VIDEOJUEGOS Y PROGRAMACIÓN: INTEGRACIÓN DE UNA COMUNIDAD ABIERTA EN UN JUEGO SERIO DE PROGRAMACIÓN

VIDEO GAMES AND PROGRAMMING: INTEGRATING AN OPEN COMMUNITY INTO A SERIOUS PROGRAMMING GAME

TRABAJO FIN DE GRADO
CURSO 2023-2024

AUTORES
ÓSCAR FERNÁNDEZ ROMANO
CÉSAR CARLOS RUBIO PASTOR

DIRECTORES
ANTONIO CALVO MORATA
BALTASAR FERNÁNDEZ MANJÓN

GRADO EN DESARROLLO DE VIDEOJUEGOS / GRADO EN INGENIERÍA INFORMÁTICA
FACULTAD DE INFORMÁTICA
UNIVERSIDAD COMPLUTENSE DE MADRID

VIDEOJUEGOS Y PROGRAMACIÓN: INTEGRACIÓN DE UNA COMUNIDAD ABIERTA EN UN JUEGO SERIO DE PROGRAMACIÓN

VIDEO GAMES AND PROGRAMMING: INTEGRATING AN OPEN COMMUNITY INTO A SERIOUS PROGRAMMING GAME

TRABAJO DE FIN DE GRADO EN DESARROLLO DE VIDEOJUEGOS / INGENIERÍA INFORMÁTICA

AUTORES

ÓSCAR FERNÁNDEZ ROMANO

CÉSAR CARLOS RUBIO PASTOR

DIRECTORES

ANTONIO CALVO MORATA

BALTASAR FERNÁNDEZ MANJÓN

CONVOCATORIA: JUNIO 2024

GRADO EN DESARROLLO DE VIDEOJUEGOS / GRADO EN INGENIERÍA INFORMÁTICA

FACULTAD DE INFORMÁTICA

UNIVERSIDAD COMPLUTENSE DE MADRID

MAYO DE 2024

DEDICATORIA

Óscar Fernández Romano

A mis queridos padres, gracias por su apoyo incondicional y por creer en mí en todo momento. Este logro es tanto suyo como mío. Sin vosotros, no habría llegado hasta donde estoy ahora.

A mi hermana, gracias por compartir toda una vida de crecimiento juntos. Tu presencia ha sido un regalo invalorable en mi vida.

A mi novia, gracias por motivarme siempre y celebrar cada logro a mi lado. Tu apoyo inflexible ha sido mi mayor fortaleza.

César Carlos Rubio Pastor

A mis padres, por su infinita paciencia y regalarme incalculables enseñanzas que me acompañarán siempre.

A mi amigo Manuel, por sacarme de más de un aprieto.

AGRADECIMIENTOS

A Henar Martín, por su amabilidad al reunirse con nosotros y proporcionar una detallada explicación sobre el funcionamiento del servidor y la página web.

A Miguel Ángel Garrido Blázquez, profesor del departamento de Informática y Estadística de la Universidad Rey Juan Carlos, por su colaboración al llevar a cabo pruebas con el juego y elaborar un informe detallado sobre los resultados obtenidos.

A los alumnos que participaron en el programa 4ºESO + Empresa por participar en las pruebas y dar su opinión, contribuyendo así al desarrollo de nuestro proyecto.

A Gema López Luengo y Raquel García Quina, directora y coordinadora de TIC del Colegio Arcipreste de Hita respectivamente, por brindarnos la oportunidad de llevar a cabo las primeras pruebas de Articoding en un entorno escolar. Ha sido una experiencia enriquecedora.

Por último, a Balta y Toni, nuestros tutores, les expresamos nuestra más sincera gratitud por su dedicación y orientación constante durante la realización de este TFG. Su firme respaldo y sus reuniones periódicas, así como el valioso *feedback* proporcionado, han sido fundamentales en nuestro camino hacia la culminación de este proyecto.

RESUMEN

VIDEOJUEGOS Y PROGRAMACIÓN: INTEGRACIÓN DE UNA COMUNIDAD ABIERTA EN UN JUEGO SERIO DE PROGRAMACIÓN.

Articoding es un juego serio educativo desarrollado en 2020, dentro de la Universidad Complutense de Madrid, por alumnos del Grado de Desarrollo de Videojuegos. El juego busca promover el desarrollo del pensamiento computacional (*Computational Thinking*) mediante la enseñanza de conceptos fundamentales de programación. El jugador debe superar niveles resolviendo los problemas que se plantean en un escenario en forma de tablero utilizando la programación visual por bloques.

El objetivo principal de este proyecto es integrar un modelo de comunidad abierta mediante mejoras en el juego, el servidor y la plataforma web de Articoding. Para lograrlo, se ha realizado un análisis exhaustivo de otros juegos similares para determinar las características esenciales que debe tener la Comunidad de Articoding. Posteriormente, se ha diseñado la estructura de los menús, las interfaces y la arquitectura del sistema, seguido por su implementación.

Durante el proyecto también se han llevado a cabo tareas de actualización y mantenimiento, optimizando el código, corrigiendo errores y actualizando los menús. Todo esto ha permitido mejorar la experiencia del usuario. Por último, se han llevado a cabo pruebas con usuarios reales para evaluar la efectividad del nuevo diseño y recopilar retroalimentación sobre su experiencia de uso.

Palabras clave

Juego, Serio, Lógica, Computacional, Programación, Comunidad, Educación, Aprendizaje, Puzles, Multijugador.

ABSTRACT

VIDEO GAMES AND PROGRAMMING: INTEGRATING AN OPEN COMMUNITY INTO A SERIOUS PROGRAMMING GAME.

Articoding is a serious educational game developed in 2020, within the Complutense University of Madrid, by students of the Video Game Development Degree. The game seeks to promote the development of Computational Thinking by teaching fundamental programming concepts. The player must overcome levels by solving the problems posed in a board-shaped scenario using visual programming by blocks.

The main objective of this project is to integrate an open community model through improvements in the game, the server and the Articoding web platform. To achieve this, an exhaustive analysis of other similar games has been carried out to determine the essential characteristics that the Articoding Community should have. Subsequently, the menu structure, interfaces and system architecture were designed, followed by their implementation.

During the project, updating and maintenance tasks have also been carried out, optimizing the code, correcting bugs and updating the menus. All this has improved the user experience. Finally, tests have been carried out with real users to evaluate the effectiveness of the new design and gather feedback on their user experience.

Keywords

Game, Serious, Logic, Computational, Programming, Community, Education, Learning, Puzzles, Multiplayer.

ÍNDICE DE CONTENIDOS

Capítulo 1 - Introducción 1
1.1 Motivación 1
1.2 Objetivos 2
1.3 Plan de trabajo 2
Capítulo 2 - Estado de la cuestión 5
2.1 Enseñanza de la Programación a través de Juegos Serios 5
2.2 Criterios de selección de juegos a analizar 6
2.2.1 Geometry Dash 6
2.2.2 I Wanna Maker 7
2.2.3 MakeCode Arcade 8
2.2.4 Rabbids Coding 9
2.2.5 Scratch 10
2.2.6 Super Mario Maker 2 11
2.3 Conclusiones 12
Capítulo 3 - Articoding: Juego y Comunidad 13
3.1 Introducción a Articoding 13
3.2 Evolución a lo largo del tiempo 14
3.2.1 Desarrollo durante el curso 2020/2021 14
3.2.2 Desarrollo durante el curso 2021/2022 15
3.2.3 Desarrollo durante el curso 2022/2023 15
Capítulo 4 - Desarrollo 17
4.1 Apropiación de código 17
4.1.1 Apropiación de código - Articoding 17
4.1.2 Apropiación de código - Server 20
4.2 Corrección de errores y optimizaciones 23

4.3 Mejoras para la experiencia de usuario....24
4.3.1 Nuevas animaciones....24
4.3.2 Imágenes de los niveles guardados....25
4.3.3 Medallas del perfil....26
4.3.4 Rediseño de la interfaz gráfica del menú principal....27
4.4 Rediseño de la Comunidad....30
4.4.1 Concepto de la Comunidad Abierta....30
4.4.2 Escenarios hipotéticos....32
4.4.3 Diseño de la interfaz gráfica....35
4.4.4 Implementación....38
Capítulo 5 - Pruebas con usuarios....53
5.1 Estructura de las pruebas de usuarios....53
5.1.1 Prueba interna 4ºESO + Empresa....53
5.1.2 Prueba externa 4ºESO + Empresa....55
5.1.3 Prueba Colegio Arcipreste de Hita....57
Capítulo 6 - Conclusiones y trabajo futuro....63
6.1 Conclusiones....63
6.2 Trabajo futuro....64
Chapter 1 - Introduction....67
Chapter 6 - Conclusions and future work....71
Contribuciones Personales....73
Bibliografía....79
Apéndice A - Repositorios....81
Apéndice B - Backup de la base de datos....81
Apéndice C - Script de análisis de los datos json....81
Apéndice D - Reuniones....82

ÍNDICE DE FIGURAS

Figura 1-1. Programas utilizados durante el desarrollo de este proyecto....3
Figura 2-1. Comunidad de Geometry Dash....7
Figura 2-2. Niveles creados en I Wanna Maker....8
Figura 2-3. Listas de niveles en I Wanna Maker....8
Figura 2-4. Pantalla de nivel y certificado al completarlo en MakeCode Arcade....9
Figura 2-5. Pantalla de nivel de Rabbids Coding....10
Figura 2-6. Menú explorar niveles en Scratch....11
Figura 2-7. Perfil del usuario y menú de búsqueda de niveles en Super Mario Maker 2 ..11
Figura 3-1. Logo Articoding....13
Figura 3-2. Pantalla del nivel y menú principal durante el año 2020/2021....14
Figura 3-3. Menús rediseñados durante el año 2021/2022....15
Figura 3-4. Menús de la comunidad durante el año 2022/2023....16
Figura 4-1. Ejemplo de un script refactorizado....18
Figura 4-2. Ejemplo de una escena refactorizada....19
Figura 4-3. Pestaña Utils creada en Unity....19
Figura 4-4. Diagrama de paquetes y dependencias....20
Figura 4-5. Ejemplo de un método que devuelve niveles públicos....21
Figura 4-6. Diagrama entidad-relación del servidor....21
Figura 4-7. Código previo a la apropiación....22
Figura 4-8. Código posterior a la apropiación....23
Figura 4-9. Creación de animaciones del pingüino en Blender....24
Figura 4-10. Ventana de resultado del nivel con el pingüino animado....25
Figura 4-11. Vista previa de niveles creados....25

Figura 4-12. Página del perfil del usuario con las medallas 26
Figura 4-13. Bocetos a lápiz del rediseño del menú principal 27
Figura 4-14. Bocetos a lápiz del rediseño del menú de los niveles 28
Figura 4-15. Prototipo del rediseño del menú principal del juego en Figma 29
Figura 4-16. Comparativa del menú principal previo y posterior al rediseño 29
Figura 4-17. Comparativa del menú de niveles previo y posterior al rediseño 30
Figura 4-18. Comparativa de los elementos del nuevo diseño de la comunidad 32
Figura 4-19. Bocetos del nuevo diseño de la comunidad de Óscar 35
Figura 4-20. Bocetos del nuevo diseño de la comunidad de César 36
Figura 4-21. Prototipo del nuevo diseño de la comunidad en Figma 37
Figura 4-22. Resultados del nuevo diseño de la comunidad en el juego 37
Figura 4-23. Menús de inicio de sesión y registro de usuarios 38
Figura 4-24. Menú de comunidad 39
Figura 4-25. Menú selección imagen de usuario 40
Figura 4-26. Carpeta levelImages en el servidor 41
Figura 4-27. Diseño tarjetas de niveles y la página de buscar niveles 41
Figura 4-28. Menú de búsqueda avanzada 42
Figura 4-29. Función getPlaylists en la clase PlaylistService 43
Figura 4-30. Menú para subir niveles 44
Figura 4-31. Diseño tarjetas de listas y la página de buscar listas 45
Figura 4-32. Menú de creación de listas 46
Figura 4-33. Menú de clases 47
Figura 4-34. Tarjetas con los niveles de una clase 47
Figura 4-35. Captura de estado de la tabla de unión triple 48
Figura 4-36. Definición de la clase ClassRoomLevelCompleted 49

Figura 4-37. Tabla con el progreso de los alumnos en una clase 49
Figura 4-38. Opción de acceder a una clase 50
Figura 4-39. Mensajes informativos de la comunidad.....50
Figura 5-1. Fotos de la prueba de 4ºESO + Empresa 54
Figura 5-2. Gráfico con los resultados de las pruebas de 4ºESO + Empresa 55
Figura 5-3. Niveles de las clases utilizadas para la prueba 58
Figura 5-4. Información de la instancia E2 y despliegue del túnel ssh.....58
Figura 5-5. Tabla resultados prueba de la comunidad visible en el cliente.....60
Figura 5-6. Gráfico con los resultados de las pruebas del colegio Arcipreste de Hita 61
Figura 5-7. Fotos de la prueba del colegio Arcipreste de Hita 62
Figura 6-1. Calendario con las reuniones realizadas a lo largo del proyecto 82

Capítulo 1 - Introducción

"La educación es el pasaporte hacia el futuro, el mañana pertenece a aquellos que se preparan para el hoy." - Malcolm X

1.1 Motivación

La digitalización ha tenido un profundo impacto en la sociedad. Ha alterado fundamentalmente cómo vivimos, trabajamos, nos comunicamos y nos relacionamos con el mundo que nos rodea [1]. Debido a ello, surge la necesidad de enseñar *pensamiento computacional* [2] para equipar a las personas con las habilidades necesarias para entender y participar activamente en un mundo cada vez más digital.

Sin embargo, los sistemas educativos actuales siguen centrados en métodos tradicionales de enseñanza que priorizan la memorización y la repetición sobre el desarrollo de habilidades críticas y prácticas. En este contexto, los juegos serios se presentan como una herramienta más efectiva para enseñar pensamiento computacional, ya que combinan el aprendizaje práctico y la resolución de problemas con una experiencia interactiva y atractiva, fomentando así la creatividad y el compromiso de los estudiantes. Articoding nació con esta misma idea de proporcionar juegos educativos para enseñar programación.

Articoding es un juego serio diseñado para enseñar a los jóvenes los conceptos básicos de la programación y mejorar las habilidades del pensamiento computacional. Fue desarrollado en Unity en 2020 y se ha seguido actualizando con nuevas funcionalidades desde entonces-. Además, mantiene su código abierto, lo que permite a cualquier persona modificarlo y añadir nuevas características libremente.

Este proyecto proporciona la oportunidad de contribuir al desarrollo continuo de Articoding, además de una oportunidad para profundizar en la teoría y práctica de los juegos serios y su aplicación en el aula, explorando nuevas metodologías educativas y evaluando su impacto en el aprendizaje.

1.2 Objetivos

El propósito principal de este trabajo de fin de grado consiste en la ampliación del proyecto Articoding, mediante la mejora de su interfaz y la ampliación de su comunidad abierta, permitiendo a estudiantes y profesores compartir niveles creados por ellos mismos y mejorando la experiencia de usuario. Para alcanzar este objetivo, se han establecido los siguientes subobjetivos más específicos:

- Estudiar y entender aplicaciones y juegos centrados en el desarrollo del pensamiento computacional y su comunidad.
- Realizar la apropiación de código.
- Identificar y corregir los errores no abordados previamente.
- Mejorar la experiencia de usuario.
- Rediseñar el sistema de comunidad y aplicar los cambios necesarios para añadir mejoras.
- Validar los cambios con pruebas de usuario reales.

1.3 Plan de trabajo

Antes de iniciar el desarrollo, con el objetivo de entender y estudiar aplicaciones similares, se llevará a cabo un análisis exhaustivo de diversos juegos para identificar interfaces efectivas y mecánicas atractivas.

En primer lugar, se analizará el estado actual del proyecto y se estandarizará de cara a futuras mejoras e implementaciones, para lo que se realizará una apropiación de código.

Tras esto, para identificar y corregir los errores, se revisarán los informes generados a partir de las pruebas de usuario llevadas a cabo el año anterior.

Con el fin de mejorar la experiencia del usuario, se realizarán modificaciones en la interfaz, se incluirá un mayor sistema de feedback dentro del juego y se implementarán animaciones adicionales.

Para rediseñar el sistema de comunidad, primero se presentará la nueva idea, seguida del diseño y prototipado para verificar su solidez [3]. Luego, se implementará en

el juego, integrando los cambios necesarios en el servidor y la base de datos. Finalmente, se adaptará el servicio web del juego para cumplir con estas nuevas necesidades. Adicionalmente, se intentarán realizar pruebas con usuarios reales para validar los cambios, detectar nuevos errores y recopilar feedback.

En relación con las herramientas que se utilizarán, Google Drive y Gmail se emplearán para la gestión de documentos y la comunicación, respectivamente. *GitHub* será empleado para el control de versiones, creando una copia de los repositorios actuales. *Unity* se mantendrá como el motor de videojuego, junto con *Visual Studio* como editor de código. Para la creación de *Assets*, se utilizará *Photoshop* para la edición de imágenes y *Blender* para la elaboración de animaciones. En cuanto al desarrollo del servidor, se empleará *Intellij* como IDE principal, *HeidiSQL* para la manipulación de la base de datos, y *Postman* para probar las peticiones HTTP. Para el desarrollo del cliente, se utilizará *Visual Studio Code*. Los diseños de las interfaces se crearán inicialmente en papel y lápiz, y posteriormente se elaborarán prototipos con *Figma*.

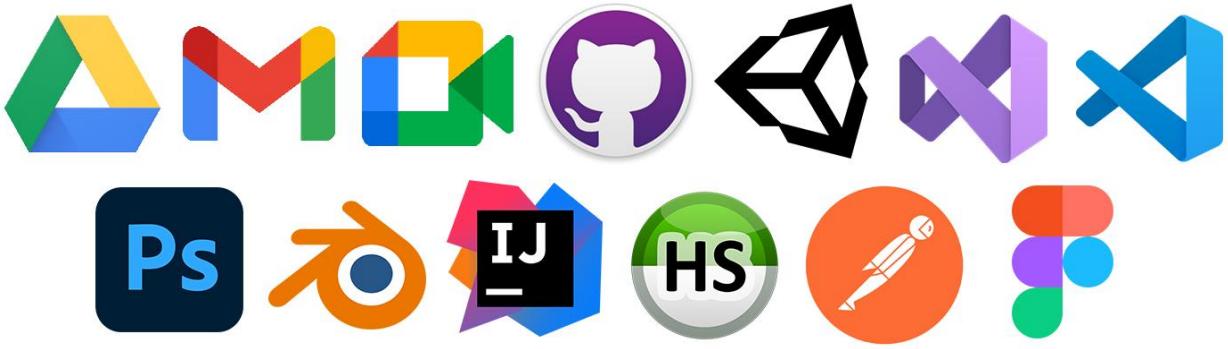

Figura 1-1. Programas utilizados durante el desarrollo de este proyecto

Con el fin de monitorizar el progreso y recibir *feedback* sobre los avances realizados, se llevarán a cabo reuniones periódicas presenciales o a través de Google Meet. Durante estas reuniones, se presentarán los avances desde el encuentro anterior y se definirán los próximos pasos a seguir. Se proporciona una descripción más detallada de estas reuniones en el Apéndice D.

Capítulo 2 - Estado de la cuestión

"La creatividad es la capacidad de ver lo que los demás ven y pensar lo que nadie más ha pensado." - Albert Einstein

2.1 Enseñanza de la Programación a través de Juegos Serios

La enseñanza tradicional es un enfoque educativo donde el maestro es la figura central y los estudiantes son receptores pasivos del conocimiento. Este modelo de enseñanza ha arraigado su práctica en la transmisión oral de conocimientos y en la repetición mnemotécnica como principales pilares de aprendizaje. Sin embargo, en un mundo caracterizado por avances sociales y tecnológicos sin precedentes, resulta evidente que estos métodos tradicionales se han vuelto insuficientes para satisfacer las demandas contemporáneas.

Como señala María Isabel Torres Salas en su estudio [4], *"Las tecnologías educativas deben adaptarse a las nuevas tecnologías, con el fin de facilitar el acceso de los ciudadanos a la educación, en el marco del desarrollo tecnológico de la informática y de las telecomunicaciones".* Es en este contexto que surgen los juegos serios, con el objetivo de modernizar y enriquecer el proceso educativo.

Los *juegos serios* son videojuegos diseñados con un propósito principal más allá del entretenimiento, como la educación, el entrenamiento, la simulación, o la concienciación sobre ciertos temas. Estos juegos pueden integrarse en el proceso de enseñanza-aprendizaje sin comprometer la diversión. El término *serio* se refiere específicamente a aquellos videojuegos utilizados en sectores como la educación, la ciencia, la atención médica, la planificación urbana, la ingeniería o la política.

