



UNIVERSIDAD RAGIONAL I

Centro de Estudio: San Raymundo

Curso: Análisis y diseño de sistemas

Tutor: Ing. William Gonzales

Documentación análisis y sistemas

Integrantes:

Bryan Leonardo Rafael Santiagos Galicia

Andersón Osbelí Tzunux Tojín

Juan Jose Antonio Perez Garcia

Mario Adonias Galvez Yoc

Cesar Eduardo Gutiérrez Lima

Carnet:

2349135

2349023

2349044

2349105

2349071

Contenido

CASOS DE USO	4
Gestión de Productos	4
Consultar producto	4
Agregar nuevo producto	5
Actualizar información de productos	7
Eliminar producto	9
gestión de clientes.....	11
Registrar cliente	11
Actualizar cliente.....	13
Eliminar cliente	15
Compras.....	17
Realizar compra.....	17
Emitir comprobante de compra.....	18
Visualizar detalle de compra.....	20
Categorías.....	21
Agregar nueva categoría	21
Editar categoría	23
Eliminar categoría.....	25
Usuarios	27
Agregar usuario	27
Editar usuario.....	29
Eliminar usuario	31
Inicio de sesión.....	34
DIAGRAMAS DEL PROYECTO	36

Diagrama de clases	36
Diagrama de arquitectura general del sistema	37
METODOLOGÍA DE DESARROLLO “CASCADA”	38
Justificación del uso de la Metodología en Cascada	38
Descripción del proceso de desarrollo aplicado.....	39
Diagrama de modelo de cascada	40
CRONOGRAMA.....	41
PRESUPUESTO FINAL	42

CASOS DE USO

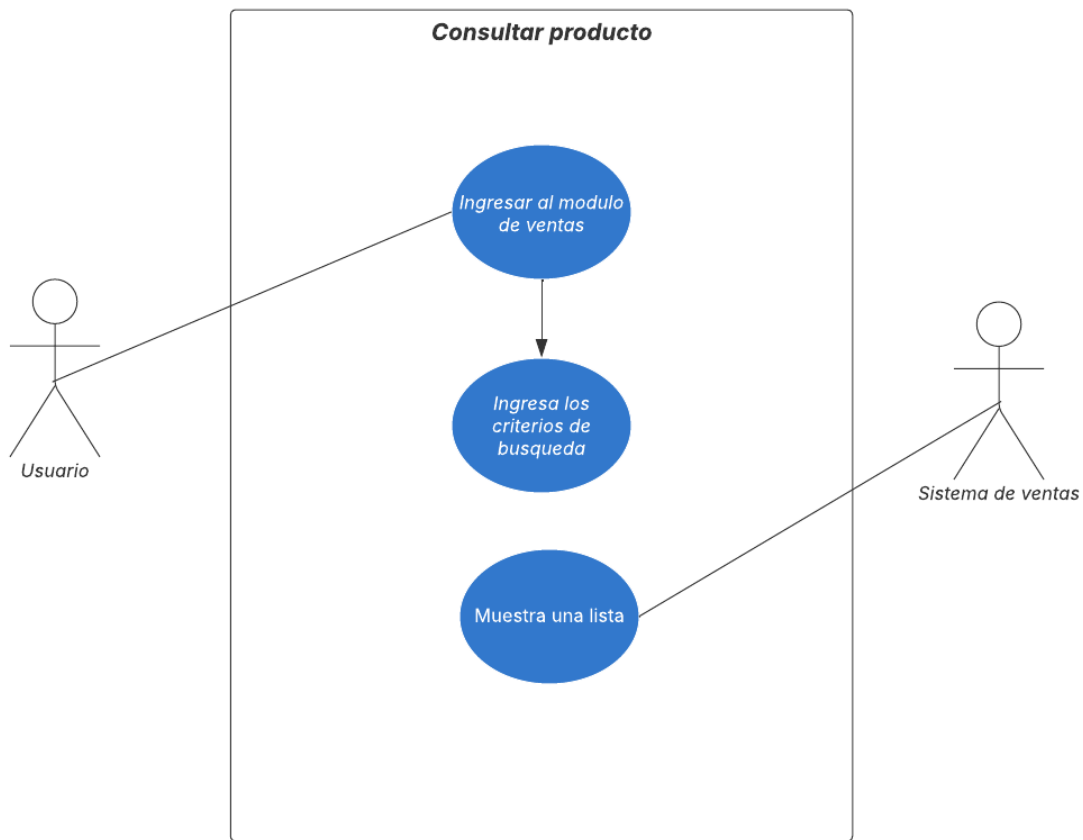
Gestión de Productos

Consultar producto

Descripción: este caso de uso permite al cajero o administrador buscar y visualizar información detallada de los productos registrados en el sistema.

- **Actores:**
 - Usuario.
 - Sistema de ventas.
- **Precondiciones:**
 - El producto debe estar registrado en el sistema.
- **Flujo básico del evento:**
 1. El usuario ingresa al módulo de productos.
 2. Ingresa los criterios de búsqueda.
 3. El sistema muestra una lista de productos que coinciden con el criterio ingresado.
 4. El usuario selecciona un producto de la lista.
 5. El usuario puede visualizar los detalles del producto seleccionado.
 6. El usuario puede cerrar la lista o realizar otra búsqueda.
- **Flujo alternativo:**
 - A2: si el usuario no ingresa ningún criterio en la búsqueda no se mostrará nada en la lista.
- **Excepciones:**
 - E1: si no hay coincidencias, el sistema muestra un mensaje: No se encontraron productos con ese criterio.
- **Resultado esperado:**

El usuario puede visualizar fácilmente la información actualizada de los productos almacenados en el sistema.



Agregar nuevo producto

Descripción: este caso de uso permite al administrador registrar un nuevo producto con su información básica y detalles.

- **Actores:**
 - Administrador.
 - Sistema de ventas.
- **Precondiciones:**
 - El producto no debe estar previamente registrado con el mismo nombre o id.
- **Flujo básico del evento:**
 1. El usuario accede al catálogo de productos.
 2. El sistema genera un formulario de registro.

3. El usuario completa los campos.
4. El usuario guarda el registro haciendo clic en la opción agregar.
5. El sistema valida la información.
6. Si todo está correcto, se guarda en la base de datos.
7. Se muestra un mensaje de: se guardó correctamente.

- **Flujo alternativo:**

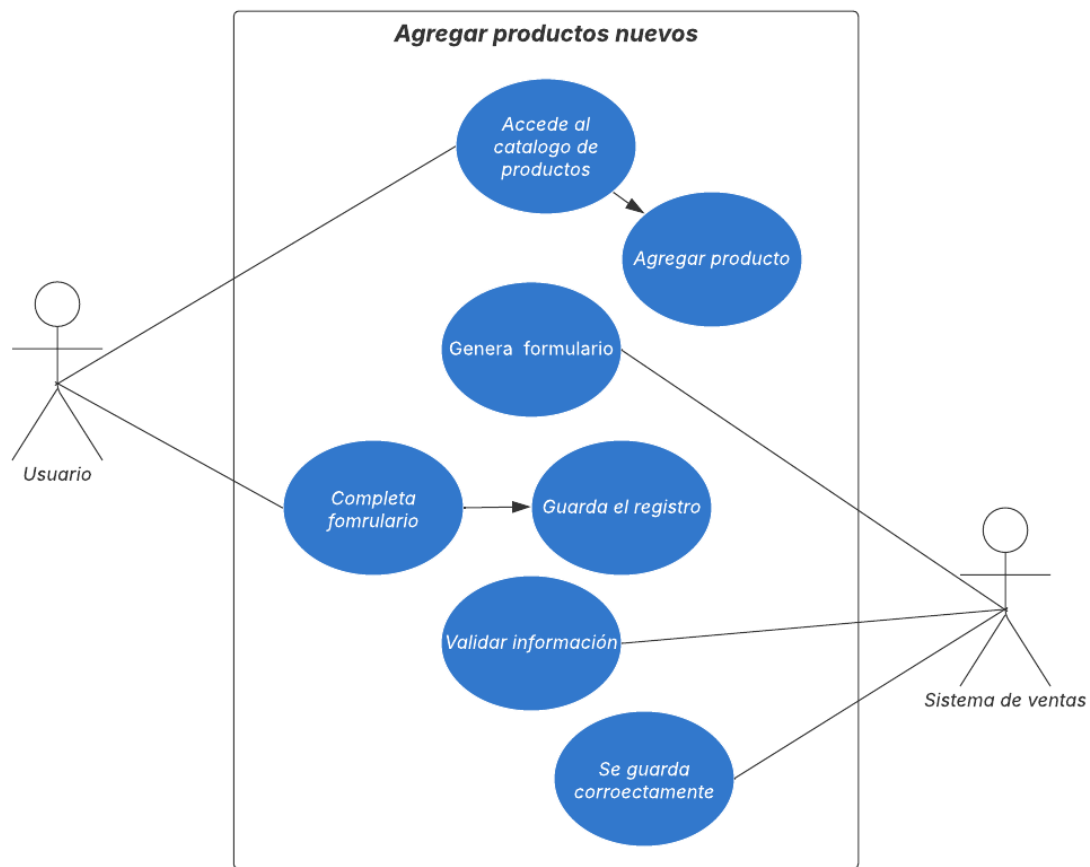
- A1: si un producto similar ya existe el sistema sugerirá revisar el producto para evitar duplicados.

- **Excepciones:**

- E1: los campos requeridos vacíos e inválidos no se permite guardar hasta que se corrijan.

- **Resultado esperado:**

El nuevo producto queda correctamente agregado al catálogo, disponible para ventas y gestión de inventario.



Actualizar información de productos

Descripción: Este caso de uso permite modificar la información de productos ya registrados en el sistema.

- **Actores:**

- Administrador.
- Sistema de ventas.

- **Precondiciones:**

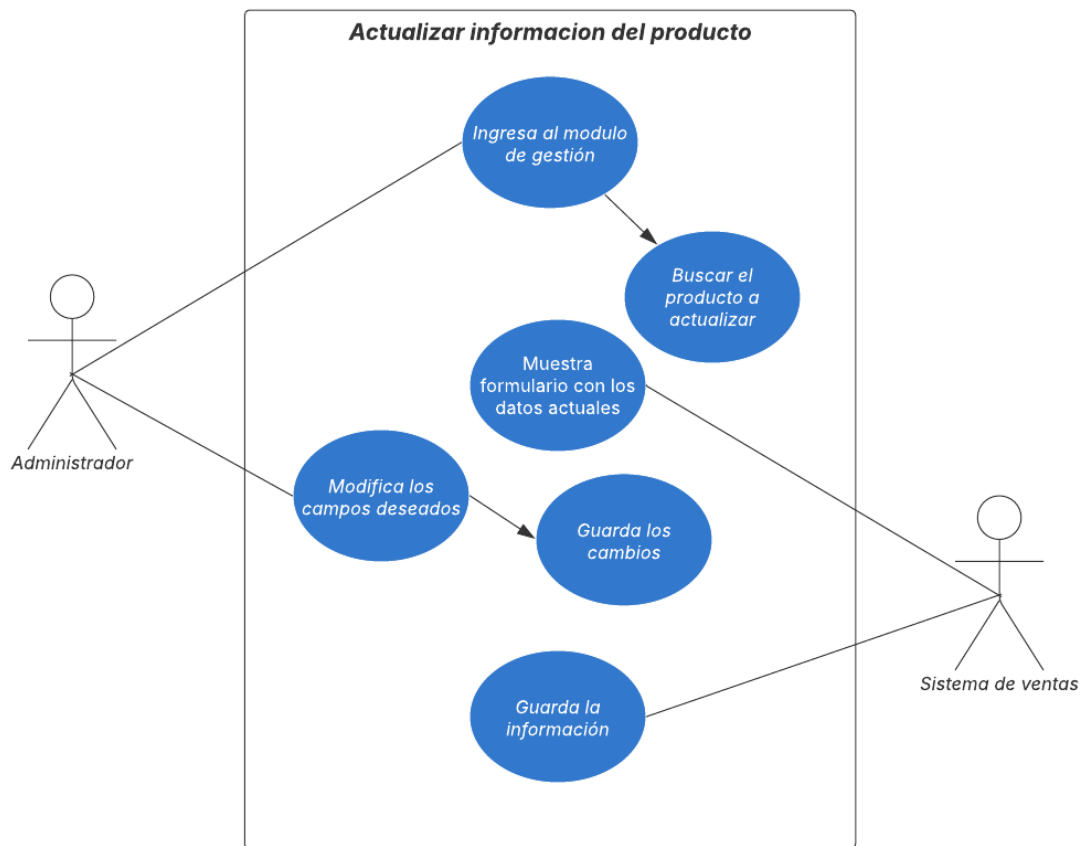
- El producto ya debe existir en la base de datos.

