Servicio Nacional de Aprendizaje
(SENA)

Técnica

Construcción de aplicaciones con Java GA7-220501096-AA2-EV01

Santiago Rodríguez Vallejo

Ficha N° 2721431

Medellín, Antioquia

Introducción

En el desarrollo de aplicaciones, la gestión de bases de datos es fundamental para el éxito de cualquier sistema. Java Database Connectivity (JDBC) es una API que permite a los desarrolladores conectar aplicaciones Java con bases de datos relacionales de manera sencilla y eficiente. Este trabajo tiene como objetivo desarrollar un pequeño CRUD de las funcionalidades fundamentales del sistema en desarrollo: GAVI (Gestión Administrativa de Ventas e Inventario) para tener un acercamiento del funcionamiento de la API y entender cómo desde Java se puede manipular información con la base de datos.

Visite el siguiente repositorio para poder revisar el código:

https://github.com/Sanrvallejo/Actividades-Sena.git

Desarrollo de las funcionalidades

Requerimientos a tener en cuenta:

RF02	El software permitirá registrar productos con la información pertinente para calcular el precio y llevar contabilidad de gastos.
RF05	El software permitirá la creación de órdenes de ventas, incluyendo información sobre los productos vendidos, la cantidad, el precio unitario, el cliente y otros detalles relevantes.
RF08	El software guardará un registro histórico de las ventas realizadas para futuras consultas y análisis.
RF13	El software calculará automáticamente el monto total de la venta, incluyendo impuestos y descuentos aplicables.
RNF08	El software utilizará MYSQL como herramienta de base de datos.

Los requerimientos anteriores se atienden de la siguiente manera:

RF02: Para el registro de productos se crea una entidad llamada Productos que será la tabla productos en la base de datos. La lógica de negocio pertinente a esta funcionalidad se da por medio del DAO de productos con los métodos insertar, obtener, obtener todos, actualizar y eliminar.

RF05: este requerimiento se atiende por medio de las clases Ventas y Detalle y sus tablas respectivas. Esta funcionalidad se maneja mediante los DAO de ventas y detalle de venta en los métodos crear venta, insertar venta, obtener venta, crear detalle de venta, insertar detalle de venta.

RF08: se atiende por medio de la clase Ventas que es la tabla Ventas en la base de datos, la cual es encargada de llevar un histórico de las ventas con su respectiva información.

RF13: se atiende por medio de los métodos crear venta y crear detalle de venta en los DAO de ventas y detalle de ventas respectivamente.

Decisiones:

1. El uso de PreparedStatement en lugar de Statement se hace con el fin de emplear un código seguro en la inserción de datos a la base de datos, puesto que el método PreparedStatemetn no permite los ataques de inyección SQL porque envía sus valores como parámetros y no directamente en la sentencia de SQL.
2. Implementar un gestor DAO para manejar toda la conexión desde una clase.
3. La no implementación de una clase Servicio se debe a la intención de simplificar el ejercicio, sin embargo, se reconoce que es mejor implementarla como buena práctica para no sobre cargar funciones en clases que no deben encargarse de ello.
4. El único CRUD completo es el de productos para demostrar la elaboración de uno. Para las demás entidades no se consideró completarlo ya que el ejercicio solo es para ejemplificar la funcionalidad principal del programa en cuanto a la gestión de las ventas.

Posibles mejoras:

1. El método insertar debe devolver el objeto que crea para dar mejor manejo a funcionalidades que dependen de la creación simultanea de dos registros en tablas diferentes como es el caso de Venta y Detalle de venta. Esto puede lograrse mediante la devolución de un ResultSet con el objeto completo o con el ID del registro empleando el método getGeneratedKeys.
2. El método obtener pude emplearse con diferentes métodos de búsqueda para facilitar el manejo del software y mejorar la experiencia de usuario.

Conclusiones

A pesar de ser una tecnología antigua, JDBC sigue siendo fundamental para conectar aplicaciones Java con bases de datos relacionales. He aprendido que usar PreparedStatement mejora la seguridad y eficiencia, ResultSet facilita la manipulación de datos, y el manejo de transacciones con commit y rollback asegura la integridad de los datos. Además, este aprendizaje prepara un camino de fundamentos sobre el funcionamiento de la conexión a bases de datos para entrar a explorar tecnologías como JPA, que es más moderna y eficiente al poseer repositorios que facilitan la manipulación de datos.