

UNIVERSIDAD DE BURGOS

ESCUELA POLITÉCNICA SUPERIOR



Grado en Ingeniería Informática

Monitorización del proceso de pruebas con Git-Travis-Codecov

Medición y Calidad

Alumnos	Antonio de los Mozos Alonso Miguel Ángel León Bardavío
Tutor	Carlos López Nozal DEPARTAMENTO DE INGENIERÍA CIVIL Área de Lenguajes y Sistemas Informáticos

Burgos, 25 de febrero de 2018



Este documento está licenciado bajo [Creative Commons Attribution 3.0 License](https://creativecommons.org/licenses/by/3.0/)

Índice de contenido

- 1 Preguntas3
 - 1.1¿Se ha realizado trabajo en equipo?.....3
 - 1.2¿Tiene calidad el conjunto de pruebas disponibles?.....5
 - 1.3¿Cuál es el esfuerzo invertido en realizar la actividad?.....5
 - 1.4¿Cuál es el número de fallos encontrados en el código?.....6

Índice de ilustraciones

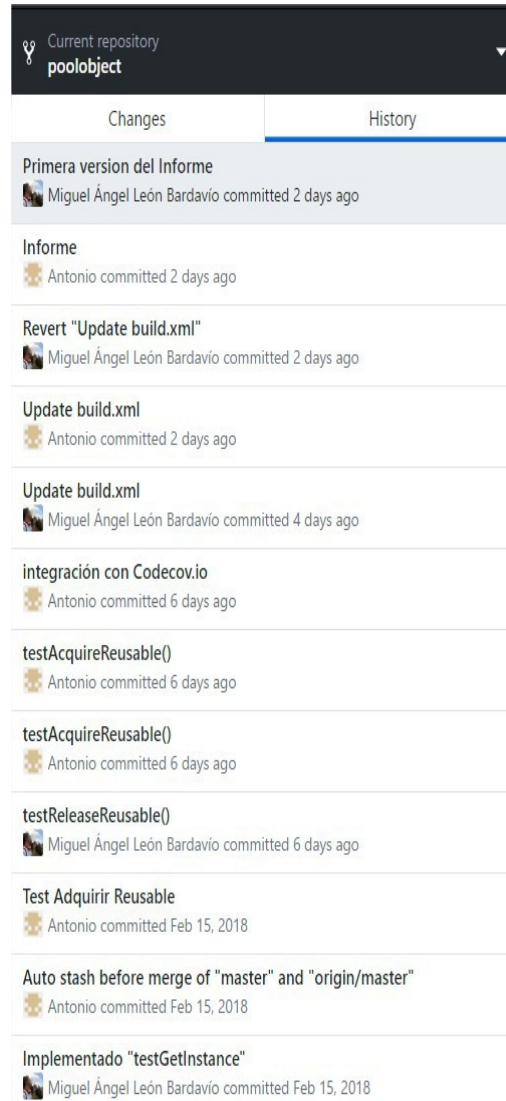
Ilustración 1: Listado con commits realizados. Obtenido con “GitHub Desktop”	3
Ilustración 2: Contribuciones al proyecto.....	4
Ilustración 3: Cobertura de líneas y de complejidad del proyecto. Usando la herramienta "Codecov"	5
Ilustración 4: Cobertura de líneas y de complejidad de las distintas clases del proyecto. Obtenido con "Codecov"	5

Índice de tablas

1 PREGUNTAS

1.1 ¿Se ha realizado trabajo en equipo?

Si, se ha repartido el trabajo entre los integrantes del grupo. El primer commit realizado tiene el título de 'Implementado "testGetInstance"' y se ha intentado alternar los commits entre un integrante del grupo y otro.



*Ilustración 1: Listado con commits realizados.
Obtenido con "GitHub Desktop"*

En la Ilustración 1 se puede observar los commits realizados por los dos integrantes del grupo.

En la Ilustración 2 se puede apreciar las contribuciones realizadas al proyecto. Se puede apreciar que ama0114 ha realizado 10 commits añadiendo 817 líneas de código y eliminando 7. Mientras que mlb0029 ha realizado 5 commits añadiendo 27 líneas de código y eliminando 9. La diferencia en commits se debe a que puede haber varios commits realizados para el mismo propósito y la diferencia en líneas de código se debe principalmente a las 782 adiciones de líneas en el fichero “codecov.yml” que realizó ama0114 en el commit titulado “integración con Codecov.io”.

