

Sistema de evaluación de prácticas con feedback

Llombart Codes, Toni

Resumen– La automatización de procesos, a día de hoy, es una actividad clave en el ámbito profesional y educativo. Por esta razón, lo que se busca en este trabajo es proporcionar a los profesores y estudiantes de la universidad, una herramienta que evalúe automáticamente las prácticas del laboratorio de estudiantes, mostrándoles cómo respuesta los resultados obtenidos. De esta forma, cualquier estudiante de la universidad tendrá a su disposición una plataforma web en la cual podrá obtener un feedback de su práctica al momento por parte del profesor. A más a más, el sistema usará técnicas de gamificación con el fin de aumentar la motivación del estudiante, ocasionando una mejora en la calidad de las prácticas y una mejora en el aprendizaje. Este sistema, gracias al uso de la herramienta Docker, a la programación de Scripting Engines y a las bajas restricciones de funcionalidad, podrá ser usado en cualquier asignatura.

Palabras clave– Tests, Autoevaluación, Gamificación, Container Docker, Prototipo, FormDa-ta, Seguridad Web.

Abstract– The process automation, to date, is a key activity in the professional and educational fields. For this reason, what is sought in this project is to provide teachers and students of the university with a tool that automatically corrects the practices of the student laboratory, showing them how to respond to the results obtained. In this way, any student at the university will have at their disposal a web platform in which they can obtain a feed-back of their practice at the moment by the teacher. Furthermore, the system will use gamification techniques in order to increase student motivation, leading to an improvement in the quality of the practices and an improvement in learning. This system, thanks to the use of the Docker tool, the programming of Scripting Engines and the low functionality restrictions, can be used in any subject.

Keywords– Tests, Self-evaluation, Gamification, Docker Container, Prototype, FormData, Web Security.



1 INTRODUCCIÓN

ACTUALMENTE la mayoría de las prácticas que tienen que entregar los estudiantes no tienen un seguimiento obligatorio por el profesor, teniendo que solicitar tutorías con ellos para poder obtener un feedback, ya que el tiempo en los laboratorios es limitado.

El problema principal que se encuentran los estudiantes es no haber tenido suficientes impresiones, sensaciones o sugerencias por parte del profesor antes de las validaciones. Las dos causas principales por las que los estudiantes no han recibido este feedback son: no asistir a tutorías con los profesores y falta de motivación [11]. El no asistir a tu-

torías se puede deber a incompatibilidad de horarios o no creer conveniente pedir tutoría, ya que las pruebas que habían realizado ellos mismos eran satisfactorias, pero a la vez insuficientes comparadas con las que el profesor usaría para validar la práctica, encontrándose una nota menor a la esperada el día de la entrega.

Para poder solucionar este problema, en este documento se hace una explicación del desarrollo y la creación de un sistema que permitirá al estudiante obtener un feedback del profesor al instante de la práctica, teniendo la posibilidad de mejorarla si procede. Y para que los estudiantes no se conformen con un aprobado, se incluirán técnicas de gamificación [5, 7] para motivar al estudiante a completar las prácticas en su totalidad, obteniendo una calidad mas alta en las prácticas y una mejora en su aprendizaje.

La estructura que sigue el documento para explicar el desarrollo de este sistema es la siguiente: definición de los objetivos que se han marcado para el proyecto, breve documentación de sistemas comunes al que se quiere desarrollar,

- E-mail de contacte: tonillombartcodes@gmail.com
- Menció realitzada: Tecnologies de la Informació
- Treball tutoritzat per: Sergi Robles Martínez (DEIC)
- Curs 2017/18

explicación de la metodología usada, planificación de las tareas del proyecto, redacción del desarrollo del proyecto que estará dividida en tres partes, cada una de estas partes define el desarrollo de un prototipo, y en cada una de ellas se explica las partes de análisis y diseño, implementación, pruebas y mejoras. Y los últimos apartados del documento encontraremos los resultados, conclusiones, agradecimientos y la bibliografía.

Como se acaba de comentar, en el siguiente apartado se redactan los objetivos propuestos para el proyecto.

1.1 Objetivos

En este apartado se redactan los objetivos propuestos, empezando con el objetivo principal seguido de los demás objetivos, indicando para cada uno de ellos el nivel de prioridad para su realización, pudiendo ser: Muy alta, alta, media y baja.

- O1. Desarrollar una herramienta web de validación de prácticas que permita a los estudiantes obtener automáticamente un feed-back por parte de los profesores. (Muy alta)
- O2. Analizar las diferentes fuentes de requisitos con el fin de obtener la totalidad de los requisitos del sistema. (Alta)
- O3. Diseñar los diferentes diagramas y diseños del sistema atendiendo a los requisitos obtenidos. (Media)
- O4. Implementar y codificar los módulos del sistema. (Muy alta)
- O5. Realizar pruebas de integración, diseño y de sistema. (Alta)
- O6. Realizar una prueba piloto con una de las prácticas de la asignatura de Xarxes. (Baja)

Estos son los objetivos del proyecto, ahora en el siguiente apartado veremos que sistemas hay ya en el mercado o creados que podrían ser usados como referencia para este.

2 ESTADO DEL ARTE

En este apartado se definen los conceptos y las ideas más relevantes del trabajo, juntamente con algunos de los programas, herramientas o aplicaciones ya existentes que están relacionadas con estos conceptos e ideas.

Una de las características principales de nuestro proyecto es que ha de poder gestionar con facilidad elementos, información y datos de la universidad, y que sea fácil de usar para los estudiantes y los profesores. Un buen programa que sirve para la gestión de cursos educativos es Moodle [13]. Este programa se está usando en la actualidad, es un buen programa gestión del material para los cursos pero no tiene la funcionalidad principal propuesta en este trabajo, corregir prácticas automáticamente.

La búsqueda de programas que se asemejen al de este proyecto ha sido difícil, ya que es algo muy específico, pero se han podido encontrar dos herramientas que se están usando en dos universidades que su objetivo principal es el mismo que el de este trabajo. La primera herramienta se llama

Autotest [14], ha sido desarrollada en la Universitat Politècnica de Catalunya, y es un módulo creado y agregado para Moodle. Este sistema es muy parecido a lo se quiere crear, pero solo funciona con ciertas prácticas de una asignatura ya creadas y configuradas. La segunda es una plataforma web [12], desarrollada para la asignatura de Compiladores de la Universitat Autònoma de Barcelona, pero al igual que la anterior, solo tiene la capacidad de funcionar con prácticas específicas de esa asignatura, y el feed-back muestra al estudiante, es la puntuación obtenida.

En este proyecto, se va a implementar la funcionalidad de autocorrección de prácticas de estudiantes, pero no tan solo para unas pocas prácticas específicas, sino para cualquier tipo de prácticas, consiguiendo que sea un sistema mucho más flexible y fácil de gestionar que los anteriores. En adición, para que sea un sistema completamente distinto a los anteriores, se implementarán técnicas de gamificación para aumentar la motivación de los estudiantes.

En este apartado se han expuesto alguno de los sistemas ya existentes que son parecidos al que se va a desarrollar en este trabajo. En el siguiente apartado se hablará sobre que metodología se va a usar en el proyecto.

3 METODOLOGÍA

En este apartado se explica cual es la metodología elegida para el proyecto con la que podremos lograr todos los objetivos listados en el apartado anterior, y también se comentan que aplicaciones y programas se han usado como herramientas de trabajo en este proyecto.

La metodología elegida para el desarrollo de este proyecto es una metodología basada en el desarrollo evolutivo: el modelo de prototipos [3]. Se ha elegido esta metodología ya que las ventajas y puntos fuertes que ofrece se ajustan mucho a las necesidades y recursos del proyecto. Para que esta metodología sea efectiva, el proyecto debe ser un sistema con el que se pueda experimentar, deba desarrollarse rápidamente, el equipo de desarrollo sea reducido, el sistema no tiene que estar totalmente acabado para poder mostrárselo al cliente, y otras características que nuestro proyecto cumple. El proyecto a desarrollar se dividirá en tres partes principales:

1. Desarrollo del sistema con la funcionalidad básica utilizando un sólo test.
2. Avanzar en la implementación de las funcionalidades en los módulos de profesores y estudiantes. Realizar pruebas con tests más complejos. Usar mecanismos de gamificación [8] en la vista del estudiante que siukllen: ranking, niveles, puntuaciones, premios, logros, progresión en la práctica, entre otros.
3. Implementación total de las funcionalidades del sistema. Realización de pruebas en el entorno final con tests que se usarán en las validaciones.

