



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática



TFG del Grado en Ingeniería
Informática

TeachMePlay



Presentado por Estela Victoria Ballester Delgado
en Universidad de Burgos — 28 de mayo de 2023

Tutor: José Manuel Galán Ordax y Virginia
Ahedo García



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática



D. José Manuel Galán Ordax y Da Virginia Ahedo García, profesores del departamento de Ingeniería Civil, área de Ingeniería de Organización.

Expone:

Que el alumno Dña. Estela Victoria Ballester Delgado, con DNI 71565429D, ha realizado el Trabajo final de Grado en Ingeniería Informática titulado TeachMePlay.

Y que dicho trabajo ha sido realizado por el alumno bajo la dirección del que suscribe, en virtud de lo cual se autoriza su presentación y defensa.

En Burgos, 28 de mayo de 2023

Vº. Bº. del Tutor:

Vº. Bº. del co-tutor:

D. nombre tutor

D. nombre co-tutor

Resumen

Este proyecto es una aplicación web que consiste en un repositorio de juegos docentes. Permite a los usuarios, según su rol de acceso, buscar y filtrar juegos, interactuar mediante valoraciones, y también añadir y modificar nuevos juegos.

Descriptores

Gamificación, aplicación web, Flask, Python, PostgreSQL, HTML, CSS, Heroku, SCRUM, Overleaf, GitHub...

Abstract

This project is a web application that consists of a repository of educational games. It allows users, according to their access role, to search and filter games, interact through ratings, and also add and modify new games.

Keywords

Gamification, web application, Flask, Python, PostgreSQL, HTML, CSS, Heroku, SCRUM, Overleaf, GitHub...

Índice general

Índice general	iii
Índice de figuras	v
Índice de tablas	vi
Introducción	1
1.1. Estructura de la memoria	2
1.2. Estructura de los anexos	2
1.3. Materiales adjuntos	3
Objetivos del proyecto	5
2.1. Objetivos generales	5
2.2. Objetivos técnicos	6
2.3. Objetivos personales	6
Conceptos teóricos	7
3.1. Gamificación	7
3.2. Serious games	8
3.3. E-learning	8
Técnicas y herramientas	11
4.1. Gestión del proyecto	11
4.2. Herramientas de documentación	13
4.3. Herramientas de diseño	15
4.4. Desarrollo	15
4.5. Librerías y módulos	21

Aspectos relevantes del desarrollo del proyecto	25
5.1. Inicio del proyecto	25
5.2. Primeros pasos	25
5.3. Metodologías	26
5.4. Formación	27
5.5. Desarrollo del proyecto	27
5.6. Resolución de problemas	33
Trabajos relacionados	35
6.1. The Gamification Repository	35
6.2. Web corporativa e intranet para una consultora tecnológica . . .	36
6.3. Desarrollo de Aplicación Web para Administración de Sensores Inalámbricos Mediante el Estándar BACnet	36
6.4. Desarrollo de una aplicación Web RESTfull basada en Spring y Angular	37
6.5. RACO	37
Conclusiones y Líneas de trabajo futuras	39
7.1. Conclusiones	39
7.2. Líneas de trabajo futuras	40
Bibliografía	41

Índice de figuras

4.1. Metodología ágil: Scrum	12
4.2. GitHub-logo	12
4.3. GitHub Desktop-logo	13
4.4. ZenHub-logo	13
4.5. Zube-logo	13
4.6. LaTeX-logo	14
4.7. Overleaf-logo	14
4.8. Zotero-logo	14
4.9. Gitbook-logo	15
4.10. Figma-logo	15
4.11. Draw.io-logo	15
4.12. Esquema protocolo HTTP	17
4.13. Arquitectura MVC	19
4.14. Python-logo	20
4.15. Flask-logo	20
4.16. PostgreSQL-logo	20
4.17. Heroku-logo	21
5.1. Captura de código	30
5.2. Captura de código	30
5.3. Archivos JSON	31
5.4. Botones para cambiar de idioma	32
5.5. Conexión con repositorio de GitHub	32
5.6. Add-on Heroku Postgres	32

Índice de tablas

4.1. Diferencias entre los métodos GET y POST	18
---	----

Introducción

La gran cantidad de horas lectivas que se invierten en las formaciones académicas puede ser un obstáculo en el aprendizaje de los alumnos, ya que disminuye el grado de concentración durante las clases. Por ello, en los últimos años han surgido dos nuevos conceptos como posible solución a este problema: gamificación y serious games.

[53] La gamificación es un sistema de enseñanza que busca el aprendizaje a través de juegos. El principal objetivo que pretende conseguir es obtener la motivación de los alumnos a través de sistemas de puntuación que les proporcionen puntos de recompensa al cumplir diferentes propósitos. De esta forma, se emplean los medios y mecanismos de juego para aumentar tanto el conocimiento académico, como sus resultados.

Por ello, las principales ventajas que proporciona son la motivación, el aumento del rendimiento y de la colaboración, la utilización y familiarización de las TICs, junto con el progreso personal.

Los avances tecnológicos y la aparición del e-learning están permitiendo el aumento de la creación y disposición de los cada vez más conocidos como juegos docentes. Actualmente, cada vez se encuentran disponibles más herramientas educativas. Entre las mas conocidas se incluyen: Kahoot, Edmodo, Socrative, Quizizz, entre otras.

[28] Los serious games son aquellos juegos que se emplean principalmente en diferentes ámbitos de la enseñanza con el objetivo de que los estudiantes aprendan de una forma más interactiva y divertida.

[37] La gamificación y los serious games se diferencian fundamentalmente en que la gamificación no se considera un juego, sino que emplea mecanismos de juego con el fin de obtener puntos de recompensa al cumplir objetivos. En

cambio, los serious games son juegos completos que pretenden el aprendizaje y formación en distintas áreas.

Este proyecto tiene como objetivo la creación de una aplicación web que permita tanto la visualización como la interacción con el conjunto de juegos docentes relevantes para la docencia de distintas materias. De esta forma, al disponer de los juegos cargados en la base de datos se facilitará su gestión y organización entre los docentes, además gracias a la interacción mediante sistemas de puntuación posibilitará a los usuarios búsquedas adaptadas a sus intereses.

1.1. Estructura de la memoria

En la memoria se encuentran los siguientes apartados:

- **Introducción:** presentación del tema del proyecto que se va a desarrollar. Se incluye la estructura de la memoria y la estructura de los anexos.
- **Objetivos del proyecto:** definición de los objetivos generales, técnicos y personales a alcanzar durante el desarrollo del proyecto.
- **Conceptos teóricos:** explicación de los conceptos teóricos fundamentales para la adecuada comprensión del proyecto.
- **Técnicas y herramientas:** exposición de las técnicas y herramientas que se van a emplear para el desarrollo y cumplimiento de los objetivos establecidos del proyecto.
- **Aspectos relevantes del desarrollo:** muestra de las etapas más notables durante la elaboración del proyecto.
- **Trabajos relacionados:** exposición de distintos trabajos y proyectos ya finalizados en el campo del proyecto en desarrollo.
- **Conclusiones y líneas de trabajo futuras:** presentación de las conclusiones obtenidas al final del desarrollo, junto con posibles mejoras que se puedan aplicar al resultado final.

1.2. Estructura de los anexos

En el anexo se encuentran los siguientes apartados:

- **Plan de proyecto de software:** presentación de la planificación temporal del proyecto junto con su análisis económico y legal.
- **Especificación de requisitos:** descripción de los requisitos establecidos que el proyecto debe alcanzar.
- **Especificación de diseño:** muestra el diseño de datos, diseño procedimental, diseño arquitectónico e interfaces.
- **Manual del programador:** guía que incluye los aspectos más importantes respecto al código fuente del proyecto.
- **Manual del usuario:** guía que explica al usuario el uso adecuado de la aplicación.

1.3. Materiales adjuntos

Además de la documentación que incluye la memoria y los anexos, se adjuntan los siguientes materiales:

- Repositorio en GitHub.
- Código del proyecto.
- Proyecto desplegado.

Objetivos del proyecto

En este apartado se explican los objetivos que se pretenden alcanzar durante el desarrollo del proyecto.

Estos objetivos se clasifican en objetivos generales y objetivos técnicos, siendo estos últimos los necesarios para el cumplimiento de los objetivos generales.

