



# Trabajo de Fin de Máster Sistemas y Tecnologías Web Aplicadas (SyTWA)

---

## Shell para corrección automática de repositorios de GitHub

*CLI tool for automatic correction of GitHub's  
repositories .*

Juan José Labrador González

---

La Laguna, 30 de junio de 2017

D. **Casiano Rodríguez León**, con DNI número 42.020.072-S profesor Titular de Universidad adscrito al Departamento de Ingeniería Informática y de Sistemas de la Universidad de La Laguna, como tutor

## **C E R T I F I C A**

Que la presente memoria titulada:

*“Sistemas y Tecnologías Web Aplicadas. Shell para corrección automática de repositorios de GitHub.”*

ha sido realizada bajo su dirección por D. **Juan José Labrador González**, con DNI número 78.729.778-L.

Y para que así conste, en cumplimiento de la legislación vigente y a los efectos oportunos firman la presente en La Laguna a 30 de junio de 2017

# Agradecimientos

La realización de esta asignatura de Trabajo de Fin de Máster no hubiera sido posible sin la ayuda de la Sección de Ingeniería Informática de la Escuela Superior de Ingeniería y Tecnología, que ha llevado a cabo todos los trámites necesarios.

Mención especial para mi familia, pareja y amigos, quienes me han alentado para no rendirme y lograr mis objetivos pese a las dificultades y contratiempos encontrados durante la realización de este Trabajo de Fin de Máster.

Y por último, especialmente agradecer a Casiano Rodríguez León su labor como tutor del Trabajo de Fin de Máster. Además de aprender muchísimo junto a él, me ha aconsejado, animado y resuelto mis dudas de manera incansable en la realización de este trabajo. Estoy seguro de que la experiencia y conocimientos adquiridos gracias a él, me ayudarán en mis próximos retos profesionales y personales.

# Licencia



© Esta obra está bajo una licencia de Creative Commons  
Reconocimiento-NoComercial-CompartirIgual 4.0  
Internacional.

## Resumen

*El objetivo de este Trabajo de Fin de Máster ha sido integrar los conocimientos adquiridos durante los estudios del Máster y, en especial, del itinerario de Tecnologías de la Información, aproximando al alumno a la resolución de problemas de aplicaciones Web y favoreciendo el desarrollo de destrezas propias de la Ingeniería Web: se centra en el aprendizaje y puesta en práctica de metodologías, aproximaciones, técnicas y herramientas para abordar la creciente complejidad de este tipo de aplicaciones en el marco de las metodologías ágiles. Cada vez ésta cobra más importancia, siendo constante el aumento del número de aplicaciones de escritorio, smartphones y tablets.*

*En este Trabajo de Fin de Máster se propone el desarrollo de un paquete Node.js (NPM) que facilite la descarga y corrección de repositorios GitHub de alumnos. Existe un buen número de herramientas de Control de Versiones que permiten alojar proyectos software y agruparlos en organizaciones lógicas, pero carecen de mecanismos para automatizar funciones de uso cotidiano como la descarga de los mismos, la preparación del entorno de cada proyecto o la ejecución de pruebas.*

*En nuestra propuesta, se ha realizado una primera aproximación a la automatización de descargas y correcciones de repositorios, recopilando todos los datos inherentes de estas acciones y generando los informes correspondientes en formato PDF y HTML. Todo ello mediante un sencillo uso y sentando las bases para proporcionar más funcionalidades a la herramienta en un futuro próximo.*

**Palabras clave:** Consola, CLI, Shell, Terminal, Node.js, GitHub, Corrección, Automatización

## Abstract

Insert here an abstract in an EU language (preferably English)

