



Trabajo de Fin de Grado

Codelab
Samuel Ramos Barroso

La Laguna, 8 de abril de 2018

D. **Casiano Rodríguez León**, con N.I.F. 42.020.072-S profesor Titular de Universidad adscrito al Departamento de Nombre del Departamento de la Universidad de La Laguna, como tutor

C E R T I F I C A

Que la presente memoria titulada:

“Codelab.”

ha sido realizada bajo su dirección por D. **Samuel Ramos Barroso**, con N.I.F. 51.165-611-H.

Y para que así conste, en cumplimiento de la legislación vigente y a los efectos oportunos firman la presente en La Laguna a 8 de abril de 2018

Agradecimientos

XXX

XXX

XXX

XXX

Licencia



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

Resumen

Este proyecto trata sobre la creación plataforma web destinada al apoyo del profesorado para la realización de prácticas intentando resolver las limitaciones que tienen otras plataformas para la gestión de prácticas. El profesor podrá crear tareas que serán asignadas a sus alumnos de forma individual o grupal. Cada alumno realizará su realización en un repositorio git, una vez finalizada, podrá ser revisada por el profesor.

Palabras clave: Git, Github, Tareas, clases, prácticas.

Abstract

This project is about the creation of a web platform designed to support teachers for the realization of practices trying to solve the limitations that other platforms have for the practice management. The teacher can create tasks that will be assigned to his students individually or group. Each student will perform their work in a git repository, once completed, they can be reviewed by the teacher.

Keywords: *Git, Github, Tareas, clases, prácticas.*

Índice general

1. Introducción	1
1.1. Herramientas para el apoyo del profesorado de informática . . .	1
1.2. Github Education	2
2. Plan de trabajo	4
2.1. Objetivos	4
2.2. Plan de trabajo	5
3. Desarrollo del proyecto	6
3.1. Tecnologías	6
3.1.1. Tecnologías populares	6
3.1.2. Tecnologías Escogidas	6
3.2. Segunda sección de este capítulo	9
3.3. Tercera sección de este capítulo	9
4. Título del Capítulo Cuatro	10
5. Conclusiones y líneas futuras	11
6. Summary and Conclusions	12
6.1. First Section	12
7. Presupuesto	13
7.1. Sección Uno	13
A. Título del Apéndice 1	14

A.1. Algoritmo XXX	14
A.2. Algoritmo YYY	14
B. Título del Apéndice 2	16
B.1. Otro apéndice: Sección 1	16
B.2. Otro apéndice: Sección 2	16
Bibliografía	16

Índice de figuras

Índice de tablas

2.1. Tabla resumen del plan de trabajo	5
7.1. Tabla resumen de los Tipos	13

Capítulo 1

Introducción

1.1. Herramientas para el apoyo del profesorado de informática

En los últimos años se han desarrollado las nuevas tecnologías, lo que ha permitido que lleguen nuevas herramientas de aprendizaje y de apoyo a la docencia. Es el caso de la exitosa plataforma Moodle, un LCMS, sigla de Learning Content Management System.

El Moodle, como otros LCMS, se utiliza para crear y manejar el contenido de un programa educativo, como por ejemplo un curso. Normalmente se crean partes de contenido en forma de módulos que se pueden personalizar.

Existen otras herramientas como el LMS sigla de Learning Management System. El LMS es un software instalado en un servidor que se emplea para administrar, distribuir y controlar las actividades de formación de una institución. Un LCMS suele estar integrado en un LMS.

Las principales funciones del LMS son: gestionar usuarios, recursos, materiales y actividades de formación, administrar el acceso, controlar y hacer seguimiento del proceso de aprendizaje, realizar evaluaciones, generar informes, gestionar servicios de comunicación como foros de discusión, videoconferencias y correo electrónico entre otros.

Como ejemplos reales de las definiciones mencionadas, se encuentra el servicio LCMS de Google, Google Classroom que está basado en Moodle, con la diferencia de que usa la Suite de Google Drive para el apoyo de almacenamiento y gestión de tareas.

En el caso del apoyo al profesorado de informática encontramos los VPL sigla de Virtual Programing Lab, que es un módulo para la plataforma Moodle. Los VPL permiten editar y ejecutar el código fuente de las tareas de programación

en el navegador, además permite que el profesor pueda corregir sin necesidad de descargar el código fuente para ejecutarlo, también se permite añadir restricciones como pegar código externo y permite comparar la similitud entre códigos de los alumnos, entre otras cosas.

Por último, la única herramienta para la gestión de tareas está desarrollada por Github, se trata de Github Classroom, una plataforma aprovecha las organizaciones y repositorios de Github como estructura para las aulas y tareas, haciendo uso de la herramienta de control de versiones Git. En el siguiente punto hablaremos en profundidad de Github Education y Classroom.

