



Universidad Nacional Del Nordeste

Facultad de Ciencias Exactas y Naturales y  
Agrimensura

Ingeniería del Software II

**“Sistema de Administración de una Tienda de Juegos de  
Mesa”**

**Profesores:**

Lic. Ferraro, María de los Ángeles

Lic. Gómez Solís, Laura

Dr. Carruthers, Juan Andrés

**Alumnos:**

Marcori, Joel Jesús

Parras, Alan Emanuel

Piazza, Giovanni Oscar

Pintos, Julio César

**Año Lectivo:** 2025

**Grupo:** 7

## 1. Tabla de contenido

1.	Tabla de contenido	1
2.	Índice de Figuras	3
3.	Índice de Tablas	4
1.	Capítulo I: Introducción	5
1.1	Breve estado del arte:	5
1.2	Objetivo(s):	5
1.3	Fundamentación:	6
2.	Capítulo II: Metodología	7
2.1	Ciclo de vida del sistema:	7
	¿Qué es SCRUM?	7
2.2	Método de educación de requerimientos:	9
2.2.1	Brainstorming	9
2.2.2	Casos de Uso	10
2.2.3	Análisis de Factores Críticos	10
2.3	Arquitectura	11
2.4	Especificación de requerimientos de software:	12
2.4.1	Introducción	12
2.4.2	Descripción general	13
2.5	Análisis de Riesgos:	16
2.6	Planificación de Proyecto:	19
	Historias de Usuario	19
	Pila de Producto (Backlog):	20
	Sprints	21
	Calendarización del proyecto	23
2.7	Modelado del Sistema	25
2.7.1	Diagramas de Casos de uso	26
2.7.2	Conversaciones	30
2.7.3	Diagramas de Secuencia	34

2.7.4 Contratos de Operaciones Críticas	40
2.7.5 Diagrama de clases	43
2.7.6. Diagrama de Entidad Relación	44
<b>2.8 Estimación de Costos</b>	<b>44</b>
<b>3. Capítulo III: Herramientas</b>	<b>46</b>
<b>4. Resultados</b>	<b>47</b>
4.1. Esquema de Colores Definido	47
4.2. Pantalla de Inicio de Sesión	48
4.3. Pantalla de Bienvenida con todos los botones disponibles	49
<b>5. Conclusiones</b>	<b>50</b>
<b>Bibliografía</b>	<b>51</b>
<b>Anexos</b>	<b>52</b>

## 2. Índice de Figuras

Fig. 2.1. 1. Partes del Framework SCRUM	8
Fig. 2.1. 2. Gráfico del Framework SCRUM	9
Fig. 2.6. 1. Vista del Sprint 1 en Trello	21
Fig. 2.6. 2. Vista del Sprint 2 en Trello	22
Fig. 2.6. 3. Diagrama de Gantt correspondiente a la planificación del proyecto	25
Fig. 2.7. 1. Diagrama de Casos de Uso General	26
Fig. 2.7. 2. Diagrama de Casos de Uso del Administrador	27
Fig. 2.7. 3. Diagrama de Casos de Uso del Analista de Ventas	28
Fig. 2.7. 4. Diagrama de Casos de Uso del Vendedor	29
Fig. 2.7. 5. Diagrama de Casos de Uso del DBA	29
Fig. 2.7. 6. Diagrama de Secuencia de Registrar Ventas – Curso Normal	34
Fig. 2.7. 7. Diagrama de Secuencia de Registrar Ventas – Curso Alternativo	35
Fig. 2.6. 8. Diagrama de Secuencia de Agregar Productos – Curso Normal	35
Fig. 2.7. 9. Diagrama de Secuencia de Agregar Productos – Curso Alternativo	36
Fig. 2.7. 10. Diagrama de Secuencia de Buscar Ventas – Curso Normal y Alternativo	37
Fig. 2.7. 11. Diagrama de Secuencia de Buscar Productos – Curso Normal y Alternativo	38
Fig. 2.7. 12. Diagrama de Secuencia de Buscar Empleados – Curso Normal y Alternativo	39
Fig. 2.7. 13. Diagrama de Secuencia de Ver Estadísticas	40
Fig. 2.7. 14. Diagrama de Clases	43
Fig. 2.7. 15. Diagrama Entidad Relación	44
Fig. 4. 1. Esquema de Colores Definido para el Sistema	47
Fig. 4. 2. Pantalla de Inicio de Sesión del Sistema	48
Fig. 4. 3. Pantalla de Bienvenida del Sistema con todas las Opciones Disponibles	49

### 3. Índice de Tablas

TABLA I .....	13
TABLA II .....	13
TABLA III .....	15
TABLA IV .....	16
TABLA V .....	17
TABLA VI .....	17
TABLA VII .....	18
TABLA VIII .....	19
TABLA IX .....	19
TABLA X .....	20
TABLA XI .....	20
TABLA XII .....	20
TABLA XIII .....	23
TABLA XIV .....	23
TABLA XV .....	30
TABLA XVI .....	31
TABLA XVII .....	31
TABLA XVIII .....	32
TABLA XIX .....	32
TABLA XX .....	33
TABLA XXI .....	40
TABLA XXII .....	41
TABLA XXIII .....	42

## **1. Capítulo I: Introducción**

Se propone diseñar un sistema de administración de tiendas de juegos de mesa que aborde las necesidades actuales del sector y aproveche las últimas tecnologías disponibles. Se pretende explorar diferentes funcionalidades que podrían mejorar la eficiencia operativa y la satisfacción del cliente.

Se busca identificar los requisitos clave para un sistema efectivo y evaluar su impacto en la gestión diaria. Al abordar estas cuestiones, este estudio busca contribuir al conocimiento en el campo de la gestión y proporcionar funcionalidades valiosas para profesionales y usuarios en el área.

Este documento servirá como base de comunicación entre los desarrolladores, los usuarios y los stakeholders del sistema.

### **1.1 Breve estado del arte:**

La gestión de inventarios, ventas y personal ha evolucionado considerablemente en los últimos años, impulsado por avances tecnológicos y las preferencias de los usuarios.

Actualmente existen soluciones genéricas como Odoo, Lightspeed o Square POS, las cuales ofrecen funcionalidades robustas pero que, en muchos casos, resultan excesivamente complejas o costosas para negocios especializados o de nicho como las tiendas de juegos de mesa.

La necesidad de herramientas personalizadas, escalables y adaptables al crecimiento del negocio sigue siendo un área de oportunidad.

Este proyecto busca desarrollar una aplicación enfocada en la simplicidad de uso, eficiencia operativa y adaptabilidad, considerando las particularidades de un mercado nicho donde la relación entre producto, cliente y comunidad es fundamental.

### **1.2 Objetivo(s):**

El objetivo principal es desarrollar una aplicación de escritorio para la gestión integral de ventas, inventario y empleados en una tienda de juegos de mesa, con capacidad de escalabilidad y usabilidad optimizada para usuarios administrativos y operativos.

Los objetivos específicos son los siguientes:

- Implementar funcionalidades de alta prioridad como gestión de productos, gestión de ventas y gestión de personal.
- Proporcionar alertas inteligentes relacionadas con el stock de productos.
- Facilitar la visualización y análisis de estadísticas de ventas.

- Desarrollar una arquitectura modular que permita futuras expansiones (por ejemplo: integración de sucursales, sistema de recompensas, entre otras).

### **1.3 Fundamentación:**

El crecimiento de la industria de los juegos de mesa y su reciente popularización ha generado la necesidad de contar con sistemas administrativos que se adapten a sus dinámicas comerciales específicas: amplios catálogos de productos con ciclos de vida variados, y cambios constantes en la demanda.

La aplicación propuesta no solo permite una mejora en la eficiencia operativa diaria, sino que también sienta las bases para la toma de decisiones estratégicas a través de la visualización de estadísticas de ventas.

Además, la flexibilidad de la arquitectura permitirá a la tienda escalar su operación conforme crezca, ya sea abriendo nuevas sucursales o integrando funcionalidades avanzadas.

## **2. Capítulo II: Metodología**

### **2.1 Ciclo de vida del sistema:**

El desarrollo de cualquier aplicación informática implica llevar a cabo una serie de tareas repartidas en cuatro etapas, llamadas: Análisis, Diseño, Codificación y Pruebas. Estas cuatro etapas conforman lo que se conoce como ciclo de vida de un producto de software, el cual representa las distintas fases que atravesará dicho producto durante toda su existencia. Decidimos utilizar la metodología SCRUM, debido a que es la que mejor se adapta a los requerimientos estipulados para el desarrollo de esta aplicación en particular. Este ciclo de vida permite realizar entregas parciales y funcionales del sistema de manera iterativa, priorizando las funcionalidades críticas en las primeras entregas y permitiendo mejoras continuas basadas en el feedback real del cliente.

Cada incremento será evaluado mediante pruebas internas y revisión de requerimientos, asegurando calidad progresiva y adaptación al cambio.

#### **¿Qué es SCRUM?**

SCRUM es un marco de trabajo ágil ampliamente utilizado en el desarrollo de software que se centra en la colaboración, la flexibilidad y la entrega continua de productos de calidad. En SCRUM, los equipos trabajan en iteraciones cortas llamadas "sprints", generalmente de 2 a 4 semanas, donde se planifican, desarrollan, prueban y entregan incrementos del producto.

El proceso está impulsado por roles clave como el Scrum Master, responsable de facilitar el proceso y eliminar obstáculos, y el Product Owner, encargado de definir las prioridades del backlog y asegurar el valor del producto.

Una de las características distintivas de SCRUM es su enfoque en la retroalimentación continua y la adaptación, permitiendo a los equipos ajustar sus estrategias y prioridades en cada sprint. Esto fomenta la transparencia, la comunicación efectiva y la mejora continua del proceso de desarrollo, lo que resulta en una mayor eficiencia, calidad y satisfacción tanto para el equipo, como para los clientes.

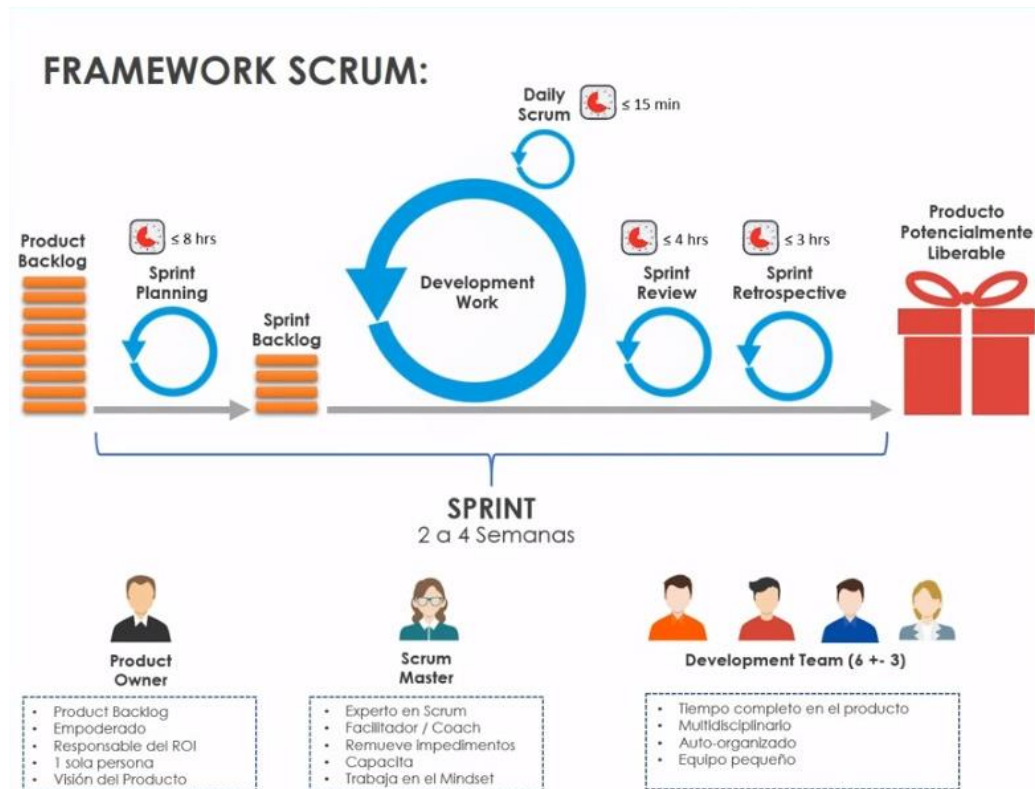




**Fig. 2.1. 1.** Partes del Framework SCRUM

Como podemos observar en la **¡Error! No se encuentra el origen de la referencia.** SCRUM se divide en 4 eventos o etapas clave que guían el proceso de desarrollo de principio a fin:

- **Planificación del Sprint:** En esta etapa, el equipo de desarrollo, junto con el Product Owner, selecciona las tareas más importantes del backlog para el próximo sprint. Se define el objetivo del sprint y se establece un plan detallado para lograrlo.
- **SCRUM Diario:** Es una reunión diaria breve y enfocada. Su objetivo principal es sincronizar al equipo de desarrollo sobre el progreso del trabajo y coordinar las actividades del día.
- **Revisión del Sprint:** Al finalizar el sprint, el equipo realiza una revisión para mostrar el trabajo completado al Product Owner y otros stakeholders. Se recopila retroalimentación y se evalúa si se cumplieron los objetivos del sprint.
- **Retrospectiva del Sprint:** Esta etapa involucra una reunión interna del equipo para reflexionar sobre el sprint, identificar qué funcionó bien y qué se puede mejorar. Se establecen acciones para implementar mejoras en el próximo sprint.



