



**Escuela Superior
de Ingeniería y Tecnología**
Universidad de La Laguna

Trabajo de Fin de Grado

Grado en Ingeniería Informática

Simuladores didácticos de Arquitectura de Computadores

Didactic computer architecture simulator

Leonardo Alfonso Cruz Rodríguez

La Laguna, 14 de julio de 2023

D. **Iván Castilla García**, con N.I.F. 78.565.451-G profesor Contratado Doctor adscrito al Departamento de Ingeniería Informática y de Sistemas de la Universidad de La Laguna, como tutor

CERTIFICA

Que la presente memoria titulada:

“Simuladores didácticos de Arquitectura de Computadores”

ha sido realizada bajo su dirección por D. **Leonardo Alfonso Cruz Rodríguez**,
con N.I.F. 43.486.599-Q.

Y para que así conste, en cumplimiento de la legislación vigente y a los efectos oportunos firman la presente en La Laguna en la fecha que figura en la firma electrónica del documento.

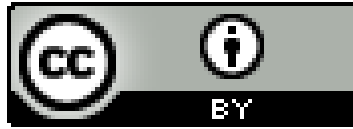
Agradecimientos

En primer lugar, me gustaría agradecer a mi profesor y tutor de TFG Iván Castilla, cuando recibí clases tuyas en mi primer año de carrera me llevé una muy buena impresión. Ahora después de varios años, fue una de las razones para escoger este trabajo y pude comprobar que sigue siendo un docente excelente.

A mi familia, por depositar su apoyo y confianza en mí desde siempre. Aunque siempre acababan preguntando de qué trataba este trabajo porque no se acordaban ni del nombre. No puedo describir lo fundamental que fueron para mi desarrollo en el ámbito académico, y siempre estaré agradecido por ello, tanto a los familiares que siguen a mi lado como los que ya no están.

Por último, agradecer a los compañeros de carrera que acabaron siendo amigos, entre nosotros también nos conseguimos ayudar para sacar la carrera adelante y con este trabajo no faltó el apoyo y los ánimos para culminar esta etapa.

Licencia



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

Resumen

Hasta el momento, los simuladores de computadores han ayudado mucho a la comprensión de su funcionamiento. No solo para las personas que los programan, sino también para aquellos que los utilizan con la intención de obtener determinados resultados.

Este proyecto tiene como objetivo llevar esta ventaja un paso más allá, partiendo de una aplicación web con distintos simuladores. Dotándola de funcionalidades que favorecen la ludificación, la participación, la competitividad y sobre todo el aprendizaje acerca del funcionamiento de estas máquinas.

Como resultado final, hemos logrado integrar a una aplicación web previamente desarrollada, la cual incluye simuladores de máquinas superescalares y VLIW, un servicio de gestión de usuarios que permitirá la participación en las actividades presentadas en la plataforma.

Palabras clave: Superescalar, Simulador, Aplicación Web, React, Redux, Firebase, Bootstrap

Abstract

At this point, computer simulators have helped in understanding their operation. Not only for the people who program them, but also for those who use them intending to obtain certain results.

This project aims to take this advantage a step further, starting from a web application with different simulators. Equipping it with functionalities that promotes gamification, participation, competitiveness, and mostly learning about the operation of these machines.

As a result, we have managed to integrate into a previously developed web application, which includes superscalar and VLIW machine simulators, a user management service that will allow participation in the activities presented on the platform.

Keywords: Superscalar, Simulator, Web Application, React, Redux, Firebase, Bootstrap

Índice general

Capítulo 1	Introducción	1
1.1	Introducción	1
1.2	Ludificación	1
1.3	Aplicaciones Web	2
Capítulo 2	Antecedentes	3
2.1	SIMDE	3
2.2	SIMDE Web	3
2.3	Estado actual	4
Capítulo 3	Objetivos y tareas	5
3.1	Objetivos	5
3.2	Tareas por hitos	5
3.2.1	Primer hito	5
3.2.2	Segundo hito	6
3.2.3	Tercer hito	6
3.2.4	Cuarto hito	6
Capítulo 4	Tecnologías, lenguajes y servicios	7
4.1	Lenguajes de programación	7
4.1.1	TypeScript	7
4.1.2	JavaScript XML	7
4.2	Tecnologías para el desarrollo	8
4.2.1	React	8
4.2.2	Redux	8
4.2.3	Bootstrap con CSS	9
4.3	Despliegue, integración y mantenimiento	9
4.3.1	GitHub	9
4.3.2	NPM	10
4.3.3	Webpack	10
4.3.4	Docker	11
4.4	Gestión de usuarios y Base de Datos	11
4.4.1	Firebase Authentication	11
4.4.2	Firebase Realtime Database	11
Capítulo 5	Arquitectura de la aplicación	13

5.1	Gestión de usuarios	14
5.2	Base de datos.....	14
5.3	Almacenamiento local	15
Capítulo 6	Funcionalidades	17
6.1	Cuenta de usuario.....	17
6.1.1	Registro e Inicio de sesión	17
6.1.2	Perfil de usuario.....	18
6.2	Participación en Actividades	19
6.2.1	Actividades públicas y privadas.....	19
6.2.2	Mis actividades	21
6.2.3	Fase de pruebas.....	22
Capítulo 7	Conclusiones y líneas futuras	23
7.1	Conclusiones.....	23
7.2	Líneas futuras	23
Capítulo 8	Summary and Conclusions.....	24
8.1	Summary.....	24
8.2	Future work lines	24
Capítulo 9	Presupuesto	25

Índice de figuras

Figura 2.1: SIMDE Original.....	3
Figura 2.2: SIMDE Web.....	4
Figura 4.1: Funcionamiento de Redux	8
Figura 4.2: Flujo de trabajo GitHub	9
Figura 4.3: Firebase Realtime Database	12
Figura 5.1: Arquitectura de la aplicación Web.....	13
Figura 5.2: Modelo de datos	14
Figura 6.1: Esquema de páginas web	17
Figura 6.2: Página de Registro/Inicio de sesión	18
Figura 6.3: Página de perfil de usuario	19
Figura 6.4: Página de actividades públicas	19
Figura 6.5: Registro a actividades privadas.....	20
Figura 6.6: Selección de grupo	21
Figura 6.7: Página Mis Actividades	22

Índice de tablas

Tabla 3.2.1: Tareas del primer hito.....	5
Tabla 3.2.2: Tareas del segundo hito.....	6
Tabla 3.2.3: Tareas del tercer hito.....	6
Tabla 3.2.4: Tareas del cuarto hito.....	6
Tabla 9: Presupuesto general.....	25

Capítulo 1

Introducción

1.1 Introducción

Durante la última década, el ámbito académico ha sido testigo de una serie de cambios tecnológicos significativos, reformando progresivamente las metodologías de aprendizaje tradicionales. Cada vez más, los estudiantes optan por programas digitales para tomar apuntes, reemplazando el antiguo paradigma de papel y bolígrafo. Simultáneamente, las redes sociales y los foros online se están convirtiendo en herramientas esenciales para facilitar el acceso a la información pertinente.

Los docentes no se quedan atrás en la adopción de estas innovaciones. La implementación de plataformas virtuales para asignar tareas, interactuar con los estudiantes, evaluar trabajos, corregir exámenes y distribuir contenidos se ha convertido en una práctica común.

A medida que estos cambios tecnológicos se integran en el ámbito académico, se ha despertado un interés creciente en una nueva tendencia pedagógica conocida como ludificación. Esta práctica, que implica la integración de elementos y técnicas de juego en contextos no lúdicos, ha demostrado tener un impacto positivo en el aprendizaje. La ludificación tiene la capacidad de aumentar la motivación y la participación de los estudiantes, fomentar el aprendizaje experiencial y mejorar la retención de información.

El objetivo de este proyecto consiste en unificar este método pedagógico, en combinación con las ventajas que ofrecen las tecnologías emergentes, para proporcionar una manera para mejorar y enriquecer el proceso educativo en la era digital.

1.2 Ludificación

La ludificación es un término cada vez más familiar en el ámbito académico. Sin embargo, su efectividad y aplicabilidad en la educación no se basan en simples suposiciones, sino que están respaldadas por un creciente cuerpo de investigación.

Kapp define la ludificación como "el uso de elementos de diseño de juego en contextos no lúdicos" [1]. En el contexto educativo, esto puede traducirse en el uso de puntos, niveles, logros, feedback inmediato y competencia amigable en el proceso de aprendizaje.

Según Hamari, la ludificación puede tener un efecto positivo en la motivación y el rendimiento de los estudiantes [2]. A través de la utilización de recompensas y reconocimientos, los estudiantes se sienten más comprometidos y motivados para aprender, lo que puede conducir a una mayor retención y comprensión del material.

Además, la ludificación puede contribuir a una experiencia de aprendizaje más personalizada. Según una investigación de Hanus & Fox, la personalización y la autonomía en un entorno de aprendizaje lúdico pueden aumentar el compromiso y la motivación de los estudiantes [3].

Sin embargo, no todas las implementaciones de la ludificación son igualmente efectivas. Dicheva argumentan que el diseño y la implementación de la ludificación son cruciales para su éxito [4]. Las estrategias de ludificación deben estar bien alineadas con los objetivos de aprendizaje, y los elementos de juego deben ser relevantes y atractivos para los estudiantes.

1.3 Aplicaciones Web

La tecnología y el internet han revolucionado diversos aspectos de nuestra vida, incluyendo la educación. A medida que avanzamos en la era digital, las aplicaciones web desempeñan un papel cada vez más integral en el ámbito académico. Estas herramientas digitales, especialmente cuando incorporan la ludificación, pueden enriquecer significativamente el proceso de aprendizaje.