Para cada una de estas partes se realizará un prototipo. Estos prototipos serán prototipos reutilizables ya que el esfuerzo efectuado en la creación de cada parte será usado para la creación del siguiente y finalmente para el sistema final. Para cada prototipo, se realizarán las etapas de análisis, diseño, implementación, evaluación, mejoras del prototipo y realización de pruebas.

Para la gestión de la creación de estos prototipos, se usaran diferentes herramientas que ayudarán a la finalización satisfactoria del proyecto. Se usará la herramienta jira para facilitar la gestión de las tareas del proyecto. También se tendrá un control de versiones gracias a la herramienta Github. Y para la codificación del sistema se usarán herramientas como WebStorm o Sublime Text.

En este apartado se ha explicado que metodología se ha elegido y el porqué. También se han definido los diferentes prototipos que se realizarán en el proyecto. En el siguiente apartado se situarán en el tiempo estos prototipos.

3.1 Planificación

En este apartado se definen las actividades que se realizarán para alcanzar los objetivos listados en la sección 2. Para representar esta información se ha creado un diagrama de Gantt que se puede encontrar en *A.1 Diagrama de Gantt*. Las actividades definidas con un rombo identifican actividades que corresponden a tareas de la gestión del proyecto. Y las actividades representadas con una barra son actividades del desarrollo del proyecto. La anchura de las barras representa el rango de fechas propuesto para finalizar esa tarea. Y las horas en la columna de la izquierda indican el tiempo previsto que vamos a necesitar para acabar esa actividad. En algunas actividades donde se define el nombre de la actividad, se puede observar con qué objetivo está relacionada. El Cuadro 1, representa la información del diagrama de Gantt en una tabla.

Cuadro 1: TABLA DEL DIAGRAMA DE GANTT

Id	Tarea	Dur.	Fecha Fin	Obj. Rel.
1	Planificación	10h	27 Feb	
1.1	Definir objetivos	10h	27 Feb	O2
2	Análisis requerimientos	5h	2 Mar	
2.1	Identificar necesidades	2h	2 Mar	O2
2.2	Priorizar y seleccionar necesidades	2h	2 Mar	O2
2.3	Redactar requerimientos	1h	2 Mar	O2
3	Primer prototipo	80h	30 Mar	
3.1	Análisis y diseños	10h	6 Mar	O2, O3
3.2	Desarrollo prototipo	45h	20 Mar	O4
3.3	Pruebas	10h	23 Mar	O5
3.4	Mejoras	15h	30 Mar	O4
4	Segundo prototipo	70h	18 Abr	
4.1	Análisis y diseños	5h	2 Abr	O2, O3
4.2	Desarrollo prototipo	50h	13 Abr	O4
4.3	Pruebas	5h	14 Abr	O5
4.4	Mejoras	10h	18 Abr	O4
5	Tercer prototipo	88h	5 Jun	
5.1	Análisis y diseños	8h	20 Abr	O2, O3
5.2	Desarrollo prototipo	60h	25 Jun	O4
5.3	Pruebas	20h	5 Jun	O5, O6

En el siguiente apartado se explican todas las actividades desarrolladas.

4 DESARROLLO DEL PROYECTO

Seguidamente se expondrá en este apartado el proceso de realización del proyecto, dando una explicación de las actividades más importantes de cada una de las partes en la que se ha dividido el proyecto. El orden que siguen las actividades es el mismo que el planteado en la planificación, empezando por inicio y planificación del proyecto, seguido de análisis del sistema actual y requisitos. Y para finalizar, se explica el desarrollo de los tres prototipos desarrollados, y en cada uno de ellos las etapas de análisis, implementación, pruebas y mejoras.

4.1 Inicio y planificación

En esta primera fase se realizaron varias reuniones con el cliente para poder entender que problema tenían y que querían desarrollar para solucionar este problema. De estas reuniones se obtuvieron los requisitos del sistema. Una vez obtenidos estos requisitos, se pudieron planificar las tareas que se deberían realizar en los prototipos para poder acabar construyendo el sistema final.

En la siguiente fase de análisis, se explica que tecnologías, lenguajes, librerías, etc. Se eligieron para la creación del proyecto.

4.2 Análisis y diseños

En esta segunda fase, se definen los requerimientos funcionales y no funcionales, y luego se explican todas las tecnologías, lenguajes de programación, librerías y herramientas que se usarán para crear el sistema e implementar todas sus funcionalidades.

Los requerimientos funcionales se pueden encontrar en un diagrama de casos de uso definido en *A.2 Diagrama de casos de uso*, aunque en los apartados de análisis y diseño de cada prototipo se indican cuales son los que van a ser implementados. Y los requerimientos no funcionales se pueden encontrar en *A.3 Requerimientos no funcionales*.

Al no estar usando ningún sistema o herramienta parecida al sistema a desarrollar, se hizo una búsqueda de sistemas existentes que su uso fuera parecido a lo que se necesitaba, o bien, que tuvieran algunas de las características no funcionales de las que necesitábamos. Estos sistemas se pueden encontrar en el apartado "Estado del arte". Se hizo una búsqueda de tecnologías, lenguajes, patrones, herramientas y librerías que fueran las más apropiadas para la implementación del sistema. Se decidió que se iba a usar un servidor Apache donde ejecutar el sistema ya que es muy portable, tiene una gran comunidad que da soporte, es open source y dispone de grandes funcionalidades para poder trabajar juntamente con una base de datos MySQL [4]. Se utilizará la herramienta Docker y sus containers [9] para obtener un entorno aislado y poder controlar los recursos destinados a las ejecuciones de las prácticas de los estudiantes y mantener seguro nuestro sistema. También se decidió que la web seguiría el patrón Modelo Vista Controlador (MVC) [10]. Y para concluir se hizo una búsqueda de librerías JavaScript para simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones, agregar interacción con la técnica AJAX a páginas web y mantener un diseño responsive [2].

En el siguiente apartado se explican los pasos que se han realizado para la creación el primer prototipo.

5 PRIMER PROTOTIPO

En esta sección, se explica las diferentes actividades que se realizaron para crear el primer prototipo, empezando con el análisis de los requisitos y creación de diseños y diagramas UML. Seguidamente se explica la implementación de estos diseños en el sistema. Después, se desarrolla cómo se ha implementado cada uno de los tres bloques del sistema, la página principal, la vista del profesor y la vista del estudiante. Luego se listan las diferentes pruebas realizadas en el sistema. Y para acabar con la última etapa, se hace un listado de las mejoras y cambios establecidos después de mostrar el primer prototipo al cliente. Estos cambios y mejoras serán realizados para el segundo prototipo.

En el siguiente apartado se describe que se realizó en la primera etapa del primer prototipo.

5.1 Análisis y diseño

En esta primera etapa se indican que diseños se realizaron para la implementación de este primer prototipo y los requerimientos funcionales para este prototipo.

Lo primero que se realizó fue analizar todos los requerimientos obtenidos y generar el diagrama de casos de uso del sistema. Para este prototipo, los requerimientos funcionales que necesitamos implementar son:

- El sistema ha de diferenciar usuarios estudiantes y usuarios profesores.
- Un profesor podrá crear una prueba de tests que será utilizada por el estudiante.
- Estas pruebas se definirán con un nombre, fechas, tiempo de espera y los diferentes tests a ejecutarse.
- Un profesor podrá modificar estas pruebas.
- Un profesor podrá eliminar estas pruebas.
- Un estudiante podrá seleccionar una prueba del profesor subir su práctica al sistema y obtener un feedback de esta cuando se ejecuten los tests en su práctica.
- Solo se le puede permitir al estudiante seleccionar las pruebas que estén dentro de las fechas de oberutra y cierre.

También, se realizaron los principales diagramas para la implementación de este prototipo, los cuales fueron:

1. Diagrama ER para el diseño de la base de datos.
2. Diagrama de secuencia para la funcionalidad base del sistema.
3. Diseños para cada una de las vistas del sistema usando paper prototyping.

En la siguiente fase, se hace una explicación cronológica de las diferentes partes y funcionalidades que se implementaron del sistema a partir de estos diseños.

