# DOCUMENTO DE DISEÑO

<<Nombre de la solución informática>>.

Versión: <<Número de la versión del documento>>

<<Nombre de la firma o proveedor que realiza el documento>>

<<Nombre del autor del documento>>

Documento de diseño. Secretaría Distrital de Salud

<<Nombre de la solución informática>>. Bogotá, <<Fecha de creación del documento>>

# REVISIÓN Y CONTROL DE CAMBIOS

**Revisión y Versiones**

|  |  |  |  |
| --- | --- | --- | --- |
| **Nombres y Apellidos** | **Cargo** | **Versión Aprobada** | **Fecha** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Control de Cambios**

|  |  |  |  |
| --- | --- | --- | --- |
| **Fecha** | **Autor** | **Versión** | **Descripción del cambio** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**TABLA DE CONTENIDO**

<<Agregue aquí la tabla de contenido>>

## TABLA DE IMÁGENES

<<Agregue aquí la tabla de ilustraciones>>

# Resumen

<<Descripción: En esta sección se debe realizar una presentación del documento especificando el propósito y enumerar las secciones y contenidos que lo componen>>

# Generalidades

## Propósito del sistema.

El propósito fundamental de este sistema se centra en la creación y gestión eficiente de una aplicación diseñada para facilitar la administración integral de un establecimiento dedicado a la venta de verduras y frutas. La arquitectura del sistema se concibe con el objetivo de optimizar la operatividad y la efectividad en la gestión de la base de datos, así como en la supervisión de las transacciones de venta y compra.

La aplicación se desarrollará con una visión global que abarcará desde la interfaz de usuario hasta la estructura de la base de datos, asegurando una experiencia fluida y transparente para los usuarios finales. Se tomarán decisiones cuidadosas en cuanto a la eficiencia del sistema y su portabilidad, buscando un equilibrio que garantice un rendimiento óptimo sin comprometer la accesibilidad en diferentes entornos.

Además, este párrafo también abordará las influencias de los requisitos funcionales y no funcionales del sistema, estableciendo las prioridades y decisiones clave en la fase de diseño. Se destacarán consideraciones como la capacidad de respuesta del sistema, la seguridad de los datos y la interoperabilidad con otras plataformas, asegurando así que el desarrollo de la aplicación se alinee de manera integral con los objetivos generales del proyecto.

## Objetivos del diseño

1. **Distribución Eficiente:**
   * **Descripción:** Garantizar una distribución eficiente de frutas y verduras en un área específica.
   * **Justificación:** Optimizar la logística de distribución contribuye a la frescura de los productos y mejora la satisfacción del cliente.
2. **Ventas por Mayoreo y Menudeo:**
   * **Descripción:** Permitir ventas tanto por mayoreo (por caja) como por menudeo (por piezas o kilos).
   * **Justificación:** Ofrecer flexibilidad en las opciones de compra satisface las necesidades de diferentes tipos de clientes.
3. **Gestión de Gastos de Traslado:**
   * **Descripción:** Integrar un sistema para gestionar y registrar los gastos de traslado.
   * **Justificación:** Facilitar el control financiero y evaluar la rentabilidad de las operaciones de distribución.
4. **Planes para Productos Próximos a Vencer:**
   * **Descripción:** Desarrollar estrategias para minimizar pérdidas en caso de productos cercanos a su fecha de vencimiento.
   * **Justificación:** Optimizar la gestión de inventarios y reducir pérdidas económicas.
5. **Compatibilidad de Métodos de Venta:**
   * **Descripción:** Permitir la implementación simultánea de los métodos de venta por mayoreo y menudeo.
   * **Justificación:** Aprovechar la versatilidad del negocio para llegar a diferentes segmentos de clientes.
6. **Control de Stock y Ventas para Vendedores:**
   * **Descripción:** Crear una aplicación móvil exclusiva para vendedores con funciones de control de stock y ventas.
   * **Justificación:** Facilitar la gestión diaria de los vendedores sin acceso a PC, mejorando la eficiencia operativa.
7. **Registro Detallado de Cajas:**
   * **Descripción:** Implementar un sistema detallado que registre la distribución, ventas y pérdidas de cada caja de productos.
   * **Justificación:** Proporcionar información precisa para la toma de decisiones y la mejora continua del proceso.
8. **Gestión de Créditos para Clientes:**
   * **Descripción:** Asignar límites de crédito a clientes que optan por esta forma de pago.
   * **Justificación:** Evitar pérdidas importantes al limitar la provisión de mercancía cuando se excede el crédito establecido.

