

MANUAL TÉCNICO - SAN CARLITA SHOP

San Carlita Shop es un sistema de gestion para pequeñas tiendas que permite manejar productos, clientes, pedidos y ventas de forma organizada y eficiente.

San Carlita Shop es un sistema para tiendas que te ayuda a:

- Controlar productos en stock
- Llevar la lista de clientes y sus compras
- Gestionar pedidos de forma ordenada
- Ver en tiempo real cómo va tu negocio

El sistema funciona como un asistente digital que mantiene todo en orden permite registrar cada producto con su precio y cantidad disponible. Los clientes pueden ser registrados con sus datos basicos para llevar un historial de sus compras.

Cuando un cliente realiza un pedido, el sistema sigue todo el proceso desde que se crea hasta que se entrega. Los vendedores pueden ver qué pedidos están pendientes y marcarlos como completados cuando se despachan.

El programa esta construido para ser facil de usar sin requerir conocimientos técnicos especiales. Puede funcionar en computadoras comunes.

Arquitectura del Proyecto MVC.

Controlador.

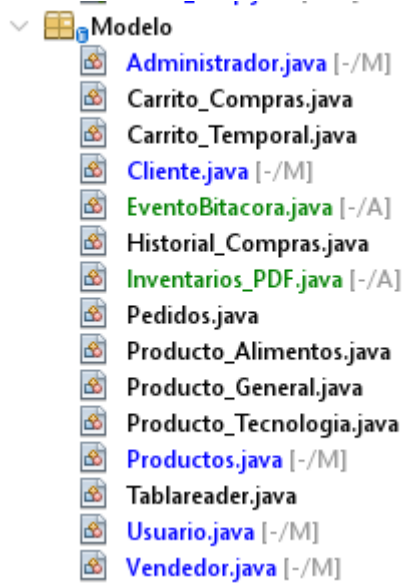


Gestiona la interacción entre Modelo y Vista

Ubicación: Paquete Controlador/

Contiene: Clases con métodos estáticos para operaciones

Característica: Orquesta el flujo de la aplicación



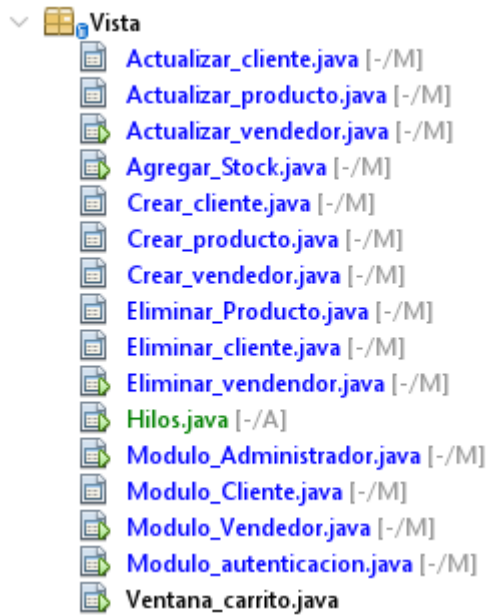
Modelo

Representa los datos y la lógica de negocio

Ubicación: Paquete Modelo/

Contiene: Entidades del sistema (Usuario, Producto, Pedido, etc.)

Característica: Independiente de la interfaz gráfica



Vista

Responsabilidad: Interfaz de usuario y presentación de datos

Ubicación: Paquete Vista/

Contiene: Formularios JFrame y componentes Swing

Característica: No contiene lógica de negocio

Características Principales

Gestión Multi-usuario con tres tipos de roles: Administrador, Vendedor y Cliente

Control de Inventario en tiempo real con categorización de productos

Procesamiento de Ventas con carrito de compras integrado

Sistema de Bitácoras para auditoria y seguimiento de actividades

Importación de datos mediante archivos CSV

Interfaz Intuitiva desarrollada con Java Swing

Dependencias Externas: (Archivo pom, pero En NetBeans necesitamos el .Jar)

```
<dependency>
  <groupId>com.opencsv</groupId>
  <artifactId>opencsv</artifactId>
  <version>5.7.1</version>
</dependency>
```

DESCRIPCIÓN DETALLADA DE CLASES

Jerarquia de Usuarios

CLASE USUARIO (CLASE BASE)

Proposito: Clase abstracta que representa la entidad base de todos los usuarios del sistema se define la estructura común que comparten todos los tipos de usuarios.