- **Flujo básico del evento:**

1. El usuario o administrador ingresa al módulo de gestión de productos.

2. Busca el producto que desea actualizar.
 3. Selecciona el producto de la lista.
 4. El sistema muestra un formulario con los datos actuales.
 5. El usuario modifica los campos deseados.
 6. El usuario hace clic en la opción editar.
 7. Si todo es correcto, se guarda la información en la base de datos.
 8. El sistema muestra un mensaje de confirmación.
- **Flujos alternativos:**
 - A2: el usuario decide no realizar cambios y sale del formulario.
 - **Excepciones:**
 - Si se ingresa datos inválidos el sistema muestra un mensaje de error y no permite guardar hasta corregir.
 - **Resultado esperado:**

El producto queda actualizado en el sistema con los nuevos datos, y se refleja en el catálogo, inventario y punto de venta si aplica.



Eliminar producto

Descripción: este caso permite al usuario eliminar productos en el catálogo.

- **Actores:**

- Administrador.
- Sistema de ventas.

- **Precondiciones:**

- El producto debe ser existente en el catálogo de productos.

- **Flujo básico del evento:**

1. El administrador accede al módulo de productos.
2. Se busca el producto que se desea eliminar.

3. Selecciona el producto de lista.
4. Hacer clic en la opción Eliminar.
5. El sistema solicita confirmación.
6. El usuario confirma la eliminación.
7. Si todo es válido, el sistema elimina el producto de la base de datos.
8. Se muestra el mensaje: producto eliminado.

- **Flujo alternativo:**

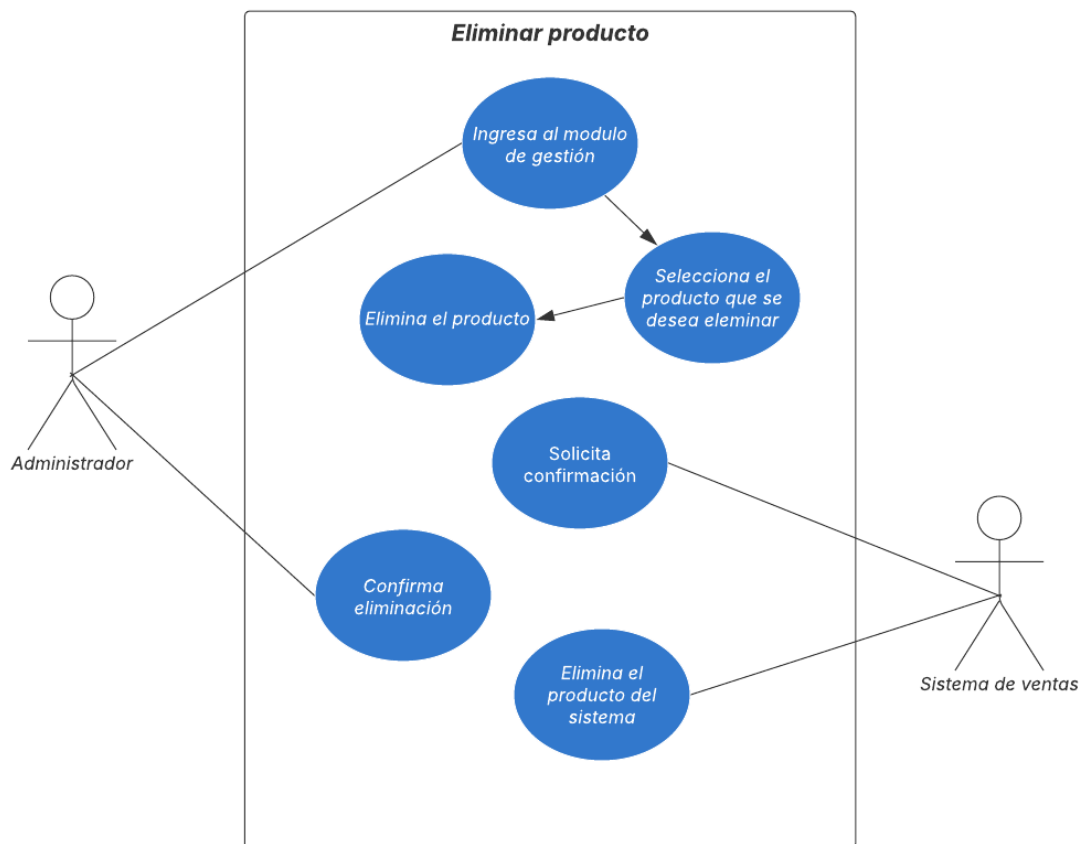
- A1: el usuario no realiza ningún cambio y regresa al menú.

- **Excepciones:**

- E1: el sistema muestra un mensaje de error y no realiza la eliminación.

- **Resultado esperado:**

El producto se elimina correctamente del catálogo (o se inactiva si hay restricciones), y ya no está disponible para venta o consulta general.



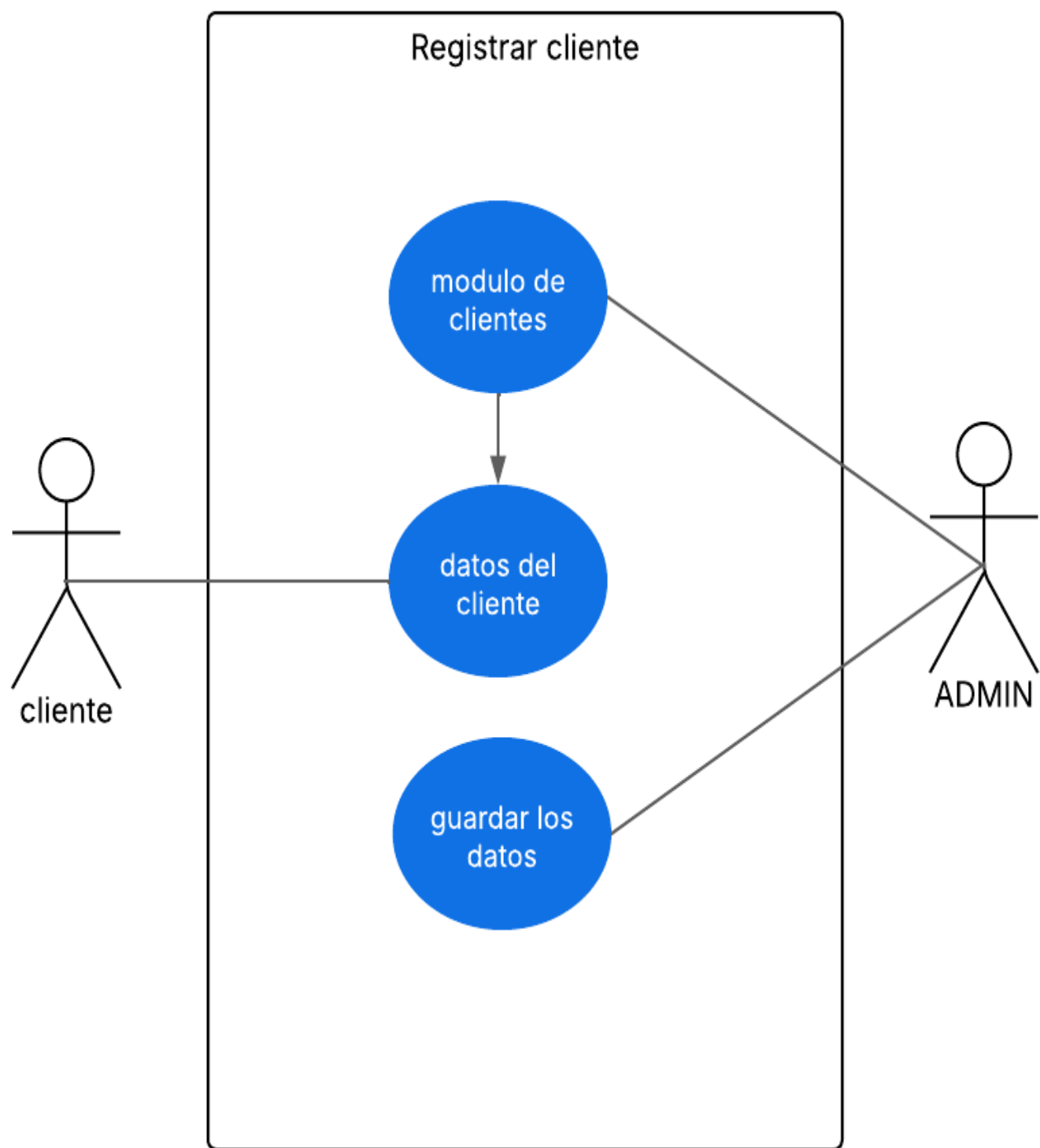
gestión de clientes

Registrar cliente

Descripción: este caso de uso permite registrar un nuevo cliente en el sistema.

- **Actores:**
 - Usuario
 - Administrador del sistema.
 - Base de datos.
- **Precondiciones:**
 - Validar que todos los campos se completen correctamente.
- **Flujo básico del evento:**
 1. El administrador accede al módulo de clientes.
 2. El sistema solicita los datos personales del cliente: nombre, apellido, NIT, correo electrónico, teléfono.
 3. El administrador ingresa la información requerida.
 4. El administrador hace clic en la opción agregar.
 5. El sistema guarda los datos del nuevo cliente en la base de datos
 6. El sistema crea un id único para el cliente.
 7. El sistema muestra un mensaje: "cliente registrado correctamente".
- **Flujos alternativos:**
 - El sistema muestra un mensaje de error indicando los campos inválidos.
- **Excepciones:**
 - El sistema ofrece la opción de actualizar datos existentes o cancelar
- **Resultado esperado:**

El cliente queda registrado correctamente, la base de datos se actualiza, y el cliente recibe su usuario.