5.2 Implementación

En este apartado se hace una explicación de cómo se implementaron los requerimientos funcionales listados en el apartado anterior, entrando en detalle de las implementaciones mas importantes y explicando problemas encontrados y cómo se solucionaron. La implementación se dividió en cuatro bloques que englobaban todas las funcionalidades que fueron los siguientes: implementación de la base de datos, implementación de la estructura del sistema, módulo de los profesores y módulo de los estudiantes.

Lo primero que se realizó fue implementar la base de datos siguiendo la estructura del diseño e insertando usuarios y elementos de prueba para poder probar el prototipo una vez terminado. El primer modelo de la base de datos constaba de una tabla para profesores, una tabla para alumnos, una tabla para las pruebas y una última tabla que almacenaría información de los resultados de los estudiantes y las pruebas. Una prueba esta compuesta por un nombre identificativo, una descripción, una fecha de inicio y otra de cierre, los archivos llamados tests que se ejecutaran sobre la práctica del alumno y por último un fichero que será el que ejecutará el container docker y en el cual debe estar el código para que se ejecuten los tests en la práctica del alumno.

Para la implementación de la estructura del sistema, se siguió una estructura MVC, creando tres vistas y tres controladores con sus respectivos modelos. El sistema estaría dividido en la vista de estudiantes, vista de profesores y la vista inicial en la que en esta última se implementó un formulario para que el usuario pudiera entrar al sistema. Este formulario se programó de forma que fuera inmune a ataques de SQL injection o cross site-scripting [1].

La implementación de la parte de los profesores consistió en la generación de tres formularios: uno para la creación de las pruebas ,Figura 1, otro para modificar pruebas ya creadas y otro para eliminar pruebas. El formulario de creación de pruebas permitía subir los diferentes tipos de tests para esa prueba a la vez, ordenándolos antes si fuera necesario.

Figura 1: Formulario profesores prototipo 1

El proceso que se genera en el sistema cuando el profe-

El profesor crea una prueba es el siguiente: se origina un directorio con el nombre de la prueba y dentro se suben todos los test que el profesor ha elegido. También, se sube el fichero de configuración que ejecutará Docker y tres ficheros donde se guarda el orden de ejecución de cada tipo de test. Y por supuesto, se genera una nueva query en la base de datos con la información de esta prueba.

Y en la parte de los estudiantes, se implementó un formulario donde aparecen las pruebas disponibles para ese estudiante. Cuando el estudiante selecciona una prueba y sube su práctica, el proceso que se genera en el sistema es el siguiente: se genera un directorio temporal en el que se almacena la práctica del alumno y una copia de los archivos de la prueba seleccionada. Seguidamente se crea un container Docker con la opción de sistema de archivos compartidos, concretamente se comparte el directorio temporal, e indicando al container que ejecute el fichero de configuración de esa prueba. En *A.4 Generación container Docker* se puede ver el script que genera este proceso.

Una vez el proceso del container termina, se lee el fichero de resultados creado por el fichero de configuración y se muestran por pantalla al usuario, a la vez que se elimina el container creado y el directorio temporal creado al inicio del proceso. Al no tener ninguna práctica ni tests reales, en este primer prototipo se ejecutaba un script que simulaba la creación del fichero con los resultados de los tests. Cuando la ejecución del container terminaba, se leía este archivo de resultados y se mostraba al usuario.

En la siguiente etapa se hace una explicación de las pruebas que se realizaron en el prototipo número uno.

5.3 Pruebas

En la etapa de pruebas se realizaron distintas pruebas en el sistema para verificar que lo que se había implementado era correcto. Se realizaron pruebas de caja negra, pruebas exploratorias y pruebas de usabilidad.

En las pruebas de caja negra, se aplicaron técnicas de valores límites y fronteras, particiones equivalentes y pruebas de casos de uso en las siguientes actividades.

Para las pruebas exploratorias, se realizaron diferentes tipos de entradas de valores en los campos de todos los formularios para forzar diferentes escenarios.

Y para las pruebas de usabilidad, se realizó una prueba con una persona que no sabía cómo funcionaba el sistema, y se le pidió que realizara un proceso de tareas.

Todas las pruebas que se realizaron se podrán encontrar en *A.5 Pruebas prototipo 1*.

En la siguiente y última etapa del primer prototipo, se redactan los cambios y mejoras obtenidas después mostrar el primer prototipo al cliente.

5.4 Mejoras

Finalmente en esta la última etapa del primer prototipo se listan los cambios y mejoras que se obtuvieron al realizar la reunión de seguimiento con el cliente. Estas mejoras serán implementadas en el segundo prototipo:

1. Se va a cambiar la forma de iniciar sesión por parte de los profesores. Ahora, cada asignatura tendrá un usuario específico y único que se le asignará al profesor de esa materia.
2. Se va a añadir el elemento “misión” al sistema. Una misión equivale a una prueba del primer prototipo. Para cada una de las prácticas de cada asignatura, se tendrán varias misiones que se deberán completar para avanzar de nivel y obtener recompensas. En la sección de análisis del segundo prototipo se hará una definición más específica de que son las misiones y que aportan al sistema.
3. Se va a modificar de la vista de los profesores para que la creación de una práctica y sus misiones sea más secuencial.
4. Se mejorará la forma de subir los tests al sistema permitiendo al profesor subir tests para la misma misión que estén en directorios distintos de su máquina.
5. Se va a modificar el diseño de la base de datos permitiendo a un profesor crear N prácticas para la asignatura y para cada práctica N misiones. Con esto se conseguirá que el sistema sea mucho más dinámico.
6. Se añadirá otra variable de control a parte del tiempo de espera por cada intento, las misiones también podrán tener un número máximo de intentos.
7. Las misiones podrán ser obligatorias o no. Si son obligatorias, hasta que no se supere esa misión no se permitirá al usuario avanzar a la siguiente.
8. Se creará una nueva funcionalidad para que el estudiante obtenga un logro oculto al realizar un test especificado por el profesor.
9. Se añadirán nuevos componentes de gamificación para que los estudiantes se involucren más en el proyecto. Cómo por ejemplo: logros, coleccionables, contenido desbloqueable, niveles y ranking.
10. Se va a implementar un "muro" con los avances de todos los estudiantes simulando un ranking de puntuaciones.

Esta ha sido la última etapa que se ha realizado para el primer prototipo. En el siguiente prototipo se volverán a realizar las actividades de análisis, diseño, implementación, pruebas y presentación del prototipo al cliente.

6 SEGUNDO PROTOTIPO

En este apartado se hace una explicación de las funcionalidades que el segundo prototipo logra alcanzar y las etapas que se llevaron a cabo para la realización de este segundo prototipo. Las etapas fueron las mismas que para el primer prototipo: análisis y diseños, implementación, pruebas y mejoras.

La meta de este segundo prototipo es implementar los cambios y mejoras obtenidas de las mejoras del primer prototipo: una mejora en la interface y la funcionalidad de la vista de los profesores, modificación de la estructura de la base de datos para una mejor gestión más flexible y añadir funcionalidades que no se habían previsto en el módulo de estudiantes. A más a más, también se deben realizar las implementaciones previstas en la planificación.

En el siguiente apartado de análisis y diseño se explicará el tipo de análisis, diseños y decisiones que se realizaron para la implementación del segundo prototipo.

6.1 Análisis y diseño

En esta primera etapa del segundo prototipo se explican los cambios necesarios que se realizaron en el sistema para la correcta implementación de las nuevas funcionalidades, obtenidas gracias a la revisión con el cliente del primer prototipo. También se definen los requerimientos funcionales que debían implementarse juntamente con los nuevos obtenidos después del primer prototipo. Los requerimientos funcionales son:

- El profesor debe ser capaz de crear una practica
- Las prácticas están formadas por un nombre, definición, una recompensa y un conjunto de misiones
- El profesor podrá añadir, modificar, eliminar y visualizar misiones en una práctica
- Las misiones se definirán con un nombre, fechas, tiempo de espera, numero de intentos y un conjunto de tests
- El estudiante puede visualizar las recompensas que tiene
- El estudiante puede visualizar el avance que tiene de cada práctica
- El estudiante obtendrá un feedback al elegir una misión y subir su practica

Al analizar los cambios que se debían efectuar para el segundo prototipo, se decidió mover la tarea de realizar pruebas con tests reales para el tercer y último prototipo ya que no iba a dar tiempo al tener que añadir las mejoras obtenidas del primer prototipo.

