



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática



**TFG del Grado en Ingeniería
Informática**

**Comparador de métricas de
evolución en repositorios
software**



Presentado por Miguel Ángel León Bardavío
en Universidad de Burgos — 19 de noviembre
de 2018

Tutor: Carlos López Nozal



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática



D. Carlos López Nozal, profesor del departamento de nombre departamento, área de nombre área.

Expone:

Que el alumno D. Miguel Ángel León Bardavío, con DNI 71362165L, ha realizado el Trabajo final de Grado en Ingeniería Informática titulado "Comparador de métricas de evolución en repositorios software" de TFG.

Y que dicho trabajo ha sido realizado por el alumno bajo la dirección del que suscribe, en virtud de lo cual se autoriza su presentación y defensa.

En Burgos, 19 de noviembre de 2018

Vº. Bº. del Tutor:

D. Carlos López Nozal

Resumen

Aplicación Web en Java que toma como entrada un conjunto de direcciones de repositorios públicos o privados y calcula medidas de la evolución que permiten comparar los repositorios.

Descriptores

gitlab, proyectos, repositorios, métricas, análisis, comparador, software

Abstract

Java Web Application that takes a set of addresses of public or private repositories and calculates measures of evolution that allow comparing the repositories.

Keywords

gitlab, projects, repositories, metrics, analysis, comparator, software

Índice general

Índice general	III
Índice de figuras	IV
Índice de tablas	V
Introducción	1
Objetivos del proyecto	3
2.1. Requisitos	3
Conceptos teóricos	5
Técnicas y herramientas	7
4.1. Herramientas utilizadas	7
Aspectos relevantes del desarrollo del proyecto	9
5.1. Definición de métricas	9
Trabajos relacionados	15
Conclusiones y Líneas de trabajo futuras	17
Bibliografía	19

Índice de figuras

Índice de tablas

Introducción

El trabajo se centra en crear una aplicación web que permita, a partir de URLs de repositorios GitLab, comparar dos o varios repositorios mediante métricas de evolución. Las métricas que se van a trabajar, en principio, son:

- Número total de issues (I1)
- Commits por issue (I2)
- Porcentaje de issues cerradas (I3)
- Media de días en cerrar una issue (TI1)
- Media de días entre commits (TC1)
- Días que han pasado entre el primer y último commit (TC2)
- Ratio de actividad de commits por mes (TC3)
- Número de commits en el mes pico (C1)

Se pretende que el proyecto pueda ser ampliado a más gestores de repositorios como GitHub o Bitbucket y que pueda calcular más métricas de las que originalmente se han trabajado, por tanto deberá presentar un diseño que facilite el mantenimiento de la misma y añadir nuevas funcionalidades.

Objetivos del proyecto

2.1. Requisitos

Requisitos funcionales

Requisitos no funcionales

Conceptos teóricos

Técnicas y herramientas

4.1. Herramientas utilizadas

- Eclipse IDE for Java EE Developers
- Apache Maven 3.5.4
- Apache Tomcat 9.0.12
- Java SE 11 (JDK)
- java-gitlab-api
- gitlab4j-api

Aspectos relevantes del desarrollo del proyecto

5.1. Definición de métricas

I1 - Número total de issues

- **Categoría:** Proceso de Orientación
- **Nombre:** NumeroTotalIssues
- **Descripción:** Número total de issues creadas en el repositorio
- **Propósito:** ¿Cuántas issues se han definido en el repositorio?
- **Fórmula:** NTI. $NTI = \text{número total de issues}$
- **Fuente de medición:** Repositorio de un gestor de repositorios
- **Interpretación:** $NTI \geq 0$. Mejor valores bajos
- **Tipo de escala:** Absoluta
- **Tipo de medida:** $NTI = \text{Contador}$

I2 - Commits por issue

- **Categoría:** Proceso de Orientación
- **Nombre:** CommitsPorIssue

- **Descripción:** Número de commits por issue
- **Propósito:** ¿Cuántos commits realizados por cada issue?
- **Fórmula:** $CI = NTC/NTI$. NTI = Numero total de issues, NTC = Número total de commits
- **Fuente de medición:** Repositorio de un gestor de repositorios
- **Interpretación:** $CI \geq 0$, Si se acerca a 1 se definen bien las issues, si alto: no se definen bien las issues, si bajo: desarrollo del proyecto lento
- **Tipo de escala:** Ratio
- **Tipo de medida:** NTI, NTC = Contador

