



UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE CUENCA
CARRERA DE COMPUTACIÓN

DESARROLLO E IMPLEMENTACIÓN DE UN MÓDULO COMO REPOSITORIO
DE RECURSOS COLABORATIVOS MEDIANTE SPRING, ANGULAR Y
POSTGRESQL PARA LA PLATAFORMA DE EDUTAINMENT DEL GRUPO DE
GIHP4C DE LA UNIVERSIDAD POLITÉCNICA SALESIANA

Trabajo de titulación previo a la obtención del
título de Ingeniero en Ciencias de la Computación

AUTORES:

MICHAEL ANDRES ALVAREZ VALDEZ
JHON SANTIAGO LLIVICOTA LEÓN

TUTOR:

ING. OMAR GUSTAVO BRAVO QUEZADA, PHD.

Cuenca - Ecuador

2024

**CERTIFICADO DE RESPONSABILIDAD
Y AUTORÍA DEL TRABAJO DE TITULACIÓN**

Nosotros, Michael Andres Alvarez Valdez con documento de identificación N° 0150874915 y Jhon Santiago Llivicota León con documento de identificación N° 0150628949; manifestamos que:

Somos los autores y responsables del presente trabajo; y, autorizamos a que sin fines de lucro la Universidad Politécnica Salesiana pueda usar, difundir, reproducir o publicar de manera total o parcial el presente trabajo de titulación.

Cuenca, 29 de enero del 2024

Atentamente,

Michael Andres Alvarez Valdez
0150874915

Jhon Santiago Llivicota León
0150628949

CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE TITULACIÓN A LA UNIVERSIDAD POLITÉCNICA SALESIANA

Nosotros, Michael Andres Alvarez Valdez con documento de identificación N° 0150874915 y Jhon Santiago Llivicota León con documento de identificación N° 0150628949, expresamos nuestra voluntad y por medio del presente documento cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del Proyecto técnico: “Desarrollo e implementación de un módulo como repositorio de recursos colaborativos mediante Spring, Angular y PostgreSQL para la plataforma de Edutainment del grupo de GIHP4C de la Universidad Politécnica Salesiana”, el cual ha sido desarrollado para optar por el título de: Ingeniero en Ciencias de la Computación, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En concordancia con lo manifestado, suscribimos este documento en el momento que hacemos la entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.

Cuenca, 29 de enero del 2024

Atentamente,

Michael Andres Alvarez Valdez

0150874915

Jhon Santiago Llivicota León

0150628949

CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN

Yo, Omar Gustavo Bravo Quezada con documento de identificación N° 0102777257, docente de la Universidad Politécnica Salesiana, declaro que bajo mi tutoría fue desarrollado el trabajo de titulación: DESARROLLO E IMPLEMENTACIÓN DE UN MÓDULO COMO REPOSITORIO DE RECURSOS COLABORATIVOS MEDIANTE SPRING, ANGULAR Y POSTGRESQL PARA LA PLATAFORMA DE EDUTAINMENT DEL GRUPO DE GIHP4C DE LA UNIVERSIDAD POLITÉCNICA SALESIANA, realizado por Michael Andres Alvarez Valdez con documento de identificación N° 0150874915 y por Jhon Santiago Llivicota León con documento de identificación N° 0150628949, obteniendo como resultado final el trabajo de titulación bajo la opción Proyecto técnico que cumple con todos los requisitos determinados por la Universidad Politécnica Salesiana.

Cuenca, 29 de enero del 2024

Atentamente,

Ing. Omar Gustavo Bravo Quezada, PhD.

0102777257

DEDICATORIA

En primer lugar, con todo mi amor y profunda gratitud, dedico este trabajo a mi querida madre, Ximena León. Su amor incondicional, sabiduría, y apoyo constante han sido el faro que ha iluminado mi camino en los momentos más oscuros. Su ejemplo de fortaleza, integridad y perseverancia ha sido la mayor fuente de inspiración en mi vida. Madre, cada página de este trabajo lleva impresa tu dedicación, tu sacrificio y tu amor. Gracias por enseñarme el valor de la educación, por tus consejos sabios y por creer en mí incluso cuando yo mismo dudaba.

Extiendo esta dedicatoria a mi abuela, María Tenempaguay, cuya presencia en mi vida ha sido un regalo invaluable. Abuela, tu amor, tus historias y tu sabiduría ancestral han enriquecido mi mundo de maneras que las palabras no pueden expresar. Eres el corazón y el alma de nuestra familia, y tu fortaleza y coraje son un testimonio de la poderosa linaje del cual somos parte. Gracias por tus oraciones, tus enseñanzas y por ser siempre una fuente de amor inagotable.

Jhon Santiago Llivicota León

DEDICATORIA

Este trabajo es el fruto de un viaje lleno de aprendizaje y crecimiento, y lo dedico con todo mi corazón a quienes han sido pilares fundamentales en esta aventura. A mi querida madre Jenny Valdez, cuya fuerza, sabiduría y amor incondicional han sido mi faro en la oscuridad, guiándome hacia este logro con su ejemplo inspirador. A mis compañeros, quienes han sido más que colegas en este viaje académico: han sido amigos, cómplices en cada desafío y celebración, y una fuente inagotable de apoyo y camaradería. Y a mis profesores, cuya dedicación y pasión por la enseñanza han encendido en mí una llama de curiosidad y me han equipado con las herramientas para explorar nuevos horizontes. A todos ustedes, mi más profundo agradecimiento y la promesa de que este logro es también suyo.

Michael Andres Alvarez Valdez

AGRADECIMIENTO

Queremos expresar nuestro más profundo agradecimiento a Omar Gustavo Bravo Quezada, quien ha sido más que un tutor para este trabajo de titulación; ha sido un mentor, un guía y un inspirador incansable. Su conocimiento experto, su paciencia y su habilidad para motivar y desafiar nuestro pensamiento han sido fundamentales en cada etapa de este proceso.

Gracias por creer en nosotros, por sus valiosos consejos y por dedicar su tiempo y energía para ayudarnos a alcanzar este importante hito. Este trabajo no solo lleva nuestra firma, sino también la impronta de su dedicación y excelencia. Su apoyo ha sido un regalo invaluable en nuestro desarrollo académico y profesional.

Michael Andres Alvarez Valdez

Jhon Santiago Llivicota León

RESUMEN

Este proyecto desarrolla un módulo para la plataforma Edutainment, permitiendo la gestión colaborativa de recursos digitales para proyectos de aprendizaje y turismo en Ecuador. Utiliza una arquitectura de microservicios con Spring, Angular, y PostgreSQL, promoviendo la reutilización de recursos y simplificando la colaboración en proyectos educativos y de entretenimiento, con el objetivo de enriquecer la enseñanza y el aprendizaje a través de la innovación tecnológica.

Palabras Claves: repositorio colaborativo, gamificación, realidad virtual, realidad aumentada, Edutainment, recursos digitales.

ABSTRACT

This project develops a module for the Edutainment platform, enabling collaborative management of digital resources for learning and tourism projects in Ecuador. It uses a microservices architecture with Spring, Angular, and PostgreSQL, promoting resource reuse and simplifying collaboration in educational and entertainment projects, with the goal of enriching teaching and learning through technological innovation.

Keywords: collaborative repository, gamification, virtual reality, augmented reality, Edutainment, digital resources.

ÍNDICE DE CONTENIDO

1. Introducción	14
2. Problema	16
2.1. Antecedentes	16
2.2. Importancia y Alcances	16
2.3. Delimitación	17
3. Objetivos	18
3.1. Objetivo General	18
3.2. Objetivos Específicos	18
4. Revisión de la literatura	19
4.1. Gamificación	19
4.2. Realidad Virtual/Aumentada	19
4.3. Arquitectura de Microservicios	20
4.4. Gestión de Repositorios Digitales	21
4.5. Tecnologías de Desarrollo Web	21
4.5.1. Angular	22
4.5.2. Spring	22
4.5.3. PostgreSQL	23
4.5.4. Docker	23
4.5.5. Eureka Netflix	24
4.5.6. Zipkin	25
4.6. Metodologías de Desarrollo Ágil	26
5. Marco Metodológico	27
5.1. Metodología SCRUM	27
5.1.1. Sprint 1	28
5.1.1.1. Comprender necesidades y expectativas.	28
5.1.1.2. Examinar la documentación previa de Edutainment.	28

5.1.1.3. Investigar mejores prácticas en gestión de contenido en plataformas similares.	29
5.1.1.4. Investigar prácticas de diseño de interfaz	29
5.1.1.5. Definir la arquitectura del módulo de repositorio	29
5.1.2. Sprint 2	31
5.1.2.1. Diseñar el esquema de la base de datos	31
5.1.2.2. Desarrollar los servicios seguros de carga y descarga de scripts y objetos	32
5.1.2.3. Implementar una interfaz de usuario para acceder y descargar contenido	36
5.1.2.4. Desarrollar funciones de visualización de scripts y objetos.	37
5.1.2.5. Implementar un sólido control de acceso con roles y permisos.	38
5.1.3. Sprint 3	41
5.1.3.1. Crear una jerarquía de etiquetas.	41
5.1.3.2. Agregar metadatos a scripts y objetos.	41
5.1.3.3. Implementar una función de búsqueda avanzada.	42
5.1.4. Sprint 4	43
5.1.4.1. Elaborar documentación detallada de las funciones del módulo	43
5.1.4.2. Realizar encuestas a usuarios sobre la usabilidad del módulo.	43
5.1.4.3. Lanzar el módulo para su uso en proyectos colaborativos	44
6. Resultados	45
7. Cronograma	54
8. Presupuesto	58
9. Conclusiones	59
10. Recomendaciones	60
Referencias Bibliográficas	61
Anexos	63

ÍNDICE DE FIGURAS

<i>Figura 1: Comunicación entre las diversas capas de la arquitectura</i>	30
<i>Figura 2: Esquema de la Base de Datos del Módulo de Repositorio de Recursos</i>	32
<i>Figura 3: Inicialización del Servicio de Descarga y Método de Carga de Recursos</i>	33
<i>Figura 4: Método de Compresión en el Servicio de Descarga</i>	33
<i>Figura 5: Inicialización del Servicio de Carga y Configuración de Almacenamiento</i>	34
<i>Figura 6: Método de Almacenamiento de Archivos del Servicio de Carga</i>	35
<i>Figura 7: Validación y Formateo en el Servicio de Carga</i>	36
<i>Figura 8: Prototipo de la Interfaz de Usuario para el Módulo de Repositorio de Recursos</i>	37
<i>Figura 9: Prototipo de la Interfaz de Visualización de Scripts y Objetos</i>	38
<i>Figura 10: Diagrama de Flujo del Control de Acceso Basado en Roles</i>	39
<i>Figura 11: Página principal del repositorio</i>	45
<i>Figura 12: Página del CRUD de las etiquetas</i>	46
<i>Figura 13: Página del CRUD de los repositorios</i>	46
<i>Figura 14: Página individual de cada repositorio</i>	47
<i>Figura 15: Página crear un recurso</i>	48
<i>Figura 16: Página visualizar los recursos</i>	48
<i>Figura 17: Página individual de cada recurso</i>	49
<i>Figura 18: Página para editar un recurso</i>	50
<i>Figura 19: Página para crear un juego</i>	51
<i>Figura 20: Página para ver todos los juegos</i>	52
<i>Figura 21: Página para ver los detalles del juego</i>	52
<i>Figura 22: Página para editar un juego</i>	53

ÍNDICE DE TABLAS

<i>Tabla 1: Actividades Realizadas en el Objetivo Especifico 1</i>	54
<i>Tabla 2: Actividades Realizadas en el Objetivo Especifico 2</i>	54
<i>Tabla 3: Actividades Realizadas en el Objetivo Especifico 3</i>	54
<i>Tabla 4: Actividades Realizadas en el Objetivo Especifico 4</i>	55
<i>Tabla 5: Actividades Realizadas en el Objetivo Especifico 5</i>	55
<i>Tabla 6: Actividades Realizadas en el Objetivo Especifico 6</i>	56
<i>Tabla 7: Actividades Realizadas en el Objetivo Especifico 7</i>	56
<i>Tabla 8: Cronograma por Cada responsable</i>	57
<i>Tabla 9: Presupuesto del Proyecto Técnico</i>	58

1. Introducción

En un mundo donde la tecnología evoluciona rápidamente, el ámbito educativo no es la excepción. La necesidad de herramientas que faciliten la interacción y colaboración entre estudiantes de distintas disciplinas es cada vez más imperante. Ante este desafío, el presente proyecto técnico, auspiciado por el grupo GIHP4C de la Universidad Politécnica Salesiana, se enfoca en el desarrollo de un módulo innovador para la plataforma Edutainment. Este módulo no solo actúa como un repositorio de recursos colaborativos sino también como un puente entre las competencias técnicas y creativas de los estudiantes, específicamente aquellos de las carreras de Computación y Diseño Multimedia.

El proyecto surge de la necesidad de abordar dos problemas fundamentales en el entorno educativo actual: la falta de una plataforma unificada que permita la colaboración interdisciplinaria eficaz y la ausencia de un sistema centralizado para la gestión y reutilización de recursos educativos digitales. Estos desafíos son particularmente notables en proyectos que involucran gamificación, realidad virtual/aumentada, donde la integración de habilidades técnicas y creativas es crucial.

Los estudiantes de Computación, con su sólida formación en programación, a menudo se ven limitados por su falta de experiencia en diseño gráfico y modelado 2D y 3D, habilidades esenciales para la creación de contenido interactivo y visualmente atractivo. Por otro lado, los estudiantes de Diseño Multimedia, aunque dotados de una creatividad excepcional y habilidades de diseño, enfrentan barreras en la implementación técnica de sus ideas. Este módulo propuesto se diseña para superar estas barreras, facilitando una colaboración efectiva y el intercambio de recursos entre ambos grupos. Además, busca mejorar la eficiencia en la evaluación y revisión de proyectos tanto por parte de los docentes como de la comunidad estudiantil.

En cuanto a la implementación técnica, el proyecto se basa en una arquitectura de microservicios utilizando tecnologías avanzadas como Spring y Angular, y emplea PostgreSQL para la gestión de la base de datos. El enfoque del proyecto se centra en la optimización y seguridad de los recursos, garantizando una experiencia de usuario intuitiva y eficiente. Además, se adopta una metodología de desarrollo ágil que permite una adaptación flexible a las cambiantes necesidades y prioridades del entorno educativo contemporáneo. Con todo esto, el proyecto no solo atiende a las demandas actuales de las instituciones educativas, sino que

también se posiciona como un referente en la integración de tecnología y creatividad para el enriquecimiento del proceso de enseñanza-aprendizaje.

Adicionalmente, este proyecto técnico busca trascender los límites tradicionales del aula, creando una plataforma que no sólo sea un recurso para estudiantes y docentes, sino que también sirva como un modelo para futuras iniciativas educativas y de desarrollo de software. Al alinear las necesidades educativas con las tendencias tecnológicas actuales, este módulo se configura como una herramienta clave en la formación de una nueva generación de profesionales capaces de combinar habilidades técnicas y creativas de manera efectiva y eficiente. Su implementación promete abrir nuevas vías para la colaboración interdisciplinaria, fomentando un enfoque más integrador y holístico en la educación y el desarrollo profesional, alineado con las dinámicas del mundo moderno.

2. Problema

2.1. Antecedentes

La intersección entre la tecnología y la creatividad ha marcado un punto de inflexión en el desarrollo de aplicaciones interactivas, particularmente en el ámbito educativo donde la gamificación y la realidad virtual/aumentada (RV/RA) han emergido como herramientas didácticas clave. Históricamente, la colaboración entre áreas técnicas y creativas ha sido un desafío en contextos académicos, donde la falta de experiencia complementaria entre estudiantes de diferentes disciplinas ha limitado el alcance y la profundidad de los proyectos conjuntos. En la Universidad Politécnica Salesiana, los estudiantes de la carrera de Computación exhiben una notable capacidad en programación, pero a menudo carecen de las destrezas necesarias en diseño de objetos 2D y 3D, críticos para proyectos de gamificación y RV/RA.

Paralelamente, los estudiantes de Diseño Multimedia poseen una formación sólida en la creación de objetos visuales, pero enfrentan barreras al carecer de habilidades de programación para materializar sus conceptos en aplicaciones funcionales. Esta disparidad ha revelado la necesidad de una plataforma unificada que no solo fomente la colaboración interdisciplinaria, sino que también propicie un espacio de intercambio y enriquecimiento de conocimientos técnicos y creativos. La ausencia de tal sinergia ha impedido la optimización de recursos y la realización de proyectos integrados que podrían beneficiarse mutuamente de las habilidades de ambos grupos de estudiantes.

2.2. Importancia y Alcances

La importancia de este estudio radica en la creciente demanda de soluciones tecnológicas que integren efectivamente elementos de diseño y programación. La gamificación y la realidad virtual/aumentada (RV/RA) están redefiniendo el panorama educativo, ofreciendo nuevas vías para el aprendizaje interactivo y la capacitación profesional. En este escenario, los estudiantes de Computación y Diseño Multimedia son actores clave cuyas habilidades combinadas pueden generar productos educativos y de entretenimiento de alto impacto. La colaboración efectiva entre estas disciplinas es esencial no solo para la Universidad Politécnica Salesiana sino también para las instituciones educativas y empresas del sector tecnológico que buscan innovar en sus métodos de enseñanza y en la oferta de contenidos digitales.