Como demuestra *Reham Ayman* en su análisis [5], el enfoque de incorporar conceptos básicos y fundamentales dentro de un contexto práctico, como lo hace un videojuego, es una estrategia altamente efectiva en el proceso de aprendizaje. En su estudio se compararon dos métodos de enseñanza: uno basado en un juego interactivo y otro en el método educativo tradicional. Con este experimento se llegó a la siguiente

conclusión: *El resultado demostró que el juego tiene más efecto en el aprendizaje y el nivel de compromiso de los niños que el método educativo tradicional.*

Articoding es un juego serio con una estética juvenil diseñada para atraer a jóvenes. Este proyecto se encuentra específicamente orientado hacia el ámbito educativo, buscando introducir a los jóvenes los principios esenciales de la programación. Además, su código fuente está disponible de manera abierta, permitiendo que cualquier individuo pueda acceder a él, descargarlo y realizar sus propias adaptaciones y mejoras según sus necesidades y preferencias.

2.2 Criterios de selección de juegos a analizar

Para profundizar en el conocimiento sobre otros juegos serios dentro del ámbito de la pedagogía informática e identificar características destacadas, se realizó una cuidadosa selección de juegos. Los criterios empleados para esta selección abarcaron juegos serios centrados en el desarrollo del pensamiento computacional, así como aquellos que ofrecen una plataforma comunitaria donde los usuarios pueden crear y compartir niveles. Además, se examinaron plataformas educativas que enseñan programación a través de sistemas de bloques.

2.2.1 Geometry Dash

Geometry Dash [6] es un juego de plataformas desafiante desarrollado por RobTop Games, donde los jugadores controlan un icono a través de niveles llenos de obstáculos. Aunque principalmente es un juego para un jugador, presenta elementos sociales como tablas de clasificación en línea y la capacidad de crear y compartir niveles con la comunidad. Estos aspectos brindan a los jugadores la oportunidad de competir y contribuir en un entorno en línea.

Figura 2-1. Comunidad de Geometry Dash

El juego en cuestión proporciona una serie de conceptos relevantes para el diseño de la comunidad. Por un lado, destaca la presentación de información en cada nivel, la cual incluye el nombre del nivel, el nombre del creador, el recuento de reproducciones y el número de interacciones positivas, como los "me gusta". Por otro lado, se observa la presencia de un sistema de búsqueda de niveles que permite a los usuarios encontrar niveles específicos mediante la introducción de términos de búsqueda como el nombre del nivel, el nombre del autor o la identificación única del nivel (ID). Estas características contribuyen significativamente a la experiencia del usuario al facilitar el descubrimiento y la interacción con el contenido generado por la comunidad dentro del juego.

2.2.2 I Wanna Maker

I Wanna Maker [7] es un juego de plataformas inspirado en *I Wanna Be The Guy* [8]. Su enfoque principal radica en proporcionar a la comunidad herramientas sólidas para la creación y compartición de niveles en línea.

Los niveles en el juego exhiben una serie de métricas, incluyendo la tasa de completitud, el número de *likes* recibidos y etiquetas que definen aspectos como la dificultad, temática o funcionalidad del nivel. Además, el juego facilita búsquedas simples, permitiendo a los usuarios explorar niveles populares o recién creados, así como búsquedas avanzadas que posibilitan el filtrado por distintos parámetros.

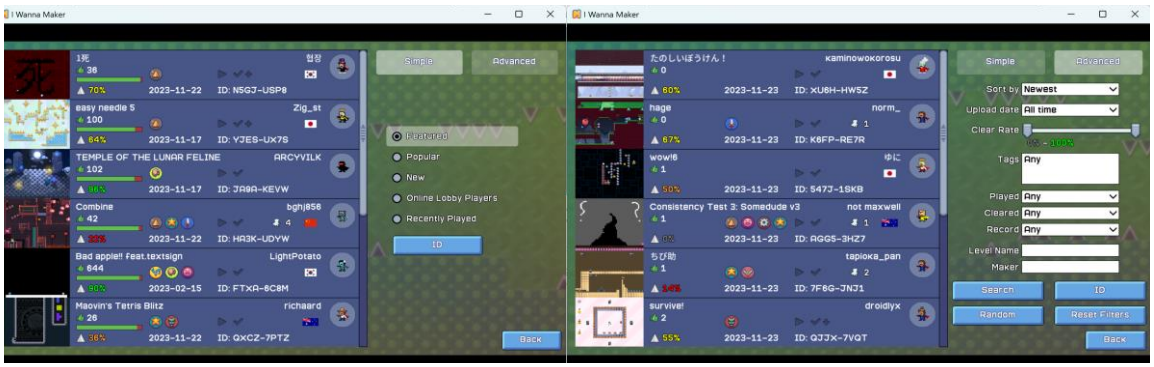

Figura 2-2. Niveles creados en I Wanna Maker

Otra característica destacable es la funcionalidad de crear listas de niveles, lo que brinda a los jugadores la oportunidad de organizar y compartir conjuntos de niveles con la comunidad.

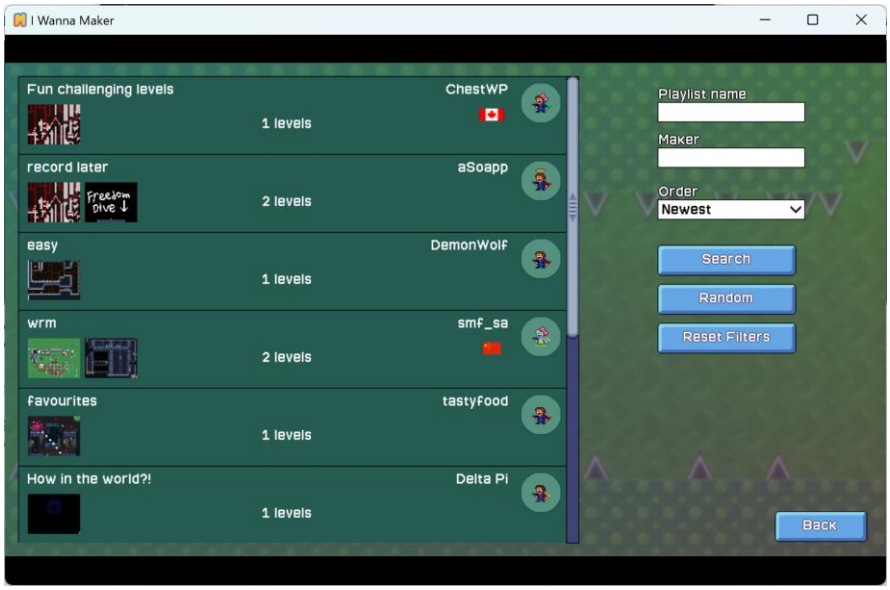

Figura 2-3. Listas de niveles en I Wanna Maker

Todas estas funcionalidades y herramientas resultaron sumamente interesantes para ser implementadas en el proyecto, con el objetivo de orientar a la comunidad hacia un modelo similar al exitoso enfoque de juegos como *I Wanna Maker*.

2.2.3 MakeCode Arcade

MakeCode Arcade [9] es una plataforma web desarrollada por Microsoft con el propósito de facilitar la creación de juegos y el aprendizaje de programación. Esta plataforma ofrece la posibilidad de construir juegos utilizando un sistema de

programación basado en bloques, similar al de Scratch, aunque también brinda la opción de utilizar lenguajes de programación más avanzados como JavaScript y Python.

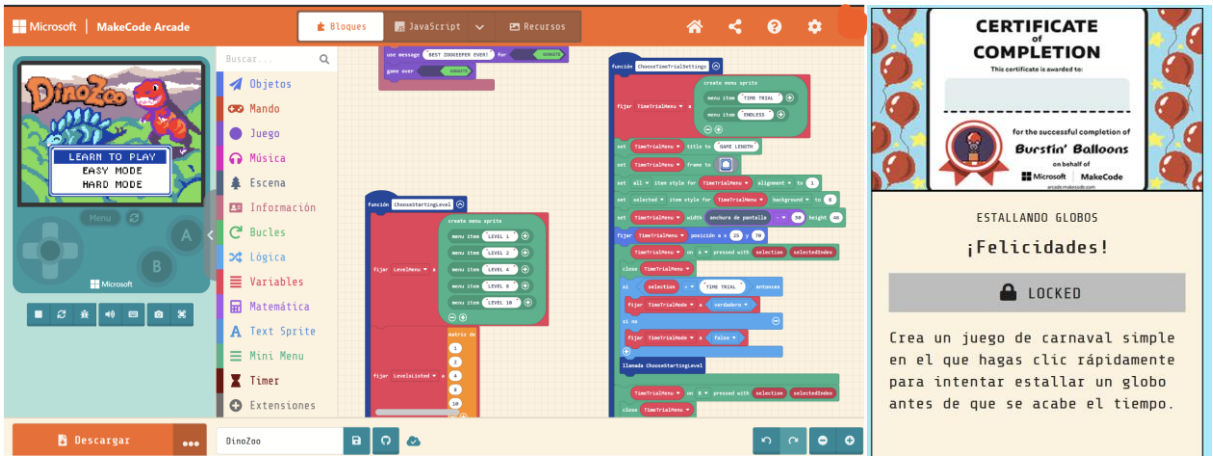

Figura 2-4. Pantalla de nivel y certificado al completarlo en MakeCode Arcade

La estructura de *MakeCode Arcade* presenta similitudes significativas con la de Scratch. El menú principal se divide en sublistas que incluyen niveles, tutoriales y creadores destacados. Además, la plataforma incorpora *skillmaps*, que son listas de desafíos diseñadas para recompensar a los usuarios con certificados al completarlos satisfactoriamente.

2.2.4 Rabbids Coding

Rabbids Coding [10] es un videojuego educativo desarrollado por Ubisoft basado en la popular franquicia de los Rabbids. Este juego está diseñado para enseñar conceptos básicos de programación a niños y principiantes de todas las edades de una manera divertida y accesible. Al igual que en Articoding, el jugador debe colocar una serie de bloques con instrucciones en el orden correspondiente para superar el nivel. Dando al botón Play se ejecutarán las instrucciones para comprobar el nivel.

Figura 2-5. Pantalla de nivel de Rabbids Coding

En el análisis de este juego, no se han identificado elementos innovadores relevantes para la investigación en curso. Sin embargo, su notable similitud con Articoding proporciona una validación adicional sobre la idoneidad del diseño de la interfaz durante la experiencia de juego y la efectividad del ritmo de presentación de nuevos conceptos, así como su interiorización a través de la resolución progresiva de niveles. Este hallazgo refuerza la percepción de que el enfoque adoptado en cuanto a la estructura y dinámica del juego es apropiado para los objetivos planteados en el estudio.

2.2.5 Scratch

Scratch [11] es un motor de videojuegos desarrollado por el MIT Media Lab. Permite a los usuarios crear proyectos directamente en la web utilizando un lenguaje de programación de alto nivel basado en bloques, similar al utilizado en Articoding. La facilidad de comprensión de este lenguaje lo convierte en una herramienta ideal para fomentar el pensamiento computacional, razón por la cual es ampliamente utilizado en la educación de niños entre 8 y 16 años.

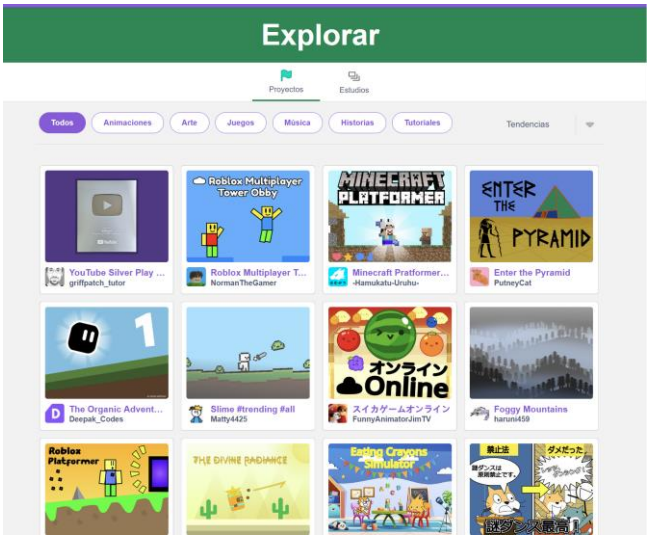

Figura 2-6. Menú explorar niveles en Scratch

Al acceder a la plataforma de Scratch, los usuarios se encuentran con listados de niveles, proyectos y estudios destacados. Al explorar los proyectos o estudios, la página ofrece la posibilidad de ordenarlos según diversos criterios, como *recientes*, *tendencias* y popularidad, además de permitir filtrar según el tipo de proyecto deseado. Estas funcionalidades contribuyen a mantener activa la comunidad de usuarios, y son características recurrentes en gran parte de los juegos analizados.

2.2.6 Super Mario Maker 2

Super Mario Maker 2 [12] es un videojuego de plataformas y creación de niveles desarrollado por Nintendo para Nintendo Switch. Se centra principalmente en su editor de niveles, permitiendo a los jugadores crear y compartir nuevos niveles para mantener activa la comunidad.

Figura 2-7. Perfil del usuario y menú de búsqueda de niveles en Super Mario Maker 2

Dos aspectos destacables del juego considerados para implementar en el proyecto son la obtención de medallas por alcanzar hitos y la información detallada de los niveles. En cuanto a la información de los niveles, esta se resume en un contador de *likes* y el número de intentos realizados por otros jugadores para completar el nivel. Además, se proporciona la opción de añadir etiquetas a los niveles, lo que contribuye a organizar y categorizar los niveles creados por la comunidad.

Estos aspectos son relevantes para la propuesta de rediseño de la comunidad, ya que promueven la interacción entre los jugadores, elemento clave para mantener una comunidad activa y comprometida.

2.3 Conclusiones

Como conclusiones, se destacan mecánicas de ciertos juegos que se integran de manera óptima en Articoding, tales como la implementación de medallas para reconocer los logros alcanzados o el diseño de menús principales que ofrecen una vista previa de cada nivel.

Asimismo, tras el análisis de estos juegos, se han identificado una serie de características comunes en sus comunidades. Entre estas características se incluyen un sistema de valoración de niveles junto con un contador que registra el número de veces que un nivel ha sido jugado, un sistema de filtrado avanzado que facilita la búsqueda de niveles específicos, un sistema de etiquetado de niveles para su categorización y la mención del autor de cada nivel.

Capítulo 3 - Articoding: Juego y Comunidad

"¡Me encanta el olor a aventuras por la mañana!"

- Skipper, Los pingüinos de Madagascar

3.1 Introducción a Articoding

Articoding [13] se presenta como un juego educativo cuyo objetivo principal es fomentar el *pensamiento computacional* y facilitar el aprendizaje de conceptos fundamentales de programación. El jugador se enfrenta a niveles donde debe resolver desafíos planteados en un entorno de tablero virtual.

Figura 3-1. Logo Articoding

La meta del jugador consiste en dirigir un haz de luz láser hacia un objetivo utilizando la programación visual por bloques, similar a la interfaz de *Scratch*. Al alcanzar el objetivo, el jugador avanza al siguiente nivel.

Este juego está diseñado para estudiantes de entre 12 y 16 años que tienen un conocimiento limitado en programación y que pueden beneficiarse significativamente del desarrollo de su Pensamiento Computacional, no sólo en el ámbito de la informática, sino también en otras disciplinas *STEM* (Science, Technology, Engineering and Mathematics).

Aunque Articoding está orientado principalmente al ámbito educativo, también puede ser utilizado de forma independiente. En ambos casos, el usuario juega de manera individual, enfrentándose a diversos desafíos que enseñan diferentes conceptos de programación.

Su aplicación en entornos educativos ofrece una alternativa interesante para enseñar habilidades básicas de programación en comparación con métodos más convencionales o aplicaciones más formales. La componente lúdica del juego puede aumentar la motivación del jugador, mientras que la narrativa sutil que lo acompaña puede mejorar la inmersión en la experiencia de juego.

3.2 Evolución a lo largo del tiempo

Articoding fue creado como parte de un TFG de la UCM en el curso 2020-2021, y ha sido ampliado en los sucesivos cursos. A continuación, se describen los cambios realizados en cada una de estas ampliaciones.

3.2.1 Desarrollo durante el curso 2020/2021

Durante el primer año de desarrollo de Articoding en el curso 2020/2021 como parte del TFG "*Juegos Serios para Promover el Pensamiento Computacional y la Programación*" por Dany Faouaz Santillana, Arturo García Cárdena y Álvaro Poyatos Morate [14], se establecieron las bases del juego.

Se desarrolló la jugabilidad principal, permitiendo a los usuarios completar niveles y desbloquear nuevos progresivamente. Se implementó un sistema de puntuación basado en estrellas para evaluar el desempeño del jugador y un sistema de pistas para ayudar en la resolución de problemas. También se introdujo un sistema de ventanas emergentes para explicar nuevos conceptos, un sistema de temario para proporcionar una referencia de conceptos y un sistema de guardado de progreso para que los usuarios pudieran retomar el juego en otro momento. Además, se integró *uBlockly,* una librería que da soporte a la programación mediante bloques, y se estableció un sistema de analíticas con *Simva* para recopilar datos sobre las acciones del usuario. Por último, se agregó un sistema de creación de niveles en el que los usuarios podían guardar sus propios diseños.

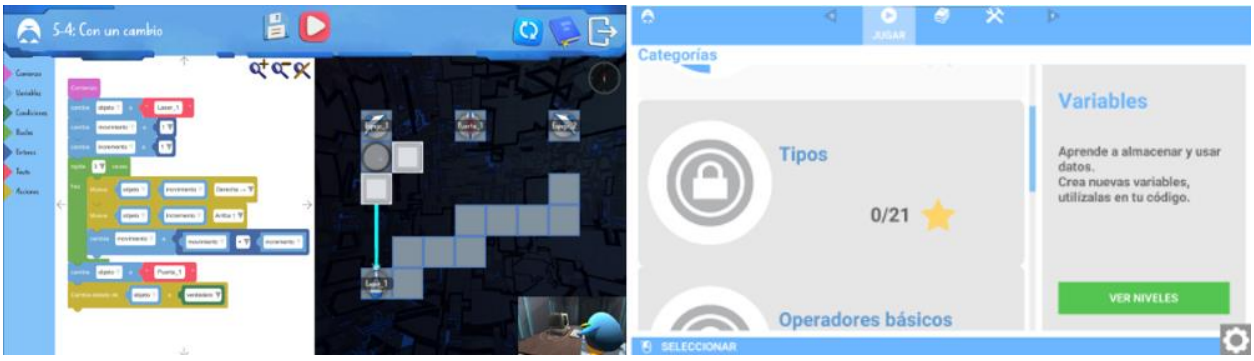

Figura 3-2. Pantalla del nivel y menú principal durante el año 2020/2021

3.2.2 Desarrollo durante el curso 2021/2022

En el segundo año de desarrollo, como parte del TFG "*Uso de Juegos Serios para Mejorar el Aprendizaje de la Programación en la Escuela*" por Tatiana Duarte Balvís, Ana Martín Sánchez y Paula Martínez Martínez [15], Se trató de mejorar la experiencia del usuario.

Después de hacer pruebas con usuarios, se realizaron ajustes significativos. Se redujo la carga textual y se agregó la funcionalidad de reiniciar niveles para mejorar la jugabilidad. Se introdujo una brújula y símbolos visuales para mejorar la comprensión de los movimientos. Se mejoró el diseño de los niveles y las ventanas emergentes, y se actualizó *uBlockly* para mejorar la programación basada en bloques. También se refinó el sistema de puntuación y se agregó la vista del perfil de usuario, junto con cambios en la interfaz del menú principal.

Figura 3-3. Menús rediseñados durante el año 2021/2022

3.2.3 Desarrollo durante el curso 2022/2023

Durante el tercer año de desarrollo en el curso 2022/2023, como parte del TFG "*Videojuegos y Programación: Evaluación de Conocimientos sobre Conceptos de Programación en Secundaria a través de un Videojuego*" por Henar Martín Domínguez y Gonzalo Cidoncha Pérez [16], el proyecto se enfocó en la expansión y la interactividad de la plataforma.

Se introdujeron mejoras en la creación de niveles, permitiendo establecer restricciones y guardar estados iniciales. Se implementó un sistema de comunidad que permitía a los alumnos subir niveles y a los profesores crear clases, utilizando un servidor API REST y un servidor web para administrar niveles, clases y usuarios. Estas adiciones ampliaron la funcionalidad del juego y lo convirtieron en una herramienta más interactiva y colaborativa.

Figura 3-4. Menús de la comunidad durante el año 2022/2023

Capítulo 4 - Desarrollo

" *La tecnología avanza a un ritmo exponencial, y aquellos que no se adaptan a* estos cambios están destinados a quedarse atrás." - John F. Kennedy

4.1 Apropiación de código

Como recién llegados al proyecto, fue crucial para nosotros comprender la arquitectura y el diseño del sistema, así como familiarizarnos con sus funcionalidades, código fuente, documentación y las herramientas y tecnologías utilizadas. Por esta razón, una de nuestras primeras acciones fue llevar a cabo una fase de apropiación del código.

La apropiación de código es el proceso mediante el cual una persona o un grupo adquieren, comprenden y utilizan activamente el código fuente de un programa o aplicación. Este proceso en primer lugar implica el estudio del código, con la intención de comprender cómo está estructurado, cómo funciona y qué propósito cumple cada parte del mismo. En segundo lugar la modificación y la personalización de este, con la finalidad de añadir modificaciones según las necesidades y preferencias. A continuación se describen los cambios realizados en las apropiaciones.