Estos objetivos de diseño buscan cumplir con los requisitos no funcionales y los estándares de calidad del sistema, priorizando la eficiencia operativa, la flexibilidad en las ventas, y la optimización de procesos para garantizar el éxito

## Definiciones, estereotipos UML y abreviaturas

## Grupo de Términos Relacionados con la Gestión del Sistema:

* Aplicación Móvil: La aplicación diseñada para dispositivos móviles que permite a los vendedores gestionar el stock y realizar ventas.
* CompraManager: Clase encargada de gestionar y asignar procesos de compra de productos.
* ControlCajas: Clase que gestiona el control detallado del número de cajas, distribución, y registra ventas y pérdidas.
* ControlCredito: Clase encargada de administrar los límites de crédito de los clientes.
* DistribucionManager: Clase responsable de coordinar la logística de distribución y asignar procesos de traslado.
* Gastos de Traslado: Los costos asociados al transporte de productos desde el proveedor hasta el negocio.
* Producto: Representa una unidad de fruta o verdura con información asociada, como nombre, tipo y fecha de vencimiento.
* ProductoAdmin: Clase encargada de dar de alta productos en el sistema.
* Subsistema: Un conjunto de componentes o clases que trabajan juntos para cumplir una función específica dentro del sistema.
* VentaManager: Clase responsable de gestionar las estrategias de venta, incluyendo la venta por caja o por unidad.
* VentaProcesoManager: Clase encargada de asignar y gestionar procesos de venta, teniendo en cuenta los gastos de traslado.
* AppMovilVendedor: Representa la aplicación móvil exclusiva para vendedores, facilitando la gestión de stock y ventas.
* Grupo de Términos Relacionados con Clientes:
* Cliente: Representa a un individuo o entidad que realiza compras en el sistema de gestión de frutas y verduras.
* Crédito: La posibilidad para los clientes de realizar compras y pagar en un momento posterior.
* Límite de Crédito: El monto máximo permitido para compras a crédito por parte de un cliente.
* Pago en Efectivo: La transacción de compra se realiza mediante el intercambio de dinero físico en el momento de la compra.

## Referencias

<<Referencias a otros documentos e información de trazabilidad – documento/s de análisis de requisitos relacionados, sistemas existentes, restricciones que afectan a la arquitectura software, etc .>>

# Arquitectura del sistema.

## Arquitectura propuesta

**Arquitectura de Tres Capas:**

1. **Capa de Presentación (Interfaz de Usuario):**
   * **Descripción:**
     + La capa de presentación se encarga de la interfaz de usuario, proporcionando una experiencia amigable para los usuarios finales, incluyendo vendedores, administradores y clientes.
   * **Características:**
     + Diseño intuitivo y fácil de usar.
     + Gestión de la interacción con el usuario.
     + Presentación de información relevante.
   * **Componentes:**
     + Aplicación móvil para vendedores.
     + Interfaz web para administradores.
     + Interfaz de usuario para clientes.
2. **Capa de Lógica de Negocio (Control):**
   * **Descripción:**
     + La capa de lógica de negocio contiene la lógica central y las reglas del sistema, gestionando las operaciones y procesos relacionados con la distribución, ventas, administración de productos y clientes.
   * **Características:**
     + Implementación de reglas de negocio.
     + Procesamiento de operaciones y lógica de aplicación.
     + Coordinación de flujos de trabajo.
   * **Componentes:**
     + Clases y servicios relacionados con la gestión de productos.
     + Gestión de procesos de compra y venta.
     + Lógica de control de créditos y límites.
3. **Capa de Datos (Persistencia):**
   * **Descripción:**
     + La capa de datos se encarga de la persistencia de la información, gestionando el almacenamiento y recuperación de datos relacionados con productos, transacciones y configuraciones del sistema.
   * **Características:**
     + Almacenamiento y recuperación de datos.
     + Gestión de la base de datos.
     + Acceso eficiente a la información.
   * **Componentes:**
     + Base de datos relacional para el almacenamiento de productos, transacciones y usuarios.
     + Servicios de acceso a datos para la interacción con la base de datos.

