

# Universidad Autónoma De Tamaulipas

## Facultad de Ingeniería Tampico

### Ingeniería En Sistemas Computacionales

#### Requisitos De Software

Materia: Programación Avanzada

Alumna: Castillo Aviles Anahi

Matricula: a2233336121

Profesor: Ing. Alvarez Navarro Eduardo

Fecha:

Semestre: 3°

Grupo: "I"

# Índice

1.-Portada .....	1
2.- Índice.....	2
3.- Introducción.....	4
3.1.- Propósito Del Documento .....	4
3.1.1.- Objetivo del proyecto.....	4
3.2.- Alcance del producto .....	5
3.3.- Definiciones, siglas y abreviaturas .....	6
4.- Descripción General.....	8
4.1.- Descripción del Proceso del Punto de Venta .....	8
4.2.- Evidencia de negocios visitados .....	12
4.3.- Usuarios Finales y Características .....	13
4.4.- Funciones del sistema .....	14
4.5.- Entorno de Operación .....	17
5.- Requerimientos Funcionales .....	19
5.1.- Flujo básico (Requisitos funcionales).....	19
5.2.- Modelo de caso de uso.....	20
5.3.- Precondiciones .....	23
5.4.- Postcondiciones .....	24
6.- Diseño del sistema .....	26
6.1.- Arquitectura General.....	26
6.2.- Uso de interfaces y clases abstractas .....	26
6.3.- Patrones de diseño utilizados .....	26
7.- Diccionario de Datos de las clases.....	27
1.- Clase: Carrito.java .....	27
2.- Clase: Producto.java .....	28
3.- Clase: Venta.java .....	30
4.- Clase: Cliente.java .....	31
5.- Clase: Inventario.java.....	31
6.- Clase: Usuario.java .....	32
7.- Clase: Proveedor.java.....	33
8.- Clase: Devolución.java.....	34

9.- Reportes.java.....	36
8.- Modelo UML de Clases .....	37
8.1.- UML Caso de uso .....	41
9.- Modelo de UML de Actividades .....	42
Inventario .....	42
Clientes.....	46
Proveedores .....	47
Reportes.....	48
Usuarios .....	49
Venta.....	50
10.- Relaciones detectadas .....	51
10.1.- Relaciones de Base de Datos.....	53
11.- Interfaces de usuarios .....	56
12.- Flujo de las interfaces de Usuario por cada Requerimiento funcional.....	73
13.- Requerimientos No funcionales .....	88
13.1.- Seguridad .....	88
13.2.- Usabilidad .....	88
13.3.- Hardware .....	89

## 3.- Introducción

El presente documento se describe lo más detalladamente posible los requerimientos del sistema informático denominado Sistema de Punto de Venta, desarrollado en lenguaje Java con interfaz gráfica construida en Swing y almacenamiento de datos mediante Microsoft Access.

Este sistema fue desarrollado con el propósito de atender las necesidades básicas de automatización en negocios de tipo comercial, específicamente en tiendas, minisúper, farmacias o comercios minoristas, que requieren control sobre ventas, productos, inventario y clientes, así como reportes que respalden su operación diaria.

El documento sirve como base de referencia para profesores, desarrolladores y usuarios evaluadores, permitiendo comprender de forma estructurada que hace el sistema, como se comporta y que limitaciones o condiciones debe cumplir. Además, funciona como guía para la implementación técnica y futura evolución del software, asegurando la trazabilidad de cada requerimiento.

### 3.1.- Propósito Del Documento

El propósito principal de este documento es recopilar, organizar y explicar los requisitos que debe cumplir el sistema de punto de venta, tanto desde la perspectiva funcional (lo que debe hacer) como desde la no funcional (como debe comportarse). También establece las restricciones técnicas del sistema, su entorno de operación, y el perfil esperado de los usuarios finales.

Este documento busca:

- Alinear las funcionalidades del sistema con las necesidades del usuario final.
- Establecer expectativas claras sobre lo que el sistema puede y no puede hacer.
- Servir como soporte técnico para futuras modificaciones, mantenimiento o auditoria del proyecto.
- Establecer una base formal para la evaluación del cumplimiento del sistema.

#### 3.1.1.- Objetivo del proyecto

El sistema punto de venta está diseñado para ser una herramienta accesible para personas con poco conocimiento técnico, permitiendo a los trabajadores y administradores, realizar las operaciones necesarias de manera eficiente. A través de este proyecto, se demostrará la aplicación práctica de los patrones de diseño para crear un software robusto, mantenible y fácil de usar.

### 3.2.- Alcance del producto

El producto desarrollo es una aplicación escritorio (entrada teclado y mouse u touch) enfocada en la gestión integral de las operaciones de venta destinado a satisfacer las necesidades de tiendas pequeñas y misceláneas que buscan una forma más eficiente de gestionar sus ventas, inventarios, reportes y clientes, sin tener que recurrir a sistemas complicados o costosos. Su interfaz gráfica amigable y con distribución de colores permite al usuario interactuar con las siguientes funcionalidades:

- Módulo de productos: Registro, consulta, modificación y eliminación de productos. Asignación de categorías, precios, cantidades y proveedores.
- Módulo de clientes: Alta y gestión de clientes frecuentes.
- Módulo de proveedores: Registro y administración de proveedores.
- Módulo de inventario: Visualización del inventario, control de stock y actualización automática al realizar ventas.
- Módulo de ventas: permite seleccionar productos, generar un carrito, aplicar descuentos y generar un ticket de venta con detalles.
- Módulo de devoluciones: Registro de productos devueltos tanto por clientes o proveedores, con control de inventario.
- Módulo de reportes: Consultas y reportes de ventas, clientes, proveedores e inventario. Todos los reportes pueden exportarse a PDF, XML, CSV, EXCEL.
- Login de usuario: Sistema de autenticación con validación de usuario y contraseña. Puede registrar el acceso a las funciones del sistema.
- Módulo de reimpresión de tickets.

El sistema se ejecuta de manera local y no requiere conexión a internet, siendo ideal para entornos donde se necesite un sistema funcional, confiable y de bajo costo.

El alcance del proyecto incluye el desarrollo, pruebas e implementación del sistema de POS para su uso en un entorno único y controlado. El sistema estará preparado para escalar a medida que el negocio crezca, pero su implementación inicial se concentrará en satisfacer las necesidades de pequeñas tiendas y comercios locales.

### 3.3.- Definiciones, siglas y abreviaturas

Termino/Sigla	Definición
<b>POS (Point of sale)</b>	Sistema de punto de venta. Aplicación que permite registrar operaciones de venta, gestionar inventario y emitir comprobantes o tickets.
<b>DAO</b>	Data Access Object. Patrón de diseño que permite abstraer y encapsular el acceso a la base de datos, mejorando la organización del código y facilitando el mantenimiento.
<b>CRUD</b>	Créate, Read, Update, Delete. Conjunto de operaciones básicas que se pueden realizar sobre registros en una base de datos.
<b>MVC</b>	Model-View-Controller. Patrón de arquitectura de software que separa la lógica del negocio (modelo), la interfaz de usuario(vista) y el control de flujo (controlador). El sistema este organizado parcialmente bajo este esquema.
<b>Access</b>	Base de datos relacional de Microsoft. Se utiliza como sistema de almacenamiento local para guardar productos, usuarios, ventas clientes y proveedores.
<b>Swing</b>	Librería de java utilizada para construir interfaces graficas de usuario (GUI). Permite la creación de formularios interactivos en el sistema.
<b>Ticket</b>	Documento generado automáticamente por el sistema al realizar una venta. Contiene información como fecha, hora, productos vendidos, cantidad, subtotal, impuestos y total.
<b>Reporte</b>	Documento que presenta la información consolidada y organizada sobre ventas, inventarios, clientes y proveedores. En este sistema, se generan reportes dinámicos que se pueden exportar a PDF, EXCEL, CSV, JSON, HTML.
<b>Inventario</b>	Registro actualizado de todos los productos disponibles en existencias. Se ajusta automáticamente cada vez que se realiza una venta o devolución.
<b>Carrito</b>	Representa los productos seleccionados temporalmente por el usuario durante el proceso de venta. Se vacía tras finalizar la transacción.
<b>Controlador</b>	Parte del sistema encargada de recibir acciones del usuario, procesar la lógica y comunicarse con los modelos para actualizar datos.
<b>Modelo</b>	Componente que representa las estructuras de datos y las reglas de negocio, como productos, clientes, ventas, etc.
<b>Vista</b>	Elementos de la interfaz gráfica del sistema. Formularios o pantallas que permiten al usuario interactuar con el sistema.
<b>Interfaz Gráfica (GUI)</b>	Acrónimo de Graphical User Interface. Es la forma visual mediante la cual el usuario interactúa con el sistema a través de botones, formularios y menús.

<b>Usuario</b>	Persona que interactúa con el sistema. Existen distintos tipos de usuarios con diferentes niveles de acceso.
<b>Panel</b>	Componente visual de swing que representa una sección de la interfaz. Se utiliza, por ejemplo, para mostrar los distintos reportes.
<b>JFrame / JPanel</b>	Clases de java Swing usadas para representar ventanas (JFrame) y secciones internas (JPanel) de la interfaz del sistema.
<b>Exportación de datos</b>	Función que permite guardar la información generada por el sistema en archivos externos (PDF), facilitando su impresión o respaldo.
<b>Validación</b>	Proceso que verifica que los datos ingresados por el usuario sean correctos, completos y seguros antes de almacenar los o procesarlos.
<b>Reimpresión</b>	Funcionalidad que permite volver a imprimir un ticket de venta anterior, recuperado desde la base de datos.
<b>Devolución</b>	Registro de productos que fueron regresados por el cliente. Esta operación afecta el inventario y puede generar reportes.
<b>Base de datos (BD)</b>	Estructura organizada para almacenar, gestionar y recuperar datos. En este caso, se usa Microsoft Access como motor de base de datos.
<b>PDFExporter</b>	Clase del sistema que permite la generación de documentos PDF a partir de datos internos (como reportes).
<b>Proveedor</b>	Entidad que abastece productos. En el sistema, los proveedores se registran y se vinculan a productos.
<b>Cliente</b>	Persona o entidad que compra productos. Los clientes registrados pueden recibir beneficios o aparecer en reportes personalizados.
<b>Interfaz de búsqueda</b>	Espacio en la GUI donde se permite filtrar o localizar rápidamente registros (productos, clientes, ventas) mediante palabras clave.
<b>Evento de interfaz</b>	Acciones del usuario (clics, escritura, selección) que activan funciones internas del sistema, mediante listeners o controladores.
<b>Botones de acción</b>	Componentes visuales que permiten ejecutar funciones específicas del sistema como “Agregar producto”, “Guardar”, “Cancelar”, etc.
<b>ConexionAccess.java</b>	Clase responsable de establecer la conexión con la base de datos Access (.mdb). utiliza JDBC con un driver ODBC-JDBC compatible.
<b>PreparedStatement</b>	Clase de java que permite ejecutar consultas SQL precompiladas con parámetros, mejorando la seguridad y el rendimiento.
<b>ResultSet</b>	Objeto de java que contiene los resultados de una consulta SQL, generalmente utilizados para probar tablas dentro de la interfaz.
<b>ODBC</b>	Open Database Connectivity. Interfaz estándar para conectar aplicaciones con sistemas de gestión de bases de datos. Usado implícitamente al conectarse con Access a través de JDBC.
<b>Entidad</b>	Representación de un objeto del mundo real dentro del sistema, como Producto, Cliente o Proveedor.
<b>Método estático</b>	Función que pertenece a la clase y no a una instancia. En el proyecto, se usa en algunos controladores y funciones auxiliares.

## 4.- Descripción General

El sistema de **Punto de Venta (POS)** desarrollado es una aplicación de escritorio construida en **Java con interfaz gráfica Swing**, orientada a facilitar la gestión de ventas, inventarios y administración básica en una tienda minorista. El programa está diseñado para funcionar de forma local y utilizar como base de datos un archivo de **Microsoft Access**, sin requerir conexión a internet.

La estructura del sistema sigue el patrón de diseño **Modelo-Vista-Controlador (MVC)**, lo cual permite mantener una separación clara entre la lógica de negocio, la interfaz de usuario y la gestión de datos. Esto facilita futuras modificaciones o mantenimientos.

Su propósito principal es automatizar procesos rutinarios del comercio, como el registro de productos, clientes y proveedores, el control de inventario, la realización de ventas y la emisión de reportes o tickets.

### 4.1.- Descripción del Proceso del Punto de Venta

El punto de venta inicia con un login sencillo y practico, en este se puede iniciar sesión tanto como el administrador y el trabajador, se iba a descartar la acción de prioridades en usuarios, porque el trabajador es el mismo administrador, pero a veces recibe ayuda así que teníamos que restringir al trabajador solo en la reimpresión de reportes y tickets.

En el mismo login se encuentra un botón escondido en el que se puede hacer un registro temporal de usuario, el cliente nos comentó que su capacidad para recordar contraseñas es muy mala así que con ese usuario temporal podía dirigirse al menú principal y de ahí a Gestión de usuario y ver su contraseña para iniciar sesión ya como administrador.

Después del inicio de sesión se abre la ventana de Menú principal, en ella se encuentra botones que redirigirá a otras ventanas específicas, en Menú principal se encuentra Ventas, Clientes, Proveedores, Reportes, Gestión de usuario, Inventario, también se encuentra el botón de cerrar sesión.

Al ingresar en Ventas se observará la interfaz donde se podrá dar clic en los productos que el cliente desea comprar sin la necesidad de código de barras ni escáner, solo viendo el producto, si este no lo encuentra en la parte izquierda, se encuentran los botones de filtrado por categoría, también la interfaz cuenta con una sección donde se podrá ingresar el nombre del producto y al darle al botón de buscar este aparecerá, el producto se puede seleccionar, procediendo a agregarse al carrito, si esto fue planeado se puede seguir con la venta, pero si no fue tal cosa puede seleccionarse en el carrito y dar clic en el botón de Eliminar, este se borrará y podrá seleccionar el producto que desee.

Si quiere cambiar la cantidad del producto que desee agregar, se encuentra una sección de Cantidad en la parte inferior izquierda, con los botones puede aumentar la cantidad o disminuir y después seleccione el producto que desee agregar con esa cantidad y en el carrito se mostrara, si el producto ya fue agregado y se olvidó de modificar la cantidad este se puede volver a seleccionar o eliminar dependiendo de lo que el usuario requiera.



En la parte central se encuentra una sección donde se refleja el total de la compra y debajo de él se puede seleccionar el método de pago que el cliente presente, puede ser efectivo o tarjeta. En efectivo antes de terminar la compra pide que ingrese un monto, este puede ingresarse por teclado o por los botones que están en el interfaz, estos botones son un teclado numérico integrado por si el usuario no cuenta con teclado o este se le descompuso, después de ingresar monto puede dirigirse al botón de pagar y finalizar la compra.

Si el método de pago seleccionado es tarjeta, esta no pide ningún requerimiento y simplemente se dirige a el botón de Pagar y finaliza la compra. Debajo de la sección de método de pago se encuentra la sección de clientes en esa sección se puede ingresar el numero de un cliente que desee registrarse para generar puntos en la tienda o un cliente ya registrado estos puntos se aumentaran por cada compra siendo que 5 puntos es 1 peso, al tener suficientes puntos se puede hacer descuento en los productos, esta sección puede ser opcional ya que no todos los clientes se registran, así que puede seguir la venta normalmente aunque no ingrese nada a esta sección.

Debajo de esta sección se encuentra un teclado numérico de botones y un botón de cancelar, el teclado numérico como lo mencione antes puede utilizarse para ingresar monto o el numero teléfono del cliente, es una alternativa por si nuestro cliente tiene problemas en teclado y el botón de cancelar como su nombre lo indica tiene la acción de cancelar la venta reiniciando el total. Al terminar la venta y dar clic al botón de pagar se generará automáticamente un ticket de venta que se abrirá, donde aparecerá toda la información de la venta, este se puede imprimir, compartir o lo que desea el usuario.

En la parte superior de la interfaz, para ser más precisos, aun lado del logo se encuentra el nombre del usuario que inicio sesión, si el usuario se equivocó al iniciar sesión con otra cuenta, esta se puede cambiar dando clic en esa parte, a lo me refiero es dar clic en el nombre del usuario y se abrirá una ventana donde se cambiará de usuario. Al aceptar se redirigirá a el login y podrá iniciar sesión con el usuario deseado.

Debajo de esa sección se encuentra un menú horizontal, donde se podrá acceder a las otras ventanas y funciones que ofrece el POS, y así no hay necesidad de regresar al menú principal, claro se puede volver a acceder a él en el botón de salir que esta en este mismo menú.

Al dar clic en Reportes en el menú horizontal se accederá a la ventana de reportes.java, donde se podrá observar el mismo diseño de ventas solo quitando lo relacionado a él, en esta ventana se encuentran 5 botones Ventas, Clientes, Inventario, Proveedores y Reimpresión.

Como sus nombres indican en estos botones se pueden generar los reportes correspondientes a su sección, en Ventas al ingresar se podrá observar una estructura dividida en 4 partes, en la primera tenemos una sección de Filtrar reporte donde se podrá filtrar el reporte de venta por periodos estos siendo Todos, Hoy, Semana, Mes y Rango personalizado (donde se podrá ingresar las fechas deseadas del reporte).

En la segunda sección se encuentra Exportar reporte, en esta se puede escoger el formato del reporte que se desee generar siendo estas PDF, EXCEL, HTML, CSV, al dar clic en el botón de

Exportar se generará el reporte automáticamente y se mostrara. En la tercera sección se encuentra un grupo de pestañas Resumen, Detalle y Gráficos, mostraran visualmente la información obtenida del filtrado del reporte, siendo también la información que se exportara.

Y en la cuarta y última sección se encuentran los cuadros de información, en estos se muestra el total de ventas, ventas hoy, productos vendidos y ticket promedio que se obtiene desde la base de datos en Microsoft Access, actualizándose por los filtrados, también se encuentra un botón para regresar al menú de reportes.

