



Universidad
de La Laguna

Escuela Superior de
Ingeniería y Tecnología
Sección de Ingeniería Informática

Trabajo de Fin de Grado

Plataforma para la gestión de exámenes: Social Quiz

Platform for managing exams: Social Quiz

Eduardo Javier Acuña Ledesma

La Laguna, 4 de marzo de 2015

D. **Casiano Rodríguez León**, con N.I.F. 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:

“Plataforma para la gestión de exámenes: Social Quiz.”

ha sido realizada bajo su dirección por D. **Eduardo Javier Acuña Ledesma**, con N.I.F. 78.704.710-K.

Y para que así conste, en cumplimiento de la legislación vigente y a los efectos oportunos firman la presente en La Laguna a 4 de marzo de 2015.

Agradecimientos

A mi tutor de este Trabajo de Fin de Grado, Casiano Rodríguez León, por su apoyo incondicional. He aprendido de uno de los mejores profesores. Sin él no hubiera sido posible.

A la profesora Isabel Sánchez Berriel, por dedicarme ese tiempo tan valioso y prestarme su ayuda. Se lo agradezco muchísimo.

A mi familia, mi novia, mi hermana y en especial a mi madre, porque siempre apostaron por mí y han formado parte de esta etapa tan importante en mi vida.

A todos los compañeros de la facultad, porque he compartido momentos inolvidables y me han ayudado hasta el último momento.

Y finalmente a la escuela, por la calidad de la enseñanza y porque he tenido el privilegio de formarme tanto profesionalmente como personalmente.

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 Grado ha sido el de poner en práctica los conocimientos adquiridos durante los años de estudio del Grado, en especial, los conocimientos acerca del desarrollo de aplicaciones Web, del itinerario de Tecnologías de la Información. Se basa en la puesta en práctica y en el aprendizaje de tecnologías, metodologías, herramientas y técnicas actuales para abordar la compleja labor que es la realización de este tipo de aplicaciones en un marco profesional.

Este proyecto se basa en la gestión, creación y realización de exámenes creados por los propios usuarios del sistema. Existen multitud de aplicaciones de este tipo, como pueden ser SurveyMonkey o BrainCog, y otras que engloban un ámbito mayor, como la conocida Moodle. Todas ellas, ofrecen la posibilidad de crear preguntas con respuestas del tipo verdadero-falso, respuesta corta, selección múltiple, y otros tipos sencillos de respuestas. Sin embargo, intentaremos ofrecer funcionalidades con las que actualmente no cuentan estas plataformas, por ejemplo la posibilidad de creación de otro tipo de preguntas en las que la respuesta es gestionada mediante expresiones regulares.

Palabras clave: Ruby, Sinatra, Sequel, MathJax, JavaScript, AJAX, Bootstrap, OAuth, Lenguaje de Dominio Específico, Mapeo Objeto Relacional, Heroku

Abstract

The purpose of this Final Degree Project has been to implement the knowledge acquired during the degree studies, particularly about Web Applications development in the Information Technology specialty. This specialty focuses on the implementation and learning technologies, methodologies, tools, and techniques to deal with the complex task of developing these applications in a professional context.

This project is based on the management, creation and realization of exams created by system's users. There are many applications of this type, such as SurveyMonkey or BrainCog, and others that covers a larger area, known as Moodle. They all offer the ability to create questions with answers such true-false, short answer, multiple choice, and other simple types of answers. However, we will try to provide extra functionality for these platforms, such as the possibility of creating another type of questions where the answer is managed using regular expressions.

Keywords: Ruby, Sinatra, Sequel, MathJax, JavaScript, AJAX, Bootstrap, OAuth, Domain Specific Language, Object Relational Mapping, Heroku

Índice General

Capítulo 1. Introducción	6
1.1 Motivación	6
1.2 Objetivos y actividades a realizar	6
1.3 Alcance del proyecto	7
1.4 Plan de trabajo.....	8
1.5 Propuesta de evaluación.....	9
1.6 Tecnologías y herramientas utilizadas	9
Capítulo 2. Estado del arte	11
Capítulo 3. Desarrollo	13
3.1 Metodología.....	13
3.2 GitHub.....	13
3.3 Base de datos	15
3.3.1 Diagrama Entidad-Relación	16
3.3.2 Modelo Relacional.....	18
3.3.3 Sequel.....	19
3.4 Sinatra	23
3.4.1 Carpeta public	24
3.4.2 Carpeta views	25
3.5 MathJax.....	28
3.6 Bootstrap	28
3.7 AJAX.....	30
3.8 OAuth.....	31
Capítulo 4. Conclusiones y líneas futuras	33
4.1 Conclusiones.....	33
4.2 Líneas futuras.....	34

Capítulo 5. Summary and Conclusions	35
5.1 Conclusions	35
5.2 Future work	35
Capítulo 6. Presupuesto	37
6.1 Diseño e implementación de la base de datos	37
6.2 Desarrollo web con Sinatra.....	37
6.3 Presupuesto final.....	38
Bibliografía	39

Índice de figuras

Figura 3.1. Pantalla principal del repositorio en GitHub.....	14
Figura 3.2. Ramas utilizadas en el repositorio en GitHub.	14
Figura 3.3. Pantalla de <i>issues</i> abiertos con algunas de las etiquetas utilizadas en el repositorio en GitHub.	15
Figura 3.4. Diagrama Entidad-Relación.	16
Figura 3.5. Modelo Relacional de la base de datos.	18
Figura 3.6. Tabla en Sequel para Usuarios.....	19
Figura 3.7. Tabla en Sequel para Exámenes.....	20
Figura 3.8. Tabla en Sequel para Preguntas.....	20
Figura 3.9. Tabla en Sequel para Grupos.....	20
Figura 3.10. Tabla en Sequel para Usuario_Grupo.....	21
Figura 3.11. Tabla en Sequel para Respuestas.	21
Figura 3.12. Tabla en Sequel para Examen_Pregunta.....	21
Figura 3.13. Tabla en Sequel para Usuario_Examen.	22
Figura 3.14. Tabla en Sequel para Usuario_Examen_Respuesta.....	22
Figura 3.15. Consulta en Sequel para obtener un registro de la tabla Preguntas pasándole como parámetro un idPregunta.....	22
Figura 3.16. Consulta en Sequel para insertar un registro en la tabla Preguntas.	23
Figura 3.17. Consulta en Sequel utilizando las cláusulas SQL JOIN y WHERE.	23
Figura 3.18. Árbol de directorios de Social Quiz.	24
Figura 3.19. Contenido de la carpeta public.....	24
Figura 3.20. Extracto de la página index.	25
Figura 3.21. Contenido de la carpeta views.....	26
Figura 3.22. Extracto de uso del método GET para la ruta /preguntas.....	27

Figura 3.23. Extracto de uso del método POST para la ruta /preguntas.....	27
Figura 3.24. Raíz cuadrada utilizando MathJax con el código $\sqrt{a^2+b^2}$	28
Figura 3.25. Pantalla de creación de una pregunta con Diseño <i>Responsive</i> . ..	29
Figura 3.26. Campo para la introducción de fechas.....	30
Figura 3.27. Campo para la selección de preguntas de un examen.	30
Figura 3.28. Visualización de preguntas utilizando la librería en AJAX.....	31
Figura 3.29. Botón de acceso utilizando una cuenta de Google	32
Figura 3.30. Botón de acceso utilizando una cuenta de Google	32

Índice de tablas

Tabla 1.1. Tabla de la propuesta de evaluación en función de las actividades implementadas.	9
Tabla 6.1. Tabla de duración y precios del diseño e implementación de la base de datos.....	37
Tabla 6.2. Tabla de duración y precios del desarrollo web con Sinatra.	38
Tabla 6.3. Tabla de duración y precios del presupuesto final.....	38

Capítulo 1.