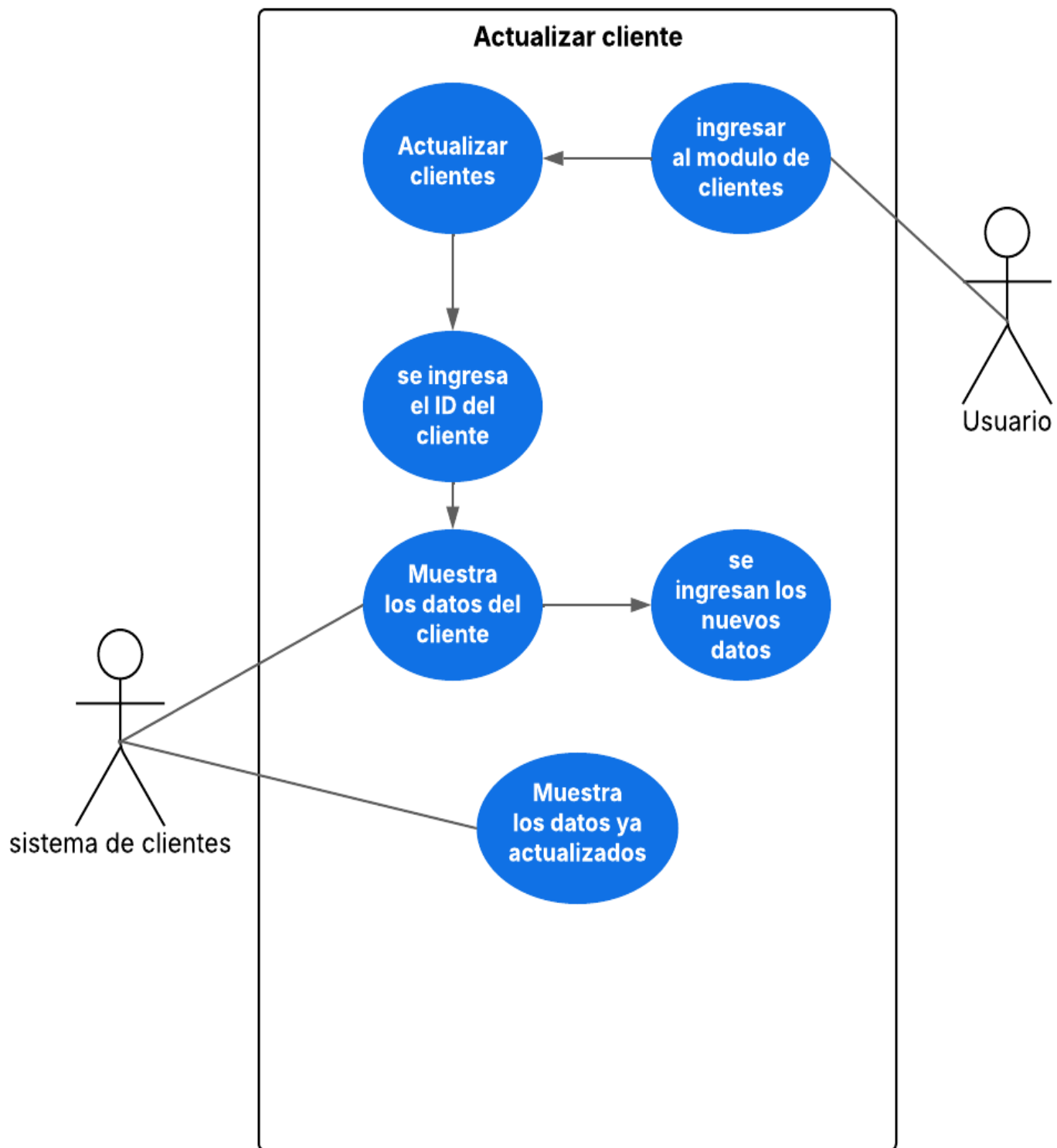


Actualizar cliente

Descripción: este caso de uso permite actualizar un cliente registrado en el sistema.

- **Actores:**
 - Administrador del sistema.
 - Base de datos.
- **Precondiciones:**
 - El cliente ya está registrado en la base de datos.
- **Flujo básico del evento:**
 1. El administrador accede al módulo de clientes.
 2. El sistema muestra la información del cliente en una lista.
 3. El sistema permite modificar los datos del cliente.
 4. Presiona editar para actualizar los datos cliente.
 5. Se guardan los cambios realizados, y se actualiza la base de datos.
 6. El sistema muestra un mensaje: "cliente actualizado correctamente".
- **Flujos alternativos:**
 - El sistema muestra un mensaje de error indicando los campos inválidos.
- **Excepciones:**
 - No se debe permitir actualizar el documento de identidad (campo único).
- **Resultado esperado:**

El administrador puede modificar los datos de cliente teléfono, correo, etc, mas no puede modificar el dato de identificación único del cliente.



Eliminar cliente

Descripción: este caso de uso permite eliminar un cliente registrado en la base de datos desde el sistema de clientes.

- **Actores:**

- Usuario
- Administrador del sistema.
- Base de datos.

- **Precondiciones:**

El cliente tiene que estar de acuerdo con eliminar su usuario del sistema y de la base de datos.

- **Flujo básico del evento:**

1. El administrador accede al módulo de clientes.
2. El administrador selecciona la información del cliente desde una lista.
3. El sistema permite visualizar los datos del cliente que se van a eliminar.
4. El administrador pulsa el botón de eliminar
5. El sistema muestra un mensaje: "cliente eliminado correctamente".

- **Flujos alternativos:**

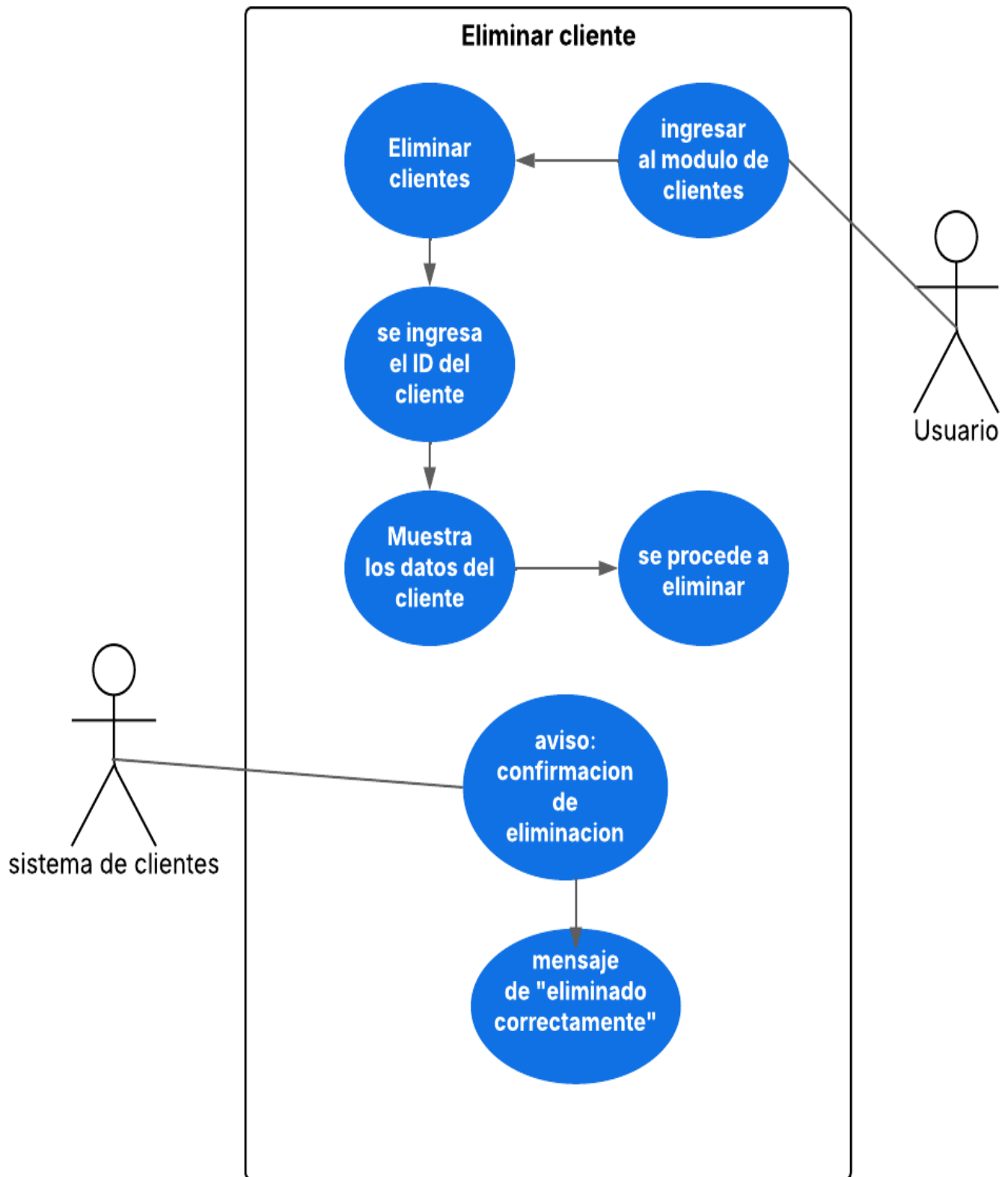
- El sistema da un aviso/advertencia que si desea eliminar

- **Excepciones:**

- Se informa que no existe el cliente o que el ID es invalido.

- **Resultado esperado:**

Que se puedan eliminar los clientes de una manera sutil y sin complicaciones con la ayuda del modulo de clientes y la opción eliminar cliente.



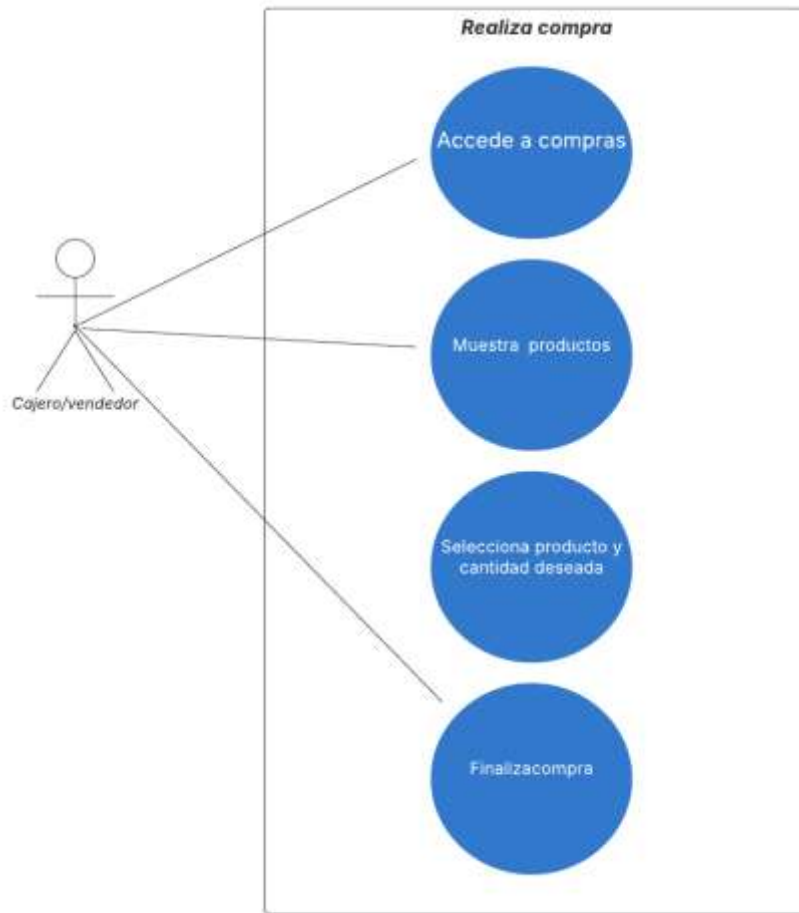
Compras

Realizar compra

Descripción: Este caso de uso permite al actor buscar y seleccionar productos disponibles en el sistema para agregarlos a una compra.

