

Trabajo Práctico Integrador - Programación II

Fecha: 19/11/2025

Alumnos:

- Federico Garcia Bengolea - feddericogarciaa@gmail.com - Legajo 101613
- Jazmin Herrera - herrerajazmin934@gmail.com - Legajo
- Alan Jofre - alan91_10@hotmail.com - Legajo 4846
- Akier Aguirrezabala - akieraguirrezabala@gmail.com - Legajo 611581

Profesores:

- Profesor: Alberto Cortez
- Tutor: Ramiro Hualpa

Índice

1. Introducción
2. Objetivo General del Proyecto
3. Descripción Funcional del Sistema
4. Arquitectura del Proyecto
5. Tecnologías Utilizadas
6. Dependencias y Configuración Inicial
7. Estructura de Carpetas del Proyecto
8. Modelo de Datos y Estructura de Tablas
9. Validaciones y Manejo de Excepciones
10. Funcionamiento del Menú Principal
11. Guía de Instalación (Windows + NetBeans + XAMPP/MySQL)

12. Configuración del Driver JDBC
13. Ejecución del Proyecto
14. Errores Comunes y Soluciones
15. Casos de Uso
16. Flujo Transaccional del Alta de Producto + Código
17. Mejoras Futuras Propuestas
18. Conclusión

1. Introducción

El presente documento describe el desarrollo del Trabajo Práctico Integrador correspondiente a la materia Programación II, cuyo objetivo es implementar un sistema de gestión orientado a productos y sus códigos de barras asociados.

El proyecto fue desarrollado íntegramente en Java 21, siguiendo buenas prácticas de programación, modularización, manejo de excepciones y acceso seguro a datos mediante JDBC. Su persistencia se implementó sobre MySQL utilizando XAMPP como servidor local.

Este README cumple una doble función:

- servir como guía de uso del sistema
- funcionar como documentación técnica base para el informe PDF del proyecto

2. Objetivo General del Proyecto

El objetivo central del proyecto es construir un sistema de software de gestión de productos que permita manipular información almacenada de manera persistente. Para ello, se propone diseñar una aplicación que:

- Permita crear, editar, consultar y eliminar productos.
- Asocie a cada producto un código de barras único de tipo EAN-13.
- Garantice la integridad mediante una relación uno a uno obligatoria y exclusiva.
- Valide estrictamente los datos ingresados (nombre, precio y código).
- Evite duplicados y preserve la unicidad del código de barras.
- Utilice transacciones para asegurar la atomicidad de todas las operaciones de inserción relacionadas.
- Aplique buenas prácticas de manejo de excepciones.

- Cumpla con los principios de arquitectura multicapa utilizados en la industria.

3. Descripción Funcional del Sistema

El sistema permite realizar las siguientes operaciones:

Gestión de Productos

Este módulo permite crear, consultar, modificar y eliminar productos almacenados en la base de datos. Cada producto contiene información básica como nombre y precio, y además debe tener asignado un código de barras único.

Las principales funciones son:

- Crear un producto junto con su código de barras correspondiente (operación transaccional).
- Listar todos los productos registrados.
- Actualizar el nombre o el precio de un producto existente.
- Eliminar un producto del sistema.
- Validar que los datos ingresados (nombre y precio) sean correctos antes de almacenarlos.

Gestión de Códigos de Barras

Este módulo permite administrar los códigos de barras EAN-13 asociados a los productos.

Incluye las siguientes funcionalidades:

- Listar todos los códigos existentes.
- Buscar un código puntual ingresando su valor numérico.
- Eliminar un código de barras específico.
- Verificar que cada código cumpla con el formato EAN-13 (13 dígitos numéricos y sin duplicados).

Reglas del Sistema

Para asegurar la integridad de los datos, se establecen reglas claras:

- Cada producto debe tener exactamente un código de barras.
- Cada código pertenece de forma exclusiva a un único producto.

- El código es único en toda la base de datos.
- El nombre del producto debe ser válido y el precio debe ser mayor que cero.
- Toda operación de creación de producto y código debe realizarse como una transacción, garantizando que ambos datos se guarden correctamente o que ninguno se registre en caso de error.