Introducción

1.1 Motivación

El presente proyecto tiene por objeto principal el estudio de la gestión de exámenes online añadiendo preguntas con respuestas que no se incluyen en otras páginas web de este tipo. Como resultado del estudio, se desarrollará una plataforma online empleando esas características y otras indicadas a lo largo de este documento.

La motivación de este proyecto se fundamenta por varias razones:

- **Oportunidad de desarrollo de una plataforma online:** desde el punto de vista educativo hay un fuerte crecimiento en el uso de plataformas online como complemento, o incluso como soporte a la hora de llevar a cabo un aprendizaje.
- **Desarrollo multidispositivo:** el uso de las nuevas tecnologías hace casi obligatorio la adaptación de cualquier página web al dispositivo que la esté visualizando.
- **Autenticación segura con OAuth:** haciendo uso de las cuentas de Google, podremos acceder de forma aún más segura a la plataforma online.

1.2 Objetivos y actividades a realizar

Los objetivos propuestos a alcanzar en este Trabajo de Fin de Grado han sido los siguientes:

- Investigar, dominar y utilizar sistemas de base de datos mediante mapeo objeto-relacional (**ORM**).
- Investigar, dominar y utilizar lenguajes de desarrollo de aplicaciones en el **servidor**.

- Investigar, dominar y utilizar lenguajes y librerías en el cliente para mejorar la experiencia del usuario.
- Investigar, dominar y utilizar **DSLs** para la creación de aplicaciones web.
- Investigar, dominar y utilizar estilos **responsive-design** para la correcta visualización de las páginas web en función del dispositivo.
- Investigar, dominar y utilizar **metodologías ágiles** para el desarrollo del software.
- Investigar, dominar y utilizar mecanismos de **despliegue** de la aplicación web.
- Investigar, dominar y utilizar gemas de **autenticación de usuarios** a través de las cuentas de otras plataformas conocidas.

Las actividades a realizar, descritas en la propuesta de **Modelo de Proyecto de Trabajo de Fin de Grado**, firmada por el director del proyecto y el alumno en la actividad Proyecto de Trabajo de Fin de Grado del “Módulo I: Preparación” de la asignatura, son las que se describen a continuación:

- Estudio de la mecánica de los cuestionarios (ACT01).
- Descripción de requisitos y análisis (ACT02).
- Estudio de las tecnologías requeridas (ACT03).
- Realización del modelo Entidad-Relación y del modelo Relacional de la base de datos (ACT04).
- Implementación de la funcionalidad básica (ACT05).
- Mejora de la funcionalidad de la aplicación web (ACT06).
- Implementar la parte “social” de la aplicación; red social (ACT07).
- Testear el comportamiento de la plataforma (ACT08).
- Preparación de la memoria y presentación (ACT09).

1.3 Alcance del proyecto

El alcance comprenderá lo establecido en el capítulo 1.2, pág. 6. Aun así, se llevarán a cabo una serie de iteraciones en las que generarán los siguientes entregables:

- **Entregable 1:** creación de la estructura básica (pantallas principales)
- **Entregable 2:** implementación de la funcionalidad básica
- **Entregable 3:** mejorar la funcionalidad y autenticación con OAuth

1.4 Plan de trabajo

En la realización del Trabajo de Fin de Grado, los tiempos que se estiman para cada una de las actividades son los siguientes:

Febrero: ACT01, ACT02 y ACT03.

- Estudio de la documentación de las tecnologías a utilizar, como:
 - DataMapper
 - Sequel
 - Sinatra
 - Ruby and Rails
- Estudio de los distintos cuestionarios online existentes.
- Descripción de los requisitos de la plataforma a desarrollar.

Marzo y Abril: ACT04 y ACT05.

- Durante estos meses se realizarán las siguientes tareas:
 - Diseño del modelo Entidad-Relación de la plataforma web.
 - Diseño del modelo Relacional a partir del modelo Entidad-Relación.
 - Desarrollo inicial de la web con la funcionalidad básica.

Mayo: ACT06, ACT07 y ACT08.

- En este mes, se pretenden abordar las siguientes tareas:
 - Implementación de más tipos de respuestas.
 - Mejora de funcionalidad y diseño de la web.
 - Presentación de una primera aplicación testeada.

Junio: ACT09.

- Para el último mes:
 - Finalización de la memoria del TFG y presentación.

1.5 Propuesta de evaluación









Se muestra la propuesta de evaluación haciendo referencia a las actividades indicadas en la sección 1.4, pág. 8:







Si se consigue	Calificación
ACT01, ACT02, ACT03, ACT04, ACT05	0-5
ACT06, ACT07	5-8
ACT08	8-10

Tabla 1.1. Tabla de la propuesta de evaluación en función de las actividades implementadas.

1.6 Tecnologías y herramientas utilizadas

A continuación se mencionarán las tecnologías utilizadas a lo largo del desarrollo del Trabajo Fin de Grado, utilizando como base del desarrollo el lenguaje de programación Ruby.

Ruby		Lenguaje de programación
Sinatra		DSL para la creación de páginas web
Sequel		ORM para la implementación de la base de datos
Haml		Lenguaje de marcado ligero
Bootstrap		Framework HTML, CSS y JS para la creación <i>responsive</i>
CSS3		Hojas de estilo en cascada
HTML5		Lenguaje de marcado para la elaboración de las páginas web
JavaScript		Lenguaje de programación interpretado del lado del cliente

jQuery		Librería JavaScript que añade funcionalidad a la hora de elaborar páginas web
MathJax		Motor de visualización JavaScript para la visualización de fórmulas matemáticas
OAuth		Protocolo de autenticación segura para una API
Heroku		Plataforma de computación <i>Cloud</i>
Github		Repositorio utilizando el sistema de control de versiones Git
Google Drive		Servicio de alojamiento de archivos de Google

Capítulo 2.

Estado del arte

La **World Wide Web** (WWW) publicada en 1992, ha ido sufriendo una serie de continuos cambios. Desde páginas web estáticas, pensadas para mostrar información permanente, pasando por páginas web dinámicas enriquecidas con **JavaScript**, se llega a lo que actualmente se conoce como **Web 2.0**, en la que los usuarios pasan de ser usuarios pasivos a activos, participando y contribuyendo en la generación de contenido en la red, a través de las redes sociales, las wikis, blogs, etc.

Desde el punto de vista educativo ha habido también un cambio con respecto a la manera y medios utilizados para impartir la enseñanza. Se han creado portales educativos mediante video-tutoriales, blogs, presentaciones, baterías de ejercicios resueltos, entre otros, permitiendo al usuario poder recibir formación sin necesidad de estar físicamente en un aula. Otro ejemplo de enseñanza online que está teniendo bastante éxito es la llamada **gamificación** (ludificación), basándose en introducir estructuras provenientes de videojuegos para convertir la actividad, en principio aburrida, en otra atractiva que motive al usuario a seguir usándola con el fin de cumplir unos objetivos.

Existen multitud de plataformas orientadas a la creación de encuestas online, y todas ellas poseen un denominador común, que son los distintos tipos de preguntas que pueden llegar a crear.

A continuación, se nombrarán algunos ejemplos de aplicaciones conocidas y no tan conocidas:

- **Moodle**: el claro ejemplo de plataforma para la creación de cuestionarios online, además de otros tipos de contenidos, dado que es una herramienta pensada para complementar el aprendizaje presencial, permitiendo subir archivos, crear foros, bases de datos, chats y algunas otras posibilidades. Es por ello que abarca muchos aspectos y no se centra sólo en la creación de cuestionarios. Es de distribución libre, y permite crear preguntas del tipo respuesta múltiple, verdadero/falso, numérica, calculada, por emparejamiento, etc.