- **Actores:**
 - Cajero
 - Vendedor
 - Sistema de ventas
- **Precondiciones:**
 - El inventario tiene productos disponibles.
- **Flujo básico del evento:**
 1. El cajero accede al módulo de compras.
 2. El sistema muestra un formulario con campos para completar la compra.
 3. El sistema muestra la Id de los productos disponibles.
 4. El sistema muestra la Id de los clientes existentes.
 5. El cajero selecciona el producto y selecciona la cantidad deseada.
 6. El cajero completa agregando el registro en la tabla haciendo clic en agregar.
 7. El cajero realiza el mismo proceso con otros productos si es necesario.
 8. El cajero finaliza la compra.
- **Flujo alternativo:**
 - A7: el cajero puede eliminar el producto del carrito de compras.
- **Excepciones:**
 - E1: el producto no existe o no se encuentra en el catálogo de productos.
 - E2: No hay suficiente stock del producto seleccionado.
- **Resultado esperado:**

Los productos seleccionados se agregan correctamente a la operación de venta con sus respectivas cantidades, listos para calcular el total y proceder con el pago.



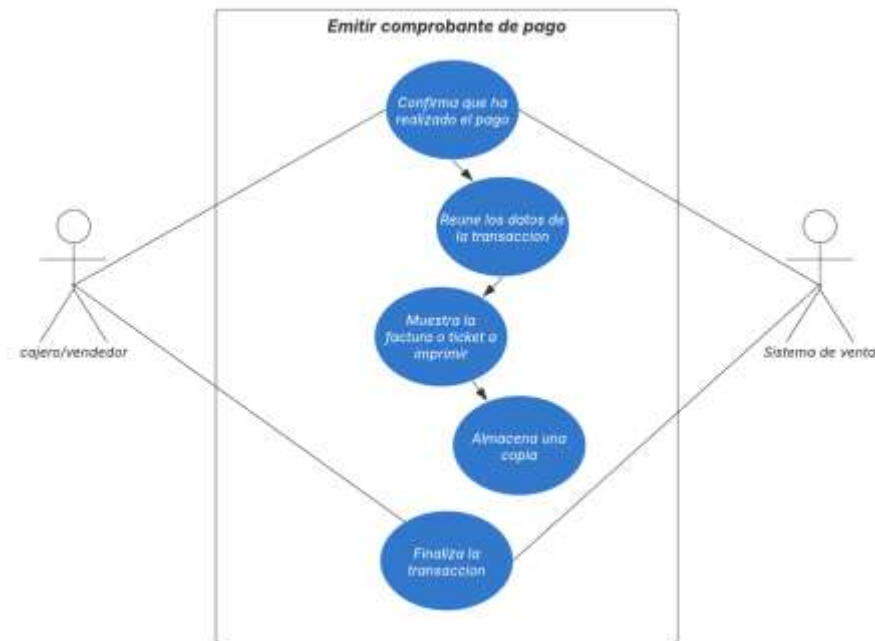
Emitir comprobante de compra

Descripción: Este caso de uso describe cómo el sistema genera y entrega un comprobante (ticket o factura) tras completarse una compra.

- **Actores:**
 - Sistema de ventas

- Cajero
- **Precondiciones:**
 - La venta ha sido registrada correctamente.
 - El pago ha sido completado y confirmado.
- **Flujo básico del evento:**
 1. El sistema confirma que se ha realizado correctamente.
 2. Reúne todos los datos de la transacción.
 3. Presiona la opción generar recibo.
 4. Genera un comprobante de pago.
 5. El sistema muestra la factura o ticket para imprimir.
- **Flujo alternativo:**
 - A3: El cliente no quiere generar factura.
- **Excepciones:**
 - E1: la venta no se completa o es inconsistente.
- **Resultado esperado:**

El cliente recibe un comprobante válido de la compra, ya sea impreso o digital, y el sistema guarda una copia para historial y control fiscal.



Visualizar detalle de compra

Descripción: Este caso de uso permite al usuario consultar la información detallada de una compra previamente realizada.

• **Actores:**

- Administrador.
- Cajero.

• **Precondiciones:**

- El usuario ha iniciado sesión correctamente.
- Existen compras registradas asociadas al usuario.

• **Flujo básico del evento:**

1. El usuario accede al módulo de compras.
2. Presiona el botón detalles de compras.
3. El sistema muestra los detalles de la compra.
4. El usuario revisa la información en pantalla.

• **Flujos alternativos:**

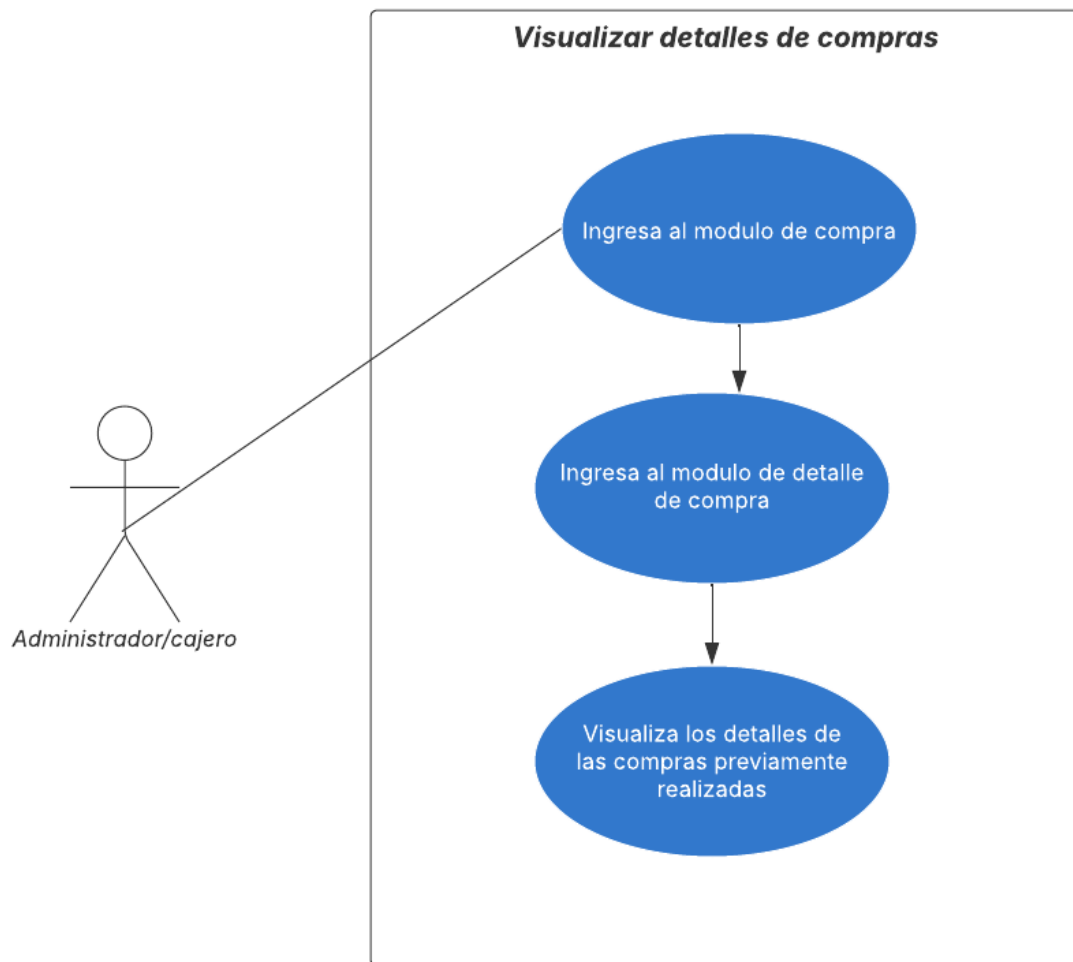
- El usuario regresa a la lista visualizar los detalles de compra.

• **Excepciones:**

- El sistema muestra: "Ocurrió un error al cargar los detalles."

• **Resultado esperado:**

El usuario visualiza correctamente toda la información relacionada con una compra específica registrada en el sistema.



Categorías

Agregar nueva categoría

Descripción: este caso de uso permite al administrador crear una nueva categoría de productos.

- **Actores:**
 - Administrador.
 - Sistema de ventas.
- **Precondiciones:**
 - El administrador ha iniciado sesión correctamente.

- **Flujo básico del evento:**

1. El administrador accede al módulo de gestión de categorías.
2. El administrador completa el formulario.
3. El administrador hace clic en la opción agregar
4. El sistema registra la nueva categoría en la base de datos.
5. El sistema muestra un mensaje de confirmación.
6. La nueva categoría aparece disponible para asignar productos.

- **Flujos alternativos:**

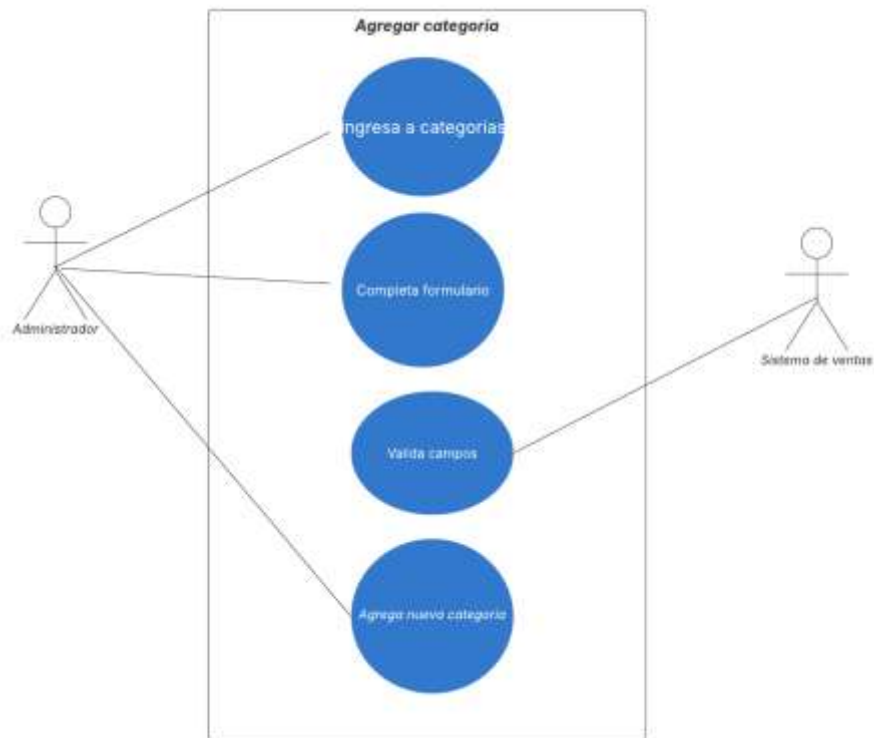
- A3: El administrador cancela el registro para agregar una categoría.

- **Excepciones:**

- E1: El sistema muestra: *“Complete todos los campos obligatorios antes de continuar.”*
- E2: El sistema muestra: *“Ocurrió un error al guardar la categoría. Intente nuevamente más tarde.”*

- **Resultado esperado:**

La nueva categoría queda registrada correctamente y disponible para la gestión de productos.



Editar categoría

Descripción: Este caso de uso permite al administrador modificar los datos de una categoría de productos previamente registrada.

- **Actores:**

- Administrador.
- Sistema de ventas.

- **Precondiciones:**

- El administrador ha iniciado sesión correctamente.
- La categoría para editar ya existe en el sistema.

- **Flujo básico del evento:**

1. El administrador accede al módulo de gestión de categorías.
2. El administrador selecciona la categoría que desea editar.
3. El sistema carga los datos actuales de la categoría.

