****

Facultad de Ingeniería

Ingeniería de Software I

**PROYECTO INGENIERÍA DE SOFTWARE PRIMER INCREMENTO SCRUM++: SISTEMA DE GESTIÓN INTEGRAL PARA AMASANDERÍA Y ALMACÉN 981**

Autores:

Cristóbal Alvarado P.

Diego Barría S.

Vicente Gaete H.

Joaquín López C.

Felipe Riquelme S.

Benjamín Teplizky D.

Profesor:

Paulo Quinsacara J.

Santiago, Chile

2025

**Índice de Contenidos**

1. **Introducción** pág. 05
2. **Metodología de desarrollo** pág. 06
3. **Product Backlog** pág. 08
4. **Sprint Backlog** pág. 13
5. **Vista Lógica** pág. 19
6. **Vista Proceso** pág. 21
7. **Vista Desarrollo** pág. 29
8. **Vista Física** pág. 31
9. **Árbol de Navegación** pág. 33
10. **Casos de pruebas** pág. 34
11. **Gráficos de esfuerzos** pág. 49
12. **Conclusión** pág. 51

**Índice de Figuras**

1. **Vista Lógica**

**Figura 1.1** Diagrama de clases primer sprint pág. 07

**Figura 1.2** Modelo de datos primer sprint pág. 08

1. **Vista Proceso**

**Figura 2.1 a Figura 2.15** Diagrama de secuencias pág. 21

1. **Vista Desarrollo**

**Figura 3.1** Diagrama de componentes pág. 29

1. **Vista Física**

**Figura 4.1** Diagrama de despliegue pág. 39

1. **Árbol de Navegación**

**Figura 5.1** Árbol de navegación pág. 31

1. **Casos de pruebas**

**Figura 6.1 a Figura 6.12** Evidencias de casos de pruebas pág. 32

1. **Gráficos de esfuerzos**

**Figura 7.1** Gráfico Burn-Up pág. 46

**Figura 7.2** Gráfico Burn-Down pág. 47

**Índice de Tablas**

1. **Product Backlog**

**Tabla 1.1** Requerimientos del sistema pág. 08

**Tabla 1.2** Requerimientos para el primer sprint pág. 11

**Tabla 1.3** Product Backlog List primer sprint pág. 12

1. **Casos de Uso Extendido**

**Tablas 2.1 a 2.12.** Casos de uso extendido del sistema pág. 13

1. **INTROUDCCIÓN**

El presente documento es una continuación del proyecto para el “Software de Ventas e Inventario, Amasandería y Panadería 981”, primeramente fue el documento 0, donde se contextualizo la empresa, además se definió los requerimientos a realizar, mientras que este informe corresponde al Primer Incremento del proyecto desarrollado bajo la metodología Scrum++, una variante del marco ágil Scrum que incorpora entregables documentales estructurados para facilitar el seguimiento y validación del avance. Este primer incremento cubre el 20% de los requerimientos del sistema, priorizando aquellos de mayor valor para el funcionamiento operativo de la Amasandería y Panadería 981, un negocio tradicional de producción y venta de pan en un entorno de barrio.

Durante este sprint, el equipo se enfocó en implementar funcionalidades esenciales que permitieran digitalizar y optimizar procesos críticos como el inicio de jornada, la planificación de la producción, el control de insumos básicos y el registro de productos disponibles para la venta. Este conjunto de funcionalidades representa la base sobre la cual se construirá el sistema completo, y fue validado mediante entregables funcionales y documentación técnica conforme a las prácticas de Scrum++.

1. **METODOLOGIA DE DESARROLLO**

Scrum++ es una metodología híbrida que surge de la combinación entre el marco ágil Scrum y el modelo de vistas 4+1 propuesto por Philippe Kruchten. Esta integración se justifica debido a una debilidad inherente en Scrum: la escasa documentación técnica y arquitectónica del sistema, que puede afectar negativamente el mantenimiento, la escalabilidad o la evolución del software a largo plazo.

Por su parte, el modelo de vistas 4+1 proporciona una estructura sólida para documentar la arquitectura del software desde cinco perspectivas complementarias, apoyándose en distintos tipos de diagramas UML. De esta manera, Scrum++ logra mantener la agilidad en el desarrollo, sin sacrificar la claridad estructural y la trazabilidad técnica del sistema.

Esta metodología busca equilibrar flexibilidad y documentación, integrando en cada Sprint tanto funcionalidades operativas como sus respectivas vistas arquitectónicas. Esto permite que cada incremento no solo aporte valor funcional al cliente, sino que además quede debidamente respaldado a nivel de diseño y arquitectura.

Características principales de Scrum++:

* Sprint independiente y autosuficiente: Cada sprint representa una unidad funcional y técnica completa. Esto facilita la integración modular del sistema, reduciendo la complejidad del ensamblaje y mejorando el mantenimiento del código.
* Product Backlog: Contiene los requerimientos del sistema, priorizados en función del valor que entregan al negocio.
* Sprint Backlog: Incluye tanto las funcionalidades a desarrollar (usualmente descritas como casos de uso) como los elementos documentales y de diseño que acompañan al desarrollo.
* Documentación arquitectónica por vistas (modelo 4+1):
  + Vista lógica: Representada mediante diagramas de clases y modelo de datos, describe la funcionalidad del sistema desde el punto de vista de los desarrolladores.
  + Vista de procesos: Utiliza diagramas de secuencia para mostrar los flujos de ejecución y la interacción entre componentes.
  + Vista de desarrollo: Refleja la estructura de los componentes del software y su organización en módulos o paquetes.
  + Vista física: Describe la infraestructura de despliegue mediante diagramas de nodos y servidores.
  + Vista de casos de uso (*+1*): Ilustra las funcionalidades del sistema desde la perspectiva del usuario final.

Gracias a esta integración, Scrum++ permite obtener incrementos funcionales ágiles, acompañados de una documentación clara, estructurada y sostenible, mejorando la mantenibilidad del sistema en el tiempo.

1. **PRODUCT BACKLOG**

4.1. Requerimientos

En esta sección se muestran los requerimientos del sistema que definen criterios de calidad como rendimiento, seguridad y usabilidad, pero más importante la prioridad, con la cual luego se escogerán los requerimientos para el primer sprint, a través de la ***Tabla 1.1***.

*Tabla 1.1: “Requerimientos para el software de Amasandería y Panadería 981”*

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **ID** | **Descripción** | **Necesidad** | **Prioridad** | **Estabilidad** | **Claridad** | **Verificabilidad** | **Fuente** |
| **1.-** | **Punto de Venta** | |  |  |  |  |  |  |
| **UR** | **1.1** | **Lectura de códigos de barra** | 1 | 1 | 1 | 1 | 1 | 2 |
|  | Operario autorizado puede realizar ventas en el sistema. | | | | | | | |
| **UR** | **1.2** | **Emisión y Anulación de boletas no fiscales** | 2 | 2 | 1 | 1 | 1 | 2 |
|  | Operario autorizado puede emitir y anular boletas al finalizar una venta. | | | | | | | |
| **UR** | **1.3** | **Registro del método de pago** | 2 | 2 | 2 | 1 | 1 | 2 |
|  | Operario autorizado puede registrar el método de pago en el sistema. | | | | | | | |
| **UR** | **1.4** | **Búsqueda de productos** | 1 | 1 | 1 | 1 | 1 | 2 |
|  | Operario autorizado puede buscar productos en el sistema. | | | | | | | |
| **UR** | **1.5** | **Registro automático de ventas** | 1 | 1 | 1 | 1 | 1 | 2 |
|  | Sistema automáticamente registra la venta. | | | | | | | |
| **UR** | **1.6** | **Aplicación de descuentos** | 2 | 3 | 1 | 1 | 1 | 2 |
|  | Operario autorizado debe aplicar descuentos válidos. | | | | | | | |
| **UR** | **1.7** | **Aplicación automática de promociones** | 2 | 3 | 1 | 1 | 1 | 2 |
|  | Sistema automáticamente aplica la promoción válida. | | | | | | | |
| **UR** | **1.8** | **Ingreso manual de productos** | 1 | 1 | 1 | 1 | 1 | 2 |
|  | Operario autorizado puede ingresar productos en el sistema. | | | | | | | |
| **UR** | **1.9** | **Visualización en tiempo real del total** | 1 | 1 | 1 | 1 | 1 | 2 |
|  | Sistema automáticamente actualiza el total de ventas. | | | | | | | |
| **UR** | **1.10** | **Impresión de ticket detallado** | 1 | 2 | 1 | 1 | 1 | 2 |
|  | Sistema automáticamente imprime el ticket. | | | | | | | |
| **UR** | **1.11** | **Pago con múltiples métodos** | 1 | 1 | 1 | 1 | 1 | 2 |
|  | Operario autorizado debe registrar pagos combinados en el sistema. | | | | | | | |
| **UR** | **1.12** | **Accesos rápidos a productos frecuentes** | 2 | 3 | 1 | 1 | 1 | 2 |
|  | Operario autorizado puede revisar los productos frecuentes en el sistema. | | | | | | | |
| **UR** | **1.13** | **Historial filtrable de ventas** | 2 | 2 | 1 | 1 | 1 | 2 |
|  | Operario autorizado puede revisar historial de ventas. | | | | | | | |
| **UR** | **1.14** | **Ventas por cantidad o peso** | 2 | 2 | 1 | 1 | 1 | 2 |
|  | Operario autorizado puede realizar ventas por cantidad o peso en el sistema. | | | | | | | |
| **UR** | **1.15** | **Suspensión temporal de venta** | 1 | 1 | 1 | 1 | 1 | 2 |
|  | Operario autorizado puede suspender ventas en el sistema. | | | | | | | |
| **UR** | **1.16** | **Notas personalizadas por venta** | 2 | 3 | 2 | 1 | 1 | 2 |
|  | Operario autorizado puede agregar una nota por venta en el sistema. | | | | | | | |
| **UR** | **1.17** | **Cierre de caja** | 1 | 1 | 1 | 1 | 1 | 2 |
|  | Operario autorizado debe realizar el cierre de caja en el sistema. | | | | | | | |
| **UR** | **1.18** | **Apertura y cierre de caja por turno** | 1 | 1 | 1 | 1 | 1 | 2 |
|  | Operario autorizado debe registrar apertura y cierre de turno en el sistema. | | | | | | | |
| **UR** | **1.19** | **Integración con balanzas electrónicas** | 2 | 2 | 2 | 1 | 1 | 2 |
|  | Operario autorizado puede leer el peso desde balanzas conectadas al sistema. | | | | | | | |
| **UR** | **1.20** | **Bloqueo de productos fuera de stock** | 2 | 2 | 1 | 1 | 1 | 2 |
|  | Sistema automáticamente impide la venta de productos sin stock. | | | | | | | |
| **UR** | **1.21** | **Gestión de promociones por fecha y hora** | 2 | 3 | 2 | 1 | 1 | 2 |
|  | Operario autorizado puede definir promociones solo en ciertos días y horarios en sistema. | | | | | | | |
| **UR** | **1.22** | **Registro de devoluciones** | 1 | 1 | 1 | 1 | 1 | 2 |
|  | Operario autorizado puede ingresar productos devueltos en el sistema. | | | | | | | |
| **2.-** | **Módulo de inventario** | |  |  |  |  |  |  |
| **UR** | **2.1** | **Ingreso de nuevos productos** | 1 | 1 | 1 | 1 | 1 | 2 |
|  | Operario autorizado puede registrar nuevos productos en sistema. | | | | | | | |
| **UR** | **2.2** | **Actualización de stock** | 1 | 1 | 1 | 1 | 1 | 2 |
|  | Operario autorizado puede sumar o restar stock de un producto en el sistema. | | | | | | | |
| **UR** | **2.3** | **Control de productos con stock mínimo** | 2 | 2 | 1 | 1 | 1 | 2 |
|  | Sistema automáticamente alerta de stock mínimo de un producto. | | | | | | | |
| **UR** | **2.4** | **Inventario por categoría o proveedor** | 1 | 2 | 1 | 1 | 1 | 2 |
|  | Operario autorizado puede visualizar el inventario en el sistema. | | | | | | | |
| **UR** | **2.5** | **Carga masiva de productos vía Excel (.xlsx)** | 1 | 2 | 1 | 1 | 1 | 2 |
|  | Operario autorizado puede cargar un archivo Excel en el sistema. | | | | | | | |
| **UR** | **2.6** | **Historial de movimientos de inventario** | 1 | 1 | 2 | 1 | 1 | 2 |
|  | Operario autorizado puede revisar historial de movimientos de inventario en el sistema. | | | | | | | |
| **UR** | **2.7** | **Registro de productos vencidos o dañados** | 2 | 2 | 2 | 1 | 1 | 2 |
|  | Operario debe marcar productos como vencidos o dañados en el sistema. | | | | | | | |
| **UR** | **2.8** | **Inventario físico** | 2 | 2 | 1 | 1 | 1 | 2 |
|  | Sistema automáticamente compara stock registrado, y permite aprobar o ajustar diferencias. | | | | | | | |
| **3.-** | **Módulo de reportes** | |  |  |  |  |  |  |
| **UR** | **3.1** | **Reporte de ventas diarias** | 1 | 1 | 2 | 1 | 1 | 2 |
|  | Sistema automáticamente genera un reporte de ventas diaria. | | | | | | | |
| **UR** | **3.2** | **Reporte de productos más vendidos** | 1 | 1 | 2 | 1 | 1 | 2 |
|  | Sistema automáticamente genera un reporte de productos más vendidos. | | | | | | | |
| **UR** | **3.3** | **Reporte de productos con baja rotación** | 2 | 2 | 2 | 1 | 1 | 2 |
|  | Sistema automáticamente genera reporte de productos con baja rotación. | | | | | | | |
| **UR** | **3.4** | **Reporte de movimiento de inventario** | 1 | 1 | 2 | 1 | 1 | 2 |
|  | Operario autorizado puede generar un reporte de movimientos de inventario en el sistema. | | | | | | | |
| **UR** | **3.5** | **Reporte de diferencias de inventario** | 2 | 2 | 2 | 1 | 1 | 2 |
|  | Sistema automáticamente genera reporte de diferencias en inventario. | | | | | | | |
| **UR** | **3.6** | **Exportación de reportes agendada** | 2 | 2 | 2 | 1 | 1 | 2 |
|  | Operario puede agendar reportes para enviar automáticamente por correo electrónico. | | | | | | | |
| **4.-** | **Módulo de métricas** | |  |  |  |  |  |  |
| **UR** | **4.1** | **Margen de ganancia por producto** | 1 | 1 | 1 | 1 | 1 | 2 |
|  | Sistema automáticamente calcula el margen de ganancia por producto. | | | | | | | |
| **UR** | **4.2** | **Promedio de venta diaria y mensual** | 1 | 1 | 2 | 1 | 1 | 2 |
|  | Administrador puede observar promedio diario y mensual de ventas. | | | | | | | |
| **UR** | **4.3** | **Rotación de inventario** | 2 | 2 | 2 | 1 | 1 | 2 |
|  | Sistema automáticamente calcula la rotación de stock. | | | | | | | |
| **UR** | **4.4** | **Tasa de anulación de ventas** | 2 | 2 | 2 | 1 | 1 | 2 |
|  | Administrador puede observar la tasa de anulación de ventas en el sistema. | | | | | | | |
| **5.-** | **Módulo de seguridad** | |  |  |  |  |  |  |
| **UR** | **5.1** | **Perfiles de usuario** | 1 | 1 | 1 | 1 | 1 | 2 |
|  | Administrador puede otorgar diferentes perfiles a los empleados en el sistema. | | | | | | | |
| **UR** | **5.2** | **Autenticación obligatoria** | 1 | 1 | 1 | 1 | 1 | 2 |
|  | Cualquier usuario debe acceder con usuario y contraseña en el sistema. | | | | | | | |
| **UR** | **5.3** | **Control de sesiones activas** | 1 | 1 | 1 | 1 | 1 | 2 |
|  | Sistema automáticamente registra cada sesión activa. | | | | | | | |
| **UR** | **5.4** | **Bitácora de acciones** | 1 | 1 | 1 | 1 | 1 | 2 |
|  | Sistema automáticamente registra las acciones relevantes en el sistema. | | | | | | | |
| **UR** | **5.5** | **Bitácora de acciones por usuario** | 1 | 1 | 1 | 1 | 1 | 2 |
|  | Sistema automáticamente registra las acciones realizas de cada usuario. | | | | | | | |
| **6.-** | **Módulo de usabilidad** | |  |  |  |  |  |  |
| **UR** | **6.1** | **Interfaz para computador de escritorio** | 1 | 1 | 1 | 1 | 1 | 2 |
|  | Usuario puede acceder al sistema desde una interfaz adaptada para computador de escritorio. | | | | | | | |
| **UR** | **6.2** | **Diseño intuitivo** | 1 | 1 | 1 | 1 | 1 | 2 |
|  | Usuario puede navegar fácilmente por el sistema en un diseño claro e intuitivo. | | | | | | | |
| **UR** | **6.3** | **Mensajes de validación claros** | 1 | 1 | 1 | 1 | 1 | 2 |
|  | Usuario recibe mensajes de validación al ingresar datos erróneos. | | | | | | | |
| **UR** | **6.4** | **Accesos rápidos en pantalla principal** | 1 | 1 | 1 | 1 | 1 | 2 |
|  | Usuario puede acceder rápidamente a las funciones principales desde la pantalla inicial. | | | | | | | |
| **UR** | **6.5** | **Manual de usuario integrado** | 1 | 1 | 1 | 1 | 1 | 2 |
|  | Usuario puede consultar un manual de ayuda integrado en el sistema. | | | | | | | |