2.1. Objetivos generales

Los objetivos generales establecidos son los siguientes:

- Crear una aplicación web que ofrezca una amplia variedad de juegos educativos para su uso en el aula, incluyendo juegos para diversas materias y niveles educativos.
- Desarrollar una base de datos bien estructurada y eficiente que almacene toda la información relevante sobre los juegos docentes, incluyendo detalles sobre los juegos, las cuentas de usuario y las solicitudes de rol.
- Proporcionar a los usuarios diferentes roles y permisos según sus necesidades y objetivos, incluyendo roles de usuario, profesor y administrador, cada uno con diferentes niveles de acceso y funcionalidades.
- Permitir que los usuarios interactúen con la aplicación y con los juegos mediante puntuaciones y comentarios.
- Fomentar el aprendizaje y la educación a través de juegos educativos, proporcionando una herramienta efectiva para el aprendizaje.

2.2. Objetivos técnicos

Los objetivos técnicos establecidos son los siguientes:

- Garantizar la disponibilidad y accesibilidad de la aplicación en cualquier momento siempre que se tenga conexión a la red.
- Crear una interfaz visualmente atractiva que facilite su uso a los usuarios.
- Crear una aplicación interactiva que permita a los usuarios realizar un feedback.
- Aplicar de la metodología ágil Scrum para la organización del desarrollo del proyecto.
- Emplear el gestor de referencias bibliográficas Zotero.
- Usar la herramienta Zube para la planificación, seguimiento y gestión de las tareas y objetivos del proyecto

2.3. Objetivos personales

Los objetivos personales establecidos son los siguientes:

- Aprender a utilizar el framework Flask en Python para crear aplicaciones web desde cero.
- Mejorar mi conocimiento y habilidades en múltiples lenguajes de programación como Python, HTML, CSS y SQL.
- Desplegar la aplicación en línea para que pueda ser accesible desde cualquier lugar.

Conceptos teóricos

En esta sección, se presentan los conceptos teóricos clave que sustentan el desarrollo del proyecto. Estos conceptos son fundamentales para comprender el contexto en el que se desarrolla el proyecto y su importancia en el ámbito en el que se enmarca.

3.1. Gamificación

[9] La gamificación es un método de aprendizaje que emplea la mecánica de los juegos en el ámbito educativo y profesional, con el objetivo de aumentar y mejorar los resultados mediante el uso de entornos digitales y educativos.

Gracias a su carácter lúdico, la gamificación facilita la obtención de conocimientos, ya que ofrece una experiencia positiva al usuario y su objetivo principal es la motivación de los estudiantes. Para ello, se emplean técnicas mecánicas y dinámicas de los juegos.

[30] Las técnicas mecánicas son aquellas que buscan recompensar al usuario según el cumplimiento de los requisitos del juego. Alguno de los elementos de compensación empleados son:

- Clasificaciones: clasificar a los estudiantes en función de sistemas de puntos o logros.
- Acumulación de puntos: asignar valores cuantitativos en función de tareas logradas.
- Misiones o retos: superación de objetivos establecidos.
- Escalado de niveles: superación de niveles según los objetivos logrados.

Mientras que las técnicas dinámicas son aquellas que buscan la motivación del usuario para que continúe y complete sus objetivos. Algunas de las técnicas dinámicas empleadas son:

- Estatus: establecer niveles jerárquicos.
- Recompensa: obtener beneficios.
- Logro: superación personal.

De esta forma la integración de la gamificación en entornos educativos y profesionales, hace que se creen experiencias más atractivas e interactivas para los alumnos, por lo que aumenta su compromiso y rendimiento.

Así, la gamificación se conoce como un método innovador y eficaz para mejorar la motivación y el rendimiento en el ámbito educativo.

3.2. Serious games

[29] Los serious games son juegos diseñados para proporcionar formación en diversos ámbitos, como la educación, la ciencia y la ingeniería. Este concepto surge de la investigación de las distintas modalidades en las que los juegos pueden incluir métodos de aprendizaje y enseñanza sin comprometer la diversión.

Las estrategias lúdicas empleadas por los serious games permiten que los alumnos se involucren más en los contenidos educativos y disfruten de una experiencia de aprendizaje más atractiva y motivadora.

En España, existen diversas empresas e iniciativas que utilizan los serious games para la formación educativa, tales como Humantiks, Gamelearn®, Binnakle, Netlanguages y Chiara.

3.3. E-learning

[54] El e-learning es un concepto que se refiere a las actividades formativas que se realizan a través de dispositivos conectados a la red, lo que lo convierte en una forma de aprendizaje conocida como formación en línea o virtual. Esta modalidad facilita la educación online y permite a los estudiantes la formación a distancia.

Las ventajas y características más destacadas del e-learning son la flexibilidad que proporciona a los estudiantes y la riqueza de aprendizaje que ofrece

la amplia variedad de recursos educativos que se encuentran en internet. Estos recursos abarcan todo tipo de modalidades, desde vídeos y juegos hasta herramientas interactivas y simulaciones.

[34] La evolución del e-learning ha llevado al desarrollo de plataformas avanzadas, como Moodle, un software que ha estandarizado la creación de distintas plataformas de e-learning y ha permitido a los profesores crear entornos virtuales para la interacción y compartición de recursos docentes.

El e-learning ha impulsado la creación de nuevas estrategias, como el aprendizaje basado en la gamificación, en proyectos o la personalización del aprendizaje. Estrategias que permiten a los estudiantes aprender de manera más lúdica y participativa, y a los profesores adaptarse a las necesidades de los estudiantes.

Técnicas y herramientas

En este apartado se muestran las distintas técnicas y herramientas empleadas para el desarrollo del proyecto.

Entre estas técnicas y herramientas se distinguen las utilizadas para la gestión del proyecto, la documentación y los entornos de desarrollo.

4.1. Gestión del proyecto

Scrum

[35] Es un marco de trabajo colaborativo que permite trabajar en equipo el desarrollo de un proyecto sistematizado en entregas organizadas parcialmente del producto final, en función de la utilidad que ofrecen al destinatario del proyecto.

[57] Esta técnica de metodología ágil permite el avance de los requisitos establecidos del proyecto en planes temporales de duración corta y fija, de manera que a los requisitos se les establece una prioridad según su importancia. De esta forma, permite al equipo organizarse y sincronizarse para alcanzar sus objetivos y mostrar al cliente los resultados obtenidos al finalizar cada sprint.

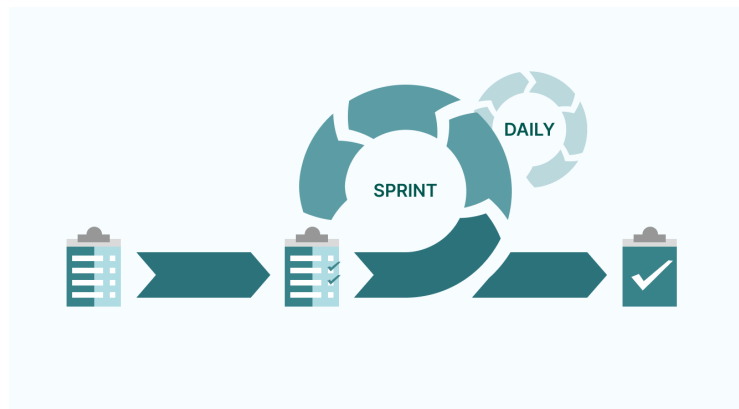


Figura 4.1: Metodología ágil: Scrum

GitHub

[20] Es una plataforma que permite la creación de repositorios de código con el fin de almacenarlos en la nube mediante un sistema de control de versiones conocido como Git. Permite la gestión de proyectos de forma cooperativa en tiempo real.

[15] Facilita el trabajo cooperativo proporcionando distintas herramientas para la gestión de las versiones de las páginas, un sistema para el control de problemas, herramientas para la comprobación de código y un visualizador de las ramas.



Figura 4.2: GitHub-logo

GitHub Desktop

[22] Es una aplicación que posibilita la interacción con GitHub proporcionando una interfaz gráfica de usuario en sustitución a la línea de comandos o buscadores web.

Se trata de la versión de escritorio de GitHub y permite la subida, extracción y clonación de repositorios empleando mecanismos colaborativos.