**Fig. 2.1. 2.** Gráfico del Framework SCRUM

Concretamente en nuestro proyecto, creamos las historias de usuarios con la información obtenida en la educación de requerimientos (estas se encuentran en *Historias de Usuario*), dichas historias de usuarios, se agruparán en sprints de 3 meses, con reuniones semanales para verificar los avances en el proyecto (menores a 2 horas), adicionales a las reuniones diarias (menores a 15 minutos).

## 2.2 Método de educación de requerimientos:

Para capturar los requerimientos del sistema de manera adecuada, se seleccionó un enfoque combinado que aprovecha distintas técnicas de educación, considerando las características del proyecto, el tamaño del equipo y las necesidades del cliente:

### 2.2.1 Brainstorming

En la fase inicial, se realizaron sesiones de **brainstorming (lluvia de ideas)** con los miembros del equipo de desarrollo.

El objetivo fue generar de manera rápida un conjunto amplio de ideas sobre funcionalidades esperadas, problemas actuales en el contexto de la gestión del negocio, y oportunidades de mejora.

Se priorizó que cada integrante aportara desde su perspectiva, ya sea técnica o de usuario final, fomentando un entorno de creatividad libre, pero manteniendo el foco en el contexto de una tienda de juegos de mesa.

#### **Resultados del Brainstorming:**

- Identificación de funcionalidades básicas: gestión de productos, ventas, empleados.
- Necesidad explícita de control de stock crítico.
- Necesidad de reportes de ventas claros y visuales.
- Deseo de funcionalidades futuras: manejo de eventos, membresías, recompensas.

#### **2.2.2 Casos de Uso**

Posteriormente, se formalizaron los requerimientos a través de la creación de **Casos de Uso**. Cada Caso de Uso describe de manera precisa una interacción significativa entre el usuario y el sistema.

Esta técnica permitió organizar los requerimientos de manera coherente, facilitar la validación, y construir una base sólida para el diseño posterior (diagramas de secuencia, contratos de operaciones, etc.).

#### **2.2.3 Análisis de Factores Críticos**

Se identificaron y analizaron **Factores Críticos de Éxito** para el sistema. Estos factores son condiciones que deben cumplirse para garantizar que el sistema cumpla sus objetivos y sea aceptado por los usuarios finales.

#### **Factores Críticos Detectados:**

- **Usabilidad:** La interfaz debe ser intuitiva y simple, considerando que los empleados de la tienda podrían no tener formación técnica.

- **Velocidad de respuesta:** Especialmente en la búsqueda de productos y en el registro de ventas, donde se requiere agilidad.
- **Fiabilidad del stock:** Las alertas de stock crítico deben ser precisas para evitar pérdidas de ventas por falta de productos.
- **Escalabilidad:** La arquitectura debe permitir agregar módulos o conectar sucursales en el futuro sin necesidad de rediseño completo.
- **Integridad de los datos:** Los datos de ventas, productos y empleados deben estar correctamente protegidos y respaldados.

#### Acciones tomadas en base a los factores:

- Diseño de interfaces simplificadas.
- Planificación de optimización de consultas en bases de datos.
- Implementación de validaciones estrictas en ingreso de productos y ventas.
- Estructuración modular del sistema para permitir crecimiento progresivo.

## 2.3 Arquitectura

La arquitectura seleccionada es la de arquitectura de capas, especializada para este sistema en un modelo de 4 capas lógicas y 2 niveles físicos.

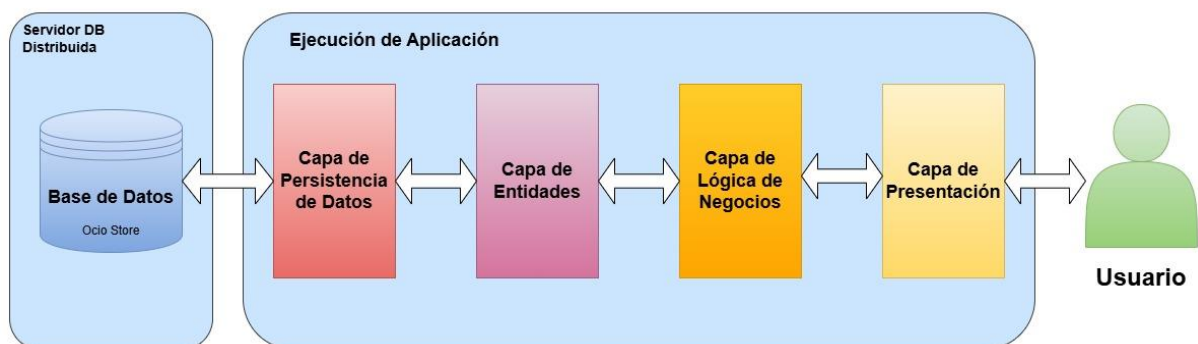


Fig. 2.3. Diagrama de la Arquitectura del Sistema

### 2.3.1. Modelo de Capas

Las capas que componen el modelo son: persistencia de datos, entidades, lógica de negocios y presentación. Las características de cada una de ellas son:

- Capa de Presentación: encargada de la interacción con el usuario y de mostrar las vistas y formularios para la captación de datos y eventos, o bien la impresión de los mismos.
- Capa de Lógica de Negocios: la encargada de albergar las funcionalidades orientadas al negocio y las demandas del sistema, transportando lo capturado por la capa de

presentación y adaptándolas al modelo de entidades, o bien realizando las solicitudes y llamados a la capa de entidades y persistencia de datos para luego manipular los datos y enviar los resultados a la capa de presentación.

-Capa de Entidades: sujeta al modelo de objetos relacionales del diagrama del sistema, en el cual se establecen las reglas estructurales que sigue cada dato antes de ser manipulado por la lógica de negocios o ser enviado a la base de datos por la capa de persistencia.

-Capa de Persistencia: sigue las reglas estructurales de las entidades y realiza la interacción con la base de datos alojada de manera distribuida.

### **2.3.2. Modelo de Niveles**

-Nivel Físico de Aplicación: es el componente del sistema donde se alojará la ejecución del sistema por el usuario.

-Nivel Físico de Datos: es el servidor que alberga la base de datos.

## **2.4 Especificación de requerimientos de software:**

### **2.4.1 Introducción**

#### **Propósito**

- Las especificaciones estipuladas en el presente tienen como propósito definir las especificaciones funcionales, no funcionales para el desarrollo de un Sistema de Digitalización y Automatización de una entidad, estando destinadas a encargados del proyecto, desarrolladores, analistas funcionales, encargados de pruebas, administradores, administrativos, docentes y usuarios finales.

#### **2.4.1.1 Alcance:**

- Será un sistema de administración de un local de ventas de juegos de mesa que permitirá gestionar a los usuarios participantes, en este caso serían el administrador, administrador de base de datos, vendedores y analistas de ventas.
- El sistema estará diseñado para uso en computadoras de escritorio en el entorno de la tienda, pero podrá escalar a una arquitectura más distribuida si el negocio crece.

#### **2.4.1.2 Definiciones, acrónimos y abreviaturas:**

- RF: Requerimiento funcional
- RNF: Requerimiento no funcional
- ERS: Especificación de requisitos de software
- HU: Historias de Usuario
- CRUD: Create, Read, Update, Delete.

- Stock Crítico: Nivel de stock por debajo del cual debe generarse una alerta para reposición.

#### 2.4.1.3 Referencias:

**TABLA I**  
REFERENCIAS

Referencias	Título
IEEE	Standard IEEE 830 - 1998
Licenciatura en Sistema de la Información.	PFC (Proyecto final de Carrera)
Ingeniería del Software I	Plantillas de clase y bibliografía de la materia.
Ingeniería del Software II	Plantillas de clase y bibliografía de la materia.

#### 2.4.2 Descripción general

##### 2.4.2.1 Perspectiva del producto

La aplicación será un sistema independiente de escritorio desarrollado en C# .NET utilizando Windows Forms, con una arquitectura en capas.

##### 2.4.2.2 Funcionalidad del producto

Las funciones que permitirá el Sistema son las siguientes:

- Registrar, modificar y eliminar productos.
- Registrar y gestionar ventas.
- Administrar información de empleados.
- Visualizar estadísticas de ventas de productos.
- Controlar niveles de stock, alertando en casos críticos.
- Alertar al gerente/administrador sobre stock crítico cuando corresponda.

##### 2.4.2.3 Características Generales de los usuarios

**TABLA II**  
CARACTERÍSTICAS DE LOS USUARIOS

<b>Tipo de usuario</b>	<b>Formación (nivel educativo)</b>	<b>Experiencia y Especialización técnica</b>
Administrador	Secundario completo	Conocimiento básico en informática, Conocimiento en Gestión de comercios y Sistemas Informáticos de pequeña-mediana escala.
Analista de Ventas	Secundario Completo	Conocimientos básicos de aplicaciones de escritorio y ofimática, análisis de mercado, estadísticas y sistemas informáticos de pequeña-mediana escala.
Administrador de la Base de Datos	Secundario Completo. alguna certificación en manejo de Bases de Datos Relacionales.	Conocimientos intermedios de aplicaciones de escritorio y ofimática, y sistemas informáticos de pequeña-mediana escala. Conocimiento de Bases de Datos y manejo de Microsoft SQL Server.
Vendedor	Secundario completo	Conocimientos básicos de aplicaciones de escritorio y ofimática.

#### **2.4.2.4 Restricciones**

Para el desarrollo ágil del software, se seguirá las bases de la Metodología Scrum para un desarrollo evolutivo en el equipo. Además, se mantendrá un control de versionado mediante el uso de un repositorio en GitHub.

El sistema utilizará Microsoft SQL Server Management Studio 20 para desarrollar la base de datos con estructura SQL. El diseño y desarrollo de la aplicación se hará utilizando C# .NET utilizando Windows Forms, en el IDE Visual Studio.

Escalabilidad planeada mediante arquitectura modular (capas).

El sistema debe funcionar en equipos Windows 10 o superior.

#### **2.4.2.5 Suposiciones y dependencias**

##### **Suposiciones:**

**Estabilidad de la Base de Datos SQL:** Se asume que la base de datos SQL incluida en Microsoft SQL Server será estable y compatible con el sistema durante todo el desarrollo.

**Disponibilidad del Entorno de Desarrollo:** Se considera que Visual Studio y las herramientas asociadas estarán disponibles y serán compatibles con las versiones de C# y .NET utilizadas a lo largo del ciclo de vida del proyecto.

**Compatibilidad del Sistema Operativo:** Se asume que la aplicación será compatible con las versiones de Windows especificadas en los requisitos del proyecto. Los cambios en las versiones o actualizaciones del sistema operativo podrían requerir ajustes en la aplicación.

**Requerimientos de Hardware:** Se asume que los usuarios finales contarán con hardware que cumple con los requisitos mínimos necesarios para ejecutar la aplicación de manera eficiente. Cambios significativos en los requerimientos de hardware pueden afectar el rendimiento.

**Disponibilidad del Control de Versiones:** Se asume que GitHub estará accesible para la gestión del código fuente y el control de versiones. Problemas con el acceso a GitHub pueden impactar el desarrollo y la integración continua.

### 2.4.3 Requisitos específicos

#### 2.4.3.1 Requisitos funcionales

**TABLA III**

DESCRIPCIÓN DE LOS REQUISITOS FUNCIONALES

N.º	Descripción de requisitos
RF#1	El sistema debe permitir registrar nuevos productos validando que no estén duplicados.
RF#2	El sistema debe permitir registrar una venta asociada a uno o más productos.
RF#3	El sistema debe reducir automáticamente el stock de un producto tras una venta.
RF#4	El sistema debe alertar al analista cuando un producto esté en stock crítico.
RF#5	El analista debe poder alertar al gerente sobre productos en stock crítico desde la vista de stock.
RF#6	El sistema debe mostrar estadísticas de ventas por producto en gráficos o tablas.
RF#7	El gerente debe poder agregar, modificar y eliminar empleados.



RF#8	El sistema debe permitir buscar productos y empleados utilizando filtros dinámicos.
RF#9	El sistema debe registrar auditoría básica de acciones críticas (altas/bajas/modificaciones).

### 2.4.3.2 Requisitos no funcionales

#### 2.4.3.2.1 Requisitos del producto

**TABLA IV**

DESCRIPCIÓN DE REQUISITOS NO FUNCIONALES

N.º	Descripción de requisitos
RNF#1	El sistema deberá realizarse bajo la especificación de requerimientos de software según la norma IEEE 830
RNF#2	El sistema debe ser fácil de usar, con una curva de aprendizaje de menos de una hora.
RNF#3	El sistema debe validar todas las entradas del usuario para prevenir errores de carga.
RNF#4	El sistema debe permitir registrar usuarios.
RNF#5	El diseño debe ser amigable, intuitivo y consistente con guías de UI/UX.
RNF#6	La base de datos debe garantizar la integridad referencial entre productos, ventas y empleados.
RNF#7	El sistema debe tener diferentes niveles de acceso para administradores, vendedores, administradores de base de datos y analistas de ventas, asegurando que cada usuario solo pueda acceder a la información y funcionalidades correspondientes a su rol.
RNF#8	El sistema debe permitir backup de la base de datos manual.