*Fuente: Elaborado por el estudiante de acuerdo con el proyecto*

4.2. Planificación del primer sprint

Se planificará cuales son los requerimientos pertenecientes de los casos de usos extendidos del Product Backlog List que se abordaran en este sprint, para esto solo se deben escoger mínimamente el 20% de los 57 CU, haciendo el cálculo, serían 12 CU a escoger a partir de la prioridad del sistema, abordando las prioridades desde la más relevante (requerimientos que tengan una prioridad 1) hasta la menos relevante (requerimientos que tengan una prioridad 3), se muestra en la ***Tabla 1.2*** con los requerimientos pertenecientes de los CU escogidos.

*Tabla 1.2: “Requerimientos para el primer sprint”*

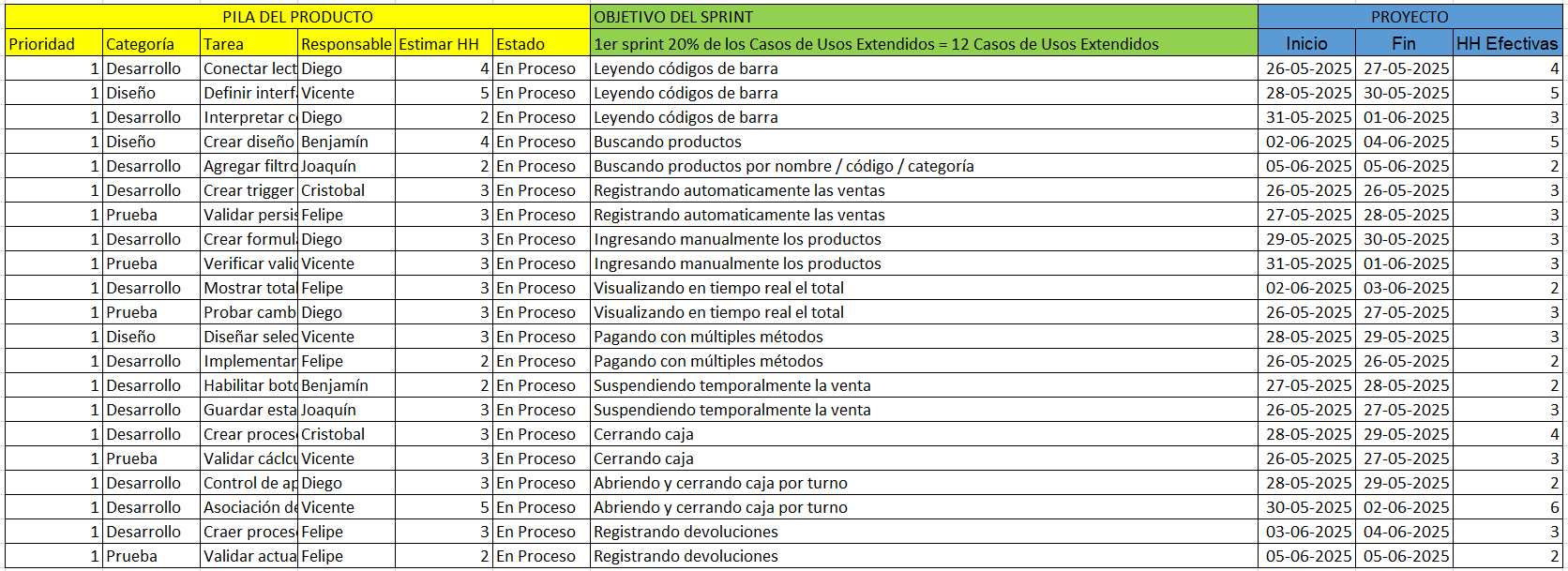
|  |  |  |
| --- | --- | --- |
| ID | Nombre del Requerimiento | Prioridad |
| 1.1 | Lectura de códigos de barra | 1 |
| 1.4 | Búsqueda de productos | 1 |
| 1.5 | Registro automático de ventas | 1 |
| 1.8 | Ingreso manual de productos | 1 |
| 1.9 | Visualización en tiempo real del total | 1 |
| 1.11 | Pago con múltiples métodos | 1 |
| 1.15 | Suspensión temporal de venta | 1 |
| 1.17 | Cierre de caja | 1 |
| 1.18 | Apertura y cierre de caja por turno | 1 |
| 1.19 | Registro de devoluciones | 1 |

*Fuente: Elaborado por el estudiante de acuerdo con el proyecto*

4.3. Product Backlog List

El Product Backlog List corresponde al listado priorizado de los CU Extendido del sistema, en este caso del sprint 1, recopilados desde las necesidades del cliente y los objetivos del proyecto. Cada ítem representa una funcionalidad o característica a implementar, y se prioriza en función de su valor para el usuario y su viabilidad técnica. A continuación, se muestra la ***Tabla 1.3***.

*Tabla 1.3: “Product Backlog List primer sprint”*



*Fuente: Elaborado por el estudiante de acuerdo con el proyecto*

1. **SPRINT BACKLOG**

Este capítulo constará con las vistas externas, que serían los casos de usos extendidos. Los requerimientos anteriores nos ayudaran a el desarrollo de las actividades.

5.1. Vista externa: Casos de uso extendido

Los casos de uso extendido describen escenarios concretos de interacción entre los usuarios (actores) y el sistema. Cada uno especifica un objetivo particular que un actor desea alcanzar, así como las condiciones necesarias, las respuestas del sistema y las posibles variaciones en el flujo de eventos, como podemos ver en los siguientes en las ***Tablas 2.1 a 2.12***.

*Tablas de 2.1 a 2.12: “Casos de uso extendido de proyecto”.*

|  |  |
| --- | --- |
| Caso de uso N° 1 | Leyendo códigos de barra |
| Actores | Operario autorizado |
| Propósito | Permitir al operario realizar ventas en el sistema mediante lectura de códigos de barra |
| Precondiciones | El operario está autenticado en el sistema. |
| Resumen | El operario utiliza un lector de códigos de barra para agregar productos a la venta. |
| Postcondiciones | Los productos son agregados correctamente al carrito de ventas. |
| Tipos | Principal |
| Curso normal de eventos | |
| Acciones de los actores | Respuesta del sistema |
| 1. El operario escanea el código de barra de un producto. | 2. El sistema muestra el producto escaneado  3. El sistema agrega el producto al carrito de ventas. |

|  |  |
| --- | --- |
| Caso de uso N° 2 | Buscando productos por nombre |
| Actores | Operario autorizado |
| Propósito | Permitir al operario buscar productos por nombre en el sistema. |
| Precondiciones | El operario está autenticado y tiene acceso al sistema. |
| Resumen | El operario realiza una búsqueda de productos por nombre. |
| Postcondiciones | Los productos relevantes se muestran en los resultados de búsqueda. |
| Tipos | Principal |
| Curso normal de eventos | |
| Acciones de los actores | Respuesta del sistema |
| 1. El operario ingresa el término de búsqueda nombre. | 2. El sistema muestra una lista de los productos que coinciden con la búsqueda. |

|  |  |
| --- | --- |
| Caso de uso N° 3 | Buscando producto por código |
| Actores | Operario autorizado |
| Propósito | Permitir al operario buscar el producto por categoría en el sistema. |
| Precondiciones | El operario está autenticado y tiene acceso al sistema. |
| Resumen | El operario realiza una búsqueda del producto por código. |
| Postcondiciones | Los productos relevantes se muestran en los resultados de búsqueda. |
| Tipos | Principal |
| Curso normal de eventos | |
| Acciones de los actores | Respuesta del sistema |
| 1. El operario ingresa el término de búsqueda código. | 2. El sistema muestra el producto que coincida con la búsqueda. |

|  |  |
| --- | --- |
| Caso de uso N° 4 | Buscando productos por categoría |
| Actores | Operario autorizado |
| Propósito | Permitir al operario buscar productos por categoría en el sistema. |
| Precondiciones | El operario está autenticado y tiene acceso al sistema. |
| Resumen | El operario realiza una búsqueda de productos por categoría. |
| Postcondiciones | Los productos relevantes se muestran en los resultados de búsqueda. |
| Tipos | Principal |
| Curso normal de eventos | |
| Acciones de los actores | Respuesta del sistema |
| 1. El operario ingresa el término de búsqueda categoría. | 2. El sistema muestra una lista de los productos que coinciden con la búsqueda. |

|  |  |
| --- | --- |
| Caso de uso N° 5 | Registrando automáticamente las ventas |
| Actores | Sistema |
| Propósito | El sistema registra automáticamente la venta una vez completada. |
| Precondiciones | El operario ha completado el proceso de pago y la venta está lista para ser registrada. |
| Resumen | El sistema registra los detalles de la venta automáticamente. |
| Postcondiciones | La venta queda registrada correctamente en el sistema. |
| Tipos | Principal |
| Curso normal de eventos | |
| Acciones de los actores | Respuesta del sistema |
| 1. Ninguna acción por parte del operario (automático). | 2. El sistema guarda la información de la venta en el registro de ventas. |

|  |  |
| --- | --- |
| Caso de uso N°6 | Ingresando manualmente los productos |
| Actores | Operario autorizado |
| Propósito | Permitir al operario ingresar productos manualmente al sistema. |
| Precondiciones | El operario tiene acceso al sistema y está autenticado. |
| Resumen | El operario ingresa manualmente los detalles de un producto. |
| Postcondiciones | El producto es agregado correctamente al inventario. |
| Tipos | Principal |
| Curso normal de eventos | |
| Acciones de los actores | Respuesta del sistema |
| 1. El operario ingresa los datos del producto (nombre, precio, etc.) según estime conveniente. | 2. El sistema guarda la información del producto ingresado. |

|  |  |
| --- | --- |
| Caso de uso N°7 | Visualizando en tiempo real del total de venta |
| Actores | Operario autorizado. |
| Propósito | Actualizar automáticamente el total de la venta en tiempo real conforme se agregan productos. |
| Precondiciones | El operario ha iniciado una venta y está agregando productos al carrito. |
| Resumen | El sistema calcula y muestra el total acumulado de la venta en tiempo real. |
| Postcondiciones | El total reflejado corresponde a los productos y descuentos aplicados hasta el momento. |
| Tipos | Principal |
| Curso normal de eventos | |
| Acciones de los actores | Respuesta del sistema |
| 1. El operario agrega o elimina productos del carrito. | 2. El sistema recalcula el nuevo total de la venta en tiempo real.  3. El sistema muestra el nuevo total de la venta en tiempo real. |