4. El administrador modifica los datos necesarios en el formulario.
5. El administrador hace clic en la opción "Editar".
6. El sistema actualiza los datos de la categoría en la base de datos.
7. El sistema muestra un mensaje de confirmación.
8. La categoría actualizada queda disponible con los nuevos datos para la gestión de productos.

• **Flujos alternativos:**

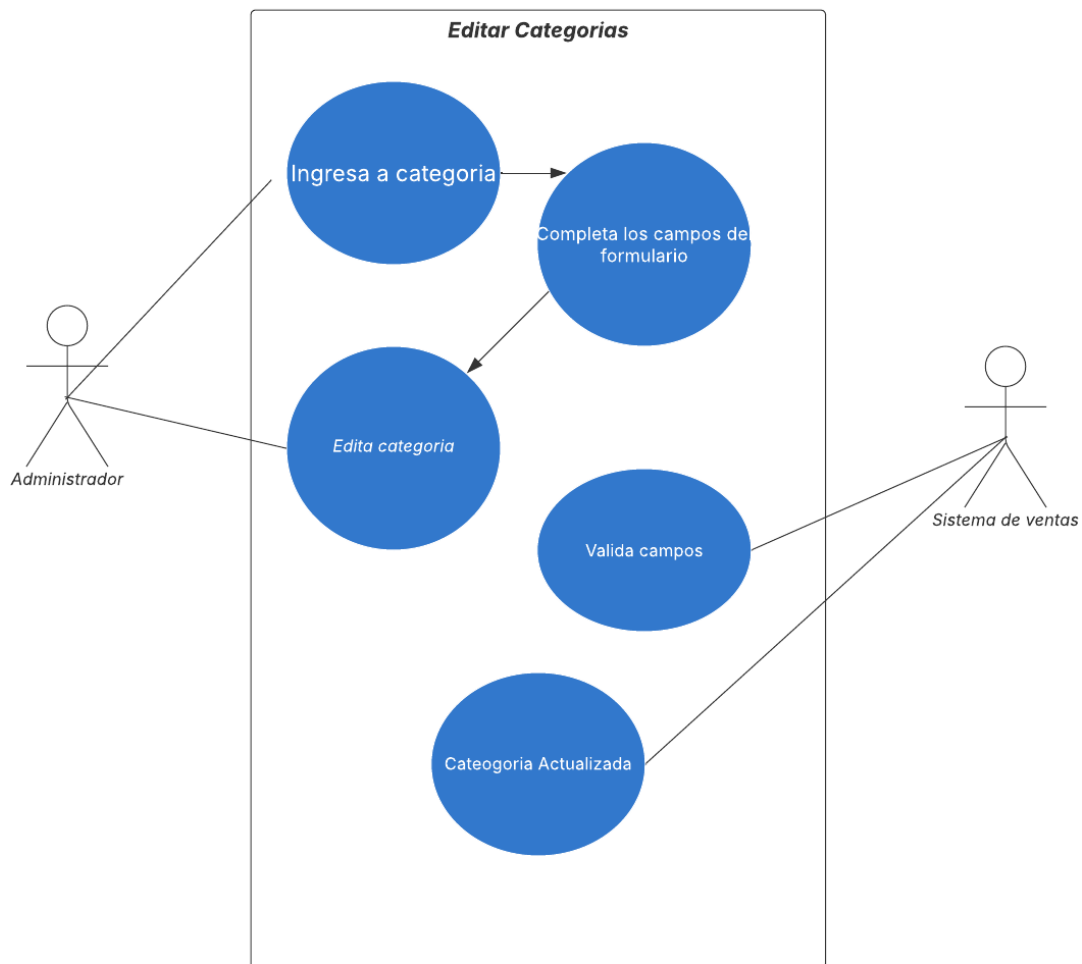
- A1: El administrador cancela la edición de la categoría.

• **Excepciones:**

- E1: El sistema muestra: "Complete todos los campos obligatorios antes de continuar."
- E2: El sistema muestra: "Ocurrió un error al actualizar la categoría. Intente nuevamente más tarde."

• **Resultado esperado:**

La categoría queda actualizada correctamente y disponible con los nuevos datos para su uso en el sistema.



Eliminar categoría

Descripción: Este caso de uso permite al administrador eliminar una categoría de productos previamente registrada en el sistema.

- **Actores:**

- Administrador.
- Sistema de ventas.

- **Precondiciones:**

- El administrador ha iniciado sesión correctamente.

- **Flujo básico del evento:**

1. El administrador accede al módulo de gestión de categorías.
2. El administrador selecciona la categoría que desea eliminar.
3. El administrador hace clic en la opción "Eliminar".
4. El sistema solicita confirmación de la acción.
5. El administrador confirma la eliminación.
6. El sistema elimina la categoría de la base de datos.
7. El sistema muestra un mensaje de confirmación.
8. La categoría deja de estar disponible en el sistema.

- **Flujos alternativos:**

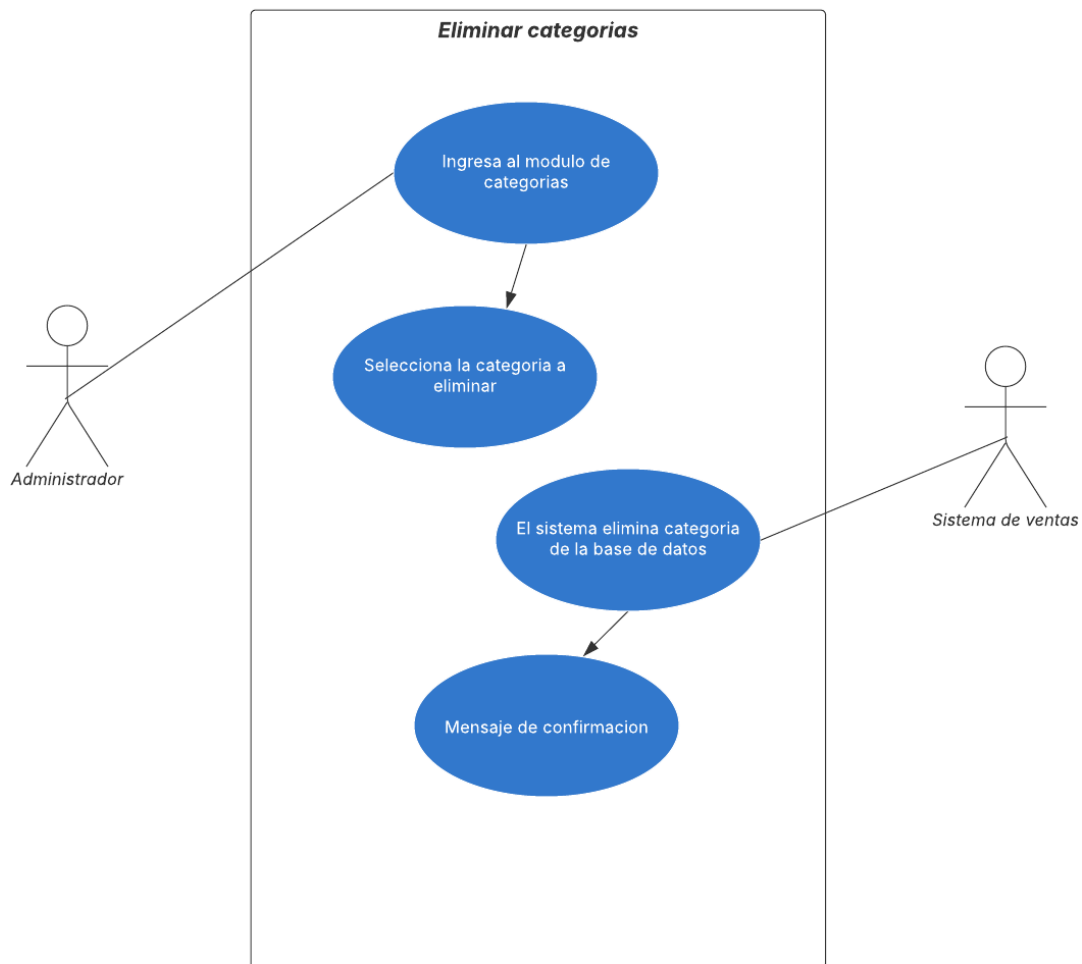
- A1: El administrador cancela la acción de eliminar la categoría.

- **Excepciones:**

- E1: El sistema muestra: "No se puede eliminar la categoría porque está asociada a uno o más productos."
- E2: El sistema muestra: "Ocurrió un error al eliminar la categoría. Intente nuevamente más tarde."

- **Resultado esperado:**

La categoría es eliminada correctamente del sistema y ya no está disponible para su asignación o consulta.



Usuarios

Agregar usuario

Descripción: Este caso de uso permite al administrador registrar un nuevo usuario en el sistema.

• **Actores:**

- Administrador.
- Sistema de ventas.

- **Precondiciones:**

- El administrador ha iniciado sesión correctamente.

- **Flujo básico del evento:**

1. El administrador accede al módulo de gestión de usuarios.
2. El administrador completa el formulario con los datos del nuevo usuario
3. El administrador hace clic en la opción "Agregar".
4. El sistema registra el nuevo usuario en la base de datos.
5. El sistema muestra un mensaje de confirmación.
6. El nuevo usuario queda habilitado para iniciar sesión.

- **Flujos alternativos:**

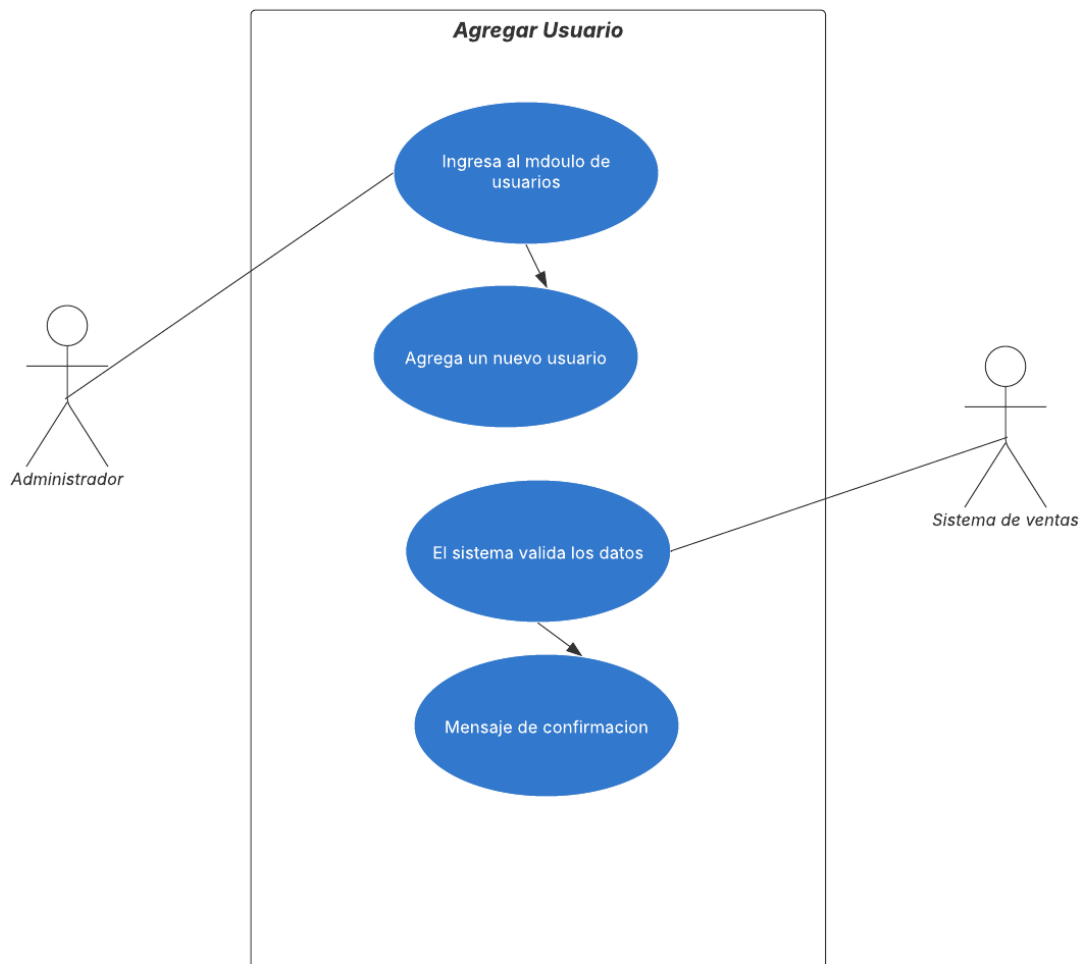
- A2: El administrador cancela el registro del nuevo usuario.

