
Inyección de Dependencias

Kenneth Rodriguez

Resumen

Este proyecto busca ejemplificar el uso y ventajas de la Inyección de Dependencias en Java para acoplar distintas clases de manera más indirecta.

Arquitectura del Sistema

El programa cuenta con 5 clases principales.

- Clase Negocio: Esta clase representa un negocio de e-commerce, que cuenta utiliza procesadores de pago y servicios de paquetería de terceros para cobrar y enviar pedidos, respectivamente.
- Clase ProcesadorPago: Esta clase representa distintos procesadores de pago digitales, como pueden ser PayPal, Stripe, Shopify, y demás herramientas.
- Clase ServicioPaqueteria: Esta clase representa servicios de paquetería física (como FedEx, DHL, Estafeta, etc.) utilizados para enviar las órdenes y pedidos del negocio.
- Clase Orden: Representa un pedido específico realizado al negocio. Incluye un id que lo representa, la dirección de envío, así como identificadores del estado de pago y de envío.
- Clase Inyector: Esta clase es la que se encargará de crear una instancia de Negocio que tenga un procesador de pago y servicio de paquetería de confianza. Esta clase será la que realice la inyección de dependencias mediante el constructor de Negocio.

Además de estas 5 clases principales, se realizaron múltiples clases hijo de ProcesadorPago y ServicioPaqueteria para ejemplificar distintos procesadores y servicios existentes. Además, el programa cuenta con 2 enumeradores, usados para determinar el tipo de pago y paquetería que el inyector utilizará.

Guía del Usuario

Las instancias del programa están hard-coded, por lo que no se modificarán o crearán por el usuario. Sin embargo, se puede modificar el código en Main para cambiar las órdenes a procesar, o alguna de las dependencias inyectadas.

Al iniciar el programa, la clase Inyector ejecuta un método estático para crear la instancia de Negocio a utilizar. Al método envíamos 3 argumentos: el nombre del negocio, el tipo de procesador de pago, y el tipo de servicio de paquetería. Estos 3 parámetros servirán para determinar las dependencias que serán inyectadas a la instancia de Negocio.

Cuando la instancia de Negocio crea una orden, debe recibir la dirección de envío. Posteriormente, puede realizar el procesamiento del pago y envío del paquete, en ese orden. Enviar el paquete antes de procesar el pago, pagar múltiples veces o enviar el mismo paquete varias veces resultará en mensaje de error, pero el programa continuará su ejecución hasta finalizar.

Conclusión

Antes de empezar con el tema, Inyección de Dependencias era un concepto del que desconocía por completo. Sin embargo, esto no me desanima de aprender del tema y realizar este proyecto para tratar de aplicar el concepto.

Esta técnica para desacoplar dependencias no solo me parece interesante, sino que completamente lógica, ya que permite separar las instancias e implementación para un código más limpio. Definitivamente es una estrategia que trataré de usar en mis proyectos (cuando sea útil, claro).