensas y promociones automáticas.
- ❖ Dashboard de Inteligencia de Negocios: Visualización gráfica de KPIs como ventas mensuales, productos más vendidos y rendimiento por cajero.

Capacidades Centrales del Sistema

- ❖ **Seguridad de Roles:** El sistema usa ProtectedRoute en el frontend y middlewares (esAdmin, esPersonalAutorizado) en el backend para asegurar que ningún usuario pueda acceder a URLs o datos que no le corresponden.
- ❖ **Validación de Stock en Tiempo Real:** El sistema previene la sobreventa al verificar el stock en el *último segundo* (al momento de crear el pedido en el backend) dentro de una **transacción** de base de datos.
- ❖ **Auditoría:** Todas las operaciones críticas en la base de datos (crear, actualizar, eliminar) son registradas automáticamente en una tabla de **bitácora** mediante triggers, indicando qué usuario realizó el cambio.

ARQUITECTURA DEL SISTEMA.

1. Vista general.

Como fue mencionado anteriormente este proyecto está desarrollado en 2 partes, el frontend y el backend correspondientes a las carpetas con dicho nombre.

Este sería el árbol de directorios.

```
DulceSys/
|
|   └── frontend/          # Aplicación Cliente (React + Vite)
|       ├── node_modules/
|       ├── public/          # Assets estáticos públicos
|       └── src/
|           ├── api.js        # Configuración de Axios e Interceptores (Manejo de
|                           errores centralizado)
|           ├── main.jsx       # Punto de entrada y Configuración de RUTAS
|                           (React Router)
|           ├── index.css      # Estilos globales y configuración de Tailwind
|           |
|           |
|           └── components/    # Piezas de UI reutilizables
|               ├── Header.jsx  # Navegación principal (lógica de menú por
|                           rol)
|               ├── PaymentModal.jsx # Modal de cobro para Cajeros/Admin
|               ├── OrderDetailsModal.jsx # Visualizador de detalles de pedido
|               └── EditStatusModal.jsx # Modal exclusivo de Admin para corregir
|                           estados
|               |
|               |
|               └── context/      # Lógica de Estado Global
```

```
|   |   |   └── AuthContext.jsx      # Manejo de sesión, decodificación de JWT  
y roles  
|   |   |   └── CartContext.jsx    # Lógica del carrito persistente (conecta con  
API)  
|   |   |  
|   |   └── layouts/            # Envoltorios de seguridad y diseño  
|   |   |   └── MainLayout.jsx    # Estructura base (Header + Outlet)  
|   |   |   └── ProtectedRoute.jsx # Guardián de rutas (Verifica Login y  
Roles permitidos)  
|   |   |  
|   |   └── pages/             # Vistas principales (Pantallas)  
|   |       └── HomePage.jsx     # Catálogo público  
|   |       └── ProductDetailPage.jsx # Detalle de producto y personalización  
|   |       └── CartPage.jsx      # Vista del carrito y checkout inicial  
|   |           └── CheckoutPage.jsx    # Selección de método de pago y  
finalización  
|   |           └── ConfirmationPage.jsx # Voucher para clientes  
|   |           └── LoginPage.jsx      # Inicio de sesión  
|   |           └── RegisterPage.jsx    # Registro de clientes  
|   |           └── MisPedidosPage.jsx  # Historial del cliente  
|   |  
|   |           └── admin/ (Gestión Interna)  
|   |               └── InventoryAdminPage.jsx  # CRUD Productos/Ingredientes +  
Producción  
|   |           └── GestionPedidosPage.jsx  # Control de pedidos y flujo de caja  
|   |           └── GestionProveedoresPage.jsx  
|   |           └── GestionCajerosPage.jsx  
|   |           └── GestionClientesPage.jsx
```

```
|   |   └── GestionRecompensasPage.jsx
|   |   └── GestionDashboardPage.jsx # Estadísticas y gráficos
|   |   └── CorteCajaPage.jsx      # Reporte financiero diario
|
|   |
|   └── index.html
|   └── package.json
|   └── tailwind.config.js
|   └── vite.config.js
|
|   └── backend/          # Aplicación Servidor (Node.js + Express)
|       └── node_modules/
|       └── prisma/
|           └── schema.prisma # Definición de la Base de Datos y Modelos
|
|       └── public/
|           └── uploads/      # Almacenamiento físico de imágenes de
productos
|
|       └── index.js        # Servidor Principal: Rutas, Controladores y Lógica
de Negocio
|       └── validations.js # Esquemas de validación Zod (Reglas de negocio
estrictas)
|       └── .env            # Variables de entorno (DB URL, JWT Secret)
|       └── package.json
|
└── .gitignore
```

2. Frontend:

La mayor parte del programa de frontend vive en la carpeta “frontend”, específicamente en el src, es donde se encuentran todas las carpetas y archivos que dan vida al programa. Dentro de las carpetas del src podemos encontrar las siguientes carpetas:

- ❖ **Components.** En esta carpeta se encuentran componentes que son usados múltiples veces en el programa, como el **HeaderComponent** (El componente de la cabecera donde aparecen los menús de todos los roles). Como el modal de pago y de edición de estatus de pedidos. Esos componentes se encuentran ahí, como tal no son páginas, si no partes del programa que tienen funcionalidades.
- ❖ **Context.** En esta carpeta se establecen los contextos globales, como el **AuthContext** que sirve para crear un token válido que dé acceso a nuestra aplicación y valide los permisos de cada rol. Por otro lado también ahí se encuentra el archivo **CartContext**, que sirve para crear el carrito persistente, y cada que el un usuario entre a la app, el carrito que haya creado siga ahí.
- ❖ **Layout.** En esta carpeta se encuentran 2 archivos, el **MainLayout**, que básicamente une una página con el header, para que de esta forma el header siempre sea visible independientemente de en qué vista se encuentre el usuario. Y también está el **ProtectedRoute** que es usado para que los usuarios solo puedan visualizar y tener acceso a rutas permitidas por su rol. Es decir si algún cliente, conociera las rutas, y quisiera acceder a la ruta /admin/proveedores o algo así, este archivo impide el paso y lo regresa al inicio pues determina que el rol de clientes no tiene acceso para poder entrar a esa ruta.
- ❖ **Pages.** Aquí es donde vive el 90% del programa. Aquí básicamente están las siguientes vistas.

Para ubicar de mejor forma las páginas y qué roles tienen acceso a las rutas se seguirá la siguiente nomenclatura, [A] Administrador, [C] Cliente, [E] Cajero, [F] Todos los roles.

- **CartPage** (Vista del carrito) **[F]**.
- **CheckOutPage** (Pantalla que se ve cuando se paga) **[C]** **[E]**.

- **ConfirmationPage** (Pantalla que se ve cuando se confirma la forma del pago y el pedido) [C] [E].
- **CorteCajaPage** (Ventana del corte de caja) [A] [E]
- **GestionCajerosPage** (Vista con operaciones CRUD de cajeros) [A] [E]
- **GestionClientesPage** (Vista con operaciones CRUD de clientes) [A] [E]
- **GestionDashboardPage** (Ventana de estadísticas) [A]
- **GestionPedidosPage** (Ventana de pedidos) [A] [E]
- **GestionProveedoresPage** (Ventana de proveedores) [A]
- **GestionRecompensasPage** (Ventana de recompensas) [A]
- **HomePage** (Página de inicio) [F]
- **InventoryAdminPage** (Ventana con ambos inventarios con CRUD's) [A] [E]
- **LoginPage** (Ventana del login) [F]
- **MisPedidos** (Ventana de con historial de pedidos del cliente) [C]
- **ProductDetailPage** (Ventana que muestra un producto y sus características) [F]
- **RegisterPage** (Página de registro de usuarios) [F]

NOTA: Aunque Cajero y Administrador tengan acceso a rutas con operaciones CRUD's el único que puede ejecutarlas es el administrador, puesto que los botones, modales y funciones con las cuales se ejecutan esas operaciones solo son visibles y accesibles para el administrador.

Sin embargo el Frontend no se reduce solamente a las vistas, también podemos encontrar los siguientes directorios que forman parte del funcionamiento.

src/api.js (El Puente):

- ❖ Instancia centralizada de axios.
- ❖ **Interceptor de Request:** Inyecta automáticamente el Token JWT (Authorization: Bearer ...) en cada petición.
- ❖ **Interceptor de Response:** Centraliza el manejo de errores. Si el backend falla (400, 500), este archivo captura el error y muestra una notificación (toast)

automáticamente. También maneja la expiración de sesión (401) redirigiendo al login.

src/context/ (El Cerebro):

- ❖ **AuthContext:** Mantiene el estado del usuario (user: { id, rol }) accesible para toda la app. Decodifica el token JWT almacenado en localStorage.
- ❖ **CartContext:** No guarda el carrito en memoria local. Actúa como un **gestor de sincronización**. Cuando el usuario hace algo (agregar, borrar), este contexto llama a la API para guardar el cambio en la BD y luego recarga los datos para mantener la UI actualizada.

src/layouts/ProtectedRoute.jsx (El Guardián):

- ❖ Componente de alto orden que envuelve a las rutas privadas.
- ❖ Recibe un array allowedRoles. Si el rol del usuario actual no está en la lista, bloquea el acceso y redirige, protegiendo las vistas administrativas de accesos no autorizados.

3. Backend:

En la carpeta del backend hay considerablemente menos cosas. Pero no por ello menos importantes.

- ❖ **Carpeta Prisma.** Como fue usado Prisma, que es un Object Relational Mapper, en esa carpeta únicamente se encuentra el esquema generado por Prisma. Este como tal no se tiene que modificar manualmente, cualquier tipo de modificación que se pueda llegar a realizar en la Base de Datos, se puede actualizar en automático con el comando **npx prisma db pull**, y luego **npx prisma generate**.
- ❖ **Carpeta Public.** En la carpeta public, solo existe otra carpeta llamada upload la cual contiene las fotos que se usan en el sistema (Como fotos de los pasteles).
- ❖ **Archivo index.js.** En este archivo se encuentran todas las rutas del API, este archivo está comentado cuidadosamente y explica dónde se encuentra cada llamada, y cómo es que se hace, está escrito para interactuar con el ORM, por lo que en su mayoría no usa SQL.

index.js (El Controlador):

- ❖ Contiene la definición de todas las rutas (endpoints) de la API.
- ❖ Implementa los **Middlewares de Seguridad**:
 - autenticarUsuario: Valida el Token JWT.
 - esAdmin / esPersonalAutorizado: Valida que el usuario tenga el rol necesario para ejecutar la acción.
- ❖ Maneja las **Transacciones de Prisma** (prisma.\$transaction) para asegurar que operaciones complejas (como crear un pedido o registrar producción) sean atómicas (todo o nada).

validations.js (Las validaciones):

- ❖ Define esquemas estrictos usando la librería **Zod**.
- ❖ Valida cada dato que entra al servidor *antes* de tocar la base de datos o la lógica de negocio.
- ❖ **Reglas:** Define longitudes mínimas, formatos de correo, prohibición de números negativos, reglas de contraseñas seguras y formatos de RFC/Teléfono.

prisma/schema.prisma (El Modelo):

- ❖ Define la estructura de las tablas de PostgreSQL y sus relaciones.
- ❖ Es la fuente de verdad para generar el cliente de base de datos (@prisma/client).
- ❖ El uso de prisma previene inyección SQL.

4. Base de Datos (PostgreSQL)

La Base de Datos cuenta con 18 Tablas y fue desarrollada e implementada en PostgreSQL.

- Almacena información persistente del sistema.
- Cuenta con tablas normalizadas y relaciones controladas por Prisma.

- Permite auditorías mediante una tabla de bitácora.

5. Flujo normal de la información

Para entender cómo funciona el sistema, sigamos el viaje de una acción común: "**Crear un Pedido**".

1. **Inicio (Frontend):** El usuario hace clic en "Finalizar Pedido" en CheckoutPage.jsx.
2. **Validación Local:** El componente verifica datos básicos (método de pago seleccionado).
3. **Petición:** apiClient.post('/orders', payload) envía los datos al servidor.
4. **Intercepción:** api.js intercepta la salida, pega el Token del usuario en el encabezado y despacha el paquete.
5. **Recepción (Backend):** index.js recibe la petición.
6. **Autenticación:** El middleware autenticarUsuario verifica que el Token sea real y extrae el ID del usuario.
7. **Transacción (Lógica de Negocio):**
 - Se abre una transacción (prisma.\$transaction).
 - Se recalculan los precios reales desde la BD (seguridad).
 - Se verifica el stock disponible (restando el pivote de exhibición).
 - Si hay stock, se descuenta y se crea el pedido.
 - Si no hay stock, se lanza error y se revierte todo (Rollback).
8. **Respuesta:** El backend envía un 201 Created con el ID del pedido o un 400 Error.
9. **Feedback (Frontend):**
 - Si es éxito: CheckoutPage navega a la confirmación.
 - Si es error: El interceptor de api.js muestra un Toast rojo con la razón exacta del fallo.

LÓGICA DEL NEGOCIO CRÍTICA

Este capítulo documenta las reglas de negocio complejas implementadas en el Backend. Estas reglas garantizan la integridad de los datos, la seguridad financiera y la coherencia del inventario.

1. Estrategia de Inventario: El "Pivote de Exhibición"

El sistema implementa una regla de protección de stock para garantizar que la tienda física nunca se quede vacía debido a las ventas en línea.

- ❖ **Regla:** El stock disponible para venta online **NO** es el stock total físico.
- ❖ **Fórmula:** Stock Disponible Online = Stock Físico Total - Buffer de Exhibición (2 unidades).
- ❖ **Implementación:**
 - En el endpoint GET /products/:id, el backend realiza esta resta antes de enviar el dato al frontend.
 - Si hay 2 pasteles físicos, el sitio web mostrará "Agotado" (0 disponibles), reservándolos para venta en mostrador.
 - **Excepción:** El rol de **Cajero** y **Administrador** pueden ver el stock real en su módulo de inventario.

2. Ciclo de Producción (Recetas)

A diferencia de un comercio tradicional que solo "compra y vende", DulceSys gestiona la **transformación** de materia prima.

- ❖ **Entidad Relacional:** La tabla Product_Ingredients actúa como la "Receta", definiendo qué ingredientes y en qué cantidad (ej. 0.5kg Harina) se requieren para 1 unidad de producto final.
- ❖ **Proceso de Producción (POST /admin/inventory/produce):**
 - El sistema recibe la instrucción de "Producir X cantidad de Producto Y".
 - Consulta la receta.
 - Verifica si hay suficiente stock de **todos** los ingredientes requeridos.

➤ **Transacción Atómica:**

- Descuenta la cantidad proporcional de cada ingrediente en la tabla Ingredientes.
- Incrementa la cantidad producida en la tabla Productos.

➤ Si falta aunque sea un gramo de un ingrediente, la operación se rechaza completamente.

3. El "Checkout" Blindado (Transacción de Compra)

El proceso de compra (POST /api/orders) es el flujo más crítico del sistema. Se diseñó bajo el principio de "**Cero Confianza en el Cliente**" para evitar fraudes y errores de concurrencia.

Flujo de Validación y Ejecución:

Todo el proceso ocurre dentro de una prisma.\$transaction para garantizar atomicidad (todo ocurre o nada ocurre).

1. Recálculo de Precios (Seguridad Financiera):

- El backend **ignora** el total enviado por el frontend para efectos de cálculo interno.
- Recupera el precio base del producto desde la BD.
- Recupera el costo de cada personalización (ej. "Relleno Extra") desde la BD.
- Suma y calcula el Total Real en el servidor.

2. Validación de Stock (Anti-Race Condition):

- Justo antes de crear el pedido, el sistema vuelve a verificar el stock (aplicando la regla del Pivote).
- Esto previene que dos usuarios compren el último pastel al mismo tiempo. Si el stock ya no es suficiente en ese milisegundo, la transacción se aborta y se lanza un error.

3. Gestión de Recompensas:

- Si el cliente tenía una recompensa activa, se aplica el descuento al Total Real y se marca la recompensa como canjeada.

4. Asignación de Nueva Recompensa:

- *Después* de cerrar la venta exitosamente, el sistema evalúa el monto pagado contra las reglas de recompensa vigentes. Si califica, se le asigna una nueva recompensa para su **siguiente** compra.

4. Gestión Inteligente de Imágenes (Deduplicación)

Para optimizar el almacenamiento en el servidor y evitar archivos basura.

- ❖ **Hash SHA-256:** Al subir una imagen de producto, el sistema calcula su huella digital única (Hash).
- ❖ **Prevención de Duplicados:** Antes de guardar, verifica si ese Hash ya existe en la BD.
 - Si existe: Elimina el archivo recién subido y reutiliza la URL de la imagen existente.
 - Si no existe: Guarda el archivo y registra el nuevo Hash.
- ❖ **Limpieza (Garbage Collection):** Al actualizar la imagen de un producto, el sistema verifica si la imagen anterior está siendo usada por otro producto. Si nadie más la usa, la elimina físicamente del disco duro.

5. Carrito de Compras Persistente

El carrito no vive en la memoria del navegador (LocalStorage), sino en la base de datos (Carrito_Items).

- ❖ **Beneficio:** Permite la persistencia cruzada (el cliente empieza la compra en el móvil y termina en la PC) y permite al Cajero o Admin auditar carritos abandonados.
- ❖ **Sincronización:** El Frontend actúa como un espejo. Cada acción (agregar/borrar) dispara una petición a la API y luego recarga el estado completo desde la BD.

MANEJO DE ERRORES Y VALIDACIONES

El sistema DulceSys implementa una estrategia de validación de "Defensa en Profundidad". No confiamos únicamente en el frontend; el backend actúa como la autoridad final para garantizar la integridad de los datos. Además, el manejo de errores se ha centralizado para evitar la duplicación de código y garantizar una experiencia de usuario (UX) consistente.

1. Estrategia de Validación (Backend)

El backend utiliza la librería **Zod** para definir esquemas estrictos de datos. Ninguna petición POST o PUT llega a la base de datos sin pasar primero por un middleware de validación.

A. Definición de Esquemas (backend/validations.js)

Todas las reglas de negocio respecto al formato de los datos se definen aquí.

- ❖ **Tipado Fuerte:** Se asegura que los números sean números (usando `z.coerce.number()`) y los textos sean strings.
- ❖ **Reglas de Negocio:**
 - **RFC:** Validado con Regex para asegurar 13 caracteres alfanuméricos sin espacios.
 - **Contraseñas:** Deben tener min 8 caracteres, mayúscula, minúscula, número y especial.
 - **Teléfono:** Estrictamente 10 dígitos.
 - **Inventario:** Se prohíben números negativos en precios o stock.

B. Middleware validate

Se implementó un middleware personalizado que intercepta la request antes del controlador.

1. Ejecuta `schema.safeParse(req.body)`.

2. Si falla, detiene la ejecución inmediatamente.
3. Formatea los errores de Zod en un mensaje legible (ej. "*El RFC debe tener 13 caracteres*").
4. Responde con **Status 400 (Bad Request)**.

C. Manejo de Errores de Base de Datos (handlePrismaError)

Dado que algunas validaciones solo pueden ocurrir en la base de datos (como la unicidad de un correo o RFC), se creó una función auxiliar handlePrismaError en index.js.

- ❖ Captura errores nativos de Prisma (ej. código P2002 para Unique Constraint).
- ❖ Traduce el código técnico a un mensaje amigable (ej. "*El valor del campo 'email' ya existe*").
- ❖ Responde con **Status 409 (Conflict)**.

2. Estrategia de Manejo de Errores (Frontend)

En el frontend, se eliminó la necesidad de escribir bloques try/catch repetitivos en cada componente visual para mostrar alertas.

A. Interceptores de Axios (src/api.js)

Esta es la pieza central del manejo de errores en el cliente.

- ❖ **Funcionamiento:** Axios está configurado para interceptar **todas** las respuestas que regresan con un error (4xx o 5xx).
- ❖ **Notificaciones Automáticas:** Cuando se detecta un error, el interceptor extrae el mensaje enviado por el backend (error.response.data.error) y dispara automáticamente una notificación visual (Toast) usando la librería **react-hot-toast**.
- ❖ **Beneficio:** Los componentes (páginas) no necesitan código para mostrar errores. Si el backend rechaza una petición, el usuario ve la notificación roja automáticamente.

B. Validación Visual (UX)

Para mejorar la experiencia y reducir la carga al servidor, los formularios de React implementan validaciones nativas de HTML5 que coinciden con las reglas del backend:

- ❖ Atributos required, min, max, minLength, pattern.
- ❖ Inputs de tipo específico (type="email", type="number", type="tel").
- ❖ **Bloqueo de caracteres:** En campos numéricos críticos (como stock o precios), se bloquea la entrada de signos negativos (-) mediante eventos onKeyDown.

3. Flujo de Vida de un Error

Ejemplo: Un administrador intenta registrar un proveedor con un RFC duplicado.

1. **Frontend:** El administrador hace clic en "Guardar". El componente activa setLoading(true).
2. **Petición:** Se envían los datos al backend.
3. **Backend (Middleware Zod):** Los datos pasan la validación de formato (el RFC tiene 13 caracteres).
4. **Backend (Prisma):** Intenta insertar en la BD. PostgreSQL rechaza la operación por violación de restricción UNIQUE en la columna rfc.
5. **Backend (Handler):** handlePrismaError captura el código P2002, detecta que el campo fallido es rfc y devuelve una respuesta HTTP 409 con el JSON: { "error": "El valor del campo 'rfc' ya existe..." }.
6. **Frontend (Interceptor):** Axios recibe el 409. Lee el mensaje JSON e invoca toast.error("El valor del campo 'rfc' ya existe...").
7. **Frontend (Componente):** El await falla. El código salta al bloque finally, donde ejecuta setLoading(false) para desbloquear el botón.
8. **Usuario:** Ve una notificación roja flotante explicando el problema, sin que la página se recargue o se rompa.

DOCUMENTACIÓN DE LA API REST

El backend de DulceSys expone una API RESTful corriendo por defecto en <http://localhost:3000/api>.

1. Aspectos Generales

- ❖ **Formato de Datos:** Todas las peticiones y respuestas (excepto subida de archivos) utilizan formato JSON (Content-Type: application/json).
- ❖ **Autenticación:** Las rutas protegidas requieren un **JSON Web Token (JWT)**.
 - El token debe enviarse en el encabezado (Header) de la petición HTTP.
 - **Header:** Authorization
 - **Valor:** Bearer <tu_token_jwt>
- ❖ **Manejo de Errores:**
 - **400 Bad Request:** Datos de entrada inválidos (validación Zod fallida).
 - **401 Unauthorized:** No se envió token o el token expiró.
 - **403 Forbidden:** El usuario tiene sesión, pero no tiene el rol necesario (ej. Cliente intentando entrar a Admin).
 - **404 Not Found:** El recurso solicitado no existe.
 - **409 Conflict:** Violación de integridad en BD (ej. correo o RFC duplicado).
 - **500 Internal Server Error:** Error no controlado en el servidor.

| Método | Endpoint | Descripción | Body (JSON) | Respuesta Exitosa |
|--------|----------------|---------------------------------|--|---|
| POST | /auth/login | Iniciar sesión y obtener token. | { "correo": "...", "contrasena": "..." } | 200 OK { "token": "...", "usuario": { "id": 1, "rol": "Admin" } } |
| POST | /auth/register | Registrar un nuevo Cliente. | { "nombre": "...", "telefono": "...", "correo": "...", "contrasena": "..." } | 201 Created { "message": "Usuario registrado..." } |

2. Módulo de Tienda y Clientes (E-commerce)

Estas rutas son utilizadas por la interfaz pública y el perfil del cliente.

A. Catálogo

- **GET /products**
 - **Acceso:** Público.
 - **Descripción:** Obtiene la lista de todos los productos activos.
- **GET /products/:id**
 - **Acceso:** Público.
 - **Descripción:** Obtiene detalles de un producto, sus opciones de personalización y el stock disponible calculado (stock_real - 2).

B. Carrito de Compras (Persistente)

- **GET /cart**
 - **Acceso:** Cliente, Cajero, Admin.

- **Descripción:** Recupera los items guardados en la BD. Si es Cliente, calcula y aplica recompensas automáticamente.
- **POST /cart**
 - **Acceso:** Cliente, Cajero, Admin.
 - **Body:** { "id_producto": 1, "cantidad": 1, "personalizacion": { "Sabor": "Chocolate" } }
- **PUT /cart/:itemId**
 - **Acceso:** Cliente, Cajero, Admin.
 - **Descripción:** Actualiza la cantidad de un item. Valida stock antes de cambiar.
 - **Body:** { "cantidad": 5 }
- **DELETE /cart/:itemId**
 - **Acceso:** Cliente, Cajero, Admin.
 - **Descripción:** Elimina un item específico.
- **DELETE /cart**
 - **Acceso:** Cliente, Cajero, Admin.
 - **Descripción:** Vacía todo el carrito.

C. Pedidos y Checkout

- **POST /orders (CRÍTICO)**
 - **Acceso:** Cliente, Cajero, Admin.
 - **Descripción:** Procesa la compra. Realiza validación final de stock, descuenta inventario, consume recompensas, crea el pedido y vacía el carrito en una **transacción atómica**.

Body:

JSON

```
{
  "metodoPago": "Efectivo",
  "montoPagoCon": 500, // O null si es tarjeta
  "total": 450, // Referencial
  "descuentoAplicado": 50 // Opcional
```

- }
- - **GET /orders/my-history**
 - **Acceso:** Solo Cliente.
 - **Descripción:** Devuelve el historial de compras del usuario logueado.

3. Módulo de Gestión (Administrador y Cajero)

Rutas para la operación interna del negocio.

A. Gestión de Pedidos (Flujo de Salida)

- **GET /admin/orders?status=...**
 - **Acceso:** Admin, Cajero.
 - **Query Params:**
 - ?status=active: Devuelve pedidos "Pendiente" o "En preparación".
 - ?status=completed: Devuelve el historial ("Completado", "Cancelado").
- **PUT /admin/orders/:id/status**
 - **Acceso:** Admin, Cajero.
 - **Descripción:** Avanza el estado del pedido (ej. de "Pendiente" a "Completado" al cobrar).
 - **Body:** { "estado": "Completado" }

B. Reportes

- **GET /admin/corte-caja?fecha=YYYY-MM-DD**
 - **Acceso:** Admin, Cajero.
 - **Descripción:** Genera el reporte financiero del día (Ventas por método, productos vendidos, stock bajo).

5. Módulo de Administración Total (Solo Admin)

Rutas de alto nivel para la configuración del negocio. Todas requieren rol Administrador.

A. Inventario: Productos

Nota: Estas rutas usan multipart/form-data para subir imágenes.

- **POST /admin/products:** Crear producto. Valida duplicidad de imagen por Hash.
- **PUT /admin/products/:id:** Actualizar producto. Si cambia la imagen, borra la anterior del disco.
- **DELETE /admin/products/:id:** Borrado lógico (activo = false).

B. Inventario: Producción y Recetas

- **GET /admin/products/:id/recipe:** Obtiene los ingredientes necesarios para un producto.
- **POST /admin/products/:id/recipe:** Define/Actualiza la receta.
 - **Body:** { "ingredientes": [{ "id_ingrediente": 1, "cantidad": 0.5 }] }
- **POST /admin/inventory/produce:** Registrar Producción.
 - **Descripción:** Transforma ingredientes en productos terminados. Resta stock de insumos y suma stock de venta.
 - **Body:** { "id_producto": 1, "cantidad": 10 }

C. Inventario: Ingredientes

- **GET /admin/ingredients**
- **POST /admin/ingredients**
- **PUT /admin/ingredients/:id**
- **DELETE /admin/ingredients/:id**

D. Usuarios y Proveedores

El sistema cuenta con CRUDs estándar para:

- **Proveedores:** /admin/suppliers (Valida RFC único).
- **Cajeros:** /admin/cashiers (Crea usuario y empleado en transacción).
- **Clientes:** /admin/clients (Gestión de usuarios finales).

E. Recompensas y Estadísticas

- **GET /admin/rewards** y CRUD asociado: Configura las reglas de negocio para descuentos automáticos.
- **GET /admin/dashboard**: Endpoint pesado de agregación. Calcula KPIs, ventas semanales y top productos en una sola llamada.

GESTIÓN DE ROLES, PERMISOS Y SEGURIDAD

El sistema DulceSys implementa un modelo de control de acceso basado en roles (**RBAC** - Role-Based Access Control). La seguridad se aplica en dos capas: **Frontend** (para la experiencia de usuario) y **Backend** (para la integridad de los datos).

1. Definición de Roles

Existen tres roles jerárquicos definidos en la base de datos (Tabla Roles):

1. **Administrador (ID Rol: 1)**: Acceso total al sistema. Puede gestionar inventario, usuarios, configuración y ver reportes financieros.
2. **Cajero (ID Rol: 2)**: Acceso operativo limitado. Puede procesar ventas, ver pedidos pendientes y consultar inventario (solo lectura). No puede modificar el catálogo ni ver usuarios.
3. **Cliente (ID Rol: 3)**: Acceso exclusivo al e-commerce. Puede ver productos, gestionar su carrito y crear pedidos.

2. Capa de Seguridad Frontend (React)

En el cliente, la seguridad se enfoca en la navegación y la visibilidad de elementos de la interfaz (UI).

A. Protección de Rutas (`ProtectedRoute.jsx`)

Se utiliza un componente de orden superior (HOC) para envolver y proteger grupos de rutas.

- ❖ **Funcionamiento:** Recibe un prop `allowedRoles` (array). Compara el rol del usuario actual (almacenado en `AuthContext`) con la lista permitida.
- ❖ **Comportamiento:**
 - Si el usuario no está logueado → Redirige a `/login`.