Al dar clic en Inventario en reportes, se abrirá una ventana que tiene una estructura parecida a la ventana de reporte ventas, contando con las 4 secciones, filtrar inventario siendo estos por bajo stock, sobre stock, categoría y proveedor, también se encuentra la sección de exportar reporte que hace la misma sección que en reporte ventas, en la tercera sección solamente se encuentran 2 gráficos, que se adaptaran dependiendo del el filtrado que se requiera, y en la última sección se encuentra la información del total productos, bajo stock y sin stock, también el botón de regresar al menú de reportes.

Reportes clientes cuanta con la misma estructura que en inventario, lo diferente es la información de filtrado siendo esto por todos, activos, inactivos, con puntos y sin puntos, los gráficos se actualizan dependiendo del filtrado y la información mostrada arriba del botón de regresar al menú reportes son el total clientes, activos e inactivos.

Reportes proveedores cuenta con la misma estructura que el anterior, siendo su filtrado con vista reciente, sin vista reciente y por producto, su información presentada es el total de proveedores, visitados este mes y última visita, antes del botón de regresar al menú.

En reimpresión como su nombre lo indica, al dar clic en el botón se abre un dialogo, donde se puede buscar el tipo de documento que queremos reimprimir, siendo estos ticket de venta, reporte de venta, reporte de inventario, reporte de clientes y reporte de proveedores, se puede ingresar en ese dialogo el ID o referencia de ese documento o podemos dar clic en el botón de buscar y se abrirá otra ventana, donde se puede ingresar el ID, las fechas desde, hasta y el método de pago, esto ayudara a buscar mejor el documento que queremos reimprimir, debajo de estas secciones se puede apreciar una tabla donde se mostrara el documento que se encontró después de ingresar los datos necesario y dar clic en el botón de buscar, si se encuentra el documento se puede seleccionar directamente y posteriormente exportarse o seleccionarlo y se regresara a la ventana anterior y podremos reimprimirlo, si no se encontró este documento podemos cancelar y regresar, esto sucede cuando buscamos el ticket de venta, en los demás tipos de documentos nos pide en un dialogo el id del documento o el nombre del documento y nos dejara reimprimirlo.

En la sección de inventario en el menú horizontal al inicio y al ingresar nos mostrara los productos que tenemos en inventario, tanto como sus botones de filtrado en la parte izquierda de la ventana, se puede agregar categorías al filtrado y al seleccionarlase puede observar a los productos de esas categorías.

Se puede buscar los productos en la sección de abajo, ingresando su nombre y dando en el botón circular que aparece a un lado, también se puede agregar productos al dar click en el botón de Agregar, apareciendo una ventana donde se podrá ingresar todos los datos requeridos del producto, estructurado por secciones tenemos la información básica que abarca la imagen del producto, su nombre, descripción, categoría, proveedor, estado (puede ser activo, discontinuado, dañado) y unidad de medida que le corresponda, cuenta con su otra sección de stock y precios, con cantidad disponible, stock mínimo, stock máximo, precio compra, precio venta, IVA y descuento, se puede guardar o cancelar.

En el botón de búsqueda avanzada, al dar click su ventana esta dividida en 4 pestañas diferentes, tenemos búsqueda en donde se puede ingresar nombre, categoría, proveedor, estado, rango de precio, rango de stock, fecha de ingreso, si necesita reposición, si tiene descuento, o si aplica IVA, puedes buscar con los datos solicitados, limpiar lo ingresado o cancelar la búsqueda.

En la pestaña de movimientos está dividida por otras 3 pestañas, entradas donde se puede ingresar el ID del producto, su cantidad a ingresar y la fecha, registrando la entrada del producto mediante el botón, salidas donde se ingresa el id del producto, su cantidad, el motivo y las observaciones registrando la salida del producto por el botón y ajustes en este podemos ingresar el id del producto, el tipo de ajuste, la cantidad, su fecha y la descripción registrando el ajuste al producto,

En la pestaña de historial se puede ingresar el teléfono del cliente al queremos verle su historial, observándolo en la tabla. En la pestaña de devoluciones está dividida por otras 3 pestañas, devolución cliente donde se ingresa el id del producto, cantidad, motivo, id venta original y las observaciones registrando la devolución del cliente con el botón.

Devolución a proveedor se le ingresan los mismos datos así registrando su devolución, y en la última pestaña se muestra en una tabla el historial de las devoluciones de productos, esta tabla se puede actualizar o generar un reporte.

A los productos se les puede hacer clic en ellos y se abrirá una venta donde se mostrará toda su información, esta información se puede editar, eliminar, también se puede contactar su proveedor por si su producto está bajo en stock.

En el botón de proveedor nos dirigirá a la misma ventana de reporte inventarios ya mencionada, donde se podrá exportar el reporte en el formato que deseemos.

En la sección cliente en el menú horizontal se observará una tabla donde se mostrará los clientes registrados, tanto como su nombre, id, teléfono, puntos y la fecha de su última compra. En esta ventana se puede agregar el cliente, se puede seleccionar uno en la tabla y editarlo o eliminarlo.

Al agregar un cliente nos pide la información del cliente que corresponde a su teléfono, nombre, la fecha de registro o compra, se puede guardar o cancelar la acción. Al seleccionar un cliente para editarlo, solo nos deja editar el teléfono, nombre y la fecha, se puede guardar o cancelar, al eliminar el cliente seleccionado este también se elimina en la base de datos.

En la sección de proveedores en el menú horizontal, tiene la misma estructura de clientes, donde se puede agregar, eliminar o editar. Al agregar un proveedor los datos que nos pide son el nombre, teléfono, dirección, producto, y su fecha de última visita (esta se ira editando cada que llegue el proveedor), se puede registrar o cancelar, sucede lo mismo con editar, e eliminar después de seleccionar.

En la sección de usuarios en el menú principal tiene la misma estructura que proveedores y clientes. Se puede agregar, seleccionar y editar o eliminar. Al agregar la información que piden es el nombre del usuario, contraseña y su rol (Admin, Trab), se puede guardar y cancelar. Al seleccionar y editar solamente lo que puede cambiarse es el rol del usuario.

## 4.2.- Evidencia de negocios visitados

En este proyecto se visitaron tres negocios (tiendas), dos de ellas son tiendas de abarrotes, la diferencia es que una tenía solamente lo esencial de supervivencia abarrotes, bebidas, botanas, cuidado personal, su mercancía no pasaba de 50 productos, el dueño del lugar un adulto de 60 años o más, en la entrevista comento que tiene esta tienda por 10 años, siempre vendió solamente lo esencial para los estudiantes de la zona.

No cuenta con punto de venta, para llevar el control de su tienda tiene cuadernos donde registra las ventas del día, y la entrada de producto de los proveedores, le pregunte si gustaría ingresar un punto de venta en su tienda para llevar un mejor control de la tienda, pero el me comento que por su edad no le entendía mucho a la tecnología.

Tiene dos trabajadores que le ayudan a descargar el producto cuando lo traen los proveedores, y le ayudan a cuidar la tienda cuando él no puede, no cuenta con reportes solo anotaciones como comenté, cuando se le acaba el producto lo anota y pide al proveedor, si llega a ver una devolución lo cumplen sin problema.

La segunda tienda también es una de abarrotes, pero tiene más mercancía que la primera, cuentan con bebidas tanto alcohólico o refrescos, venden cigarrillo y muchas más de abarrotes, también verdura y fruta, ellos llevan un control en Excel, crean libros en Excel y anotan ahí las ventas del día, no anotan producto solo montos, así saben cuánto ganan y cuanto debería estar en la caja.

Cuentan con tres trabajadores familiares, su dueño un señor de 40 años o más me comento que la tienda la tienen hace cuatro años, y que han tenido dificultades a veces con el Excel, porque comprar la membresía de Microsoft para que este funcione, pero es un gasto más que dan a la tienda, los proveedores tienen un día recurrido que llegan, así que se hace una lista de productos que faltan y que llegarían a faltar para esa fecha y se hace el pedido, los reportes son los libros pero no los imprimen ni llevan un control deficiente, ya que solo lo tienen para saber lo vendido del día.

La tercera tienda visitada fue una taquería, esta taquería cuenta con el turno de la noche, pero se abre desde las 6 de la tarde, se hace un conteo de los productos que hay que comprar para las ventas, esta tienda si cuenta con un punto de venta, los trabajadores lo

tienen en el teléfono, tanto meseros como los que trabajan en cocina, lo de trabajo en cocina registran en el punto de venta lo gastado el día anterior de ingredientes

Este punto de venta esta sencillo cuenta con la venta de productos, su inventario ya antes mencionado, donde llevan el control de los ingredientes, este inventario lo controlan más los de la cocina ya que ellos se encargan de sabes cuantos ingredientes gastaron, la venta de productos se encargan los meseros y el recepcionista.

Este punto de venta se divide en esos usuarios, cocina, meseros, admin, el administrador lleva un control de todo lo que hacen los usuarios, cada que termina la jornada laboral, los usuarios mandan el reporte del día de sus ventas, e inventario, cuenta con generación de tickets, y puntos para clientes, los puntos se registran con tu número de teléfono y al obtener cierta cantidad de puntos se puede descontar de la comida que pidas.

Las ventas se dividen en comer aquí o para llevar, así el precio varía, hay ofertas del día que se marcan en el punto de venta, y el mesero o recepcionista se encarga de mencionarlas al cliente, no se agrega producto nuevo ya que siempre se vende lo mismo y si se requiere de agregar algo se debe de hablar al que lleva el mantenimiento del sistema para que del permiso para agregar. También se hacen y registran la venta a domicilio por didi, rapid o Uber.

### 4.3.- Usuarios Finales y Características

Los usuarios finales se dividen en dos roles:

- El administrador que podrá acceder a todo el programa no tiene limitantes.
- El trabajador que la única limitante que tiene es no poder agregar usuarios nuevos, ya que necesitara el permiso del administrador.

No se agregaron más limitantes, porque la mayoría del tiempo quien operara el POS será el mismo dueño o administrador de la tienda, los trabajadores solo serán temporales y no operaran tanto tiempo en el POS.

Sus responsabilidades de registrar ventas, generar tickets, recibir los pagos, dar el cambio, gestionar inventario, agregar o eliminar productos, actualizar precios, aplicar descuentos, generar reportes detallados de ventas u otros, son responsabilidades compartidas, pero con ciertas limitantes.

## 4.4.- Funciones del sistema

El sistema POS debe incluir una serie de funcionalidades esenciales que permite a los usuarios realizar sus tareas de manera eficiente, las principales funciones del POS son:

### 1) Inicio de sesión (login)

- a) Autenticación de usuario con nombre y contraseña.
- b) Roles: Administrador (Admin) y trabajador (Trab).
- c) Restricción de funciones para trabajadores (sin acceso a agregar usuarios).
- d) Botón oculto para registrar un usuario temporal (para recuperar contraseña o ver datos).

### 2) Menú principal

- a) Acceso rápido a módulos:
  - i) Ventas
  - ii) Clientes
  - iii) Proveedores
  - iv) Inventario
  - v) Gestión de usuario
  - vi) Reportes
  - vii) Cerrar sesión
- b) Visualización del nombre del usuario actual.
- c) Cambio de usuario desde el nombre mostrado.

### 3) Módulo de ventas

- a) Selección visual de productos sin necesidad de código de barras.
- b) Filtros por categoría.
- c) Búsqueda por nombre de producto.
- d) Carrito de compras editable (agregar, eliminar productos).
- e) Control de cantidad por producto antes de añadir al carrito.
- f) Modificación de cantidad de productos ya añadidos.
- g) Visualización del total de compra.
- h) Método de pago: efectivo (requiere ingresar monto), tarjeta (proceso directo).
- i) Teclado numérico en pantalla para ingresar montos o número de cliente.
- j) Asociación de ventas con clientes para acumulación de puntos (5 puntos = 1 peso).
- k) Cancelación de ventas con botón dedicado.
- l) Generación automática de ticket de venta al finalizar.
- m) Impresión o visualización de ticket.

#### 4) Módulo de reportes

Acceso desde menú principal o menú horizontal superior.

- a) Submódulo de reporte de ventas
  - viii) Filtros: todos, hoy, semana, mes, rango personalizado.
  - ix) Exportación en: PDF, EXCEL, CSV.
  - x) Pestañas: resumen, detalle, gráficos.
  - xi) Datos mostrados:
    - (1) Total, de ventas
    - (2) Ventas del día
    - (3) Productos vendidos
    - (4) Ticket promedio
- b) Submódulo de reporte de inventario
  - xii) Filtros: bajo stock, sobre stock, por categoría, por proveedor.
  - xiii) Exportación igual que en ventas.
  - xiv) Gráficos de inventario.
  - xv) Información:
    - (1) Total, productos
    - (2) Bajo stock
    - (3) Sin stock
- c) Submódulo de reportes de clientes
  - xvi) Filtros: todos, activos, inactivos, con puntos, sin puntos.
  - xvii) Gráficos e información:
    - (1) Total, clientes
    - (2) Clientes activos
    - (3) Clientes inactivos
- d) Submódulo de reporte de proveedores
  - xviii) Filtros: recientes, sin visita reciente, por producto.
  - xix) Información:
    - (1) Total, proveedores
    - (2) Visitados este mes
    - (3) Ultima visita
- e) Reimpresión de documentos
  - xx) Búsqueda por ID, fecha o método de pago.
  - xxi) Reimpresión de:
    - (1) Tickets de venta
    - (2) Reportes de ventas
    - (3) Reportes de inventarios
    - (4) Reporte de clientes
    - (5) Reporte de proveedores



## **5) Módulo de inventario**

- a) Visualización general de productos.
- b) Filtros por categoría y proveedor.
- c) Búsqueda rápida y avanzada por nombre, precio, estado, IVA, etc.
- d) Agregar productos con datos:
  - I. Nombre, descripción, imagen, categoría, proveedor, estado, unidad
  - II. Stock actual, mínimo y máximo
  - III. Precio de compra y venta
  - IV. Iva y descuento
- a) Editar y eliminar productos.
- b) Ver historial de movimientos:
  - I. Entradas (cantidad, fecha)
  - II. Salidas (motivo, observación)
  - III. Ajustes (tipo, cantidad, descripción)
- a) Devoluciones:
  - I. Devolución de cliente (id venta, motivo)
  - II. Devolución a proveedor
  - III. Historial de devoluciones

## **6) Módulo de clientes**

- a) Ver lista de clientes registrados
- b) Agregar cliente (nombre, teléfono, fecha)
- c) Editar cliente (nombre, teléfono, fecha).
- d) Eliminar cliente.
- e) Visualización de puntos acumulados y fecha de ultima compra.

## **7) Módulo de proveedores**

- a) Ver lista de proveedores registrados.
- b) Agregar proveedor (nombre, dirección, teléfono, producto, fecha última visita).
- c) Editar proveedor.
- d) Eliminar proveedor

## **8) Módulo de gestión de usuario**

- a) Ver lista de usuarios.
- b) Agregar nuevo usuario (nombre, contraseña, rol).
- c) Editar usuario (solo cambio de rol).
- d) Eliminar usuario.

## **9) Funciones adicionales / avanzadas**



- a) Navegación entre módulos desde menú horizontal (sin volver al menú principal solo al dar salir).
- b) Registro temporal de usuarios para recuperación rápida.
- c) Contactar proveedor desde vista de producto.
- d) Actualización dinámica de datos desde base de datos Access.
- e) Diseño gráfico amigable e intuitivo con Swing.
- f) Sistema sin conexión a internet, completamente local.

## 4.5.- Entorno de Operación

Con base en la descripción detallada, el entorno de operación del sistema de punto de venta (POS) puede definirse:

### 1) Usuario del sistema

- I. Administrador
  - a. Acceso total al sistema.
  - b. Puede gestionar productos, clientes, proveedores, usuarios, reportes e inventario.
- II. Accede directamente al menú principal tras iniciar sesión.
- III. Trabajador (empleado de apoyo)
- IV. Acceso restringido a funciones específicas.
- V. Puede reimprimir tickets y reportes, pero no modificar configuraciones ni registros críticos.
- VI. Usuario temporal
- VII. Acceso limitado al menú principal para consultar su contraseña desde “Gestión de usuarios”.
- VIII. Función creada para usuarios con dificultades para recordar contraseñas.

### 2) Hardware requerido

- I. PC o laptop o tableta, con sistema operativo compatible con java y base de datos Microsoft Access.
- II. Pantalla táctil opcional para facilitar la interacción (dado el uso de botones grandes y teclado numérico virtual).
- III. Impresora de tickets.
- IV. Lector de código de barras **no necesario** (la selección de productos es visual y por búsqueda manual).
- V. Teclado físico opcional (el sistema incluye un teclado numérico visual en la interfaz para situaciones donde el teclado físico no funcione).

### 3) Software utilizado

- I. Sistema operativo: Windows (ideal por la compatibilidad con Microsoft Access),
- II. Lenguaje de desarrollo: Java (Swing para GUI).
- III. Base de datos: Microsoft Access(.accdb).
- IV. Soporte para exportación de reportes a: PDF, CSV, etc.

- V. Herramientas externas: posibles librerías java para gráficos, exportación, y conexión a Access (UCanAccess o similar).

#### **4) Requisitos de funcionamiento**

- I. Java runtime Environment (JRE) instalado.
- II. Conectividad local (no requiere conexión a internet, todo el sistema es local).
- III. Resolución de pantalla suficiente para mostrar interfaces con múltiples secciones.
- IV. Permisos de escritura en disco para:
  - a. Guardar tickets, reportes, exportaciones.
  - b. Mantener archivos temporales o configuraciones.

#### **5) Entorno físico de operación**

- I. Local comercial físico (tienda, minisúper, abarrotes).
- II. Uso principal en punto de venta y mostrador.
- III. Operado por una o dos personas simultáneamente.
- IV. Flujo constante de clientes durante horarios pico (requiere fluidez en la interfaz).

#### **6) Manejo de seguridad**

- I. Acceso por login con nombre de usuario y contraseña.
- II. Control de sesión activa visible (nombre del usuario siempre en pantalla).
- III. Opción para cambiar de usuario desde cualquier ventana.
- IV. Gestión de roles y privilegios básicos (admin/trabajador).

#### **7) Restricciones del sistema**

- I. No contempla conexión remota ni multi-sucursal.
- II. No está diseñado para operación en línea o móvil.
- III. La base de datos es local, por lo que es sensible a fallos de disco o cierre inesperado del sistema.
- IV. Facilidad de uso
- V. Interfaz amigable e intuitiva, con acceso visual a todos los productos.
- VI. Botones grandes, iconos claros y distribución de elementos por secciones.
- VII. Uso de filtros, teclados y herramientas visuales para usuarios con poca experiencia técnica.

