Estructuras de datos

Manual técnico

Jesús Pencue

Diego Gaviria

Docente:

Jimmy Roman Valdes Santacruz

Unicomfacauca

Decanatura de División de Educación Presencial

Ingeniería de Sistemas

Popayán

2024

Manual Técnico para el Sistema de Inventario de Tienda de Celulares

Índice

Introducción

Este manual técnico describe el funcionamiento interno del sistema de inventario de tienda de celulares. El sistema permite gestionar el inventario de celulares, incluyendo operaciones como agregar, actualizar, mostrar, eliminar y vender dispositivos móviles.

Requisitos del Sistema

- Java Development Kit (JDK) versión 8 o superior.
- Entorno de Desarrollo Integrado (IDE) como IntelliJ IDEA, Eclipse o NetBeans.
- Bibliotecas estándar de Java.

Estructura del Código

El código se organiza en una clase principal **“TiendaCelulares”**, que contiene métodos para manejar diversas operaciones de inventario. Las operaciones se realizan sobre dos listas:
listaCelulares: Contiene los celulares disponibles para la venta.

listaCelularesVendidos: Contiene los celulares que han sido vendidos.

Descripción de Clases y Métodos

Clase **"TiendaCelulares"**
Atributos
"private ArrayList<Celulares> listaCelulares": Lista de celulares disponibles para la venta.
"private ArrayList<Celulares> listaCelularesVendidos": Lista de celulares vendidos.

Métodos

1. "private Celulares pedirDatosCelular()"

- Solicita al usuario ingresar los datos de un celular y devuelve una instancia de **"Celulares".**
- **Validaciones:**
  - ✓ La cantidad de SIM debe ser 1 o 2.
  - ✓ El código del celular debe ser único.

2. "**public void agregar()"**

- Agrega un nuevo celular a **"listaCelulares"**.
- Proceso:
  - ✓ Llama a **"pedirDatosCelular().**
  - ✓ Si los datos son válidos, agrega el celular a la lista y muestra un mensaje de confirmación.

3. "public void actualizar()"

- Actualiza la información de un celular en "listaCelulares".
- Proceso:
  - ✓ Solicita el índice del celular a actualizar.
  - ✓ Llama a **"pedirDatosCelular()"**.
  - ✓ Si los datos son válidos, actualiza el celular en la lista.

4. “public void mostrarListaCelulares()”

- Muestra la lista de celulares disponibles para la venta.

5. “public void mostrarListaCelularesVendidos()”

- Muestra la lista de celulares vendidos.

6. “public void eliminar()”

- Elimina un celular de “listaCelulares”.
- Proceso:
  - ✓ Solicita el índice del celular a eliminar.
  - ✓ Si el índice es válido, elimina el celular de la lista y muestra un mensaje de confirmación.

7. “public void vender()”

- Registra la venta de un celular, moviéndolo de “listaCelulares” a “listaCelularesVendidos” y muestra una factura.
- Proceso:
  - ✓ Solicita el nombre del vendedor y el índice del celular a vender.
  - ✓ Si el índice es válido, genera y muestra una factura, mueve el celular a “listaCelularesVendidos” y lo elimina de “listaCelulares”.

8“private void mostrarFactura(String nombreVendedor, Celulares celular, String fechaDeCompra)”

- Muestra la factura de la venta de un celular.
- Detalles:
  - Incluye fecha de compra, nombre del vendedor, detalles del celular y precio.

Flujo de Ejecución

1. Inicialización

- Se crea una instancia de **“TiendaCelulares”.**
- Se llaman los métodos apropiados según las operaciones deseadas.

2. Agregar Celular

- Llama a **“agregar()”.**
- Solicita datos del celular.
- Agrega el celular a **“listaCelulares”.**

3. Actualizar Celular

- Llama a **“actualizar()”.**
- Solicita índice y datos del celular.
- Actualiza el celular en **“listaCelulares”.**

4. Mostrar Listas

- Llama a **“mostrarListaCelulares()”** o **“mostrarListaCelularesVendidos()”.**

5. **Eliminar Celular**

- Llama a **“eliminar()”.**
- Solicita índice.
- Elimina el celular de **“listaCelulares”.**

6. **Vender Celular**

- Llama a **“vender()”.**
- Solicita nombre del vendedor e índice del celular.
- Genera factura, mueve el celular a **“listaCelularesVendidos”** y lo elimina de **“listaCelulares”.**

Consideraciones de Diseño

- Validación de Datos: Asegura que los datos ingresados por el usuario sean correctos y únicos donde sea necesario.
- Manejo de Listas: Utiliza **“ArrayList”** para gestionar dinámicamente las listas de celulares.

- Interfaz de Usuario: Interacción basada en consola para simplificar la entrada y salida de datos.

Pruebas y Depuración

- **Pruebas Unitarias:** Se recomienda escribir pruebas unitarias para cada método utilizando un framework como JUnit.
- **Depuración:** Utilizar las capacidades de depuración del IDE para poner puntos de interrupción y seguir el flujo del programa.
- **Mensajes de Error:** Verificar que se muestran mensajes de error adecuados para entradas inválidas.

Ejemplo de Uso

```
public class Main {
    public static void main(String[] args) {
        TiendaCelulares tienda = new TiendaCelulares();

        // Agregar un celular
        tienda.agregar();

        // Mostrar la lista de celulares
        tienda.mostrarListaCelulares();

        // Actualizar un celular
        tienda.actualizar();

        // Eliminar un celular
        tienda.eliminar();

        // Vender un celular
        tienda.vender();

        // Mostrar la lista de celulares vendidos
        tienda.mostrarListaCelularesVendidos();
    }
}
```

Conclusion

Este manual técnico proporciona una guía detallada sobre la estructura, funcionamiento y consideraciones de diseño del sistema de inventario de tienda de celulares. Siga las instrucciones y buenas prácticas descritas para mantener y ampliar el sistema de manera eficiente.