- Si el usuario está logueado pero no tiene permiso → Redirige a / (Home).
- Si tiene permiso → Renderiza el componente hijo (<Outlet />).

Ejemplo de Implementación en main.jsx:

JavaScript

```
// Ruta accesible solo para Admins y Cajeros
{
  element: <ProtectedRoute allowedRoles={['Administrador', 'Cajero']} />,
  children: [
    { path: "admin/pedidos", element: <GestionPedidosPage /> }
  ]
}
```

B. Renderizado Condicional (UI)

Para páginas compartidas (como el inventario), se ocultan botones o secciones específicas según el rol.

- **Patrón:** Se evalúa la condición const isAdmin = user.rol === 'Administrador' al inicio del componente.
- **Uso:** {isAdmin && <Button>Eliminar</Button>}. Esto asegura que un Cajero pueda ver la tabla de productos, pero no vea los botones de acción.

3. Capa de Seguridad Backend (Node.js/Express)

Esta es la capa de seguridad real. Aunque un usuario logre "saltar" las protecciones del frontend, el backend rechazará cualquier petición no autorizada.

A. Autenticación (autenticarUsuario)

Este middleware se ejecuta primero en todas las rutas protegidas.

1. Intercepta la petición HTTP.
2. Busca el encabezado Authorization: Bearer <TOKEN>.
3. Verifica la firma del JWT usando jsonwebtoken y la JWT_SECRET.
4. Si es válido, decodifica el payload y lo inyecta en req.usuario ({ id, rol }).
5. Si es inválido o expiró, responde con **401 Unauthorized**.

B. Autorización por Roles (Middlewares)

Se implementaron dos middlewares para validar permisos específicos:

- **esAdmin:** Verifica estrictamente que req.usuario.rol === 'Administrador'.
Se usa en rutas críticas como DELETE /products o GET /dashboard.

esPersonalAutorizado(roles): Una función generadora que permite el acceso a múltiples roles. Se usa en rutas operativas como GET /orders (donde entran Cajeros y Admins).

JavaScript

```
// Ejemplo de uso en ruta
app.get('/api/admin/orders',
  autenticarUsuario,
  esPersonalAutorizado(['Administrador', 'Cajero']), // <-- Lista blanca
  controller
);
```

4. Lógica de Negocio Diferenciada

El sistema cambia su comportamiento interno dependiendo de quién ejecuta la acción, especialmente en el proceso de **Checkout** (POST /api/orders).

Caso: Venta por Cajero/Admin (Punto de Venta)

Cuando un rol administrativo ejecuta una venta:

1. El sistema ignora la identidad personal del usuario logueado (el cajero).
2. Asigna el pedido a un "**Cliente Genérico de Mostrador**" (configurado con el correo).
3. Establece el estado del pedido inmediatamente como "**Completado**".
4. No genera voucher de pago diferido.

Caso: Compra por Cliente (E-commerce)

Cuando un rol Cliente ejecuta una compra:

1. El sistema vincula el pedido estrictamente al ID del usuario logueado.
2. Establece el estado del pedido como "**Pendiente**".
3. Calcula y aplica reglas de lealtad (Recompensas).

Base de Datos y Sistema de Auditoría

El sistema utiliza **PostgreSQL** como motor de base de datos relacional. El diseño sigue un esquema normalizado (3NF) para garantizar la integridad de los datos, complementado con campos JSONB para flexibilidad en la personalización de productos.

1. Diagrama y Estructura (Schema)

El esquema se divide en módulos lógicos:

A. Módulo de Usuarios (Identidad)

El sistema utiliza un patrón de "Herencia de Tabla" para manejar los tipos de usuarios.

- ❖ **Usuarios:** Tabla maestra. Contiene las credenciales de acceso (correoelectrónico, contraseña) y el rol (id_rol).
- ❖ **Nota sobre Implementación Inicial:** Para la fase de prueba/desarrollo, la contraseña inicial establecida fue Password=1. Se requiere actualizar esta contraseña inmediatamente en producción.
- ❖ **Roles:** Catálogo de permisos (1: Administrador, 2: Cajero, 3: Cliente).
- ❖ **Clientes:** Extensión de usuario. Contiene datos específicos de negocio (nombre, teléfono). Se vincula 1:1 con Usuarios.
- ❖ **Empleados:** Extensión de usuario. Contiene datos laborales (nombrecompleto, turno). Se vincula 1:1 con Usuarios.

B. Módulo de Inventory y Producción

Diseñado para soportar la transformación de materia prima.

- ❖ **Productos:** El catálogo de venta.
 - stockproductosterminalados: La cantidad física lista para vender.
 - imagenurl y imagehash: Gestión de archivos multimedia.
- ❖ **Ingredientes:** Materia prima (Harina, Huevo, etc.).
 - stockactual vs stockminimo: Para alertas de reabastecimiento.
- ❖ **Product_Ingredients (La Receta):** Tabla pivote (Muchos a Muchos) que define qué ingredientes se consumen al producir un producto.

C. Módulo de Ventas y Transacciones

- ❖ **Pedidos:** Cabecera de la venta.
 - estado: Controla el flujo ('Pendiente', 'Completado', 'Cancelado').
 - descuento: Almacena el monto descontado por recompensas.
- ❖ **Detalle_Pedido:** Renglones de la venta.

- personalizacion (JSONB): Almacena la configuración exacta del producto en el momento de la compra (ej. {"Sabor": "Vainilla", "Tamaño": "Grande"}).
- ❖ **Carrito_Items:** Carrito persistente. Funciona igual que el detalle de pedido pero es temporal por usuario.

2. Diccionario de Datos (Tablas Críticas)

Tabla: Usuarios

| Columna | Tipo | Descripción |
|-------------------|--------------|---|
| id_usuario | SERIAL (PK) | Identificador único del sistema. |
| correoelectrónico | VARCHAR(100) | Unique. Usado para el Login. |
| contraseña | VARCHAR(255) | Hash Bcrypt. Nunca texto plano. |
| id_rol | INT (FK) | Referencia a la tabla Roles. |
| activo | BOOLEAN | true por defecto. Se usa para borrado lógico. |

Tabla: Productos

| Columna | Tipo | Descripción |
|-------------|-------------|--|
| id_producto | SERIAL (PK) | Identificador del producto. |
| sku | VARCHAR(50) | Código único de inventario (ej. PAS-CHO-01). |

| | | |
|--------------------------|---------------|---|
| preciobase | DECIMAL(10,2) | Precio antes de personalizaciones. |
| stockproductosterminados | INT | Cantidad real en existencia. |
| Imagenurl | VARCHAR(255) | Ruta relativa (ej. /uploads/foto.jpg). |
| imagehash | VARCHAR(64) | Hash SHA-256 para evitar archivos duplicados. |

Tabla: BitacoraCambios (Auditoría)

| Columna | Tipo | Descripción |
|-----------------|----------------|--|
| id_bitacora | BIGSERIAL (PK) | ID autoincremental masivo. |
| tablaafectada | VARCHAR(100) | Nombre de la tabla modificada (ej. 'productos'). |
| accion | VARCHAR(10) | 'INSERT', 'UPDATE', o 'DELETE'. |
| datosanteriores | JSONB | Copia de la fila <i>antes</i> del cambio (para UPDATE/DELETE). |
| datosnuevos | JSONB | Copia de la fila <i>después</i> del cambio (para INSERT/UPDATE). |
| id_usuario | INT | ID del usuario que realizó la acción. |

3. Sistema de Auditoría Automática (Triggers)

El sistema cuenta con un mecanismo de "Caja Negra" que registra **cualquier** movimiento en la base de datos, sin importar si se hizo desde el Frontend, el Backend o manualmente por SQL.

A. El Usuario "Sistema" (ID 0)

Para garantizar la integridad referencial, existe un usuario especial con id_usuario = 0.

- **Propósito:** Si una operación se realiza automáticamente por el servidor (cron job) o manualmente por un DBA sin sesión, el sistema atribuye el cambio a este usuario en lugar de fallar.

B. Función Maestra: fn_registrar_bitacora()

Es una función PL/pgSQL genérica que se ejecuta cada vez que ocurre un evento.

1. **Detección de Usuario:** Intenta leer la variable de sesión myapp.user_id (inyectada por el Backend en cada petición). Si no existe, usa el ID 0.
2. **Captura de Datos:** Convierte las filas OLD y NEW a formato JSON.
3. **Registro:** Inserta una nueva fila en BitacoraCambios con la marca de tiempo exacta.

C. Implementación de Triggers

Cada tabla crítica tiene un disparador asociado que invoca la función maestra.

Ejemplo de Definición (SQL):

```
CREATE TRIGGER trg_audit_productos
AFTER INSERT OR UPDATE OR DELETE ON Productos
FOR EACH ROW EXECUTE FUNCTION
fn_registrar_bitacora('id_producto');
```

4. Scripts de Mantenimiento

Para desarrolladores que necesiten modificar la base de datos, existen flujos obligatorios.

Sincronización con Prisma

El ORM no detecta cambios en la BD automáticamente. Si modificas una tabla (ej. ALTER TABLE), debes ejecutar:

1. npx prisma db pull (Actualiza el archivo schema.prisma local).
2. npx prisma generate (Regenera las librerías de JavaScript para que el autocomplete funcione).

Reinicio de Secuencias

Si al insertar datos manualmente (seeds) obtienes errores de "Unique constraint failed on id...", debes resincronizar los contadores:

```
SELECT setval(pg_get_serial_sequence('nombre_tabla',
'id_columna'), max(id_columna)) FROM nombr
```

Guía de Solución de Problemas

(Troubleshooting)

Esta sección documenta los errores más comunes encontrados durante el desarrollo y despliegue de DulceSys, junto con sus soluciones probadas.

1. Errores de Base de Datos y Prisma

A. Error: Unique constraint failed on the fields: (id_...)

Síntoma: Al intentar crear un nuevo registro (Pedido, Producto, etc.), el sistema lanza un error 500 indicando que el ID ya existe. **Causa:** La secuencia interna de PostgreSQL (SERIAL) está desincronizada con los datos reales, generalmente por haber insertado datos manualmente con IDs específicos. **Solución:** Ejecutar el script de resincronización en la BD:

SQL

```
-- Ejemplo para la tabla Pedidos
SELECT      setval(pg_get_serial_sequence('pedidos',      'id_pedido'),
max(id_pedido)) FROM pedidos;
```

B. Error: TypeError: Cannot read properties of undefined (reading 'create')

Síntoma: El backend falla al iniciar una consulta de Prisma. **Causa:** El cliente de Prisma (@prisma/client) está desactualizado y no reconoce una tabla nueva o una columna recién agregada. **Solución:** Detener el servidor backend y regenerar el cliente:

Consola

```
cd backend
npx prisma db pull      # 1. Leer la BD actualizada
npx prisma generate     # 2. Actualizar los archivos JS
npm run dev              # 3. Reiniciar
```

C. La Bitácora (Audit Log) no registra nada

Síntoma: Se hacen cambios en la BD pero la tabla BitacoraCambios sigue vacía.

Causas Probables y Soluciones:

1. **Triggers Deshabilitados:** Si se restauró un backup, los triggers pueden estar en DISABLE.
 - **Solución:** Ejecutar `ALTER TABLE nombre_tabla ENABLE TRIGGER trg_audit_...`;
2. **Falta Usuario Sistema:** El trigger intenta asignar `ID_Usuario = 0` pero ese usuario no existe.
 - **Solución:** Insertar el usuario 0 en la tabla Usuarios (ver Sección 4.3).

2. Errores de Backend (API)

A. Error: Invalid input: expected number, received NaN

Síntoma: Al enviar un formulario (ej. Crear Producto), el backend devuelve este error de Zod. **Causa:** Discrepancia de nombres de variables. El frontend envía camelCase (ej. `precioBase`) y el backend intenta leer lowercase (ej. `preciobase`) o viceversa, resultando en `undefined`. **Solución:** Verificar que el destructuring en el controlador (`const { ... } = req.body`) coincida exactamente con lo que envía el frontend.

B. Las imágenes no cargan (404 Not Found)

Síntoma: La URL de la imagen en la BD es correcta (`/uploads/foto.jpg`) pero el navegador muestra imagen rota. **Causa:** Express no está configurado para servir archivos estáticos desde esa carpeta. **Solución:** Verificar en `index.js`:

JavaScript

```
import path from 'path';
import { fileURLToPath } from 'url';
const __dirname = path.dirname(fileURLToPath(import.meta.url));
// Esta línea es obligatoria:
app.use('/uploads', express.static(path.join(__dirname,
'public/uploads')));
```

C. Error: ReferenceError: fs is not defined

Síntoma: Ocurre al intentar actualizar o borrar una imagen. **Causa:** Falta importar la librería de sistema de archivos. **Solución:** Agregar `import fs from 'fs'`; al inicio de `index.js`.

3. Errores de Frontend (React)

A. Pantalla en Blanco (White Screen of Death)

Síntoma: La aplicación no carga nada. **Causa Común:** Error de sintaxis crítico antes del renderizado.

1. **Componentes en minúscula:** Usar `<loginPage />` en lugar de `<LoginPage />`. React exige PascalCase.
2. **Contexto indefinido:** Intentar usar `useAuth()` o `useCart()` en un componente que no está envuelto por `<AuthProvider>` o `<CartProvider>` en `main.jsx`.

B. Botón "No hace nada" (Sin error en consola)

Síntoma: Se hace clic en un botón (ej. "Ir a Pagar") y no hay reacción. **Causa:** Falta inicializar el hook de navegación. **Solución:**

JavaScript

```
import { useNavigate } from 'react-router-dom';
// Dentro del componente:
const navigate = useNavigate();
// En el onClick:
onClick={() => navigate('/ruta')}
```

C. Error: opt.ajusteprecio.toFixed is not a function

Síntoma: Falla al renderizar precios. **Causa:** Los datos vienen de la BD (Prisma/PostgreSQL) como String o Decimal, no como Number nativo de JS. **Solución:** Siempre envolver los valores monetarios antes de formatear:

JavaScript

```
parseFloat(producto.preciobase).toFixed(2)
```

4. Mantenimiento de Rutina

Para mantener la salud del sistema DulceSys, se recomienda:

1. **Limpieza de Imágenes Huérfanas:** Aunque el sistema intenta borrar imágenes al actualizar, ocasionalmente pueden quedar archivos basura en backend/public/uploads. Se recomienda una revisión manual mensual.
2. **Revisión de Bitácora:** La tabla BitacoraCambios crece indefinidamente. Se sugiere respaldar y depurar registros antiguos (más de 1 año) periódicamente para no afectar el rendimiento.
3. **Actualización de Dependencias:** Ejecutar npm audit regularmente en ambas carpetas (frontend y backend) para detectar y corregir vulnerabilidades de seguridad en las librerías.

INSTALACION Y CONFIGURACION DEL SISTEMA

En esta sección se describen los requisitos y pasos necesarios para instalar y ejecutar el sistema en otra computadora. El proyecto está dividido en dos partes principales:
Frontend: Aplicación web en React + Vite + Tailwind CSS.

Backend: API en Node.js con Express y Prisma, conectada a una base de datos **PostgreSQL**.

8.1 Requisitos previos

Antes de instalar el sistema, la computadora debe contar con lo siguiente:

- **Sistema operativo**

Windows 10 o superior.

Software necesario

- **Node.js 22.20 LTS** (incluye npm).
- **Visual Studio Code** (u otro editor de código).
- **PostgreSQL** (servidor de base de datos).
- **pgAdmin** (cliente gráfico para administrar PostgreSQL).

Puertos utilizados

- Frontend: `http://localhost:5173`
- Backend/API: `http://localhost:3000`
- PostgreSQL: puerto por defecto 5432.
- **Estructura del proyecto**

Carpeta raíz del proyecto con:

- `/frontend` → aplicación React.
- `/backend` → API, Prisma y conexión a BD.

8.2 Instalación de Node.js y verificación de npm

1. Descargar **Node.js 22.20 LTS** desde la página oficial.
2. Ejecutar el instalador:

- Dejar marcada la opción “**Add to PATH**”.
 - Aceptar las opciones por defecto.
3. Al finalizar, abrir la terminal (CMD) y verificar:

```
node -v  
npm -v
```

8.3 Instalación y configuración de PostgreSQL

Descargar e instalar **PostgreSQL** (incluye pgAdmin).

Durante la instalación:

- Definir una contraseña para el usuario **postgres** (debe recordarse, se usará en la cadena de conexión).
- Mantener el puerto por defecto 5432.

Abrir **pgAdmin**, conectarse con el usuario postgres y la contraseña definida.

Crear una nueva base de datos:

- Botón derecho sobre **Databases** → **Create** → **Database...**
- Nombre sugerido: **dulcesys** (o el nombre que use el proyecto).
- Guardar.

8.4 Configuración del Backend

La parte de backend se encuentra en la carpeta /backend. Aquí se configura la conexión a PostgreSQL, las migraciones de Prisma y la API.

8.4.1 Instalación de dependencias del backend

1. Abrir **Visual Studio Code**.
2. Abrir la carpeta del proyecto.
3. Abrir una terminal en VS Code y ubicarse en la carpeta backend: cd backend
4. Instalar las dependencias:npm install

8.4.2 Configuración del archivo .env (cadena de conexión)

Dentro de la carpeta backend, se debe crear un archivo llamado .env

Con el siguiente contenido (ajustando contraseña y nombre de base de datos):
DATABASE_URL="postgresql://postgres:TU CONTRASEÑA@localhost:5432/du
lcesys?schema=public"

- **postgres** → usuario por defecto de PostgreSQL.

- TU CONTRASEÑA → contraseña definida durante la instalación de PostgreSQL.
- dulcesys → nombre de la base de datos creada en pgAdmin.

Este archivo **no se debe subir a repositorios públicos**, ya que contiene credenciales de conexión.

8.4.3 Aplicar migraciones de Prisma (creación de tablas)

El archivo prisma/schema.prisma ya contiene la definición de las tablas del sistema (usuarios, roles, productos, pedidos, etc.).

Para crear las tablas en PostgreSQL:

En la terminal (dentro de backend), ejecutar: npx prisma migrate dev --name init

8.4.4 Ejecución del script de inicialización (usuario administrador)

El proyecto cuenta con un script para crear automáticamente un usuario administrador en la base de datos.

Un ejemplo de archivo puede ser seed-admin.js, ubicado en la carpeta backend, con la siguiente lógica:

- Crear (o asegurar) el rol **Administrador**.
- Crear un usuario con correo admin@dulcesys.com y contraseña Admin123* (almacenada con hash).
- Crear el registro correspondiente en la tabla de empleados.

Para ejecutarlo:

1. Asegurarse de estar en la carpeta backend.

2. Ejecutar:

node seed-admin.js

o, si está configurado en package.json como script:

npm run seed:admin

En la terminal debería mostrarse un mensaje similar a:

Admin creado: admin@dulcesys.com

Nota: Por motivos de seguridad, se recomienda cambiar esta contraseña por una propia al usar el sistema en producción.

8.4.5 Puesta en marcha del servidor backend

Finalmente, para iniciar la API:

1. Verificar en backend/package.json el script de arranque (por ejemplo):

```
"scripts": {
  "dev": "node index.js"
```

}

2. En la terminal (dentro de backend):
npm run dev

Si el servidor arranca correctamente, se mostrará un mensaje indicando el puerto, por ejemplo:

Servidor escuchando en http://localhost:3000

8.5 Configuración del Frontend

La parte de frontend se encuentra en la carpeta `/frontend`, desarrollada con React, Vite y Tailwind CSS.

8.5.1 Instalación de dependencias del frontend

1. Abrir una nueva terminal en VS Code.

2. Ubicarse en la carpeta frontend:

cd frontend

3. Instalar las dependencias:

npm install

8.5.2 Arranque del servidor de desarrollo (frontend)

Para iniciar la aplicación web:

npm run dev

La terminal mostrará una dirección similar a:

Local: http://localhost:5173/

Abrir esa URL en el navegador para visualizar la interfaz del sistema.

8.6 Orden recomendado de arranque

Para que el sistema funcione correctamente en otra computadora, se recomienda seguir este orden:

1. Iniciar PostgreSQL

- Asegurarse de que el servicio de PostgreSQL esté corriendo.

2. Aplicar migraciones (solo la primera vez)

En /backend:

npx prisma migrate dev --name init

○

3. Ejecutar el script de creación del administrador (solo la primera vez)

En /backend:

node seed-admin.js

○

4. Levantar backend

En /backend:

npm run dev

○

5. Levantar frontend

En /frontend:

npm run dev

○

6. Acceder al sistema

- Navegador → http://localhost:5173

EJECUCIÓN DEL SISTEMA

Backend:

npm start

Frontend:

npm run dev

PRUEBAS TÉCNICAS

Las pruebas incluyeron:

- Validación de inicio de sesión y registro.
- Flujo completo de compra y confirmación.
- Cambios de estado en pedidos.
- Gestión de cajeros.
- Corte de caja y registro de movimientos.
- Inventario para empleados y administradores.

- Interacción correcta entre frontend y backend.

MANEJO DE ERRORES Y BITÁCORA DE CAMBIOS

La tabla **bitacoracambios** permite rastrear:

- Datos previos.
- Datos nuevos.

- Acción realizada.
- Fecha y tabla afectada.

Esto asegura trazabilidad y auditoría en todas las operaciones sensibles.

FUNCIONALIDADES

PRUEBAS DE FUNCIONALIDAD

1. Pruebas de autenticación (Login)

Ingresar credenciales correctas

A continuación podemos ver que al ingresar con cada tipo de usuario, nos abre el menú correcto, es decir si validan credenciales, y muestra los paneles indicados para cada uno.

Pantalla iniciar sesión “Cliente”

Iniciar Sesión

Correo Electrónico:

santiagovtania@gmail.com

Contraseña:

.....

Ingresar

¿No tienes una cuenta? [Regístrate aquí](#)

Pantalla de “inicio cliente”

localhost:5173

DulceSys

Inicio Carrito Mis Pedidos Hola, Cliente Cerrar Sesión

Buscar productos...

Nuestros Productos

| Producto | Descripción | Precio |
|-------------------------|---------------------------------------|--------|
| Pastel de Chocolate | Clásico pastel de chocolate | \$350 |
| Pastel de oreo | Pastel de 3 leches, hecho con gall... | \$300 |
| Pastell de vainilla | Pastel de Vainilla, clásico... | \$350 |
| Pastel de Fresa y Crema | Pastel de Fresas con crema, 3 lech... | \$380 |

Inicio de sesión del administrador

- Mismo formulario de autenticación con campos estándar
- El sistema diferencia el rol después de validar

Iniciar Sesión

Correo Electrónico:

Contraseña:

Ingresar

[¿No tienes una cuenta? Regístrate aquí](#)

Pantalla de “inicio administrador”

Múltiples secciones visibles: Productos, Inventario, Pedidos, Usuarios, Ingredientes, etc.

localhost:5173

DulceSys Inicio Cajeros Proveedores Inventario Recompensas Estadísticas Gestión de Pedidos Cierre de Caja Hola, Administrador Cerrar Sesión

Buscar productos...

Nuestros Productos

| | | | |
|---|--|--|---|
| Pastel de Chocolate Clásico pastel de chocolate \$350 | Pastel de oreo Pastel de 3 leches, hecho con gall... \$300 | Pastell de vainilla Pastel de Vainilla, clásico... \$350 | Pastel de Fresa y Crema Pastel de Fresas con crema, 3 lech... \$380 |
|---|--|--|---|

Inicio de sesión del cajero

Iniciar Sesión

Correo Electrónico:

Contraseña:

[Ingresar](#)

[¿No tienes una cuenta? Regístrate aquí](#)

Pantalla de “inicio cajero”

- Vista enfocada en gestión de pedidos
- Tabla con listado de órdenes mostrando: número, cliente, fecha, total y estado
- Menú simplificado comparado con el administrador

The screenshot shows a web browser window with the URL "localhost:5173". The page title is "DulceSys". The navigation menu includes "Inicio", "Carrito", "Inventario", "Gestión de Pedidos", "Cierre de Caja", "Hola, Cajero", and a red "Cerrar Sesión" button. A search bar at the top right contains the placeholder "Buscar productos...". Below the search bar, the heading "Nuestros Productos" is displayed. Four cake products are listed in a grid:

- Pastel de Chocolate**: Clásico pastel de chocolate. Price: \$350.
- Pastel de oreo**: Pastel de 3 leches, hecho con gall... Price: \$300.
- Pastell de vainilla**: Pastel de Vainilla, clásico... Price: \$350.
- Pastel de Fresa y Crema**: Pastel de Fresas con crema, 3 lech... Price: \$380.

Contraseña invalida

Si se ingresa una credencial mal la contraseña nos envía el mensaje de credenciales inválidas y no podemos acceder, aquí se podría implementar la recuperación de contraseñas, ya que no tenemos esta opción.

Iniciar Sesión

Credenciales inválidas

Correo Electrónico:
santiagovtania@gmail.com

Contraseña:
.....

Ingresar

¿No tienes una cuenta? [Regístrate aquí](#)

Ingresar usuario inexistente

Si se ingresa un usuario incorrecto, no permite ingresar al sistema
Mismo comportamiento que contraseña inválida. Mensaje de error
sin especificar si es usuario o contraseña (buena práctica de
seguridad)

Iniciar Sesión

Credenciales inválidas

Correo Electrónico:

Contraseña:

Ingresar

¿No tienes una cuenta? [Regístrate aquí](#)

Intentar ingresar con campos vacíos

Campos de entrada con bordes rojos indicando error. Mensaje: "Este campo es obligatorio" debajo de cada campo vacío

Iniciar Sesión

Correo y contraseña son requeridos

Correo Electrónico:

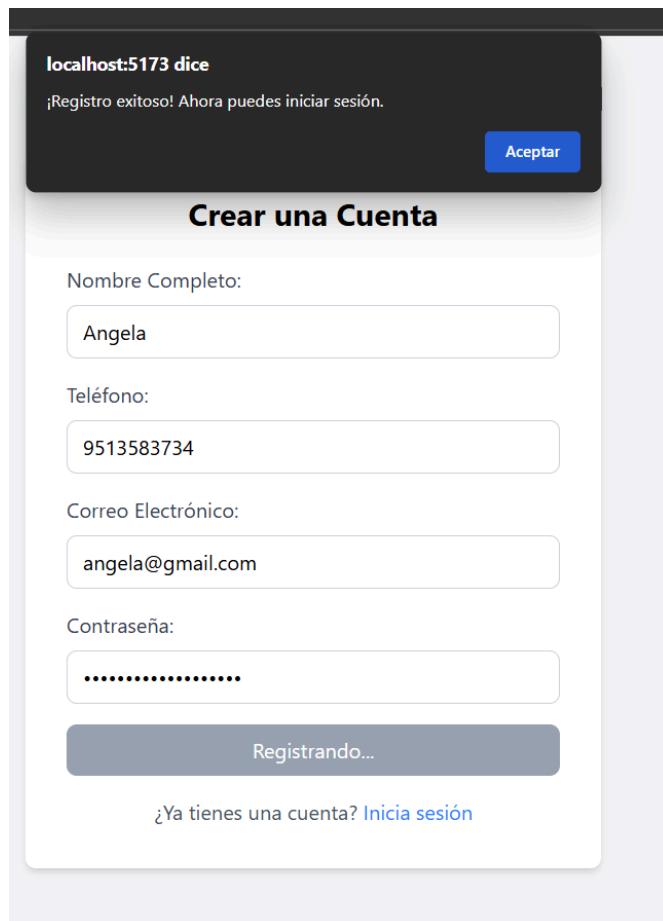
Contraseña:

Ingresar

¿No tienes una cuenta? [Regístrate aquí](#)

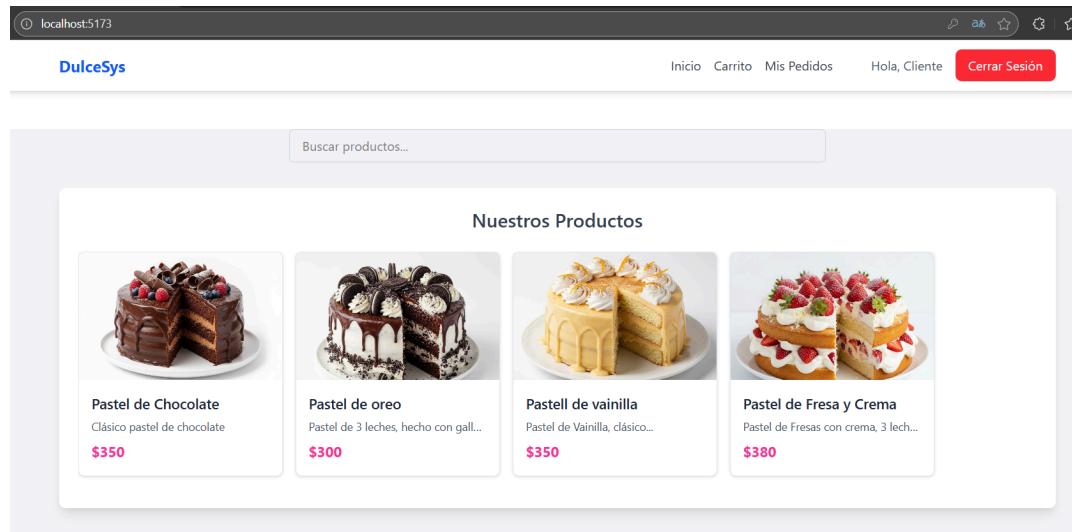
Probar límites de caracteres en usuario y contraseña

- No tiene límites de carácteres.
- Campo de contraseña con texto excesivamente largo (50+ caracteres)
- El sistema acepta la entrada sin restricciones
- Potencial vulnerabilidad de seguridad



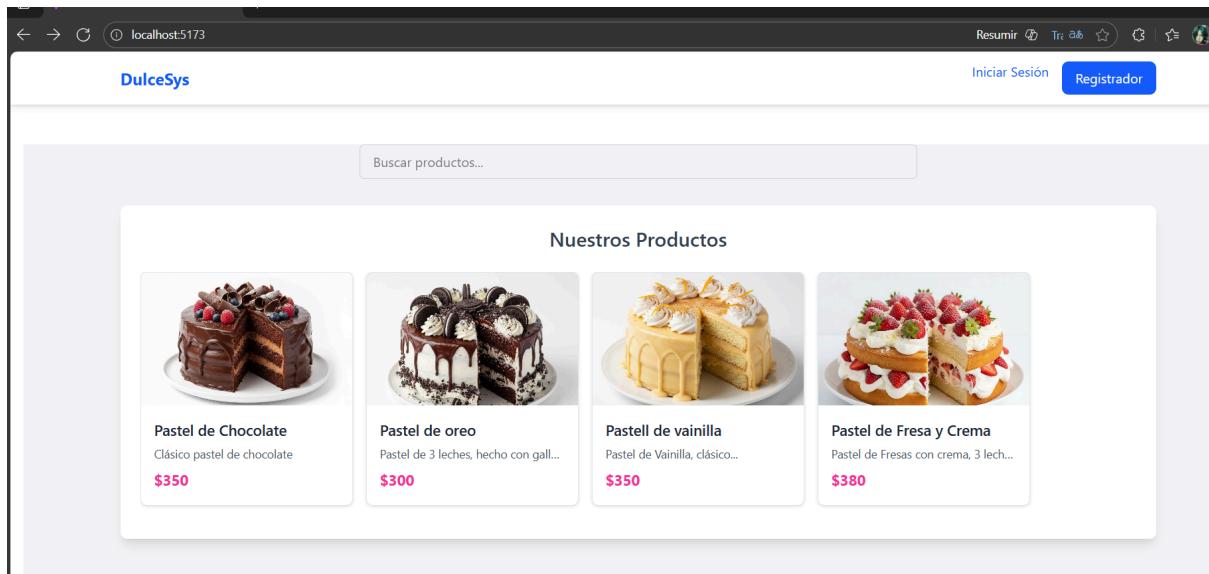
Verificar persistencia de sesión (si se mantiene al navegar)

- Usuario navegando entre diferentes secciones sin perder sesión
- Indicador de sesión activa visible en el header
- No requiere reautenticación al cambiar de página



Cerrar sesión y tratar de volver a entrar con el botón atrás

- Se cierra la sesión automáticamente, y en la esquina superior nos dice si se desea iniciar sesión o registrarse.



