UNIVERSIDAD POLITÉCNICA SALESIANA

SEDE CUENCA

CARRERA DE COMPUTACIÓN

DIFUSIÓN Y DESPLIEGUE DE VIDEOJUEGOS GAMIFICADOS DESARROLLADOS EN UNITY Y UNREAL ENGINE MEDIANTE MICROSERVICIOS EN LA PLATAFORMA EDUTAINMENT DEL GRUPO GIHP4C

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

AUTOR: CHRISTIAN MAURICIO BUESTAN VILLA

TUTOR: ING. OMAR GUSTAVO BRAVO QUEZADA, PHD.

Cuenca - Ecuador

2024

CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE TITULACIÓN

Yo, Christian Mauricio Buestan Villa con documento de identificación N° 0106467251, manifiesto que:

Soy el autor y responsable del presente trabajo; y, autorizo 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 julio del 2024

Atentamente,

Christian Mauricio Buestan Villa
0106467251

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

Yo, Christian Mauricio Buestan Villa con documento de identificación N° 0106467251, expreso mi voluntad y por medio del presente documento cedo a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que soy autor del Proyecto técnico: "Difusión y despliegue de videojuegos gamificados desarrollados en Unity y Unreal Engine mediante microservicios en la Plataforma Edutainment del grupo GIHP4C", 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, suscribo este documento en el momento que hago la entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.

Cuenca, 29 de julio del 2024

Atentamente,

Christian Mauricio Buestan Villa
0106467251

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: DIFUSIÓN Y DESPLIEGUE DE VIDEOJUEGOS GAMIFICADOS DESARROLLADOS EN UNITY Y UNREAL ENGINE MEDIANTE MICROSERVICIOS EN LA PLATAFORMA EDUTAINMENT DEL GRUPO GIHP4C, realizado por Christian Mauricio Buestán Villa con documento de identificación N° 0106467251, 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 julio del 2024

Atentamente,

Ing. Omar Gustavo Bravo Quezada, PhD.
0102777257

Dedicatoria y agradecimientos

Dedico el presente documento a mi madre, a mi padre y a mis hermanos por haber cuidado de mi por tanto tiempo, permitirme estudiar y crecer como persona en un hogar acogedor y lleno de amor de su parte. Les dedico este documento para que sepan que yo si los amo con toda mi alma.

Agradezco a Dios por darme salud, conocimiento, tolerancia y sabiduría en vida para mantener mi vida estable conforme a estándares de vida y respeto mutuo al prójimo, tener fe y siempre creer en su palabra para la vida y muerte.

Agradezco a todos los tutores y profesores que he tenido a lo largo de mi vida, aguantando mis acciones, resentimientos y ayudándome a ser mejor estudiante con cada año de estudio y de saberes fuera de la carrera.

Agradecimientos mi tutor el PhD. Gustavo Bravo Quezada por los ánimos al momento de realizar el trabajo, apoyarme en los momentos de mayor debilidad frente a la carga de trabajo.

Agradezco también al compañero German Tobar por ser eje central para que el proyecto se terminará, siendo un apoyo en este proceso de aprendizaje y finalización del trabajo.

Resumen

Este proyecto tuvo como eje principal la construcción de aplicaciones web a partir de proyectos de Unity de los estudiantes de desarrollo multimedia, las cuales serán desplegadas dentro de la plataforma de edutainment por medio de un repositorio de GitHub asignado exclusivamente a los juegos gamificados, permitiendo a los desarrolladores resguardar sus aplicaciones y proyectos sin la necesidad de los usuarios de directamente descargarlos para probar dichos juegos, también se fomenta participación de los usuarios para conocer y aprender con estos juegos gamificados dentro de la plataforma de Edutainment. El presente desarrollo del trabajo hizo uso de la metodología de prototipado para el despliegue correcto de las aplicaciones web generadas, teniendo varias versiones del despliegue local de los juegos para probar funcionalidades de Unity Web Player(WEBGL) antes ser cargados en la plataforma, así también se tomaron decisiones de diseño del despliegue para mejorar el desempeño y ser más atractivo para los jugadores; siendo el resultado final aplicaciones web desplegadas dentro de la plataforma de edutainment para el disfrute de usuarios y respaldo de proyectos de los estudiantes de la carrera de Diseño multimedia.

Palabras clave

Unity, WEBGL, Prototipado, Edutainment, Gamificación, GitHub.

Abstract.

This project had as its main axis the construction of web applications from Unity projects by multimedia development students, which will be deployed within the edutainment platform through a GitHub repository assigned exclusively to gamified games, allowing the developers protect their applications and projects without the need for users to directly download them to try these games. User participation is also encouraged to know and learn with these gamified games within the Edutainment platform. The present development of the work made use of the prototyping methodology for the correct deployment of the generated web applications, having several versions of the local deployment of the games to test Unity Web Player (WEBGL) functionalities before being loaded on the platform, as well as display design decisions were made to improve performance and be more attractive to players; The final result being web applications deployed within the edutainment platform for the enjoyment of users and support of projects by students of the Multimedia Design career.

KeyWords

Unity, WEBGL, Prototyping, Edutainment, Gamification, GitHub.

Índice

Dedicatoria y agradecimientos 5

Resumen 6

Palabras clave 6

Abstract. 6

KeyWords 6

Índice 7

Índice de ilustraciones 8

Introducción 10

Problema 10

Justificación 11

Grupo Objetivo 13

Objetivos 14

Revisión de la literatura o fundamentos teóricos. 14

Marco metodológico 20

Etapas de la Metodología de Prototipado 20

Herramientas y Tecnologías Utilizadas 22

Resultados 22

Cronograma 38

Presupuesto 39

Conclusiones 39

Recomendaciones 41

Referencias bibliográficas 43

Anexos 45

Índice de ilustraciones

Imagen 1. Frameworks Surática.(2014).. 15
Imagen 2.Angular Logo. 15
Imagen 3.Springboot Logo. 16
Imagen 4.Motores gráficos de desarrollo de videojuegos. Delgado M. (12 de febrero de 2021). 16
Imagen 5.Unreal Engine Logo. 16
Imagen 6.Unity Logo. 17
Imagen 7.Microservicios. Luis J. (8 de julio de 2022). 17
Imagen 8. Gamificación. 17
Imagen 9. Web. 18
Imagen 10. Servicios Web. 18
Imagen 11. EduEntretenimeinto. 19
Imagen 12. Icono del lenguaje de C++. 19
Imagen 13. Aplicaciones web en un navegador. 19
Imagen 14. Icono de Git. 20
Imagen 15. Icono de GitHub. 20
Imagen 16. Estructura del Prototyping. Maida, EG, Pacienzia, J. (diciembre de 2015). 22
Imagen 17. Carga correcta de un proyecto de Unity. 23
Imagen 18. Configuración de compresión del proyecto. 24
Imagen 19. Resolución del proyecto. 24
Imagen 20. Ventana de configuración de Build. 25
Imagen 21. Cambio de plataforma. 26
Imagen 22. Selector de escenas de la construcción de aplicación. 26
Imagen 23. Botón Build para la construcción de la aplicación web. 27
Imagen 24. Resultados en consola de la construcción de la aplicación web. 27
Imagen 25. Archivos resultantes de la exportación. 28
Imagen 26. Archivo server.js. 29
Imagen 27. Archivos de dependencias generados. 29
Imagen 28. Despliegue de la aplicación web en el navegador. 30
Imagen 29. Agregar el componente a las rutas de la plataforma. 31
Imagen 30. Instanciar el componente creado.. 31
Imagen 31. Creación y configuración de los subcomponentes. 31
Imagen 32. Índice de la página de juegos usando sus subcomponentes. 32
Imagen 33. Página de los juegos localmente. 32
Imagen 34. Repositorio de los juegos. 33
Imagen 35. Carta de juego. 33
Imagen 36. Configuración para redimensionar la ventana de juego. 34
Imagen 37. Texto y botón de regreso en la ventana de juego. 34
Imagen 38. Estilo de los elementos en el HTML. 35
Imagen 39. Componente Juegos de la plataforma localmente. 35
Imagen 40. Juego BioBee. 36
Imagen 41. Juego Batalla Bucal. 36

Imagen 42. Plataforma edutainment modificada. 37

Introducción

En la actualidad, la gamificación ha emergido como una herramienta poderosa en el ámbito educativo, transformando la forma en la que los estudiantes interactúan con el contenido y mejorando su compromiso y motivación en el proceso de aprendizaje. La gamificación se basa en la utilización de elementos y técnicas propias de los juegos para fomentar el aprendizaje y la resolución de problemas en contextos no lúdicos. Esta estrategia educativa ha demostrado ser efectiva en diversos estudios, ya que integra el aprendizaje con el entretenimiento, creando experiencias educativas más atractivas y efectivas.

En particular, los videojuegos gamificados combinan el entretenimiento con objetivos educativos, permitiendo a los usuarios aprender mientras se divierten. Estos juegos pueden influir positivamente en el desarrollo personal y educativo de los usuarios, abriendo puertas para su uso en la educación formal e informal. Sin embargo, en muchas instituciones educativas, existe una carencia significativa de plataformas centralizadas para el almacenamiento y difusión de estos videojuegos gamificados, lo que limita su accesibilidad y aprovechamiento.

En particular, los videojuegos gamificados combinan el entretenimiento con objetivos educativos, permitiendo a los usuarios aprender mientras se divierten. Estos juegos pueden influir positivamente en el desarrollo personal y educativo de los usuarios, abriendo puertas para su uso en la educación formal e informal. Sin embargo, en muchas instituciones educativas, existe una carencia significativa de plataformas centralizadas para el almacenamiento y difusión de estos videojuegos gamificados, lo que limita su accesibilidad y aprovechamiento (Intel, 2021).

La importancia de este proyecto radica en la necesidad de preservar y promover los videojuegos gamificados desarrollados por los estudiantes de la carrera de Diseño Multimedia. Estos juegos, al estar alojados en una plataforma accesible y funcional, pueden alcanzar un mayor impacto educativo y social. Además, al integrar elementos culturales y educativos, estos contribuyen al aprendizaje significativo y al enriquecimiento cultural de los usuarios. El desarrollo de este proyecto beneficia a los estudiantes que desarrollan los videojuegos, a la comunidad educativa y al público en general, proporcionando contenido educativo de alta calidad de manera sencilla y atractiva (Núñez Estela et al., 2020).

El desarrollo de este proyecto beneficia a los estudiantes que desarrollan los videojuegos, a la comunidad educativa y al público, que pueden acceder a contenido educativo de alta calidad de manera sencilla y atractiva. La plataforma Edutainment se convierte así en un repositorio dinámico de conocimiento y entretenimiento, incentivando el aprendizaje y la competencia sana entre los usuarios a través de tablas de puntuación y otros mecanismos de gamificación.

Problema

En el ámbito educativo y de desarrollo de software, los videojuegos gamificados han surgido como herramientas valiosas para la enseñanza y el aprendizaje. Estos juegos combinan elementos lúdicos con objetivos educativos, ofreciendo experiencias interactivas y atractivas que facilitan la adquisición de conocimientos. Sin embargo, muchas instituciones educativas, especialmente aquellas que imparten la carrera de Diseño Multimedia, enfrentan un problema

significativo: la ausencia de una plataforma centralizada para el almacenamiento y difusión de los videojuegos gamificados desarrollados en Unity y Unreal Engine.

Una vez completados, muchos de estos videojuegos quedan relegados al olvido, almacenados en dispositivos locales o, peor aún, se pierden con el tiempo. Esta falta de accesibilidad no solo desperdicia los valiosos recursos invertidos en su desarrollo, sino que también desaprovecha oportunidades de aprendizaje tanto para los desarrolladores como para los usuarios finales. Según Núñez Estela et al. (2020), "los videojuegos pueden influir positivamente en el desarrollo personal y educativo de los usuarios, abriendo puertas para su uso en la educación"; sin una plataforma adecuada, este potencial queda sin explotar.

La situación actual presenta varios desafíos críticos:

1. **Accesibilidad y Preservación**:

   La falta de una plataforma centralizada dificulta el acceso continuo a los videojuegos gamificados, limitando su uso y preservación a largo plazo. Esto impide que los estudiantes y desarrolladores compartan y difundan sus proyectos, reduciendo el impacto potencial de sus trabajos.

2. **Aprovechamiento de Recursos:**

   Los recursos invertidos en el desarrollo de videojuegos gamificados, incluyendo tiempo, esfuerzo y conocimientos técnicos, se desperdician cuando estos juegos no se almacenan ni difunden adecuadamente. Esto también significa que otros estudiantes y educadores no pueden beneficiarse de las experiencias y aprendizajes incorporados en estos juegos.

3. **Falta de Interacción y Feedback:**

   Sin una plataforma que facilite la interacción entre los desarrolladores y los usuarios finales, los videojuegos gamificados carecen de un mecanismo efectivo para recibir retroalimentación. Esto es crucial para mejorar la calidad de los juegos y adaptar los contenidos a las necesidades educativas de los usuarios.

Para abordar este problema, se propone la creación de una plataforma web que utilice microservicios para el despliegue y gestión de los videojuegos gamificados. Los microservicios, según Intel (2021), “son componentes ágiles, escalables y fáciles de implementar en sistemas distribuidos lo que los convierte en una solución ideal para manejar aplicaciones basadas en la nube”. Esta plataforma no solo proporcionará un espacio seguro y accesible para almacenar los videojuegos, sino que también permitirá a los usuarios descubrir, jugar y aprender de una variedad de videojuegos educativos.

Justificación

La creación de una plataforma web centrada en la difusión y acceso de videojuegos gamificados desarrollados en Unity y Unreal Engine mediante microservicios se presenta como

una solución efectiva para abordar la problemática actual de accesibilidad y preservación de estos recursos educativos. Esta plataforma proporcionará un espacio centralizado donde los desarrolladores de videojuegos puedan compartir sus creaciones y donde los usuarios puedan descubrir, jugar y aprender de una variedad de videojuegos educativos.

La justificación de este proyecto se sustenta en varios aspectos clave:

1. **Mejora de la Accesibilidad y Preservación:**
   Al centralizar el almacenamiento de los videojuegos gamificados, se garantiza su accesibilidad a largo plazo y se evita la pérdida de valiosos recursos educativos. Esto permite a los estudiantes y desarrolladores preservar su trabajo y a los usuarios finales acceder fácilmente a contenidos educativos de calidad.
2. **Aprovechamiento Óptimo de Recursos:**
   La plataforma permitirá un mejor aprovechamiento de los recursos invertidos en el desarrollo de los videojuegos gamificados. Los esfuerzos y conocimientos técnicos dedicados a estos proyectos no se perderán, sino que estarán disponibles para futuras generaciones de estudiantes y educadores, fomentando un ciclo continuo de aprendizaje y desarrollo.
3. **Facilitación de la Interacción y Feedback:**
   Al proporcionar un espacio donde los desarrolladores pueden compartir sus juegos, la plataforma facilita la interacción con los usuarios finales y la obtención de retroalimentación valiosa. Esto es crucial para mejorar continuamente la calidad de los juegos y adaptar los contenidos a las necesidades y preferencias de los usuarios.
4. **Integración de Elementos Culturales y Educativos:**
   Los videojuegos desarrollados integran elementos culturales y educativos que enriquecen el aprendizaje de los usuarios. La plataforma Edutainment no solo difunde estos contenidos, sino que también contribuye al enriquecimiento cultural de la comunidad, promoviendo el conocimiento de la historia, mitología y leyendas de la región.
5. **Escalabilidad y Flexibilidad:**
   La utilización de microservicios permite una mayor escalabilidad y flexibilidad en la gestión y ejecución de aplicaciones web. Los microservicios son componentes independientes que ejecutan procesos específicos y se comunican entre sí a través de interfaces bien definidas, lo que reduce la complejidad y mejora la capacidad de respuesta del sistema.
6. **Beneficios Educativos y Sociales:**
   La plataforma beneficia a los estudiantes de la carrera de Diseño Multimedia, a la comunidad educativa y al público en general, proporcionando contenido educativo de alta calidad de manera sencilla y atractiva. Además, fomenta la competencia sana y el aprendizaje significativo a través de mecanismos de gamificación como tablas de puntuación y desafíos semanales.
7. **Innovación Tecnológica en la Educación:**
   Este proyecto promueve el uso de tecnologías innovadoras en la educación, apoyando el aprendizaje y el desarrollo de habilidades digitales en un entorno lúdico y estimulante. La integración de herramientas modernas como Unity y Unreal Engine,

junto con la arquitectura de microservicios, posiciona a la Universidad Politécnica Salesiana como un referente en la adopción de tecnologías emergentes para la mejora educativa

Grupo Objetivo

El grupo objetivo principal de este proyecto está compuesto por los estudiantes de la carrera de Diseño Multimedia de la Universidad Politécnica Salesiana. Estos estudiantes son los principales beneficiarios, ya que la plataforma les permitirá respaldar y difundir sus aplicaciones y proyectos de videojuegos gamificados. Sin embargo, el impacto de la plataforma se extiende a otros grupos clave:

1. **Estudiantes de Diseño Multimedia:**
   La plataforma proporciona un espacio seguro y accesible para almacenar y compartir sus proyectos, lo que les permite obtener reconocimiento por su esfuerzo y creatividad. Además, facilita la preservación de sus trabajos, garantizando que no se pierdan con el tiempo y estén disponibles para futuras referencias y mejoras.
   Al tener la posibilidad de recibir retroalimentación de otros usuarios y desarrolladores, los estudiantes pueden mejorar sus habilidades y conocimientos, lo que contribuye a su desarrollo profesional y académico.
2. **Comunidad Educativa:**
   Profesores y educadores pueden utilizar la plataforma como una herramienta educativa, integrando los videojuegos gamificados en sus planes de estudio. Esto les permite ofrecer una experiencia de aprendizaje más interactiva y atractiva, lo que puede aumentar el compromiso y la motivación de los estudiantes.
   La plataforma también sirve como un repositorio de recursos educativos innovadores, facilitando el acceso a materiales didácticos de alta calidad y fomentando el intercambio de conocimientos y buenas prácticas entre educadores.
3. **Desarrolladores de Videojuegos:**
   Además de los estudiantes, otros desarrolladores de videojuegos pueden beneficiarse de la plataforma al tener un espacio para mostrar y compartir sus proyectos. Esto puede incluir a exalumnos y profesionales del sector que deseen contribuir con sus conocimientos y experiencias.
   La plataforma ofrece un entorno colaborativo donde los desarrolladores pueden interactuar, intercambiar ideas y colaborar en proyectos, lo que puede llevar a la creación de videojuegos más innovadores y efectivos.
4. **Usuarios Finales (Estudiantes y Público en General):**
   Los usuarios finales de la plataforma, incluidos estudiantes de diferentes niveles educativos y el público en general, pueden acceder a una variedad de videojuegos gamificados que combinan entretenimiento y educación. Esto les permite aprender de manera lúdica sobre diversos temas, desde la historia y la cultura hasta conceptos científicos y técnicos.
   Al participar en los juegos y desafíos propuestos en la plataforma, los usuarios pueden mejorar sus habilidades cognitivas, su conocimiento cultural y su motivación por el aprendizaje continuo.

5. **Institución Educativa:**
   La Universidad Politécnica Salesiana se beneficia al posicionarse como una institución innovadora que utiliza tecnologías avanzadas para mejorar la educación. La plataforma Edutainment puede atraer a nuevos estudiantes interesados en carreras tecnológicas y fortalecer la reputación de la universidad como un líder en la integración de la gamificación en el proceso educativo.

Objetivos

Objetivo general

Difundir y desplegar los videojuegos gamificados desarrollados en Unity y Unreal Engine mediante microservicios en la Plataforma Edutainment del grupo GIHP4C.