**Ventajas del Estilo Arquitectónico:**

1. **Separación de Responsabilidades:**
   * La división en capas permite una clara separación de responsabilidades, facilitando el mantenimiento y la evolución del sistema.
2. **Escalabilidad y Reutilización:**
   * La modularidad de las capas facilita la escalabilidad del sistema, permitiendo la adición de nuevas funcionalidades o la mejora de las existentes sin afectar otras partes del sistema.
3. **Mantenimiento Simplificado:**
   * Cambios en la interfaz de usuario, reglas de negocio o persistencia pueden realizarse de manera independiente, simplificando el mantenimiento y reduciendo el impacto de las modificaciones.
4. **Adaptabilidad a Futuros Cambios:**
   * La arquitectura de tres capas proporciona flexibilidad para adaptarse a futuros cambios en los requisitos del sistema sin afectar la totalidad de la aplicación.

Diseña un diagrama de actividades que modele el sistema completo

## Visión global.

<<Presenta la estructura global de la arquitectura del sistema>>

1. **Subsistema de Distribución:**
   * *Funcionalidad:*
     + Coordinar la logística de distribución de frutas y verduras.
     + Asignar procesos de traslado y entrega de productos.
     + Gestionar estrategias para minimizar pérdidas por productos cercanos a la fecha de vencimiento.
2. **Subsistema de Ventas:**
   * *Funcionalidad:*
     + Desarrollar estrategias de venta por mayoreo y menudeo.
     + Coordinar procesos de compra y venta.
     + Asignar y gestionar procesos de venta, asegurando un flujo eficiente.
3. **Subsistema Administrativo:**
   * *Funcionalidad:*
     + Dar de alta productos en el sistema.
     + Asignar procesos de compra y venta.
     + Gestionar los gastos asociados a la distribución.
4. **Subsistema de Vendedores (Aplicación Móvil):**
   * *Funcionalidad:*
     + Gestionar el stock de productos disponible.
     + Realizar ventas a través de la aplicación móvil.
     + Proporcionar información sobre las cajas distribuidas, vendidas y perdidas.
5. **Subsistema de Clientes:**
   * *Funcionalidad:*
     + Manejar los pagos en efectivo.
     + Asignar créditos a clientes, con límites establecidos.
     + Controlar y gestionar límites de crédito para evitar pérdidas significativas.

**Descripción de la Arquitectura Software Elegida:**

La arquitectura del sistema se basa en un modelo en capas para proporcionar una estructura modular y bien organizada. Cada capa tiene funciones específicas y se comunica con capas adyacentes de manera clara. A continuación, se describen las capas y su funcionalidad:

1. **Capa de Presentación:**
   * *Funcionalidad:*
     + Manejo de la interfaz de usuario.
     + Recepción de entradas del usuario.
     + Presentación de información relevante.
2. **Capa de Lógica de Negocio:**
   * *Funcionalidad:*
     + Coordinación de procesos de negocio.
     + Implementación de la lógica central del sistema.
     + Gestión de la interacción entre subsistemas.
3. **Capa de Acceso a Datos:**
   * *Funcionalidad:*
     + Acceso y manipulación de la base de datos.
     + Almacenamiento y recuperación de información.
     + Gestión de transacciones de datos.

**Ventajas de la Arquitectura:**

* **Modularidad:** Cada subsistema opera como un módulo independiente, facilitando el desarrollo y mantenimiento.
* **Separación de Responsabilidades:** Las capas permiten una clara separación de responsabilidades, mejorando la comprensión y mantenimiento del sistema.
* **Flexibilidad:** La arquitectura en capas facilita la adaptación a cambios en requisitos sin afectar otras partes del sistema.

**Comunicación entre Subsistemas:**

* El Subsistema de Distribución se comunica con el Subsistema de Ventas para recibir información sobre la demanda y estrategias de venta.
* El Subsistema de Ventas se comunica con el Subsistema Administrativo para obtener detalles sobre los productos y con el Subsistema de Distribución para coordinar la logística de distribución.
* Los Subsistemas de Vendedores y Clientes interactúan con el Subsistema de Ventas para realizar transacciones y obtener información sobre productos y procesos.

Esta arquitectura proporciona un marco robusto para la distribuidora de frutas y verduras, asegurando una colaboración efectiva entre subsistemas y una estructura que permite la evolución del sistema.