## 5.- Requerimientos Funcionales

Los requerimientos funcionales definen las funcionalidades que el POS debe ofrecer.

### 5.1.- Flujo básico (Requisitos funcionales)

A continuación, se describe el flujo básico de uso del sistema, que involucra las tareas esenciales que un usuario realizara en el sistema.

Los principales procesos funcionales incluyen:

- **Inicio de sesión:**

- **Funcionabilidad:** el sistema debe permitir a los usuarios ingresar con un nombre de usuario y contraseña. Dependiendo del rol del usuario, el acceso se configura para mostrar un conjunto específico de funcionalidades.
- **Requisitos:**
  - El sistema deberá verificar que el nombre de usuario y la contraseña coincidan con los registros en la base de datos.
  - Si el usuario es válido, se mostrará la interfaz de acuerdo con su rol.

- **Gestión de productos (Inventario):**

- **Funcionalidad:** los usuarios deben poder agregar, modificar o eliminar productos del inventario.
- **Requisitos:**
  - Los productos deben tener un nombre, precio unitario, cantidad en stock, categoría, etc.
  - El sistema debe permitir buscar productos por nombre o código.
  - Los cambios realizados en los productos deben reflejarse inmediatamente en el inventario.
  - Se podrá realizar devoluciones o entradas de productos.

- **Registro de venta (venta):**

- **Funcionalidad:** el usuario podrá registrar los productos que el cliente desea comprar en el carrito, seleccionándolos sin necesidad de código de barras.
- **Requisitos:**
  - El sistema debe calcular automáticamente el subtotal de la venta, el impuesto (IVA) y los descuentos aplicables.
  - El sistema debe generar un ticket de venta con todos los detalles de la transacción (productos comprados, precio unitario, cantidad, total, IVA, descuentos y monto a pagar).
  - El inventario debe actualizarse después de cada venta,

- **Generación de reportes (Reportes):**

- **Funcionalidad:** el sistema debe permitir a los usuarios generar reportes detallados de ventas, productos, proveedores, inventarios y clientes.

- **Requisitos:**
  - Los reportes deben ser exportables a formatos como PDF, Excel, CSV, etc.
  - Los reportes deben poder filtrarse por fecha, producto, categoría, proveedor, etc.
- **Devoluciones y ajuste de inventario:**
  - **Funcionalidad:** Los usuarios deben poder registrar devoluciones o cambios de productos.
  - **Requisitos:**
    - El sistema debe permitir ajustar el inventario en caso de una devolución o cancelación de venta tanto para cliente o proveedor.
    - El sistema debe reflejar las devoluciones en los reportes de ventas,
- **Gestión de usuarios:**
  - **Funcionalidad:** Los administradores deben poder crear, modificar o eliminar usuarios con diferentes roles (Admin, Trabj)
  - **Requisito:**
    - El sistema debe permitir asignar diferentes permisos según el rol del usuario.
    - Los administradores deben poder restablecer contraseñas y gestionar accesos.

## 5.2.- Modelo de caso de uso

El modelo de casos de uso describe las interacciones entre los usuarios y el sistema de punto de venta. Cada funcionalidad clave se representa mediante un caso de uso, especificando los actores involucrados y el flujo básico de las tareas que se llevan a cabo dentro del sistema.

### Caso de Uso 1: Inicio de sesión

- **Actor principal:** Usuario (Administrador, Trabajador (Cajero))
- **Descripción:** El usuario ingresa sus credenciales para acceder al sistema. Dependiendo del rol, se habilitan diferentes funciones.
- **Flujo básico:**
  1. El usuario introduce su nombre de usuario y contraseña.
  2. El sistema valida las credenciales con la base de datos.
  3. El sistema redirige al usuario a la interfaz correspondiente (menú principal con opciones habilitadas según rol).
  4. Si el usuario se olvida de sus credenciales se puede realizar un registro de usuario temporal.

### Caso de Uso 2: Registrar una venta

- **Actor principal:** Trabajador (Cajero)

- **Descripción:** El trabajador registra los productos que el cliente desea comprar, el sistema calcula totales, aplica descuentos si corresponde y genera un ticket de venta.
- **Flujo básico:**
  1. El cajero selecciona productos desde la interfaz.
  2. El sistema calcula el subtotal, aplica descuentos e IVA.
  3. El sistema actualiza automáticamente el inventario.
  4. Se genera e imprime el ticket de venta.

### **Caso de Uso 3: Registrar productos (Inventario)**

- **Actor principal:** Administrador
- **Descripción:** Permite agregar nuevos productos al inventario incluyendo detalles como nombre, categoría, proveedor, precio y cantidad.
- **Flujo básico:**
  1. El administrador accede al formulario de productos.
  2. Ingresar los datos requeridos.
  3. El sistema guarda el producto en la base de datos y actualiza la tabla visual.

### **Caso de Uso 4: Gestión de inventario**

- **Actor principal:** Administrador
- **Descripción:** Permite al administrador revisar existencias, agregar o modificar cantidades, y ver historial de movimientos.
- **Flujo básico:**
  1. El administrador accede al módulo de inventario.
  2. Consulta productos con bajo stock.
  3. Puede registrar movimientos de entrada o salida.

### **Caso de Uso 5: Registrar cliente**

- **Actor principal:** Trabajador (Cajero)/ Administrador
- **Descripción:** Permite ingresar información de nuevos clientes para asociarlos a ventas o mantener un historial.
- **Flujo básico:**

1. El usuario accede al módulo de clientes.
2. Introduce los datos del cliente.
3. El sistema guarda al cliente en la base de datos.

#### **Caso de Uso 6: Gestión de proveedores**

- **Actor principal:** Administrador
- **Descripción:** Permite agregar, editar o eliminar proveedores asociados a productos del sistema.
- **Flujo básico:**
  1. El administrador accede al módulo de proveedores.
  2. Gestiona la información de contacto y productos suministrados.
  3. El sistema actualiza la información.

#### **Caso de Uso 7: Generar reportes**

- **Actor principal:** Administrador
- **Descripción:** El administrador puede generar reportes sobre ventas, inventario, clientes y proveedores. Estos pueden exportarse en formatos PDF o Excel.
- **Flujo básico:**
  1. El administrador selecciona el tipo de reporte.
  2. El sistema recupera los datos desde la base.
  3. El sistema muestra el reporte y permite exportarlo.

#### **Caso de Uso 8: Gestión de usuarios**

- **Actor principal:** Administrador
- **Descripción:** Permite crear, editar o eliminar usuarios del sistema y asignarles roles.
- **Flujo básico:**
  1. El administrador accede a la sección de usuarios.
  2. Registra un nuevo usuario o edita uno existente.
  3. El sistema guarda los cambios.

### 5.3.- Precondiciones

Las precondiciones son las condiciones que deben cumplirse antes de ejecutar ciertas funcionalidades del sistema de punto de venta. Aseguran que los procesos tengan el contexto y los datos necesarios para su correcto funcionamiento.

#### Inicio de sesión

- El usuario debe tener una cuenta registrada previamente en el sistema (almacenada en archivo o base de datos).
- El usuario debe conocer su nombre de usuario y contraseña válidos.
- El sistema debe estar operativo y mostrar la pantalla de inicio de sesión.
- La conexión al archivo de usuarios debe estar disponible.

#### Registrar una venta

- El usuario debe haber iniciado sesión con el rol de trabajador o administrador.
- El producto debe estar registrado en el inventario
- El producto debe tener una cantidad en stock mayor a cero.
- El sistema debe tener acceso al archivo/ base de datos del inventario.
- El sistema debe poder generar e imprimir ticket de venta.

#### Registrar un producto

- El usuario debe tener rol de administrador.
- Debe haber al menos una categoría y un proveedor registrados en el sistema.
- Todos los campos requeridos (nombre, precio, cantidad, proveedor, etc.) deben estar completos y ser válidos.
- El sistema debe tener acceso de escritura al almacenamiento de productos,

#### Registrar un cliente

- El usuario debe haber iniciado sesión.
- El cliente por registrar no debe existir ya (validación por nombre y datos únicos).
- Debe completarse toda la información obligatoria (nombre, teléfono, etc.)
- El sistema debe tener acceso al almacenamiento de clientes,

#### Generar reportes

- El usuario debe tener rol de administrador.
- El tipo de datos necesarios (ventas, productos, inventario, clientes) debe estar disponibles en el sistema.
- El sistema debe tener acceso a los archivos o base de datos requeridos.
- El sistema debe tener configurada una ruta válida para exportar los reportes (PDF, Excel, etc.)

### Consultar inventario

- El usuario debe haber iniciado sesión como administrador.
- Deben existir productos registrados en el sistema.
- El archivo de inventario debe estar disponible y sin errores,

### Registrar proveedor

- El usuario debe haber iniciado sesión como administrador.
- Los datos del proveedor (nombre contacto et.) deben estar completos.
- El proveedor no debe estar previamente registrado.

### Registrar usuario

- Solo el administrador puede registrar nuevos usuarios.
- Los campos obligatorios (usuario, contraseña, rol) deben estar llenos y ser valido.
- El nombre de usuario no debe estar repetido.

## 2.4.- Postcondiciones

Las postcondiciones representan el estado del sistema **después de la ejecución exitosa** de una funcionalidad. Aseguran que se haya producido un cambio esperable en los datos, el sistema o la interfaz.

### Inicio de sesión

- El sistema verifica las credenciales y permite el acceso al usuario.
- Se despliega la interfaz principal correspondiente al rol del usuario (Trabajador o Administrador).
- Se habilitan solo las funcionalidades autorizadas según el rol del usuario.

### Registrar una venta

- El sistema actualiza el inventario, reduciendo las existencias de los productos vendidos.
- Se genera un ticket de venta con todos los detalles (productos, precios, IVA, total).
- La información de la venta queda almacenada en el historial para reportes futuros.
- (Opcional) Se asocia la venta a un cliente si fue seleccionado.

### Registrar un producto

- El producto se guarda en la base de datos (u otro archivo) con todos sus datos.
- El sistema actualiza la lista visual de productos disponibles.



- El producto queda disponible para ventas e inventario.

#### **Registrar un cliente**

- El nuevo cliente se guarda en el sistema.
- La lista de clientes se actualiza.
- El cliente puede ser seleccionado en futuras ventas.

#### **Generar reportes**

- El reporte solicitado se genera correctamente con los datos actuales del sistema.
- El usuario puede visualizarlo y/o exportarlo en formato PDF o Excel.
- (Opcional) Se guarda una copia del reporte en la ubicación definida.

#### **Consultar inventario**

- Se muestra la lista actual de productos con su cantidad, categoría y proveedor.
- Los productos con bajo stock pueden marcarse para reposición.
- El usuario puede realizar movimientos de stock desde esta vista.

#### **Registrar proveedor**

- El proveedor se guarda correctamente en el sistema.
- El proveedor queda disponible para ser asignado a nuevos productos.
- Se actualiza la lista de proveedores disponibles.

#### **Registrar usuario**

- El nuevo usuario se guarda en el sistema con su rol y credenciales.
- El usuario puede iniciar sesión con sus datos en futuras sesiones.
- La lista de usuarios se actualiza automáticamente.

## 6.- Diseño del sistema

El diseño del sistema busca garantizar la modularidad, escalabilidad y mantenibilidad del software de punto de venta. Se adoptó una arquitectura orientada a objetos, donde cada componente desempeña un rol claro y bien definido, facilitando futuras extensiones y correcciones. A continuación, se detallan los principales aspectos del diseño:

### 6.1.- Arquitectura General

El sistema está estructurado bajo el patrón de arquitectura **Modelo-Vista-Controlador (MVC)**:

- **Modelo:** Encapsula la lógica de negocio y el acceso a datos. Incluye clases como Producto, Cliente, Carrito, Inventario, y sus respectivos DAO (Data Access Object), como ClienteDAO y InventarioDAO.
- **Vista:** Implementada con Java Swing, gestiona la interfaz gráfica del usuario, permitiendo realizar ventas, registrar clientes, consultar inventario, etc.
- **Controlador:** Gestiona la lógica de aplicación entre vista y modelo. Clases como VentaContro, ProductoControlador y ClientesContro manejan eventos, validaciones y coordinación entre componentes.

### 6.2.- Uso de interfaces y clases abstractas

El sistema emplea interfaces y clases abstractas para promover la extensibilidad y el desacoplamiento:

- **Interfaces DAO:** Interfaces como ClienteDAO y DevolucionDAO definen los métodos estándar de acceso a datos (insertar, eliminar, buscar), permitiendo cambiar la implementación sin afectar el resto del sistema.
- **Clases Abstractas o Base:** Aunque en esta versión concreta no se detectan clases abstractas explícitas, muchas clases modelo implementan estructuras comunes que podrían beneficiarse de una futura abstracción.

### 6.3.- Patrones de diseño utilizados

El diseño incorpora diversos patrones de diseño que facilitan la organización y mantenibilidad:

- **DAO (Data Access Object):** Aísla la lógica de persistencia de datos en clases como ClienteDAOImpl, facilitando la migración de base de datos o pruebas.
- **Factory (potencial):** La creación de instancias de productos o clientes podría automatizarse mediante un Factory para evitar dependencias fuertes y facilitar pruebas o modularidad.

- **Singleton:** La clase ConexionAccess implementa el patrón Singleton al asegurar una única conexión compartida a la base de datos.
- **MVC:** Como se mencionó, el sistema sigue fielmente este patrón, separando la interfaz de usuario, la lógica de negocio y la lógica de acceso a datos.

## 7.- Diccionario de Datos de las clases

### 1.- Clase: Carrito.java

Gestiona los productos seleccionados para una compra, con operaciones para agregar, eliminar y calcular totales.

#### Atributos:

Nombre	Tipo	Descripción	Restricciones
carrito	List<Producto>	Lista de productos en el carrito.	Inicializado como ArrayList<>.
totalVenta	double	Suma total de los precios (precioUnitario × cantidad).	≥ 0.0, se actualiza automáticamente.

#### Métodos:

Método	Parámetros	Retorno	Descripción
agregarProducto	id (String), nombre (String), precio (double), cantidad (int)	void	Agrega o actualiza un producto (si ya existe). Actualiza totalVenta.
actualizarTotal	—	void	Recalcula totalVenta sumando precioUnitario × cantidad de cada producto.
cancelarVenta	—	void	Vacía el carrito y establece totalVenta = 0.0.

Método	Parámetros	Retorno	Descripción
eliminarProducto	idProducto (String)	boolean	Elimina un producto por su ID. Retorna true si se eliminó, false si no.
getTotalVenta	—	double	Devuelve el total actual de la venta.
setTotalVenta	total (double)	void	(Uso interno) Establece el total manualmente.
getProductos	—	List<Producto>	Devuelve una <b>copia</b> de la lista para evitar modificaciones externas.
getCarrito	—	List<Producto>	Devuelve la lista original de productos (riesgo de modificación externa).

## 2.- Clase: Producto.java

Representa un producto en el inventario o en una venta.

### Atributos:

Nombre	Tipo	Descripción	Restricciones
id	String	Identificador único del producto.	No nulo.
nombre	String	Nombre del producto.	No nulo.
cantidad	int	Cantidad en el carrito (para ventas).	$\geq 0$ .

Nombre	Tipo	Descripción	Restricciones
precioUnitario	double	Precio por unidad.	$\geq 0.0$ .
precioVenta	double	Precio de venta al público.	$\geq 0.0$ , sincronizado con precioUnitario.
descripcion	String	Descripción detallada.	Opcional.
categoria	String	Categoría del producto (ej. "Abarrotes").	Valores predefinidos.
stockMinimo	int	Límite inferior para alertas de reposición.	$\geq 0$ .
stockMaximo	int	Límite superior para alertas de exceso.	$\geq \text{stockMinimo}$ .
tieneIVA	boolean	Indica si el producto incluye IVA.	true/false.
unidadMedida	String	Unidad de medida (ej. "kg", "litros").	Opcional.

#### Métodos Clave:

Método	Descripción
getPrecioConIVA()	Retorna precioVenta * 1.16 si tieneIVA=true.
necesitaReposicion()	true si cantidadDisponible $\leq$ stockMinimo.
tieneExcesoStock()	true si cantidadDisponible $>$ stockMaximo.

### 3. -Clase: Venta.java

Registra una transacción de venta.

#### Atributos:

Nombre	Tipo de Dato	Descripción
id	int	Identificador único de la venta.
fecha	Date / Timestamp	Fecha y hora en que se realizó la venta.
total	double	Monto total de la venta antes de descuentos.
descuento	double	Monto de descuento aplicado a la venta.
metodoPago	String	Medio de pago utilizado (efectivo, tarjeta, etc.).
montoRecibido	double	Cantidad de dinero que entregó el cliente.
cliente	Cliente	Objeto que representa al cliente asociado con la venta.
productos	List<Producto>	Lista de productos incluidos en la venta.

#### Métodos:

Método	Descripción
agregarProducto()	Añade un producto a la lista productos.
getFechaTimestamp()	Convierte la fecha a Timestamp para compatibilidad con bases de datos.
calcularTotal()	Suma precioConIVA de todos los productos.
aplicarDescuento()	Reduce si el cliente tiene puntos.

#### 4. -Clase: Cliente.java

Representa la información de los clientes del sistema.

**Atributos:**

Nombre	Tipo	Descripción
id	String	Identificador único.
telefono	String	Número de contacto.
puntos	int	Puntos de fidelidad acumulados.
fechaRegistro	Date	Fecha de alta (automática).

**Métodos:**

Método	Descripción
actualizarPuntos()	Añade puntos según el monto de compra.

#### 5.- Clase: Inventario.java

Gestiona el stock de productos.

**Atributos:**

Atributo	Tipo	Descripción
inventarioDAO	InventarioDAO	Conexión a la base de datos.
categorias	List<String>	Lista predefinida de categorías.