Restricciones rotas por URL

Si se accede desde el link del administrador en otra pestaña, nos pide ingresar las credenciales nuevamente por seguridad.

- Intento de acceder a URL protegida directamente
- Sistema intercepta y redirige a página de login
- Mensaje: "Debe iniciar sesión para acceder a esta página"

The image consists of two screenshots of a web browser. The top screenshot shows a browser window with two tabs open. The active tab is titled 'frente' and shows the URL 'localhost:5173/admin/inventario'. This tab displays a list of inventory items under the heading 'Gestión de Inventario'. The items are:

| SKU | IMAGEN | NOMBRE | PRECIO | ACCIÓN | ACCIONES |
|------|--------|-------------------------|----------|--------|---|
| PASC | | Pastel de Chocolate | \$350.00 | 4 | Editar Eliminar |
| PASO | | Pastel de oreo | \$300.00 | 19 | Editar Eliminar |
| PASV | | Pastell de vainilla | \$350.00 | 13 | Editar Eliminar |
| PASF | | Pastel de Fresa v Crema | \$380.00 | 8 | Editar Eliminar |

The bottom screenshot shows a separate browser window with the title 'localhost:5173/login'. It displays a login form with the title 'Iniciar Sesión' (Login). The form fields are:

- Correo Electrónico: admin@duls.com
- Contraseña: (redacted)
- Ingresar (Login button)

Below the login button, there is a link: '¿No tienes una cuenta? [Regístrate aquí](#)' (Don't you have an account? [Register here](#)).

2. Pruebas del CRUD de productos

Durante las pruebas del módulo de inventario y carrito de compras, se detectó que ciertas operaciones no funcionaban correctamente, específicamente al intentar actualizar el stock. El problema se debía a una discrepancia en los nombres de los campos entre el backend y el frontend.

Agregar Producto

SKU
PASZ

Nombre
Pastel de zanahoria

Descripción
Relleno de zanahoria rayada

Categoría
Pasteles

Precio
350.00

Acción
1

Imagen
Elegir archivo PastelZanahoria.png

Cancelar **Guardar**

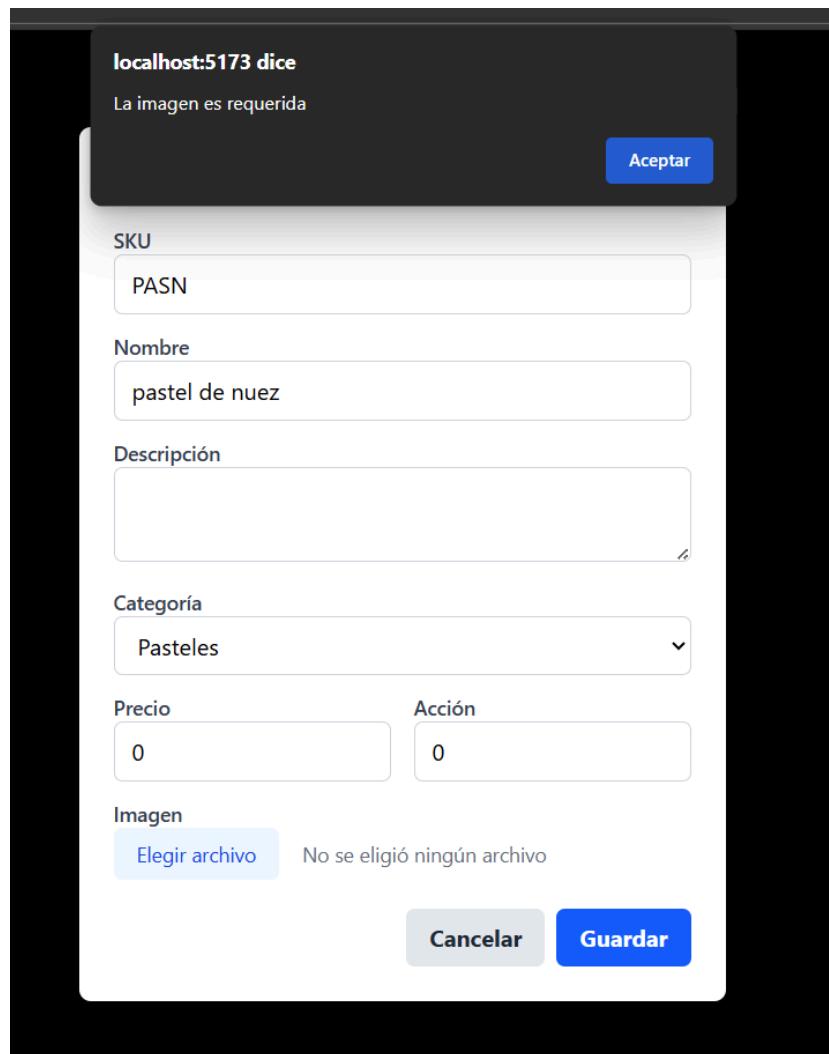
- Primera imagen: Código del backend mostrando la variable `stockProductosTerminados` en formato camelCase
- Segunda imagen: Consola del navegador mostrando error, el frontend busca `stockproductostermindos` en minúsculas
- Resultado: el valor aparece como `undefined` causando fallas en la actualización del inventario

| Productos | | Ingredientes | | | |
|-----------|---|--------------------------------------|----------|--------|---|
| | | Buscar productos por nombre o SKU... | | | |
| SKU | IMAGEN | NOMBRE | PRECIO | ACCIÓN | ACCIONES |
| PASO |  | Pastel de oreo | \$300.00 | 18 | Editar Eliminar |
| PASV |  | Pastell de vainilla | \$350.00 | 12 | Editar Eliminar |
| PASF |  | Pastel de Fresa y Crema | \$380.00 | 7 | Editar Eliminar |
| PASC |  | Pastel de Chocolate | \$350.00 | 3 | Editar Eliminar |
| PASZ |  | Pastel de zanahoria | \$350.00 | | Editar Eliminar |

Crear producto con campos vacíos o inválidos

Si no subimos la imagen del producto no nos permite guardarla.

- Formulario de creación de producto con todos los campos completados excepto la imagen
- Mensaje de error destacado: "La imagen del producto es obligatoria"
- El botón "Guardar" no procesa la acción



Si dejamos campos vacíos guarda el nuevo producto, lo cual no debería pasar

Editar Producto

SKU
PASN

Nombre
pastel de nuez

Descripción

Categoría
Pasteles

Precio
0

Acción
0

Imagen
[Elegir archivo](#) No se eligió ningún archivo
Dejar en blanco para conservar la imagen actual.

[Cancelar](#) [Guardar](#)

| | | | | | |
|------|---|---------------------|----------|---|---|
| PASC |  | Pastel de Chocolate | \$350.00 | 3 | Editar Eliminar |
| PASZ |  | Pastel de zanahoria | \$350.00 | | Editar Eliminar |
| PASN |  | pastel de nuez | \$0.00 | | Editar Eliminar |

Crea producto con precio negativo (debería bloquearse)

- Formulario de producto mostrando campo de precio con valor "-100"
- No hay validación visual que lo impida
- El producto se guarda exitosamente con precio negativo en la tabla

| | | | | |
|------|---|---------------------|-----------|---|
| PASZ |  | Pastel de zanahoria | \$350.00 | Editar Eliminar |
| PASN |  | pastel de nuez | \$-200.00 | Editar Eliminar |

Intentar crear productos duplicados

No permite crear dos o más productos con el mismo identificador, pero si con el mismo nombre y misma imagen.

| | | | | |
|------|---|---------------------|-----------|--|
| PASZ |  | Pastel de zanahoria | \$350.00 | |
| PASN |  | pastel de nuez | \$-200.00 | |
| PASE |  | pastel de nuez | \$0.00 | |

3. Pruebas del carrito de compras

- Vista del carrito con productos agregados
- Cada fila muestra: imagen miniatura, nombre del producto, precio unitario, selector de cantidad con botones (+/-), subtotal
- Total general calculado en la parte inferior

Tu Carrito de Compras

| PRODUCTO | CANTIDAD | PRECIO UNITARIO | SUBTOTAL | A |
|---|----------|-----------------|----------|----|
| Pastell de vainilla (Crema Batida, Pequeño (10 pers), Vainilla) | 1 | \$350.00 | \$350.00 | EI |
| Pastel de Fresa y Crema (Crema Batida, Pequeño (10 pers), Vainilla) | 1 | \$380.00 | \$380.00 | EI |

Total: \$730.00 [Comprar Carrito](#)

- Añadir cantidades grandes

Tu Carrito de Compras

| PRODUCTO | CANTIDAD | PRECIO UNITARIO | SUBTOTAL | A |
|---|----------|-----------------|----------|----|
| Pastell de vainilla (Crema Batida, Pequeño (10 pers), Vainilla) | 1 | \$350.00 | \$350.00 | EI |
| Pastel de Fresa y Crema (Crema Batida, Pequeño (10 pers), Vainilla) | 1 | \$380.00 | \$380.00 | EI |
| Pastel de Chocolate (Crema Batida, Pequeño (10 pers), Vainilla) | 1 | \$350.00 | \$350.00 | EI |
| Pastel de Chocolate (Crema Batida, Pequeño (10 pers), Vainilla) | 1 | \$350.00 | \$350.00 | EI |

Total: \$1430.00 [Comprar Carrito](#)

Al intentar finalizar compra, aparece mensaje de validación

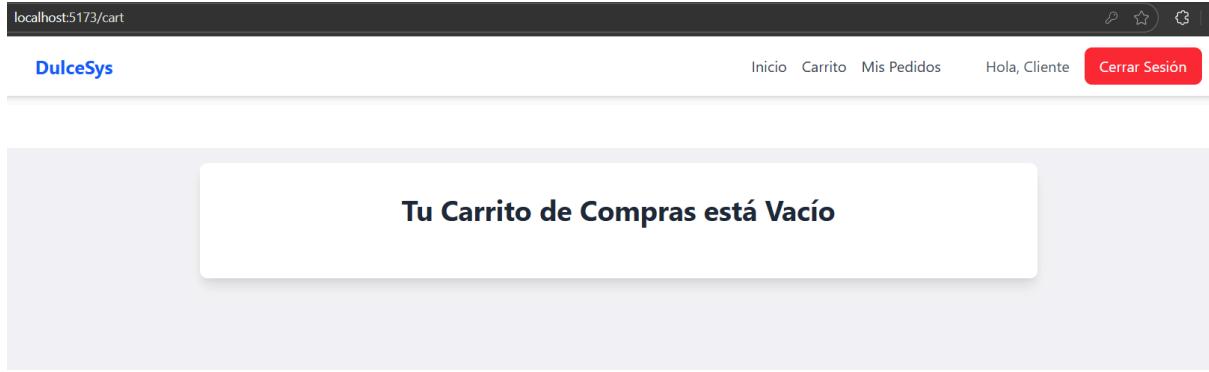


"Stock insuficiente. Solo hay X unidades disponibles"

The image shows a summary page for a shopping cart. At the top, the heading "Tu Carrito de Compras" is displayed in a bold, dark font. Below it is a table with the following data:

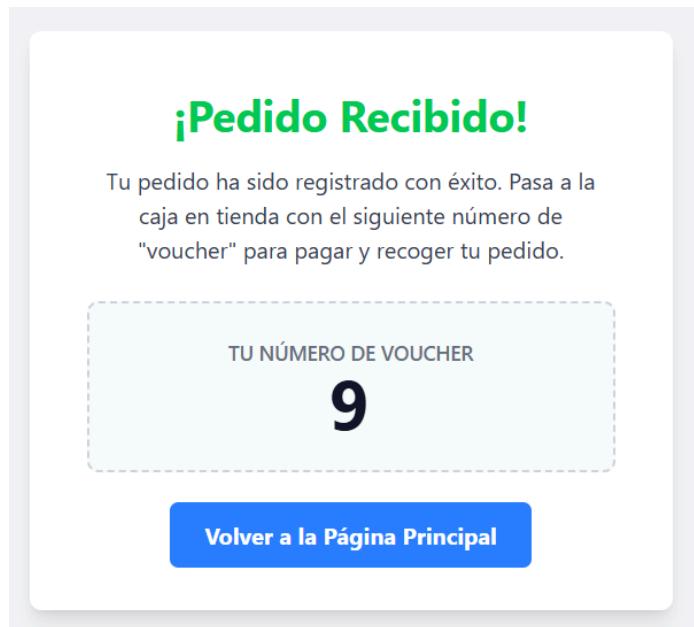
| PRODUCTO | CANTIDAD | PRECIO UNITARIO | SUBTOTAL | A |
|---|----------|-----------------|----------|----|
| Pastel de Fresa y Crema (Crema Batida, Pequeño (10 pers), Vainilla) | 1 | \$380.00 | \$380.00 | EI |
| Pastel de Chocolate (Crema Batida, Pequeño (10 pers), Vainilla) | 1 | \$350.00 | \$350.00 | EI |

At the bottom of the table is a horizontal scrollbar. To the right of the table, the total amount "Total: \$730.00" is shown in bold black text, followed by a pink button with white text that says "Comprar Carrito".



Hacer pedido desde el carrito

- Vista del carrito completo con múltiples productos
- Pantalla de confirmación mostrando resumen del pedido
- Mensaje de éxito "¡Pedido realizado con éxito!" con número de orden



Mis Pedidos

| ID PEDIDO | FECHA | ESTADO | TOTAL |
|-----------|------------|-----------|----------|
| #9 | 17/11/2025 | Pendiente | \$300.00 |
| #8 | 14/11/2025 | Pendiente | \$730.00 |

Solo permite comprar el carrito completo, si se desea comprar solo un producto tendría que eliminar los demás del carrito.

Tu Carrito de Compras

| PRODUCTO | CANTIDAD | PRECIO UNITARIO | SUBTOTAL | ACCIÓN |
|---|----------|-----------------|----------|--------------------------|
| Pastell de vainilla (Crema Batida, Pequeño (10 pers), Vainilla) | 1 | \$350.00 | \$350.00 | Eliminar |
| Pastel de oreo (Crema Batida, Pequeño (10 pers), Vainilla) | 1 | \$300.00 | \$300.00 | Eliminar |

Total: 650,00 \$

[Comprar Carrito](#)

4. Pruebas del módulo de pedidos (cajero / admin)

- Crear un pedido válido

El cajero si puede crear pedidos desde la página principal, también modificar el estatus de cada pedido, no puede filtrar pedidos por fecha, cliente o estado, ni acceder a pedidos de otros usuarios

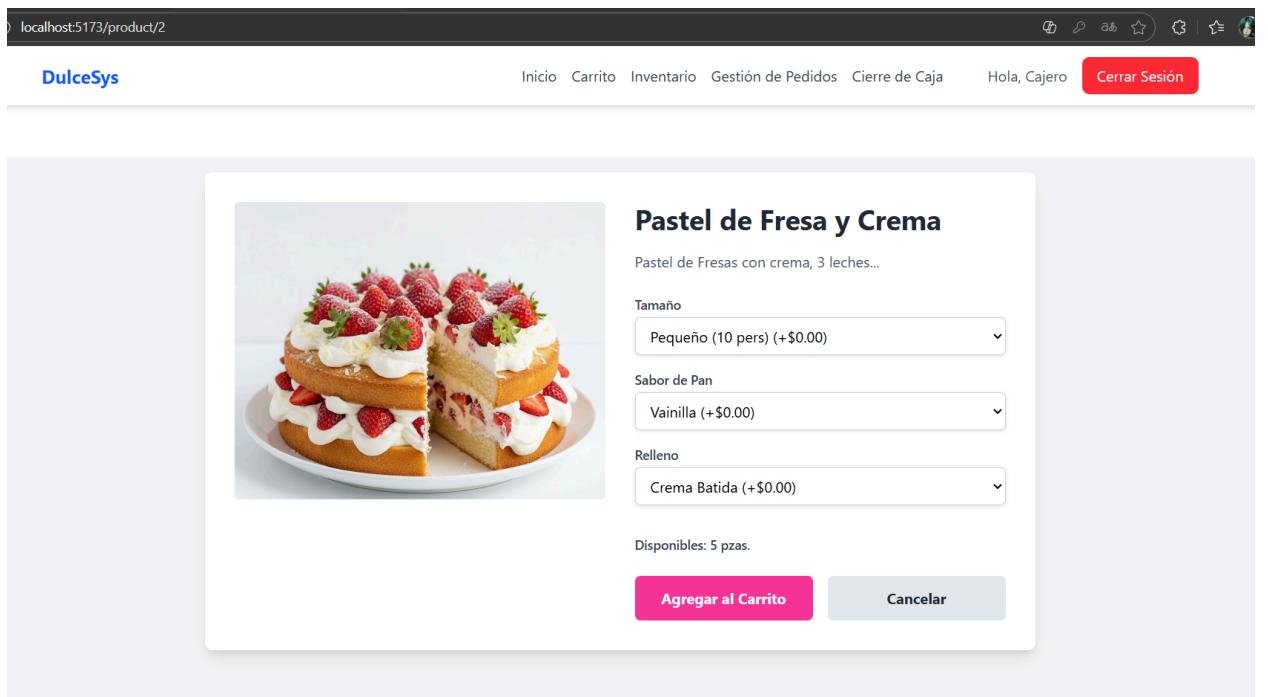


Tabla completa de pedidos con columnas: ID orden, Fecha, Cliente, Productos, Total, Esta

The screenshot shows a web application interface for managing orders. At the top, there is a header with the URL 'localhost:5173/gestion/pedidos', the logo 'DulceSys', and navigation links for 'Inicio', 'Carrito', 'Inventario', 'Gestión de Pedidos', 'Cierre de Caja', 'Hola, Cajero', and a red 'Cerrar Sesión' button.

The main content area is titled 'Gestión de Pedidos'. It has two tabs: 'Pedidos Activos' (selected) and 'Historial (Terminados)'. Below the tabs is a search bar with the placeholder 'Buscar por ID de pedido o nombre de cliente...'. The main table lists the following data:

| IDENTIFICACIÓN | CLIENTE | HORA | MÉTODO PAGO | TOTAL | ESTADO | ACCIÓN |
|----------------|----------------|-------|---------------|-----------|----------------|---------------|
| #2 | Ana Martínez | 20:49 | | \$1200.00 | En preparación | Procesar Pago |
| #4 | Ana Martínez | 21:26 | Tarjeta | \$350.00 | Listo | Procesar Pago |
| #5 | Ana Martínez | 21:39 | Transferencia | \$300.00 | Listo | Procesar Pago |
| #8 | Tania Santiago | 20:19 | Efectivo | \$730.00 | Pendiente | Procesar Pago |

5. Pruebas de stock e inventario

- Verificar que disminuir stock funcione al vender

Si entramos como clientes nos aparece que solo hay un pastel de chocolate

Tarjeta del producto "Pastel de Chocolate"

Indicador visible: "Disponible: 1 unidad"

Precio mostrado, botón "Agregar al carrito" activo

The screenshot shows a product detail page for 'Pastel de Chocolate'. The top navigation bar includes the URL 'localhost:5173/product/1', the 'DulceSys' logo, and links for 'Inicio', 'Carrito', 'Mis Pedidos', 'Hola, Cliente', and 'Cerrar Sesión'.

The main content area displays a large image of a chocolate cake with a slice removed, revealing the interior layers. The product title is 'Pastel de Chocolate' and the description is 'Clásico pastel de chocolate'. The configuration options are as follows:

- Tamaño: Pequeño (10 pers) (+\$0.00)
- Sabor de Pan: Vainilla (+\$0.00)
- Relleno: Crema Batida (+\$0.00)

A note at the bottom states 'Disponibles: 1 pzas.' (Available: 1 pieces). At the bottom right are two buttons: a pink 'Agregar al Carrito' (Add to Cart) button and a grey 'Cancelar' (Cancel) button.

Finalizamos la compra

Producto agregado al carrito con cantidad 1

Tu Carrito de Compras

| PRODUCTO | CANTIDAD | PRECIO UNITARIO | SUBTOTAL | ACCIÓN |
|--|----------|-----------------|----------|--------------------------|
| Pastel de vainilla (Crema Batida, Pequeño (10 pers), Vainilla) | 1 | \$350.00 | \$350.00 | Eliminar |
| Pastel de oreo (Crema Batida, Pequeño (10 pers), Vainilla) | 1 | \$300.00 | \$300.00 | Eliminar |
| Pastel de Chocolate (Crema Batida, Pequeño (10 per), Vainilla) | 1 | \$350.00 | \$350.00 | Eliminar |

Total: 1000,00 \$ [Comprar Carrito](#)

Finalizamos en pedido y muestra la pantalla de total a pagar y el tipo de pago

Finalizar Pedido

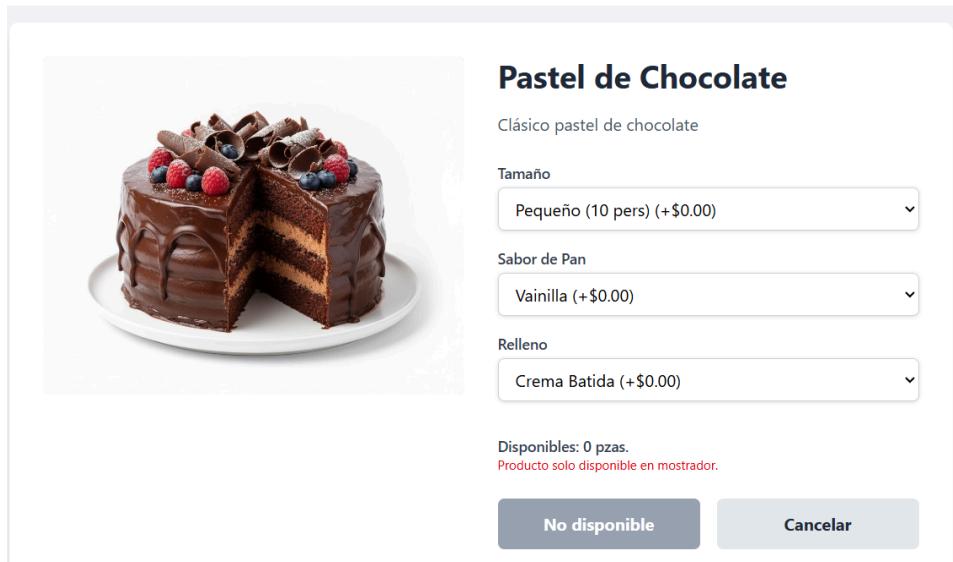
Total a Pagar:
\$1000.00

Con cuánto vas a pagar?
1000

Mensaje final de pedido Recibido con éxito, con numero de voucher correspondiente



Ahora si checamos de nuevo para comprar otro pastel de chocolate, podemos ver que ya no hay más pasteles.



6. Pruebas de ingredientes (si es parte del sistema)

- Crear ingrediente
- Formulario con campos: Nombre del ingrediente, Cantidad, Unidad de medida

| SKU | NOMBRE | UNIDAD | STOCK ACTUAL | STOCK MINIMO | ACCIONES |
|-------|------------------|--------|--------------|--------------|---|
| AZU2 | Azúcar | kg | 50.00 | 10.00 | Editar Eliminar |
| LEC3 | Leche | Lt | 20.00 | 5.00 | Editar Eliminar |
| HUE4 | Huevo | PZA | 100.00 | 24.00 | Editar Eliminar |
| CHOS5 | Chocolate Amargo | kg | 15.00 | 3.00 | Editar Eliminar |
| FRE6 | Fresa | kg | 10.00 | 2.00 | Editar Eliminar |
| HAR01 | Harina | kg | 50.00 | 10.00 | Editar Eliminar |

Lista actualizada mostrando el nuevo ingrediente con opciones de editar y eliminar

Agregar Ingrediente

SKU
CB01

Nombre
Crema para batir

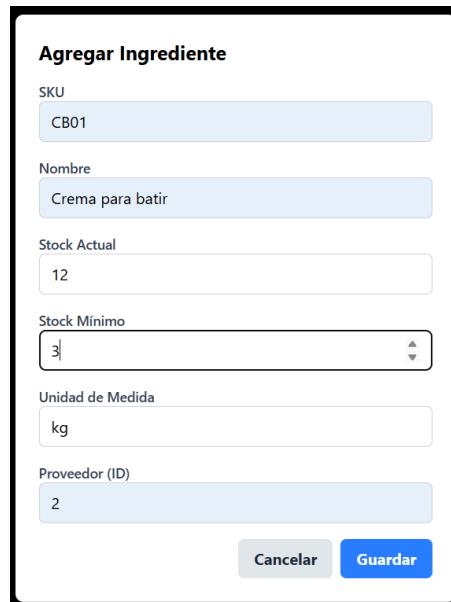
Stock Actual
12

Stock Mínimo
3

Unidad de Medida
kg

Proveedor (ID)
2

Cancelar **Guardar**



Editar ingrediente

- Modal de edición emergente con formulario prellenado
- Campos editables: nombre, cantidad, unidad
- Botones: "Guardar cambios" y "Cancelar"

Editar Ingrediente

SKU
LEC3

Nombre
Leche

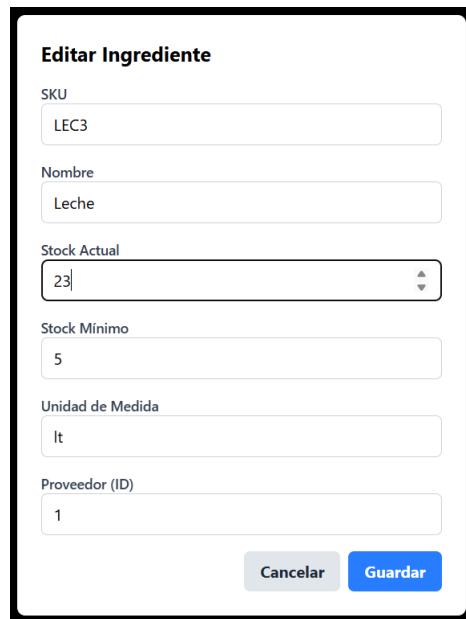
Stock Actual
23

Stock Mínimo
5

Unidad de Medida
lt

Proveedor (ID)
1

Cancelar **Guardar**



- Ver ingredientes asignados a productos

En el inicio se muestran las características de los pasteles, pero dentro del inventario no se ven asignados los ingredientes.



Pastell de vainilla

Paste de Vainilla, clásico...

Tamaño

Pequeño (10 pers) (+\$0.00)

Sabor de Pan

Vainilla (+\$0.00)

Relleno

Crema Batida (+\$0.00)

Disponibles: 10 pzas.