Figura 4.3: GitHub Desktop-logo

ZenHub

[43] Es una plataforma de gestión de proyectos integrada en GitHub que permite la conexión de repositorios y la organización de los proyectos mediante tableros de trabajo que facilitan la planificación de sprints, la creación de tareas, la estimación de story points y la generación de gráficos e informes estadísticos.



Figura 4.4: ZenHub-logo

Zube

[31] Es un sistema de gestión de proyectos gratuito integrado en GitHub que facilita la asignación y gestión de tareas, permitiendo especificar la prioridad de cada una y promoviendo la colaboración entre los miembros del equipo. Además, su funcionalidad de actualización en tiempo real permite que los cambios realizados se reflejen instantáneamente en el proyecto. Por otra parte, esta plataforma se enfoca en la detección y corrección de errores, contribuyendo a garantizar la calidad y eficiencia del proyecto en todo momento.



Figura 4.5: Zube-logo

4.2. Herramientas de documentación

LaTeX

[60] Es un sistema de composición de textos basado en TeX. Su función consiste en la creación de distintos documentos como libros o artículos técnicos y científicos, proporcionando una alta calidad tipográfica.

Este procesador de textos se basa en instrucciones, por lo que la principal diferencia que ofrece respecto a otros procesadores de texto es la facilidad de incluir el contenido sin necesidad de manejar los detalles del formato permitiendo estructurar de forma muy sencilla el documento, gracias además a sus capacidades gráficas.



Figura 4.6: LaTeX-logo

Overleaf

[48] Es una herramienta en línea para la redacción colaborativa y su publicación. Gracias al editor LaTeX permite la cooperación en tiempo real y la compilación de la salida de forma automática en segundo plano durante la redacción de los textos.



Figura 4.7: Overleaf-logo

Zotero

[61] Es un gestor de referencias bibliográficas gratuito y de código abierto que permite la recopilación y organización de distintos recursos, incluyendo su citación, sincronización y colaboración.



Figura 4.8: Zotero-logo

Gitbook

[49] Es una herramienta que permite la creación de la documentación de proyectos y libros técnicos.



Figura 4.9: Gitbook-logo

4.3. Heramientas de diseño

Figma

[56] Es una herramienta de prototipado principalmente web que permite el diseño de las interfaces de usuario y el diseño de experiencia de las aplicaciones, posibilitando la colaboración en tiempo real.

[32] La principal ventaja de esta herramienta es la posibilidad que ofrece de usar funcionalidades y librerías de componentes creadas y compartidas por la comunidad, lo cual facilita y agiliza el diseño de las interfaces.



Figura 4.10: Figma-logo

Draw.io

[58] Es un software gratuito y de código abierto que permite el dibujo de gráficos, incluyendo la creación de diagramas como diagramas de flujo, diagramas de casos de uso, diagramas de secuencia, entre otros.



Figura 4.11: Draw.io-logo

4.4. Desarrollo

Frontend

[39] El frontend es la parte de la aplicación a la que el usuario puede acceder e interactuar directamente. Se trata de todas las tecnologías de diseño y

desarrollo web que se utilizan en el navegador para proporcionar funcionalidad de interacción con los usuarios. Esto incluye elementos como botones, formularios, menús desplegables, imágenes y otros elementos visuales que constituyen la interfaz de usuario.

Algunos de los principales lenguajes que se utilizan para el desarrollo del frontend de una aplicación son HTML5 y CSS3.

HTML5 se utiliza para definir la estructura de la aplicación web, incorporando elementos que se muestran en la interfaz de usuario.

Por otro lado, CSS3 se encarga del estilo y de la presentación visual de la interfaz. Así, se puede personalizar la apariencia y la distribución de los elementos de la interfaz, proporcionando una experiencia de usuario atractiva y fácil de usar.

Backend

[39] El backend se refiere a la parte de la aplicación que interactúa con la base de datos y el servidor correspondiente.

Es la capa de acceso a los datos del software que no es accesible directamente por los usuarios. Además, el Backend incluye la programación encargada de manejar los datos y procesar las acciones de la aplicación.

[12] El Backend se encarga de comunicarse con el servidor, de forma que el servidor recibe las solicitudes del navegador, y después de que el Backend procese los datos, envía una respuesta al servidor para que este se la devuelva al navegador.

Framework

[4] Un framework es una herramienta utilizada en el desarrollo de software que proporciona una estructura que integra diversos componentes básicos para facilitar la creación de nuevos proyectos.

Existen numerosos frameworks, por lo que dependiendo del lenguaje de programación utilizado y el tipo de proyecto, se debe seleccionar el framework más adecuado.

Protocolo HTTP

[8] El Protocolo de transferencia de hipertexto (HTTP) es el protocolo utilizado para aprobar la transferencia de información a través de distintos tipos de archivos, como XML o HTML.

HTTP se encarga de establecer las convenciones sintácticas y semánticas utilizadas por los elementos de software en la arquitectura web para la comunicación.

Se trata de un protocolo orientado a transacciones, que sigue el esquema petición-respuesta entre el cliente y el servidor. En una transacción HTTP, el cliente envía una petición en un formato determinado al servidor, y este le responde con un mensaje.

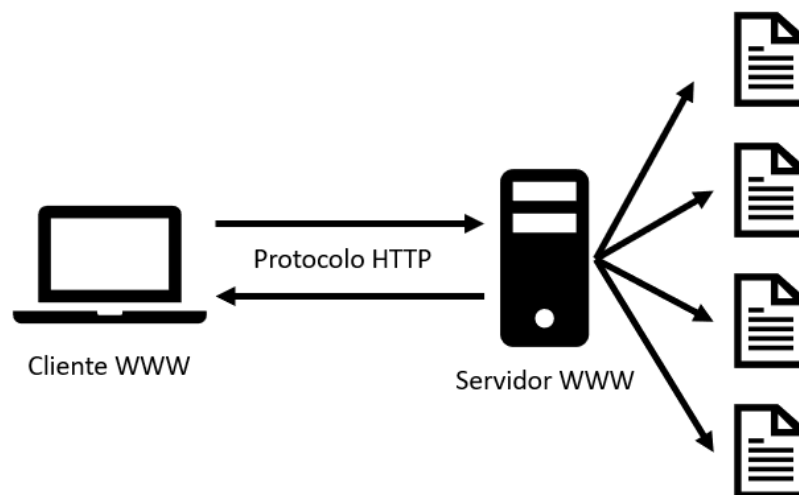


Figura 4.12: Esquema protocolo HTTP

GET

[18] El método GET se emplea para solicitar recursos del servidor web.

POST

[18] El método POST se emplea para enviar datos del cliente al servidor web.

Diferencias entre el método GET y POST

[6] [7] A continuación se incluye una tabla mostrando las diferencias entre los métodos GET y POST.

	GET	POST
<i>Parámetros</i>	URI	Cuerpo
<i>Caché</i>	Los parámetros URL se guardan sin cifrar.	Los parámetros URL no se guardan automáticamente
<i>Marcadores e historiales</i>	Los parámetros URL se guardan junto al URL	Los parámetros URL no se guardan junto al URL
<i>Propósito</i>	Recuperación de documentos	Actualización de datos
<i>Visibilidad</i>	Visible en la barra de direcciones para el usuario	Invisible para el usuario
<i>Tamaño variable</i>	Hasta 2000 caracteres	Hasta 8 Mb
<i>Longitud de datos</i>	Limitado al máximo del URL (2048 caracteres)	Ilimitado
<i>Tipo de datos</i>	Solo caracteres ASCII	Caracteres ASCII y datos binarios

Tabla 4.1: Diferencias entre los métodos GET y POST

Arquitectura MVC

[16] La arquitectura MVC es un tipo de arquitectura de software que se compone de tres componentes: modelo, vista y controlador. Esta arquitectura tiene como objetivo separar los datos de la aplicación, la interfaz de usuario y la lógica de control.

- **Modelo:** se encarga de manejar los datos y la lógica de negocio. En mi aplicación los archivos `database.py`, `juego.py` y `usuario.py`, son los encargados de interactuar con la base de datos, manejando las operaciones de creación, lectura, actualización y eliminación de los datos.
- **Vista:** es la presentación visual e interacción del sistema con el usuario. De forma que en mi aplicación son los templates HTML los que muestran la información al usuario.
- **Controlador:** es el intermediario entre el modelo y la vista, ya que se encarga de recibir las peticiones del usuario y de solicitar los datos al modelo y comunicárselo a la vista. En mi aplicación el controlador es el archivo `app.py` ya que contiene todos los métodos GET y POST, manejando la interacción entre el modelo y la vista, junto con las solicitudes del usuario y devolviendo las respuestas adecuadas.