4.1.1 Apropiación de código - Articoding

El juego está desarrollado en *Unity [17]* y consta de 5 escenas principales. La primera escena es *LoadingScene*, donde se lleva a cabo el proceso de inicialización de los distintos sistemas del juego. La segunda es *MenuScene*, que sirve como el menú principal del juego y desde la cual se puede acceder a diversas áreas, como jugar niveles, ver el perfil, crear niveles, acceder a la comunidad, entre otras opciones. La tercera es *LevelScene*, una escena dedicada a cargar los diferentes niveles y permitir la jugabilidad principal para resolverlos. La cuarta es *BoardCreationScene*, que permite a los usuarios crear nuevos tableros y personalizar su distribución. Por último, la quinta escena es *EndScene*, que simplemente muestra los créditos del juego. Dentro del proyecto de *Unity* se realizaron los siguientes cambios:

- Scripts:
    - Renombrado de Scripts siguiendo la convención de nombres "UpperCamelCase", ejemplo: "PenguinLoadingBehavior".
    - Descripción de las clases mediante la etiqueta <summary>.
    - Mejora del alcance y accesibilidad de ciertas variables "public" que no eran necesarias en otros scripts sustituyéndolo por "[SerializeField] private".
    - Renombrado de Scripts siguiendo la convención de nombres "lowerCamelCase", ejemplo: "initialPoint".

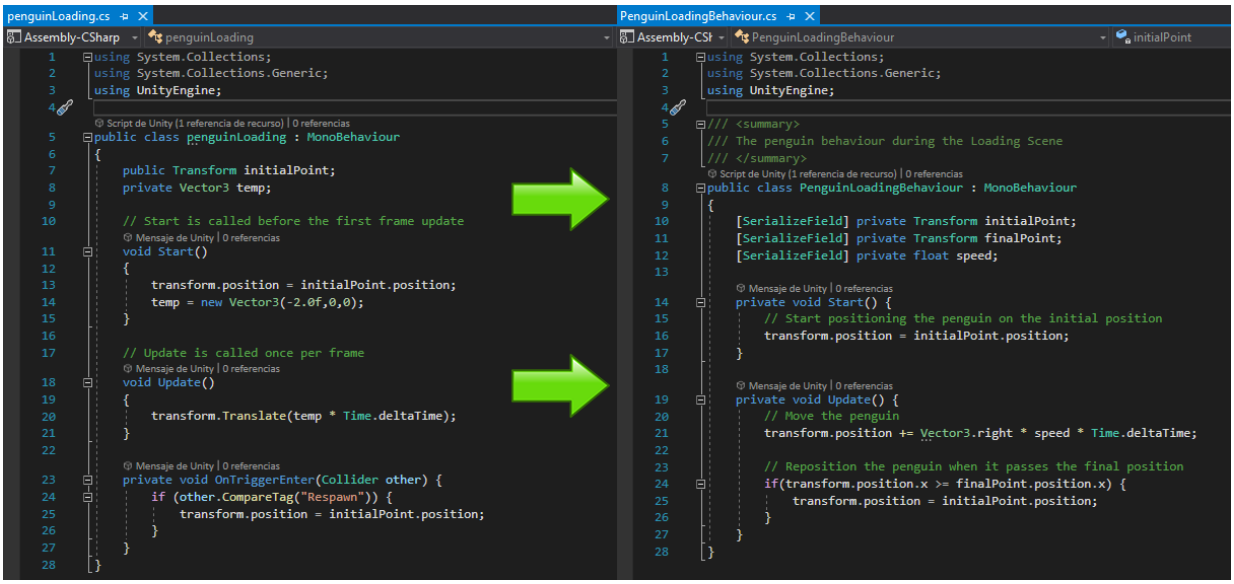

Figura 4-1. Ejemplo de un script refactorizado

- Escena:
    - Uniformado de los nombres de todos los elementos en inglés.
    - Separación entre palabras con espacios.
    - Adición de *GameObjects* vacíos llamados "----------" para separar elementos o grupos importantes en la escena.
    - Mejora de la jerarquía.

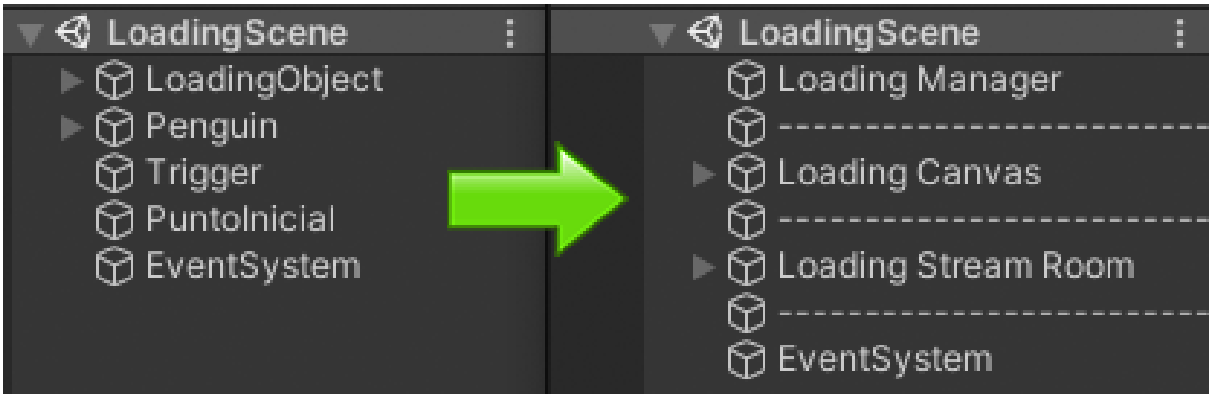

Figura 4-2. Ejemplo de una escena refactorizada

- Creación de la pestaña "Utils" con el propósito de facilitar la transición entre escenas y simplificar la función de "Play-Unplay" debido a la necesidad del juego de arrancar siempre desde la escena "LoadingScene".

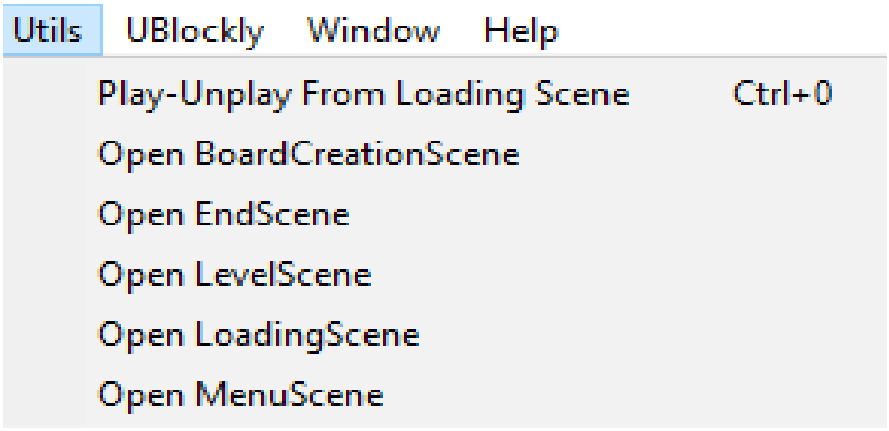

Figura 4-3. Pestaña Utils creada en Unity

- Actualización del plugin *Localizables* de *Unity* de la versión 0.10.0 a la versión 1.4.5 con sus respectivas refactorizaciones en los scripts dependientes.

- Optimización del sistema de guardado para eliminar la información redundante en los archivos y agilizar la carga de datos.

4.1.2 Apropiación de código - Server

ArticodingServer es una aplicación Java basada en Spring Boot, que utiliza Maven como gestor de dependencias y construcción del proyecto. Este servidor está diseñado para administrar usuarios, niveles y clases dentro de la comunidad de Articoding. El servidor mantiene funcionalidades para la definición de usuarios con el rol de profesor, quienes tienen la capacidad de crear y gestionar clases (*Classrooms*), así como controlar el acceso a las mismas.

La aplicación almacena información sobre los componentes necesarios y define la estructura de la base de datos, así como el manejo de las peticiones del juego y del cliente.

ArticodingServer está organizado en paquetes interdependientes. Los controladores (*controllers*) son responsables de recibir las peticiones HTTP tanto del cliente web como del juego, haciendo uso de las funcionalidades definidas en las clases del paquete servicios (*services*).

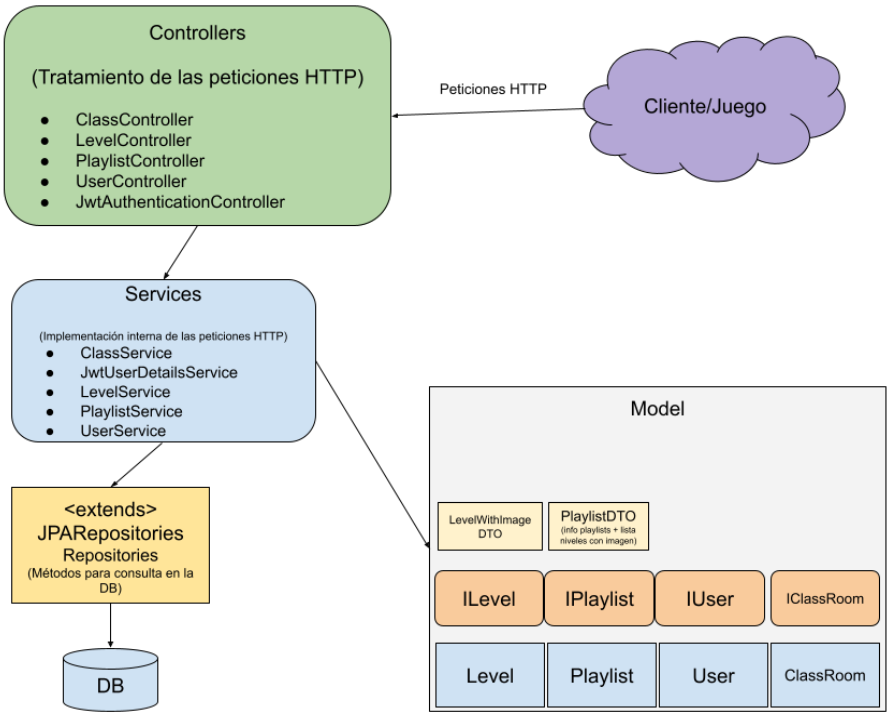

Figura 4-4. Diagrama de paquetes y dependencias

En el paquete *Repository* se encuentran las interfaces que facilitan la consulta de la base de datos, extendiendo *JPArepository*. Estas interfaces generan las consultas necesarias sobre la base de datos interpretando el nombre del método definido.

```
<T> Page<T> findByPublicLevelTrueAndTitleContains(Pageable pageable, Class<T> type, String title);
```

Figura 4-5. Ejemplo de un método que devuelve niveles públicos

El paquete *model* contiene las estructuras de datos para cada componente del juego, como usuarios o niveles. Estas estructuras también definen la estructura de las tablas entidad-relación en la base de datos.

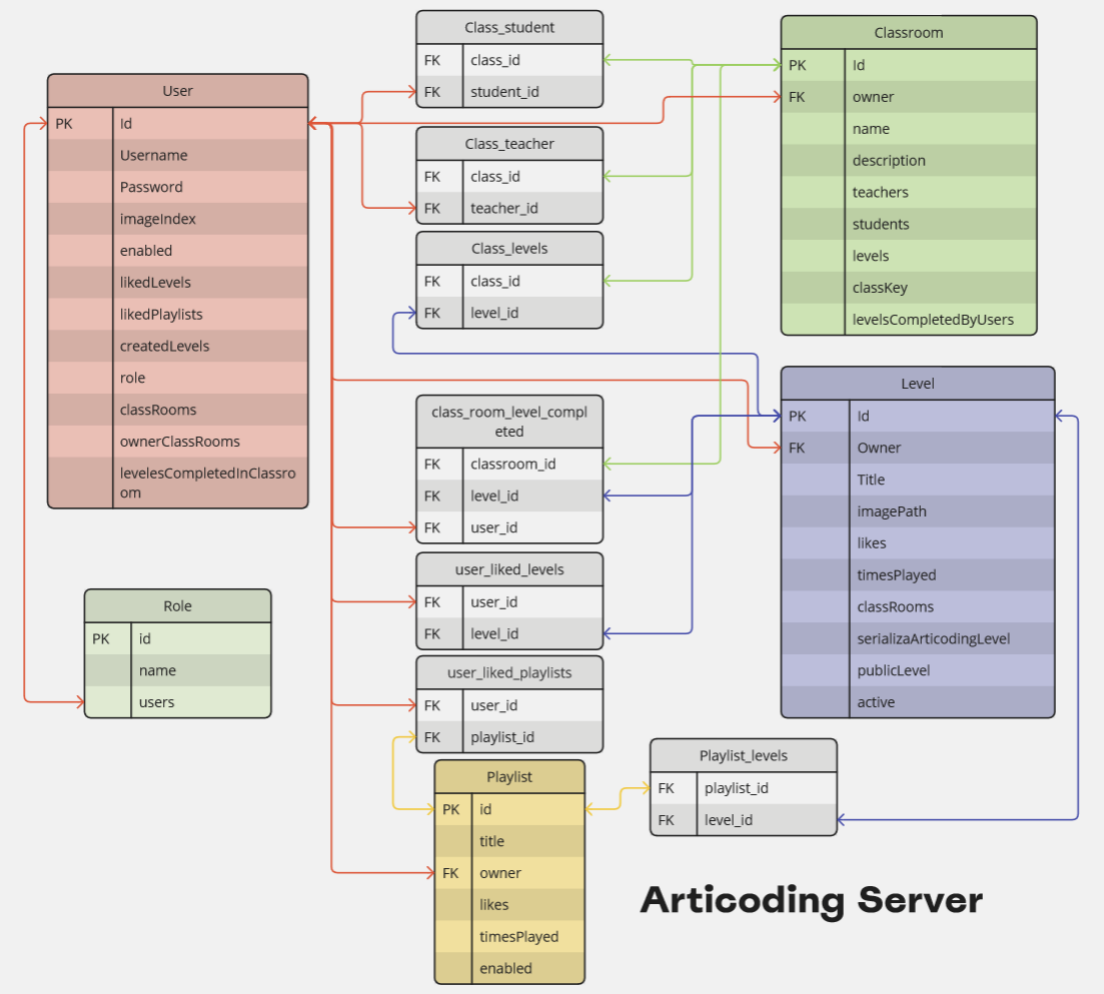

Figura 4-6. Diagrama entidad-relación del servidor

Para facilitar la comprensión y apropiación del código, se añadió al repositorio un documento descriptivo sobre la estructura de Articoding Server, donde se explica con mayor detalle lo mencionado anteriormente. También se amplió la información en el archivo README del proyecto, con el fin de simplificar el despliegue del servidor.

Además, se realizó una refactorización de algunos de los métodos en el código para mejorar su legibilidad y escalabilidad.

```
public Page<ILevel> getLevels(PageRequest pageRequest, Optional<Long> classId,  1 usage  HenarMD
                              Optional<Long> userId, Optional<Boolean> publicLevels, Optional<String> title) {
    User actualUser = userService.getActualUser();
    /** Si quiere todos los niveles de una clase*/
    if(classId.isPresent()) {
        /** Comrpobamos que exista la clase*/
        ClassRoom classRoom = classRepository.findById(classId.get())
                .orElseThrow(() -> new ErrorNotFound("clase", classId.get()));

        /**comprobamos que es su alumno/profesor o tiene role admin*/
        if(!roleHelper.isAdmin(actualUser)) {
            if (!classRoom.getStudents().stream().anyMatch(s -> s.getId() == actualUser.getId()) &&
                    !classRoom.getTeachers().stream().anyMatch(s -> s.getId() == actualUser.getId())) {
                throw new NotAuthorization("ver niveles de la clase " + classId.get());
            }
        }
        if(title.isPresent()) {
            return levelRepository.findByClassRoomsAndActiveTrueAndTitleContains(classRoom, title.get(), pageRequest, ILevel.class);
        } else {
            return levelRepository.findByClassRoomsAndActiveTrue(classRoom, pageRequest, ILevel.class);

        }
    } else if (userId.isPresent()) {
        /** Si quiere todos los niveles de un usuario, comprobamos que sea ADMIN */
        if(!roleHelper.isAdmin(actualUser)) {
            throw new NotAuthorization("ver niveles del usuario " + userId.get());
        } else {
            return levelRepository.findByOwnerAndActiveTrue(actualUser, pageRequest, ILevel.class);
        }
    } else {
        if(publicLevels.isPresent()) {
            /** Si quiere todos los niveles publicos*/
            if (title.isPresent()) {
                return levelRepository.findByPublicLevelTrueAndTitleContains(pageRequest, ILevel.class, title.get());
            } else {
                return levelRepository.findByPublicLevelTrue(pageRequest, ILevel.class);
            }
        } else {
```

Figura 4-7. Código previo a la apropiación

```java
public Page<LevelWithImageDTO> getLevels(PageRequest pageRequest, Comparator<ILevel> comparator,   ± CesarCRP97 +2 *
                                         Optional<Long> classId, Optional<Long> userId,
                                         Optional<Boolean> publicLevels, Optional<Boolean> liked,
                                         Optional<String> title, Optional<String> owner, Optional<Long> levelId) {
    List<ILevel> levels;
    User actualUser = userService.getActualUser();
    /** If there is a classId then returns levels from the classroom */
    if (classId.isPresent()) {
        levels = getLevelsFromClass(pageRequest, actualUser, classId, title);
    } else if (userId.isPresent()) {
        levels = getLevelsFromUserId(actualUser, userId);
    } else {
        if (publicLevels.isPresent()) {
            /** If publicLevels is true, returns all public levels. */
            levels = getPublicLevels(liked, title, owner, levelId);
        } else {
            /** If it's ADMIN then it returns every level */
            levels = getOwnedLevels(title, actualUser);
        }
    }

    //Converts the list of levels to a Page given PageRequest
    Page<ILevel> page = filteredLevelsToPage(pageRequest, comparator, levels);
    return toLevelWithImageDTO(page);
}
```

Figura 4-8. Código posterior a la apropiación

4.2 Corrección de errores y optimizaciones

Durante el año 2023, el profesor Miguel Ángel Garrido Blázquez de la Universidad Rey Juan Carlos probó y analizó el proyecto Articoding. Posteriormente, elaboró un informe que detallaba los errores encontrados durante la realización de estas pruebas. Dado que los alumnos del curso 22/23 no pudieron abordar la solución de estos errores a tiempo, nosotros nos encargamos de hacerlo.

La mayoría de los errores identificados estaban relacionados con aspectos visuales, como imágenes que no se ajustaban correctamente al tamaño en diferentes escalados de pantalla, así como errores en las traducciones. Tras realizar diversas pruebas, se implementaron ajustes en el redimensionamiento de todos los elementos del juego para garantizar su adecuación a una amplia variedad de tamaños de pantalla.

Durante el mes de enero, ante la posibilidad de llevar a cabo pruebas con usuarios reales, se priorizó la realización de pruebas intensivas del juego básico. Al mismo tiempo, se añadieron nuevas funciones con el objetivo de asegurar que la versión que se utilizara en las pruebas fuera estable y libre de problemas. Se prestó especial atención al sistema de guardado, el cual presentaba dificultades.

4.3 Mejoras para la experiencia de usuario

Tras conocer el estado del proyecto, se decidió agregar una serie de mejoras basándonos en las ideas obtenidas a través del análisis de otros juegos similares y proceder a la corrección de errores detectados en las pruebas de usuario del año anterior.

4.3.1 Nuevas animaciones

Se consideró que otorgarle mayor protagonismo al pingüino era una decisión acertada, dado que se ha verificado que los usuarios responden positivamente a su diseño. En consecuencia, se procedió a la incorporación de nuevas animaciones para enriquecer su presencia dentro del juego.

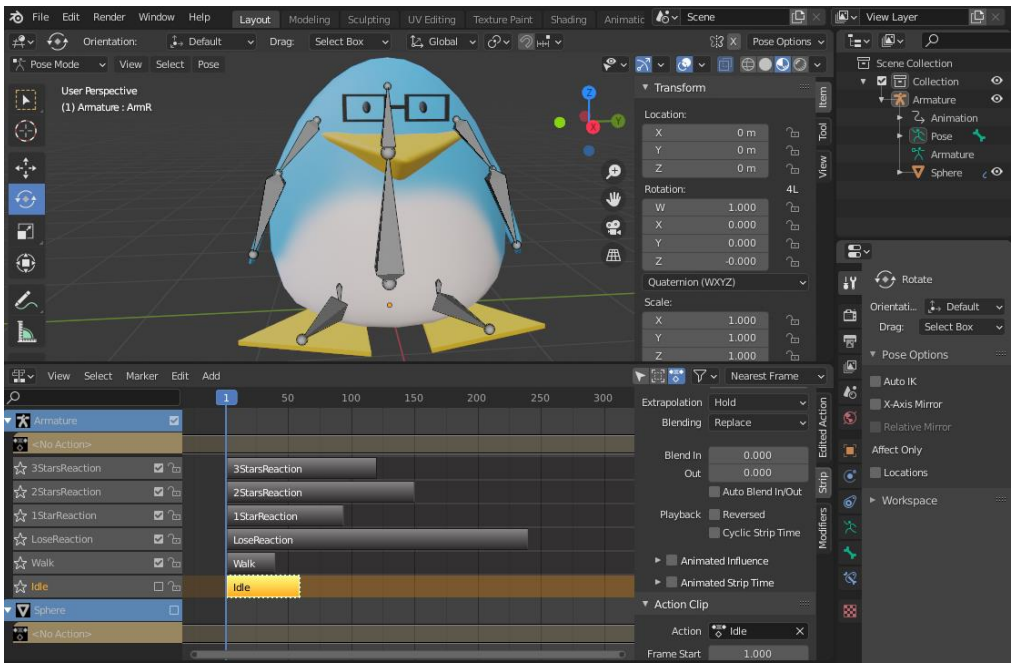

Figura 4-9. Creación de animaciones del pingüino en Blender

Con este propósito, se empleó el proyecto previo *penguinAnimated* de Blender para desarrollar cuatro animaciones, cada una representando un resultado posible al completar un nivel (desde fallar con 0 estrellas hasta obtener las 3). Una vez creadas las diversas animaciones, estas se integraron en el juego dentro de la ventana de solución de nivel, donde se muestran las animaciones correspondientes al resultado obtenido.