Antes de empezar la implementación, se rediseño la base de datos en un diagrama ER, se puede encontrar en *A.10 Diagrama Entidad Relación prototipo 2*, para que fuera capaz de abastecer los nuevos requisitos. También se diseñaron nuevos diseños a papel de las nuevas vistas para los profesores y estudiantes.

Una vez hechos los diseños y con los nuevos requisitos se podía pasar a la siguiente fase, en la cual se implementaron todos los cambios y se añadieron las funcionalidades que aún no estaban implementadas.

6.2 Implementación

En este apartado se explica cómo se realizaron todos los cambios, mejoras e implementación de nuevas funcionalidades del segundo prototipo. La implementación se dividió en cuatro grandes tareas que englobaban todas las mejoras, nuevas funcionalidades y cambios, y fueron las siguientes: Implementación de la nueva base de datos, rediseño de la vista del profesor e implementación de mejoras, implementar al completo la vista de estudiantes con las funcionalidades previstas y las obtenidas del primer prototipo, y por último mejorar la ejecución de los containers Docker.

La nueva base de datos, debía de ser más flexible que la anterior y que permitiera gestionar cada práctica de una forma única. Para ello, la nueva base de datos debía de satisfacer los siguientes requisitos: ahora cada profesor puede gestionar y personalizar cada una de las prácticas que quiera de forma única, así que cada asignatura puede contener diferentes prácticas y estas prácticas estarán formadas por unos

elementos llamados “misiones” que equivale a un conjunto de tests que serán ejecutados sobre la práctica del estudiante e información que hace que estas prácticas sean únicas. El objetivo principal que se quería conseguir era una motivación extra al estudiante para realizar las prácticas, haciendo que se embarcará en una aventura única en cada práctica y que pudiera ir avanzando en esta a medida que fuera superando misiones, motivándolo con recompensas, nuevos niveles, misiones desbloqueables, etc. Estas misiones solo pueden ser superadas si la práctica del estudiante está correctamente programada.

Una vez implementada la nueva base de datos, se modificaron funcionalidades de los profesores, como por ejemplo la forma de entrar en el sistema, ahora cada asignatura tendría un usuario administrador, que sería asignado al profesor de esa asignatura. Se modificó también el diseño de la vista de los profesores, haciéndola más simple, elegante y con la capacidad de gestionar prácticas y dentro de cada una de ellas las misiones. Ahora, las misiones equivaldrían a las prácticas del primer prototipo, pero con nuevas opciones como por ejemplo número de intentos que puede realizar un usuario esa misión, la recompensa que obtendrá si la supera o el tipo de misión pudiendo ser una misión obligatoria o no obligatoria. La diferencia entre obligatoria y no obligatoria es que una misión obligatoria debe ser superada antes de poder pasar a la siguiente obligatoria.

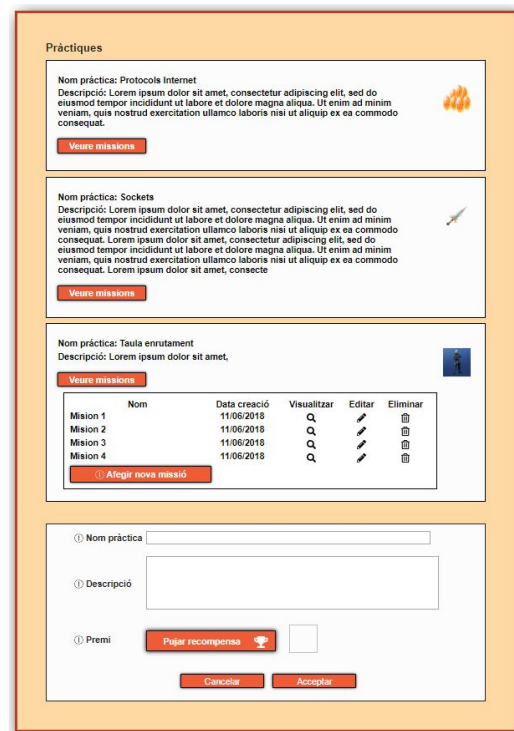


Figura 2: Vista profesor creación prácticas

Terminada la vista de los profesores, Figura 2, se siguió con la implementación de la vista de los estudiantes y todas sus funcionalidades. Ahora en la vista de los estudiantes pueden ver las prácticas que tendrán que realizar a lo largo del curso y para cada práctica, un listado de misiones que tendrán que superar con la práctica que desarrollen para poder completar todas las misiones de esa práctica, Figura 3. Para cada práctica, los estudiantes pueden ver en qué estado se encuentra cada misión, como por ejemplo, si está

superada o no, cuantos intentos quedan para esa misión y el tipo de misión que es. También pueden ver el proceso que llevan de cada práctica y cuanto les queda para llegar a completarla. Además, hay una zona en la que pueden ver las recompensas obtenidas al completar misiones y cada una de las prácticas.

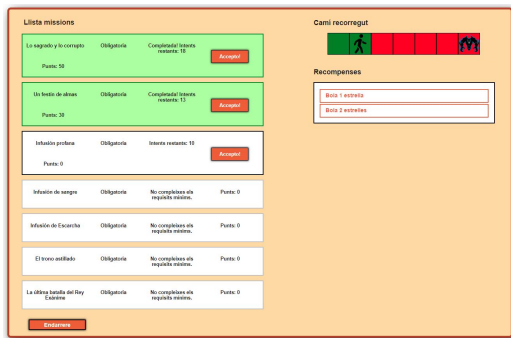


Figura 3: Vista estudiante seleccionando misión

Y por último, en esta etapa de implementación se ha mejorado el proceso de creación y ejecución de containers docker. Ahora el proceso se ejecuta de forma remota gracias al modo detached de docker y a una mejora en la llamada del script que ejecuta este comando. Esto permite que el estudiante una vez confirme el comienzo del proceso de validación de la práctica, pueda seguir trabajando sin que se le quede la pantalla congelada. Con esta mejora, se ha implementado en la barra de menú del estudiante un espacio de notificación que avisa al estudiante cuando ha acabado el proceso y puede obtener los resultados. El proceso que sigue el sistema para poder obtener los resultados a cambiado ya que ahora el proceso no es secuencial y necesita ser notificado una vez se haya creado el archivo de resultados en el directorio. Ahora, una vez se crea el container docker, el sistema pasa a ejecutar una función recursiva cada cierto tiempo. Esta función se encarga de comprobar si existe el fichero de resultados, en el caso que no exista esperará un cierto tiempo y volverá a probar, así hasta que el proceso del container docker genere el fichero y pueda leerlo y mostrar los resultados. Esto nos permite también la opción de poder parar el proceso del container docker si ha pasado cierto tiempo y no se ha generado el fichero con los resultados, y poder informar al estudiante que ha surgido un error y se ha cancelado la ejecución.

En la siguiente etapa de pruebas, se explican todas las pruebas efectuadas en el prototipo dos.

6.3 Pruebas

En este apartado se explican los tipos de pruebas realizados sobre el prototipo número dos y para cada una de ellas se definen las técnicas aplicadas para intentar encontrar errores.

Las pruebas que se realizaron en este segundo prototipo han sido del mismo estilo que para el primero, siendo estas pruebas de caja negra, pruebas exploratorias y pruebas de usabilidad. En las pruebas de caja negra, se aplicaron técnicas de valores límites y fronteras, particiones equivalentes y pruebas de casos de uso.

En las pruebas exploratorias, se realizaron el mismo tipo de técnicas de valores de entrada que se usaron en el primer

prototipo.

Y para las pruebas de usabilidad, se volvió a realizar una prueba con la misma persona que realizó la prueba en el primer prototipo, pidiéndole que realizara un proceso de acciones distintas.

Todas las pruebas que se realizaron se podrán encontrar en *A.6 Pruebas prototipo 2*.

El siguiente apartado es la última etapa del segundo prototipo, donde se redactan los cambios y mejoras obtenidas después de mostrar el segundo prototipo al cliente.

6.4 Mejoras

Para acabar este prototipo, en esta etapa se listan las que serán las últimas mejoras, que se implementarán en el último prototipo, obtenidas del segundo prototipo.