## 2.5 Análisis de Riesgos:

**TABLA V**  
IDENTIFICACIÓN DE RIESGOS

Riesgo	Tipo	Descripción
Riesgo de Personal	Proyecto	Un colaborador se integre al proyecto o lo abandone
Riesgo de Tecnología	Producto	Falta de experiencia del equipo en una tecnología específica
Riesgo de Personal	Proyecto	Problemas de comunicación entre integrantes
Riesgo de Requerimientos	Producto	Dificultades en la identificación y corrección oportuna de defectos durante el ciclo de desarrollo
Riesgo de Personal	Proyecto	Falta de compromiso o apoyo por parte de los interesados clave del proyecto
Riesgo de Tecnología	Proyecto	La tecnología fundamental sobre la que se construirá el sistema, se sustituye por nueva tecnología
Riesgo de Requerimiento	Negocio	El cliente no está satisfecho con el producto
Riesgo de Tecnología	Producto	Fallos en la conexión a la base de datos o corrupción de datos locales.

**TABLA VI**  
ANÁLISIS DE RIESGOS

Riesgo	Probabilidad	Efecto
Un colaborador se integre al proyecto o lo abandone.	Baja (10-25%)	Serio

Falta de experiencia del equipo en una tecnología específica.	Moderado (25-50%)	Tolerable
Problemas de comunicación entre integrantes.	Moderado (25-50%)	Tolerable
Dificultades en la identificación y corrección oportuna de defectos durante el ciclo de desarrollo.	Bajo (10-2%)	Tolerable
Falta de compromiso o apoyo por parte de los interesados clave del proyecto.	Moderado (25-50%)	Tolerable
La tecnología fundamental sobre la que se construirá el sistema, se sustituye por nueva tecnología.	Bajo (10-25%)	Catastrófico
El cliente no está satisfecho con el producto	Moderado (25-50%)	Catastrófico
Fallos en la conexión a la base de datos o corrupción de datos locales.	Bajo (10-25%)	Catastrófico

**TABLA VII**  
**PLANIFICACIÓN DE RIESGOS**

Riesgo	Estrategia
Cambio/s en el equipo	Asignación de roles claros y realizar evaluaciones periódicas del desempeño del equipo y brindar feedback constructivo.
Falta de experiencia en una tecnología	Brindar capacitaciones o cursos de las tecnologías a usar.
Problemas de comunicación	Promover un ambiente donde todos se sientan cómodos expresando sus ideas y preocupaciones sin temor a represalias.
Dificultades en la identificación y corrección de defectos	Aplicar ciclos cortos de desarrollo, priorización dinámica, pruebas automatizadas, revisiones de código y procesos de mejora continua.

Falta de compromiso con el proyecto	Demostrar resultados tangibles y valor continuo del proyecto a los interesados clave a lo largo del tiempo, reforzando su compromiso y apoyo.
Cambios de tecnologías	Desarrollar un plan detallado para la transición tecnológica, identificando recursos, evaluando riesgos y garantizando la continuidad del servicio para minimizar impactos negativos en las operaciones del negocio.
Disconformidad del cliente	Revisar y modificar nuevamente la documentación para subsanar las necesidades del usuario final.
Fallos en la conexión a la base de datos o corrupción de datos locales.	Desarrollar mecanismos de backup y validaciones de entrada.

## 2.6 Planificación de Proyecto:

A continuación, se detallan las Historias de Usuario, la pila de producto y la calendarización del proyecto.

### Historias de Usuario

**TABLA VIII**

HISTORIA DE USUARIO 1

<b>HU N.º 1: Registrar Venta</b>
<b>Descripción:</b> Como vendedor quiero poder registrar una nueva venta para poder caracterizarla y facturarla.
<b>Criterios de Aceptación:</b> Dado un vendedor que quiere registrar una venta cuando ingresa el/los producto/s, información de pago y fecha.

**TABLA IX**

HISTORIA DE USUARIO 2

<b>HU N.º 2: Seleccionar Forma de Pago</b>
--

<b>Descripción:</b> Como vendedor quiero poder seleccionar un método de pago para poder realizar el cobro de la venta al cliente.
<b>Criterios de Aceptación:</b> Dado un vendedor que quiere seleccionar un método de pago para efectuar el cobro de una venta.

**TABLA X**  
HISTORIA DE USUARIO 3

<b>HU N.º 3: Buscar Ventas</b>
<b>Descripción:</b> Como analista quiero poder buscar las ventas realizadas para poder realizar un informe estadístico.
<b>Criterios de Aceptación:</b> Dado un analista que quiere buscar ventas para realizar un informe de las mismas.

**TABLA XI**  
HISTORIA DE USUARIO 4

<b>HU N.º 4: Agregar Producto</b>
<b>Descripción:</b> Como administrador quiero poder agregar nuevos productos para poder llevar un registro de su situación en el negocio.
<b>Criterios de Aceptación:</b> Dado un administrador que quiere agregar productos para aumentar el inventario de su negocio.

Pila de Producto (Backlog):

**TABLA XII**  
PILA DE PRODUCTO

Id	Historia de Usuario	Prioridad	Estimación
1	Registrar venta	Alta	13
2	Seleccionar forma de pago	Media	5
3	Buscar ventas	Media	3
4	Agregar producto	Alta	13

## Sprints



Fig. 2.6. 1. Vista del Sprint 1 en Trello

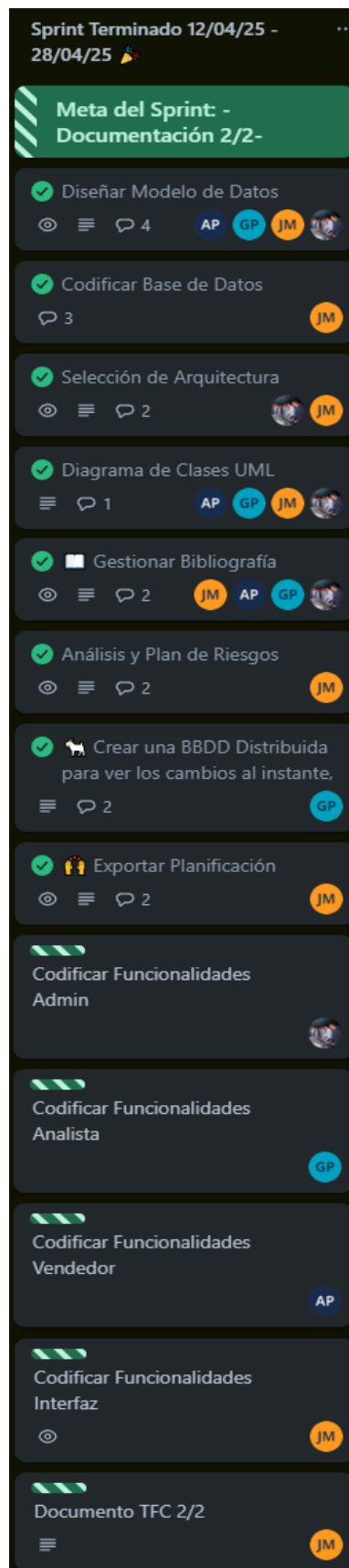


Fig. 2.6. 2. Vista del Sprint 2 en Trello

### 1er Sprint:

El primer sprint fue dedicado al análisis, diseño y modelado del sistema a desarrollar.

- Crear tablero de gestión del proyecto
- Crear Especificación de Requerimientos de Software (ERS)
- Elaborar Diagramas de Casos de Uso
- Modelar Conversaciones (Buscar, Ver Estadísticas, Registrar Venta, Agregar Producto)
- Modelar Diagramas de Secuencia (Buscar, Ver Estadísticas, Registrar Venta, Agregar Producto)
- Elaborar Contratos de Operaciones Críticas
- Redactar Documento TFC (Parte 1/2)

### 2do Sprint:

**TABLA XIII**

SEGUNDO SPRINT

Id	Descripción de la tarea	Persona responsable	Estado de la tarea
1	Registrar venta	Parras, Pintos	Pendiente
2	Seleccionar forma de pago	Piazza, Marcori	Pendiente
3	Buscar ventas	Piazza, Parras	Pendiente
4	Agregar producto	Marcori, Pintos	Pendiente

### Calendarización del proyecto

**TABLA XIV**

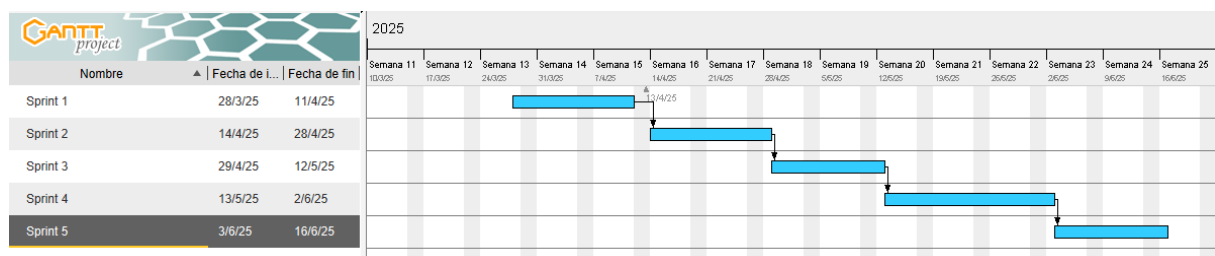
PLANIFICACIÓN DE ACTIVIDADES

Nombre	Fecha de Inicio	Fecha de Fin
Análisis	28/03/25	28/03/25
Educir Requisitos	28/03/25	28/03/25
Diseño Sprint 1: Product Backlog Planear el Sprint	29/03/25	30/03/25



Sprint 1	30/04/25	10/04/25
Testing	10/04/25	11/04/25
Diseño Sprint 2:  Refinar el Product Backlog Rever el Sprint 1 Planear el Sprint 2	12/04/25	14/04/25
Sprint 2:  HU1 HU2 HU3 HU4	14/04/25	27/04/25
Testing	28/04/25	29/04/25
Diseño Sprint 3:  Refinar el Product Backlog Rever el Sprint 2 Planear el Sprint 3	29/04/25	30/04/25
Sprint 3	30/04/25	09/05/25
Testing	10/05/25	11/05/25
Rever el Sprint	12/05/25	12/05/25
Diseño Sprint 4:  Refinar el Product Backlog Rever el Sprint 3 Planear el Sprint 4	13/05/25	14/05/25
Sprint 4	15/05/25	30/05/25
Testing	31/05/25	01/06/25

Rever el Sprint	02/06/25	02/06/25
Diseño Sprint 5:  Refinar el Product Backlog Rever el Sprint 4 Planear el Sprint 5	03/06/25	04/06/25
Sprint 5	04/06/25	14/06/25
Testing	15/06/25	16/06/25
Rever el Sprint	16/06/25	16/06/25
Entrega del Producto	17/06/25	17/06/25



**Fig. 2.6. 3.** Diagrama de Gantt correspondiente a la planificación del proyecto

## 2.7 Modelado del Sistema

A continuación, pueden observarse los distintos diagramas que hacen al modelado del sistema.