- **Excepciones:**

- El sistema muestra un mensaje: "Complete todos los campos obligatorios antes de continuar."

- **Resultado esperado:**

El nuevo usuario queda registrado correctamente en el sistema y habilitado para iniciar sesión.



Editar usuario

Descripción: Este caso de uso permite al administrador modificar los datos de un usuario previamente registrado en el sistema.

• **Actores:**

- Administrador.
- Sistema de ventas.

• **Precondiciones:**

- El administrador ha iniciado sesión correctamente.

- El usuario para editar ya existe en el sistema.

• **Flujo básico del evento:**

1. El administrador accede al módulo de gestión de usuarios.
2. El administrador selecciona el usuario que desea editar.
3. El sistema muestra el formulario con los datos actuales del usuario.
4. El administrador modifica los campos necesarios.
5. El administrador hace clic en la opción "Editar".
6. El sistema valida los datos modificados.
7. El sistema actualiza la información del usuario en la base de datos.
8. El sistema muestra un mensaje de confirmación.
9. El usuario queda actualizado con los nuevos datos.

• **Flujos alternativos:**

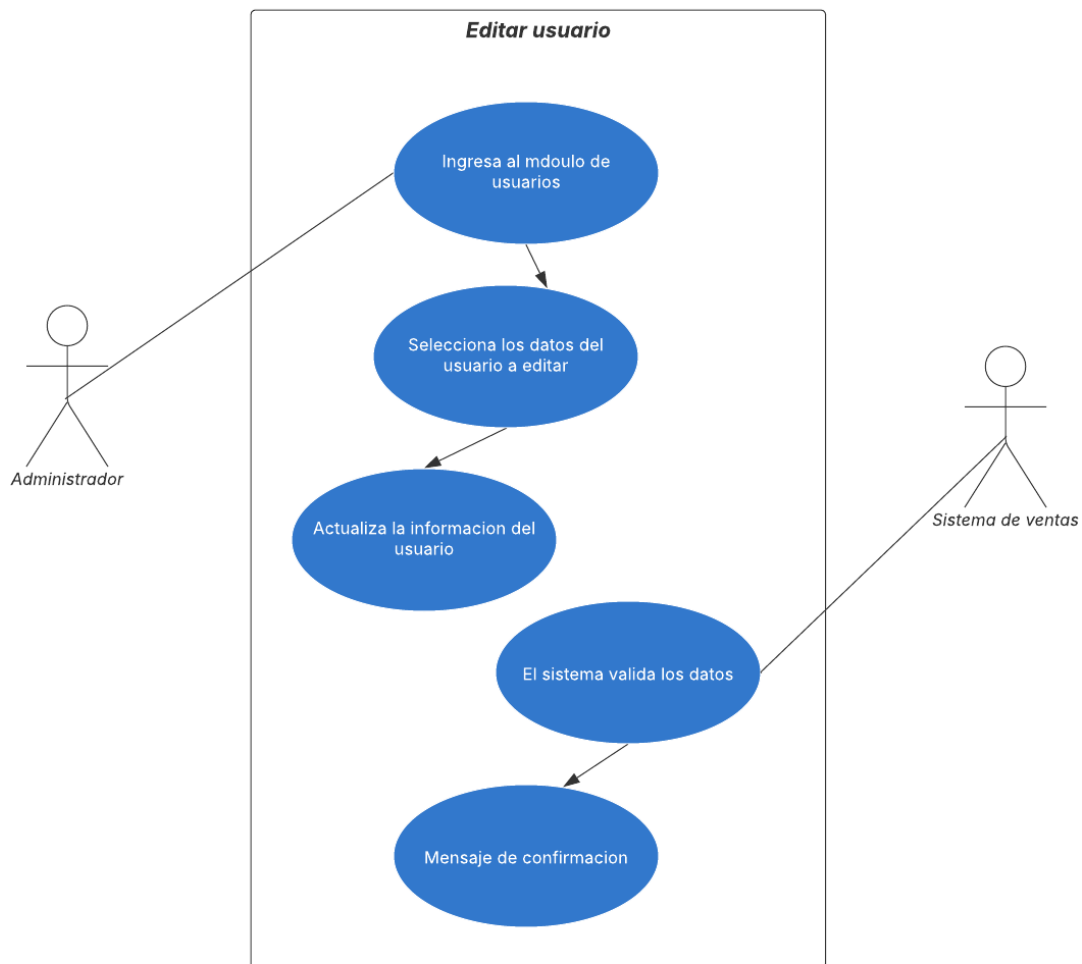
- A5: El administrador cancela la edición del usuario.

• **Excepciones:**

- E1: El sistema muestra: "Complete todos los campos obligatorios antes de continuar."

• **Resultado esperado:**

El usuario queda actualizado correctamente en el sistema con la nueva información registrada.



Eliminar usuario

Descripción: Este caso de uso permite al administrador eliminar un usuario previamente registrado en el sistema.

- **Actores:**

- Administrador.
- Sistema de ventas.

- **Precondiciones:**

- El administrador ha iniciado sesión correctamente.

- El usuario para eliminar debe existir en el sistema.

• **Flujo básico del evento:**

1. El administrador accede al módulo de gestión de usuarios.
2. El administrador selecciona al usuario que desea eliminar.
3. El administrador hace clic en la opción “Eliminar”.
4. El sistema solicita confirmación de la acción.
5. El administrador confirma la eliminación.
6. El sistema elimina al usuario de la base de datos.
7. El sistema muestra un mensaje de confirmación.
8. El usuario deja de estar disponible para iniciar sesión o realizar operaciones en el sistema.

• **Flujos alternativos:**

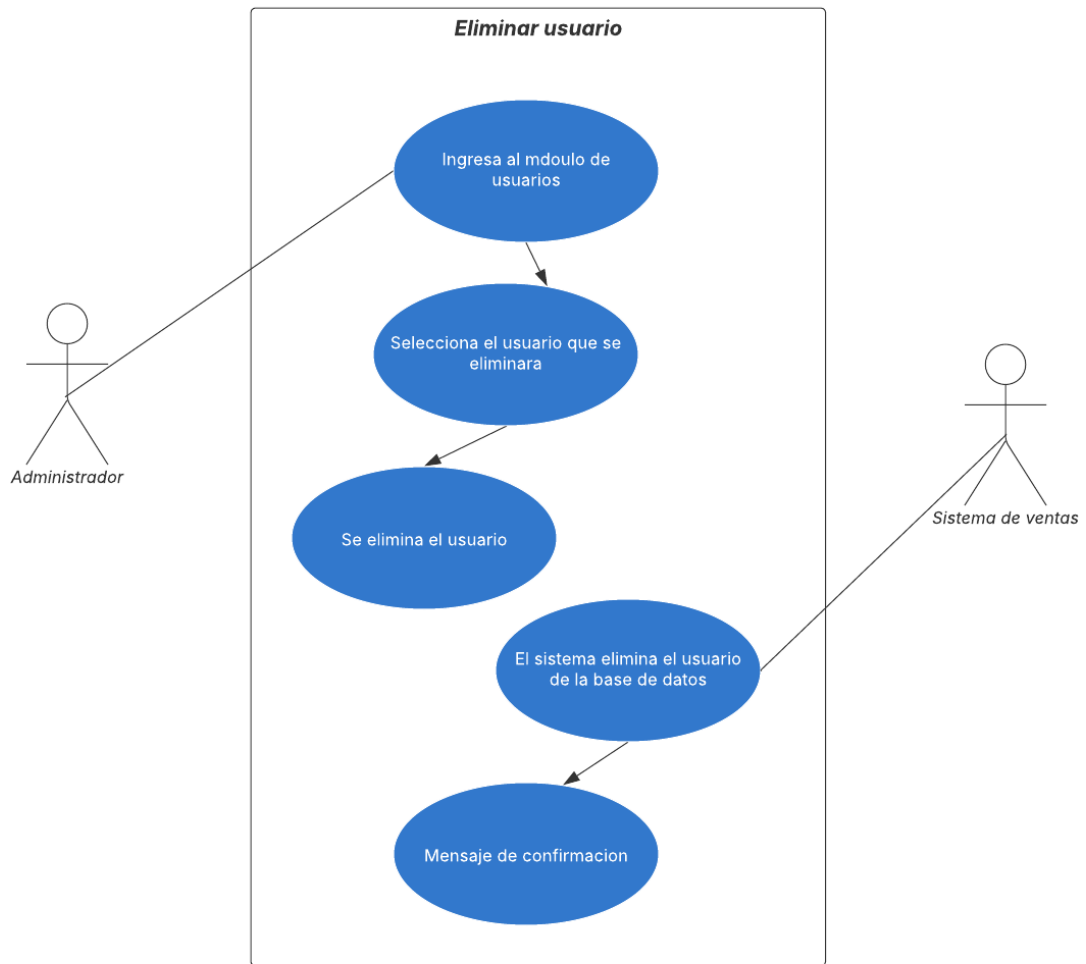
- El administrador cancela la acción de eliminar el usuario.

• **Excepciones:**

- El sistema muestra: “Ocurrió un error al eliminar el usuario. Intente nuevamente más tarde.”

• **Resultado esperado:**

El usuario es eliminado correctamente del sistema y ya no puede acceder ni realizar actividades.



Inicio de sesión

Descripción: Este caso de uso permite a los usuarios acceder al sistema mediante la autenticación con sus credenciales (usuario y contraseña).

• **Actores:**

- Usuario del sistema (Administrador, Cajero)
- Sistema de ventas.

• **Precondiciones:**

- El usuario está registrado en el sistema.
- El sistema está operativo y accesible.

• **Flujo básico del evento:**

1. El usuario accede a la pantalla de inicio de sesión.
2. El usuario introduce su nombre de usuario y contraseña.
3. El usuario hace clic en el botón "Iniciar sesión".
4. El sistema valida las credenciales.
5. El sistema verifica que la cuenta esté activa.
6. El sistema permite el acceso al usuario y lo redirige a la pantalla principal correspondiente según su rol.
7. El sistema muestra un mensaje de bienvenida.