Objetivos específicos

OE1. Investigar sobre las herramientas de desarrollo de videojuegos Unreal y Unity; generación de aplicaciones web en los diferentes ámbitos.

OE2. Modificar las aplicaciones que no cuenten con microservicios para el registro de información para posteriormente crearlas.

OE3. Implementar los microservicios a la aplicación y a la plataforma de Edutainment; obteniendo los registros de usuario respectivos.

OE4. Testear y evaluar el Frontend de la plataforma de Edutainment con los microservicios, aplicar cambios de ser necesarios para su posterior despliegue.

Revisión de la literatura o fundamentos teóricos.

La revisión de la literatura es una parte fundamental de cualquier proyecto de investigación, ya que proporciona el contexto y el marco teórico necesario para entender el problema de estudio y las soluciones propuestas. En el caso del presente proyecto, que se centra en la difusión y despliegue de videojuegos gamificados en una plataforma educativa, es esencial explorar y analizar la literatura existente sobre varios temas clave. Estos temas incluyen la gamificación en la educación, las tecnologías y metodologías de desarrollo de videojuegos, la arquitectura de microservicios y su aplicación en plataformas web, y el impacto cultural y educativo de los videojuegos.

Esta sección ofrece una revisión exhaustiva de los conceptos teóricos y estudios previos relacionados con estos temas. Se examinarán las bases teóricas y prácticas de la gamificación, las características y funcionalidades de los motores de videojuegos como Unity y Unreal Engine, y las ventajas y desafíos de utilizar una arquitectura de microservicios para el despliegue de aplicaciones web. Además, se analizará el papel de los videojuegos en la educación y su potencial para enriquecer el aprendizaje y la experiencia cultural de los usuarios.

La comprensión de estos fundamentos teóricos no solo respaldará el desarrollo y la implementación del proyecto, sino que también ayudará a justificar su relevancia e impacto en el ámbito educativo y tecnológico. Al revisar la literatura relevante, se identificarán las mejores

prácticas, se reconocerán las brechas existentes en el conocimiento y se establecerá una base sólida para las contribuciones del presente trabajo.

- Framework: Conocido también como marco de trabajo es una estructura básica donde se emplean un conjunto de reglas y estándares para el óptimo desarrollo de software. Su funcionalidad ahorra tiempo y gastos durante el desarrollo, además de proporcionar un referente para que cualquier programador pueda comenzar con sus aplicaciones. Dentro de este se pueden encontrar varios ejemplos, como Angular, Springboot, React, Vue.js entre otros.

Imagen 1. Frameworks Surática.(2014).

- Angular: Este framework ayudará a desplegar la aplicación web de forma dinámica y sin ningún tipo de enlace externo, más que nada presentará una ventana de juego dentro de la misma plataforma de Edutainment. Esto es posible gracias a sus módulos que permiten mantener una aplicación web en una sola página, así se puede tener y ejecutar más de un solo videojuego gamificado.

Imagen 2.Angular Logo.

- SpringBoot: Es un framework principalmente orientado a facilitar la creación de microservicios y despliegue plataformas web, esto gracias a que las aplicaciones son autocontenidas para su posterior implementación en la plataforma.

Imagen 3.Springboot Logo.

- Motores de creación de videojuegos: También conocidos como Engines, son software adecuados para la creación de videojuegos, contando con funcionalidades para que los desarrolladores se enfoquen más en el buen diseño de un videojuego mediante la creatividad y no tanto en el aspecto técnico que estos puedan presentar eventualmente, los principales motores son: Unity, Unreal Engine y Game Maker Studio.

Imagen 4.Motores gráficos de desarrollo de videojuegos. Delgado M. (12 de febrero de 2021).

- Unreal Engine: Es un motor de creación de videojuegos proporcionado de forma gratuita por Epic Games, además es una forma en la que los videojuegos gamificados serán proporcionados, por lo que el conocimiento de esta herramienta es importante para el desarrollo de una aplicación web desarrollada con este motor de alta gama. Cabe destacar que el motor cuenta con versatilidad para creación de interfaces dedicadas al usuario.

Imagen 5.Unreal Engine Logo.

- Unity: Es otro motor multiplataforma de videojuego proporcionado por Unity Technologies, el juego a implementar puede desarrollarse en base a este motor, siendo una buena alternativa un proceso de creación de aplicaciones web nativamente por su software debido a la versatilidad que puede brindar al desarrollar los videojuegos;

contando con un gran número de tutoriales que explican cómodamente su uso para el desarrollo de la aplicación, exportando a una aplicación web para posteriormente importarla dentro de la plataforma.

Imagen 6.Unity Logo.

- Microservicios: Los microservicios se usarán para administrar sesión y posterior registro de datos de usuario, se basarán en puntaje y fecha de juego para la base de datos, con esta data se mostrará en la misma plataforma en una sección de estadística del usuario o jugador que haya probado el videojuego. La razón de usar microservicios es para tener un sistema libre de caídas por errores aislados y un mejor mantenimiento de las tareas que estos van a administrar, caso que sería imposible de implementar con una estructura de tipo monolito.

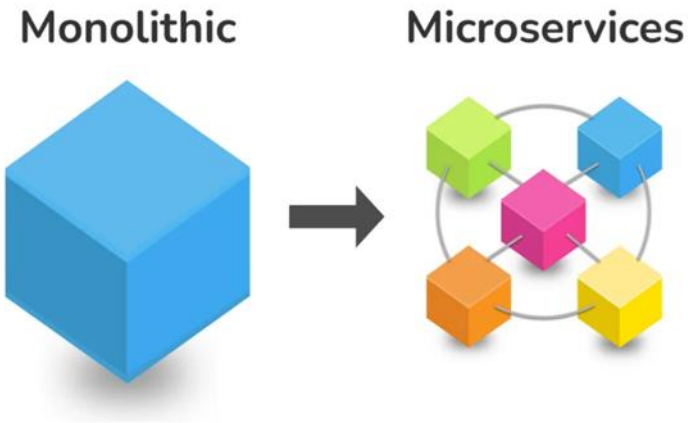

Imagen 7.Microservicios. Luis J. (8 de julio de 2022).

- Gamificación: Es la base de este proyecto, el implementar mecanismos de juegos al mundo educativo para que los jóvenes puedan divertirse y aprender por igual; teniendo en cuenta lo difícil que es mantener el interés actual por la tecnología, es un gran incentivo para la formación cultural y educativo de las personas.

Imagen 8. Gamificación.

- Desarrollo web: Es la forma en que se crean productos digitales o sitios web en base a una correlación entre el usuario y una base de datos por medio de un navegador que

permita realizar tareas (Formularios, consultas, juegos, etc.) en base al sitio desarrollado o simplemente mostrar información (Blogs, Videos, Redes sociales, Noticieros, etc.) siendo una base para desplegar los videojuegos gamificados dentro de una plataforma web.

Imagen 9. Web.

- Web Services: Es la forma que una API va a comunicarse entre sí de la plataforma, teniendo en cuenta modelos de la plataforma y su estructura, esto es más importante para el montado de las aplicaciones web de los juegos gamificados, siendo un conocimiento ideal para el correcto despliegue de estos sin tener la necesidad de alterar el código o estructura.

Imagen 10. Servicios Web.

- Edutainment: Es una combinación de lo educativo y lo divertido de realizar una actividad, en este caso que tan educativo sería el videojuego montado a la plataforma; también estará relacionado el nivel de diversión que se alcanzará para que los usuarios jueguen y aprendan más en base a un apego por estos videojuegos, siendo un enforque pedagógico que se aplica a la vida real.

Imagen 11. EduEntretenimeinto.

- C++: Es el código por el que está desarrollado principalmente los juegos hechos con Unity o Unreal Engine, el conocimiento de este lenguaje está relacionado con las modificaciones que un juego exportado como aplicación web pueda tener, como la recolección de datos de usuario o alteraciones para que sea más sencillos de ejecutar desde la web.

Imagen 12. Icono del lenguaje de C++.

- Aplicaciones web: Es la base del proyecto, una forma de que videojuegos gamificados y hechos con el motor de Unity y/o Unreal Engine puedan ser ejecutados desde un navegador web o en este caso desde una plataforma educativa orientada a preservar los proyectos de la carrera de multimedia, es la forma más sencilla y optima de implementar juegos en la nube para que los usuarios puedan disfrutar de una sesión educativa y entretenida.

Imagen 13. Aplicaciones web en un navegador.

- Git: Es el sistema de administración de versiones de proyectos principalmente usado para un trabajo en paralelo de proyectos, facilitando el trabajo en equipo para el desarrollo de la plataforma de Edutainment de forma remota.

Imagen 14. Icono de Git.

- GitHub: Es el servidor donde se almacenan las versiones de la plataforma de Edutainment, permitiendo el trabajo de varios usuarios sobre el mismo proyecto de forma remota y con la seguridad proporcionada por los servicios de la nube.

Imagen 15. Icono de GitHub.

Marco metodológico

El presente proyecto se llevó a cabo utilizando la metodología de prototipado, que es especialmente adecuada para el desarrollo iterativo y la validación continua de aplicaciones web y videojuegos gamificados. La metodología de prototipado permite crear versiones preliminares del producto final, facilitando la identificación y corrección de errores en etapas tempranas del desarrollo. Este enfoque asegura que el producto final cumpla con los requisitos y expectativas de los usuarios y stakeholders.

Etapas de la Metodología de Prototipado

1. **Recolección y Refinamiento de Requisitos:**
   Acceso a Repositorios: Se solicitó acceso a los repositorios de los proyectos desarrollados en Unity y Unreal Engine. Esta etapa involucró la recopilación de información sobre las funcionalidades y características de los videojuegos a ser desplegados.
   Investigación del Funcionamiento: Se investigó el funcionamiento de los sistemas de Unity y Unreal Engine para la creación de aplicaciones web, asegurando que los proyectos se pudieran exportar y adaptar adecuadamente para la plataforma Edutainment.
2. **Diseño Rápido:**
   Definición del Prototipo: Se definió la forma en la que sería más conveniente implementar los videojuegos dentro de la plataforma web. Esta etapa incluyó la selección de tecnologías y herramientas adecuadas para el despliegue de aplicaciones web, como el uso de Angular para la interfaz de usuario.

Planificación de Interfaces: Se diseñaron interfaces intuitivas y amigables para el usuario, que facilitaran la interacción con los videojuegos gamificados y mejoraran la experiencia del usuario.

3. **Construcción del Prototipo:**
   Exportación de Aplicaciones Web: Se utilizó Unity y Unreal Engine para exportar los proyectos como aplicaciones web. Este proceso involucró varias configuraciones específicas para garantizar la compatibilidad con navegadores web y la plataforma Edutainment.
   Configuraciones Técnicas: Se realizaron configuraciones en los proyectos, como la compresión de archivos, la selección de resoluciones adecuadas y la eliminación de escenas no compatibles con WEBGL, para optimizar el rendimiento y la experiencia del usuario.
4. **Evaluación del Prototipo:**
   Pruebas de Funcionamiento: Se probaron las aplicaciones web exportadas en navegadores web para verificar su correcto funcionamiento. Esta etapa incluyó pruebas de funcionalidad, rendimiento y compatibilidad.
   Feedback y Correcciones: Se recopilaron comentarios de los usuarios y se realizaron las correcciones necesarias para mejorar el prototipo, asegurando que cumpla con los requisitos y expectativas.
5. **Refinamiento del Prototipo:**
   Implementación de Microservicios: Se integraron microservicios para la administración de sesiones y el registro de datos de usuarios. Esto permitió una gestión más eficiente y escalable de las aplicaciones web dentro de la plataforma Edutainment.
   Optimización de Desempeño: Se realizaron optimizaciones adicionales para mejorar el rendimiento de las aplicaciones web, asegurando una experiencia de usuario fluida y atractiva.
6. **Producto Final:**
   Despliegue en la Plataforma Edutainment: Se montaron las aplicaciones web dentro de la plataforma Edutainment, donde los usuarios pueden acceder y disfrutar de los videojuegos gamificados. Este despliegue incluyó la creación de componentes y rutas específicas dentro de la plataforma para facilitar el acceso a los juegos.
   Documentación y Guía para Futuros Desarrolladores: Se elaboró una guía detallada para futuros desarrolladores, describiendo el proceso de exportación y despliegue de aplicaciones web en la plataforma. Esta guía incluye configuraciones técnicas y mejores prácticas para asegurar la calidad y eficiencia de los proyectos (ver Anexo 1).

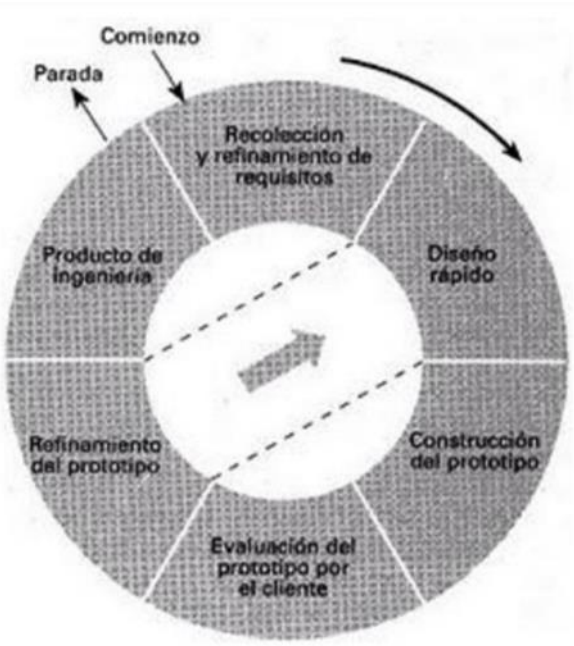

Imagen 16. Estructura del Prototyping. Maida, EG, Pacienzia, J. (diciembre de 2015).

Herramientas y Tecnologías Utilizadas

Motores de Creación de Videojuegos: Unity y Unreal Engine fueron las principales herramientas utilizadas para el desarrollo y exportación de los videojuegos gamificados. Estos motores proporcionan funcionalidades avanzadas para la creación de aplicaciones interactivas y educativas.

Frameworks para Desarrollo Web: Angular fue utilizado para el desarrollo de la interfaz de usuario de la plataforma Edutainment, proporcionando una estructura robusta y escalable para la gestión de aplicaciones web.

Microservicios: La arquitectura de microservicios permitió una gestión eficiente y escalable de las aplicaciones web, mejorando la flexibilidad y capacidad de respuesta del sistema.

Plataformas de Pruebas y Despliegue: Se utilizaron servidores locales y NodeJS para las pruebas iniciales de las aplicaciones web, asegurando que funcionaran correctamente antes de su despliegue en la plataforma Edutainment.

Unreal Engine 5 fue el motor usado para uno de los juegos gamificados, lamentablemente este motor ya no cuenta con soporte de aplicaciones web como la versión 4 del mismo motor, se decidió usar 2 proyectos de Unity, los desarrolladores confirmaron una modificación de su proyecto para que sea posible subirlo como aplicación web posteriormente.

Para los proyectos desarrollados Unity, su despliegue y construcción de las aplicaciones web siguió el siguiente proceso.

Resultados

El proceso de desarrollo y despliegue de los videojuegos gamificados en la plataforma Edutainment se llevó a cabo mediante varias etapas específicas y técnicas. A continuación, se detallan los resultados obtenidos en cada fase del proyecto:

1. Carga del proyecto de Unity.

Se requiere que el juego sea abierto desde una unidad C:/ accesible por el programa de Unity Editor, esto para que la exportación sea libre de inconvenientes. En este documento se usa el proyecto Bio Bee como ejemplo a compilar la aplicación web. Se cargará dentro de Unity Hub como proyecto para exportarse como aplicación web, con cuidado con las versiones requeridas, si hay que instalar las especificadas por el mismo editor. No se recomienda usar archivos corruptos o no funcionales para el desarrollo de las aplicaciones web.

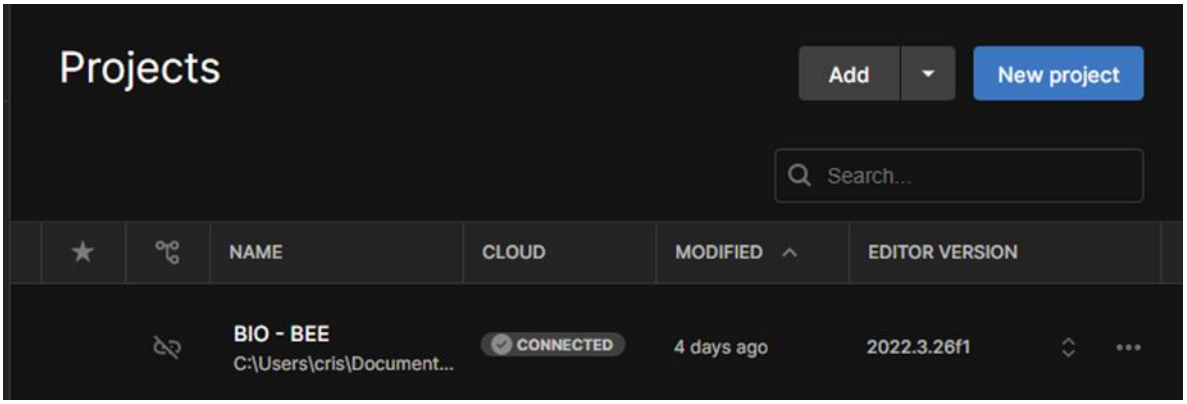

Imagen 17. Carga correcta de un proyecto de Unity.

2. Configuraciones generales del proyecto

a. Modificar configuraciones de la compresión.

En Edit>Project Settings>Player buscar el campo Publishing Settings donde estará el selector Compression Format que permite la compresión del juego al momento de construirlo, esto debe estar deshabilitado para que la aplicación web sea compatible con servidores web.

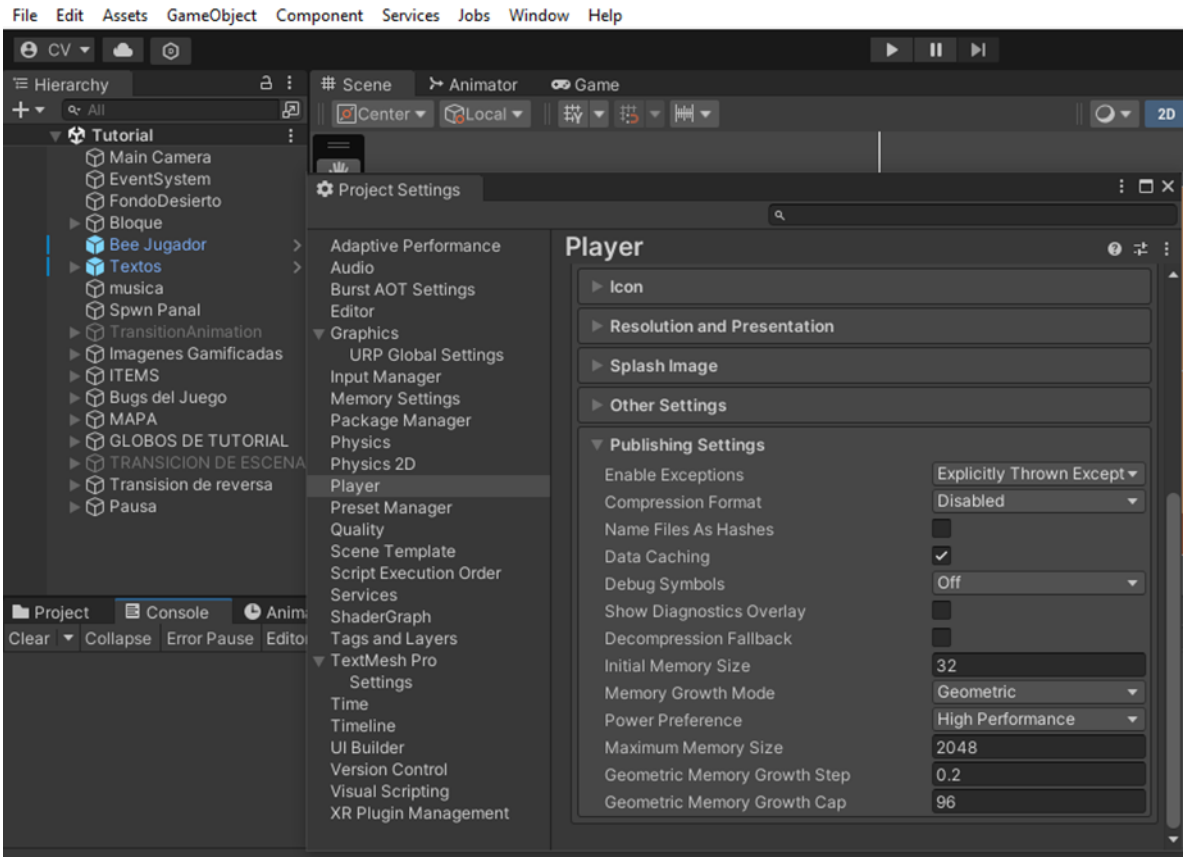

Imagen 18. Configuración de compresión del proyecto.

b. Selección de resolución

La resolución es algo importante, pues permitirá al juego redimensionarse sin necesidad de cambiar el código directamente.