- **SurveyMonkey:** plataforma web enfocada al ámbito estudiantil permitiendo la creación de encuestas educativas, escolares y académicas en línea. Permite la elección de varias categorías como comunidad, educación, eventos, investigación de mercado, y algunas más. Entre las distintas preguntas que se pueden crear están: respuesta corta, menú desplegable, opción múltiple, matriz/escala de valoración, matriz de menús desplegables, ranking, etc.
- **BrainCog:** plataforma que permite la creación de exámenes online con notificación de los resultados. Esta plataforma se presenta como la más limitada de las nombradas hasta el momento, debido a que sólo permite crear preguntas de respuesta corta, quitando funcionalidad y la posibilidad de creación de preguntas más complejas que requieran otro tipo de tratamiento.

Este proyecto, se circunscribe en un ámbito más limitado que el de propia aplicación Moodle: el de la gestión y creación de exámenes similar a algunas plataformas existentes nombradas posteriormente. Intentaremos, a lo largo de esta mejoría, ofrecer funcionalidades con las que actualmente estas plataformas no cuentan aún. Ejemplos de estas carencias son otro tipo de pregunta, en la que la respuesta es gestionada mediante expresiones regulares.

Capítulo 3.

Desarrollo

A continuación, se describirán cada uno de los aspectos importantes llevados a cabo para la realización del Trabajo de Fin de Grado, tanto la metodología seguida durante el desarrollo, como la base de datos y la tecnología implementada.

3.1 Metodología

Durante el desarrollo del Trabajo Fin de Grado, se ha seguido una metodología ágil, muy común en la ingeniería del software basada en el desarrollo iterativo e incremental, con reuniones (iteraciones) semanales la mayoría de las veces, en las que se definían una serie de tareas u objetivos los cuales se presentaban la semana siguiente. De este modo, se han corregido problemas no contemplados en las fases previas del diseño y añadido nuevas funcionalidades.

3.2 GitHub

Como herramienta para la puesta en funcionamiento de la metodología escogida y como repositorio de control de versiones (CVS), se ha utilizado GitHub. Así, todo el código se encuentra disponible para posibles actualizaciones.

Se muestra a continuación una captura de la pantalla principal del repositorio:

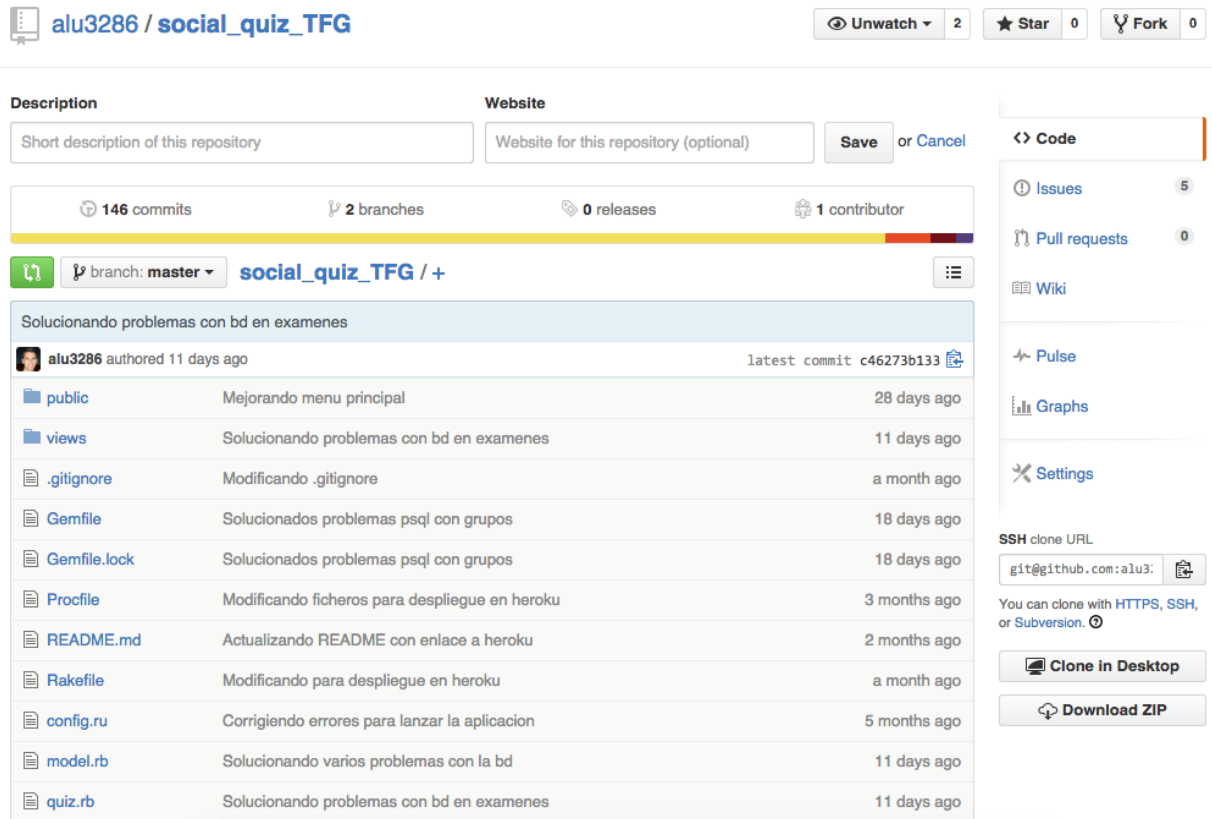


Figura 3.1. Pantalla principal del repositorio en GitHub.

El trabajo se ha dividido utilizando 2 ramas (*branches*):

- **Master:** en ella se aloja el código listo para ser desplegado en producción.
- **Develop:** aloja cualquier modificación para añadir funcionalidades nuevas o corrección de errores.

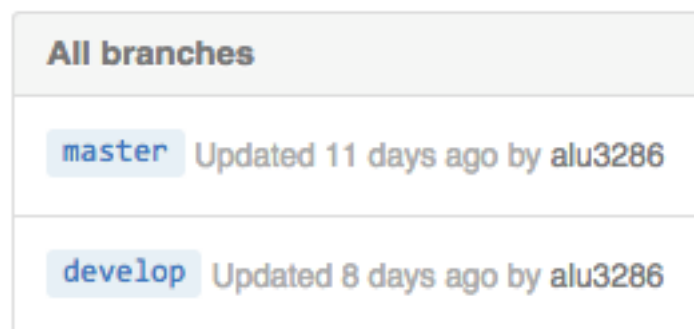


Figura 3.2. Ramas utilizadas en el repositorio en GitHub.

Además, se han creado ramas locales en momentos puntuales por motivos organizativos, permitiendo así una correcta organización de las tareas a realizar.

Aparte de servir de control de versiones, GitHub incorpora la sección *issues*, un *tracking* para organizar las tareas a realizar e hitos a cumplir. A cada tarea añadida se le asigna una etiqueta (*label*) para diferenciar las distintas partes del desarrollo. Las etiquetas usadas son:

- **Feature:** nuevas funcionalidades en local
- **Heroku:** despliegue y testeo de nuevas funcionalidades en Heroku
- **Bbdd:** creación, modificación y/o actualización en la base de datos
- **View:** creación, modificación y/o actualización en las vistas
- **Research:** investigación en un ámbito en concreto

