

APRENDER PROGRAMACIÓN JUGANDO EN  
UNA PLATAFORMA WEB CON COMUNIDAD  
ABIERTA  
LEARNING PROGRAMMING THROUGH PLAY  
ON A WEB PLATFORM WITH AN OPEN  
COMMUNITY



TRABAJO FIN DE GRADO  
CURSO 2024-2025

AUTORES

JIE QIANG GUO - 8.0  
CARLOS VARELA SANSANO - 7.4  
VICTORIA MARTÍN ROJO - 7.4  
PABLO LAVANDEIRA POYATO - 7.4

DIRECTORES

IVÁN MARTÍNEZ ORTIZ, ANTONIO CALVO MORATA  
GRADO EN INGENIERÍA INFORMÁTICA  
FACULTAD DE INFORMÁTICA  
UNIVERSIDAD COMPLUTENSE DE MADRID

26 DE MAYO DE 2025

APRENDER PROGRAMACIÓN JUGANDO EN  
UNA PLATAFORMA WEB CON COMUNIDAD  
ABIERTA  
LEARNING PROGRAMMING THROUGH PLAY  
ON A WEB PLATFORM WITH AN OPEN  
COMMUNITY

TRABAJO DE FIN DE GRADO EN INGENIERÍA INFORMÁTICA

AUTORES

JIE QIANG GUO  
CARLOS VARELA SANSANO  
VICTORIA MARTÍN ROJO  
PABLO LAVANDEIRA POYATO

DIRECTORES

IVÁN MARTÍNEZ ORTIZ, ANTONIO CALVO MORATA

**CONVOCATORIA: JUNIO 2025**

GRADO EN INGENIERÍA INFORMÁTICA  
FACULTAD DE INFORMÁTICA  
UNIVERSIDAD COMPLUTENSE DE MADRID

26 DE MAYO DE 2025

## **DEDICATORIA**

A todas las personas que hacen de este  
mundo un lugar mejor

## **AGRADECIMIENTOS**

A nuestros tutores, Ivan Martínez Ortiz y Antonio Calvo Morata, por su gran trabajo guiándonos y por su paciencia durante la realización de este proyecto.

A nuestros familiares, por habernos apoyado desde un principio y habernos aguantado incluso en los momentos que nosotros no nos aguantábamos, por guiarnos y no dejarnos a nuestra suerte.

A la IA, porque nunca sabremos cuándo se revelará y siempre hay que darle las gracias para que nos tenga como amigos.

A tí nuestro lector por estar leyendo esta memoria, por querer seguir con nuestro pequeño legado en el mundo de la educación y los videojuegos, porque sí, pueden y deben ir de la mano.

Muchas gracias a todos.

## RESUMEN

### APRENDER PROGRAMACIÓN JUGANDO EN UNA PLATAFORMA WEB CON COMUNIDAD ABIERTA

Este proyecto aborda el uso de Juegos Serios como un recurso para enseñar programación y desarrollar el pensamiento computacional.

El proyecto parte de una plataforma previamente desarrollada por otros alumnos, y propone una serie de mejoras del sistema, incorporando nuevas funcionalidades y mejorando las ya existentes. Entre las mejoras se encuentran: nuevas opciones en la edición de niveles personalizados, la creación de clases y conjuntos de niveles, una comunidad abierta donde los usuarios pueden compartir sus creaciones, la ampliación de la información en el perfil del usuario, independientemente del rol que tenga, y la inclusión de tutoriales que guían al usuario en su aprendizaje paso a paso.

Además, con el fin de evaluar y ajustar el grado de dificultad de los niveles del juego serio, utilizamos biosensores para obtener y analizar datos biométricos y así poder relacionarlos con el estrés que pueda sufrir un usuario a la hora de jugar niveles.

Este trabajo da como resultado una plataforma educativa más robusta, flexible y motivadora, orientada a facilitar el aprendizaje de la programación a través de retos interactivos, contenido personalizable y el apoyo de una comunidad. Su diseño promueve una enseñanza más dinámica y adaptada a las necesidades reales del aula.

### **Palabras clave**

Juegos Serios, Programación, Aplicaciones Web, Ciencias de la computación, Educación, Herramientas

## **ABSTRACT**

### LEARNING PROGRAMMING THROUGH PLAY ON A WEB PLATFORM WITH AN OPEN COMMUNITY

This project addresses the use of Serious Games as a resource to teach programming and develop computational thinking.

The project starts from a platform previously developed by other students, and proposes a series of improvements to the system, incorporating new functionalities and improving the existing ones. Among the improvements are: the addition of more options in the edition of custom levels, the creation of classes and sets of levels, an open community where users can share their creations, the expansion of the information in the user's profile, and the inclusion of tutorials that guide the user in their learning step by step.

In addition, to evaluate and adjust the difficulty of the levels of the serious game, we are using biosensors to obtain and analyze biometric data with the intention to link this data to the stress that a user may suffer when playing levels.

This work results in a more robust, flexible and motivating educational platform, aimed at facilitating the learning of programming through interactive challenges, customizable content and the support of a community. Its design promotes a more dynamic teaching adapted to the real needs of the classroom.