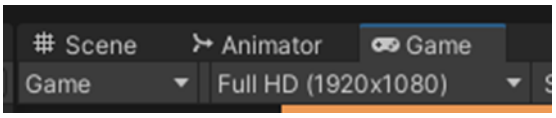

Imagen 19. Resolución del proyecto.

c. Cambiar la plataforma de exportación

En File>Build Settings se encontrará un selector de plataformas que permite el proyecto. Así como el selector de escenas para la exportación de este, pudiendo quitar o seleccionar a voluntad las escenas a compilar en el proyecto.

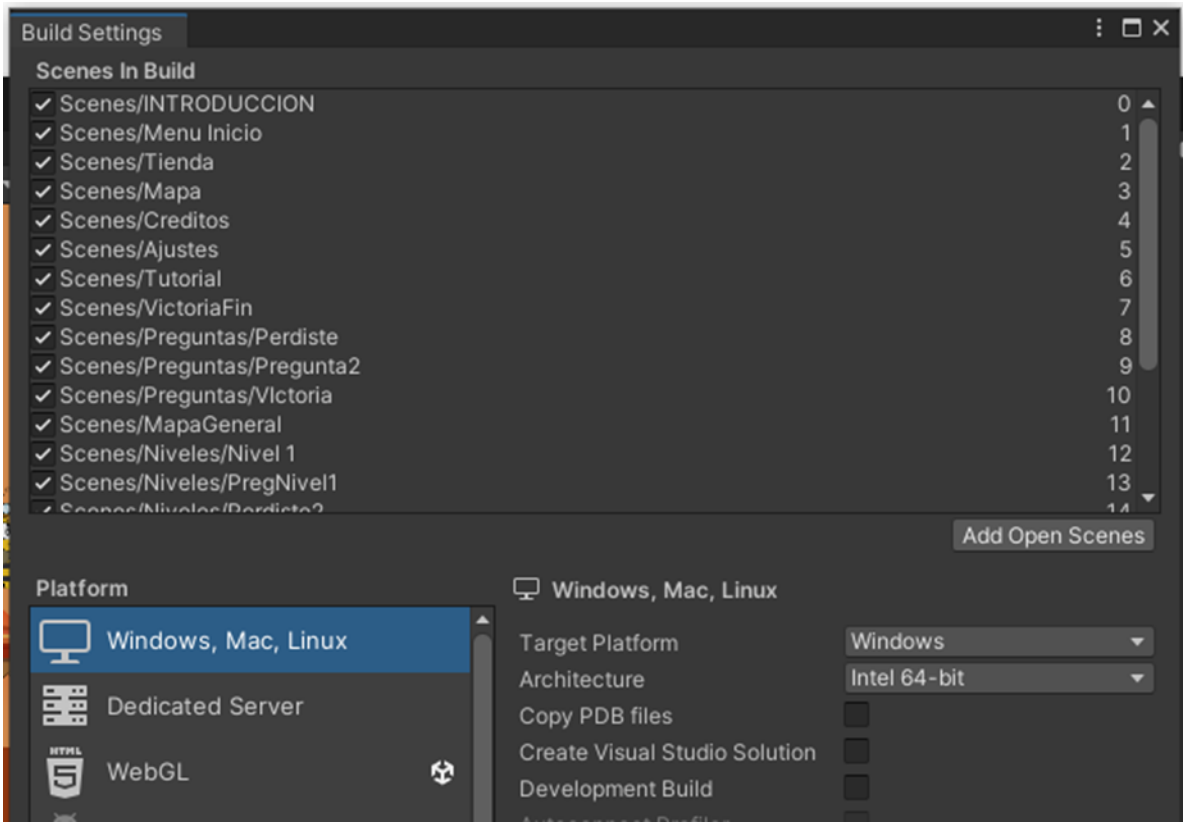

Imagen 20. Ventana de configuración de Build.

Este paso es importante, ya que Unity permite la exportación del juego a diversas plataformas, para dispositivos móviles, sobremesa y el que nos interesa, WEBGL para aplicaciones web. Este será el que se seleccionará para cambiar la plataforma del juego con el botón Switch Platform, notando como el icono de Unity se posiciona sobre la plataforma seleccionada para la exportación.

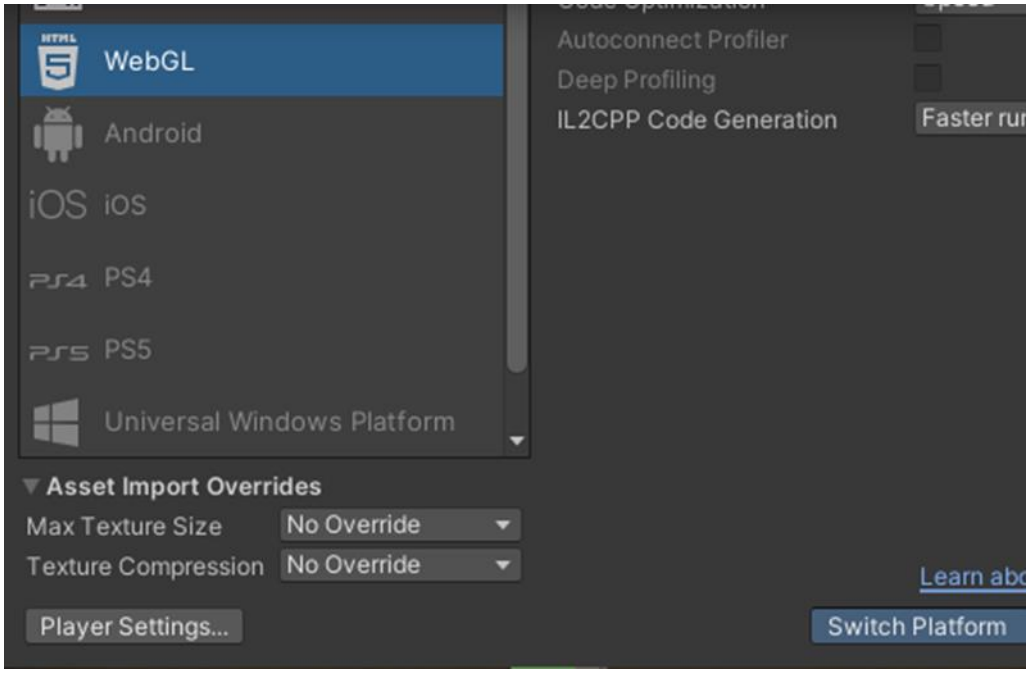

Imagen 21. Cambio de plataforma.

d. Selección de escenas.

Si su proyecto tiene escenas de introducción o animaciones, los cuales WEBGL no puede desplegarlas, por esto es recomendable no seleccionar estas para no afectar el funcionamiento o rendimiento de la aplicación web. Igualmente puede obviar algunas escenas no indispensables para disminuir el peso de la construcción de la aplicación web, al finalizar la construcción de la app web se debe revisar que no exista archivos de peso mayor a 100 Mb, porque es el máximo de peso que GitHub soporta para cargar al repositorio de Edutainment.

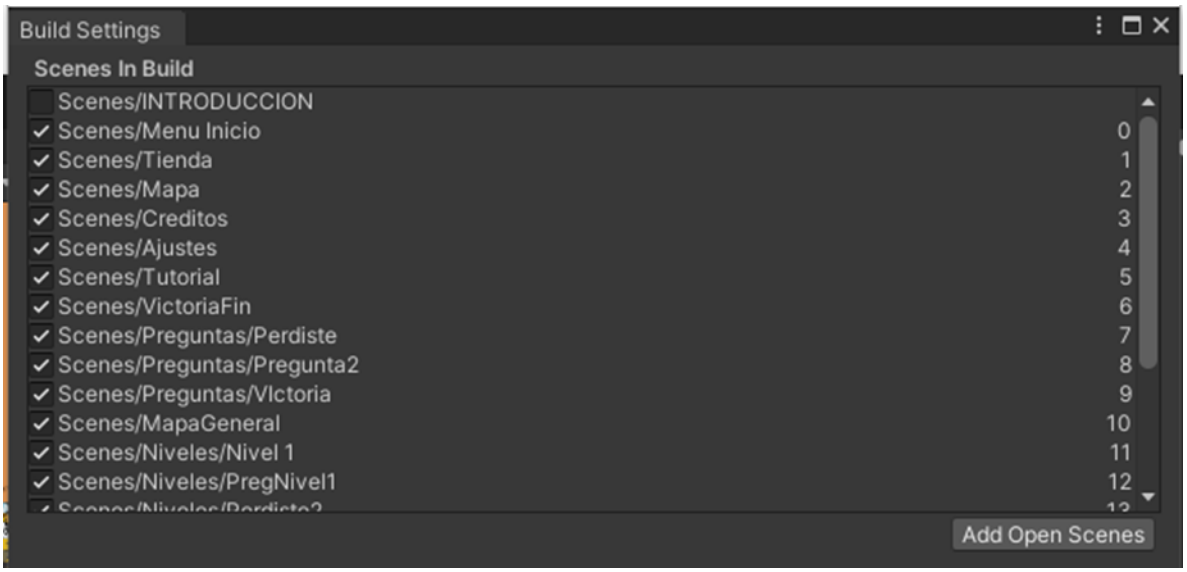

Imagen 22. Selector de escenas de la construcción de aplicación.

3. Construcción de la aplicación web

Una vez configurado el proyecto previamente, se deberá ir a la ventana de configuración con la plataforma WEB GL seleccionada, donde se encontrará 2 opciones.

- Build: Creación de la aplicación web en una carpeta especifica.
- Build and run: Creación y despliegue de la aplicación web con Unity.

Se seleccionará la opción Build, la cual pedirá una carpeta donde serán exportado el proyecto, esto es recomendable ponerlo en una carpeta C: del equipo; conociendo esto se dará clic al botón Build especificando la carpeta destino de exportación, esto iniciará la construcción de la aplicación web.

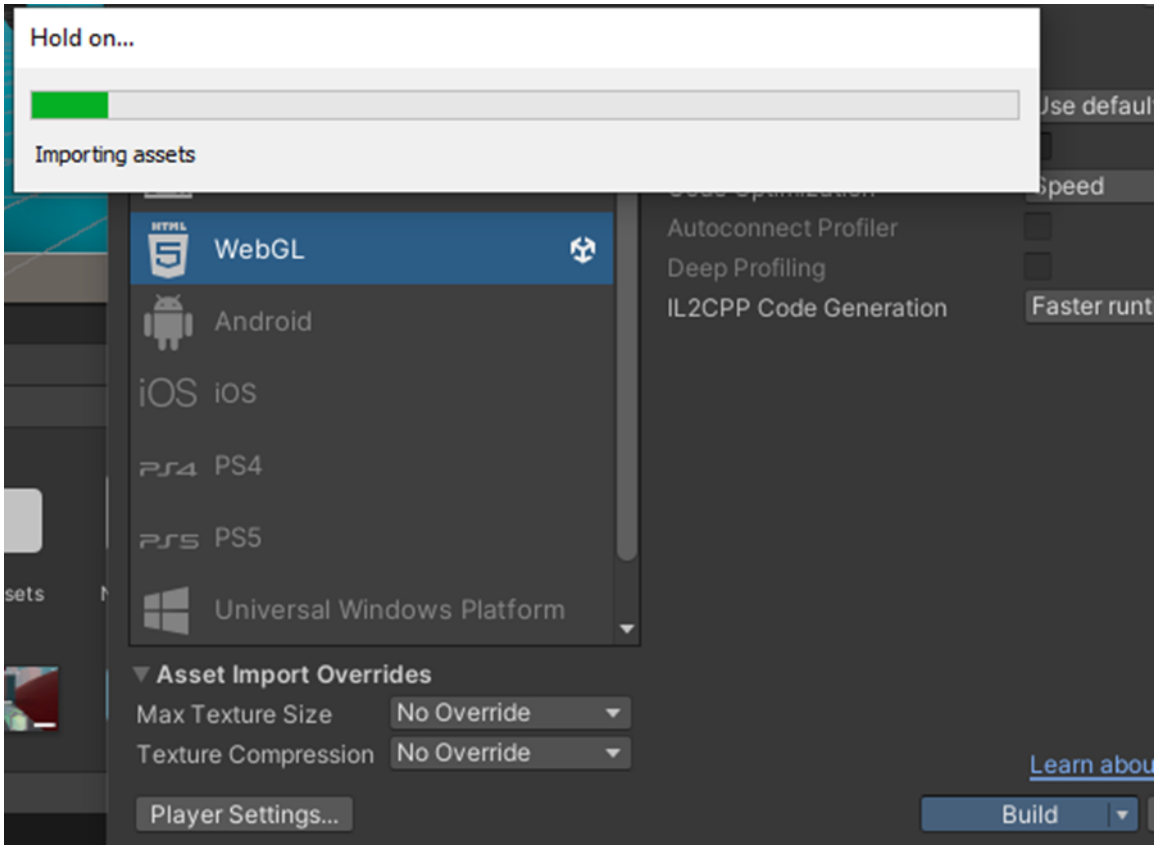

Imagen 23. Botón Build para la construcción de la aplicación web.

Para verificar que el proceso fue terminado con éxito debe de ser revisado en la consola del proyecto donde las salidas deben indicar que el proceso fue exitoso.

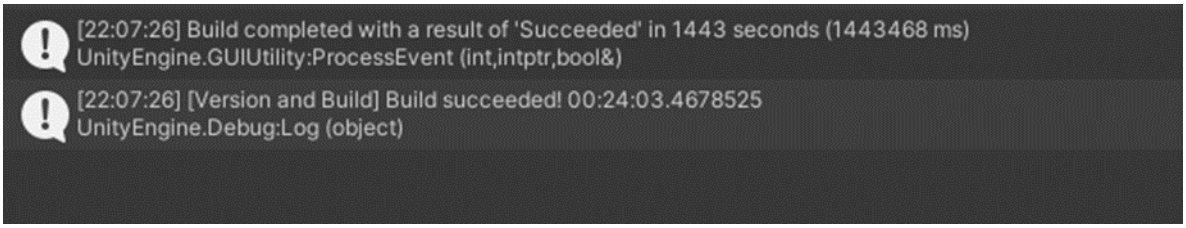

Imagen 24. Resultados en consola de la construcción de la aplicación web.

El proceso fue finalizado con éxito, donde al revisar la carpeta destino especificada se tendrá el siguiente contenido:

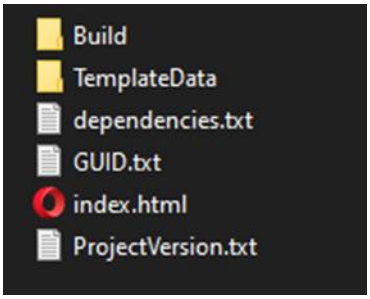

Imagen 25. Archivos resultantes de la exportación.

Si bien se tiene varios archivos, los de interés para despliegue en servidores web son los siguientes.

- **Build**: Carpeta donde se encuentra la construcción del proyecto, así como configuraciones de arranque de la aplicación compatible con navegadores web.
    - **Archivo JSON**: configuración del proyecto y los datos necesarios para cargar la aplicación.
    - **Archivo WASM**: Web assembly de formato binario permitirá ejecutar código de alto rendimiento en el navegador.
    - **Archivo DATA**: Contiene los datos del juego, como texturas, modelos, escenas y otros recursos necesarios para la ejecución del juego. Este archivo es el que más peso tiene luego de la construcción, es recomendable que no tenga un peso mayor a 100 Mb para que sea posible subirlo al GitHub de la plataforma.
    - **Archivo Framework JavaScript**: Contiene el código JavaScript que carga y ejecuta el archivo WASM, manejando la interacción del motor de Unity y el navegador.
    - **Archivo Loader JavaScript**: Responsable de cargar los archivos JSON, WASM y DATA para inicializar la aplicación Unity.
- **TemplateData**: Diseño de la página HTML del juego.
    - **Archivo Style CSS: Contiene los estilos básicos de la página HTML, que pueden modificarse para una interacción y diseño más dinámicos.**
    - **Archivos PNG**: Son imágenes usadas al cargar el juego, funcionan como indicador de que el juego se está cargando y funcionando correctamente.
- **Index.html**: Ventana del juego, es el punto central para el jugador, mostrando el juego en el navegador, usa los archivos de la carpeta Build y el archivo CSS para funcionar correctamente. Posee un elemento canvas WEBGL encargado de renderizar el juego.

Con ello se ha finalizado la construcción de la aplicación web del proyecto de gamificación, ahora solo se requeriría de un servidor web para su despliegue a los usuarios. Para el otro juego de gamificación llamado Batalla Bucal el proceso fue el mismo, pero teniendo que eliminar archivos mp4 y un nivel 3 para que el archivo DATA de la app web construida no superará los 100 Mb. Con eso ambos proyectos de Unity exportados como aplicación para un despliegue mediante un servidor web.

4. Despliegue de la aplicación web con NodeJS

Como parte del proceso de despliegue, primero se verificará el funcionamiento de la aplicación web generada de forma local con un servidor NodeJS como primer prototipo. Puede copiar los archivos y carpetas de interés a un directorio diferente para generar el servidor de forma individual.

Se deberá crear un archivo JavaScript en el mismo directorio que los archivos de la aplicación web, este se llamará server.js. Este archivo contendrá el siguiente contenido.

```
const express = require('express');
const path = require('path');
const app = express();

const PORT = process.env.PORT || 8080;

// Sirve archivos estáticos desde la carpeta raíz
app.use(express.static(path.join(__dirname)));

app.get('/', (req, res) => {
    res.sendFile(path.join(__dirname, 'index.html'));
});

app.listen(PORT, () => {
    console.log(`Servidor escuchando en http://localhost:${PORT}`);
});
```

Imagen 26. Archivo server.js.

Este archivo requerirá de dependencias instaladas con el comando **npm install express** desde la consola en el directorio del proyecto. Ejecutar este comando creara los archivos JSON y una carpeta node_modules que me ayudarán a desplegar la aplicación.

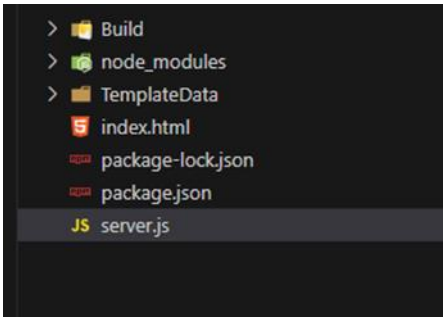

Imagen 27. Archivos de dependencias generados.