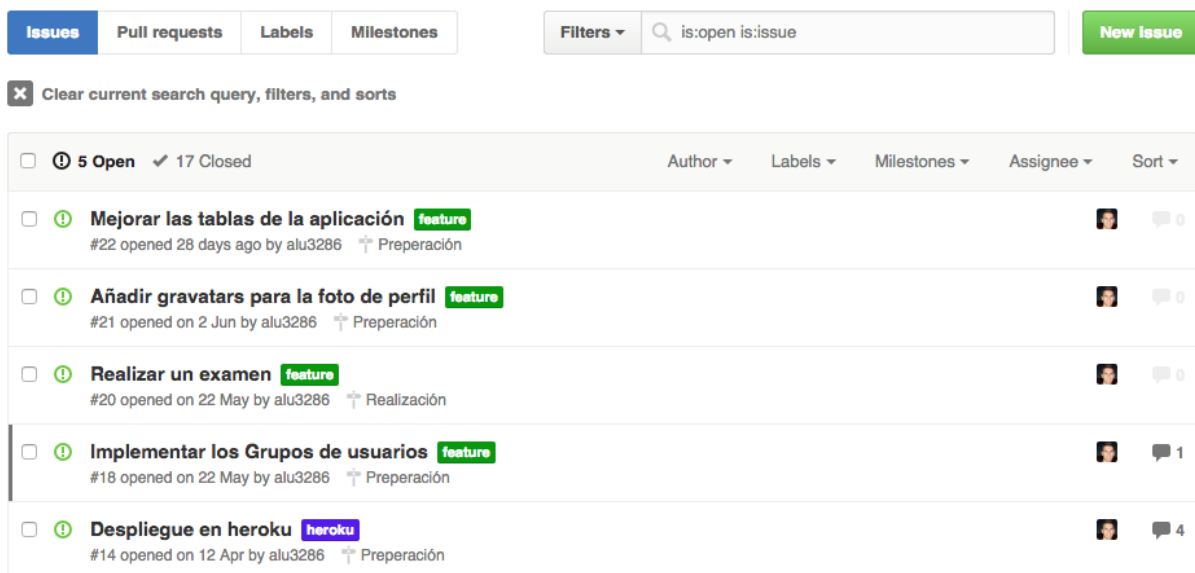


Figura 3.3. Pantalla de *issues* abiertos con algunas de las etiquetas utilizadas en el repositorio en GitHub.

3.3 Base de datos

La plataforma web gira en torno a una base de datos robusta, que ha sido especialmente cuidada elaborando un diseño óptimo. Es la columna vertebral de la plataforma, diseñada casi en su totalidad en las primeras fases del desarrollo.

En ella se pretende almacenar:

- Los usuarios registrados, por lo tanto, los que utilizan la web
- Los grupos
- Las preguntas con sus respuestas correctas
- Los exámenes
- Los usuarios ligados al examen
- Las respuestas que dan los usuarios al examen

Antes de su implementación se ha creado el correspondiente **Diagrama Entidad-Relación**, transformándolo después al **Modelo Relacional**. Estos puntos se explican en los siguientes apartados.

3.3.1 Diagrama Entidad-Relación

A continuación, se muestra el Diagrama Entidad-Relación diseñado para el correcto funcionamiento de la plataforma web, ajustado a las necesidades funcionales:

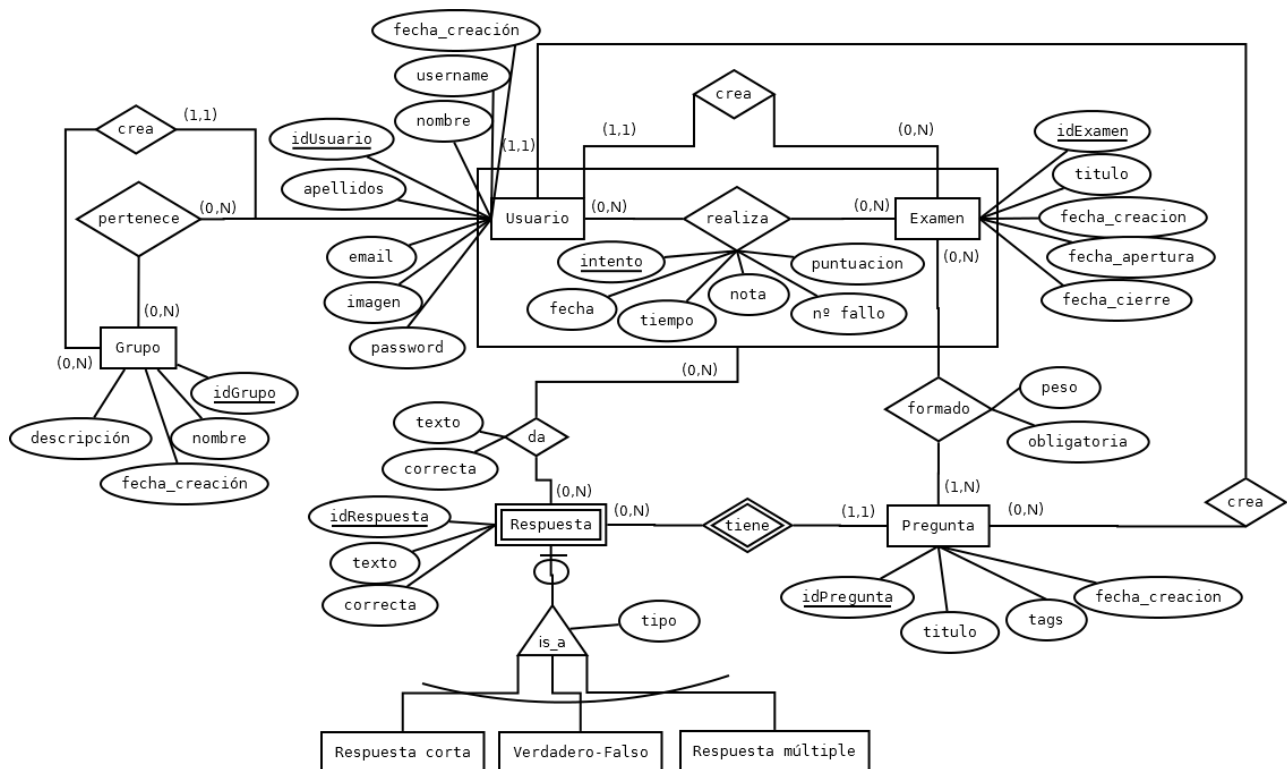


Figura 3.4. Diagrama Entidad-Relación.

Consideraciones previas

- Un usuario puede crear exámenes
- Un usuario puede crear preguntas
- Un usuario puede crear grupos
- Un usuario puede realizar exámenes
- Un usuario puede pertenecer a uno o varios grupos
- Un examen está formado por preguntas
- Cada pregunta está asociada a un tipo de respuesta
- Un usuario que realiza un examen, da una respuesta

Entidades

- Usuario: idUsuario, nombre, apellidos, username, email, imagen, password y fecha_creación
- Examen: idExamen, título, fecha_creación, fecha_apertura y fecha_cierre
- Pregunta: idPregunta, título, fecha_creación y tags
- Grupo: idGrupo, nombre, descripción y fecha_creación
- Respuesta: idRespuesta, texto, correcto y tipo
- Usuario-Examen: se trata de una entidad formada por las entidades Usuario y Examen. Esto se diseñó así para poder decir que un usuario realiza un examen y da una respuesta a ese examen

Relaciones

Se nombran cada una de las relaciones entre entidades:

- Usuario-*realiza*-Examen
- Usuario-*crea*-Examen
- Usuario-*crea*-Grupo
- Usuario-*pertenece*-Grupo

- Usuario-crea-Pregunta
- Examen-formado-Pregunta
- Pregunta-tiene-Respuesta
- Usuario-realiza-Examen-da-Respuesta

3.3.2 Modelo Relacional

Una vez diseñado el Diagrama Entidad-Relación, se hace la conversión a Modelo Relacional, que será el modelo utilizado para la creación de las tablas finales que tendrá la plataforma web. El Modelo Relacional resultante se muestra a continuación:

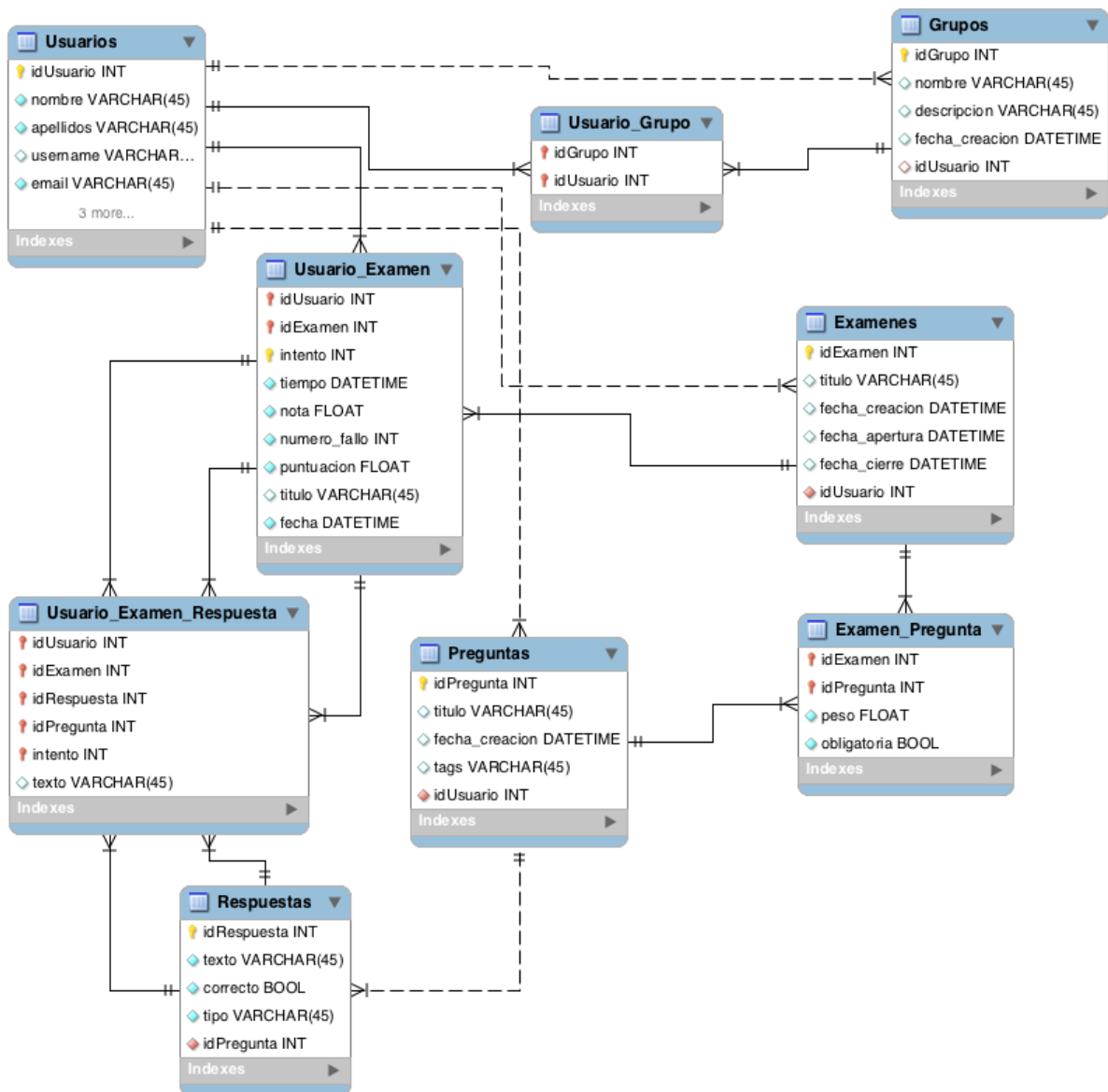


Figura 3.5. Modelo Relacional de la base de datos.

Como se aprecia en la Figura 3.2, se ha creado una tabla por cada entidad del modelo Entidad-Relación. Asimismo, las entidades con relaciones muchos a muchos (X, N) han formado una tabla intermedia.

3.3.3 Sequel

Como se menciona en la sección 1.4, pág. 8, y más concretamente con respecto a la base de datos, se hizo un estudio previo del ORM a utilizar. Entre los ORM investigados existían dos posibilidades: DataMapper y Sequel.

DataMapper es un ORM escrito en Ruby rápido, seguro y funcional. Además, es el utilizado en la asignatura Sistemas y Tecnologías Web del Grado. Tiene una apariencia como cualquier ORM, pero no incluye la calidad documental que sí incorpora Sequel, además de ser este último, mucho más intuitivo y fácil de utilizar.

Por este motivo se decidió utilizar Sequel como ORM en Ruby para la implementación de la base de datos.

A continuación se mostrarán las tablas implementadas en Sequel usadas en la plataforma web:

```
DB.create_table? :usuarios do
  primary_key    :idUsuario
  String         :nombre
  String         :apellidos
  String         :username
  String         :email
  Text           :imagen
  String         :password
  DateTime       :fecha_creacion
end
```

Figura 3.6. Tabla en Sequel para Usuarios.

```
DB.create_table? :examenes do
  primary_key    :idExamen
  String         :titulo
  DateTime       :fecha_creacion
  DateTime       :fecha_apertura
  DateTime       :fecha_cierre
  foreign_key    :idUserario, :usuarios
end
```

Figura 3.7. Tabla en Sequel para Exámenes.

```
DB.create_table? :preguntas do
  primary_key    :idPregunta
  String         :titulo
  DateTime       :fecha_creacion
  String         :tags
  foreign_key    :idUserario, :usuarios
end
```

Figura 3.8. Tabla en Sequel para Preguntas.

```
DB.create_table? :grupos do
  primary_key    :idGrupo
  String         :nombre
  String         :descripcion
  DateTime       :fecha_creacion
  foreign_key    :idUserario, :usuarios
end
```

Figura 3.9. Tabla en Sequel para Grupos.

```

DB.create_table? :usuario_grupo do
  primary_key [:idGrupo, :idUsuario]
  foreign_key :idGrupo, :grupos
  foreign_key :idUsuario, :usuarios
end

```

Figura 3.10. Tabla en Sequel para Usuario_Grupo.

```

DB.create_table? :respuestas do
  primary_key :idRespuesta
  String :texto
  Boolean :correcto
  String :tipo
  foreign_key :idPregunta, :preguntas
end

```

Figura 3.11. Tabla en Sequel para Respuestas.

```

DB.create_table? :examen_pregunta do
  primary_key [:idExamen, :idPregunta]
  Float :peso
  Integer :obligatoria
  foreign_key :idExamen, :examenes
  foreign_key :idPregunta, :preguntas
end

```

Figura 3.12. Tabla en Sequel para Examen_Pregunta.

```

DB.create_table? :usuario_examen do
  primary_key [:intento, :idUsuario, :idExamen]
  Integer     :intento
  Time        :tiempo
  Float       :nota
  Integer     :numero_fallo
  Float       :puntuacion
  String      :titulo
  DateTime    :fecha
  foreign_key :idUsuario, :usuarios
  foreign_key :idExamen, :examenes
end

```

Figura 3.13. Tabla en Sequel para Usuario_Examen.

```

DB.create_table! :usuario_examen_respuesta do
  primary_key [:idUsuario, :idExamen, :idRespuesta, :idPregunta, :intento]
  String      :texto
  foreign_key :idUsuario, :usuario_examen
  foreign_key :idExamen, :usuario_examen
  foreign_key :intento, :usuario_examen
  #foreign_key :idExamen, :examen_pregunta
  #foreign_key :idPregunta, :examen_pregunta
  foreign_key :idRespuesta, :respuestas
  foreign_key :idPregunta, :respuestas
end

```

Figura 3.14. Tabla en Sequel para Usuario_Examen_Respuesta.