## Diseño de la Arquitectura

<<En este apartado se establece la descomposición en subsistemas y las responsabilidades asignadas a cada uno. Corresponde a los productos obtenidos de las fases de análisis y diseño arquitectónico. En la sección de topología del sistema se describirán la asignación del software al hardware – diagrama de despliegue- y a los componentes – diagrama de componentes. La sección de “Gestión de la Persistencia” describe la infraestructura de almacenamiento requerida y los datos que serán almacenados por el sistema. La sección de “Aspectos Globales y de Seguridad” se describen las políticas de seguridad y acceso que serán establecidas en el sistema – mecanismos de autentificación, encriptación, certificados, etc.- y los aspectos de sincronización, concurrencia e inicialización de subsistemas. Por último, la sección de “Aspectos de rendimiento y tamaño” establecerá los acuerdos de nivel de servicio establecidos con el usuario en relación a tiempos de respuesta aceptables, productividad y capacidad de almacenamiento requerido.>>

## Diseño de los subsistemas

1. **Subsistema de Distribución:**
   * **Servicios:**
     + Coordinación de la logística de distribución.
     + Asignación de procesos de traslado y entrega.
     + Estrategias para minimizar pérdidas por productos cercanos a la fecha de vencimiento.
   * **Interfaces:**
     + Interfaz con el Subsistema de Ventas para recibir información sobre la demanda.
     + Interfaz con el Subsistema Administrativo para obtener detalles sobre los productos a distribuir.
2. **Subsistema de Ventas:**
   * **Servicios:**
     + Desarrollo de estrategias de venta por mayoreo y menudeo.
     + Coordinación de procesos de compra y venta.
     + Asignación y gestión de procesos de venta.
   * **Interfaces:**
     + Interfaz con el Subsistema de Distribución para enviar información sobre la demanda.
     + Interfaz con el Subsistema Administrativo para obtener detalles sobre los productos.
3. **Subsistema Administrativo:**
   * **Servicios:**
     + Alta de productos en el sistema.
     + Asignación de procesos de compra y venta.
     + Gestión de gastos asociados a la distribución.
   * **Interfaces:**
     + Interfaz con el Subsistema de Ventas para recibir información sobre la demanda y estrategias de venta.
     + Interfaz con el Subsistema de Distribución para obtener detalles sobre la logística de distribución.
4. **Subsistema de Vendedores (Aplicación Móvil):**
   * **Servicios:**
     + Gestión del stock de productos.
     + Realización de ventas a través de la aplicación móvil.
     + Proporcionar información sobre cajas distribuidas, vendidas y perdidas.
   * **Interfaces:**
     + Interfaz con el Subsistema de Ventas para realizar transacciones de venta.
     + Interfaz con el Subsistema Administrativo para obtener información sobre productos y procesos.
5. **Subsistema de Clientes:**
   * **Servicios:**
     + Manejo de pagos en efectivo.
     + Asignación de créditos a clientes con límites establecidos.
     + Control de límites de crédito para evitar pérdidas.
   * **Interfaces:**
     + Interfaz con el Subsistema de Ventas para realizar transacciones.
     + Interfaz con el Subsistema Administrativo para obtener información sobre productos y procesos.

**Establecimiento de Límites y Interfaces:**

* El Subsistema de Distribución establece límites en la asignación de procesos de traslado y entrega, basándose en la demanda proporcionada por el Subsistema de Ventas.
* El Subsistema de Ventas establece límites en la asignación y gestión de procesos de venta, coordinando con el Subsistema de Distribución y obteniendo estrategias del Subsistema Administrativo.
* El Subsistema Administrativo establece límites en la asignación de procesos y gestión de gastos, coordinando con el Subsistema de Ventas y obteniendo información logística del Subsistema de Distribución.
* El Subsistema de Vendedores se comunica con el Subsistema de Ventas y el Subsistema Administrativo para realizar transacciones y obtener información sobre productos y procesos.
* El Subsistema de Clientes se comunica con el Subsistema de Ventas y el Subsistema Administrativo para realizar transacciones y obtener información sobre productos y procesos.

Estas interfaces y límites bien definidos permiten una colaboración efectiva entre los subsistemas, asegurando un funcionamiento coherente y eficiente del sistema de gestión de frutas y verduras.

Principio del formulario

## Vista de casos de uso

<<En esta sección se enumeran los casos de uso incluidos en el modelo de casos de uso que representan las partes de funcionalidad central y que son significativos para el sistema final o que cubren una parte importante de la arquitectura del sistema final.