Las aplicaciones web son programas accesibles a través de un navegador web, lo que significa que no requieren instalación y pueden ser accesibles desde cualquier dispositivo con conexión a internet. En el ámbito académico, estas aplicaciones abarcan desde sistemas de gestión de aprendizaje, plataformas de colaboración en línea, hasta aplicaciones de evaluación y retroalimentación.

Una aplicación web que utiliza la ludificación puede presentar contenido educativo de una manera más interactiva y atractiva. Puede incluir desafíos, competencias, recompensas, tableros de clasificación y otros elementos de juego para incentivar a los estudiantes a participar más activamente en su propio aprendizaje. Estas aplicaciones pueden hacer que el aprendizaje sea más atractivo y divertido, fomentando así una mayor participación y compromiso de los estudiantes.

Además, estas aplicaciones pueden proporcionar retroalimentación instantánea, permitiendo a los estudiantes comprender inmediatamente qué conceptos necesitan mejorar. También pueden personalizar el contenido según el ritmo y el nivel de habilidad del estudiante, lo que puede ayudar a los estudiantes a aprender de manera más eficaz.

Capítulo 2

Antecedentes

2.1 SIMDE

SIMDE es un simulador pedagógico creado en 2004 por Iván Castilla Rodríguez, un estudiante en ese momento, ahora profesor y tutor de este proyecto, para la enseñanza de la asignatura arquitectura de computadores [5]. Esta herramienta tiene la capacidad de emular arquitecturas que buscan maximizar el paralelismo a nivel de instrucción (ILP) utilizando tanto técnicas de programación dinámica como estática, en la Figura 2.1 se puede observar la interfaz del programa SIMDE original.

El enfoque de SIMDE es estrictamente educativo, evitando la complicación de las tecnologías avanzadas y enfocándose en los principios fundamentales para entender el ILP. La herramienta opera con un conjunto de instrucciones que se asemeja al MIPS, aunque mucho más simplificado.

Originalmente se trataba de un software programado en el lenguaje C++, lo que lo limitaba a una aplicación de escritorio. Actualmente las aplicaciones de escritorio están transicionando a la obsolescencia, debido a problemas de mantenimiento, integración o escalabilidad. Es por ello por lo que el proyecto se trasladó a una aplicación web y fue renombrado a SIMDE Web.

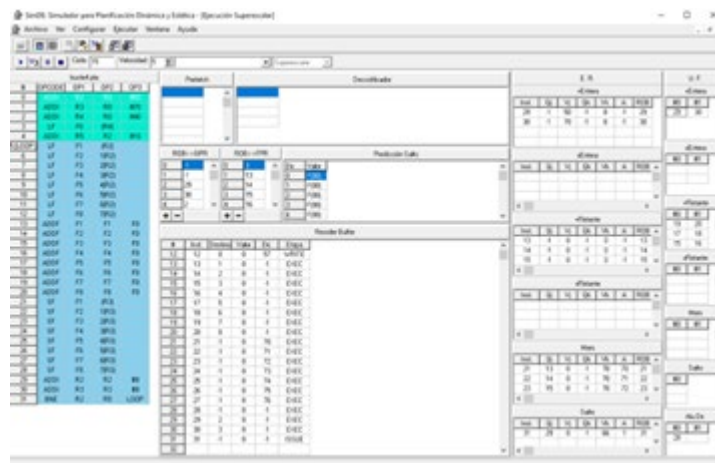


Figura 2.1: SIMDE Original

2.2 SIMDE Web

Durante varios años, el proyecto ha pasado por distintos exalumnos de la Universidad de La Laguna, en 2017 el alumno Adrián Abreu González contribuyó al gran proyecto que resultó ser SIMDE. Desarrollando como Trabajo de Fin de Grado una versión web de la aplicación haciendo uso de tecnologías de actualidad como TypeScript y React (Figura 2.2) [6].

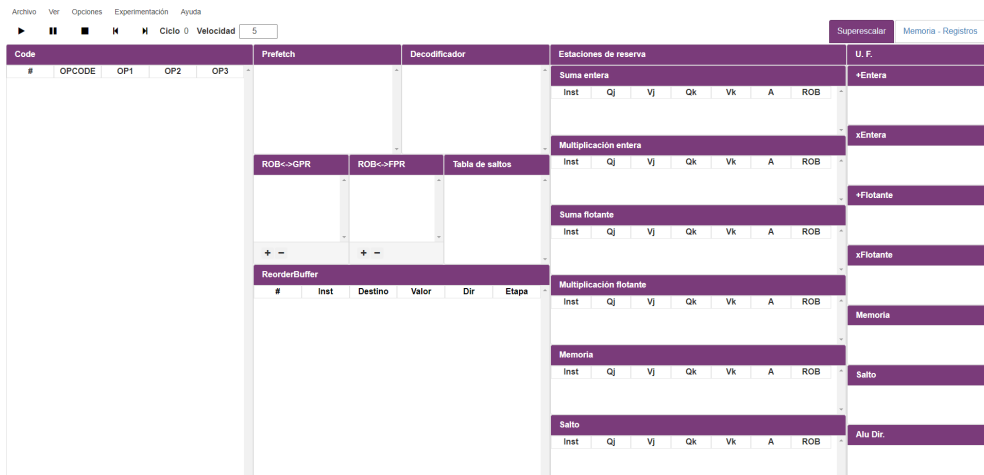


Figura 2.2: SIMDE Web

Al año siguiente, en 2018, Antonio Jesús López Garnier retomó el trabajo y propuso el primer diseño de ludificación para la plataforma en su proyecto titulado "Plataforma de ludificación de un simulador didáctico de arquitectura de computadores" [7].

En ese mismo año, Melissa Díaz Arteaga logró implementar un nuevo simulador en la plataforma. Este simulador estaba basado en la arquitectura de la máquina Very Long Instruction Word (VLIW), dando lugar a su proyecto "Simulador didáctico de una Arquitectura de planificación estática" [8].

La contribución más reciente en este proyecto se realizó en 2022 por Óscar Carrasco Benítez, en su proyecto fue replanteado "Simulador didáctico de arquitectura de computadores: aplicación de metodologías de integración y mantenimiento" el flujo de trabajo en el proyecto, añadiendo integración continua y mejorando el mantenimiento del proyecto gestionando las dependencias de este [9].

2.3 Estado actual

SIMDE Web es utilizada en las clases prácticas de la asignatura Arquitectura de Computadores. Actualmente se encuentra desplegado en GitHub Pages [10], en un repositorio de código abierto y cuenta con un simulador de una máquina superescalar y otro de una máquina VLIW.

A raíz de este repositorio y del trabajo realizado por Óscar, el método de desarrollo actual consiste en realizar un *fork* del repositorio, para posteriormente solicitar la validación de un administrador y unificarlo con la rama principal del proyecto.

Capítulo 3

Objetivos y tareas

3.1 Objetivos

La planificación de este proyecto se organizó de una manera jerárquica con prioridades, teniendo tareas de baja, media o alta prioridad, además de las opcionales. Fueron planteados cinco objetivos principales, los cuales tienen una serie de tareas para alcanzar dichos hitos.

- Diseño del modelo de datos
- Configuración e implementación del servicio de gestión de usuarios
- Configuración e implementación del servicio de base de datos
- Diseño, desarrollo e implementación de interfaces en la aplicación
- Implementación del simulador secuencial de programas por instrucciones

Adicionalmente, se plantearon tareas opcionales como por ejemplo incluir traducciones en las interfaces o crear una interfaz de gestión para el profesorado.

3.2 Tareas por hitos

Para la organización y el seguimiento de este proyecto, he utilizado la plataforma Notion [11]. La cual permite organizar tareas por hitos, estado de desarrollo y prioridades, además de la posibilidad de añadir subtareas dentro de las propias tareas con los mismos atributos.

3.2.1 Primer hito

Tareas	Prioridad
Interfaz de <i>LogIn</i> y <i>SignUp</i>	Alta
Configuración de Firebase Authentication	Alta
Implementación del servicio <i>LogIn</i> y <i>SignUp</i> en la aplicación	Media

Tabla 1: Tareas del primer hito

3.2.2 Segundo hito

Tareas	Prioridad
Diseño del modelo de datos	Alta
Configuración de Firebase Realtime Database	Alta
Pruebas de lectura y escritura desde la aplicación	Media

Tabla 2: Tareas del segundo hito

3.2.3 Tercer hito

Tareas	Prioridad
Diseño de interfaz de actividades	Media
Listar y visualizar actividades con sus propiedades	Alta
Registro a actividades nuevas	Media

Tabla 3: Tareas del tercer hito

3.2.4 Cuarto hito

Tareas	Prioridad
Implementar un simulador secuencial por instrucciones	Baja
Fase de pruebas dentro de las actividades	Media
Interfaz de gestión para profesores	Opcional
Incluir traducciones en inglés	Opcional

Tabla 4: Tareas del cuarto hito

Capítulo 4

Tecnologías, lenguajes y servicios

En este capítulo se presentarán las distintas tecnologías, herramientas y servicios que han sido utilizados en los distintos aspectos de este proyecto. Algunos de estos fueron escogidos en versiones anteriores del proyecto, por lo tanto, no resulta conveniente buscar alternativas ni realizar cambios en las decisiones tomadas. En cambio, hay otras que no se han utilizado previamente, o por lo menos no se llegaron a integrar en el proyecto finalmente.

4.1 Lenguajes de programación

4.1.1 TypeScript

TypeScript es un lenguaje de programación de código abierto desarrollado por Microsoft. Es un superconjunto tipado de JavaScript, lo que significa que cualquier código JavaScript es también código TypeScript.

En anteriores versiones de este proyecto se ha decidido utilizar TypeScript en combinación con React, un popular marco de trabajo para el desarrollo de interfaces de usuario en JavaScript. React y TypeScript se complementan bien, ya que React se centra en la construcción de componentes de interfaz de usuario y TypeScript aporta un sistema de tipos que ayuda a garantizar que estos componentes se utilicen de la manera correcta.

