Universidad de La Laguna

CodeLab

A Tool to automate repository creation and access control, giving support to distribute starter code, collect assignments and evaluate the students work on GitHub.

Autor: Samuel Ramos Barroso
Director: Casiano Rodríguez León

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

14 de Junio de 2018

Índice

Introducción I

CodeLab es una plataforma web basada en Javascript destinada al apoyo del profesorado para la realización de prácticas intentando resolver las limitaciones que tienen otras plataformas para la gestión de prácticas.

Introducción II

- El profesor podrá crear tareas que serán asignadas a sus alumnos de forma individual o grupal.
- Cada alumno realizará su trabajo en un repositorio git, el cual una vez finalizado, podrá ser revisado por el profesor.
- La generación de aplicaciones correctoras de exámenes provistas de lo necesario para su despliegue y puesta en funcionamiento.

Introducción III

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.

Introducción IV

La única herramienta para la gestión de tareas está desarrollada por Github, se trata de Github Classroom, que forma parte de Github Education:

- GitHub Classroom
- Classroom Desktop
- Teachers Pet
- Student Pack

Introducción V

Classroom simplifica la asignación de tareas, automatizando la creación de repositorios, es una herramienta útil y sencilla de usar, tanto para profesores como para alumnos, pero tiene ciertos defectos:

- No se puede crear un repositorio de evaluación que contenga las tareas de todos los alumnos
- Tampoco se puede acceder a los enlaces de travis, si la práctica requiere su uso
- Tiene un sistema para asociar información a cada alumno que es bastante complejo de usar

Objetivos I

- Analizar otras plataformas web existentes para la gestión del código de las prácticas de informática y su metodología de trabajo.
- Estudiar el funcionamiento de otras plataformas web existentes para la gestión del código de las prácticas de informática.
- Estudiar las tecnologías a usar y enfocar el diseño de la plataforma web.
- Estudiar las funcionalidades que se van a incluir en la plataforma web.

Objetivos II

- Crear una aplicación web básica que permita al usuario iniciar sesión con su cuenta de Github.
- 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.
- Diseñar y desarrollar los estilos de las vistas.

Tecnología usada I

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

- Express.js es bastante fácil de aprender y usar.
- Se usa JS tanto en Backend como en frontend, ahorrando tiempo de desarrollo.
- El gestor de paquetes NPM.

Tecnología usada II

Otras tecnologías usadas:

- Github API
- MongoDB - Mongoose
- El gestor de paquetes NPM.
- Pug.js
- Materialize

Desarrollo de la plataforma

Github API

Para poder usar Github como estructura para la creación de aulas y tareas de código usaremos la Github REST API V3. La API permite acceder a las funcionalidades de Github, A continuación se detallará que funcionalidades de la Github API se han usado

- OAuth
- Organizaciones, repositorios y equipos.

OAuth

Desde el punto de vista de Codelab, OAuth proporciona un método de acceso a los datos de Github a la vez que se protegen los credenciales de la cuenta. Solicitamos al usuario una serie de accesos y permisos en sus datos, para poder operar con las organizaciones y repositorios de Github.

Organizaciones repositorios y equipos

Las operaciones que realizaremos con Organizaciones, repositorios y equipos son las siquientes:

- Obtener las organizaciones del usuario.
- Añadir usuarios a la organización.
- Crear un repositorio en las organizaciones.
- Añadir colaboradores al repositorio.
- Crear equipos.
- Añadir un equipo a un repositorio.
- Comprobar si un usuario es miembro de un equipo.
- Obtener los repositorios de una organización.
- Crear un fichero en un repositorio.

MVC

El modelo vista controlador es una arquitectura de software que separa la lógica de la aplicación de la interfaz de usuario. Lo hace separando la aplicación en tres partes: el modelo, la vista y el controlador.

Diseño de la base de datos I

Se usará una base de datos NoSQL, en concreto MongoDB como sistema gestor y Mongoose como ODM, por lo que nos referimos las tablas como colecciones, columnas como claves y filas como objeto.

La base de datos es una parte muy importante de la plataforma, ya que en ella recae toda la responsabilidad de simular toda la estructura de aulas y tareas.

Diseño de la base de datos II

Las colecciones son las siguientes:

- Usuarios
- Organizaciones
- Asignaciones
- Asignaciones en grupo
- Asignaciones individuales
- Equipos
- Alumnos

Funcionalidades I

El proyecto se divide en tres paquetes de funcionalidades, Cada paquete incluye funcionalidades para cada rol:

- Funcionalidades básicas
- Funcionalidades para profesores.
- Funcionalidades para el alumno

Funcionalidades II

Las funcionalidades básicas son comunes a todos los roles que participan en la plataforma, tanto alumnos como profesores, todos pueden hacer Log in, Log out y consultar un perfil.