Figura 4-10. Ventana de resultado del nivel con el pingüino animado

4.3.2 Imágenes de los niveles guardados

Resulta evidente la importancia de contar con imágenes de vista previa en los niveles para una rápida identificación del contenido antes de ingresar, y aunque dichas imágenes estaban disponibles para los niveles básicos del juego, no se registraban en ningún momento las imágenes correspondientes a los niveles generados por los usuarios, lo cual afectaba negativamente la presentación visual de estos últimos. Además, teniendo en cuenta que uno de los objetivos primordiales de este proyecto era mejorar el sistema de comunidad, se anticipó la importancia que tendría la inclusión de una imagen previa para cada nivel.

Figura 4-11. Vista previa de niveles creados

Por consiguiente, se procedió a implementar la funcionalidad de guardado de dicha imagen mediante la adición de una cámara auxiliar que capturara la vista de su *viewport* como una textura 2D, la cual se almacenaba en la carpeta *LevelPreviewIcons* en el momento de guardar un nivel. Un aspecto considerado fue la necesidad de mantener un tamaño uniforme para todas las imágenes. Debido a la variación en las dimensiones de los niveles, fue necesario ajustar el nivel y centrarlo apropiadamente en la imagen resultante.

4.3.3 Medallas del perfil

Uno de los elementos que destacó en el juego *Super Mario Maker 2* fue el uso de medallas para recompensar al jugador al alcanzar hitos específicos dentro del juego. Por consiguiente, se decidió integrarlas en el juego, ya que se consideró que complementaban de manera óptima las mecánicas ya existentes.

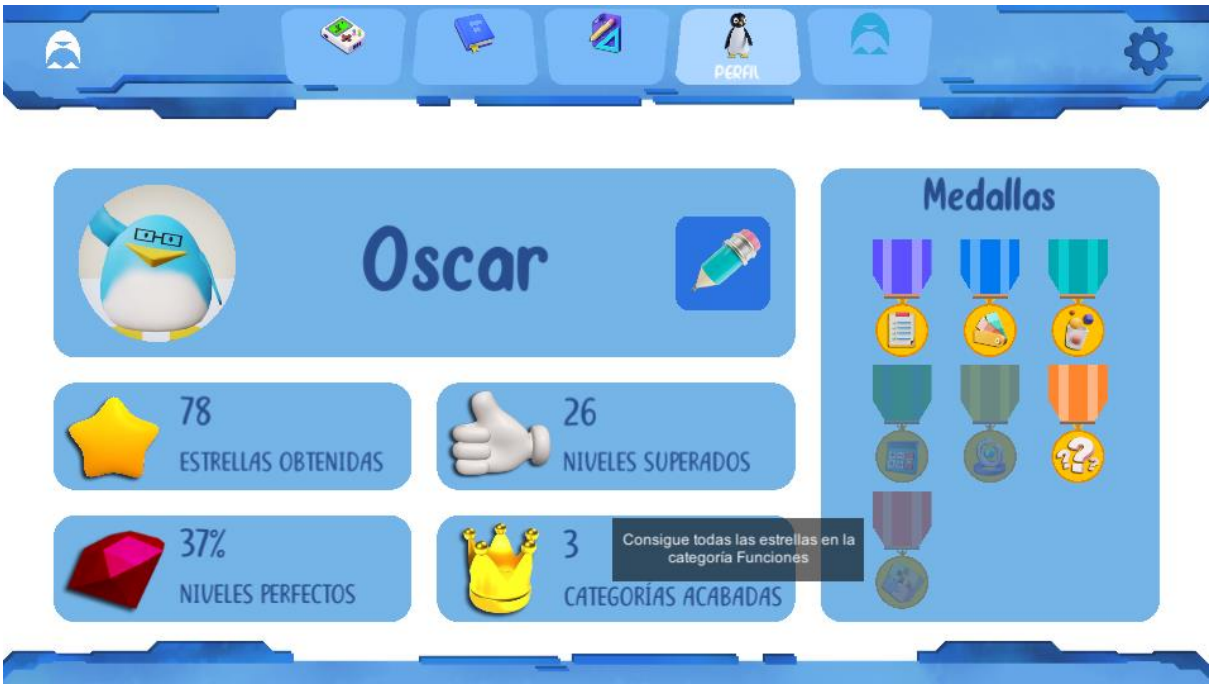

Figura 4-12. Página del perfil del usuario con las medallas

Las medallas solo se obtienen cuando el usuario logra completar todos los niveles de una categoría con 3 estrellas. Este sistema de recompensa proporciona una mayor satisfacción a los usuarios que se comprometen a completar el juego en su totalidad, sirviendo así como un incentivo para alcanzar dicho objetivo.

4.3.4 Rediseño de la interfaz gráfica del menú principal

Se rediseñó la interfaz del menú principal para mejorar la experiencia del usuario, haciéndola más fácil, intuitiva y atractiva. Para ello, primero se realizaron unos bocetos en papel.

El nuevo diseño se basa en el *modelo de carrusel lateral* que proporciona una forma intuitiva de navegar por las diferentes categorías de contenido. Los usuarios pueden deslizar horizontalmente para ver más opciones, tocar en los elementos del carrusel para dirigirse a ellos o utilizar los botones inferiores que actúan como atajos. Este diseño organiza las categorías mediante tarjetas en las que se incluyen: el título, el icono, la descripción, el número de estrellas conseguidas y el botón para seleccionarla.

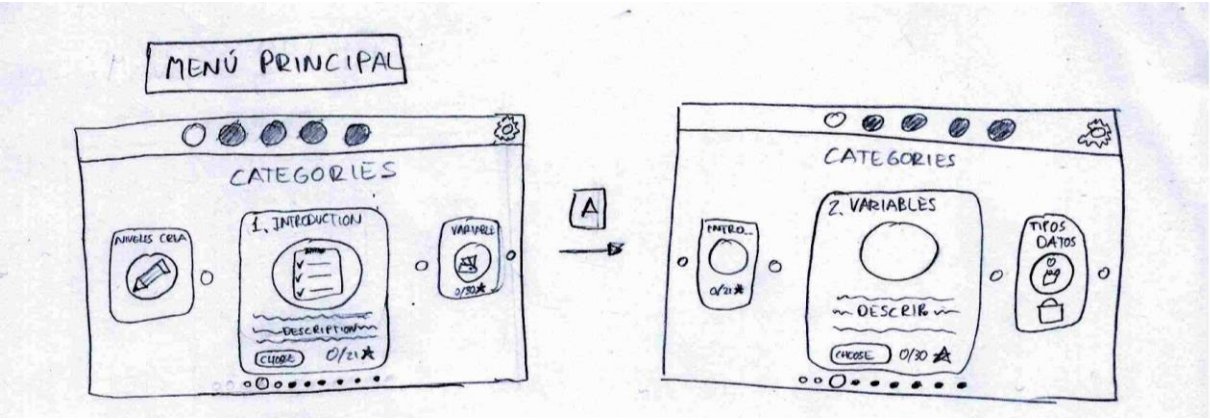

Figura 4-13. Bocetos a lápiz del rediseño del menú principal

Al seleccionar una categoría se abre un menú en el que se pueden ver todos los niveles a la vez con la información necesaria en cada uno: el título, la vista previa, el número de estrellas, un candado indicando que está bloqueado y un botón para jugar.

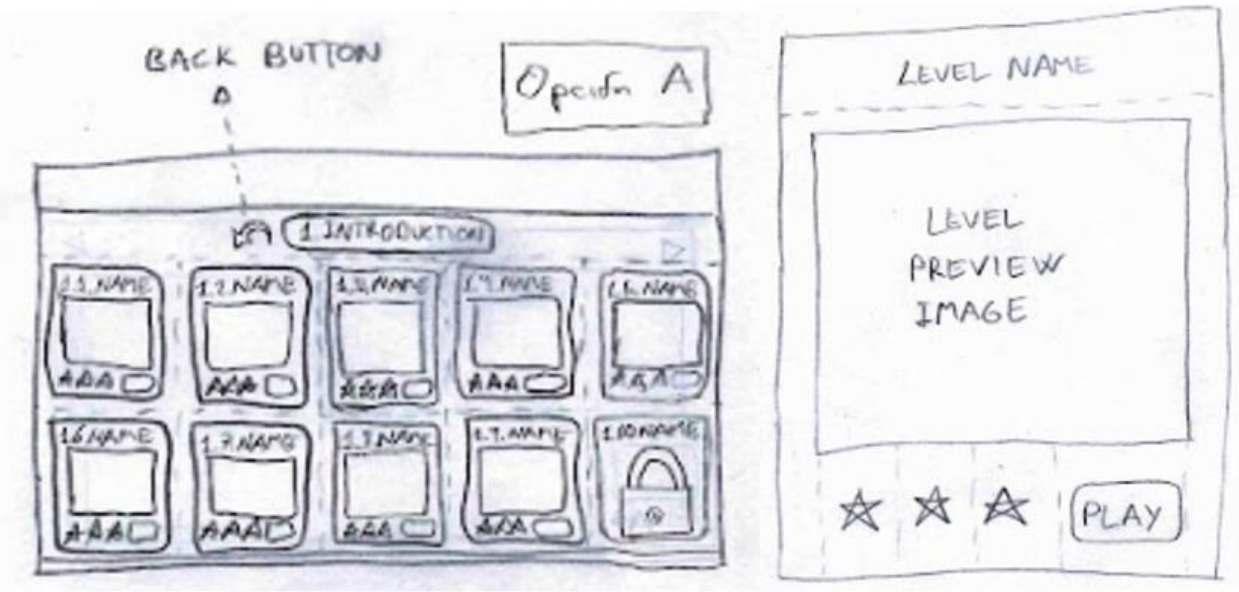

Figura 4-14. Bocetos a lápiz del rediseño del menú de los niveles

Tras refinar estos diseños sobre el papel, se procedió a la implementación de un prototipo funcional en *Figma [18]*, una herramienta de diseño de interfaz de usuario y experiencia de usuario que permite a los diseñadores crear, colaborar y prototipar diseños de manera efectiva en un entorno en línea.

Sobre este prototipo se realizaron varias iteraciones en las que primero se analizaron las fortalezas y debilidades del diseño y después se introdujeron cambios. Uno de los cambios fue incluir animaciones de expandir y contraer una categoría ocultando la información menos relevante al no ser la seleccionada. Otro fue la de asociar cada categoría a un color, manteniendo un equilibrio en la gama cromática y una consistencia en la saturación.

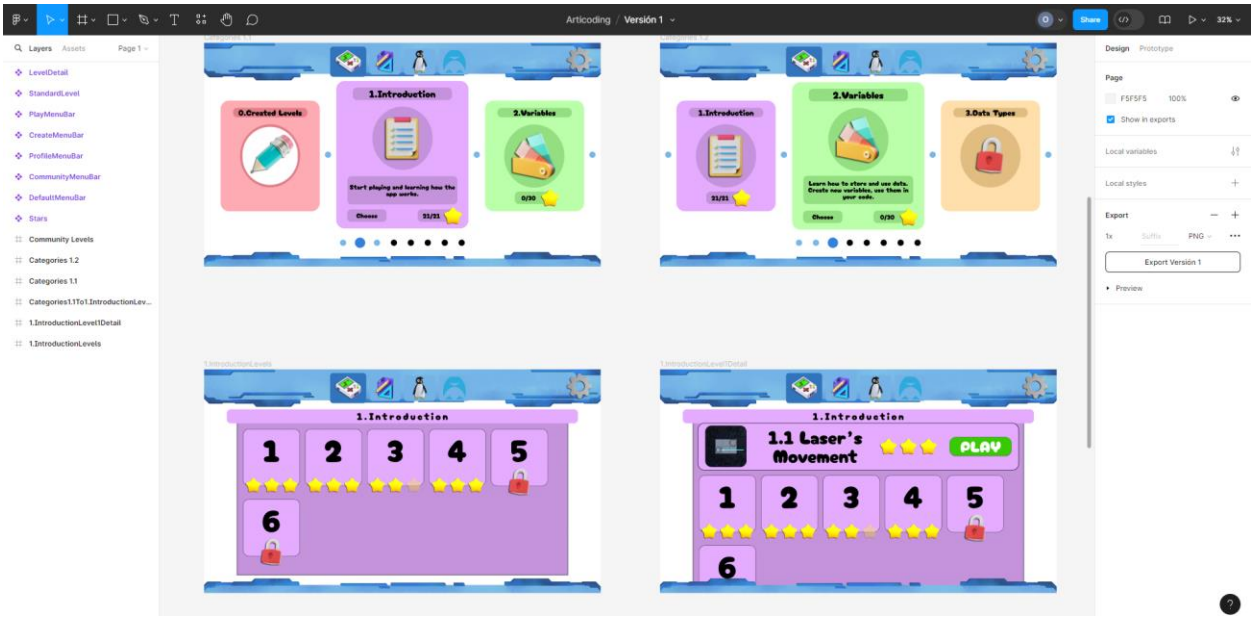

Figura 4-15. Prototipo del rediseño del menú principal del juego en Figma

Una vez se dispuso del diseño final, se procedió a su implementación dentro del juego, lo cual necesitó de la creación de nuevos *prefabs* y *scripts* asociados que permitieran añadir la funcionalidad pertinente para el nuevo diseño. El resultado fue el siguiente:

Figura 4-16. Comparativa del menú principal previo y posterior al rediseño

Figura 4-17. Comparativa del menú de niveles previo y posterior al rediseño

4.4 Rediseño de la Comunidad

Se reestructuró el sistema de la comunidad con la finalidad de proponer una visión de comunidad dirigida a un juego con un alcance más amplio, trascendiendo así la limitación de entornos cerrados exclusivamente destinados a estudiantes y profesores. Para lograr este objetivo, se tomó como referencia la inspiración obtenida del análisis de juegos anteriores y sus características distintivas. Además, se conservó la estética presente en los menús del juego base, proporcionando un desglose detallado de los requisitos esenciales para llevar a cabo la reconfiguración de dicha comunidad.

4.4.1 Concepto de la Comunidad Abierta

Se planteó una visión de comunidad destinada a un juego que tenga mayor alcance y no que tan solo sea jugable en entornos cerrados y limitados a estudiantes y profesores. Para ello, se tuvieron que añadir los siguientes elementos:

- Usuarios: los usuarios deben contar con la capacidad de acceder al sistema mediante la utilización de un nombre de usuario y una contraseña. El proceso de registro está disponible para los usuarios, permitiendo también que un docente pueda crear cuentas desde la interfaz web. El docente encargado de la creación de cuentas posee pleno acceso a las mismas, incluyendo los datos de inicio de sesión (nombre de usuario y contraseña), facultándolo para modificar las contraseñas en caso de ser necesario, como por ejemplo, en situaciones de olvido por parte del estudiante. Cada usuario cuenta con la capacidad de

personalizar su perfil mediante la selección de una imagen de entre las opciones predeterminadas.

- Niveles: cada nivel está identificado por un ID único y contiene información detallada, incluyendo su nombre, autor, imagen asociada, cantidad de *likes*, número de veces jugado y etiquetas descriptivas. Esta estructura permite a los usuarios filtrar los niveles según cada uno de estos parámetros. Los niveles pueden ser compartidos de manera pública por cualquier usuario, o añadidos por los profesores a sus clases, manteniendo el acceso restringido para los alumnos que no pertenezcan a dicha clase.
- Listas: las listas públicas de niveles son creadas por la comunidad y están disponibles para todos los usuarios. Cada lista se identifica con un ID único y contiene información como su nombre, autor, imagen representativa, cantidad de *likes*, número de veces jugadas y etiquetas descriptivas. La imagen de la lista se actualiza dinámicamente con las imágenes de los niveles que contiene.
- Clases: las clases representan listas privadas conformadas por niveles tanto privados, subidos por el profesor, como públicos. Dado su carácter privado, no requieren contar con números de *likes*, número de veces jugadas o etiquetas descriptivas. Únicamente los profesores tienen la facultad de crear clases. Estos tendrán acceso a una tabla que refleja el progreso de los alumnos, detallando qué niveles han completado cada uno de ellos. Para unirse a una clase, el usuario podrá ingresar un código proporcionado por el profesor o ser añadido directamente por este último mediante el servicio web. Este código identificador será único y se generará automáticamente al crear la clase.
- Clases con tiempo límite: las clases con límite de tiempo se caracterizan por tener una fecha de finalización establecida. Una vez transcurrido este plazo, ningún nivel está disponible para su juego, conservándose los datos en ese momento para su evaluación por parte de los profesores. La duración de la fecha límite puede variar, desde periodos cortos como *un examen de 2 horas* hasta plazos más amplios como *un trabajo de 7 días*, según las necesidades específicas de la actividad académica.

	Niveles	Listas	Clases
Visibilidad	Públicos por usuarios Privados por profesores	Públicas	Privadas
Tipos de niveles	-	Públicos	Públicos de usuarios Privados de profesores
Edición	No	Sí	Sí
Estadísticas	No	No	Por el profesor

Figura 4-18. Comparativa de los elementos del nuevo diseño de la comunidad

Las clases se conciben como entidades independientes debido a una serie de razones fundamentales. Primero, restringir la capacidad de subir niveles o listas de reproducción de manera privada fuera del entorno de clase contribuye al enriquecimiento de la comunidad, ya que todas las contribuciones son públicas y accesibles para todos los usuarios.

Otro motivo para considerar las clases como entidades independientes de las listas de reproducción es garantizar que los alumnos puedan acceder a ellas sin complicaciones y sin perderse en la búsqueda. Esta separación facilita el acceso directo a las clases para los alumnos, especialmente porque el profesor puede agregarlos directamente, lo que les permite aparecer automáticamente al iniciar sesión con su usuario.

Además, las clases ofrecen al profesor la capacidad de monitorizar el progreso de los alumnos. Asimismo, es fundamental proteger la privacidad de los usuarios, evitando que los creadores de listas tengan acceso a información sobre qué usuarios han jugado sus listas.

4.4.2 Escenarios hipotéticos

Para reafirmar las elecciones tomadas en el nuevo concepto de comunidad se plantearon una serie de escenarios hipotéticos. Estos escenarios son herramientas valiosas en el diseño y desarrollo de aplicaciones, ya que permiten prever y analizar el

comportamiento del sistema en situaciones variadas que podrían no estar cubiertas por casos de uso convencionales.

4.4.2.1 Escenario 1: Profesor Matías

Matías, quien ha descargado el juego gratuitamente desde *Github* pero aún no posee el rol de profesor, decide introducirlo a sus alumnos. Durante las primeras tres sesiones, los alumnos completan los niveles de la aventura principal y quedan encantados con la experiencia. Motivado por el entusiasmo de sus estudiantes, Matías planea que en la cuarta sesión cada alumno cree su propio nivel.

Después de que los alumnos creen sus niveles, le proporcionan a Matías los ID de cada uno. Con esta información, Matías compila una lista de reproducción con los niveles de sus estudiantes y les informa cuál es la lista para que jueguen y valoren los niveles creados por sus compañeros.

Dado el éxito de las sesiones de juego y el aprendizaje significativo que los alumnos están experimentando en cuanto a los conceptos básicos de programación, Matías decide ponerse en contacto con los administradores para obtener el rol de profesor. Una vez que lo obtiene, Matías decide que un porcentaje de la nota de cada alumno se determinará mediante una serie de niveles.

Matías crea una clase con los 10 niveles públicos más populares de la comunidad, los prueba para asegurarse de que no sean demasiado difíciles y los selecciona para la evaluación. En el día de la clase, Matías proporciona a los alumnos el código de acceso a la clase, permitiéndoles utilizar los usuarios que crearon previamente. Después de la clase, Matías revisa qué usuarios han completado cada nivel y utiliza esta información para evaluar a sus alumnos, reconociendo así su progreso y desempeño en la materia.

4.4.2.2 Escenario 2: Estudiando Sofia

Sofía, una estudiante de 2º de la ESO, muestra un gran interés por la asignatura de Informática y ya posee conocimientos previos en programación, especialmente en *Scratch*, lo que le permite encontrar los niveles básicos del juego Articoding bastante

simples. Cuando la profesora les indica que ingresen a la página de la comunidad, Sofía lo hace con facilidad y de inmediato se dirige a la sección de niveles.