Al utilizar TypeScript con React, podemos definir propiedades (props) y estados con tipos estáticos, lo que nos permite detectar errores en tiempo de compilación si, por ejemplo, intentamos pasar un tipo de datos incorrecto a un componente.

Además, TypeScript viene con un sistema de módulos integrado que facilita la organización del código en módulos más pequeños. Esto ha sido útil en este proyecto, debido a que se ha tenido que manejar una cantidad significativa de lógica de la aplicación y datos de estado.

4.1.2 JavaScript XML

JSX, o JavaScript XML, es una extensión de la sintaxis de JavaScript que permite escribir estructuras similares al HTML directamente en el código JavaScript. JSX ha sido utilizado en el desarrollo de anteriores versiones del proyecto con React, se trata de una biblioteca JavaScript para la construcción de interfaces de usuario. En términos simples, JSX permite la creación de elementos y componentes de React utilizando una sintaxis que se asemeja a la del HTML, lo que facilita la lectura y la escritura del código.

JSX permite a los desarrolladores construir componentes dinámicos de React. Los componentes de React definidos con JSX se traducen en llamadas, lo que permite a React construir una representación de la interfaz de usuario en memoria (conocida como el "DOM virtual") que luego puede ser renderizada en el navegador.

4.2 Tecnologías para el desarrollo

En este apartado mencionaremos las librerías orientadas a metodologías de desarrollo y los lenguajes de programación que se utilizaron para el desarrollo de la aplicación web.

4.2.1 React

React es una biblioteca de JavaScript desarrollada con el objetivo de construir interfaces de usuario. React se destaca por su eficiencia y flexibilidad, permitiendo a los desarrolladores crear aplicaciones web de una sola página (SPA) que pueden actualizar y renderizar componentes eficientemente en respuesta a cambios de datos, sin tener que recargar toda la página.

Aunque React está orientado al desarrollo de aplicaciones de una página, instalando librerías adicionales es posible crear más de una página y además conectarlas entre sí mediante rutas. En este proyecto y en anteriores versiones se ha optado por esta opción, debido al aumento de funcionalidades que ha tenido en sus últimas versiones.

4.2.2 Redux

Redux es una biblioteca de JavaScript de código abierto que se utiliza para gestionar el estado de una aplicación. En aplicaciones complejas de React, Redux proporciona un "almacén" global que permite a los diferentes componentes de una aplicación compartir y manejar el estado de una manera predecible.

Al momento de desarrollar aplicaciones web nos ofrece una gran ventaja, la conservación de datos mediante el manejo de estados. Esta ventaja nos permite conocer el estado actual y transicionar a otro estado sin perder datos independientemente de si la página es cerrada o actualizada por el usuario.

El funcionamiento del almacenaje de estados y toda la información se puede ver representado en la Figura 4.1. Para almacenar datos nuevos se necesita disparar una acción, dicha acción será captada por un "reducer". Los "reducers" se encargan de capturar las acciones disparadas y cambiar el estado actual en el "almacén" o mejor conocido como "Store". Por último, para leer los datos contenidos en el "Store" necesitamos utilizar "selectores". Los selectores son funciones que reciben el estado de la aplicación como argumento y devuelven la parte del estado que necesitamos. Esto nos proporciona un método fácil y eficiente para acceder a los datos en nuestro "store".

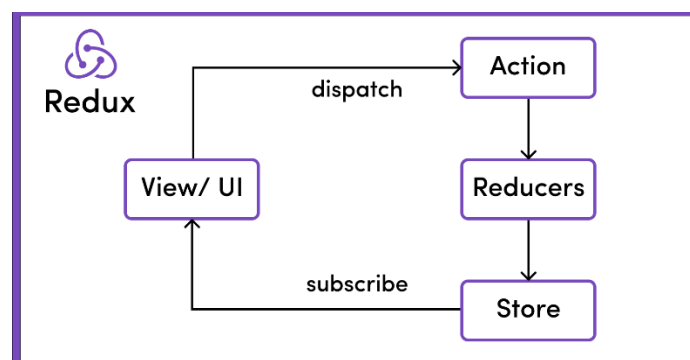


Figura 4.1: Funcionamiento de Redux

Fuente: What is Redux? Store, Actions, and Reducers Explained for Beginners [12]

4.2.3 Bootstrap con CSS

Bootstrap es una biblioteca de diseño de código abierto que facilita la creación de interfaces de usuario responsivas y atractivas. [13] Proporciona una variedad de componentes predefinidos de HTML, CSS y JavaScript, que los desarrolladores pueden utilizar para crear rápidamente sitios web y aplicaciones web. Los componentes de *Bootstrap* incluyen botones, formularios, tarjetas, carruseles, menús de navegación, y muchos otros.

En combinación con CSS, *Bootstrap* permite a los desarrolladores escribir menos código y lograr más. En el desarrollo de esta aplicación hemos podido utilizar las reglas CSS predefinidas de *Bootstrap* y personalizarlas según sea necesario, lo que ha permitido un ahorro significativo de tiempo y además asegura la consistencia en el diseño.

Adicionalmente, se ha añadido la librería de estilo *Sweet Alerts* [14], una alternativa a las alertas por defecto con mejores efectos visuales que además permite personalizar iconos, títulos, descripciones, botones y otros aspectos visuales de las alertas.

4.3 Despliegue, integración y mantenimiento

En este apartado mencionaremos las herramientas y servicios que se utilizaron para el despliegue, integración y mantenimiento del proyecto.

4.3.1 GitHub

GitHub es una plataforma de alojamiento de código que permite a los desarrolladores colaborar en proyectos y mantener un registro de las modificaciones realizadas en su código fuente a lo largo del tiempo. Su funcionalidad se basa en el sistema de control de versiones Git, permitiendo a los desarrolladores trabajar en proyectos sin pisar los cambios de los demás, manteniendo una historia completa de todas las modificaciones.

El código de este trabajo se encuentra en un repositorio “fork” [15] proveniente del repositorio con la última versión de este proyecto [16]. Realizar este tipo de colaboraciones es posible gracias a que GitHub ofrece una serie de herramientas y servicios integrados que facilitan el desarrollo colaborativo, como la gestión de problemas para el seguimiento de bugs y solicitudes de características, revisiones de código para garantizar la calidad del código, y GitHub Actions para la automatización de tareas como pruebas y despliegue. En este proyecto, se utilizaron todas estas características para asegurar un proceso de desarrollo fluido y eficiente.

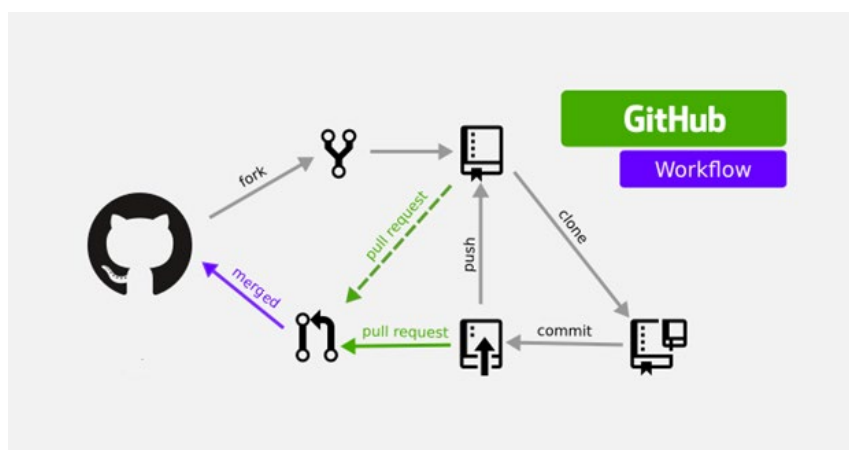


Figura 4.2: Flujo de trabajo GitHub

Fuente: *How to Sync Forked Repositories Using Git or Github* [17]

Por último, el repositorio de GitHub sirvió como base para el despliegue del proyecto a través

de GitHub Pages, permitiendo una transición fluida desde el desarrollo hasta la puesta en producción. Gracias a la integración directa entre GitHub y GitHub Pages, cada vez que se realizaba un cambio en la rama principal del repositorio, el sitio se actualizaba automáticamente para reflejar estos cambios.

4.3.2 NPM

Dentro de este proyecto, NPM ha sido utilizado no solo para manejar las dependencias, sino también para automatizar tareas de desarrollo mediante scripts NPM. Estos scripts son un conjunto de comandos que se pueden ejecutar desde la línea de comandos y son útiles para automatizar tareas repetitivas.

A continuación, se presenta un desglose de algunos de los scripts utilizados en este proyecto:

"npm run dev" - Este script lanza el servidor de desarrollo de Webpack en el puerto 9060, facilitando el desarrollo en un entorno que refleja de cerca el entorno de producción.

"npm run build" - Este script elimina la carpeta build, luego utiliza el compilador de TypeScript (tsc) para compilar el proyecto de acuerdo a la configuración especificada en tsconfig.json.

"npm run dist" - Este script crea una versión de producción de la aplicación usando Webpack.

"npm run lint" - Este script pasa el linter (tslint) por todos los archivos TypeScript en el directorio src, asegurando que el código cumple con las guías de estilo y formato predefinidas.

"npm run test" o "npm run unit" - Este script ejecuta pruebas unitarias en el código utilizando la herramienta de pruebas Ava.

"npm run docs" - Este script genera documentación para el código a partir de los comentarios del código utilizando TypeDoc.

4.3.3 Webpack

Webpack [18] es un empaquetador de módulos de JavaScript de código abierto. Se utiliza para compilar módulos modernos de JavaScript en paquetes estáticos que los navegadores pueden interpretar. Webpack permite a los desarrolladores manejar dependencias complejas y puede ser configurado para manejar una variedad de archivos de entrada y salida, incluyendo CSS, imágenes, y más.

