

# Devolución final — Trabajo Integrador

**Grupo evaluado:** Grupo115\_Producto--CodigoBarras

**Materia:** Programación II

**Fecha:** 21/11/2025

**Docente evaluador/a:** Marcos Vega

## Devolución general

El proyecto demuestra un muy buen manejo de la arquitectura en capas, un uso adecuado de DAOs concretos, buen dominio de PreparedStatement y una correcta implementación del CRUD con soft delete. Además, destaca por el uso de JOIN y el manejo de valores NULL, algo que pocos grupos implementaron correctamente. Se observa un trabajo sólido, ordenado y con buenas prácticas en general.

## Fortalezas principales observadas

- **Arquitectura en capas bien implementada:** La estructura Models → DAO → Service → Main está claramente definida y se respeta a nivel conceptual.
- **JOIN y manejo de NULL:** Excelente uso de SQL JOIN y rs.wasNull(), algo destacable en esta entrega.
- **DAOs concretos completos:** Implementación correcta tanto para Producto como para CodigoBarras.
- **PreparedStatement y try-with-resources:** Buen uso de ambos en la mayoría de los métodos.
- **Servicios con lógica de negocio:** Se observan validaciones y operaciones transaccionales en ambas capas de servicio.
- **Soft delete implementado:** El uso del campo eliminado está correctamente aplicado en el CRUD.
- **Código ordenado y legible:** Buena modularidad, nombres claros y comentarios apropiados.

## Sugerencias para seguir mejorando

Estas recomendaciones apuntan a llevar el proyecto a un nivel más profesional:

- **SQL como constantes:** Actualmente las consultas están directamente en los métodos DAO. Centralizarlas como constantes mejora mantenibilidad y legibilidad.
- **Mayor abstracción:** El uso de GenericDao es correcto, pero incorporar una clase abstracta para DAOs reduciría duplicación de código.
- **Completar sobrescrituras:** Se implementa toString(), pero faltan equals() y hashCode(), necesarios para robustez en comparaciones y colecciones.
- **Transacciones más completas:** Algunas operaciones no están dentro de bloques transaccionales. Ideal unificar el manejo para mayor consistencia.
- **Validaciones y sanitización:** Las validaciones están bien encaminadas, pero falta sanitizar entradas para cubrir casos borde.
- **Integridad referencial en la base:** Aunque la lógica está en servicios, sería deseable modelar FK y restricciones directamente en la base de datos.
- **Excepciones:** En algunos métodos aún se usa printStackTrace(). Mejor reemplazar por excepciones más precisas o un logger.
- **Métodos largos:** Algunos métodos podrían dividirse para mejorar claridad.
- **Presentación mediante video:** Estuvo muy bien en líneas generales, pero utilizar el fondo del IDE blanco, tamaño de fuente mediano y alguna herramienta para nivelar el volumen de las distintas grabaciones hubiese enriquecido mejor el video en general.

## Mensaje final

El grupo realizó un trabajo muy sólido, con varios aspectos destacados como el manejo de JOIN, la arquitectura en capas y el uso adecuado de DAOs y servicios. Las sugerencias tienen como objetivo acompañar la mejora continua hacia un diseño más robusto y profesional. ¡Felicitaciones por el gran trabajo realizado!