Atributos:

Atributo	Tipo	Descripción	Modificador
codigo	String	Identificador único del usuario en el sistema	protected
nombre	String	Nombre completo del usuario	protected
contra	String	Contraseña de acceso al sistema	protected

Metodos Principales:

Método	Tipo Retorno	Descripción
getCodigo()	String	Obtiene el codigo del usuario
setCodigo(String)	void	Establece el codigo del usuario
getNombre()	String	Obtiene el nombre del usuario
setNombre(String)	void	Establece el nombre del usuario
getContra()	String	Obtiene la contraseña del usuario
setContra(String)	void	Establece la contraseña del usuario

Relaciones:

Clases que heredan: Cliente, Vendedor, Administrador

CLASE CLIENTE (HEREDA DE USUARIO)

Proposito: Representa a los clientes que realizan compras en el sistema y almacena informacion especifica de clientes para personalización y seguimiento.

Atributos:

Atributo	Tipo	Descripción	Modificador
genero	String	Género del cliente (Masculino/Femenino)	private
cumple	String	Fecha de cumpleaños en formato DD/MM/AAAA	private

Metodos Especificos:

Método	Tipo Retorno	Descripción
getGenero()	String	Obtiene el género del cliente
setGenero(String)	void	Establece el género del cliente
getCumple()	String	Obtiene la fecha de cumpleaños
setCumple(String)	void	Establece la fecha de cumpleaños

Flujo de Uso:

Se crea al registrar nuevo cliente en el sistema

Se utiliza para personalizar experiencias de compra

Se referencia en transacciones y pedidos

CLASE VENDEDOR (HEREDA DE USUARIO)

Proposito: Representa a los vendedores autorizados para gestionar ventas, clientes y operaciones del sistema.

Atributo Especificos:

Atributo	Tipo	Descripción	Modificador
genero	String	Género del vendedor	private
numero_venta	int	Contador de ventas realizadas	private
confirmacion	String	Estado de confirmación del vendedor	private

Metodos Especificos:

Método	Tipo Retorno	Descripción
getGenero()	String	Obtiene el genero del vendedor
setGenero(String)	void	Establece el genero del vendedor
getNumero_venta()	int	Obtiene el numero de ventas realizadas
setNumero_venta(int)	void	Establece el numero de ventas
getConfirmacion()	String	Obtiene el estado de confirmación
setConfirmacion(String)	void	Establece el estado de confirmación

CLASE ADMINISTRADOR (HEREDA DE USUARIO)

Proposito: Usuario con privilegios totales sobre el sistema con acceso completo a todas las funcionalidades.

Caracteristicas:

Credenciales predefinidas (hardcodeadas)

Acceso a módulos administrativos

Capacidad de gestionar todos los usuarios

Gestión de Productos

CLASE PRODUCTOS (CLASE BASE)

Proposito: Representa los productos disponibles en el inventario del sistema. Clase base para especializaciones por categoría.

Atributos:

Atributo	Tipo	Descripción	Modificador
codigo_producto	String	Código único identificador del producto	protected
nombre_producto	String	Nombre descriptivo del producto	protected
categoria_producto	String	Categoría (Alimentos, Tecnología, General)	protected
precio_producto	double	Precio de venta al público	protected
atributo_unico	String	Atributo específico por categoría	protected
stock_productos	int	Cantidad disponible en inventario	protected
ventas_acumuladas	int	Total histórico de ventas del producto	protected

Metodos de Negocio:

Método	Tipo Retorno	Descripción
agregarVentas(int)	void	Establece las ventas acumuladas
verificarStock(int)	boolean	Verifica si hay stock suficiente
reducirStock(int)	void	Reduce el stock disponible

CLASE PRODUCTO_ALIMENTOS (HEREDA DE PRODUCTOS)

Proposito: Representa productos de la categoría alimentos con atributos específicos como fecha de expiración.

Atributos:

Atributo	Tipo	Descripción	Modificador
fecha_expiracion	String	Fecha de caducidad del producto	private
tipo_alimento	String	Tipo específico (Perecedero, No perecedero)	private

CLASE PRODUCTO_TECNOLOGIA (HEREDA DE PRODUCTOS)

Proposito: Representa productos tecnológicos con atributos como garantía y marca.

Atributos:

Atributo	Tipo	Descripción	Modificador
garantia	String	Período de garantía del producto	private

Atributo	Tipo	Descripción	Modificador
marca	String	Marca o fabricante del producto	private

CLASE PRODUCTO_GENERAL (HEREDA DE PRODUCTOS)

Proposito: Representa productos que no pertenecen a categorías específicas como alimentos o tecnología.