|  |  |
| --- | --- |
| Caso de uso N°8 | Guardando en el sistema automáticamente el ticket detallado. |
| Actores | Sistema |
| Propósito | Permitir al sistema guardar el ticket detallado. |
| Precondiciones | El operario ha iniciado el proceso de cobro. |
| Resumen | El sistema guarda un ticket que incluye detalles de los productos, precios, descuentos y total pagado. |
| Postcondiciones | El pago se registra correctamente y la venta se completa. |
| Tipos | Principal |
| Curso normal de eventos | |
| Acciones de los actores | Respuesta del sistema |
| 1. No aplica. | 2. El sistema guarda el ticket generado en la base de datos. |

|  |  |
| --- | --- |
| Caso de uso N°9 | Suspendiendo temporalmente la venta |
| Actores | Operario autorizado |
| Propósito | Permitir pausar una venta sin perder los datos ya ingresados. |
| Precondiciones | La venta debe estar activa. |
| Resumen | El operario opta por suspender la venta para retomarla posteriormente. |
| Postcondiciones | La venta queda guardada en estado temporal. |
| Tipos | Principal |
| Curso normal de eventos | |
| Acciones de los actores | Respuesta del sistema |
| 1. El operario selecciona la opción suspender venta. | 2. El sistema guarda el estado actual de la venta.  3. El sistema lo asocia a una lista de ventas en pausa. |

|  |  |
| --- | --- |
| Caso de uso N°10 | Cerrando la caja |
| Actores | Operario autorizado |
| Propósito | Registrar el cierre de caja al finalizar una jornada o turno. |
| Precondiciones | Debe haber una caja abierta con operaciones registradas. |
| Resumen | El operario revisa el resumen de caja y confirma el cierre. |
| Postcondiciones | La caja queda cerrada y los datos almacenados para reportes. |
| Tipos | Principal |
| Curso normal de eventos | |
| Acciones de los actores | Respuesta del sistema |
| 1. El operario accede al módulo de cierre  2. El operario revisa el resumen  3. El operario selecciona confirmar resumen | 3. El sistema cierra la caja.  4. El sistema genera el registro.  5. El sistema guarda el registro. |

|  |  |
| --- | --- |
| Caso de uso N°11 | Abriendo la caja. |
| Actores | Operario autorizado |
| Propósito | Controlar el inicio de las actividades. |
| Precondiciones | El operario inicia su jornada. |
| Resumen | El sistema registra la hora de apertura e inicia lo necesario para empezar las ventas. |
| Postcondiciones | Se genera un reporte del turno asociado al operario. |
| Tipos | Principal |
| Curso normal de eventos | |
| Acciones de los actores | Respuesta del sistema |
| 1. El operario registra la apertura de caja, y al finalizar, registra el cierre. | 2. El sistema inicia lo necesario para empezar las ventas  3. El sistema guarda los datos del turno con hora, montos y ventas realizadas. |

|  |  |
| --- | --- |
| Caso de uso N°12 | Registrando devoluciones |
| Actores | Operario autorizado |
| Propósito | Permitir registrar productos devueltos por clientes. |
| Precondiciones | Debe existir una venta previa registrada. |
| Resumen | El operario busca la venta, selecciona el producto y registra la devolución con su motivo. |
| Postcondiciones | Se actualiza el stock y queda el registro de la devolución. |
| Tipos | Principal |
| Curso normal de eventos | |
| Acciones de los actores | Respuesta del sistema |
| 1. El operario accede al historial de ventas  2. El operario selecciona la venta y producto  3. El operario ingresa los productos devueltos | 4. El sistema actualiza el stock.  5. El sistema guarda el registro con motivo, fecha y hora de devolución. |

*Fuente: Elaborado por el estudiante de acuerdo con el proyecto*

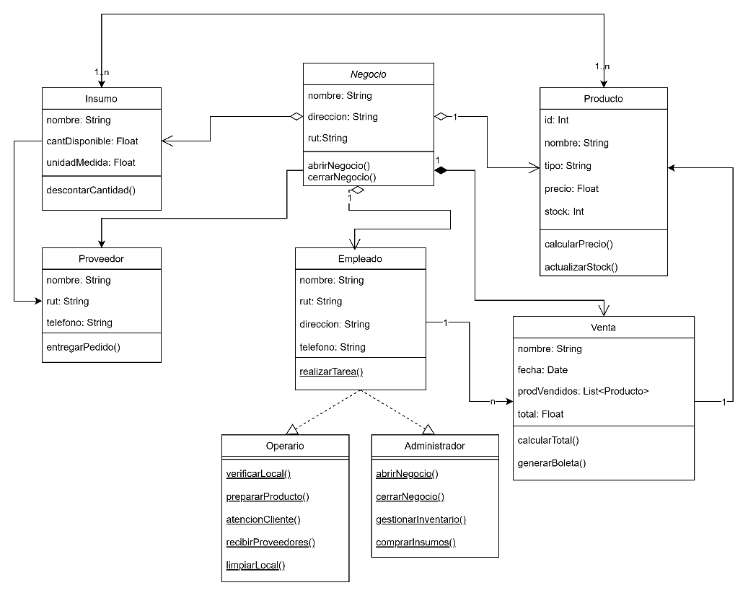
1. **VISTA LÓGICA**

En esta sección se presenta la Vista Lógica del sistema, una de las cinco perspectivas del modelo 4+1 integrado en la metodología Scrum++. Esta vista describe la estructura interna del software desde una perspectiva orientada a objetos, permitiendo visualizar cómo se organiza y relaciona el código fuente que implementa la funcionalidad del sistema.

6.1. Diagrama de clases

El diagrama de clases a mostrar, representa la estructura lógica del sistema de un negocio que elabora y vende productos de panadería. Incluye las principales entidades: Negocio, Empleado (con sus subclases Operario y Administrador), Producto, Insumo, Proveedor y Venta. Se muestran las relaciones entre estas clases, como la asociación entre productos e insumos, la relación de ventas con productos y empleados, y la gestión de insumos por parte de proveedores. También se representa la herencia desde la clase Empleado, reflejando los distintos roles en la empresa. El diagrama permite visualizar cómo interactúan los distintos elementos clave del sistema, como se ve a continuación en la ***Figura 1.1***.

*Figura 1.1: “Diagrama de clases primer sprint”*

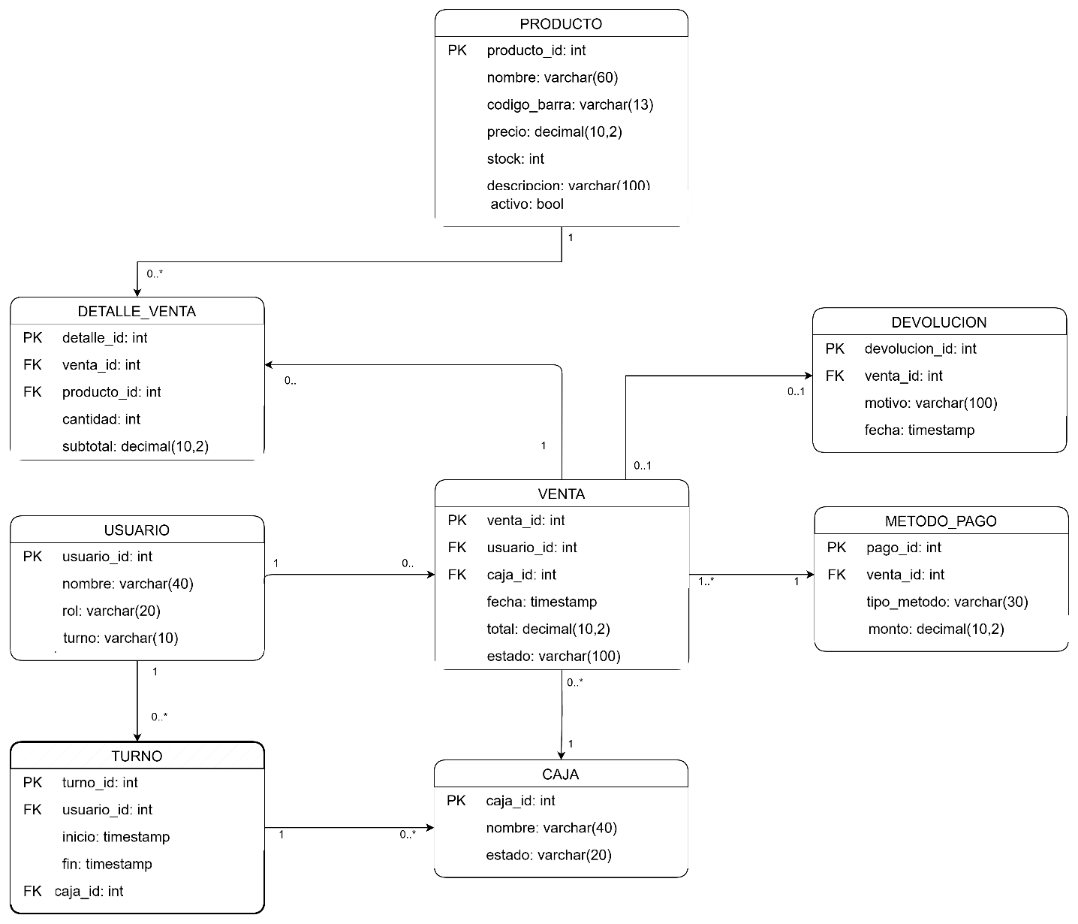


*Fuente: Elaborado por el estudiante de acuerdo con el proyecto*

6.2. Modelo de datos

A continuación, se describe el modelo de datos derivado del diagrama de clases presentado. Este modelo considera las entidades principales del sistema para el primer sprint, sus atributos, claves primarias (PK), claves foráneas (FK) y las relaciones entre ellas, como se muestra en la ***Figura 1.2***.

*Figura 1.2: “Modelo de datos primer sprint”*



*Fuente: Elaborado por el estudiante de acuerdo con el proyecto*

1. **VISTA PROCESO**

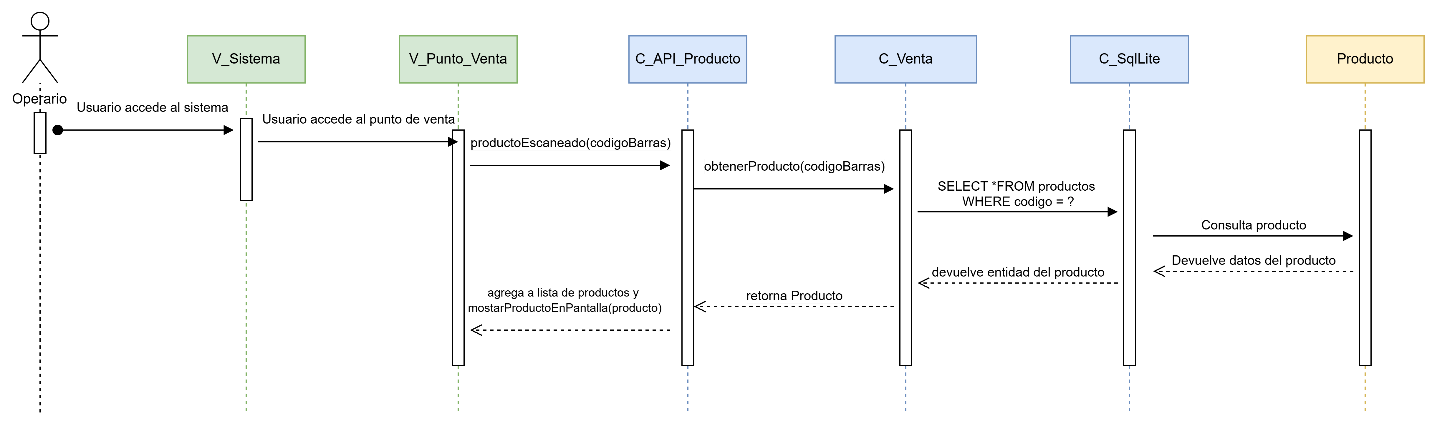
7.1. Diagrama de proceso

Los diagramas de secuencia permiten visualizar el flujo de mensajes entre los distintos actores y componentes del sistema a lo largo del tiempo. En este proyecto, estos diagramas ilustran cómo interactúan el operario, el sistema y otros elementos en cada caso de uso, detallando el orden y la lógica de las operaciones. Sirven como una herramienta clave para entender el comportamiento dinámico del sistema y asegurar una implementación coherente y eficiente.

|  |  |
| --- | --- |
| Caso de uso N° 1: | Leyendo códigos de barra |

Este diagrama muestra cómo, tras escanear un código de barras, el sistema consulta el producto en la base de datos a través de un controlador central y lo agrega al carrito de ventas, actualizando la interfaz del operario, como se ve en la ***Figura 2.1.***

*Figura 2.1: “Diagrama de secuencia CU N°1”*

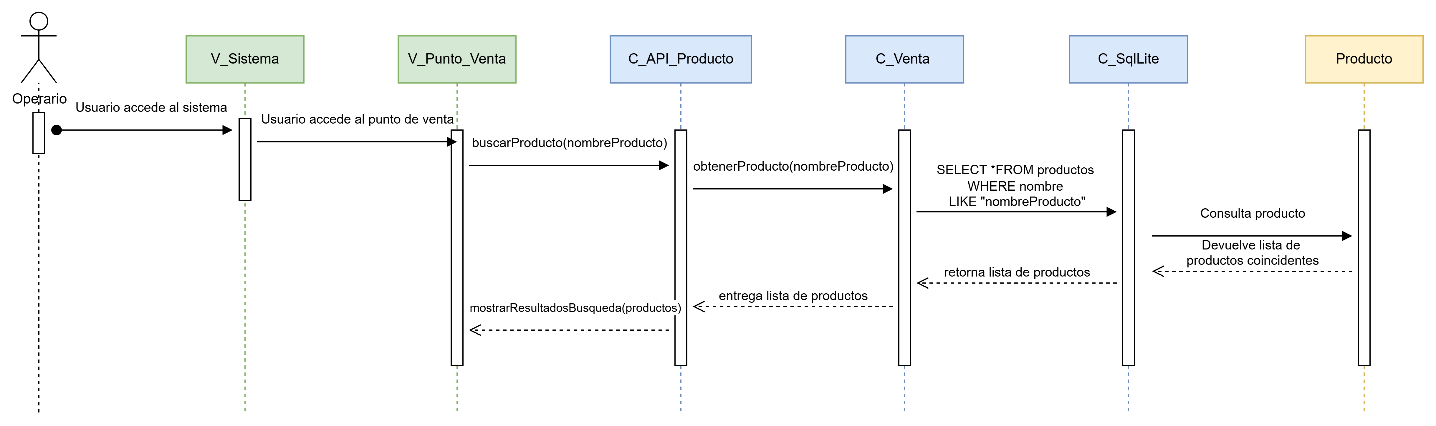


*Fuente: Elaborado por el estudiante de acuerdo con el proyecto*

|  |  |
| --- | --- |
| Caso de uso N° 2: | Buscando productos por nombre |

Desde la interfaz se envía la consulta a la API, que mediante el controlador accede a la entidad PRODUCTO para devolver coincidencias, como se ve en la ***Figura 2.2.***

