

Documentación del Sistema de Gestión de Concesionaria de Autos

Descripción General

El Sistema de Gestión de Concesionaria de Autos es una aplicación de consola basada en Java diseñada para administrar el inventario de autos nuevos y usados, registrar ventas y realizar tareas básicas de contabilidad. El sistema permite a los usuarios agregar autos al inventario, vender autos a clientes, ver la lista de clientes y mostrar informes financieros. La persistencia de datos se logra mediante archivos de texto para el inventario y la información de los clientes.

Componentes del Sistema

Clases

Autos1

- **Propósito:** Clase base que representa un auto con atributos comunes.
- **Atributos:**
 - nombrepers: Nombre de la persona asociada al auto.
 - nserie: Número de serie de 17 caracteres (alfanumérico).
 - marca: Marca del auto.
 - modelo: Modelo del auto.
 - version: Versión del auto.
 - color: Color del auto.
 - año: Año de fabricación (1950-2026).
 - precioCompra: Precio de compra.
 - precio: Precio de venta.
- **Métodos:**
 - Getters para todos los atributos.
 - toString(): Devuelve una representación en cadena de los detalles del auto.

Nuevos (extiende Autos1)

- **Propósito:** Representa autos nuevos con atributos adicionales.
- **Atributos:**
 - garantia: Período de garantía (1-5 años).
 - accesorios: Equipamiento o accesorios.
 - enganche: Pago inicial (hasta el 15% del precio de venta).
- **Métodos:**
 - Getters para los atributos adicionales.
 - Sobrescribe toString() para incluir garantía, accesorios y enganche.

Seminuevos (extiende Autos1)

- **Propósito:** Representa autos usados con atributos adicionales.
- **Atributos:**
 - kilometraje: Kilometraje del auto.
 - cantidadDueños: Número de dueños anteriores.
 - nacionalidad: Nacionalidad del auto.
- **Métodos:**
 - Getters para los atributos adicionales.
 - Sobrescribe toString() para incluir kilometraje, dueños anteriores y nacionalidad.

Ciente

- **Propósito:** Representa a un cliente que ha comprado un auto.
- **Atributos:**
 - nombre: Nombre del cliente.
 - unidadComprada: Descripción del auto comprado (marca, versión, color).
 - precioCompra: Precio pagado por el cliente.
- **Métodos:**
 - Getters para todos los atributos.

ArrayListAutos

- **Propósito:** Administra el inventario de autos y la lista de clientes, maneja entrada/salida de archivos e implementa funcionalidades de contabilidad.
- **Atributos:**
 - inventarioAutos: Lista de autos en el inventario.
 - listaClientes: Lista de clientes.
 - INVENTARIO_FILE: Archivo para almacenar datos del inventario (inventario.txt).
 - CLIENTES_FILE: Archivo para almacenar datos de clientes (clientes.txt).
- **Métodos:**
 - agregarAuto(Autos1): Agrega un auto al inventario y guarda en archivo.
 - mostrarInventario(): Muestra el inventario actual.
 - venderAuto(int, String): Vende un auto en el índice especificado a un cliente y actualiza archivos.
 - mostrarListaClientes(): Muestra la lista de clientes.
 - mostrarContabilidad(): Muestra un resumen financiero (ingresos, costos, ganancias).
 - trabajoContable(): Implementa la interfaz Contabilidad para cálculos financieros.
 - cargarInventario(): Carga el inventario desde un archivo.
 - guardarInventario(): Guarda el inventario en un archivo.
 - cargarClientes(): Carga los clientes desde un archivo.
 - guardarClientes(): Guarda los clientes en un archivo.
 - getInventarioAutos(): Devuelve la lista de inventario.

Principal

- **Propósito:** Clase principal que proporciona la interfaz de usuario mediante un menú en consola.
- **Funcionalidad:**
 - Muestra un menú con opciones: comprar (agregar) un auto, vender un auto, ver contabilidad, ver clientes o salir.
 - Maneja la validación de entrada con límites de reintentos (5 intentos por campo).
 - Permite agregar autos nuevos (Nuevos) o usados (Seminuevos) con atributos específicos.
 - Asegura que el precio de venta no sea menor al precio de compra usando PrecioInvalidoException.
 - Gestiona la interacción del usuario para vender autos y ver informes.