Aparte de la creación de tablas, hubo que realizar un estudio de la **API** de Sequel, ya que para realizar consultas simples sí es bastante intuitivo, pero cuando la consulta es más densa, hay que conocer los distintos métodos de los que se disponen. Se mostrarán a continuación algunos ejemplos de consultas utilizadas:

```
@pregunta = DB[:preguntas].first(:idPregunta => params[:num])
```

Figura 3.15. Consulta en Sequel para obtener un registro de la tabla Preguntas pasándole como parámetro un idPregunta.

```
# Añadir la pregunta a la base de datos
@objeto = DB[:preguntas].insert(:titulo => params[:titulo], :fecha_creacion => Time.now,
                                :tags => params[:tags], :idUsuario => session[:id])
```

Figura 3.16. Consulta en Sequel para insertar un registro en la tabla Preguntas.

```
@examen_realizar = DB[:usuario_examen].join(:examenes, :idExamen => :idExamen)
                                .where(:usuario_examen__idUsuario => session[:id])
```

Figura 3.17. Consulta en Sequel utilizando las cláusulas SQL JOIN y WHERE.

3.4 Sinatra

Sinatra es un Software de Dominio Específico (*DSL* en inglés) para aplicaciones web de código abierto, escrito en Ruby. Es una alternativa a otros *frameworks* para aplicaciones web como Ruby on Rails.

Sinatra es pequeño y flexible, y no sigue, a priori, el típico patrón Modelo-Vista-Controlador (MVC) que se ve en otros frameworks como Ruby on Rails. En su lugar, Sinatra se enfoca en la rápida creación de aplicaciones web en Ruby con el mínimo esfuerzo.

Por eso, y porque se utilizó en la asignatura Sistemas y Tecnologías Web del último curso, se decidió a utilizar Sinatra para el desarrollo de la plataforma web.

Algunas destacadas compañías que usan Sinatra son: BBC, 2 Engine Yard, Heroku, GitHub, y Songbird.

A continuación se muestra el árbol de directorios de Social Quiz:

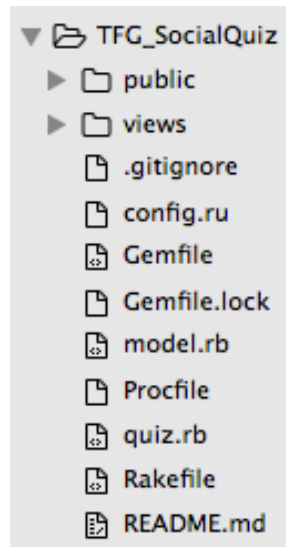


Figura 3.18. Árbol de directorios de Social Quiz.

A pesar de que Sinatra no sigue el patrón **MVC**, se ha seguido esta política, separando lo que es el modelo en el fichero **model.rb** explicado en la sección 3.3.3, pág. 19, las vistas en el directorio **views** y los controladores en el fichero **quiz.rb**.

3.4.1 Carpeta public

Es la carpeta donde se alojan todos los ficheros públicos (estáticos) de la plataforma.

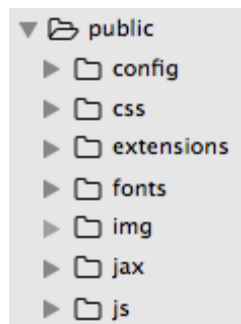


Figura 3.19. Contenido de la carpeta public.

En ella se encuentran los siguientes directorios:

- **config**: ficheros de configuración de MathJax
- **css**: ficheros de estilos de toda la plataforma web
- **extensions**: ficheros de extensiones para MathJax
- **fonts**: fuentes externas añadidas a la plataforma

- **img**: imágenes locales para la plataforma
- **jax**: archivos de MathJax
- **js**: esta carpeta contiene además de los ficheros JavaScript propios de la plataforma, los siguientes directorios:
 - Config, Jax, Moment: ficheros JavaScript para MathJax

3.4.2 Carpeta views

Para el diseño de las vistas, se utiliza **Haml** (*HTML Abstraction Markup Language* en inglés), siendo un lenguaje de marcado ligero que se utiliza para describir de manera limpia el HTML de cualquier documento web sin emplear el código embebido tradicional. Está diseñado para solucionar varios problemas de los motores de plantillas tradicionales y también para ser un lenguaje de marcado tan elegante como sea posible. Haml funciona como reemplazo de sistemas de plantillas de páginas embebidas como PHP, ERB y ASP.

A continuación se muestra un ejemplo de la vista **index.haml**, que es la utilizada para la pantalla de inicio de la plataforma web:

```

index.haml
.container
  .page-header.list-page
    %h2 Inicio

  .row
    .col-md-6
      .lista-resumen
        .list-group
          %a.list-group-item{:href => "/preguntas"}
            %span.glyphicon.glyphicon-pencil
            Preguntas
            %span.badge
              = @preguntas
          %a.list-group-item{:href => "/exámenes"}
            %span.glyphicon.glyphicon-check
            Exámenes
            %span.badge
              = @exámenes
          %a.list-group-item{:href => "/grupos"}
            %span.glyphicon.glyphicon-user
            Grupos
            %span.badge
              = @grupos
          %a.list-group-item{:href => "/exámenes"}
            %span.glyphicon.glyphicon-flag
            Realizar examen
            %span.badge
              = @examen_realizar

```

Figura 3.20. Extracto de la página index.

Como puede observarse, se utilizan etiquetas de una forma más limpia y sin la necesidad de utilizar una etiqueta de cierre. Además, Haml obliga a indentar

correctamente el código, no dejando visualizar la página si no está correctamente indentado.

A continuación se muestra el contenido de esta carpeta:

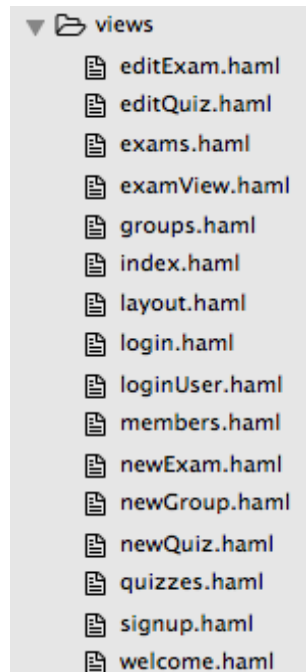


Figura 3.21. Contenido de la carpeta views.

Finalmente, el resto de ficheros se describen a continuación:

- **.gitignore**: define los archivos y directorios que no se suben al repositorio
- **auth.rb**: se encuentra la configuración para el uso de la autenticación mediante OAuth
- **config.ru**: archivo de configuración de Sinatra
- **config.yml**: archivo que contiene el identificador y el secreto para la autenticación con OAuth
- **Gemfile** y **Gemfile.lock**: archivo donde se indican las gemas que va a utilizar la aplicación
- **model.rb**: fichero de configuración del modelo de base de datos usando Sequel
- **Procfile**: fichero donde se declara el comando que ejecutará Heroku en el despliegue

- **quiz.rb:** fichero Ruby de la aplicación. Como se nombró anteriormente, este fichero es el utilizado como controlador, hace uso de Sinatra para implementar los controladores de la plataforma, de tal forma que mediante métodos GET y POST, realiza consultas a la base de datos y obtiene valores que son devueltos a las vistas para su posterior tratamiento.

A continuación se muestra un ejemplo de los métodos GET y POST haciendo uso de Sinatra:

```
get '/preguntas' do
  @actual = "preguntas"
  if (session[:username])
    @preguntas = DB[:preguntas].where(:idUsuario => session[:id]).order(:fecha_creacion).reverse

    haml :quizzes
  else
    redirect '/'
  end
end
```

Figura 3.22. Extracto de uso del método GET para la ruta /preguntas.