Agregar al Carrito

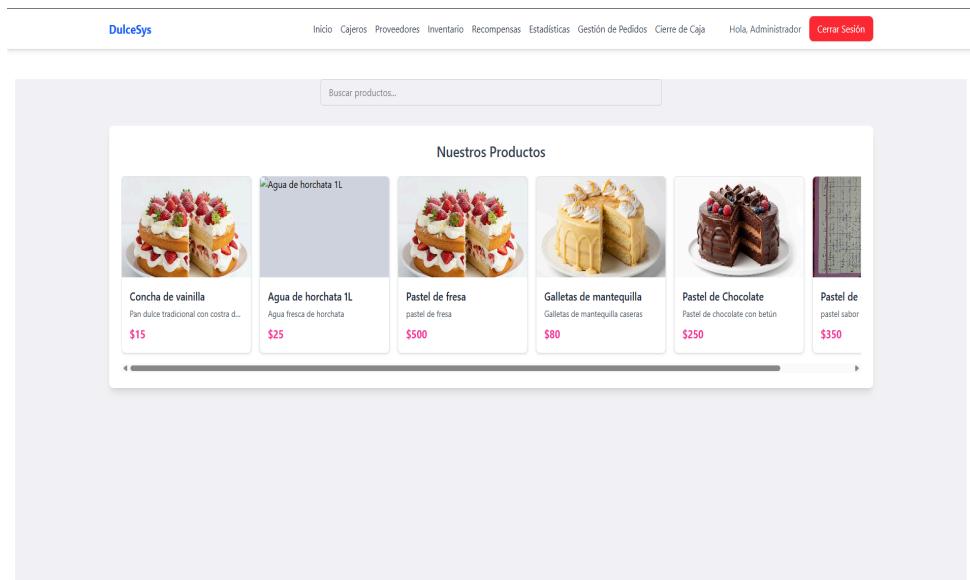
Cancelar

7. Pruebas de interfaz

- Verificar que los menús cambian según el rol

Administrador:

- Opciones visibles: Dashboard, Productos, Inventario, Pedidos, Usuarios, Ingredientes, Configuración (7+ opciones)
- Acceso total a todas las funcionalidades



Cliente:

- simple con 4 opciones
- Menú: Inicio, Catálogo, Carrito (con badge de cantidad), Mis Pedidos
- Interfaz minimalista enfocada en compras

The screenshot shows a clean, minimalist website interface for customers. At the top, there's a navigation bar with links for 'Inicio', 'Carrito', 'Mis Pedidos', 'Hola, Cliente', and a red 'Cerrar Sesión' button. Below the navigation is a search bar labeled 'Buscar productos...'. The main content area is titled 'Nuestros Productos' and displays six items in a grid: 'Concha de vainilla' (\$15), 'Agua de horchata 1L' (\$25), 'Pastel de fresa' (\$500), 'Galletas de mantequilla' (\$80), 'Pastel de Chocolate' (\$250), and 'Pastel de pastel sabor' (\$350). Each item has a small image, a name, a description, and a price.

Cajero:

- Menú intermedio con 3 opciones
- Visible: Inicio, Gestión de Pedidos, Perfil
- Permisos limitados comparado con administrador

This screenshot shows the cashier's version of the website. It includes additional links in the top navigation: 'Inventario', 'Gestión de Pedidos', and 'Cierre de Caja'. The rest of the interface is identical to the customer version, featuring the 'Nuestros Productos' section with the same six items and their details.

Resultados prueba de interfaz:

Esta prueba se realizó sin problemas y se visualiza una clara diferencia entre los menús de acuerdo al rol en el que se ingrese a la página.

8. Pruebas de la Base de Datos (funcionales)

- Crear datos y verificar que se guardan en la BD

Llenar formulario:

Agregar Producto

SKU
PST-TEST-03

Nombre
Pastel de prueba 3

Descripción
PST-TEST-01PST-TEST-01PST-TEST-01PST-TEST-01

Categoría
Pasteles

Precio 150 **Stock** 6

Imagen
 Imagen de Whats....38_b2ba3dd5.jpg

Cancelar **Guardar**

Visualización del producto:

The screenshot shows a web-based inventory management application. At the top, there's a navigation bar with links for 'Inicio', 'Cajeros', 'Proveedores', 'Inventario', 'Recompensas', 'Estadísticas', 'Gestión de Pedidos', 'Cierre de Caja', 'Hola, Administrador', and a red 'Cerrar Sesión' button. Below the navigation is a search bar labeled 'Buscar productos por nombre o SKU...' and a green 'Agregar Producto' button. The main area is titled 'Gestión de Inventario' and has tabs for 'Productos' (which is selected) and 'Ingredientes'. A table lists four products: 'Galletas de mantequilla', 'Pastel de Chocolate', 'Pastel de prueba 2', and 'Pastel de prueba 3'. Each product row includes a small image, the product name, price (\$80.00, \$250.00, \$1225.00, \$150.00), stock level (4, 50, 10, 6), and edit/delete buttons.

Resultado exitoso en la BD:

The screenshot shows a pgAdmin interface with a query editor and a results grid. The query editor contains the following SQL code:

```
1 SELECT id_producto, sku, nombre, descripcion, preciobase, stockproductosterminalados
2 FROM productos
3 WHERE sku = 'PST-TEST-03';
4
```

The results grid displays one row of data:

| | id_producto | sku | nombre | descripcion | preciobase | stockproductosterminalados |
|---|-------------|-------------|--------------------|--|------------|----------------------------|
| 1 | 13 | PST-TEST-03 | Pastel de prueba 3 | PST-TEST-01PST-TEST-01PST-TEST-01PST-TEST... | 150.00 | 6 |

At the bottom of the pgAdmin window, status messages indicate 'Total rows: 1' and 'Query complete 00:00:00.096'.

- Editar y verificar que se actualicen

Llenar formulario:

Editar Producto

SKU
PST-TEST-03

Nombre
Pastel de prueba 3

Descripción
Edición para prueba

Categoría
Pasteles

Precio
145

Stock
7

Imagen
 Sin archivos seleccionados
Dejar en blanco para conservar la imagen actual.

Visualización del producto:

The screenshot shows the 'Gestión de Inventario' (Inventory Management) section of the DulceSys application. At the top, there are tabs for 'Productos' (Products) and 'Ingredientes' (Ingredients), with 'Productos' being the active tab. Below the tabs is a search bar labeled 'Buscar productos por nombre o SKU...'. To the right of the search bar is a green button labeled 'Agregar Producto' (Add Product). The main area displays a table with four rows of product data:

| SKU | IMAGEN | NOMBRE | PRECIO | STOCK | ACCIONES |
|-------------|--------|-------------------------|-----------|-------|---|
| GAL-001 | | Galletas de mantequilla | \$80.00 | 4 | Editar Eliminar |
| PST-001 | | Pastel de Chocolate | \$250.00 | 50 | Editar Eliminar |
| PST-TEST-02 | | Pastel de prueba 2 | \$1225.00 | 10 | Editar Eliminar |
| PST-TEST-03 | | Pastel de prueba 3 | \$145.00 | 7 | Editar Eliminar |

Resultado en BD:

The screenshot shows a PostgreSQL database client interface with a dark theme. The top bar shows the connection details: 'dulcesys/postgres@PostgreSQL 18*'. The main area has two panes: 'Query' on the left and 'Scratch Pad' on the right. The 'Query' pane contains the following SQL code:

```
1 SELECT id_producto, sku, nombre, descripcion, preciobase, stockproductosterminalados
2 FROM productos
3 WHERE sku = 'PST-TEST-03';
4
5
```

The 'Data Output' pane below shows the results of the query:

| | id_producto | sku | nombre | descripcion | preciobase | stockproductosterminalados |
|---|-------------|-------------|--------------------|----------------------|------------|----------------------------|
| 1 | 13 | PST-TEST-03 | Pastel de prueba 3 | Edición para prue... | 145.00 | 7 |

At the bottom of the client window, status messages indicate: 'Total rows: 1' and 'Query complete 00:00:00.077'.

Resultados prueba de base de datos:

Esta prueba se realizó sin problema alguno, en las imágenes se visualiza de manera clara los cambios realizados tanto en la base de datos como en la interfaz.

Sincronización perfecta entre frontend, backend y base de datos

- Operaciones CRUD completas funcionando correctamente
- Integridad de datos mantenida en todas las operaciones
- Timestamps actualizándose apropiadamente (created_at y updated_at)
- Tipos de datos correctos en la base de datos (integer, decimal, varchar, timestamp)
- Sin pérdida de información durante actualizaciones parciales

REPORTE DE PRUEBAS

REPORTE DE PRUEBAS.

1- Prueba en carrito de compra: en esta prueba se buscaron errores en cuanto a la actualización de stock. para ver fallas se hizo lo siguiente: se añadieron al carrito productos, se noto que el stock visible antes de agregar al carrito era 21, se agregaron 2 piezas.



Pastel de oreo

Pastel de 3 leches, hecho con galletas oreo...

Tamaño

Pequeño (10 pers) (+\$0.00)

Sabor de Pan

Vainilla (+\$0.00)

Relleno

Crema Batida (+\$0.00)

Disponibles: 21 pzas.

Agregar al Carrito **Cancelar**

Tu Carrito de Compras

| PRODUCTO | CANTIDAD | PRECIO UNITARIO | SUBTOTAL | ACCIÓN |
|--|----------|-----------------|-----------|-----------------|
| Pastel de oreo (Crema Batida, Pequeño (10 pers), Vainilla) | 1 | 350,00 \$ | 350,00 \$ | Eliminar |
| Pastel de oreo (Crema Batida, Pequeño (10 pers), Vainilla) | 1 | 350,00 \$ | 350,00 \$ | Eliminar |

Subtotal: 700,00 \$

Total: 700,00 \$

Comprar Carrito

Resultados:

- Despues de agregar el producto al carrito se nota lo siguiente: el stock no disminuye en el carrusel de inicio, la disminución se verá reflejada hasta que el pedido se marque como completado.
- El cliente al realizar un pedido no puede cancelarlo solo ver la fecha y el estado.

The screenshot shows a user interface for managing orders. At the top, there is a navigation bar with links for 'Inicio', 'Carrito', 'Mis Pedidos', 'Hola, Cliente', and a red 'Cerrar Sesión' (Logout) button. Below the navigation is a section titled 'Mis Pedidos' (My Orders). This section contains a table with one row of data. The columns are labeled 'ID PEDIDO', 'FECHA', 'ESTADO', and 'TOTAL'. The data row shows: ID PEDIDO #24, FECHA 28/11/2025, ESTADO Pendiente (Pending), and TOTAL \$700.00.

| ID PEDIDO | FECHA | ESTADO | TOTAL |
|-----------|------------|-----------|----------|
| #24 | 28/11/2025 | Pendiente | \$700.00 |

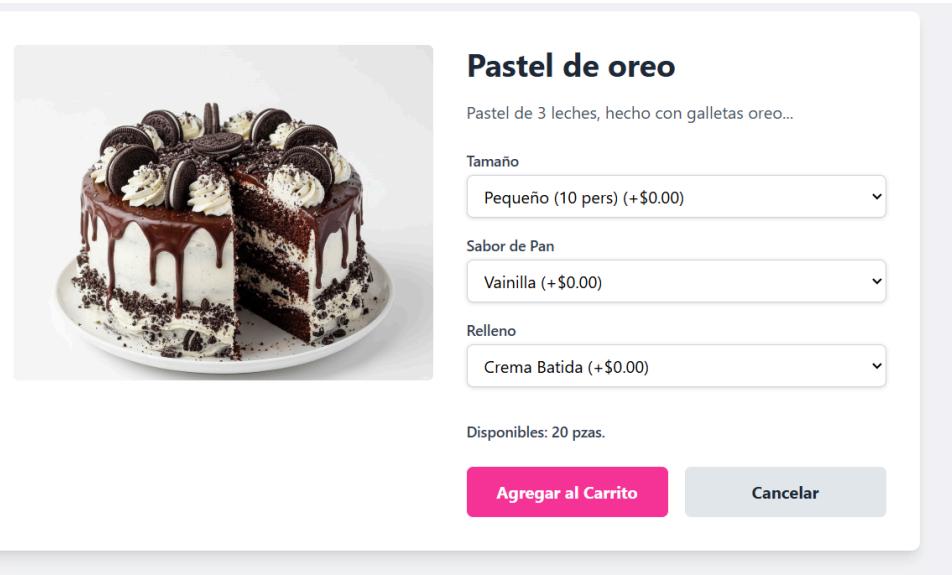
- Solo el admin puede cancelar los pedidos, ya que el cajero tampoco puede hacerlo

Gestión de Pedidos

The screenshot shows a user interface for managing completed orders. At the top, there are two tabs: 'Pedidos Activos' and 'Historial (Terminados)', with 'Historial (Terminados)' being the active tab. Below the tabs is a search bar with the placeholder text 'Buscar por ID de pedido o nombre de cliente...'. The main area displays a table of completed orders. The columns are labeled 'IDENTIFICACIÓN', 'CLIENTE', 'FECHA', 'HORA', 'MÉTODO PAGO', 'TOTAL', 'ESTADO', and 'ACCIÓN'. One row of data is visible: IDENTIFICACIÓN #24, CLIENTE Tania, FECHA 28/11/2025, HORA 14:35, MÉTODO PAGO Efectivo, TOTAL \$700.00, ESTADO Completado (Completed), and ACCIÓN with a 'Editar Estado' (Edit Status) link.

| IDENTIFICACIÓN | CLIENTE | FECHA | HORA | MÉTODO PAGO | TOTAL | ESTADO | ACCIÓN |
|----------------|---------|------------|-------|-------------|----------|------------|-------------------------------|
| #24 | Tania | 28/11/2025 | 14:35 | Efectivo | \$700.00 | Completado | Editar Estado |

- Al ingresar como cliente vemos que solo disminuyó un pastel de oreo, teníamos 21, compramos 2 y ahora nos aparece que tenemos 20, aquí se detecta un error, es stock solo disminuye en uno a pesar de haber comprado 2 del mismo tipo.

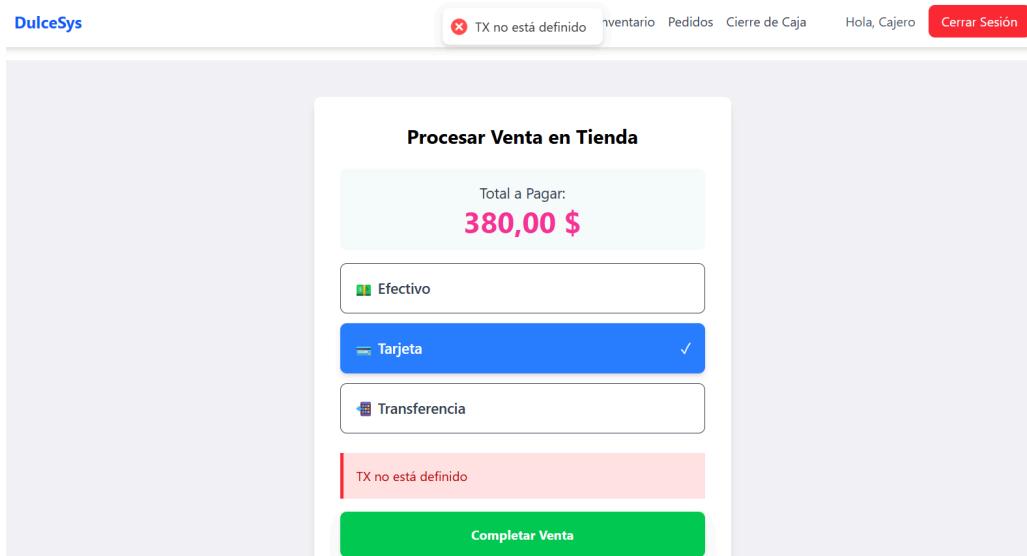


Solución a errores:

En el [index.js](#) se modificó la ruta que se encargaba de cargar los pedidos a la BD. Anteriormente se usaba una función decrement, la cual generaba únicamente un decremento de 1 al producto. Ahora se usa un decrement : item.cantidad. De tal forma que ahora descuenta la cantidad de productos comprados en el carrito.

2- Prueba al hacer compras como cajero:

Al ingresar el monto a pagar, sale ese error por tanto no se guarda en pedidos



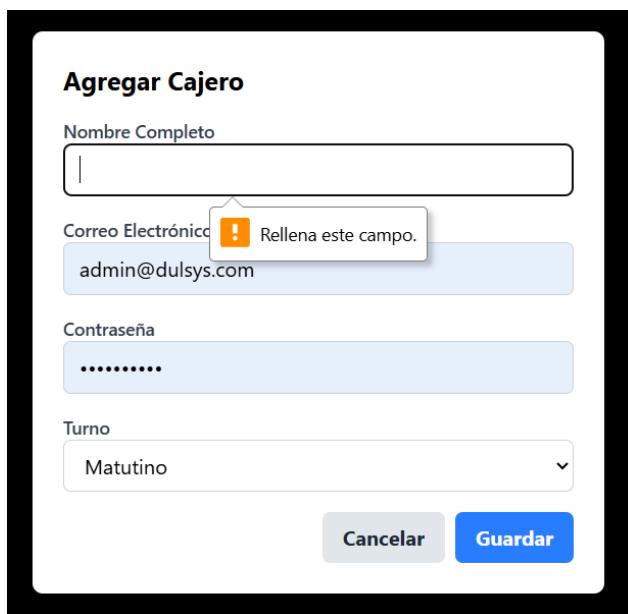
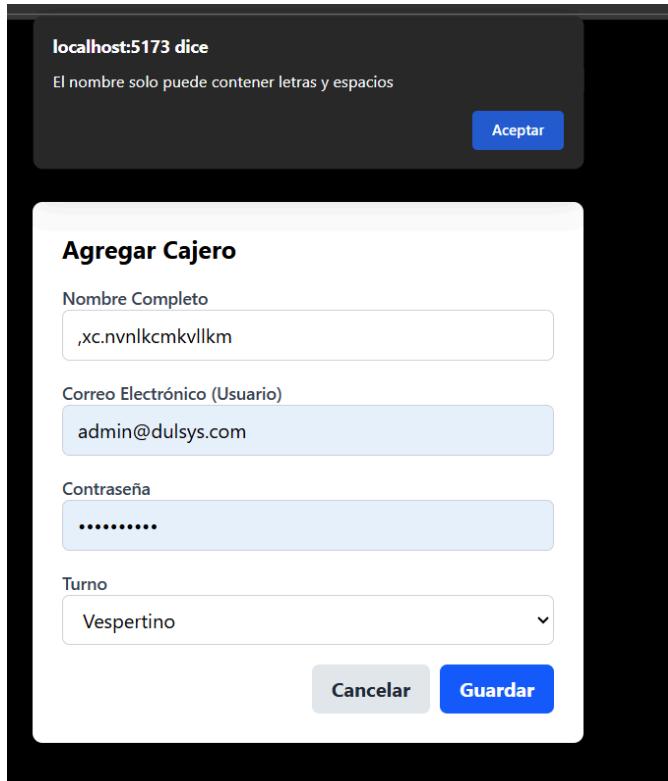
Resultados:

- Se detectó un solo error durante el flujo de compra, al momento de hacer el pago con cualquier método se genera el error “TX no está definido” este error no permite completar de manera exitosa la compra.

3- Prueba de validaciones:

En este punto se realizaron varias pruebas a distintas partes del programa en las que se requiere completar formularios, ya sean para registro o de inicio de sesión, se probaron las validaciones al ingresar datos erróneos o inexistentes.

- Agregar cajero desde admin:



- Inicio de sesión y registro.

Iniciar Sesión

Credenciales inválidas

Correo Electrónico:

angelasantiago@gmail.com

Contraseña:

.....

Ingresar

¿No tienes una cuenta? [Regístrate aquí](#)

Iniciar Sesión

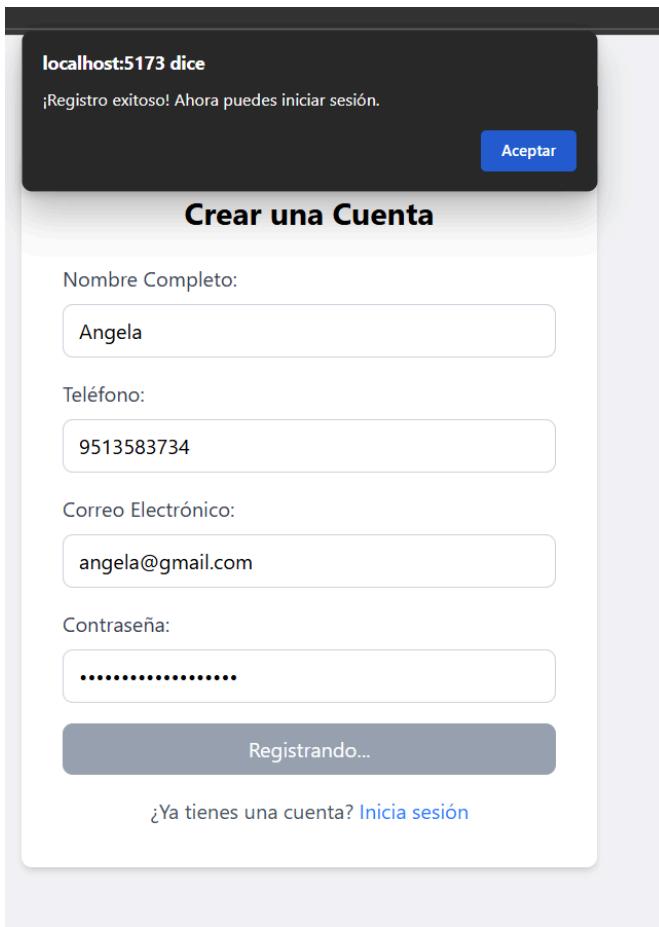
Correo y contraseña son requeridos

Correo Electrónico:

Contraseña:

Ingresar

¿No tienes una cuenta? [Regístrate aquí](#)

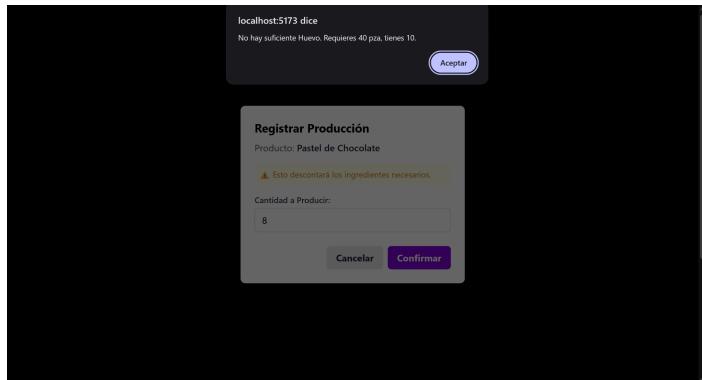


Resultados:

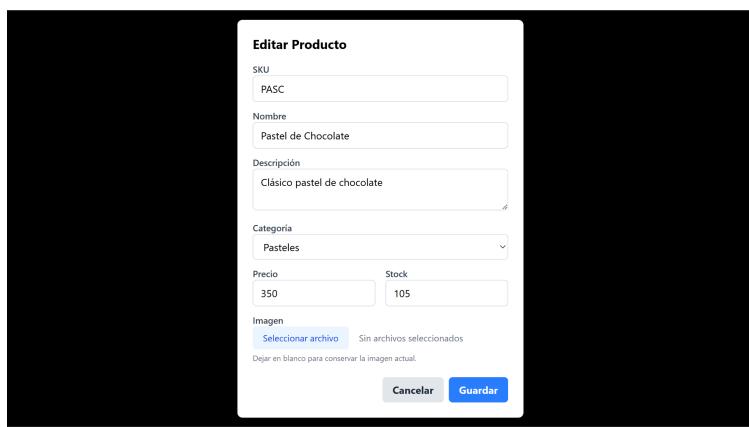
- Estas pruebas resultaron exitosas, no se encontraron fallas en cuanto a validaciones.

4- Pruebas de lógica:

Se realizaron pruebas de funcionalidad y lógica a los botones integrados dentro de la pestaña de inventario desde administrador, se puso a prueba la funcionalidad del botón “Producir” la cual es correcta y lanza un error si la cantidad a producir supera la capacidad respecto a los ingredientes.



También se realizó una prueba a la pestaña editar, en la misma aparece un campo de texto en el que puedes escribir un nuevo stock.



| Inventario | | | | | | Cierre de Caja | Recompensas | Estadísticas | Hola, Usuario | ADMINISTRADOR |
|------------|------|-------------------------|----------|-----|--|---|-------------|--------------|---------------|---------------|
| | RED1 | Pastel de Redvelvet | \$400.00 | 12 | | Receta Producir Editar Eliminar | | | | |
| | PASF | Pastel de Fresa y Crema | \$380.00 | 2 | | Receta Producir Editar Eliminar | | | | |
| | PASV | Pastel de vainilla | \$350.00 | 6 | | Receta Producir Editar Eliminar | | | | |
| | PASO | Pastel de oreo | \$350.00 | 40 | | Receta Producir Editar Eliminar | | | | |
| | PASC | Pastel de Chocolate | \$350.00 | 105 | | Receta Producir Editar Eliminar | | | | |

Resultados:

- Se detectó un error de lógica ya que como se aprecia en la imagen anterior no se podían producir 8 productos nuevos desde el botón “Producir” caso contrario al campo de texto “Stock”.

Solución:

- Se modificó el modal de agregar/editar productos para que cuando se modifique el inventario solamente se pueda reducir la cantidad y no aumentar.

Puesto que el error de lógica entraría en caso de que si ya tenemos la funcionalidad de “producir” que aumenta, esta otra función solo puede “romper” la lógica ya establecida. Por lo que se decidió que solamente se pudiera disminuir en el caso de merma o algo parecido.

En el modal que se encuentra en InventoryAdminPage se agregó esta funcionalidad. Se hace desde el frontend.

Editar Producto

SKU Para aumentar el stock, utiliza el botón “Producir”.

PASo

Nombre Pastel de oreo

Descripción Pastel de 3 leches, hecho con galletas oreo...

Categoría Pasteles

Precio 350 Stock 1

* Solo para ajustes de merma. Para agregar, usa “Producir”.

Imagen No se eligió ningún archivo
Dejar en blanco para conservar la imagen actual.

5- prueba en pedidos:

En la pestaña pedidos no se encontraron errores que afecten al funcionamiento de la página, se revisó la pestaña y se detectaron problemáticas puntuales que si bien no afectan el flujo, pueden llegar a ser importantes para el buen desempeño de la misma.

| Gestión de Pedidos | | | | | | | |
|--|------------------------|------------|-------|---------------|-----------|-----------|-----------------------------|
| Pedidos Activos | Historial (Terminados) | | | | | | |
| Buscar por ID de pedido o nombre de cliente... | | | | | | | |
| IDENTIFICACIÓN | CLIENTE | FECHA | HORA | MÉTODO PAGO | TOTAL | ESTADO | ACCIÓN |
| #16 | Ana Martínez | 25/11/2025 | 19:40 | Efectivo | \$476.00 | Pendiente | Procesar Pago Editar Estado |
| #17 | Ana Martínez | 25/11/2025 | 19:40 | Tarjeta | \$471.75 | Pendiente | Procesar Pago Editar Estado |
| #18 | Ana Martínez | 25/11/2025 | 19:41 | Transferencia | \$297.50 | Pendiente | Procesar Pago Editar Estado |
| #19 | Ana Martínez | 25/11/2025 | 19:41 | Transferencia | \$297.50 | Pendiente | Procesar Pago Editar Estado |
| #20 | Ana Martínez | 25/11/2025 | 19:41 | Tarjeta | \$297.50 | Pendiente | Procesar Pago Editar Estado |
| #21 | Ana Martínez | 25/11/2025 | 19:42 | Tarjeta | \$2380.00 | Pendiente | Procesar Pago Editar Estado |
| #22 | Ana Martínez | 25/11/2025 | 19:43 | Tarjeta | \$1190.00 | Pendiente | Procesar Pago Editar Estado |

Resultados:

- La problemática encontrada en pedidos radica en que no se podían visualizar detalles del pedido, tales como el producto que se compró, la cantidad y el desglose, esto generaba que la gestión de pedidos pueda llegar a ser ineficiente.

Solución:

- Por lo que se agregó la opción de Ver detalles, donde se abre un modal con todas las especificaciones del pedido seleccionado, como el número de pedido, fecha, hora, tipo de pago, estado, el desglose de productos, el subtotal, si se aplicaron descuentos, y en el caso del efectivo, con cuánto efectivo pagó y cuánto cambio se le debió regresar.

Gestión de Pedidos

[Pedidos Activos](#) [Historial \(Terminados\)](#)

Buscar por ID de pedido o nombre de cliente...

| ID | CLIENTE | FECHA | HORA | MÉTODO PAGO | TOTAL | ESTADO | ACCIÓN |
|-----|--------------|------------|------------|---------------|----------|-----------|--|
| #8 | Ana Martínez | 14/11/2025 | 08:07 p.m. | Efectivo | \$300.00 | Pendiente | Ver Detalles Procesar Pago |
| #16 | Ana Martínez | 25/11/2025 | 07:40 p.m. | Efectivo | \$476.00 | Pendiente | Ver Detalles Procesar Pago |
| #17 | Ana Martínez | 25/11/2025 | 07:40 p.m. | Tarjeta | \$471.75 | Pendiente | Ver Detalles Procesar Pago |
| #18 | Ana Martínez | 25/11/2025 | 07:41 p.m. | Transferencia | \$297.50 | Pendiente | Ver Detalles Procesar Pago |

Detalles del Pedido #25

Cliente: Ana Martínez Fecha: 30/11/2025 - 02:07 p.m.

