

Artículos de refactorización

ALUMNOS: Diego Alonso Herreruella (100432074@alumnos.uc3m.es)

Ignacio Marín de la Bárcena Pérez (100432039@alumnos.uc3m.es)

Artículo 1

TÍTULO: HARP: Holistic Analysis for Refactoring Python-Based Analytics Programs

NOMBRE DE LA REVISTA: IEEE

DIRECCIÓN WEB: <https://ieeexplore.ieee.org/abstract/document/9284016/authors#authors>

AUTORES: Weijie Zhou, Yue Zhao, Guoqiang Zhang y Xipeng Shen.

FECHA DE PUBLICACIÓN: 21 de diciembre de 2020

FECHA CONFERENCIA: 5 de octubre de 2020

RESUMEN:

Se nos presenta HARP, una herramienta de refactorización que permite a los usuarios optimizar su código. Esta herramienta, se emplea en aquellos programas dónde se emplean distintas librerías y código para dar cohesión entre las mismas para la creación de computación gráfica. Los programas de computación gráfica escritos en Python se componen de una serie de complejidades y barreras que complican la refactorización. Es por ello que se decidió crear un asistente que ayudará a la refactorización en dicho programas, HARP. Este asistente, es capaz de analizar el código, extraer el que realmente es importante unificando de manera estructurada (tanto el escrito como el proporcionado en las librerías), para así realizar innumerables sugerencias para su optimización aunque puede que no todas sean válidas. Se nos indica con qué plataformas es compatible HARP siendo las más destacadas Pytorch, Pycharm por medio de IntelliJ y TensorFlow. Y cómo esta herramienta ha sido probada por medio de distintos experimentos, siendo el más destacable para nuestro proyecto aquel relacionado con la refactorización. Experimento en el que 12 programadores expertos desarrollaron un código y aplicaban HARP a la hora de optimizarlo y refactorizarlo. Obteniéndose unos resultados favorables para HARP ya que todos los programadores coincidieron en la ayuda de las sugerencias que éste realiza y de cómo optimizar el código empleado. Finalmente, se nos exponen varios proyectos relacionados con el uso de HARP, y una conclusión de porqué se debería usar HARP para optimizar programas de computación gráfica.

Artículo 2

TÍTULO: PYREF: Refactoring Detection in Python Projects

NOMBRE DE LA REVISTA: IEEE

DIRECCIÓN WEB: <https://ieeexplore.ieee.org/abstract/document/9610674/authors#authors>

AUTORES: Hassan Atwi, Bin Lin, Nikolaos Tsantalos, Yutaro Kashiwa, Yasutaka Kamei, Naoyasu Ubayashi, Gabriele Bavota y Michele Lanza

FECHA DE PUBLICACIÓN: 24 de noviembre de 2021

FECHA CONFERENCIA: 27-28 de septiembre de 2021

RESUMEN:

El documento leído trata sobre PYREF, una herramienta para programadores cuya función consiste en identificar los cambios de refactoring que se han realizado en un programa. Este

tipo de herramientas ya se utilizan para otros lenguajes, pero no hay ninguna bien establecida para Python. PYREF utiliza un algoritmo similar al de otras herramientas parecidas, basado en extraer los cambios realizados, convertirlos a un formato más fácilmente legible (un tipo especial de árbol), compararlos con posibles cambios de refactoring mediante heurística, y finalmente asignarlos. Cuando se evaluó su rendimiento comparándolo frente a otras herramientas para Python (aún en desarrollo también). Se obtuvieron unos resultados bastante positivos, ya que, para los mismos programas de prueba, PYREF consiguió detectar más operaciones de refactoring que la otra herramienta. Sin embargo, los resultados variaron dependiendo del método exacto identificado, con una precisión bastante baja en dos de ellos, debido a un bug encontrado. Tras la evaluación, se propuso que la precisión de la herramienta, aunque buena, puede ser aún mejor. También se encontraron una serie de posibles amenazas, debidas principalmente a errores humanos, la falta de situaciones donde probar PYREF, o errores propios de las herramientas en las que se basas. También se mencionan las diferentes limitaciones de PYREF, como que solo se detecta el refactoring en métodos y no en clases; que depende de un módulo externo para su funcionamiento, lo que puede producir errores dependiendo de la compatibilidad de versiones; o que se pueden generar problemas en caso de que una operación produzca varias revisiones. Finalmente, se ofrece un pequeño resumen, presentando PYREF como nueva herramienta de detección de refactoring.