El proyecto de grado propuesto apunta a beneficiar directamente a los estudiantes de ambas carreras, facilitando un entorno de colaboración que potenciará su capacidad para desarrollar proyectos conjuntos más robustos y de mayor calidad. Indirectamente, el proyecto beneficiará a la comunidad educativa en su conjunto, proporcionando un repositorio de recursos y herramientas que puede ser adoptado por otras instituciones para fomentar la interdisciplinariedad y la innovación.

Actualmente, la falta de una plataforma integrada se manifiesta en la dificultad para compartir conocimientos y recursos entre los estudiantes, resultando en una limitada transferencia de habilidades y en proyectos que no explotan completamente el potencial creativo y técnico de los participantes. Esta situación también se refleja en el sector educativo más amplio, donde las instituciones luchan por ofrecer experiencias de aprendizaje que involucren las últimas tecnologías. La propuesta de este trabajo de grado tiene el potencial de influir positivamente en la dinámica de trabajo actual, fomentando una cultura de colaboración que puede extenderse más allá de las aulas universitarias y tener un efecto transformador en los sectores económicos y empresariales enfocados en la educación y el entretenimiento digital.

2.3. Delimitación

El trabajo realizado se establece en las siguientes dimensiones fundamentales:

1. Espacial: La implementación del módulo está restringida a la plataforma Edutainment utilizada dentro de la Universidad Politécnica Salesiana, con la intención de expandir su uso a otras instituciones en el futuro, según los resultados y aceptación del proyecto.
2. Temporal: El desarrollo del módulo se llevará a cabo durante el período académico estipulado para el proyecto de titulación, con fases de diseño, implementación y evaluación claramente definidas y sujetas a la calendarización académica.
3. Sectorial: El enfoque se centra en la colaboración entre los estudiantes de las carreras de Computación y Diseño Multimedia, abordando sus necesidades específicas y buscando superar las limitaciones actuales en cuanto a la colaboración y el intercambio de recursos didácticos y creativos.
4. Institucional: Se limita a los recursos y capacidades ofrecidas por las facultades involucradas, así como a las políticas y normativas de la Universidad que rigen los proyectos de titulación y la colaboración interdepartamental.

3. Objetivos

3.1. Objetivo General

Desarrollar e implementar un módulo como repositorio de recursos colaborativos mediante Spring, Angular y PostgreSQL para la plataforma de Edutainment del grupo de GIHP4C de la Universidad Politécnica Salesiana.

3.2. Objetivos Específicos

OE1. Investigar y definir de manera precisa las necesidades de almacenamiento y gestión de scripts y objetos de diseño para la plataforma Edutainment.

OE2. Diseñar la arquitectura y la interfaz del módulo de repositorio, asegurando la accesibilidad y facilidad de uso tanto para los estudiantes de Computación como para los de Diseño Multimedia.

OE3. Desarrollar el módulo de repositorio, implementando de manera efectiva funciones de carga, descarga y visualización de scripts y objetos.

OE4. Establecer un sistema de etiquetado y categorización de scripts y objetos que permita una búsqueda y organización eficientes.

OE5. Realizar pruebas exhaustivas del módulo para identificar y corregir posibles problemas de funcionamiento y optimizar su rendimiento.

OE6. Elaborar documentación detallada y material de capacitación destinado a estudiantes y profesores para facilitar el uso adecuado del módulo de repositorio.

OE7. Evaluar de manera rigurosa el funcionamiento del módulo en la colaboración interdisciplinaria y la calidad de los proyectos, recopilando retroalimentación valiosa de los usuarios y ajustando el sistema en consecuencia.

4. Revisión de la literatura

En esta sección se presentarán los conceptos fundamentales que sustentan la creación de un repositorio para recursos de gamificación, realidad virtual y aumentada. Se explorará no solo la importancia y el impacto de estas tecnologías en el ámbito educativo y de entretenimiento, sino también cómo un repositorio centralizado puede facilitar el acceso y la reutilización de recursos educativos y lúdicos en estos campos. Se discutirán los principios de la gamificación, cómo estas prácticas pueden mejorar el aprendizaje y la participación, y el potencial de la realidad virtual/aumentada para crear experiencias inmersivas y efectivas. Además, se abordará la gestión de datos y las mejores prácticas en el desarrollo de repositorios digitales, enfatizando la necesidad de sistemas seguros, accesibles y eficientes para la gestión y recuperación de recursos, apoyándose en arquitecturas de microservicios y herramientas como Spring y Angular. Este análisis integral busca proveer una base teórica sólida que respalde la viabilidad y eficacia del repositorio propuesto.

4.1. Gamificación

La gamificación se ha integrado en diversos campos como la educación, el marketing y la capacitación corporativa, demostrando ser una herramienta efectiva para aumentar la participación y la retención de conocimientos (Borrás Gené, 2015). Utilizando elementos de juego en contextos no lúdicos, la gamificación estimula la motivación y el compromiso, haciendo que las actividades educativas sean más atractivas y memorables. En el ámbito educativo, se ha observado que la gamificación puede mejorar significativamente la experiencia de aprendizaje, promoviendo un enfoque más activo y participativo por parte de los estudiantes. Su inclusión en los programas educativos ha mostrado resultados positivos en la mejora de la comprensión y retención del material educativo (Ortiz-Colón, Jordán, & Agredal, 2018).

4.2. Realidad Virtual/Aumentada

La Realidad Virtual (RV) y la Realidad Aumentada (RA), reconocidas por su versatilidad, ofrecen aplicaciones innovadoras en diversos ámbitos. Desde la educación, donde transforman el aprendizaje en una experiencia inmersiva, hasta el sector de la salud, donde facilitan procedimientos quirúrgicos y rehabilitación, estas tecnologías están revolucionando la manera en que interactuamos con el mundo digital. En el entretenimiento, la RV y la RA crean

experiencias de juego y espectáculos visuales sin precedentes, mientras que, en el sector empresarial, mejoran la capacitación y la visualización de datos complejos. Estas herramientas no solo mejoran las experiencias existentes, sino que también abren la puerta a nuevas posibilidades en campos como la arquitectura, el diseño y más allá, señalando un futuro donde lo virtual y lo real se fusionan de maneras cada vez más integradas y sorprendentes (Gutiérrez, 2020).

Sin embargo, en este proyecto nos centramos en como la RA y RV tienen un potencial significativo en el turismo, en la educación y en la representación de leyendas y mitos de la cultura ecuatoriana. En turismo, estas tecnologías recrean escenarios históricos y culturales, ofreciendo a los visitantes una experiencia inmersiva del pasado. La RA se usa para superponer imágenes y narrativas en sitios históricos, enriqueciendo la visita con contextos y leyendas. En educación, la RV es una herramienta poderosa para visualizar y experimentar eventos y lugares históricos, facilitando un aprendizaje profundo y atractivo sobre la rica historia y mitología ecuatoriana. Estas experiencias no solo educan, sino que también fomentan la apreciación y conservación del patrimonio cultural (Burbano Erazo, 2022; Zambrano, Romero, Dávila, & Balarezo, 2023).

4.3. Arquitectura de Microservicios

En la plataforma Edutainment, la adopción de la arquitectura de microservicios juega un papel fundamental. Esta arquitectura permite diseñar la plataforma como una colección de servicios independientes y ligeros, cada uno con su propia base de datos y lógica de negocio. Este enfoque ofrece ventajas significativas: facilita la escalabilidad del sistema, ya que cada servicio puede escalarse de forma independiente según la demanda; mejora la resiliencia, pues el fallo en un microservicio no afecta a los demás; y promueve una mayor agilidad en el desarrollo y el despliegue de nuevas características. Además, la arquitectura de microservicios se alinea bien con prácticas de desarrollo ágil y entrega continua, permitiéndome actualizar y mejorar continuamente la plataforma sin interrumpir su funcionamiento global (Contreras, 2018).

Al implementar la arquitectura de microservicios en la plataforma Edutainment, buscamos capitalizar su modularidad y flexibilidad. Este enfoque es especialmente ventajoso para nuestro proyecto, ya que estamos desarrollando un módulo dentro de una plataforma que ya integra múltiples módulos. La arquitectura de microservicios nos permite construir y escalar el módulo

de repositorio de recursos de manera independiente, garantizando así una integración y mantenimiento más eficientes. Además, esta arquitectura facilita la adaptación a las cambiantes necesidades tecnológicas y pedagógicas, permitiendo que cada componente de Edutainment evolucione sin obstaculizar el funcionamiento de los demás. Al final, la meta es crear un sistema cohesivo donde cada módulo, trabajando en conjunto, optimice la experiencia de aprendizaje y colaboración en la plataforma. Cada microservicio es desarrollado, desplegado y operado de manera independiente, lo que facilita la actualización y el mantenimiento sin afectar al resto del sistema. Esta arquitectura permite el uso de distintas tecnologías y lenguajes de programación para cada servicio, optimizando el rendimiento y adaptabilidad. Además, mejora la capacidad de respuesta del sistema frente a demandas fluctuantes, ya que cada servicio puede escalar según sea necesario. En términos de seguridad, esta arquitectura permite aislar fallas, reduciendo el riesgo de interrupciones en el sistema completo. La implementación de microservicios es clave para crear una plataforma robusta, eficiente y capaz de evolucionar rápidamente ante los cambios y necesidades del proyecto Edutainment (Contreras, 2018).

4.4. Gestión de Repositorios Digitales

La gestión de repositorios digitales en la plataforma Edutainment es un aspecto crucial. Este proceso implica la organización, almacenamiento y recuperación eficiente de recursos educativos digitales, como contenidos de gamificación, realidad virtual y aumentada. La efectividad de un repositorio digital depende de su capacidad para clasificar y etiquetar los recursos de manera que sean fácilmente accesibles y reutilizables. Además, la seguridad y la integridad de los datos son fundamentales, asegurando que los recursos sean accesibles solo para usuarios autorizados y estén protegidos contra accesos no autorizados o corrupción de datos. El objetivo es crear un sistema que no solo almacene información de manera eficiente, sino que también facilite la colaboración y el intercambio de conocimientos entre los usuarios (López, 2021).

4.5. Tecnologías de Desarrollo Web

Es importante destacar el papel teórico y práctico de Angular, Spring y PostgreSQL. Estas tecnologías no son solo herramientas; representan la confluencia de avanzadas prácticas de programación y principios de diseño de software. Angular, basado en el patrón de diseño MVC (Modelo-Vista-Controlador), optimiza la interacción usuario-interfaz. Spring, siguiendo el

principio de "inversión de control", facilita la creación de aplicaciones modulares y de fácil mantenimiento. PostgreSQL, por su parte, se alinea con la teoría de bases de datos relacionales, proporcionando robustez y escalabilidad. La elección de estas tecnologías refleja un compromiso con prácticas de desarrollo modernas y eficientes, fundamentales para el éxito del proyecto.

4.5.1. Angular

Angular es un framework de desarrollo frontend basado en TypeScript que facilita la construcción de aplicaciones web y móviles eficientes y sofisticadas. Con su arquitectura basada en componentes, Angular permite a los desarrolladores crear interfaces de usuario altamente interactivas y dinámicas. Estos componentes son reutilizables y fáciles de mantener, lo que hace que Angular sea ideal para proyectos grandes y complejos. Angular también maneja eficientemente la actualización del modelo de objeto de documento (DOM) para reflejar los cambios en los datos, lo que es crucial para las aplicaciones de página única (SPA), donde la fluidez y la reactividad son esenciales. Además, Angular incluye servicios integrados y un potente sistema de enrutamiento, mejorando la experiencia general del usuario y facilitando la construcción de aplicaciones ricas e interactivas (Angular Team, s.f.).

En el contexto del módulo de repositorio de recursos de la plataforma Edutainment, Angular es utilizado para desarrollar una interfaz de usuario que no solo sea atractiva y fácil de usar, sino que también permita una interacción eficiente y dinámica con los recursos educativos. La capacidad de Angular para actualizar la vista de manera eficiente cuando los datos cambian es particularmente útil para garantizar que los usuarios tengan acceso en tiempo real a la información más actualizada del repositorio.

4.5.2. Spring

Spring es un framework de desarrollo backend ampliamente utilizado, conocido por su robustez y versatilidad. Ofrece un enfoque integral para el desarrollo de aplicaciones Java, destacando por su sistema de inyección de dependencias que facilita la creación y gestión de componentes de manera eficiente. Esto es crucial en entornos donde se requiere flexibilidad y modularidad, como en arquitecturas basadas en microservicios. Spring permite a los desarrolladores construir aplicaciones desacopladas y cohesivas, mejorando la mantenibilidad y escalabilidad del software (González, 2016).

En el módulo de repositorio de recursos de la plataforma Edutainment, Spring es fundamental para la implementación de la lógica de negocio y la gestión de datos. Su capacidad para manejar microservicios permite un desarrollo más ágil y modular del backend del repositorio, facilitando la implementación de funcionalidades complejas y la integración eficiente con otros componentes y servicios de la plataforma. Además, la utilización de Spring Boot dentro del ecosistema de Spring simplifica la configuración y el despliegue, acelerando el desarrollo y permitiendo iteraciones más rápidas, lo que es esencial para adaptarse a las necesidades cambiantes del proyecto.

4.5.3. PostgreSQL

PostgreSQL es un avanzado sistema de gestión de bases de datos objeto-relacional, altamente respetado por su solidez, seguridad y su fiel adhesión a los estándares SQL. Es conocido por su arquitectura sofisticada, que soporta funcionalidades como transacciones ACID (Atomicidad, Consistencia, Aislamiento, Durabilidad), recuperación ante fallos y concurrencia robusta. Además, PostgreSQL maneja de manera eficiente tanto datos estructurados como no estructurados, lo que lo hace ideal para aplicaciones que requieren alta flexibilidad y capacidad de adaptación. Con soporte para una amplia gama de tipos de datos, incluyendo JSON y datos geoespaciales, así como un potente sistema de indexación y búsqueda de texto completo, PostgreSQL es una solución versátil y escalable para una variedad de necesidades de manejo de datos (The PostgreSQL Global Development Group, s.f.).

En el módulo de repositorio de recursos de Edutainment, PostgreSQL es utilizado debido a su habilidad para gestionar eficientemente los datos estructurados y no estructurados. Su confiabilidad y escalabilidad son fundamentales para mantener la integridad y el acceso seguro a los recursos educativos, que pueden incluir una variedad de formatos y tamaños. La elección de PostgreSQL asegura que la plataforma pueda crecer y adaptarse a las necesidades cambiantes del entorno educativo, manteniendo al mismo tiempo un alto nivel de rendimiento y seguridad.

4.5.4. Docker

Docker es una plataforma de contenedores que permite empaquetar una aplicación y sus dependencias en un contenedor virtual aislado que puede ejecutarse en cualquier sistema operativo que tenga instalado Docker. Esta tecnología se ha convertido en un estándar de facto para el despliegue de aplicaciones en entornos de desarrollo, prueba y producción debido a su

capacidad para garantizar la consistencia entre los entornos y simplificar el proceso de DevOps (Miell & Sayers, 2019).

La utilización de Docker en el proyecto es clave para respaldar el principio de "construir una vez, desplegar en cualquier lugar", lo que proporciona una gran flexibilidad y eficiencia en el ciclo de vida del desarrollo de software. Con Docker, podemos asegurar que el entorno de desarrollo del módulo de repositorio sea consistente con el entorno de producción, lo que minimiza los problemas comunes relacionados con las diferencias entre entornos. Además, la tecnología de contenedores ofrece una manera más eficiente de utilizar los recursos del sistema en comparación con las máquinas virtuales tradicionales, ya que permite ejecutar múltiples contenedores de manera simultánea en un único host (Docker, 2020).

En el módulo de repositorio de recursos de Edutainment, Docker facilita la implementación de microservicios encapsulando cada servicio y su entorno en un contenedor independiente. Esto permite un escalado y una gestión más eficiente, ya que cada contenedor puede iniciarse, detenerse, trasladarse y replicarse de manera independiente de los otros. Además, la integración con herramientas de orquestación de contenedores como Kubernetes puede llevar la administración y el escalado automático de estos servicios a un nivel superior, aumentando aún más la fiabilidad y disponibilidad del módulo de repositorio.

4.5.5. Eureka Netflix

Netflix Eureka es una herramienta dentro del ecosistema de Spring Cloud que funciona como un servicio de descubrimiento para aplicaciones basadas en microservicios. Permite a los servicios encontrar y comunicarse entre sí sin la necesidad de codificar de forma rígida las direcciones de host y puertos, lo cual es crucial en entornos dinámicos donde las direcciones de los servicios pueden cambiar con frecuencia (Baeldung & Baeldung, 2024).

En la práctica, Eureka usa un servidor de registro donde cada microservicio se registra, y un cliente que consulta este servidor para localizar otros servicios. Esta arquitectura facilita la creación de sistemas distribuidos resilientes y escalables, ya que los servicios pueden registrarse, quitar su registro y descubrir pares de forma dinámica, lo que es particularmente valioso en entornos de nube y de aplicaciones empresariales a gran escala.