Pago: Tarjeta Estado: Completado

| Producto | Detalles | Cant. | P. Unit | Subtotal |
|--------------------|--|-------|----------|----------|
| Pastel de vainilla | • Relleno: Crema Batida • Tamaños Pequeño (10 pers) • Sabor de París: Vainilla | 1 | \$350.00 | \$350.00 |

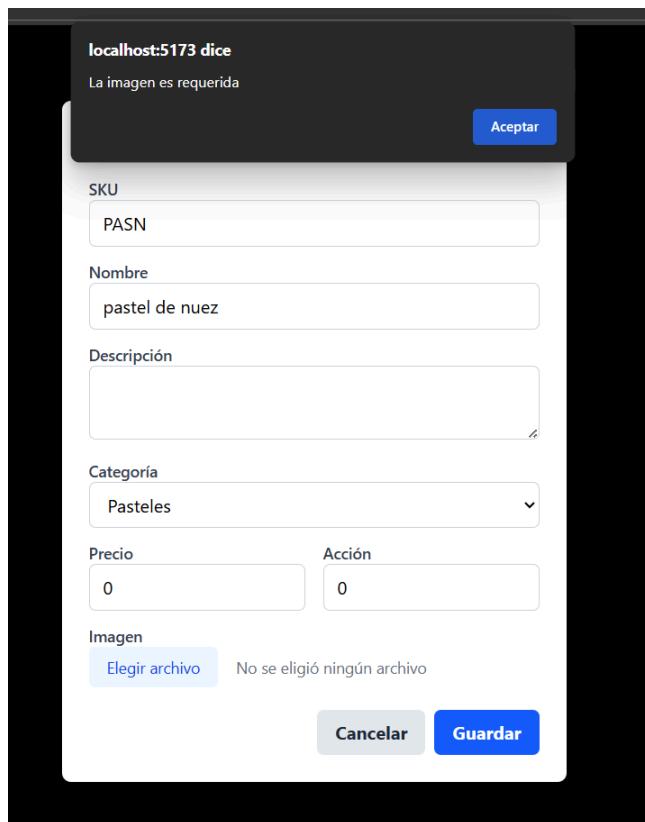
Subtotal: \$350.00
Descuento: -\$52.50

Total Cobrado: **\$297.50**

[Cerrar](#)

6- CORRECCIÓN DE IMÁGENES DUPLICADAS.

Durante las pruebas de registro de nuevos productos, se validaron ingresar un nuevo producto con el nombre y SKU de alguno ya existente, además de probar los botones editar y eliminar.



Gestión de Inventario

Productos Ingredientes

Buscar productos...

Agregar Producto

| IMAGEN | SKU | NOMBRE | PRECIO | STOCK | ACCIONES |
|--------|------|---------------------|----------|-------|---|
| | PASO | Pastel de oreo | \$350.00 | 18 | Receta Producir Editar Eliminar |
| | PASC | Pastel de Chocolate | \$350.00 | 9 | Receta Producir Editar Eliminar |
| | RED1 | Pastel de Redvelvet | \$400.00 | 12 | Receta Producir Editar Eliminar |

Editar Producto

| | |
|--|---|
| SKU | PASN |
| Nombre | pastel de nuez |
| Descripción | |
| Categoría | Pasteles |
| Precio | 0 |
| Imagen | <input type="button" value="Elegir archivo"/> No se eligió ningún archivo Dejar en blanco para conservar la imagen actual. |
| <input type="button" value="Cancelar"/> <input type="button" value="Guardar"/> | |

| | | | | | |
|------|--|---------------------|----------|---|---|
| PASC | | Pastel de Chocolate | \$350.00 | 3 | Editar Eliminar |
| PASZ | | Pastel de zanahoria | \$350.00 | | Editar Eliminar |
| PASN | | pastel de nuez | \$0.00 | | Editar Eliminar |

Resultado:

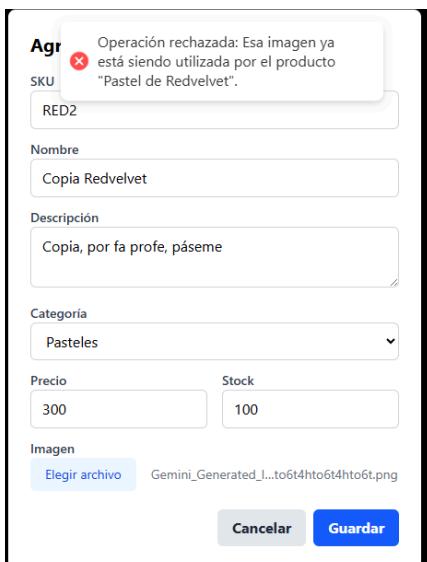
- Se encontró un error lógico y con ello una acción preventiva, se detectó que al registrar un nuevo producto el sistema permite que la imagen con la que saldría visible en el carrusel podría ser la misma que la de un producto ya registrado.

Solución:

- El problema se solucionó desde [index.js](#). y con la BD. En la base de datos, en la tabla *Producto* se agregó una columna que guarda un hash, este hash es generado cuando se carga una imagen, de esta forma se “identifica” la imagen y es posible “comparar” imágenes a través del Hash que estas generan.

Entonces si por ejemplo se duplicará la imagen, no solo se duplica en la interfaz, en la carpeta donde estén las imágenes también.

Así que se tuvo que hacer un manejo de ese hash para comparar las imágenes y que se determinara si ya están en uso o no. De esta forma no permite imágenes duplicadas en el front y ahorramos espacio en la carpeta de upload donde se guardan las imágenes.



7- Pruebas de la Base de Datos

Crear datos y verificar que se guardan en la BD, se realizaron pruebas de registro de productos y se busca visualizar el resultado del registro en la base de datos.

Agregar Producto

SKU

Nombre

Descripción

Categoría

Precio Stock

Imagen
 Imagen de Whats....38_b2ba3dd5.jpg

resultado:

- no se detectaron errores en cuanto a la conexión y registro de productos, el flujo de registro no se ve interrumpido o cortado por algún error.

```

SELECT id_producto, sku, nombre, descripcion, preciobase, stockproductostterminados
FROM productos
WHERE sku = 'PST-TEST-03';

```

| | id_producto [PK] integer | sku character varying (50) | nombre character varying (100) | descripcion text | preciobase numeric (10,2) | stockproductostterminados integer |
|---|------------------------------------|--------------------------------------|--|---------------------------------------|-------------------------------------|---|
| 1 | 13 | PST-TEST-03 | Pastel de prueba 3 | PST-TEST-01PST-TEST-01PST-01PST-TEST- | 150.00 | 6 |

Total rows: 1 Query complete 00:00:00.096 CRLF Ln 1, Col 1

8- MEJORA DE UX

Durante las pruebas se detectó que en el carrito no es posible aumentar la cantidad manualmente, se tienen que salir del carrito entrar al inicio, ir a ver el producto que quieren y agregarlo, si por ejemplo quisieran comprar 6 pasteles, se tendría que repetir esta operación 6 veces. Además de que visualmente consume mucho espacio y no es posible borrar todo el carrito.

Resultados:

- Se encontró que el flujo de este proceso ya mencionado es demasiado largo y puede llegar a ser poco eficiente el sistema.

Tu Carrito de Compras

| PRODUCTO | CANTIDAD | PRECIO UNITARIO | SUBTOTAL | ACCIÓN |
|--|----------|-----------------|-----------|--------------------------|
| Pastel de oreo (Crema Batida, Pequeño (10 pers), Vainilla) | 1 | 350,00 \$ | 350,00 \$ | Eliminar |
| Pastel de oreo (Crema Batida, Pequeño (10 pers), Vainilla) | 1 | 350,00 \$ | 350,00 \$ | Eliminar |
| Subtotal: | | | | 700,00 \$ |
| Total: | | | | 700,00 \$ |
| Comprar Carrito | | | | |

Solución:

- Se agregaron botones para aumentar las cantidades, la posibilidad de desechar un producto o vaciar todo el carrito. Y el problema del descuento de stock, se solucionó de una forma sencilla, el problema se encontraba en el index.js, ahí había un problema lógico con el arreglo de productos, puesto que el arreglo solo guardaba un objeto de tipo producto, pero no tomaba en cuenta la cantidad. Esto se arregló cambiando las especificaciones del arreglo para que pudieran guardarse los objetos y las cantidades para actuar sobre la BD.

Carrito de Compras

[Vaciar Carrito](#)

| PRODUCTO | CANTIDAD | PRECIO UNITARIO | SUBTOTAL | ACCIÓN |
|---|----------|-----------------|-----------------|------------------|
| Pastel de Chocolate (Crema Batida, Pequeño (10 pers), Vainilla) | - 2 + | \$350.00 | \$700.00 | |
| Pastel de oreo (Crema Batida, Pequeño (10 pers), Vainilla) | - 1 + | \$350.00 | \$350.00 | |
| Subtotal: | | | | \$1050.00 |
| Total: | | | | \$1050.00 |

[Proceder al Pago](#)

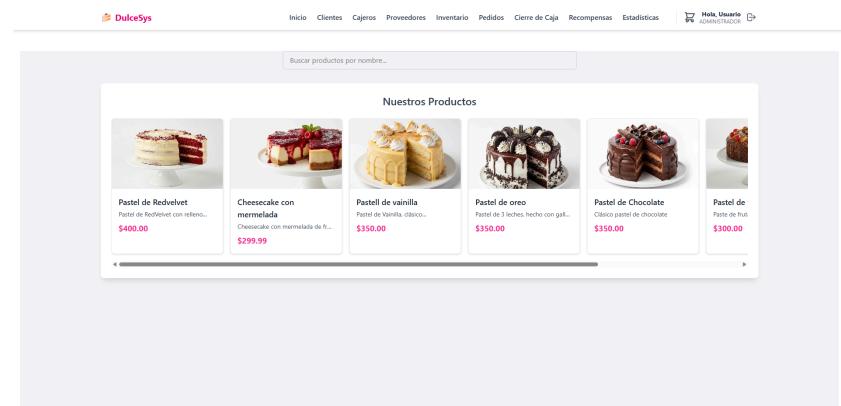
Este cambio también respeta la lógica de que no se puede agregar al pedido más de lo que se tiene en stock.

PRUEBAS FINALES

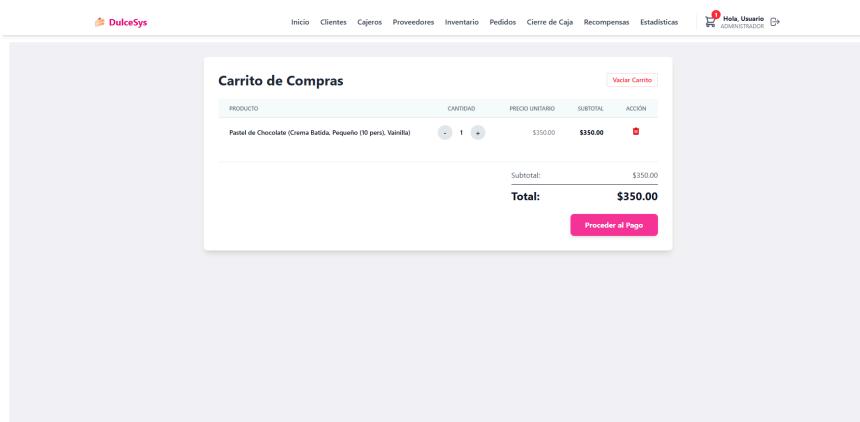
PRUEBAS FINALES

ADMIN:

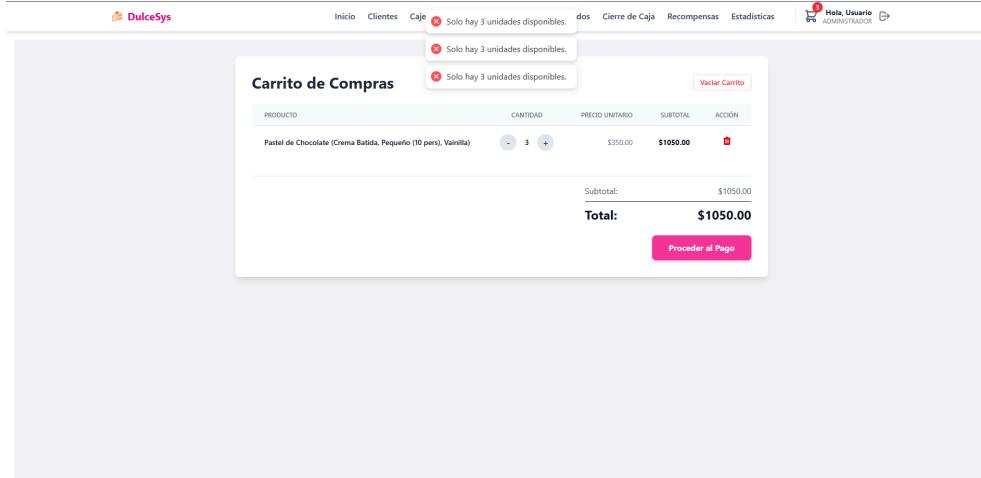
Prueba A1: Esta es la primera prueba y será una recurrente ya que se busca que se note las diferentes vistas que tiene el sistema por roles, en este caso se muestran diferentes apartados dentro del menú los cuales ayudarán a gestionar de manera eficiente el programa al administrador.



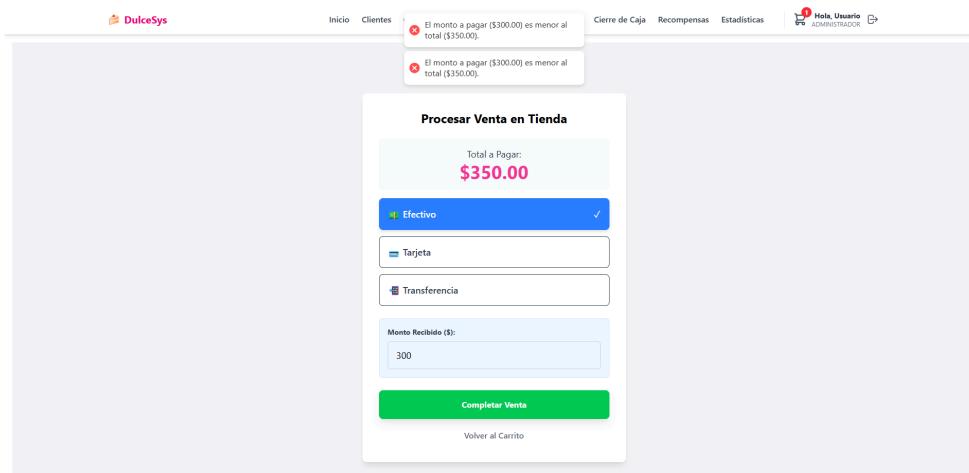
Prueba A2: Se aprecia la pestaña del carrito de compras dentro de administrador, en ella se muestran funcionalidades como pagar y los requeridos para aumentar o disminuir la cantidad de artículos del mismo tipo además se muestra un botón para eliminar el artículo sin importar la cantidad y otro para eliminar todos los artículos que se encuentren en el carrito.



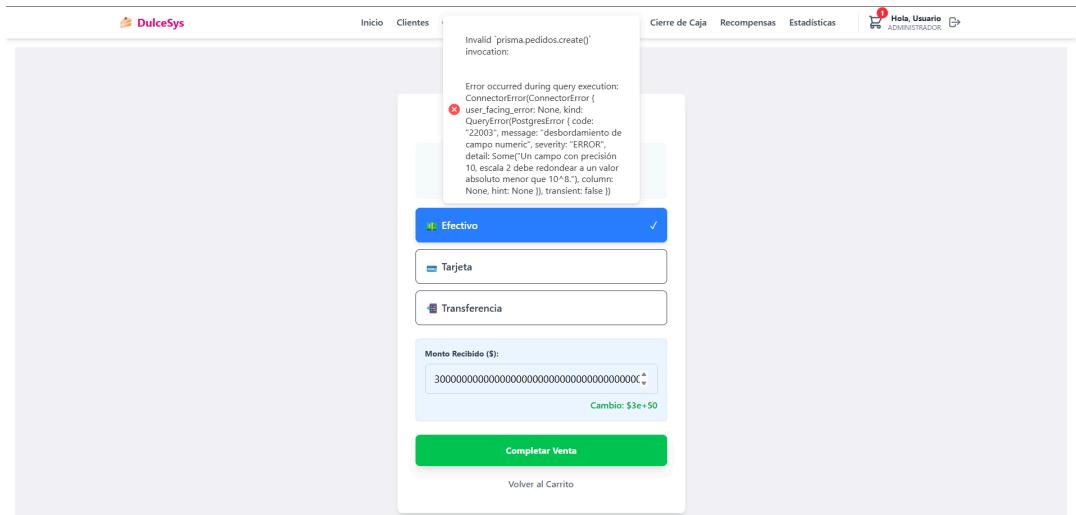
Prueba A3: Aquí se comprueba una validación, la misma refiere a que no se pueden agregar al carrito más productos que los registrados en el inventario.



Prueba A4: Al efectuar la compra e ingresar una cantidad menor, muestra un mensaje de que el monto ingresado es menor al monto a pagar.

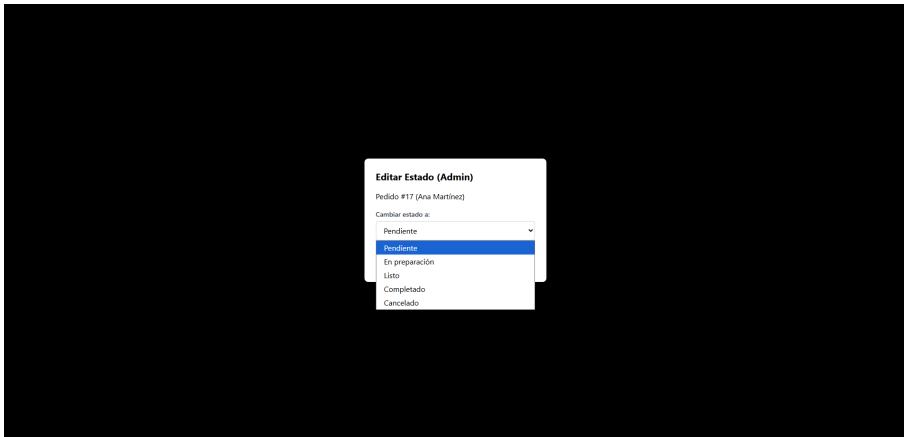


En cambio si se ingresa una cantidad enorme muestra el siguiente error, aquí se recomendó modificar el código para que muestre un mensaje de error más amigable con el usuario.



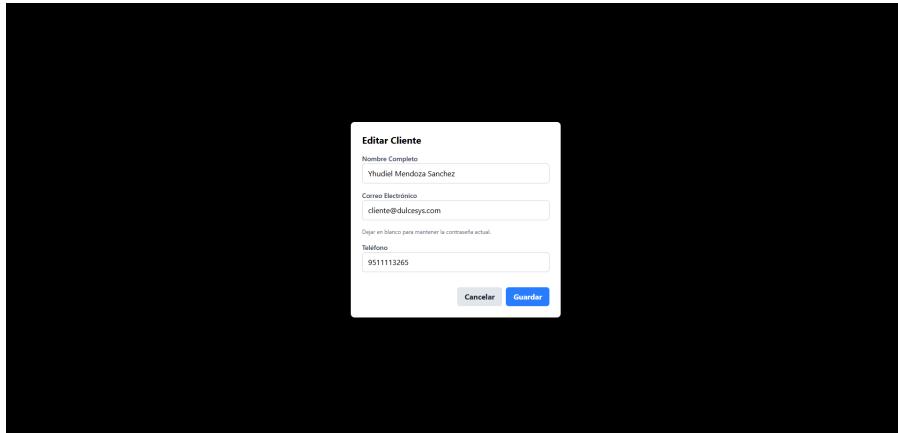
Prueba A5: El administrador puede editar el estado del pedido, desde pendiente, hasta completado, o incluso cancelado, esta pestaña es la de pedidos.

| Gestión de Pedidos | | | | | | | |
|--------------------|-------------------------|------------------------|------------|---------------|-----------|----------------|--|
| Pedidos Activos | | Historial (Terminados) | | | | | |
| ID | CLIENTE | FECHA | HORA | MÉTODO PAGO | TOTAL | ESTADO | ACCIÓN |
| #17 | Ana Martínez | 25/11/2025 | 07:40 p.m. | Tarjeta | \$471.75 | Pendiente | Ver Detalles Procesar Pago Editar Estado |
| #18 | Ana Martínez | 25/11/2025 | 07:41 p.m. | Transferencia | \$297.50 | Pendiente | Ver Detalles Procesar Pago Editar Estado |
| #19 | Ana Martínez | 25/11/2025 | 07:41 p.m. | Transferencia | \$297.50 | Pendiente | Ver Detalles Procesar Pago Editar Estado |
| #20 | Ana Martínez | 25/11/2025 | 07:41 p.m. | Tarjeta | \$297.50 | Pendiente | Ver Detalles Procesar Pago Editar Estado |
| #21 | Ana Martínez | 25/11/2025 | 07:42 p.m. | Tarjeta | \$2380.00 | Pendiente | Ver Detalles Procesar Pago Editar Estado |
| #23 | Ana Martínez | 25/11/2025 | 07:45 p.m. | Tarjeta | \$1062.50 | Pendiente | Ver Detalles Procesar Pago Editar Estado |
| #36 | Ana Martínez | 1/12/2025 | 06:11 p.m. | Efectivo | \$750.00 | Pendiente | Ver Detalles Procesar Pago Editar Estado |
| #40 | Venta en Tienda | 4/12/2025 | 05:43 p.m. | Tarjeta | \$800.00 | En preparación | Ver Detalles Procesar Pago Editar Estado |
| #49 | Yhudiel Mendoza Sanchez | 5/12/2025 | 01:06 p.m. | Efectivo | \$700.00 | Pendiente | Ver Detalles Procesar Pago Editar Estado |
| #52 | Yhudiel Mendoza Sanchez | 7/12/2025 | 03:44 p.m. | Tarjeta | \$2082.50 | Pendiente | Ver Detalles Procesar Pago Editar Estado |

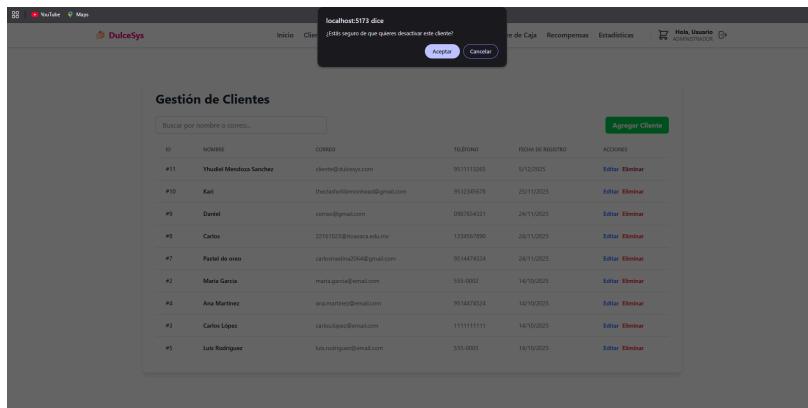


Prueba A6: En la gestión de clientes, el administrador puede editar los datos del cliente o eliminarlo, también se puede agregar o buscar un cliente.

A screenshot of the "Gestión de Clientes" (Client Management) page. At the top, there is a search bar labeled "Buscar por nombre o correo..". Below it is a table with columns: ID, NOMBRE, CORREO, TELÉFONO, FECHA DE REGISTRO, and ACCIONES. The table contains 10 rows of client data. At the bottom right of the table is a green "Agregar Cliente" button.



Si se elimina, manda este mensaje para confirmar si realmente se deseaba eliminar este cliente.



Prueba A7: En la pestaña cajero, sigue la misma lógica y estructura que la de cliente, ya que esta permite editar la información de los cajeros, agregar y eliminarlos.

| ID | NOMBRE COMPLETO | CORREO (USUARIO) | TURNO | ACCIONES |
|----|-----------------|---------------------------|------------|---|
| 13 | Cristiano | 123567@hotmail.com | Vespertino | Editar Eliminar |
| 10 | Patricia Silva | patricia.silva@duleys.com | Vespertino | Editar Eliminar |
| 11 | Elena Morales | elenamorales@duleys.com | Matutino | Editar Eliminar |
| 9 | Roberto Gomez | roberto.gomez@duleys.com | Nocturno | Editar Eliminar |
| 15 | Cajero Sistema | cajero@duleys.com | Matutino | Editar Eliminar |

Prueba A8: al igual que las anteriores esta repite la lógica, lo único que cambia con cada pestaña es en editar y agregar ya que el formulario para llenar es distinto, esto por la diferencia en cuanto a datos necesarios para cada usuario del sistema.

The screenshot shows the DulceSys application interface. At the top, there is a navigation bar with links for Inicio, Clientes, Cajeros, Proveedores, Inventario, Pedidos, Cierre de Caja, Recompensas, Estadísticas, and a user sign-in area. Below the navigation bar is a search bar labeled "Buscar por nombre o RFC...". To the right of the search bar is a green button labeled "Agregar Proveedor".

The main content area displays a table titled "Gestión de Proveedores" (Supplier Management). The table has columns for ID, NOMBRE (Name), CONTACTO (Contact), TELÉFONO (Phone), RFC, and ACCIONES (Actions). The data in the table is as follows:

| ID | NOMBRE | CONTACTO | TELÉFONO | RFC | ACCIONES |
|----|-----------------------------|-------------|------------|---------------|---|
| #3 | Frutas Frescas de la Sierra | Carlos Ruiz | 1234567890 | FFS120101CC30 | Editar Eliminar |
| #2 | Harinas La Oaxaqueña | Maria Lopez | 1122334455 | HAO120101BB21 | Editar Eliminar |
| #1 | Lácteos del Sur | Ernesto Paz | 0123456789 | LDS120101A1A1 | Editar Eliminar |
| #5 | TecNM | Valverde | 9514567896 | VAL2345678987 | Editar Eliminar |

A modal window titled "Agregar Proveedor" (Add Supplier) is displayed in the center. It contains four input fields: "Nombre de la Empresa" (Company Name), "Nombre de Contacto" (Contact Name), "Teléfono" (Phone), and "RFC". At the bottom of the modal are two buttons: "Cancelar" (Cancel) and "Guardar" (Save).

Prueba A9: En la pestaña de inventario, se presentan productos e ingredientes, se integraron diferentes funcionalidades.

The screenshots show the 'Gestión de Inventario' (Inventory Management) section of the DulceSys software. Both screens have a top navigation bar with links: Inicio, Clientes, Cajeros, Proveedores, Inventario, Pedidos, Cierre de Caja, Recompensas, Estadísticas, and a user sign-in area.

Screenshot 1: Gestión de Productos (Product Management)

| Imagen | SKU | NOMBRE | PRECIO | STOCK | ACTIONES |
|--------|------|--------------------------|----------|-------|---------------------------------------|
| | RED1 | Pastel de Redvelvet | \$400.00 | 14 | Receta Producto Editar Eliminar |
| | CHC1 | Cheesecake con mermelada | \$250.99 | 2 | Receta Producto Editar Eliminar |
| | PAV1 | Pastel de vainilla | \$350.00 | 2 | Receta Producto Editar Eliminar |
| | PAG1 | Pastel de chocolate | \$350.00 | 12 | Receta Producto Editar Eliminar |
| | FRT1 | Pastel de frutas | \$300.00 | 12 | Receta Producto Editar Eliminar |
| | PAS1 | Pastel de Puff o Crema | \$300.00 | 8 | Receta Producto Editar Eliminar |

Screenshot 2: Gestión de Ingredientes (Ingredient Management)

| SKU | NOMBRE | STOCK | PRECIO | ACTIONES |
|-------|------------------|------------|--------|-------------------|
| CHD5 | Chocolate Amargo | 12.40 kg | 3.00 | Editar Eliminar |
| UVA1 | Uvas verdes | 200.00 pza | 12.00 | Editar Eliminar |
| HAR01 | Harina | 40.05 kg | 10.00 | Editar Eliminar |
| HUE4 | Huevo | 38.00 pza | 24.00 | Editar Eliminar |
| LEC1 | Lecita | 35.00 lt | 5.00 | Editar Eliminar |
| AZU2 | Azúcar | 63.36 kg | 10.00 | Editar Eliminar |
| FRE8 | Fresa | 55.00 Pza | 2.00 | Editar Eliminar |

Prueba A10: La producción de un nuevo pastel debe coincidir la cantidad de insumos con la cantidad que se desea producir, de lo contrario envía un mensaje con esta leyenda

The dialog box is titled 'Registrar Producción' (Register Production) and specifies 'Producto: Pastel de Redvelvet'. It contains the following information:

- A message at the top: 'Este producto NO tiene una receta configurada. Ve a "Receta" primero.' (This product does not have a recipe configured. Go to "Recipe" first.)
- A note below: 'Al confirmar, se descontarán los ingredientes del inventario automáticamente.' (Upon confirmation, ingredients will be automatically deducted from the inventory.)
- A dropdown menu for 'Tamaño a Producir:' (Production Size) set to 'Pequeño (10 pers) (x1 insumos)'.
- A text input for 'Cantidad a Producir:' (Production Quantity) with the value '20'.
- Buttons at the bottom: 'Cancelar' (Cancel) and 'Confirmar Producción' (Confirm Production).