En base a estas dependencias en el archivo server.js se crea la variable app que servirá para definir el servidor, así mismo ayuda a usar el index.html en el servidor, este se despliega en el puerto especificado, en este caso será el puerto 8080. Con ello obtendrá la dirección inicial cuando despliegue "/" y un comando que me dirá que el servidor está siendo ejecutado exitosamente, para esto se ejecutara el comando **node server.js** en la consola y se visualizará con la dirección URL en un navegador.

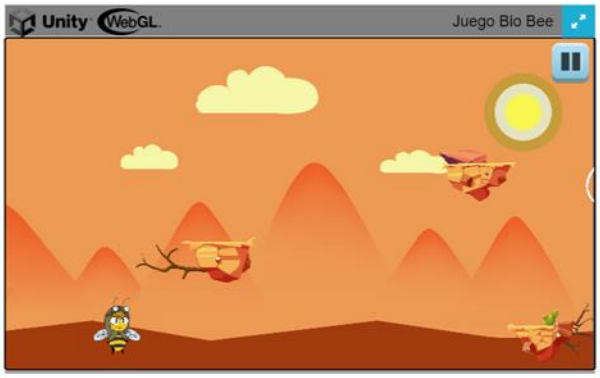

Imagen 28. Despliegue de la aplicación web en el navegador.

Con esto correctamente construido la aplicación web de nuestro proyecto de Unity, también desplegado localmente el resultado con Node.JS para una vista dinámica.

5. Carga de la aplicación web a la plataforma

Una vez se haya confirmado el correcto funcionamiento de la aplicación web de forma local, se deberá desplegar el mismo dentro de la plataforma de Edutainment, la cual usa Angular como Framework para administración de páginas, elementos y diseño de la plataforma. Esta plataforma está a cargo del estudiante German Tobar, por lo que su consultoría de acceso al repositorio es indispensable para obtener acceso al repositorio y hacer cambios; lo primero será copiar el repositorio de forma local, inicializar el git y copiar cambios de la rama principal .

Git clone <URL del git>

Git initialize

Git fetch origin main

Git pull origin main

Una vez obtenido acceso al repositorio se deberá descargar el mismo para ser modificado en base a los requerimientos, sabiendo que se tienen 2 juegos gamificados exportados como aplicaciones web lo ideal sería subirlos como un directorio nuevo en una carpeta assets del repositorio de la plataforma.

Se crea es el componente para visualizar y acceder a los juegos que se han cargado, para esto se debe dirigir en consola a la dirección de la página auth de la plataforma, ingresando el siguiente código.

cd **src/app/components/auth**

Con ello lo siguiente fue crear un componente de los juegos con el comando.

Ng g c games

Si bien se creó el componente, se debe revisar si está dentro de las rutas en auth-routing.module.ts del repositorio para las futuras referencias dentro del proyecto.

Imagen 29. Agregar el componente a las rutas de la plataforma.

Así mismo debe estar en el archivo auth.module.ts como un componente en el @NgModule.

Imagen 30. Instanciar el componente creado.

El componente games tendrá subyacentes de cabecera y cartas que ayudarán a tener un diseño similar a cartas de selector de juegos, estos se realizan ingresando a la ruta de /games desde la consola y aplicar ng g c <componente> para cada uno. Es importante saber que, al no ser principales, sus archivos spec.ts deben ser eliminados mientras que en sus archivos TypeScript deben eliminarse los atributos **standalone,** que los vuelve individuales de su componente padre. En auth.module.ts deben ser importados como componentes del proyecto.

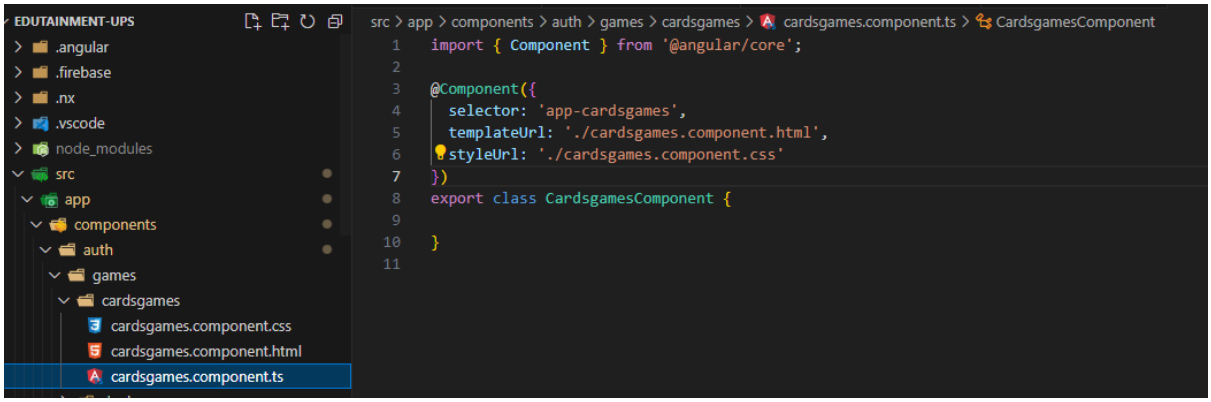

Imagen 31. Creación y configuración de los subcomponentes.

Con los subcomponentes de games, en el index.html del componente padre solo es necesario instanciar estos mediante elementos a mostrar en la misma página.

```
Go to component
<!DOCTYPE html>
<html lang="en">

<app-header-index></app-header-index>

<body class="hold-transition sidebar-collapse layout-top-nav">
  <div class="wrapper">
    <app-super-nav></app-super-nav>
    <app-navbar-index></app-navbar-index>
    <app-header-games></app-header-games>
    <app-cardsgames></app-cardsgames>
    <app-footer-index></app-footer-index>
  </div>
</body>

</html>
```

Imagen 32. Índice de la página de juegos usando sus subcomponentes.

Con esto, el diseño de la cabecera es similar a la página de proyectos de la plataforma, solo cambiando el tema asignado de los juegos.

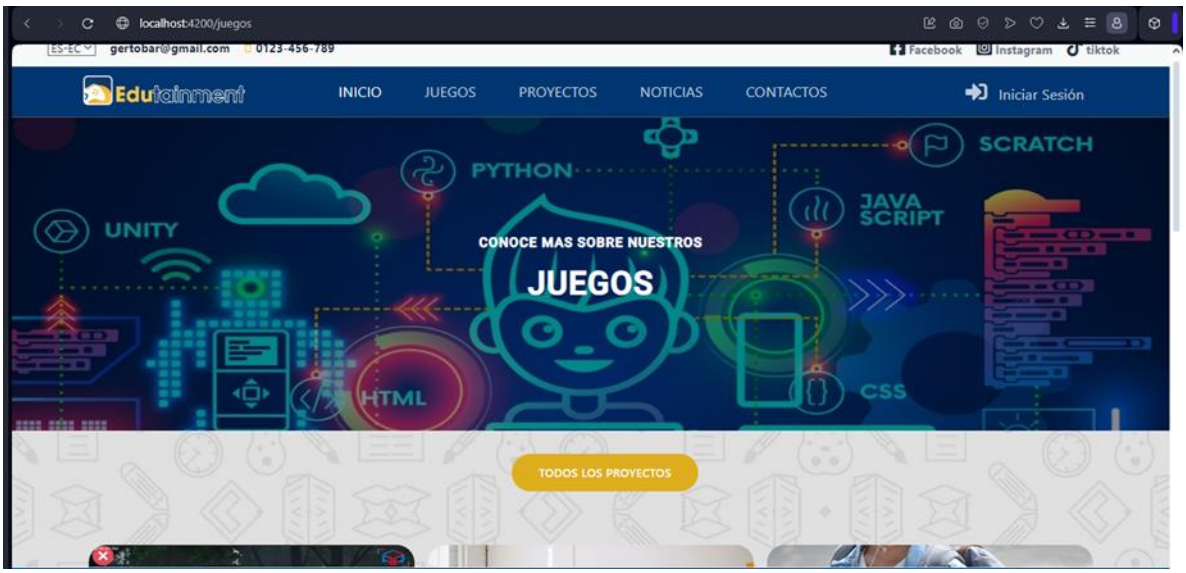

Imagen 33. Página de los juegos localmente.

Si bien el diseño está correcto, la funcionalidad de las cartas es debe ser configurado para el redireccionamiento a los juegos específicos, en este caso cada carta llevara a un index.html específico para cada juego. Esta sección es la que los desarrolladores de los videojuegos pueden usar para subir sus juegos ya exportados como aplicaciones web, sabiendo que la plataforma ya cuenta con la sección de juegos.

Lo primordial sería subir los juegos a la carpeta de assets, dividido en carpetas para cada juego, dentro de estas estarán los documentos de build, templateData y el index.html respectivamente para cada juego.

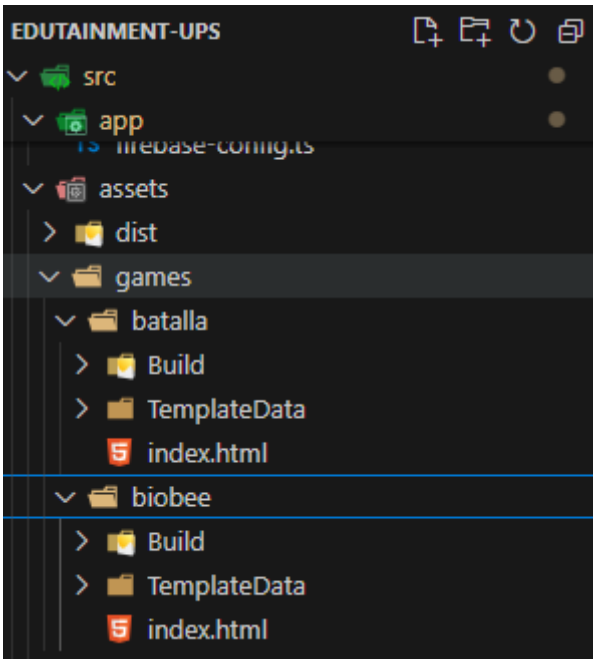

Imagen 34. Repositorio de los juegos.

Con esto se deben instanciar en cada carta un redireccionamiento a los HTML de cada juego, así mismo en assets se debería tener una miniatura para indicar al jugador que es lo que va a desplegar, también una descripción breve del juego con datos.

```
<section class="meetings-page" id="meetings">
    <div class="container">
      <div class="row">
        <div class="col-lg-12">
          <div class="row">
            <div class="col-lg-12">
              <div class="row grid">
                <div class="col-lg-4 templatemo-item-col all soon">
                  <div class="meeting-item">
                    <div class="thumb">
                      <!-- Cambiarlo por un redireccionamiento a un despliegue de juego-->
                      <a href="assets/games/biobee/index.html" >
                        <img src="../../../../../assets/dist/img/images/BioBeePrev.jpg"
                          alt=""></a>
                    </div>
                    <div class="down-content">
                      <a href="assets/games/biobee/index.html" >
                        <h4>Juego Bio Bee</h4>
                      </a>
                      <p>Videojuego gamificado por<br>estudiantes de diseño multimedia. <br>
```

Imagen 35. Carta de juego.

Con esto si se selecciona la carta, esta redirecciona al HTML del juego especificado, este puede ser modificados para tener un formato acorde a la plataforma, usando assets ya vistos, ajustar color, botones de regreso y redimensionamiento.

El redimensionado de la ventana de juego por defecto esta activado para que el WEBGL siempre se ajuste al canvas, lo que es un problema al momento de ajustar tamaño de la ventana de Unity Player, para desactivarlo debe ser descomentado la línea de configuración que desactiva el redimensionado al canvas.

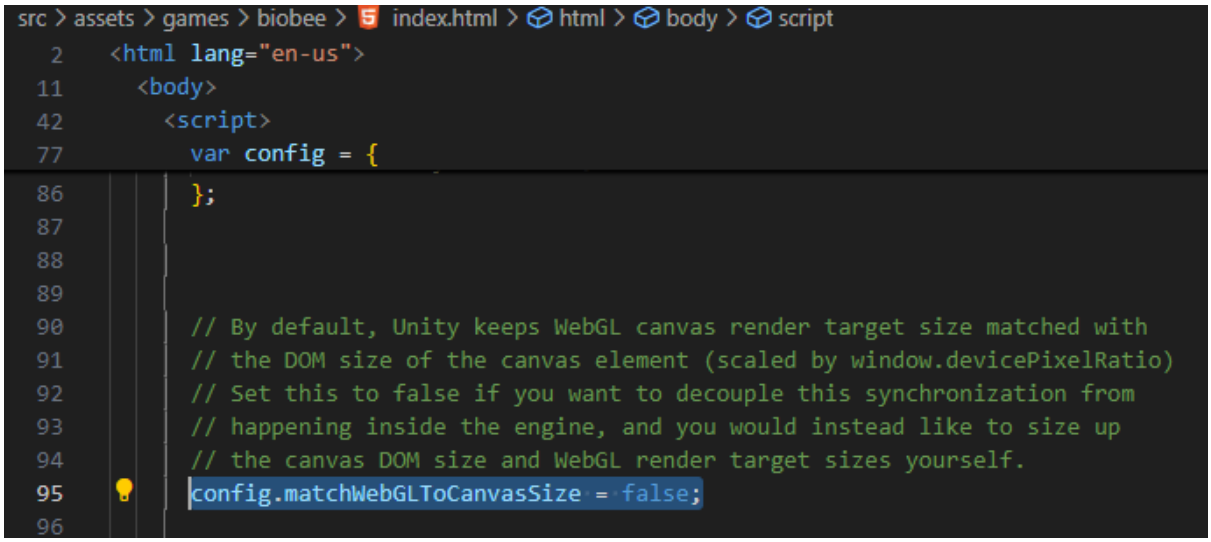

Imagen 36. Configuración para redimensionar la ventana de juego.

Con esta configuración, es posible diseñar el HTML del juego a disposición, en este caso se agregó una descripción y controles del juego, además de un botón para volver a la página de selección.

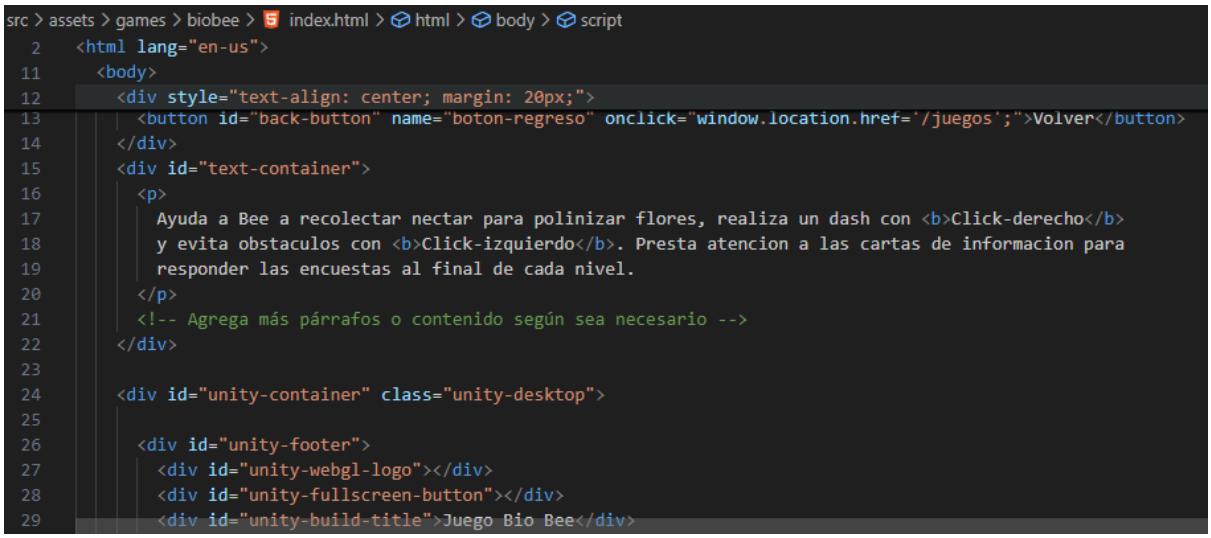

Imagen 37. Texto y botón de regreso en la ventana de juego.

Este proceso es aplicable para el diseño, con configuración de estilo en /TemplateData/style.css para usar ajustar el diseño de los elementos. También es usado para mover la ventana del contenedor de Unity a una posición más dinámica para el disfrute del jugador.

```
src > assets > games > biobee > TemplateData > style.css > #unity-container.unity-desktop
1   body { padding: 0; margin: 0; background-image: url(../../../../../assets/dist/img/images/proyectos-bg.jpg); }
2   #unity-container {
3       position: absolute ;
4       width: 50%; /* Ajustar el tamaño del contenedor al 50% del ancho de la pantalla */
5       aspect-ratio: 16 / 9; /* Mantener la relación de aspecto 16:9 */
6       background: grey;
7   }
8   #unity-container.unity-desktop { left: 30%; top: 50%; transform: translate(-50%, -50%) }
9   #unity-container.unity-mobile { position: fixed; width: 100%; height: 100% }
10  #unity-canvas { background: #FFFFFF }
11  .unity-mobile #unity-canvas { width: 100%; height: 100% }
12  #unity-loading-bar { position: absolute; left: 50%; top: 50%; transform: translate(-50%, -50%); display: none }
13  #unity-logo { width: 154px; height: 130px; background: url('unity-logo-dark.png') no-repeat center }
14  #unity-progress-bar-empty { width: 141px; height: 18px; margin-top: 10px; margin-left: 6.5px; background: url('progress
15  #unity-progress-bar-full { width: 0%; height: 18px; margin-top: 10px; background: url('progress-bar-full-dark.png') no-
16  #unity-footer { position: relative }
17  .unity-mobile #unity-footer { display: none }
18  #unity-webgl-logo { float:left; width: 204px; height: 38px; background: url('webgl-logo.png') no-repeat center }
19  #unity-build-title { float: right; margin-right: 10px; line-height: 38px; font-family: arial; font-size: 18px }
20  #unity-fullscreen-button { cursor:pointer; float: right; width: 38px; height: 38px; background: url('fullscreen-button.
```

Imagen 38. Estilo de los elementos en el HTML.

Para visualizar los cambios correctos se deberá hacer un despliegue de la plataforma de forma local con el comando.

ng serve

Esto construirá la aplicación de Angular para ser visualizada en un puerto desde un navegador. Si se accede al componente de juegos se visualizarán las cartas a los juegos que han sido cargados como aplicaciones web.