Características:

Utiliza los atributos base de Productos

Sin atributos específicos adicionales

Categoría por defecto: "General"

Gestión de Transacciones

CLASE CARRITO_COMPRAS

Proposito: Representa los ítems temporalmente agregados al carrito de compras antes de confirmar la venta.

Atributos:

Atributo	Tipo	Descripción	Modificador
codigo_producto	String	Código del producto en el carrito	private
nombre_producto	String	Nombre del producto	private
categoria	String	Categoría del producto	private

Atributo	Tipo	Descripción	Modificador
cantidad	int	Cantidad seleccionada por el cliente	private
precio	double	Precio unitario del producto	private
total	double	Total calculado (cantidad × precio)	private

CLASE PEDIDOS

Proposito: Representa pedidos confirmados y procesados en el sistema. registra transacciones completadas.

Atributos:

Atributo	Tipo	Descripción	Modificador
codigo_producto	String	Código del producto vendido	private
fecha_generacion	String	Fecha y hora de la venta	private
codigo_cliente	String	Código del cliente que realizó la compra	private
nombre_cliente	String	Nombre del cliente	private
total_pagar	double	Monto total de la venta	private

CLASE CARRITO_TEMPORAL

Proposito: Maneja la persistencia temporal del carrito de compras entre sesiones del mismo cliente.

Funcionalidad:

Almacena carritos no confirmados

Permite recuperar carritos al reiniciar sesión

Gestiona carritos por cliente

CLASE HISTORIAL_COMPRAS

Proposito: Registrar el historial completo de compras realizadas por cada cliente.

Funcionalidad:

Almacena transacciones históricas

Proporciona datos para reportes

Permite consultar compras anterior

CLASE EVENTOBITACORA

Proposito: Sistema de registro de auditoría que captura todas las actividades importantes del sistema.

Parametros:

Parámetro	Tipo	Descripción
modulo	String	Módulo del sistema donde ocurrió el evento
usuario	String	Usuario que realizó la acción
accion	String	Tipo de acción realizada
estado	String	Resultado (Éxito, Error, Advertencia)

Parámetro	Tipo	Descripción
descripcion	String	Descripción detallada del evento
detalles	String	Información adicional específica

CLASE INVENTARIOS_PDF

Proposito: Generación de reportes de inventario en formato PDF.

Funcionalidad:

Generación de reportes PDF

Formato estructurado de inventarios

Exportación de datos del sistema

CONTROLADORES PRINCIPALES

Controlador_Clientes

Propositos: Gestiona todas las operaciones relacionadas con clientes en el sistema, incluyendo creación, modificación, eliminación y consultas.

Funcionamiento:

Variable	Tipo	Descripción	Valor Inicial
crear_clientes	Cliente[]	Array de clientes registrados	new Cliente[100]
carrito	Carrito_Compras[]	Array de carritos de compra	new Carrito_Compras[100]

Variable	Tipo	Descripción	Valor Inicial
indice_clientes	int	Contador de clientes registrados	0
indice_carrito	int	Contador de ítems en carrito	0

MÉTODOS PRINCIPALES:

crearClientes(Cliente nuevo_cliente)

java

objetoCliente(String codigo)

java

validarClientesExistentes(String codigo)

java

eliminarCliente(String codi)

java

}

cargarCsvClientes(File archivo)

java

Controlador_Productos

Proposito: Gestiona el inventario de productos, incluyendo creación, búsqueda, actualización de stock y procesamiento de ventas.

Funcionamiento:

Variable	Tipo	Descripción	Valor Inicial
productos	Productos[]	Array de productos en inventario	new Productos[100]
indice_productos	int	Contador de productos registrados	0

Funcionalidades:

```
crearProductos(Productos nuevo_producto)
objetoProductos(String codigo)
actualizarStock(String codigo, int cantidad_vendida)
visualizarProductos(DefaultTableModel modelo)
```

Controlador_Usuarios

Proposito: Gestiona la autenticación y autorización de usuarios en el sistema.

Funcionalidades:

```
validarCredenciales(String codigo, String contra)
```

```
)) {
```

Controlador_Vendedor

Proposito: Gestiona las operaciones específicas de los vendedores en el sistema.