Contabilidad

- **Propósito:** Interfaz que define la funcionalidad de contabilidad.

- **Método:**
 - o `trabajoContable()`: Debe ser implementado por las clases para cálculos financieros.

PrecioInvalidoException

- **Propósito:** Excepción personalizada para escenarios de precios inválidos.
- **Atributos:**
 - o `precioInvalido`: El valor de precio inválido.
- **Constructores:**
 - o Por defecto, con mensaje, con mensaje y precio, con mensaje y causa.
- **Método:**
 - o `getPrecioInvalido()`: Devuelve el precio inválido.

Funcionalidades

1. Agregar un Auto (Opción 1)

- Los usuarios pueden agregar un auto nuevo o usado al inventario.
- Los campos de entrada incluyen nombre, número de serie, marca, modelo, versión, color, año, precio de compra y precio de venta.
- Para autos nuevos: garantía, accesorios y enganche.
- Para autos usados: kilometraje, número de dueños anteriores y nacionalidad.
- La validación de entrada asegura formatos correctos (por ejemplo, número de serie de 17 caracteres, año entre 1950-2026).
- El formulario se reinicia después de 5 intentos fallidos para cualquier campo.

2. Vender un Auto (Opción 2)

- Muestra el inventario con índices, marcas, modelos, versiones y precios de venta.
- Los usuarios seleccionan un auto por índice y proporcionan el nombre del cliente.
- El auto se elimina del inventario y el cliente se agrega a la lista de clientes.
- Los datos se guardan en los archivos.

3. Contabilidad (Opción 3)

- Muestra los ingresos totales por ventas, el costo total de los autos en inventario y las ganancias.
- Las ganancias se calculan como la diferencia entre los precios de venta y compra de los autos vendidos.

4. Lista de Clientes (Opción 4)

- Muestra una tabla de clientes con sus nombres, unidades compradas y precios de compra.

5. Salir (Opción 5)

- Cierra la aplicación y el scanner.

Persistencia de Datos

- **Inventario:** Almacenado en `inventario.txt` con valores separados por comas. Cada línea comienza con NUEVO o SEMINUEVO para indicar el tipo de auto.
- **Clientes:** Almacenado en `clientes.txt` con el nombre del cliente, la unidad comprada y el precio.

- Los archivos se cargan al iniciar y se guardan después de modificaciones.

Manejo de Errores

- Validación de entrada con límites de reintentos.
- Excepción personalizada PrecioInvalidoException para validación de precios.
- Manejo de excepciones para entrada/salida de archivos, análisis de números y límites de arreglos.

Instrucciones de Uso

1. Compila y ejecuta Principal.java.
2. Selecciona una opción del menú (1-5).
3. Sigue las indicaciones para ingresar datos, asegurándote de usar los formatos correctos.
4. Visualiza el inventario, vende autos o revisa los informes financieros según sea necesario.
5. Sal para guardar los cambios y cerrar el programa.

Limitaciones

- Sin interfaz gráfica; solo basado en consola.
- Recuperación de errores limitada para archivos corruptos.
- Los cálculos contables pueden no considerar todos los casos extremos (por ejemplo, la correspondencia de autos para el cálculo de ganancias es simplista).
- No hay soporte para modificar entradas de inventario existentes.

Mejoras Futuras

- Agregar una interfaz gráfica para mejorar la experiencia del usuario.
- Implementar la modificación y eliminación de autos.
- Mejorar la contabilidad con informes detallados (por ejemplo, por tipo de auto o fecha).
- Agregar autenticación de usuarios para un acceso seguro.
- Mejorar el manejo de archivos con una base de datos para mayor escalabilidad.

Estructura de Archivos

- Autos1.java: Clase base para autos.
- Nuevos.java: Subclase para autos nuevos.
- Seminuevos.java: Subclase para autos usados.
- Cliente.java: Clase para clientes.
- ArrayListAutos.java: Gestión de inventario y clientes.
- Principal.java: Lógica principal de la aplicación.
- Contabilidad.java: Interfaz de contabilidad.
- PrecioInvalidoException.java: Excepción personalizada.
- inventario.txt: Archivo de datos del inventario.
- clientes.txt: Archivo de datos de clientes.