La documentación a incluir en esta sección corresponde a la obtenida como consecuencia de la actividad “Realización de casos de uso”:

## Vista lógica

**Paquete Principal: GestorFrutasVerduras**

* **Descripción:**
  + Contiene todos los subsistemas y componentes principales del sistema.
* **Subpaquetes:**
  + **Distribucion**
  + **Ventas**
  + **Administracion**
  + **Vendedores**
  + **Clientes**

**2. Paquete: Distribucion**

* **Descripción:**
  + Se encarga de la distribución logística de frutas y verduras.
* **Clases Relevantes:**
  + **GestorDistribucion**
    - **Responsabilidades:**
      * Coordinar procesos de traslado y entrega.
      * Implementar estrategias para minimizar pérdidas por productos cercanos a la fecha de vencimiento.
    - **Relaciones:**
      * Se comunica con **Ventas** para obtener información sobre la demanda.
      * Interactúa con **Administracion** para coordinar gastos asociados a la distribución.

**3. Paquete: Ventas**

* **Descripción:**
  + Gestionar estrategias y procesos de venta.
* **Clases Relevantes:**
  + **GestorVentas**
    - **Responsabilidades:**
      * Desarrollar estrategias de venta por mayoreo y menudeo.
      * Coordinar procesos de compra y venta.
    - **Relaciones:**
      * Interactúa con **Distribucion** para enviar y recibir información sobre la demanda.
      * Colabora con **Administracion** para obtener detalles sobre productos.

**4. Paquete: Administracion**

* **Descripción:**
  + Manejar la administración central del sistema.
* **Clases Relevantes:**
  + **GestorAdministrativo**
    - **Responsabilidades:**
      * Dar de alta productos en el sistema.
      * Asignar procesos de compra y venta.
      * Gestionar los gastos asociados a la distribución.
    - **Relaciones:**
      * Colabora con **Ventas** para proporcionar información sobre estrategias de venta.
      * Interactúa con **Distribucion** para coordinar gastos asociados a la distribución.

**5. Paquete: Vendedores**

* **Descripción:**
  + Se encarga de las funciones específicas de los vendedores.
* **Clases Relevantes:**
  + **AplicacionMovil**
    - **Responsabilidades:**
      * Gestionar el stock de productos disponible.
      * Realizar ventas a través de la aplicación móvil.
      * Proporcionar información sobre cajas distribuidas, vendidas y perdidas.
    - **Relaciones:**
      * Se comunica con **Ventas** para realizar transacciones de venta.
      * Colabora con **Administracion** para obtener información sobre productos y procesos.

**6. Paquete: Clientes**

* **Descripción:**
  + Gestionar las funciones relacionadas con los clientes.
* **Clases Relevantes:**
  + **GestorClientes**
    - **Responsabilidades:**
      * Manejar los pagos en efectivo.
      * Asignar créditos a clientes, con límites establecidos.
      * Controlar y gestionar límites de crédito para evitar pérdidas significativas.
    - **Relaciones:**
      * Se comunica con **Ventas** para realizar transacciones.
      * Colabora con **Administracion** para obtener información sobre productos y procesos.

<<Realiza un diagrama de con todas las clases y paquetes contenidas en dicho paquete.>>

# Diseño detallado

## Introducción.

La fase de diseño detallado de este proyecto de investigación juega un papel crucial en la materialización de la aplicación destinada a la gestión integral de un establecimiento de venta de verduras y frutas. En esta etapa, se han tomado decisiones de diseño específicas que trazan el camino para el desarrollo eficiente y efectivo del sistema.

En cuanto al tipo de desarrollo, se ha optado por el desarrollo interno, basado en consideraciones estratégicas y de alineación con los objetivos específicos del proyecto. Esta elección se ha guiado por la necesidad de personalización y adaptación a los requisitos particulares del local de venta, asegurando así una solución que se ajuste de manera óptima a las necesidades del negocio.

En el ámbito del rendimiento, se ha llevado a cabo un análisis detallado que ha guiado las elecciones entre espacio en memoria y tiempo de respuesta. La prioridad recae en la velocidad de respuesta, con el objetivo de ofrecer una experiencia de usuario ágil y sin interrupciones.

Para garantizar la coherencia y calidad del desarrollo, se han establecido estándares rigurosos que abarcan desde la asignación de nombres hasta los mecanismos de gestión de excepciones. Estos estándares proporcionan una guía clara para los diseñadores e implementadores, asegurando la consistencia en la estructura del código y facilitando el mantenimiento a lo largo del ciclo de vida de la aplicación.

