

PG\_R01\_ Proceso de refactorización- Fowler



20 de MAYO de 2024

Jon Ander Incera Moreno

Miguel José Gómez López



Contenido

[¿Cómo has utilizado en la práctica tus conocimientos de defectos de diseño? 2](#_Toc167022110)

[¿Cómo has utilizado en la práctica tus conocimientos de métricas de código? 3](#_Toc167022111)

[¿Cómo has utilizado en la práctica tus conocimientos de prueba y cobertura de pruebas? 4](#_Toc167022112)

[¿Se puede automatizar completamente el proceso de refactorización a través de herramientas? 5](#_Toc167022113)

[¿Qué relación encuentras entre el proceso de refactorización y la utilización de sistemas de control de tareas y versiones? 6](#_Toc167022114)

# ¿Cómo has utilizado en la práctica tus conocimientos de defectos de diseño?

He aplicado mis conocimientos de defectos de diseño identificando y corrigiendo code smells en el proyecto del videoclub. Utilicé el catálogo de Martin Fowler para detectar problemas como métodos excesivamente largos, código duplicado y otras ineficiencias. Implementé técnicas de refactorización como "Extract Method" para dividir métodos largos en otros más pequeños y específicos, y "Move Method" para reubicar lógica inapropiada a clases más adecuadas. Estas acciones mejoraron la legibilidad, mantenibilidad y modularidad del código.

# ¿Cómo has utilizado en la práctica tus conocimientos de métricas de código?

Utilicé métricas de código para evaluar la calidad del software antes y después de la refactorización. Calculé la complejidad ciclomática para medir la complejidad lógica del código y utilicé herramientas de análisis de cobertura para asegurar que las pruebas abarquen una alta proporción del código. Estas métricas permitieron cuantificar las mejoras y asegurar que las refactorizaciones no introdujeran nuevos errores.

# ¿Cómo has utilizado en la práctica tus conocimientos de prueba y cobertura de pruebas?

Apliqué mis conocimientos de prueba y cobertura asegurando que las modificaciones en el código estuvieran bien testeadas. Realicé pruebas unitarias y de integración antes y después de cada refactorización. Utilicé herramientas de análisis de cobertura para asegurar que el nivel de cobertura se mantuviera por encima del 80%, garantizando que las áreas críticas del código estuvieran adecuadamente verificadas.

# ¿Se puede automatizar completamente el proceso de refactorización a través de herramientas?

No, el proceso de refactorización no se puede automatizar completamente, ya que requiere juicio y conocimiento del contexto del código que las herramientas automáticas no pueden proporcionar. Sin embargo, herramientas como Eclipse ofrecen funcionalidades para ejecutar refactorizaciones comunes y registrar cambios, lo cual facilita y acelera el proceso.

# ¿Qué relación encuentras entre el proceso de refactorización y la utilización de sistemas de control de tareas y versiones?

El proceso de refactorización está estrechamente relacionado con el uso de sistemas de control de tareas y versiones. Cada refactorización debe asociarse con una issue específica para facilitar el seguimiento y la trazabilidad de los cambios. Utilizar ramas independientes para cada tarea de refactorización ayuda a gestionar y revisar los cambios de manera aislada, permitiendo una integración más segura y organizada del código refactorizado.

<https://github.com/Jonandermorenoo88/refactoring-fowler-example>