**Keywords:** *Console, CLI, Shell, Terminal, Node.js, GitHub, Correction, Automation*

# Índice general

<b>1. Introducción</b>	<b>1</b>
1.1. Antecedentes . . . . .	1
1.2. Estado actual del arte . . . . .	1
1.3. Objetivos y actividades a realizar . . . . .	2
1.4. Tecnología usada . . . . .	3
<b>2. Desarrollo</b>	<b>4</b>
2.1. Metodología usada . . . . .	4
2.1.1. GitHub . . . . .	4
2.1.2. Travis-CI . . . . .	5
2.1.3. Experiencia de usuario . . . . .	5
<b>3. Resultados</b>	<b>7</b>
3.1. Problemas encontrados y soluciones . . . . .	7
3.2. Perfil del usuario de ghshell . . . . .	7
<b>4. Conclusiones y líneas futuras</b>	<b>9</b>
<b>5. Summary and Conclusions</b>	<b>10</b>
5.1. First Section . . . . .	10
<b>6. Presupuesto</b>	<b>11</b>
6.1. Introducción y coste por hora . . . . .	11
6.2. Desarrollo de la herramienta . . . . .	12
<b>A. Glosario</b>	<b>13</b>
<b>B. Guía de uso</b>	<b>17</b>
B.1. Instalación . . . . .	17
B.2. Ejecución . . . . .	17
B.2.1. Otras consideraciones . . . . .	17
<b>Índice alfabético</b>	<b>18</b>
<b>Bibliografía</b>	<b>19</b>

# Índice de Figuras

2.1. Apartado de issues cerrados . . . . .	5
2.2. Herramienta de integración continua . . . . .	5



# Índice de Tablas

6.1. Tabla de actividades, duración y precios . . . . .	12
---	----

# Capítulo 1

## Introducción

### 1.1. Antecedentes

La World Wide Web está sujeta a un cambio continuo. La llegada de HTML5, la creciente importancia de AJAX y de la programación en el lado del cliente, las nuevas fronteras de la Web Semántica, y la explosión de las redes sociales son ejemplos de esta tendencia general.

Las aplicaciones web parecen evolucionar hacia entornos cada vez más ricos y flexibles en los que los usuarios pueden acceder con facilidad a los documentos, publicar contenido, escuchar música, ver vídeos, realizar dibujos e incluso jugar usando un navegador. Esta nueva clase de software ubicuo no cesa de ganar momentum y promueve nuevas formas de interacción y cooperación.

Ante la rápida evolución del software, los sistemas de control de versiones han adquirido una mayor importancia dentro de la metodología del desarrollo del software: la gestión de las versiones del propio software se ha convertido en una actividad crítica. Estos sistemas han evolucionado a la par que el software, proporcionando nuevas funcionalidades y orientándose hacia la colaboración.

### 1.2. Estado actual del arte

Actualmente, hay numerosos sistemas de control de versiones. Todos ellos proporcionan mecanismos de almacenamiento del código, de su modificación y de consulta histórica del mismo, a la vez que proporcionan un entorno colaborativo en el que los usuarios pueden colaborar entre sí.

En el caso particular de GitHub, además de proporcionar lo mencionado anteriormente, observando el creciente número de estudiantes que utiliza la plataforma, ha creado herramientas específicas para facilitar sus desarrollos (ej: Student Developer Pack) y provee a profesores de herramientas para gestionar dichos desarrollos (ej: GitHub Classrooms).

Sin embargo, estas herramientas de gestión de desarrollos requieren una administración interactiva por parte del profesor. No cuentan aún con funcionalidades de automatización de tareas.

### 1.3. Objetivos y actividades a realizar

En este proyecto se persigue integrar los conocimientos adquiridos durante los estudios del Máster y, en especial, del itinerario de Tecnologías de la Información para solucionar problemas actuales de aplicaciones y servicios Web.

Los objetivos propuestos para alcanzar en este Trabajo de Fin de Máster han sido los siguientes:

- Conocer, dominar y practicar con lenguajes y herramientas de desarrollo de aplicaciones Web en el servidor.
- Conocer, dominar y practicar con diferentes lenguajes y librerías en el cliente.
- Conocer, practicar y dominar de herramientas de Desarrollo Dirigido por Pruebas en entornos web.
- Conocer, practicar y dominar diferentes lenguajes de marcas y de estilo.
- Conocer, practicar y dominar diferentes mecanismos de despliegue.
- Conocer, practicar y familiarizarse con diferentes mecanismos de seguridad, autenticación y autorización.
- Conocer, practicar y dominar diferentes herramientas colaborativas y de control de versiones.
- Conocer, practicar y dominar Metodologías Ágiles de desarrollo de software.