*Figura 2.2: “Diagrama de secuencia CU N°2”*

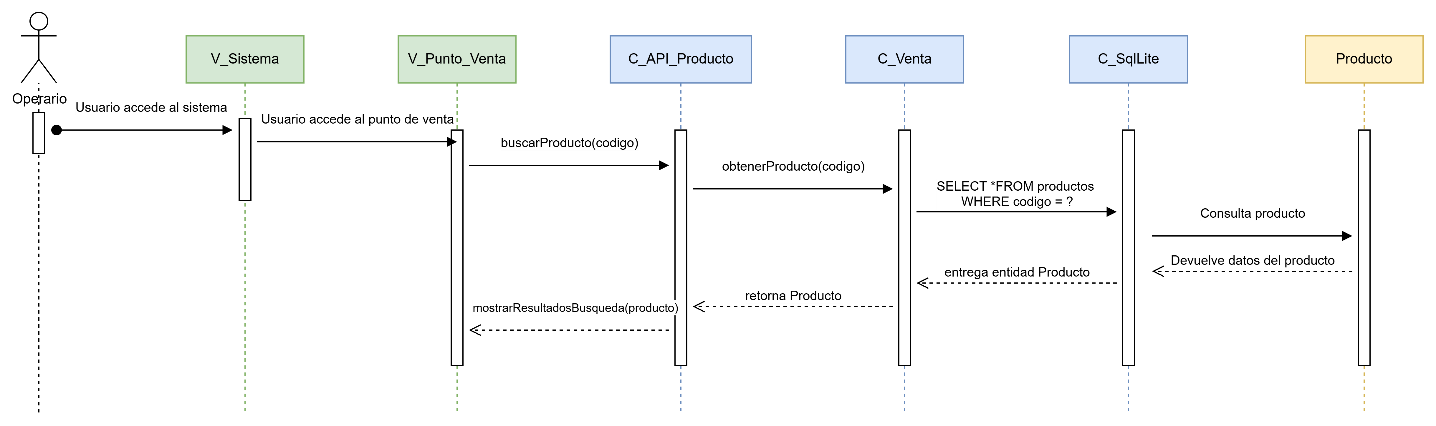


*Fuente: Elaborado por el estudiante de acuerdo con el proyecto*

|  |  |
| --- | --- |
| Caso de uso N° 3: | Buscando producto por código |

En este caso, el operario introduce un código en el campo de búsqueda. El sistema consulta la base de datos utilizando ese código y muestra el producto correspondiente si existe una coincidencia, como se ve en la *Figura 2.3.*

*Figura 2.3: “Diagrama de secuencia CU N°3”*

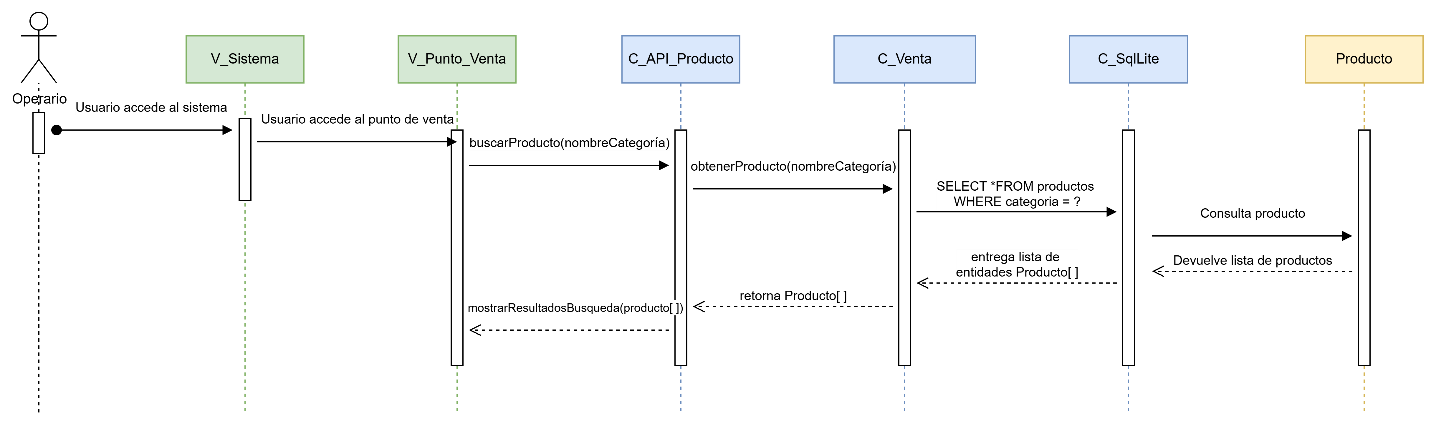


*Fuente: Elaborado por el estudiante de acuerdo con el proyecto*

|  |  |
| --- | --- |
| Caso de uso N° 4: | Buscando productos por categoría |

El diagrama refleja al operario ingresando una categoría. El sistema consulta los productos asociados a esa categoría en el inventario y devuelve la lista correspondiente al operario, como se muestra en la *Figura 2.4.*

*Figura 2.4: “Diagrama de secuencia CU N°4”*

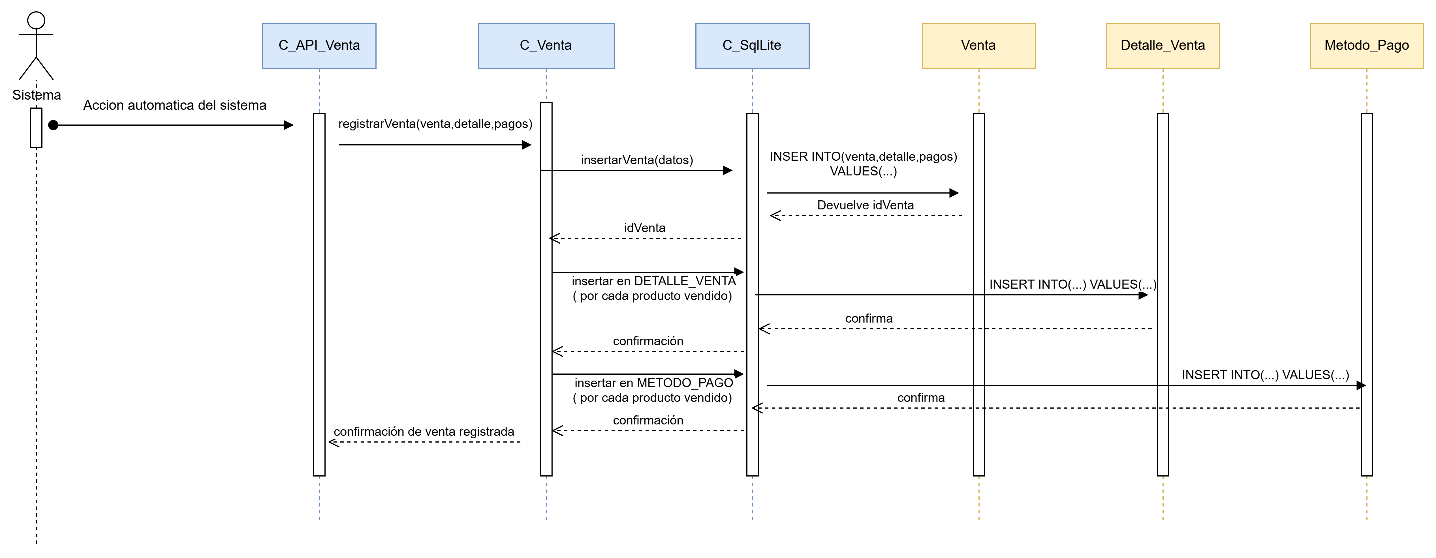


*Fuente: Elaborado por el estudiante de acuerdo con el proyecto*

|  |  |
| --- | --- |
| Caso de uso N° 5: | Registrando automáticamente las ventas |

Este diagrama muestra cómo, tras finalizar el proceso de pago, el sistema registra automáticamente la venta sin intervención del operario. Se genera una entrada en la base de datos de ventas con toda la información relevante, como se muestra en la ***Figura 2.5.***

*Figura 2.5: “Diagrama de secuencia CU N°5”*

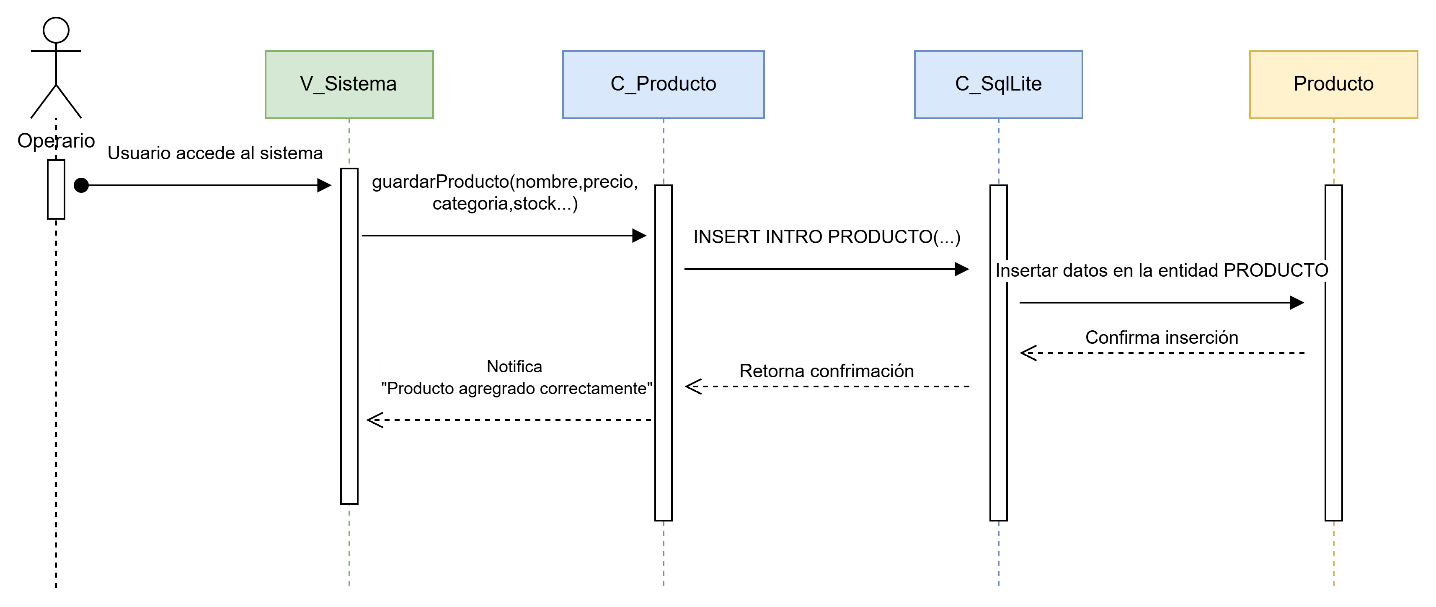


*Fuente: Elaborado por el estudiante de acuerdo con el proyecto*

|  |  |
| --- | --- |
| Caso de uso N°6: | Ingresando manualmente los productos |

En esta secuencia, el operario introduce manualmente los datos de un nuevo producto (nombre, precio, etc.). El sistema valida los campos y guarda la información en la base de datos del inventario, como se muestra en la ***Figura 2.6.***

*Figura 2.6: “Diagrama de secuencia CU N°6”*

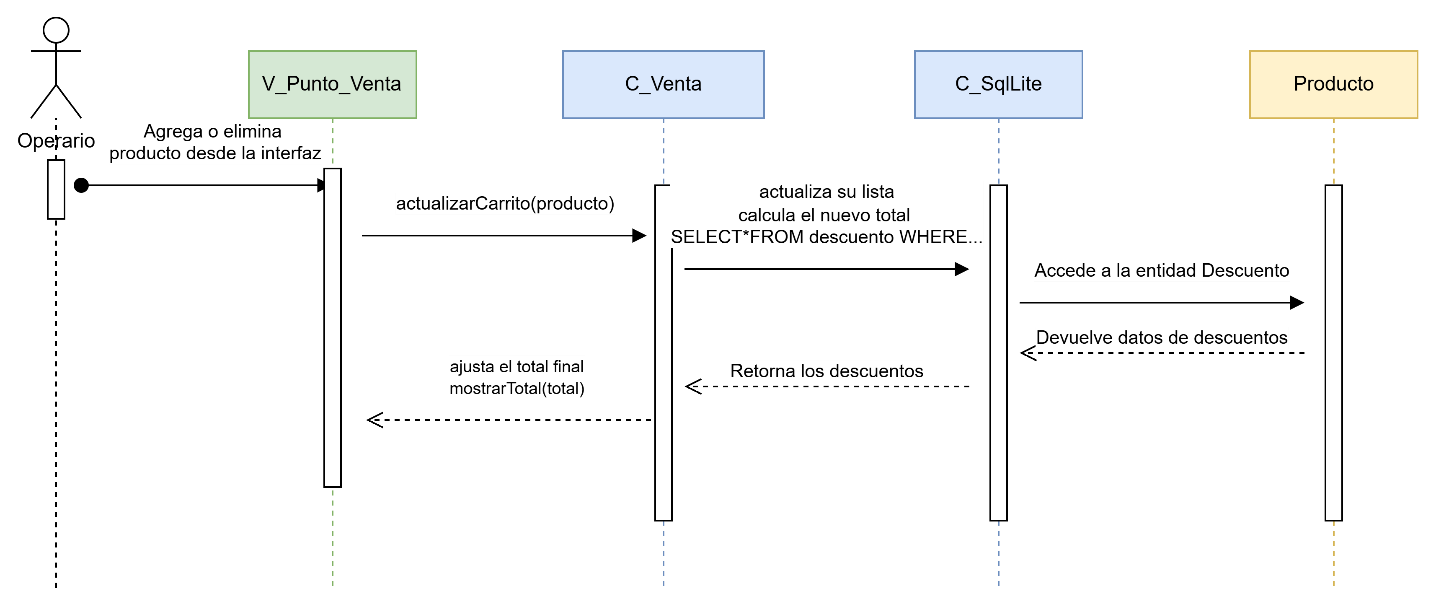


*Fuente: Elaborado por el estudiante de acuerdo con el proyecto*

|  |  |
| --- | --- |
| Caso de uso N°7: | Visualizando en tiempo real del total de venta |

El diagrama muestra que cada vez que el operario agrega o elimina un producto del carrito, el sistema recalcula el total de la venta automáticamente y actualiza la pantalla con el nuevo monto, como se muestra en la ***Figura 2.7***.