1. Añadir el elemento de 'nivel' al usuario del estudiante. Los niveles otorgan privilegios a los estudiantes.
2. Añadir 'puntuaciones' a las misiones según cuantos tests haya superado el estudiante.
3. Los tests ocultos podrán ser o no ser imprescindibles para pasar la misión.
4. Permitir al estudiante gastar todos los intentos en una misión si esta ha sido superada pero no con la máxima puntuación.
5. Habilitar un tablero donde se muestren todos los grupos de estudiantes de esa asignatura ordenados según sus puntuaciones.
6. Generar un archivo que simule una práctica de estudiante, varios tests y un archivo de configuración que permita ejecutar todos los tests sobre la práctica, simulando el proceso real que deberá realizarse.
7. Cambiar el diseño de la página principal para que no se centre en una sola temática y que muestre información del sitio web.

El siguiente apartado se define el último prototipo, como en los prototipos anteriores, se hará una explicación de las funcionalidades que ha de contener, y se explican las etapas de análisis, implementación y pruebas.

7 TERCER PROTOTIPO

A continuación se describen las etapas que se van a realizar en este último prototipo y se hace una explicación de las últimas características que debe implementar el sistema.

Como los prototipos anteriores, ha seguido las mismas etapas de análisis, implementación y pruebas, pero la última etapa de mejoras no se ha realizado ya que no habrá más prototipos. Al finalizar este prototipo el sistema contendrá todas las funcionalidades propuestas al inicio del proyecto y las añadidas en las etapas de mejoras. Y también, se crearán archivos con la misma estructura que deberán seguir los tests, archivos de configuración y prácticas reales que usaran los profesores y estudiantes, para poder hacer pruebas y obtener resultados mucho más reales.

El siguiente apartado es la primera etapa del último prototipo, en la cual se listarán los requerimientos funcionales

que se van a implementar y diseños que se han utilizado para este prototipo.

7.1 Análisis y diseño

En esta etapa se hace un listado de los requerimientos funcionales que se van a implementar en este prototipo, obtenidos del análisis principal del proyecto y de las mejoras del segundo prototipo. También una lista de los diseños usados a lo largo de este prototipo.

Los requerimientos funcionales que debe cumplir este prototipo a parte de todos los de los prototipos anteriores son:

- Los estudiantes han de ver las puntuaciones de cada misión que han realizado.
- La cuenta de iniciar sesión de los estudiantes irá subiendo de nivel a medida que vayan completando misiones.
- Los estudiantes tendrán la capacidad de reclamar recompensas que dependerán del nivel de los estudiantes.
- Los tests ocultos pueden ser necesarios para completar la misión.
- Los estudiantes obtendrán los resultados de una práctica aunque hayan cerrado sesión o el sistema mientras el proceso de validación estaba en curso.
- Los estudiantes podrán gastar todos los intentos en una misión aunque ya esté superada con el fin de obtener la máxima puntuación.
- Los estudiantes podrán visualizar un ranking el cual indica las puntuaciones de cada grupo de esa asignatura.

Y con el fin de implementar estos requisitos y otras características del sistema, se diseñaron los siguientes diseños: un nuevo diagrama ER de la base de datos, un nuevo diseño en papel de la página inicial del sistema, un diagrama de flujo que define los pasos que debe seguir el fichero de configuración de cada misión.

En este apartado se han listado los requerimientos funcionales necesarios para este prototipo y los nuevos diseños que se desarrollaron para luego implementarlos. En la siguiente etapa de implementación se explicarán cómo se implementaron estas nuevas funcionalidades.

7.2 Implementación

En la última etapa de implementación se explican todas las características y funcionalidades finales implementadas en el sistema. Las tareas a realizar se han agrupado en tres y se explicarán en el siguiente orden: implementación del diseño final de las vistas de estudiantes y la página inicial, creación de ficheros de configuración docker y tests, y por último modificación de la lectura de los resultados de los tests.

En la vista de los estudiantes, Figura 4, se han implementado dos nuevas características, la primera ha sido una

tabla en la que el estudiante puede visualizar todos los grupos de prácticas de esa asignatura, ordenados por la suma de puntos obtenidos de las misiones. Esta funcionalidad intenta motivar a los estudiantes con carácter competitivo. La segunda característica implementada ha sido un espacio en el que el estudiante puede obtener recompensas que se irán desbloqueando según el nivel de la cuenta del estudiante. Estas recompensas realizan cambios en la cuenta del estudiante, y estos cambios quedan reflejados de tal forma que los demás grupos de estudiantes son conscientes de quien ha desbloqueado esas recompensas. Por ejemplo una recompensa que se ha implementado ha sido dar la opción al usuario de modificar su nombre de prácticas y que en vez de mostrar este en el tablero de puntuaciones, muestre el nombre que elija.

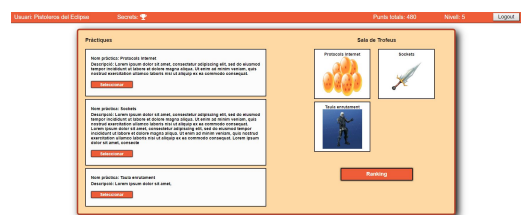


Figura 4: Vista estudiante seleccionando práctica

Por otro lado, también se ha modificado la vista de la página inicial, se han añadido diferentes módulos en los que se ofrece información sobre que es el sistema, para que sirva, que ofrece, que pueden obtener los estudiantes e información que como funciona.

La siguiente tarea fue la creación de los archivos de configuración y tests de docker. Estos serían ejemplos a seguir por los profesores para crear sus propios tests y archivos de configuración. El archivo de configuración sigue el siguiente proceso: primero de todo si es necesario ejecuta los programas o herramientas necesarios para ejecutar la práctica, seguidamente crea un archivo en el que se irán guardando los resultados, para saber que tests tiene que ejecutar, hay tres archivos llamados trivialsFile, basicsFile y ocultosFile, en los que están los nombres de los tests que el profesor adjuntó a esa misión. El archivo de configuración empezará a leer estos archivos y por cada línea ejecutará ese test a la práctica, obteniendo un 1 si lo supera o un 0 si no lo supera, estos resultados, entre otro tipo de información, los irá escribiendo en el fichero que una vez haya acabado de leer las listas de los test y ejecutarlos, se convertirá en el fichero de resultados que detectará el anfitrión a leer y mostrará los resultados al estudiante. En el apartado *A.7 Engine Script* se puede encontrar el código de este proceso.

Y por último, se han realizado dos cambios en la forma de obtención de resultados. El primer cambio ha sido una modificación en el comportamiento de los resultados de los tests ocultos. Ahora el profesor puede programar estos tests ocultos de tal forma que pueden ser necesarios o no para pasar la misión, añadiendo unos valores como resultados que indican al sistema si es un test oculto necesario para pasar la misión o no.

Y el segundo cambio que se ha implementado ha sido una mejora en la forma de obtención de los resultados por parte del estudiante. Antes, si el estudiante decidía salir del sistema dejando el proceso de validación en ejecución, el

sistema automáticamente eliminaba ese container y si el estudiante volvía a entrar con su sesión tenía que realizar el proceso de elgjr misión y subir su práctica otra vez. Ahora, si el estudiante se va del sistema en medio del proceso, el sistema a la que vuelve a entrar el estudiante, comprueba si hay un fichero de resultados para ese estudiante y si lo hay se lo muestra cómo en el proceso normal de validación.

Estas han sido las últimas funcionalidades que faltaban por implementar en el sistema. Ahora en el siguiente apartado, se describirán las pruebas que se han realizado en el sistema para comprobar su correcto funcionamiento.

7.3 Pruebas

Finalmente en esta última etapa del tercer prototipo, se explican los tipos de pruebas realizadas sobre las nuevas funcionalidades del tercer prototipo.

Las pruebas realizadas se han centrado en el proceso de validación de la práctica del estudiante, haciendo pruebas específicas para cada una de las casuísticas posibles en la validación de una práctica y con la comprobación que las puntuaciones sean las correctas. Estas pruebas se han realizado tanto cómo en el caso que el estudiante no cierra sesión y también para cuando el usuario sale del sistema sin esperar a los resultados. En A.8 *Pruebas prototipo 3* se listan estas casuísticas junto con las pruebas que se han realizado.