Y las actividades a realizar en el mismo son las que se describen a continuación:

- Revisión bibliográfica y estado del arte.
- Desarrollar una herramienta de línea de comandos escrita en Node.js que permita automatizar tareas relacionadas con repositorios de GitHub:
  - 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 (TDD, creación de entorno, evaluación de código...).
  - Recopilar la información obtenida de la automatización de tareas y presentarla al usuario (PDF, HTML...).
- Redacción de la memoria.
  - Preparación de la presentación oral.

## 1.4. Tecnología usada

Para llevar a cabo el desarrollo de esta herramienta se planteó realizar el desarrollo en **Node.js**, creando una librería modular que se pudiese instalar mediante el gestor de paquetes de Node.js (NPM).



Además, se ha hecho uso de otras tecnologías enumeradas a continuación:

- NPM 
- GitHub 
- GitBook  **GitBook**
- Travis-CI  **Travis CI**

# Capítulo 2

## Desarrollo

En el capítulo anterior se ha descrito el estado del arte actual y se ha definido el Trabajo de Fin de Máster, especificado los objetivos, actividades a desarrollar y las tecnologías empleadas para su desarrollo. A continuación, se describirá la metodología de trabajo seguida.

### 2.1. Metodología usada

Se ha llevado a cabo una metodología de trabajo ágil, común en el campo de la Ingeniería Informática, con reuniones quincenales en las que se definían una serie de tareas u objetivos (iteración) y que se presentaban la siguiente quincena. De este modo, con la entrega de prototipos funcionales de la aplicación, se han ido testeando, corrigiendo y mejorando las funcionalidades, al mismo tiempo que detectando problemas no contemplados en las fases previas de diseño.

Esta metodología, además, ha propiciado la generación de ideas que se han traducido en nuevas características.

#### 2.1.1. GitHub

Para llevar a cabo esta metodología, se ha usado GitHub como herramienta de Control de Versiones (CVS). Todo el código implementado se alojaba en dicha herramienta, permitiendo así su cómoda modificación y actualización.

[ Imagen ]

El trabajo se dividía en ramas, de modo que la versión estable de la aplicación (rama master) quedara aislada de la versión en desarrollo (rama develop) y de la rama experimental (rama test).

[ Imagen ]

La documentación adicional para llevar a cabo los desarrollos de cada iteración, así como los problemas detectados, se anotaban en el apartado de issues con el fin de que quedara constancia de ello y se reflejara el estado en el que se encontraba cada uno.