En la arquitectura de la plataforma Edutainment, Netflix Eureka juega un papel crucial en la gestión de microservicios, aportando flexibilidad y dinamismo al permitir que se registren y

localicen de manera adaptable a los cambios del entorno de desarrollo. Esto asegura no solo una mayor resiliencia y tolerancia a fallos, manteniendo la operatividad del sistema incluso ante fallas de componentes, sino que también facilita una escalabilidad eficiente. Eureka maneja de forma ágil los registros y descubrimientos de servicios, permitiendo que el sistema se adapte y crezca de acuerdo con las necesidades cambiantes del proyecto. Además, su interfaz de usuario contribuye significativamente a simplificar el monitoreo y la gestión del entorno de microservicios, ofreciendo una visión clara del estado de los servicios y mejorando la eficiencia en la administración del sistema.

4.5.6. Zipkin

Zipkin es una herramienta de seguimiento distribuido de código abierto que se utiliza para recopilar información detallada sobre el rendimiento y la latencia de las aplicaciones. Sirve principalmente para ayudar a los desarrolladores a diagnosticar y solucionar problemas en sistemas de microservicios, proporcionando una visión clara de cómo las solicitudes viajan a través de diferentes servicios. Mediante la recopilación y presentación de datos detallados sobre las peticiones que se realizan dentro de una arquitectura de servicios, Zipkin facilita la identificación de cuellos de botella y otros problemas de rendimiento (Zipkin, s.f.).

Zipkin ofrece una interfaz de usuario que visualiza trazas de servicios, mostrando cómo las solicitudes se propagan y cuánto tiempo tardan en procesarse a través de varios servicios. Esto es especialmente útil en sistemas complejos donde un solo problema puede afectar a múltiples componentes interconectados. Al analizar estas trazas, los desarrolladores pueden entender mejor las interacciones entre servicios y optimizar el rendimiento y la eficiencia del sistema en general.

La integración de Zipkin en el proyecto de Edutainment ilustra un enfoque meticuloso hacia la monitorización y diagnóstico en un entorno de microservicios complejo. Como una herramienta de seguimiento distribuido, Zipkin es vital para recoger datos sobre el rendimiento y la latencia en aplicaciones donde interactúan numerosos microservicios. Su capacidad para ofrecer una visión detallada del tráfico de solicitudes y las interacciones entre los servicios nos permite identificar y abordar eficazmente cuellos de botella y otros problemas de rendimiento. Este enfoque no solo mejora la eficiencia y efectividad del sistema, sino que también es clave para el diagnóstico en tiempo real, permitiendo un mantenimiento proactivo y contribuyendo a una mejora continua de nuestras prácticas de desarrollo y arquitectura. En última instancia, el uso

de Zipkin garantiza una experiencia de usuario fluida y sin interrupciones, alineándose con el objetivo de proporcionar una plataforma de Edutainment tecnológicamente avanzada y confiable.

4.6. Metodologías de Desarrollo Ágil

Estas Metodologías son un conjunto de guías para la gestión de proyectos de software que promueven la entrega iterativa, la colaboración del equipo y la capacidad de respuesta ante el cambio. Se centran en la creación rápida de valor y en la adaptabilidad durante todo el proceso de desarrollo, permitiendo ajustes frecuentes basados en la retroalimentación continua de los usuarios finales o de las partes interesadas (Montero, Cevallos, & Cuesta, 2018). Estas metodologías de desarrollo ágil, pero en particular Scrum, son el núcleo de la estrategia de desarrollo para mi proyecto. Estas metodologías enfatizan la flexibilidad, la adaptabilidad y la colaboración continua, lo cual es esencial en el dinámico campo del software educativo.

Scrum, en específico, es una metodología ágil que promueve la entrega incremental de software, permitiendo revisiones y adaptaciones regulares del producto en desarrollo. Organiza el trabajo en ciclos cortos y manejables conocidos como sprints, que típicamente duran entre dos semanas y un mes. Durante cada sprint, se seleccionan y completan características del producto a partir de una lista priorizada de trabajo conocida como el backlog del producto (Schwaber & Sutherland, 2020).

En el contexto del módulo de repositorio de recursos de la plataforma Edutainment, Scrum facilita la gestión efectiva del proyecto al permitir al equipo de desarrollo responder rápidamente a los cambios, ya sean nuevas ideas pedagógicas que necesitan ser incorporadas en el repositorio o ajustes basados en la retroalimentación de los usuarios. Además, la ceremonia de Scrum, que incluye reuniones diarias de pie, revisiones de sprint y retrospectivas, asegura que el equipo mantenga una comunicación constante y efectiva, y que el progreso sea transparente para todos los interesados.

La adopción de Scrum es fundamental en la coordinación del equipo, en la priorización de tareas y en la entrega de un módulo de repositorio de recursos que no solo cumpla con los requisitos actuales, sino que también tenga la capacidad de adaptarse y desenvolverse con las necesidades futuras de la plataforma Edutainment (Schwaber & Sutherland, 2020).

5. Marco Metodológico

Como se presentó en la revisión de la literatura nuestro marco metodológico se basa en las metodologías de desarrollo ágil en especial SCRUM.

5.1. Metodología SCRUM

El trabajo se organizó en períodos conocidos como "Sprints" usualmente de 2 a 4 semanas. Cada sprint inicio con una reunión de planeamiento en la que se definieron las tareas y se establecieron las metas con precisión. Además, se llevaron a cabo reuniones diarias de seguimiento durante el sprint para compartir avances y detectar posibles obstáculos. Al final de cada sprint, se realizó una revisión para evaluar el trabajo y planificar el próximo ciclo. Esta metodología se enfoca en la comunicación constante, transparencia y capacidad de adaptación, permitiendo a los equipos responder ágilmente a cambios y prioridades del proyecto.

A continuación, se detalla los roles que se establecieron para cada uno de los implicados en el proyecto:

Roles SCRUM

- Product Owner: Ing. Gustavo Bravo
- Scrum Máster: Est. Jhon Llivicota
- Scrum Team: Est. Michael Alvarez

5.1.1. Sprint 1

En el primer sprint, nuestra meta fue establecer comunicación con nuestro tutor y el desarrollador de la plataforma Edutainment. El propósito fue obtener una comprensión del estado actual de la plataforma, lo que nos permitió comenzar a planificar su escalabilidad. En este contexto las actividades realizadas fueron:

5.1.1.1. Comprender necesidades y expectativas.

En esta etapa inicial, el objetivo fue alinear nuestra visión del módulo de repositorio con las expectativas de los principales interesados: nuestro tutor y el desarrollador principal de la plataforma Edutainment. Se llevaron a cabo múltiples reuniones para discutir las funcionalidades existentes de la plataforma y las posibles mejoras. Se recopiló y revisó la documentación existente, que incluyó requisitos de software, casos de uso previos y comentarios de usuarios. Esta información fue esencial para obtener una visión holística del proyecto y establecer un marco para las metas a corto y largo plazo.

El resultado de estas discusiones fue un conjunto de requisitos funcionales y no funcionales claros y una mejor comprensión del alcance del proyecto. Los requisitos funcionales definieron las capacidades específicas que el módulo debe tener, mientras que los requisitos no funcionales se refirieron a criterios de rendimiento, seguridad y usabilidad. Esta comprensión inicial fue crucial para la planificación efectiva de las fases subsiguientes del desarrollo.

5.1.1.2. Examinar la documentación previa de Edutainment.

Una comprensión profunda de la documentación existente fue esencial para garantizar la continuidad y el progreso efectivo del proyecto. Se realizó un análisis exhaustivo de la documentación técnica y de usuario previa de la plataforma Edutainment. Este análisis incluyó manuales de usuario, especificaciones técnicas, informes de errores y actualizaciones de características, lo que nos proporcionó un conocimiento detallado del estado actual de la plataforma.

Específicamente, la documentación técnica nos ofreció insights sobre la arquitectura subyacente, los modelos de datos y las integraciones de sistemas. Por otro lado, la documentación de usuario nos ayudó a entender la experiencia del usuario actual, sus flujos de trabajo y las áreas de mejora desde una perspectiva de usuario final. La revisión detallada de

los informes de errores y las solicitudes de características nos permitió identificar patrones recurrentes y priorizar las áreas que requerían atención inmediata.

5.1.1.3. Investigar mejores prácticas en gestión de contenido en plataformas similares.

Para garantizar que el módulo de repositorio de recursos a desarrollar se alinee con los estándares y las expectativas de los usuarios, se llevó a cabo una investigación sobre las mejores prácticas en la gestión de contenido. Se analizaron plataformas similares de educación y entretenimiento, identificando características clave y funcionalidades que han demostrado éxito en el compromiso y la satisfacción del usuario.

Esta investigación se enfocó en comprender cómo otras plataformas manejan el almacenamiento de recursos, la categorización, la búsqueda y la recuperación de contenido, así como las estrategias para mantener a los usuarios comprometidos con el material. Se estudiaron casos de estudio y artículos de investigación para identificar tendencias emergentes y técnicas probadas en la gestión y entrega de contenido digital.

5.1.1.4. Investigar prácticas de diseño de interfaz

El diseño de la interfaz de usuario es un aspecto crítico para la usabilidad y la experiencia general del usuario en la plataforma Edutainment. Por ello, se dedicó una parte significativa del Sprint 1 a investigar prácticas efectivas de diseño de interfaz que pudieran aplicarse al módulo de repositorio de recursos. Esto implicó estudiar las tendencias actuales en UI/UX, como la simplicidad, la coherencia visual y la accesibilidad, que son esenciales para una plataforma educativa.

Analizando interfaces líderes en el sector y revisando literatura especializada, se identificó patrones de diseño que no solo atraen a los usuarios, sino que también facilitan la interacción y el acceso a la información. La exploración de estos patrones incluyó la evaluación de menús de navegación intuitivos, esquemas de colores efectivos y la implementación de diseños responsivos que se adaptan a diferentes dispositivos y dimensiones de pantalla.

5.1.1.5. Definir la arquitectura del módulo de repositorio

El último paso crítico en este sprint inicial fue la definición de la arquitectura del módulo de repositorio de recursos. Se eligió una arquitectura de 3 capas por su capacidad para separar claramente la lógica de negocio, la interfaz de usuario y el acceso a datos, lo cual facilita la mantenibilidad y la escalabilidad del sistema.

La primera capa es la Presentación, donde se maneja la interacción con el usuario. Utiliza Angular para proporcionar una interfaz reactiva y accesible, que permite a los usuarios interactuar con el repositorio de forma intuitiva.

La segunda capa es la Lógica de Negocio, implementada con el framework Spring y el lenguaje de Java. Aquí se define la lógica central del módulo, incluyendo la gestión de recursos, la autorización de usuarios y el procesamiento de solicitudes.

La tercera y última capa es la Persistencia de Datos, donde PostgreSQL almacena y gestiona eficientemente todos los datos relacionados con los recursos. Esta capa garantiza que la información esté bien estructurada y sea fácilmente accesible.

La Figura 1 muestra un esquema de la arquitectura de tres capas, ilustrando cómo cada una interactúa con las otras para formar un sistema cohesivo y bien organizado.

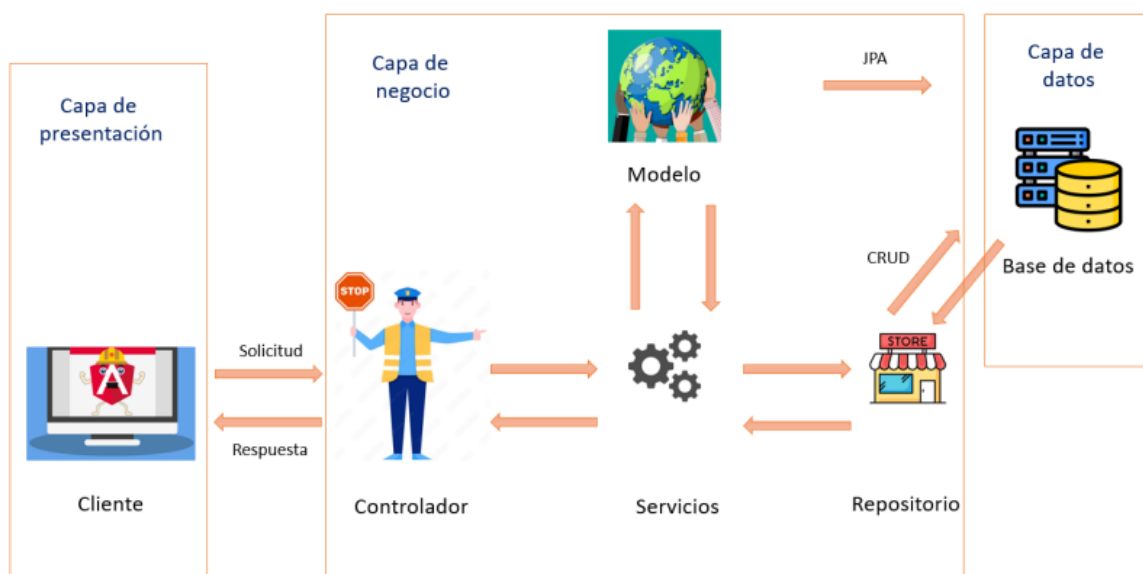


Figura 1: Comunicación entre las diversas capas de la arquitectura

Esta arquitectura no solo promueve una separación de preocupaciones, sino que también apoya la integración y el trabajo conjunto de las diferentes tecnologías que conforman la plataforma Edutainment. Con esta base sólida, el módulo de repositorio está preparado para evolucionar y expandirse conforme a las necesidades futuras del entorno educativo.

5.1.2. Sprint 2

En nuestro segundo sprint nuestro objetivo fue empezar ya el desarrollo de la base de datos, el backend y el frontend de la aplicación.

5.1.2.1. Diseñar el esquema de la base de datos

Como primer paso en este Sprint, se procedió a diseñar el esquema de la base de datos para el módulo de repositorio de recursos en la plataforma Edutainment. Este esquema es el fundamento para almacenar, organizar y recuperar todos los datos de manera eficiente y segura.

La Figura 2 ilustra el esquema relacional diseñado. El diseño se centró en asegurar la normalización de la base de datos para evitar redundancias y promover la integridad de los datos. Las entidades principales, como user, resource, media_references, game, resource_tag, y tag, están interconectadas para reflejar las relaciones y dependencias lógicas entre los diferentes tipos de datos.

La tabla user almacena información sobre los usuarios, que son fundamentales para el control de acceso y la personalización del contenido. La tabla resource contiene metadatos de los recursos disponibles en el repositorio, incluyendo referencias a los archivos de medios correspondientes en la tabla media_references. La asociación entre recursos y etiquetas (tags) se maneja a través de una tabla de relación resource_tag, lo que permite una búsqueda y clasificación efectivas de los recursos.

Cada entidad fue cuidadosamente estructurada para incluir todos los campos necesarios, como se muestra en la Figura 2, con timestamps para rastrear la creación y actualización de los registros. La integridad referencial se mantiene a través de claves foráneas, asegurando la consistencia de las relaciones entre tablas.

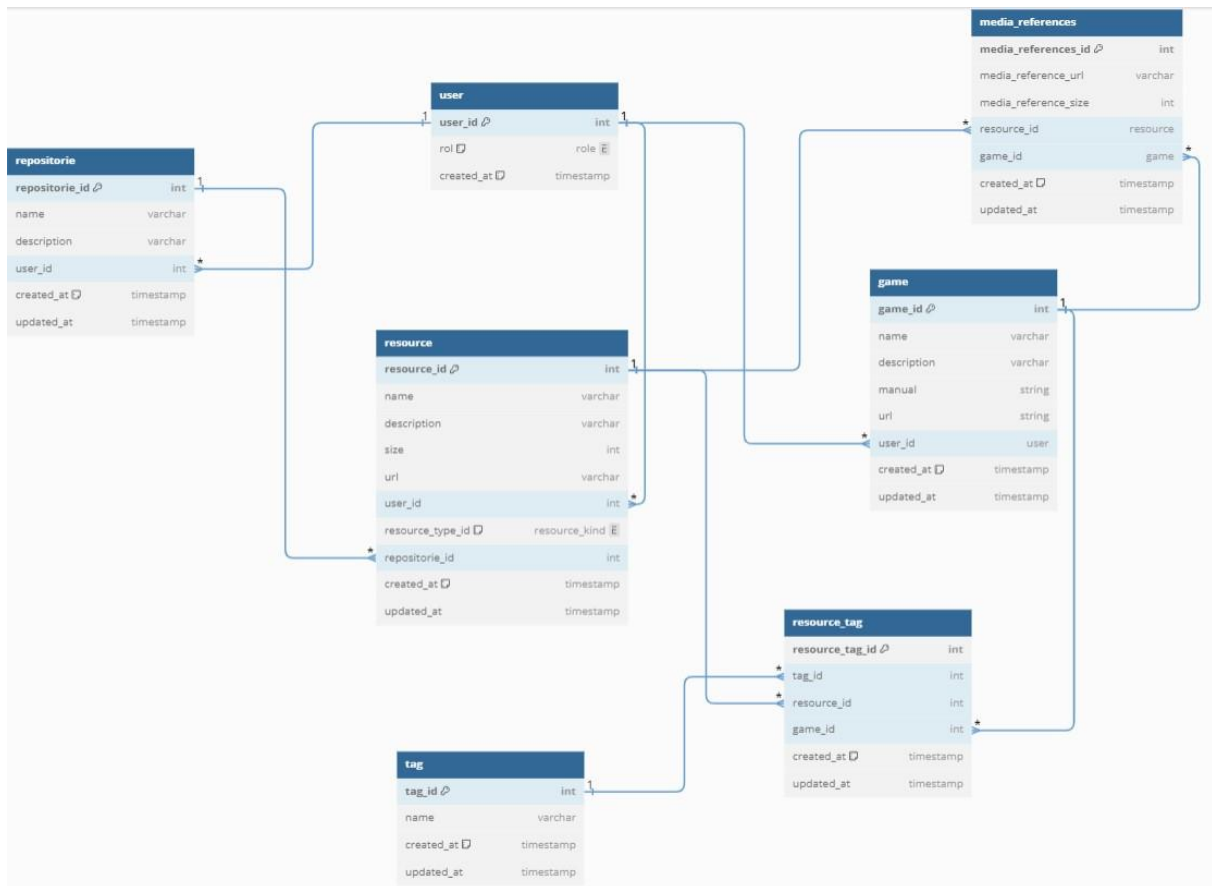


Figura 2: Esquema de la Base de Datos del Módulo de Repositorio de Recursos

Este diseño de base de datos ofrece una base sólida para el desarrollo del módulo y su integración con la plataforma Edutainment, asegurando que la gestión de datos sea escalable, segura y eficiente.