4. Arquitectura del Proyecto

El sistema se implementó bajo una arquitectura por capas:

Presentación (AppMenu / Main)



Servicio (ProductoService, CodigoBarrasService)



Acceso a Datos (DAO)



Conexión a BD (DatabaseConnection)



Base de Datos (MySQL)

Esta arquitectura se basa en:

- Claridad conceptual:** cada capa tiene un propósito claramente definido.
- Reutilización del código:** las capas inferiores (DAO, conexión, entidades) no dependen de la interfaz del usuario.
- Robustez:** los servicios controlan transacciones, validaciones y errores de manera ordenada.
- **Mantenibilidad:** permite cambiar componentes individuales sin afectar el sistema completo.
- **Escalabilidad:** el sistema podría transformarse en una API REST, un sistema con interfaz gráfica o incluso una app web sin modificar las capas inferiores.

5. Tecnologías Utilizadas

Tecnologías	Versión	Uso
-------------	---------	-----

Java	21	Lógica del sistema
NetBeans	20	IDE principal
MySQL (XAMPP)	8.x	Motor de base de datos
mysql-connector-j	8.4.0	Driver JDBC oficiales
JDBC	-	Capa de persistencia
Consola / CLI	-	Interfaz de usuario

6. Dependencias y Configuración Inicial

1. Instalar Java 21
2. Verificar con: `java -version`
3. Instalar XAMPP (incluye MySQL)
4. Asegurarse de iniciar el módulo:
5. MySQL
6. Apache (opcional)
7. Crear la base de datos
8. El sistema utiliza:

Usuario: root

Password: vacío

Base: tpi_db

7. Estructura del Proyecto

/src

```

├── config
│   └── DatabaseConnection.java
├── dao
│   ├── ProductoDAO.java
│   └── CodigoBarrasDAO.java
├── entities
│   ├── Producto.java
│   └── CodigoBarras.java
├── exceptions
│   └── DataAccessException.java
├── service
│   ├── ProductoService.java
│   └── CodigoBarrasService.java

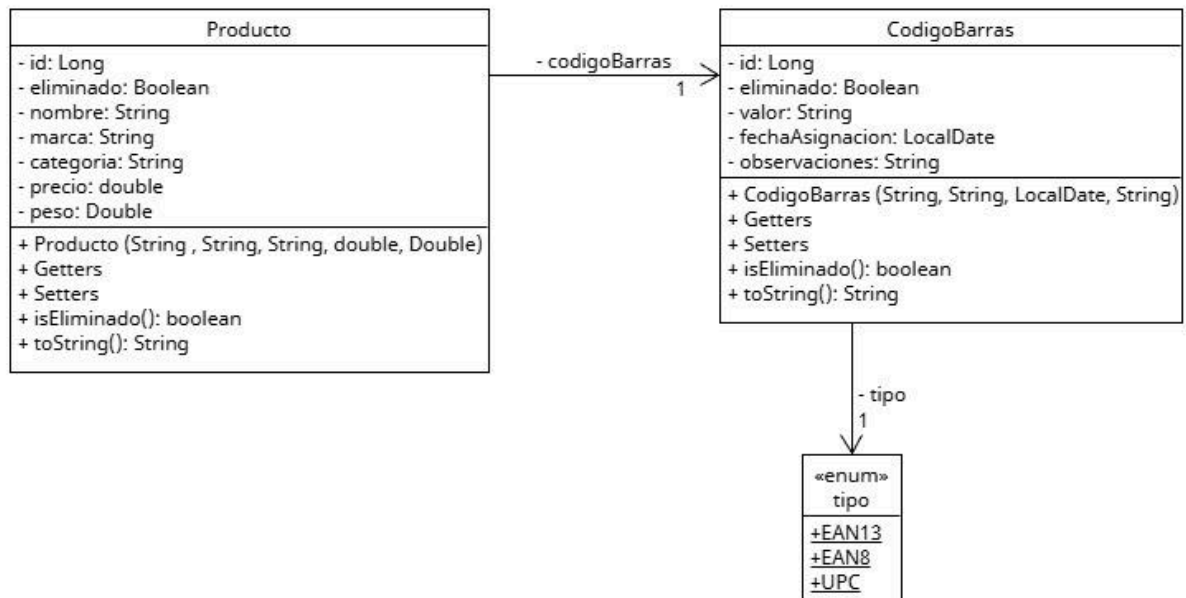
```

```

├── utils
│   └── Validador.java
├── main
│   ├── AppMenu.java
│   └── Main.java