**Métodos Principales:**

Método	Parámetros	Retorno	Descripción
agregarProducto()	Producto	boolean	Añade un producto a la BD. Retorna éxito/fallo.
buscarPorNombre()	nombre (String)	List<Producto >	Búsqueda insensible a mayúsculas.
buscarPorCategoria()	categoria (String)	List<Producto >	Filtra por categoría. Si es "TODOS", retorna completa.
obtenerProductosBajoStock()	—	List<Producto >	Productos con cantidadDisponible ≤ stockMinimo.

## 6. -Clase: Usuario.java

Maneja credenciales y roles de usuarios.

### Atributos:

Atributo	Tipo de Dato	Descripción
username	String	Nombre de usuario utilizado para iniciar sesión.
password	String	Contraseña asociada al usuario.
rol	String	Rol asignado al usuario (por ejemplo: "Admin", "Cliente").

### Métodos:

Método	Descripción
esAdministrador()	Retorna true si el rol es "Admin".



## 7.- Clase: Proveedor.java

Gestiona la información de los proveedores que suministran productos al sistema.

### Atributos:

Atributo	Tipo	Descripción
id	String	Identificador único del proveedor (ej. "PROV-001").
nombre	String	Nombre legal o comercial del proveedor.
telefono	String	Número de contacto principal.
direccion	String	Dirección física (calle, ciudad, país).
productoSuministrado	String	Tipo de productos que suministra (ej. "Lácteos", "Electrónicos").
ultimaVisita	Timestamp	Fecha y hora de la última interacción (entrega, reunión, etc.).

### Métodos:

Método	Tipo Retorno	Descripción
getId()	String	Devuelve el ID del proveedor.
setId(String id)	void	Establece el ID. Validación: formato alfanumérico.
getNombre()	String	Obtiene el nombre del proveedor.

Método	Tipo Retorno	Descripción
setNombre(String nombre)	void	Establece el nombre. Validación: no nulo, longitud mínima 3.
getUltimaVisita()	Timestamp	Devuelve la fecha/hora de la última visita.
setUltimaVisita(Timestamp)	void	Actualiza la fecha de última visita (usar new Timestamp(System.currentTimeMillis()) para fecha actual).

## 8.- Clase: Devolución.java

Gestiona el proceso de devoluciones de productos, ya sea por parte de clientes o a proveedores.

### Atributos:

Atributo	Tipo	Descripción
id	String	Identificador único (ej. "DEV-001").
idProducto	String	ID del producto devuelto.
nombreProducto	String	Nombre del producto (para referencia rápida).
cantidad	int	Cantidad devuelta.
tipo	String	<b>"CLIENTE"</b> : Devolución por cliente. <b>"PROVEEDOR"</b> : Devolución a proveedor.
motivo	String	Razón de la devolución (ej. "Defectuoso", "No solicitado").
fecha	Date	Fecha de registro de la devolución.

Atributo	Tipo	Descripción
estado	String	"PENDIENTE": En revisión. "APROBADA": Aceptada. "RECHAZADA": Denegada.
observaciones	String	Comentarios adicionales.
idTransaccionOriginal	String	ID de la venta/compra original relacionada.
idUsuario	String	ID del usuario que registró la devolución.

#### Métodos:

Método	Tipo Retorno	Descripción
getEstado()	String	Devuelve el estado actual.
setEstado(String estado)	void	Cambia el estado. Valida que sea un valor permitido.
getPrecioProducto()	double	(Propuesto) Obtiene el precio unitario del producto desde Inventario.
calcularMontoTotal()	double	(Propuesto) Retorna cantidad * precioProducto.

## 9.- Reportes.java

Centraliza la generación de reportes del sistema (ventas, inventario, tickets).

### Atributos:

Atributo	Tipo	Descripción
usuarioActual	String	Usuario que genera el reporte.
fechaGeneracion	Date	Fecha y hora de creación del reporte.
tipoReporte	String	Tipo de reporte: - "VENTAS" - "INVENTARIO" - "TICKET" - "CLIENTES"
datosReporte	Object	Datos crudos para el reporte (puede ser List, Map, JSON, etc.).
formato	String	(Propuesto) Formato de salida: "PDF", "EXCEL", "HTML".
rutaGuardado	String	(Propuesto) Ruta donde se almacena el reporte generado.

### Métodos:

Método	Parámetros	Descripción
generarReporteVentas()	Date desde, Date hasta	Genera reporte de ventas en un rango de fechas. Retorna éxito/fallo.

Método	Parámetros	Descripción
generarReporteInventario()	String categoria	Filtra por categoría ("TODOS" para reporte completo).
exportarAPDF()	String ruta	(Propuesto) Exporta datosReporte a PDF.
setFiltros()	Map<String, Object>	(Propuesto) Añade filtros personalizados (ej: {"sucursal": "Centro"}).

## 8.- Modelo UML de Clases

### Modelo

#### 1. Entidades de dominio

Clase	Responsabilidad principal	Atributos clave	Relaciones
<b>Producto</b>	Representa cada artículo que puede venderse.	id, nombre, cantidad, precioUnitario, precioVenta, tieneIVA, descuento, fechaIngreso, estado, imagenPath, unidadMedida	<ul style="list-style-type: none"> <li>• <b>Inventario</b> mantiene una colección de <i>Producto</i> (composición).</li> <li>• Es referenciado desde <i>DetalleVenta</i> y <i>Carrito</i>.</li> </ul>
<b>Cliente</b>	Persona o empresa que compra.	id, nombre, telefono, correo, etc.	<ul style="list-style-type: none"> <li>• Relacionado 1..* con <i>Venta</i> (un cliente puede tener muchas ventas).</li> </ul>
<b>Proveedor</b>	Empresa que surte productos.	id, nombre, telefono, direccion, etc.	<ul style="list-style-type: none"> <li>• Relación 1..* con <i>Producto</i> (un proveedor puede surtir varios productos).</li> </ul>
<b>Inventario</b>	Existencias físicas.	idInventario, fechaActualizacion, stockTotal	<ul style="list-style-type: none"> <li>• Contiene (<i>has-many</i>) objetos <i>Producto</i>.</li> </ul>

Clase	Responsabilidad principal	Atributos clave	Relaciones
<b>Venta</b>	Transacción de salida registrada en el punto de venta.	id, fechaHora, total, metodoPago, estado	<ul style="list-style-type: none"> <li>• Agrega (<i>has-many</i>) <i>DetalleVenta</i>.</li> <li>• Tiene referencia a un <i>Cliente</i>.</li> <li>• Puede originar una <i>Devolucion</i>.</li> </ul>
<b>DetalleVenta</b>	Ítem dentro de una venta.	id, cantidad, precioUnitario, subtotal	<ul style="list-style-type: none"> <li>• Contiene un <i>Producto</i>.</li> <li>• Pertenece a una <i>Venta</i> (composición).</li> </ul>
<b>Carrito</b>	Representa las selecciones temporales antes de confirmar la venta.	listaDetalle, totalTemporal	<ul style="list-style-type: none"> <li>• Contiene <i>DetalleVenta</i> mientras el usuario opera la interfaz.</li> </ul>
<b>Devolucion</b>	Maneja artículos devueltos.	id, fecha, monto, motivo	<ul style="list-style-type: none"> <li>• Se asocia a 0..1 <i>Venta</i> y a uno o varios <i>DetalleVenta</i> devueltos.</li> </ul>
<b>Usuario</b>	Empleado que inicia sesión.	id, username, passwordHash, rol	<ul style="list-style-type: none"> <li>• Sus credenciales las gestiona <i>GestorUsuario</i> en la capa Controlador.</li> </ul>

## 2. Patrones DAO

Cada entidad persistente tiene:

- **<Entidad>DAO** – *interfaz* que declara las operaciones CRUD.
- **<Entidad>DAOImpl** – implementación concreta; encapsula SQL/ JDBC-ODBC (en este caso, contra un archivo Access: véase *ConexionAccess*).

Ejemplos:

java

Copiar código

ClienteDAO      <interface

ClienteDAOImpl    implements ClienteDAO

ProductoDAO      <interface

ProductoDAOImpl    implements ProductoDAO

...

Las implementaciones dependen de **ConexionAccess**, la clase fábrica de conexiones.

Relación UML: *ClienteDAOImpl* —————> *ClienteDAO* (realización)

*ClienteDAOImpl* —————> *ConexionAccess* (dependencia)

### 3. Capa Controlador (adaptadores)

- **ProductoControlador, ClientesContro, ProveedoresContro, VentaContro**, etc.
  - Actúan como *facades* para la GUI Swing.
  - Contienen instancias de los respectivos DAO para orquestar operaciones.
- **ControladorInventario** actualiza stock y precios, invocando métodos de *InventarioDAO* y *ProductoDAO*.
- **GestorUsuario** valida inicios de sesión contra *UsuarioDAO*.

### 4. Capa Vista (Swing)

No forma parte estricta del modelo UML pedida, pero completa el patrón MVC:

- Formularios como *FrmVenta*, *FrmProductos*, etc., observan a los controladores y disparan eventos.

### 5. Relaciones clave

- **Asociaciones**
  - *Venta* 1 —\* *Cliente*\* (una venta pertenece a un cliente).
  - *Producto* 1 —\* *Proveedor*\* (opcional; un producto puede tener un proveedor principal).
- **Agregación/Composición**
  - *Venta* **compone** *DetalleVenta*. Si se borra la venta, se eliminan sus detalles.
  - *Inventario* **compone** la lista de *Producto* (el producto puede existir fuera, por lo que suele modelarse como agregación; aquí se maneja como contención fuerte en la lógica).
- **Dependencias**
  - Controladores dependen de DAO e interfaces de Vista.
  - DAO dependen de *ConexionAccess*.
- **Generalización**
  - Cada *DAOImpl* *realiza* su interfaz DAO.

## 6. Patrones utilizados

Patrón	Dónde aparece	Propósito
<b>DAO</b>	Paquete Modelo	Abstraer acceso a BD.
<b>MVC</b>	Paquetes Modelo, Controlador, Vista	Separar lógica de negocio, interfaz y persistencia.
<b>Singleton</b>	ConexionAccess	Mantener única conexión a la BD Access.
<b>Factory Method</b> (simple)	Métodos getConnection()	Crear objetos Connection sin exponer detalles.

## 7. Flujo típico

1. **Vista** solicita acción (por ejemplo, “Agregar producto al carrito”).
2. **ProductoControlador** recibe la petición y actualiza **Carrito** (modelo en memoria).
3. Al confirmar, **VentaContro** crea una **Venta**, pobla sus **DetalleVenta** y los persiste usando **VentaDAO & DetalleVentaDAO**.
4. **ControladorInventario** descuenta existencias vía **InventarioDAO**.
5. Se actualiza la GUI y, opcionalmente, se genera un PDF mediante **PDFExporter**.

