

# Sistemas y Tecnologías Web Aplicadas

## Shell para corrección automática de repositorios de GitHub

Autor: Juan José Labrador González  
Director: Casiano Rodríguez León

Escuela Superior de Ingeniería y Tecnología  
Departamento de Ingeniería Informática y de Sistemas  
Universidad de La Laguna

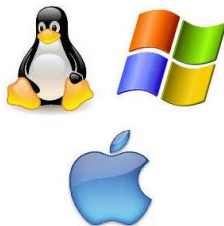
14 de julio de 2017

- 1 Introducción
- 2 Estado del arte y motivación
- 3 Perfil de Usuario
- 4 Objetivos
- 5 Tecnología usada
- 6 Metodología de desarrollo
- 7 Resultados
  - Funcionalidades requeridas
  - Funcionalidades extra
- 8 Conclusiones y Trabajos Futuros/Conclusions and Future Work
- 9 Bibliografía

Este Trabajo de Fin de Máster consistió en la creación de una herramienta de línea de comandos que permite la automatización de tareas sobre repositorios de GitHub.

## Características:

- Escrita en **Node.js**
- Distribuida mediante un **paquete NPM**.
- Multiplataforma: Linux, macOS, Windows.
- Uso sencillo.



## Sistemas de Control de Versiones

- Indispensables en la Ingeniería del Software.
- Constante crecimiento en número y funcionalidades.
- Fomento de la colaboración.



## GitHub

- Plataforma de desarrollo colaborativo consolidada.



# Estado del arte y motivación III

- Usa **Git** como sistema de control de versiones.
- Fuerte apuesta por la educación:



**GitHub**  
Student Developer Pack



## Motivación

- Imposibilidad de realizar operaciones masivas sobre repositorios.
- Ausencia de herramientas externas para tal fin.
- Mejor aprovechamiento de las herramientas educativas.
- Simplificación del trabajo del docente.
- Ahorro considerable de tiempo en tareas repetitivas.

## Ejemplo de caso real

- Asignatura cuatrimestral con 10 prácticas evaluables.
- Total de alumnos matriculados: 50.
- Total de prácticas por alumno: 10.

$$10 \times 50 = \mathbf{500 \text{ repositorios}}$$





# Perfil de Usuario

La herramienta desarrollada está principalmente orientada hacia un perfil de profesor concreto:

- Docente de alguna rama de Ingeniería Informática.
- Con conocimientos avanzados en:
  - Programación.
  - Herramientas de control de versiones.



# Objetivos

- Revisión bibliográfica y consulta del estado del arte.
- Desarrollo de una herramienta de línea de comandos escrita en **Node.js** que permita:
  - Autenticación de usuarios.
  - Listar organizaciones, asignaciones y repositorios del usuario.
  - Automatizar la descarga de repositorios y la ejecución de scripts en los mismos:
    - TDD.
    - Creación de entorno.
    - Evaluación.
  - Presentación de resultados al usuario:
    - PDF.
    - HTML.





Travis CI

## Metodología **ágil**:

- Reuniones periódicas estableciendo iteraciones cortas.
- Desarrollo y presentación de resultados y prototipos tras cada iteración.
- Solución de problemas e incorporación de nuevas características.

## GitHub

- Control de versiones usando *branching*.
- Gestión de incidencias y mejoras usando *issues*.



## Travis-CI

- Control de despliegues.



Travis CI

# Resultados

- Autenticación con GitHub
- Listar organizaciones, asignaciones y repositorios de GitHub del usuario
- Automatizar la descarga de repositorios
- Automatizar la ejecución de scripts en los repositorios
- Recopilar y presentar la información obtenida de la automatización de tareas

Funcionalidades que, a pesar de no ser requeridas, brindan al usuario de una mejor experiencia de uso del programa:

- Autocompletado de los comandos. Pull Request.
- Opción de ayuda.
- Opción de visualizar el directorio de trabajo actual
- Opción para conocer el propietario de cada repositorio



# Funcionalidades extra II

# Conclusiones y Trabajos Futuros/Conclusions and Future Work I

- Esta herramienta pretende ser un complemento para....
- a
- a

# Conclusiones y Trabajos Futuros/Conclusions and Future Work II

## Trabajos Futuros:

- Generar scripts en Bash para evaluar aplicaciones (instalación de dependencias, comprobación de calidad de código y ejecución de tests) en varios lenguajes: Node.js, C++, Ruby, Python, etc.
- Dar soporte a la ejecución de scripts escritos en otros lenguajes: Ruby, Python...
- Generar issues en cada repositorio con los resultados de los scripts que se ejecuten.
- Crear ramas en cada repositorio con los resultados de los scripts.

# Conclusiones y Trabajos Futuros/Conclusions and Future Work III

- This tool intends to ...
- It uses ....
- On the other hand...

# Conclusiones y Trabajos Futuros/Conclusions and Future Work IV

## Future Work:

- a
- a
- a
- a



“node-github NPM.”

<https://github.com/mikedeboer/node-github>.



“node-github API.”

<https://mikedeboer.github.io/node-github/>.



“Node.js API.”

<https://nodejs.org/api/>.



“ghshell NPM.”

<https://www.npmjs.com/package/ghshell>.



“simple-barrier.”

<https://www.npmjs.com/package/simple-barrier>.



“async.”

<https://github.com/caolan/async>.



“Asynchronous Patterns.”

<https://mostafa-samir.github.io/async-iterative-patterns->



“Using async/await.”

<https://blog.readme.io/using-async-await-in-node-js-7-6-0/>

Gracias por su atención