Figura 2.7: “Diagrama de secuencia CU N°7”

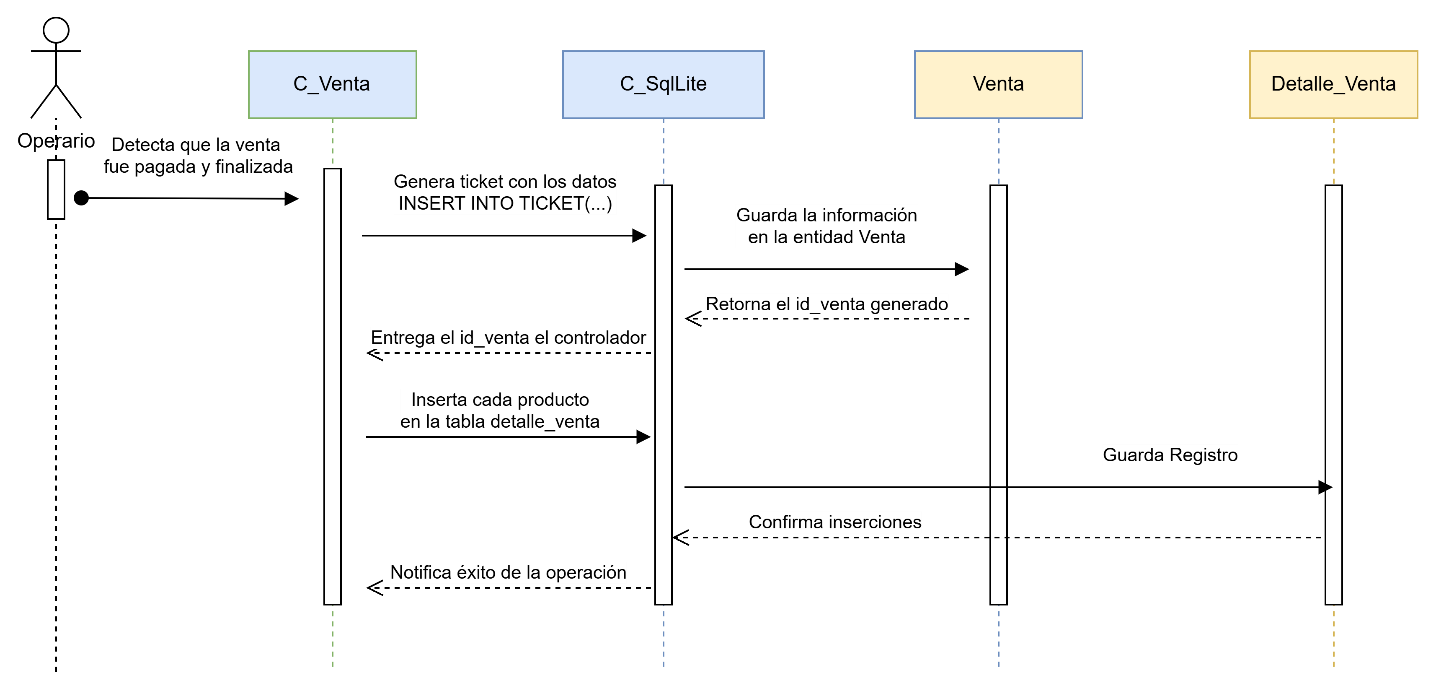


*Fuente: Elaborado por el estudiante de acuerdo con el proyecto*

|  |  |
| --- | --- |
| Caso de uso N°8: | Guardando en el sistema automáticamente el ticket detallado. |

El sistema, al momento de concretar el pago, genera un ticket con todos los detalles de la venta (productos, precios, descuentos, total) y lo guarda automáticamente en la base de datos, sin intervención manual, como se ve en la *Figura 2.8.*

*Figura 2.8: “Diagrama de secuencia CU N°8”*

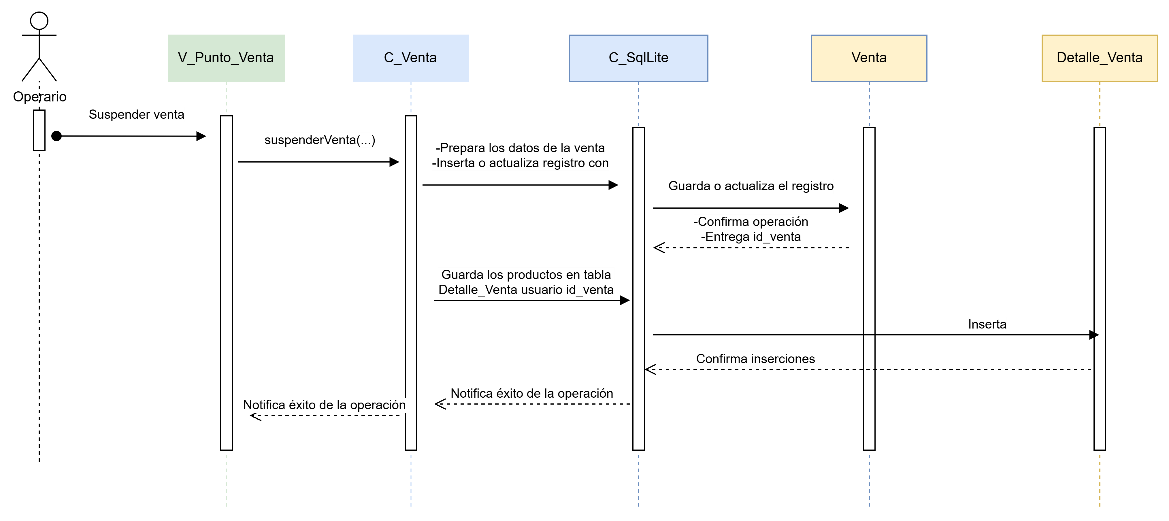
**

*Fuente: Elaborado por el estudiante de acuerdo con el proyecto*

|  |  |
| --- | --- |
| Caso de uso N°9: | Suspendiendo temporalmente la venta |

Este diagrama refleja que, al elegir la opción de suspender, el sistema guarda el estado actual del carrito y lo asocia a una lista de ventas en pausa, permitiendo reanudarla más tarde, como se ve en la ***Figura 2.9.***

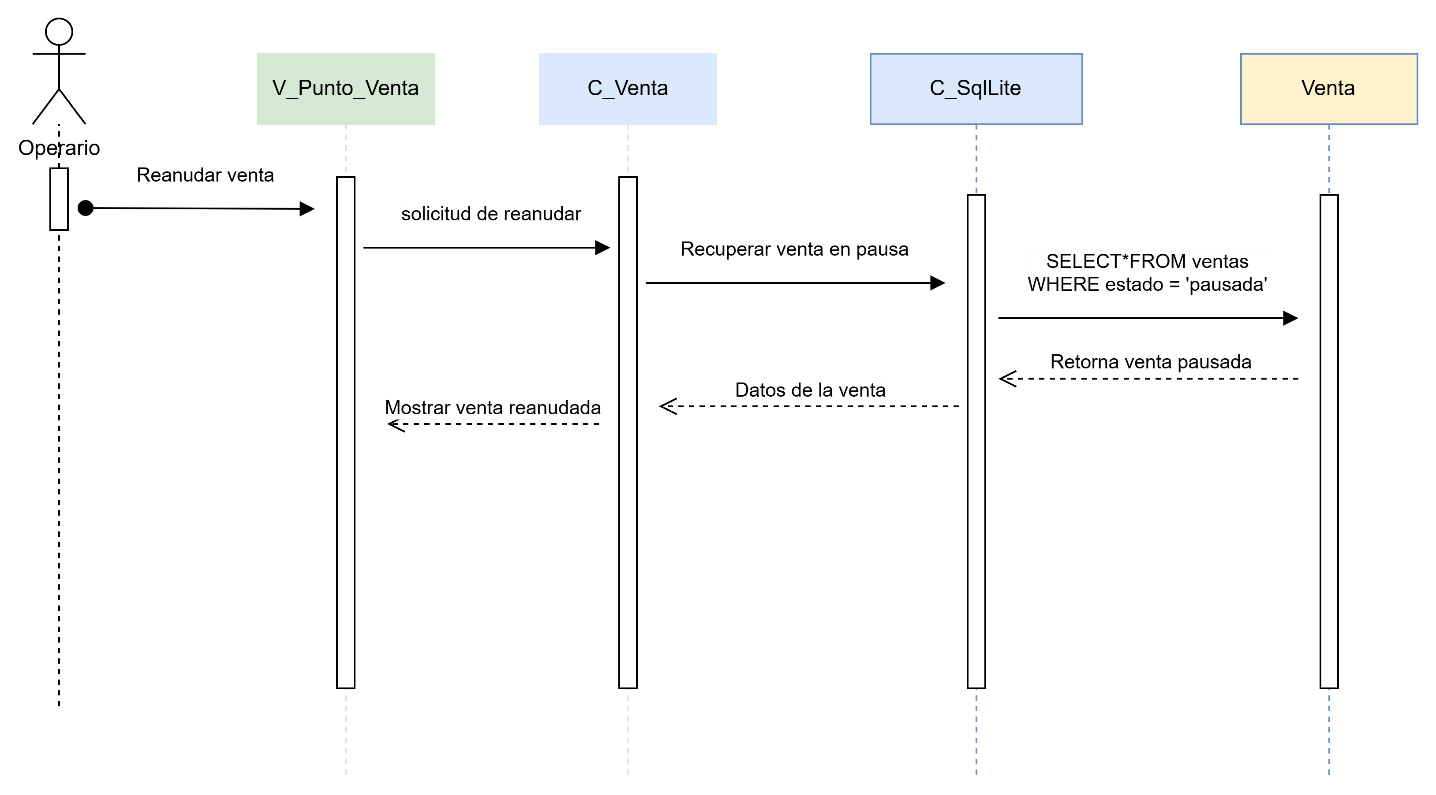
*Figura 2.9: “Diagrama de secuencia CU N°9.1”*



*Fuente: Elaborado por el estudiante de acuerdo con el proyecto*

Reanudación de venta suspendida ilustra el proceso mediante el cual el operario puede recuperar una venta previamente pausada y continuarla sin pérdida de datos, como se ve en la ***Figura 2.10***.

*Figura 2.10: “Diagrama de secuencia CU N°9.2”*

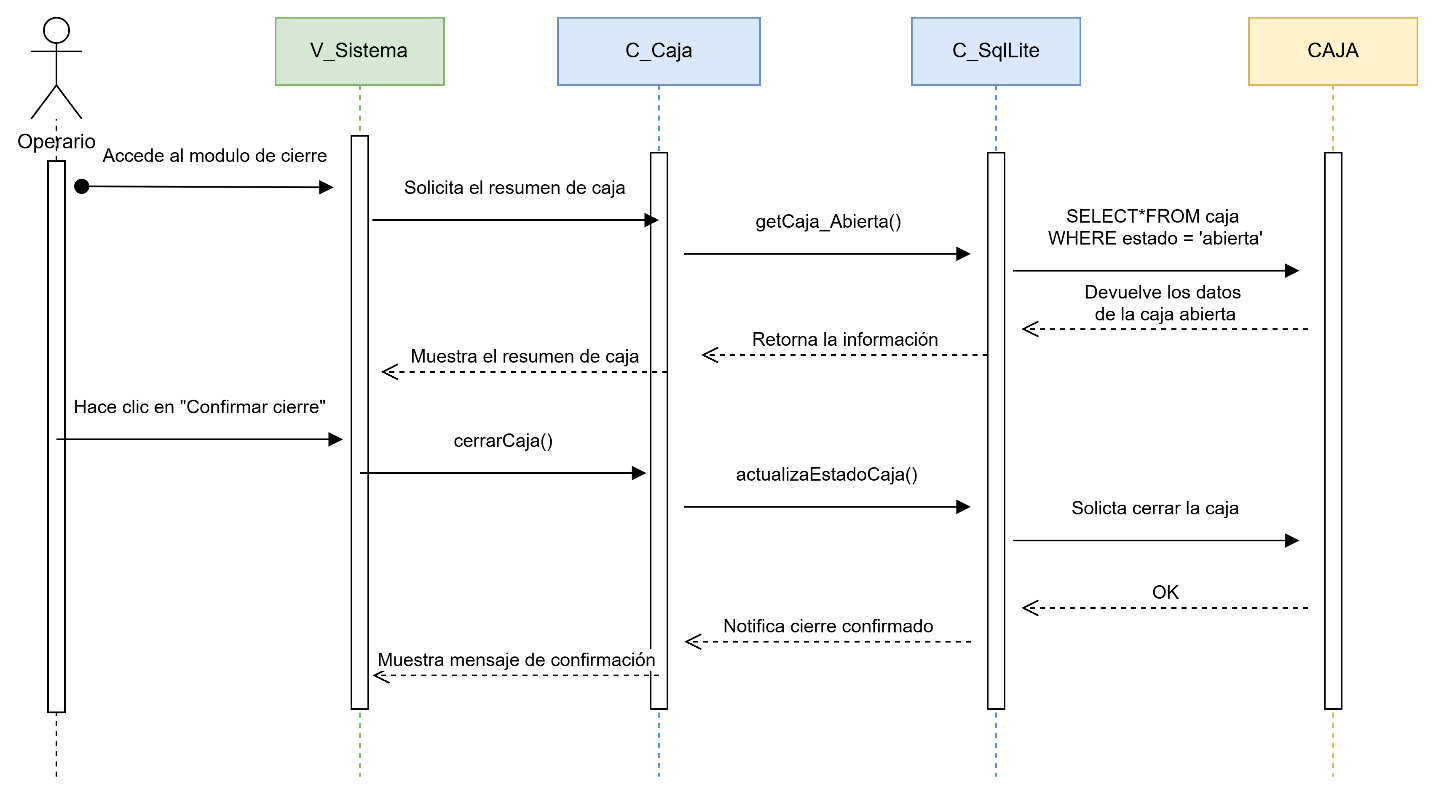


*Fuente: Elaborado por el estudiante de acuerdo con el proyecto*

|  |  |
| --- | --- |
| Caso de uso N°10: | Cerrando la caja |

El operario accede al módulo de cierre de caja, revisa el resumen y confirma. El sistema registra el cierre, genera un resumen final y guarda el reporte en el sistema, como se ve en la ***Figura 2.11.***