En este proyecto, Webpack ha sido utilizado como una herramienta esencial para la compilación y empaquetado de módulos JavaScript, para ello se han configurado tres archivos principales de Webpack, que gestionan la compilación y empaquetado en diferentes escenarios:

"webpack.common.js": Este es el archivo de configuración de Webpack común a todos los entornos, el cual establece configuraciones básicas.

"webpack.dev.js": Este archivo extiende webpack.common.js y contiene configuraciones específicas para el entorno de desarrollo.

"webpack.prod.js": Este archivo también extiende webpack.common.js pero está destinado a la configuración de producción.

4.3.4 Docker

Docker [19] es una plataforma de código abierto que facilita a los desarrolladores y administradores de sistemas la creación, implementación y ejecución de aplicaciones en contenedores. Un contenedor es una unidad de software estandarizada que empaqueta el código de la aplicación y todas sus dependencias, lo que permite que la aplicación se ejecute rápidamente y de manera confiable en cualquier entorno de computación.

En nuestro proyecto, Docker juega un papel importante para mantener un ambiente de desarrollo y producción. El archivo Dockerfile presente en el proyecto sirve para definir el contenedor en el que se alojará nuestra aplicación. Se utiliza una versión ligera de Node.js para establecer el entorno de desarrollo y también para construir la aplicación para la fase de producción.

Este flujo de trabajo con Docker nos permite tener un entorno controlado, tanto para el desarrollo como para la producción, asegurándonos de que la aplicación se comportará de la misma manera en todos los entornos en los que se despliegue. Además, facilita la tarea de despliegue y escalado de la aplicación, ya que el contenedor puede ser fácilmente replicado en múltiples instancias.

4.4 Gestión de usuarios y Base de Datos

Hasta este punto se han mencionado tecnologías que ha sido seleccionadas previamente a este proyecto. Estas funcionalidades son nuevas, por lo tanto, se indicarán los servicios seleccionados y sus alternativas junto con la justificación de dicha elección.

Para este proyecto se ha optado por utilizar el conjunto de servicios en la nube Firebase [20]. Firebase proporciona un conjunto completo de servicios de back-end que permiten a los desarrolladores centrarse más en la creación de una gran experiencia de usuario sin preocuparse por la configuración y la gestión del servidor. A diferencia de las alternativas como AWS (Amazon Web Services) o Microsoft Azure, que requieren más configuración y gestión, Firebase proporciona una experiencia más integrada y centrada en el desarrollador.

4.4.1 Firebase Authentication

Firebase Authentication [21] fue una opción bastante bien considerada debido a la simplicidad y variedad de opciones de autenticación. Con Firebase Authentication, la gestión de cuentas de usuario y los flujos de autenticación se manejan de manera segura sin la necesidad de una gestión de back-end personalizada. Este servicio proporciona una ventaja sobre alternativas como Auth0 o Passport.js, especialmente en términos de facilidad de integración y costos para aplicaciones con un gran número de usuarios.

4.4.2 Firebase Realtime Database

Firebase Realtime Database es una solución de base de datos en tiempo real que ofrece sincronización de datos en vivo (Figura 4.3). Comparada con alternativas como MongoDB o PostgreSQL, Firebase Realtime Database ofrece la ventaja de un modelo de datos en tiempo real, facilitando la creación de experiencias de usuario interactiva. Además, su capacidad para trabajar offline proporciona una gran ventaja para las aplicaciones que necesitan funcionar en condiciones de red inestables.

Dentro de la propia plataforma de Firebase existen dos alternativas de servicios de base de datos. La alternativa a Realtime Database es Cloud Firestore [22].

Se trata de una base de datos no relacional, lo que significa que no sigue un esquema SQL o de tablas. Una base de datos no relacional organiza los datos de forma jerárquica mediante rutas, similar al funcionamiento de un sistema de carpetas. Este tratamiento de datos es muy utilizado en el ámbito web debido a su escalabilidad, rendimiento, flexibilidad y tolerancia a fallos.

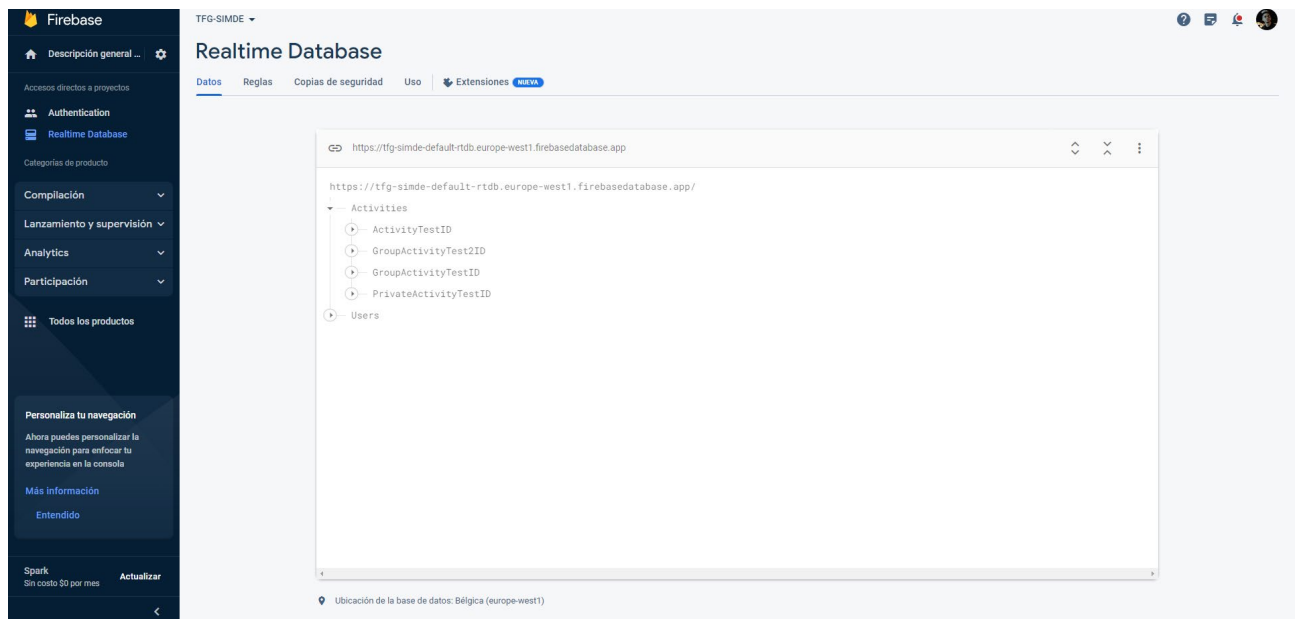


Figura 4.3: Firebase Realtime Database

Capítulo 5

Arquitectura de la aplicación

Para este proyecto se ha optado por utilizar el conjunto de servicios en la nube Firebase. Firebase proporciona un conjunto completo de servicios de back-end que permiten a los desarrolladores centrarse más en la creación de una gran experiencia de usuario sin preocuparse por la configuración y la gestión del servidor.

A diferencia de las alternativas como AWS (Amazon Web Services) o Microsoft Azure, que requieren más configuración y gestión, Firebase proporciona una experiencia más integrada y centrada en el desarrollador.

En la Figura 5.1 se muestran todas las tecnologías, lenguajes y servicios utilizados en este trabajo. Además, se indican como fluyen los datos entre los componentes principales de la aplicación y con los servicios externos a la misma.

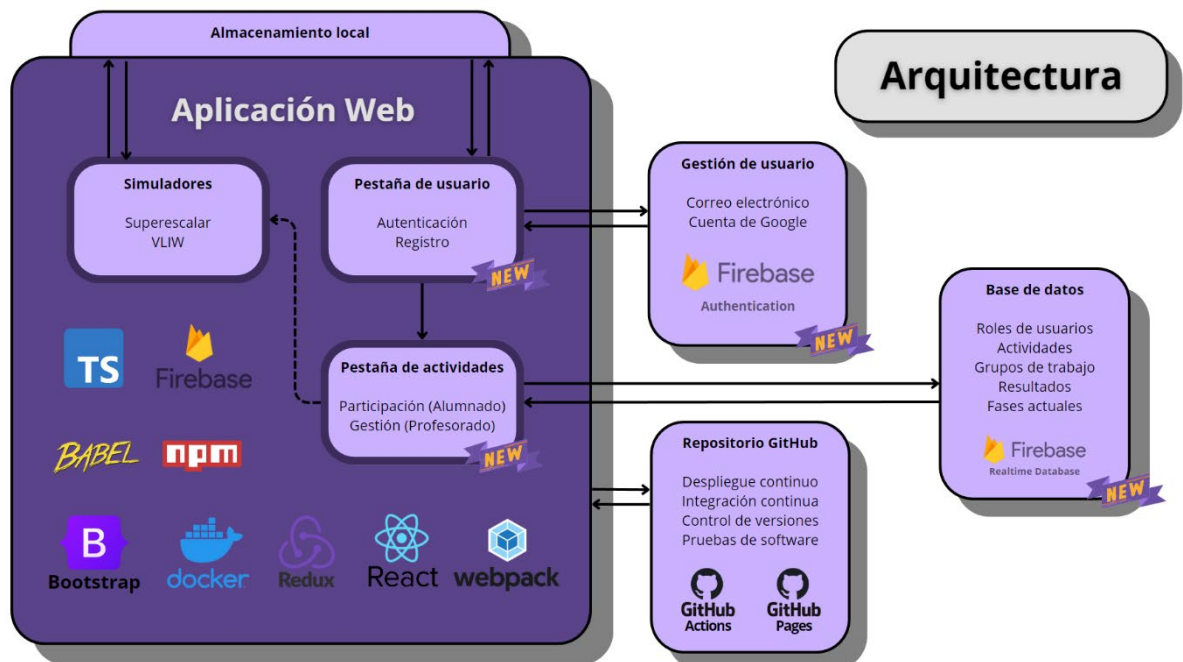


Figura 5.1: Arquitectura de la aplicación Web

5.1 Gestión de usuarios

Los usuarios tendrán la posibilidad de registrarse y autenticarse con una cuenta de correo y contraseña, que es uno de los métodos que ofrece la plataforma. En el caso de que un usuario se registre en la aplicación se almacenaría en los registros de Firebase Authentication y se crearía un nuevo objeto en los registros de Firebase Realtime Database para almacenar datos adicionales como el nombre o el rol de usuario.

