



Universidad Carlos III
Grado en ingeniería informática
Curso Desarrollo de Software 2023-24
Práctica 3
Curso 2023-24

Fecha: **26/04/2024** - ENTREGA: 3

GRUPO: 81

Alumnos: **Mario Ramos Salsón (100495849), Pablo Amor Molina (100495855) y Víctor Martínez de las Heras (100495829)**

TÍTULO: Green software: Refactoring approach

DIRECCIÓN WEB: <https://n9.cl/z37sf>

AUTORES: Rajni Sehgal

FECHA DE PUBLICACIÓN: 7/07/2022

RESUMEN:

Hoy en día, necesitamos sistemas de información y comunicación que sean lo más eficientes energéticamente hablando. Las empresas de tecnología están tratando de reducir el consumo de energía tanto en el hardware como en el software. Dichas fugas, son causadas a menudo por el software complejo que realiza tareas innecesarias que consumen bastante más energía de la necesaria. La solución de este problema, está en manos de los investigadores que poco a poco van mejorando y optimizando dicho software. No obstante cuando este código se actualiza es bastante común que se introduzca código que se repite y que por consiguiente haga que el programa sea más lento y consume más energía. Para combatir este problema, hacemos uso de la refactorización. Esto consiste en limpiar y optimizar el código. Este estudio lo que quiere mostrar es cómo afecta este proceso al consumo de energía comparando un código refactorizado con uno que no está. Para concluir y como era de esperar, un código refactorizado y limpio es mucho más eficiente tanto energéticamente como a nivel de tiempo.

TÍTULO: An Empirical Evaluation of Refactoring

DIRECCIÓN WEB:

<https://www.e-informatyka.pl/index.php/einformatica/volumes/volume-2007/issue-1/article-2/>

AUTORES: Dirk Wilking, Umar Farooq Kahn, Stefan Kowalewski

FECHA DE PUBLICACIÓN: 22/04/2024

RESUMEN:

El mantenimiento del software se considera un factor de costo importante. Una solución propuesta para reducir este esfuerzo es la refactorización, que consiste en hacer modificaciones del código de acuerdo con las reglas de diseño predeterminadas y otras técnicas. La refactorización se integra perfectamente en el proceso de desarrollo existente y se ejecuta continuamente en segundo plano. Para comprobar los beneficios de la refactorización empíricamente, se ha hecho un experimento con 12 estudiantes que tenían que hacer el mismo programa en un tiempo de 3 meses. A algunos de ellos se les requería que cada 20 minutos intentarían refactorizar su código con diferentes técnicas. La diferencia de líneas entre los que habían hecho refactorización y los que no estaban entre 745 y 2214. Estudiando el uso de la memoria RAM podemos concluir que la refactorización tiene un impacto positivo en aspectos no funcionales como la escalabilidad, la modularidad, la reutilización, la complejidad, la mantenibilidad y la eficiencia. Del estudio también se descubrió que las mejores técnicas de refactorización son 'extraer método', 'renombrar método' y 'comentarios'.

TÍTULO: Toward Understanding the Impact of Refactoring on Program Comprehension

DIRECCIÓN WEB: <https://ieeexplore.ieee.org/abstract/document/9825885>

AUTORES: Giulia Sellitto, Emanuele Iannone, Zadia Codabux, Valentina Lenarduzzi,
Andrea De Lucia, Fabio Palomba, Filomena Ferrucci

FECHA DE PUBLICACIÓN: 21/07/ 2022

RESUMEN:

Este artículo fue seleccionado debido a su relevancia en el ámbito del desarrollo de software, particularmente en lo que respecta al proceso de refactorización, en torno a la que gira el desarrollo de nuestro trabajo grupal. La refactorización de software implica cambios en la estructura interna del código fuente sin alterar su comportamiento externo, y se debate ampliamente en la literatura sus implicaciones tanto beneficiosas como perjudiciales para la mantenibilidad del software, especialmente cuando se realiza sin el respaldo de herramientas automatizadas. El estudio aborda la dimensión de la comprensión del programa, evaluando específicamente la legibilidad del código como una unidad básica de comprensión. A través de una investigación empírica a gran escala en 156 proyectos de código abierto, se cuantifica el impacto de la refactorización en la legibilidad del programa. Los resultados clave revelan que la refactorización tiene un impacto notable en la mayoría de las métricas de legibilidad consideradas, lo que resalta la importancia de entender cómo estos cambios pueden afectar la comprensión del código por parte de los desarrolladores. Este estudio proporciona una contribución significativa al campo al proporcionar evidencia empírica sobre los efectos de la refactorización en la comprensión del programa, lo que puede informar prácticas de desarrollo más efectivas y herramientas de apoyo.