El siguiente apartado trata sobre los resultados obtenidos una vez realizado el sistema.

8 RESULTADOS

En este apartado se mostrarán los resultados que se han obtenido una vez finalizado el proyecto y realizado diferentes tipos de pruebas para obtener respuesta a los objetivos propuestos.

Se puede afirmar que el sistema cumple el objetivo principal, ya que permite a los estudiantes obtener un feedback de prácticas a través de tests subidos por los profesores, pero con las pruebas que se van a realizar a continuación, se van a evaluar características del sistema que determinarán si cumple algunos de los requisitos no funcionales y en comparación con otros sistemas que aporta de más. Se han efectuado pruebas de resistencia del servidor, pruebas de usabilidad con usuarios, y comparaciones con los sistemas existentes.

Los primeros resultados que se van a mostrar son obtenidos a partir de un gráfico el cual se muestra el tiempo de espera de carga de la página web al conectarse 100 sesiones a la vez. Se han elegido 100 peticiones concurrentes ya que es un valor que sobrepasa el número de grupos de prácticas de una asignatura, solo se tiene en cuenta los grupos de una asignatura ya que para la mayoría de asignaturas tendrán a las mismas personas y no podrían entrar con dos cuentas a la vez.

En el gráfico, figura 5, se puede observar que a medida que se generan más conexiones el tiempo sube. Las peticiones tardan entre 25 y 41 ms, siendo un tiempo aceptable de espera de carga de página.

En la siguiente prueba, Cuadro 2, se muestran las funcionalidades implementadas en el sistema para aumentar la motivación a los estudiantes. Esta tabla sigue el modelo de

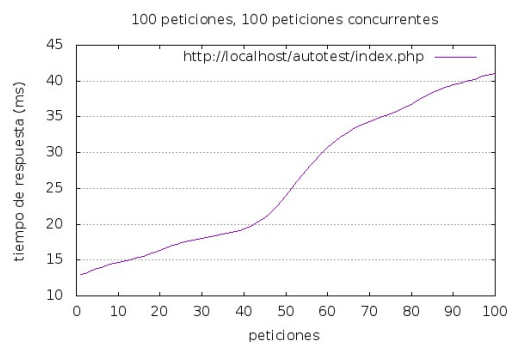


Figura 5: Prueba de estrés al servidor

la teoría de Bartle [15], el cual define cuatro tipos distintos de jugadores.

Cuadro 2: TABLA GAMIFICACIÓN

Tipo Jugador	Funcionalidades
Achievers	Desbloqueables por nivel, recompensas, premios
Explorers	Secretos, recompensas con misiones no obligatorias, visualización del avance
Socializers	Mensajes en las recompensas
Killers	Tabla de puntuaciones, niveles

Finalmente, se realizaron pruebas con distintos usuarios para obtener resultados de la usabilidad del sistema. El proceso que se realizó fue crear y editar prácticas, y realizar misiones. El proceso que tuvieron que realizar se puede encontrar en A.9 *Prueba de aprendizaje*. A continuación se muestra un gráfico con los tiempos que necesitó cada usuario para realizar el proceso, Figura 6.

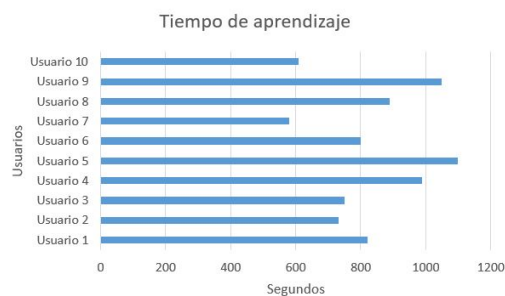


Figura 6: Prueba de usabilidad del sistema

Se observa que son unos tiempos bajos, ya que no se superan los 20 minutos al realizar el proceso usando la herramienta por primera vez.

En este apartado se han mostrado resultados obtenidos de pruebas que se han realizado al sistema, en el siguiente apartado se hablará sobre las conclusiones obtenidas de estos resultados y del trabajo realizado.

9 CONCLUSIONES

En este apartado se expondrán primeramente las conclusiones obtenidas de los resultados, luego se mostrarán las

aportaciones obtenidas del sistema, y después, se mostrarán posibles extensiones y mejoras del sistema.

A partir de los resultados expuestos en el apartado anterior, se puede concluir que el sistema podrá soportar la carga de conexiones incluso en el peor de los casos que todos los estudiantes se conecten a la vez. También, se observa que la herramienta es fácil y rápida de usar, y también fiable ya que todos los usuarios pudieron acabar el proceso sin equivocarse. Y por último se ha conseguido integrar elementos de gamificación para los diferentes tipos de "jugadores" que pueden llegar a ser los estudiantes.

Como conclusiones generales, ahora los estudiantes tienen una forma más práctica, motivadora y divertida de avanzar en las prácticas. Aparte de esto, también es de gran ayuda contar con el feed-back de los tests que los profesores usarán para puntuar esa práctica.

9.1 Propuesta de mejoras

En esta segunda parte de las conclusiones se introducen algunas de las muchas extensiones que no se han podido implementar por falta de tiempo en el proyecto y que a continuación se listarán:

- Añadir una capa de seguridad usando un filtro de código no permitido en las prácticas de los estudiantes para evitar posibles ataques al sistema.
- Creación de una aplicación móvil con la que los estudiantes puedan iniciar sesión y les permita gestionar su cuenta de usuario.
- Obtener información del análisis de los datos que se almacenan en la base de datos por cada alumno, para poder obtener estadísticas que puedan ser de ayuda al profesor.
- Usar más técnicas de gamification como por ejemplo el uso de una moneda virtual, permitiendo a los estudiantes gastar estas monedas en privilegios o elementos que hagan distinto a su usuario de los demás usuarios. Las monedas podrían ser conseguidas al completar misiones u otros retos como por ejemplo retos del "top enginyeria".
- Implementación de un chat con el que se puedan comunicar los estudiantes y sirva para poder comentar problemas, dudas de las prácticas y puedan ser solucionadas.

Este apartado de conclusiones provisionales se ha dividido en aportaciones obtenidas gracias al proyecto y en líneas futuras que podría seguir el proyecto.

AGRADECIMIENTOS

Me gustaría agradecer a todos los compañeros que me han ayudado a realizar tanto las pruebas de usabilidad cómo las pruebas de cada prototipo.

Y también un agradecimiento especial a mi tutor Sergi Robles, por haberme dado la oportunidad de realizar este proyecto y haber sabido guiarme durante todo el trabajo.

REFERENCIAS

- [1] K.B. Abhishek, *Approaches to detect SQL injection and XSS in web applications*, Proceedings of Electrical Engineering Seminar and Special Problems, Apr. 2012.
- [2] H. Eva, K. Paul, L. Steven, S. Norberts, *Responsive Web Design*, Google Scholar, 6 Dec. 2011, pp. 13-22.
- [3] C. Angel, *Metodologia y prototipo*, blog, 4 Dec. 2014: <https://es.slideshare.net/ajacj93/metodologia-y-prototipo.html> [Último acceso: 25 Feb 2018].
- [4] W. Luke, T. Laura, *PHP and MySQL Web Development*, ed. 2, Addison Wesley Pub Co Inc, 2016.
- [5] G. Kiryakova, N. Angelova, L. Yordanova, *Gamification in Education*, Proceedings of 9th International Balkan Education and Science Conference, 2014.
- [6] Z. Gave, C. Christopher, *Implementing Game Mechanics in Web and Mobile Apps*, Gamification by Design, O'Reilly Media, 2011.
- [7] J. Lee, J. Hammer, *Gamification in education: What, how, why bother?*, Academic Exchange Quarterly, 2011.
- [8] Moll, S. (2018). *Gamificación: 7 claves para entender qué es y cómo funciona*, blog, 26 Oct. 2015: <http://justificaturespuesta.com/gamificacion-7-claves-para-entender-que-es-y-como-funciona> [Último acceso: 20 Feb 2018].
- [9] T. Bui, *Analysis of Docker Security*, arXiv:1501.02967, 13 Jan 2015.
- [10] A. Lefl, J.T. Rayfield, *Web-application development using the Model/View/Controller design pattern*, Proceedings of 5th IEEE International Conference on Enterprise Distributed Object Computing, 2001.
- [11] S.C. Adrián, R. Sergi, P. Jordi, *A gamification experience to improve engineering students' performance through motivation*, Journal of Technology and Science Education, 7 Jun. 2017, pp. 150-161.
- [12] G.M. Cristian, *Estudio, diseño e implementación de una web de corrección automática de prácticas y trabajos la asignatura Compiladores*, 2015: https://ddd.uab.cat/pub/tfg/2015/tfg_27911/Articulo.pdf. [Último acceso: 1 Jun 2018].
- [13] Moodle, <https://moodle.org/?lang=es>. [Último acceso: 2 Jun 2018].
- [14] C.D. Jordi, *Desenvolupament de nous mòduls sobre la plataforma Moodle 2.0: Autotest*, 2013: <https://upcommons.upc.edu/handle/2099.1/15009> [Último acceso: 1 Jun 2018].
- [15] *Gamification: tipos de jugadores según el modelo Bartle*, blog, 2018. <https://www.iebschool.com/blog/tipos-jugadores-gamification-2-innovacion/> [Último acceso: 12 Jun 2018].