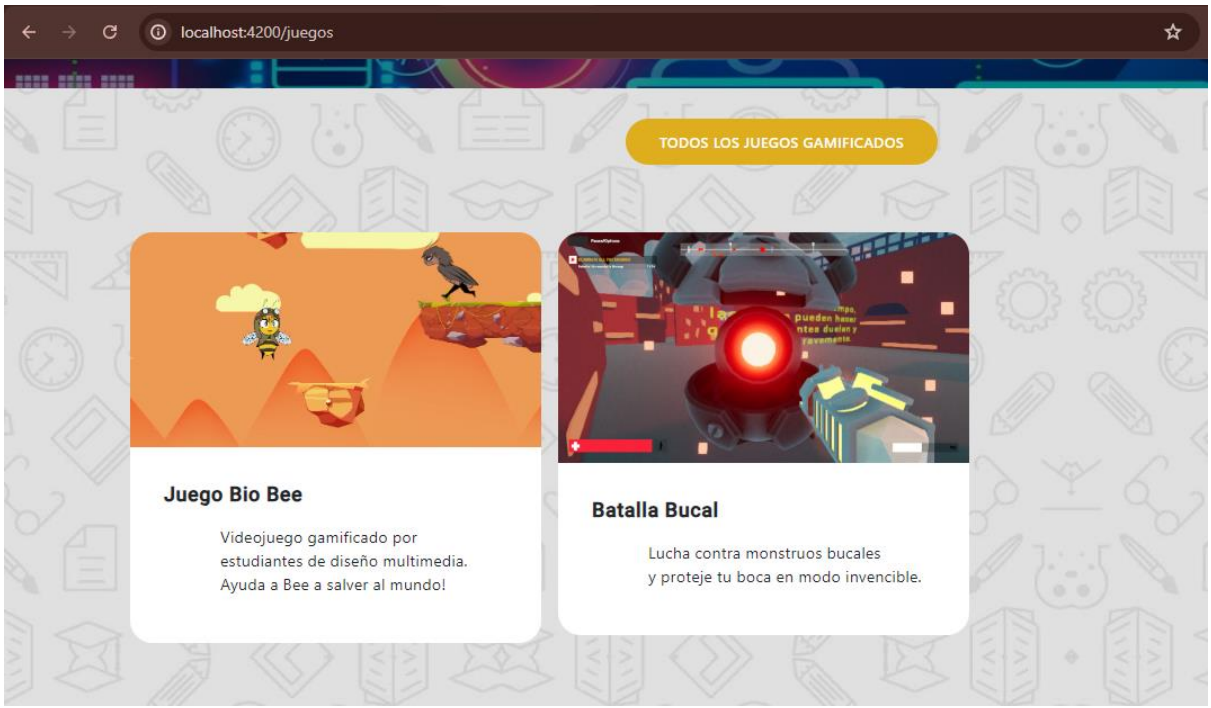

Imagen 39. Componente Juegos de la plataforma localmente.

Si se selecciona el juego este desplegará el mismo mediante el HTML que fue modificado. En este caso se seleccionó el juego BioBee.

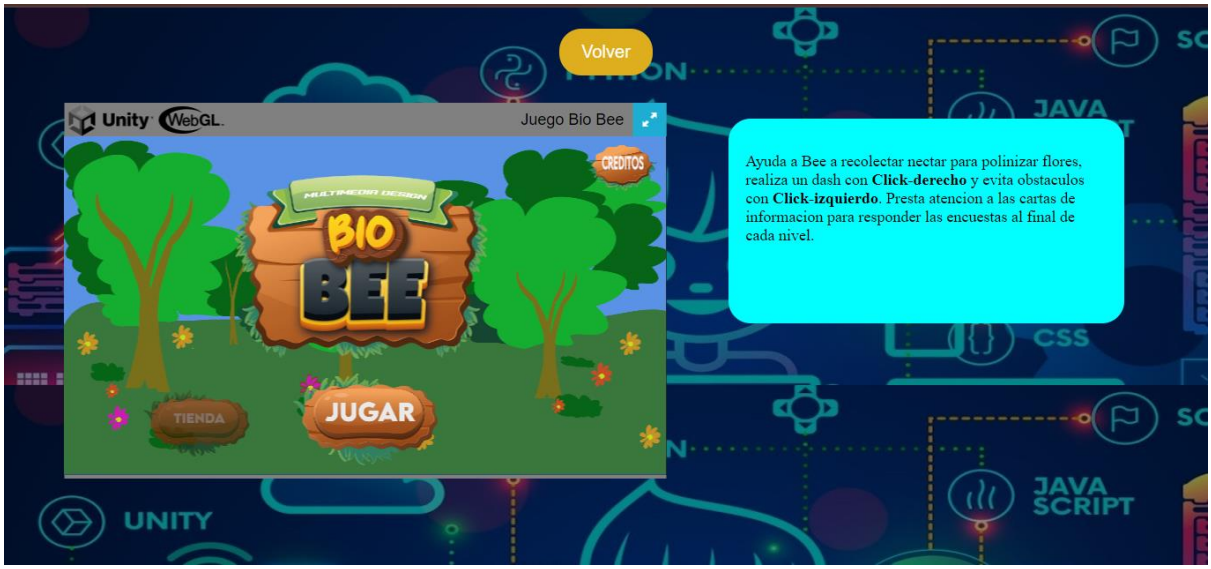

Imagen 40. Juego BioBee.

Lo mismo sucederá con el juego de batalla bucal.

Imagen 41. Juego Batalla Bucal.

Con los juegos desplegados localmente en la plataforma, ahora solo es necesario hacer un git Commit y git Push al repositorio de la plataforma para que la misma contenga los cambios realizados. Esto es útil para los desarrolladores, pues con esto podrán respaldar y compartir sus proyectos para que más personas jueguen los frutos de su esfuerzo.

Estos cambios por seguridad se subirán dentro de una rama asignada para no alterar el repositorio principal(main) por lo que es necesario cambiar la rama del repositorio a una rama generada para la carga de los cambios.

Git checkout <rama>

Con esto es seguro realizar un commit de los cambios realizados al repositorio. Por lo que debe seguirse el siguiente grupo de comandos de Git.

Git add .

Git commit –m "<nombrecommit>"

Git push origin <rama>

Con esto los cambios deberán ser cargados a la rama principal por los delegados a administrar la misma, si todo sale bien, los juegos podrán ser jugables desde el siguiente enlace donde está la plataforma de Edutainment https://edutainment.firebaseapp.com/juegos.

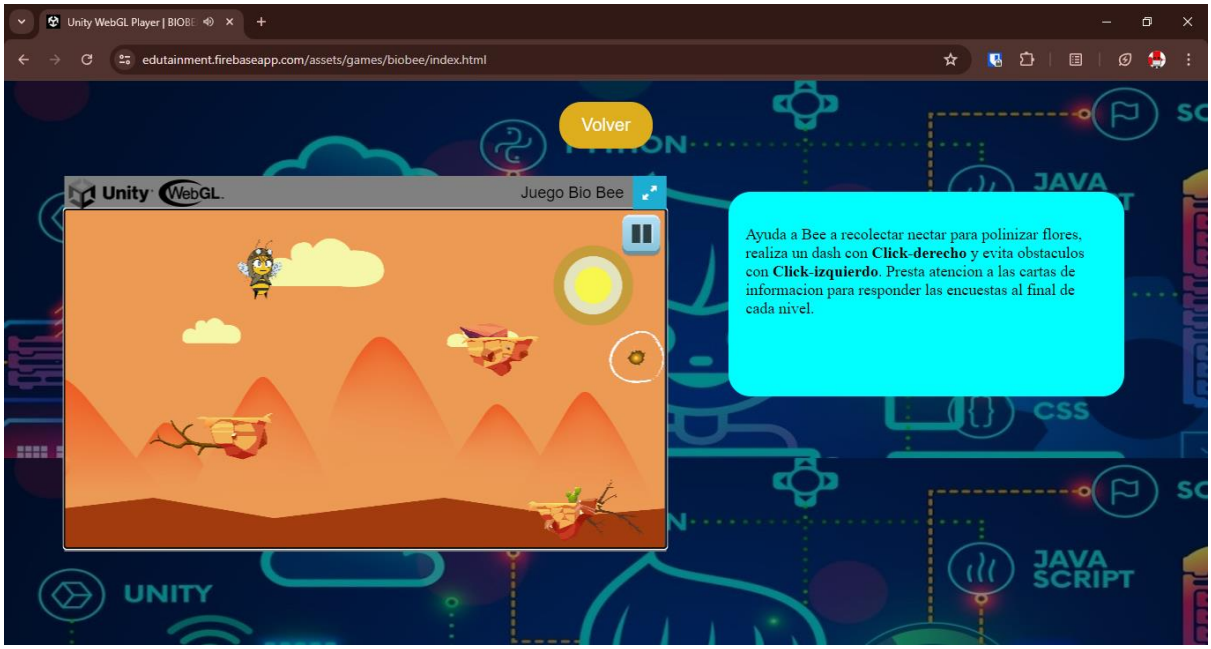

Imagen 42. Plataforma edutainment modificada.

Los 2 proyectos de Unity exportados como aplicaciones web se desplegaron correctamente dentro de la plataforma de Edutainment mediante usos de repositorio y conocimientos de Angular con las aplicaciones a desplegar, estos juegos son disfrutables desde la misma plataforma sin la necesidad de descargarlos, promoviendo el uso de la plataforma como un repositorio para que los estudiantes de Diseño Multimedia puedan cargar sus juegos en base al documento como una guía para que en el futuro los juegos sean conocidos y salvaguardados con el paso del tiempo.

El juego titulado "BioBee" funciona en base a clic izquierdo de usuario, como lo era el juego de Flappy Bird, además de un impulso con clic derecho. Con ello el juego es totalmente funcional y adaptado a la plataforma, un juego que incita a conocer más sobre las abejas y su importancia en el ecosistema.

El segundo juego cargado "Batalla Bucal" es un juego estilo "Shooter" donde el personaje debe enfrentarse a enemigos de caries dentro de un entorno bucal, este juego conciencia sobre los problemas bucales, siendo un videojuego perfecto para pasar el rato explorando zonas y derrotando enemigos constantemente.

Las aplicaciones web poseen un mínimo de 30 fotogramas por segundo, lo que los hace regulares para una plataforma web, lo cual permite a los jugadores una experiencia sin

inconvenientes. Así mismo el redimensionado visto del contenedor de Unity WebGL usa una resolución y posicionamiento del 40% para que el jugador pueda ver los consejos y jugar el juego sin molestia. Mientras que, la carga de los juegos es rápida para el juego BioBee debido a su contenido en 2D, mientras que Batalla Bucal tarda un poco más debido a que usa recursos para cargar niveles en 3D.

Ambos juegos, además de entretener, enseñan a sus jugadores conceptos importantes como el ecosistema e higiene bucal, siendo el objetivo que se espera obtener con la gamificación, pues no serán los únicos cargados de la plataforma, pues el documento sirve como una guía para que los futuros desarrolladores no tengan problemas al cargar sus proyectos dentro de la plataforma, permitiendo la enseñanza con el entretenimiento paralelamente.

Cronograma

OE1. Investigar sobre las herramientas de desarrollo de videojuegos Unreal y Unity; generación de aplicaciones web en los diferentes ámbitos.

1	Conocer el proyecto con el docente Gustavo Bravo.
2	Investigar los softwares de creación de videojuegos a montar dentro de la plataforma.
3	Familiarizarse con el entorno de trabajo y la plataforma de Edutainment.
4	Investigar sobre cómo subir los juegos hechos con Unreal Engine y Unity como aplicación web.

OE2. Modificar las aplicaciones que no cuenten con microservicios para el registro de información para posteriormente crearlas.

1	Obtener el código del juego a modificar y analizar.
2	Identificar carencias del videojuego, como la falta de microservicios de registro de información.
3	Exportar los videojuegos gamificados como aplicaciones web mediante Unity y Unreal Engine.

OE3. Implementar los microservicios a la aplicación y a la plataforma de Edutainment; obteniendo los registros de usuario respectivos.

1	Basarse en los microservicios ya hechos para crear uno nuevo para la aplicación web a montar.
2	Verificar si el código no afecta al entorno de la plataforma de Edutainment
3	Implementar las aplicaciones web dentro de la plataforma web por ventanas dinámicas.

OE4. Testear y evaluar el Frontend de la plataforma de Edutainment con los microservicios, aplicar cambios de ser necesarios para su posterior despliegue.

1	Verificar el rendimiento de la aplicación web dentro de la plataforma de Edutainment.
2	Testeo de la recolección de datos de usuario para su posterior presentación.
3	Verificar que es posible subir más de un solo videojuego por sesión de jugador.
4	Evaluado final según los requerimientos del proyecto.

Tabla de Actividades:

Objetivo	Actividad	Fecha inicio	Fecha fin	Horas por responsable	Responsables
OE1	Ac.1	21/03/2024	21/03/2024	5	Christian Buestan
OE1	Ac.2	28/03/2024	04/04/2024	15	Christian Buestan
OE1	Ac.3	11/04/2024	15/04/2024	15	Christian Buestan
OE1	Ac.4	28/04/2024	15/04/2024	10	Christian Buestan
OE2	Ac.1	17/04/2024	17/04/2024	5	Christian Buestan
OE2	Ac.2	17/04/2024	24/04/2024	10	Christian Buestan
OE2	Ac.3	24/04/2024	15/05/2024	30	Christian Buestan
OE3	Ac.1	15/05/2024	20/05/2024	10	Christian Buestan
OE3	Ac.2	28/04/2024	30/05/2024	40	Christian Buestan
OE3	Ac.3	30/05/2024	10/06/2024	20	Christian Buestan
OE4	Ac.1	12/06/2024	18/06/2024	10	Christian Buestan, Gustavo Bravo
OE4	Ac.2	18/06/2024	20/06/2024	10	Christian Buestan, Gustavo Bravo
OE4	Ac.3	21/06/2024	26/06/2024	10	Christian Buestan, Gustavo Bravo
OE4	Ac.4	27/06/2024	12/07/2024	10	Christian Buestan, Gustavo Bravo

Presupuesto

DENOMINACIÓN	CANT.	COSTO UNITARIO	COSTO TOTAL
	Unidades	Dólares	Dólares
1. BIENES			
Papel Bond-A	1	0.00	0.00
Copias	1	0.30	0.30
2. Tecnológico			
Computadora portátil	1	700.00	700.00
3. Servicios			
Servicio de transporte	100	1.40	140.00
Servicio de Internet	1	30.00	30.00
4. Personal			
Estudiante/Desarrollador	200 horas	25.00 por hora	5000.00
5. Otros			
Imprevistos	1	50.00	50.00
TOTAL			5920.30

Conclusiones

El desarrollo del proyecto de titulación ha sido un proceso fructífero que ha permitido alcanzar varios objetivos clave en la difusión y despliegue de videojuegos gamificados dentro de la plataforma Edutainment. A continuación, se presentan las conclusiones más relevantes del proyecto:

Implementación Exitosa de Aplicaciones Web: Se lograron desplegar dos proyectos de gamificación, "BioBee" y "Batalla Bucal", como aplicaciones web funcionales dentro de la plataforma Edutainment. Este despliegue facilita el acceso y uso de los videojuegos sin necesidad de descargas adicionales, promoviendo una mayor adopción y participación por parte de los usuarios.

Optimización y Adaptación Técnica: El proceso de exportación y configuración de los proyectos en Unity y Unreal Engine fue optimizado para asegurar compatibilidad y rendimiento en la plataforma web. Las configuraciones específicas, como la compresión de archivos y la selección de resoluciones, fueron cruciales para obtener una experiencia de usuario fluida y atractiva. Ambos juegos desplegados en la plataforma poseen una tasa de 30 fotogramas por segundo para una experiencia más satisfactoria de parte de los usuarios de la plataforma.

Integración de Microservicios: La implementación de microservicios permitió una gestión eficiente y escalable de las aplicaciones web. Esto no solo mejoró la capacidad de respuesta del sistema, sino que también facilitó la administración de sesiones y el registro de datos de usuarios, contribuyendo a un entorno más robusto y dinámico.

Desarrollo de una Guía Práctica para Futuros Desarrolladores: Se elaboró una guía detallada que documenta el proceso de exportación y despliegue de aplicaciones web en la plataforma Edutainment. Esta guía es un recurso valioso para futuros desarrolladores, proporcionando instrucciones claras y mejores prácticas para asegurar la calidad y eficiencia de sus proyectos.

Contribución al Enriquecimiento Educativo y Cultural: Los videojuegos desarrollados no solo ofrecen entretenimiento, sino que también incorporan elementos educativos y culturales significativos. "BioBee" educa a los jugadores sobre la importancia de las abejas en el ecosistema, mientras que "Batalla Bucal" concientiza sobre la higiene bucal. Estos juegos contribuyen a un aprendizaje significativo y al enriquecimiento cultural de los usuarios.

Posicionamiento de la Universidad Politécnica Salesiana como Líder Innovador: La creación y el despliegue de la plataforma Edutainment posicionan a la Universidad Politécnica Salesiana como una institución innovadora en la adopción de tecnologías avanzadas para la educación. Este proyecto no solo mejora la experiencia educativa de los estudiantes actuales, sino que también atrae a futuros estudiantes interesados en carreras tecnológicas.

Desafíos y Recomendaciones para Futuras Implementaciones: Aunque el proyecto fue exitoso, se identificaron desafíos, como la exportación de aplicaciones web desde Unreal Engine 5, que requiere soluciones adicionales. Se recomienda seguir trabajando en la compatibilidad de motores gráficos y explorar nuevas tecnologías para mejorar aún más la plataforma Edutainment.

El proyecto de difusión y despliegue de videojuegos gamificados en la plataforma Edutainment ha demostrado ser una iniciativa valiosa y efectiva. No solo ha logrado sus objetivos técnicos y educativos, sino que también ha establecido una base sólida para futuras expansiones y mejoras. La plataforma Edutainment se presenta como un recurso dinámico y accesible que

incentiva el aprendizaje y el entretenimiento, beneficiando a estudiantes, educadores y desarrolladores por igual.

Recomendaciones

El presente trabajo no concretó la forma de exportar la aplicación hecha con el motor de Unreal Engine 5, por lo que es recomendable esperar a los desarrolladores para tener, a sus palabras, una versión exportable como aplicación web pueda ser desplegada dentro de la plataforma de Edutainment.

Compatibilidad con Unreal Engine 5: Se recomienda continuar trabajando en la compatibilidad y exportación de proyectos desarrollados con Unreal Engine 5. Es crucial investigar y probar diferentes métodos de exportación que permitan desplegar aplicaciones web sin comprometer la calidad o funcionalidad del juego. Esto podría incluir colaboraciones con los desarrolladores del motor o la utilización de herramientas adicionales que faciliten el proceso.

Desarrollo de un Sistema de Registro de Jugadores: Implementar un sistema de registro y autenticación de jugadores en la plataforma Edutainment es fundamental para mejorar la experiencia del usuario y la recopilación de datos. Este sistema debe permitir a los jugadores crear perfiles, guardar su progreso y competir en tablas de clasificación, incentivando así el uso continuo y el aprendizaje a largo plazo.

Integración de un Sistema de Puntaje: Es recomendable que los futuros desarrolladores de videojuegos incluyan sistemas de puntaje y retroalimentación dentro de sus juegos. Esto no solo mejorará la experiencia del usuario, sino que también permitirá a la plataforma recopilar datos valiosos sobre el rendimiento y las preferencias de los jugadores. Además, estos datos pueden utilizarse para realizar análisis y mejorar los contenidos educativos.

Actualización Continua de la Plataforma: La plataforma Edutainment debe ser actualizada regularmente para incluir nuevas funcionalidades y mejorar la experiencia del usuario. Esto incluye la optimización de la interfaz de usuario, la incorporación de nuevos juegos y la mejora de la compatibilidad con diferentes dispositivos y navegadores. Mantener la plataforma actualizada asegurará su relevancia y atractivo para los usuarios.

Capacitación y Soporte para Desarrolladores: Ofrecer talleres y recursos educativos para capacitar a los desarrolladores en el uso de Unity, Unreal Engine y la plataforma Edutainment. Esto ayudará a asegurar que los desarrolladores estén bien equipados para crear y desplegar juegos de alta calidad. Además, proporcionar soporte técnico continuo puede ayudar a resolver problemas rápidamente y fomentar una comunidad activa de desarrolladores.

Expansión del Contenido Educativo y Cultural: Fomentar la creación de nuevos juegos que integren contenido educativo y cultural relevante. Colaborar con expertos en educación y cultura para desarrollar juegos que aborden temas importantes y reflejen las tradiciones y conocimientos locales. Esto enriquecerá la plataforma y proporcionará a los usuarios una variedad más amplia de contenidos educativos.