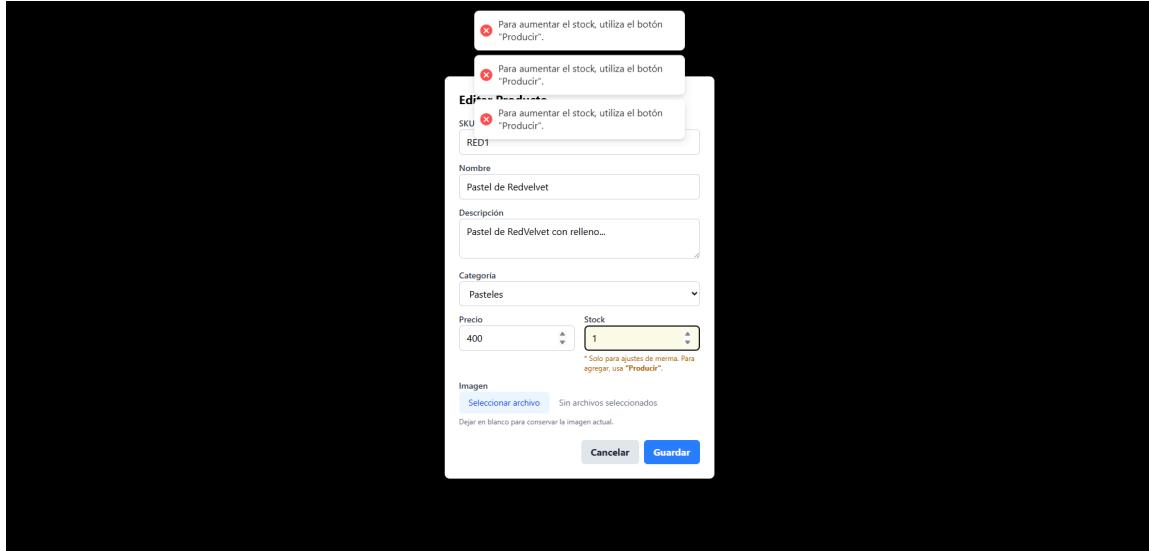
Si cumple con la condición anterior el sistema muestra un mensaje de que el producto fue guardado correctamente

The screenshot shows the 'Gestión de Inventario' (Inventory Management) section of the DulceSys application. At the top, there are tabs for 'Productos' and 'Ingredientes'. A search bar labeled 'Buscar productos...' is present. Below the search bar is a table with columns: IMAGEN, SKU, NOMBRE, PRECIO, STOCK, and ACCIONES. The table lists six products:

| IMAGEN | SKU | NOMBRE | PRECIO | STOCK | ACCIONES |
|--------|------|-------------------------|----------|-------|---------------------------------------|
| | RED1 | Pastel de Redvelvet | \$400.00 | 14 | Receta Producir Editar Eliminar |
| | PASV | Pastel de vainilla | \$350.00 | 2 | Receta Producir Editar Eliminar |
| | PASO | Pastel de oreo | \$350.00 | 12 | Receta Producir Editar Eliminar |
| | FRU1 | Pastel de frutas | \$300.00 | 12 | Receta Producir Editar Eliminar |
| | PASF | Pastel de Fresa y Crema | \$380.00 | 8 | Receta Producir Editar Eliminar |
| | DAC1 | Pastel de Chocolate | \$350.00 | 4 | Receta Producir Editar Eliminar |

A green success message at the top right of the page reads: 'Éxito! Se produjeron 1 Pequeño (10 perj.).'

Prueba A11: Si se desea aumentar el stock se debe realizar desde producir no desde editar el producto, de lo contrario el sistema manda el siguiente mensaje:



Prueba A12: Si se desea eliminar un producto, el sistema pregunta nuevamente para confirmar esta acción.

The screenshot shows a modal dialog box centered over a web application interface. The dialog box has a dark background and contains the text "localhost:5173 dice" at the top, followed by "¿Eliminar producto?". At the bottom of the dialog are two buttons: "Aceptar" (Accept) on the left and "Cancelar" (Cancel) on the right. The background of the application is a "Gestión de Inventario" (Inventory Management) screen. The screen has tabs for "Productos" and "Ingredientes", with "Productos" currently selected. It features a search bar labeled "Buscar productos..." and a green "Agregar Producto" (Add Product) button. Below these are several product entries in a table format, each with a small image, SKU, name, price, stock level, and a row of actions (Receta, Producir, Editar, Eliminar). One row, specifically for "Pastel de Chocolate" (SKU: FRC6), has its stock value (4) displayed in red, indicating it is below the minimum level.

Prueba A13: Cuando el stock está por debajo del mínimo establecido, se marca en rojo para indicar que ya se debe agregar más de dicho ingrediente

This screenshot shows the same "Gestión de Inventario" (Inventory Management) screen as the previous one, but with a different focus. The "Ingredientes" tab is now active. The table lists various ingredients with their respective SKUs, names, stock levels, and minimum levels. The stock levels are color-coded: most are in green, while the stock level for "Fresa" (SKU: FRE6) is explicitly highlighted in red, matching the color of the minimum level value (1.00 Pza) in the same row. Other visible ingredients include Chocolate Amargo, Uvas verdes, Harina, Azucar, Leche, and Huevo.

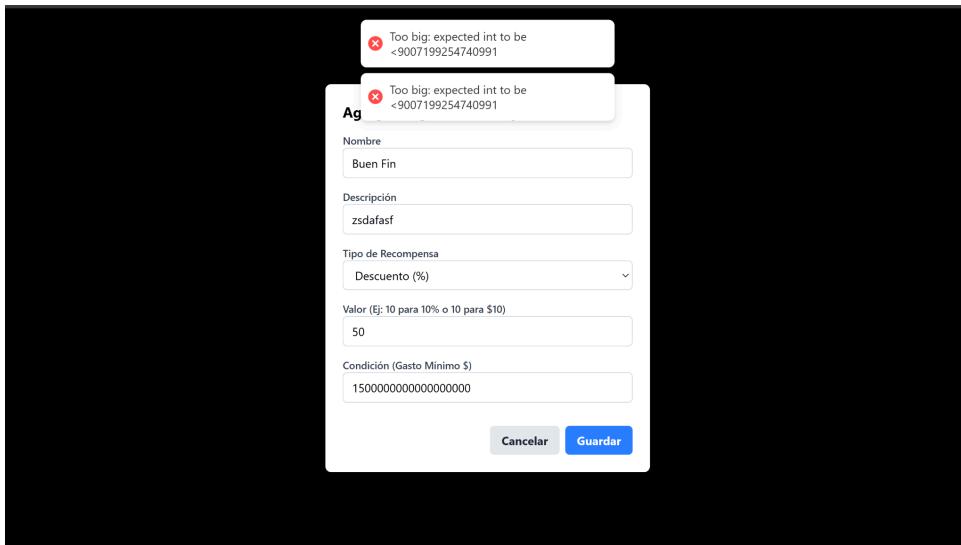
Prueba A14: en la pestaña recompensas tenemos 3 botones uno para eliminar, editar y agregar.

Gestión de Reglas de Recompensas

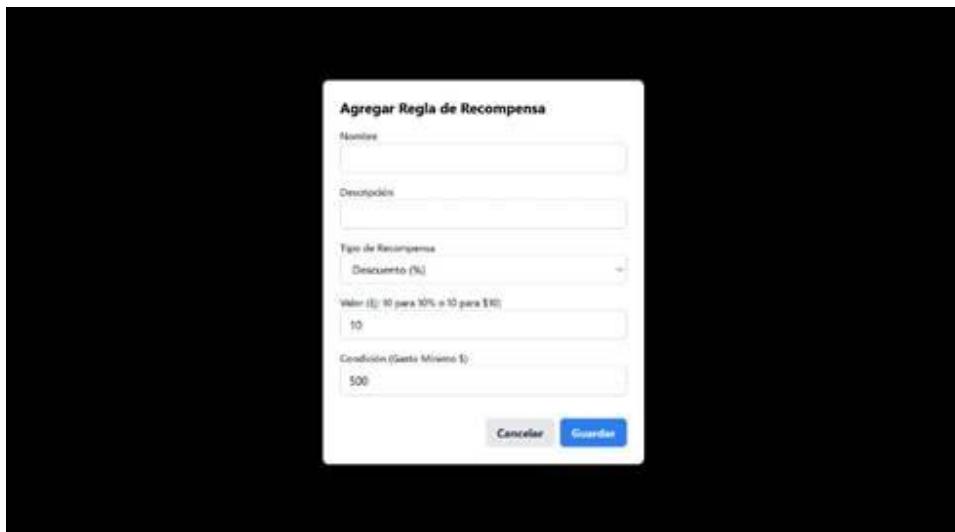
| NOMBRE | TIPO | VALOR | CONDICIÓN (GASTO MÍN) | ACCIONES |
|--------------------|-------------|-------|-----------------------|---|
| Promoción Apertura | Descuento % | 15% | \$200.00 | Editar Eliminar |

Agregar Regla

Prueba A15: Al crear un descuento, y en condición como gasto mínimo se registra una cantidad exagerada muestra el siguiente mensaje:



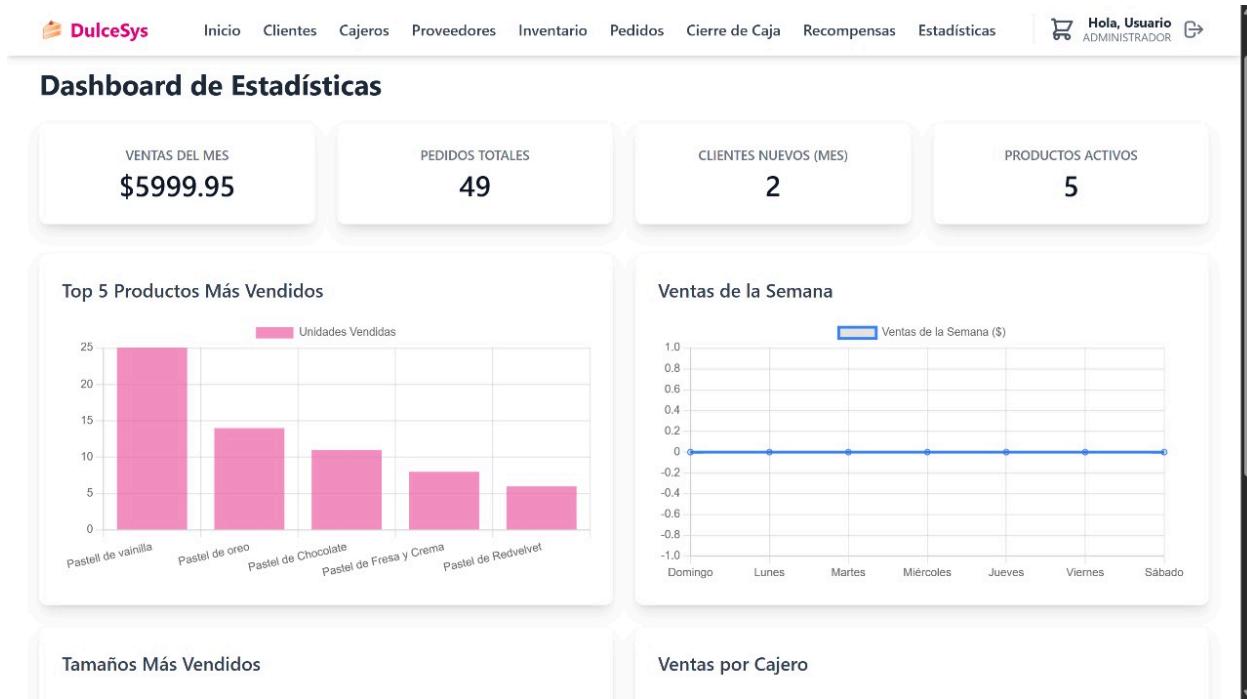
Si asigna una cantidad considerable la recompensa se guarda correctamente



Prueba A16: en esta prueba se detectó un error de lógica, este se da cuando se agrega un descuento fijo de cualquier cantidad, y la condición de gasto es menor al descuento, es decir si ponemos que queremos descontar 50\$ cuando la condición es 1\$.

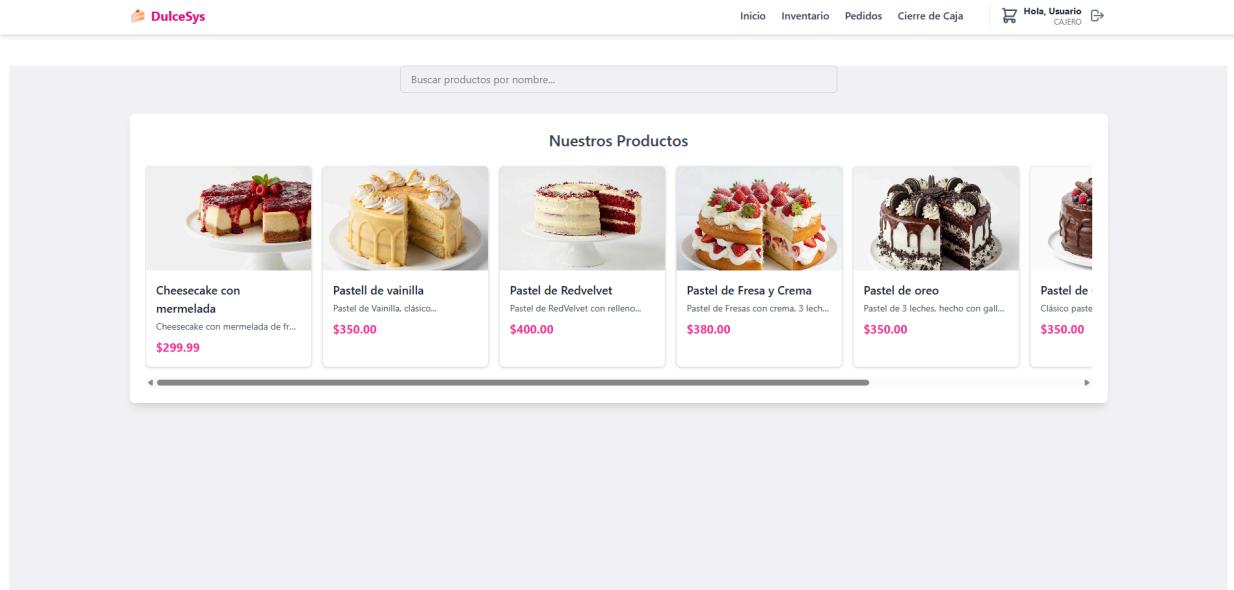
| NOMBRE | TIPO | VALOR | CONDICIÓN (GASTO MÍN) | ACCIONES |
|--------------------|----------------|---------|-----------------------|-----------------|
| Promoción Apertura | Descuento Fijo | \$50.00 | \$1.00 | Editar Eliminar |

Prueba A17: El administrador puede ver las estadísticas de las ventas del mes, como se muestra a continuación:



CAJERO.

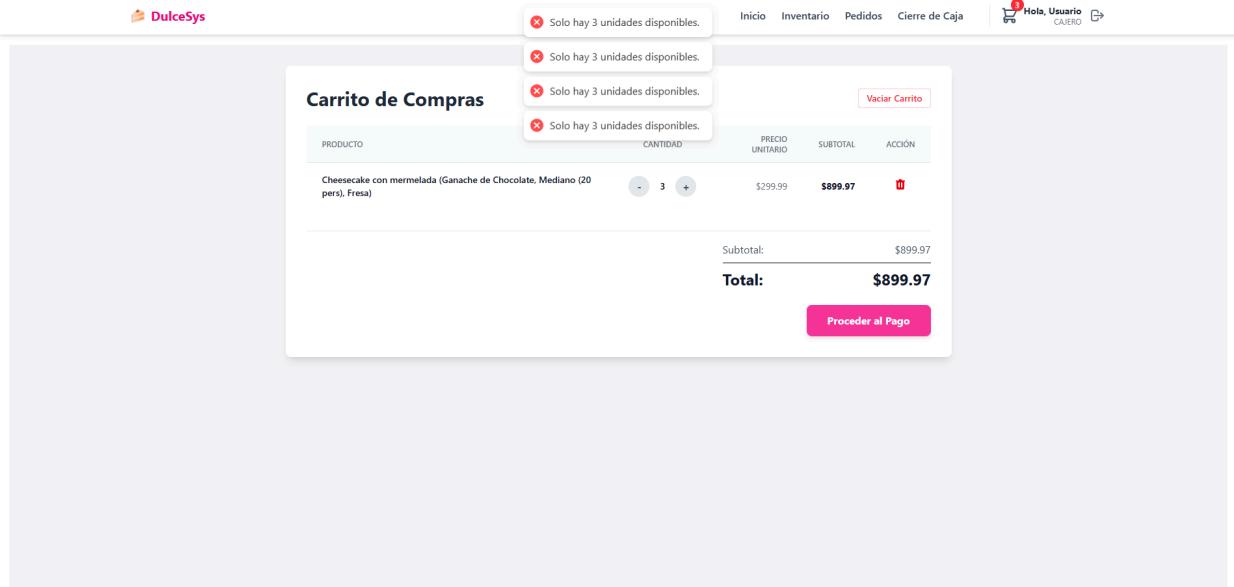
Prueba C1: en esta prueba al igual que en la Prueba A1 se busca hacer notar la diferencia entre las vistas que corresponden a cada rol.



The screenshot shows the 'Nuestros Productos' (Our Products) section of the DulceSys software. It displays six cake products with their names, descriptions, prices, and small images. The products are:

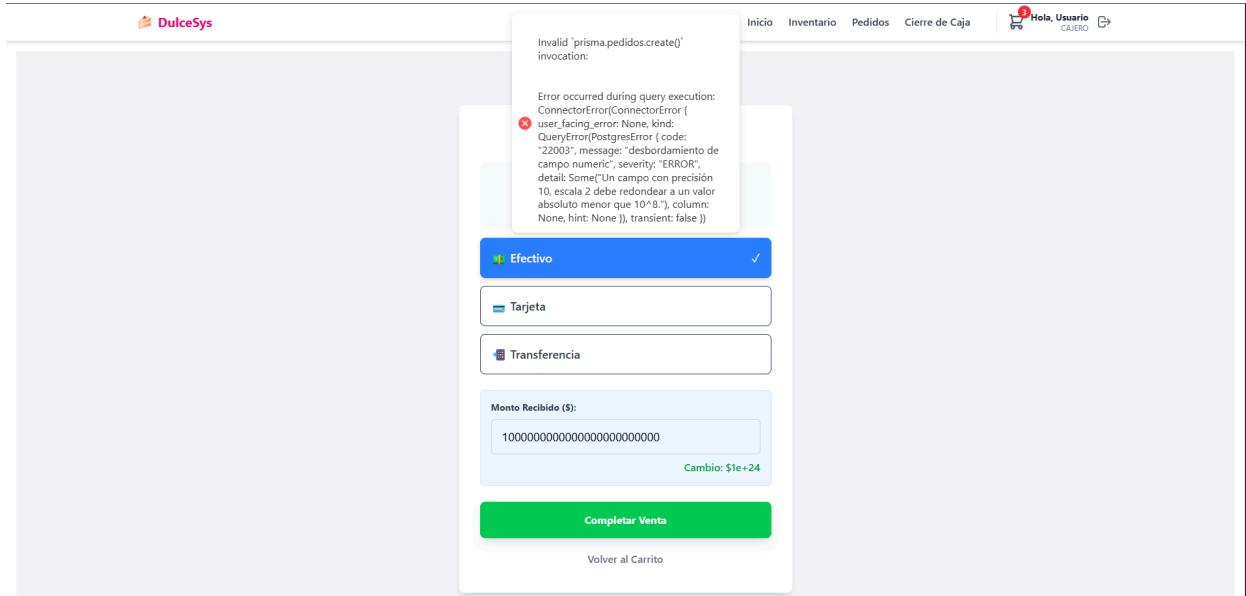
- Cheesecake con mermelada: \$299.99
- Pastell de vainilla: \$350.00
- Pastel de RedVelvet: \$400.00
- Pastel de Fresa y Crema: \$380.00
- Pastel de oreo: \$350.00
- Pastel de Clásico paste: \$350.00

Prueba C2: en este punto al igual que en el rol de administrador las opciones de compra son las mismas.



The screenshot shows the 'Carrito de Compras' (Shopping Cart) section of the DulceSys software. It displays a single item in the cart: 'Cheesecake con mermelada (Ganache de Chocolate, Mediano (20 pers), Fresa)' with a quantity of 3. The total price is \$899.97. There are four error messages above the cart table: 'Solo hay 3 unidades disponibles.' (Only 3 units available.) repeated four times. A 'Vaciar Carrito' (Empty Cart) button is visible. At the bottom, there is a 'Proceder al Pago' (Proceed to Payment) button.

Prueba C3: Se aplicó una prueba parecida a la de el administrador en la que se ingresaba un monto extremadamente alto con una cantidad de dígitos muy grande y nos muestra una correcta validación, pero una alerta no amigable con el sistema y el usuario final.



Prueba C4: al terminar la compra te dirige a la pestaña pedidos, al comprar como usuario y seleccionar cualquier método de pago el periodo pasa a pedidos activos en donde se guarda con un estado “pendiente” para poder completarlo se tiene que procesar el pago.

| ID | CLIENTE | FECHA | HORA | MÉTODO PAGO | TOTAL | ESTADO | ACIÓN |
|-----|-------------------------|------------|------------|---------------|-----------|----------------|--|
| #16 | Ana Martínez | 25/11/2025 | 07:40 p.m. | Efectivo | \$476.00 | Pendiente | Ver Detalles Procesar Pago |
| #17 | Ana Martínez | 25/11/2025 | 07:40 p.m. | Tarjeta | \$471.75 | Pendiente | Ver Detalles Procesar Pago |
| #18 | Ana Martínez | 25/11/2025 | 07:41 p.m. | Transferencia | \$297.50 | Pendiente | Ver Detalles Procesar Pago |
| #19 | Ana Martínez | 25/11/2025 | 07:41 p.m. | Transferencia | \$297.50 | Pendiente | Ver Detalles Procesar Pago |
| #20 | Ana Martínez | 25/11/2025 | 07:41 p.m. | Tarjeta | \$297.50 | Pendiente | Ver Detalles Procesar Pago |
| #21 | Ana Martínez | 25/11/2025 | 07:42 p.m. | Tarjeta | \$2380.00 | Pendiente | Ver Detalles Procesar Pago |
| #23 | Ana Martínez | 25/11/2025 | 07:45 p.m. | Tarjeta | \$1062.50 | Pendiente | Ver Detalles Procesar Pago |
| #36 | Ana Martínez | 1/12/2025 | 06:11 p.m. | Efectivo | \$750.00 | Pendiente | Ver Detalles Procesar Pago |
| #40 | Venta en Tienda | 4/12/2025 | 05:43 p.m. | Tarjeta | \$800.00 | En preparación | Ver Detalles Procesar Pago |
| #49 | Yhudiel Mendoza Sanchez | 5/12/2025 | 01:06 p.m. | Efectivo | \$700.00 | Pendiente | Ver Detalles Procesar Pago |
| #50 | Yhudiel Mendoza Sanchez | 5/12/2025 | 01:06 p.m. | Tarjeta | \$255.00 | Pendiente | Ver Detalles Procesar Pago |



Prueba C5: en las siguientes imágenes se muestran formularios para procesar pagos ya sea tarjeta o transferencia.





| Detalles del Pedido #17 | | | | | |
|-------------------------|---|--------------------------------|-------------------------|-----------------|--|
| Cliente: | Ana Martínez | Fecha: | 25/11/2025 - 07:40 p.m. | | |
| Pago: | Tarjeta | Estado: | Pendiente | | |
| Producto | Detalles | Cant. | P. Unit | Subtotal | |
| Pastel de vainilla | <ul style="list-style-type: none">Relleno: Mermelada de FresaTamaño: Grande (30 porc)Sabor: Pan Chocolate | 1 | \$555.00 | \$555.00 | |
| | | Subtotal: | \$555.00 | | |
| | | Descuento: | -\$83.25 | | |
| | | Total Cobrado: \$471.75 | | | |
| | | Cerrar | | | |

Prueba C6: el cajero tiene acceso a una pestaña llamada inventario en la cual solo puede hacer una interacción visual con los productos e ingredientes

The screenshot shows the 'Gestión de Inventario' (Inventory Management) section of the DulceSys system. At the top, there are tabs for 'Productos' (Products) and 'Ingredientes' (Ingredients), with 'Productos' being the active tab. Below the tabs is a search bar labeled 'Buscar productos...'. The main area displays a table with six rows of product information:

| IMAGEN | SKU | NOMBRE | PRECIO | STOCK |
|--------|-------|-------------------------|----------|-------|
| | PASV | Pastel de vainilla | \$350.00 | 9 |
| | RED1 | Pastel de Redvelvet | \$400.00 | 14 |
| | PASF | Pastel de Fresa y Crema | \$380.00 | 9 |
| | PASO | Pastel de oreo | \$350.00 | 13 |
| | PASC | Pastel de Chocolate | \$350.00 | 6 |
| | FRUIT | Pastel de frutas | \$500.00 | 14 |

Prueba C7: de igual manera tiene una forma de interactuar con los resultados del día, para poder así hacer un corte de caja.

The screenshot shows the 'Corte de Caja' (Day Closing) section of the DulceSys system. At the top, it displays 'Total de Ventas (7 de diciembre de 2025)' which is '\$899.97'. Below this are four colored boxes: green for 'TOTAL VENTAS \$899.97', blue for 'Efectivo \$0.00', purple for 'TARJETA \$0.00', and yellow for 'TRANSFERENCIA \$899.97'. To the right, there is a date field set to '07/12/2025' with a calendar icon. The interface is divided into two main sections: 'Productos Vendidos Hoy' (Products Sold Today) and 'Stock Actual de Ingredientes' (Current Stock of Ingredients).

| PRODUCTO | CANT. | TOTAL |
|--------------------------|-------|----------|
| Cheesecake con mermelada | 3 | \$899.97 |

| INGREDIENTE | STOCK ACTUAL | STOCK MÍNIMO |
|------------------|--------------|--------------|
| Chocolate Amargo | 12.40 kg | 3.00 kg |
| Leche | 31.00 lt | 5.00 lt |
| Huevo | 39.00 pza | 24.00 pza |
| Harina | 40.05 kg | 10.00 kg |
| Fresa | 55.00 Pza | 2.00 Pza |
| Azúcar | 63.36 kg | 10.00 kg |

CLIENTE.

Prueba U1: Al igual que en todas las pruebas iniciales de cada rol, en esta se hace notar la diferencia entre las vistas iniciales dependiendo del rol.

The screenshot shows the homepage of the DulceSys website. At the top, there is a navigation bar with links for 'Inicio' and 'Mis Pedidos'. On the right, there is a user greeting 'Hola, Usuario' followed by a dropdown arrow, and a 'CLIENTE' link. Below the navigation, there is a search bar labeled 'Buscar productos por nombre...'. The main content area is titled 'Nuestros Productos' and features a grid of six cake images. Each card contains the cake's name, a brief description, and its price. A horizontal scrollbar is visible below the cards.

| Pastel de vainilla | Pastel de RedVelvet | Pastel de Fresa y Crema | Pastel de oreo | Pastel de Chocolate | Pastel de frut |
|--------------------------------|------------------------------------|---------------------------------------|---------------------------------------|-----------------------------|----------------|
| Pastel de Vainilla, clásico... | Pastel de RedVelvet con relleno... | Pastel de Fresas con crema. 3 lech... | Pastel de 3 leches, hecho con gall... | Clásico pastel de chocolate | Paste de frut |
| \$350.00 | \$400.00 | \$380.00 | \$350.00 | \$350.00 | \$300.00 |

Prueba U2: la pestaña que muestra el pastel a comprar es la misma en todos los roles.

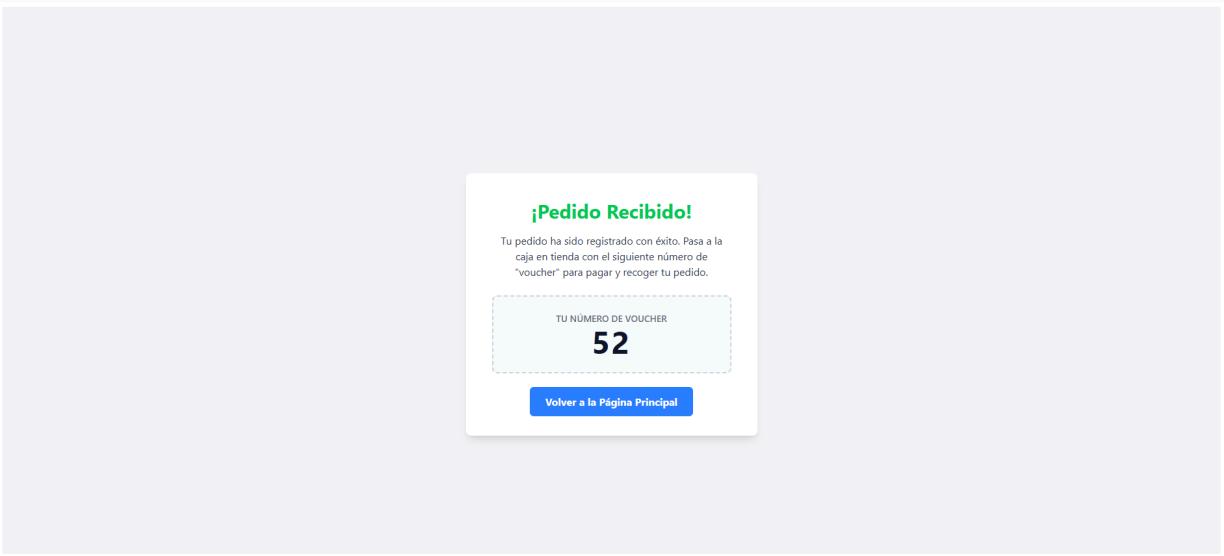
The screenshot shows a detailed product page for a 'Pastell de vainilla'. The page includes a large image of the cake, its name, a brief description, and several dropdown menus for customization. The dropdowns allow selecting the size ('Mediano (20 pers) (+\$80.00)'), flavor ('Fresa (+\$20.00)'), and filling ('Ganache de Chocolate (+\$40.00)'). Below the dropdowns, it says 'Disponibles: 7 pzas.' (Available: 7 pieces). At the bottom, there are two buttons: 'Agregar al Carrito' (Add to Cart) and 'Cancelar' (Cancel).

Prueba U3: Se hizo la misma prueba que en las anteriores y se nota correcto el funcionamiento de las validaciones.

The screenshot shows a shopping cart page from the DulceSys website. At the top, there are three validation error messages: "Solo hay 7 unidades disponibles." (Only 7 units available) appearing three times. The cart table lists a single item: "Pastel de vainilla (Ganache de Chocolate, Mediano (20 pers), Fresa)". The quantity is set to 7, which is highlighted in red. The total price is \$2450.00, and after applying a discount of -\$367.50, the final total is \$2082.50. A pink button at the bottom right says "Proceder al Pago".