APÉNDICE

A.1 Diagrama de Gantt

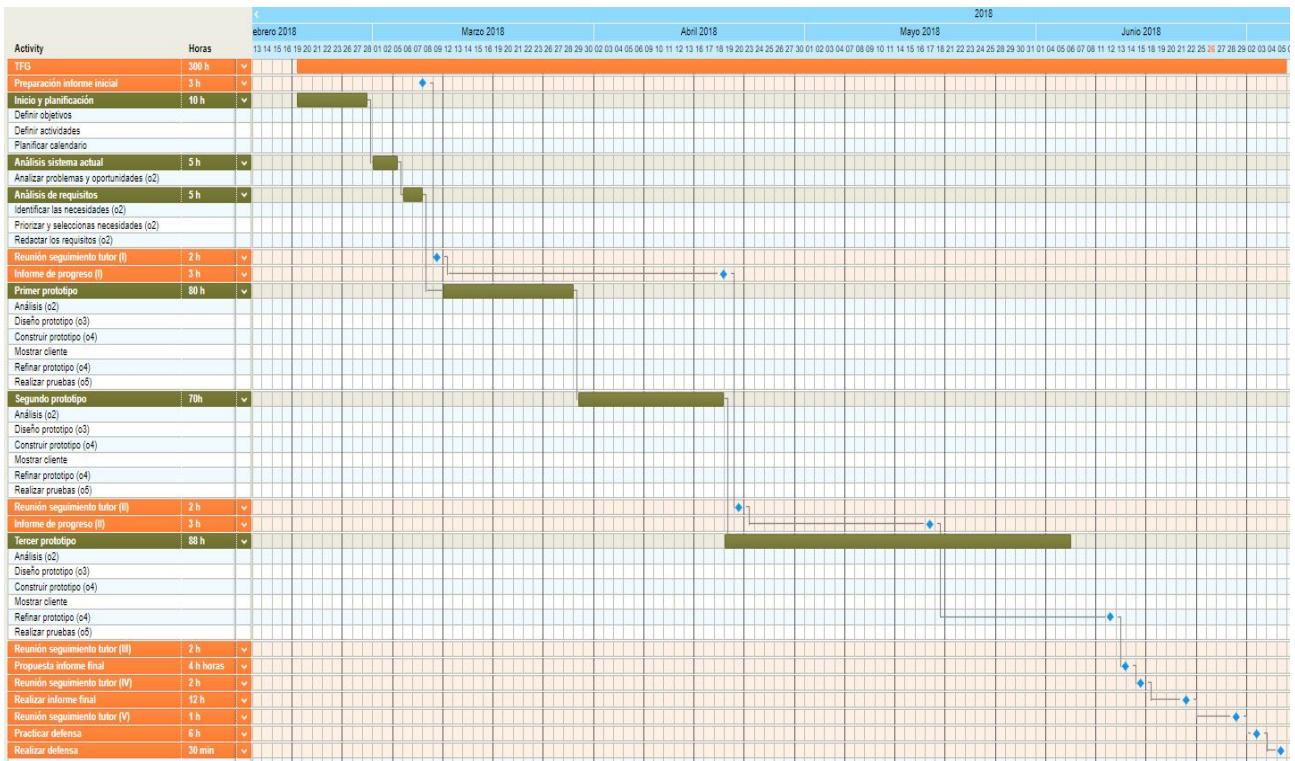


Figura 7: Planificación Diagrama de Gantt

A.2 Diagrama de casos de uso

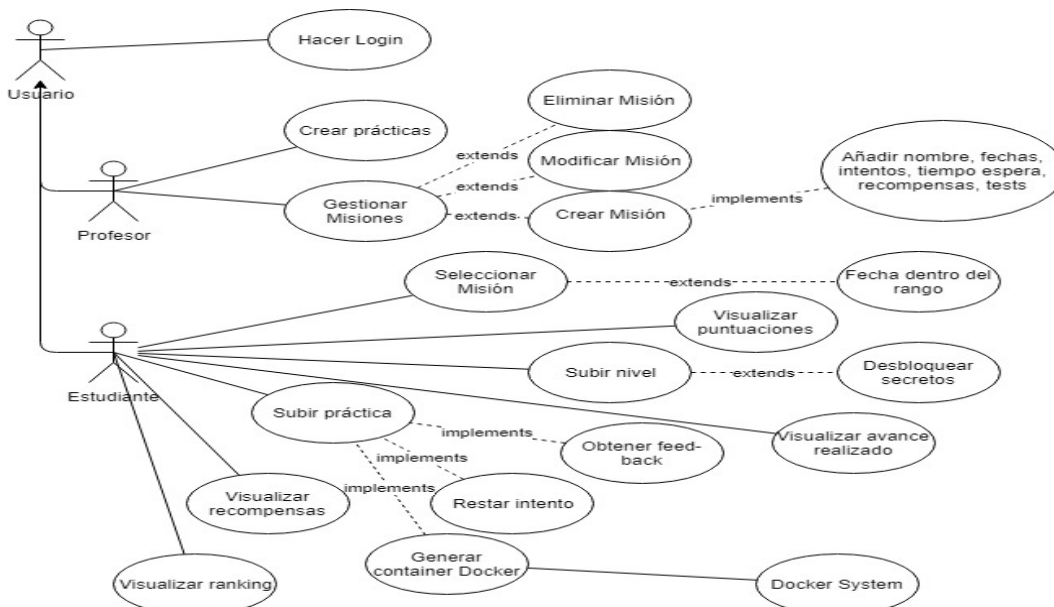


Figura 8: Diagrama casos de uso

A.3 Requerimientos no funcionales

- El sistema debe desarrollarse para que funcione en los navegadores Google Chrome, Firefox e Internet Explorer.
- El sistema debe poseer interfaces gráficas bien formadas.
- La aplicación web debe poseer un diseño “Responsive” a fin de garantizar la adecuada visualización en múltiples pantallas.
- La interfaz de usuario será implementada para navegadores web.
- La metodología del desarrollo del software será basado en un desarrollo evolutivo, específicamente en un modelo de prototipos.
- Las prácticas de los alumnos deben ejecutarse en un entorno aislado del sistema y de forma remota.
- El sistema debe cancelar la ejecución de una práctica si esta supera el minuto de durada.
- El tiempo de aprendizaje del sistema por un usuario deberá ser menor a 1 hora .
- El sistema debe ser capaz de operar adecuadamente con hasta 100 usuarios con sesiones concurrentes.