Adicionalmente, hemos decidido incluir el inicio de sesión mediante una cuenta de Google, teniendo en cuenta que en el ámbito universitario de la ULL todos los alumnos disponen de una cuenta asociada a Google y con el objetivo de facilitar el acceso a la plataforma, además de reducir el tiempo de acceso. Al igual que en el caso de registro, el usuario pasará primero por Firebase Authentication para comprobar sus credenciales y así proceder a obtener su información de la base de datos.

5.2 Base de datos

La base de datos ha sido diseñada para gestionar la información de los usuarios y las actividades en que las que podrán participar. El propósito principal de esta base de datos es almacenar, recuperar y gestionar eficazmente los datos almacenados. Dentro de la base de datos existen dos rutas principales, por una parte, existe la ruta para acceder a los datos de las actividades y otra para acceder a los datos de los usuarios registrados.

En el modelo de datos (Figura 5.2) se muestra toda las rutas y objetos de forma abstracta utilizando ejemplos. Las líneas discontinuas indican que pueden existir más elementos con los mismos atributos como los que conecta la flecha continua. Por ejemplo, en la ruta “Activities” pueden encontrarse más de una actividad. Además, los tres puntos dentro de un atributo en un objeto, indica que ese atributo puede repetirse y adoptar otra clave con otro valor. Por ejemplo, en la ruta “/Activities/{ID de la actividad}/Tasks” habrá tantos atributos como tareas, de tal manera que las claves corresponderán con un identificador para la tarea y el valor a la descripción de la misma.

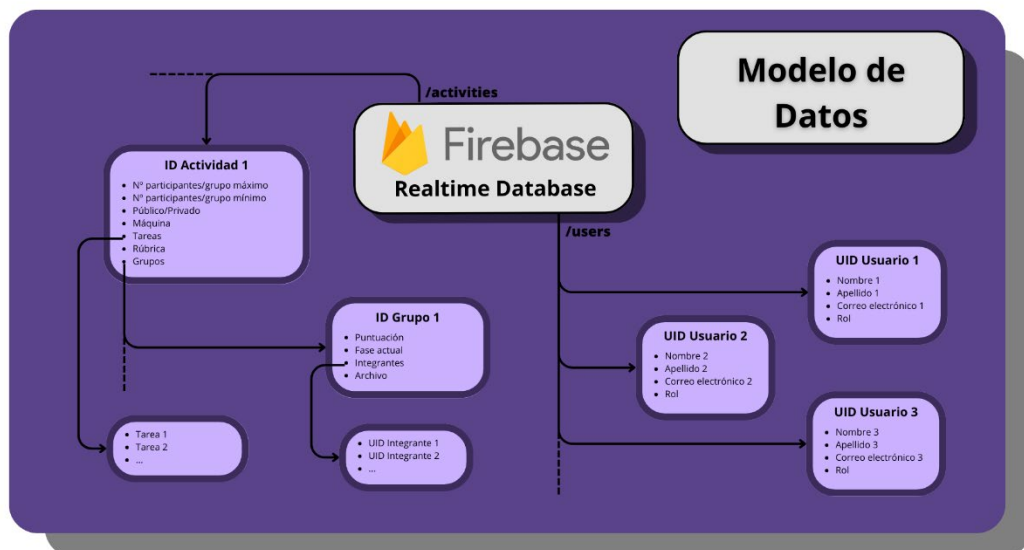


Figura 5.2: Modelo de datos

5.2.1 Actividades

En la ruta “Activities” existe un listado de objetos que almacenan los datos de una actividad, el listado de objetos puede tener cualquier tamaño siempre que el plan contratado de Firebase lo permita (Para este proyecto se utilizó el plan gratuito Spark). Cada actividad tendrá un identificador generado por la plataforma, la cual se encargará de que no se repitan dichos identificadores.

Los atributos de una actividad se componen por:

- Título: Cadena de texto con el título de la actividad.
- Número máximo y mínimo de participantes: Determina si una actividad es individual o grupal.
- Privada/pública: Atributo booleano que indica si la actividad es privada o pública.
- Máquina: Cadena de texto que indica la máquina que se utilizará en las pruebas.
- Tareas: Listado de tareas a realizar para completar la actividad.
- Rúbrica: Criterios de corrección que se deben tener en cuenta en el momento de desarrollo.
- Grupos: Listado de grupos que participan en la actividad. Los grupos pueden estar compuestos por una única persona, siendo el caso de las actividades individuales. Los grupos se almacenan como un objeto con los siguientes atributos:

- Puntuación: Número de ciclos obtenidos en la fase de pruebas.
- Fase actual: Número que indica la fase de desarrollo. En la versión final de este trabajo, esta idea fue dejada al margen del alcance del proyecto, ya que no se utilizan más etapas aparte de la fase de pruebas. Queda como un apartado pendiente que puede ser continuado o replanteado en líneas futuras.
- Integrantes: Listado con los identificadores de usuario de cada integrante del grupo.
- Archivo: Almacena el código entregado por el grupo.

5.2.2 Usuarios

La segunda ruta principal, “Users”, contiene los datos de los usuarios que se han registrado en la aplicación. De igual manera que las actividades, el número de usuarios se ajusta a la capacidad de almacenamiento que el plan permita. Cada objeto de usuario tendrá como identificador el User Identifier (UID) y como valor los datos del propio usuario.

Los atributos de un usuario se componen por el nombre, sus apellidos, su correo electrónico y su rol como usuario. Los roles de usuario planteados son “students” para los estudiantes, “professor” para los docentes y “admin” para los administradores de la aplicación. Por el momento los roles no cumplen un papel importante en la aplicación, esta idea fue implementada pensando en las futuras líneas de desarrollo.

5.3 Almacenamiento local

Además de la base de datos que proporciona Firebase y como bien comentamos en el anterior capítulo. Disponemos de otro “almacén” de datos gracias a la biblioteca Redux. En este proyecto se ha utilizado con anterioridad para el propio funcionamiento de los simuladores, en esta ocasión

hemos optado por utilizar este recurso para almacenar datos que interesan que persistan a lo largo de la navegación por la aplicación web.

Los datos que interesan que persistan son:

- Datos de Usuario: En el momento en el que un usuario se autentica en la aplicación interesa guardar sus datos para mantener la sesión iniciada. De cualquier otro modo se cerraría la sesión con cambiar de página, lo cual no sería conveniente si queremos que el usuario se mueva libremente por la aplicación.
- Código entregable: Considerando el caso de que un usuario con el rol de alumno realiza una entrega, interesa que dicha entrega persista. Ya que al realizar una entrega se cambia de una página a otra al igual que los datos de usuario.

Capítulo 6

Funcionalidades

En este capítulo se comentarán las funcionalidades que tendrán disponibles los usuarios de SIMDE Web. Indicando algunos casos de uso y la manera de moverse por la plataforma. En la Figura 6.1 se puede observar cómo están conectadas las páginas web de la aplicación.

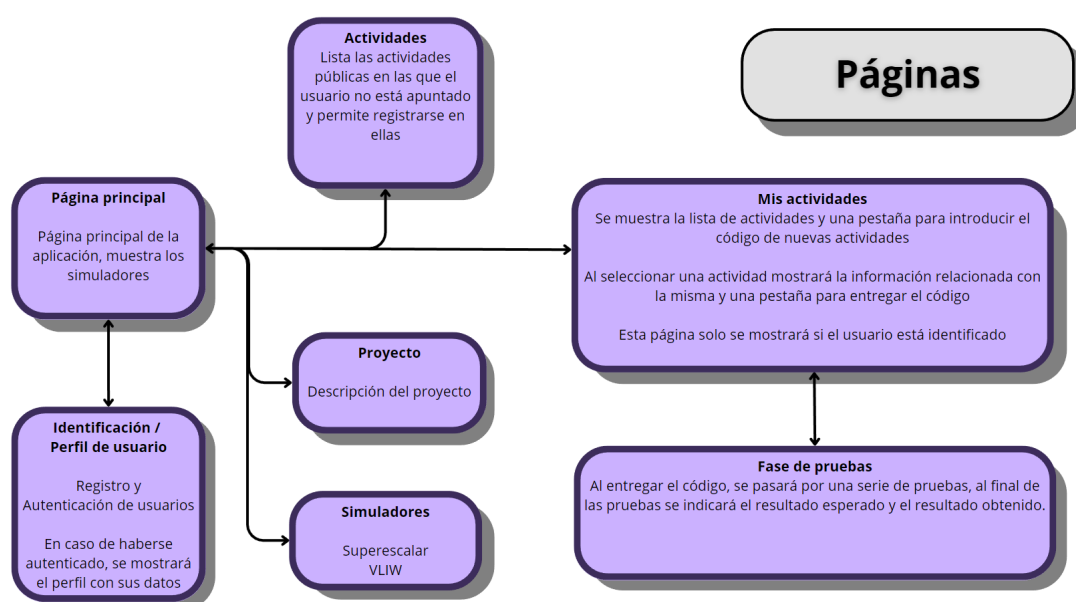


Figura 6.1: Esquema de páginas web

6.1 Cuenta de usuario

6.1.1 Registro e Inicio de sesión

Para que un usuario pueda participar en alguna actividad, es obligatorio que disponga de una cuenta registrada. Por ello esta funcionalidad fue de las primeras en implementarse, ya que basándonos en la planificación de esta plataforma el usuario no podría utilizar ninguna de las nuevas funcionalidades sin haberse registrado previamente.