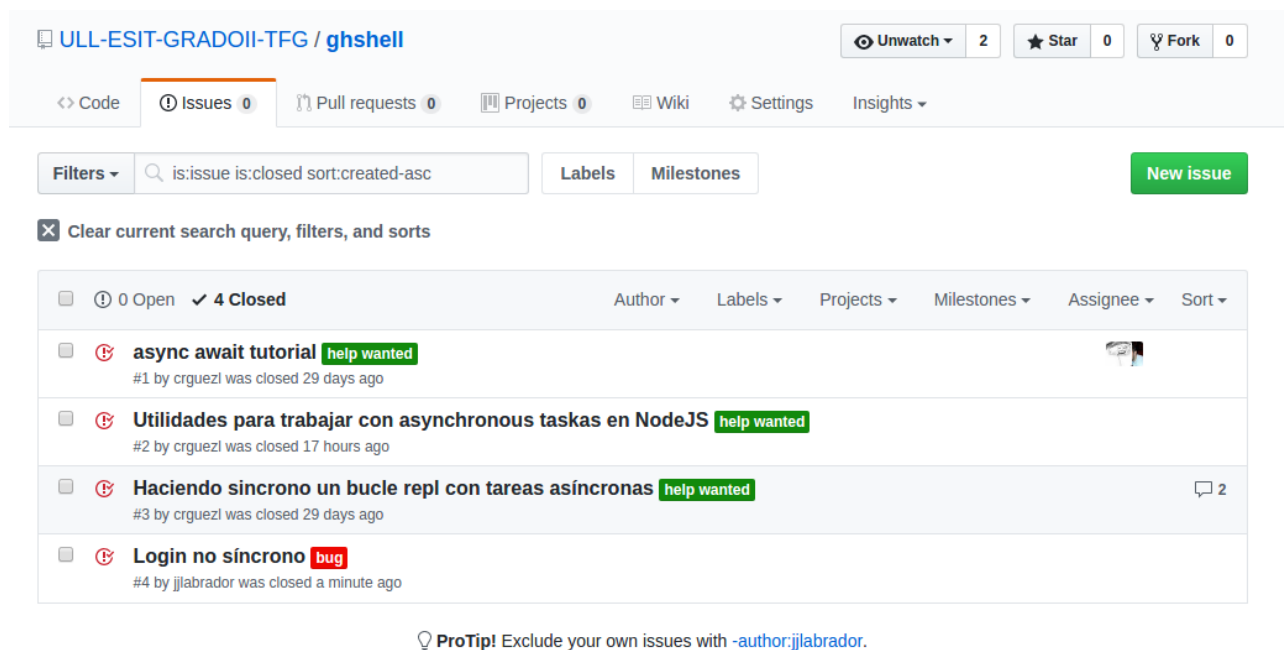


Figura 2.1: Apartado de issues cerrados

### 2.1.2. Travis-CI

Como herramienta de integración continua, se ha utilizado Travis-CI, con el fin de asegurarnos el despliegue de la aplicación era satisfactorio tras cada cambio subido a la herramienta de control de versiones (GitHub).

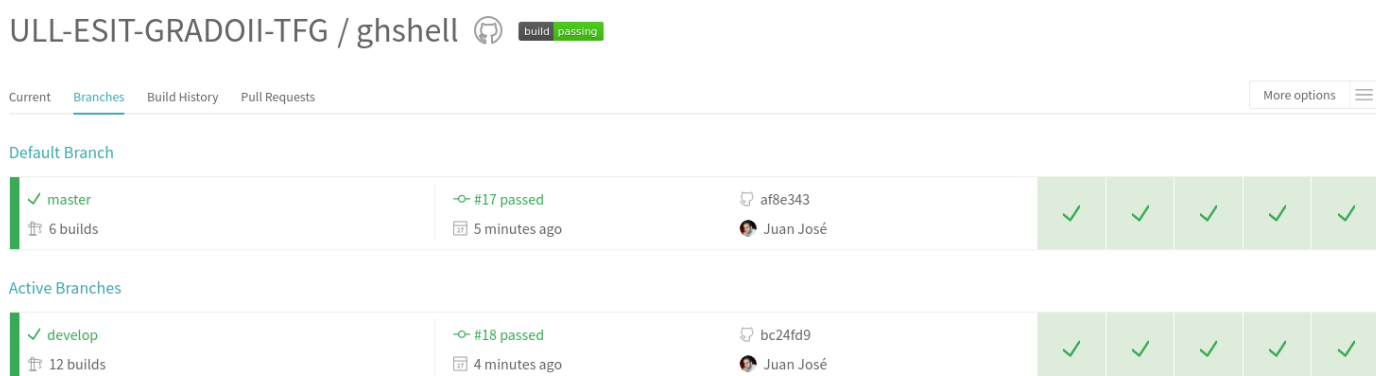


Figura 2.2: Herramienta de integración continua

### 2.1.3. Experiencia de usuario

Por otra parte, el tutor del Trabajo de Fin de Máster ha hecho pruebas reales con el resultado de cada iteración. De este modo, se comprobaba el funciona-

miento de la aplicación en un entorno real y se recibía un valioso feedback para corregir problemas o hacer mejoras en las siguientes iteraciones.

# Capítulo 3

## Resultados

Finalizada la etapa de desarrollo del Trabajo de Fin de Máster, se procede a describir la herramienta implementada.

La herramienta se ha denominado ghshell, abreviatura de 'GitHub Shell'. Se ha publicado en NPM[Enlace] para su fácil distribución e instalación.

Las funcionalidades implementadas en ghshell, se describen a continuación:

- Autenticación con GitHub. [ Imagen ]
- Listar organizaciones, asignaciones y repositorios de GitHub del usuario. [ Imagen ]
- Automatizar la descarga de repositorios. [ Imagen ]
- Automatizar la ejecución de scripts en los repositorios (TDD, creación de entorno, evaluación de código...). [ Imagen ]
- Recopilar la información obtenida de la automatización de tareas y presentarla al usuario (PDF, HTML...). [ Imagen ]

### 3.1. Problemas encontrados y soluciones

Asincronía, etc.