I3 - Porcentaje de issues cerradas

- **Categoría:** Proceso de Orientación
- **Nombre:** PorcentajeIssuesCerradas
- **Descripción:** Porcentaje de issues cerradas
- **Propósito:** ¿Qué porcentaje de issues definidas en el repositorio se han cerrado?
- **Fórmula:** $PIC = NTIC*100/NTI$. NTIC = Número total de issues cerradas, NTI = Numero total de issues
- **Fuente de medición:** Repositorio de un gestor de repositorios
- **Interpretación:** $0 \leq PIC \leq 100$. Cuanto más alto mejor
- **Tipo de escala:** Ratio
- **Tipo de medida:** NTI, NTIC = Contador

TI1 - Media de días en cerrar una issue

- **Categoría:** Proceso de Orientación
- **Nombre:** PorcentajeIssuesCerradas
- **Descripción:** Porcentaje de issues cerradas

- **Propósito:** ¿Qué porcentaje de issues definidas en el repositorio se han cerrado?
- **Fórmula:** $PIC = NTIC * 100 / NTI$. NTIC = Número total de issues cerradas, NTI = Numero total de issues
- **Fuente de medición:** Repositorio de un gestor de repositorios
- **Interpretación:** $0 \leq PIC \leq 100$. Cuanto más alto mejor.
- **Tipo de escala:** Ratio
- **Tipo de medida:** NTI, NTIC = Contador

TC1 - Media de días entre commits

- **Categoría:** Constantes de tiempo
- **Nombre:** MediaDiasEntreCommits
- **Descripción:** Media de días que pasan entre dos commits consecutivos
- **Propósito:** ¿Cuánto tiempo suele pasar desde un commit hasta el siguiente?
- **Fórmula:** $MDEC = [\text{Sumatorio de } (TC_i - TC_j) \text{ desde } i=1, j=0 \text{ hasta } i=NTC] / NTC$. NTC = Número total de commits, TC = Tiempo de Commit
- **Fuente de medición:** Repositorio de un gestor de repositorios
- **Interpretación:** $MDEC > 0$. Cuanto más pequeño mejor.
- **Tipo de escala:** Ratio
- **Tipo de medida:** NTC = Contador; TC = Tiempo

TC2 - Días entre primer y último commit

- **Categoría:** Constantes de tiempo
- **Nombre:** DiasEntrePrimerYUltimoCommit
- **Descripción:** Días transcurridos entre el primer y el ultimo commit

- **Propósito:** ¿Cuántos días han pasado entre el primer y el último commit?
- **Fórmula:** $DEPUC = TC2 - TC1$. $TC2$ = Tiempo de último commit, $TC1$ = Tiempo de primer commit.
- **Fuente de medición:** Repositorio de un gestor de repositorios
- **Interpretación:** $DEPUC \geq 0$
- **Tipo de escala:** Absoluta
- **Tipo de medida:** TC = Tiempo

TC3 - Ratio de actividad de commits por mes

- **Categoría:** Constantes de tiempo
- **Nombre:** RatioCommitPorMes
- **Descripción:** Muestra el número de commits relativos al número de meses
- **Propósito:** ¿Cuál es el número medio de cambios por mes?
- **Fórmula:** $RCM = NTC / 12$
- **Fuente de medición:** Repositorio de un gestor de repositorios
- **Interpretación:** $RCM > 0$. Cuanto más alto mejor
- **Tipo de escala:** Ratio
- **Tipo de medida:** NTC = Contador

C1 - Número de commits en el mes pico

- **Categoría:** Constantes de tiempo
- **Nombre:** ContadorCommitsPico
- **Descripción:** Número de commits en el mes que más commits se han realizado en relación con el número total de commits
- **Propósito:** ¿Cuál es la proporción de trabajo realizado en el mes con mayor número de cambios?

- **Fórmula:** $CCP = NCMP / NTC$. $NCMP$ = Número de commits en el mes pico, NTC = Número total de commits
- **Fuente de medición:** Repositorio de un gestor de repositorios
- **Interpretación:** $0 \leq CCP \leq 1$. Mejor valores intermedios
- **Tipo de escala:** Ratio
- **Tipo de medida:** $NCMP, NTC$ = Contador

Trabajos relacionados

- [Activiti-API](#)
- [Soporte de Métricas con Independencia del Lenguaje para la Inferencia de Refactorizaciones](#)
- [Software Project Assessment in the Course of Evolution - Jacek Ratzinger](#)

Conclusiones y Líneas de trabajo futuras

Bibliografía

- [1] John R. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, 1992.
- [2] Wikipedia. Latex — wikipedia, la enciclopedia libre, 2015. [Internet; descargado 30-septiembre-2015].