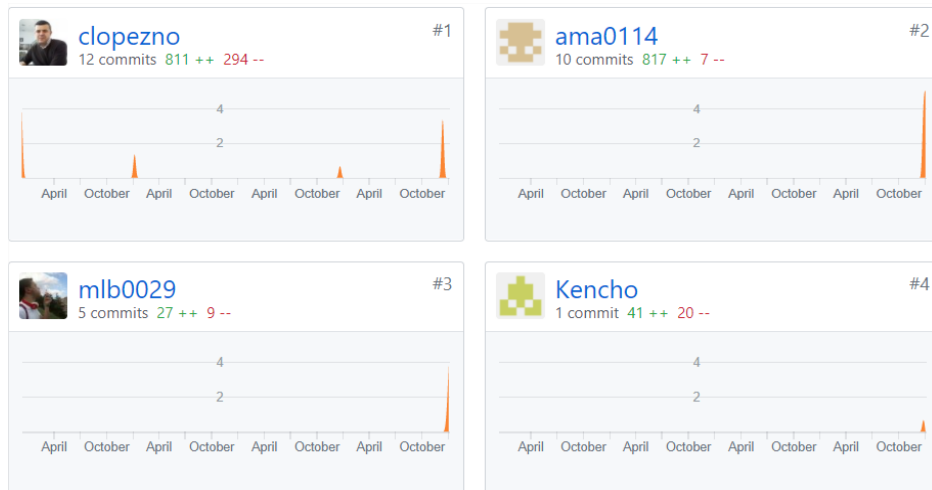


Ilustración 2: Contribuciones al proyecto

Por tanto se puede observar que el 100% de los integrantes del grupo han participado en el proyecto tanto en la Ilustración 1 como en la Ilustración 2.

1.2 ¿Tiene calidad el conjunto de pruebas disponibles?

Las pruebas cubren el 75 % del total del código del proyecto como se muestra en la Ilustración 3. Esto se debe a que hay parte del código que no se ha podido probar debido a que no se sabe la funcionalidad. Nos referimos al método “main” de la clase “Client” que no está documentado ni se explica su funcionalidad por lo que no se pueden crear los casos de prueba adecuados. Solo sabemos de la clase “Client” que usa objetos del tipo “Reusable”.

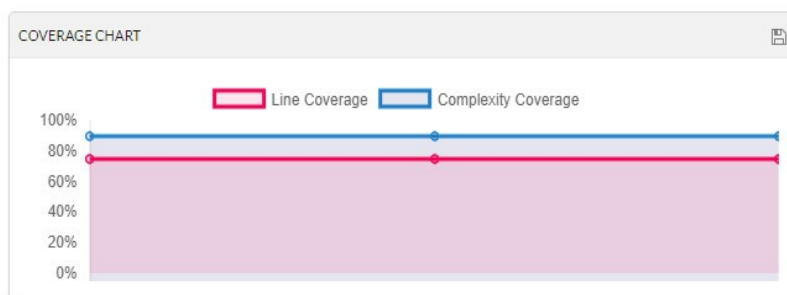


Ilustración 3: Cobertura de líneas y de complejidad del proyecto. Usando la herramienta "Codecov"

El resto de clases tienen una cobertura del 100% tanto de líneas como de complejidad como se muestra en la Ilustración 4.

Files	Complexity	Coverage
Client.java	0.00%	0.00%
DuplicatedInstanceException.java	100.00%	100.00%
NotFreeInstanceException.java	100.00%	100.00%
Reusable.java	100.00%	100.00%
ReusablePool.java	100.00%	100.00%

Ilustración 4: Cobertura de líneas y de complejidad de las distintas clases del proyecto. Obtenido con "Codecov"

1.3 ¿Cuál es el esfuerzo invertido en realizar la actividad?

En cuanto a tiempo ha llevado dos sesiones de practicas (4 horas).

Y en cuanto al esfuerzo se puede observar en la Ilustración 2 que el grupo de prácticas ha realizado 15 commits en total, añadiendo 844 líneas de código y eliminando 16.

1.4 ¿Cuál es el número de fallos encontrados en el código?

Se han encontrado 3 fallos externos al código fuente, son más bien errores de integración con las herramientas Travis y Codecov.

- El fichero “Build.xml” ha tenido que ser modificado para la correcta ejecución de los test y se ha modificado la propiedad de junit “haltonfailure” a “yes”.
- Se ha tenido que añadir el fichero “codecov.yml”
- La clase “Client” no está documentada.

A parte de estos fallos, en el código fuente no se ha encontrado ningún fallo con los test realizados.