Monitoreo y Evaluación de la Plataforma: Implementar un sistema de monitoreo y evaluación para medir el impacto y la eficacia de la plataforma Edutainment. Recopilar y analizar datos

sobre el uso de la plataforma, el rendimiento de los juegos y la satisfacción del usuario. Estos datos pueden utilizarse para realizar mejoras continuas y asegurar que la plataforma cumpla con sus objetivos educativos.

Promoción y Difusión de la Plataforma: Desarrollar estrategias de marketing y comunicación para promover la plataforma Edutainment entre estudiantes, educadores y el público en general. Utilizar redes sociales, eventos educativos y colaboraciones con otras instituciones para aumentar la visibilidad y el uso de la plataforma. Una mayor difusión ayudará a atraer más usuarios y fomentar una comunidad vibrante y participativa.

Exploración de Nuevas Tecnologías: Mantenerse al tanto de las nuevas tecnologías y tendencias en el desarrollo de videojuegos y la educación. Explorar el uso de tecnologías emergentes como la realidad aumentada (AR), la realidad virtual (VR) y la inteligencia artificial (IA) para crear experiencias de aprendizaje más inmersivas e interactivas. La adopción de estas tecnologías puede posicionar a la plataforma Edutainment a la vanguardia de la innovación educativa.

Fomento de la Colaboración Interdisciplinaria: Promover la colaboración entre diferentes disciplinas dentro de la universidad, incluyendo diseño, educación, tecnología y cultura. Esta colaboración puede llevar a la creación de juegos más integrales y enriquecedores que aborden múltiples aspectos del aprendizaje y el desarrollo personal. Además, puede fomentar un entorno de aprendizaje más holístico y creativo.

Referencias bibliográficas

Estela N, Yeray S, Rafael R. (2020). *Los videojuegos en la educación: Beneficios y perjuicios*. Revista Electrónica Educare, vol. 24, p. 240-257. Universidad Nacional. CIDE. Recuperado de: https://www.redalyc.org/journal/1941/194163269012/html/

Intel. (junio de 2021). *Microservicios y arquitectura de microservicios*. intel.la. Recuperado de: https://www.intel.la/content/www/xl/es/cloud-computing/microservices.html

Villalobos, G. M., Sánchez, G. D. C., & Gutiérrez, D. A. B. (2010). *Diseño de framework web para el desarrollo dinámico de aplicaciones*. Scientia et technica, 16(44), 178-183. Recuperado de: https://www.redalyc.org/pdf/849/84917316032.pdf

Surática. (2014). *¿Qué es un Framework?* Suratica.es. Recuperado de: https://www.suratica.es/que-es-un-framework/

Anchundia Medrano, L. A. (2022). *Análisis comparativo de tecnologías Front End Angular Js Vs React Js, en el modelo de procesos para el desarrollo de aplicaciones web* (Bachelor's thesis, Babahoyo: UTB-FAFI. 2022). Recuperado de: http://dspace.utb.edu.ec/bitstream/handle/49000/11386/E-UTB-FAFI-SIST-000271.pdf

Haro, E., Guarda, T., Peñaherrera, A. O. Z., & Quiña, G. N. (2019). *Desarrollo backend para aplicaciones web, servicios web restful: Node. js vs spring boot. Revista Ibérica de Sistemas e Tecnologias de Informação*, (E17), 309-321. Recuperado de: https://www.proquest.com/openview/a78cfaa62708fd24f38ac8d1025050eb/1?pq-origsite=gscholar&cbl=1006393

Gregory, J. (2018). *Game engine architecture. AK Peters/CRC Press*. Recuperado de: https://www.taylorfrancis.com/books/mono/10.1201/9781315267845/game-engine-architecture-third-edition-jason-gregory

Delgado M. (12 de febrero de 2021). *Los 11 mejores motores gráficos para introducirse en el desarrollo de videojuegos*. Vandal.elespanol. Recuperado de: https://vandal.elespanol.com/reportaje/los-11-mejores-motores-graficos-para-introducirse-en-el-desarrollo-de-videojuegos

Sanders, A. (2016). *An introduction to Unreal engine 4.* AK Peters/CRC Press. Recuperado de: https://www.taylorfrancis.com/books/mono/10.1201/9781315382555/introduction-unreal-engine-4-andrew-sanders

Nicoll, B., Keogh, B., Nicoll, B., & Keogh, B. (2019). *The Unity game engine and the circuits of cultural software (pp. 1-21)*. Springer International Publishing. Recuperado de: https://link.springer.com/chapter/10.1007/978-3-030-25012-6_1

López, D., & Maya, E. (2017). *Arquitectura de Software basada en Microservicios para Desarrollo de Aplicaciones Web.* Recuperado de: http://138.59.13.30/bitstream/10786/1277/1/93%20Arquitectura%20de%20Software%20basada%20en%20Microservicios%20para%20Desarrollo%20de%20Aplicaciones%20Web.pdf

Luis J. (8 de julio de 2022). *Microservices Architecture for Solving Business Needs*. Krasamo. Recuperado de: https://www.krasamo.com/microservices-architecture-for-real-world-business-needs/

Espinosa, R. S. C. (2017). *GAMIFICACION EN ESCENARIOS EDUCATIVOS. REVISANDO LITERATURA PARA ACLARAR CONCEPTOS*. Experiencias de gamificación en aulas, 15, 11-17. Recuperado de: https://gc.scalahed.com/recursos/files/r161r/w25565w/L1TE105_S3_RUBIO.pdf

Hernández, J. (2014). *Análisis y desarrollo web*. Jesús Hernández. Recuperado de: https://books.google.es/books?hl=es&lr=&id=nYDVBQAAQBAJ&oi=fnd&pg=PA4&dq=An%C3%A1lisis+y+desarrollo+web&ots=wSAs4KfzwF&sig=qhwTDnvp-SVd8iYXieAsCcKrUL4#v=onepage&q=An%C3%A1lisis%20y%20desarrollo%20web&f=false

Cobo, L. M. A. (2004). *Servicios WEB: Distribución e integración. Sistemas y Telemática*, 2(4), 107-120. Recuperado de: https://aplicaciones.icesi.edu.co/revistas/index.php/sistemas_telematica/article/download/937/962

Wallden, S., & Soronen, A. (2004). *Edutainment. From Television and Computers to Digital Television.* Recuperado de: https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=deb575d79d4449b536978b5c19168c5ab42f46f4

Carrillo, A. G. (2005). *Fundamentos de programación en C++*. Delta Publicaciones. Recuperado de: https://books.google.es/books?hl=es&lr=&id=OC17arE5xukC&oi=fnd&pg=PA1&dq=Fundamentos+de+programaci%C3%B3n+en+C%2B%2B&ots=pxjNlqxsFA&sig=MPirUKEiu4KXV6vvIGZcsNkKSj8

Ríos, J. R. M., Ordóñez, M. P. Z., Segarra, M. J. C., & Zerda, F. G. G. (2017). *Estado del arte: Metodologías de desarrollo en aplicaciones web*. 3c Tecnología: glosas de innovación aplicadas a la pyme, 6(3), 54-71. Recuperado de: https://dialnet.unirioja.es/servlet/articulo?codigo=6143045

Astigarraga, J., & Cruz-Alonso, V. (2022). *¡ Se puede entender cómo funcionan Git y GitHub!*. Ecosistemas, 31(1), 2332-2332. Recuperado de: https://www.revistaecosistemas.net/index.php/ecosistemas/article/download/2332/1505

Maida, EG, Pacienzia, J. (diciembre de 2015). *Metodologías de desarrollo de software.* Universidad Católica Argentina, 2015. Recuperado de: https://repositorio.uca.edu.ar/handle/123456789/522

Anexos

Anexo 1: Manual de desarrollo.

MANUAL DE DESARROLLADORES: UNITY A WEBGL

POR CHRISTIAN BUESTAN

JUSTIFICACION

El presente manual fue diseñado para que los desarrolladores que usan Unity puedan exportar correctamente sus juegos gamificados a formato de aplicación web, esto ayuda a su carga dentro de plataformas de interacción sin la necesidad de descarga o importación de parte de los jugadores.

CARGA DEL JUEGO

Se requiere que el juego sea abierto desde una unidad C:/ accesible por el programa de Unity Editor, esto para que la exportación sea libre de inconvenientes. En este documento se usa el proyecto Bio Bee como ejemplo a compilar la aplicación web. Este será cargado dentro de Unity Hub como un proyecto para ser exportado como aplicación web, tener cuidado con las versiones requeridas, si es necesario instalar las versiones especificadas por el mismo editor.

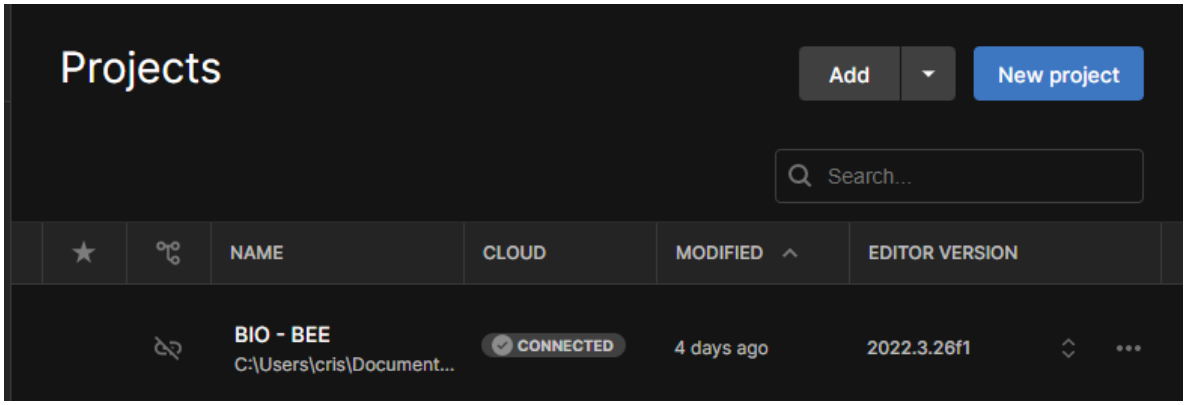

Ilustración 1. Carga correcta de un proyecto de Unity.

Tomar en cuenta que el proyecto es en base a un archivo ya terminado, su proyecto debe ser jugable para poder aplicar los pasos en este documento, en caso de usarse este proceso dentro de un proyecto corrupto o no funcional, se recomienda solventar dichos errores antes de la exportación a aplicación web.

CONFIGURACIONES DEL PROYECTO

El proyecto debe de ser configurado de la siguiente manera para que la exportación como aplicación web sea exitosa y libre de errores.

1. Modificar las configuraciones de jugador.
   En Edit>Project Settings>Player buscar el campo Publishing Settings donde estará el selector Compression Format que permite la compresión del juego al momento de construirlo, esto debe estar deshabilitado para que la aplicación web sea compatible con servidores web.

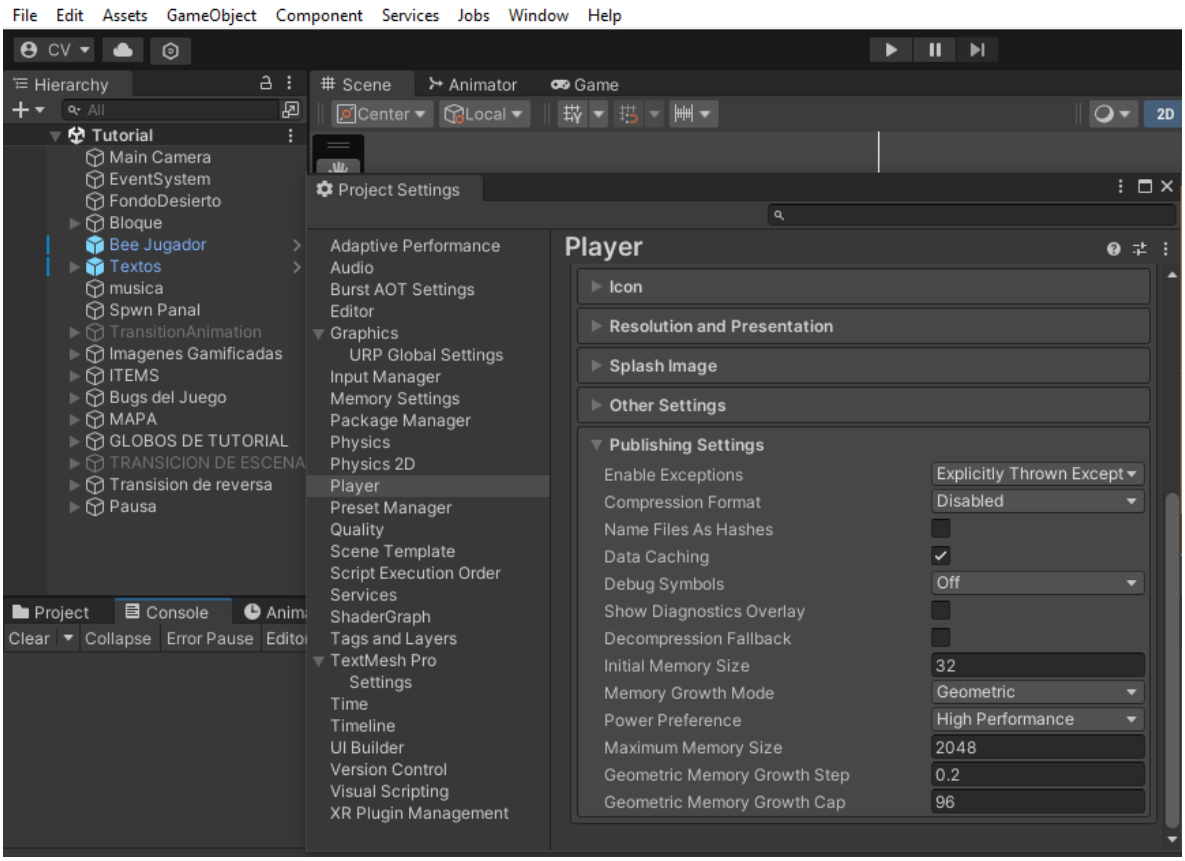

Ilustración 2. Configuración de compresión del proyecto.

2. Selección de resolución del juego.
   En la pestaña de Game se debe seleccionar la resolución que tendrá el juego, se selecciona Full HD para que el juego se vea mejor sin importar el tamaño.

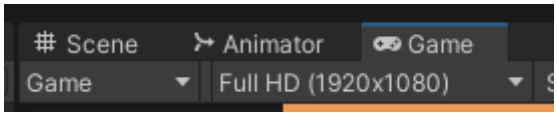

Ilustración 3. Selección de resolución.

3. Cambiar plataforma de exportación.
   En File>Build Settings se encontrará un selector de plataformas que permite nuestro proyecto. Así como el selector de escenas para la exportación del mismo, pudiendo quitar o seleccionar a voluntad las escenas a compilar en el proyecto y reducir su tamaño de exportación.

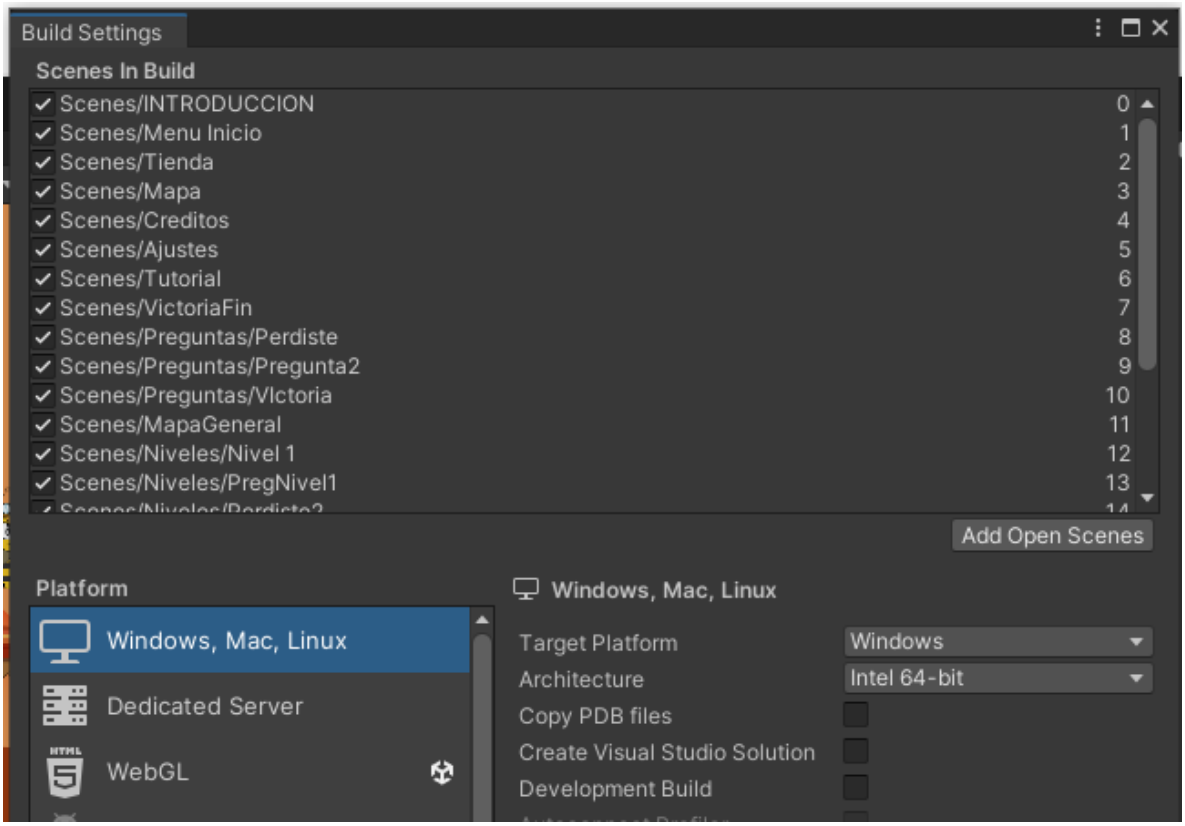

Ilustración 4. Ventana de configuración de Build.

Este paso es importante, ya que Unity permite la exportación del juego a diversas plataformas, para dispositivos móviles, sobremesa y el que nos interesa, WEBGL para aplicaciones web. Este será el que se seleccionará para cambiar la plataforma del juego con el botón Switch Platform, notando como el icono de Unity se posiciona sobre la plataforma seleccionada para la exportación.

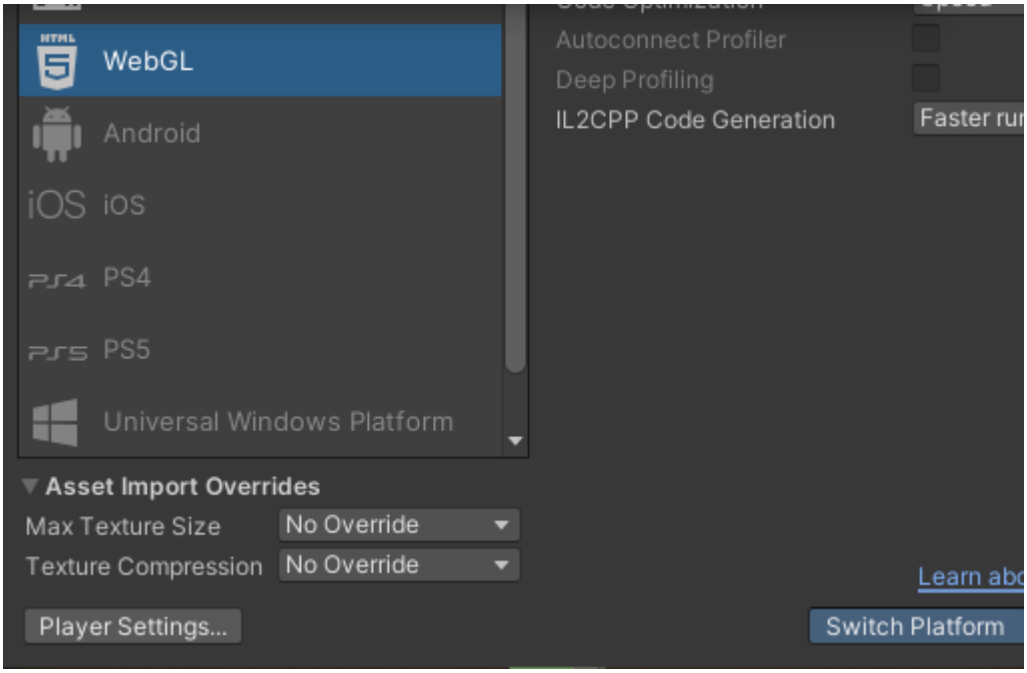

Ilustración 5. Cambio de plataforma.