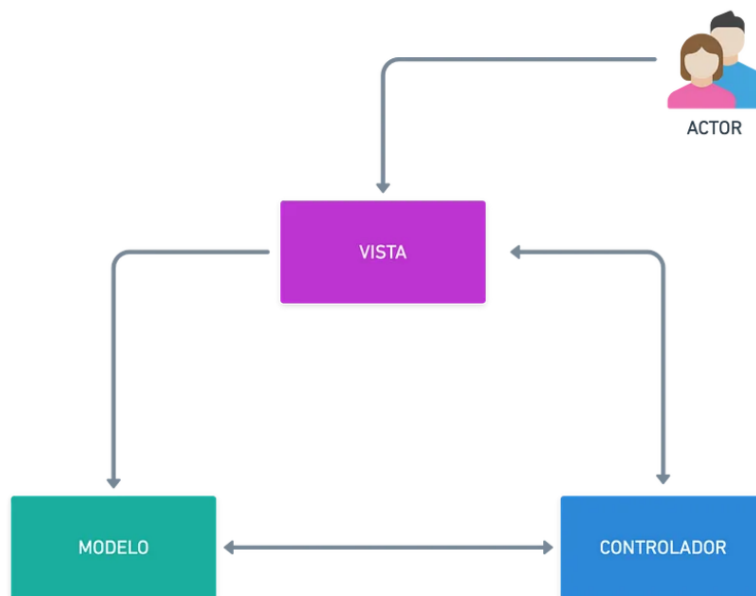


Figura 4.13: Arquitectura MVC

Python

[5] Es un lenguaje de programación de alto nivel empleado mayoritariamente para el desarrollo de software. [52] Al ser un lenguaje interpretado no es necesaria la compilación para la ejecución de las aplicaciones ya que dispone del programa conocido como interpretador, y por lo tanto no requiere la traducción a lenguaje máquina.



Figura 4.14: Python-logo

Flask

[40] Es un micro framework web que permite la creación de aplicaciones web con Python. Se basa en el Modelo-Vista-Controlador (MVC) facilitando el desarrollo de las aplicaciones, y diferenciando el modelo de datos, la vista y el controlador. Esta herramienta proporciona utilidades para el desarrollo de aplicaciones web dinámicas permitiendo su comunicación con bases de datos, la autenticación de usuarios y el desarrollo de formularios, entre otros.



Figura 4.15: Flask-logo

PostgreSQL

[13] Es un sistema de gestión de bases de datos relacionales de código abierto y gratuito. PostgreSQL es conocido por su escalabilidad y la facilidad de administrar grandes cantidades de datos, siendo compatible con herramientas y tecnologías empleados en el proyecto, incluyendo lenguajes de programación como Python y frameworks web, como Flask.



Figura 4.16: PostgreSQL-logo

Heroku

[59] Es una plataforma en la nube que admite distintos lenguajes de programación. Proporciona un servidor para el despliegue de la aplicación y la posibilidad de subir la base de datos a la nube, permitiendo así el acceso a la aplicación de forma online.



Figura 4.17: Heroku-logo

4.5. Librerías y módulos

flask

Librería empleada para el uso del framework Flask.

- **request:** [55] Objeto global que proporciona acceso a los datos enviados por el cliente en una solicitud HTTP.
- **redirect:** [46] Función de Flask que redirige al usuario a otra página en una aplicación Flask.
- **render template:** [33] Función Flask que permite renderizar plantillas HTML para visualizarlas en el navegador.
- **url for:** [36] Función Flask que permite construir URLs para funciones o vistas específicas de una aplicación web Flask.
- **session:** [47] Objeto que almacena los datos del usuario que inició sesión en la aplicación.
- **send file:** [3] Función de Flask que envía el contenido de un archivo al cliente como respuesta a una solicitud HTTP.
- **UserMixin:** [50] Clase que proporciona unas propiedades de un modelo de usuario en una aplicación web.

flask login

[2] Librería Flask que se emplea para la gestión de las sesiones de usuarios y su autenticación.

- **LoginManager:** Clase que maneja la autenticación y la gestión de sesiones de usuario en una aplicación Flask.
- **login user:** Función que permite el inicio de sesión de los usuarios en una aplicación Flask.
- **current user:** Función que permite acceder al usuario que está actualmente autenticado en la sesión de una aplicación Flask.
- **login required:** Decorador que se utiliza para restringir el acceso a las vistas solo a usuarios que hayan iniciado sesión en una aplicación Flask.
- **logout user:** Función que permite cerrar la sesión de un usuario autenticado en una aplicación Flask.

werkzeug.security

[41] Módulo de la librería Werkzeug que ofrece distintas herramientas para la seguridad de las aplicaciones.

- **check password hash:** Función que permite la comprobación de si una contraseña proporcionada coincide con la hash de la contraseña almacenada.
- **generate password hash:** Función que genera un hash seguro a partir de una contraseña proporcionada.

werkzeug.utils

[42] Módulo que proporciona distintas funciones para manipular operaciones en una aplicación web.

- **secure filename:** Función que convierte un nombre de archivo en un nombre de archivo seguro, de forma que el archivo se puede almacenar de forma segura.

datetime

[24] Librería Python que permite manejar fechas y horas mediante diferentes clases.

- **datetime:** Clase que representa una fecha y hora.
- **timedelta:** Clase que representa una duración de tiempo, es decir, la diferencia entre dos instancias de date, time o datetime, con una resolución de microsegundos.

unicodecode

[27] Librería Python que contiene una función para convertir cadenas Unicode en cadenas

math

[25] Módulo que permite el uso de funciones matemáticas definidas en el estándar de C.

os

[26] Módulo que proporciona una forma de interactuar con el sistema operativo en el que se está ejecutando Python, permitiendo el uso de las funcionalidades del sistema operativo.

Psycopg2

[51] Librería Python que ofrece una interfaz que permite la conexión con bases de datos PostgreSQL y la realización de distintas operaciones.

Font Awesome

[23] Librería que proporciona iconos vectoriales y estilos css para su uso en aplicaciones web.

Dotenv

[11] Librería que permite cargar las variables de entorno desde el archivo .env en el programa.

Bootstrap

[14] Librería que facilita el desarrollo de aplicaciones web mediante estándares de diseño, por lo que permite la adaptabilidad a dispositivos móviles.

Animate CSS

Librería de animaciones para usar en aplicaciones web.

Aspectos relevantes del desarrollo del proyecto

En este apartado se pretende comentar las fases más interesantes y significativas del desarrollo del proyecto, incluyendo sus inicios, la metodología empleada, la creación, el despliegue de la aplicación y la resolución de los problemas que han surgido durante el desarrollo.

5.1. Inicio del proyecto

El tema del proyecto fue propuesto por José Ignacio Santos Martín y José Manel Galán Ordax, ya que tuvimos una primera reunión en la que me ofrecieron la posibilidad de realizar este trabajo de fin de grado, que sería tutelado por José Manuel Galán y Virginia Ahedo.

La primera propuesta fue la creación de una base de datos que se desplegara en GitHub Pages mediante JavaScript, que permitiera la subida de contenidos, su búsqueda y consulta, y la recolección de opiniones de usuarios a través de un sistema de rating a semejanza de Amazon o similares.

5.2. Primeros pasos

En la siguiente reunión que tuve con mis tutores me propusieron los primeros pasos a realizar.

Inicialmente me plantearon gestionar el proyecto mediante la metodología ágil. Para ello, se creó un repositorio en GitHub para llevar el seguimiento del proyecto a través de sprints de dos semanas de duración.

Durante este período se empezó con la búsqueda, selección y familiarización de las distintas herramientas que se iban a utilizar para el desarrollo del proyecto.

Algunas de las recomendaciones iniciales que me propusieron mis tutores fueron emplear un gestor de referencias como Zotero, y un gestor de base de datos como MySQL, MongoDB, MariaDB o PostgreSQL.

Respecto a la propuesta inicial de desarrollar el proyecto usando JavaScript, me decanté por usar como lenguaje de programación Python, ya que era el lenguaje aprendido durante los últimos años del grado, y por tanto iba a facilitar el desarrollo del proyecto puesto que JavaScript era un lenguaje de programación que no había usado por el momento.