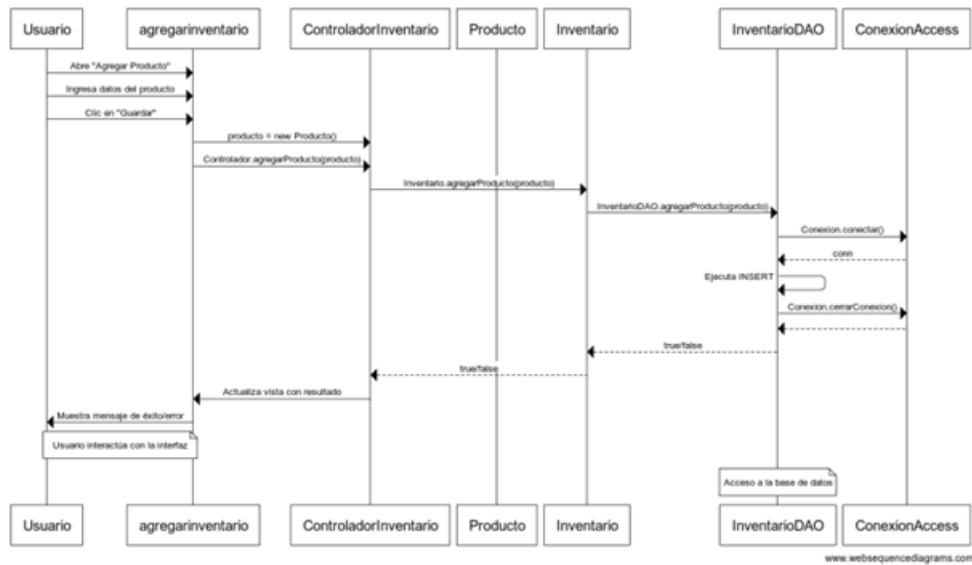


## 8.1.- UML Caso de uso

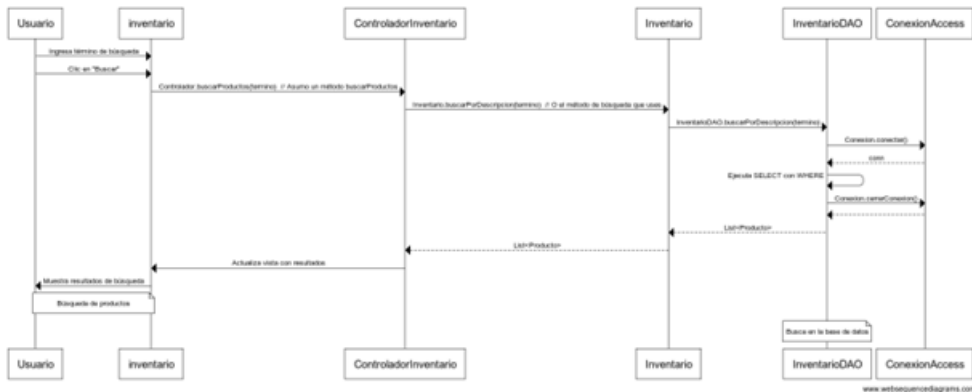


## 9.- Modelo de UML de Actividades

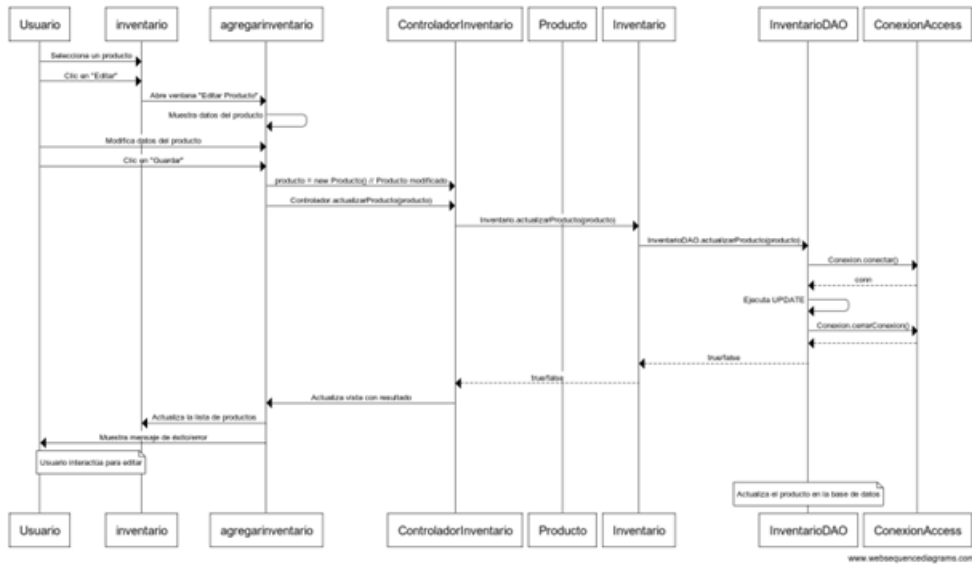
### Inventario



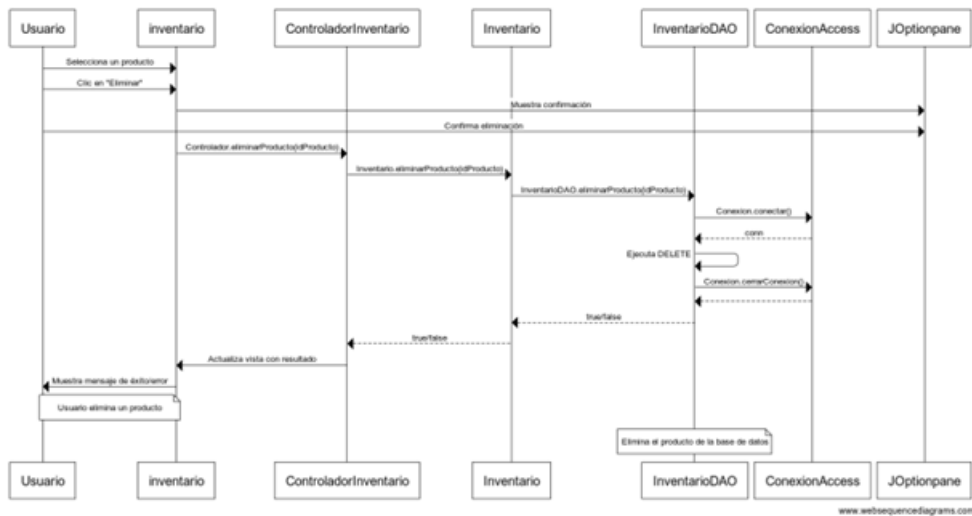
### Buscar producto



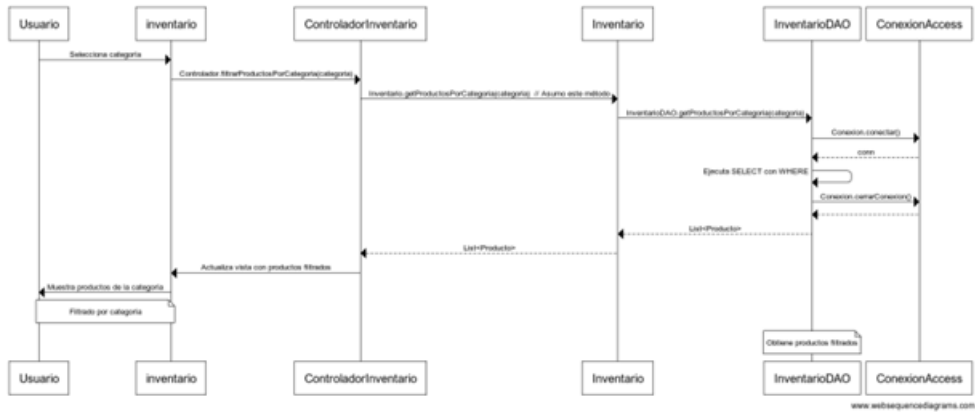
## Editar producto



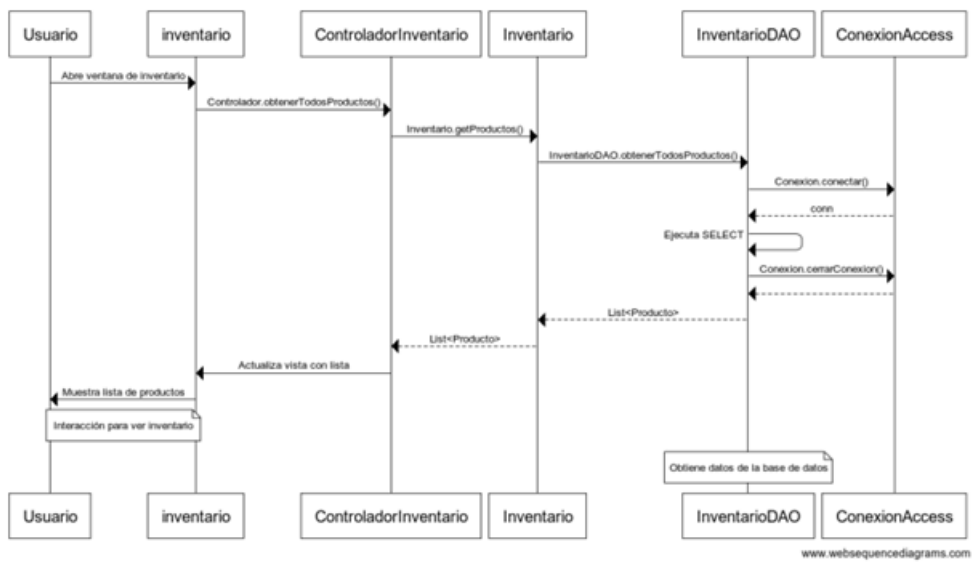
## Eliminar producto



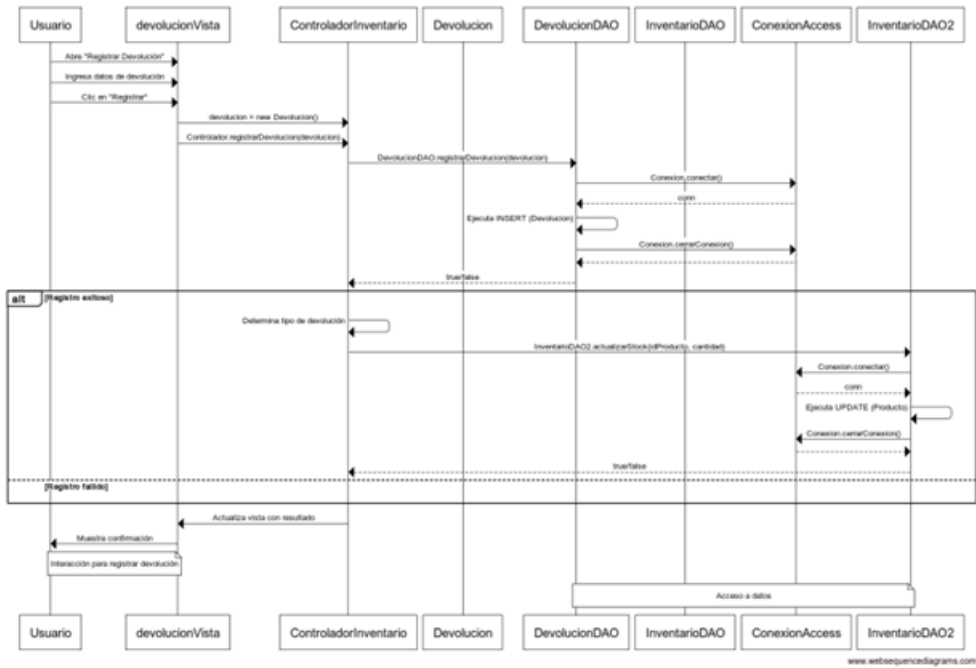
## Filtrar producto



## Mostrar lista producto



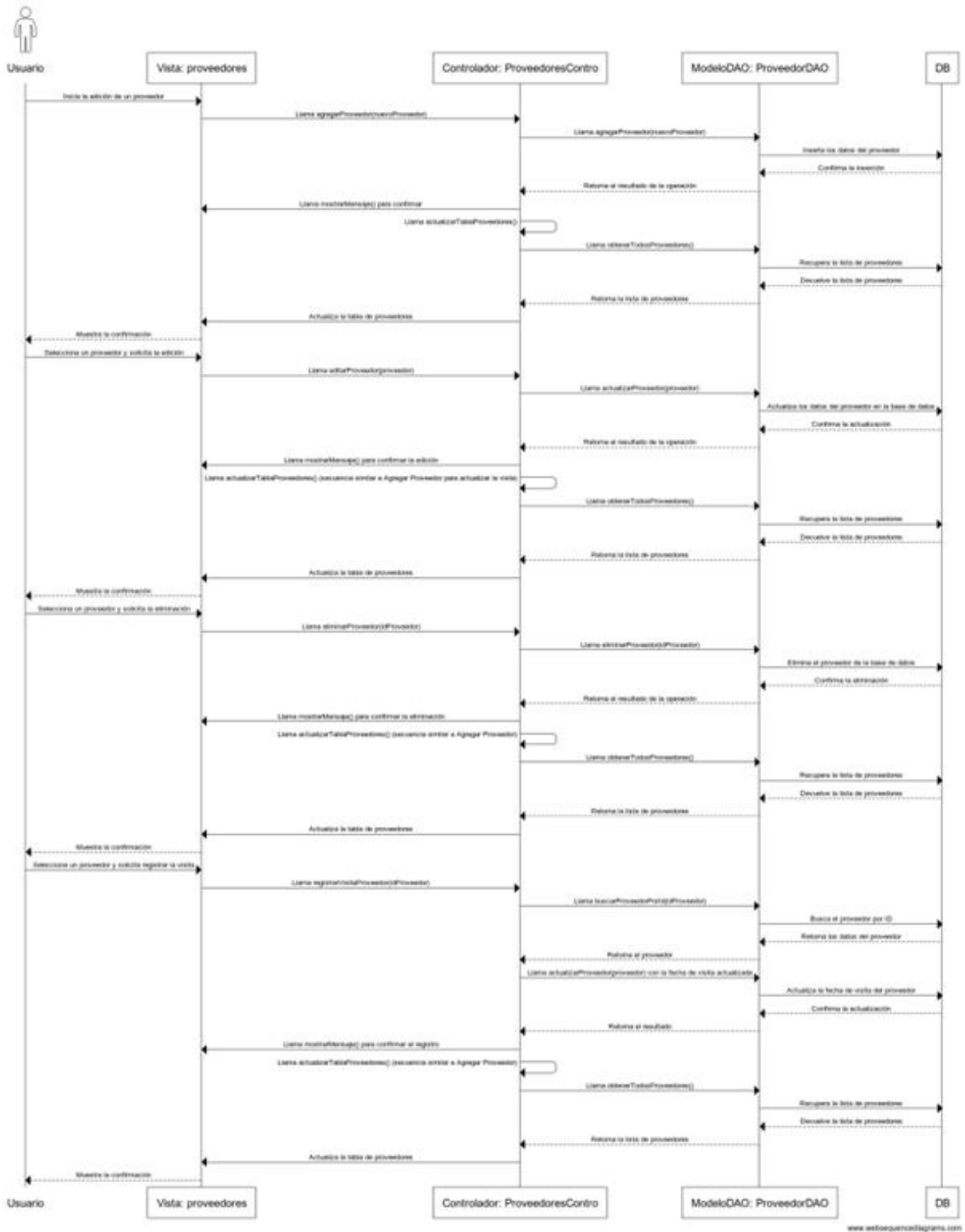
## Devolución



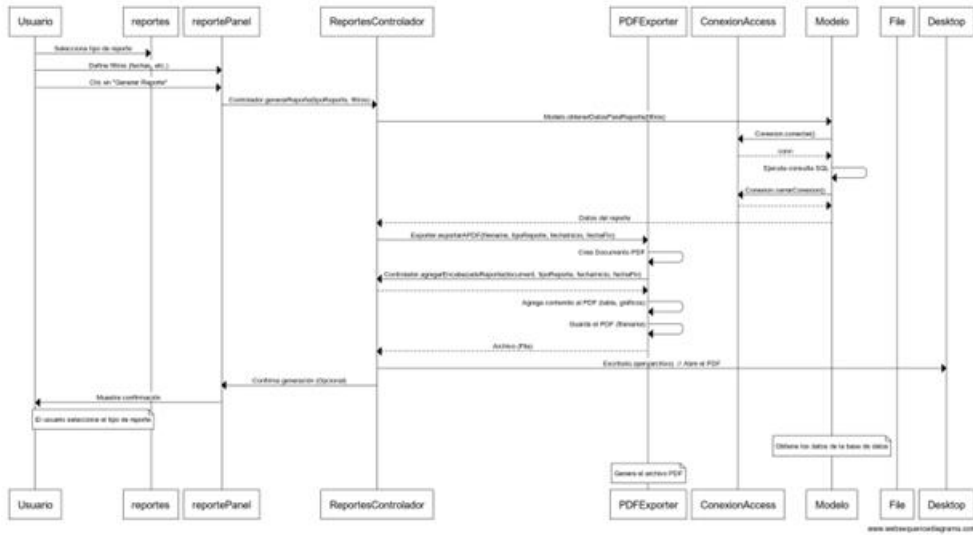
## Cientes



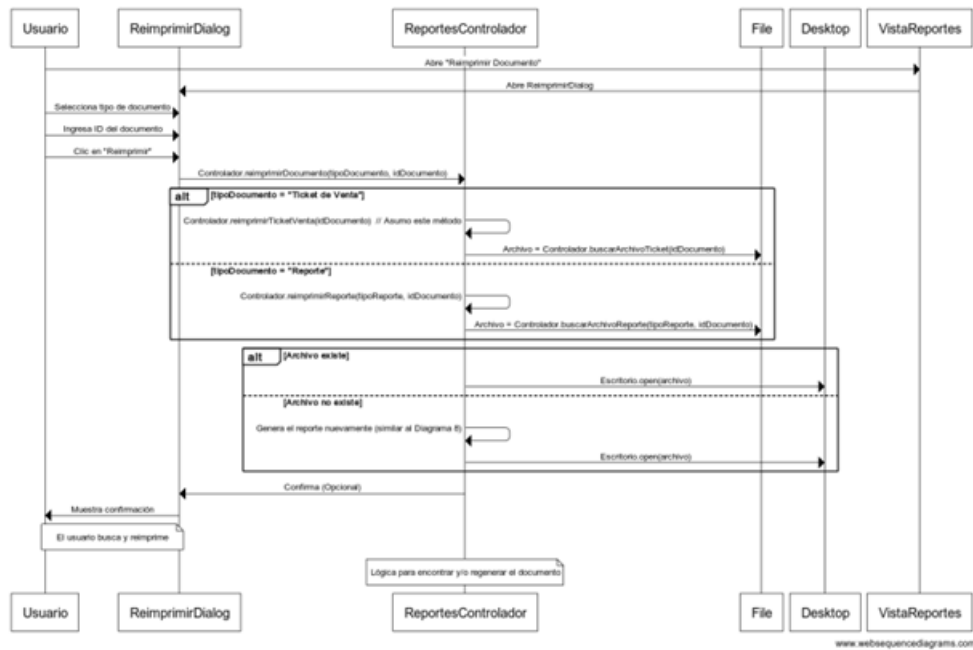
## Proveedores



## Reportes

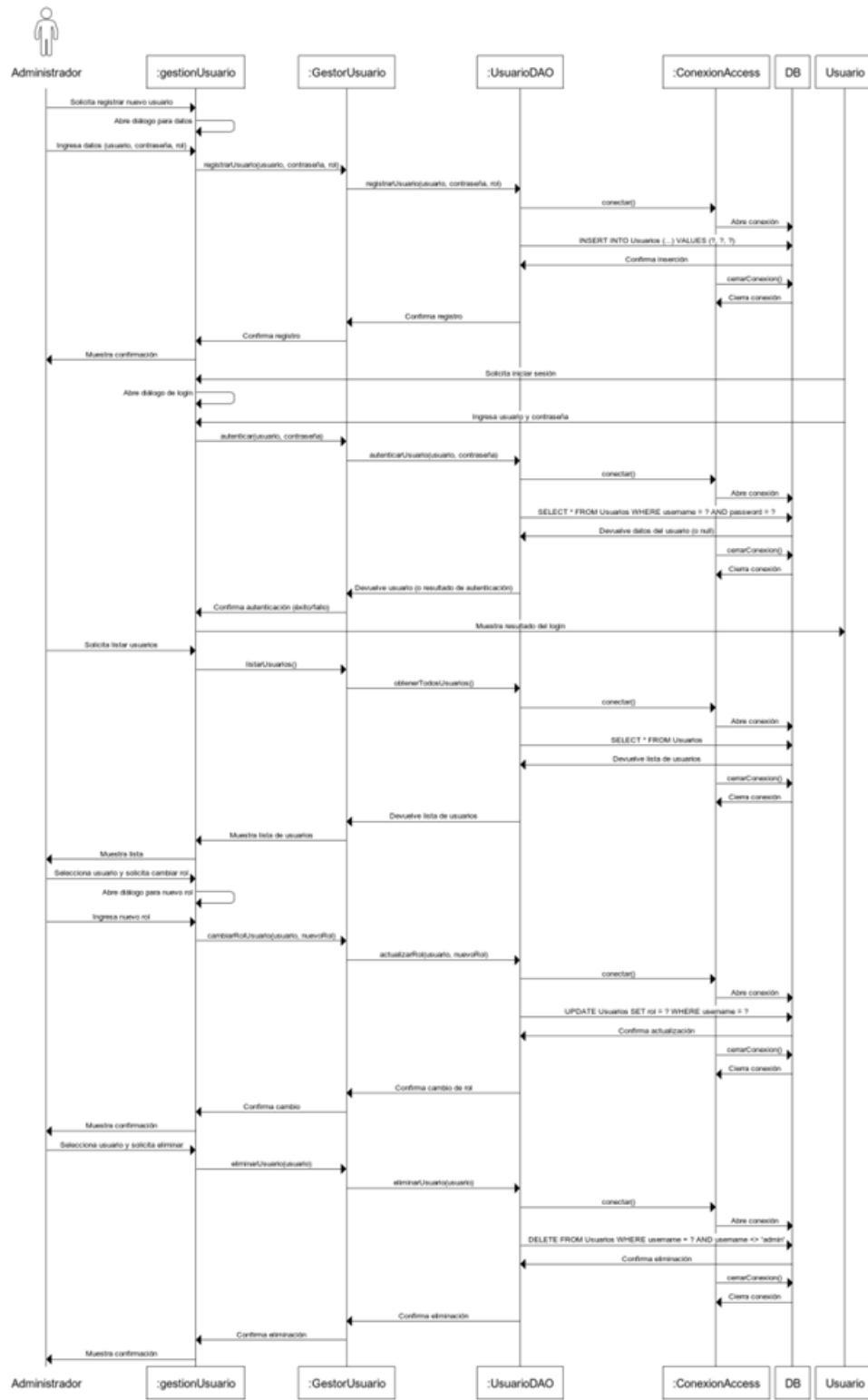


## Reimprimir

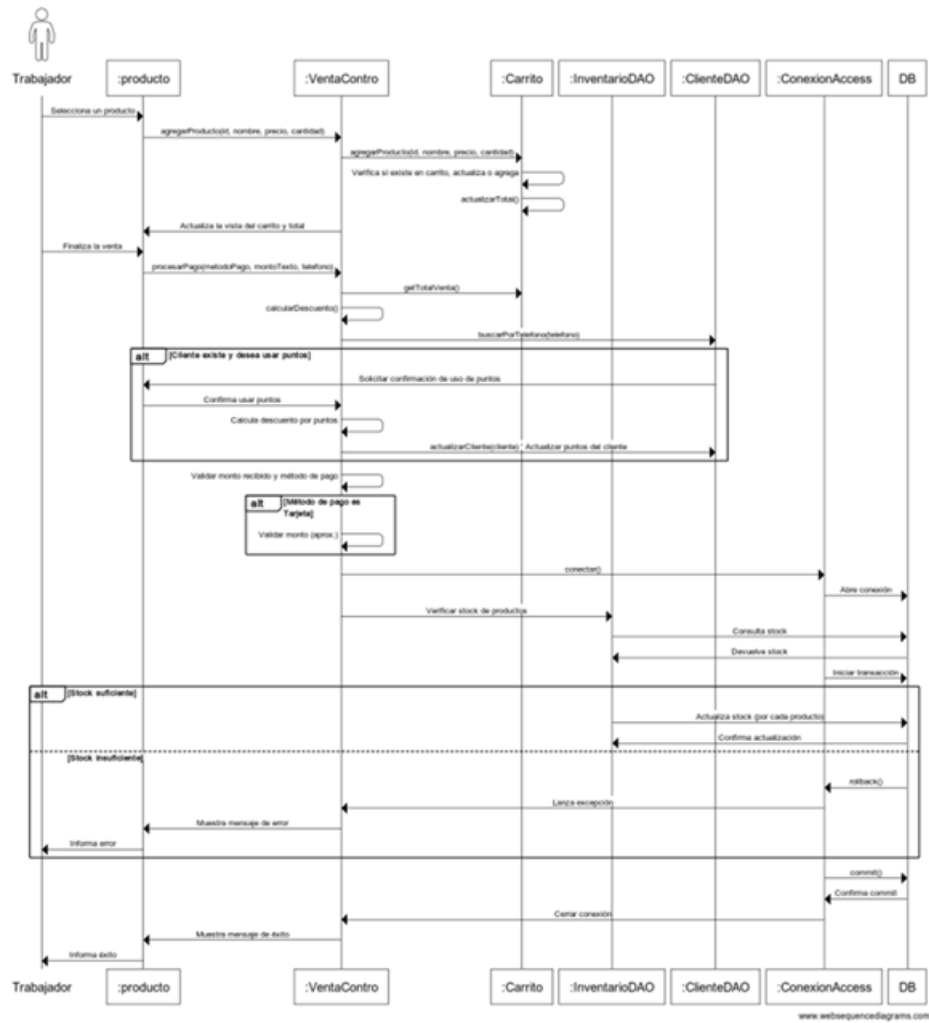




## Usuarios



## Venta



## 10.- Relaciones detectadas

El sistema se ha desarrollado bajo una arquitectura **MVC (Modelo - Vista - Controlador)**, lo que permite separar responsabilidades. A continuación, se describen las relaciones detectadas:

### Relación: Cliente – Venta

- **Tipo de relación:** Uno a muchos
- **Dirección:** Cliente → Venta
- **Justificación técnica:** En el sistema, cada venta debe estar registrada a nombre de un cliente. Esto es evidente en las clases donde se realiza una venta y se solicita el cliente correspondiente, como en interfaces de facturación.
- **Motivo del diseño:** Un cliente puede realizar múltiples compras a lo largo del tiempo, por lo que esta relación permite rastrear el historial de compras por cliente, analizar hábitos de consumo y generar reportes de ventas personalizadas.