5.1.2.2. Desarrollar los servicios seguros de carga y descarga de scripts y objetos

Servicio de descarga:

Como parte de las actividades del Sprint 2, se desarrolló un servicio de descarga fundamental para el funcionamiento del módulo de repositorio, este servicio fue implementado en Java utilizando el framework Spring. Este servicio permite a los usuarios descargar scripts y objetos de manera segura y eficiente.

En la Figura 3 se muestra la inicialización del servicio y el método principal para la carga del recurso. Se utiliza la inyección de dependencias de Spring para definir la ubicación de los archivos y se implementa la lógica para comprimir y preparar los archivos para su descarga.


```

@Service
public class DownloadService {
    private final Path rootLocation;

    @Autowired
    public DownloadService(@Value("${file.storage.location}") String storageLocation) {
        this.rootLocation = Paths.get(storageLocation).toAbsolutePath().normalize();
    }

    public Resource loadFolderAsResource(String folderPath) {
        try {
            Path folderToCompress = this.rootLocation.resolve(folderPath).normalize();
            String zipFileName = folderToCompress.getFileName().toString() + ".zip";
            Path zipFilePath = folderToCompress.getParent().resolve(zipFileName);

            // Siempre comprimir la carpeta, independientemente de si el archivo ZIP ya existe
            compressFolder(folderToCompress, zipFilePath);

            // Cargar el recurso ZIP
            Resource resource = new UrlResource(zipFilePath.toUri());
            if (resource.exists()) {
                return resource;
            } else {
                throw new RuntimeException("El archivo ZIP no pudo ser creado o no existe.");
            }
        } catch (IOException ex) {
            throw new RuntimeException("Error al cargar la carpeta como recurso", ex);
        }
    }
}

```

Figura 3: Inicialización del Servicio de Descarga y Método de Carga de Recursos

En el fragmento de código que se muestra en la Figura 4, se detalla el proceso de compresión de la carpeta de recursos en un archivo ZIP. Este método asegura que cada archivo dentro de la carpeta especificada se agregue correctamente al archivo ZIP, gestionando eficientemente las excepciones y errores.

```

private void compressFolder(Path folderToCompress, Path zipFilePath) throws IOException {
    try (ZipOutputStream zos = new ZipOutputStream(new FileOutputStream(zipFilePath.toFile()));
        Stream<Path> paths = Files.walk(folderToCompress)) {
        paths.filter(path -> !Files.isDirectory(path))
            .forEach(path -> {
                ZipEntry zipEntry = new ZipEntry(folderToCompress.getParent().relativize(path).toString());
                try {
                    zos.putNextEntry(zipEntry);
                    Files.copy(path, zos);
                    zos.closeEntry();
                } catch (IOException e) {
                    throw new RuntimeException("Error al comprimir la carpeta", e);
                }
            });
    }
}

```

Figura 4: Método de Compresión en el Servicio de Descarga

Estos fragmentos de código ilustran la implementación técnica de una característica clave del módulo de repositorio, evidenciando la aplicación práctica de los conceptos discutidos en la revisión de la literatura y resaltando la metodología ágil de desarrollo empleada en el proyecto.

Servicio de carga

Después de establecer el servicio de descarga, el siguiente paso fue desarrollar un servicio de carga seguro y eficiente, igualmente implementado en Java con el framework Spring. Este servicio proporciona a los usuarios la capacidad de subir scripts y objetos al módulo de repositorio de forma segura.

La Figura 5 muestra la inicialización del servicio de carga y la configuración de la ubicación de almacenamiento de archivos, utilizando la inyección de dependencias de Spring para asegurar que los archivos se almacenen en la ubicación especificada en `application.properties`. Además, se implementaron medidas de seguridad para evitar la creación de directorios no autorizados o el almacenamiento de archivos con nombres inapropiados.

```
@Service
@Slf4j
public class UploadService {
    private final Path rootLocation;

    @Autowired
    public UploadService(@Value("${file.storage.location}") String storageLocation) {
        this.rootLocation = Paths.get(storageLocation).toAbsolutePath().normalize();
        try {
            Files.createDirectories(this.rootLocation);
        } catch (IOException e) {
            throw new StorageException("Could not create the directory where the uploaded files will be stored.", e);
        }
    }
}
```

Figura 5: Inicialización del Servicio de Carga y Configuración de Almacenamiento

El fragmento de código en la Figura 6 inicia con la definición del método `storeFile`, que es responsable de procesar las solicitudes de carga de archivos en el sistema. Se reciben los detalles de la carga a través del objeto `UploadRequest`, que contiene la información necesaria como el archivo en sí, el nombre del recurso, el nombre del repositorio y la fuente de la carga.

Se observa en la Figura 6 la lógica para extraer y limpiar los nombres del archivo y los recursos del `UploadRequest` para asegurar que las rutas de los archivos sean seguras y no contengan secuencias de ruta inválidas. Además, la función determina la ubicación final del archivo basándose en el nombre de la fuente, el repositorio y el recurso. Se crea un sistema de carpetas jerárquico que organiza los archivos subidos según su fuente y categoría, lo que facilita su gestión y recuperación. Antes de guardar el archivo, el código verifica si los directorios necesarios existen y, si no, los crea. Esto asegura que la estructura de directorios necesaria para almacenar el archivo esté presente. El código también gestiona las excepciones para asegurar que los errores durante el proceso de carga sean comunicados de manera efectiva al usuario.

```

public UploadResponse storeFile(UploadRequest uploadRequest) {
    var file = uploadRequest.getFile();
    var resourceName = formatName(uploadRequest.getResource_name());
    var repositoryName = formatName(uploadRequest.getRepositorie_name());
    var source = uploadRequest.getSource();

    String originalFileName = StringUtils.cleanPath(Objects.requireNonNull(file.getOriginalFilename()));
    validateFileName(originalFileName, resourceName);

    String fileName = constructFileName(resourceName, originalFileName, uploadRequest.getIndex_number());

    try {
        String folderName = source.name().toLowerCase();
        Path folderLocation = rootLocation.resolve(folderName);
        Path resourceLocation;

        // Determinar la ubicación del recurso en función de si repositoryName está presente o no
        if (!repositoryName.isEmpty()) {
            Path repositoryLocation = folderLocation.resolve(repositoryName);
            resourceLocation = repositoryLocation.resolve(resourceName);
        } else {
            resourceLocation = folderLocation.resolve(resourceName); // Ruta directa a la carpeta del juego
        }

        Files.createDirectories(resourceLocation);

        Path targetLocation = resourceLocation.resolve(fileName);
        Files.copy(file.getInputStream(), targetLocation, StandardCopyOption.REPLACE_EXISTING);

        return UploadResponse.builder()
            .url(targetLocation.toString())
            .size(file.getSize())
            .resourceKind(FileTypes.getFileType(originalFileName))
            .build();

    } catch (IOException ex) {
        throw new StorageException("Could not store file " + originalFileName + ". Please try again!", ex);
    }
}

```

Figura 6: Método de Almacenamiento de Archivos del Servicio de Carga

En la figura 7 se observan las siguientes funciones:

La función `validateFileName` se asegura de que el nombre del archivo cargado no incluya secuencias de ruta que podrían ser explotadas para acceder a directorios no autorizados, una práctica esencial para prevenir vulnerabilidades de seguridad. Además, esta función verifica que el nombre del archivo no contenga el nombre del recurso como parte de su ruta, lo que es una comprobación adicional para mantener la estructura de directorios organizada y predecible.

La función `formatName` es una operación de utilidad que formatea el nombre del repositorio eliminando espacios y convirtiéndolo a minúsculas. Esta estandarización es crucial para evitar problemas relacionados con la consistencia de nombres en diferentes sistemas operativos y plataformas.

La función `constructFileName` construye el nombre final del archivo basándose en el nombre del recurso, un número de índice opcional y la extensión del archivo. Este nombre formateado

asegura un sistema de nomenclatura coherente y facilita la organización y recuperación de archivos en el sistema.

Por último, la clase `StorageException` extiende `RuntimeException` y se utiliza para manejar errores relacionados con el almacenamiento de archivos, proporcionando mensajes claros al usuario en caso de problemas durante el proceso de carga.

```
private void validateFileName(String originalFileName, String resourceName) {
    if (originalFileName.contains("..")) {
        throw new StorageException("Sorry! Filename contains invalid path sequence " + originalFileName);
    }
    if (originalFileName.contains(resourceName)) {
        throw new StorageException("Sorry! Filename contains invalid path sequence " + originalFileName);
    }
}

private String formatName(String repositoryName) {
    return repositoryName.trim().replaceAll(" ", "-").toLowerCase();
}

private String constructFileName(String resourceName, String originalFileName, String indexNumber) {
    String extension = FileTypes.getFileExtension(originalFileName);
    return (indexNumber != null && !indexNumber.isEmpty()) ? resourceName + "-" + indexNumber + "." + extension
        : resourceName + "." + extension;
}

public static class StorageException extends RuntimeException {
    public StorageException(String message) {
        super(message);
    }

    public StorageException(String message, Throwable cause) {
        super(message, cause);
    }
}
```

Figura 7: Validación y Formateo en el Servicio de Carga

Este servicio es esencial para la funcionalidad del módulo de repositorio, ya que permite a los usuarios contribuir con nuevos contenidos y mantener el repositorio actualizado. La implementación técnica de este servicio demuestra la aplicación práctica de los conceptos de desarrollo ágil y refleja el compromiso del proyecto con la creación de un sistema robusto y centrado en el usuario.

5.1.2.3. Implementar una interfaz de usuario para acceder y descargar contenido

Una vez establecido el backend con los servicios de carga y descarga, la siguiente fase fue centrarse en la interfaz de usuario (UI). La UI es el puente entre el usuario y la funcionalidad del módulo de repositorio, por lo que su diseño y desarrollo son cruciales para la experiencia del usuario. Se priorizó la creación de una interfaz intuitiva y amigable que facilitara la navegación, el acceso y la descarga de contenidos de forma eficiente.

Utilizando Angular, se implementó una interfaz que se integra perfectamente con los servicios de backend ya desarrollados. El prototipo de la interfaz que se muestra en la Figura 8, se realizó siguiendo las mejores prácticas de UX para garantizar que los usuarios pudieran encontrar fácilmente los recursos que necesitan y descargarlos sin complicaciones. Se crearon vistas de listado de recursos con capacidades de filtrado y búsqueda, y se diseñaron flujos de usuario para la selección y descarga de archivos individuales o todo el repositorio que contiene los recursos. También se aseguró la accesibilidad y la responsividad, permitiendo que la plataforma sea utilizada en una variedad de dispositivos y dimensiones de pantalla.



Figura 8: Prototipo de la Interfaz de Usuario para el Módulo de Repositorio de Recursos

Este enfoque iterativo y centrado en el usuario asegura que la interfaz no solo cumpla con los requisitos técnicos, sino que también responda a las necesidades reales de los usuarios finales, mejorando la adopción y la satisfacción del usuario con el módulo de repositorio de recursos.

5.1.2.4. Desarrollar funciones de visualización de scripts y objetos.

Continuando con la expansión del módulo, la funcionalidad para visualizar scripts y objetos fue una de las mejoras clave implementadas en este sprint. La visualización efectiva es esencial para proporcionar a los usuarios una experiencia completa, permitiéndoles inspeccionar los recursos antes de comprometerse con la descarga.

La nueva funcionalidad se centró en la creación de una interfaz interactiva que ofrece una vista previa de los scripts y objetos. Esto se complementa con metadatos detallados que brindan

información esencial sobre cada recurso, incluyendo descripciones, el tamaño del archivo y datos de actualización.

El diseño de esta interfaz de visualización se plasmó en un prototipo detallado que se muestra en la Figura 9. Este prototipo fue desarrollado con un enfoque en la accesibilidad y la usabilidad, asegurando que la funcionalidad de visualización sea intuitiva y asequible para todos los beneficiarios, independiente de su nivel de habilidad técnica.



Figura 9: Prototipo de la Interfaz de Visualización de Scripts y Objetos

La introducción de estas funciones de visualización representa un paso significativo hacia el logro de un ecosistema educativo digital más interactivo y enriquecedor. Permitiendo a los usuarios explorar y evaluar los recursos con facilidad, el módulo de repositorio se solidifica como una herramienta educativa integral y dinámica.

5.1.2.5. Implementar un sólido control de acceso con roles y permisos.

Una de las piezas fundamentales del módulo de repositorio de recursos es el control de acceso basado en roles, que garantiza que los usuarios interactúen con la plataforma de acuerdo con sus privilegios asignados. La implementación de esta característica comenzó con la definición de una matriz de roles y permisos que especifica claramente qué acciones pueden realizar diferentes usuarios, como administradores, editores y visitantes.

El esquema de control de acceso se diseñó para ser extensible y seguro, permitiendo una fácil administración de los permisos y la adición de nuevos roles según las necesidades evolutivas de la plataforma. Se establecieron políticas para la autenticación de usuarios y la autorización de acciones, utilizando tokens de seguridad y sesiones gestionadas para mantener la integridad y privacidad de los datos de usuario.

Para ilustrar en la Figura 10 se observa la lógica detrás del control de acceso, se creó un diagrama de flujo que muestra el proceso de autenticación y autorización, desde el momento en que un usuario intenta acceder a un recurso hasta la verificación de sus permisos y la consecuente concesión o denegación de acceso.

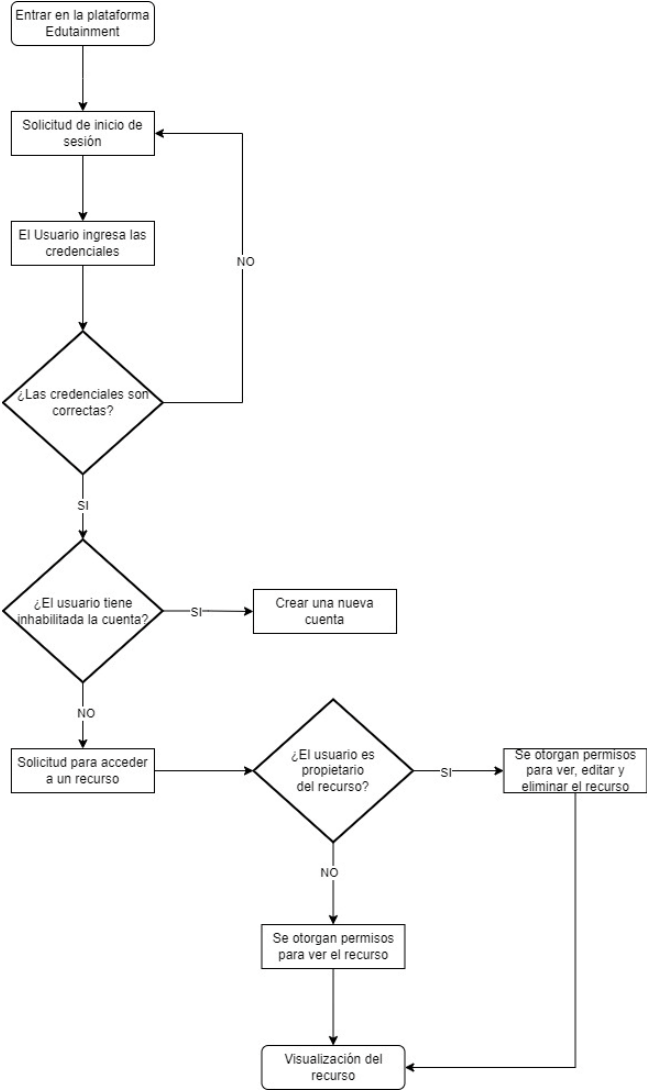


Figura 10: Diagrama de Flujo del Control de Acceso Basado en Roles

Este sistema de control de acceso se integra con todas las demás funcionalidades del módulo, asegurando que solo los usuarios autorizados puedan realizar operaciones sensibles como la carga o eliminación de recursos. La implementación de estas medidas es esencial para proteger la plataforma contra accesos no autorizados y para mantener un entorno seguro y confiable para todos los usuarios.