Funcionalidades:

Variable	Tipo	Descripción	Valor Inicial
crear_vendedores	Vendedor[]	Array de vendedores registrados	new Vendedor[50]
indice_vendedores	int	Contador de vendedores registrados	0

Funcionalidades:

crearVendedores(Vendedor nuevo_vendedor)

incrementarVentasVendedor(String codigo_vendedor)

Controlador_Carrito_Temporal

Proposito: Gestiona el carrito de compras temporal entre sesiones de usuario.

Funcionalidades:

Variable	Tipo	Descripción	Valor Inicial
carritos_temporales	Carrito_Temporal[]	Array de carritos temporales	new Carrito_Temporal[100]
indice_carritos_temp	int	Contador de carritos temporales	0

MÉTODOS PRINCIPALES:


```
guardarCarritoTemporal(String codigo_cliente, Carrito_Compras[] carrito)
}
cargarCarritosTemporalesCliente(String codigo_cliente)
```

VISTAS PRINCIPALES

6.1 Inicio_Shop.java

Proposito: Ventana principal de inicio del sistema, punto de entrada para todos los usuarios.

Componentes Principales:

Campos para código y contraseña

Botón de inicio de sesión

Selector de tipo de usuario

Enlaces para registro de nuevos usuarios

Funcionalidades:

Autenticación de usuarios

Redirección a módulos según tipo de usuario

Manejo de credenciales incorrectas

Modulo_Vendedor

Proposito: Interfaz principal para vendedores con acceso a todas las funcionalidades de ventas.

Funcionalidades:

Gestión de Clientes - CRUD de clientes

Gestión de Productos - Administración de inventario

Procesar Ventas - Carrito de compras y pedidos

Reportes - Estadísticas y bitácoras

Funcionalidades:

```
visualizacionClientes()  
mostrarPestanaActual()  
}
```

Modulo_Administrador

Proposito: Interfaz con privilegios administrativos completos sobre el sistema.

Funcionalidades:

Gestión de todos los usuarios
Configuración del sistema
Reportes completos
Backup de datos

Actualizar_cliente

Proposito: Formulario específico para modificar datos de clientes existentes.

Flujo Operación:

Buscar cliente por código
Cargar datos en formulario
Modificar campos necesarios
Guardar cambios
Actualizar datos en sistema

MANEJO DE ERRORES Y BITÁCORAS

Clasificación de Errores

Errores de Validación de Datos

Campos Vacíos: Verificación de campos obligatorios

Formatos Invalidos: Validación de fechas, números, códigos

Longitudes Excedidas: Control de límites de caracteres

Datos Duplicados: Verificación de códigos únicos

Errores de Lógica de Negocio

Stock Insuficiente: Cuando no hay productos disponibles

Saldo Insuficiente: Validación de límites de compra

Operaciones Invalidas: Acciones no permitidas por rol de usuario

Errores del Sistema

Archivos No Encontrados: Cuando no existe un recurso requerido

Problemas de Entrada/Salida: Errores de lectura/escritura

Memoria Insuficiente: Cuando se excede la capacidad del sistema

Comunicación al Usuario:

Mensajes específicos y accionables

Información sobre cómo proceder

Registro detallado para soporte técnico

Sistema de Bitácoras (Logging)

Arquitectura del Sistema de Bitácoras

Niveles de Registro:

Información: Operaciones exitosas del sistema

Advertencia: Situaciones que requieren atención

Error: Fallos que afectan funcionalidad

Implementación de Bitácoras

Estructura de Registro:

Fecha y hora exacta del evento

Módulo: Componente del sistema donde ocurrió

Usuario: Identificación del usuario involucrado

Acción: Operación específica realizada

Estado: Resultado (Éxito, Error, Advertencia)

Descripción: Resumen legible del evento

Detalles: Información técnica específica

Registro de Eventos Específicos

Eventos de Autenticación

Inicios de sesión exitosos y fallidos

Cambios de contraseña

Bloqueos de cuenta por intentos fallidos

Eventos de Gestión de Datos

Creación, modificación y eliminación de registros

Importación/exportación de datos

Eventos Transaccionales

Procesamiento de ventas y pedidos

Movimientos de inventario

Transacciones financieras

Eventos del Sistema

Inicio y cierre de la aplicación

Errores críticos del sistema

Eventos de mantenimiento

MANTENIMIENTO Y SEGURIDAD

Mantenimiento

Limpieza semanal de archivos temporales

Monitoreo constante de bitácoras

Actualizaciones (mejoramientos)

Seguridad

Autenticación por usuario y contraseña

Control de acceso por roles

Validación de todas las entradas

Bitácoras de todas las operaciones