### Relación: Venta – Producto

- **Tipo de relación:** Muchos a muchos
- **Dirección:** Bidireccional
- **Justificación técnica:** Una venta puede contener múltiples productos y, a su vez, un producto puede venderse muchas veces. Esta relación se evidencia en las estructuras que gestionan listas de productos dentro de la venta (Carrito) y en el procesamiento del total de la venta.
- **Motivo del diseño:** Es esencial para cualquier sistema de ventas manejar la relación entre productos vendidos y la venta concreta. Esto permite calcular subtotales, aplicar descuentos por producto y actualizar inventarios.

### Relación: Producto – Inventario

- **Tipo de relación:** Uno a uno o uno a muchos
- **Dirección:** Producto → Inventario
- **Justificación técnica:** Cada producto tiene una entrada de inventario asociada que indica su cantidad disponible. Esto se refleja en métodos que consultan o actualizan stock.
- **Motivo del diseño:** Esta relación permite llevar un control preciso del stock disponible de cada producto y aplicar lógicas como: evitar ventas sin stock, hacer ajustes por pérdidas o devoluciones, y consultar la disponibilidad.

### Relación: Producto – Proveedor

- **Tipo de relación:** Muchos a uno
- **Dirección:** Producto → Proveedor
- **Justificación técnica:** En el modelo, cada producto se asocia con un proveedor específico que lo suministra. Esta relación aparece en la gestión de nuevos productos, donde se selecciona un proveedor.
- **Motivo del diseño:** Permite identificar la fuente de los productos, optimizar las compras, organizar catálogos por proveedor, y contactar directamente al proveedor ante cambios de calidad o devoluciones.

#### **Relación: Carrito – Producto**

- **Tipo de relación:** Uno a muchos
- **Dirección:** Carrito → Producto
- **Justificación técnica:** El carrito contiene una lista de productos seleccionados por el cliente para realizar una compra. Esta relación es explícita en las listas internas de productos dentro del carrito.
- **Motivo del diseño:** El carrito actúa como una estructura temporal para preparar una venta antes de ser confirmada. Su existencia permite agregar o quitar productos, calcular totales en tiempo real y luego generar una venta final.

#### **Relación: Carrito – Venta**

- **Tipo de relación:** Uno a uno (en implementación)
- **Dirección:** Carrito → Venta
- **Justificación técnica:** El carrito funciona como base para generar la venta. En el momento de confirmar la compra, los productos del carrito se convierten en productos vendidos.
- **Motivo del diseño:** Esta relación facilita el flujo natural de una venta: selección → revisión → confirmación. La implementación modular lo permite sin mezclar la lógica de selección con la de facturación.

#### **Relación: Cliente – Devolucion**

- **Tipo de relación:** Uno a muchos
- **Dirección:** Cliente → Devolucion
- **Justificación técnica:** Cada devolución está asociada a un cliente que la solicita. El sistema necesita identificar quién hizo la devolución para mantener trazabilidad.

- **Motivo del diseño:** Permite registrar qué clientes han devuelto productos, identificar patrones (clientes con más devoluciones) y hacer ajustes en el inventario por devoluciones.

#### **Relación: Devolucion – Venta**

- **Tipo de relación:** Muchos a uno
- **Dirección:** Devolucion → Venta
- **Justificación técnica:** Toda devolución está relacionada a una venta previa. Esta relación asegura que solo se puedan devolver productos que efectivamente fueron comprados.
- **Motivo del diseño:** Garantiza integridad del sistema evitando devoluciones ficticias y facilita la validación de productos devueltos.

#### **Relación: Usuario – Módulos funcionales del sistema**

- **Tipo de relación:** Dependencia funcional
- **Dirección:** Usuario → Operaciones
- **Justificación técnica:** Los usuarios del sistema (empleados o administradores) no están relacionados directamente a las clases del modelo, pero son esenciales para operar módulos como ventas, inventario, y reportes.
- **Motivo del diseño:** A través de la autenticación y el acceso a módulos, se controla quién puede ejecutar operaciones. Aunque no sea una relación estructural (como atributos o clases anidadas), es una relación funcional vital para la seguridad y trazabilidad.

#### **Relación: Interfaces DAO – Implementaciones**

- **Tipo de relación:** Implementación de interfaz
- **Ejemplo:** ClienteDAOImpl implements ClienteDAO
- **Justificación técnica:** Las interfaces DAO definen operaciones como guardar, buscar, actualizar y eliminar. Las implementaciones (XXXDAOImpl) las concretan con lógica SQL o similar.
- **Motivo del diseño:** Este patrón desacopla la lógica de negocio de la lógica de acceso a datos, facilitando el mantenimiento, las pruebas, y posibles cambios futuros en la base de datos o tecnología (por ejemplo, cambiar de MySQL a MongoDB).

## **10.1.- Relaciones de Base de Datos**

las **relaciones** representan la forma en que las diferentes tablas están conectadas entre sí. Estas relaciones se establecen mediante **claves primarias (PK)** y **claves foráneas (FK)**, lo

que permite organizar la información de manera eficiente, evitar la redundancia de datos y mantener la integridad de los registros.

#### 1. Cliente (PK) → Venta (FK id\_cliente)

- **Tipo:** Uno a muchos.
- **Descripción:** Un cliente puede estar asociado a muchas ventas.
- **Justificación:** Permite mantener un historial de ventas por cliente.

#### 2. Venta ↔ Producto a través de Detalle\_Venta

- **Tipo:** Muchos a muchos.
- **Descripción:** Se usa una tabla intermedia que contiene id\_venta, id\_producto, cantidad, precio\_unitario.
- **Justificación:** Relación clásica de ventas múltiples con varios productos por transacción.

#### 3. Producto (PK) → Proveedor (FK id\_proveedor)

- **Tipo:** Muchos a uno.
- **Descripción:** Cada producto es suministrado por un proveedor.
- **Justificación:** Permite rastrear la procedencia de cada producto.

#### 4. Producto (PK) → Inventario (FK id\_producto)

- **Tipo:** Uno a uno.
- **Descripción:** Cada producto tiene una entrada específica en inventario.
- **Justificación:** Permite registrar la cantidad disponible, ubicación, alertas, etc.

#### 5. Carrito → Detalle\_Carrito

- **Tipo:** Uno a muchos.
- **Descripción:** Un carrito se compone de varios productos antes de ser vendido.
- **Justificación:** Facilita la selección previa a la venta y la modificación del pedido.

#### 6. Carrito (PK) → Venta (FK id\_carrito)

- **Tipo:** Uno a uno.
- **Descripción:** Cada carrito confirmado se convierte en una venta.
- **Justificación:** Relaciona la selección temporal del cliente con la venta concreta.

#### 7. Cliente (PK) → Devolucion (FK id\_cliente)

- **Tipo:** Uno a muchos.
- **Descripción:** Relaciona al cliente con sus devoluciones registradas.
- **Justificación:** Control de historial de devoluciones.

#### 8. Devolucion (FK id\_venta) → Venta (PK)

- **Tipo:** Muchos a uno.
- **Descripción:** Cada devolución se vincula con una venta específica.
- **Justificación:** Garantiza que solo productos vendidos puedan ser devueltos.

#### 9. Usuario (PK) → Venta (FK id\_usuario)

- **Tipo:** Uno a muchos.
- **Descripción:** Un usuario del sistema puede registrar muchas ventas.
- **Justificación:** Rastreabilidad y control administrativo.

## 11.- Interfaces de usuarios

### Gestión de Categorías (GestionCategorias.java)

#### Funcionalidad

Esta interfaz permite a los usuarios administradores gestionar las categorías de productos del sistema. Las categorías son esenciales para organizar y clasificar los productos en el inventario.

#### Componentes principales:

- **Tabla de categorías:** Muestra todas las categorías existentes con sus nombres y descripciones.
- **Botón "Agregar":** Permite crear nuevas categorías ingresando nombre y descripción.
- **Botón "Eliminar":** Elimina categorías seleccionadas de la tabla.

#### Flujo de operación:

1. El usuario accede a la interfaz desde el menú principal.
2. El sistema carga y muestra todas las categorías existentes en una tabla.
3. Para agregar:
  - El usuario hace clic en "Agregar".
  - Aparece un diálogo para ingresar nombre y descripción.
  - Al confirmar, la categoría se guarda y la tabla se actualiza.
4. Para eliminar:
  - El usuario selecciona una categoría de la tabla.
  - Hace clic en "Eliminar".
  - El sistema confirma y remueve la categoría.

#### Validaciones:

- No se permiten nombres de categoría vacíos.
- Se verifica que la categoría no exista previamente al agregar.



## Gestión de Productos (agregarinventario.java)

### Funcionalidad

Interfaz completa para agregar o editar productos en el inventario del sistema. Permite definir todas las características de un producto, incluyendo información básica, stock y precios.

### Componentes principales:

- **Pestañas organizadas:** Divide la información en "Información Básica" y "Stock y Precios".
- **Sección de imagen:** Permite asociar una imagen al producto.
- **Campos de información:** Nombre, descripción, categoría, proveedor, estado, unidad de medida.
- **Controles de stock:** Cantidad disponible, stock mínimo y máximo.
- **Datos financieros:** Precios de compra/venta, IVA, descuentos.
- **Botones de acción:** Guardar y Cancelar.

### Flujo de operación:

1. El usuario accede desde el menú de inventario.
2. Completa la información básica en la primera pestaña:
  - Selecciona o carga imagen del producto.
  - Ingresa nombre, descripción y categoría.
  - Asocia proveedor y define unidad de medida.
3. En la pestaña de Stock y Precios:
  - Establece niveles de inventario.
  - Define precios y condiciones fiscales.
4. Confirma con "Guardar" o cancela la operación.

### Validaciones:

- Nombre del producto obligatorio.
- Stock mínimo no puede ser mayor al máximo.
- Precio de venta debe ser mayor al de compra.
- Proveedor y categoría deben ser seleccionados.

## Gestión de Usuarios (gestionUsuario.java)

### Funcionalidad

Interfaz exclusiva para administradores que permite gestionar los usuarios del sistema, incluyendo creación, modificación de roles y eliminación de cuentas.

### Componentes principales:

- **Tabla de usuarios:** Muestra username, rol y estado.
- **Botón "Agregar":** Abre formulario para nuevo usuario.
- **Botón "Editar":** Permite cambiar el rol de usuarios existentes.
- **Botón "Eliminar":** Remueve usuarios seleccionados (excepto admin).
- **Menú de navegación:** Acceso a otras áreas del sistema.

### Flujo de operación:

1. El administrador accede desde el menú principal.
2. El sistema carga y muestra todos los usuarios registrados.
3. Para agregar:
  - Clic en "Agregar" abre formulario.
  - Se ingresan username, password y rol.
  - El sistema crea la cuenta y actualiza la tabla.
4. Para editar:
  - Seleccionar usuario y clic en "Editar".
  - Seleccionar nuevo rol en diálogo emergente.
5. Para eliminar:
  - Seleccionar usuario (no admin).
  - Confirmar eliminación.

### Validaciones:

- Username y password no pueden estar vacíos.
- No se puede eliminar la cuenta "admin".
- Usernames deben ser únicos.

## Gestión de Clientes (clientes.java)

### Funcionalidad

Interfaz para administrar la base de clientes del negocio, permitiendo registrar información de contacto, historial de compras y sistema de puntos.

### Componentes principales:

- **Tabla de clientes:** Muestra ID, teléfono, nombre, última compra y puntos.
- **Botón "Agregar":** Abre formulario para nuevo cliente.
- **Botón "Editar":** Modifica información de cliente existente.
- **Botón "Eliminar":** Remueve clientes del sistema.
- **Búsqueda y filtros:** Para localizar clientes específicos.

### Flujo de operación:

1. El usuario accede desde el menú principal.
2. El sistema carga y muestra todos los clientes registrados.
3. Para agregar:
  - Clic en "Agregar" abre formulario.
  - Se ingresan teléfono, nombre y fecha de última compra.
  - El sistema asigna automáticamente ID y puntos iniciales (0).
4. Para editar:
  - Seleccionar cliente de la tabla.
  - Clic en "Editar" abre formulario con datos actuales.
  - Modificar información necesaria.
5. Para eliminar:
  - Seleccionar cliente.
  - Confirmar eliminación.

### Validaciones:

- Teléfono es campo obligatorio y debe contener solo números.
- Nombre no puede estar vacío.
- Formato de fecha válido para última compra.

## Diseño y Experiencia de Usuario

Todas las interfaces comparten un diseño consistente que incluye:

- **Barra superior:** Con logo de la aplicación e información del usuario logueado.
- **Menú horizontal:** Navegación entre módulos con botones destacados según ubicación actual.
- **Feedback visual:** Colores, bordes redondeados y efectos hover para mejor usabilidad.
- **Validaciones en tiempo real:** Para prevenir errores en la entrada de datos.
- **Mensajes de confirmación/error:** Diálogos modales para operaciones críticas.

Las interfaces están diseñadas para ser intuitivas, con flujos de trabajo lógicos que guían al usuario a través de cada proceso. Se prioriza la eficiencia en operaciones frecuentes y se incluyen controles para prevenir acciones no deseadas.

Inventario (inventario.java)

El sistema de inventario está diseñado para gestionar productos, proveedores, movimientos de stock y reportes de manera eficiente. La interfaz de usuario (UI) está organizada para facilitar la navegación y el acceso rápido a las funcionalidades clave, adaptándose a los roles de los usuarios (administradores, empleados, etc.). A continuación, se describe la funcionalidad principal de las interfaces:

### Inventario (inventario.java)

#### Funcionalidad:

- Muestra una vista general de los productos en el inventario, con opciones para filtrar por categorías, buscar productos y gestionar movimientos de stock.
- Permite visualizar alertas de stock bajo o excesivo.
- Los usuarios pueden acceder a funciones avanzadas como búsqueda, reportes y gestión de devoluciones.

#### Componentes:

- **Barra superior:** Logo de la empresa, nombre de usuario y opción para cambiar de usuario.
- **Menú horizontal:** Accesos directos a módulos como Productos, Reportes, Inventario, Clientes, Proveedores y Usuarios.
- **Panel de categorías:** Filtros para categorías de productos (izquierda).
- **Tarjetas de productos:** Muestran imagen, nombre, precio, stock y estado de cada producto (centro).

- **Barra inferior:** Botones para agregar productos, búsqueda avanzada y generación de reportes.

#### **Flujo:**

1. El usuario inicia sesión y accede al módulo de inventario.
2. Desde el menú, puede navegar a otras secciones o usar los filtros para encontrar productos.
3. Al hacer clic en un producto, se abre un diálogo con detalles completos y opciones de edición o eliminación.
4. Los botones inferiores permiten agregar productos, realizar búsquedas avanzadas o generar reportes.

#### **Diálogo de Detalles de Producto**

##### **Funcionalidad:**

- Muestra información detallada de un producto seleccionado, incluyendo precios, stock, proveedor y estado.
- Permite editar o eliminar el producto, así como contactar al proveedor si está asociado.

##### **Componentes:**

- **Pestañas:** Organizan la información en secciones (Información General, Stock y Precios, Proveedor).
- **Imagen del producto:** Muestra la imagen o un placeholder si no está disponible.
- **Botones de acción:** Editar, Eliminar, Contactar Proveedor y Cerrar.

#### **Flujo:**

1. El usuario selecciona un producto de la lista principal.
2. El sistema muestra el diálogo con pestañas para explorar los detalles.
3. El usuario puede editar el producto, eliminarlo o contactar al proveedor si es necesario.

#### **Búsqueda Avanzada**

##### **Funcionalidad:**

- Permite filtrar productos por múltiples criterios: nombre, categoría, proveedor, rango de precios, stock, fechas y estados.
- Incluye opciones para movimientos de stock (entradas, salidas, ajustes) y gestión de devoluciones.

### Componentes:

- **Pestañas:** Búsqueda, Movimientos, Historial y Devoluciones.
- **Campos de filtro:** Texto, listas desplegables, selectores de fecha y rangos numéricos.
- **Botones:** Buscar, Limpiar y Cerrar.

### Flujo:

1. El usuario abre el diálogo de búsqueda avanzada desde el botón inferior.
2. Selecciona los criterios de búsqueda y aplica los filtros.
3. El sistema muestra los resultados en la pantalla principal.

### Gestión de Movimientos de Stock

#### Funcionalidad:

- Registra entradas (compras), salidas (ventas) y ajustes (mermas, daños) de productos.
- Valida las cantidades para evitar errores (ej.: no permitir salidas mayores al stock disponible).

#### Componentes:

- **Formularios:** Campos para ID del producto, cantidad, motivo, fecha y observaciones.
- **Validación:** Mensajes de error si los datos son incorrectos.
- **Botones:** Registrar y Cancelar.

#### Flujo:

1. El usuario selecciona "Movimientos" en la búsqueda avanzada.
2. Completa el formulario con los datos del movimiento (ej.: cantidad de productos vendidos).
3. El sistema valida y registra el movimiento, actualizando el stock automáticamente.

### Devoluciones

#### Funcionalidad:

- Gestiona devoluciones de clientes (productos defectuosos) o a proveedores (errores en pedidos).
- Genera historiales y reportes de devoluciones para análisis.