Los frameworks que mis tutores me recomendaron fueron varios, como Flask o Django. Al ser Python el lenguaje de programación elegido para el desarrollo decidí usar Flask como framework. Aunque son parecidos se considera que es más conveniente Flask al facilitar el desarrollo de aplicaciones web en proyectos de menor escala como este.

5.3. Metodologías

Para la gestión del proyecto se ha empleado la metodología conocida como SCRUM, ya que fue la aprendida en el curso anterior en la asignatura de gestión de proyectos.

Esta metodología permite la gestión del proyecto mediante el establecimiento de tareas trabajando colaborativamente. Por ello, no se ha aplicado estrictamente esta metodología ya que el proyecto no se desarrolla en equipo, pero sí que se han seguido la mayoría de prácticas que incluye la metodología ágil.

Para la división del desarrollo del proyecto se han usado los sprints que han permitido la organización de las issues en períodos de dos semanas. En estos sprints se marcaban las tareas a realizar para alcanzar los objetivos establecidos, junto con sus story points correspondientes. Para ello, se ha utilizado la herramienta ZenHub, que gracias al tablero canvas que proporciona, se pueden gestionar los sprints, tareas, etiquetas y story points, entre otros.

Una vez finalizado el sprint se realizaba una reunión con los tutores en la que se realizaban una serie de tareas:

- **Revisión del sprint:** en esta tarea se revisaba si se habían cumplido todas las tareas asignadas al sprint, comprobando los avances obtenidos y los problemas que habían surgido durante este.
- **Planificación del siguiente sprint:** una vez revisado el sprint finalizado se planificaban los siguientes objetivos a alcanzar en el siguiente sprint, asignando las tareas y estimando sus correspondientes story points.

5.4. Formación

Para llevar a cabo el desarrollo del proyecto se ha realizado la investigación, y por tanto la familiarización de distintos conceptos y herramientas.

Flask

Para la creación de la aplicación se empleó Flask como framework, ya que se consideró que su instalación era sencilla y su uso también. Python fue elegido como el lenguaje de programación para el desarrollo, lo que permitió crear rápidamente un servidor y facilitó la implementación del patrón de diseño MVC.

Para la familiarización con su funcionamiento, se investigó y se consultaron tutoriales sobre cómo utilizarlo [21].

Bootstrap

Para el diseño de las interfaces, se utilizó Bootstrap, ya que uno de los objetivos de la aplicación era que fuera responsive y se visualizara correctamente en distintos dispositivos.

Para comprender su funcionamiento, se consultó la documentación oficial de Bootstrap[17], que proporciona información sobre cómo comenzar con la herramienta, además de mostrar ejemplos de código de diferentes componentes ya predefinidos. Esto facilitó considerablemente su modo de empleo.

5.5. Desarrollo del proyecto

Durante las primeras semanas se determinó el entorno de desarrollo del proyecto, eligiendo las distintas herramientas que se iban a utilizar para el desarrollo del proyecto.

Una vez determinado todo el entorno se comenzó con el desarrollo, empezando con la creación del entorno virtual y la creación y conexión con la base de datos.

Creación del frontend

Para crear el frontend, lo primero que se hizo fue desarrollar un prototipo de las interfaces de la aplicación, con el objetivo de diseñar una primera versión de cómo serían las interfaces. Este primer diseño, como se trataba de un prototipo, constó de interfaces sencillas e intuitivas para facilitar la comprensión y la navegación del usuario.

El desarrollo de las interfaces de la aplicación fue evolutivo. Para su diseño visual se utilizó CSS, ya que permitía personalizar de forma manual los templates HTML de forma sencilla y eficiente.

En un primer momento, se implementaron las interfaces de inicio de sesión y registro de nuevas cuentas de usuarios. Posteriormente, se diseñó la interfaz principal del menú de juegos para el usuario, que incluía una barra de búsqueda y filtros, así como tarjetas con la información de cada juego. Además, en la barra de navegación superior había un botón para solicitar el rol de profesor. Dependiendo del rol del usuario, la interfaz del menú de juegos cambiaba para mostrar funcionalidades específicas.

La interfaz para el menú de juegos del profesor, además de la barra de búsqueda y filtros, y las tarjetas con información de cada juego, incluía tres botones adicionales para añadir nuevos juegos, modificar juegos existentes y agregar archivos a cada juego. Esta interfaz se diferenciaba de la del usuario y la del administrador.

Por su parte, la interfaz del administrador también incluía un botón para la administración, lo que la diferenciaba de la del menú de juegos del usuario. En esta interfaz, el administrador podía realizar diferentes tareas de gestión.

Después, se trabajó en el desarrollo de interfaces específicas para la visualización y modificación de cada juego, con el fin de proporcionar una experiencia más detallada y personalizada para los usuarios.

Por último, se crearon interfaces para los usuarios con rol de administrador que les permitían gestionar usuarios, juegos y solicitudes en general. Además, se incluyeron las interfaces para la valoración de los juegos por parte de los usuarios y una interfaz adicional para proporcionar información general sobre la aplicación y sus funcionalidades.

Creación del backend

Para crear el backend, lo primero que se hizo fue establecer la conexión con la base de datos y crear las tablas necesarias, lo cual se llevó a cabo en el archivo `database.py`.

Se crearon cuatro tablas que forman parte de la base de datos:

1. **Tabla de usuarios:** almacena los datos de los usuarios registrados en la aplicación, incluyendo su nombre de usuario, nombre, apellido, institución, contraseña, rol y borrado.
2. **Tabla de juegos:** almacena los datos de los juegos registrados en la aplicación, incluyendo su nombre, descripción, idioma, enlace, puntuación, disciplina, naturaleza, precio, instrucciones para el jugador, instrucciones para el instructor, objetivo, espacio de control, objetivos principales y secundarios, estructura de sesiones, aspectos adicionales, la valoración del docente respecto al entretenimiento, aprendizaje, complejidad para el alumno, complejidad para el instructor, la URL del vídeo tutorial, los datos del usuario que añade o modifica el juego, los archivos de instrucciones y juego y borrado.
3. **Tabla de solicitudes:** almacena los datos de las solicitudes para obtener el rol de profesor, incluyendo el estado de la solicitud, la fecha y el usuario que la realizó.
4. **Tabla de valoraciones:** almacena los datos de las valoraciones que realizan los usuarios de los juegos, incluyendo la puntuación, el comentario, la fecha y los datos del usuario y juego al que se refiere la valoración.

Después, se comenzó a desarrollar en el archivo `app.py` los diferentes métodos GET y POST para manejar las solicitudes y respuestas, así como los archivos `juego.py`, `usuario.py`, `valoracion.py` y `solicitud.py` para manejar las operaciones de inserción y actualización de datos en la base de datos.

Para el desarrollo del login se utilizaron varios módulos de la librería Flask-Login, ya que esto facilitó toda la implementación. La clase `LoginManager` permitió gestionar la sesión del usuario en la aplicación, permitiendo a los usuarios iniciar sesión y mantenerse autenticados. Para ello, se creó una instancia de la clase `LoginManager` en la aplicación, lo que permitió usar sus métodos y atributos para configurar la autenticación y el inicio de sesión.

Además, en el caso de que un usuario intentara acceder a una ruta protegida sin estar autenticado, Flask redirigiría al usuario a la vista de inicio de

sesión, gracias a que se definió la propiedad login-view de la instancia de LoginManager creada anteriormente.

```
59 @login_manager.user_loader
60 def load_user(usuario_id):
61     return Usuario.get(usuario_id)
62
```

Figura 5.1: Captura de código

Con el módulo check-password-hash de la librería Werkzeug se permitía comprobar si la contraseña introducida para el inicio de sesión era correcta. En caso de que el usuario introdujera la contraseña de forma incorrecta durante 3 intentos consecutivos, se bloqueaba su cuenta durante 60 segundos para evitar ataques de fuerza bruta o intentos de acceso no autorizados. Este mecanismo de seguridad se implementó mediante la configuración de dos parámetros en el objeto app.config de Flask: MAX-LOGIN-ATTEMPTS, que establece el número máximo de intentos de inicio de sesión permitidos, y LOGIN-BLOCK-DURATION, que define la duración del bloqueo de la cuenta en segundos.