Mientras la profesora explica el procedimiento para unirse a la clase, Sofía se adelanta y comienza a intentar completar el nivel con más "Me gustas" que ha encontrado, utilizando la función de ordenar por cantidad de "Me gustas".

4.4.2.3 Escenario 3: Usuario fuera de clase Jorge

Jorge, un adolescente de 14 años, ha sentido curiosidad por el mundo de la programación, pero se ha encontrado con dificultades y frustraciones al intentar aprender. Sin embargo, un día descubre Articoding y se siente atraído por su curva de aprendizaje gradual y accesible, lo que le permite superar los desafíos del juego en pocas horas. Impulsado por su entusiasmo, decide registrarse en la Comunidad para seguir explorando.

Al principio, Jorge se dedica a jugar los niveles más populares de la comunidad y también experimenta con algunas listas de reproducción. Motivado por su experiencia, decide crear sus propios niveles, los cuales reúne en una lista de reproducción. Pronto, sus niveles ganan popularidad en la comunidad, y uno de ellos se convierte en el más apreciado por los usuarios, incluso siendo utilizado por profesores como Matías en sus clases, sin que Jorge lo sepa.

Con el tiempo, Jorge comienza a perder interés en el juego, ya que ninguno de los niveles le presenta un desafío. Decide retomar un curso de programación que había abandonado anteriormente debido a su dificultad, pero ahora se encuentra sorprendido al descubrir que comprende perfectamente los conceptos que antes le resultaban complicados, como los bucles. El curso ahora le parece incluso fácil, demostrando su progreso y habilidades adquiridas a través de su experiencia en Articoding.

4.4.2.4 Escenario 4: Profesor Eduardo

Antes de la clase, Eduardo elabora una serie de niveles personalizados utilizando el editor integrado en el juego. Luego, accede a la plataforma web y crea una clase, en la cual añade los niveles que ha diseñado previamente. Posteriormente, desde la

misma plataforma web, procede a crear cuentas de usuario para cada uno de sus alumnos, asignándoles un nombre de usuario y una contraseña. Seguidamente, incorpora a los alumnos a la clase que ha creado.

Durante la clase, Eduardo distribuye las credenciales de inicio de sesión a cada alumno y les explica cómo acceder a la comunidad, dirigirse a la clase correspondiente y jugar los niveles designados.

Después de la clase, Eduardo puede consultar la plataforma web para verificar qué usuarios han completado cada nivel, utilizando esta información para evaluar el progreso y desempeño de sus alumnos.

4.4.3 Diseño de la interfaz gráfica

Se diseñó la interfaz de comunidad con idea de mantener una buena experiencia de usuario, haciéndola intuitiva y atractiva a partes iguales.

Se crearon dos bocetos diferentes, diseñados por cada miembro del grupo, en los que cada uno plasmó su idea de la comunidad para poder ponerlos en común y llevar a cabo una discusión sobre qué elementos de cada boceto eran mejores.

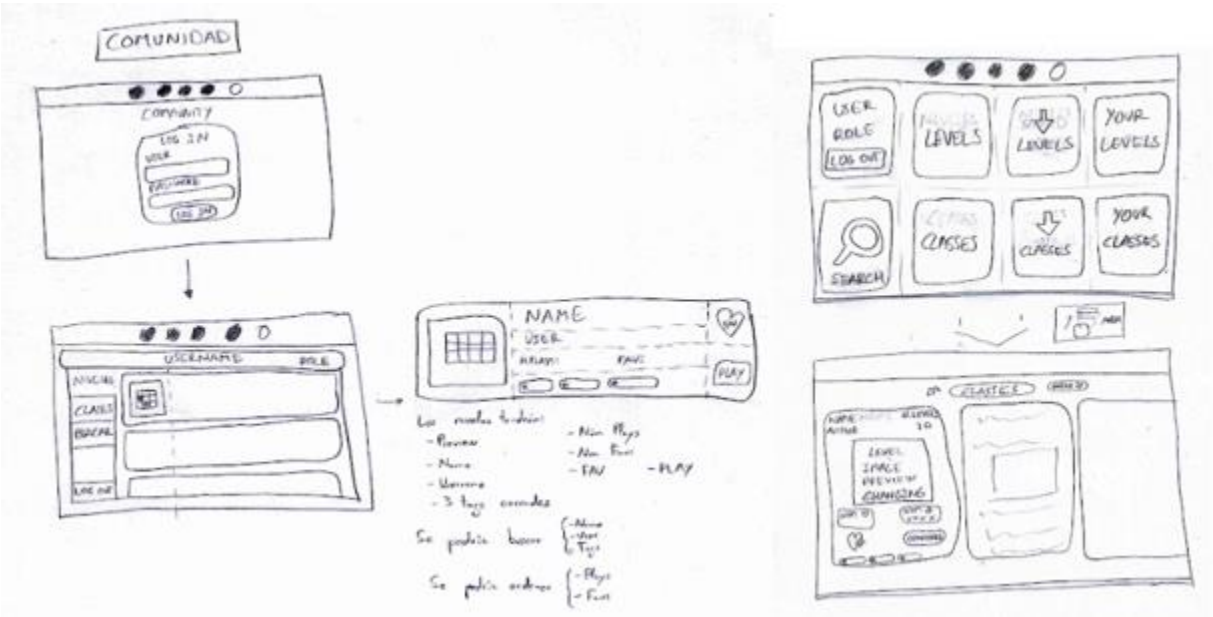

Figura 4-19. Bocetos del nuevo diseño de la comunidad de Óscar

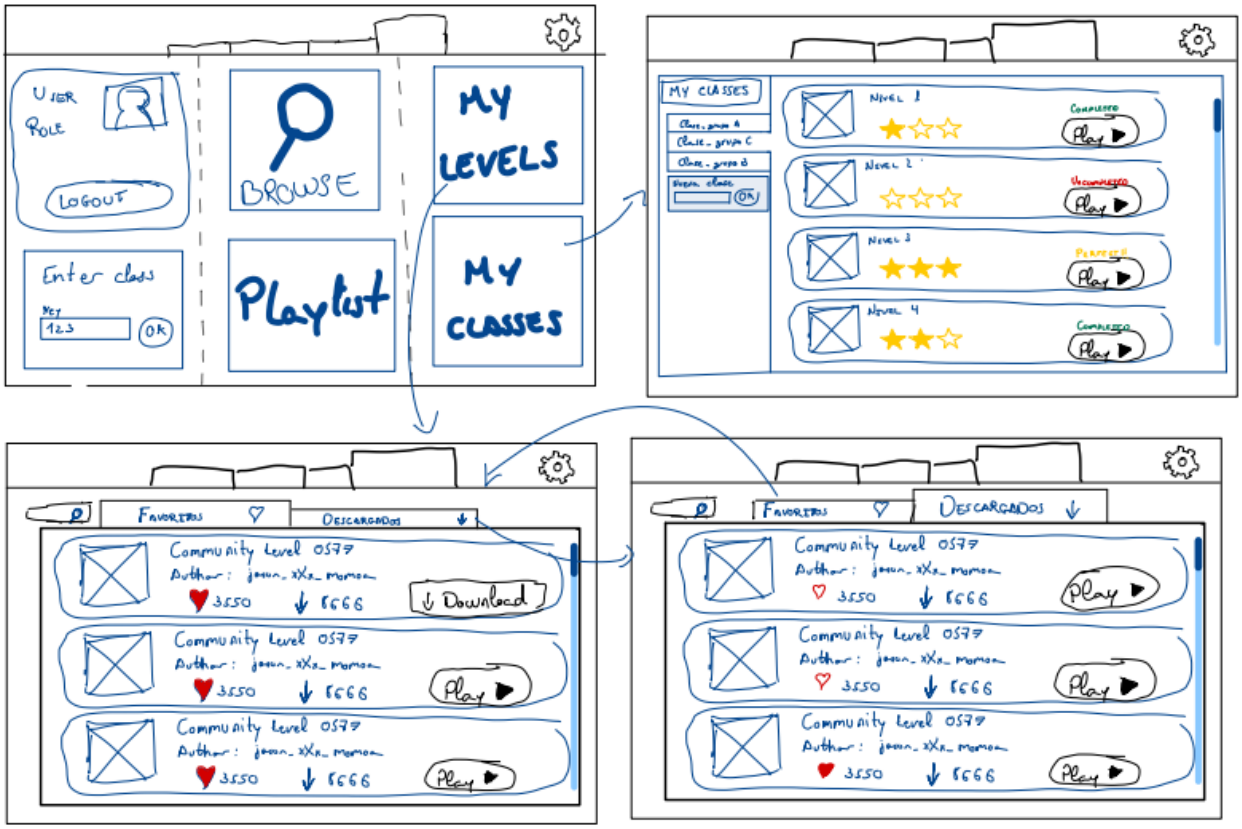

Figura 4-20. Bocetos del nuevo diseño de la comunidad de César

Ambos diseños presentaban de maneras muy similares los menús con botones para acceder de forma directa a las páginas de niveles, listas y clases. Entre los diseños se discrepaba en el diseño de las tarjetas de nivel, siendo unas horizontales y otras verticales. Tras valorar los puntos fuertes de cada diseño se tomó la decisión de utilizar las tarjetas verticales por su facilidad para ajustarse a diversos tipos de pantallas.

Figura 4-21. Prototipo del nuevo diseño de la comunidad en Figma

Tras tener los bocetos se procedió al desarrollo de un prototipo utilizando Figma. Sobre él, se realizaron las modificaciones que consideramos necesarias tras probar la experiencia de usuario con estas interfaces y contrastar nuestra visión con Baltasar y Antonio. Al final se obtuvo un prototipo efectivo que tan solo hubo que introducir en el juego, dejando el siguiente resultado:

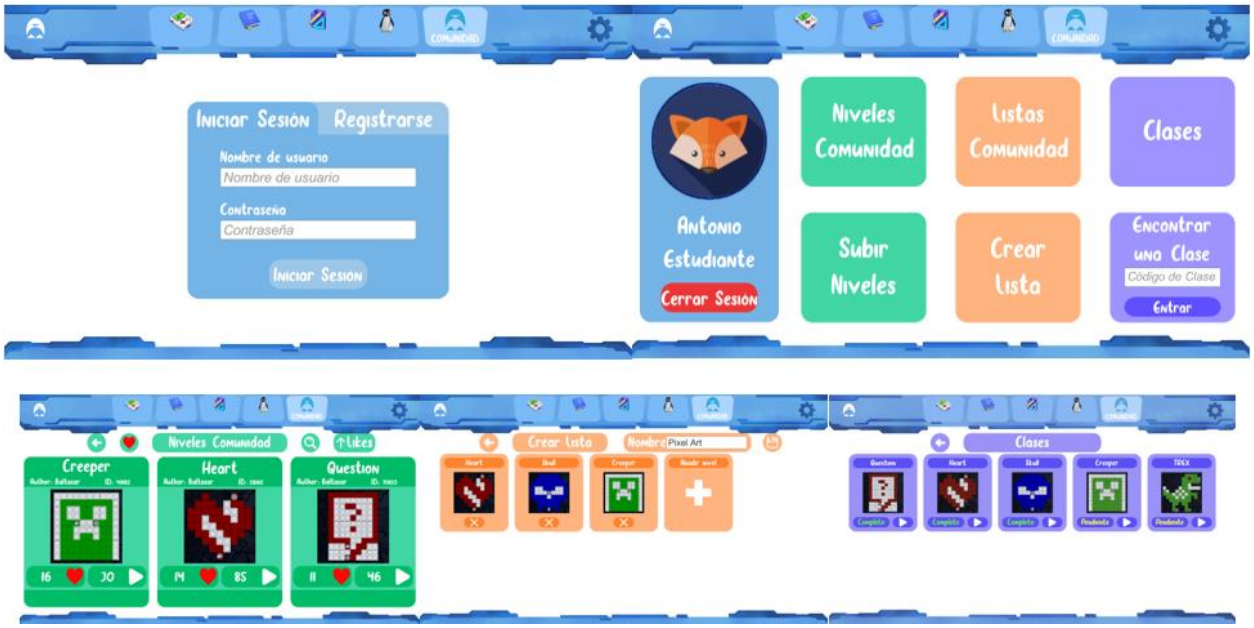

Figura 4-22. Resultados del nuevo diseño de la comunidad en el juego

4.4.4 Implementación

A continuación se detallan las funcionalidades implementadas en la Comunidad Articoding, abarcando aspectos tanto del juego como del servidor y la plataforma web.

4.4.4.1 Funcionalidad de iniciar sesión / registrarse

Al seleccionar la pestaña "Comunidad", el primer elemento que se presenta es el menú para iniciar sesión o crear una cuenta nueva. En el contexto del juego, la funcionalidad de inicio de sesión ya estaba implementada, por lo que solo se requirió añadir la opción de registro. Para ello, se incorporó un menú que permite alternar entre las pestañas de inicio de sesión y registro de usuarios. El proceso de registro de un nuevo usuario implica proporcionar un nombre de usuario, una contraseña y su confirmación. Una vez completado este proceso, el usuario es creado y se accede automáticamente con las credenciales proporcionadas.

Figura 4-23. Menús de inicio de sesión y registro de usuarios

En lo que respecta al servidor, se añadió un nuevo *endpoint* para el registro de usuarios, el cual recibe una solicitud POST con un formulario de usuario (*UserForm*), manteniendo el mismo flujo que se seguía previamente desde la página web. La diferencia radica en que ahora no es necesario solicitar una autenticación previa, ya que anteriormente solo un usuario con un rol superior podía crear nuevas cuentas.

4.4.4.2 Funcionalidad del menú principal

Después de iniciar sesión o registrarse, los usuarios acceden al menú principal, una vista intermedia que les permite visualizar la información de su perfil, incluyendo su nombre, rol e imagen de perfil. Además, en este menú principal se encuentran botones para acceder a diversas secciones específicas, donde los usuarios pueden: ver los niveles de la comunidad, subir sus propios niveles, consultar las listas de la comunidad, crear sus propias listas, visualizar las clases a las que pertenecen y unirse a otras clases mediante un código de acceso.

Figura 4-24. Menú de comunidad

4.4.4.3 Funcionalidad de cambiar imagen de perfil

Para modificar la imagen de usuario, al hacer clic en ella se despliega un menú que presenta todas las opciones disponibles. Al seleccionar una de las opciones simplemente haciendo clic en ella, se efectúa la elección. En cuanto al almacenamiento de datos, la selección de la imagen de perfil se registra como un

número entero dentro de la clase de usuario. Este número se corresponde con la imagen específica en el menú principal y se actualiza mediante una solicitud PUT.

Figura 4-25. Menú selección imagen de usuario

4.4.4.4 Funcionalidad añadir imágenes de los niveles

Los niveles emplean las imágenes que previamente han sido guardadas para cada nivel creado, tal como se mencionó en el punto 4.2. Durante el proceso de subida de niveles, las imágenes son almacenadas localmente en la máquina que hospeda el servidor. Con este propósito, se implementó un nuevo *DTO* (*Data Transfer Object*) que incorpora tanto la información del nivel como la imagen asociada. Al recibir la petición, la imagen se extrae como un array de bytes. Posteriormente, se genera un nombre único que se utiliza para almacenarla en una carpeta denominada *levelImages*. En la base de datos, se guarda únicamente el nombre del archivo, lo que permite su posterior recuperación.

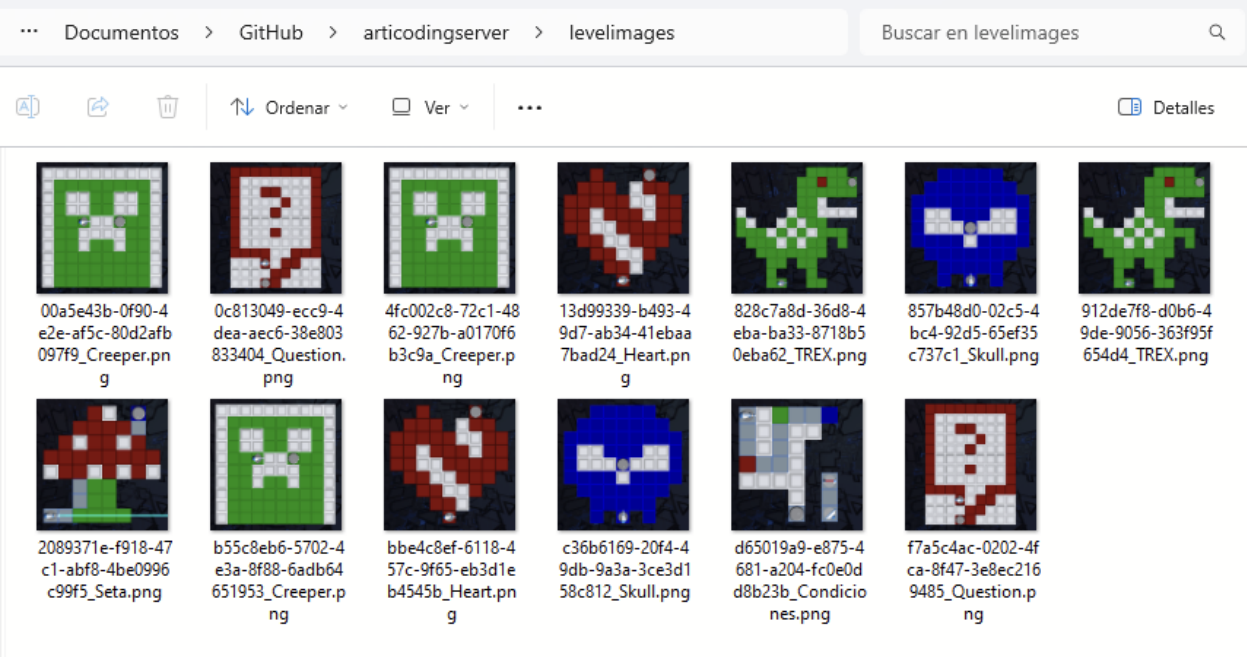

Figura 4-26. Carpeta levelImages en el servidor

4.4.4.5 Funcionalidad de ver lista de niveles

Inicialmente, se procedió a la creación de un *prefab* para las tarjetas de nivel, el cual se configuró para adaptarse a la información de cada nivel descargada desde el servidor. Posteriormente, se implementó una página que incluía los botones de volver al menú, abrir el menú de búsqueda avanzada y ordenar, junto con una lista de tarjetas de nivel dispuestas en un desplazamiento lateral.

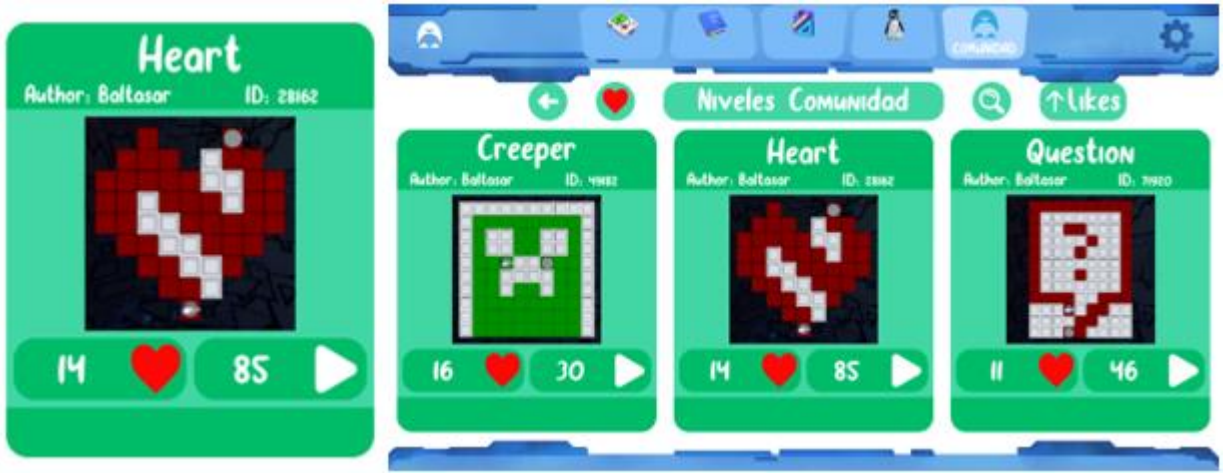

Figura 4-27. Diseño tarjetas de niveles y la página de buscar niveles

Para obtener una lista de niveles del servidor, se realiza una solicitud GET. Inicialmente, los niveles se recuperaban como un objeto del tipo Page<ILevel>, que almacenaba la información obtenida de la respectiva base de datos. Sin embargo, al introducir la funcionalidad de imágenes, fue necesario modificar la estructura para devolver la página del nuevo *DTO* que incluye la imagen. Con el enfoque adoptado, se modificaron los requisitos para la visualización del listado de niveles.

Se integraron dos contadores adicionales en las tarjetas de nivel, uno para registrar el número de veces que un nivel ha sido jugado y otro para indicar la cantidad de usuarios que le han dado *like*. Para lograr esto, se implementaron *endpoints* en el servidor que permiten registrar las solicitudes PUT destinadas a actualizar los contadores correspondientes.