## 2.7.1 Diagramas de Casos de uso

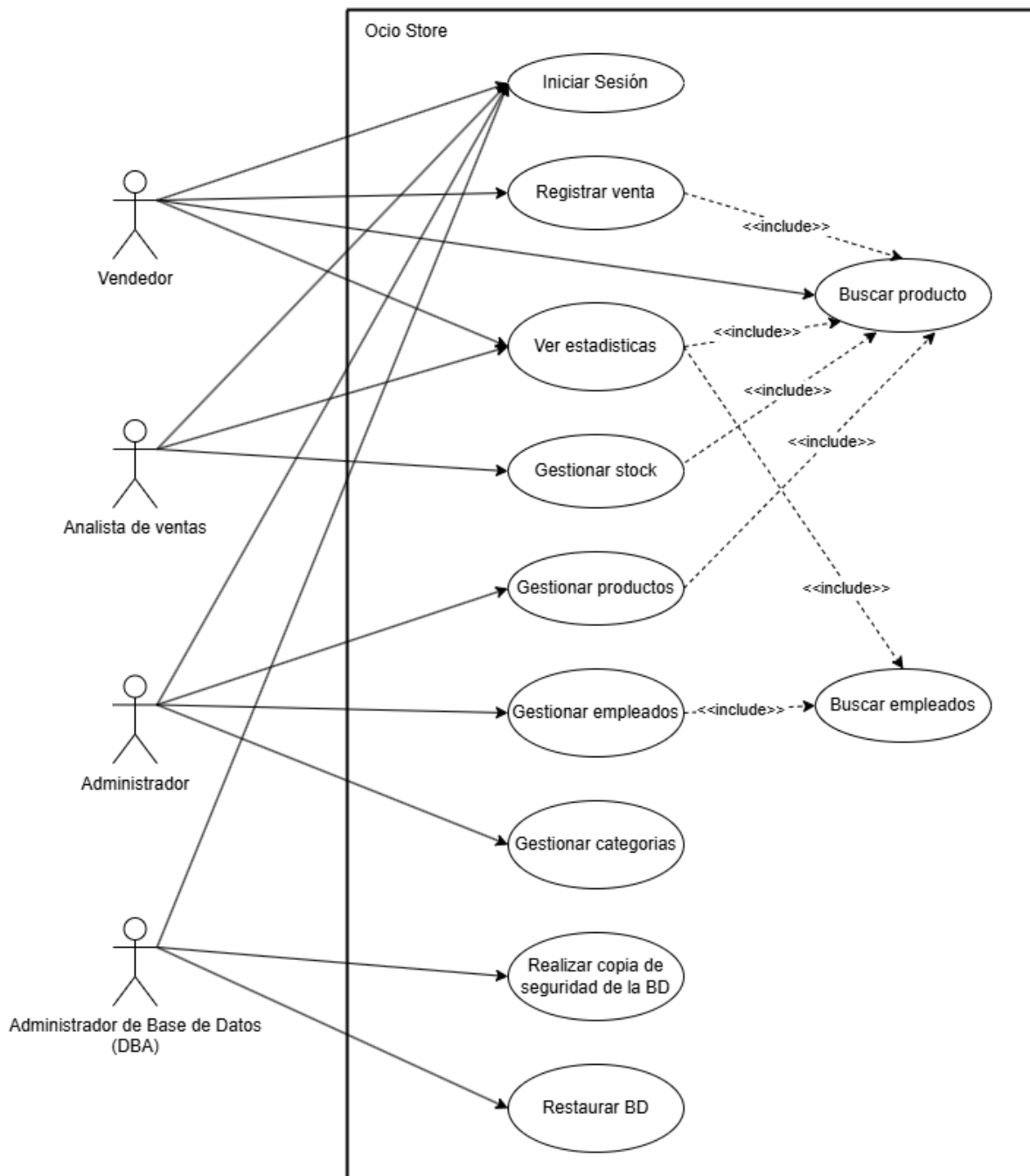
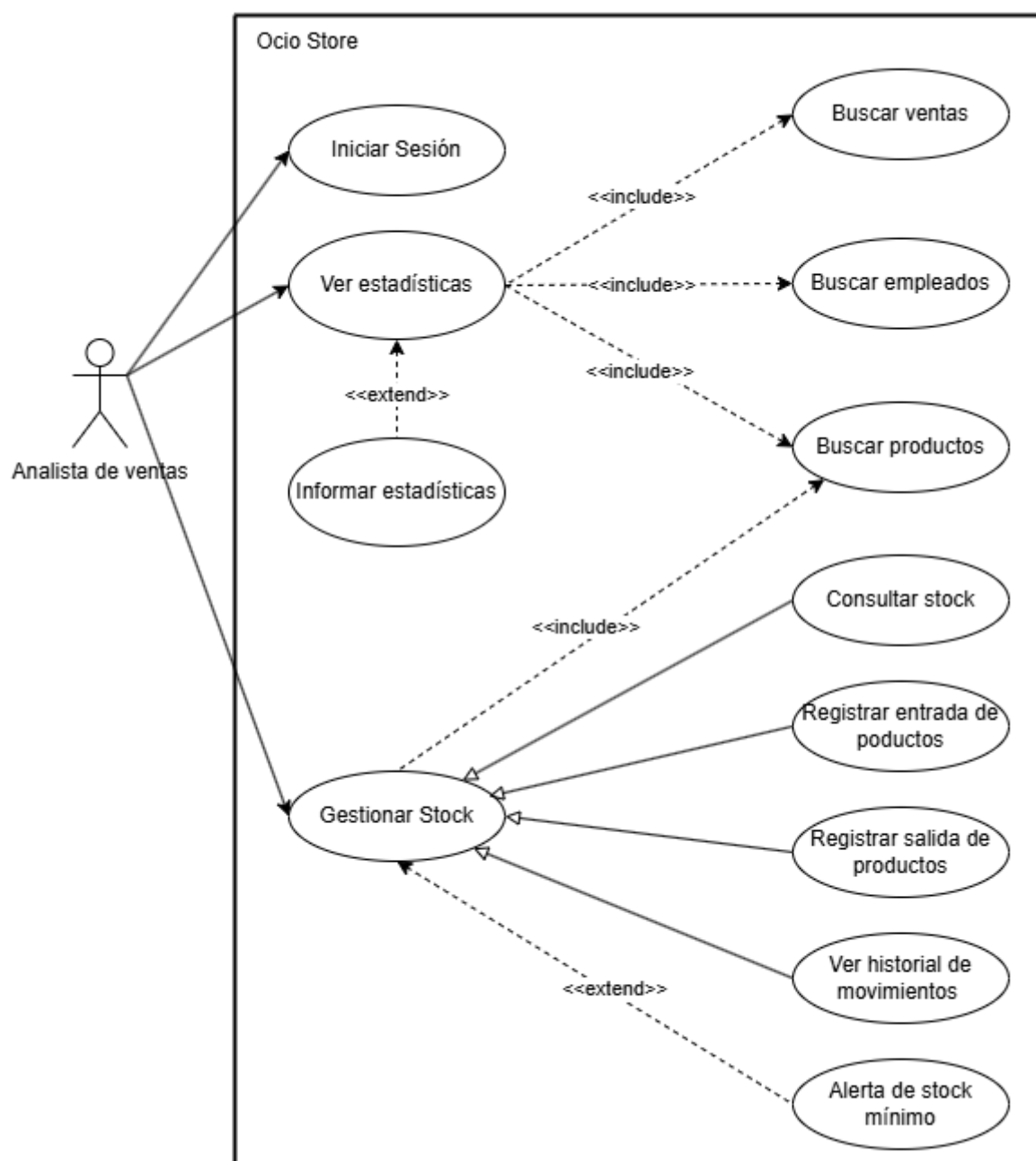


Fig. 2.7. 1. Diagrama de Casos de Uso General



Fig. 2.7. 2. Diagrama de Casos de Uso del Administrador



**Fig. 2.7. 3.** Diagrama de Casos de Uso del Analista de Ventas

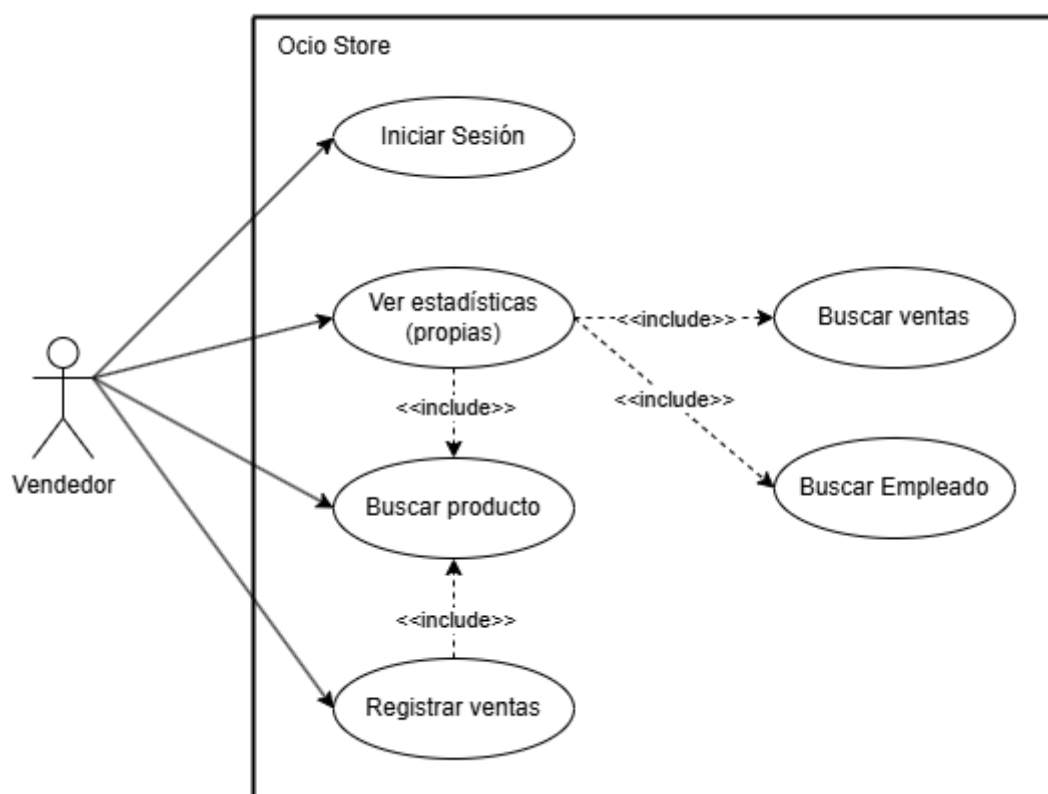


Fig. 2.7. 4. Diagrama de Casos de Uso del Vendedor

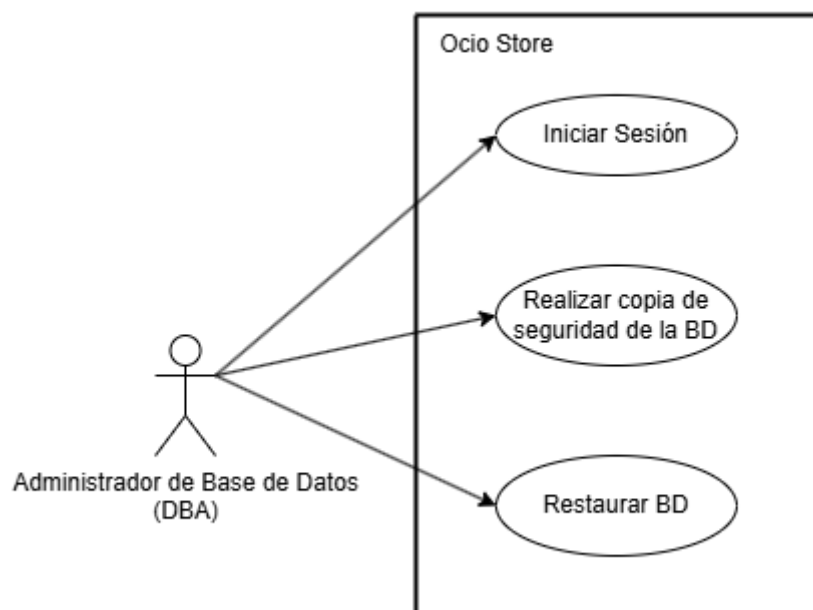


Fig. 2.7. 5. Diagrama de Casos de Uso del DBA

## 2.7.2 Conversaciones

### Conversación: 1

**Caso de uso:** Registrar ventas

**Actor:** Vendedor

**TABLA XV**  
**CONVERSACIÓN DE REGISTRAR VENTAS**

<b>Acción</b>	<b>C. Normal</b>	<b>C. Alternativo 1</b>
1. V: Inicia una nueva venta.		
2. V: Ingresar o seleccionar producto.	2.1 S: <i>Include: Caso de Uso "Buscar Productos"</i> . 2.2 Hay stock disponible. 2.3 Se agrega al detalle de venta.	2.1.1 No hay stock disponible. 2.1.2 S: Muestra el mensaje "Sin stock". 2.1.3 Vuelve a 2.
3. 3. S: Pregunta si desea agregar otro producto.	3.1 V: No desea agregar. 3.2 Continúa a 4.	3.1.1 Si desea agregar. 3.1.2 Vuelve a 2.
4. 4. S: Muestra subtotal.		
5. 5. S: Solicita forma de pago.		
6. 6. V: Selecciona forma de pago.	6.1 Método de pago válido.	6.1.1 Método de pago inválido. 6.1.2 S: Muestra el mensaje "Método de pago invalido". 6.1.3 Vuelve a 6.
7. S: Solicita confirmar la venta.	7.5.1 V: Confirma.	7.1.1 V: Cancela la venta. 7.1.2 S: muestra el mensaje "operación cancelada". 7.1.3 Continúa a 9.