Como puede observarse, el método GET es el utilizado cuando se visita la ruta “/preguntas” haciendo uso de Sequel para realizar una consulta a la base de datos y llamando a la vista “quizzes” para que se muestre.

```
post '/preguntas' do
  begin
    # Eliminar la pregunta de la base de datos.
    @objeto = DB[:preguntas].where(:idPregunta => params[:pregunta]).delete
    flash[:mensaje] = "Pregunta eliminada correctamente."

  rescue Exception => e
    puts e.message
    flash[:mensajeRojo] = "No se ha podido eliminar la pregunta. Inténtelo de nuevo más tarde."
  end
  redirect '/preguntas'
end
```

Figura 3.23. Extracto de uso del método POST para la ruta /preguntas.

El método POST es llamado por un formulario del cual se obtienen unos parámetros y para realizar la consulta, en este caso, un DELETE. De este modo, se crean los distintos accesos para las vistas de la plataforma.

- Rakefile: fichero Rake de la plataforma
- README.md: fichero README escrito en Markdown (lenguaje de marcado ligero)

3.5 MathJax

MathJax es una biblioteca Javascript que permite visualizar fórmulas matemáticas en navegadores web, utilizando los lenguajes de marcado LaTeX o MathML. MathJax tiene licencia libre y soporta múltiples navegadores.

MathJax es descargado entre el contenido de una página web, analizando el contenido de esta página para buscar expresiones matemáticas y dibujándolas. Por lo tanto, MathJax no requiere la instalación de ningún tipo de software o de tipos de letras adicionales en el sistema del lector. Esto permite que MathJax funcione en cualquier navegador con soporte JavaScript, incluidos los dispositivos móviles.

Se ha incluido esta librería en el texto de la pregunta a plantear, por lo que es posible redactar preguntas de cualquier ámbito, tanto educativo, como tecnológico, informativo o incluso matemático.

Un ejemplo del potencial que ofrece MathJax se muestra a continuación:


$$\sqrt{a^2 + b^2}$$

Figura 3.24. Raíz cuadrada utilizando MathJax con el código

`\sqrt{a^2+b^2}`.

3.6 Bootstrap

Bootstrap es el framework de Twitter que permite crear interfaces web con CSS y JavaScript de manera que se adapta a la interfaz dependiendo del tamaño del dispositivo en el que se visualice de forma nativa, es decir, utiliza lo que se conoce como **Diseño Responsive** (*Responsive Design* en inglés).

Es el proyecto más popular en GitHub y es usado por la NASA y la MSNBC junto a otras organizaciones.

Social Quiz está completamente desarrollada utilizando este framework, por lo que cada sección de la aplicación es *responsive* además de presentar un acabado visual muy atractivo.

Un ejemplo de una pantalla de la plataforma de muestra a continuación:

SocialQuiz

Pregunta > Nueva

Texto de la pregunta

Introduce un texto para tu pregunta

Previsualización

Tags

Añade tags a tu pregunta

Figura 3.25. Pantalla de creación de una pregunta con Diseño *Responsive*.

Además, aprovechando estas funcionalidades se han añadido una serie de *plug-ins* que mejoran y enriquecen la plataforma:

- Librería para la introducción de fechas utilizando **bootstrap-datepicker**.

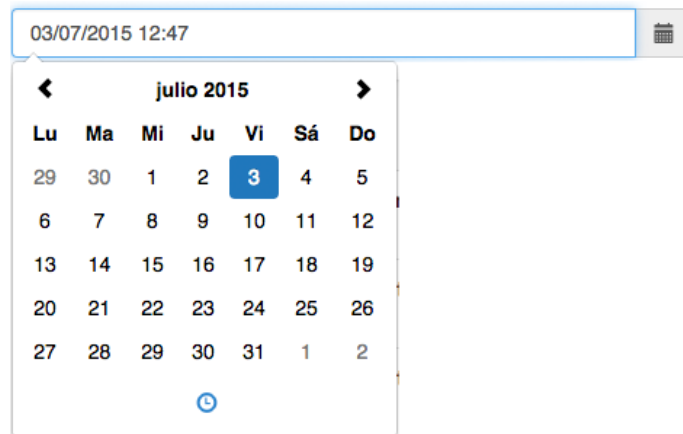


Figura 3.26. Campo para la introducción de fechas.

- Librería para la selección de preguntas de un examen utilizando **bootstrap-dual-list**.

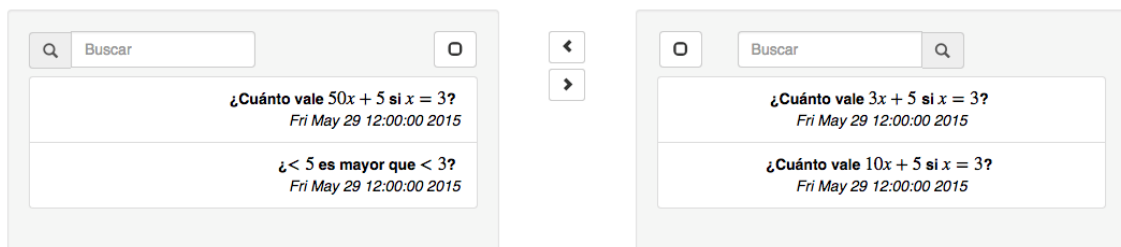


Figura 3.27. Campo para la selección de preguntas de un examen.

3.7 AJAX

AJAX permite mejorar completamente la interacción del usuario con la aplicación, evitando las recargas constantes de la página, ya que el intercambio de información con el servidor se produce en un segundo plano.

Las aplicaciones construidas con AJAX eliminan la recarga constante de páginas mediante la creación de un elemento intermedio entre el usuario y el servidor. La nueva capa intermedia de AJAX mejora la respuesta de la aplicación, ya que el usuario nunca se encuentra con una ventana del navegador vacía esperando la respuesta del servidor.

El conjunto de tecnologías que conforman AJAX son:

- XHTML y CSS: para crear una presentación basada en estándares

- DOM: para la interacción y manipulación dinámica de la presentación
- XML, XSLT y JSON: para el intercambio y la manipulación de información
- XMLHttpRequest: para el intercambio asíncrono de información
- JavaScript: para unir todas las demás tecnologías

Además de AJAX, se utiliza JavaScript para la interacción de algunos elementos de las páginas.

JavaScript es un lenguaje de programación que se utiliza principalmente en su forma del lado del cliente (*client-side*) para crear páginas web dinámicas. Es un lenguaje interpretado, por lo que no es necesario compilar los programas para ejecutarlos.

Existen multitud de plug-ins que se apoyan en AJAX para mejorar elementos específicos en una página web. En el desarrollo de este Trabajo de Fin de Grado, se ha utilizado un plug-in AJAX para la representación de los listados, llamado **Table plug-in for jQuery**, que mejora las tablas HTML añadiendo controles avanzados. Se muestra a continuación un ejemplo de un listado en Social Quiz utilizando esta librería:

ID	Pregunta	Creado	Usuario	Opciones
1	¿Cuánto vale $3x + 5$ si $x = 3$?	Friday, 29 de May de 2015, 12:00	edu	  
2	¿Cuánto vale $10x + 5$ si $x = 3$?	Friday, 29 de May de 2015, 12:00	edu	  
3	¿Cuánto vale $50x + 5$ si $x = 3$?	Friday, 29 de May de 2015, 12:00	edu	  
4	¿ < 5 es mayor que < 3 ?	Friday, 29 de May de 2015, 12:00	edu	  

Mostrando 1 de 4 de un total de 4 entradas

Search:

Mostrar 10  entradas

Previous 1 Next

Figura 3.28. Visualización de preguntas utilizando la librería en AJAX

3.8 OAuth

OAuth (*Open Authorization* en inglés) es un protocolo que permite flujos simples de autorización para sitios web o aplicaciones informáticas. Este protocolo permite la autorización segura de una API para aplicaciones de

escritorio, móviles y web, proporcionando un acceso a los datos de los usuarios y protegiendo las credenciales de la cuenta.