Además, se han definido estándares de interfaces y codificación que servirán como pilares fundamentales en el proceso de desarrollo. Estos criterios guiarán a los diferentes equipos involucrados, fomentando la cohesión y la eficiencia en la implementación de módulos específicos.

En resumen, esta etapa de diseño detallado se erige como el cimiento sobre el cual se construirá una aplicación robusta, adaptada a las necesidades específicas del negocio y orientada a proporcionar una experiencia de usuario óptima. La cuidadosa consideración de estas elecciones de diseño garantiza un desarrollo coherente y eficiente que respalda los objetivos del proyecto de manera integral.

## Diseño de casos de uso reales

<<En esta sección se enumeran los casos de uso incluidos en el modelo de casos de uso que representan las partes de funcionalidad central y que son significativos para el sistema final o que cubren una parte importante de la arquitectura del sistema final.

La documentación a incluir en esta sección corresponde a la obtenida como consecuencia de la actividad “Realización de casos de uso”:

Flujos de eventos(Plantillas de caso des uso) - Diseño: descripción textual de cómo se realiza el caso de uso en términos de los objetos que colaboran. Resumen de los diagramas conectados con el caso de uso y explicación de sus relaciones.

Diagramas de interacción: Diagramas de secuencia ó Diagramas de colaboración, objetos participantes, Diagramas de clases.

Requisitos derivados: Descripción textual que recoge todos los requisitos, normalmente los no funcionales, de la realización del caso de uso no que han de tenerse en cuenta durante la implementación>>

**Tabla 1. Formatos de casos de uso (Tomar como referencia el formato propuesto en clase)**

|  |  |
| --- | --- |
| **Número y nombre del caso de uso** | < Escriba aquí el nombre del caso de uso [CU número del caso de uso]> |
| Descripción | <Breve descripción de lo que hace el Caso de uso, el objetico que logra> |
| Actores Principales | <Nombre del Rol>:<Descripción del Rol del  actor que esta interactuando con el sistema> |
| Actores Secundarios | <Aquí se puede incluir, por ejemplo, a los actores de soporte> |
| Precondiciones | <Lo que debe cumplirse para ejecutar el caso de uso> |
| Requerimientos. especiales | Reglas de Negocio: <son las especificaciones de los procesos que maneja el usuario> |
| Formatos (de captura) asociados al Caso: <se |

|  |  |
| --- | --- |
|  | debe especificar el número de la forma1 a la que se hace referencia > |
| Formatos de “Salida” asociados al Caso: <la relación de las formas de salida del sistema en este caso de uso> |
| Poscondiciones | <Lo que debe cumplirse una vez finalizado el caso de uso con éxito> |
| Requerimientos no funcionales | De Comunicación: <Detalle aquí los requerimientos no funcionales de comunicaciones, que hacen que el caso de prueba funcione con éxito o no aplica (N/A) si es el caso>  De Persistencia: <Detalle aquí los requerimientos no funcionales de bases de datos, que hacen que el caso de prueba funcione con éxito o no aplica (N/A) si es el caso>  De Seguridad: <Detalle aquí los requerimientos no funcionales de seguridad, que hacen que el caso de prueba funcione con éxito o no aplica (N/A) si es el caso>  De Escalabilidad: <Detalle aquí los requerimientos no funcionales de escalabilidad, que hacen que el caso de prueba funcione con éxito o no aplica (N/A) si es el caso>  De rendimiento: <Detalle aquí los requerimientos no funcionales de rendimiento, que hacen que el caso de prueba funcione con  éxito o no aplica (N/A) si es el caso> |

Flujo de eventos (Escenario normal)

|  |  |  |
| --- | --- | --- |
| **Paso** | **Actor** | **Descripción** |
| <Paso No.> | <Cuál es la condición que desencadena la extensión o variante> | <Descripción de lo que ocurre en la variación o extensión> |

Flujo Alternos

|  |  |  |
| --- | --- | --- |
| **Paso** | **Actor** | **Descripción** |
| <Paso No.> | <Cuál es la condición que desencadena la extensión o  variante> | <Descripción de lo que ocurre en la variación o extensión> |

Puntos Abiertos

|  |  |
| --- | --- |
| **Punto** | **Descripción** |
| <Id Punto> | <Descripción de las dudas o indefiniciones existentes en el caso de uso> |

1 Forma: se refiere a la especificación de los campos que hacen parte de ella. (referirse al documento formato de formas)

## Diseño detallado de las clases.

En la fase de diseño detallado, se han identificado y definido varias clases fundamentales utilizando el enfoque CRC (Clase-Responsabilidad-Colaborador). A continuación, se describen completamente algunas de las clases clave junto con sus atributos, métodos y documentación de dependencias.