Una vez terminadas las funciones que se encargaban del registro de usuarios y de su inicio de sesión, se desarrollaron los diferentes menús de los juegos. Para ello, lo primero que se hizo fue la creación de las tarjetas que iban a contener información sobre los juegos del menú. Para una mejor visualización se desarrolló la paginación del menú, de forma que en cada página solo se iban a mostrar tres tarjetas de juego.

Después, en el archivo busqueda.py se desarrolló la funcionalidad de la barra de búsqueda y aplicación de filtros. Para asegurar el correcto funcionamiento de las búsquedas personalizadas y únicas de los usuarios, se empleó la combinación de la librería unicode y el método .lower() para garantizar que las búsquedas coincidieran con la información almacenada en la base de datos.

```
# Convertir contenido de busqueda e idioma en minúsculas y normalizarla
busqueda = unicode(busqueda.lower())
idiomaF = unicode(idiomaF.lower())
```

Figura 5.2: Captura de código

Una vez terminados los menús generales de cada rol, se fueron implementando las diferentes funcionalidades de cada rol.

Internacionalización

Para implementar la internacionalización en mi aplicación, consideré dos opciones: el uso de la librería Flask-Babel o la inclusión de todas las traducciones en distintos archivos JSON.

Finalmente, opté por la segunda opción ya que me resultó más sencilla de implementar y gestionar debido a que solo necesitaba dos opciones de idioma. Flask-Babel ofrece características avanzadas, como la detección automática del idioma y de los formatos de fecha y hora, la funcionalidad de pluralización para la correcta traducción de los textos, y la capacidad de generar formularios de selección de idioma de forma automática, entre otras, por lo que consideré que era una solución más adecuada para proyectos más complejos.

Para cada template HTML se creó un archivo JSON, lo que permitió mantener todas las traducciones de cada página en un solo archivo, facilitando su llamado en el código.



Figura 5.3: Archivos JSON

La aplicación ofrecía la posibilidad de visualizar el contenido en español o inglés, y se proporcionaron botones para que los usuarios seleccionen el idioma en la página de inicio. Si el usuario no seleccionaba un idioma, se utilizaba el español por defecto.



Figura 5.4: Botones para cambiar de idioma

Despliegue con Heroku

Una vez la aplicación estaba terminada en local, se empezó con el proceso de despliegue en Heroku. Tras el alta en la plataforma, se tuvo que realizar la conexión y configuración con el repositorio de GitHub que contenía el proyecto.

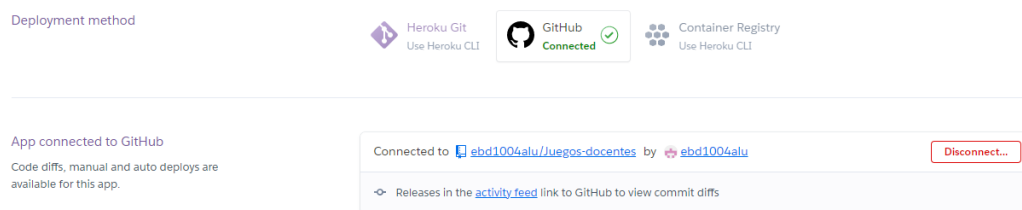


Figura 5.5: Conexión con repositorio de GitHub

Debido a que el despliegue de la base de datos en la nube era de pago, se tuvo que obtener el paquete de desarrollo para estudiantes de GitHub para así obtener beneficios por ser estudiante, lo cual iba a permitir obtener unos créditos para usar dentro de Heroku.

Para el acceso a la base de datos se incluyó en Heroku un add-on de Heroku-PostgreSQL y así poder desplegar la base de datos en la nube.

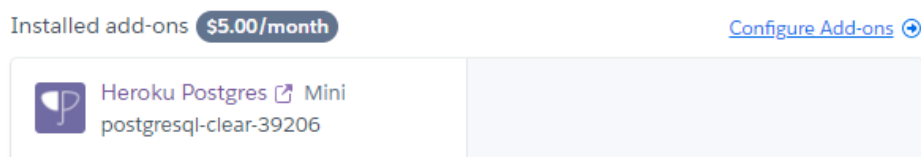


Figura 5.6: Add-on Heroku Postgres

Se obtuvieron las credenciales de Heroku-PostgreSQL y se modificaron en la aplicación los anteriores credenciales que estaban establecidos en la base de datos local.

Una vez desplegado el proyecto en Heroku, era posible utilizar la aplicación de manera web a través del enlace que facilitaba la plataforma.

5.6. Resolución de problemas

Durante el desarrollo del proyecto, surgieron varios problemas que fueron resueltos con éxito.

Problemas con ZenHub

El primer problema que surgió fue con la plataforma de gestión de proyectos ZenHub. Al principio, utilicé esta herramienta para la gestión del proyecto, pero después de un mes recibí una notificación informándome de que la versión de prueba gratuita había finalizado.

Busqué alternativas que proporcionasen funcionalidades similares a las de ZenHub y finalmente opté por Zube, un sistema de gestión de proyectos gratuito integrado en GitHub que ofrece la planificación de sprints, la creación de tareas, la estimación de story points y la generación de gráficos e informes estadísticos de manera similar a ZenHub.

Problemas con la visualización de las interfaces

Una vez que se terminaron todos los templates junto con sus diseños, surgió un inconveniente con los estilos diseñados. El problema radicaba en que estos estilos no eran responsive y solo se veían correctamente cuando la pantalla estaba maximizada. Si el usuario reducía la ventana del navegador, todos los elementos se sobreponían unos encima de otros, lo que impedía la correcta visualización y usabilidad de la aplicación. Esto también significaba que la aplicación no era adecuada para su ejecución en distintos tipos de dispositivos.

Por ello, se empleó la librería de Bootstrap para lograr que la aplicación fuera responsive y que todos los elementos se visualizaran correctamente, ya fuera reduciendo la ventana del navegador, haciendo zoom o en otros tipos de dispositivos.

Trabajos relacionados

En este apartado se presentan varios proyectos que guardan relación con el presente.

Aunque el objetivo específico de este proyecto ha dificultado la identificación de proyectos con el mismo propósito, se han encontrado TFGs de otras universidades que emplean herramientas similares a las que se utilizan en este.

Además, los tutores me proporcionaron un repositorio de juegos para el aula que resultaba muy similar al objetivo final de este proyecto.

Estos trabajos relacionados han facilitado la orientación del desarrollo y la aplicación de distintas herramientas y técnicas en este.

6.1. The Gamification Repository

[38] Se trata de una aplicación para la gamificación en "Dirección de Proyectos" desarrollado por Lorenzo Borreguero Cortón y dirigido por Adolfo López-Paredes, de la Universidad de Valladolid.

Este repositorio recopila juegos en el aula relevantes para la docencia de materias de áreas de la Ingeniería de Organización y Organización de Empresas.

La aplicación web [1] proporciona al usuario tres secciones diferentes:

- **Gamificación:** proporciona al usuario información sobre el concepto de gamificación.
- **Buscador:** permite al usuario buscar los juegos de interés que se encuentran en la base de datos. Estos juegos se pueden buscar por el nombre, la asignatura relacionada o por el idioma.

Una vez buscado el juego por las opciones disponibles se muestran los distintos juegos que cumplen el criterio, proporcionando la url donde se obtiene el juego, su descripción, sus instrucciones y una descripción.

- **INSISOC Official Web:** redirecciona al usuario a la web oficial de INSISOC Grupo de Investigación Reconocido (GIR) de la Universidad de Valladolid, dedicado a la Ingeniería de los Sistemas Sociales.

6.2. Web corporativa e intranet para una consultora tecnológica

[44] Se trata de un trabajo de fin de grado realizado por Pablo Sánchez Ponce en junio de 2020.

El objetivo de su proyecto consiste en el desarrollo de una aplicación web para la exposición de los servicios que proporciona una empresa, junto con la venta online de diversos cursos de formación para los clientes.

Tanto mi proyecto como el trabajo de grado mencionado anteriormente tienen como objetivo la creación de una plataforma web que permita a los usuarios buscar, acceder y utilizar ciertos recursos. Ambos proyectos involucran la gestión de usuarios y recursos, con diferentes roles y permisos.

En su caso las tecnologías usadas para el desarrollo coinciden en el lenguaje de programación Python y el framework empleado Flask. Aunque, para el almacenamiento de datos se emplea el sistema MySQL a diferencia del mío en el que se emplea PostgreSQL.