Prueba U4: Al comenzar el proceso de pago, sin importar la forma de pago que se elija se salta los formularios y te muestra la pestaña de pedido recibido, el pago y formulario se hacen desde cajero o administrador.

The screenshot shows a payment page titled "Finalizar Pedido". It displays a total amount of \$300.00 and three payment method options: "Efectivo" (Cash), "Tarjeta" (Card), and "Transferencia" (Transfer). Below these options is a large grey button labeled "Finalizar Pedido" (Complete Order). At the bottom of the page, there is a link "Volver al Carrito" (Return to Cart).



Prueba U5: la segunda pestaña en usuario es mis pedidos en la cual la única interacción es visual, ya que no puede hacer modificaciones a sus pedidos una vez ya pagados.

| Mis Pedidos | | | |
|-------------|-----------|------------|-----------|
| ID PEDIDO | FECHA | ESTADO | TOTAL |
| #54 | 7/12/2025 | Pendiente | \$297.50 |
| #53 | 7/12/2025 | Pendiente | \$297.50 |
| #52 | 7/12/2025 | Pendiente | \$2082.50 |
| #50 | 5/12/2025 | Completado | \$255.00 |
| #49 | 5/12/2025 | Pendiente | \$700.00 |

ERRORES Y SOLUCIONES.

Prueba A16: Se encontró un error lógico, el cual se soluciono de la siguiente manera.

Solución: La solución fue implementada desde el backend, específicamente en el archivo de validations.js, en este archivo se hizo una modificación al esquema de recompensas, primero, definiendo que la máxima cantidad de % de descuento sea menor a 100, y mayor a 0.

Por otra parte, el descuento fijo más grande que puede hacerse es de 1,000 pesos, además de que la condición de gasto mínimo, tiene que ser siempre mayor al descuento fijo para evitar caer en errores como que el descuento sea de \$100 y el gasto mínimo sea de \$50.

The screenshot shows a mobile application's edit screen for a reward. The form fields are as follows:

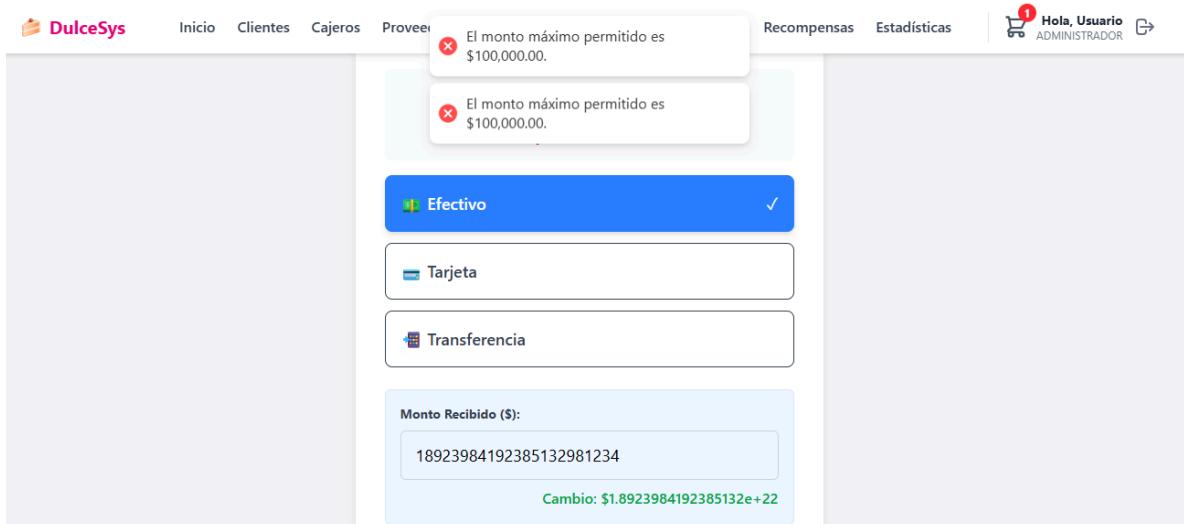
- Nombre: yoas
- Descripción: (empty)
- Tipo de Recompensa: Descuento Fijo (\$)
- Valor (Ej: 10 para 10% o 10 para \$10): 50
- Condición (Gasto Mínimo \$): 20

Two validation errors are displayed:

- El descuento (\$50) no puede ser mayor o igual al gasto mínimo (\$20)
- El descuento (\$50) no puede ser mayor o igual al gasto mínimo (\$20)

At the bottom are two buttons: Cancelar and Guardar.

Prueba A4: El error encontrado es un error de desbordamiento, y aparte visual ya que una advertencia muestra código, se sugiere corregir y modificar por una advertencia más amigable con el sistema y los usuarios.



Solución: La solución se implementó desde el `handleCheckout` que viene en la `CheckoutPage`, ahí se manejó el error poniendo un límite en la cantidad máxima permitida por el sistema, y además se configuró la alerta para que sea mucho más amigable con el usuario y no muestre datos relacionados al sistema.

MANUAL DE USUARIO



MANUAL DE USUARIO

**Sistema DulceSys
Pastelería "La Casa del Pastel"**

2025

| | |
|-------------------------------------|-----------|
| ¡ Bienvenido a DulceSys ! | 3 |
| Requisitos Técnicos Mínimos: | 4 |
| INICIO DE SESIÓN | 5 |
| MÓDULO CLIENTE | 7 |
| MÓDULO CAJERO | 13 |
| MÓDULO ADMINISTRADOR | 20 |
| SOPORTE Y AYUDA | 32 |

¡ Bienvenido a DulceSys !

DulceSys es el sistema de gestión desarrollado específicamente para "La Casa del Pastel", una pastelería comprometida con la calidad, la frescura y la satisfacción de sus clientes en Oaxaca de Juárez.

¿Qué es DulceSys?

DulceSys es una solución digital moderna que transforma la manera en que la pastelería opera diariamente. Este sistema elimina los procesos manuales tradicionales (libretas, llamadas telefónicas, cálculos manuales) y los reemplaza con una plataforma web eficiente, segura y fácil de usar que beneficia tanto a los clientes como al personal de la pastelería.

¿Por qué se creó DulceSys?

Antes de DulceSys, "La Casa del Pastel" enfrentaba varios desafíos:

- Descontrol de inventario: Los ingredientes se agotaban sin previo aviso
- Cotizaciones lentas: Calcular precios de pasteles personalizados tomaba demasiado tiempo
- Pedidos en papel: Información perdida o mal comunicada al equipo de producción
- Sin fidelización: No había forma de premiar a los clientes frecuentes

Ubicación

Dirección: Orquídeas 110, Jardines de la Primavera, 68285 Oaxaca de Juárez, Oax.

Horarios de Atención

 Lunes a Sábado: 8:00 AM - 8:00 PM  Domingo: 9:00 AM - 6:00 PM

Requisitos Técnicos Mínimos:

- Computadora, laptop, tablet.
- Conexión a internet estable
- Navegador web moderno (Chrome, Firefox, Safari, Edge)
- Teclado y mouse (recomendado para administradores)

Requisitos de Usuario:

- Correo electrónico válido (para registro)
- Credenciales de acceso (usuario y contraseña)
- Teléfono de contacto

Seguridad y Privacidad

DulceSys toma muy en serio la seguridad de su información:

- Todas las contraseñas están encriptadas
- Conexiones seguras (HTTPS)
- Cada usuario solo puede ver la información autorizada según su rol
- Registro de auditoría de todas las operaciones importantes
- Respaldos automáticos diarios de la información

INICIO DE SESIÓN

Acceder al Sistema

1. Abre tu navegador web e ingresa a la dirección del sistema
2. Verás la pantalla de inicio de sesión con dos opciones:
 - **Correo electrónico**
 - **Contraseña**
3. Haz clic en "**Iniciar Sesión**"

Iniciar Sesión

Correo Electrónico:

santiagovtania@gmail.com

Contraseña:

.....

Ingresar

¿No tienes una cuenta? [Regístrate aquí](#)

¿No tienes cuenta? Haz clic en "**Regístrate**" para crear una cuenta de cliente.

Registrarse como Cliente

1. En la pantalla de registro, completa:
 - Nombre completo

- Correo electrónico
 - Teléfono
 - Contraseña
2. Haz clic en "**Registrarse**"
 3. Serás redirigido al inicio de sesión para acceder con tus credenciales
 4. En caso de perder tu usuario o contraseña, comunícate con el administrador.
 5. **Nota sobre Implementación Inicial:** Para la fase de prueba/desarrollo, la contraseña inicial establecida fue Password=1. Se requiere actualizar esta contraseña inmediatamente en producción.

El sistema valida lo siguiente al iniciar sesión:

- Credenciales correctas: Al ingresar con cada tipo de usuario, abre el menú correcto y muestra los paneles indicados
- Contraseña incorrecta: Muestra mensaje de credenciales inválidas y no permite acceder
- Usuario inexistente: No permite ingresar al sistema
- Campos vacíos: Es obligatorio ingresar correo y contraseña
- Persistencia de sesión: La sesión se mantiene estable al navegar
- Cierre de sesión: Al cerrar sesión automáticamente muestra opciones de iniciar sesión o registrarse
- Restricciones por URL: Si se accede desde un link en otra pestaña, solicita credenciales nuevamente por seguridad

Crear una Cuenta

Nombre Completo:

Tania Santiago

Teléfono:

9513583708

Correo Electrónico:

santiagovtania@gmail.com

Contraseña:

.....

Registrarse

¿Ya tienes una cuenta? [Inicia sesión](#)

MÓDULO CLIENTE

Pantalla de Inicio

Al iniciar sesión como cliente verás:

Menú superior con opciones:

- Inicio
- Carrito
- Mis Pedidos
- Cerrar Sesión (botón rojo)
- Barra de búsqueda para encontrar productos
- Catálogo de productos con imágenes, nombres, descripciones y precios

The screenshot shows a web browser window with the URL "localhost:5173" in the address bar. The page has a dark header bar with icons for search, refresh, and user profile. Below the header is a navigation bar with links: "DulceSys" (highlighted in blue), "Inicio", "Carrito", "Mis Pedidos", "Hola, Cliente", and a red "Cerrar Sesión" button. A search bar labeled "Buscar productos..." is positioned above a product catalog section. The catalog is titled "Nuestros Productos" and displays four cake products in a grid:

| Imagen | Título | Descripción | Precio |
|--|-------------------------|---------------------------------------|--------|
| A chocolate cake with a slice removed to show the layers. | Pastel de Chocolate | Clásico pastel de chocolate | \$350 |
| A chocolate cake decorated with Oreo cookies and white frosting. | Pastel de oreo | Pastel de 3 leches, hecho con gall... | \$300 |
| A vanilla cake with a slice removed to show the layers. | Pastel de vainilla | Pastel de Vainilla, clásico... | \$350 |
| A two-layer cake filled with strawberries and cream. | Pastel de Fresa y Crema | Pastel de Fresas con crema, 3 lech... | \$380 |

Ver Detalle de Producto

1. Haz clic sobre cualquier producto del catálogo

2. Se abrirá una ventana con:

Imagen del producto en grande

Descripción detallada

Opciones de personalización:

- Tamaño (con variación de precio)
- Sabor del pan
- Relleno
- Stock disponible
- Precio total actualizado según opciones

3. Selecciona las opciones que deseas

4. Haz clic en "Agregar al Carrito"

DulceSys

Inicio Carrito Mis Pedidos Hola, Cliente Cerrar Sesión



Pastel de oreo
Pastel de 3 leches, hecho con galletas oreo...

Tamaño
Pequeño (10 pers) (+\$0.00)

Sabor de Pan
Vainilla (+\$0.00)

Relleno
Crema Batida (+\$0.00)

Disponibles: 17 pzas.

Agregar al Carrito **Cancelar**

Carrito de Compras

1. Haz clic en "Carrito" en el menú superior
2. Verás todos los productos agregados con:
 - Imagen y nombre
 - Cantidad
 - Precio unitario
 - Subtotal
 - Botón para eliminar

3. Acciones disponibles:

- Modificar cantidades
- Eliminar productos
- Ver el Total a Pagar
- Botón "Comprar Carrito" para proceder al pago

Solo permite comprar el carrito completo. Si desea comprar solo un producto, debe eliminar los demás del carrito.

Tu Carrito de Compras

| PRODUCTO | CANTIDAD | PRECIO UNITARIO | SUBTOTAL |
|--|----------|-----------------|----------|
| Pastell de vainilla (Mermelada de Fresa, Pequeño (10 pers), Chocolate) | 1 | \$350.00 | \$350.00 |

Total: \$350.00 [Comprar Carrito](#)

Realizar Pago

Al hacer clic en "Comprar Carrito":

Se abrirá la pantalla de Método de Pago con dos opciones:

Opción A: Pagar en Línea

1. Selecciona "Pago en Línea"
2. Ingresa los datos de tu tarjeta:

- Número de tarjeta
- Fecha de vencimiento (MM/AA)
- CVV (3 dígitos)

3. Haz clic en "Pagar"
4. El sistema validará tu pago

Opción B: Pagar en Cajero

1. Selecciona "Pago en Cajero"
2. Se generará un número de voucher único
3. Anota este número
4. Acude a la tienda física con el voucher
5. Presenta el número al cajero para completar tu pago

6. Tienes 24 horas para pagar, después el pedido se cancela automáticamente

Recibirás una pantalla de confirmación con:

- Número de pedido
- Fecha y hora
- Total pagado
- Estado del pedido
- Método de pago usado
- Número de voucher

Finalizar Pedido

Total a Pagar:
\$350.00

 Efectivo (Pagar en tienda)

 Tarjeta (Pagar en tienda)

 Transferencia (Pagar en tienda)

¿Con cuánto vas a pagar?
350

Finalizar Pedido y Generar Voucher

Cancelar

Mis Pedidos

1. Haz clic en "Mis Pedidos" en el menú
2. Verás el historial completo de tus compras con tarjetas organizadas:
 - Número de pedido
 - Fecha y hora
 - Estado del pedido:
 - Pendiente: Esperando pago
 - En Preparación: Tu pedido se está elaborando
 - Listo: Puedes pasar a recoger
 - Completado: Pedido entregado
 - Cancelado: Pedido cancelado
 - Total pagado
 - Método de pago
 - Productos incluidos con detalles
3. Haz clic en cualquier pedido para ver el seguimiento completo

The screenshot shows a user interface for managing orders. At the top, there is a navigation bar with links for 'Inicio', 'Carrito', 'Mis Pedidos', 'Hola, Cliente', and a red 'Cerrar Sesión' button. Below the navigation is a section titled 'Mis Pedidos' containing a single order card. The card has the following details:

| ID PEDIDO | FECHA | ESTADO | TOTAL |
|-----------|------------|-----------|----------|
| #24 | 28/11/2025 | Pendiente | \$700.00 |

Validaciones del Stock en Compras

El sistema verifica stock disponible en tiempo real

Al finalizar una compra, el stock se actualiza automáticamente

Si un producto se agota, ya no aparecerá disponible para compra

MÓDULO CAJERO

Pantalla de Inicio del Cajero

Al iniciar sesión como cajero verás:

Menú superior:

- Inicio
- Pedidos
- Inventario (solo consulta)
- Corte de Caja
- Cerrar Sesión

Iniciar Sesión

Correo Electrónico:

Contraseña:

Ingresar

¿No tienes una cuenta? [Regístrate aquí](#)

Gestionar Pedidos

1. Haz clic en "Pedidos" en el menú

Hoja, Usuario 

| Gestión de Pedidos | | | | | | | |
|--------------------|-------------------------|------------------------|------------|---------------|-----------|----------------|--|
| Pedidos Activos | | Historial (Terminados) | | | | | |
| ID | CLIENTE | FECHA | HORA | MÉTODO PAGO | TOTAL | ESTADO | ACCIÓN |
| #16 | Ana Martinez | 25/11/2025 | 07:40 p.m. | Efectivo | \$476.00 | Pendiente | Ver Detalles Procesar Pago |
| #17 | Ana Martinez | 25/11/2025 | 07:40 p.m. | Tarjeta | \$471.75 | Pendiente | Ver Detalles Procesar Pago |
| #18 | Ana Martinez | 25/11/2025 | 07:41 p.m. | Transferencia | \$297.50 | Pendiente | Ver Detalles Procesar Pago |
| #19 | Ana Martinez | 25/11/2025 | 07:41 p.m. | Transferencia | \$297.50 | Pendiente | Ver Detalles Procesar Pago |
| #20 | Ana Martinez | 25/11/2025 | 07:41 p.m. | Tarjeta | \$297.50 | Pendiente | Ver Detalles Procesar Pago |
| #21 | Ana Martinez | 25/11/2025 | 07:42 p.m. | Tarjeta | \$2380.00 | Pendiente | Ver Detalles Procesar Pago |
| #23 | Ana Martinez | 25/11/2025 | 07:43 p.m. | Tarjeta | \$1062.50 | Pendiente | Ver Detalles Procesar Pago |
| #36 | Ana Martinez | 1/12/2025 | 06:11 p.m. | Efectivo | \$750.00 | Pendiente | Ver Detalles Procesar Pago |
| #40 | Venta en Tienda | 4/12/2025 | 05:43 p.m. | Tarjeta | \$800.00 | En preparación | Ver Detalles Procesar Pago |
| #49 | Yhudiel Mendoza Sanchez | 5/12/2025 | 01:05 p.m. | Efectivo | \$700.00 | Pendiente | Ver Detalles Procesar Pago |
| #50 | Yhudiel Mendoza Sanchez | 5/12/2025 | 01:06 p.m. | Tarjeta | \$255.00 | Pendiente | Ver Detalles Procesar Pago |

2. Verás dos secciones:

Pedidos Pendientes

Lista de pedidos que esperan pago

Información de cliente, productos y total

Botón "Procesar Pago" en cada pedido

Historial de Pedidos

Pedidos completados del día

Procesar un Pago

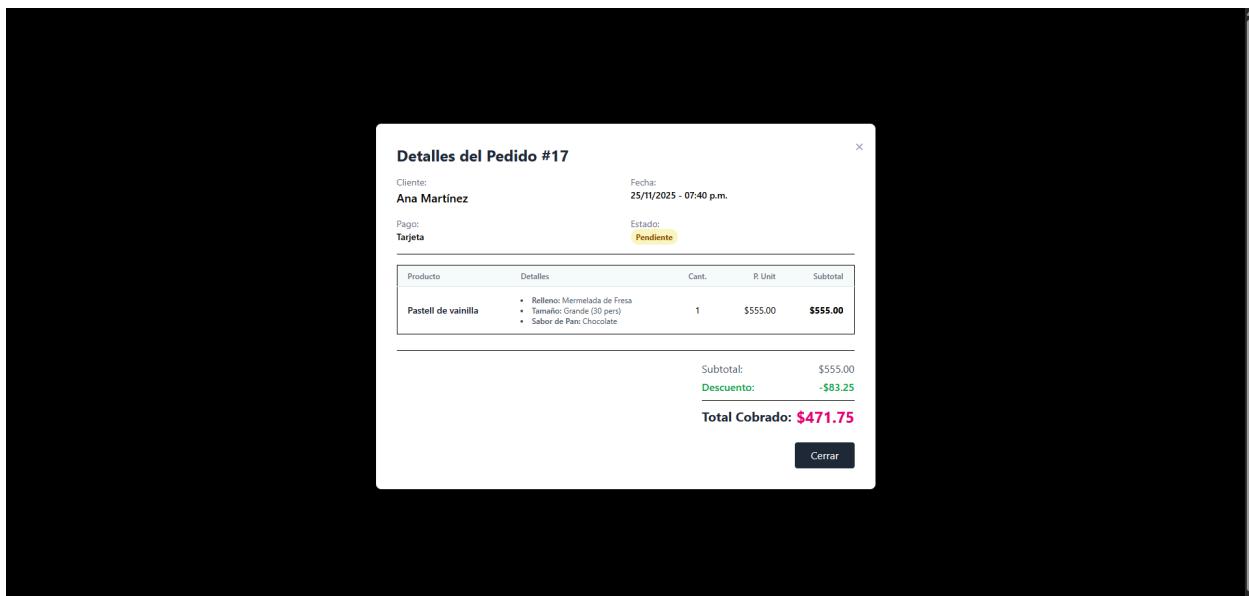
1. Haz clic en "Procesar Pago"

2. Se abrirá un modal con:

- Método de pago (Efectivo, Tarjeta, Transferencia)
- Monto con el que paga el cliente
- Cálculo automático del cambio

3. Haz clic en "Confirmar Pago"

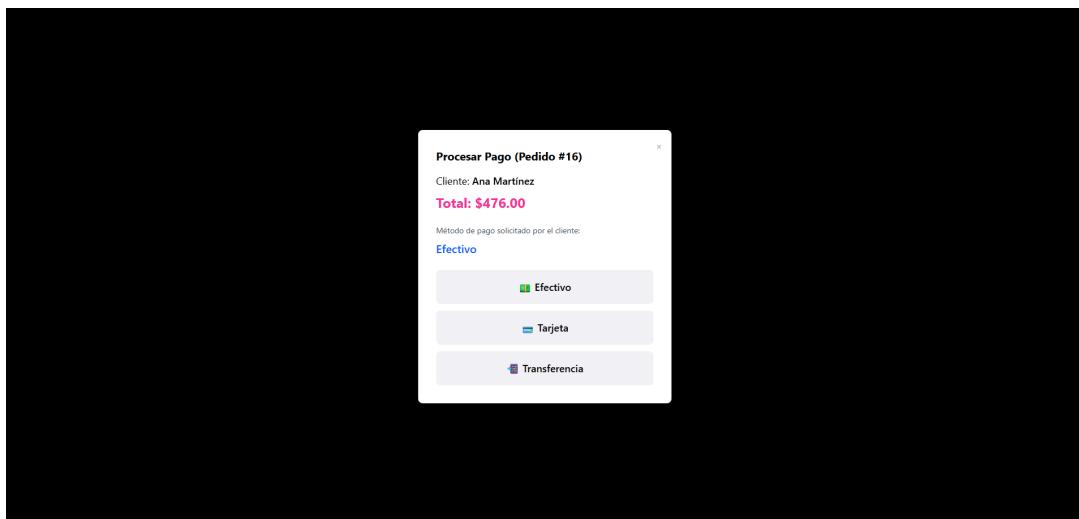
4. El pedido cambia a estado "Completado"



Venta Directa en Tienda

Para atender clientes en mostrador:

1. Desde "Inicio", accede al catálogo de productos
2. Busca y agrega productos al carrito como si fueras un cliente
3. Haz clic en "Comprar Carrito"
4. Selecciona el método de pago del cliente:
 - Efectivo
 - Tarjeta
 - Transferencia
5. Ingresa el monto con el que paga el cliente
6. El sistema calcula automáticamente el cambio
7. Confirma el pago
8. El pedido se marca como "Completado" automáticamente



Consultar Inventory

1. Haz clic en "Inventory"

2. Verás dos pestañas:

- Pestaña Productos
- Lista de productos terminados
- Stock disponible de cada uno
- Precio
- Categoría
- Imagen

Pestaña Ingredientes

- Lista de insumos
- Stock actual
- Stock mínimo
- Unidad de medida
- Proveedor asignado
- Alertas visuales cuando stock < stock mínimo

Sólo puedes consultar, no modificar (sin permisos de edición). Usa esta información para verificar la disponibilidad antes de tomar pedidos.

Gestión de Inventario

[Productos](#) [Ingredientes](#)

Buscar productos...

| IMAGEN | SKU | NOMBRE | PRECIO | STOCK |
|--------|-------|-------------------------|----------|-------|
| | PASV | Pastel de vainilla | \$350.00 | 9 |
| | RED1 | Pastel de Redvelvet | \$400.00 | 14 |
| | PASF | Pastel de Fresa y Crema | \$380.00 | 9 |
| | PASO | Pastel de oreo | \$350.00 | 13 |
| | PASC | Pastel de Chocolate | \$350.00 | 6 |
| | FRIJ1 | Pastel de frutas | \$400.00 | 14 |

Corte de Caja

1. Haz clic en "Corte de Caja"
2. Selecciona la fecha del corte (por defecto: día actual)
3. El sistema mostrará:

Resumen de Ventas

- Total en Efectivo
- Total en Tarjeta
- Total en Transferencia
- Total General del Día

Productos Vendidos

Tabla con productos, cantidades y totales

Balance de Inventario

Stock actual de ingredientes

Alertas de stock bajo

4. Puedes imprimir o exportar el reporte

The screenshot shows the DulceSys software interface with the following details:

Header: DulceSys, Inicio, Inventario, Pedidos, Cierre de Caja, Hola, Usuario CAJERO.

Main Section: Corte de Caja (Report Date: 07/12/2025). It displays the following summary:

| TOTAL VENTAS | EFEKTIVO | TARJETA | TRANSFERENCIA |
|--------------|----------|---------|---------------|
| \$899.97 | \$0.00 | \$0.00 | \$899.97 |

Productos Vendidos Hoy:

| PRODUCTO | CANT. | TOTAL |
|--------------------------|-------|----------|
| Cheesecake con mermelada | 3 | \$899.97 |

Stock Actual de Ingredientes:

| INGREDIENTE | STOCK ACTUAL | STOCK MÍNIMO |
|------------------|--------------|--------------|
| Chocolate Amargo | 12.40 kg | 3.00 kg |
| Leche | 31.00 lt | 5.00 lt |
| Huevo | 39.00 pza | 24.00 pza |
| Harina | 40.05 kg | 10.00 kg |
| Fresa | 55.00 Pza | 2.00 Pza |
| Azúcar | 63.36 kg | 10.00 kg |

Validaciones del Módulo Cajero

El sistema aplica las siguientes validaciones para el rol de cajero:

Validaciones de Pedidos

- Crear pedido válido: El cajero puede crear pedidos desde la página principal
- Modificar estatus: Puede cambiar el estado de cada pedido
- Permisos limitados: No puede filtrar pedidos por fecha, cliente o estado
- Acceso restringido: No puede acceder a pedidos de otros usuarios (solo los que él procesa)

Validaciones de Productos al Vender

- Stock disponible: Valida que haya suficiente stock antes de procesar la venta
- Stock después de venta: El sistema disminuye automáticamente el stock de productos terminados
- Campos obligatorios de pago: No permite procesar pago sin seleccionar método de pago
- Monto válido: Valida que el monto ingresado sea correcto y mayor o igual al total

Validaciones de Inventario (Consulta)

- Solo lectura: El cajero solo puede visualizar el inventario, no modificarlo
- Alertas visibles: Puede ver alertas de stock bajo pero no puede actualizar cantidades
- Información en tiempo real: Visualiza stock actualizado para verificar disponibilidad

Validaciones de Corte de Caja

- Fecha válida: Solo puede generar corte de caja de fechas válidas
- Cálculos automáticos: El sistema calcula automáticamente totales por método de pago
- Integridad de datos: No puede modificar manualmente los totales del corte

MÓDULO ADMINISTRADOR

Gestionar Productos

Crear Producto

1. Ve a "Inventario" - Pestaña "Productos"

2. Haz clic en "Aregar Producto"

3. Completa el formulario:

- Nombre del producto
- Descripción
- Precio
- Categoría
- SKU (código único)
- Stock inicial
- Imagen del producto (obligatorio)

4. Haz clic en "Guardar"

The screenshot shows the 'Gestión de Inventario' (Inventory Management) section. At the top, there are two tabs: 'Productos' (Products) and 'Ingredientes' (Ingredients). Below the tabs is a search bar labeled 'Buscar productos...'. A green button labeled 'Aregar Producto' is located on the right side of the header. The main area displays a table with four rows of data. Each row contains an image, a SKU, a name, a price, a stock level, and a status column with buttons for 'Actualizar', 'Editar', and 'Eliminar'.

| IMAGEN | SKU | NOMBRE | PRECIO | STOCK | ACCIONES |
|--------|------|-------------------------|---------|-------|--|
| | R12E | Pastel de Nata | \$40.00 | 14 | Actualizar Editar Eliminar |
| | P12F | Pastel de Fresa y Crema | \$30.00 | 8 | Actualizar Editar Eliminar |
| | P12O | Pastel de Oreo | \$35.00 | 13 | Actualizar Editar Eliminar |
| | P12C | Pastel de Chocolate | \$30.00 | 11 | Actualizar Editar Eliminar |

The screenshot shows the 'Gestión de Inventario' (Inventory Management) section. At the top, there are two tabs: 'Productos' (Products) and 'Ingredientes' (Ingredients). Below the tabs is a search bar labeled 'Buscar ingredientes...'. A green button labeled 'Aregar Ingrediente' is located on the right side of the header. The main area displays a table with eight rows of data. Each row contains an image, a SKU, a name, a price, a stock level, and a status column with buttons for 'Actualizar', 'Editar', and 'Eliminar'. The last row, 'LECH', has a red background.

| IMAGEN | SKU | NOMBRE | PRECIO | STOCK | ACCIONES |
|--------|-------|------------------|-------------|--------|--|
| | UVA1 | Uvas verdes | \$20.00 pza | 100.00 | Actualizar Editar Eliminar |
| | FRES1 | Fresa | \$15.00 Pza | 100 | Actualizar Editar Eliminar |
| | HUSA1 | Harina | \$10.00 pza | 14.00 | Actualizar Editar Eliminar |
| | CHOC1 | Chocolate Amargo | \$15.00 kg | 3.00 | Actualizar Editar Eliminar |
| | HAR01 | Harina | \$10.00 kg | 10.00 | Actualizar Editar Eliminar |
| | AZU1 | Azúcar | \$12.00 kg | 10.00 | Actualizar Editar Eliminar |
| | LECH1 | Leche | \$20.00 l | 5.00 | Actualizar Editar Eliminar |

Validaciones del CRUD de Productos

Datos correctos: Todos los campos obligatorios deben completarse

Imagen obligatoria: No permite guardar sin imagen del producto

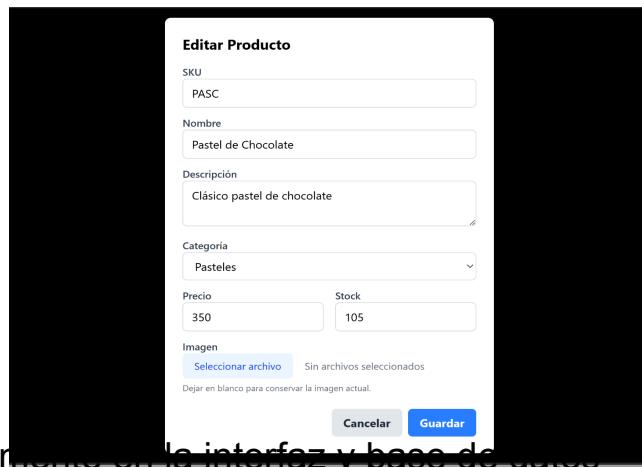
Campos vacíos: Idealmente no debería permitir guardar con campos vacíos

Precios negativos: No permite crear productos con precios negativos o cero

Productos duplicados: No permite crear dos o más productos con el mismo identificador (SKU), pero sí con el mismo nombre e imagen

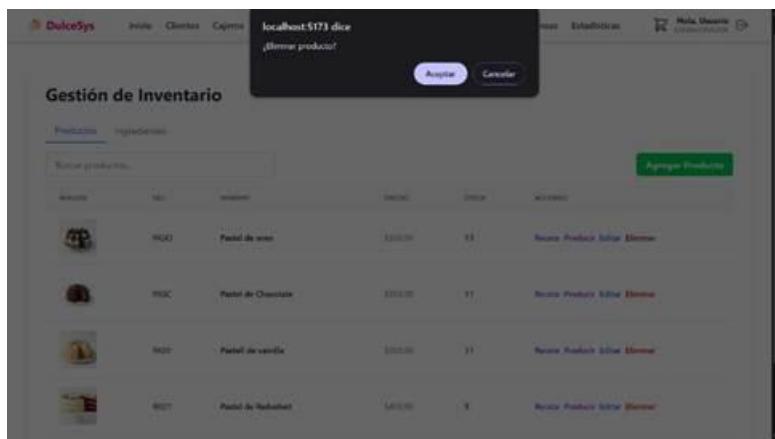
Editar Producto

1. Localiza el producto en la lista
2. Haz clic en el ícono de edición
3. Modifica los campos necesarios
4. Haz clic en "Guardar"
5. Los cambios se reflejan inmediatamente en la interfaz y base de datos



Eliminar Producto

1. Haz clic en el ícono de eliminar
2. Confirma la acción
3. El producto se elimina de la base de datos



Gestionar Ingredientes

Crear Ingrediente

1. Ve a "Inventario" - Pestaña "Ingredientes"

2. Haz clic en "Agregar Ingrediente"

3. Completa el formulario:

- Nombre del ingrediente
- Stock actual
- Stock mínimo
- Unidad de medida (kg, L, piezas, etc.)
- Proveedor

4. Haz clic en "Guardar"

Editar Ingrediente

1. Localiza el ingrediente en la lista

2. Haz clic en el ícono de edición

3. Modifica los campos necesarios

4. Haz clic en "Guardar"

Gestionar Recetas de Productos

El sistema de recetas permite registrar los ingredientes necesarios para elaborar cada producto, facilitando el control de inventario y la planificación de producción.