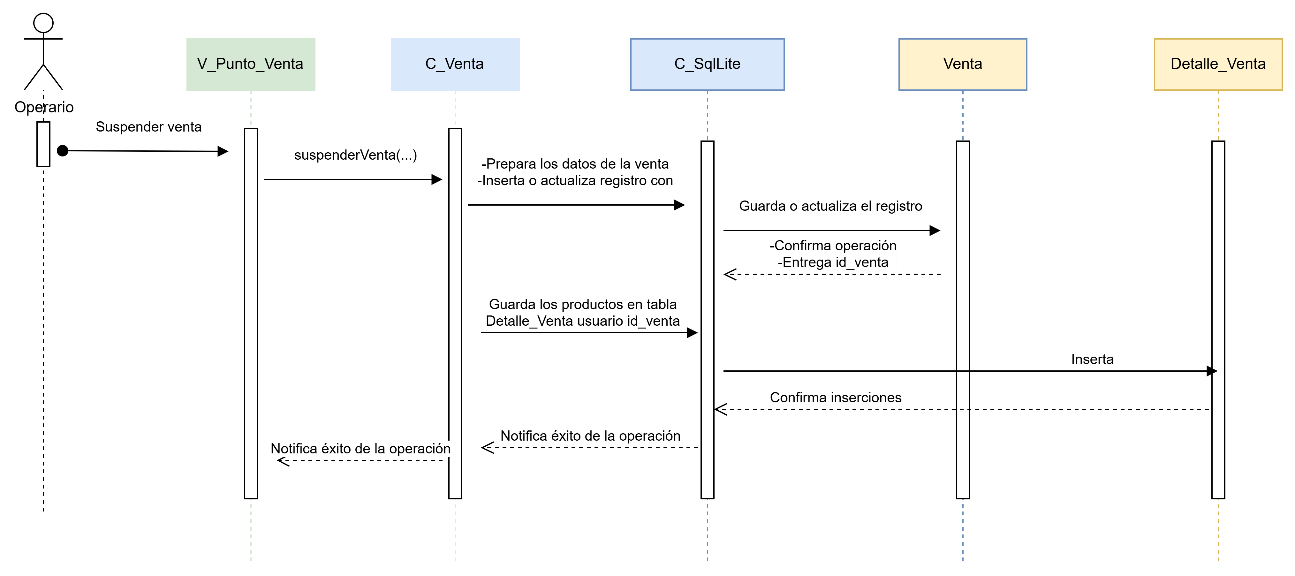
*Figura 2.11: “Diagrama de secuencia CU N°10”*



*Fuente: Elaborado por el estudiante de acuerdo con el proyecto*

|  |  |
| --- | --- |
| Caso de uso N°11: | Abriendo la caja. |

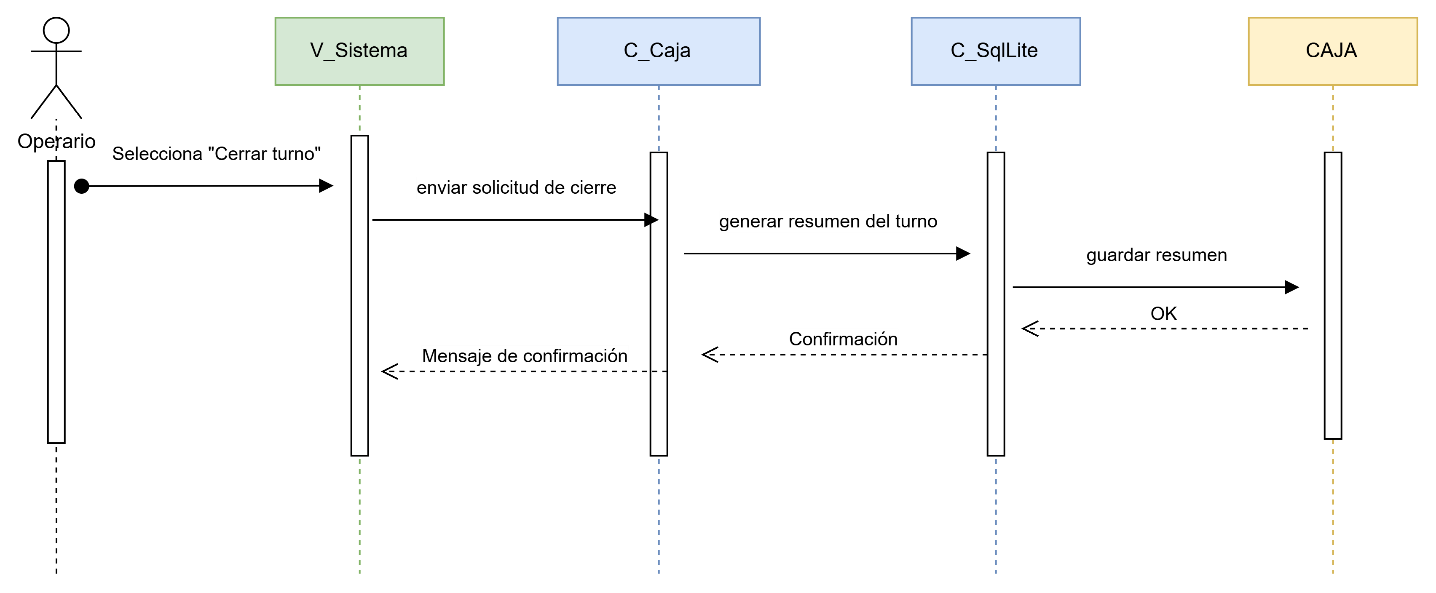
El operario inicia su jornada abriendo la caja. El sistema registra la hora de inicio y habilita el módulo de ventas, como se muestra en la ***Figura 2.12.***

*Figura 2.12: “Diagrama de secuencia CU N°11.1”*

*Fuente: Elaborado por el estudiante de acuerdo con el proyecto*

Cierre de turno: muestra el flujo que ocurre al finalizar la jornada laboral, registrando el cierre de caja y generando el resumen correspondiente, como se ve en ***la Figura 2.13.***

*Figura 2.13:” Diagrama de secuencia CU N°11.2”*

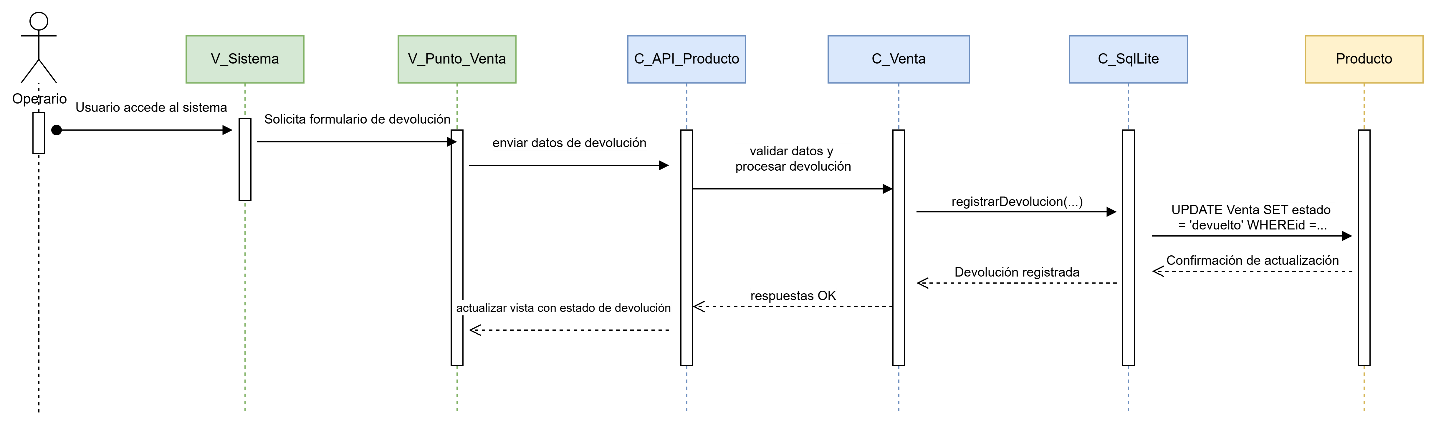


*Fuente: Elaborado por el estudiante de acuerdo con el proyecto*

|  |  |
| --- | --- |
| Caso de uso N°12: | Registrando devoluciones |

El operario accede al historial de ventas, selecciona una venta previa y el producto a devolver. Luego, ingresa el motivo y confirma. El sistema actualiza el stock e ingresa el registro de la devolución con fecha y hora, como se muestra en la ***Figura 2.14.***

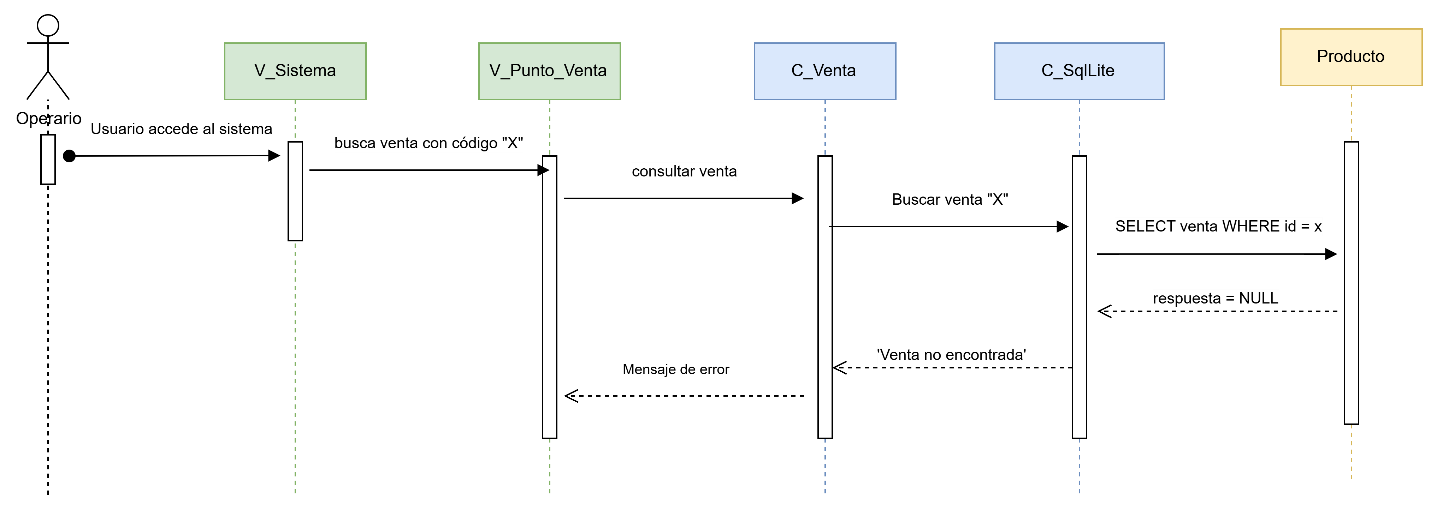
*Figura 2.14: “Diagrama de secuencia CU N°12.1”*



*Fuente: Elaborado por el estudiante de acuerdo con el proyecto*

Venta no encontrada en devolución: representa el manejo de error cuando se intenta registrar una devolución sobre una venta inexistente en el sistema, como se muestra en la ***Figura 2.15.***