1. \*\*Clase: `GestorInventario`\*\*

- \*\*Atributos:\*\*

- `productos: Lista[Producto]` - Almacena la lista de productos disponibles en el inventario.

- `proveedores: Lista[Proveedor]` - Contiene información sobre los proveedores asociados a los productos.

- \*\*Métodos:\*\*

- `agregarProducto(producto: Producto): void` - Añade un nuevo producto al inventario.

- `actualizarStock(producto: Producto, cantidad: int): void` - Actualiza el stock disponible de un producto.

- `generarReporteInventario(): Reporte` - Genera un informe detallado del estado actual del inventario.

- \*\*Dependencias:\*\*

- Colabora con la clase `Producto` para la gestión detallada de los productos en el inventario.

- Requiere la clase `Proveedor` para mantener información sobre los proveedores asociados.

2. \*\*Clase: `InterfazUsuario`\*\*

- \*\*Atributos:\*\*

- `sesionActiva: bool` - Indica si un usuario ha iniciado sesión en la interfaz.

- \*\*Métodos:\*\*

- `iniciarSesion(usuario: Usuario, contraseña: str): bool` - Inicia sesión para acceder a las funciones de la aplicación.

- `realizarVenta(producto: Producto, cantidad: int): void` - Registra una venta en el sistema.

- \*\*Dependencias:\*\*

- Interactúa con la clase `Usuario` para gestionar el inicio de sesión y autorización.

- Utiliza la clase `Producto` para realizar operaciones de venta.

3. \*\*Clase: `Reporte`\*\*

- \*\*Atributos:\*\*

- `contenido: str` - Almacena el contenido del informe.

- \*\*Métodos:\*\*

- `exportarPDF(): void` - Genera un archivo PDF del informe actual.

- \*\*Dependencias:\*\*

- No tiene dependencias directas con otras clases, pero utiliza información proporcionada por las clases `GestorInventario` y otras para construir el contenido del informe.

4. \*\*Clase: `Producto`\*\*

- \*\*Atributos:\*\*

- `nombre: str` - Nombre del producto.

- `precioUnitario: float` - Precio por unidad del producto.

- \*\*Métodos:\*\*

- `calcularTotal(cantidad: int): float` - Calcula el precio total de un producto según la cantidad.

- \*\*Dependencias:\*\*

- No tiene dependencias directas con otras clases, pero se utiliza en conjunto con `GestorInventario` y `InterfazUsuario`.

Estas descripciones son esbozos iniciales y deberán ser detalladas con mayor profundidad durante el proceso de implementación, asegurando una coherencia y modularidad efectiva en el desarrollo del sistema. La documentación de dependencias proporciona una guía clara sobre cómo estas clases interactúan entre sí y con otras partes del sistema.

## Diseño de la arquitectura de módulos del sistema

En el diseño de la arquitectura de módulos del sistema, se han identificado varios módulos y submódulos que colaboran para ofrecer la funcionalidad integral de la aplicación. A continuación, se describen estos componentes y se proporciona un esbozo de un diagrama para visualizar la relación entre ellos.

1. \*\*Módulo: `Gestión de Inventario`\*\*

- \*\*Submódulos:\*\*

- `GestorInventario`: Encargado de la gestión y control del inventario de productos.

- `ProveedorManager`: Responsable de la gestión de la información relacionada con los proveedores.

- \*\*Funcionalidades:\*\*

- Agregar, actualizar y eliminar productos en el inventario.

- Gestionar la información de los proveedores asociados a los productos.

2. \*\*Módulo: `Ventas`\*\*

- \*\*Submódulos:\*\*

- `InterfazUsuario`: Proporciona una interfaz para que los usuarios realicen ventas y accedan a otras funciones del sistema.

- `ReporteGenerator`: Genera informes detallados sobre las transacciones de ventas.

- \*\*Funcionalidades:\*\*

- Iniciar y gestionar sesiones de usuario.

- Registrar ventas y mantener un historial de transacciones.

- Generar informes de ventas.

3. \*\*Módulo: `Seguridad y Autenticación`\*\*

- \*\*Submódulos:\*\*

- `AutenticaciónManager`: Gestiona el proceso de inicio de sesión y autenticación de usuarios.