### 3.2. Perfil del usuario de ghshell

El uso de ghshell está especialmente dirigido a un determinado grupo de profesores: nos referimos al perfil de un profesor, principalmente docente en alguna rama de Ingeniería, con conocimientos avanzados en programación y en herramientas de control de versiones.

No obstante, ya que la curva de aprendizaje de ghshell no es excesiva y dado que el uso de las herramientas de control de versiones no se limita exclusivamente a repositorios de código fuente, se puede extender su uso para el resto



de profesorado y usuarios con otros roles. Basta con tener claras unas nociones básicas de informática, junto con la lectura y asimilación previa de la documentación de la herramienta.

## Capítulo 4

# Conclusiones y líneas futuras

Este capítulo es obligatorio. Toda memoria de Trabajo de Fin de Máster ha de incluir unas conclusiones y unas líneas de trabajo futuro

Desde hace unos años hasta ahora, ha tenido lugar un enorme crecimiento de las herramientas de control de versiones. Se han convertido en una herramienta imprescindible en la metodologías de desarrollo del software y las instituciones de enseñanza saben que incorporarlas a sus sistemas educativos es clave para ofrecer un servicio puntero y de calidad.

Ésto es lo que se pretende con la herramienta obtenida tras la realización de este Trabajo de Fin de Máster: que sea posible su implantación dentro del marco académico de la Universidad de La Laguna, partiendo de la premisa de que, actualmente, el desarrollo de un proyecto software sin tener detrás un sistema de control de versiones, no es viable.

.....

Nuevo paradigma: Sincronía/asincronía Motivación de aprender algo nuevo

Futuro: ampliar funcionalidades: push, issues, despliegues...

# Capítulo 5

## Summary and Conclusions

This chapter is compulsory. The memory should include an extended summary and conclusions in english.

### 5.1. First Section

# Capítulo 6

## Presupuesto

En este capítulo se especifica un presupuesto que indica cuánto costaría realizar este Trabajo de Fin de Máster si se tratase de un trabajo encargado por un cliente.

### 6.1. Introducción y coste por hora

Se definirá una tabla con la lista de actividades realizadas en este Trabajo de Fin de Máster. Otra columna indicará la duración en horas que se han empleado para dicha actividad junto con el precio por hora calculado.

El precio por hora que se considerará en este presupuesto es de 30€/hora.

## 6.2. Desarrollo de la herramienta

Actividad	Duración	Precio
Autenticación con GitHub	xx horas	xx €
Listar organizaciones, asignaciones y repositorios	xx horas	xx €
Automatizar la descarga de repositorios	xx horas	xx €
Automatizar la ejecución de scripts en los repositorios	xx horas	xx €
Exportar la información obtenida de la automatización de tareas	xx horas	xx €
<b>Total</b>	<b>xx horas</b>	<b>xx €</b>

Cuadro 6.1: Tabla de actividades, duración y precios

# Apéndice A

## Glosario

### A

**AJAX** : acrónimo de *Asynchronous JavaScript And XML* (JavaScript asíncrono y XML). Es una técnica de desarrollo web para crear aplicaciones interactivas o RIA (*Rich Internet Applications*). Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas, mejorando la interactividad, velocidad y usabilidad en las aplicaciones.

**API** : (*Application Programming Interface* o Interfaz de Programación de Aplicaciones). Conjunto de funciones y procedimientos o métodos que ofrece cierta librería para ser utilizados por otro software como una capa de abstracción.

### C

**CVS** : (*Control Versioning System* o Sistema de Control de Versiones). Aplicación informática que implementa un sistema de control de versiones: mantiene el registro de todo el trabajo y los cambios en los ficheros (código fuente principalmente) que forman un proyecto y permite que distintos desarrolladores (potencialmente situados a gran distancia) colaboren.

### D

**DSL** : (*Domain Specific Language* o Lenguaje de Dominio Específico). En desarrollo de software, un lenguaje de dominio específico es un lenguaje de programación o especificación de un lenguaje dedicado a resolver un problema en particular, representar un problema específico y proveer una técnica para solucionar una situación particular.

