



Framework para refactorización de código Java

Árbol de llamadas y métricas para análisis de código

Facultad de Informática UPV/EHU

Ingeniería del Software

Autor:
Iñaki García Noya
Directora:
Maider Azanza Sese

Introducción

El **mantenimiento del software** es una actividad esencial para cualquier sistema de software. Entre el 60% y el 90% de los costes del *software* están relacionados con las actividades de mantenimiento. Este mantenimiento del *software* se puede prolongar si no se identifican los **code smells**, violaciones de los principios del diseño del código. Para eliminar estos *code smell* se utilizan diferentes mecanismos de *refactoring*.

Contexto

Este TFG se ha realizado con la empresa **INDABA CONSULTORES S.L.**, la cual realiza muchas tareas de modificación de código para gestionar la calidad de aplicaciones. Para encontrar oportunidades de *refactoring* dentro de una clase; como por ejemplo extraer un método de una clase, realizaban una tabla introduciendo de manera manual las llamadas que realizaban los métodos entre ellos y sus métricas. El objetivo es **automatizar** este proceso con el fin de reducir el tiempo y los posibles errores que conlleva el proceso manual.

Objetivos principales

1. Analizar las herramientas existentes para refactoring de código.
2. Analizar herramientas existentes para obtener métricas y las llamadas entre clases y métodos. Teniendo como entrada una clase o un conjunto de clases, se obtendrán las relaciones clase-clase y método-método(s).
3. Realizar los análisis con filtros personalizados (p. ej.: paquetes).
4. Exportar los resultados en diferentes ficheros, con la posibilidad de disponer diferentes formatos.
5. Realizar pruebas exhaustivas para comprobar el resultado del análisis.

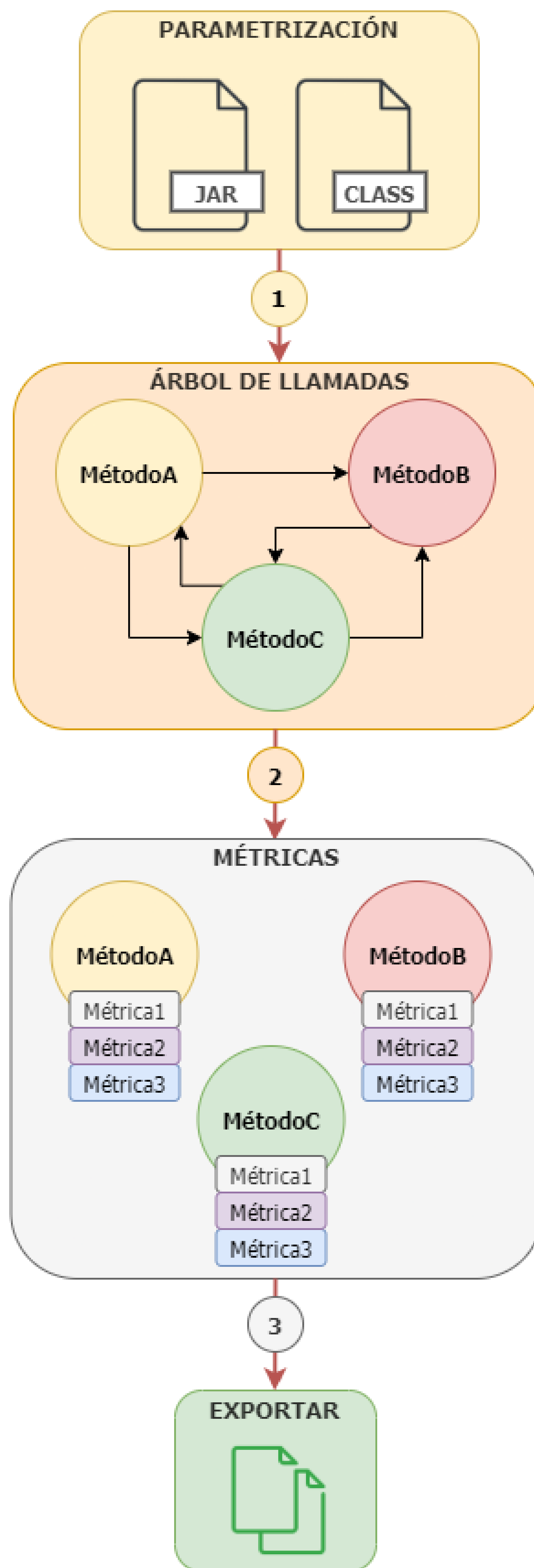
Solución

El programa realiza el análisis siguiendo los siguientes pasos:

1. Analiza los parámetros de entrada (ficheros comprimidos y código fuente) para obtener el árbol de llamadas.
2. Calcula las métricas de los métodos y clases.
3. Exporta los resultados del análisis en diferentes ficheros. Estos ficheros pueden ser en distintos formatos.

Resultados

Se ha realizado un programa que sigue los requisitos definidos por la empresa. La herramienta se ha probado con múltiples proyectos, gracias a una fase de pruebas exhaustivas. Por ello, se puede asegurar que la herramienta construye un árbol de llamadas bien formado, con métricas exactas y con un buen rendimiento.



Bibliografía principal

- [1] T. Honglei, S. Wei, and Z. Yanan. The Research on Software Metrics and Software Complexity Metrics. 2009.
- [2] J. Jász., I. Siket., E. Pengő., Z. Ságodi., and R. Ferenc. Systematic comparison of six open-source java call graph construction tools. 2019.
- [3] Lacerda, Guilherme, Petrillo, Fabio, Pimenta, Marcelo, Guéhéneuc, and Yann-Gaël. Code smells and refactoring: A tertiary systematic review of challenges and observations. 2020.