**Ejercicio Guiado 4**

Desarrollo de Software

Raúl Aguilar Arroyo, 100472050

Carlos Seguí Cabrera, 100472060

Grupo 3

11 feb 2023

ARTÍCULO 1 (Carlos Seguí Cabrera)

Título: Sistemas de información

Nombre de la revista/congreso donde se ha publicado: Congreso Nacional de Ingeniería en Informática

Dirección web: UCASAL-Sistema de Bibliotecas - Catálogo en línea y Repositorio Institucional

Autores: Cardacci, Dario Guillermo

Fecha de publicación: 2016 nov. 17

Resumen del artículo:

El último artículo visitado trata acerca de la reutilización de código. Un concepto común en el desarrollo de software actual debido a razones económicas, tiempos de desarrollo y código heredado. Cuando diferentes piezas de código desarrolladas con diferentes técnicas y paradigmas coexisten o han sido sometidas a muchos cambios por mantenimiento, pueden presentar características no deseadas para su integración. Por lo tanto, es natural tener una predisposición a refactorizar el código para hacerlo más legible y manejable. Esto implica modificar líneas internas de código, que aunque no deberían afectar la funcionalidad ofrecida, podrían alterar la complejidad lógica.

Este trabajo estudia cómo se comporta la complejidad lógica al aplicar refactorizaciones en piezas de código estructurado para representar la misma funcionalidad en un esquema orientado a objetos. Se analizan estructuras estándar de piezas de código estructurado y su resultado en un modelo orientado a objetos después de refactorizarlas. El estudio ha permitido determinar que la complejidad lógica se ve afectada no solo por la descomposición estructural promovida al refactorizar el código de un modelo a otro, sino también por decisiones de diseño que van más allá de mantener la funcionalidad intacta como propone la técnica de refactorización.

ARTÍCULO 2 (Raúl Aguilar Arroyo)

Título: Refactoring asistido mediante la detección de code smells

Nombre de la revista/congreso donde se ha publicado: SEDICI

Dirección web: [Refactoring asistido mediante la detección automática de code smells (unlp.edu.ar)](http://sedici.unlp.edu.ar/handle/10915/63701)

Autores: Castilla Matias

Fecha de publicación: Octubre 2014

Resumen del artículo:

La refactorización de código es un proceso que se lleva a cabo en tres fases.

Búsqueda de sección del sistema que necesite mejoras en su diseño.

Determinar qué técnica de refactorización es la más adecuada para solucionar el problema.

Transformación del código.

Aunque muchos entornos de desarrollo ofrecen herramientas para automatizar la refactorización, estas herramientas suelen centrarse en la última fase del proceso, es decir, en la transformación del código. La identificación de problemas y la elección de la técnica de refactorización adecuada no siempre están cubiertas por estas herramientas.

En este trabajo, se propone una solución para asistir al programador en todas las fases del proceso de refactorización. Esto incluye la detección automática de problemas de diseño, utilizando la idea de “code smell” y reglas “lint” para identificarlos. También se proponen técnicas de refactorización que podrían solucionar estos problemas y se lleva a cabo la transformación del código de manera interactiva.