6.3. Desarrollo de Aplicación Web para Administración de Sensores Inalámbricos Mediante el Estándar BACnet

[45] Se trata de un trabajo de fin de grado realizado por Ciprian Ilut Pop en septiembre de 2020.

El objetivo de su proyecto consiste en el desarrollo de una aplicación web para la gestión de sensores a través del estándar BACnet.

En cuanto a las similitudes en objetivos, ambos proyectos tienen como objetivo el desarrollo de una aplicación web, y utilizan tecnologías similares,

como Flask y PostgreSQL. Además, ambos proyectos tienen como objetivo proporcionar información y funcionalidades a los usuarios.

Aunque el objetivo final de este proyecto es diferente al objetivo del propio, varias de las herramientas y técnicas que se emplean son las mismas. En este caso, para el almacenamiento de datos también se emplea el sistema de PostgreSQL. Para la implementación de la aplicación web se ha usado Flask como framework y Python como lenguaje de programación.

6.4. Desarrollo de una aplicación Web RESTfull basada en Spring y Angular

[10] Se trata de un trabajo de fin de grado realizado por Jorge Zapata Alburquerque.

El objetivo de su proyecto consiste en la creación de una aplicación web para la administración de currículums en una empresa. Su objetivo principal es mantener un registro actualizado de las habilidades de los empleados.

Los objetivos y funcionalidades son diferentes en cada proyecto, ya que mi aplicación se enfoca en la búsqueda y filtrado de juegos docentes, permitiendo a los usuarios puntuar los juegos, y su objetivo principal es mantener un registro de las habilidades de los empleados, enfocándose en la gestión de datos de la empresa y no en la interacción de los usuarios con la aplicación.

En cuanto a las similitudes, ambos proyectos utilizan PostgreSQL para la gestión de la base de datos. También se utilizan tecnologías web para el desarrollo de la aplicación, aunque mi proyecto utiliza Python con Flask y el trabajo de fin de grado de Zapata se basa en Spring y Angular.

6.5. RACO

[19] RACO (Revistes Catalanes amb Accés Obert) es una aplicación web que actúa como un repositorio que permite acceder y consultar artículos completos de revistas científicas, culturales y eruditas de Cataluña.

Los objetivos de este repositorio son la gestión y publicación de revistas académicas en línea. Por lo tanto, mi proyecto se asemeja a RACO en el sentido de que ambos son repositorios que buscan facilitar la búsqueda y gestión de contenido para los usuarios en sus respectivos campos.

Conclusiones y Líneas de trabajo futuras

En este apartado, se expondrán las conclusiones obtenidas tras finalizar el desarrollo del proyecto, así como las posibles líneas de trabajo que podrían llevarse a cabo en el futuro.

7.1. Conclusiones

Una vez finalizado el proyecto, se puede considerar que se han cumplido satisfactoriamente los objetivos generales, técnicos y personales establecidos inicialmente. La aplicación desarrollada ha logrado ser una herramienta efectiva y accesible para el aprendizaje a través de juegos docentes, cumpliendo con los requisitos establecidos.

Además, se han añadido funcionalidades adicionales que enriquecen la experiencia de usuario y amplían las capacidades de la aplicación. Estas mejoras incluyen la implementación de filtros de búsqueda y un sistema de gestión completo para administrar juegos, usuarios y solicitudes.

Durante el desarrollo del proyecto, se han adquirido y fortalecido conocimientos en diversas herramientas y lenguajes. Se ha mejorado la comprensión y habilidades en HTML, CSS y se ha utilizado el framework Bootstrap para agilizar el desarrollo y lograr diseños responsive.

En cuanto a Flask, se ha profundizado en su uso y se ha aprovechado su versatilidad para el desarrollo de aplicaciones web. Se ha trabajado en la definición de rutas, diseño de vistas y utilización de plantillas para generar contenido dinámico. También se ha implementado formularios y validaciones para mejorar la interacción con el usuario.

En Python, se han ampliado los conocimientos y habilidades, especialmente en el contexto del desarrollo web. Se han utilizado diversas funcionalidades y bibliotecas para la manipulación de datos, interacción con la base de datos PostgreSQL y la implementación de la lógica de negocio de la aplicación.

Además, se ha adquirido experiencia en la metodología ágil Scrum y se han utilizado herramientas como Zube para la planificación y gestión eficiente de tareas.

En resumen, el proyecto ha permitido ampliar y aplicar los conocimientos adquiridos durante el grado, desarrollando una aplicación web educativa con éxito.

7.2. Líneas de trabajo futuras

Algunas ideas para futuras líneas de desarrollo de la aplicación web pueden ser:

- Incorporar una función de mensajería interna para que los usuarios puedan comunicarse entre sí, de forma que los docentes puedan compartir ideas y conocimientos sobre los distintos juegos docentes.
- Mejorar la experiencia de búsqueda de juegos. Agregando filtros adicionales como el tema del juego o su nivel de dificultad, de forma que los usuarios puedan encontrar juegos más específicos.
- Implementar una página personalizada para los profesores donde puedan ver sus juegos favoritos o los juegos que han creado.
- Añadir una sección de informes para que los administradores puedan ver distintas estadísticas sobre el uso de la plataforma. Incluyendo información como la cantidad de usuarios activos, la cantidad de juegos subidos en un período determinado de tiempo y las solicitudes de rol aceptadas o rechazadas.
- Integrar la plataforma con las redes sociales, permitiendo que los usuarios puedan compartir los juegos docentes.
- Incluir un foro de discusión para que los usuarios puedan hacer preguntas y compartir ideas sobre cómo utilizar los juegos educativos en el aula.

Bibliografía

- [1] <http://wordpress.insisoc.org/gamification/index.php/searcher/>.
- [2] Flask-Login — Flask-Login 0.7.0 documentation. <https://flask-login.readthedocs.io/en/latest/>.
- [3] flask.send_file — Flask API. https://tedboy.github.io/flask/generated/flask.send_file.html.
- [4] Framework: ¿Qué es y qué tipos hay? <https://open-bootcamp.com/aprender-programar/que-es-un-framework>.
- [5] ¿Qué es Python? - Explicación del lenguaje Python - AWS. <https://aws.amazon.com/es/what-is/python/>.
- [6] Diferencias entre el Método GET y POST. <https://pc-solucion.es/tecnologia/diferencias-entre-el-metodo-get-y-post/>, June 2018.
- [7] GET vs. POST: los dos métodos de petición HTTP más conocidos cara a cara. <https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/get-vs-post/>, August 2020.
- [8] Protocolo de transferencia de hipertexto. https://es.wikipedia.org/w/index.php?title=Protocolo_de_transferencia_de_hipertexto&oldid=150180985, March 2023.
- [9] EDUCACIÓN 3.0. ¿Qué es la gamificación y cuáles son sus objetivos? | EDUCACIÓN 3.0. <https://www.educaciontrespuntocero.com/noticias/gamificacion-que-es-objetivos/>, August 2019. Section: Noticias.

- [10] Jorge Zapata Alburquerque. Desarrollo de una aplicación web rest-full basada en spring y angular. <https://repositorio.upct.es/xmlui/bitstream/handle/10317/9055/tfg-zap-apl.pdf?sequence=1&isAllowed=y>, January 2016.
- [11] Miguel Angel Alvarez. Variables de entorno en PHP con archivos .env. <https://desarrolloweb.com/articulos/variables-entorno-php-env.html>, February 2020.
- [12] Arimetrics. Qué es Backend - Definición, significado y ejemplos, January 2020.
- [13] Santiago Borges. ¿Qué es PostgreSQL? - Para qué sirve, Características e Instalación. <https://blog.infranetworking.com/servidor-postgresql/>, November 2019.
- [14] Ernesto G. Bustamante. Qué es Bootstrap: Significado y Definición. <https://aulacm.com/que-es/bootstrap-significado-definicion/>.
- [15] Luciano Castillo. Introducción — Conociendo GitHub 0.1 documentation. <https://conociendogithub.readthedocs.io/en/latest/data/introduccion/>, 2012.
- [16] Codeicus. Arquitectura MVC - Parte 1. <https://medium.com/somos-codeicus/arquitectura-mvc-conceptos-b%C3%A1sicos-481062755df9>, April 2020.
- [17] contributors Jacob Thornton and Bootstrap Mark Otto. Get started with Bootstrap. <https://getbootstrap.com/docs/5.3/getting-started/introduction/>.
- [18] Mozilla Corporation's. Métodos de petición HTTP - HTTP | MDN. <https://developer.mozilla.org/es/docs/Web/HTTP/Methods>, November 2022.
- [19] CSUC. RACO – RACO. <https://raco.cat/raco/index.php/es/inicio/>.
- [20] Diegocamachop. Qué es GitHub y cómo empezar a usarlo de forma correcta. <https://platzi.com/blog/que-es-github-como-funciona/>, 2021.
- [21] DigitalOcean. Cómo crear una aplicación Web usando Flask en Python 3 | DigitalOcean. <https://www.digitalocean.com/community/tutorials/how-to-make-a-web-application-using-flask-in-python-3-es>.