De este modo se proporciona acceso a la aplicación usando una cuenta Google:

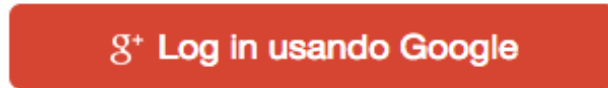


Figura 3.29. Botón de acceso utilizando una cuenta de Google

Para permitir el acceso con una cuenta de Google, tan solo hay que Aceptar las condiciones:

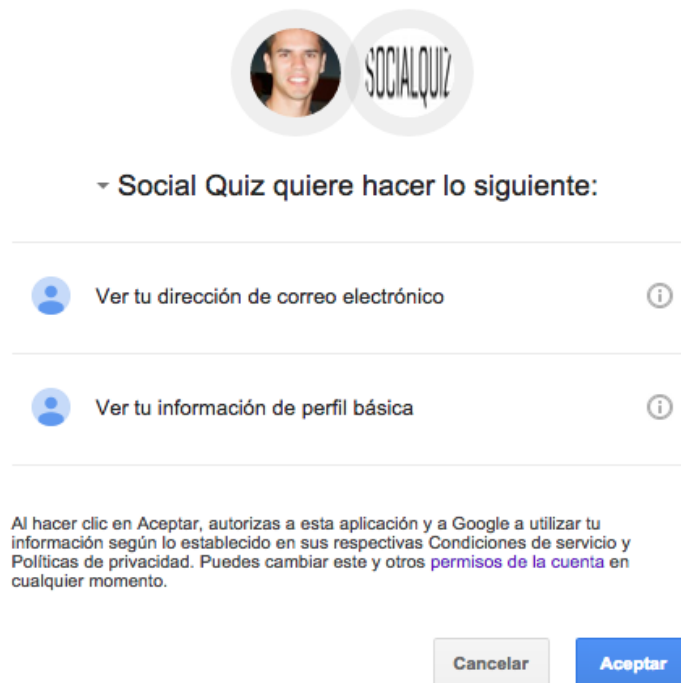


Figura 3.30. Botón de acceso utilizando una cuenta de Google

Capítulo 4.

Conclusiones y líneas futuras

4.1 Conclusiones

Desde siempre he estado interesado en el desarrollo de páginas web y este trabajo surgió como la mejor de las oportunidades para sumergirme en este mundo.

No obstante, supuso un reto en cuanto a complejidad y utilización de tecnologías, ya que aúna todas las tecnologías web vistas en el Grado de Ingeniería Informática, e incorpora otras nuevas surgidas de la necesidad de realizar un trabajo en el marco profesional. No se trata de crear una web estática y añadir contenido, sino que hay que hacerla dinámica, apetecible, elegante, sencilla, que interactúe con la base de datos de una manera casi transparente y que permita su correcta visualización con distintos tipos de pantallas y dispositivos.

En cuanto a los objetivos definidos en las fases iniciales del Trabajo de Fin de Grado, se puede afirmar que se han ido alcanzando cada uno de los ellos, pudiendo así darlo como superando.

Esta plataforma proporciona varias ventajas con respecto a otras plataformas creadas hasta la fecha:

- Permite crear grupos de usuarios para asignarlos a un examen.
- Posibilidad de añadir preguntas evaluadas mediante expresiones regulares.
- Creación de preguntas independientes y reutilización de las mismas en otros exámenes.

4.2 Líneas futuras

Como era de esperar, se ha abierto un amplio abanico de posibilidades, y el desarrollo no finaliza aquí. Queda mucho trabajo que realizar y se pueden añadir más funcionalidades que enriquezca la experiencia del usuario. Por lo tanto, las principales líneas de desarrollo para continuar serían las enumeradas a continuación:

- Ampliar los tipos de respuesta posibles a la hora de crear una pregunta, como por ejemplo, preguntas en las que la respuesta es gestionada mediante código proveído por el profesor.
- Importación de preguntas a través de fichero.
- Creación de un visor estadístico del usuario.
- Añadir seguidores.

Capítulo 5.

Summary and Conclusions

5.1 Conclusions

I have always been interested in web pages development and this project came as the best chance to immerse myself in this world.

However, it was a challenge in complexity and use of technologies, as it combines all the web technologies seen in the Computer Engineering Degree, and incorporates new ones arising from the need to develop a project in the professional context. It is not about creating some static web pages and adding content, but these web pages have to be dynamic, inviting, elegant and simple, to interact with the database in a transparent mode and they have to be displayed correctly in different screens types and devices.

We can say that the objectives defined at the beginning phases of the Final Degree have been completed.

This platform provides several advantages compare to other platforms created to date:

- Create user groups to assign them to a test.
- Ability to add questions evaluated using regular expressions.
- Create independent questions and reuse them in other tests.

5.2 Future work

As expected, it has opened a wide range of possibilities, and development does not end here. There is still a lot of work to be done like adding more features to enrich the user experience. Therefore, the main lines of development to continue would be listed below:

- Expanding the types of possible answers when the user creates a question, for example where the answer is managed by the teacher.
- Import questions via file.
- Creation of a statistical display user.
- Add followers.

Capítulo 6.

Presupuesto

En este capítulo se detalla el presupuesto de lo que supondría llevar a cabo la realización de este Trabajo de Fin de Grado, en el caso de que fuese un encargo de un cliente.

El coste por cada hora de trabajo establecido es de 15€/hora.

6.1 Diseño e implementación de la base de datos

A continuación se especifican la duración y el precio de todo el desarrollo de la base de datos desglosada en actividades:

Actividad	Duración (horas)	Precio (€)
Estudio inicial de los distintas plataformas de cuestionarios existentes	4	60,00
Especificaciones sobre la herramienta a desarrollar	8	120,00
Elaboración del diagrama Entidad-Relación	20	300,00
Elaboración del modelo Relacional	4	60,00
Estudio de los distintos ORM	1	15,00
Estudio de la API de Sequel	5	75,00
Implementación de las tablas en Sequel	8	120,00
TOTAL	50	750,00

Tabla 6.1. Tabla de duración y precios del diseño e implementación de la base de datos.

6.2 Desarrollo web con Sinatra

Se especifican la duración y el precio del proceso de desarrollo de la plataforma utilizando Sinatra desglosado en actividades:

Actividad	Duración (horas)	Precio (€)
Investigación de los dos OSL propuestos: Ruby and Rails y Sinatra	2	30,00
Configuración de la base de datos con Sinatra	24	360,00
Implementación de preguntas	54	810,00
Implementación de exámenes	80	1200,00
Implementación de grupos	40	600,00
Mejoras de la plataforma mediante AJAX y JavaScript	45	675,00
Implementación de la realización de un examen	40	600,00
Configuración de OAuth	8	120,00
Mejoras en el diseño final	24	360,00
TOTAL	317	4755,00

Tabla 6.2. Tabla de duración y precios del desarrollo web con Sinatra.

6.3 Presupuesto final

Se especifican la duración y el precio del presupuesto final del Trabajo de Fin de Grado:

Actividad	Duración (horas)	Precio (€)
Diseño e implementación de la base de datos	50	750,00
Desarrollo web con Sinatra	317	4.755,00
TOTAL	367	5.505,00

Tabla 6.3. Tabla de duración y precios del presupuesto final.

Bibliografía

- [1] Sinatra. <http://www.sinatrarb.com/>.
- [2] Sequel: The Database Toolkit for Ruby. <http://sequel.jeremyevans.net/>.
- [3] MathJax. <https://www.mathjax.org/>.
- [4] Bootstrap. <http://getbootstrap.com/>.
- [5] Moodle: Open-source learning platform. <https://moodle.org/>.
- [6] SurveyMonkey: Herramienta gratuita de software y cuestionarios para crear encuestas online. <https://es.surveymonkey.com/>.
- [7] Darren Jones. *Jump Start Sinatra*. SitePoint, 2013.
- [8] Chang Sau Sheong. *Cloning Internet Applications with Ruby*. Packt Publishing, 2010.