5.1.3. Sprint 3

En nuestro tercer sprint nuestro objetivo fue añadir funcionalidad extra al desarrollo anterior para mejorar la organización y comodidad de los usuarios.

5.1.3.1. Crear una jerarquía de etiquetas.

Primeramente, es este Sprint, nos centramos en el desarrollo de una jerarquía de etiquetas para la organización eficiente de los recursos en el módulo de repositorio. Esta jerarquía es esencial para mantener un sistema coherente y fácil de navegar.

Se diseñó una estructura de etiquetas que incluye categorías y subcategorías, adaptadas a la naturaleza y tipo de los recursos disponibles. Para garantizar la consistencia, se establecieron etiquetas predeterminadas que facilitan la clasificación uniforme de los contenidos. Estas etiquetas predeterminadas aseguran que todos los recursos se categoricen de manera coherente, lo que ayuda a mantener un orden lógico en el repositorio.

La implementación de esta jerarquía de etiquetas se integró en la interfaz de usuario, diseñada con Angular, para proporcionar una experiencia de usuario coherente y eficiente. La interfaz permite a los usuarios y colaboradores asignar y modificar etiquetas de manera intuitiva, manteniendo la facilidad de uso y la accesibilidad como prioridades.

El enfoque en una jerarquía de etiquetas bien definida y la inclusión de etiquetas predeterminadas son pasos esenciales para mejorar la gestión y organización de los recursos, facilitando un sistema más ordenado y eficaz que responde a las necesidades de los usuarios del módulo de repositorio de recursos.

5.1.3.2. Agregar metadatos a scripts y objetos.

La incorporación de metadatos a scripts y objetos en el repositorio representa un avance crucial en la gestión y organización de los recursos. Se identificaron y definieron los tipos de metadatos más relevantes para los recursos del repositorio, incluyendo información como título, autor, fecha de creación, versión, y palabras clave. Esta definición de metadatos facilita la gestión y categorización de los recursos, aportando una estructura y claridad esenciales.

Posteriormente, se realizó la implementación de estos metadatos en la base de datos, asegurando una integración eficiente y coherente con el sistema existente. Paralelamente, se diseñó una interfaz de usuario en Angular, intuitiva y amigable, que permite a los usuarios y colaboradores añadir y editar metadatos de forma sencilla y eficiente.

Finalmente, se llevaron a cabo pruebas exhaustivas para asegurar que los metadatos se gestionaran correctamente y que su integración con el sistema fuese fluida, garantizando así una mejora significativa en la organización y accesibilidad de los recursos del repositorio. Este enfoque en la adición de metadatos a scripts y objetos mejora la estructura y eficiencia del sistema, potenciando la experiencia del usuario y facilitando la gestión de los recursos.

5.1.3.3. Implementar una función de búsqueda avanzada.

El desarrollo de una función de búsqueda avanzada en el repositorio es una mejora significativa que permite a los usuarios buscar recursos utilizando etiquetas, nombres o tipos de recursos. Esta funcionalidad avanzada facilita la localización de recursos específicos de manera rápida y eficiente, mejorando significativamente la experiencia del usuario.

Se diseñó e implementó una herramienta de búsqueda en la interfaz de usuario, que se integra de manera fluida con la estructura de etiquetas y metadatos ya establecidos. Esta herramienta permite búsquedas por múltiples criterios, incluyendo categorías, subcategorías, autor, fecha, entre otros, y muestra resultados relevantes de una manera organizada y accesible.

La incorporación de esta función de búsqueda avanzada es un paso clave para maximizar la utilidad y accesibilidad del repositorio, asegurando que los usuarios puedan encontrar los recursos que necesitan de manera eficiente y precisa.

5.1.4. Sprint 4

Nuestro último sprint fue para elaborar la documentación del desarrollo realizado, recibir retroalimentación del servicio y finalmente lanzar al público las nuevas funcionalidades.

5.1.4.1. Elaborar documentación detallada de las funciones del módulo

Se realizó la elaboración de una documentación detallada para el módulo de repositorio de recursos. Esta documentación es esencial para proporcionar una comprensión clara de las funcionalidades desarrolladas, asegurando que los usuarios y futuros desarrolladores puedan utilizar y mantener el sistema eficazmente.

La documentación abarca todos los aspectos del módulo, incluyendo la jerarquía de etiquetas, la implementación de metadatos y la función de búsqueda avanzada. Se enfoca en explicar no solo cómo usar estas funciones, sino también cómo fueron implementadas, proporcionando así una visión completa tanto a nivel de usuario como técnico.

Se emplearon formatos y prácticas estándar de documentación técnica, asegurando que la información sea accesible, comprensible y útil para todos los usuarios del sistema. La documentación se diseñó para ser un recurso integral, facilitando la adopción y el uso efectivo del módulo de repositorio de recursos.

Esta documentación detallada, es un componente clave para el lanzamiento exitoso de las nuevas funcionalidades al público y para recibir retroalimentación constructiva que permita futuras mejoras.

5.1.4.2. Realizar encuestas a usuarios sobre la usabilidad del módulo.

Se llevó un proceso crucial de recopilación de retroalimentación de los usuarios a través de encuestas. El objetivo de estas encuestas era evaluar la usabilidad del módulo de repositorio de recursos y comprender mejor la experiencia del usuario.

Se diseñaron y distribuyeron encuestas enfocadas en aspectos claves como la facilidad de uso, la eficacia de las funciones de búsqueda y la claridad de la jerarquía de etiquetas. Estas encuestas permitieron obtener una visión valiosa de la percepción y las necesidades de los usuarios.

El análisis de las respuestas ayudó a identificar tanto las fortalezas del sistema como las áreas que requerían mejoras. Esta información fue fundamental para realizar ajustes que aumentarían

la eficiencia y satisfacción del usuario, asegurando que el módulo de repositorio de recursos no solo cumpla con los requisitos técnicos, sino que también responda eficazmente a las necesidades de los usuarios.

5.1.4.3. Lanzar el módulo para su uso en proyectos colaborativos

El último y crucial paso del Sprint 4 fue el lanzamiento del módulo para su uso en proyectos colaborativos. Este lanzamiento marca la culminación de nuestro trabajo de desarrollo, llevando nuestras mejoras y nuevas funcionalidades al entorno real donde los usuarios pueden empezar a aplicarlas en sus proyectos.

La preparación para este lanzamiento incluyó pruebas finales, asegurando que todas las características funcionaran de manera óptima. Se realizó un esfuerzo coordinado para comunicar el lanzamiento a los usuarios potenciales, incluyendo la presentación de las funcionalidades del módulo y las instrucciones de uso.

Este lanzamiento no solo representa la finalización exitosa del proyecto, sino también el comienzo de una fase de uso y colaboración real, abriendo nuevas posibilidades para el trabajo colaborativo y el intercambio de conocimientos.

Nota:

Para una comprensión más profunda y detallada de las funcionalidades del módulo, se ha incluido una documentación exhaustiva en los anexos de este trabajo. En estos anexos, los lectores encontrarán guías de usuario, descripciones técnicas y ejemplos de código que complementan y amplían la información presentada en esta sección. Se recomienda consultar estos anexos para obtener una visión completa y detallada del módulo de repositorio de recursos.

6. Resultados

En esta sección, nos enfocaremos en los resultados finales del desarrollo, poniendo especial atención en la forma en que todos los servicios creados son consumidos desde un cliente web. Nuestro objetivo es demostrar la interacción efectiva entre el cliente y los servicios, destacando la funcionalidad y eficiencia del sistema en un entorno real.



Figura 11: Página principal del repositorio

La Figura 11 muestra la página principal del repositorio, diseñada para ofrecer una navegación intuitiva al usuario. Desde aquí, es posible acceder directamente a las secciones de mayor interés, incluyendo etiquetas, repositorios, recursos y juegos. Esta disposición facilita la exploración y el acceso rápido a las diferentes áreas del sitio.

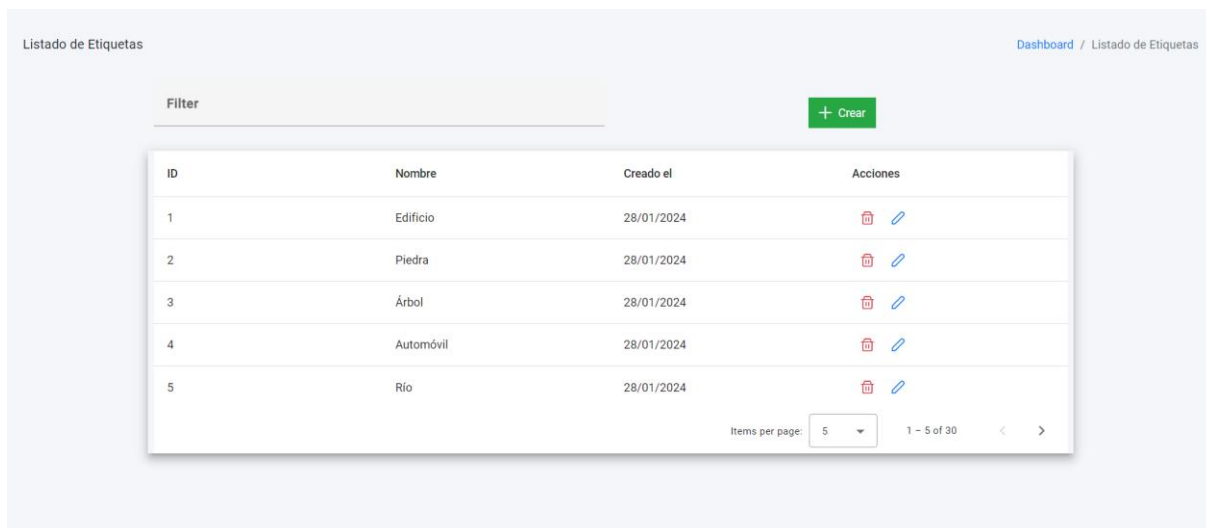


Figura 12: Página del CRUD de las etiquetas

La Figura 12 muestra la interfaz para realizar operaciones CRUD en las etiquetas, accesible únicamente por usuarios con roles de administrador o desarrollador. Los usuarios comunes solo pueden visualizar las etiquetas. Esta página presenta una tabla paginada con funciones de filtrado por nombre y ordenación por cualquier campo de datos. Además, incluye botones para crear, eliminar y editar etiquetas, mejorando así la gestión y el manejo de la información. Con esta interfaz hemos completado el objetivo de crear etiquetas para facilitar la búsqueda y el acceso a los datos.

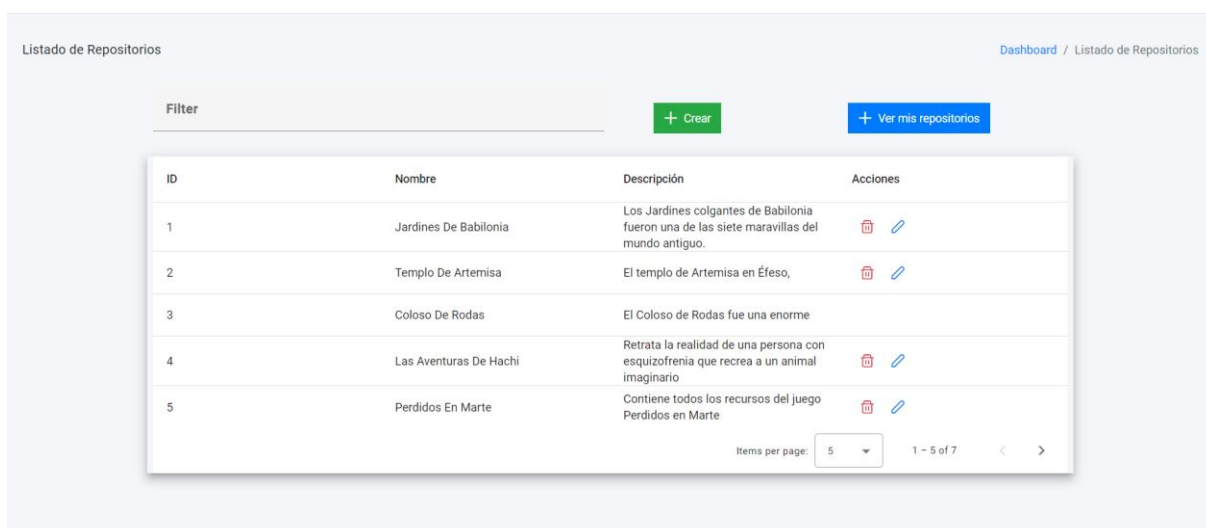


Figura 13: Página del CRUD de los repositorios

La Figura 13 muestra la interfaz para el CRUD de los repositorios, la cual es muy similar a la de las etiquetas. Sin embargo, esta página se distingue por incluir un botón adicional que permite filtrar exclusivamente los repositorios del usuario. Además, se observa que las opciones de editar o eliminar están condicionadas a que el usuario sea el propietario del repositorio.

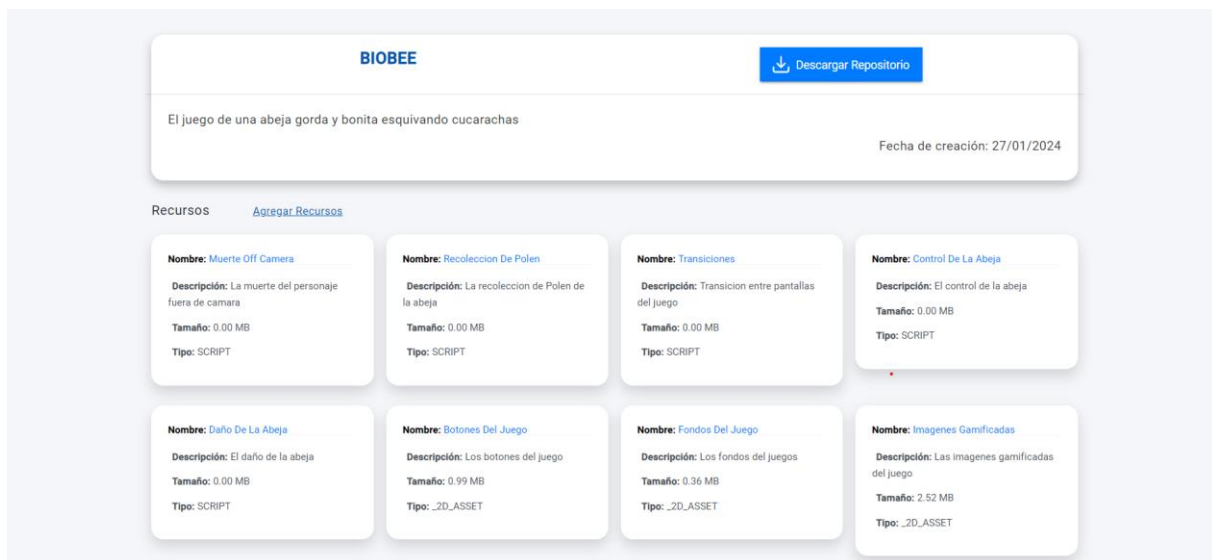


Figura 14: Página individual de cada repositorio

En la Figura 14 se puede observar la página diseñada para ofrecer una vista detallada del repositorio. Los propietarios del repositorio tienen la capacidad de añadir más recursos, mientras que el botón de descarga está accesible para todos los usuarios, permitiendo la obtención de contenidos sin restricciones de propiedad. Con esta interfaz hemos cumplido el objetivo de agregar metadatos a los diversos objetos que podemos manipular en el repositorio.

Figura 15: Página crear un recurso

En la Figura 15 se puede ver la página diseñada para la creación de recursos de manera individual. Se solicita a los usuarios ingresar toda la información relevante necesaria para completar esta tarea. La posibilidad de subir diferentes tipos de archivos, así como la activación o desactivación de ciertas opciones, variará en función del tipo de recurso que se esté gestionando.

