

PROGRAMACIÓN II

INFORME FINAL – TRABAJO PRÁCTICO INTEGRADOR

Sistema de Gestión de Pedidos y Envíos

Grupo 174_Pedido-Envío

Alumna: Mónica Martín C -17

Índice

1. *Introducción*
2. *Objetivo del sistema*
3. *Descripción funcional*
4. *Modelo de base de datos*
5. *Script SQL de creación*
6. *Arquitectura del sistema en Java*
7. *Diagrama UML*
8. *Desarrollo del sistema*
 - 8.1 *Entidades*
 - 8.2 *DAO*
 - 8.3 *Servicios*
 - 8.4 *Menú principal*
9. *Pruebas realizadas*
10. *Capturas (indicadas)*
11. *Conclusiones*

1. Introducción

El presente Trabajo Práctico Integrador tiene como objetivo aplicar los contenidos vistos en la asignatura Programación II de la UTN, desarrollando un sistema funcional que integra Java, MySQL, JDBC y el manejo de estructuras orientadas a objetos.

El trabajo consiste en implementar un sistema completo de gestión de pedidos y envíos, que permita registrar, consultar, actualizar y eliminar información, manteniendo integridad de datos y una correcta interacción entre las entidades.

2. Objetivo del sistema

El sistema tiene como finalidad gestionar:

- **Envíos:** datos del despacho, estado logístico y costos.
- **Pedidos:** datos del cliente, fecha, importe y estado comercial.
- **Relación 1:1:** Cada pedido posee exactamente un envío asociado.

Además, el sistema debe permitir:

- Inserción de registros
- Borrado lógico
- Listado completo de información
- Búsquedas
- Actualizaciones
- Manejo de transacciones al crear Pedido + Envío

3. Descripción funcional del sistema

El sistema implementa las siguientes funciones principales:

✓ Creación de Envíos

Carga de tracking, empresa, tipo de envío, fechas, costo y estado logístico.

✓ Creación de Pedido + Envío (transacción)

Primero se crea el Envío, luego el Pedido.

Si algo falla → se aplica **rollback** y no se guarda nada.

✓ Listado de Envíos y Pedidos

Incluye filtrado por eliminados lógicos.

✓ Búsquedas

- Envío por tracking
- Pedido por ID

✓ Actualizaciones

Cambiar el estado del envío o del pedido.

✓ Eliminación lógica

No borra físicamente; marca registros como eliminados.

4. Modelo de Base de Datos

Entidad envio

- *id (PK, auto_increment)*
- *tracking (único)*
- *empresa (ENUM)*
- *tipo (ENUM)*
- *costo (decimal)*
- *fechaDespacho*
- *fechaEstimada*
- *estado (ENUM)*
- *eliminado (boolean)*

Entidad pedido

- *id (PK, auto_increment)*

- *numero* (*único*)
- *fecha*
- *clienteNombre*
- *total*
- *estado* (*ENUM*)
- *envio* (*FK UNIQUE a envio.id*)
- *eliminado* (*boolean*)

Relación:

pedido.envio = envio.id

Relación 1:1 con restricción UNIQUE.

5. Script SQL de creación de la base

CREATE DATABASE IF NOT EXISTS mdb_tpi;

USE mdb_tpi;

```
CREATE TABLE envio (
    id BIGINT PRIMARY KEY AUTO_INCREMENT,
    tracking VARCHAR(50) UNIQUE NOT NULL,
    empresa ENUM('OCA', 'ANDREANI', 'CORREO_ARG') NOT
    NULL,
    tipo ENUM('ESTANDAR', 'EXPRESS') NOT NULL,
    costo DECIMAL(10,2) NOT NULL CHECK (costo > 0),
    fechaDespacho DATE NOT NULL,
    fechaEstimada DATE NOT NULL,
    estado
ENUM('EN_PREPARACION','EN_TRANSITO','ENTREGADO') NOT
NULL,
```

```
eliminado BOOLEAN DEFAULT 0  
);
```

```
CREATE TABLE pedido (  
    id BIGINT PRIMARY KEY AUTO_INCREMENT,  
    numero VARCHAR(50) UNIQUE NOT NULL,  
    fecha DATE NOT NULL,  
    clienteNombre VARCHAR(100) NOT NULL,  
    total DECIMAL(10,2) NOT NULL CHECK (total > 0),  
    estado ENUM('NUEVO','FACTURADO','ENVIADO') NOT NULL,  
    envio BIGINT UNIQUE,  
    eliminado BOOLEAN DEFAULT 0,  
    FOREIGN KEY (envio) REFERENCES envio(id)  
);
```

6. Arquitectura del sistema en Java

El proyecto está organizado en paquetes:

- **config:** manejo de conexión JDBC
- **entities:** clases Pedido y Envio
- **dao:** acceso a datos (CRUD + búsquedas)
- **service:** lógica de negocio y transacciones
- **main:** menú principal y ejecución

Este enfoque respeta una arquitectura **en capas**, separando responsabilidades.

7. Diagrama UML del sistema

(Insertar capturas de los diagramas que ya generamos)

- *Diagrama de Clases*
- *Diagrama de Paquetes*
- *Diagrama de Secuencia (opcional pero recomendado)*

8. Desarrollo del sistema

8.1 Entidades

Las entidades representan objetos del dominio:

Envío:

- *atributos básicos + estado logístico*
- *eliminación lógica*
- *validaciones en el service*

Pedido:

- *representa la compra*
- *referencia 1:1 al envío*
- *validaciones en el service*

8.2 DAO

Los DAO implementan toda la interacción con MySQL mediante JDBC:

Incluyen:

- *crear()*
- *leer()*
- *leerTodos()*
- *actualizar()*
- *eliminar() (lógico)*

- búsquedas específicas
- y mapeo de *ResultSet* → objetos Java.

8.3 Servicios

La capa de servicios:

- Valida datos
- Controla transacciones
- Ejecuta lógica de negocio

La operación clave

crearPedidoConEnvio(pedido, envio)

Crea ambos objetos en una transacción:

- Si sale bien → commit
- Si algo falla → rollback

8.4 Menú principal

El menú orientado a consola permite:

- crear
- listar
- buscar
- actualizar
- eliminar

Incluye captura de errores para evitar caídas

9. Pruebas realizada

Se probaron todos los casos:

- ✓ Alta de Envío → ID autogenerado correcto
- ✓ Alta de Pedido + Envío → transacción OK

- ✓ **Listados completos**
- ✓ **Búsquedas por tracking y por ID**
- ✓ **Actualización de estados**
- ✓ **Eliminación lógica**
- ✓ **Integridad referencial**
- ✓ **Comportamiento ante datos inválidos**
- ✓ **Comprobación en MySQL Workbench**

Las pruebas funcionaron sin errores.

10. Capturas (instrucciones)

Insertar en el documento:

1. *Menú principal ejecutándose*
2. *Creación de envío*
3. *Creación de pedido + envío*
4. *Listado de envíos*
5. *Listado de pedidos*
6. *Búsqueda por tracking*
7. *Actualización*
8. *Eliminación*
9. *Base de datos en Workbench mostrando tablas*
10. *Consultas SQL mostrando resultados*

11. Conclusiones

El desarrollo del sistema permitió integrar conocimientos de:

- *Programación orientada a objetos*
- *Persistencia con JDBC*

- *Arquitectura en capas*
- *Manejo de transacciones*
- *Diseño de base de datos*
- *Validación y manejo de excepciones*

Se logró implementar un sistema funcional y completo, cumpliendo todas las consignas del integrador de Programación II.