- `AutorizaciónManager`: Controla los niveles de acceso y autorización dentro del sistema.

- \*\*Funcionalidades:\*\*

- Verificar credenciales de usuario y gestionar sesiones de inicio de sesión.

- Definir y aplicar niveles de acceso para garantizar la seguridad.

4. \*\*Módulo: `Reportes`\*\*

- \*\*Submódulos:\*\*

- `GeneradorReportes`: Crea informes específicos del estado del inventario y de las transacciones de venta.

- \*\*Funcionalidades:\*\*

- Crear informes detallados y exportarlos en diferentes formatos (PDF, CSV, etc.).

- Personalizar la información que se incluye en los informes.

5. \*\*Módulo: `Base de Datos`\*\*

- \*\*Submódulos:\*\*

- `ConectorDB`: Establece y gestiona la conexión con la base de datos.

- `ScriptsSQL`: Contiene los scripts para la creación y actualización de la base de datos.

- \*\*Funcionalidades:\*\*

- Gestionar la persistencia de datos del inventario, transacciones de venta y otros registros.

**Diagrama de Arquitectura de Módulos:**

|  |
| --- |
| Aplicación |
| Gestión de Inventario  - GestorInventario     - ProveedorManager |
| Ventas                 - InterfazUsuario            - ReporteGenerator |
| Seguridad y Autenticación       - AutenticaciónManager        - AutorizaciónManager |
| Reportes  - GeneradorReportes |
| Base de Datos  - ConectorDB  - ScriptsSQL |

Este diagrama proporciona una representación visual de la estructura modular del sistema, mostrando cómo los distintos módulos y submódulos interactúan entre sí para ofrecer las funcionalidades requeridas. La documentación detallada de cada módulo y submódulo contribuye a una comprensión clara de su propósito y responsabilidades dentro de la arquitectura del sistema.

## Diseño físico de datos

1. **Modelo de datos y diccionario de datos**

<<En esta sección del documento se agrega el diagrama clases con el diagrama de entidades de la base de datos. Adicionalmente se deberá complementar el diagrama entidad relación con la respectiva descripción detallada de cada una de las entidades y atributos como se muestra a continuación..>>

## Modelo Entidad – Relación. (Diagrama de Clases)

<< En esta sección se deberá incluir el diagrama entidad relación desarrollado.>>

## Diccionario de datos.

**Tabla 2 Diccionario de datos**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Número y nombre de**  **la entidad** | | TBL - <#> | < Escriba aquí el nombre de la entidad (Tabla) > | | | | | |
| Descripción | | < Escriba aquí la descripción de la nombre de la entidad (Tabla) > | | | | | | |
| Nro. | Nombre de la Columna | Descripción | | Tipo de datos | Permitir Valores NULL | Llave | Valor Predeter minado | Auto numér ico |
| <#> | < Escriba aquí | < Escriba aquí la | | < | < | < | < | < |
|  | el nombre de | descripción de la | | Escriba | Marque | Indiqu | Indique | Marqu |
|  | la columna > | columna > | | aquí el | SI o NO | e PK – | aquí | e SI o |
|  |  |  | | tipo de | permite | Para | cual es | NO es |
|  |  |  | | datos de | valores | llaves | el valor | un |
|  |  |  | | la | NULL > | primari | predeter | valor |
|  |  |  | | columna |  | as o | minado | auto |
|  |  |  | | > |  | FK – | si aplica | numér |
|  |  |  | |  |  | para | > | ico > |
|  |  |  | |  |  | llaves |  |  |
|  |  |  | |  |  | secun |  |  |
|  |  |  | |  |  | darias |  |  |
|  |  |  | |  |  | > |  |  |

## Prototipo de pantallas

<<En esta sección se incluirá la descripción de navegación a nivel de interfaz de usuario (Diagrama de navegación entre pantallas), se deberá mostrar el prototipo de pantallas del mismo. Se podrá usar algún software de prototipos previamente pactado con la SDIS. Se incluirá la propuesta de diseño de interfaz de usuario con los colores y logos definidos conjuntamente con la SDIS y acorde con la imagen corporativa y las disposiciones de la Comisión Distrital de Sistemas y la normatividad vigente.>>

Realiza un mapa de navegación y posteriormente agrega las imágenes del prototipo elaborado en blasamiq, AdobeXD.