1.2. Github Education

Github Education es una plataforma desarrollada por Github que contiene varias herramientas para apoyar a los profesores en la asignación de tareas de programación:

- Classroom
- Classroom Desktop
- Teachers Pet
- Education Community
- Student Pack

Classroom es una herramienta destinada a profesores para gestionar el uso educativo de GitHub. Simplifica la asignación de tareas, automatizando la creación de repositorios git usando las organizaciones de Github como aulas y los repositorios como asignaciones, es una herramienta útil y sencilla de usar, tanto para profesores como para alumnos, pero tiene ciertos defectos, por ejemplo, no se puede crear un repositorio de evaluación que contenga las tareas de todos los alumnos, para eso se debe usar Classroom Desktop, es decir, se requiere de otra herramienta para una única funcionalidad, tampoco se puede acceder a los enlaces de travis, si la práctica requiere su uso, por último, tiene un sistema para asociar información a cada alumno que es bastante complejo de usar, ya que requiere introducir a mano cada usuario y la información que se desea asociar.

Teachers Pet es un CLI destinado a la administración de Education vía terminal, pero dejaron de desarrollarla porque para realizar ciertas tareas requería demasiadas opciones y los comandos eran demasiado largos.

Education Comunity es un foro para el intercambio de información entre alumnos, profesores, investigadores y desarrolladores.

Student Pack es un paquete que ofrece Github a los alumnos para mejorar su experiencia haciendo uso de Github para el desarrollo de tareas.

Capítulo 2

Plan de trabajo

2.1. Objetivos

Los objetivos a desarrollar en este trabajo se definen a continuación:

- **A1.** Analizar otras plataformas web existentes para la gestión del código de las prácticas de informática y su metodología de trabajo.
- **A2.** Estudiar el funcionamiento de otras plataformas web existentes para la gestión del código de las prácticas de informática.
- **A3.** Estudiar las tecnologías a usar y enfocar el diseño de la plataforma web.
- **A4.** Estudiar las funcionalidades que se van a incluir en la plataforma web.
- **A5.** Crear una aplicación web básica que permita al usuario iniciar sesión con su cuenta de Github.
- **A6.** Continuar con el desarrollo de la aplicación incluyendo las funcionalidades que solucionen las dificultades de otras plataformas web existentes para la gestión del código de las prácticas de informática.
- **A7.** Diseñar y desarrollar los estilos de las vistas.

2.2. Plan de trabajo

Objetivo	Fecha
A1	30 Enero - 3 Febrero
A2	4 - 9 Febrero
A3	10 - 15 Febrero
A4	16 - 21 Febrero
A5	22 Febrero - 22 Marzo
A6	23 Marzo - 23 Mayo
A7	25 Abril - 20 Junio

Tabla 2.1: Tabla resumen del plan de trabajo

Capítulo 3

Desarrollo del proyecto

3.1. Tecnologías

3.1.1. Tecnologías populares

Django - Python Django es un framework para aplicaciones web gratuito y open source, escrito en Python. Un conjunto de componentes que te ayudan a desarrollar sitios web más fácil y rápidamente siguiendo el patrón Modelo Vista Controlador.

Ruby on Rails - Ruby Ruby on Rails, también conocido como RoR o Rails, es un framework de aplicaciones web de código abierto escrito en el lenguaje de programación Ruby, siguiendo el patrón Modelo Vista Controlador (MVC).

Express.js - Javascript Express.js es un framework de desarrollo de aplicaciones web minimalista y flexible para Node.js. Está inspirado en Sinatra, además es robusto, rápido, flexible y muy simple, además, es compatible con el patrón Modelo Vista Controlador.

Aquí tenemos una pequeña comparación de las búsquedas que han tenido las tecnologías en el último año.

3.1.2. Tecnologías Escogidas

Finalmente se escogió Express.js, Node.js y Javascript como tecnologías principales a usar en este proyecto por las siguientes razones:

En primer lugar, el uso de Node.js como tecnología de servidor aporta una gran ventaja ya que Javascript se usa en ambos lados, en el backend y en el frontend, reduciendo complejidad y tiempo de desarrollo ya que Express es bastante fácil de aprender y usar. Además, JavaScript es el lenguaje de

programación más popular en el desarrollo web.

Además de eso, la comunidad de Node.js está en constante crecimiento: la cantidad de preguntas de StackOverflow aumenta constantemente, por lo que la base de conocimiento para la tecnología es amplia. También hay que destacar el hecho de que Node.js sea de código abierto y gratuito.

Finalmente, Node nos ofrece NPM, un gestor de paquetes que dispone de una gran cantidad de paquetes que crece rápidamente y aporta una forma sencilla de gestionar los paquetes que necesitemos en nuestro proyecto.

NPM

NPM es un administrador de paquetes para Node.js con cientos de miles de paquetes. Aunque crea parte de la estructura del directorio del proyecto, este no es el objetivo principal.