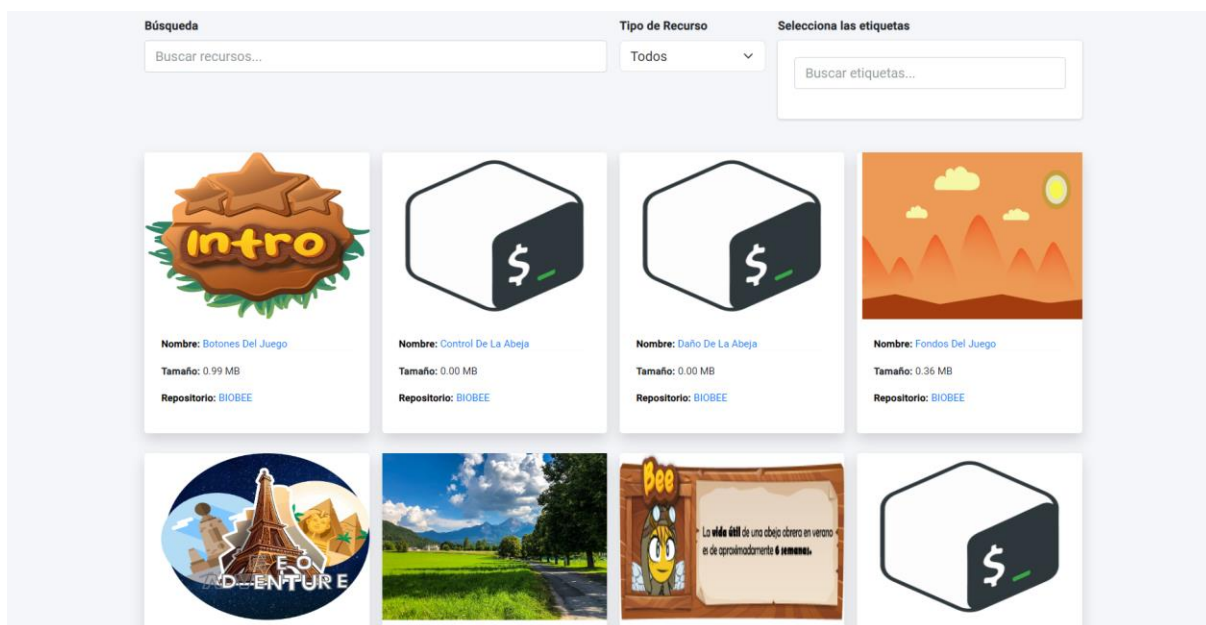


Figura 16: Página visualizar los recursos

Tal como se puede observar en la Figura 16 en esta página seremos capaces de visualizar todos los recursos disponibles en el repositorio. Así mismo podremos filtrarlos en base al nombre, al tipo de recurso y también mediante etiquetas. Con esta interfaz hemos completado el objetivo de ofrecer una búsqueda avanzada, ya que podemos obtener los recursos ya sea por su nombre, el tipo de recurso y las etiquetas de este.

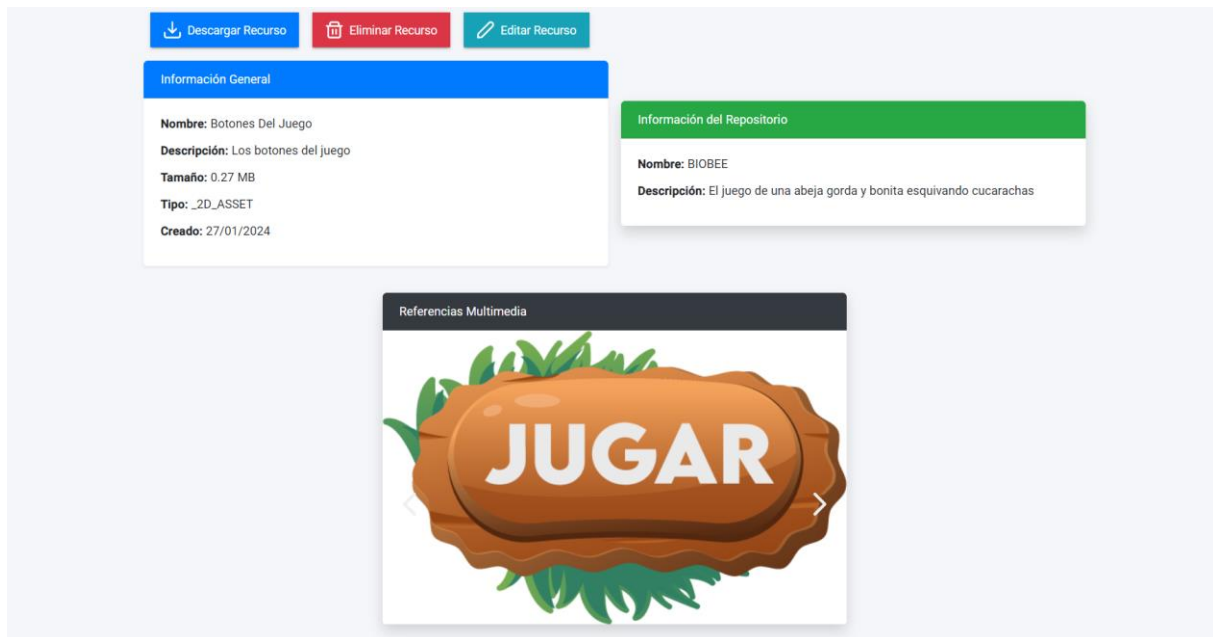


Figura 17: Página individual de cada recurso

La Figura 17 presenta la interfaz diseñada para mostrar información detallada de cada recurso. Dependiendo del tipo de archivos subidos, los usuarios pueden visualizar diferentes formatos de contenido multimedia, tales como imágenes, videos o audios. Las funcionalidades de edición o eliminación están exclusivamente habilitadas para los propietarios de los recursos. Por otro lado, el botón para descargar está accesible para todos los usuarios, facilitando la obtención de los contenidos.

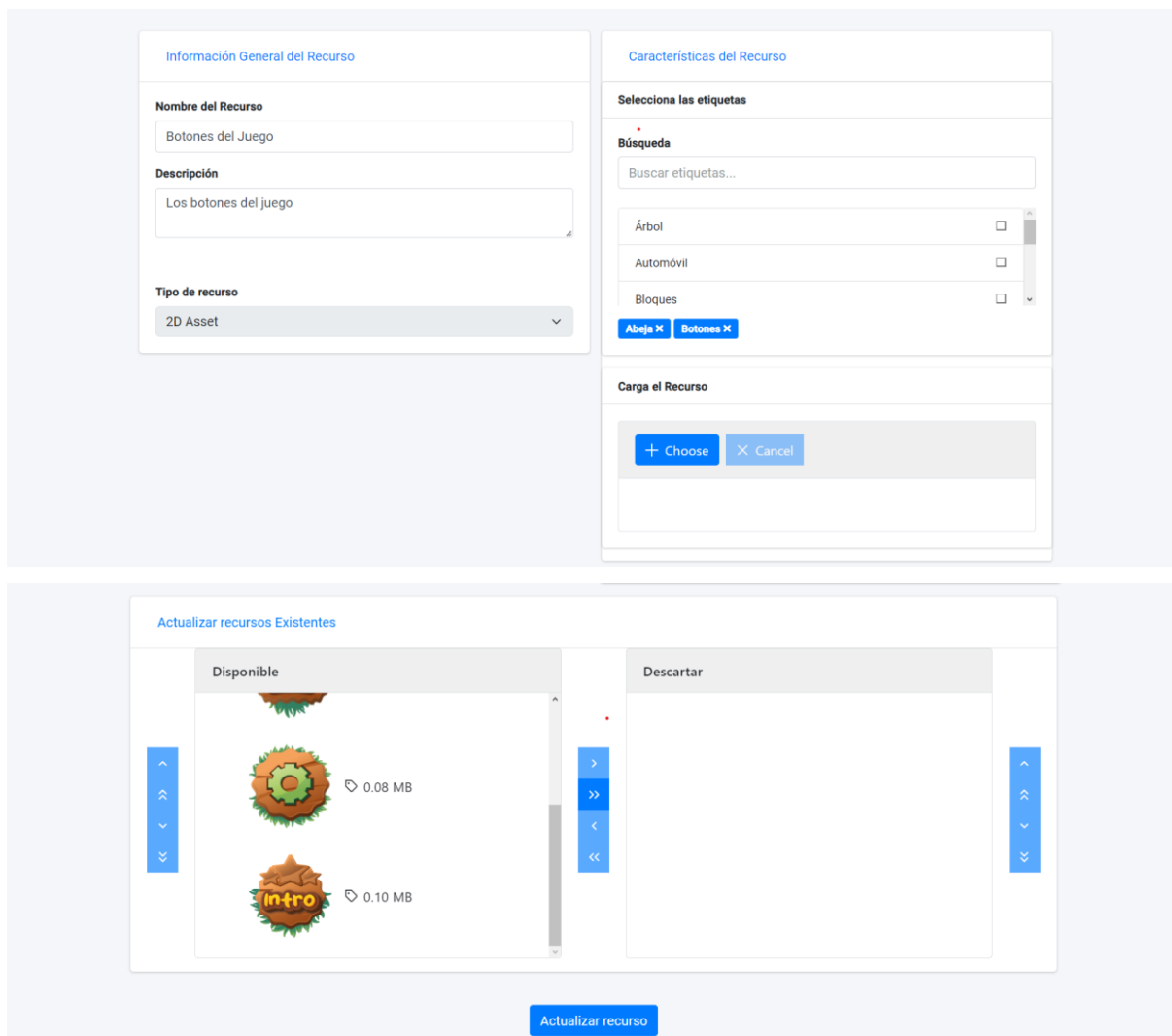


Figura 18: Página para editar un recurso

La Figura 18 muestra la interfaz utilizada para editar recursos existentes. Esta página es similar a la de creación de recursos, con la diferencia de que ya se muestran los datos previamente ingresados. Además, esta versión de la interfaz incluye dos listas: una muestra los recursos multimedia ya cargados, los cuales pueden ser seleccionados para eliminación y movidos a la segunda lista. Para confirmar las eliminaciones y aplicar los cambios, los usuarios simplemente deben hacer clic en el botón de actualización.

The image shows a web form for creating a game, divided into several sections:

- Información General del Juego:** Contains a text input for 'Nombre del Juego', a larger text area for 'Descripción', and a dropdown menu for 'Tipo de recurso' which is currently set to 'Juego'.
- Características del Juego:** Includes a section 'Selecciona las etiquetas' with a search box 'Búsqueda' containing the text 'Buscar etiquetas...'. Below it is a list of tags: 'Abeja', 'Árbol', and 'Automóvil', each with an unchecked checkbox. A note below the list says 'Seleccionar una etiqueta como mínimo'.
- Carga el Juego y Manual:** Features a '+ Choose' button and a 'Cancel' button above a file upload area.
- Carga imágenes, videos o gifs del GAME:** Also features a '+ Choose' button and a 'Cancel' button above another file upload area.

At the bottom center of the form is a blue button labeled 'Guardar juego'.

Figura 19: Página para crear un juego

La Figura 19 ilustra la interfaz para la creación de juegos. Esta página, que comparte similitudes con la de creación de otros recursos, difiere en que el tipo de recurso ya está predefinido y no se puede modificar. Además, se requiere que los usuarios carguen el juego en formato .zip y proporcionen un manual o informe del proyecto en formato .pdf. Para complementar la presentación, se debe subir material multimedia que sirva como referencia visual para los usuarios, permitiéndoles obtener una idea clara del juego y su temática.

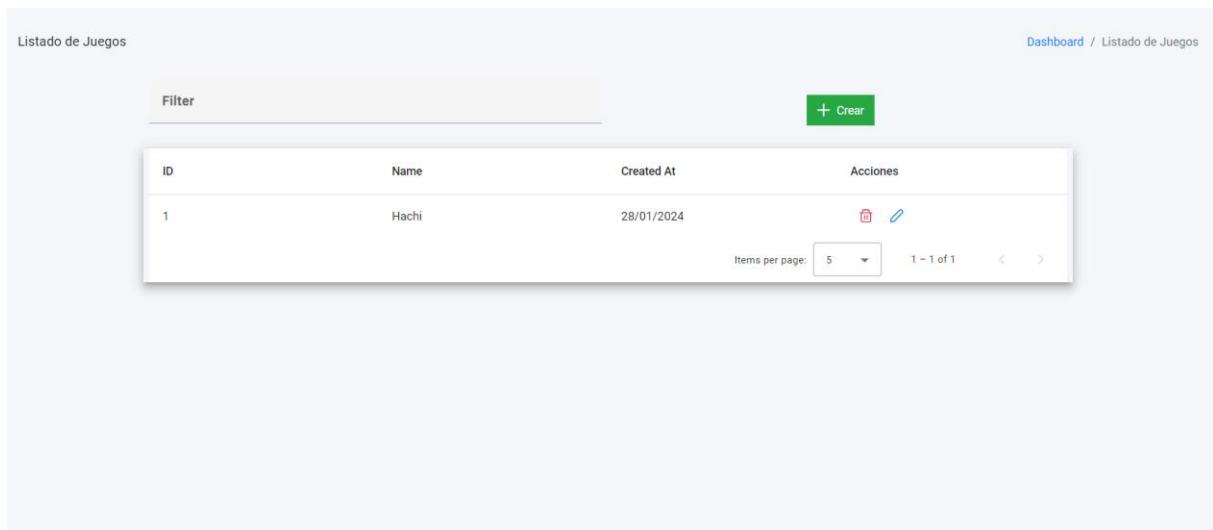


Figura 20: Página para ver todos los juegos

La Figura 20 muestra la página donde se pueden visualizar todos los juegos disponibles. Esta página comparte características similares con las páginas de etiquetas y repositorios, pero se diferencia en que los botones para 'Crear' o 'Editar' redirigen a páginas específicas para estas acciones en el contexto de los juegos. Así, los usuarios cuentan con una interfaz intuitiva y especializada para la gestión de juegos dentro de la plataforma.

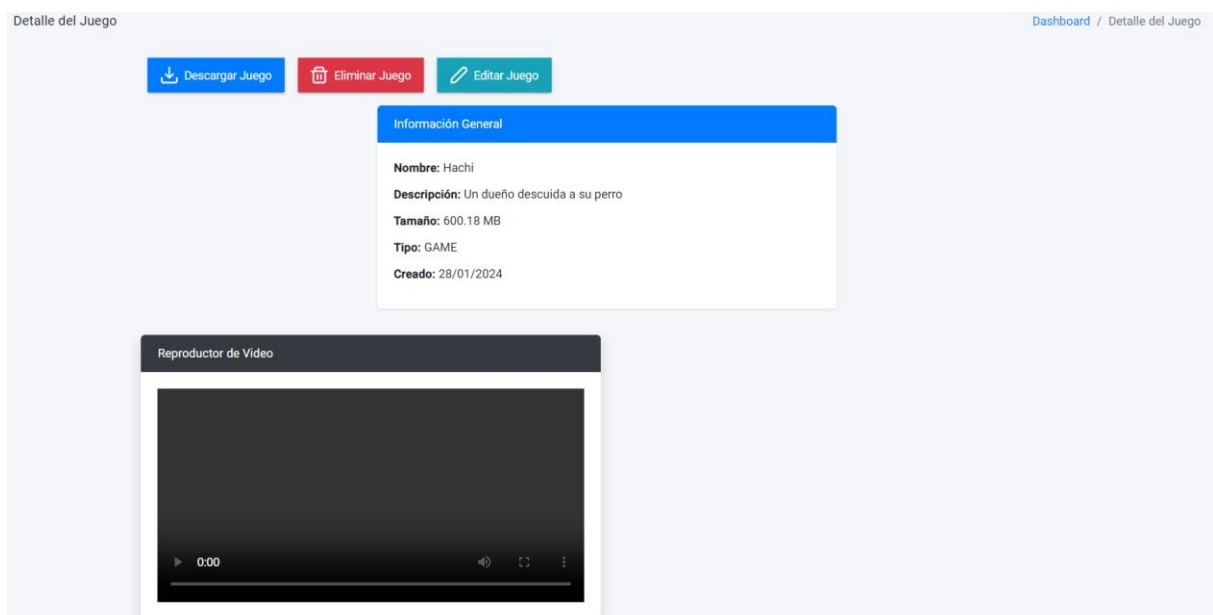


Figura 21: Página para ver los detalles del juego

La Figura 21 muestra la interfaz para visualizar los detalles de un juego específico. En esta página, los usuarios tienen la capacidad de descargar, editar o eliminar el juego, dependiendo de los permisos asignados tanto a la aplicación como al recurso en cuestión.