```

Diagrama Uml del Sistema:



8. Modelo de Datos y Estructura de Tablas

Modelo relacional: 1 producto → 1 código de barras

Tabla: producto

```

CREATE TABLE producto (
    id INT AUTO_INCREMENT PRIMARY KEY,
    nombre VARCHAR(100) NOT NULL,
    precio DECIMAL(10,2) NOT NULL
);

```

Tabla: codigo_barras

```

CREATE TABLE codigo_barras (
    id INT AUTO_INCREMENT PRIMARY KEY,
    codigo VARCHAR(13) NOT NULL UNIQUE,
    producto_id INT UNIQUE,
    FOREIGN KEY (producto_id) REFERENCES producto(id) ON DELETE CASCADE
);

```

);

9. Validaciones y Manejo de Excepciones

Validación de Producto

- Nombre no puede estar vacío ni tener menos de 3 caracteres.
- Se convierte a mayúsculas automáticamente.
- Precio debe ser numérico y mayor que cero.

Validación de Código de Barras (EAN-13)

- 13 dígitos exactos.
- Sin letras.
- No duplicado.
- Debe cumplir integridad referencial.

Excepciones manejadas

- SQLException
- DataAccessException personalizada
- IllegalArgumentException
- NumberFormatException
- Errores transaccionales y rollback automático

10. Funcionamiento del Menú Principal

Ejemplo de ejecución:

=====

MENU PRINCIPAL

=====

1. Gestión de Productos
2. Gestión de Códigos de Barras
0. Salir

Menú de Productos:

1. Crear Producto con Código

2. Listar Productos
3. Actualizar Producto
4. Eliminar Producto
0. Volver

Menú de Códigos:

1. Listar Códigos
2. Buscar Código
3. Eliminar Código
0. Volver

11. Guía de Instalación (Windows)

1. Instalar XAMPP
2. Iniciar MySQL
3. Crear la base tpi_db
4. Instalar Java 21
5. Abrir NetBeans 20
6. Descargar mysql-connector-j 8.4.0
7. Agregar el JAR al proyecto: Project > Properties > Libraries > Add JAR
8. Ejecutar Main.java

12. Configuración del Driver JDBC

URL utilizada:

jdbc:mysql://localhost:3306/tpi_db?useSSL=false&allowPublicKeyRetrieval=true
&serverTimezone=UTC

Usuario: root

Contraseña: vacía

13. Ejecución del Proyecto

Desde NetBeans:

Run → Run Project

14. Errores Comunes y Soluciones

Error	Causa	Solución
Unknown database	Nombre incorrecto	Verificar tpi_db
No suitable driver	Driver JDBC faltante	Agregar el .jar
Access denied	Pass incorrecta	Configurar DatabaseConnection
Duplicate entry	Código duplicado	Validar antes del insert
SQLSyntaxError	Error en script	Revisar estructura

15. Casos de Uso

- Crear producto + código
- Evitar duplicados
- Actualización de precios
- Eliminación lógica o física
- Búsqueda rápida por código

16. Flujo Transaccional del Alta

- Iniciar transacción
- Insertar producto
- Insertar código
- Si falla → rollback
- Si todo correcto → commit

17. Mejoras Futuras Propuestas

- Reemplazar eliminación física por baja lógica
- Interfaz gráfica (JavaFX o Swing)
- Exportación a CSV/Excel
- Inclusión de categorías o stock
- Transformación a API REST con Spring Boot
- Cálculo y validación automática del dígito verificador EAN-13

18. Conclusión

El proyecto permitió integrar de forma práctica los contenidos fundamentales de Programación II, aplicando conceptos de programación orientada a objetos, estructura por capas, uso de JDBC, manejo de excepciones y diseño de bases de datos. A través del sistema de gestión de productos y códigos de barras, se implementaron operaciones CRUD completas, validaciones y un flujo transaccional que asegura la integridad de los datos.

El trabajo cumple con los requisitos de la materia y ofrece una base sólida para futuras mejoras, como agregar una interfaz gráfica, nuevas funcionalidades o incluso evolucionar hacia una API REST. En síntesis, el desarrollo del sistema permitió consolidar conocimientos y demostrar la capacidad de construir aplicaciones robustas, organizadas y escalables.