4.4.4.6 Funcionalidad de filtrar y ordenar

Se implementó un botón para permitir el cambio rápido entre diferentes métodos de ordenación de los niveles, ya sea por el número de veces jugado o por el número de *likes*. Además, se agregó una ventana emergente que permite a los usuarios introducir el identificador de un nivel, el nombre del nivel o el nombre del autor para realizar una búsqueda filtrada basada en estos datos.

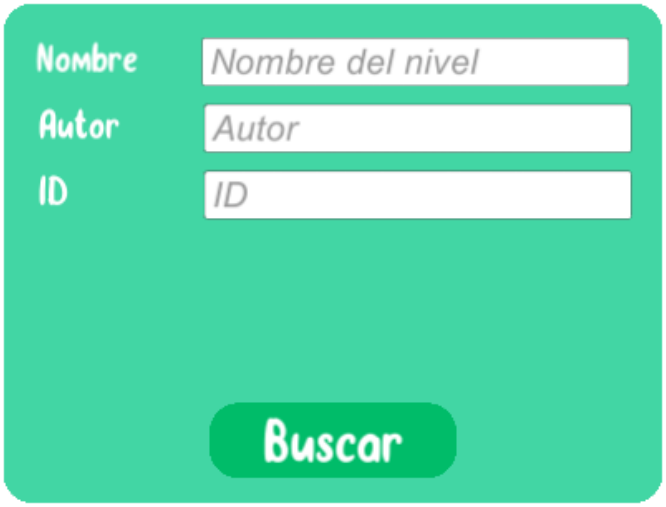

Figura 4-28. Menú de búsqueda avanzada

Durante el proceso de implementación de filtrado y ordenación, nos dimos cuenta de que la manera previa de hacer las peticiones a la base de datos se volvía poco manejable y el código era poco escalable. Por ello, permitir el uso simultáneo de todos los filtros requería la definición de tantos métodos como combinaciones únicas de filtros existieran, lo que resultaba en un aumento exponencial de código.

Ante esta situación, fue necesario replantear el código y el enfoque de las consultas con idea de mejorar la escalabilidad. En primer lugar, se descompuso el código del método principal en métodos auxiliares más específicos y reducidos. Asimismo, se realizaron modificaciones en los métodos de consulta a la base de datos para que devolvieran los datos en forma de *Streamable*, lo que facilitaba la realización de las transformaciones necesarias.

```
public Page<PlaylistDTO> getPlaylists(PageRequest pageRequest, Comparator<IPlaylist> comparator,
                                      Optional<Long> userId, Optional<Long> playlistId,
                                      Optional<Boolean> liked, Optional<Boolean> publicPlaylists,
                                      Optional<String> title, Optional<String> owner) {
    List<IPlaylist> playlists;
    User actualUser = userService.getActualUser();

    if (publicPlaylists.isPresent() && publicPlaylists.get()) {
        playlists = getPublicPlaylists(userId, title, owner, playlistId, liked);
    } else {
        playlists = getOwnedLevels(title, actualUser);
    }

    Page<IPlaylist> page = filteredPlaylistsToPage(pageRequest, comparator, playlists);
    return toPlaylistDTO(page);
}
```

Figura 4-29. Función getPlaylists en la clase PlaylistService

4.4.4.7 Funcionalidad de subir niveles

En el menú de subida de niveles, se presentan todos los niveles que se encuentran almacenados localmente en el juego, junto con un botón que permite su envío (siempre y cuando dicho nivel no haya sido subido anteriormente).

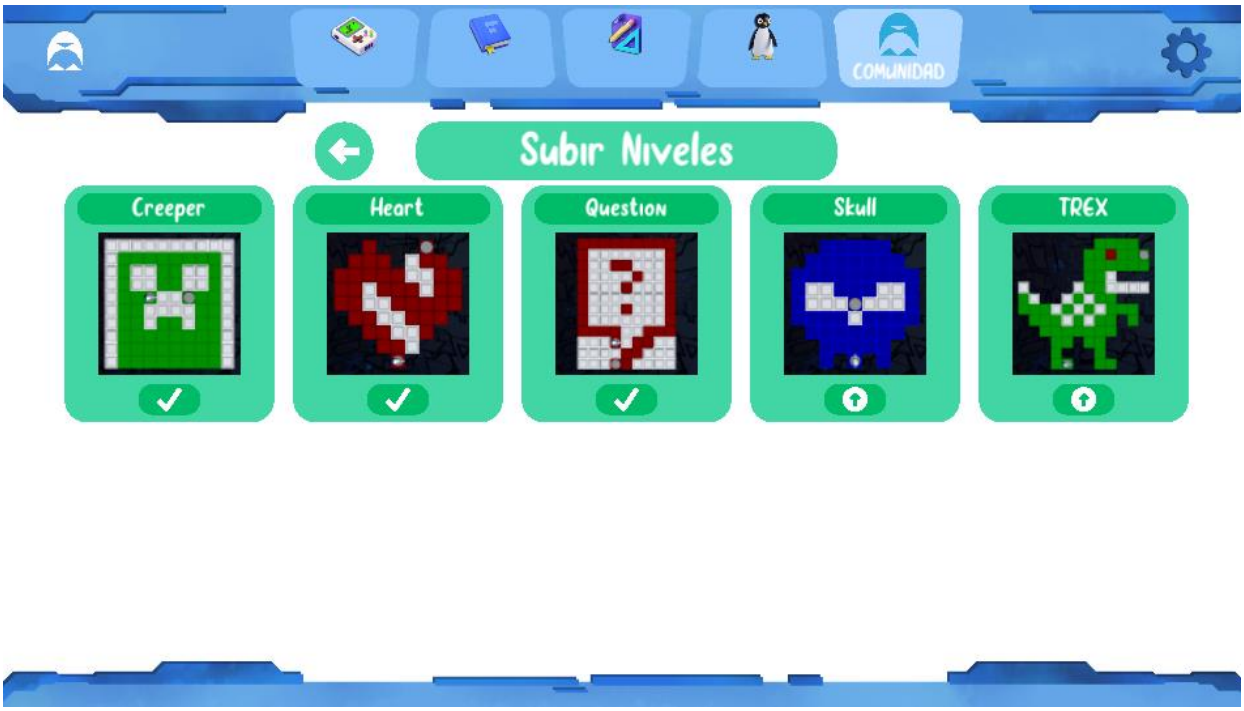

Figura 4-30. Menú para subir niveles

Es importante destacar que la subida de niveles solo es posible desde el juego. Cuando el usuario pulsa el botón para subir un nivel, el juego emite una solicitud POST con un formulario de nivel (*LevelForm*) en el cuerpo de la petición. Posteriormente, dicho formulario se formatea adecuadamente, la imagen asociada se guarda localmente y tanto la información del nivel como la imagen se registran en la base de datos.

4.4.4.8 Funcionalidad de ver lista de listas

Se desarrolló una variante del prefab de los niveles que incorporaba la funcionalidad de alternar entre las imágenes asociadas a cada nivel dentro de la lista. Posteriormente, se reutilizó la página de visualización de niveles de la comunidad, sustituyendo simplemente las tarjetas de niveles por tarjetas de listas. Al iniciar la reproducción de una lista, se despliega una página que presenta todos los niveles que forman parte de dicha lista.

Figura 4-31. Diseño tarjetas de listas y la página de buscar listas

Para implementar esta funcionalidad en el servidor, se crearon una serie de clases análogas a las ya definidas para otras entidades. Estas incluyen: un controlador (*controller*) para la recepción de peticiones, un servicio (*service*) para encapsular la lógica interna del servidor y un repositorio (*repository*) para gestionar las llamadas a la base de datos.

4.4.4.9 Funcionalidad de crear listas

Para la creación de listas, se presenta una página de interfaz sencilla donde los usuarios pueden agregar o eliminar niveles, asignar un nombre a la lista de reproducción y guardar los cambios realizados. Al hacer clic en el botón de "añadir niveles", se abre una página de búsqueda de niveles donde se pueden utilizar diversos filtros para encontrar el nivel deseado, con la opción de agregarlo directamente desde la tarjeta del nivel.

Figura 4-32. Menú de creación de listas

El servidor devuelve los niveles de manera similar a como lo haría al visualizar la lista de niveles públicos. Una vez creada la lista y asignado un nombre, al presionar el botón de "guardar", el juego envía una solicitud POST al servidor para registrar la lista. Cabe destacar que cualquier usuario registrado tiene la capacidad de crear una lista.

4.4.4.10 Funcionalidad de ver clases

Al acceder a la página de clases, la primera sección que se muestra son las tarjetas correspondientes a las clases a las cuales el alumno está inscrito. Estas tarjetas incluyen el nombre de la clase y una descripción opcional proporcionada por el profesor. Al seleccionar una clase específica, se despliegan tarjetas que contienen información sobre los niveles incluidos en dicha clase. Estas tarjetas muestran menos detalles que las utilizadas en las páginas de niveles anteriores, pero incorporan información adicional sobre el estado del nivel (completado o pendiente).

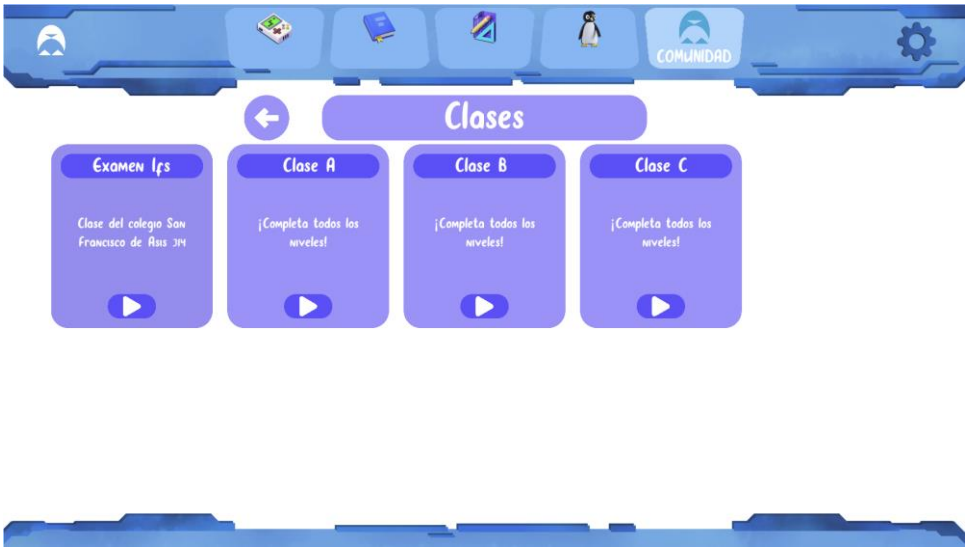

Figura 4-33. Menú de clases

Figura 4-34. Tarjetas con los niveles de una clase

En el servidor, la gestión de una solicitud para visualizar clases varía según el rol asignado al usuario que envía la petición. Un profesor tiene la capacidad de ver las clases que ha creado y aquellas en las que está inscrito, mientras que un usuario normal únicamente puede acceder a las clases en las que está registrado.

Dentro de cada clase, el jugador puede visualizar y jugar los niveles incluidos en la misma. Al completar un nivel, el juego envía una solicitud al servidor para registrar que el usuario ha finalizado dicho nivel en esa clase específica.

Para implementar esta funcionalidad en el servidor, se añadió una clase como una tabla de unión triple que incluye los identificadores de usuario, clase y nivel. De esta manera, la combinación de estos identificadores asegura que cada instancia sea única en todo momento.

Esto permite mostrar el progreso de los jugadores en las tarjetas dentro del juego, así como proporcionar a los profesores una visualización más clara a través de una tabla en la plataforma web, lo que facilita la comprensión de qué usuarios han completado cada nivel.

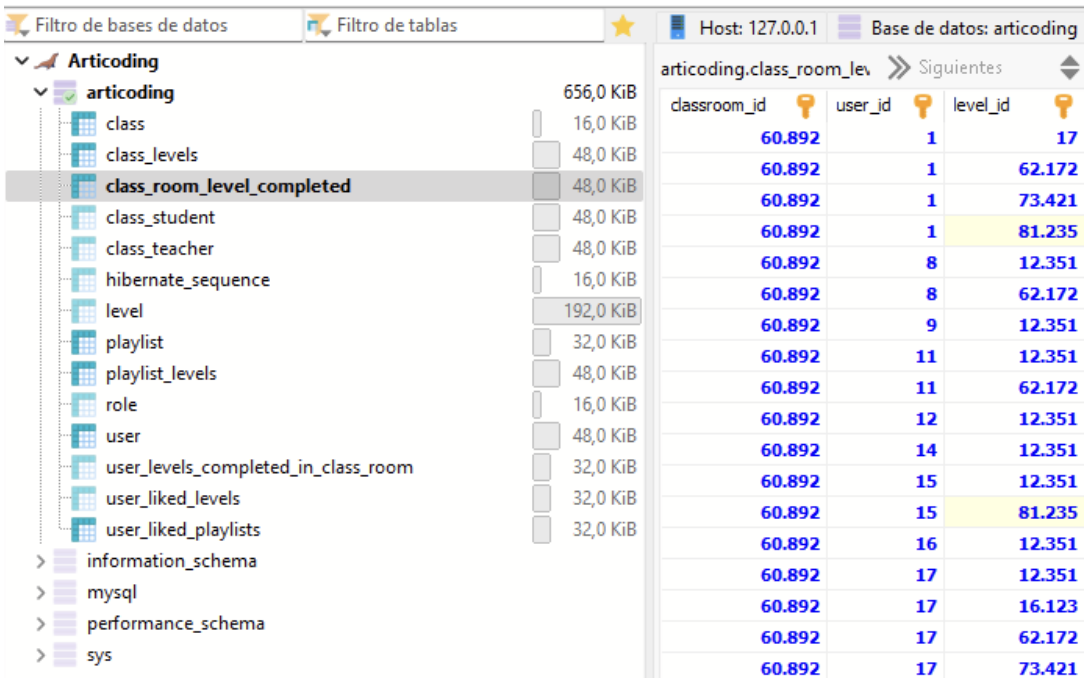

Figura 4-35. Captura de estado de la tabla de unión triple

```java
@Entity  11 usages  CesarCRP97
@IdClass(ClassRoomLevelCompleted.class)
public class ClassRoomLevelCompleted implements Serializable {

    @Id  3 usages
    @ManyToOne
    @JoinColumn(name = "user_id")
    private User user;

    @Id  3 usages
    @ManyToOne
    @JoinColumn(name = "classroom_id")
    private ClassRoom classRoom;

    @Id  3 usages
    @ManyToOne
    @JoinColumn(name = "level_id")
    private Level level;
```

Figura 4-36. Definición de la clase ClassRoomLevelCompleted

Por último, se añadió la vista de una tabla en la plataforma web para ver que el profesor pudiera ver el avance de cada alumno. Esta tabla contiene los nombres de los niveles que forman la clase en las columnas y los nombres de cada alumno en las filas.

Estado de la clase

Alumnos	TREX	Question	Creeper	Heart	Skull	Condiciones	Seta
Pepito	✓	✓	✗	✓	✗	✗	✗
Carlosroa	✗	✓	✗	✗	✗	✓	✗
Vilaviejo	✗	✗	✗	✗	✓	✗	✓

Figura 4-37. Tabla con el progreso de los alumnos en una clase

4.4.4.11 Funcionalidad de entrar a una clase por código

En el menú principal de la comunidad, se incluye un campo donde los usuarios pueden introducir códigos para unirse a una clase específica. Cuando se introduce un código válido, se notifica al usuario que ha sido añadido a la clase correspondiente, la cual luego se mostrará en la página de clases.

Al crear una clase, únicamente es posible hacerlo a través de la plataforma web. En este proceso, se genera automáticamente un código aleatorio compuesto por 7 caracteres alfanuméricos. Para visualizar este código, es necesario acceder a la plataforma web.

Figura 4-38. Opción de acceder a una clase

4.4.4.12 Funcionalidad de mensajes de aviso

Se implementaron mensajes de retroalimentación para informar a los usuarios sobre sus acciones dentro de la comunidad. Se optó por utilizar el color azul para los mensajes informativos y el color rojo para aquellos que indican un error.

Figura 4-39. Mensajes informativos de la comunidad

La lista de mensajes informativos incluye los siguientes eventos: inicio de sesión exitoso, registro exitoso, creación exitosa de una lista de reproducción, carga correcta de un nivel, adición exitosa a una clase, y la situación de no pertenecer a ninguna clase.

La lista de mensajes de error abarca los siguientes casos: inicio de sesión fallido, registro fallido, nombre de lista de reproducción demasiado corto, nombre de lista de reproducción demasiado largo, error al crear una lista de reproducción, campo de nombre de usuario vacío, campo de contraseña vacío, error al cargar el nivel, lista de reproducción vacía, clase no encontrada, servidor no encontrado, niveles no encontrados, listas no encontradas, contraseñas no coincidentes y carácteres no permitidos.

Capítulo 5 - Pruebas con usuarios

"La retroalimentación es el desayuno de los campeones." - Ken Blanchard

5.1 Estructura de las pruebas de usuarios

Con el fin de validar los cambios, identificar nuevos errores y recopilar feedback, se llevaron a cabo una serie de pruebas con usuarios reales.

La estructura de estas pruebas varió según el tiempo disponible y los objetivos específicos de cada prueba, pero todas incluyeron: una encuesta previa a la prueba para recoger datos sobre los intereses académicos y uso de dispositivos electrónicos; la interacción directa con el juego; y una encuesta tras probar el juego para pedir retroalimentación sobre la dificultad, los conceptos aprendidos y sugerencias.

Durante la interacción directa con el juego, el sistema de analíticas recopila datos sobre cada acción relevante realizada por los usuarios. Cada acción registrada incluye información sobre el usuario que la llevó a cabo, la acción específica realizada y el momento exacto en que ocurrió.

Además, en todas las pruebas se contó con la presencia de evaluadores, quienes estaban disponibles para asistir a los participantes que encontrarán dificultades.

5.1.1 Prueba interna 4ºESO + Empresa

A principios del mes de abril, en la Facultad de Informática de la UCM, tuvo lugar *4º ESO + Empresa,* una iniciativa financiada por la Comunidad de Madrid dirigida a alumnos que cursan 4º de Educación Secundaria Obligatoria en centros educativos sostenidos con fondos públicos [19]. Gracias a este programa se recibió un grupo de 15 estudiantes de 4º de la ESO que pudieron probar la versión de Articoding sin el rediseño de la comunidad. El objetivo de esta prueba fue comprobar el correcto funcionamiento del juego tras haber realizado la apropiación del código, las mejoras diversas, las optimizaciones y el rediseño del menú principal.

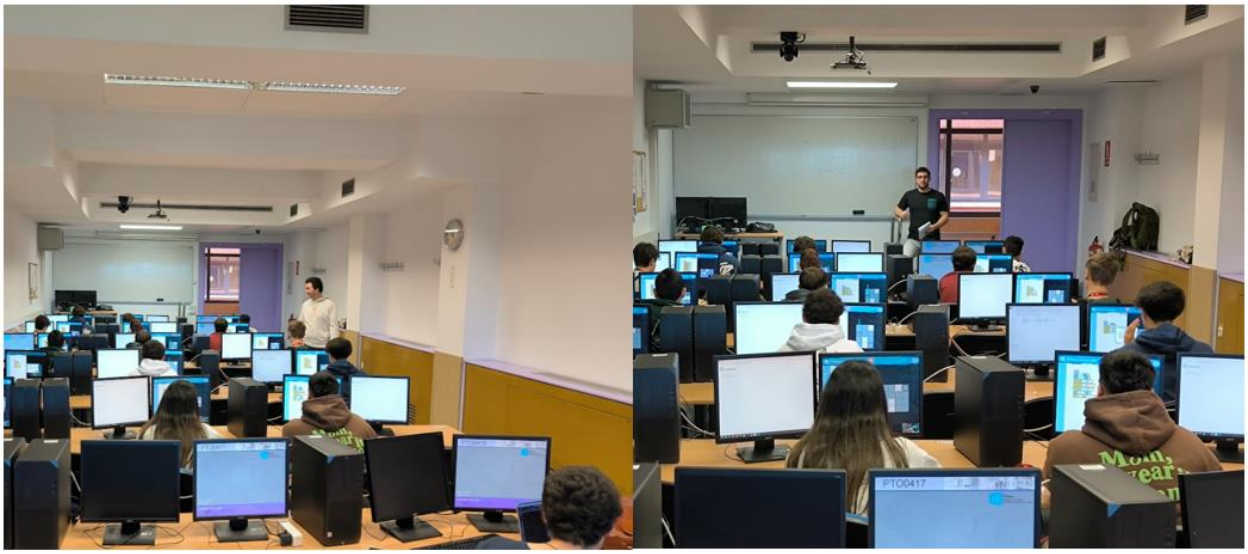

Figura 5-1. Fotos de la prueba de 4ºESO + Empresa

El objetivo se cumplió porque todo el mundo pudo jugar perfectamente sin notificar ninguna incidencia grave. Solamente se detectaron algunos errores visuales que fueron arreglados posteriormente. A pesar de no entrar dentro del objetivo de esta prueba, aprovechamos los datos recogidos mediante las encuestas y el *tracker* para realizar el siguiente análisis:

El análisis muestra la relación entre el tiempo jugado y el último nivel superado. Como se puede observar, todos los alumnos jugaron entre 80-90 minutos ya que fue el tiempo que duró la sesión, pero la diferencia entre los niveles alcanzados es muy grande.