8. 8. S: Registra venta e imprime comprobante.		
9. Fin del caso de uso		

## Conversación 2

**Caso de uso:** Buscar Ventas

**Actor:** Analista de Ventas

**TABLA XVI**

CONVERSACIÓN DE BUSCAR VENTAS

Acción	C. Normal	C. Alternativo
1. S: Muestra interfaz de búsqueda con filtros		
2. V: Selecciona filtros		
3. V: Realiza la búsqueda	3.1: S: Muestra las ventas según los filtros	3.1.1: S: Muestra un mensaje cuando ninguna venta coincide con los filtros 3.1.2: S: Vuelve a 2
4. Finaliza caso de uso		

## Conversación 3

**Caso de uso:** Buscar Productos

**Actor:** Analista de Ventas, Vendedor, Administrador

**TABLA XVII**

CONVERSACIÓN DE BUSCAR PRODUCTOS

Acción	C. Normal	C. Alternativo
1. S: Muestra interfaz de búsqueda con filtros		
2. V: Selecciona filtros		



3. V: Realiza la búsqueda	3.1: S: Muestra los productos según los filtros	3.1.1: S: Muestra un mensaje cuando ningún producto coincide con los filtros 3.1.2: S: Vuelve a 2
4. Finaliza caso de uso		

#### Conversación 4

**Caso de uso:** Buscar Empleados

**Actor:** Analista de Ventas, Administrador

**TABLA XVIII**

CONVERSACIÓN DE BUSCAR EMPLEADOS

Acción	C. Normal	C. Alternativo
1. S: Muestra interfaz de búsqueda con filtros		
2. V: Selecciona filtros		
3. V: Realiza la búsqueda	3.1: S: Muestra los empleados según los filtros	3.1.1: S: Muestra un mensaje cuando ningún empleado coincide con los filtros 3.1.2: S: Vuelve a 2
4. Finaliza caso de uso		

#### Conversación 5

**Caso de uso:** Ver estadísticas de ventas

**Actor:** Analista de Ventas

**TABLA XIX**

CONVERSACIÓN DE VER ESTADÍSTICAS DE VENTAS

Acción	C. Normal	C. Alternativo
--------	-----------	----------------

1. S: Muestra opciones de filtros para ver estadísticas		
2. V: Selecciona un producto	2.1: S: <i>Include: Caso de Uso "Buscar Productos"</i>	
3. S: Procesa estadísticas de ventas	3.1: S: <i>Include: Caso de Uso "Buscar Ventas"</i>	
4. S: Muestra estadísticas de venta		
5. Finaliza caso de uso		

## Conversación 6

**Caso de uso:** Agregar Productos

**Actor:** Administrador

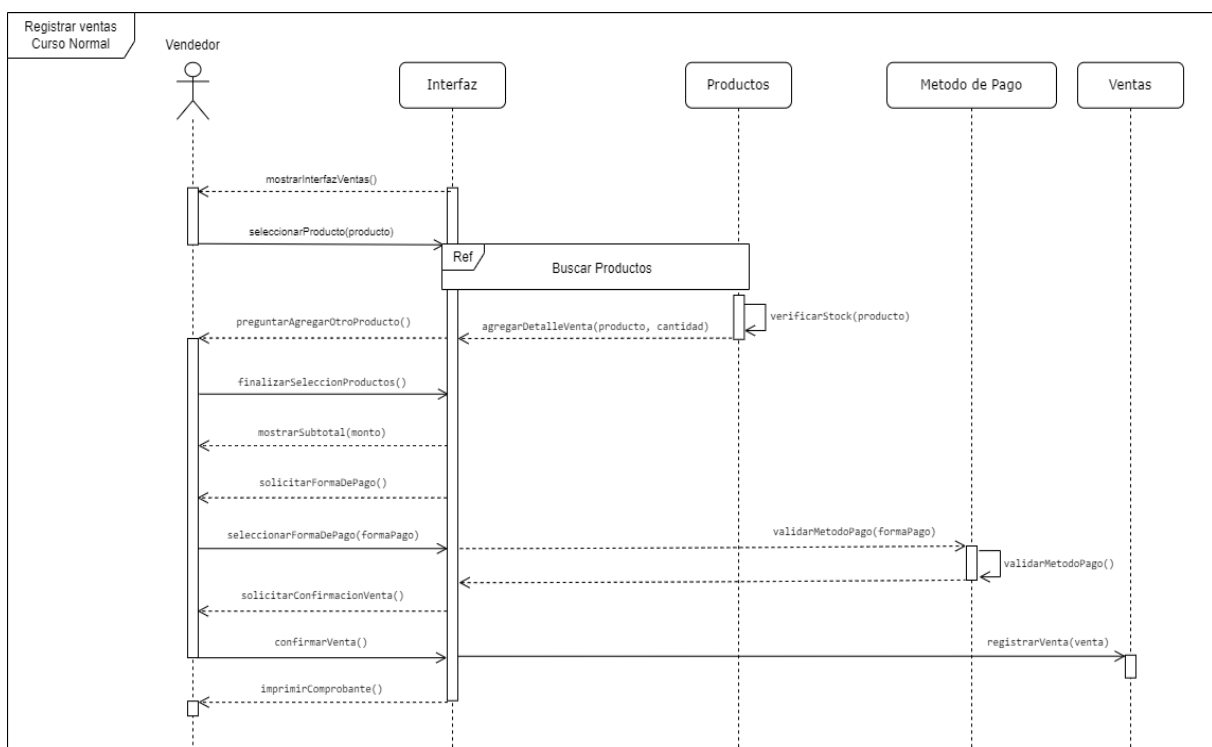
**TABLA XX**

CONVERSACIÓN DE AGREGAR PRODUCTOS

<b>Acción</b>	<b>C. Normal</b>	<b>C. Alternativo 1</b>	<b>C. Alternativo 2</b>
El Sistema muestra la interfaz			
El usuario ingresa la información del producto (nombre, precio de lista, stock, categoría)			
El sistema verifica si el producto ya existe	3.1.1. El sistema permite agregar el producto 3.1.2. El sistema muestra la	3.2.1. El sistema muestra un mensaje "el producto ya existe" 3.2.2. Vuelve a 2	3.3.1. El sistema muestra un mensaje "producto tiene información necesaria incompleta" 3.3.2. Vuelve a 2

	información del producto		
El sistema permite agregar otro producto	4.1. El usuario decide no ingresar otro producto	4.2.1. El usuario decide agregar otro producto 4.2.2. Vuelve a 2	
Finaliza caso de uso			

### 2.7.3 Diagramas de Secuencia



**Fig. 2.7. 6.** Diagrama de Secuencia de Registrar Ventas – Curso Normal

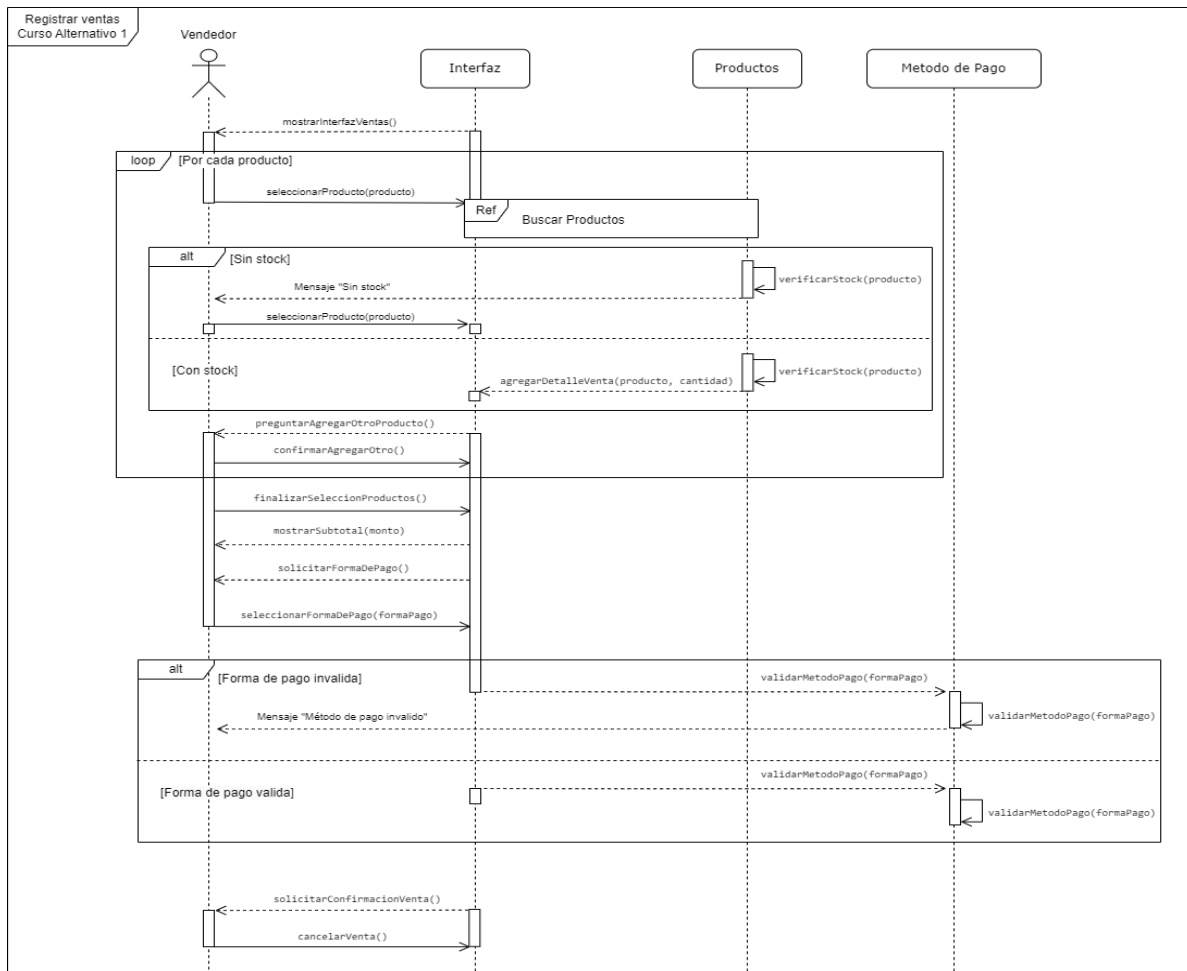


Fig. 2.7. 7. Diagrama de Secuencia de Registrar Ventas – Curso Alternativo

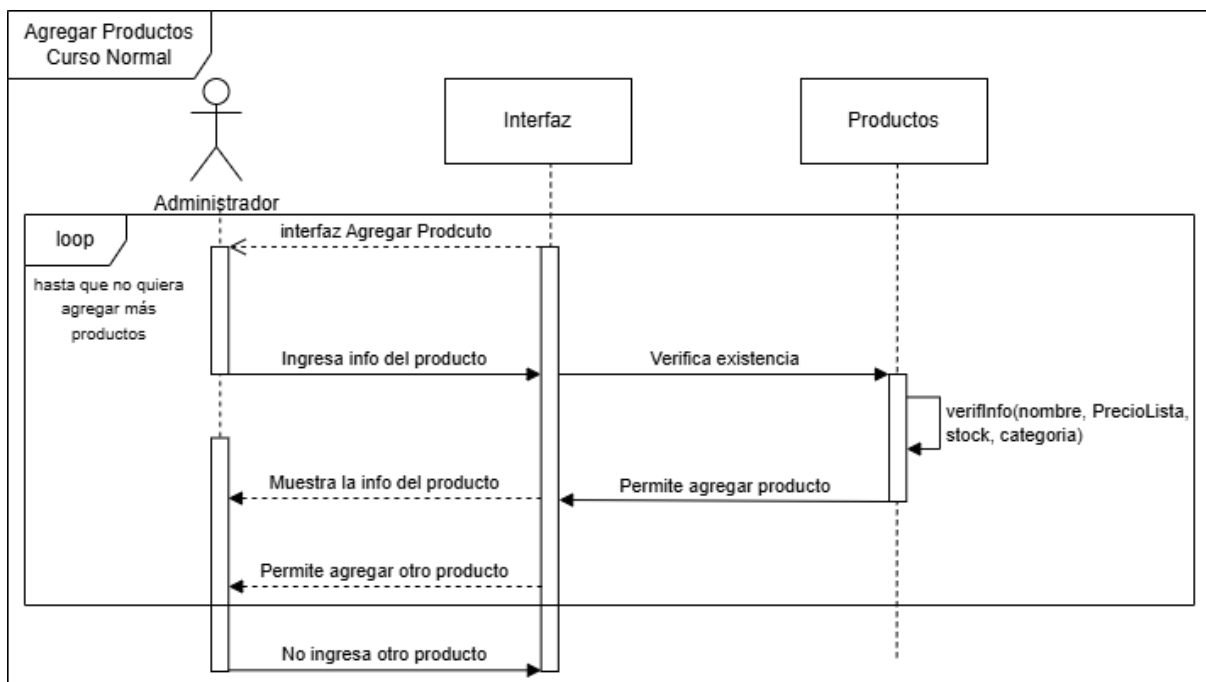
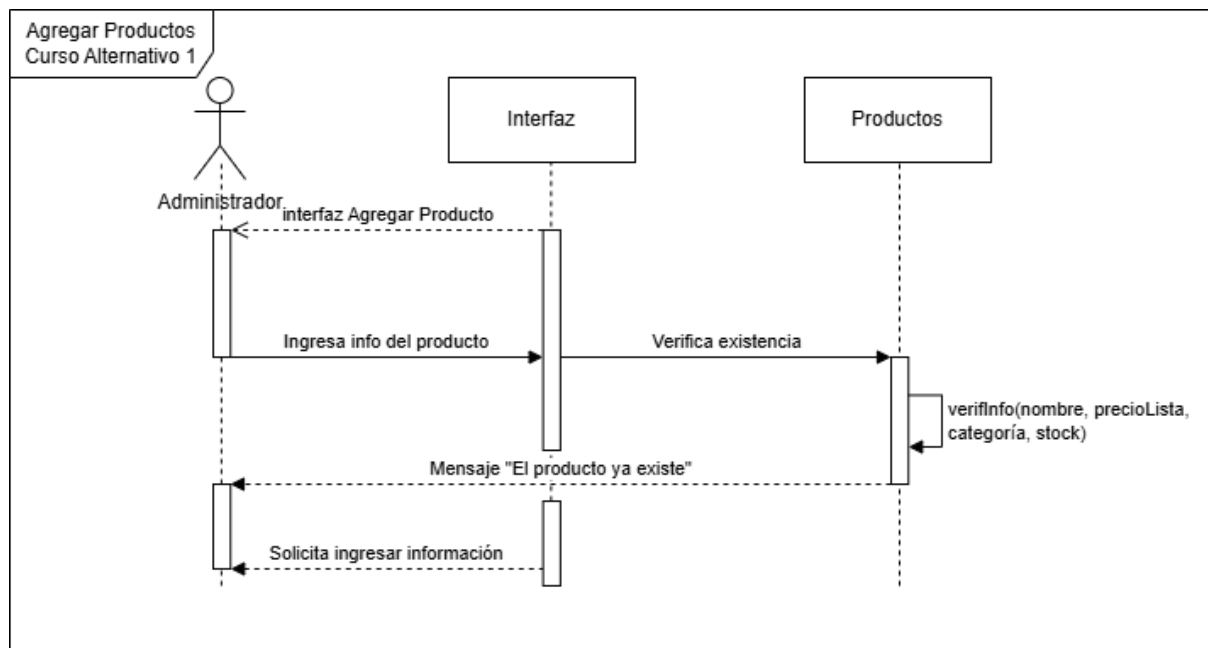