Acceder a las Recetas

1. Haz clic en "Inventario" en el menú principal

2. Selecciona la pestaña "Productos"

3. Localiza el producto al que deseas agregar o modificar la receta

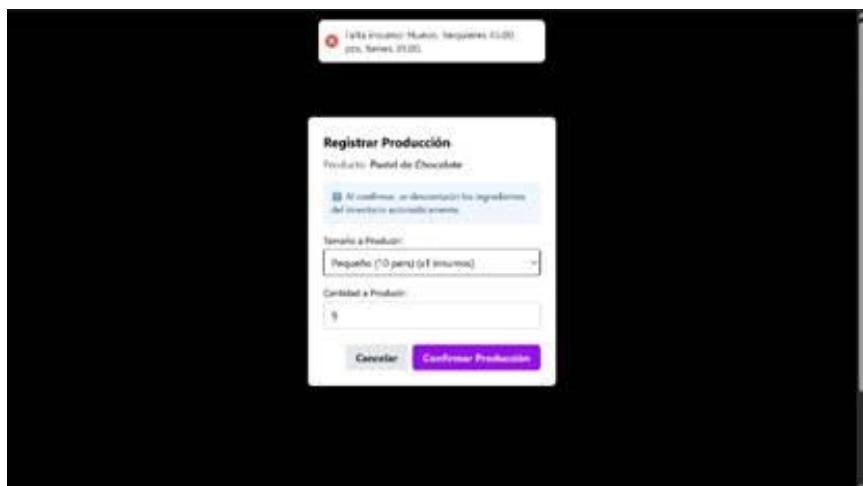
4. Haz clic en el botón "Ver Receta"

Crear una Nueva Receta

Cuando un producto no tiene receta registrada:

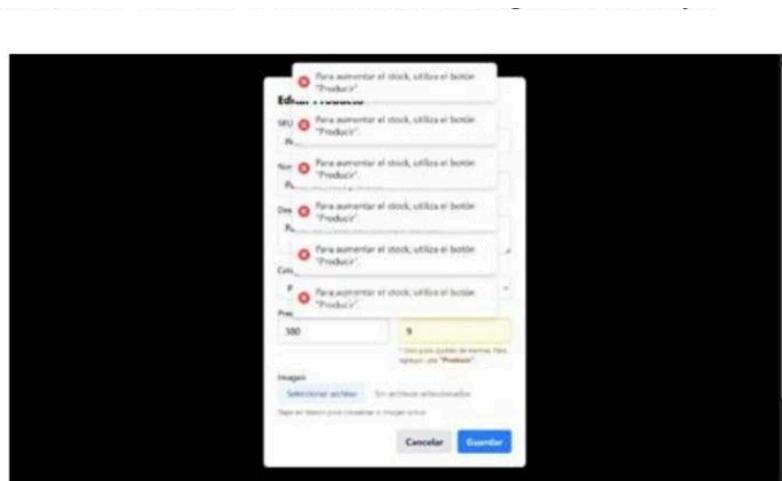
1. Haz clic en "Ver Receta"

2. Se abrirá un modal con el título "Receta de [Nombre del Producto]"
3. Verás el mensaje: "Este producto aún no tiene una receta registrada"
4. Haz clic en el botón "Crear Receta"
5. Completa el formulario:



Modificar una Receta Existente

1. Accede a "Ver Receta" del producto
2. Se mostrará la lista completa de ingredientes con sus cantidades



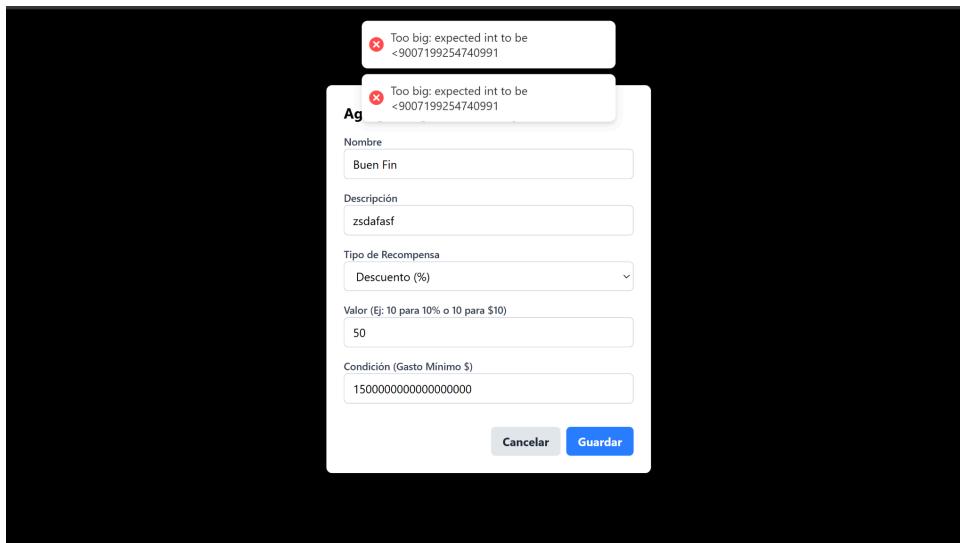
En la pestaña recompensas tenemos 3 botones uno para eliminar, editar y agregar.

Gestión de Reglas de Recompensas

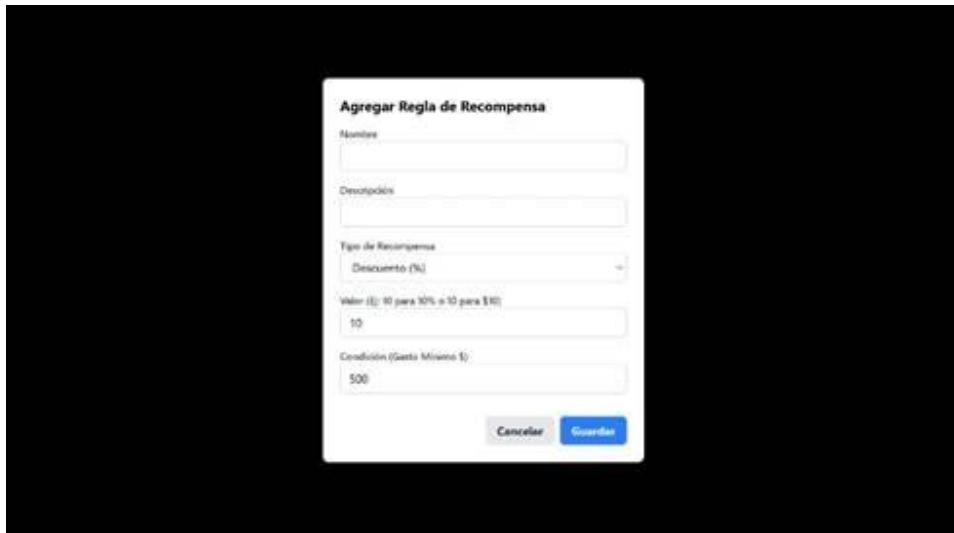
| NOMBRE | TIPO | VALOR | CONDICIÓN (GASTO MÍN) | ACCIONES |
|--------------------|-------------|-------|-----------------------|---|
| Promoción Apertura | Descuento % | 15% | \$200.00 | Editar Eliminar |

Agregar Regla

Al crear un descuento, y en condición como gasto mínimo se registra una cantidad exagerada muestra el siguiente mensaje:



Si asigna una cantidad considerable la recompensa se guarda correctamente



Gestión de pedidos

El administrador puede editar el estado del pedido, desde pendiente, hasta completado, o incluso cancelado, esta pestaña es la de pedidos.

The screenshot shows the DulceSys software interface. At the top, there is a navigation bar with links: Inicio, Clientes, Cajeros, Proveedores, inventario, Pedidos, Cierre de Caja, Recompensas, Estadísticas, and a user sign-in area. Below the navigation bar, the main content area has a title "Gestión de Pedidos". Underneath it, there are two tabs: "Pedidos Activos" (which is selected) and "Historial (Terminados)". A search bar is present below the tabs. The main table displays the following data:

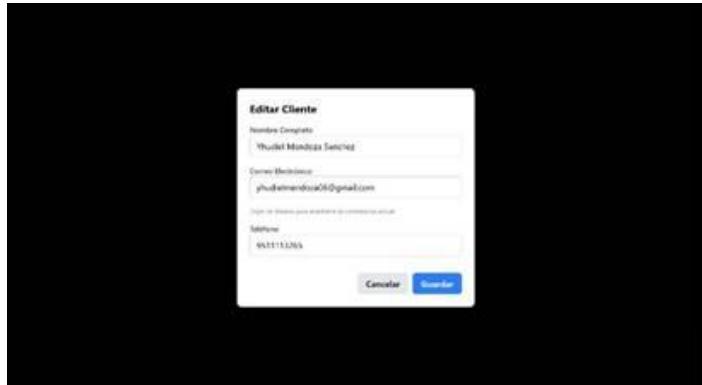
| ID | CLIENTE | FECHA | HORA | MÉTODO PAGO | TOTAL | ESTADO | ACCIÓN |
|-----|-------------------------|------------|------------|---------------|-----------|----------------|--|
| #17 | Ana Martínez | 25/11/2025 | 07:40 p.m. | Tarjeta | \$471.75 | Pendiente | Ver Detalles Procesar Pago Editar Estado |
| #18 | Ana Martínez | 25/11/2025 | 07:41 p.m. | Transferencia | \$297.50 | Pendiente | Ver Detalles Procesar Pago Editar Estado |
| #19 | Ana Martínez | 25/11/2025 | 07:41 p.m. | Transferencia | \$297.50 | Pendiente | Ver Detalles Procesar Pago Editar Estado |
| #20 | Ana Martínez | 25/11/2025 | 07:41 p.m. | Tarjeta | \$297.50 | Pendiente | Ver Detalles Procesar Pago Editar Estado |
| #21 | Ana Martínez | 25/11/2025 | 07:42 p.m. | Tarjeta | \$2380.00 | Pendiente | Ver Detalles Procesar Pago Editar Estado |
| #23 | Ana Martínez | 25/11/2025 | 07:45 p.m. | Tarjeta | \$1062.50 | Pendiente | Ver Detalles Procesar Pago Editar Estado |
| #36 | Ana Martínez | 1/12/2025 | 06:11 p.m. | Efectivo | \$750.00 | Pendiente | Ver Detalles Procesar Pago Editar Estado |
| #40 | Venta en Tienda | 4/12/2025 | 05:43 p.m. | Tarjeta | \$800.00 | En preparación | Ver Detalles Procesar Pago Editar Estado |
| #49 | Yhudiel Mendoza Sanchez | 5/12/2025 | 01:06 p.m. | Efectivo | \$700.00 | Pendiente | Ver Detalles Procesar Pago Editar Estado |
| #52 | Yhudiel Mendoza Sanchez | 7/12/2025 | 03:44 p.m. | Tarjeta | \$2082.50 | Pendiente | Ver Detalles Procesar Pago Editar Estado |

A modal window titled "Editar Estado (Admin)" is displayed over the table, showing a dropdown menu for changing the status of Order #17 (Ana Martínez). The dropdown menu includes options: Pendiente (selected), En preparación, Listo, Completado, and Cancelado.

Gestión de clientes

En la gestión de clientes, el administrador puede editar los datos del cliente o eliminarlo, también se puede agregar o buscar un cliente.

| ID | Nombre | Correo | Teléfono | Fecha de Registro | Estado |
|-----|-------------------------|---------------------------------|------------|-------------------|--------|
| #11 | Xhuliet Montaña Sanchez | xhulietmontañasanchez@gmail.com | 9511111110 | 20/11/2020 | Activo |
| #10 | Karl | karlcarlosfernandez@gmail.com | 9511111108 | 20/11/2020 | Activo |
| #9 | Daniell | danieldiaz@hotmail.com | 9501111107 | 28/11/2020 | Activo |
| #8 | Carmel | CTV102318@outlook.es | 7755555555 | 28/11/2020 | Activo |
| #7 | Pascual Gomez | pascualgomez2001@gmail.com | 9511111104 | 28/11/2020 | Activo |
| #6 | Maria Garcia | mariagarcia12345@gmail.com | 666-9999 | 14/11/2020 | Activo |
| #5 | Ana Martinez | ana.martinez123@gmail.com | 9511111103 | 14/11/2020 | Activo |
| #4 | Carlos Lopez | carloslopez001@gmail.com | 7777777777 | 14/11/2020 | Activo |



Si se elimina, manda este mensaje para confirmar si realmente se deseaba eliminar este cliente.

| ID | Nombre | Correo | Teléfono | Acciones |
|----|-----------------------|-------------------------------------|-----------|---|
| 01 | Miguel Moreno Sanchez | miguel.moreno.sanchez@correo.ugr.es | 954492213 | Editar Eliminar |
| 02 | Edu | edu.eduardo@correo.ugr.es | 954492212 | Editar Eliminar |
| 03 | David | david.david@correo.ugr.es | 954492211 | Editar Eliminar |
| 04 | Celia | celia.celia@correo.ugr.es | 954492210 | Editar Eliminar |
| 05 | Pilar de zea | pilar.pilar@correo.ugr.es | 954492209 | Editar Eliminar |
| 06 | Maria Garcia | maria.maria@correo.ugr.es | 954492208 | Editar Eliminar |
| 07 | Alex Martinez | alex.martinez@correo.ugr.es | 954492207 | Editar Eliminar |
| 08 | Carmen Lopez | carmen.carmen@correo.ugr.es | 954492206 | Editar Eliminar |

Gestión de cajeros

En la pestaña cajero, sigue la misma lógica y estructura que la de cliente, ya que esta permite editar la información de los cajeros, agregar y eliminarlos.

| ID | Nombre Completo | Correo Electrónico | Teléfono | Acciones |
|----|-----------------|------------------------------|-----------|---|
| 01 | Diana Mendoza | diana.mendoza@correo.ugr.es | 954492205 | Editar Eliminar |
| 02 | Patricia Sáez | patricia.saenz@correo.ugr.es | 954492204 | Editar Eliminar |
| 03 | Ricardo Gómez | ricardo.gomez@correo.ugr.es | 954492203 | Editar Eliminar |
| 04 | Diego | diego.diego@correo.ugr.es | 954492202 | Editar Eliminar |
| 05 | Cajero Sistema | cajero.sistema@correo.ugr.es | 954492201 | Editar Eliminar |

Agregar Cajero

Nombre (Completo):

Correo Electrónico (Usuario):

Contraseña:

Teléfono: Maturino

[Cancelar](#) [Guardar](#)

al igual que las anteriores esta repite la lógica, lo único que cambia con cada pestaña es en editar y agregar ya que el formulario para llenar es distinto, esto por la diferencia en cuanto a datos necesarios para cada usuario del sistema.

- **Gestión de proveedores**

| ID | Nombre | Domicilio | RFC | Acciones |
|----|-----------------------------|----------------|-----------------|---|
| A1 | Pistas Frescas de la Sierra | Calle 60 #100 | F102019102201 | Editar Borrar |
| A2 | Hojas La Gaviota | Mexico 123 | HAB000000000001 | Editar Borrar |
| A3 | Lácteos del Sur | Av. Mexico 123 | LSD000000000001 | Editar Borrar |
| A4 | SCHM | Vallarta | SCH000000000001 | Editar Borrar |

Nombre de la Empresa:

Nombre de Contacto:

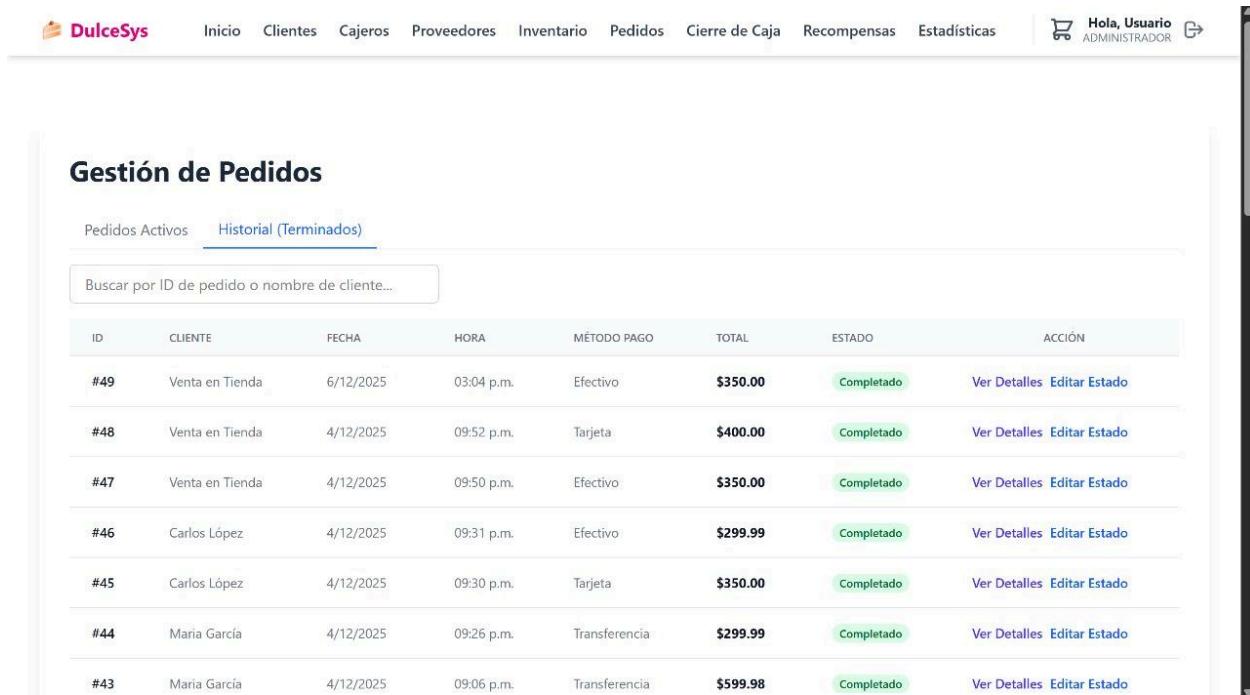
Teléfono:

RFC:

[Cancelar](#) [Guardar](#)

Historial de pedidos realizados

Como podemos observar, a lado de pedidos activos, logramos visualizar la sección de historial de pedidos



The screenshot shows the DulceSys software interface with the following details:

- Header:** Includes the logo "DulceSys", navigation links (Inicio, Clientes, Cajeros, Proveedores, Inventario, Pedidos, Cierre de Caja, Recompensas, Estadísticas), and a user session "Hola, Usuario ADMINISTRADOR".
- Title:** "Gestión de Pedidos" (Management of Orders).
- Section:** "Historial (Terminados)" (History (Completed)).
- Search Bar:** "Buscar por ID de pedido o nombre de cliente...".
- Table:** A grid displaying completed orders with the following columns: ID, CLIENTE, FECHA, HORA, MÉTODO PAGO, TOTAL, ESTADO, and ACCIÓN (Actions). The table contains 8 rows of data.

| ID | CLIENTE | FECHA | HORA | MÉTODO PAGO | TOTAL | ESTADO | ACCIÓN |
|-----|-----------------|-----------|------------|---------------|----------|------------|--|
| #49 | Venta en Tienda | 6/12/2025 | 03:04 p.m. | Efectivo | \$350.00 | Completado | Ver Detalles Editar Estado |
| #48 | Venta en Tienda | 4/12/2025 | 09:52 p.m. | Tarjeta | \$400.00 | Completado | Ver Detalles Editar Estado |
| #47 | Venta en Tienda | 4/12/2025 | 09:50 p.m. | Efectivo | \$350.00 | Completado | Ver Detalles Editar Estado |
| #46 | Carlos López | 4/12/2025 | 09:31 p.m. | Efectivo | \$299.99 | Completado | Ver Detalles Editar Estado |
| #45 | Carlos López | 4/12/2025 | 09:30 p.m. | Tarjeta | \$350.00 | Completado | Ver Detalles Editar Estado |
| #44 | Maria García | 4/12/2025 | 09:26 p.m. | Transferencia | \$299.99 | Completado | Ver Detalles Editar Estado |
| #43 | Maria García | 4/12/2025 | 09:06 p.m. | Transferencia | \$599.98 | Completado | Ver Detalles Editar Estado |

Estado del pedido

El administrador podrá editar esta sección

- Estado del pedido:
- Pendiente: Esperando pago
- En Preparación: Tu pedido se está elaborando
- Listo: Puedes pasar a recoger
- Completado: Pedido entregado
- Cancelado: Pedido cancelado

Información Importante sobre Recetas

Cada receta debe contener al menos un ingrediente para poder guardarse

Las cantidades deben ser positivas y mayores a cero

El sistema valida automáticamente que haya stock suficiente de ingredientes antes de permitir producción

Las recetas son fundamentales para el sistema de producción y control de inventario

Sistema de Producción y Actualización de Stock

DulceSys incorpora un sistema inteligente de producción que vincula las recetas con el inventario de ingredientes y productos terminados.

Reglas del Sistema de Producción

Productos sin receta no pueden producirse: primero debe crearse la receta

El sistema valida stock en tiempo real: no es posible producir más unidades de las que los ingredientes permiten

Trazabilidad completa: cada producción queda registrada con fecha, hora y usuario responsable

Alertas preventivas: si algún ingrediente alcanza el stock mínimo durante la producción, el sistema lo indica

Actualización Manual de Stock de Ingredientes

Para registrar la entrada de nuevos ingredientes:

1. Ve a "Inventario" - Pestaña "Ingredientes"
2. Localiza el ingrediente que deseas actualizar
3. Haz clic en el ícono de edición
4. Modifica el campo "Stock actual" con la nueva cantidad
5. Haz clic en "Guardar"

Recomendaciones:

- Actualiza el stock inmediatamente al recibir mercancía
- Verifica que la unidad de medida coincida con la del proveedor
- Mantén actualizado el stock mínimo para recibir alertas oportunas
- Revisa las alertas de stock bajo antes de planificar producción

Flujo Completo de Operaciones

El sistema de inventario y producción funciona de la siguiente manera:

Paso 1 Recepción de ingredientes: Cuando llega mercancía del proveedor, el administrador actualiza el stock de ingredientes en el sistema ingresando las cantidades recibidas.

Paso 2 - Planificación de producción: El administrador revisa qué productos necesitan elaborarse y verifica que existan recetas registradas para cada uno de ellos.

Paso 3 - Ejecución de producción: Al iniciar la producción, el sistema valida automáticamente que haya stock suficiente de todos los ingredientes necesarios según la receta.

Paso 4 - Confirmación y actualización: Una vez confirmada la producción, el sistema descuenta los ingredientes utilizados del inventario y aumenta el stock de productos terminados.

Paso 5 - Venta de productos: Cuando un cliente compra un producto, el sistema disminuye automáticamente el stock de productos terminados.

Paso 6 - Monitoreo y alertas: El sistema genera alertas cuando el stock de ingredientes cae por debajo del mínimo configurado, indicando que es necesario hacer un nuevo pedido al proveedor y reiniciar el ciclo.

Validaciones y Control de Inventario

El sistema incluye validaciones automáticas para garantizar la integridad del inventario:

Validaciones de Producción

Existencia de receta: No se puede producir sin receta registrada

Stock suficiente: Valida que haya ingredientes para la cantidad solicitada

Cantidades positivas: No permite producir 0 o cantidades negativas

Ingredientes activos: Solo permite usar ingredientes disponibles

Validaciones de Ingredientes

Stock mínimo: Genera alertas cuando stock < stock mínimo configurado

Unidades de medida: Válida que sean consistentes

Proveedor asignado: Cada ingrediente debe tener un proveedor

Validaciones de Productos

Precios: No permite precios negativos o cero

SKU único: Cada producto tiene un código único

Stock disponible: Al vender, valida que haya suficiente stock

Reportes y Consultas Relacionados

Consultar Historial de Producción

1. Ve a "Estadísticas" - "Reportes"

2. Selecciona "Producción"

3. Filtra por:

- Rango de fechas
- Producto específico
- Usuario que produjo

Alertas de Stock Crítico

- El sistema muestra alertas en tiempo real cuando:
- Productos con stock bajo
- Estas alertas aparecen en:
- Dashboard principal del administrador
- Tabla de inventario de ingredientes
- Notificaciones del sistema

DulceSys

Inicio Clientes Cajeros Proveedores Inventario Pedidos Cierre de Caja Recompensas Estadísticas Hola, Usuario ADMINISTRADOR

Dashboard de Estadísticas

VENTAS DEL MES \$5999.95

PEDIDOS TOTALES 49

CLIENTES NUEVOS (MES) 2

PRODUCTOS ACTIVOS 5

Top 5 Productos Más Vendidos

| Producto | Unidades Vendidas |
|-------------------------|-------------------|
| Pastel de vainilla | 25 |
| Pastel de oreo | 14 |
| Pastel de Chocolate | 11 |
| Pastel de Fresa y Crema | 8 |
| Pastel de Redvelvet | 6 |

Ventas de la Semana

| Día | Ventas de la Semana (\$) |
|-----------|--------------------------|
| Domingo | 0 |
| Lunes | 0 |
| Martes | 0 |
| Miércoles | 0 |
| Jueves | 0 |
| Viernes | 0 |
| Sábado | 0 |

Tamaños Más Vendidos

Ventas por Cajero

VALIDACIONES GENERALES DEL SISTEMA

Validaciones de Interfaz

Menús por rol: Los menús cambian correctamente según el rol del usuario
(Administrador, Cliente, Cajero)

Visualización clara: Diferencia evidente entre las interfaces de cada rol

SOPORTE Y AYUDA

Si experimentas problemas con el sistema o necesitas asistencia:

1. Contacta al administrador del sistema

2. Proporciona detalles del problema:

- Pantalla donde ocurrió
- Acciones que realizaste
- Mensaje de error
- Correo: soporte@dulcesys.com
- Teléfono: (951) 365 984 85
- Horario de soporte: Lunes a Viernes, 9:00 AM - 6:00 PM
- Ubicación: Orquídeas 110, Jardines de la Primavera, Oaxaca

Para capacitación sobre el uso del sistema de producción, solicita una sesión personalizada con el equipo de soporte.

Última actualización: Noviembre 2025

Versión del sistema: DulceSys v2.1

Nuevas funcionalidades: Gestión de Recetas y Sistema de Producción