- [22] GitHub Docs. Comenzar con GitHub Desktop. <https://ghdocs-prod.azurewebsites.net/es/desktop/installing-and-configuring-github-desktop/overview/getting-started-with-github-desktop>.
- [23] eduk2. Font Awesome ¿Qué es y cómo se usa? <https://www.aquihaydominios.com/blog/font-awesom-que-es-y-como-se-usa/>, May 2014.
- [24] Python Software Foundation. datetime — Tipos básicos de fecha y hora. <https://docs.python.org/3/library/datetime.html>, 2021.
- [25] Python Software Foundation. math — Funciones matemáticas — documentación de Python - 3.10.11. <https://docs.python.org/es/3.10/library/math.html>, 2021.
- [26] Python Software Foundation. os — Interfaces misceláneas del sistema operativo — documentación de Python - 3.10.11. <https://docs.python.org/es/3.10/library/os.html>, 2021.
- [27] Python Software Foundation. Unidecode: ASCII transliterations of Unicode text. <https://pypi.org/project/Unidecode/>, 2023.
- [28] Karina Fuerte. ¿Qué son los Serious Games? <https://observatorio.tec.mx/edu-news/que-son-los-serious-games/>, September 2018.
- [29] Karina Fuerte. ¿Qué son los Serious Games? <https://observatorio.tec.mx/edu-news/que-son-los-serious-games/>, September 2018.
- [30] Virginia Gaitán. Gamificación: el aprendizaje divertido | educativa. <https://www.educativa.com/blog-articulos/gamificacion-el-aprendizaje-divertido/>.
- [31] Alex Galera. Zube, un sistema de gestión de proyectos para programadores. <https://www.whatsnew.com/2015/10/04/zube-un-sistema-de-gestion-de-proyectos-para-programadores/>, October 2015.
- [32] Germán Garcés. Ventajas de usar Figma como herramienta de diseño UI - Blog de Hiberus Tecnología. <https://www.hiberus.com/crecemos-contigo/ventajas-de-usar-figma-como-herramienta-de-diseno-ui/>, January 2022.
- [33] grokkeepcoding. ¿Qué es render template de Flask? | KeepCoding Bootcamps. <https://keepcoding.io/blog/que-es-render-template-de-flask/>, August 2022.

- [34] Andrés Herrera. Qué es el eLearning y qué modalidades hay. <https://www.innovacionycualificacion.com/plataforma-elearning/que-es-elearning/>, July 2021.
- [35] Javier Sáez Hurtado. Cómo funciona la Metodología Scrum: Qué es y cómo utilizarla. <https://www.iebschool.com/blog/metodologia-scrum-agile-scrum/>, December 2021.
- [36] javatpoint. Flask URL Building - Javatpoint. <https://www.javatpoint.com/flask-url-building>, 2021.
- [37] Ana García Jiménez. Serious games y Gamificación, ¿en qué se diferencian? <https://blog.gestazion.com/serious-games-y-gamificaci3n>, November 2017.
- [38] JMgalan. Repositorio de juegos para el aula – CIO docencia. Innovación Docente. <https://epsapps.udg.edu/cioblog/index.php/2019/12/13/repositorio-de-juegos-para-el-aula/>, December 2019.
- [39] Maldeadora. Qué es Frontend y Backend: diferencias y características. <https://platzi.com/blog/que-es-frontend-y-backend/>, 2018.
- [40] José Domingo Muñoz. Qué es Flask y ventajas que ofrece. <https://openwebinars.net/blog/que-es-flask/>, November 2017.
- [41] Pallets. Utilities — Werkzeug Documentation (1.0.x). <https://werkzeug.palletsprojects.com/en/1.0.x/utils/>, 2007.
- [42] Pallets. Utilities — Werkzeug Documentation (2.3.x). <https://werkzeug.palletsprojects.com/en/2.3.x/utils/>, 2007.
- [43] Juan Diego Polo. zenhub, plataforma de gestión de proyectos integrada en Github, ahora gratis para estudiantes. <https://www.whatsnew.com/2015/09/24/zenhub-plataforma-de-gestion-de-proyectos-integrada-en-github-ahora-gratis-para-estudiantes/>, September 2015.
- [44] Pablo Sánchez Ponce. Web corporativa e intranet para una consultora tecnológica. <https://openaccess.uoc.edu/bitstream/10609/116207/7/pabloponceTFG0620memoria.pdf>, June 2020.
- [45] Ciprian Ilut Pop. Desarrollo de Aplicación Web para Administración de Sensores Inalámbricos Mediante el Estándar BACnet. https://oa.upm.es/64755/1/TFG_CIPRIAN_ILUT_POP.pdf, September 2020.

- [46] pythonbasics. Flask redirect and errors - Python Tutorial. <https://pythonbasics.org/flask-redirect-and-errors/>, 2021.
- [47] pythonbasics. Session data in Python Flask - Python Tutorial. <https://pythonbasics.org/flask-sessions/>, 2021.
- [48] Alberto Albarracín Ramírez. Guías BibUpo: LaTeX: redacción de documentos científicos: Trabajando con Overleaf. <https://guiasbib.upo.es/latex/Trabajando-con-Overleaf>, January 2023.
- [49] Txema Rodríguez. GitBook, crea documentación técnica y libros usando Markdown y Git/Github de forma flexible. <https://www.genbeta.com/desarrollo/gitbook-crea-documentacion-tecnica-y-libros-usando-markdown-y-git-github-de-forma-flexible>, April 2014. Section: desarrollo.
- [50] Daniel Seyi. "What is the UserMixin in Flask?". <https://stackoverflow.com/a/63232145>, August 2020.
- [51] Patricio Soriano. Psycopg2 para acceder a una base de datos PostGIS con Python. <https://geoinnova.org/blog-territorio/libreria-psycopg2-para-acceder-a-una-base-de-datos-postgis-con-python/>, February 2023.
- [52] Santander Universidades. ¿Qué es Python? | Blog Becas Santander. <https://www.becas-santander.com/es/blog/python-que-es.html>, April 2021.
- [53] Santander Universidades. Ventajas de la gamificación en el aula: qué herramientas utilizar y cómo. <https://www.becas-santander.com/es/blog/gamificacion-en-el-aula.html>, March 2022.
- [54] Santander Universidades. ¿Qué es el e-learning y cómo está transformando la educación? <https://www.becas-santander.com/es/blog/e-learning.html>, March 2022.
- [55] Manual Web. Request Flask | Manual Web. <https://www.manualweb.net/flask/request-flask/>, 2023.
- [56] Wikipedia. Figma. <https://es.wikipedia.org/w/index.php?title=Figma&oldid=147926756>, December 2022.
- [57] Wikipedia. Scrum (desarrollo de software). [https://es.wikipedia.org/w/index.php?title=Scrum_\(desarrollo_de_software\)&oldid=148233785](https://es.wikipedia.org/w/index.php?title=Scrum_(desarrollo_de_software)&oldid=148233785), December 2022.

- [58] Wikipedia. Diagrams.net. <https://en.wikipedia.org/w/index.php?title=Diagrams.net&oldid=1142089111>, February 2023.
- [59] Wikipedia. Heroku. <https://es.wikipedia.org/w/index.php?title=Heroku&oldid=151170100>, May 2023.
- [60] Wikipedia. LaTeX. <https://es.wikipedia.org/w/index.php?title=LaTeX&oldid=149680576>, March 2023.
- [61] Wikipedia. Zotero. <https://es.wikipedia.org/w/index.php?title=Zotero&oldid=148468399>, January 2023.