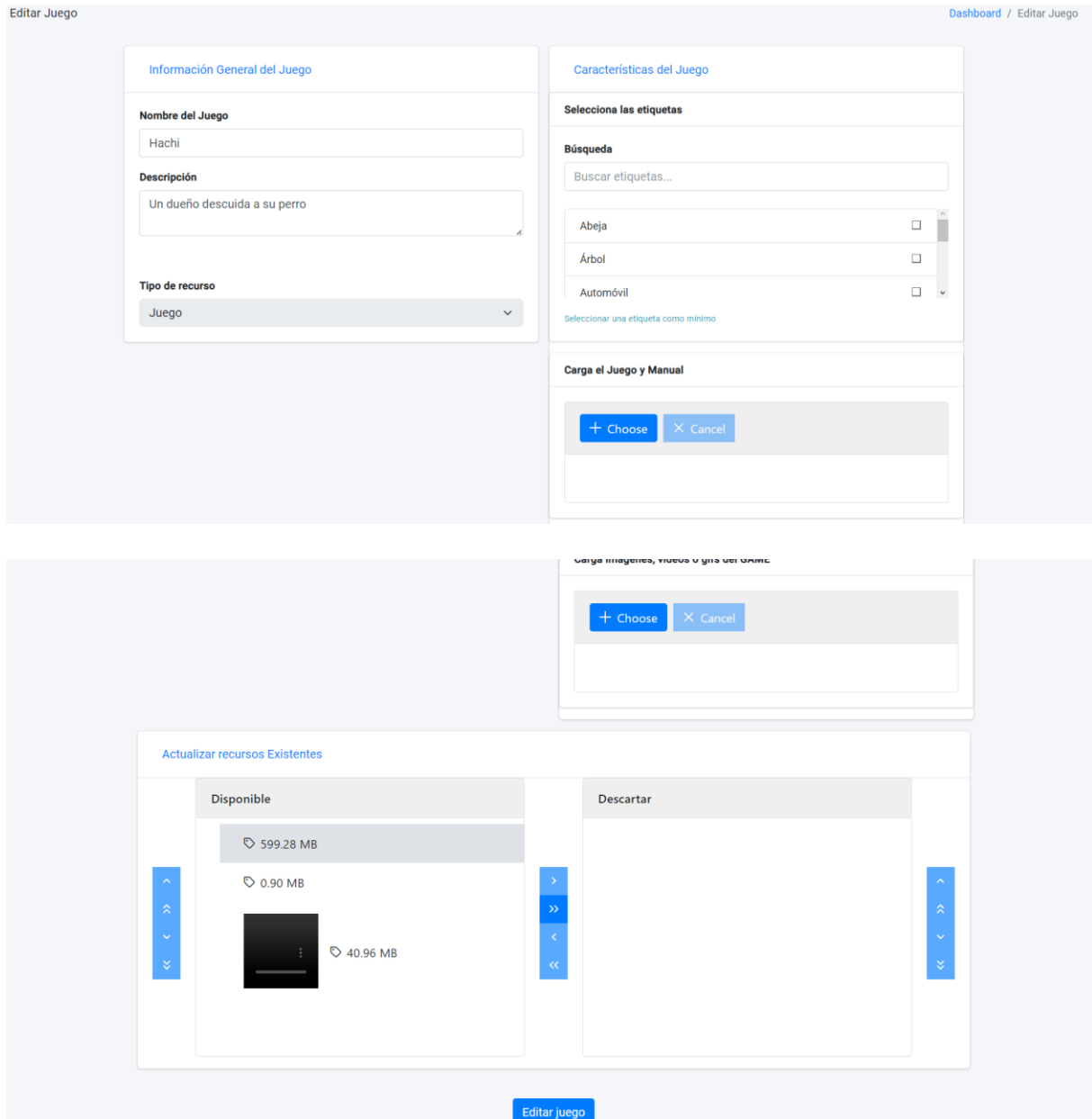


Figura 22: Página para editar un juego

La Figura 22 ilustra la página de edición de un juego, donde es posible modificar diversos aspectos relacionados con el mismo. Esta funcionalidad permite tanto eliminar como agregar recursos adicionales, ofreciendo así flexibilidad y control completo sobre la configuración y el contenido del juego.

7. Cronograma

A continuación, se presentan las diferentes actividades realizadas por cada objetivo.

Tabla 1: Actividades Realizadas en el Objetivo Especifico 1

OE1	Actividad
1	Reunirse con el equipo de desarrollo y los interesados para comprender las necesidades y expectativas del proyecto.
2	Examinar documentación previa sobre la plataforma Edutainment.
3	Realizar una investigación exhaustiva y recopilar las mejores prácticas en la gestión de contenido similar en otras plataformas educativas o de entretenimiento.

Tabla 2: Actividades Realizadas en el Objetivo Especifico 2

OE2	Actividad
1	Investigar y recopilar las mejores prácticas de diseño de interfaz que se adapten a las necesidades de ambos grupos de usuarios.
2	Definir la arquitectura técnica del módulo de repositorio, incluyendo la estructura de almacenamiento y acceso a los recursos.
3	Crear prototipos de la interfaz de usuario que permitan visualizar la estructura y la disposición de los elementos en la plataforma.

Tabla 3: Actividades Realizadas en el Objetivo Especifico 3

OE3	Actividad
1	Basado en la arquitectura definida en el OE2, realizar un diseño detallado del módulo de repositorio, incluyendo la estructura de la base de datos, las rutas de acceso y la interfaz de usuario.
2	Crear la base de datos que albergará los scripts y objetos de diseño, definiendo las tablas y relaciones necesarias.
3	Desarrollar funciones de carga que permitan a los usuarios cargar scripts y objetos de diseño en el repositorio. Asegurarse de que las cargas sean seguras y validen los tipos de archivo.

4	Implementar funciones que permitan a los usuarios descargar los scripts y objetos de diseño almacenados en el repositorio, garantizando un proceso de descarga sencillo y eficaz.
5	Crear la interfaz de usuario que permita a los usuarios navegar y acceder a los scripts y objetos de diseño en el repositorio.
6	Desarrollar funciones de visualización que permitan a los usuarios ver los scripts y objetos de diseño directamente en la plataforma.

Tabla 4: Actividades Realizadas en el Objetivo Especifico 4

OE4 Actividad

1	Crear una taxonomía que organice las etiquetas y categorías en una jerarquía lógica, lo que facilita la búsqueda y navegación.
2	Ofrecer una lista de etiquetas predeterminadas que los usuarios pueden elegir al cargar contenido para mantener la consistencia en la categorización.
3	Agregar metadatos a los scripts y objetos de diseño, incluyendo información relevante como autor, fecha de creación y tema, para facilitar la organización y búsqueda.
4	Implementar una función de búsqueda avanzada que permita a los usuarios buscar por etiquetas, metadatos y otros criterios, y que muestre resultados relevantes.

Tabla 5: Actividades Realizadas en el Objetivo Especifico 5

OE5 Actividad

1	Realizar pruebas funcionales para verificar que todas las funciones del módulo (carga, descarga, visualización, etiquetado, etc.) funcionen según lo esperado.
2	Evaluar la facilidad de uso del módulo a través de pruebas de usabilidad con usuarios reales, identificando posibles puntos de confusión o dificultad.
3	Probar la integración del módulo de repositorio con otros componentes de la plataforma para asegurarse de que funcione sin problemas.
4	Evaluar el rendimiento del módulo al simular cargas de usuarios y volúmenes de datos significativos, optimizando el tiempo de respuesta y la escalabilidad.

Tabla 6: Actividades Realizadas en el Objetivo Especifico 6

OE6 Actividad

1	Crear material de capacitación adecuado para la audiencia, que puede incluir manuales, tutoriales y guías rápidas.
2	Elaborar documentación detallada que describa cómo utilizar las funciones del módulo, incluyendo carga, descarga, búsqueda, etiquetado y gestión de permisos.

Tabla 7: Actividades Realizadas en el Objetivo Especifico 7

OE7 Actividad

1	Recolectar datos de referencia sobre la colaboración interdisciplinaria y la calidad de los proyectos antes de implementar el módulo.
2	Lanzar el módulo de repositorio y permitir que los usuarios comiencen a utilizarlo en sus proyectos colaborativos.
3	Realizar encuestas y entrevistas a usuarios para obtener una comprensión cualitativa de cómo el módulo ha influido en su colaboración y en la calidad de sus proyectos.
4	Utilizar la retroalimentación de los usuarios y los resultados de las evaluaciones para realizar ajustes y mejoras en el módulo de repositorio.

Total de horas: 400

Horas completadas por Alvarez Valdez Michael Andres (MA): 200 horas

Horas completadas por Llivicota León Jhon Santiago (JL): 200 horas

Fecha de Inicio: lunes 30-10-2023

Fecha de Finalización: martes 23-01-2024

Tabla 8: Cronograma por Cada responsable

Objetivo	Actividad	Fecha Inicio	Fecha Fin	Horas por responsable	Responsables
OE1	Ac. 1	30/10/2023	31/10/2023	5	MA y JL
	Ac. 2	31/10/2023	01/11/2023	5	MA y JL
	Ac. 3	02/11/2023	03/11/2023	5	MA y JL
OE2	Ac. 1	04/11/2023	06/11/2023	5	MA y JL
	Ac. 2	06/11/2023	11/11/2023	10	MA y JL
	Ac. 3	12/11/2023	15/11/2023	10	MA y JL
OE3	Ac. 1	16/11/2023	30/11/2023	50	JL
	Ac. 2	1/12/2023	5/12/2023	10	JL
	Ac. 3	16/11/2023	22/11/2023	35	MA
	Ac. 4	22/11/2023	30/11/2023	35	MA
	Ac. 5	6/12/2023	9/12/2023	5	JL
	Ac. 6	9/12/2023	11/12/2023	5	JL
OE4	Ac. 1	12/12/2023	14/12/2023	10	MA
	Ac. 2	12/12/2023	14/12/2023	5	JL
	Ac. 3	14/12/2023	18/12/2023	5	MA
	Ac. 4	14/12/2023	18/12/2023	10	JL
OE5	Ac. 1	19/12/2023	24/12/2023	15	MA Y JL
	Ac. 2	24/12/2023	01/01/2024	10	MA
	Ac. 3	01/01/2024	05/01/2024	10	MA
	Ac. 4	24/12/2023	01/01/2024	10	JL
OE6	Ac. 1	01/01/2024	05/01/2024	10	JL
	Ac. 2	05/01/2024	09/01/2024	10	MA
OE7	Ac. 1	05/01/2024	09/01/2024	10	JL
	Ac. 2	09/01/2024	10/01/2024	5	MA Y JL
	Ac. 3	10/01/2024	15/01/2024	8	MA Y JL
	Ac. 4	15/01/2024	23/01/2024	17	MA Y JL

Total en horas: 400

8. Presupuesto

Tabla 9: Presupuesto del Proyecto Técnico

Denominación	Cantidad	Costo Unitario	Costo Total
1. Bienes			
Copias	25	0.05	1.25
Libros	2	25.00	50.00
2. Tecnológico			
Computadora portátil	2	1100.00	2200.00
Celular	2	600	1200.00
3. Servicios			
Servicio de transporte	15	1.80	27.00
Servicios de Internet	3 meses	30.00	90.00
4. Personal			
Estudiante	200 horas (2 estudiantes)	5 por hora	2000.00
5. Otros			
Imprevistos	1	100.00	100.00
TOTAL:			5668.25

9. Conclusiones

Este trabajo de titulación, centrado en el desarrollo e implementación de un módulo de repositorio de recursos colaborativos, ha cumplido con los objetivos propuestos, demostrando la viabilidad y eficacia de las soluciones tecnológicas empleadas. La integración de tecnologías como Spring, Angular y PostgreSQL, junto con la metodología SCRUM, ha resultado en un sistema robusto, funcional y amigable para el usuario.

Las funcionalidades desarrolladas, incluyendo la jerarquía de etiquetas, la incorporación de metadatos y la búsqueda avanzada, han mejorado significativamente la accesibilidad y gestión de los recursos. La retroalimentación de los usuarios ha sido fundamental para refinar estas funcionalidades, asegurando que el sistema no solo cumpla con los requisitos técnicos, sino que también satisfaga las necesidades reales de los usuarios.

Este proyecto contribuye al campo de la gestión de recursos digitales, ofreciendo un modelo replicable y eficiente para otras instituciones y proyectos. Las lecciones aprendidas y los conocimientos adquiridos durante este trabajo sientan las bases para futuras investigaciones y desarrollos en la misma línea.

En conclusión, este trabajo representa un significativo avance en la compartición de recursos de gamificación, realidad virtual y realidad aumentada, elementos claves en la transformación del entorno educativo y turístico. La implementación de este módulo en la institución marca un hito en el ámbito del diseño multimedia, facilitando la subida y descarga de estos recursos innovadores. Esto no solo optimiza el proceso actual, sino que también establece una base sólida para la realización de futuros proyectos, promoviendo la reutilización y la eficiencia en el manejo de recursos en estos campos emergentes. Además, que este trabajo demuestra que con un enfoque metodológico y la integración adecuada de tecnologías, es posible crear soluciones efectivas que mejoren la colaboración y el cambio de conocimientos en entornos educativos y de investigación.

10. Recomendaciones

Continuar la expansión de recursos, dada la naturaleza dinámica de la gamificación, la realidad virtual y la realidad aumentada, se recomienda la continua actualización y expansión del repositorio de recursos. Esto incluye la incorporación de nuevas herramientas y tendencias en diseño multimedia.

Fomentar la colaboración interdisciplinaria, se sugiere promover proyectos que involucren colaboración entre diferentes áreas académicas, aprovechando la versatilidad del módulo para enriquecer el aprendizaje y la investigación.

Capacitación y talleres, implementar programas de capacitación para estudiantes y docentes sobre el uso efectivo de estos recursos en contextos educativos y turísticos, potenciando su aplicación práctica.

Evaluación continua y retroalimentación, establecer mecanismos de evaluación y retroalimentación continuos para adaptar y mejorar el módulo según las necesidades y avances tecnológicos.

Expansión del módulo, se recomienda explorar la posibilidad de expandir el uso del módulo de repositorio de recursos más allá de la universidad, hacia otras instituciones educativas y organizaciones interesadas en gamificación, realidad virtual y realidad aumentada. Esto puede lograrse a través de alianzas estratégicas y programas de colaboración, compartiendo conocimientos y recursos, y fomentando un ecosistema de aprendizaje e innovación más amplio. La expansión externa no solo incrementaría el impacto del proyecto, sino que también facilitaría el intercambio interinstitucional de ideas y tecnologías avanzadas.

Referencias Bibliográficas

Borrás Gené, O. (2015). Fundamentos de gamificación.

Ortiz-Colón, A. M., Jordán, J., & Agredal, M. (2018). Gamificación en educación: una panorámica sobre el estado de la cuestión. *Educação e pesquisa*, 44.

Gutiérrez, A. M. L. (2020). ¿Un mundo nuevo? Realidad virtual, realidad aumentada, inteligencia artificial, humanidad mejorada, Internet de las cosas. *arbor*, 196(797), a572-a572.

Burbano Erazo, R. D. (2022). *Interfaz de realidad aumentada para recomendador de sitios turísticos en Quito* (Bachelor's thesis, Quito: EPN, 2022).

Zambrano, R. L. C., Romero, M. E. Y., Dávila, K. E. D., & Balarezo, C. E. B. (2023). Realidad virtual y aumentada en la educación superior: experiencias inmersivas para el aprendizaje profundo. *Religación. Revista de Ciencias Sociales y Humanidades*, 8(37).

Contreras, D. A. B. (2018). Arquitectura de microservicios. *Tecnología Investigación y Academia*, 6(1), 36-46.

López, D. R. (2021). Archivos, repositorios y bibliotecas digitales: Hacia una historia digital de la Edad de Plata. *Signa: Revista de la Asociación Española de Semiótica*, (30), 17-30.

Angular Team. (s.f.). What Is Angular? Angular.

<https://angular.io/guide/what-is-angular>

González, G. M. (2016). *Aprende a Desarrollar con Spring Framework: 2ª Edición*. IT Campus Academy.

The PostgreSQL Global Development Group. (s.f.). PostgreSQL 16.1 Documentation. <https://www.postgresql.org/docs/>

Miell, I., & Sayers, A. (2019). *Docker in practice*. Simon and Schuster.

Docker, I. (2020). Docker. *linea*. [Junio de 2017]. Disponible en: <https://www.docker.com/what-docker>.

Baeldung, & Baeldung. (2024, 8 enero). Introduction to Spring Cloud Netflix - Eureka | Baeldung. Baeldung. <https://www.baeldung.com/spring-cloud-netflix-eureka>

OpenZipkin (s.f.). A distributed tracing system. <https://zipkin.io/>

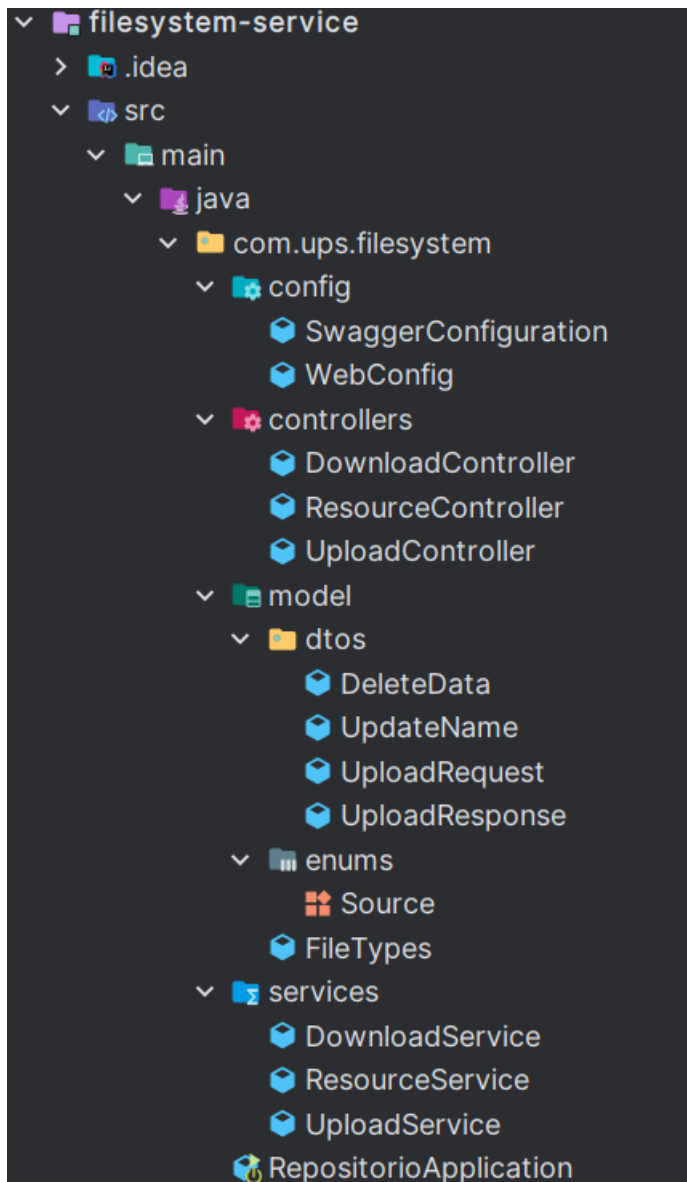
Montero, B. M., Cevallos, H. V., & Cuesta, J. D. (2018). Metodologías ágiles frente a las tradicionales en el proceso de desarrollo de software. *Espiraes revista multidisciplinaria de investigación*, 2(17), 114-121.