Fig. 2.6. 8. Diagrama de Secuencia de Agregar Productos – Curso Normal



**Fig. 2.7. 9.** Diagrama de Secuencia de Agregar Productos – Curso Alternativo

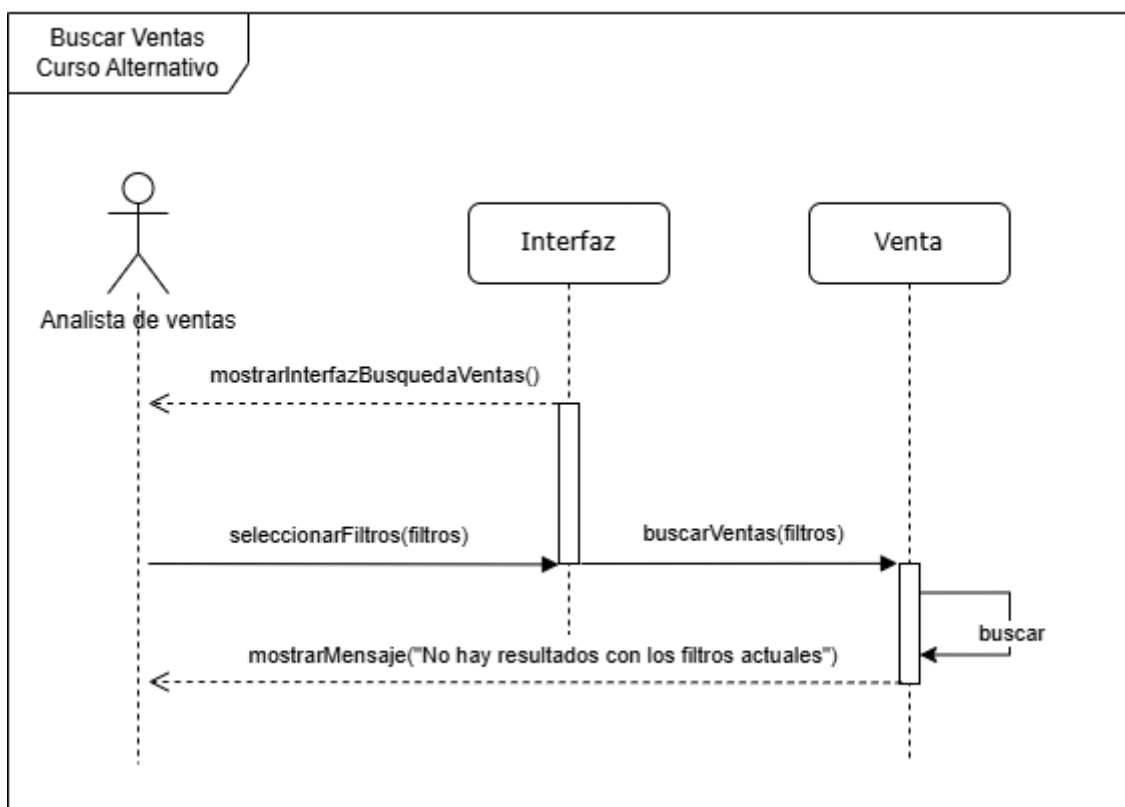
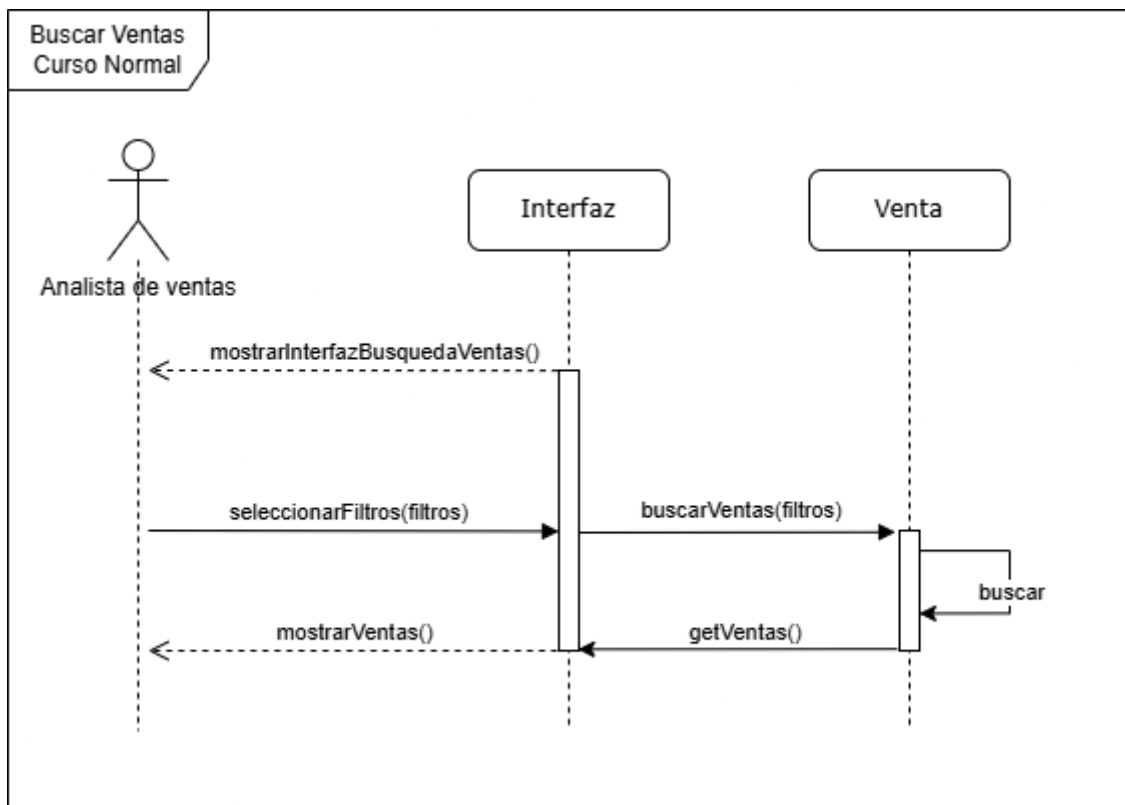
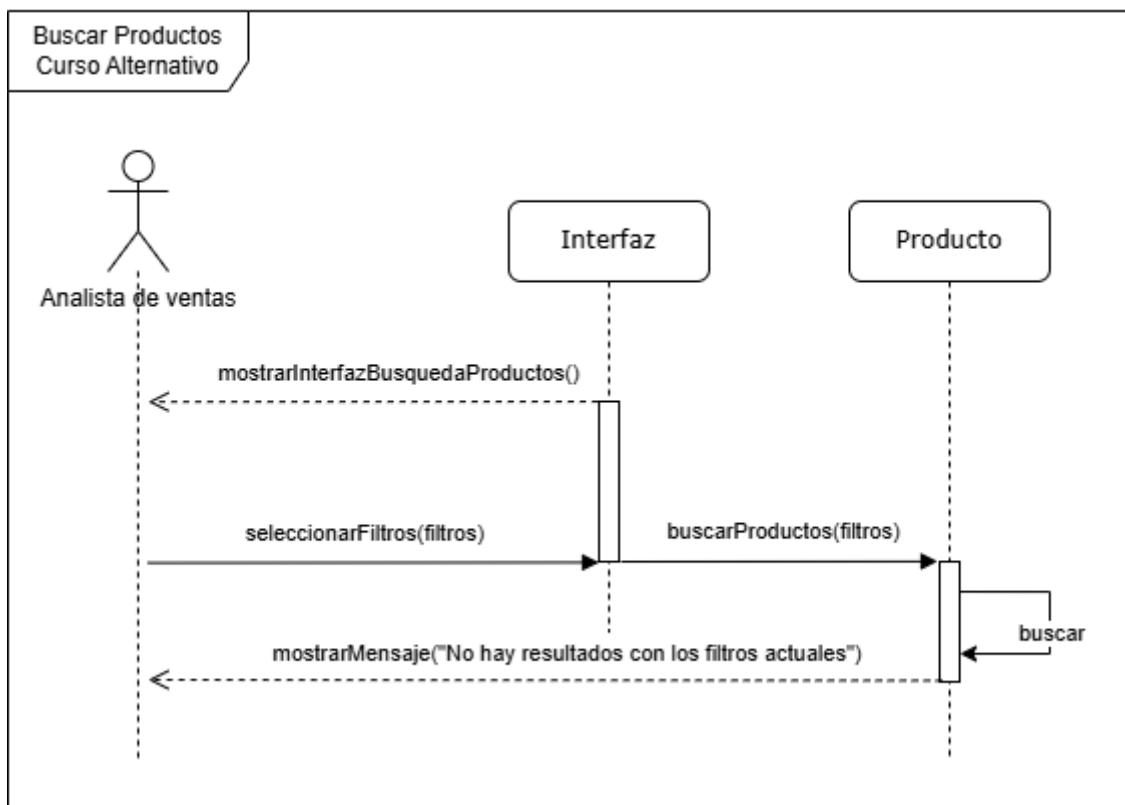
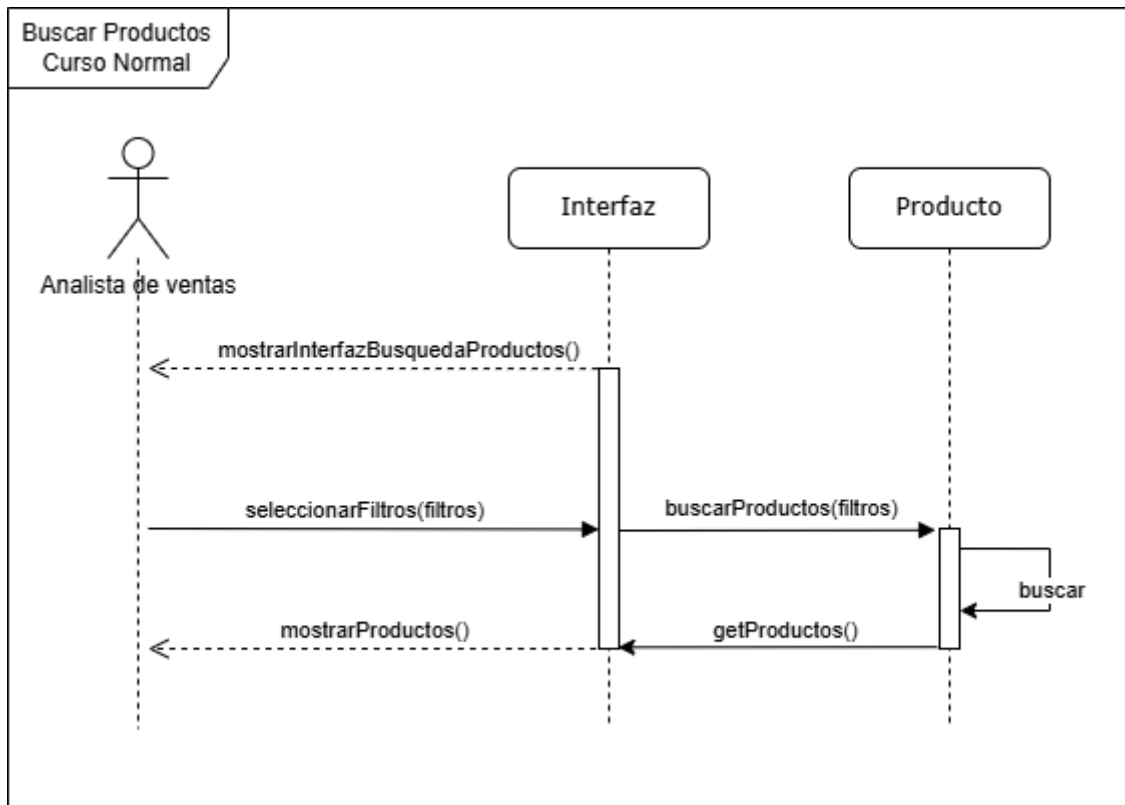
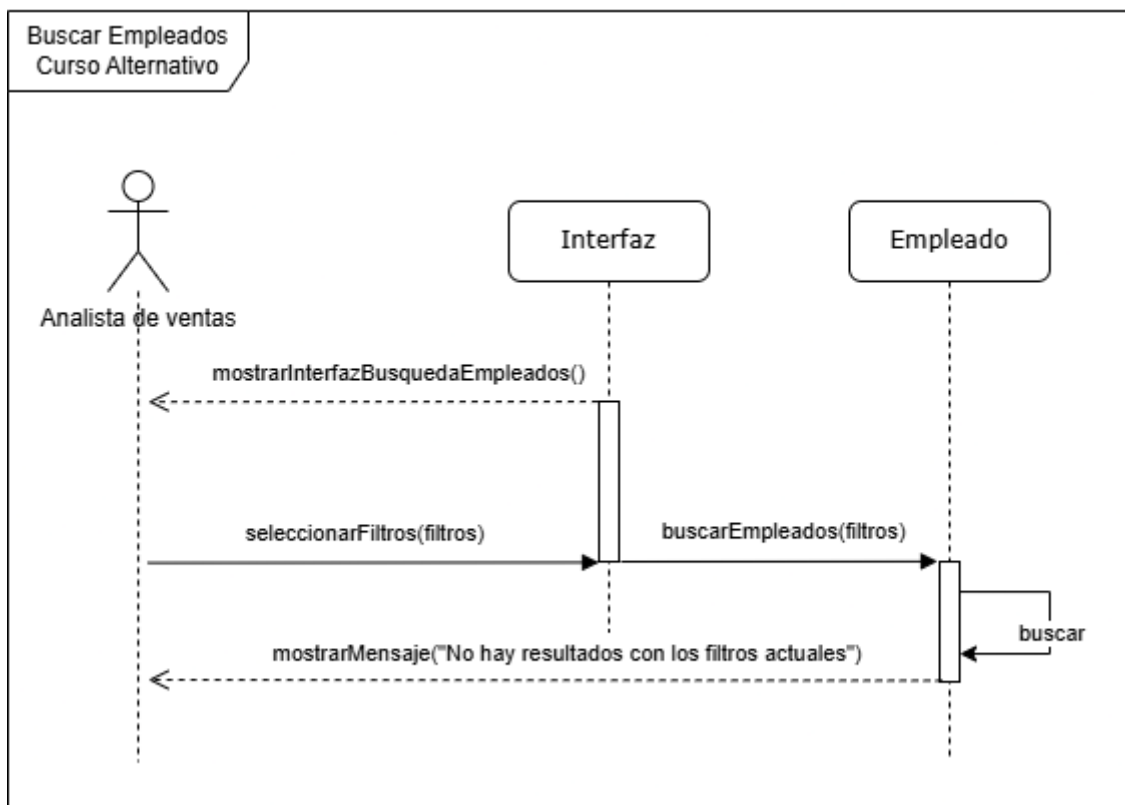
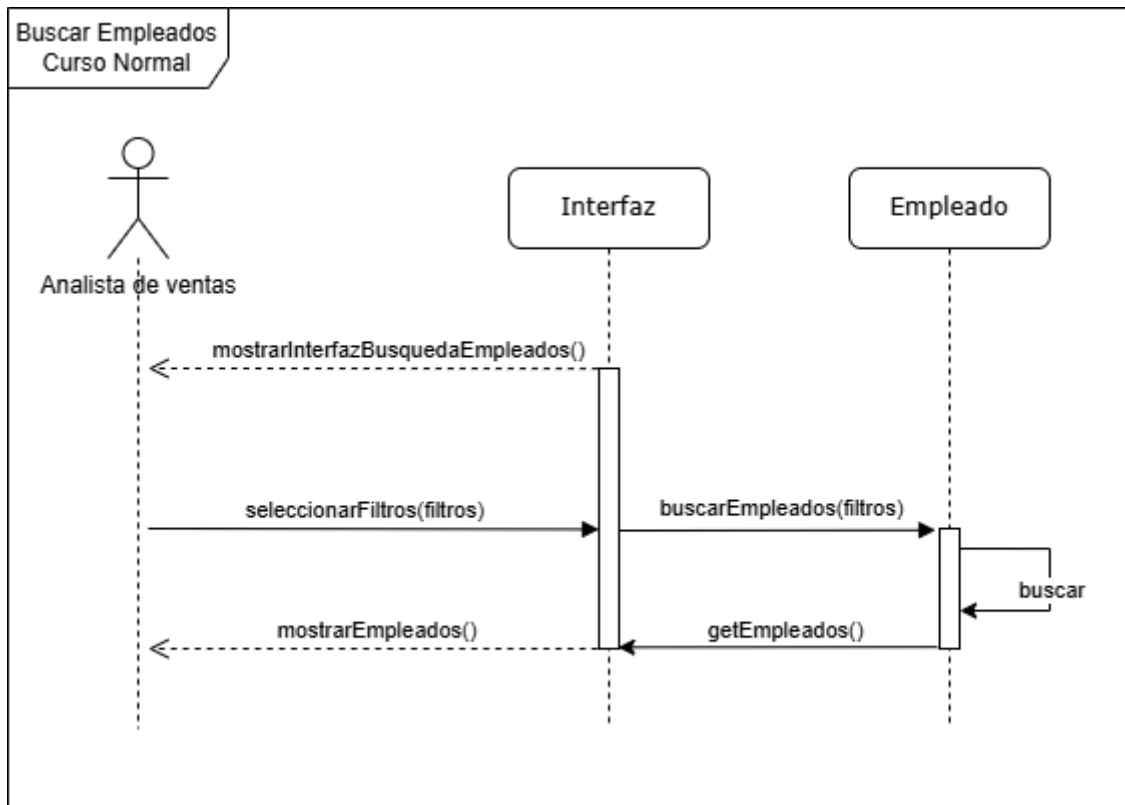


