



U N I V E R S I D A D
COMPLUTENSE
M A D R I D

Facultad de Ingeniería Informática

APLICACIÓN WEB DE SOPORTE AL APRENDIZAJE-SERVICIO

Autores

Daniela Nicoleta
Jesus Sanchez Granado
Victoria Gnatiuk Romaniuk

Tutores

Simon Pickin
Manuel Montenegro Montes

Índice

1. Resumen

Este trabajo de fin de grado es una continuación del trabajo de fin de grado de David Jiménez del Rey en el que se fundamentan las bases de nuestro proyecto. Esta es la cuarta parte de un proyecto de desarrollo de una comunidad web de Aprendizaje-Servicio.

En este proyecto abarca las fases de análisis, diseño y desarrollo de una aplicación web capaz de gestionar las demandas de servicios sociales de las entidades y las ofertas de servicio de los profesores de las universidades. Este tipo de asociación entre entidades y centros educativos es una metodología de aprendizaje innovadora llamada Aprendizaje-Servicio o APS.

El APS consiste en combinar los conocimientos obtenidos en los centros educativos con los servicios proporcionados por a la comunidad.

Aunque una de las ventajas de estos proyectos es la obtención de competencias profesionales, en realidad su principal objetivo es la reflexión, el pensamiento crítico y la responsabilidad social de los alumnos.

REVISAR—El principal problema de una plataforma orientada para el APS es casar las demandas de las entidades con las ofertas de los profesores. Las entidades suelen ser muy genéricas a la hora de definir requisitos para un proyecto de este estilo y los profesores suelen ofrecer conocimientos más técnicos. Nuestro reto consistía en que la aplicación fuera capaz de casar estas ofertas y demandas y convertirlas en partenariados para posteriormente poder desarrollar un proyecto con dichas características. TO DO (Describir todo lo que se ha hecho).

2. Abstract

3. Introducción

3.1. Pruebas de robustez

Lo primero que hicimos al ponernos a trabajar en el TFG fue realizar una prueba de robustez a la aplicación web de David desplegada en Heroku.

Al hacerlo identificamos varios problemas, algunos más graves que otros. Todos estos problemas los fuimos anotando para poder arreglarlos posteriormente en la etapa de desarrollo.

Uno de los problemas más graves que encontramos fue el bloqueo de la página web causada por varios acontecimientos como la subida de un fichero a un partenariado o proyecto existentes, la subida y posterior eliminación de una foto de perfil o la subida de una imagen a una iniciativa. Nos percatamos de que todos estos problemas tienen en común la tabla de base de datos llamada "uploads". Esto lo tendremos en cuenta a la hora de corregir los problemas posteriormente.

3.2. Diseño de la base de datos

La base de datos se va a cambiar a una base de datos SQL ya que nos damos cuenta que la estructura de la base de datos es una base de datos relacional.

3.3. Objetivos

Realizamos nuestro TFG en base al trabajo de David Jiménez. Nuestros principales objetivos fueron rediseñar la aplicación, para que en el futuro se pueda adaptar a un entorno de ejecución real, e implementar una funcionalidad esencial que es el emparejamiento de ofertas y demandas. A continuación, se detallan estas tareas:

- **Pruebas de robustez:** se realizaron pruebas de depuración al código de David. Se observó que el sistema era muy robusto y estable, pero a pesar de ello se detectaron unos pocos fallos menores que fueron arreglados posteriormente.
- **Diseño del modelo de dominio:** se observó que el diseño de David no estaba preparado para un entorno de ejecución real y por ello se volvió a rediseñar la aplicación teniendo en cuenta su futura integración en un sistema real. Este diseño fue reflejado en un modelo de dominio que permitió reflejar las entidades, los atributos, las relaciones y las restricciones que gobiernan el ámbito de la aplicación.
- **Diseño de la BD:** David había cambiado la BD de MySQL a MongoDB alegando que así podría tener acceso a más documentación. Junto con nuestros tutores, consideramos que la estructura de la aplicación se adapta mejor a una BD relacional y por ello se ha rediseñado la base de datos del sistema a una base de datos SQL.
- **Creación de un DAO:** usando la librería knex de Node.js se ha implementado el patrón DAO para ofrecer una interfaz común entre la aplicación y la base de datos, separando de esta manera la lógica de acceso a datos de la lógica de negocio.
- **Implementación de un sistema de emparejamiento:** se ha implementado la lógica de emparejamiento de las ofertas de servicio, ofertadas por los profesores, y las demandas de servicio, solicitadas por las entidades beneficiarias resolviendo de esta manera el principal problema de esta aplicación.

Referencias