La página de Registro e Inicio de sesión se encuentran a la derecha de la barra de navegación, se presenta la posibilidad de registrarse y autenticarse con correo electrónico y contraseña, en el que las contraseñas deben tener al menos seis caracteres y los nombres deben tener al menos uno. Además, se permite una cuenta oficial de Google tal y como se menciona en el anterior capítulo.

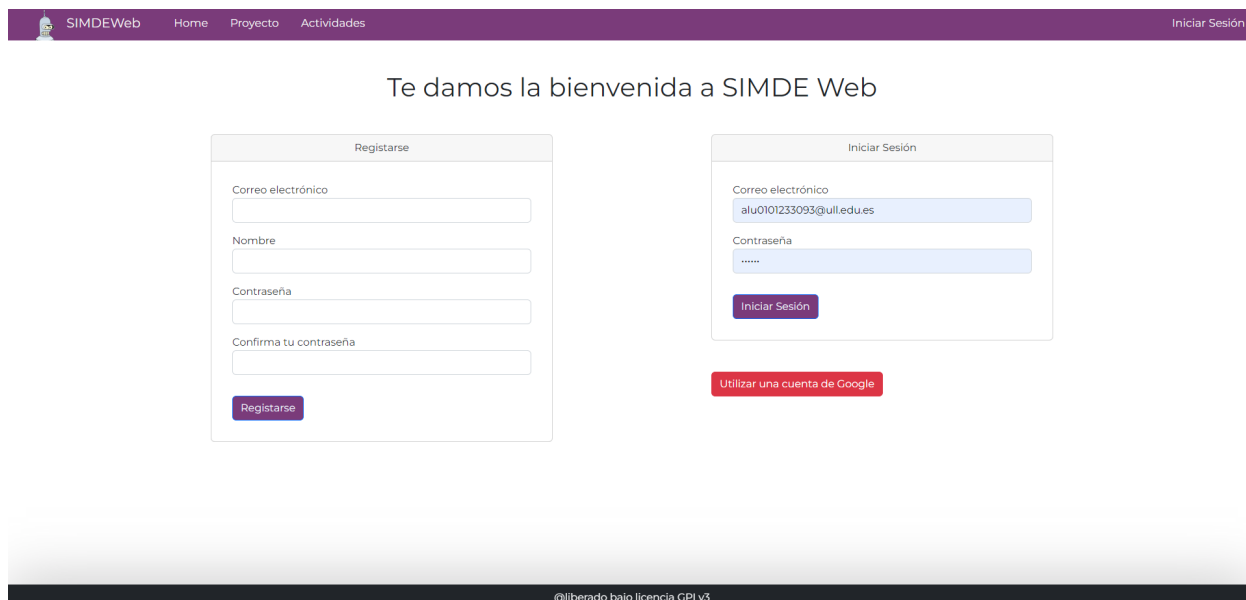


Figura 6.2: Página de Registro/Inicio de sesión

6.1.2 Perfil de usuario

El perfil de usuario lógicamente no se mostrará hasta que el usuario se autentique. Por ello, hasta que no se logre identificar al usuario esta página permanece oculta en la barra de navegación.

Dentro de la página de perfil se pueden observar los datos del usuario y dos botones (Figura 6.3). Uno de los botones permite cerrar sesión, lo que permite cambiar de cuenta si el usuario lo desea o si se dispone a abandonar la aplicación. El segundo botón tiene la funcionalidad de eliminar al usuario, borrando sus datos en la plataforma de autenticación y en la propia base de datos.

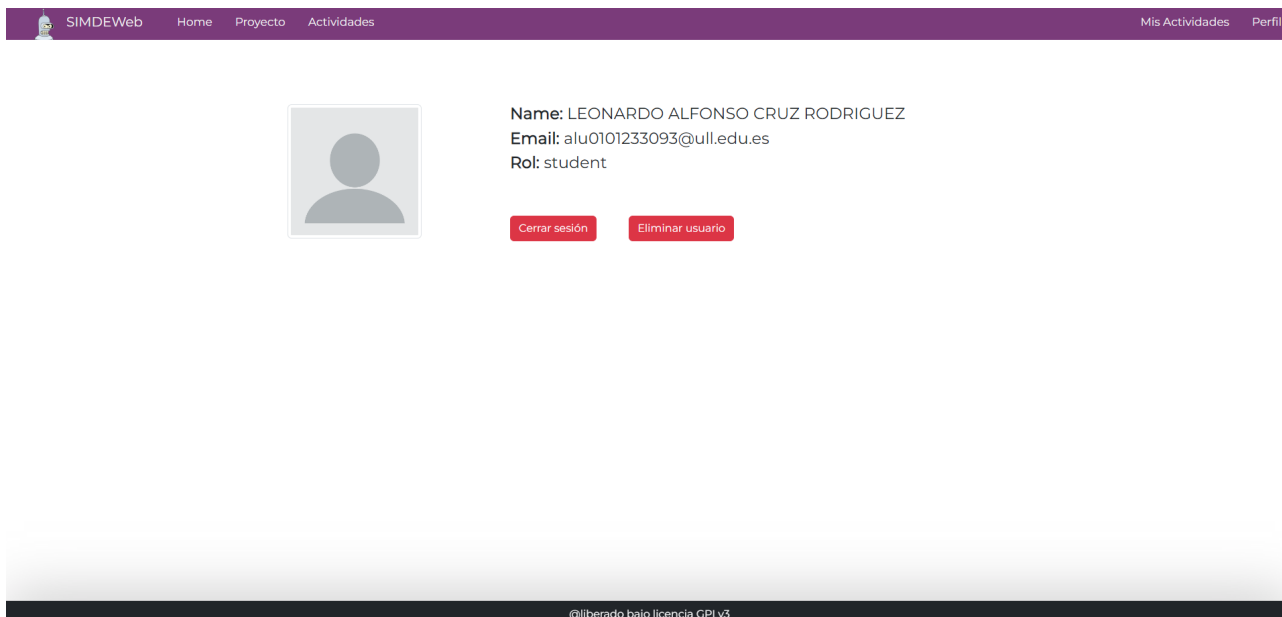


Figura 6.3: Página de perfil de usuario

6.2 Participación en Actividades

6.2.1 Actividades públicas y privadas

En esta aplicación web están contemplados varios tipos de actividades, las principales categorías son las actividades públicas y las privadas.

Las actividades públicas son aquellas a las que cualquier usuario registrado puede apuntarse, accediendo a la página de “Actividades” ubicada en la barra de navegación se muestra un listado de las actividades públicas a las que el usuario no está registrado (Figura 6.4). Simplemente con seleccionar alguna el usuario queda registrado en la actividad y tendrá acceso a la información de dicha actividad.

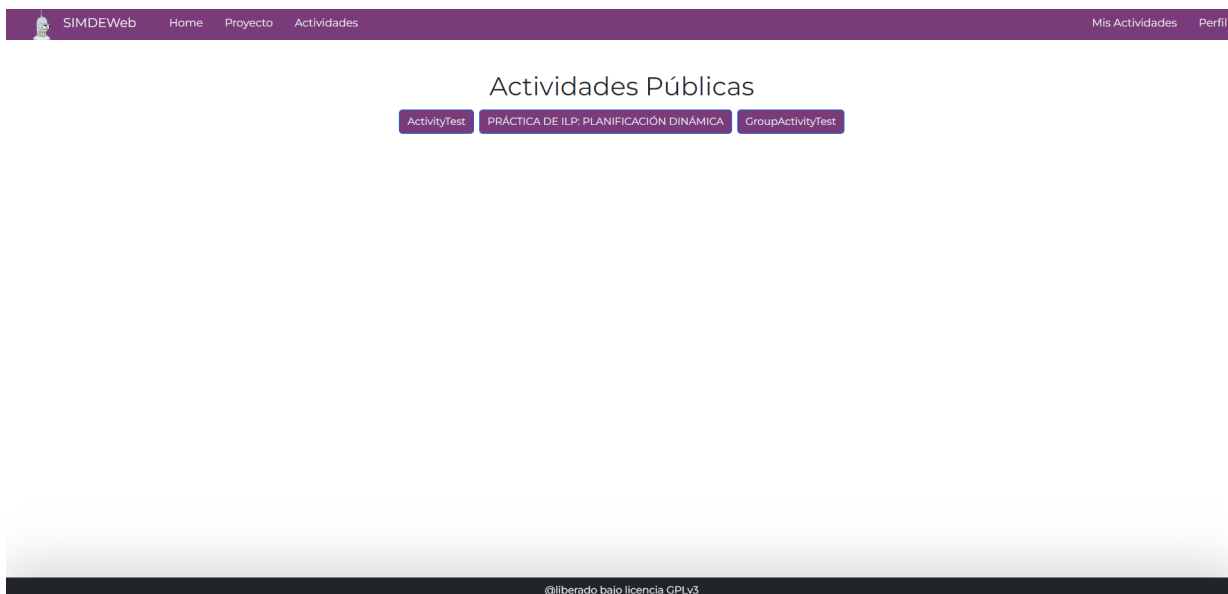


Figura 6.4: Página de actividades públicas

Por otra parte, existe también la categoría de actividad privada. La principal diferencia con respecto a las actividades públicas es que son accesibles únicamente conociendo el identificador de la actividad. En la teoría, los docentes serán los encargados de proporcionar este código a los estudiantes, de tal manera que los profesores administradores son los encargados de difundir la identificación a un grupo de personas seleccionadas.

