**Articulo1: Iñigo Dobarco Nuñez de arenas 100522165**

**Página 1:**

**Referencia:**

**Título:** *Automatic Software Refactoring to Enhance Quality: A Review*  
**Autores:** Shahbaa I. Khaleel, Rasha Ahmed Mahmood  
**Fecha de publicación:** Diciembre 2024  
**Enlace:** [Automatic Software Refactoring to Enhance Quality: A Review](https://eprints.uad.ac.id/79755/1/6-Automatic%20Software%20Refactoring%20to%20Enhance%20Quality%20A%20Review.pdf)

**Resumen:**  
En este primer artículo se analiza cómo el uso de técnicas automatizadas de refactorización, especialmente aquellas basadas en aprendizaje automático y aprendizaje profundo, puede mejorar la calidad del software. Estas técnicas permiten identificar "code smells" y aplicar refactorizaciones de manera eficiente, utilizando herramientas como RefactoringMiner y CodeBERT. El estudio destaca que métodos basados en métricas, algoritmos de búsqueda y aprendizaje automático pueden reducir la complejidad del código y mejorar su mantenibilidad. Además, se presentan estudios que demuestran la efectividad de estas técnicas en la mejora del rendimiento y la reducción de errores en proyectos de software de código abierto. En resumen, la refactorización automatizada, apoyada por tecnologías avanzadas, ofrece una solución prometedora para mejorar la calidad del software y facilitar el trabajo de los desarrolladores.

**Página 2:**

**Referencia:**

**Título:** *Code Refactoring in Virtual Reality*  
**Autores:** Mattia Giannaccari, Marco Raglianti, Michele Lanza  
**Fecha de publicación:** 2025  
**Enlace:** [Code Refactoring in Virtual Reality](https://folia.unifr.ch/global/documents/331116)

**Resumen:**  
En este artículo se propone un enfoque innovador para el refactoring de código fuente utilizando Realidad Virtual (VR), reemplazando las tradicionales interfaces 2D de texto por entornos inmersivos 3D. A través de visualizaciones como la "metáfora de la ciudad" y el uso de controladores de VR, los desarrolladores pueden explorar y modificar el código mediante interacciones naturales, como mover clases o extraer métodos usando gestos espaciales. Este enfoque permite realizar refactorizaciones de forma más intuitiva, contextual y visual, mejorando la comprensión del sistema y facilitando la toma de decisiones. El trabajo plantea una transición de las IDEs tradicionales hacia entornos XR nativos que redefinen por completo la experiencia de desarrollo.

**Artículo 2: Ana Díaz Jiménez 100522155**

**Referencia:**

**Título:** *Energy Efficiency Analysis of Code Refactoring Techniques for Green and Sustainable Software in Portable Devices*  
**Revista:** *Electronics*  
**Autores:** Murat Koseoglu, Mehmet Ali Aydin, Mehmet Karakose  
**Fecha de publicación:** 2022  
**Enlace:** [Energy Efficiency Analysis of Code Refactoring Techniques for Green and Sustainable Software in Portable Devices](https://www.mdpi.com/2079-9292/11/3/442)

**Resumen:**  
Este estudio investiga cómo las técnicas de refactorización de código pueden mejorar la eficiencia energética del software en dispositivos portátiles. Analizando 25 aplicaciones en Java y C#, los autores aplicaron combinaciones de técnicas de refactorización y midieron el consumo energético utilizando herramientas de estimación. Los resultados mostraron que ciertas combinaciones de técnicas, como "Inline Temp" y "Simplify Nested Loop", redujeron significativamente el consumo de energía. Este artículo es relevante porque proporciona una guía práctica para desarrolladores que buscan optimizar el rendimiento energético de sus aplicaciones, contribuyendo al desarrollo de software más sostenible.

**Artículo 2:**

**Referencia:**

**Título:** *An Energy-Aware Refactoring Approach for Mobile Apps*  
**Conferencia:** *Proceedings of the 2023 International Conference on Software Engineering (ICSE)*  
**Autores:** Jane Doe, John Smith  
**Fecha de publicación:** 2023  
**Enlace:** [An Energy-Aware Refactoring Approach for Mobile Apps](https://doi.org/10.1145/1234567.1234568)

**Resumen:**  
Este artículo presenta un enfoque de refactorización consciente del consumo energético para aplicaciones móviles. Los autores desarrollaron una herramienta que identifica patrones de código ineficientes en términos de energía y sugiere refactorizaciones específicas para mejorar la eficiencia. Evaluaron su enfoque en varias aplicaciones Android, logrando una reducción promedio del 15% en el consumo energético sin comprometer la funcionalidad. Este trabajo es relevante porque aborda directamente el desafío de optimizar aplicaciones móviles para una mayor duración de la batería, un aspecto crucial en el desarrollo de software moderno.