• **Flujos alternativos:**

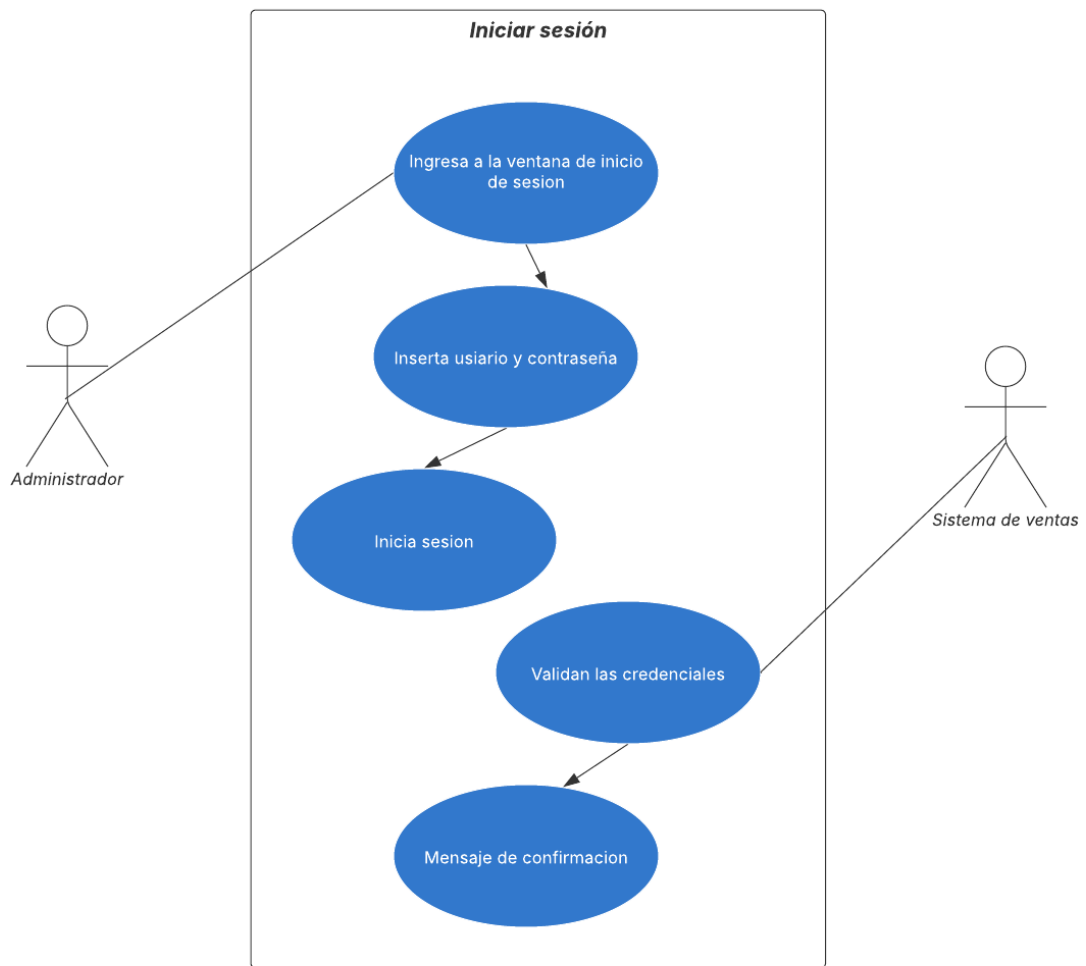
- El usuario cancela el proceso de inicio de sesión.

• **Excepciones:**

- El sistema muestra: "credenciales incorrectas."

• **Resultado esperado:**

El usuario accede correctamente al sistema con sus permisos y funcionalidades habilitadas según su rol.



DIAGRAMAS DEL PROYECTO

Diagrama de clases

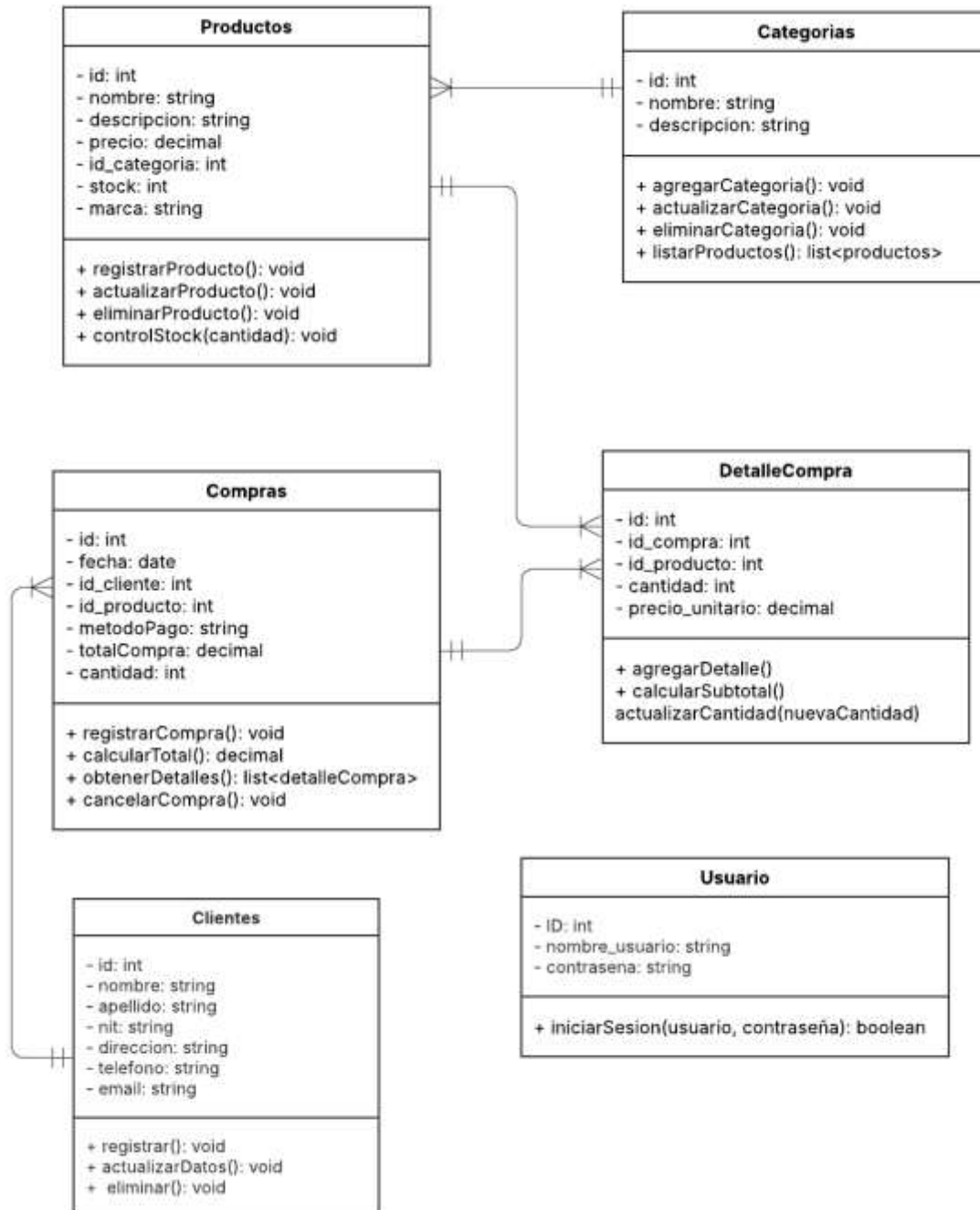
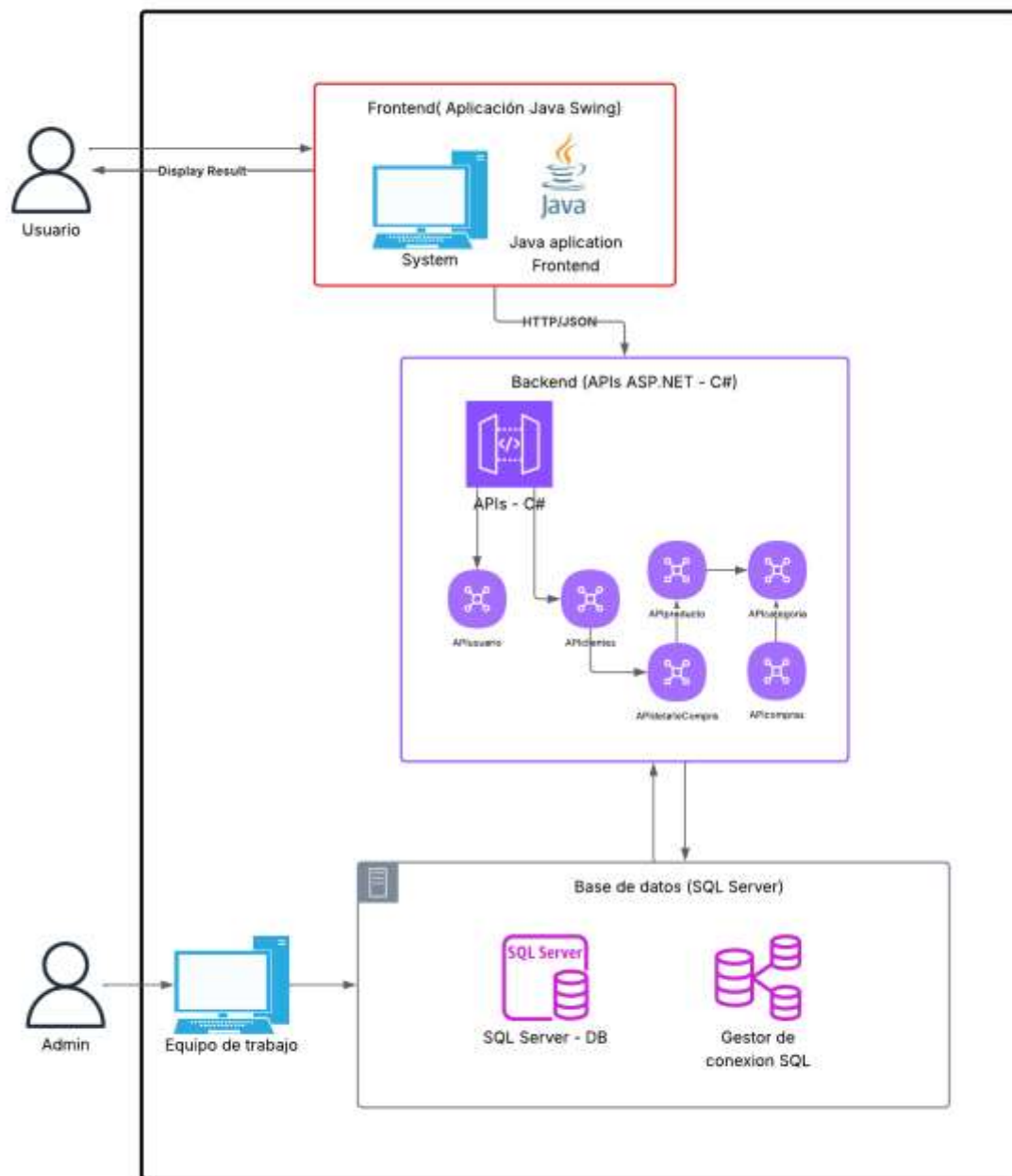


Diagrama de arquitectura general del sistema



METODOLOGÍA DE DESARROLLO “CASCADA”

Justificación del uso de la Metodología en Cascada

Para el desarrollo del sistema de ventas, que incluye los módulos de Gestión de Productos y Gestión de Clientes, se optó por emplear la metodología de desarrollo en cascada. Esta decisión fue tomada considerando las características del proyecto, la estabilidad de los requisitos y la organización del equipo de trabajo.