A.4 Generación container Docker

Código PHP de generación de un container Docker

```
<?php

session_start();
include '../config/pathToDocker.php';
include '../config/connectBBDD.php';

$path = getPath();
$conne = connect();

$practicaSeleccionada = $_SESSION['nomPractica'];
$grupo = $_SESSION['grupo'];
$dir = $grupo."-".$practicaSeleccionada;
$path += "/" . $dir;

//Get confHeadFile BBDD
$query = "SELECT ConfHeader
        FROM 'Test'
        WHERE NombrePractica =
        '$practicaSeleccionada'";
$resultado = $conne->query($query);
$fila = mysqli_fetch_array($resultado);
$fileConHead = $fila['ConfHeader'];

//Create docker container
shell_exec
('docker run -d --user 1000 -v "'. $path.'" :/resultsDir
--name "'. $grupo.'" ubuntu:toni
sh /resultsDir/" . $fileConHead . "'');

?>
```

A.5 Pruebas prototipo 1

Pruebas caja negra:

- Log in al sistema
- Creación, modificación y eliminación de pruebas
- Validación de la prueba de profesores
 - Valores de los campos
 - Formato de los archivos
- Validación de la práctica de los estudiantes
 - Tiempo entre cada intento
 - Formato de la práctica
- Resultados de los tests al finalizar el proceso del container

Pruebas exploratorias:

- Entradas para que aparecieran al menos una vez todos los mensajes de error.
- Entradas para forzar que el software establezca valores predeterminados.
- Entradas con cadenas de caracteres que pueden tener otro significado al ser ejecutado.
- Entradas que sobrecargan el buffer del campo o el de la base de datos.
- Repetir la misma entrada muchas veces.
- Entradas inválidas.
- Entradas que fuercen la salida a generar algún error.

Pruebas usabilidad:

- Log in en el sistema.
- Crear una prueba con un nombre, fechas y tests específicos.
- Modificar esa prueba.
- Eliminar una prueba.
- Subir una práctica como si fuera un estudiante.

A.6 Pruebas prototipo 2

Pruebas caja negra:

- Log in al sistema siendo un profesor
- Creación de las nuevas prácticas
- Creación de misiones
 - Con valores límite en la entrada de “tiempo de espera” e “intentos”
 - Diferentes tipos de misiones
 - Diferentes número de misiones
- Validación de las nuevas misiones

- Valores de los campos
- Formato de los archivos
- Entradas para que aparecieran al menos una vez todos los mensajes de error.
- Entradas para forzar que el software establezca valores predeterminados.
- Entradas con cadenas de caracteres que pueden tener otro significado al ser ejecutado.
- Entradas que sobrecargan el buffer del campo o el de la base de datos.
- Repetir la misma entrada muchas veces.
- Entradas inválidas.
- Entradas que fuercen la salida a generar algún error.

Pruebas usabilidad:

- Log in en el sistema como profesor.
- Crear una práctica con un nombre, descripción y recompensa.
 - Crear una lista de misiones con distintos atributos dentro de esta práctica.
 - Modificar algunas misiones.
 - Eliminar alguna misión.
- Log in en el sistema como alumno.
- Realizar las misiones que había creado anteriormente.
 - Seleccionando la practica creada.
 - Realizar el listado de misiones subiendo un archivo en concreto.

A.7 Engine Script

Ejemplo de Engine Script

```
#!/bin/bash
#Inicializar programas si es necesario
#Descomprimir practica si es necesario
#Crear fichero resultados
> /results/fileTmp.txt
#Leer archivo de tests trivials
while IFS='' read -r line || [[ -n "$line" ]]; do
#Ejecutar tests y escribir resultado en fichero
RES=$(sh "/results/$line")
echo "$RES#0#1" >> /results/fileTmp.txt
done < "/results/listTrivials"
#Leer archivo de tests basics
while IFS='' read -r line || [[ -n "$line" ]]; do
RES=$(sh "/results/$line")
echo "$RES#0#1" >> /results/fileTmp.txt
done < "/results/listBasics"
#Leer archivo de tests ocults
while IFS='' read -r line || [[ -n "$line" ]]; do
RES=$(sh "/results/$line")
echo "$RES" >> /results/fileTmp.txt
done < "/results/list0cults"
#Renombrar fichero
mv /results/fileTmp.txt /results/results.txt
```

A.8 Pruebas prototipo 3

Casuísticas y pruebas realizadas con los tests:

- Estudiante supera la misión con la máxima puntuación.
 - Misión con tests triviales, básicos y ocultos no necesarios.
 - Misión con tests triviales, básicos y ocultos necesarios.
 - Misiones sólo con tests triviales, básicos o ocultos.
- Estudiante no supera la misión.
 - No superada porque no ha superado 1 test básico.
 - No superada porque no ha superado 1 test trivial.
 - No superada porque no ha superado 1 test oculto necesario.
 - No superada porque no ha superado mas de un test básico, trivial y/u oculto.
- El tiempo de la validación supera el límite establecido.

A.9 Prueba de aprendizaje

Paso 1:

Entrar en el portal login con las credenciales:

- grupo lab: tai-admin
- password: admin

Paso 2:

Crear una nueva práctica con los valores:

- Nombre: PtaiUser
- Descripción: Texto max. 200 palabras
- Premio: espada.jpg (adjunto en el mail)

Paso 3:

Crear 1 mision dentro de esa práctica

- Nombre: Mision1
- Fecha obertura: 10/3/2018
- Fecha cierre: 1/7/2018
- Intentos: 10
- Tiempo de espera: 5
- Obligatoria: si
- Premio: 10k
- Pujar config header: subimos el archivo adjunto test.sh,
- Pujar test basic: subimos los archivos test1.sh, test2.sh, test3.sh
- Los demas los dejamos vacios
- Eliminamos el archivo test2.sh
- Movemos el archivo test3.sh delante de test1.sh

- Creamos la misión

Paso 4:

Editar la misión que acabais de crear cambiando el nombre a cómo os llamais vosotros.

Paso 5:

Visualizar la misión.

Paso 6:

Salir de la pantalla, clicando el boton de log out. Entrar en el portal login con las credenciales:

- grupo lab: x-d-1
- password: 111111

Paso 7:

Entrar en la segunda práctica y realizar la primera misión

- Subir el archivo practicaPing.sh
- Esperar y visualizar la puntuación
- Repetir el proceso con las otras misiones de la práctica

A.10 Diagrama entidad-relación protitio 2

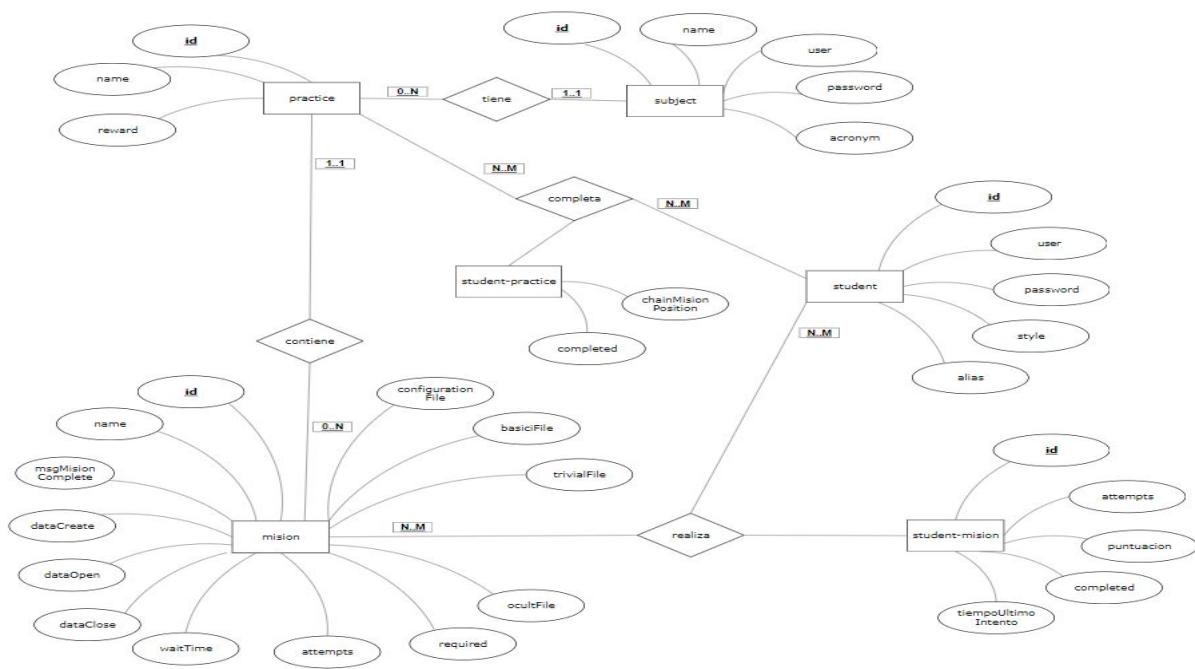


Figura 9: Diagrama entidad-relación