En consecuencia, se han definido tres regiones que demarcan distintos niveles de rendimiento académico. La región roja identifica aquellos casos en los que los alumnos no han alcanzado los niveles de competencia esperados. La región amarilla representa el desempeño típico, donde los alumnos han alcanzado los estándares previstos. Por otro lado, la región verde destaca el excepcional rendimiento de los estudiantes que han superado ampliamente las expectativas establecidas.

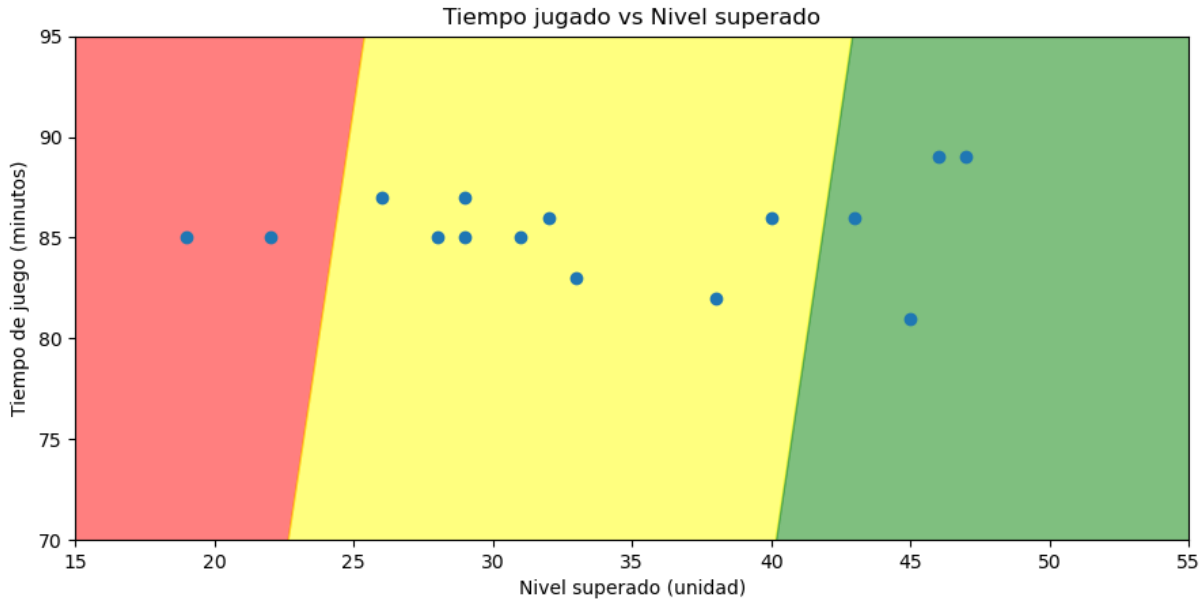

Figura 5-2. Gráfico con los resultados de las pruebas de 4ºESO + Empresa

En la gráfica se puede apreciar que todos los alumnos jugaron entre 80 y 90 minutos, dependiendo del tiempo que emplearon en realizar la encuesta previa. Asimismo, se observa que la mayoría de ellos alcanzaron niveles entre 25 y 40, lo cual era lo esperado. Cabe destacar que dos alumnos no lograron llegar a este rango, probablemente debido al desinterés que mostraron en el juego. Sin embargo, otros cuatro alumnos superaron las expectativas establecidas.

En cuanto a las encuestas realizadas, se observó que las asignaturas preferidas por la mayoría del grupo eran las matemáticas y la informática. Además, la mayoría de los alumnos mostraba un gran interés en el área de la programación, y las respuestas en general indican que la actividad les pareció interesante y a la par que útil.

5.1.2 Prueba externa 4ºESO + Empresa

Miguel Ángel Garrido Blázquez, docente del Departamento de Informática y Estadística de la Universidad Rey Juan Carlos, utilizó Articoding para realizar 2 sesiones en el mismo programa de 4ºESO + Empresa. Estas sesiones fueron conducidas de manera independiente, sin asistencia externa, con la realización de una encuesta previa antes del inicio de la actividad y otra al concluir. El objetivo de esta prueba fue verificar la

capacidad del juego para ser utilizado por un profesor externo al equipo de desarrollo sin experimentar ningún inconveniente, y adicionalmente, respaldar los resultados de la prueba previa.

Las pruebas transcurrieron sin problemas y todos los participantes pudieron disfrutar del juego sin experimentar ningún problema significativo durante las dos sesiones. Esto indica que el juego opera adecuadamente después de haber incorporado todas las mejoras, y su rediseño gráfico se percibe como práctico e intuitivo.

Una vez finalizadas las sesiones, Miguel elaboró un informe detallado para documentar los resultados y acontecimientos ocurridos durante las pruebas. De dicho informe podemos extraer los siguientes puntos clave.

El primer taller contó con 30 participantes y una duración de aproximadamente 3 horas con un descanso de 20 minutos. La actividad transcurrió sin incidencias técnicas relevantes. Los participantes mostraron un progreso adecuado en el juego, con el participante más avanzado alcanzando el nivel 6.9 y el menos avanzado llegando al nivel 2.7.

Los resultados de la encuesta previa proporcionaron una visión detallada del perfil de los participantes. Se constató un claro interés en la tecnología, con una inclinación notable hacia la creación de aplicaciones y programas. Además, se destacó la preferencia por áreas como la ingeniería y la arquitectura en términos de una futura formación académica. En la encuesta posterior, los estudiantes expresaron una valoración positiva general de la actividad, subrayando la utilidad de Articoding como herramienta para aprender a programar. Sin embargo, se hicieron sugerencias para mejorar la experiencia, como hacer el juego más entretenido y agregar más explicaciones y ejemplos visuales.

En la segunda sesión, participaron 26 estudiantes de los cuales la mayoría mostró receptividad. La dinámica del taller siguió una estructura similar a la primera sesión, con intervalos de juego y descanso.

Los resultados de la encuesta previa reflejaron nuevamente el interés en la tecnología, con una destacada preferencia por aprender a crear aplicaciones,

programas y videojuegos. En la encuesta posterior, se reiteró la valoración positiva de la actividad, resaltando nuevamente el aprendizaje de programación como punto fuerte. Se renovaron las sugerencias para mejorar la experiencia, centrándose en la necesidad de hacer el juego más entretenido y proporcionar más explicaciones visuales.

En resumen, el juego funcionó sin contratiempos y ambas pruebas se llevaron a cabo sin necesidad de asistencia externa de desarrolladores, lo que cumplió con el objetivo principal. Ambas sesiones de talleres de Articoding en la URJC fueron bien recibidas por los estudiantes, quienes valoraron positivamente la experiencia de aprendizaje. Los resultados de las encuestas proporcionaron información sobre los intereses y percepciones de los participantes, así como sugerencias útiles para mejorar futuras sesiones.

5.1.3 Prueba Colegio Arcipreste de Hita

A principios de mayo, se tuvo la oportunidad de probar el nuevo sistema de la comunidad en el Colegio Arcipreste de Hita de El Espinar, Segovia, con las 3 clases de 6º de educación primaria. El diseño experimental comprendía múltiples fases distribuidas a lo largo de un período de 90 minutos. Inicialmente, los participantes dedicarían 10 minutos al relleno de una encuesta previa. Posteriormente, durante la primera mitad de la sesión (40 minutos), participarían en el juego principal. Seguidamente, se les solicitaría registrarse en la comunidad y proporcionar un código específico para unirse a la clase correspondiente, donde deberían completar 5 niveles en un lapso de 30 minutos. Finalmente, se destinarían 10 minutos al rellenado de una encuesta posterior.

El propósito principal de esta prueba consistió en evaluar el funcionamiento del nuevo sistema de la comunidad, y el propósito secundario fue analizar el desempeño de los alumnos de 6º de primaria en el juego por primera vez.

Para la creación de la clase, debido a la ausencia de un profesor encargado de diseñar la batería de niveles, asumimos dicho rol y diseñamos los niveles nosotros mismos. Conscientes de la limitación de tiempo de los alumnos para avanzar en el juego, desarrollamos 5 niveles de dificultad mínima que pudieran completarse con los conocimientos adquiridos en la primera parte de la sesión. Asimismo, nos esforzamos por

incorporar diseños atractivos con el fin de mantener su interés. El resultado final fue el siguiente:

Figura 5-3. Niveles de las clases utilizadas para la prueba

Previamente a la prueba, se anticipó que sería imposible desplegar el servidor en la red local del colegio. Por consiguiente, se optó por asignar una IP elástica accesible a través de internet. Para utilizar esta dirección IP, se adquirió una instancia *E2* de *AWS* y se empleó para establecer un túnel *SSH* que redirigiera las solicitudes recibidas en la IP pública directamente al portátil en el que se había desplegado el servidor. Este enfoque permitiría administrar la base de datos durante la prueba en caso de surgir inconvenientes.

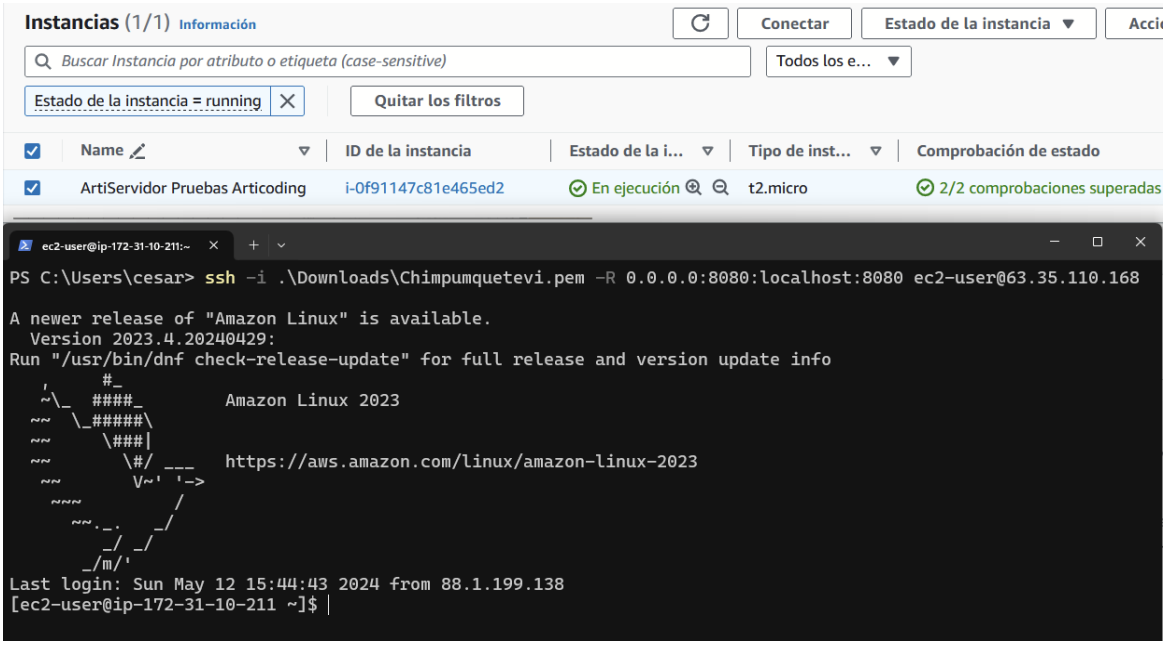

Figura 5-4. Información de la instancia E2 y despliegue del túnel ssh

La primera clase en realizar la prueba fue 6ºA, quienes se conectaron a la red wifi del colegio. Pudieron completar las encuestas y jugar al juego Articoding básico. Sin embargo, al intentar acceder a la parte de la comunidad, descubrimos que la red del colegio no les permitía conectarse al servidor del juego debido a restricciones. Dada la limitación de tiempo para resolver esta cuestión, les indicamos que continuaran jugando a los niveles normales.

La segunda clase en realizar la prueba fue 6ºC. Con el propósito de evitar el problema experimentado con la clase anterior, sugerimos que se conectaran a nuestras redes móviles en lugar de la red del instituto. Por consiguiente, les proporcionamos el nombre y la contraseña de cada red. Aunque inicialmente todo transcurrió sin contratiempos, la red de uno de los móviles colapsó debido al exceso de usuarios conectados (tenía 16 dispositivos conectados). Como resultado, al igual que la primera clase, solo pudieron acceder a la parte básica del juego.

La tercera clase en realizar la prueba fue 6ºB. La solución que implementamos fue solicitar a varios profesores de clases adyacentes que nos prestaran sus móviles para distribuir a los usuarios en diferentes puntos de acceso. De este modo, cada red únicamente tenía que soportar entre 4 y 5 usuarios, y todos pudieron conectarse al servidor del juego. A los alumnos se les requirió registrarse, iniciar sesión, modificar la imagen de usuario, unirse a la clase mediante una clave y completar todos los niveles disponibles en el orden deseado. Gracias a la nueva versión del sitio web, pudimos examinar los resultados:

Estado de la clase

Alumnos	TREX	Question	Heart	Skull	Creeper
SHAKIRA	✓	✓	X	X	X
chorizote	✓	X	X	X	X
waelinchielchevere	✓	✓	X	X	X
Lordenador	X	X	X	X	X
JOSEFA	X	X	X	X	X
manolo99	✓	X	X	X	X
mertupo	X	X	X	X	X
pitogordo	✓	X	X	X	X
MariaUnPajote	✓	✓	X	✓	✓
suna	✓	X	X	X	X
Gonzalo	X	X	X	X	X
AITANA	X	X	X	✓	X
lolaindigo	✓	X	X	X	X
maxneymar	X	X	✓	X	X
Milonista	✓	✓	X	X	X
karolg	X	X	X	X	X
pegate	X	X	X	X	X
manosrotas	X	X	X	X	X
chupamela	✓	X	✓	X	X
follarbbb	X	X	X	X	X

Figura 5-5. Tabla resultados prueba de la comunidad visible en el cliente

Con respecto al desempeño de los alumnos jugando a Articoding, observamos dos grupos distinguidos por sus niveles de comprensión. Por un lado, estaban aquellos alumnos que comprendieron el concepto de variables y progresaron en el juego sin dificultades. Por otro lado, algunos alumnos no lograron comprender el concepto de *variables* a pesar de nuestros intentos por explicárselo, lo que resultó en su estancamiento en el juego. Esta dificultad se manifestó a partir del nivel 2.2. Es relevante señalar que entre estos alumnos hubo un pequeño grupo que no logró avanzar hasta las variables debido a la dificultad percibida desde el inicio del juego, mientras que otros demostraron habilidades para alcanzar los niveles más avanzados.

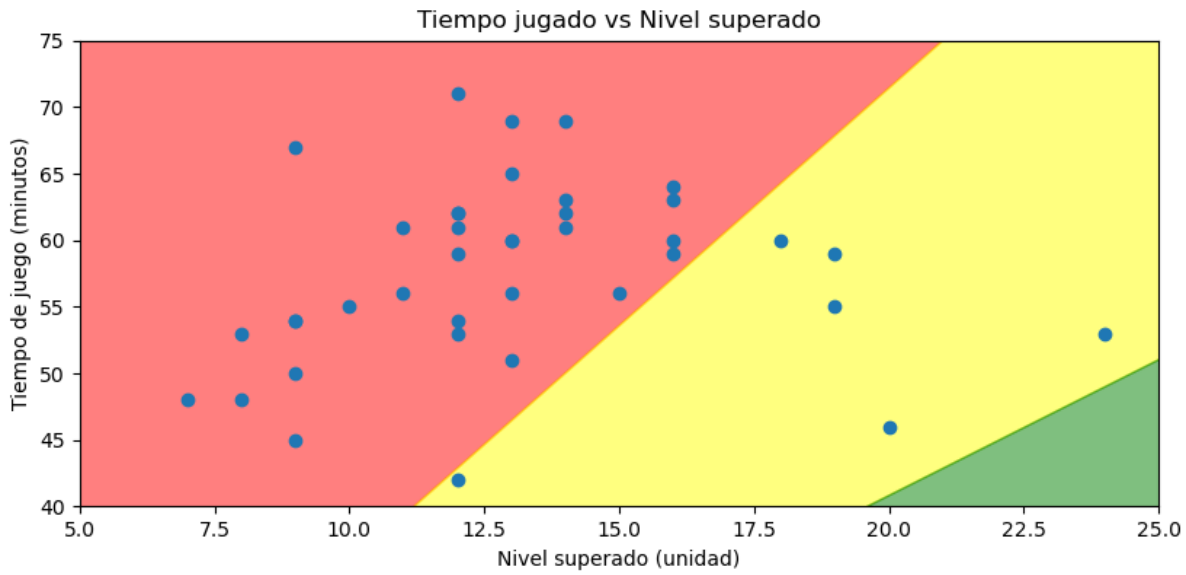

Figura 5-6. Gráfico con los resultados de las pruebas del colegio Arcipreste de Hita

Al contrastar los rangos de niveles estimados para los estudiantes de 4º de la ESO con los de primaria, como era previsible, se observa un peor desempeño de estos últimos. Resulta perceptible que la densidad más significativa de puntos se sitúa en el intervalo comprendido entre el nivel 10 y el nivel 16, coincidiendo con la introducción del concepto de variables. Es probable que esto se deba al hecho de que en primaria aún no se han introducido las ecuaciones ni el concepto de incógnita.

En relación a los resultados de las encuestas, se considera que no son especialmente relevantes debido a que la mayoría de las preguntas se centran en los campos en los que los participantes desean desarrollarse profesionalmente. Dado que se trata de niños de 11-12 años, estas preguntas pueden no tener mucho sentido en este momento. Sin embargo, lo único concluyente que se puede extraer de estas encuestas es que los niños disfrutaron mucho de la actividad, aunque expresaron que les resultó bastante difícil.

Figura 5-7. Fotos de la prueba del colegio Arcipreste de Hita

En resumen, el análisis del funcionamiento de la comunidad y la detección de errores resultaron exitosos, proporcionando el feedback necesario para evaluar el nuevo sistema en un entorno escolar y corregir las deficiencias identificadas. Asimismo, la evaluación de Articoding con alumnos de 11-12 años reflejó la dificultad que representa el concepto de variables para la mayoría de los estudiantes en este grupo de edad.

Capítulo 6 - Conclusiones y trabajo futuro

"No tengo nada que ofrecer, excepto sangre,

sudor, lágrimas y fatiga." - Winston Churchill

6.1 Conclusiones

Consideramos que el proyecto alcanzó satisfactoriamente los objetivos establecidos. A pesar de la abrumadora complejidad inicial al involucrarnos en un proyecto grande ya desarrollado y de nuevas, logramos comprender todas sus facetas mientras realizábamos ajustes para estandarizarlo en vista de futuras mejoras e implementaciones. Cuando abordamos la corrección de errores informados el año anterior, ya teníamos un entendimiento claro del funcionamiento del proyecto, lo que facilitó considerablemente el proceso.

Gracias a los análisis de varios juegos similares a Articoding, pudimos incorporar nuevas ideas que enriquecieron la experiencia de juego. El trabajo de rediseño de la comunidad nos obligó a adoptar un enfoque de trabajo iterativo e incremental, que incluyó la especificación de requisitos, la creación de prototipos y su posterior refinamiento. Durante la fase de implementación, realizamos mejoras tanto en el juego como en el servidor de forma simultánea, utilizando *Pair Programming* para integrar ambas partes de manera eficiente.

La realización de pruebas con usuarios reales ha sido una experiencia sumamente enriquecedora. Estas pruebas nos han brindado una valiosa perspectiva sobre cómo Articoding es percibido y utilizado por aquellos para quienes fue diseñada. A través de la interacción directa con los usuarios, hemos podido identificar tanto fortalezas como áreas de mejora en nuestro diseño y funcionalidad.

Ha sido un honor participar en este proyecto, en el cual numerosas personas han dedicado su esfuerzo y compromiso a lo largo de los últimos años. Nos complace haber tenido la oportunidad de contribuir con nuestro trabajo. Podemos afirmar con satisfacción que el juego ha experimentado mejoras significativas, tanto con sus menús más dinámicos como con mayores incentivos gracias a las nuevas animaciones y

medallas, lo que los motiva a esforzarse aún más. Además, el sistema de comunidad ha sido renovado con un enfoque fresco, marcando las bases para su expansión y mejora en los próximos años.

6.2 Trabajo futuro

A continuación se presentan algunas ideas que podrían contribuir significativamente a mejorar la experiencia del juego y su uso en futuros años:

- Desbloqueo de elementos de personalización: introducir elementos de personalización para el pingüino podría aumentar el interés de los usuarios por completar los niveles. Estos elementos podrían incluir colores, sombreros, trajes, entre otros.
- Incorporación de estilos temáticos: una mejora significativa podría consistir en variar la estética del nivel en función de la categoría a la que pertenece. Estas categorías podrían incluir diferentes tipos de climas dentro de la nave, como áreas desérticas, selváticas, árticas, entre otras.
- Corrección de errores: Es crucial abordar los errores identificados durante las pruebas, como la superposición de bloques, la nomenclatura de variables confusa y la generación de archivos sensibles a caracteres no permitidos en Windows al nombrar niveles.
- Mejora del menú de edición: Este aspecto del juego requiere un análisis exhaustivo y un rediseño radical que aproveche sus fortalezas y resuelva sus debilidades y errores.
- Consejos del pingüino: Durante el juego, el pingüino podría ofrecer consejos útiles a través de bocadillos. Estos consejos podrían ser del estilo: "Ponle nombres descriptivos a las variables", "¿Sabías que haciendo clic derecho sobre un bloque lo puedes duplicar?", "¡Ojo con dejar bloques por ahí tirados!¡Qué desorden!", etc.
- Etiquetado de niveles y listas: La adición de etiquetas a los niveles y listas permitiría tematizar los niveles y facilitar la búsqueda según los conceptos que se abordan en cada nivel.