*Figura 2.15: “Diagrama de secuencia CU N°12.2”*

**

*Fuente: Elaborado por el estudiante de acuerdo con el proyecto*

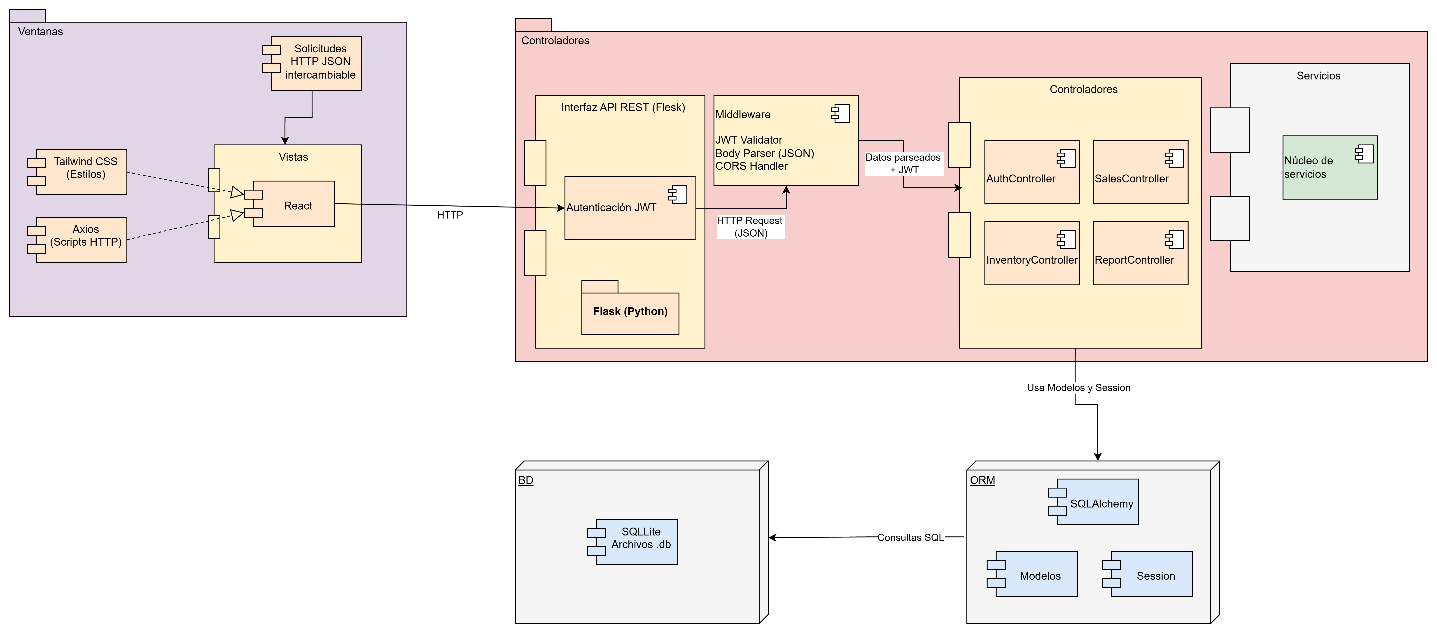
1. **VISTA DE DESARROLLO**

La vista de desarrollo representa la organización del software desde una perspectiva de implementación. En esta vista se identifican los principales componentes del sistema, sus tecnologías asociadas y las relaciones entre ellos.

8.1. Diagrama de componentes

Este diagrama representa la arquitectura del sistema, construido con las tecnologías de React, Flask (Python) y MySQL. Muestra la interacción entre los módulos clave, desde la recepción de peticiones HTTP hasta el acceso a la base de datos, pasando por capas de seguridad, procesamiento y lógica de negocio, como se muestra en la ***Figura 3.1.***

*Figura 3.1: “Diagrama de componentes”*



*Fuente: Elaborado por el estudiante de acuerdo con el proyecto*

1. **VISTA FISÍCA**

La vista física en el modelo 4+1 representa la arquitectura del sistema desde el punto de vista del despliegue en hardware y redes. Describe cómo los componentes de software se distribuyen en los distintos nodos físicos, como servidores, dispositivos y otros recursos de infraestructura. Esta vista muestra la topología del sistema, conexiones, comunicación entre nodos y cómo se implementan los requisitos no funcionales, como la escalabilidad, la disponibilidad y el rendimiento.

9.1. Diagrama de despliegue

El diagrama muestra cómo se distribuyen los componentes del sistema: el usuario accede desde su navegador al servidor web, que procesa las solicitudes y se comunica con la base de datos para almacenar o recuperar información. Todo está pensado para funcionar en línea, asegurando una arquitectura organizada y eficiente, como se muestra en la ***Figura 4.1.***

*Figura 4.1: “Diagrama de despliegue”*

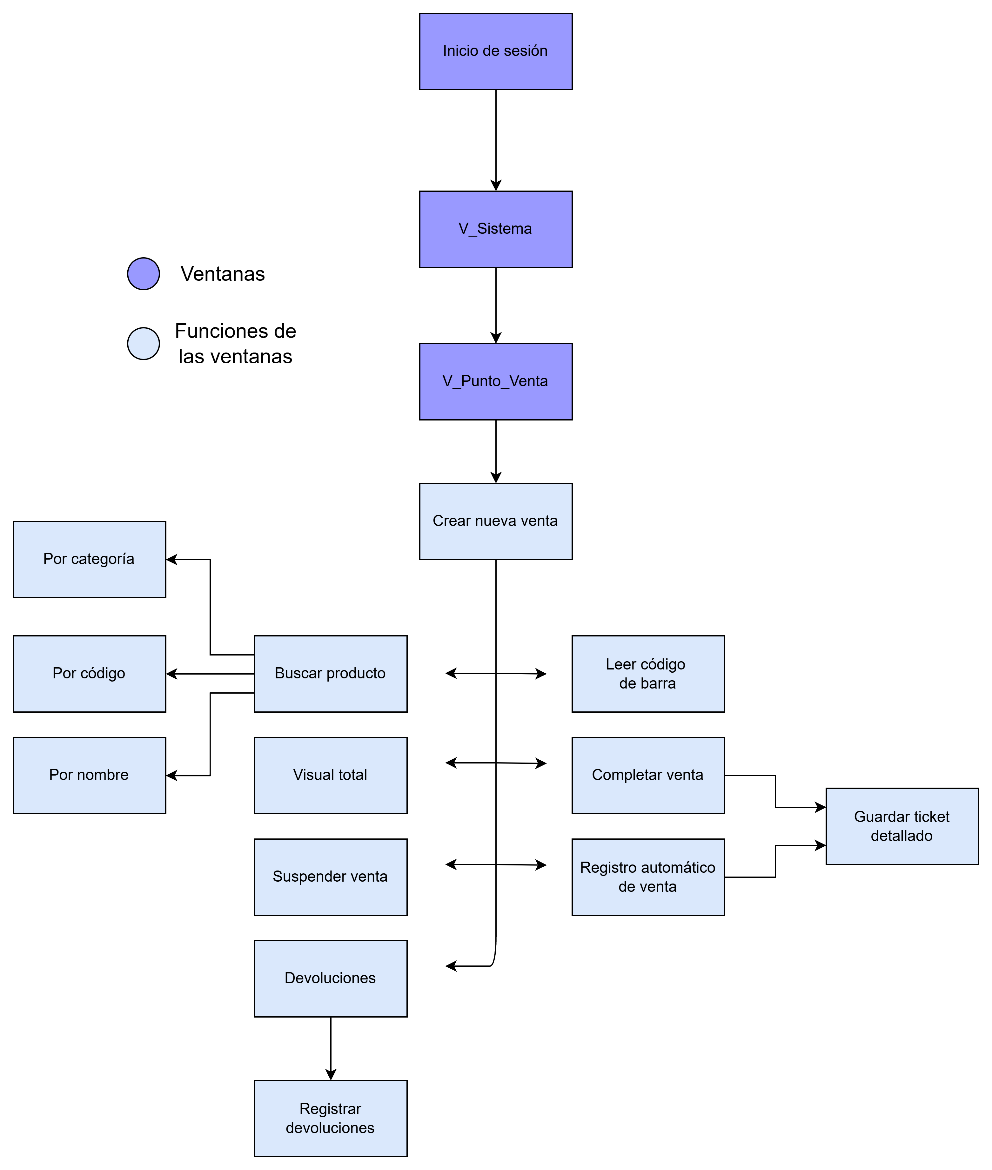
**

*Fuente: Elaborado por el estudiante de acuerdo con el proyecto*

1. **ÁRBOL DE NAVEGACIÓN**

El árbol de navegación muestra cómo el operario accede y realiza las funciones principales del sistema, desde abrir la caja, gestionar ventas con distintas opciones de ingreso de productos, hasta cerrar la caja y registrar devoluciones. Facilita una navegación ordenada y eficiente durante el proceso de venta, como se ve en la ***Figura 5.1.***

*Figura 5.1: “Árbol de navegación primer sprint”*

****

*Fuente: Elaborado por el estudiante de acuerdo con el proyecto*

1. **CASOS DE PRUEBAS**

En este capítulo se presentan los casos de prueba diseñados para validar el correcto funcionamiento del sistema desarrollado. Cada caso de prueba está orientado a verificar que las funcionalidades clave cumplan con los requerimientos establecidos, asegurando la calidad, estabilidad y confiabilidad del sistema ante distintos escenarios de uso.

|  |  |
| --- | --- |
| Caso de uso N° 1: | Leyendo códigos de barra |

Casos de Prueba:

1. Verificar que el lector de código de barra sea reconocido correctamente por el sistema.
2. Verificar que el producto escaneado se muestre correctamente en la interfaz.
3. Verificar que el producto se agregue automáticamente al carrito de ventas al escanearlo.

Resultados Esperados:

1. El sistema reconoce el lector de códigos de barra y lo habilita para lectura.
2. Al escanear un código, se muestran los datos del producto correspondiente.
3. El producto escaneado se agrega al carrito sin errores y con su cantidad ajustada.

Resultados Obtenidos:

1. El sistema reconoce el lector sin problemas y está listo para escanear.
2. Al escanear, el producto correcto se muestra en pantalla.
3. El producto es añadido al carrito de forma inmediata con todos sus datos.

Evidencia:

Se muestra ***Figura 6.1*** la cual es una imagen de interfaz mostrando el carrito de ventas con un producto recién escaneado y agregado automáticamente.

*Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.Figura 6.1: “Evidencia de caso de prueba N°1”*

*Fuente: Elaborado por el estudiante de acuerdo con el proyecto*

|  |  |
| --- | --- |
| Caso de uso N° 2: | Buscando productos por nombre |

Casos de Prueba:

1. Verificar que el campo de búsqueda esté disponible y sea editable.
2. Verificar que se muestren resultados correctos al ingresar un nombre válido.
3. Verificar que se gestione correctamente cuando no hay coincidencias.

Resultados Esperados:

1. El campo de búsqueda aparece visible y permite ingresar texto.
2. Al ingresar un nombre válido, el sistema muestra una lista de productos que coinciden.
3. Si no hay coincidencias, se muestra un mensaje informativo sin errores.

Resultados Obtenidos:

1. El campo de búsqueda está disponible y es funcional.
2. Se muestran los productos relacionados correctamente al ingresar el nombre.
3. Se muestra mensaje “No se encontraron resultados” cuando corresponde.

Evidencia:

Se muestra ***Figura 6.2***, una imagen mostrando el campo de búsqueda con resultados listados tras ingresar un nombre.

*Figura 6.2: “Evidencia de caso de prueba N°2”*

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.  
*Fuente: Elaborado por el estudiante de acuerdo con el proyecto*

|  |  |
| --- | --- |
| Caso de uso N° 3: | Buscando producto por código |

Casos de Prueba:

1. Verificar que se permita ingresar un código de producto.
2. Verificar que al ingresar un código válido, se muestre el producto correspondiente.
3. Verificar el comportamiento ante un código no existente.

Resultados Esperados:

1. El campo para ingresar el código está visible y funcional.
2. El producto exacto se muestra cuando el código es correcto.
3. Si el código no existe, se informa al usuario.

Resultados Obtenidos:

1. El campo es funcional y admite texto numérico.
2. El sistema muestra el producto exacto asociado al código.
3. Se muestra un mensaje “Producto no encontrado” si no existe coincidencia.

Evidencia:

Se muestra ***Figura 6.3***, imagen mostrando el resultado de una búsqueda exitosa por código.

*Figura 6.3:” Evidencia de caso de prueba N°3”*

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.  
*Fuente: Elaborado por el estudiante de acuerdo con el proyecto*

|  |  |
| --- | --- |
| Caso de uso N° 4: | Buscando productos por categoría |

Casos de Prueba:

1. Verificar que se pueda seleccionar o escribir una categoría.
2. Verificar que se muestren los productos que pertenecen a la categoría seleccionada.
3. Verificar el manejo cuando no existen productos en dicha categoría.

Resultados Esperados:

1. La interfaz permite seleccionar una categoría válida.
2. Se muestran los productos asociados a la categoría elegida.
3. Si no hay productos, el sistema lo informa sin errores.

Resultados Obtenidos:

1. El sistema despliega el listado de categorías correctamente.
2. Al seleccionar una categoría, se muestran sus productos asociados.
3. Se muestra mensaje informativo si no hay productos disponibles.

Evidencia:

Se muestra ***Figura 6.4***, imagen con lista de productos filtrados por categoría.

*Figura 6.4: “Evidencia de caso de prueba N°4”*

Interfaz de usuario gráfica, Texto

El contenido generado por IA puede ser incorrecto.  
*Fuente: Elaborado por el estudiante de acuerdo con el proyecto*

|  |  |
| --- | --- |
| Caso de uso N° 5: | Registrando automáticamente las ventas |

Casos de Prueba:

1. Verificar que al finalizar la venta, el sistema registre automáticamente la información.
2. Verificar que los datos de la venta aparezcan en el historial.
3. Verificar que no haya duplicación ni pérdida de datos.

Resultados Esperados:

1. Al finalizar la venta, se almacena correctamente en el sistema.
2. La venta registrada es visible en el historial con todos sus detalles.
3. No se duplican ventas ni hay inconsistencias.

Resultados Obtenidos:

1. El sistema registra automáticamente la venta sin intervención del operario.
2. La venta aparece reflejada correctamente en el historial.
3. No se detectaron errores ni duplicaciones.

Evidencia:

Se muestra ***Figura 6.5***, imagen del historial de ventas mostrando el último registro guardado automáticamente.

Figura 6.5: “Evidencia de caso de prueba N°5”

Una captura de pantalla de una red social

El contenido generado por IA puede ser incorrecto.  
*Fuente: Elaborado por el estudiante de acuerdo con el proyecto*

|  |  |
| --- | --- |
| Caso de uso N°6: | Ingresando manualmente los productos |

Casos de Prueba:

1. Verificar que el formulario de ingreso manual esté disponible y permita ingresar todos los datos.
2. Verificar que el sistema almacene correctamente la información ingresada.
3. Verificar el manejo de errores en caso de campos vacíos o inválidos.