Fig. 2.7. 10. Diagrama de Secuencia de Buscar Ventas – Curso Normal y Alternativo

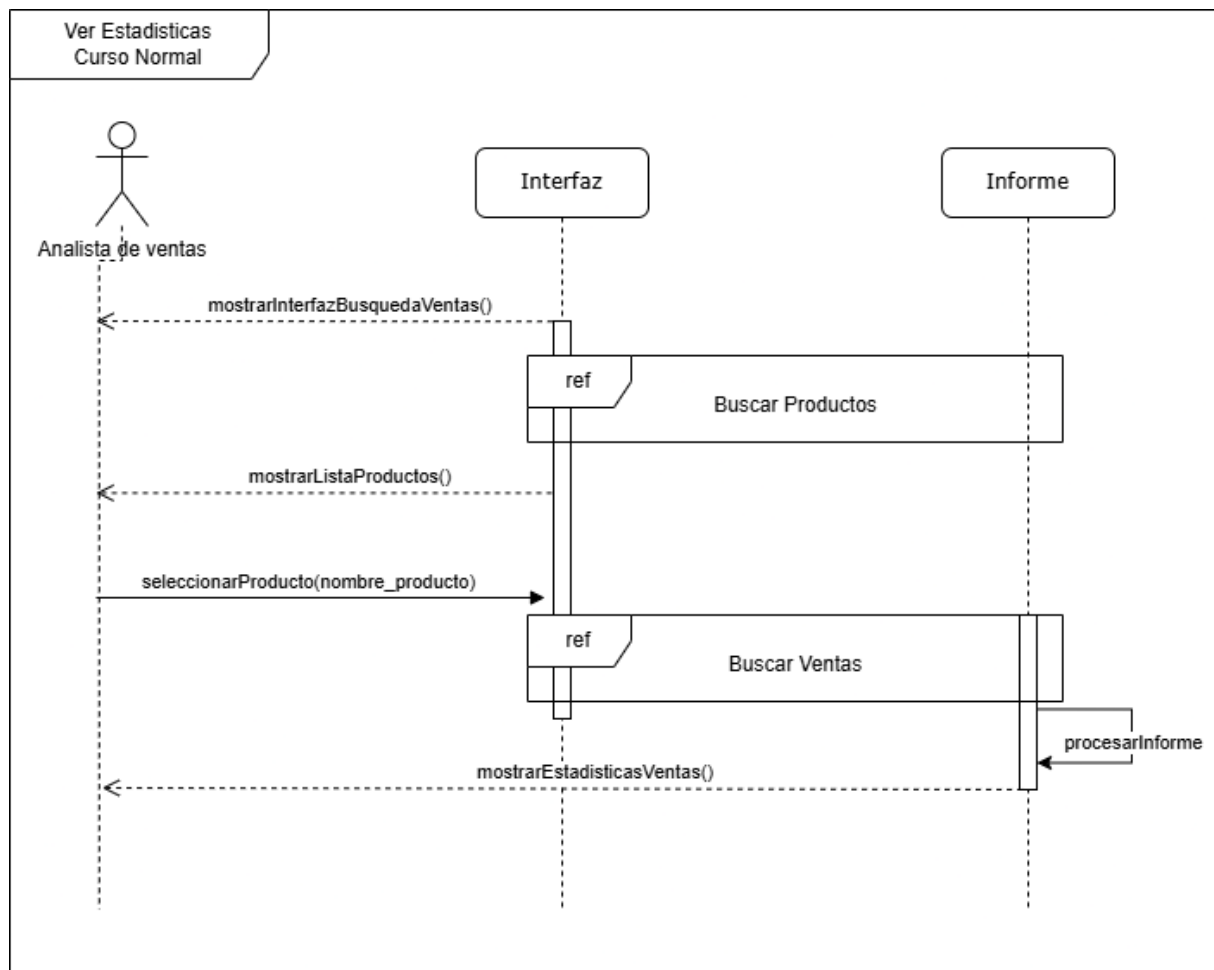


**Fig. 2.7. 11.** Diagrama de Secuencia de Buscar Productos – Curso Normal y Alternativo



**Fig. 2.7. 12.** Diagrama de Secuencia de Buscar Empleados – Curso Normal y Alternativo





**Fig. 2.7. 13.** Diagrama de Secuencia de Ver Estadísticas

## 2.7.4 Contratos de Operaciones Críticas

**Contrato de Operaciones Críticas:** registrarVenta()

**TABLA XXI**

CONTRATO DE REGISTRAR VENTAS

Nombre:	registrarVenta()
Responsabilidades:	- Crear una nueva transacción de venta.
Tipo:	Sistema
Referencias Cruzadas:	Diagrama de Casos de Uso del Vendedor

	Diagrama de Secuencia de Registrar Ventas – Curso Normal y Diagrama de Secuencia de Registrar Ventas – Curso Alternativo
Excepciones:	El usuario no tiene el rol “Vendedor”. El usuario no fue dado de alta o fue dado de baja. Base de datos no disponible.
Precondiciones:	El usuario debe de estar registrado y tener permisos.
Postcondiciones:	El sistema genera un id de venta. El sistema inicializa un detalle de venta vacío. El sistema muestra una pantalla de selección de productos.

**Contrato de Operaciones Críticas:** mostrarEstadisticasVentas()

**TABLA XXII**

CONTRATO DE MOSTRAR ESTADÍSTICAS DE VENTAS

Nombre:	mostrarEstadisticasVentas(producto: Producto)
Responsabilidades:	- Mostrar las estadísticas de ventas correspondientes al producto seleccionado. - Presentar indicadores relevantes (ventas totales, frecuencia, ingresos generados, etc.).
Tipo:	Sistema
Referencias Cruzadas:	Diagrama de Casos de Uso del Analista de Ventas Diagrama de Casos de Uso del Vendedor Diagrama de Secuencia de Ver Estadísticas
Excepciones:	Error en el procesamiento de estadísticas. Fallo en la obtención de datos desde la base de datos.
Precondiciones:	Producto seleccionado por el analista.
Postcondiciones:	Estadísticas de ventas mostradas en la interfaz.

**Contrato de Operaciones Críticas** agregarProducto()

**TABLA XXIII**

**CONTRATO DE AGREGAR PRODUCTO**

Nombre:	agregarProducto (nombre: String, precioLista: Float, stock: Integer, categoria: Categoria)
Responsabilidades:	<ul style="list-style-type: none"><li>- Validar que los datos ingresados para el nuevo producto sean correctos y completos.</li><li>- Verificar que el producto no exista previamente en el catálogo.</li><li>- Registrar el nuevo producto en el sistema.</li><li>- Confirmar el registro mostrando la información almacenada al usuario.</li></ul>
Tipo:	Sistema
Referencias Cruzadas:	Diagrama de Casos de Uso del Administrador Diagrama de Secuencia de Agregar Productos – Curso Normal Diagrama de Secuencia de Agregar Productos – Curso Alternativo
Excepciones:	Si el producto ya existe, se notifica el error Si la información está incompleta, se notifica el error Error de conexión a la base de datos
Precondiciones:	El sistema conoce la información necesaria para el producto El sistema conoce los productos y categorías existentes El sistema conoce qué tipo de usuario tiene acceso a la interfaz
Postcondiciones:	El sistema agrega el producto El sistema muestra la información del producto agregado El sistema da la opción de agregar otro producto El sistema da la opción de finalizar