- Renovación del diseño del servicio web: Tanto a nivel visual como funcional, se podría considerar la inclusión de nuevas características, como la capacidad para que los profesores gestionen las cuentas de los alumnos que ellos mismos creen, o la posibilidad de poner los niveles en modo privado.

Chapter 1 - Introduction

"Education is our passport to the future, for tomorrow belongs only to the people who prepare for it today." - Malcolm X

1. Motivation

Digitalization has had a profound impact on society. It has fundamentally altered how we live, work, communicate and engage with the world around us [1]. Due to this, the need arises to teach computational thinking [2] to equip people with the necessary skills to understand and actively participate in an increasingly digital world.

However, current educational systems remain focused on traditional teaching methods that prioritize memorization and repetition over the development of critical and practical skills. In this context, serious games are presented as a more effective tool for teaching computational thinking, since they combine practical learning and problem solving with an interactive and engaging experience, thus promoting student creativity and engagement. Articoding was born with this same idea of providing educational games to teach programming.

Articoding is a serious game designed to teach young people the basics of programming and improve computational thinking skills. It was developed in Unity in 2020 and has continued to be updated with new features since then. Additionally, it keeps its source open, allowing anyone to modify it and add new features freely.

This project provides the opportunity to contribute to the continued development of Articoding, as well as an opportunity to delve deeper into the theory and practice of serious games and their application in the classroom, exploring new educational methodologies and evaluating their impact on learning.

2. Goals

The main purpose of this final degree project is to expand the Articoding project, by improving its interface and expanding its open community, allowing students and

teachers to share levels created by themselves and improving the user experience. To achieve this objective, the following more specific sub-objectives have been established:

- Study and understand applications and games focused on the development of computational thinking and its community.
- Perform code appropriation.
- Identify and correct previously unaddressed bugs.
- Improve the user experience.
- Redesign the community system and apply the necessary changes to add improvements.
- Integrate server and database changes with the new game client.
- Validate changes with real user testing.

3. Work plan

Before starting development, with the aim of understanding and studying similar applications, an exhaustive analysis of various games will be carried out to identify effective interfaces and attractive mechanics.

First, the current state of the project will be analyzed and standardized for future improvements and implementations, for which a code appropriation will be carried out.

After this, to identify and correct errors, the reports generated from the user tests carried out the previous year will be reviewed.

In order to improve the user experience, modifications will be made to the interface, an expanded in-game feedback system will be included, and additional animations will be implemented.

To redesign the community system, the new idea will be presented first, followed by design and prototyping to verify its robustness [3]. It will then be deployed to the game, integrating the necessary server and database changes. Finally, the game's web service will be adapted to meet these new needs. Additionally, we will try to carry out tests with real users to validate the changes, detect new errors and collect feedback.

In relation to the tools that will be used, Google Drive and Gmail will be used for document management and communication, respectively. GitHub will be used for version control, creating a copy of the current repositories. Unity will remain as the video game engine, along with Visual Studio as the code editor. To create Assets, Photoshop will be used for image editing and Blender for creating animations. Regarding the development of the server, Intellij will be used as the main IDE, HeidiSQL for database manipulation, and Postman for testing HTTP requests. For client development, Visual Studio Code will be used. The interface designs will initially be created in paper and pencil, and later prototypes will be developed with Figma.

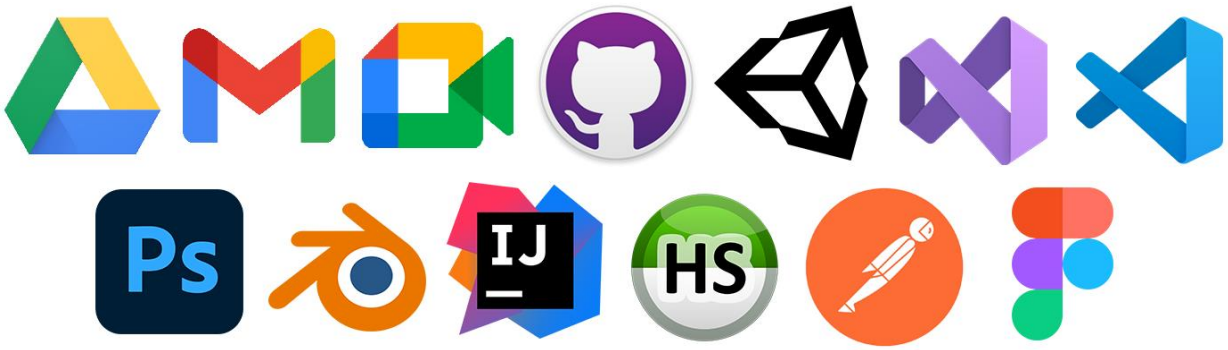

Figure 1-1. Programs used during the development of this project

In order to monitor progress and receive feedback on the progress made, periodic meetings will be held in person or through Google Meet. During these meetings, progress since the previous meeting will be presented and the next steps to follow will be defined. A more detailed description of these meetings is provided in Appendix D.

Chapter 6 - Conclusions and future work

"I have nothing to offer but blood, toil, tears and sweat."

Winston Churchill

1. Conclusions

We consider that the project satisfactorily achieved the established objectives. Despite the overwhelming initial complexity of getting involved in a large project already developed and new, we managed to understand all its facets while making adjustments to standardize it in view of future improvements and implementations. By the time we tackled the bug fixes reported the previous year, we already had a clear understanding of how the project worked, which made the process considerably easier.

Thanks to the analysis of several games similar to Articoding, we were able to incorporate new ideas that enriched the game experience. The community redesign work forced us to adopt an iterative and incremental work approach, which included requirements specification, prototyping and subsequent refinement. During the implementation phase, we made improvements to both the game and the server simultaneously, using Pair Programming to integrate both parts efficiently.

Testing with real users has been an extremely enriching experience. These tests have given us valuable insight into how Articoding is perceived and used by those for whom it was designed. Through direct interaction with users, we have been able to identify both strengths and areas for improvement in our design and functionality.

It has been an honor to participate in this project, to which numerous people have dedicated their efforts and commitment over the past few years. We are pleased to have had the opportunity to contribute our work. We can say with satisfaction that the game has undergone significant improvements, both with its more dynamic menus and with greater incentives thanks to the new animations and medals, which motivates them to work even harder. In addition, the community system has been revamped with a fresh approach, laying the groundwork for its expansion and improvement in the coming years.

2. Future Work

Here are some ideas that could significantly contribute to improving the game experience and its usability in future years:

- Unlocking customization elements: introducing customization elements for the penguin could increase users' interest in completing levels. These elements could include colors, hats, costumes, among others.
- Incorporation of thematic styles: a significant improvement could be to vary the aesthetics of the level depending on the category to which it belongs. These categories would include different types of climates within the ship, such as desert, jungle, arctic areas, among others.
- Bug fixes: It is crucial to address bugs identified during testing, such as block overlapping, confusing variable nomenclature, and generating files sensitive to characters not allowed in Windows when naming levels.
- Improved editing menu: This aspect of the game requires a thorough analysis and radical redesign that takes advantage of its strengths and addresses its weaknesses and bugs.
- Penguin tips: During the game, the penguin could offer useful tips through speech bubbles. These tips could be along the lines of: "Give descriptive names to variables", "Did you know that by right-clicking on a block you can duplicate it?", "Beware of leaving blocks lying around, what a mess!", etc.
- Labeling of levels and lists: The addition of labels to the levels and lists would make it possible to thematize the levels and facilitate the search according to the concepts addressed in each level.
- Renewal of the web service design: Both visually and functionally, the inclusion of new features could be considered, such as the ability for teachers to manage student accounts that they create themselves, or the ability to set levels to private mode.

CONTRIBUCIONES PERSONALES

Al ser 2 alumnos de distintos grados, se optó por dividir el trabajo en función de sus conocimientos y experiencias previas.

Óscar, quien está cursando Desarrollo de Videojuegos, se enfocó en los nuevos diseños y en el desarrollo interno del juego.

César, estudiante de Ingeniería Informática, se centró en el desarrollo del servidor y la aplicación web.

Para integrar las partes donde el juego se conecta con el servidor, se decidió utilizar la técnica de *Pair Programming*.

Además, si bien cada uno de nosotros redactó individualmente ciertas secciones de esta memoria, es importante destacar que la mayor parte del trabajo se llevó a cabo de manera colaborativa durante nuestras sesiones de videollamadas conjuntas.

Óscar Fernández Romano

- Creación del repositorio Articoding23/24.
- Investigación del juego Geometry Dash.
- Investigación del juego Rabbids Coding.
- Apropiación de Articoding.
- Creación de la pestaña de *Utils* en el editor de Unity.
- Mejora del sistema de guardado.
- Actualización del sistema de localización.
- Corrección de textos con errores.
- Implementación de animaciones del pingüino.
- Eliminación de carpetas y activos obsoletos.
- Desarrollo de un comando para iniciar automáticamente el juego.
- Reescalado de imágenes mediante la IA *Upscayl* mejorando su resolución.
- Creación de nuevas imágenes mediante Photoshop.
- Mejora de la organización dentro de los Assets del juego.
- Rediseño del interfaz del menú principal.
- Implementación del nuevo diseño del menú principal.
- Integración del paquete *DOTween*.
- Adición de animaciones en el menú principal.
- Actualización de la página de selección de nivel.
- Implementación de guardado de imagen previa del nivel creado.
- Creación de las nuevas tarjetas para los niveles creados.
- Corrección de pequeños errores en la creación de niveles.
- Actualización de la página de perfil del usuario.
- Incorporación del sistema de medallas.
- Actualización de la página de temario.
- Replanteamiento de la comunidad.

- Rediseño del interfaz de la comunidad.
- Realización de las pruebas internas de 4ºESO + Empresa.
- Análisis de los resultados de las pruebas internas 4ºESO + Empresa.
- Implementación de la página de inicio de sesión y registro de usuarios.
- Desarrollo de la pestaña principal de la comunidad.
- Personalización de perfiles de usuario mediante la selección de iconos.
- Implementación de la página de niveles de la comunidad.
- Integración de la pestaña para filtrar niveles.
- Desarrollo de las funcionalidades para ordenar niveles.
- Implementación del sistema de guardado de niveles como favoritos.
- Creación de las tarjetas de nivel de la comunidad.
- Implementación de la página de subir niveles de la comunidad.
- Implementación de la página de listas de la comunidad.
- Integración de la pestaña para filtrar listas.
- Desarrollo de las funcionalidades para ordenar listas.
- Implementación del sistema de guardado de listas como favoritas.
- Creación de las tarjetas de lista de la comunidad.
- Implementación de la página de crear listas de la comunidad.
- Integración de la clave de acceso para unirse a clases
- Implementación de la página de clases de la comunidad.
- Creación de las tarjetas de clases de la comunidad.
- Adición de mensajes informativos.
- Implementación de una animación de carga.
- Integración de la opción de habilitar/deshabilitar el modo online desde el archivo de configuración.
- Actualización de la escena de créditos.
- Análisis de los resultados de las pruebas externas 4ºESO + Empresa.

- Creación de un script de python que analiza los datos recogidos por el *tracker*.
- Realización de las pruebas en colegio Arcipreste de Hita.
- Análisis de los resultados de las pruebas en colegio Arcipreste de Hita.
- Corrección de errores encontrados en las pruebas.
- Publicación continua de builds estables.
- Redacción de esta memoria.

César Carlos Rubio Pastor

- Creación del repositorio ArticodingServer.
- Creación del repositorio ArticodingClient.
- Investigación del juego I Wanna Maker.
- Investigación del juego Super Mario Maker 2.
- Investigación de la plataforma Scratch.
- Investigación de la plataforma MakeCode Arcade.
- Apropiación de ArticodingServer.
- Apropiación de ArticodingClient.
- Actualización de paquetes Maven en ArticodingServer.
- Eliminación de archivos compilados de ArticodingServer.
- Actualización del fichero README.md de ArticodingServer.
- Eliminación de comentarios y valoraciones del servidor.
- Traducción de mensajes de fallo y comentarios de ArticodingServer.
- Rediseño del interfaz del menú principal.
- Refactorización de *LevelService*.
- Testeo a fondo del juego.
- Replanteamiento de la comunidad.
- Rediseño del interfaz de la comunidad.
- Implementación del guardado de imágenes en el servidor.
- Creación de un nuevo DTO *LevelWithImageDTO*.
- Actualización de las clases existentes para que cumplieran con los nuevos requisitos de la comunidad.
- Realización de las pruebas internas de 4ºESO + Empresa.
- Análisis de los resultados de las pruebas internas 4ºESO + Empresa.
- Implementación del registro de los niveles que ha dado *like* el jugador.
- Refactorización del filtrado de niveles.

- Creación de clases *Comparator* que ordena los niveles según distintos criterios.
- Creación del componente *Playlist*, con sus respectivos *service, repository, controller y DTO*.
- Adición de documento descriptivo al repositorio de ArticodingServer.
- Generación automática de claves únicas para las clases.
- Adición de funcionalidad de guardado de imagen de perfil de la comunidad.
- Desarrollo de tabla de unión triple *ClassRoomLevelCompleted* en el servidor.
- Adición de la tabla de resultados a las clases en la plataforma web.
- Mejora del sistema para añadir niveles, profesores y alumnos en la plataforma web.
- Despliegue de la *E2 instance* de *AWS* en las pruebas del colegio Arcipreste de Hita.
- Realización de las pruebas en colegio Arcipreste de Hita.
- Análisis de los resultados de las pruebas en colegio Arcipreste de Hita.
- Redacción de esta memoria.

BIBLIOGRAFÍA

[1] Digitalization and development | Digital Development Observatory. https://desarrollodigital.cepal.org/en/digitalization-development

[2] Aho, A. V. (2011). Ubiquity symposium: Computation and Computational Thinking. Ubiquity, 2011(January). https://doi.org/10.1145/1922681.1922682

[3] Matveeva, E. (2022). UNIVERSITAT POLITÈCNICA DE VALÈNCIA Facultad de Bellas Artes. https://riunet.upv.es/handle/10251/185729

[4] Salas Torres, M. I. (2010). La enseñanza tradicional de las ciencias versus las nuevas tendencias educativas. Revista Electrónica Educare, 14(1), 131–142. https://doi.org/10.15359/REE.14-1.11

[5] Ayman, R., Sharaf, N., Ahmed, G., Abdennadher, S. (2018). MiniColon; Teaching Kids Computational Thinking Using an Interactive Serious Game. In: Göbel, S., et al. Serious Games. JCSG 2018. Lecture Notes in Computer Science(), vol 11243. Springer, Cham. https://doi.org/10.1007/978-3-030-02762-9_9

[6] RobTop Games. https://www.robtopgames.com/

[7] I Wanna Maker. https://www.iwannamakergame.com/

[8] IWBTG! A Very Hard Game About a Boy and 8-bit Masochism! https://iwbtg.kayin.moe/

[9] MakeCode Arcade. https://arcade.makecode.com/

[10] Rabbids Coding.

https://store.ubisoft.com/es/rabbids-coding/5d96f9b05cdf9a2eacdf68cb.html

[11] Scratch - Imagine, Program, Share. https://scratch.mit.edu/

[12] Inicio - Super Mario Maker 2TM para la consola Nintendo SwitchTM - Sitio oficial. https://supermariomaker.nintendo.com/es/

[13] Articoding – eUCM. https://www.e-ucm.es/es/portfolio-item/articoding/

[14] Faouaz Santillana, D., García Cárdenas, A., & Poyatos Morate, Á. (2021). Juegos Serios para Promover el Pensamiento Computacional y la Programación. http://hdl.handle.net/20.500.14352/10444

[15] Duarte Balvís, T., Martín Sánchez, A., & Martínez Martínez, P. (2022). Uso de juegos serios para mejorar el aprendizaje de la programación en la escuela. http://hdl.handle.net/20.500.14352/3240

[16] Martín Domínguez, H., & Cidoncha Pérez, G. (2023). Videojuegos y Programación: Evaluación de Conocimiento sobre los Conceptos de Programación en Secundaria a través de un Videojuego.

[17] Bucher, N. (2017). Introducing design patterns and best practices in unity. Proceedings of the SouthEast Conference, ACMSE 2017, 243–247. https://doi.org/10.1145/3077286.3077322

[18] Figma. https://www.figma.com/

[19] Programa 4o ESO+Empresa | Comunidad de Madrid. https://www.comunidad.madrid/servicios/educacion/programa-4o-esoempresa

APÉNDICES

Apéndice A - Repositorios

- Articoding

  https://github.com/OskarFreestyle/Articoding23-24

- Servidor Articoding

  https://github.com/CesarCRP97/articodingserver

- Cliente Web Articoding

  https://github.com/CesarCRP97/articodingclient

Apéndice B - Backup de la base de datos

Se ha generado un *backup de una base de datos* que incluye varios niveles, listas y clases, con el fin de que los futuros desarrolladores puedan emplearla y evaluar la comunidad sin necesidad de invertir esfuerzos en la creación de estos elementos. Dicha copia de seguridad se encuentra almacenada en el repositorio del servidor, específicamente en la carpeta denominada *databaseTemplate*:

https://github.com/CesarCRP97/articodingserver/blob/master/databaseTemplate/basicBackup.sql

Apéndice C - Script de análisis de los datos json

Se ha incorporado un script de Python diseñado para leer los datos del archivo JSON generado por el tracker, facilitando así su análisis. Actualmente, el script recopila información sobre el tiempo de juego de cada jugador y el número de niveles completados, presentándolos en forma de tabla. No obstante, es fácilmente adaptable para utilizar cualquier otro dato contenido en el archivo JSON. Este script se localiza en el repositorio del juego, dentro de la carpeta denominada *AnálisisTracker*:

https://github.com/OskarFreestyle/Articoding23-24/tree/main/AnalisisTracker

Apéndice D - Reuniones

Con el propósito de monitorear el progreso y obtener retroalimentación sobre los avances realizados, se llevaron a cabo reuniones periódicas tanto presenciales como a través de *Google Meet*. Durante estas reuniones, se presentaban los avances desde el encuentro anterior y se definían los próximos pasos a seguir. En el siguiente calendario, se resaltan los días en los que se llevaron a cabo estas reuniones para destacar el trabajo continuo en el proyecto a lo largo de todo el curso.

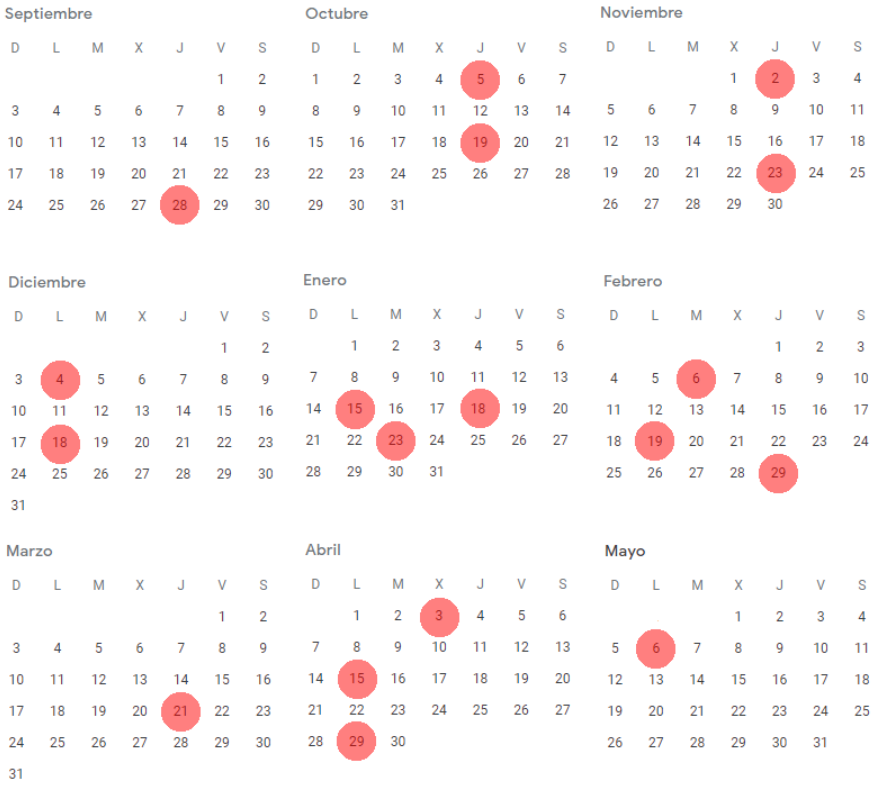

Figura 6-1. Calendario con las reuniones realizadas a lo largo del proyecto