Como alumno el usuario puede visitar el perfil, donde encontrará información básica de Github y dos pestañas, en las que el alumno tiene un historial de las tareas que ha realizado de forma grupal e individual.

Los profesores tendrán el grupo de funcionalidades más completo, ya que ellos son los protagonistas de la app.

Funcionalidades III

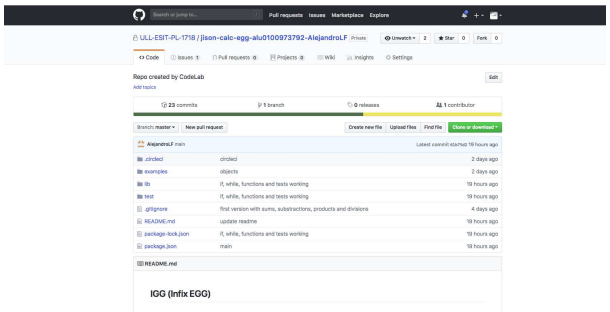
Los profesores podrán desempeñar las siguientes tareas:

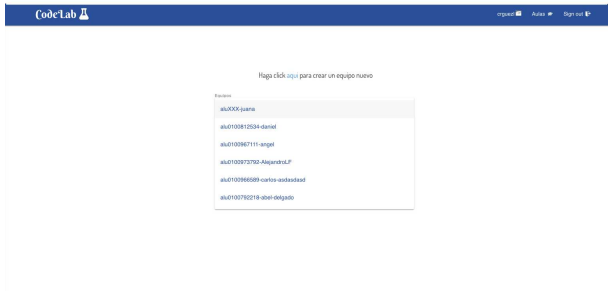
- Añadir una organización como aula.
- Invitar alumnos al aula.
- Crear una tarea.
- Añadir un fichero de alumnos asociado al aula.
- Editar las opciones del aula.
- Invitar alumnos a la tarea.
- Editar las opciones de la tarea.
- Crear un repositorio de evaluación de cada tarea.

Caso de uso I

Con vistas a probar y testear que todo funcionaba de forma correcta, Casiano me sugirió probar la plataforma para la realización de algunas prácticas individuales y grupales. Por ello, decidimos realizar algunas tareas para la asignatura de Procesadores de Lenguajes en CodeLab.

Caso de uso II





Conclusiones y Trabajos Futuros/Conclusions and Future Work I

- Esta herramienta pretende ser un complemento para plataformas como Moodle al ofrecer la posibilidad de especificar preguntas de programación.
- Ofrece una solución innovadora para el almacenamiento de los datos de los exámenes: **Google Drive**. Permite gestionar de forma más cómoda los datos generados en lugar de usar bases de datos.
- Por otra parte, considerando aspectos éticos y de seguridad, se hace uso de **OAuth** para la autentificación de usuarios con el fin de evitar posibles problemas de phishing y exposición de datos sensibles a terceras personas.

Conclusiones y Trabajos Futuros/Conclusions and Future Work II

Trabajos Futuros:

- Resolver los problemas de seguridad relacionados con evaluar el código escrito de los alumnos.
- Dar soporte a preguntas con respuestas de código en otros lenguajes de programación.
- Ofrecer una alternativa de despliegue distinta a Heroku.
- Escribir *renderers* para dar soporte a otros formatos usados por diversas plataformas educativas (Ej: MoodleXML, Gift, etc.).

Conclusiones y Trabajos Futuros/Conclusions and Future Work III

- This tool intends to complement learning management systems with new capabilities like the possibility to specify **programming questions**.
- It uses **Google Drive** for the storage of exams data (instead using databases), providing in this way a more natural solution from the lecturer perspective.
- On the other hand, keeping in mind ethic and legal topics, we use **OAuth** to delegate the authentication to Google. This way, we avoid security bugs as the phishing or the exposure of sensitive information to third people.

Conclusiones y Trabajos Futuros/Conclusions and Future Work IV

Future Work:

- Solve the security problem related with the evaluation of student code in the server.
- Provide support to questions with answers written in other programming languages.
- Provide a deployment alternative different to Heroku.
- To write renderers giving support to other formats (MoodleXML, Gift, etc.) used by a variety of learning platforms.

Bibliografía I


"Node JS."
https://nodejs.org/es/docs/.


"Github REST API V3."
https://developer.github.com/v3/.


"Express JS."
http://expressjs.com/es/.


"Pug JS."
https://pugjs.org/api/getting-started.html.


"Mongoose JS."
http://mongoosejs.com/.


"Mongo DB."
http://mongoosejs.com/.


"Stack Overflow."
https://es.stackoverflow.com/.

Bibliografía II


"Heroku."
https://devcenter.heroku.com/.


"GitHub Education."
https://education.github.com/.


"NPM."
https://www.npmjs.com/.

Fin de la presentación

Gracias por su atención