#### Componentes:

- **Pestañas:** Devolución de Cliente, Devolución a Proveedor, Historial.
- **Formularios:** Motivo, cantidad, ID de transacción original.
- **Tabla de historial:** Muestra devoluciones registradas con filtros por tipo y estado.

#### Flujo:

1. El usuario selecciona "Devoluciones" en la búsqueda avanzada.
2. Completa el formulario según el tipo de devolución (cliente o proveedor).
3. El sistema registra la devolución y actualiza el stock o genera una notificación al proveedor.

#### Reportes

##### Funcionalidad:

- Genera reportes en PDF, Excel, CSV o HTML con datos de inventario, movimientos y devoluciones.
- Permite filtrar por fechas, categorías o estados.

##### Componentes:

- **Opciones de formato:** Selección entre PDF, Excel, etc.
- **Filtros:** Fechas, tipo de reporte (ej.: stock bajo).
- **Botones:** Generar y Cancelar.

#### Flujo:

1. El usuario selecciona "Generar Reporte" desde el panel inferior.
2. Elige los filtros y el formato del reporte.
3. El sistema genera el archivo y lo abre automáticamente para su visualización o descarga.

#### Resumen del Flujo General

1. **Acceso:** El usuario inicia sesión y navega al módulo de inventario.
2. **Gestión:**
  - Agrega, edita o elimina productos.
  - Filtra y busca productos según necesidades.
3. **Movimientos:** Registra entradas, salidas y ajustes de stock.
4. **Devoluciones:** Gestiona reclamos de clientes o proveedores.

5. **Reportes:** Genera documentos para análisis o auditorías.

La interfaz prioriza la usabilidad, con diseños intuitivos, validaciones en tiempo real y retroalimentación clara para el usuario (ej.: alertas de stock bajo en rojo). Los administradores tienen acceso completo, mientras que otros roles pueden tener permisos restringidos.

### **Pantalla de Inicio de Sesión (Login.java)**

#### **Funcionalidad:**

- Permite a los usuarios autenticarse en el sistema ingresando su nombre de usuario y contraseña.
- Valida las credenciales contra la base de datos mediante UsuarioDAO.
- Redirige al menú principal según el rol del usuario (administrador, empleado, etc.).
- Incluye una función oculta para registro rápido de usuarios temporales (accesible haciendo clic en una zona específica del logo).

#### **Componentes:**

- **Campos de texto:**
  - usernameField: Para ingresar el nombre de usuario.
  - passwordField: Para la contraseña (con caracteres ocultos).
- **Botón de acceso:**
  - loginButton: Valida los datos y abre el menú principal si son correctos.
- **Diseño:**
  - Fondo con gradiente suave y círculos decorativos en colores pastel.
  - Logo de la empresa (clickeable para función oculta).

#### **Flujo:**

1. El usuario ingresa sus credenciales.
2. El sistema verifica con UsuarioDAO.autenticarUsuario().
3. Si son válidas:
  - Cierra la ventana de login.
  - Abre el menuprincipal con los permisos correspondientes al rol.
4. Si son inválidas: Muestra un mensaje de error.



## Menú Principal (menuprincipal.java)

### Funcionalidad:

- Es el hub central del sistema después del login.
- Muestra opciones según el rol del usuario (ej.: administradores ven "Gestión de Usuarios", empleados no).
- Permite navegar a módulos como:
  - **Gestión de Usuarios** (solo admin).
  - **Cientes, Ventas, Inventario, Reportes, Proveedores.**
- Incluye un botón para cerrar sesión y volver al login.

### Componentes:

- **Botones principales:**
  - Diseñados con bordes redondeados y efectos hover.
  - Se adaptan dinámicamente según el rol del usuario (configurarVisibilidadBotones()).
- **Cabecera:**
  - Muestra el logo, nombre de la empresa y usuario actual (lblUsuario).
- **Diseño:**
  - Fondo con gradiente y círculos decorativos (similar al login).

### Flujo:

1. Tras un login exitoso, se carga el menú principal con el nombre y rol del usuario.
2. El usuario selecciona un módulo (ej.: "Inventario").
3. El sistema abre la interfaz correspondiente (controlado por ventanas.java).
4. Al cerrar sesión, vuelve a la pantalla de login.

### Diferencias Clave Entre las Interfaces

Aspecto	Login	Menú Principal
<b>Propósito</b>	Autenticación de usuarios.	Navegación a módulos del sistema.

Aspecto	Login	Menú Principal
<b>Interacción</b>	Campos de texto + botón de acceso.	Botones para cada funcionalidad.
<b>Dinamismo</b>	Estático (solo login).	Adaptable (oculta opciones por rol).
<b>Función oculta</b>	Registro rápido de usuarios.	No aplica.

### Integración con el Sistema

- **Controlador** (ventanas.java):
  - Gestiona la transición entre ventanas (ej.: al hacer clic en "Inventario", abre inventario.java).
  - Ajusta permisos según el rol del usuario pasado desde Login.
- **Modelo** (UsuarioDAO):
  - Verifica credenciales en la base de datos.
  - Registra usuarios temporales (función oculta).

### Interfaz de Gestión de Clientes (movimientocliente.java)

#### Funcionalidad:

- Permite registrar, editar y eliminar información de clientes en el sistema.
- Facilita la gestión de datos clave como identificación, nombre, contacto y dirección.
- Valida los datos ingresados para asegurar su integridad.

#### Componentes principales:

- Formulario con campos para ID, nombre completo, teléfono, fecha de registro, dirección y correo electrónico.
- Botones de acción: Agregar, Editar y Eliminar.
- Validaciones automáticas para formato de email y números de teléfono.

#### Flujo de trabajo:

1. El usuario ingresa los datos del cliente en el formulario.
2. El sistema valida que los campos obligatorios estén completos.
3. Al guardar, se verifica que no haya errores en los formatos.
4. Los datos se almacenan en la base de datos y pueden ser modificados posteriormente.

### **Interfaz de Ventas (producto.java)**

#### **Funcionalidad:**

- Sistema completo para procesar ventas con interfaz táctil optimizada.
- Permite buscar productos por categoría o nombre.
- Gestiona carrito de compras con cantidades variables.
- Calcula totales, impuestos y cambio automáticamente.
- Registra información del cliente para programas de fidelización.

#### **Componentes principales:**

- Catálogo visual de productos organizado por categorías.
- Panel de carrito con lista de productos seleccionados.
- Calculadora integrada para ingresar montos de pago.
- Sección para registrar teléfono del cliente y consultar puntos.
- Opciones de pago (efectivo o tarjeta) con cálculo automático de cambio.

#### **Flujo de venta:**

1. El cajero selecciona productos del catálogo.
2. El sistema muestra el carrito con cantidades y subtotales.
3. Se registra el método de pago y el monto recibido.
4. Se genera el ticket de venta y actualiza el inventario.

### **Interfaz de Proveedores (proveedores.java)**

#### **Funcionalidad:**

- Gestiona el registro de proveedores y su información de contacto.
- Permite registrar visitas y productos suministrados.
- Facilita la búsqueda y filtrado de proveedores.

### **Componentes principales:**

- Tabla con lista completa de proveedores.
- Formulario para agregar/editar datos de proveedores.
- Selector de fecha para registrar última visita.
- Filtros para buscar proveedores específicos.

### **Flujo de trabajo:**

1. El usuario visualiza la lista de proveedores registrados.
2. Puede agregar nuevos o editar existentes mediante formulario.
3. El sistema valida y almacena la información.
4. Las búsquedas permiten localizar proveedores rápidamente.

### **Interfaz de Reimpresión (ReimprimirDialog.java)**

#### **Funcionalidad:**

- Permite reimprimir tickets de venta y diversos reportes.
- Ofrece búsqueda avanzada de documentos por múltiples criterios.
- Exporta resultados de búsqueda a formato CSV.

### **Componentes principales:**

- Selector de tipo de documento (tickets o reportes).
- Campo para ingresar ID de documento.
- Panel de búsqueda avanzada con filtros por fecha y método de pago.
- Tabla de resultados con opción de selección.

### **Flujo de trabajo:**

1. El usuario selecciona el tipo de documento a reimprimir.
2. Puede ingresar directamente un ID o realizar búsqueda avanzada.
3. El sistema muestra los resultados coincidentes.
4. Al seleccionar un documento, se genera una nueva impresión.

### **Panel de Reportes de Clientes (ReporteClientePanel.java)**

#### **Funcionalidad:**

- Genera reportes analíticos sobre la base de clientes.

- Muestra estadísticas visuales mediante gráficos.
- Permite filtrar clientes por diferentes criterios.
- Exporta reportes a múltiples formatos.

#### **Componentes principales:**

- Tabla con datos detallados de clientes.
- Gráficos de registros por mes y distribución de puntos.
- Filtros para segmentar la información.
- Opciones de exportación (PDF, Excel, HTML).

#### **Flujo de generación de reportes:**

1. El usuario aplica filtros según necesidades.
2. El sistema actualiza tablas y gráficos en tiempo real.
3. El usuario puede exportar los resultados.
4. Los gráficos permiten análisis visual rápido.

#### **Panel de Reportes de Inventario (ReporteInventarioPanel.java)**

##### **Funcionalidad:**

- Proporciona análisis detallado del estado del inventario.
- Identifica productos con stock bajo o agotado.
- Muestra distribución de productos por categoría.
- Genera reportes exportables para gestión.

##### **Componentes principales:**

- Tabla con niveles de stock y datos de productos.
- Gráficos de niveles de stock promedio y distribución por categoría.
- Indicadores rápidos de estado del inventario.
- Filtros y opciones de exportación.

##### **Flujo de trabajo:**

1. El usuario visualiza el estado general del inventario.
2. Los gráficos destacan áreas que requieren atención.
3. Los filtros permiten focalizar el análisis.

4. Los reportes pueden exportarse para compartir.

### Módulo de Reportes (reportes.java)

El módulo de reportes del sistema está diseñado para proporcionar a los usuarios (administradores y gerentes) herramientas de análisis y visualización de datos relacionados con ventas, inventario, clientes y proveedores. Las interfaces están organizadas para facilitar la generación, filtrado y exportación de reportes, con un enfoque en la usabilidad y la presentación clara de la información.

#### Descripción de Funcionalidad

##### Pantalla Principal de Reportes

##### Funcionalidad:

- Es el punto de entrada al módulo de reportes, donde los usuarios pueden seleccionar el tipo de reporte que desean generar (ventas, inventario, clientes o proveedores).
- Cada opción está representada por una tarjeta visual con iconos descriptivos para facilitar la navegación.

##### Componentes:

- Tarjetas de acceso rápido:
  - **Ventas:** Genera reportes detallados de ventas por período.
  - **Inventario:** Muestra el estado actual del inventario y alertas.
  - **Clientes:** Analiza el comportamiento y fidelidad de los clientes.
  - **Proveedores:** Evalúa el desempeño y la relación con los proveedores.
  - **Reimpresión:** Permite reimprimir tickets de venta o reportes.
- Efectos visuales (hover) para mejorar la interacción.

##### Flujo:

- El usuario selecciona una tarjeta para acceder al reporte correspondiente.
- El sistema redirige a la interfaz específica del reporte seleccionado.

##### Reporte de Ventas

##### Funcionalidad:

- Permite filtrar ventas por período (hoy, semana, mes o rango personalizado).
- Muestra un resumen gráfico de las ventas, métodos de pago y productos más vendidos.

- Ofrece opciones para exportar el reporte en formatos como PDF, Excel o HTML.

#### **Componentes:**

- **Filtros:**
  - Selección de período (combo box).
  - Selectores de fecha para rangos personalizados.
  - Botón "Filtrar" para aplicar los criterios.
- **Gráficos:**
  - Resumen de ventas (barras).
  - Métodos de pago (torta).
  - Productos más vendidos (barras horizontales).
- **Detalle de ventas:**
  - Área de texto que lista las ventas con información como ID, fecha, producto, cantidad y total.
- **Estadísticas rápidas:**
  - Total, de ventas, ventas del día, productos vendidos y ticket promedio.

#### **Flujo:**

1. El usuario selecciona un período o define un rango de fechas.
2. El sistema filtra los datos y actualiza los gráficos y el detalle.
3. El usuario puede exportar el reporte o imprimirlo.

#### **Reporte de Proveedores**

##### **Funcionalidad:**

- Muestra información detallada de los proveedores, como nombre, teléfono, dirección y productos suministrados.
- Incluye gráficos para visualizar visitas por mes y distribución por producto.
- Permite filtrar proveedores (todos, con visita reciente, sin visita reciente, por producto).

#### **Componentes:**

- **Tabla de proveedores:**

- Columnas: ID, nombre, teléfono, dirección, producto suministrado y última visita.
- **Gráficos:**
  - Visitas por mes (barras).
  - Distribución por producto (torta).
- **Filtros:**
  - Combo box para seleccionar el tipo de filtro.
  - Botón "Filtrar" para aplicar.
- **Estadísticas rápidas:**
  - Total, de proveedores, visitados este mes y última visita registrada.

#### **Flujo:**

1. El usuario selecciona un filtro (opcional).
2. El sistema actualiza la tabla y los gráficos con los datos filtrados.
3. El usuario puede exportar el reporte en diferentes formatos.

#### **Reporte de Clientes e Inventario**

##### **Funcionalidad:**

- **Clientes:** Analiza datos como frecuencia de compra, preferencias y fidelidad.
- **Inventario:** Muestra niveles de stock, productos agotados y alertas de reabastecimiento.

##### **Componentes (similares a los anteriores):**

- Tablas con información detallada.
- Gráficos para visualizar tendencias.
- Filtros y opciones de exportación.

##### **Flujo General del Módulo de Reportes**

1. **Inicio:** El usuario accede al módulo desde el menú principal.
2. **Selección:** Elige el tipo de reporte (ventas, proveedores, etc.).
3. **Filtrado:** Aplica filtros para personalizar la información.
4. **Visualización:** Los datos se muestran en tablas y gráficos interactivos.
5. **Exportación:** El usuario puede exportar el reporte o imprimirlo.



### Consideraciones de Diseño

- **Interfaz intuitiva:** Uso de tarjetas, gráficos y filtros para facilitar la navegación.
- **Responsividad:** Adaptable a diferentes tamaños de pantalla.
- **Feedback visual:** Efectos hover y mensajes de error claros.
- **Accesibilidad:** Diseño limpio y jerarquía visual para destacar información clave.

Este módulo está diseñado para ayudar a los usuarios a tomar decisiones basadas en datos, con herramientas flexibles y visualmente atractivas.

## 12.- Flujo de las interfaces de Usuario por cada Requerimiento funcional

A continuación, se describe el flujo de las interfaces de las clases para cada requerimiento funcional del sistema, incluyendo las interacciones clave y cómo se integran estas clases.

### 1. Autenticación de Usuario

#### Requerimiento:

*Permitir a los usuarios iniciar sesión con credenciales válidas y redirigirlos según su rol.*

#### Flujo:

##### 1. Pantalla de Login (Login.java)

- El usuario ingresa su nombre y contraseña.
- El botón "**Acceder**" llama a UsuarioDAO.autenticarUsuario().
- Si las credenciales son válidas:
  - Se obtiene el objeto Usuario con su rol.
  - Se cierra el Login y se abre el menuprincipal (pasando el usuario como parámetro).
- Si son inválidas: Muestra un mensaje de error.

##### 2. Menú Principal (menuprincipal.java)

- Muestra un mensaje de bienvenida con el nombre y rol del usuario (setBienvenida()).
- Filtra las opciones visibles según el rol (ej.: oculta "Gestión de Usuarios" para no administradores).

### Clases Involucradas:

- Login → Valida credenciales.
- UsuarioDAO → Consulta la base de datos.
- menuprincipal → Muestra interfaz adaptada al rol.

## 2. Gestión de Inventario

### Requerimiento:

*Permitir a los usuarios autorizados ver, agregar, editar o eliminar productos del inventario.*

### Flujo:

1. **Desde el Menú Principal** (menuprincipal.java)
  - El usuario hace clic en "**Inventario**".
  - El controlador ventanas abre la clase inventario.java, pasando el objeto Usuario.
2. **Interfaz de Inventario** (inventario.java)
  - **Visualización:**
    - Muestra tarjetas de productos con imagen, nombre, precio y stock.
    - Filtros por categoría (panel izquierdo) y búsqueda rápida.
  - **Gestión:**
    - **Agregar:** Botón "AGREGAR" abre un formulario (agregarinventario.java).
    - **Editar/Eliminar:** Al hacer clic en un producto, se abre un diálogo con pestañas de detalles y botones de acción.
  - **Alertas:**
    - Productos con stock bajo se marcan en rojo.

### Clases Involucradas:

- menuprincipal → Navegación.
- inventario → Lógica y visualización del inventario.
- ControladorInventario → Gestiona operaciones CRUD.

### 3. Generación de Reportes

#### Requerimiento:

*Permitir a administradores generar reportes de inventario en PDF, Excel, etc.*

#### Flujo:

1. **Desde el Menú Principal** (menuprincipal.java)
  - Solo visible para administradores (configurarVisibilidadBotones()).
  - El usuario hace clic en "**Reportes**".
2. **Interfaz de Reportes** (en inventario.java)
  - Se abre un diálogo con pestañas para seleccionar:
    - **Filtros:** Por fechas, categorías, stock, etc.
    - **Formato:** PDF, Excel, CSV o HTML.
  - Al confirmar, se genera el archivo y se descarga automáticamente.

#### Clases Involucradas:

- menuprincipal → Control de acceso.
- inventario → Lógica de reportes (usa Apache POI para Excel, iText para PDF).

### 4. Gestión de Usuarios (Admin)

#### Requerimiento:

*Solo los administradores pueden crear o modificar usuarios.*

#### Flujo:

1. **Desde el Menú Principal** (menuprincipal.java)
  - El botón "**Gestión de Usuarios**" solo aparece para roles administrativos.
  - Al hacer clic, abre la clase gestionUsuario.java (no mostrada en el código, pero vinculada por ventanas).

#### Clases Involucradas:

- menuprincipal → Control de visibilidad.
- ventanas → Navegación a gestionUsuario.