## G

**Google Drive** : servicio de alojamiento de archivos accesible desde su página web mediante ordenadores o mediante aplicaciones móviles. Para más información, visitar <https://drive.google.com>.

**GitHub** : forja para alojar proyectos utilizando el Sistema de Control de Versiones **Git**. Para más información, visitar <https://github.com>.

## H

**HTML5** : (*HyperText Markup Language*). Lenguaje de marcado para la elaboración de páginas web. Es un estándar que sirve de referencia para la elaboración de páginas web definiendo una estructura básica y un código para la definición del contenido de la misma.

# J

**JavaScript** : lenguaje de programación interpretado. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico. Se utiliza principalmente en su forma del lado del cliente (*client-side*), implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas.

# M

**Metodologías ágiles** : conjunto de métodos de ingeniería del software basados en el desarrollo iterativo e incremental, donde los requisitos y soluciones evolucionan mediante la colaboración de grupos auto organizados y multidisciplinarios. Se caracterizan además por la minimización de riesgos desarrollando software en iteraciones cortas de tiempo.

# O

**OAuth 2.0** : (*Open Authorization*): protocolo abierto que permite la autorización segura de una API de modo estándar y simple para aplicaciones de escritorio, web y móviles. Para más información, visitar <http://oauth.net/2/>



# T

**TDD** : (*Test-Driven Development* o Desarrollo Dirigido por Pruebas). Práctica de programación que involucra otras dos prácticas: escribir las pruebas primero (*Test First Development*) y Refactorización de código (*Refactoring*).

# W

**Web semántica** : idea de añadir metadatos semánticos y ontológicos a la World Wide Web. Esas informaciones adicionales, que describen el contenido, el significado y la relación de los datos, se deben proporcionar de manera formal, para que sea posible evaluarlas automáticamente por máquinas de procesamiento. El objetivo es mejorar Internet ampliando la interoperabilidad entre los sistemas informáticos usando **agentes inteligentes**, es decir, programas en las computadoras que buscan información sin necesidad de interacción humana.

**World Wide Web** : (WWW). Sistema de distribución de documentos de hipertexto o hipermedios interconectados y accesibles vía Internet. Con un navegador web, un usuario visualiza sitios web compuestos de páginas web que pueden contener texto, imágenes, vídeos u otros contenidos multimedia, y navega a través de esas páginas usando hiperenlaces.

# Apéndice B

## Guía de uso

El objetivo de esta guía de usuario es proporcionar a los usuarios un ejemplo para la puesta a punto y ejecución de las funcionalidades implementadas en el gema RuQL durante el Trabajo de Fin de Grado.

### B.1. Instalación

Para instalar la gema, basta con ejecutar el siguiente comando:

```
[~]$ gem install ruql
```

### B.2. Ejecución

```
1 var foo = function(){  
2   console.log('foo');  
3 }  
4 foo();
```

#### B.2.1. Otras consideraciones

Para que

# Índice alfabético

Node.js, 3

# Bibliografía

- [1] ACM LaTeX Style. [http://www.acm.org/publications/latex\\_style/](http://www.acm.org/publications/latex_style/).
- [2] FACOM OS IV SSL II USER'S GUIDE, 99SP0050E5. Technical report, 1990.
- [3] D. H. Bailey and P. Swarztrauber. The fractional Fourier transform and applications. *SIAM Rev.*, 33(3):389–404, 1991.
- [4] A. Bayliss, C. I. Goldstein, and E. Turkel. An iterative method for the Helmholtz equation. *J. Comp. Phys.*, 49:443–457, 1983.
- [5] C. Darwin. *The Origin Of Species*. November 1859.
- [6] C. Goldstein. Multigrid methods for elliptic problems in unbounded domains. *SIAM J. Numer. Anal.*, 30:159–183, 1993.
- [7] P. Swarztrauber. *Vectorizing the FFTs*. Academic Press, New York, 1982.
- [8] S. Taásan. *Multigrid Methods for Highly Oscillatory Problems*. PhD thesis, Weizmann Institute of Science, Rehovot, Israel, 1984.