Schwaber, K., & Sutherland, J. (2020). La guía definitiva de Scrum: las reglas del juego.

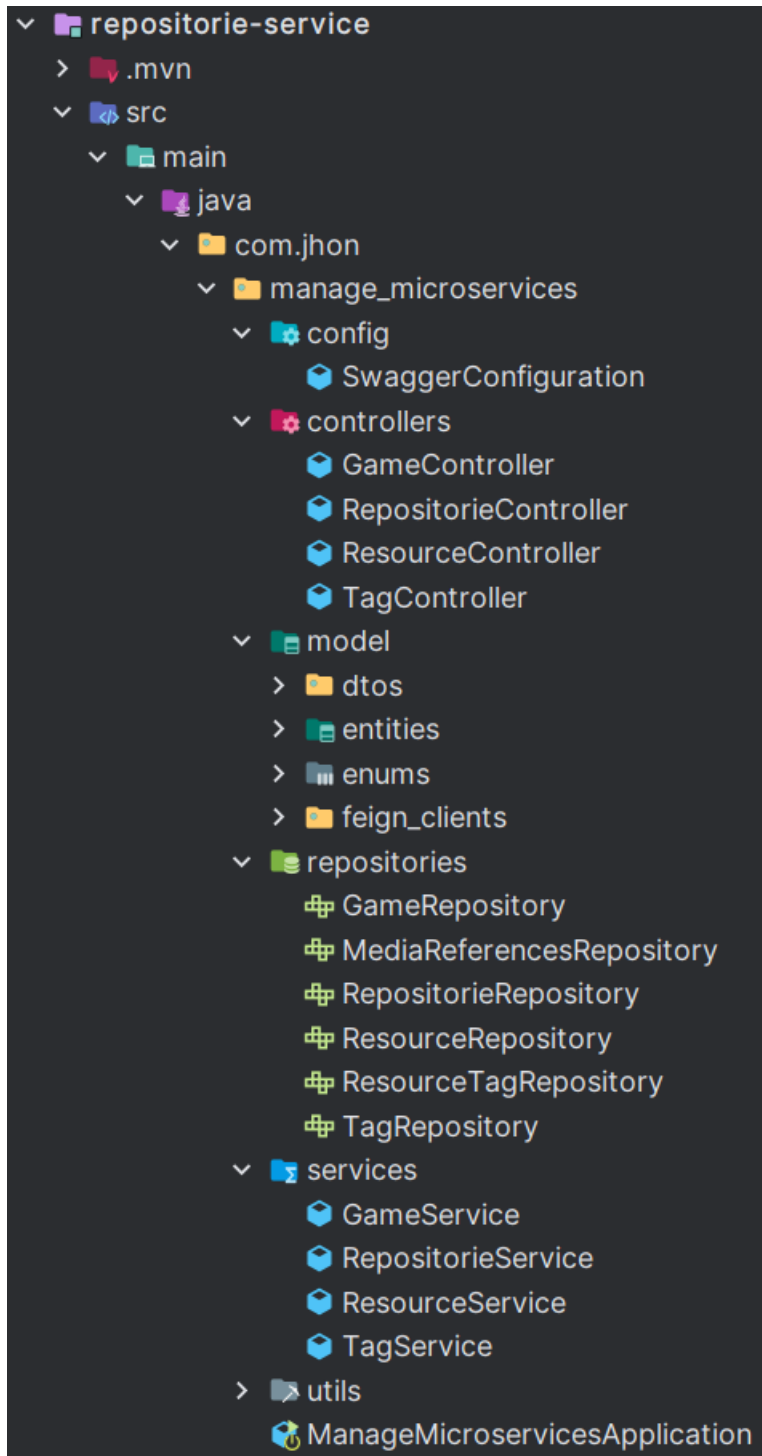
Anexos

Estructura del proyecto

1. Estructura del microservicio encargado de la carga y descarga de archivos. Como se puede observar el proyecto cuenta con varias carpetas y archivos apropiadamente nombrados y separados según su funcionalidad.



2. Estructura del microservicio encargado de trabajar con la base de datos para asegurar la persistencia de los datos. Nuevamente, las carpetas y archivos están apropiadamente nombrados con lo cual se puede inferir la funcionalidad de cada uno rápidamente.



Conectividad entre Microservicios

3. Microservicios registrados en Eureka

Instances currently registered with Eureka

Application	AMIs	Availability Zones	Status
API-GATEWAY	n/a (1)	(1)	UP (1) - api-gateway:1e1aed62a329b655b5e1cb88aae1f6e2
FILESYSTEM-SERVICE	n/a (1)	(1)	UP (1) - filesystem-service:1c7c953e9ed596767bf4307b69fc7455
REPOSITORIE-SERVICE	n/a (1)	(1)	UP (1) - repositorie-service:bf5d4c4d4ec8b9296c1d79711f673e67

4. Trazabilidad con Zipkin

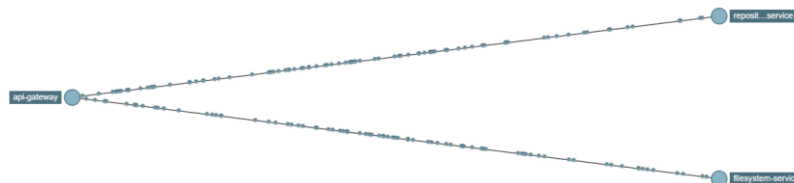
Zipkin Find a trace Dependencies ENGLISH Search by trace ID

10 Results EXPAND ALL COLLAPSE ALL Service filters

Root	Start Time	Spans	Duration	
api-gateway: http delete	5 minutes ago (01/28 12:43:41:510)	3	234.440ms	SHOW
api-gateway: http post	5 minutes ago (01/28 12:43:03:860)	3	119.576ms	SHOW
api-gateway: http get	5 minutes ago (01/28 12:43:41:780)	3	91.664ms	SHOW
api-gateway: http post	5 minutes ago (01/28 12:43:03:790)	3	58.185ms	SHOW
api-gateway: http post	5 minutes ago (01/28 12:43:03:798)	3	55.213ms	SHOW
api-gateway: http get	5 minutes ago (01/28 12:43:04:030)	3	33.352ms	SHOW
api-gateway: http get	5 minutes ago (01/28 12:43:07:536)	3	24.019ms	SHOW
filesystem-service: http post /api/data/delete/resource	5 minutes ago (01/28 12:43:41:683)	1	10.573ms	SHOW
api-gateway: http options	5 minutes ago (01/28 12:43:41:507)	1	1.403ms	SHOW
api-gateway: http options	5 minutes ago (01/28 12:43:03:857)	1	1.168ms	SHOW

Start Time: 01/27/2024 13:00:07 - End Time: 01/28/2024 13:00:07 RUN QUERY

Filter



- Para establecer la conexión entre microservicios usando OpenFeign, se crea una interfaz que facilita la comunicación con el servicio deseado, identificado por su nombre registrado en Eureka. Esta interfaz define los endpoints y especifica el método de la solicitud HTTP, así como los parámetros requeridos para cada operación. Con esta configuración, OpenFeign permite una integración sencilla y eficiente entre diferentes servicios, aprovechando las capacidades de descubrimiento de servicios de Eureka y simplificando las llamadas a otros microservicios.

```
6 usages
@FeignClient(name = "FILESYSTEM-SERVICE")
public interface FeignClientFilesystem {

    @PostMapping("/api/data/delete/resource")
    public ResponseEntity<String> deleteResource(@RequestBody DeleteData deleteData);

    @PostMapping("/api/data/delete/repositorie")
    public ResponseEntity<String> deleteRepo(@RequestBody DeleteData deleteData);

    @PutMapping("/api/data/update/resource")
    public ResponseEntity<String> updateResource(@RequestBody DeleteData deleteData);

    @PutMapping("/api/data/update-names")
    public ResponseEntity<String> updateName(@RequestBody UpdateName updateName);

}
```

- Hacemos la inyección de dependencias y llamamos al método.

```
private final FeignClientFilesystem clientFilesystem;

// ACTUALIZAR LAS URL EN EL FILESYSTEM SERVICE PORQUE HAN CAMBIADO
if (isUrlChanged){
    this.clientFilesystem.updateResource(DeleteData.builder()
        .repositorie_name(null)
        .resource_name(oldName)
        .urls(mediaReferencesList.stream() Stream<MediaReferences>
            .map(MediaReferences::getMediaReferenceUrl) Stream<String>
            .toList())
        .build());
}
```

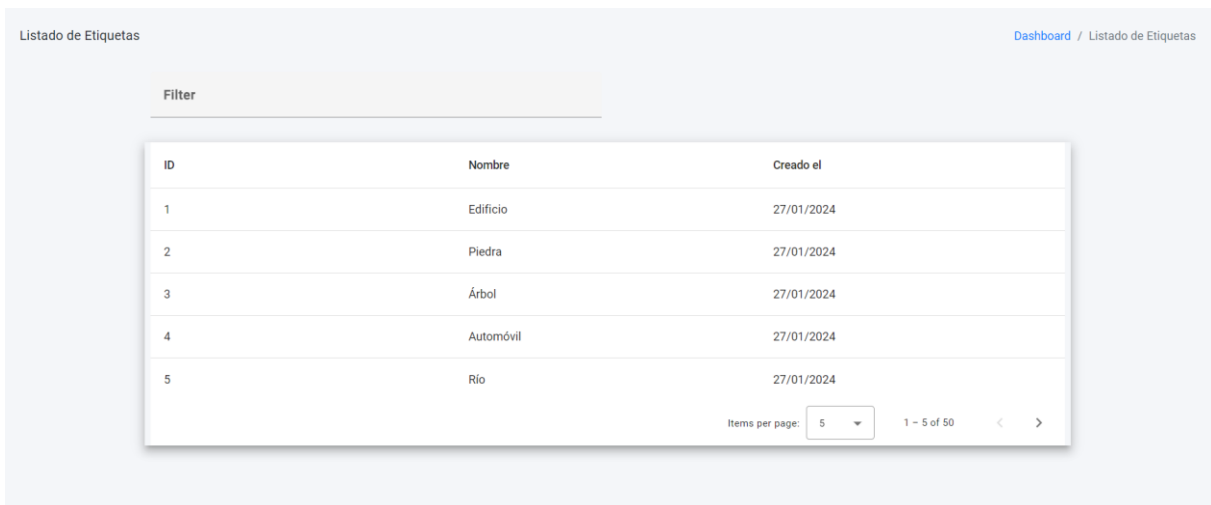
Manual de Usuario

La página en sí es muy simple de usar y aquí vamos a presentar un flujo de trabajo común en el cual un usuario entra a la página principal y realiza una serie de acciones como ver todas las etiquetas, crear un repositorio, ver entrar al repositorio, agregar recursos, visualizar todos los recursos, visualizar todos los recursos, crear un juego, visualizar todos los juegos y editar los recursos y juegos que haya creado.

1. Usuario en la página principal.

















2. Usuario viendo el listado de etiquetas.



3. Usuario viendo sus repositorios.

Listado de Repositorios Dashboard / Listado de Repositorios

Filter + Crear + Ver todos los repositorios

ID	Nombre	Descripción	Acciones
1	Jardines De Babilonia	Los Jardines colgantes de Babilonia fueron una de las siete maravillas del mundo antiguo.	 
2	Templo De Artemisa	El templo de Artemisa en Éfeso,	 
4	Las Aventuras De Hachi	Retrata la realidad de una persona con esquizofrenia que recrea a un animal imaginario	 
5	Perdidos En Marte	Contiene todos los recursos del juego Perdidos en Marte	 
7	Variedad De Rocas	Contiene una vasta cantidad de rocas listas para usarse	 
8	Biobee	El juego de una abeja gorda y bonita esquivando cucarachas	 
9	Geoadventure	GeoAdventure	 

Items per page: 100 1 - 9 of 9

4. Usuario en la página individual del repositorio.

Detalle del Repositorio Dashboard / Detalle del Repositorio

Jardines de Babilonia Descargar Repositorio

Los Jardines colgantes de Babilonia fueron una de las siete maravillas del mundo antiguo.

Fecha de creación: 27/01/2024

Recursos [Agregar Recursos](#)

Nombre: Personajes

Descripción: Personajes

Tamaño: 2.67 MB

Tipo: _2D_ASSET

Nombre: Geoadventure

Descripción: geoadventure

Tamaño: 14.03 MB

Tipo: _3D_OBJECT

Nombre: Hola

Descripción: hola

Tamaño: 0.13 MB

Tipo: _3D_OBJECT

5. Usuario en la página para agregar más recursos al repositorio.

Crear Recurso Dashboard / Crear Recurso

Información General del Recurso

Nombre del Recurso

Descripción

Tipo de recurso
Selecciona el tipo

Características del Recurso

Selecciona las etiquetas

Búsqueda

- Abeja
- Árbol
- Automóvil

Seleccionar una etiqueta como mínimo

Carga el Recurso


6. Usuario en la página de todos los recursos.

Listado de Recursos Dashboard / Listado de Recursos


Búsqueda

Tipo de Recurso
Todos


Selecciona las etiquetas



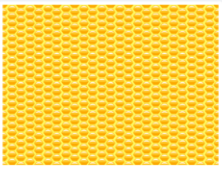
Nombre: Botones Del Juego
Tamaño: 0.27 MB
Repositorio: BIOBEE



Nombre: Control De La Abeja
Tamaño: 0.00 MB
Repositorio: BIOBEE



Nombre: Daño De La Abeja
Tamaño: 0.00 MB
Repositorio: BIOBEE



Nombre: Fondos Del Juego
Tamaño: 0.36 MB
Repositorio: BIOBEE

7. Usuario en la página individual de cada recurso

Detalle del Recurso Dashboard / Detalle del Recurso

[Descargar Recurso](#) [Eliminar Recurso](#) [Editar Recurso](#)

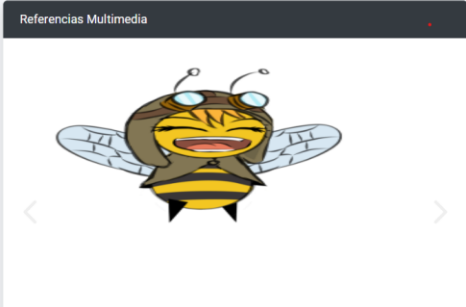
Información General

Nombre: Personajes
Descripción: Personajes
Tamaño: 2.67 MB
Tipo: _2D_ASSET
Creado: 28/01/2024

Información del Repositorio

Nombre: Jardines de Babilonia
Descripción: Los Jardines colgantes de Babilonia fueron una de las siete maravillas del mundo antiguo.

Referencias Multimedia



8. Usuario en la página para editar el Recurso.

Editar Recurso Dashboard / Editar Recurso

Información General del Recurso

Nombre del Recurso

Descripción

Tipo de recurso
2D Asset

Características del Recurso

Selecciona las etiquetas

Búsqueda

- Etiquetas
- Boda
- Botones

[Abeja X](#) [Personaje X](#) [Bosque X](#) [Árbol X](#)

Carga el Recurso

[+ Choose](#) [X Cancel](#)

9. Usuario en el Recurso tras editarlo.

Detalle del Recurso Dashboard / Detalle del Recurso

[Descargar Recurso](#) [Eliminar Recurso](#) [Editar Recurso](#)


Información General

Nombre: Botones Del Juego
Descripción: Los botones del juego
Tamaño: 0.05 MB
Tipo: _2D_ASSET
Creado: 27/01/2024

Información del Repositorio

Nombre: BIOBEE
Descripción: El juego de una abeja gorda y bonita esquivando cucarachas

Referencias Multimedia



10. Página para ver el listado de juegos.

Listado de Juegos Dashboard / Listado de Juegos

Filter [+ Crear](#)

ID	Name	Created At	Acciones
1	Hachi	28/01/2024	Eliminar Editar

Items per page: 1 - 1 of 1 < >

11. Página para editar el juego.

Editar Juego Dashboard / Editar Juego

Información General del Juego

Nombre del Juego

Descripción

Tipo de recurso

Características del Juego

Selecciona las etiquetas

Búsqueda

- Abeja
- Árbol
- Automóvil

Seleccionar una etiqueta como mínimo

Carga el Juego y Manual

12. Página individual de cada juego.

Detalle del Juego Dashboard / Detalle del Juego

Información General

Nombre: Hachi
Descripción: Un dueño descuida a su perro
Tamaño: 600.18 MB
Tipo: GAME
Creado: 28/01/2024

Reproductor de Video

0:00

13. Descargar el Juego