A continuación, se detallan las razones principales por las que se seleccionó esta metodología:

1. **Claridad en los requisitos desde el inicio:** desde la fase de análisis, los requerimientos funcionales del sistema fueron definidos de forma clara, concreta y sin ambigüedades. Esto permitió estructurar un plan de desarrollo donde cada fase tuviera objetivos delimitados y resultados esperados, tal como lo propone el modelo en cascada.
2. **Proceso secuencial ordenado y definidas:** la naturaleza lineal de esta metodología favorece la planificación y ejecución ordenada del proyecto. En este caso, se comenzó por la documentación y análisis, se continuó con el diseño de la base de datos y la creación de la API (backend), y posteriormente se desarrollará la interfaz gráfica con Java Swing (frontend). Esta estructura secuencial es coherente con el enfoque en cascada.
3. **Énfasis en la documentación técnica:** el enfoque en cascada prioriza una documentación exhaustiva antes de iniciar cada fase. Esto resultó especialmente útil para nuestro equipo, ya que se elaboraron especificaciones técnicas que facilitaron el desarrollo y permitieron una comprensión integral del sistema por parte de todos los miembros.
4. **Facilidad de gestión y control del proyecto:** el modelo en cascada permite llevar un control riguroso del avance del proyecto, con entregables definidos al finalizar cada etapa. Esto ha permitido una asignación eficiente de tareas,

mayor control de tiempos y una verificación clara del cumplimiento de los objetivos.

5. **Adecuado para Proyectos Académicos o de Corto Alcance:** dado que se trata de un sistema de ventas con funcionalidades bien definidas desde el inicio y con bajo riesgo de cambios a mitad del desarrollo, la metodología en cascada resulta adecuada. Además, su aplicación es común en entornos académicos o proyectos con objetivos concretos y plazos definidos.

Descripción del proceso de desarrollo aplicado

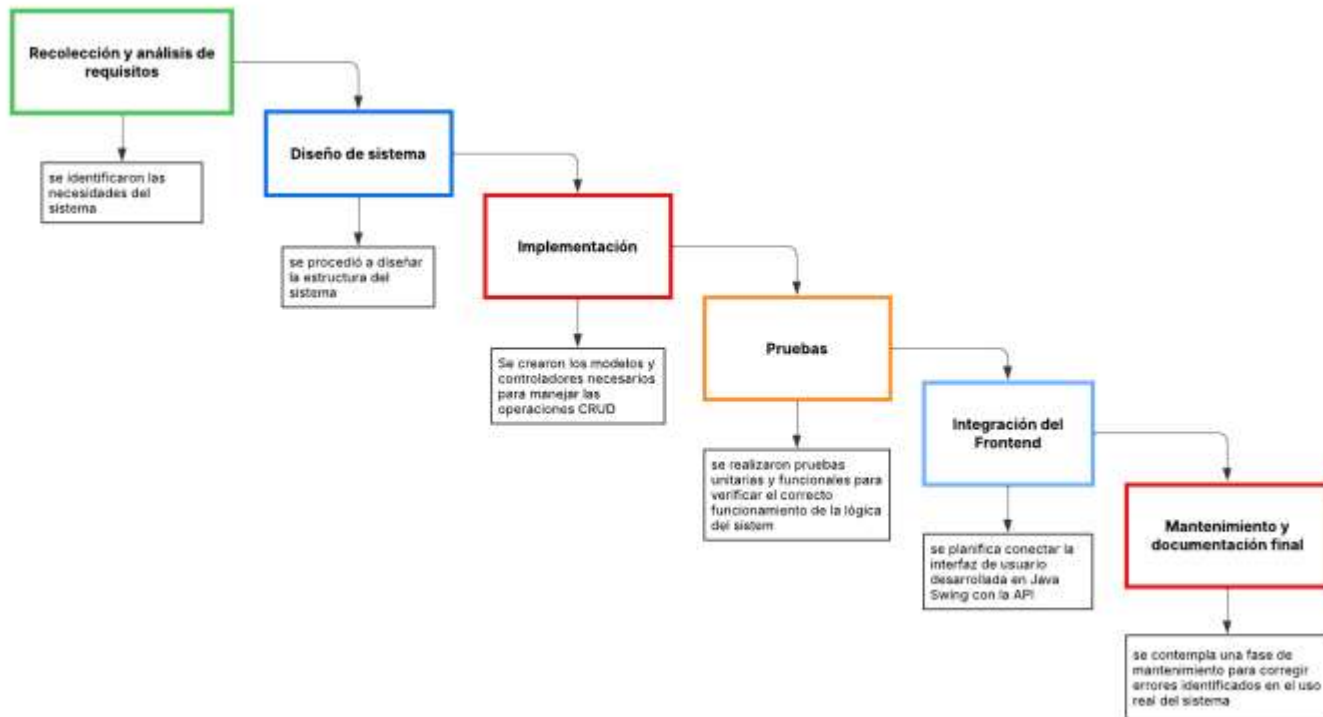
A continuación, se describe cada una de estas etapas y cómo fueron aplicadas en el proyecto:

1. **Fase de Recolección y Análisis de Requisitos:** en esta primera etapa se identificaron las necesidades del sistema, a través de reuniones de grupo y análisis de casos similares. Se definieron los módulos principales (Gestión de Productos y Gestión de Clientes), así como las funcionalidades necesarias, como el CRUD de productos y clientes, el control de stock, y el registro de compras.
2. **Fase de Diseño del Sistema:** con los requisitos claros, se procedió a diseñar la estructura del sistema. Se elaboraron los diagramas entidad-relación para la base de datos, definiendo tablas como **categoría, productos, clientes, compras, detalleCompra y usuarios**. También se diseñó la arquitectura de la API y su integración futura con la interfaz gráfica.
3. **Fase de Implementación:** en esta etapa se desarrolló la base de datos en SQL Server y se programó la API utilizando un lenguaje backend (ASP.NET – C#, según lo utilizado por el grupo). Se crearon los modelos y controladores necesarios para manejar las operaciones CRUD de productos y clientes.
4. **Fase de Pruebas:** una vez finalizada la implementación del backend, se realizaron pruebas unitarias y funcionales para verificar el correcto

funcionamiento de la lógica del sistema, validación de datos, y manejo de errores. Esto aseguró la integridad de la base de datos y la correcta respuesta de la API a distintas peticiones.

5. **Fase de Integración del Frontend:** en esta fase se planifica conectar la interfaz de usuario desarrollada en Java Swing con la API previamente construida. Esta conexión permitirá realizar las operaciones desde una interfaz gráfica, como la gestión de productos, clientes y visualización de registros de compras.
6. **Fase de Mantenimiento y Documentación Final:** finalmente, se contempla una fase de mantenimiento para corregir errores identificados en el uso real del sistema, así como la entrega de la documentación técnica completa que detalle la arquitectura del sistema, los endpoints de la API, y las instrucciones de uso del software.


Diagrama de modelo de cascada:



CRONOGRAMA

ruta del proyecto

Fecha de Entrega: 7-06-2025 | Fecha Actual: 19-04-2025
Duración Total: = 7 semanas (49 días)

	PLANIFICACIÓN INICIAL	DESARROLLO DEL BACKEND	DESARROLLO DEL FRONTEND	PRUEBAS FINALES	INFORME FINAL	ENTREGA
RESPONSABLES	(19/04 - 25/04) DURACION: 1 SEMANA	(26/04 – 16/05) DURACION: 3 SEMANAS	(17/05 – 30/05) DURACION: 2 semanas	(31/05 – 04/06) DURACION: 4 días	(05/06 – 06/06) DURACION: 2 días	(07/06)
BRYAN JOSE MARIO OSVELI CESAR	Definir metodología: Reunión para elegir una metodología Agile o de cascada. Oyvel documentará la decisión.	Pruebas Postman: Verificar que todos los endpoints respondan correctamente (200/400/500).	Pruebas de integración: Asegurar que frontend y backend interactúan sin errores.	PRUEBAS ENTREGABLES	INFORME FINAL	ENTREGA FINAL DEL PROYECTO DIA SABADO 7 DE JUNIO DEL 2025
CESAR	Asignar tareas: Se reparten las tareas según habilidades de cada estudiante.	Modelar base de datos: Esquema SQL con tablas: Productos (id, nombre, stock, precio) y Clientes (id, nombre, correo).	Diseñar UI (Clientes): Pantallas para gestionar clientes y registrar compras.	Presentación final: Preparar slides para justificar diseño y demostrar funcionalidades.	REVISION: Base de datos (SQL), Diseño UI (Java – Clientes), Pruebas de datos, Presentación final.	
JOSE	Configurar repositorio Git: Crear repo en GitHub y crear las ramas dev, pre, pro para el control de versiones.	Implementar endpoints (Productos): APIs para: GET /productos, POST /productos, PUT /productos/{id}, DELETE /productos/{id}.	Validar datos en frontend: Implementar reglas de validación (ej. stock no negativo).	Pruebas unitarias/integración: frontend, Reportar bugs en GitHub.	REVISION: Backend (APIs CR – Productos), Validación frontend, Pruebas integración.	
MARIO	Diseñar diagrama de arquitectura: Búsqueda inicial de la arquitectura Cliente-Servidor (APIs RESTful).	Integrar control de stock: Lógica para actualizar stock al realizar ventas (ej. PATCH /productos/{id}/stock).	Diseñar UI (Productos): Pantallas para listar, agregar, editar y eliminar productos (CRUD).	Pruebas unitarias/integración: BD, Reportar bugs en GitHub.	REVISION: Diseño UI (Java – Productos), Lógica de stock, Diagrama de arquitectura inicial.	
BRYAN	BASE DE DATOS: Creación de la base de datos en sql server.	Implementar endpoints (Clientes): APIs para: GET /clientes, POST /clientes, PUT /clientes/{id}, registro-compras.	Conectar frontend con APIs: Usar httpURLConnector o librerías como Retrofit para consumir endpoints.	Pruebas unitarias/integración: backend, Reportar bugs en GitHub.	REVISION: Backend (APIs CR – Clientes), Conexión frontend-APIs, Pruebas Postman.	
OSBELI	Documentar plan inicial: Redactar objetivos, alcance y herramientas en la documentación.	Documentación: Se crean los casos de uso y los diagramas de casos de uso.	Documenta avances: Se documentan los avances y los logros sobre las conexiones exitosas.		REVISION: Documentación General, UML, manual, casos de uso.	
HERRAMIENTAS A UTILIZAR		DESCRIPCION				
GIT HUB		Para el control de versiones.				
SQL SERVER		Motor de base de datos.				
VISUAL STUDIO		Programación de las APIs, java.				
POSTMAN		Gestor y consumidor de APIs.				
APACHE NETBEAS-JAVA		Entorno de frontend y consumidor de APIs.				

PRESUPUESTO FINAL

Categoría	Detalle	días trabajados	horas trabajadas	Tarifa / Costo Unitario (Q)	Subtotal (Q)
Costos de Personal	José (Backend/Frontend)	10 días	55	Q 50.00	Q 2,750.00
	Bryan (Frontend/DBA)	15 días	82	Q 50.00	Q 4,100.00
	Mario (Frontend/Backend)	30 días	165	Q 50.00	Q 8,250.00
	Osbeli (Analista/Documentador)	30 días	165	Q 50.00	Q 8,250.00
	César (UI/Documentador)	25 días	137	Q 50.00	Q 6,850.00
Infraestructura	Sin costos (uso de herramientas gratuitas)	-		-	Q -
Costos Adicionales	Computadoras y equipos	-		-	Q 3,000.00
	Internet	2 meses		800 Q/mes	Q 1,600.00
	Electricidad	2 meses		150 Q/mes	Q 300.00
Gastos extras	10% sobre costos personales + adicionales	-		-	Q 3,703.00
TOTAL DEL PROYECTO					Q 38,803.00