4. Selección de escenas.

Esto es algo importante, si su proyecto tiene escenas de introducción o animaciones de introducción es recomendable deseleccionarlas para que no afecten el funcionamiento o rendimiento de la aplicación web.

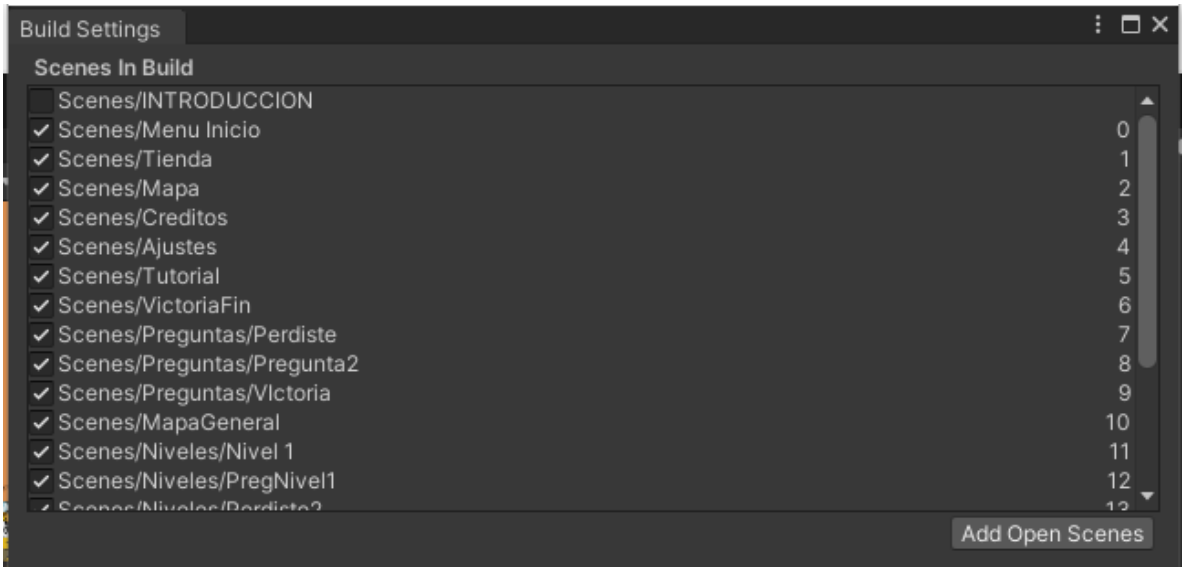

Ilustración 6. Selector de escenas de la construcción de aplicación.

Debe tomarse en cuenta que tanto escenas como videos de mp4 de los recursos del juego pueden hacer que la construcción de la aplicación web resulte en un archivo demasiado pesado para ser cargado en una plataforma en la nube o repositorio de GitHub.

CONSTRUCCION DEL PROYECTO

Una vez configurado el proyecto previamente, se deberá ir a la ventana de configuración con la plataforma WEB GL seleccionada, donde se encontrará 2 botones

1. Build: Creación de la aplicación web en una carpeta especificada.
2. Build and run: Creación y despliegue de la aplicación web generada.

Para ambos casos de ser seleccionados, se pedirá una carpeta donde serán exportado el proyecto, esto es recomendable ponerlo en una carpeta C: del equipo; conociendo esto se dará clic al botón Build especificando la capeta destino de exportación, esto iniciará comenzará la construcción de la aplicación web.

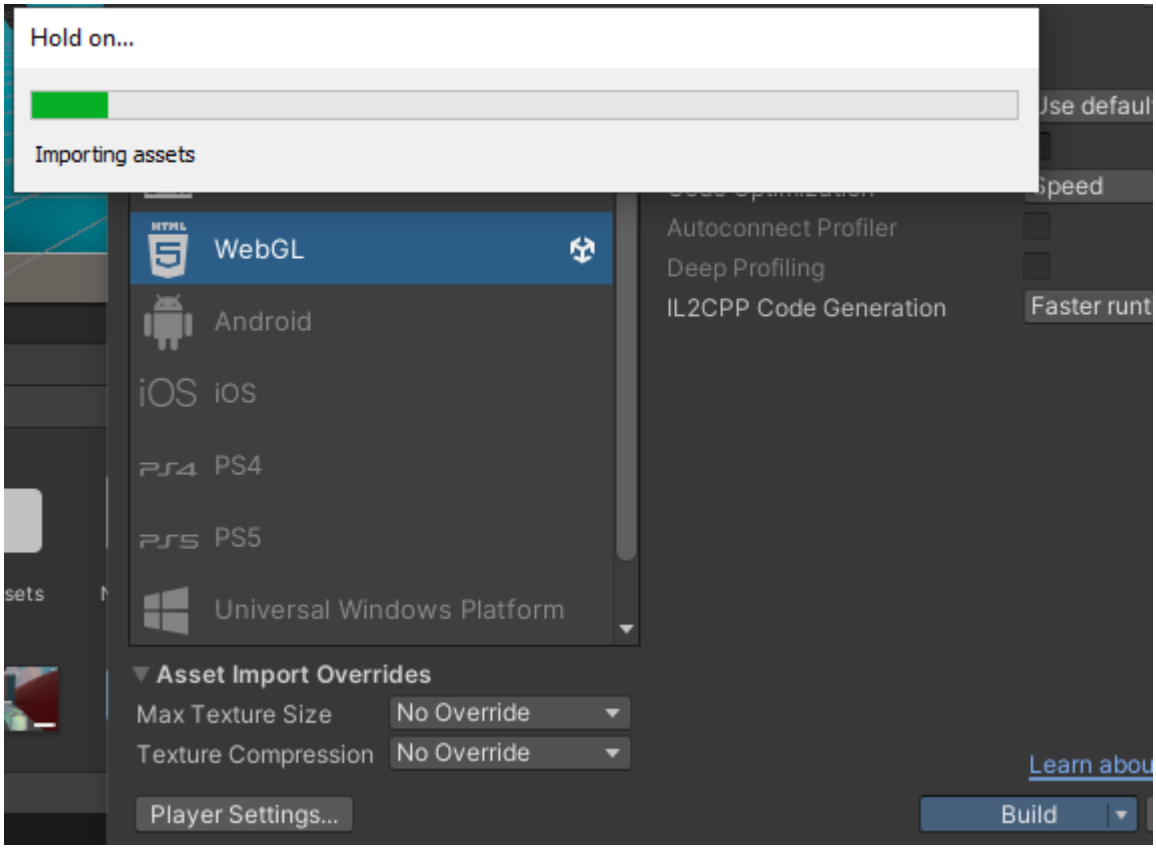

Ilustración 7. Construcción de la aplicación web.

Para verificar que el proceso fue terminado con éxito debe de ser revisado en la consola del proyecto donde las salidas deben indicar que el proceso fue exitoso.

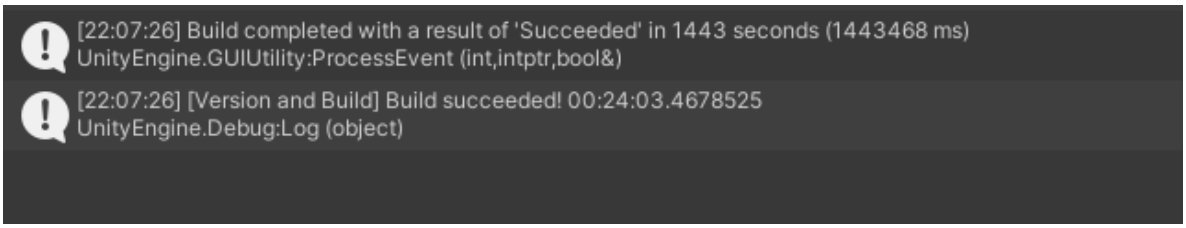

Ilustración 8. Consola con la salida de proceso de construcción.

El proceso fue finalizado con éxito, donde al revisar la carpeta destino especificada se tendrá el siguiente contenido:

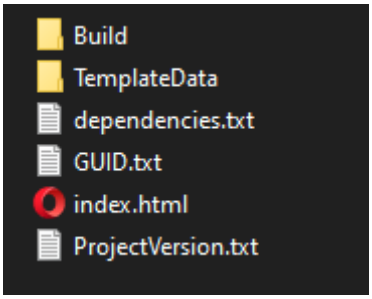

Ilustración 9. Resultado de exportación.

Si bien se tiene varios archivos, los de interés para despliegue en servidores web son los siguientes.

- **Build**: Carpeta donde se encuentra la construcción del proyecto, así como configuraciones de arranque de la aplicación compatible con navegadores web.
    - **Archivo JSON**: configuración del proyecto y los datos necesarios para cargar la aplicación.
    - **Archivo WASM**: Web assembly de formato binario permitirá ejecutar código de alto rendimiento en el navegador.
    - **Archivo DATA**: Contiene los datos del juego, como texturas, modelos, escenas y otros recursos necesarios para la ejecución del juego. Este archivo es el que más peso tiene luego de la construcción, es recomendable que no tenga un peso mayor a 100 Mb para que sea posible subirlo al GitHub de la plataforma.
    - **Archivo Framework JavaScript**: Contiene el código JavaScript que carga y ejecuta el archivo WASM, manejando la interacción del motor de Unity y el navegador.
    - **Archivo Loader JavaScript**: Responsable de cargar los archivos JSON, WASM y DATA para inicializar la aplicación Unity.
- **TemplateData**: Diseño de la página HTML del juego.
    - **Archivo Style CSS**: Contiene los estilos básicos de la página HTML, los cuales pueden ser modificados para una interacción y diseño más dinámicos.
    - **Archivos PNG**: Son imágenes usadas al cargar el juego, funcionan como indicador de que el juego se está cargando y funcionando correctamente.
- **Index.html**: Ventana del juego, es el punto central para el jugador, mostrando el juego en el navegador, usa los archivos de la carpeta Build y el archivo CSS para funcionar correctamente. Posee un elemento canvas WEBGL encargado de renderizar el juego.

Con ello se ha finalizado la construcción de la aplicación web del proyecto de gamificación, ahora solo se requeriría de un servidor web para su despliegue a los usuarios.

DESPLIEGE DE LA APLICACIÓN WEB CON NODEJS

Una vez creado los archivos de la aplicación web, solo quedaría desplegarlos en un servidor local para verificar su funcionamiento, Node JS es una forma muy fácil de despliegue, ya que no usa muchos recursos, y es de fácil implementación.

Primero se deberá usar un Framework de edición de códigos como Visual Studio Code, tener instalado Node JS en su máquina local y otras dependencias.

Se deberá crear un archivo JavaScript en el mismo directorio que los archivos de la aplicación web, este se llamará **server.js.** Este archivo contendrá el siguiente contenido.

```
const express = require('express');
const path = require('path');
const app = express();

const PORT = process.env.PORT || 8080;

// Sirve archivos estáticos desde la carpeta raíz
app.use(express.static(path.join(__dirname)));

app.get('/', (req, res) => {
    res.sendFile(path.join(__dirname, 'index.html'));
});

app.listen(PORT, () => {
```

```
    console.log(`Servidor escuchando en http://localhost:${PORT}`);
});
```

Tabla 1. Archivo server.js.

Este archivo se le ingresa las dependencias de las librerías express y path que se instalan ingresando el comando en la terminal:

- **npm install express**

Ejecutar este comando creara los archivos JSON y una carpeta node_modules que me ayudarán a desplegar la aplicación.

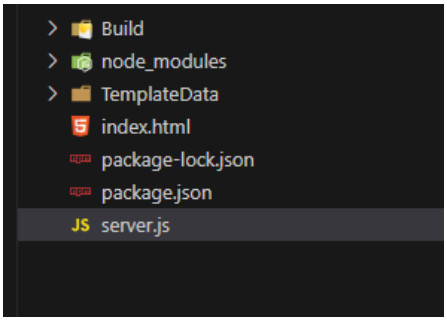

Ilustración 10. Archivos generados.

En base a estas dependencias en el archivo server.js se crea la variable app que servirá para definir el servidor, así mismo me ayuda a usar el index.html en el servidor, este se despliega en el puerto especificado, en este caso será el puerto 8080. Con ello obtendrá la dirección inicial cuando despliegue "/" y un comando que me dirá que el servidor está siendo ejecutado exitosamente, para esto voy a ejecutar el comando.

- **node server.js**

Con ello el mismo me dirá que el servidor se está escuchando desde la dirección URL http://localhost:8080 . Con lo cual, al acceder, debería visualizarse la siguiente ventana del WEBGL con el juego desplegado.

Ilustración 11. Despliegue del juego en navegador con NodeJS de forma local.

Con esto esta correctamente construido la aplicación web de nuestro proyecto de Unity, también se desplego el resultado con Node.JS para una vista dinámica del mismo.

CARGA DE LA APP WEB A LA PLATAFORMA CON ANGULAR

Una vez se haya confirmado el correcto funcionamiento de la aplicación web de forma local, se deberá desplegar el mismo dentro de la plataforma de Edutainment, la cual usa Angular como Framework para administración de páginas, elementos y diseño de la plataforma. Esta plataforma está a cargo del estudiante German Tobar, por lo que su consultoría de acceso al repositorio es indispensable para obtener acceso al repositorio y hacer cambios.

Git clone <URL del git>

Git initialize

Git fetch origin main

Git pull origin main

Una vez obtenido acceso al repositorio se deberá descargar el mismo para ser modificado en base a los requerimientos, sabiendo que se tienen 2 juegos gamificados exportados como aplicaciones web lo ideal sería subirlos como un directorio nuevo en una carpeta assets del repositorio de la plataforma. Lo primero que fue realizado es el componente para visualizar y acceder a los juegos que se han cargado, para esto se debe dirigir en consola a la dirección de auth de la plataforma, ingresando el siguiente código.

cd src/app/components/auth

Con ello lo siguiente fue crear un componente de los juegos con el comando.

Ng g c games

Si bien se creó el componente automáticamente, se debe revisar si está dentro de las rutas en authrouting.module.ts del repositorio.

Ilustración 12. Agregar el componente a las rutas de la plataforma.

Así mismo debe estar en el archivo auth.module.ts como un componente en el @NgModule.

Ilustración 13.Instanciar el componente creado.

El componente games tendrá subyacentes de cabecera y cartas que ayudarán a tener un diseño similar a cartas de selector de juegos, estos se realizan ingresando a la ruta de /games desde la consola y aplicar ng g c <componente> para cada uno. Es importante saber que, al no ser principales, sus archivos spec.ts deben ser eliminados mientras que en sus archivos TypeScript deben eliminarse los atributos standalone, que los vuelve individuales de su componente padre. En auth.module.ts deben ser importados como componentes del proyecto.

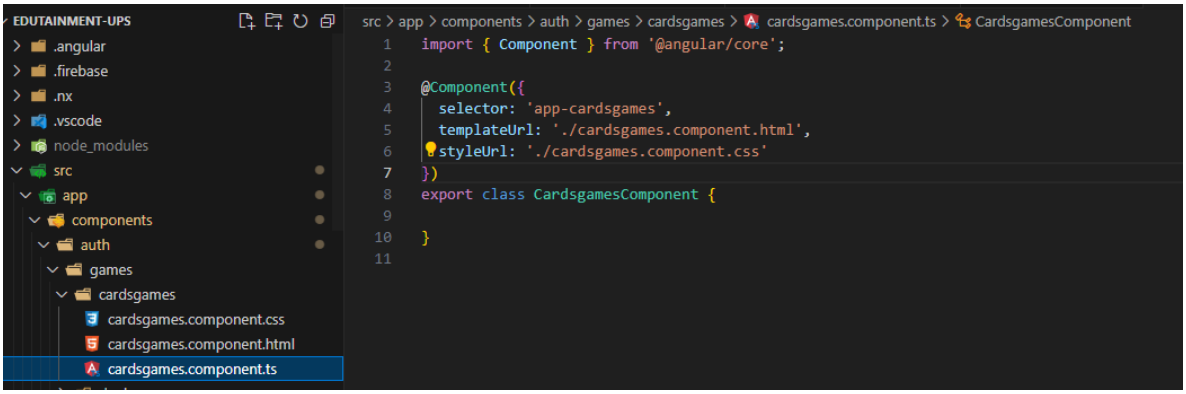

Ilustración 14.Creación y configuración de los subcomponentes.

Con los subcomponentes de games, en el index.html del componente padre solo es instanciar estos mediante elementos a mostrar.

```
Go to component
<!DOCTYPE html>
<html lang="en">

<app-header-index></app-header-index>

<body class="hold-transition sidebar-collapse layout-top-nav">
  <div class="wrapper">
    <app-super-nav></app-super-nav>
    <app-navbar-index></app-navbar-index>
    <app-header-games></app-header-games>
    <app-cardsgames></app-cardsgames>
    <app-footer-index></app-footer-index>
  </div>
</body>

</html>
```

Ilustración 15.Índice de la página de juegos usando sus subcomponentes.

Con esto, el diseño de la cabecera es similar a la página de proyectos de la plataforma, solo cambiando el tema asignado de los juegos.

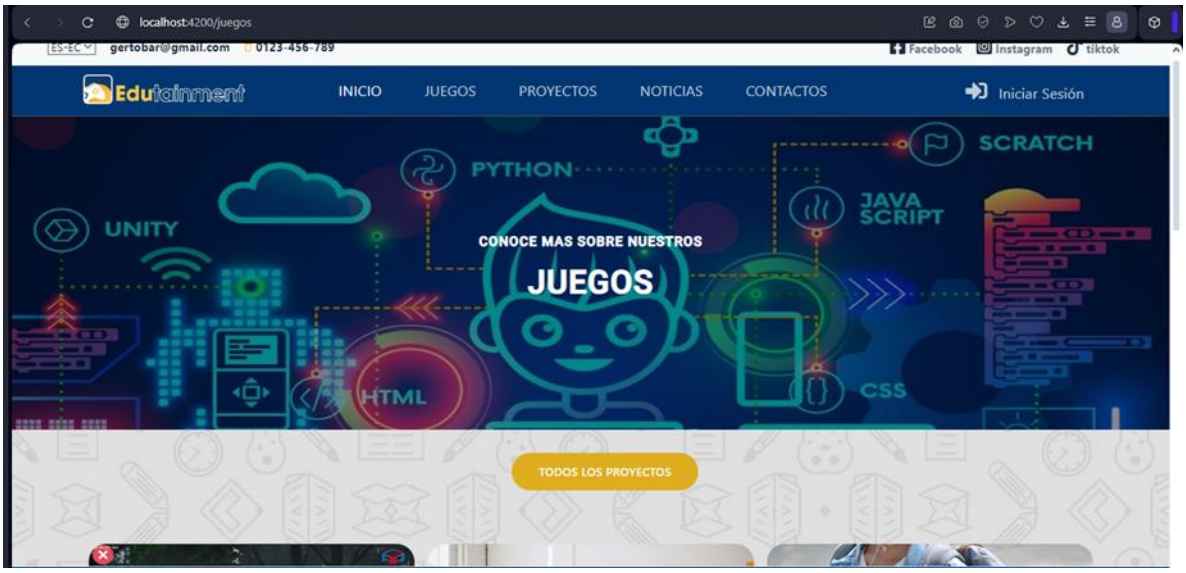

Ilustración 16. Página de los juegos localmente.