Resultados Esperados:

1. El formulario aparece completo y editable.
2. El producto se agrega exitosamente al inventario.
3. El sistema valida correctamente los campos.

Resultados Obtenidos:

1. El formulario está disponible y funcional.
2. El producto es guardado correctamente al completar los datos.
3. Se muestran mensajes de error si faltan campos.

Evidencia:

Se muestra ***Figura 6.6***, imagen del formulario completado y confirmación de producto agregado.

*Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.Figura 6.6: “Evidencia de caso de prueba N° 6”*

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

*Fuente: Elaborado por el estudiante de acuerdo con el proyecto.*

|  |  |
| --- | --- |
| Caso de uso N°7: | Visualizando en tiempo real del total de venta |

Casos de Prueba:

1. Verificar que el total de la venta se actualice al agregar un producto.
2. Verificar que se actualice también al eliminar un producto.
3. Verificar que el total mostrado sea correcto en todo momento.

Resultados Esperados:

1. Al agregar productos, el total se recalcula instantáneamente.
2. Al eliminar productos, el total se ajusta correctamente.
3. El total siempre refleja el contenido actual del carrito.

Resultados Obtenidos:

1. El total se actualiza automáticamente al modificar el carrito.
2. Se recalcula sin errores ni retrasos.
3. El total mostrado es correcto en todas las pruebas.

Evidencia:

Se muestra ***Figura 6.7***, imagen con carrito actualizado y total ajustado en tiempo real.

*Figura 6.7: “Evidencia de caso de prueba N°7”*

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.  
*Fuente: Elaborado por el estudiante de acuerdo con el proyecto*

|  |  |
| --- | --- |
| Caso de uso N°8: | Guardando en el sistema automáticamente el ticket detallado. |

Casos de Prueba:

1. Verificar que, al finalizar la venta, se genere y guarde automáticamente un ticket.
2. Verificar que el ticket contenga todos los detalles necesarios.
3. Verificar que el ticket esté disponible para consultas posteriores.

Resultados Esperados:

1. El ticket se genera automáticamente.
2. Contiene productos, precios, descuentos y totales.
3. Es accesible desde el historial de ventas.

Resultados Obtenidos:

1. El ticket se crea y guarda automáticamente.
2. Se validó que los datos contenidos sean completos.
3. El ticket aparece en la base de datos y puede ser consultado.

Evidencia:

Se muestra ***Figura 6.8***, imagen del ticket generado con detalle completo de la venta.

*Figura 6.8: “Evidencia de caso de prueba N°8”*

Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.  
*Fuente: Elaborado por el estudiante de acuerdo con el proyecto*

|  |  |
| --- | --- |
| Caso de uso N°9: | Suspendiendo temporalmente la venta |

Casos de Prueba:

1. Verificar que la opción “suspender venta” esté disponible.
2. Verificar que la venta actual se guarde en estado temporal.
3. Verificar que la venta pueda retomarse luego sin pérdida de datos.

Resultados Esperados:

1. El sistema muestra la opción de suspender.
2. La venta queda en estado “en pausa”.
3. La venta se puede reanudar con todos sus productos intactos.

Resultados Obtenidos:

1. La opción está disponible y funcional.
2. La venta queda correctamente guardada en pausa.
3. Se retoma con éxito sin pérdida de información.

Evidencia:  
Se muestra ***Figura 6.9***, imagen mostrando una venta en estado suspendido y opción para retomarla.

*Figura 6.9: “Evidencia de caso de prueba N°9”*

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

*Fuente: Elaborado por el estudiante de acuerdo con el proyecto*

|  |  |
| --- | --- |
| Caso de uso N°10: | Cerrando la caja |

Casos de Prueba:

1. Verificar que se pueda acceder al módulo de cierre.
2. Verificar que se muestre correctamente el resumen de caja.
3. Verificar que el sistema registre y guarde el cierre correctamente.

Resultados Esperados:

1. El módulo de cierre está disponible.
2. El resumen presenta todas las operaciones del turno.
3. El cierre queda registrado con hora y totales.

Resultados Obtenidos:

1. Se accede sin problemas al módulo.
2. El resumen es completo y claro.
3. El sistema guarda el cierre correctamente.

Evidencia:

Se muestra ***Figura 6.10***, imagen del resumen de caja y confirmación de cierre registrado.

*Figura 6.10: “Evidencia de caso de prueba N°10”*

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

*Fuente: Elaborado por el estudiante de acuerdo con el proyecto*

|  |  |
| --- | --- |
| Caso de uso N°11: | Abriendo la caja. |

Casos de Prueba:

1. Verificar que se pueda iniciar la jornada desde el sistema.
2. Verificar que el sistema registre la apertura con hora.
3. Verificar que al cierre se registre toda la información del turno.

Resultados Esperados:

1. El operario puede abrir la caja al iniciar su turno.
2. Se registra correctamente la hora de inicio.
3. El sistema asocia los datos del turno (ventas, montos).

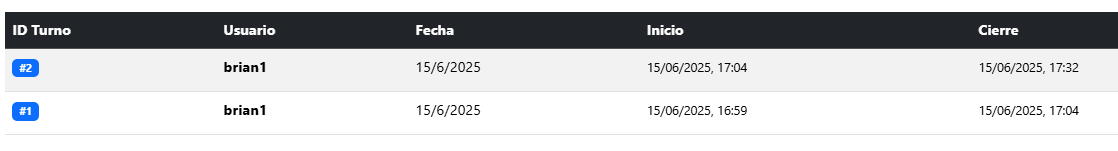
Resultados Obtenidos:

1. La apertura se realiza correctamente.
2. La hora queda registrada automáticamente.
3. Los datos del turno quedan asociados al operario.

Evidencia:

Se muestra ***Figura 6.1***, imagen del registro de apertura de caja con hora y datos iniciales.

*Figura 6.11: “Evidencia de caso de prueba N°11”*

  
*Fuente: Elaborado por el estudiante de acuerdo con el proyecto*

|  |  |
| --- | --- |
| Caso de uso N°12: | Registrando devoluciones |

Casos de Prueba:

1. Verificar que el operario pueda acceder al historial de ventas.
2. Verificar que pueda seleccionar la venta y los productos a devolver.
3. Verificar que el sistema actualice el stock y registre la devolución con motivo.

Resultados Esperados:

1. El historial de ventas está disponible para consultas.
2. El sistema permite seleccionar productos devueltos.
3. La devolución queda registrada con motivo, fecha y hora.

Resultados Obtenidos:

1. El operario accede fácilmente al historial.
2. Los productos son seleccionados sin errores.
3. La devolución se guarda correctamente y se actualiza el stock.

Evidencia:  
Se muestra ***Figura 6.12***, imagen mostrando la pantalla de devolución con productos seleccionados y motivo ingresado.

*Figura 6.12: “Evidencia de caso de prueba N°12”*

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

*Fuente: Elaborado por el estudiante de acuerdo con el proyecto*

1. **Gráficos de esfuerzos**

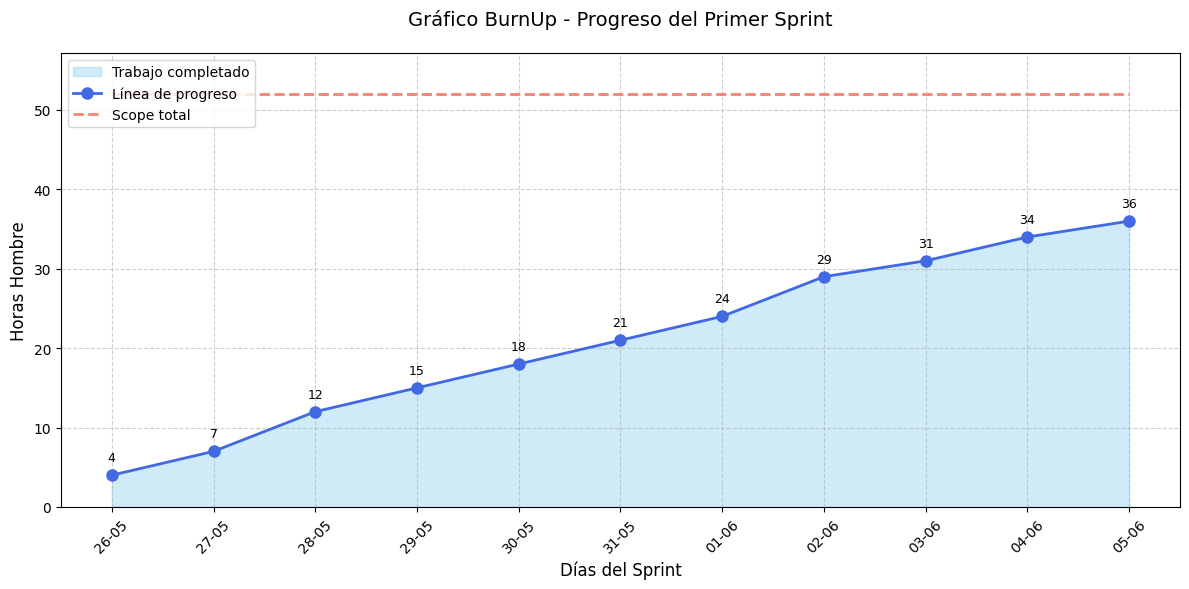
Para asegurar un control eficiente del desarrollo y cumplimiento de los objetivos del sistema, se utilizaron herramientas visuales de seguimiento que permiten medir el avance real frente a lo planificado. En esta sección se presentan los gráficos de esfuerzos, Burn Up y Burn Down, fundamentales en metodologías ágiles como Scrum o XP.

Estos gráficos entregan información clara sobre el trabajo realizado, lo que aún queda por completar y cómo se está comportando el equipo frente a los plazos establecidos. Su análisis permite detectar posibles desviaciones, identificar cuellos de botella y tomar decisiones tempranas que contribuyan al éxito del proyecto.

A continuación, se presentan los gráficos generados durante el desarrollo, junto a una breve interpretación de cada uno.

El gráfico Burn-Up muestra el progreso acumulado del equipo durante el sprint, comparando el trabajo completado (línea azul) con el scope total planeado (línea roja). Permite visualizar cuánto del trabajo comprometido se ha finalizado y si se han agregado nuevos requisitos durante la iteración, como se ve en la ***Figura 7.1.***

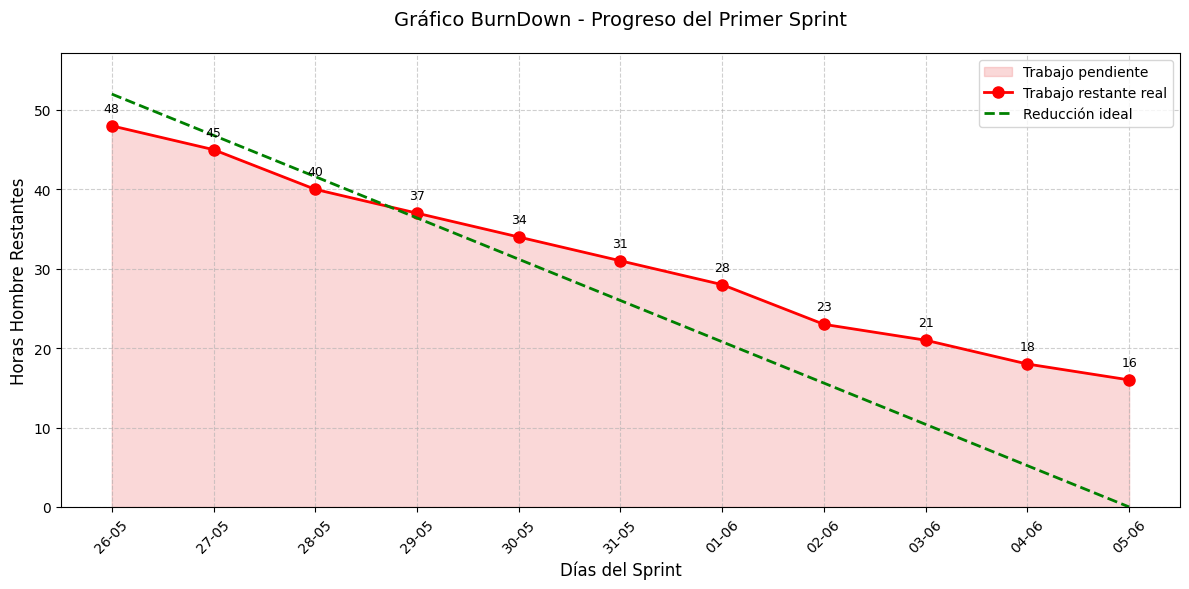
*Figura 7.1: “Gráfico Burn-Up”*



*Fuente: Elaborado por el estudiante de acuerdo con el proyecto.*

El gráfico Burn-Down ilustra el trabajo pendiente a lo largo del sprint, contrastando el progreso real (línea roja) con la reducción ideal (línea verde). Es útil para identificar desviaciones del plan y prever si el equipo completará todo el trabajo comprometido a tiempo, como se ve en la ***Figura 7.2***.

*Figura 7.2: “Gráfico Burn-Down”*



*Fuente: Elaborado por el estudiante de acuerdo con el proyecto.*

1. **CONCLUSIÓN**

El primer incremento del proyecto “Scrum++: Sistema de Gestión Integral para Amasandería y Almacén 98” estableció las funcionalidades esenciales para la operación diaria del negocio, priorizando procesos clave como ventas y gestión de caja. La documentación técnica, respaldada por diagramas UML (clases, secuencias, componentes y despliegue), asegura una base sólida para el desarrollo futuro, facilitando la escalabilidad y el mantenimiento del sistema.

Los casos de uso implementados fueron validados mediante pruebas, demostrando que el sistema cumple con los objetivos iniciales de eficiencia y automatización. Este avance representa un primer paso crucial, pero el proyecto requerirá iteraciones posteriores para integrar el resto de los requerimientos y optimizar su desempeño.

En resumen, el equipo logró entregar un producto funcional y bien estructurado, alineado con las necesidades del cliente y con un enfoque claro en la mejora continua. El trabajo realizado sienta las bases para evolucionar hacia un sistema completo y robusto.