El objetivo principal, es la automatización de dependencias y gestión de paquetes de cada proyecto. Esto significa que puede especificar todas las dependencias de su proyecto dentro del fichero package.json usando el comando `npm i -s nombrepaquete`, de forma que cada vez que un usuario clone el proyecto en su máquina simplemente tenga ejecutar `npm install` e inmediatamente se descargan e instalan todas las dependencias necesarias para el funcionamiento del proyecto en el directorio `node_modules/`. Además, también sirve para especificar en qué versión está su proyecto.

También es posible descargar manualmente los paquetes, copiarlo en el directorio `nodemodules/` y usarlo de esa manera. Sin embargo, a medida que crezca el proyecto y la lista de dependencias, llevará mucho tiempo instalar los paquetes necesarios y será una tarea engorrosa. También hace que colaborar y compartir tu proyecto sea mucho más difícil

Express.js

Express es un framework “minimalista” que permite crear una infraestructura web simple sobre Node.js. Express permite crear una API REST, un tipo de arquitectura de desarrollo web que se apoya totalmente en el estándar HTTP:

- GET: Para consultar y leer recursos
- POST: Para crear recursos
- PUT: Para editar recursos
- DELETE: Para eliminar recursos
- PATCH: Para editar partes concretas de un recurso

Express también permite crear una aplicación web siguiendo el patrón de diseño Modelo Vista Controlador, que separa los datos y la lógica de una aplicación de su representación.

Github API

Para poder usar Github como estructura necesitaremos hacer uso de su Github REST API v3. La API nos permitirá acceder a los servicios que ofrece Github de forma externa, es decir, podremos crear repositorios, usar organizaciones como clases y manejar otros servicios, como consulta de información sobre el usuario. Para poder acceder a la clave de cada usuario, debemos usar un Token generado por Github, es decir, una cadena que nos permitirá identificarnos para realizar transacciones usando la API, en este caso, el token será generado usando las aplicaciones OAuth que nos ofrece Github para que el usuario haga login con su cuenta de usuario de github, sin necesidad de poner su usuario y contraseña de Github en la plataforma Codelab. La Github REST API v3 tiene además varias librerías de terceros y tres librerías oficiales escritas en Ruby, Javascript y .Net.

MongoDB

MongoDB es un sistema gestor de base de datos NoSQL, es decir, MongoDB es un sistema de bases de datos no relacional, en lugar de almacenar los datos en tablas, se almacenan colecciones o documentos, que está formado por objetos, y estos a su vez por claves y cada clave tiene un valor asociado. Es decir, sería algo parecido a un documento JSON, aunque en MongoDB usan una distribución llamada BSON.

Mongoose

Mongoose es una herramienta para el diseño de objetos de MongoDB. Mongoose provee una serie de métodos y funciones para manejar la base de datos en MongoDB.

Motor de vistas

Como motor de vistas para el proyecto se ha elegido Pug, antes conocido como Jade, un motor de vistas implementado en Javascript para su uso en NodeJS y que dispone de una fácil integración con Express.JS.

Automatización de tareas

Para la automatización de tareas como generar el fichero de variables de entorno o poner en marcha un servidor nodemon se ha usado Gulp, una herramienta que permite automatizar tareas en Node.js, es simple y bastante fácil de usar. **Framework de CSS** Como framework para los estilos de CSS se ha usado Materialize CSS, basado en material design. Ofrece varias opciones a la hora de diseñar la plataforma, permitiendo un diseño más minimalista o un diseño más cargado y vistoso.

3.2. Segunda sección de este capítulo

3.3. Tercera sección de este capítulo

Capítulo 4

Título del Capítulo Cuatro

Los capítulos intermedios servirán para cubrir los siguientes aspectos: antecedentes, problemática o estado del arte, objetivos, fases y desarrollo del proyecto.

En el capítulo 1 se describió bla, bla, bla.....

Capítulo 5

Conclusiones y líneas futuras

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

Capítulo 6

Summary and Conclusions

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

6.1. First Section

Capítulo 7

Presupuesto

Este capítulo es obligatorio. Toda memoria de Trabajo de Fin de Grado debe incluir un presupuesto.

7.1. Sección Uno

Objetivo	Descripcion
AAAA	BBBB
CCCC	DDDD
EEEE	FFFF
GGGG	HHHH

Tabla 7.1: Tabla resumen de los Tipos

Apéndice A

Título del Apéndice 1

A.1. Algoritmo XXX

```
*****
*
* Fichero .h
*
*****
* AUTORES
*
*
* FECHA
*
*
* DESCRIPCION
*
*
*****/
```

A.2. Algoritmo YYY

```
/*****
*
* Fichero .h
*
*****
* AUTORES
*
* FECHA
*
* DESCRIPCION
*
*****/
```


*

*****/

Apéndice B

Título del Apéndice 2

B.1. Otro apéndice: Sección 1

Texto

B.2. Otro apéndice: Sección 2

Texto

Bibliografía

- [1] ACM 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.