Si bien el diseño está correcto, la funcionalidad de las cartas es debe ser configurado para el redireccionamiento a los juegos específicos, en este caso cada carta llevara a un index.html específico para cada juego. Esta sección es la que los desarrolladores de los videojuegos pueden usar para subir sus juegos ya exportados como aplicaciones web, sabiendo que la plataforma ya cuenta con la sección de juegos.

Lo primordial sería subir los juegos a la carpeta de assets, dividido en carpetas para cada juego, dentro de estas estarán los documentos de build, templateData y el index.html respectivamente para cada juego.

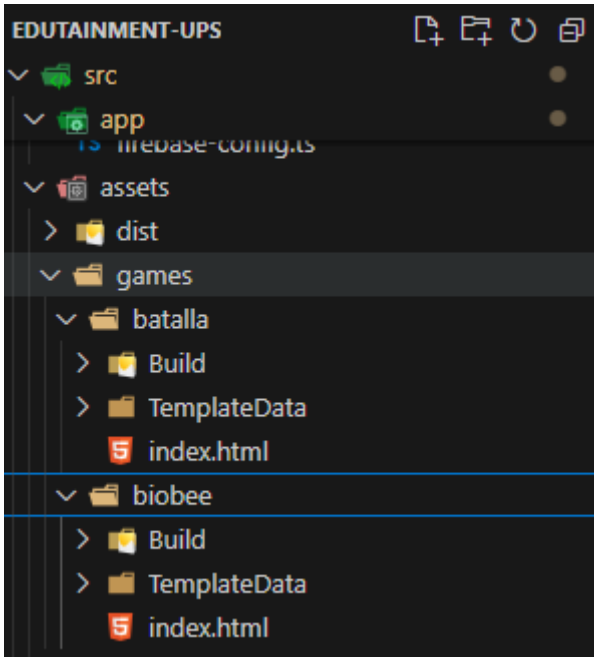

Ilustración 17. Repositorio de los juegos.

Con esto se deben instanciar en cada carta un redireccionamiento a los HTML de cada juego, así mismo en assets se debería tener una miniatura para indicar al jugador que es lo que va a desplegar, también una descripción breve del juego con datos.

```
<section class="meetings-page" id="meetings">
    <div class="container">
      <div class="row">
        <div class="col-lg-12">
          <div class="row">
            <div class="col-lg-12">
              <div class="row grid">
                <div class="col-lg-4 templatemo-item-col all soon">
                  <div class="meeting-item">
                    <div class="thumb">
                      <!-- Cambiarlo por un redireccionamiento a un despliegue de juego-->
                      <a href="assets/games/biobee/index.html" >
                        <img src="../../../../../assets/dist/img/images/BioBeePrev.jpg"
                          alt=""></a>
                    </div>
                    <div class="down-content">
                      <a href="assets/games/biobee/index.html" >
                        <h4>Juego Bio Bee</h4>
                      </a>
                      <p>Videojuego gamificado por<br>estudiantes de diseño multimedia. <br>
```

Ilustración 18. Carta de juego.

Con esto si se selecciona la carta, esta redirecciona al HTML del juego especificado, este puede ser modificados para tener un formato acorde a la plataforma, usando assets ya vistos, ajustar color, botones de regreso y redimensionamiento.

El redimensionado de la ventana de juego por defecto esta activado para que el WEBGL siempre se ajuste al canvas, lo que es un problema al momento de ajustar tamaño de la ventana de Unity Player, para desactivarlo debe ser descomentado la línea de configuración que desactiva el redimensionado al canvas.

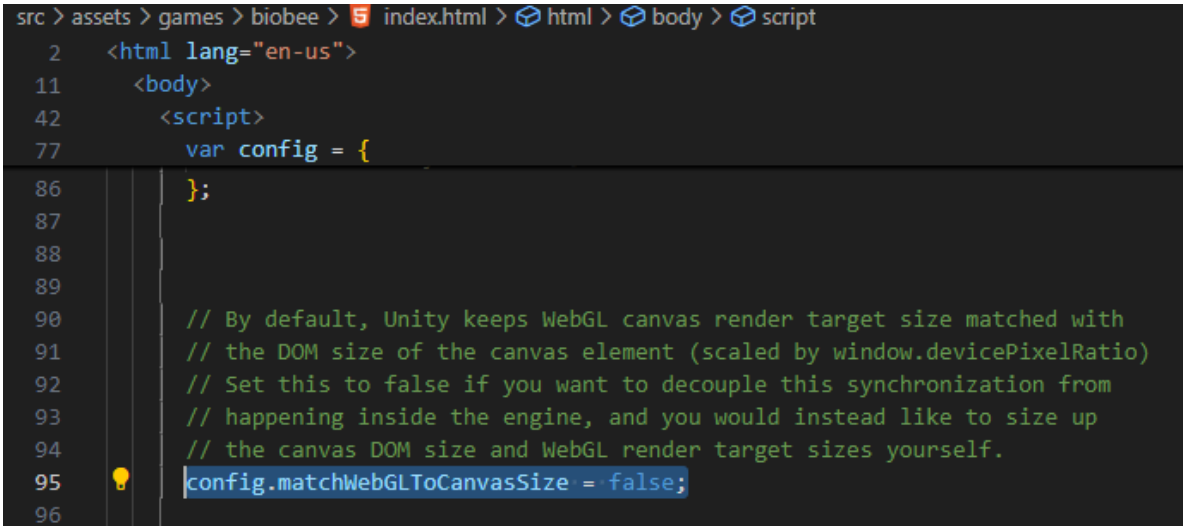

Ilustración 19. Configuración para redimensionar la ventana de juego

Con esta configuración, es posible diseñar el HTML del juego a disposición, en este caso se agregó una descripción y controles del juego, además de un botón para volver a la página de juego.

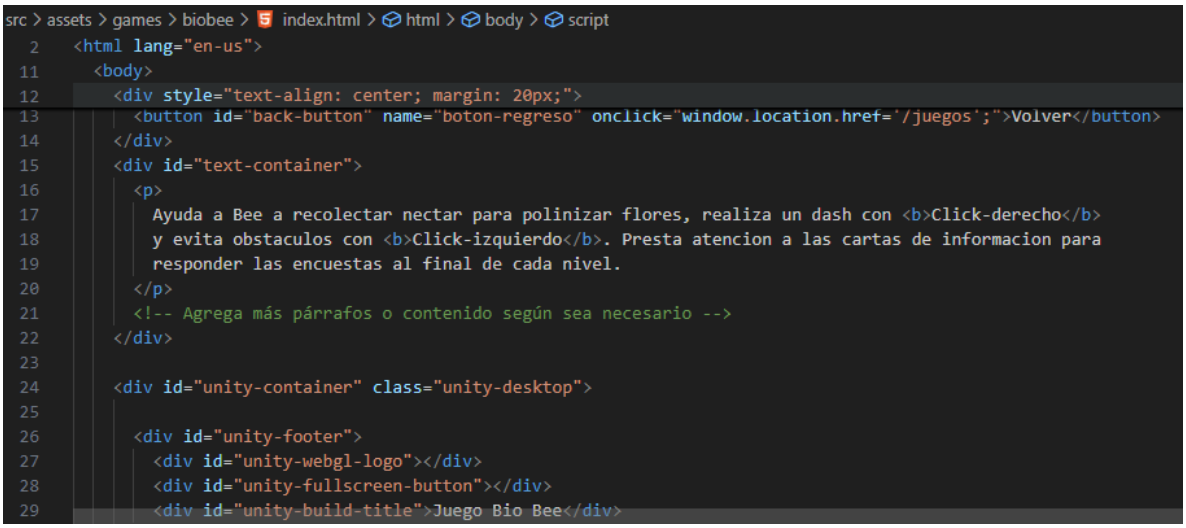

Ilustración 20.Texto y botón de regreso en la ventana de juego

Este proceso es aplicable para el otro juego, con configuración de estilo en /TemplateData/style.css para usar ajustar el diseño de los elementos. También es usado para mover la ventana del contenedor de Unity a una posición más dinámica para el jugador.

```
src > assets > games > biobee > TemplateData > style.css > #unity-container.unity-desktop
1   body { padding: 0; margin: 0; background-image: url(../../../../../assets/dist/img/images/proyectos-bg.jpg); }
2   #unity-container {
3       position: absolute ;
4       width: 50%; /* Ajustar el tamaño del contenedor al 50% del ancho de la pantalla */
5       aspect-ratio: 16 / 9; /* Mantener la relación de aspecto 16:9 */
6       background: grey;
7   }
8   #unity-container.unity-desktop { left: 30%; top: 50%; transform: translate(-50%, -50%) }
9   #unity-container.unity-mobile { position: fixed; width: 100%; height: 100% }
10  #unity-canvas { background: #FFFFFF }
11  .unity-mobile #unity-canvas { width: 100%; height: 100% }
12  #unity-loading-bar { position: absolute; left: 50%; top: 50%; transform: translate(-50%, -50%); display: none }
13  #unity-logo { width: 154px; height: 130px; background: url('unity-logo-dark.png') no-repeat center }
14  #unity-progress-bar-empty { width: 141px; height: 18px; margin-top: 10px; margin-left: 6.5px; background: url('progress
15  #unity-progress-bar-full { width: 0%; height: 18px; margin-top: 10px; background: url('progress-bar-full-dark.png') no-
16  #unity-footer { position: relative }
17  .unity-mobile #unity-footer { display: none }
18  #unity-webgl-logo { float:left; width: 204px; height: 38px; background: url('webgl-logo.png') no-repeat center }
19  #unity-build-title { float: right; margin-right: 10px; line-height: 38px; font-family: arial; font-size: 18px }
20  #unity-fullscreen-button { cursor:pointer; float: right; width: 38px; height: 38px; background: url('fullscreen-button.
```

Ilustración 21.Estilo de los elementos en el HTML

Se para visualizar los cambios correctos se deberá hacer un despliegue de la plataforma de forma local con el comando.

ng serve

Esto construirá la aplicación de Angular para ser visualizada en un puerto desde un navegador. Si accedemos al componente de juegos se visualizarán las cartas a los juegos que hemos cargado como aplicaciones web.

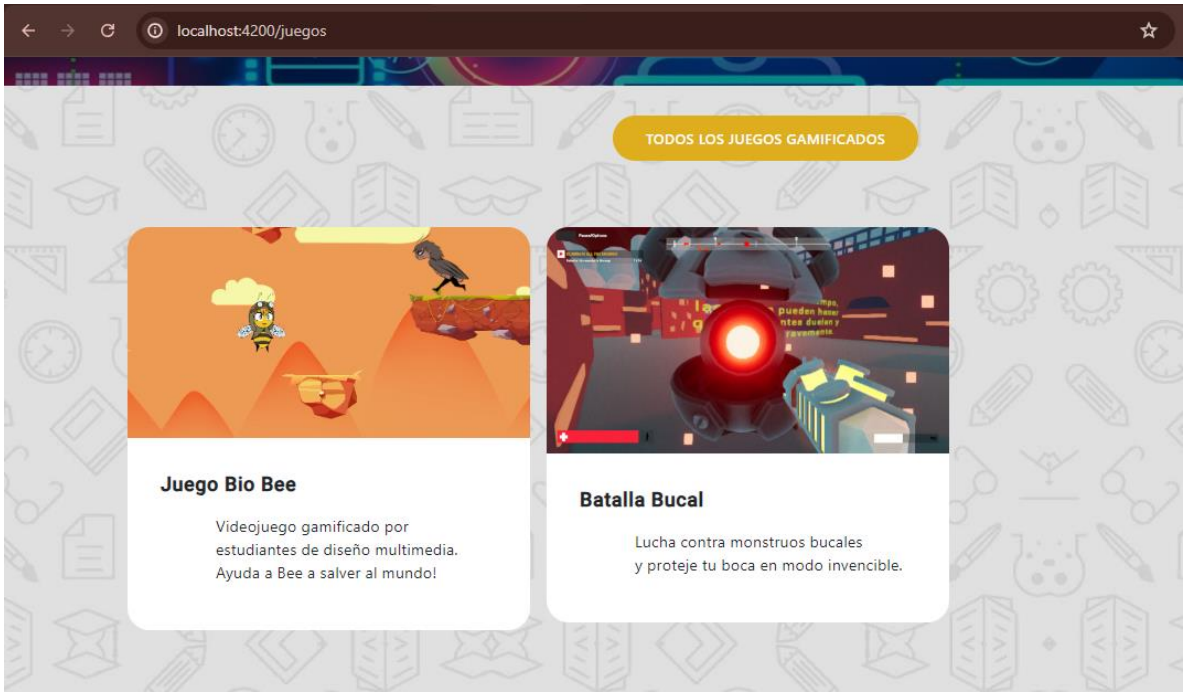

Ilustración 22. Componente Juegos de la plataforma localmente.

Si se selecciona el juego este desplegará el mismo mediante el HTML que fue modificado. En este caso se seleccionó el juego BioBee.

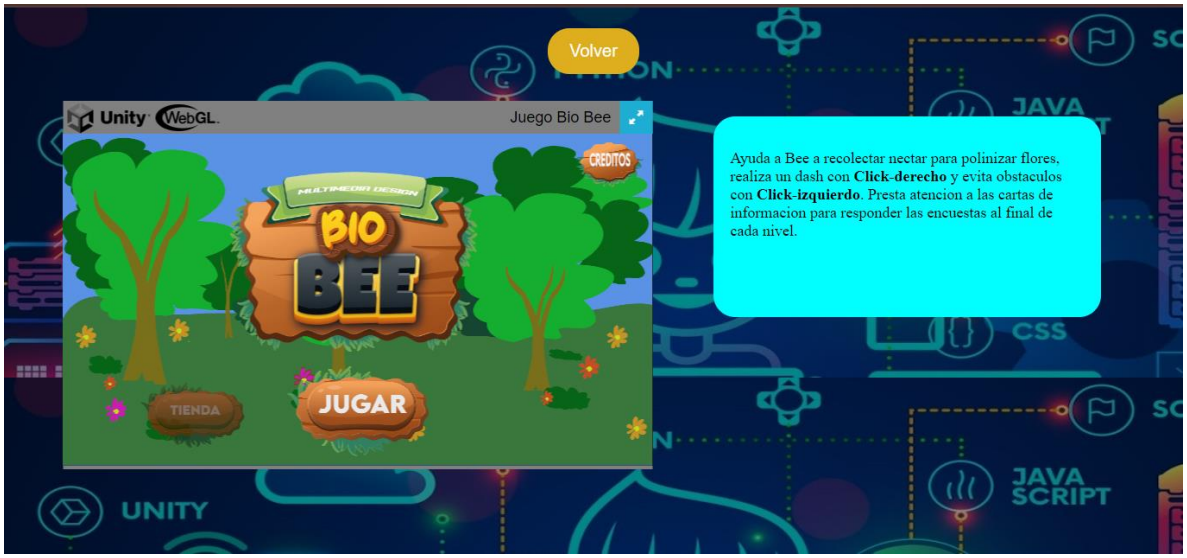

Ilustración 23. Juego BioBee

CARGA DE LOS CAMBIOS A LA RAMA DE LA PLATAFORMA

Con los juegos desplegados localmente en la plataforma, ahora solo es necesario hacer un git Commit y git Push al repositorio de la plataforma para que la misma contenga los cambios realizados. Estos cambios por seguridad se subirán dentro de una rama asignada para no alterar el repositorio principal(main) por lo que es necesario cambiar la rama del repositorio a una rama generada para la carga de los cambios.

Git checkout <rama>

Con esto es seguro realizar un commit de los cambios realizados al repositorio. Por lo que debe seguirse el siguiente grupo de comandos de Git.

Git add .

Git commit –m "<nombrecommit>"

Git push origin <rama>

Con esto los cambios deberán ser cargados a la rama principal por los delegados a administrar la misma, si todo sale bien, los juegos podrán ser jugables desde el siguiente enlace donde está la plataforma de Edutainment. Se accede al siguiente enlace: https://edutainment.firebaseapp.com/juegos

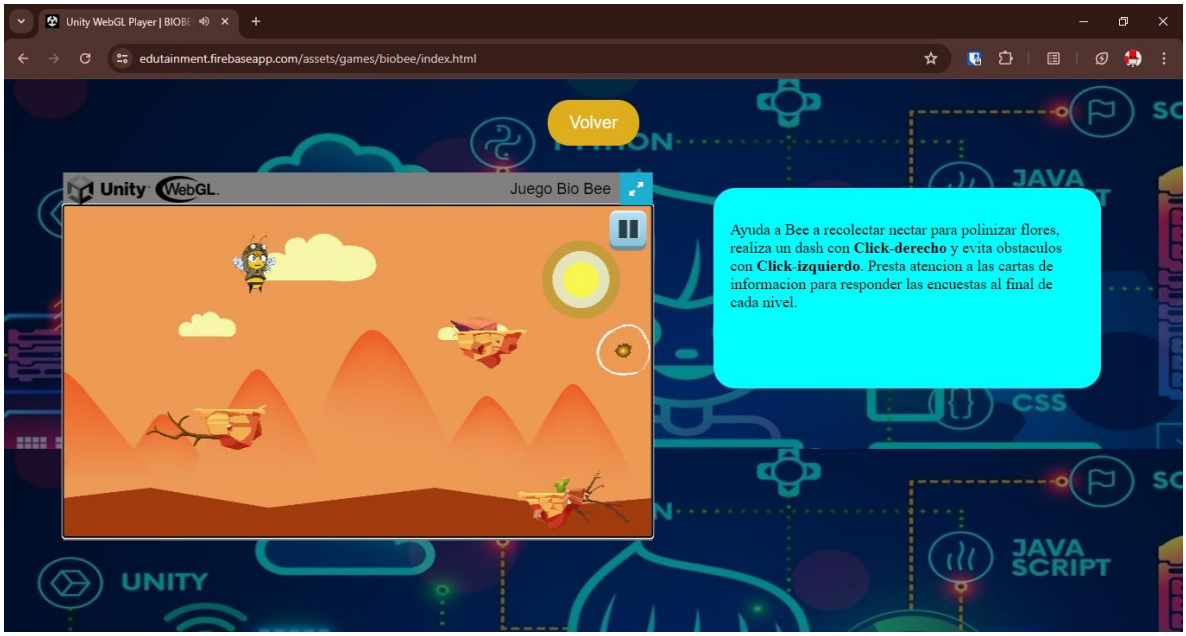

Ilustración 24. Plataforma edutainment modificada.

Con esto los cambios realizados ya dependerían de los desarrolladores.

RECOMENDACIONES

Siempre revisar los assets de sus proyectos, no deberían tener elementos muy pesados o cinemáticas con animaciones que puedan llevar a problemas futuros cuando se exporten como aplicaciones web.

Los archivos de APPWEB deben tener un mínimo de 100MB de peso, esto para que sea posible realizar la carga de los cambios a la plataforma, caso contrario el mismo Git rechazara los cambios por lo pesado de los mismos.

Revisión de elementos de HTML y CSS, para que no surjan inconvenientes sobre datos alterados o errores visuales que afecten la experiencia de los jugadores.