Para registrarse en una actividad privada, el usuario deberá entrar en la pestaña de Mis Actividades (Figura 6.5). Al entrar se encontrará con una entrada de texto en la que podrá introducir el código y registrarse.

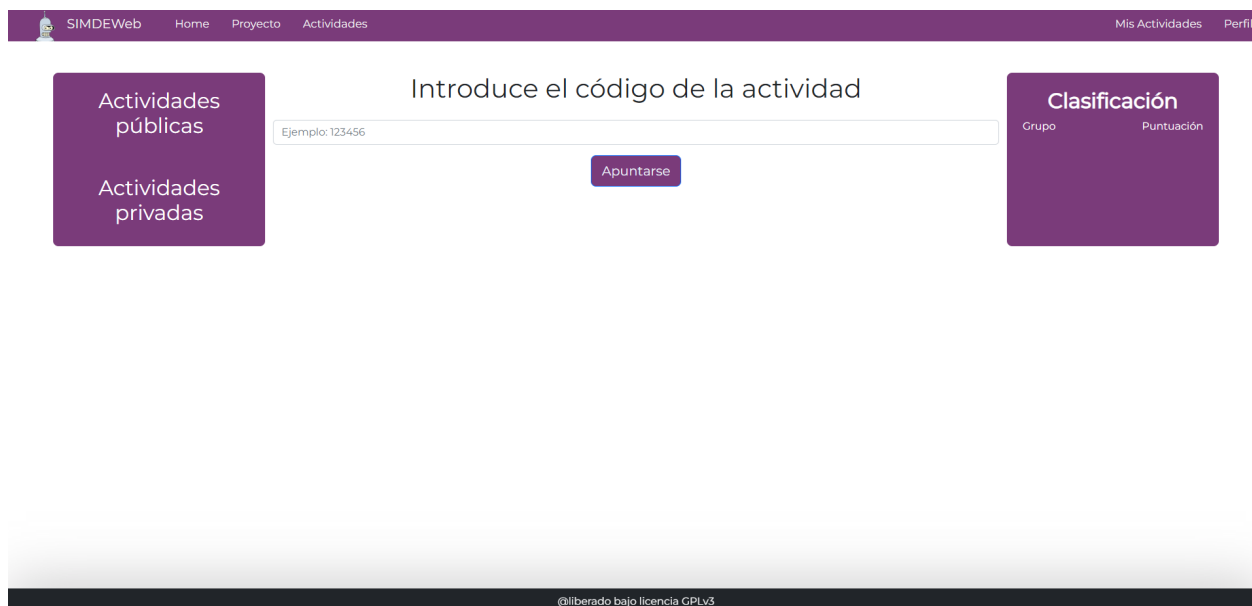


Figura 6.5: Registro a actividades privadas

Adicionalmente a estas dos categorías, toda actividad debe definirse como actividad individual o actividad grupal. A nivel lógico, todas las actividades poseen grupos, sin embargo, en el caso de las actividades que se consideran individuales los grupos no pueden poseer más de un integrante. En cualquier otro caso en el que el máximo de integrantes sea mayor que uno, la actividad se considerará grupal.

En el caso de las actividades grupales, el equipo docente/administrativo será el encargado de crear los grupos. De esta forma, los usuarios estudiantes se les mostrarán los grupos con plazas disponibles y este elegirá en base a ello, como puede verse en la Figura 6.6.

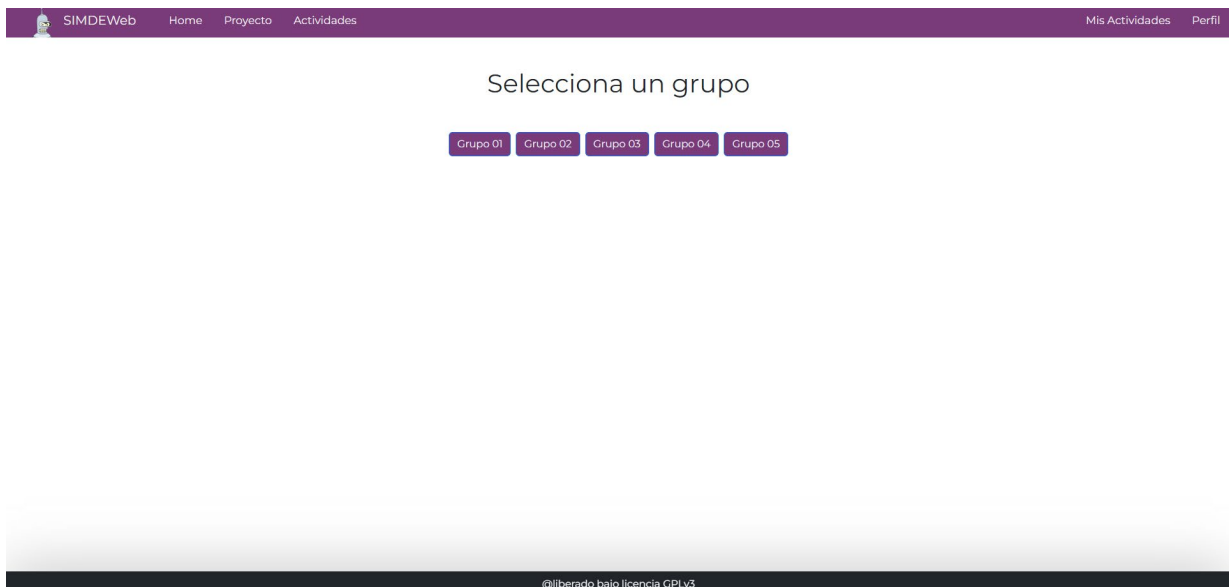


Figura 6.6: Selección de grupo

6.2.2 Mis actividades

Las actividades a las que el usuario se registra son mostradas en esta página (Figura 6.7), que al igual que la página de perfil de usuario, únicamente es accesible cuando el usuario inicia sesión. En la parte de la izquierda se muestran las actividades a las que el usuario se ha registrado organizadas por las categorías públicas y privadas.

Si una actividad de esta pestaña es seleccionada, en la parte central de la página se mostrará la información de la actividad (título, rúbrica y tareas). A la derecha se mostrará la clasificación ordenada de mayor a menor puntuación de cada grupo o participante, dependiendo si se trata de una actividad individual o grupal. Debajo de los laterales se presentan dos botones, en la zona inferior izquierda el botón “Nueva actividad”, con la utilidad de mostrar la pestaña de registro a actividades privada. Por otra parte, en la zona inferior derecha el botón “Fase de pruebas” que permitirá entregar el código que se encuentre en la caja de texto situado en la zona central y pasar a la siguiente fase.

En la anteriormente mencionada zona central, se mostrará una pestaña para escribir el entregable de la actividad y adicionalmente una zona “Drag & drop” en caso de que el usuario tenga el código en un archivo de texto. Cuando el usuario esté conforme con su entrega deberá hacer clic en el botón “Fase de pruebas” para pasar el código por distintas pruebas planteadas por los docentes.

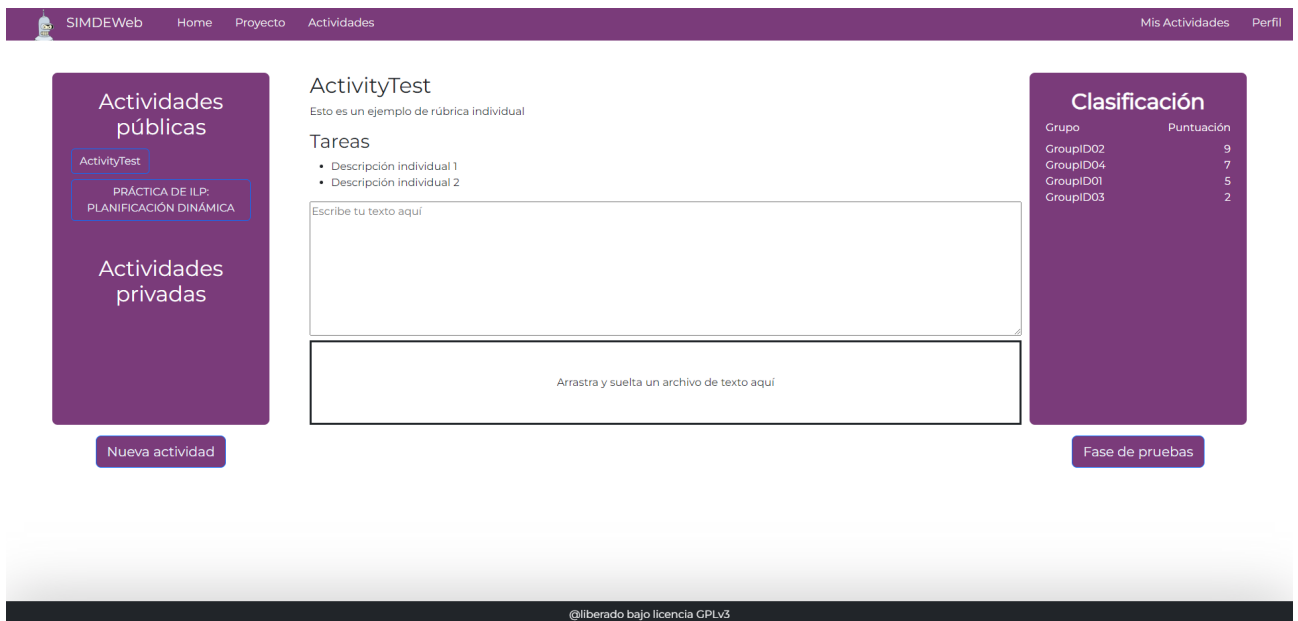


Figura 6.7: Página Mis Actividades

6.2.3 Fase de pruebas

Una vez el usuario haya desarrollado el código de cualquier actividad a la que esté suscrito, utilizando la pestaña de escritura o la zona “Drag & Drop”, tendrá la posibilidad de comprobar el número exacto de ciclos que necesitó la máquina que se utilice en la actividad.