## 5. Cierre de Sesión

### Requerimiento:

*Permitir a los usuarios cerrar sesión y volver al login.*

### Flujo:

#### 1. Botón en Menú Principal (menuprincipal.java)

- Al hacer clic en "**Cerrar Sesión**":
  - Cierra la ventana actual (menuprincipal).
  - Abre una nueva instancia de Login.java.

### Clases Involucradas:

- menuprincipal → Ejecuta dispose() y abre Login.

Gestión de Categorías (GestionCategorias.java)

**Requerimiento Funcional:** Administrar las categorías de productos del sistema de inventario.

### Flujo de Interacción:

#### 1. Acceso:

- El usuario administrador selecciona "Gestión de Categorías" desde el menú de Inventario.
- El sistema carga la ventana modal mostrando la tabla de categorías existentes.

#### 2. Agregar Categoría:

- Usuario hace clic en "Agregar".
- Sistema muestra diálogo con campos: Nombre y Descripción.
- Usuario completa los datos y confirma.
- Sistema valida que el nombre no exista y no esté vacío.
- Si es válido, agrega la categoría y actualiza la tabla.
- Si hay error, muestra mensaje específico.

#### 3. Eliminar Categoría:

- Usuario selecciona una categoría de la tabla.
- Hace clic en "Eliminar".

- Sistema confirma la acción (sin diálogo explícito en el código actual).
- Elimina la categoría y actualiza la tabla visualmente.

#### **Validaciones Clave:**

- Nombre de categoría obligatorio
- Unicidad del nombre de categoría
- Selección requerida para eliminar

### **6. Gestión de Productos (agregarinventario.java)**

**Requerimiento Funcional:** Registrar y actualizar productos en el inventario con toda su información asociada.

#### **Flujo de Interacción:**

##### **1. Acceso:**

- Usuario selecciona "Agregar Producto" desde el módulo de Inventario.
- Sistema abre ventana modal con pestañas organizadas.

##### **2. Completar Información Básica (Pestaña 1):**

- Usuario carga imagen del producto (opcional).
- Completa nombre, descripción, selecciona categoría y proveedor.
- Define estado y unidad de medida.
- Puede agregar proveedor rápido con botón "+" si no existe.

##### **3. Definir Stock y Precios (Pestaña 2):**

- Establece cantidades: disponible, mínimo y máximo.
- Ingresa precios de compra y venta (con formato monetario).
- Marca si aplica IVA y porcentaje de descuento.

##### **4. Confirmación:**

- Usuario hace clic en "Guardar".
- Sistema valida:
  - Nombre obligatorio
  - Stock mínimo  $\leq$  máximo

- Precio venta > precio compra
- Proveedor válido seleccionado
- Si todo es válido, guarda producto y cierra ventana.
- Si hay errores, muestra mensajes específicos.

#### Casos Especiales:

- **Edición:** Si se abre en modo edición, carga automáticamente los datos del producto seleccionado.
- **Imágenes:** Permite seleccionar y guardar imágenes en carpeta específica.

#### 3. Gestión de Usuarios (gestionUsuario.java)

**Requerimiento Funcional:** Administrar cuentas de usuario y permisos del sistema.

#### Flujo de Interacción:

##### 1. Acceso:

- Administrador selecciona "Gestión de Usuarios" desde el menú.
- Sistema carga ventana principal con tabla de usuarios.

##### 2. Agregar Usuario:

- Clic en "Agregar" abre diálogo modal.
- Ingresa username, password y selecciona rol (Admin/Trab).
- Confirma con "Guardar".
- Sistema valida:
  - Campos no vacíos
  - Username único
- Agrega usuario y actualiza tabla.

##### 3. Editar Rol:

- Selecciona usuario en tabla (no admin).
- Clic en "Editar".
- Sistema muestra opción para cambiar rol.
- Selecciona nuevo rol y confirma.
- Actualiza inmediatamente en tabla.

#### 4. Eliminar Usuario:

- Selecciona usuario (no admin).
- Clic en "Eliminar".
- Sistema pide confirmación.
- Si confirma, elimina y actualiza tabla.

#### Restricciones:

- Usuario "admin" no se puede eliminar ni editar.
- Solo administradores acceden a esta interfaz.

### 7. Gestión de Clientes (clientes.java)

**Requerimiento Funcional:** Mantener registro de clientes con su información de contacto y historial.

#### Flujo de Interacción:

##### 1. Acceso:

- Usuario selecciona "Clientes" desde el menú principal.
- Sistema carga ventana con tabla de clientes y botones de acción.

##### 2. Agregar Cliente:

- Clic en "Agregar Cliente" abre formulario.
- Completa teléfono, nombre, fecha última compra (automática).
- Puntos se asignan automáticamente (0 inicial).
- Confirma con "Guardar".
- Sistema valida:
  - Teléfono no vacío y numérico
  - Nombre no vacío
- Agrega cliente y actualiza tabla.

##### 3. Editar Cliente:

- Selecciona cliente en tabla.
- Clic en "Editar" abre formulario con datos actuales.

- Modifica información necesaria.
- Confirma cambios.
- Sistema valida datos y actualiza registro.

#### **4. Eliminar Cliente:**

- Selecciona cliente en tabla.
- Clic en "Eliminar".
- Sistema pide confirmación.
- Si confirma, elimina y actualiza tabla.

#### **Características Especiales:**

- Selector de fecha gráfico para última compra.
- Visualización formateada de fechas en tabla.
- Búsqueda y filtrado de clientes.

### **8. Gestión de Clientes (movimientocliente.java)**

#### **Requerimiento Funcional:**

- Registrar, editar y eliminar información de clientes.
- Validar datos de contacto (teléfono, email).
- Consultar historial de compras (si se integra con ventas).

#### **Flujo de la Interfaz:**

##### **1. Acceso:**

- El usuario (administrador o cajero) selecciona "Cliente" en el menú principal.
- Se abre el formulario de gestión de clientes (movimientocliente).

##### **2. Registro/Edición:**

- El usuario ingresa datos como ID, nombre, teléfono, dirección y correo.
- El sistema valida que los campos obligatorios estén completos y el formato sea correcto (ej. teléfono solo números, email válido).

##### **3. Acciones:**

- **Guardar:** Confirma el registro o modificación del cliente.



- **Eliminar:** Solicita confirmación antes de borrar un cliente.
- **Limpiar:** Reinicia el formulario para un nuevo registro.

#### 4. **Salida:**

- Al cerrar, redirige al menú principal.

## 9. Procesamiento de Ventas (producto.java)

### Requerimiento Funcional:

- Registrar ventas con múltiples productos.
- Calcular totales, impuestos y cambio.
- Asociar cliente a la venta para programas de fidelización.

### Flujo de la Interfaz:

#### 1. **Acceso:**

- El cajero selecciona "Productos" (o "Ventas") en el menú.
- Se abre la interfaz de ventas (producto).

#### 2. **Selección de Productos:**

- El cajero navega por categorías o busca productos por nombre.
- Al hacer clic en un producto, se agrega al carrito con la cantidad especificada.

#### 3. **Gestión del Carrito:**

- Muestra lista de productos, cantidades, precios y subtotal.
- Permite eliminar productos o ajustar cantidades.

#### 4. **Pago:**

- Selección de método de pago (efectivo/tarjeta).
- Si es efectivo, calcula automáticamente el cambio.
- Si el cliente está registrado, se captura su teléfono para acumular puntos.

#### 5. **Finalización:**

- Al confirmar, se genera un ticket y se actualiza el inventario.
- Opción de reimprimir ticket desde ReimprimirDialog.

## 10. Gestión de Proveedores (proveedores.java)

### Requerimiento Funcional:

- Mantener registro de proveedores (nombre, contacto, productos suministrados).
- Registrar visitas y actualizar información.

### Flujo de la Interfaz:

#### 1. Acceso:

- El administrador selecciona "Proveedores" en el menú.
- Se despliega una tabla con todos los proveedores registrados.

#### 2. Acciones:

- **Agregar:** Abre un formulario para ingresar datos del proveedor.
- **Editar:** Permite modificar información existente.
- **Eliminar:** Solicita confirmación antes de borrar.

#### 3. Búsqueda/Filtrado:

- Permite filtrar por nombre, producto suministrado o fecha de visita.

#### 4. Salida:

- Al cerrar, regresa al menú principal.

## 11. Reimpresión de Documentos (ReimprimirDialog.java)

### Requerimiento Funcional:

- Reimprimir tickets de venta perdidos.
- Generar reportes históricos (ventas, inventario).

### Flujo de la Interfaz:

#### 1. Acceso:

- Desde el menú de "Reportes" o una opción directa en ventas.
- Se abre el diálogo ReimprimirDialog.

#### 2. Búsqueda:

- El usuario selecciona el tipo de documento (ticket, reporte de ventas, etc.).

- Puede ingresar un ID específico o usar filtros avanzados (fecha, método de pago).

### 3. Resultados:

- Muestra una tabla con documentos coincidentes.
- Al seleccionar uno, se reimprime o regenera el archivo.

### 4. Exportación:

- Opción para guardar resultados en CSV.

## 12. Reporte de Clientes (ReporteClientePanel.java)

### Requerimiento Funcional:

- Analizar base de clientes (activos, inactivos, puntos acumulados).
- Generar gráficos de registro mensual y distribución de puntos.

### Flujo de la Interfaz:

#### 1. Acceso:

- El administrador selecciona "Reportes" → "Clientes".
- Se carga el panel ReporteClientePanel.

#### 2. Filtrado:

- Permite filtrar por:
  - Clientes activos/inactivos.
  - Rango de puntos.
  - Fechas de registro.

#### 3. Visualización:

- Muestra tabla con datos detallados.
- Gráficos de barras (registros por mes) y pastel (distribución de puntos).

#### 4. Exportación:

- Opciones para guardar en PDF, Excel o HTML.

### 13. Reporte de Inventario (ReporteInventarioPanel.java)

#### Requerimiento Funcional:

- Monitorear niveles de stock (bajo stock, sobre stock).
- Analizar distribución de productos por categoría.

#### Flujo de la Interfaz:

##### 1. Acceso:

- El administrador selecciona "Reportes" → "Inventario".
- Se carga el panel ReporteInventarioPanel.

##### 2. Filtrado:

- Permite filtrar por:
  - Productos con stock crítico.
  - Categorías específicas.
  - Proveedores.

##### 3. Visualización:

- Tabla con ID, nombre, stock actual, mínimo y máximo.
- Gráficos de barras (niveles de stock) y pastel (distribución por categoría).

##### 4. Exportación:

- Opciones para guardar en PDF, Excel o HTML.

### 14. Generación de Reportes de Ventas

#### Requerimiento Funcional:

Permitir al usuario filtrar, visualizar y exportar reportes de ventas por período (diario, semanal, mensual o personalizado).

#### Flujo de Interfaz:

##### 1. Pantalla Principal de Reportes

- El usuario selecciona la tarjeta "**Ventas**" desde el menú de reportes.
- La clase reportes redirige al panel ReporteVentasPanel.

##### 2. Filtrado de Datos

- En ReporteVentasPanel, el usuario selecciona un período en el combo box (comboFiltro):
  - **Opciones:** "Hoy", "Esta semana", "Este mes", "Rango personalizado".
- Si elige "Rango personalizado", se habilitan los selectores de fecha (dateChooserInicio y dateChooserFin).
- El usuario hace clic en "**Filtrar**" (btnFiltrar), lo que dispara el método filtrarReporte() del controlador.

### 3. Visualización de Resultados

- El sistema muestra:
  - **Gráfico de resumen** (barras con ventas por fecha).
  - **Gráfico de métodos de pago** (torta).
  - **Gráfico de productos más vendidos** (barras horizontales).
  - **Detalle en texto** (txtAreaVentas), con información como ID de venta, productos, cantidades y totales.
  - **Estadísticas rápidas** (ej: "Total Ventas: \$X", "Ventas Hoy: Y").

### 4. Exportación/Impresión

- El usuario selecciona un formato (comboExportar: PDF, Excel, HTML, CSV).
- Clic en "**Exportar**" (btnExportar) o "**Imprimir**" (btnImprimir).

## Generación de Reportes de Proveedores

### Requerimiento Funcional:

Visualizar y filtrar datos de proveedores, incluyendo visitas recientes y productos suministrados.

### Flujo de Interfaz:

#### 1. Pantalla Principal de Reportes

- El usuario selecciona la tarjeta "**Proveedores**" en el menú.
- La clase reportes carga el panel ReporteProveedoresPanel.

#### 2. Filtrado de Datos

- El usuario elige un filtro (comboFiltro):

- **Opciones:** "Todos", "Con Visita Reciente", "Sin Visita Reciente", "Por Producto".

- Clic en "**Filtrar**", lo que ejecuta `filtrarProveedores()` en el controlador.

### 3. Visualización de Resultados

- **Tabla de proveedores:** Muestra ID, nombre, teléfono, producto suministrado y última visita.
- **Gráficos:**
  - *Visitas por mes* (barras).
  - *Distribución por producto* (torta).
- **Estadísticas rápidas:** Total de proveedores, visitados este mes, última visita registrada.

### 4. Exportación

- Selección de formato (comboExportar: PDF, Excel, HTML).
- Clic en "**Exportar**" (`btnExportar`).

## Generación de Reportes de Clientes

### Requerimiento Funcional:

Analizar comportamiento de clientes (frecuencia de compra, preferencias).

### Flujo de Interfaz:

*(Similar a proveedores, pero enfocado en clientes)*

1. **Selección desde el menú principal** → Tarjeta "**Clientes**".
2. **Filtrado:** Por rango de fechas o segmentación (ej: "Clientes frecuentes").
3. **Visualización:**
  - Tabla con datos de clientes (nombre, compras totales, última compra).
  - Gráficos de tendencias (ej: clientes nuevos vs. recurrentes).
4. **Exportación:** Igual que en otros reportes.

## 4. Reimpresión de Tickets/Reportes

### Requerimiento Funcional:

Reimprimir tickets de ventas o reportes previamente generados.

### Flujo de Interfaz:

1. **Selección desde el menú principal** → Tarjeta **"Reimpresión"**.
2. **Búsqueda:**
  - El usuario ingresa el ID de la venta o selecciona un reporte guardado.
3. **Previsualización:** Muestra el ticket/reporte en pantalla.
4. **Acción final:**
  - **"Reimprimir"** → Envía a la impresora configurada.
  - **"Guardar como PDF"** → Exporta el documento.

---

### Navegación General

#### Flujo Común a Todas las Interfaces:

1. **Menú horizontal:** Permite cambiar entre módulos (ej: "Ventas", "Inventario", "Reportes").
2. **Botón "Regresar":** En cada panel, permite volver al menú principal de reportes.
3. **Header:** Muestra el usuario activo y opciones como "Cambiar usuario" o "Salir".

## 13.- Requerimientos No funcionales

Los requerimientos no funcionales son aquellas características del sistema que no están directamente relacionadas con las funciones específicas del software, pero que son igualmente cruciales para asegurar su desempeño, usabilidad, seguridad y mantenimiento. A continuación, se detallan los principales requerimientos no funcionales para el sistema de punto de venta.

### 13.1.- Seguridad

El sistema implementa medidas de seguridad básicas que son comunes en aplicaciones de escritorio desarrolladas en Java. A partir del análisis de clases como Login.java, Usuario.java y sus interacciones, se identifican los siguientes aspectos:

- **Autenticación de usuarios:**  
El acceso al sistema requiere autenticación mediante credenciales (nombre de usuario y contraseña). La clase Login gestiona esta autenticación y valida el ingreso contra los datos almacenados.
- **Control de acceso:**  
El sistema puede restringir funcionalidades dependiendo del tipo de usuario. Aunque no se observó un sistema de roles implementado explícitamente, la arquitectura permite su futura inclusión.
- **Integridad de datos:**  
Las clases de modelo separadas (Cliente, Producto, Venta, etc.) aseguran que cada operación se realice con objetos bien definidos, evitando errores comunes de integridad de datos.
- **Persistencia segura:**  
Aunque el proyecto no contiene una base de datos embebida o acceso explícito a una externa, se infiere que la lógica está diseñada para conectarse a una base de datos, donde se podría aplicar cifrado o uso de usuarios con permisos específicos.

Justificación: La seguridad básica es necesaria para evitar el acceso no autorizado, proteger los datos del cliente y mantener la confidencialidad de las transacciones comerciales.

### 13.2.- Usabilidad

- **Interfaz gráfica amigable (GUI):**  
El sistema utiliza Java Swing para crear interfaces visuales. Se observan pantallas como Vista\_Productos, Vista\_Clientes, y Vista\_Ventas, que organizan los datos de forma clara con botones, tablas y paneles.
- **Flujo lógico de uso:**  
Las ventanas están diseñadas para ser utilizadas de forma secuencial y coherente,



por ejemplo: se inicia sesión, luego se accede al panel principal, se elige una sección (productos, clientes, ventas), y se opera en ella.

- **Paneles diferenciados:**  
Se implementa el uso de paneles de colores (anaranjados, grises, etc.) y secciones bien delimitadas para mejorar la comprensión visual del usuario, siguiendo principios básicos de diseño centrado en el usuario.
- **Mensajes y botones claros:**  
Botones como “Guardar cambios”, “Cancelar”, “Limpiar”, etc., están correctamente etiquetados y ubicados, lo cual reduce la curva de aprendizaje del sistema.

Justificación: Una buena usabilidad reduce errores humanos, mejora la productividad de los empleados y hace más eficiente la operación diaria del negocio.

### 13.3.- Hardware

Dado que el sistema está desarrollado en Java Swing como una aplicación de escritorio, los requerimientos de hardware son relativamente bajos y accesibles:

- **Procesador:** Intel i3 o superior.
- **Memoria RAM:** Mínimo 4 GB (recomendado 8 GB si se usa simultáneamente con otros programas).
- **Disco duro:** Al menos 500 MB libres para el sistema, sin contar la base de datos externa.
- **Pantalla:** Resolución mínima de 1280x720 para que los paneles se muestren correctamente.
- **Sistema operativo:** Windows 7, 8, 10 u 11, o cualquier sistema con soporte para Java Runtime Environment (JRE).
- **Otros dispositivos:** Lector de código de barras (opcional), impresora de tickets (opcional), mouse y teclado.

Justificación: El sistema no requiere procesamiento avanzado ni gráficos intensivos, por lo tanto, puede ejecutarse en computadoras estándar del entorno comercial.