## 2.7.5 Diagrama de clases

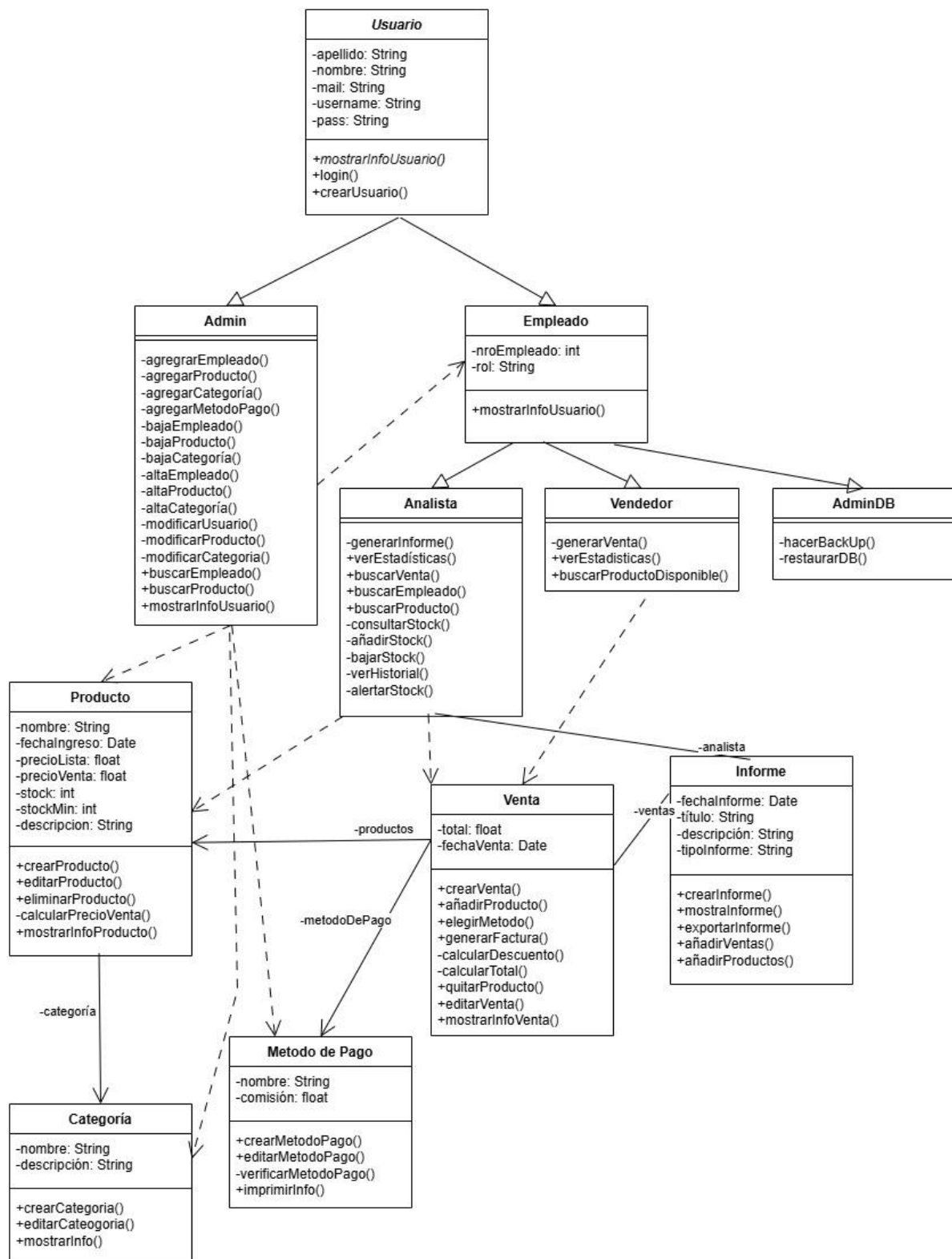


Fig. 2.7. 14. Diagrama de Clases

## 2.7.6. Diagrama de Entidad Relación

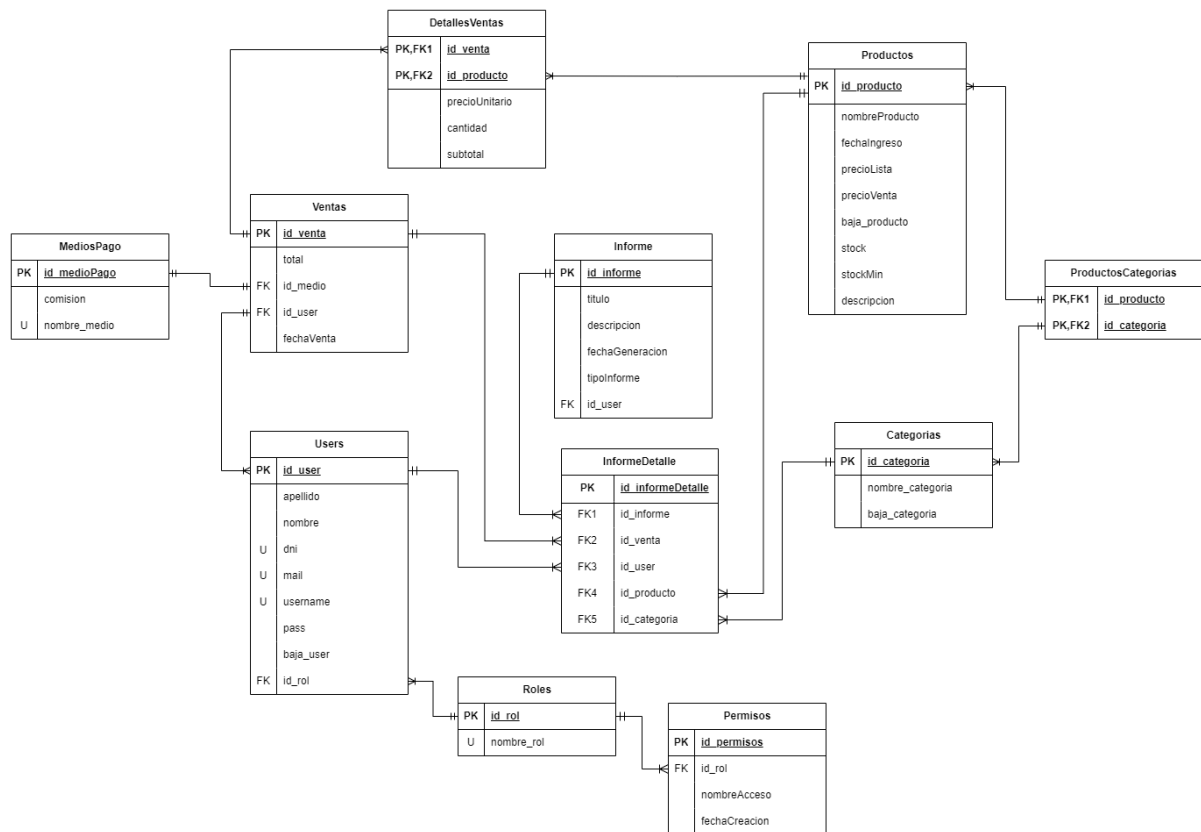


Fig. 2.7. 15. Diagrama Entidad Relación

## 2.8 Estimación de Costos

Para la estimación de costos del proyecto, se empleó la técnica **Planning Poker** combinada con la asignación de **Story Points**.

Esta metodología ágil nos permitió calcular el esfuerzo relativo de cada historia de usuario y caso de uso en función de su complejidad técnica, riesgos asociados y cantidad de trabajo esperado.

Durante las sesiones de planificación cada miembro del equipo asignó de manera individual un número de Story Points a cada funcionalidad o requisito identificado.

Los valores utilizados corresponden a la sucesión de Fibonacci modificada (0, 1, 2, 3, 5, 8, 13, 21), que ayuda a representar de forma más realista las diferencias de complejidad entre las tareas.

En caso de discrepancia significativa entre las asignaciones, se promovió la discusión argumentativa para alcanzar un consenso razonado.

Este enfoque permitió tener en cuenta no solo el tamaño del trabajo, sino también la incertidumbre respecto a la definición de algunos requisitos, la experiencia previa en tareas similares, y el riesgo tecnológico relacionado con nuevas herramientas o integraciones.

Posteriormente, se estableció una equivalencia entre los Story Points y las horas estimadas de trabajo en función de la velocidad promedio del equipo (cantidad de Story Points resueltos por sprint, proyectada a partir de trabajos similares y experiencia previa).

Siendo 1 Story Point  $\approx$  2-4 horas de trabajo efectivo (dependiendo de la tarea).

**Beneficios de la técnica aplicada**

- Mejora la precisión en la estimación inicial frente a métodos tradicionales.
- Fomenta la participación activa de todo el equipo, logrando mayor compromiso.
- Permite ajustar de forma dinámica las proyecciones a medida que avanza el proyecto.

### 3. Capítulo III: Herramientas

#### Herramientas Utilizadas

1. **Draw.io:** Utilizado para la creación de diagramas de secuencia, diagramas de casos de uso, y diagramas de entidad relación, facilitando la visualización de procesos y la estructuración de la información recolectada.
2. **GanttProject:** Empleado para la planificación y seguimiento del proyecto mediante diagramas de Gantt, permitiendo una gestión efectiva de tiempos y recursos.
3. **Figma:** Herramienta de diseño utilizada para la creación de prototipos de interfaces de usuario, asegurando una experiencia visual coherente y funcional.
4. **Trello:** Utilizado para la gestión de tareas y sprints dentro del marco Scrum, facilitando la organización y seguimiento del backlog del producto.
5. **Casos de Uso, Brainstorming y Análisis de Factores Críticos:** Métodos de educación de requerimientos que permitieron una comprensión profunda de las necesidades del usuario y la definición de funcionalidades clave del sistema.
6. **Google Docs:** Herramienta utilizada para la documentación inicial conjunta del proyecto. También fue utilizada para dejar comentarios en caso de que hubiese errores a corregir.
7. **GitHub:** Plataforma utilizada para el control de versiones del código fuente y la colaboración entre los integrantes del equipo. Permitió mantener un repositorio centralizado, gestionar issues, y asegurar la trazabilidad de los cambios realizados durante el ciclo de vida del proyecto.
8. **Azure Data Studio:** Utilizado por uno de los integrantes del equipo para hostear y gestionar una base de datos distribuida, lo cual permitió el acceso remoto y simultáneo a los datos durante el desarrollo colaborativo.
9. **SQL Server Management Studio (SSMS):** Herramienta de administración de bases de datos empleada para modelar, consultar y mantener la base de datos relacional del sistema. Su integración con SQL Server permitió una gestión eficiente y segura de la información.

Estas herramientas combinadas han permitido un desarrollo estructurado y colaborativo, asegurando la entrega de un producto de alta calidad que satisface las expectativas y necesidades del equipo.

## 4. Resultados

Como resultado obtenemos los siguientes prototipos de pantalla, generados en base a la educación de requerimientos utilizando la aplicación Figma.

### 4.1. Esquema de Colores Definido



**Fig. 4. 1.** Esquema de Colores Definido para el Sistema



## 4.2. Pantalla de Inicio de Sesión



The image shows a login screen for a system named "Ocio Store". The background is a solid dark blue. At the top left, there is a logo consisting of two white dice. To the right of the dice, the text "Ocio Store" is written in a white, sans-serif font. Below the logo and text, the words "Iniciar Sesión" are centered in a white, sans-serif font. Underneath this text, there are two white input fields with rounded corners. The first field is labeled "Usuario" and the second field is labeled "Contraseña". Below these fields, there are two dark blue buttons with white text. The left button is labeled "Iniciar Sesión" and the right button is labeled "Cancelar".

**Fig. 4. 2.** Pantalla de Inicio de Sesión del Sistema

#### 4.3. Pantalla de Bienvenida con todos los botones disponibles



**Fig. 4. 3.** Pantalla de Bienvenida del Sistema con todas las Opciones Disponibles

## 5. Conclusiones

El presente trabajo de campo sobre el "Sistema de Administración de una Tienda de Juegos de Mesa" fue planteado aplicando **metodologías de ingeniería de software modernas**, basadas en prácticas de análisis estructurado y métodos ágiles de planificación.

A través del uso de Scrum, se facilitó una planificación iterativa y flexible que permitió una constante adaptación y mejora del sistema a lo largo de su desarrollo.

La implementación de herramientas de diseño y gestión de proyectos, como draw.io, GanttProject y Figma, permitió una visualización clara y un seguimiento efectivo del progreso. Además, la correcta documentación del proceso garantiza **trazabilidad** y **facilita el mantenimiento**, así como la comprensión del sistema por futuros desarrolladores o stakeholders.

Finalmente, cabe destacar que la aplicación de buenas prácticas de modelado UML, la atención al detalle en los contratos y la correcta evaluación de riesgos posicionan a este proyecto como un ejemplo de desarrollo ordenado y profesional dentro del contexto académico.

En conclusión, este proyecto no solo proporciona una plataforma eficiente para la gestión de una tienda de nicho, como una de juegos de mesa, sino que también demuestra la eficacia de Scrum como metodología para el desarrollo de software.

## Bibliografía

- [1] R. S. Pressman, *Ingeniería de software: un enfoque práctico*, 7ª ed., McGraw-Hill, 2010.
- [2] I. Sommerville, *Ingeniería de software*, 7ª ed., Pearson Addison Wesley, ISBN: 84-7829-074-5.
- [3] A. Silberschatz, H. F. Korth, y S. Sudarshan, *Fundamentos de bases de datos*, McGraw-Hill, 2014.
- [4] L. Bass, P. Clements, y R. Kazman, *Software Architecture in Practice*, 3ª ed., Addison-Wesley, 2012.
- [5] C. Larman, *UML y Patrones*, 2ª ed., Prentice Hall, 2004.
- [6] O. A. Vallejos, *Plantillas de Clases de Teoría de Ingeniería del Software 1*, Material de clase, 2024.
- [7] L. Gómez Solís, *Plantillas de Clases de Práctica de Ingeniería del Software 1*, Material de clase, 2024.
- [8] M. A. Ferraro, *Plantillas de Clases de Teoría de Ingeniería del Software 2*, Material de clase, 2025.
- [9] L. Gómez Solís y J. A. Carruthers, *Plantillas de Clases de Práctica de Ingeniería del Software 2*, Material de clase, 2025.
- [10] A. Matoso. *Plantillas de Diagramas de Secuencia de Ingeniería del Software 1*, Material de clase, 2021.

## **Anexos**