En el momento de pulsar el botón “Fase de pruebas” se redirigirá al usuario a una página en la que se le mostrará si hubo algún error, el tipo de máquina utilizada y el número de ciclos empleados para el algoritmo que le ha indicado el usuario, como se puede observar en la Figura 6.8. Una vez el usuario haya leído la información y decida volver a la página “My Activities”, se registrará en la base de datos tanto su código, como su puntuación (número de ciclos).

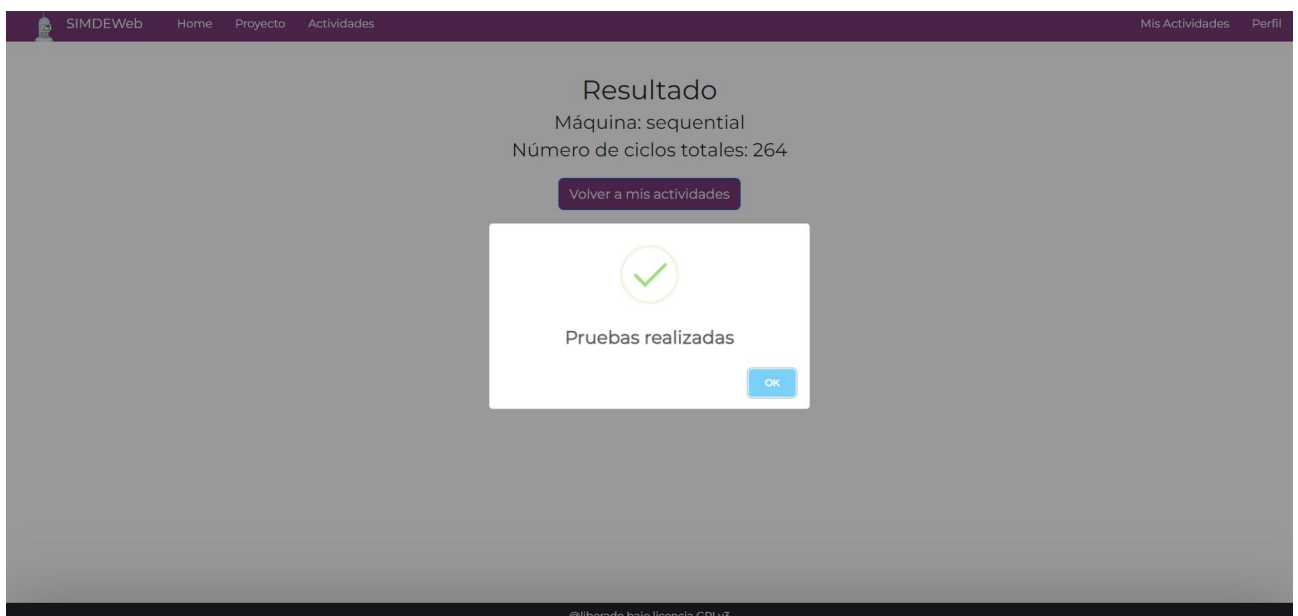


Figura 6.8: Página para fase de pruebas

Capítulo 7

Conclusiones y líneas futuras

7.1 Conclusiones

Realizando este trabajo he podido mejorar mucho mis conocimientos a cerca del desarrollo de aplicaciones web. Además de adquirir experiencia planificando proyectos de una magnitud mayor a cualquier práctica que se haya podido plantear en la carrera.

Como resultado se ha logrado una aportación a la aplicación SIMDE Web con nuevas funcionalidades que permitieron ludificar la plataforma a través de actividades planteadas por los docentes para los estudiantes.

7.2 Líneas futuras

Finalizando este trabajo, quedan abiertas diversas líneas de desarrollo:

- **Mejorar los entregables:** Actualmente la aplicación solo permite un entregable, que es un único código. En un futuro se puede ampliar las opciones de entrega y permitir distintos códigos para una misma actividad.
- **Ampliar las opciones de registro:** Se podría considerar la opción de registrarse a través de otros servicios externos, ya que Firebase ofrece una gran variedad de opciones, como por ejemplo GitHub, Facebook o Twitter.
- **Incluir más fases:** En un principio se planteó incluir tres fases de desarrollo en las actividades. En la primera se plantearía el problema a resolver, indicando toda la información necesaria de la actividad. La segunda fase consistiría en pasar el entregable por una serie de pruebas (equivalente a la fase de pruebas ya implementada) y una tercera fase en la que se muestra el ranking y puedes volver a la fase uno en caso de querer mejorar la puntuación.
- **Definir plazos:** En las actividades podrían definirse distintos plazos. Un primer plazo para apuntarse y formar grupos en caso de que se trate de una actividad grupal, un segundo plazo que consistiría en la realización de la actividad y por último un tercer plazo en el que únicamente se pueden ver los resultados obtenidos. Dando por finalizada la actividad en ese punto.
- **Proteger zonas de memoria:** En el momento de realizar las pruebas, sería interesante proteger ciertas zonas de memoria en los simuladores. De esa forma habría más control a la hora de tomar registros y se valoraría mejor la optimización de los entregables.

Capítulo 8

Summary and Conclusions

8.1 Summary

By completing this work, I have enhanced my knowledge about web application development. I have also gained experience in planning projects of a larger magnitude than any other practice proposed throughout my studies.

As a result, I have contributed to SIMDE Web application by adding new features that allow gamification of the platform through activities designed by teachers for students.

8.2 Future work lines

At the end of this work, there are lines of development that remain open:

- **Improve deliverables:** Currently, the application only allows a single deliverable, which is a unique piece of code. In the future, the delivery options could be expanded to allow different codes for the same activity.
- **Expand SignUp options:** The option of registering via other external services could be considered, as Firebase offers a wide variety of options, such as GitHub, Facebook, or Twitter.
- **Include more phases:** Initially, the idea was to include three development phases in the activities. The first one would present the problem to be solved, providing all the necessary information about the activity. The second phase would involve passing the deliverable through a series of tests (equivalent to the testing phase already implemented), and a third phase in which the ranking is displayed, and you can return to phase one if you want to improve the score.
- **Define deadlines:** Activities could define different deadlines. An initial period for sign-ups and group formation in the case of group activities, a second period for the activity itself, and finally a third period where only the results obtained can be viewed, effectively ending the activity at that point.
- **Protect memory areas:** When conducting tests, it would be interesting to protect certain memory areas in the simulators. This would give more control over record-taking and better assess the optimization of the deliverables.

Capítulo 9

Presupuesto

Descripción	Coste por hora	Coste total
195 horas de trabajo	35€	6825€
Nota: En el precio por hora se incluye la cuota de autónomo, gastos de suministros y amortización de equipos para el desarrollo.		

Tabla 5: Presupuesto general

Bibliografía

- [1] Kapp, K. M. (2012). *The Gamification of Learning and Instruction: Game-based Methods and Strategies for Training and Education*. John Wiley & Sons.
- [2] Hamari, J., Koivisto, J., & Sarsa, H. (2016). Does gamification work? A literature review of empirical studies on gamification. In *Proceedings of the Annual Hawaii International Conference on System Sciences (HICSS)*, 3025-3034.
- [3] Hanus, M. D., & Fox, J. (2015). Assessing the effects of gamification in the classroom: A longitudinal study on intrinsic motivation, social comparison, satisfaction, effort, and academic performance. *Computers & Education*, 80, 152-161.
- [4] Dicheva, D., Dichev, C., Agre, G., & Angelova, G. (2015). Gamification in education: A systematic mapping study. *Educational Technology & Society*, 18(3), 75-88.
- [5] Iván Castilla Rodríguez. *Simulador didáctico de arquitectura de computadores*. PhD thesis, Universidad de La Laguna, 2004.
- [6] Adrián Abreu González. «Simulador didáctico de arquitectura de computadores».
- [7] Antonio Jesús López Garnier. «Plataforma de ludificación de un simulador didáctico de arquitectura de computadores».
- [8] Melissa Díaz Arteaga. «Simulador didáctico de una arquitectura de planificación estática».
- [9] Óscar Carrasco Benítez. «Simulador didáctico de arquitectura de computadores: aplicación de metodologías de integración y mantenimiento»
- [10] Documentación de GitHub Pages. <https://docs.github.com/es/pages>
- [11] Tabla de Hitos y tareas en Notion. <https://real-rotate-abb.notion.site/1b6787e369e8474c9b390012845ed7fc?v=6e6825ebf99c46cea6bc89183e31e3df>
- [12] What is Redux? Store, Actions, and Reducers Explained for Beginners. <https://www.freecodecamp.org/news/what-is-redux-store-actions-reducers-explained/>
- [13] Documentación de Bootstrap. <https://getbootstrap.com/docs/5.3/getting-started/introduction>
- [14] Guía de Sweet Alerts. <https://sweetalert.js.org/>
- [15] Repositorio de este trabajo. <https://github.com/alu0101233093/SIMDE>

- [16] Repositorio Original de Óscar Carrasco Benítez. <https://github.com/oxcabe/SIMDE>
- [17] How to Sync Forked Repositories Using Git or Github. <https://levelup.gitconnected.com/how-to-sync-forked-repositories-using-git-or-github-2933e497fa16>
- [18] Documentación de WebPack. <https://webpack.js.org/concepts/>
- [19] Documentación de Docker. <https://docs.docker.com/>
- [20] Documentación general de Firebase. <https://firebase.google.com/docs>
- [21] Documentación de Firebase Authentication. <https://firebase.google.com/docs/auth>
- [22] Cómo elegir tu base de datos: Cloud Firestore o Realtime Database. <https://firebase.google.com/docs/database/rtdb-vs-firestore>