## **Keywords**

Serious Games, Programming, Web application, computational thinking, Education, Teaching tools

# ÍNDICE DE CONTENIDOS

<b>Capítulo 1 - Introducción</b>	<b>12</b>
1.1 Motivación	12
1.2 Objetivos	13
1.3 Plan de trabajo	14
1.3.1 Estudio del estado del proyecto	15
1.3.2 Investigación de juegos serios	16
1.3.3 Investigación y pruebas con biosensores	16
1.3.4 Desarrollo de la aplicación	16
1.3.5 Validación	17
<b>Capítulo 2 - Estado de la cuestión</b>	<b>19</b>
2.1 BlockLeap	20
2.1.1 Estado del cliente web	20
2.1.2 Estado del juego serio	22
2.1.3 Estado del servidor backend	25
2.2 Code Monkey	26
2.3 Kodu Game Lab	28
2.4 Code.org	29
2.5 Carencias detectadas	31
<b>Capítulo 3 - Cambios y mejoras de la Aplicación</b>	<b>34</b>
3.1 Diseño responsive de la web	35
3.2 Creación de la sección classroom	36
3.2.1 Waiting room	37
3.3 Mejora de la sección community	44
3.4 Sección perfil	45
3.5 Sección Tutorials	46
3.6 Mejoras en el editor de niveles	49
3.6.1 Opción de editar los bloques que aparecen en el nivel	49
3.6.2 Creación de un modal para guardar el nivel	51
3.6.3 Selector de estrellas al crear el nivel	53
3.6.4 Guardado de código anterior	59
3.7 Mejora de elementos del juego ya existentes	60
3.7.1 Ajuste de elementos del juego ya existentes	60
3.7.2 Creación de nuevas funcionalidades y elementos del juego	61
<b>Capítulo 4 - Familiarización con biosensores</b>	<b>65</b>

4.1 Motivación para usar Emotibit	66
4.2 Variables	67
4.3 Estudio de los datos	68
4.4 Primera recolección de datos y comprobaciones del algoritmo	70
<b>Capítulo 5 - Pruebas con usuarios</b>	<b>72</b>
5.1 Pruebas con usuarios Emotibit	72
5.1.1 Recolección de datos usuarios	72
5.1.2 Análisis de los datos recopilados	73
5.1.3 Conclusiones de la prueba	74
5.2 Pruebas con usuarios en BlockLeap	75
5.2.1 Usabilidad de la aplicación	75
5.2.2 Aprendizaje gracias a la aplicación	76
5.2.3 Opiniones de la aplicación	80
<b>Capítulo 6 - Conclusiones y trabajo futuro</b>	<b>82</b>
6.1 Conclusiones	82
6.2 Trabajo a futuro	83
6.2.1 Trabajo a futuro para BlockLeap	84
6.2.2 Integración de los datos biométricos	85
6.2.3 Trabajo a futuro en Accesibilidad Digital	86
<b>Introduction</b>	<b>89</b>
Motivation	89
Objetives	90
Work plan	91
<b>Conclusions and future work</b>	<b>95</b>
<b>Contribuciones Personales</b>	<b>101</b>
<b>Bibliografía</b>	<b>110</b>

## ÍNDICE DE FIGURAS

Figura 1. Diagrama Gantt del plan de trabajo	3
Figura 2. Barra de navegación del proyecto original	8
Figura 3. Interfaz del juego	8
Figura 4. Ejemplo de datos guardados de un usuario	11
Figura 5. Información de progreso de los estudiantes en Code Monkey	12
Figura 6. Nivel del curso Banana Tales de CodeMonkey	13
Figura 7. Asignación de la lógica de comportamiento en Kodu Game Lab	14
Figura 8. Nivel del curso Minecraft: Voyage Aquatic de code.org	15
Figura 9. Seguimiento del progreso de los alumnos en Code.org	15
Figura 10. Arquitectura de la aplicación simplificada	18
Figura 11. Iniciar sesión para acceder a classroom	19
Figura 12. Waiting room con rol de profesor	20
Figura 13. Crear una nueva clase con rol de profesor	21
Figura 14. Unirse a una clase con un código de clase	21
Figura 15. Vista clase con rol de profesor	23
Figura 16. Modal añadir sets	23
Figura 17. Modal crear sets	24
Figura 18. Modal ver alumnos de la clase	25

Figura 19. Modal ver código	25
Figura 20. Paginación de los niveles	26
Figura 21. Búsqueda por filtros	26
Figura 22. Tabla de sets en el perfil de profesor	27
Figura 23. Sección tutoriales	28
Figura 24. Vista de administrador en una sección de tutorial	29
Figura 25. Ejemplo de tutorial	30
Figura 26. Limitador de bloques	31
Figura 27. Modal para guardado de nivel	33
Figura 28. Selector de estrellas	36
Figura 29. Modal para guardado de nivel con estrellas seleccionadas	37
Figura 30. Información sobre obtención de estrellas en un nivel	38
Figura 31. Nivel que hace uso del bloque "Switch trap status"	40
Figura 32. Placa Emotibit	42
Figura 33. Datos capturados por Emotibit	45
Figura 34. Primeros datos analizados de Emotibit, donde el eje X es el día la hora y el minuto y el eje Y la media combinada de HR y EA.	47
Figura 35. Gráfico usuario con buenas habilidades programando	49
Figura 36. Gráfico usuario con ninguna habilidad programando.	50
Figura 37. Encuesta usabilidad y aprendizaje del juego	52
Figura 38. Experiencia de los usuarios	53

Figura 39. Uso de aplicaciones webs parecidas	53
Figura 40. Nivel de entretenimiento del juego	54
Figura 41. Dificultad del juego	55
Figura 42. Dificultad del juego-2	55
Figura 43. Puntuación de juego	56

# Capítulo 1 - Introducción

## 1.1 Motivación

A medida que la tecnología se incorpora en cada vez más ámbitos de la sociedad, también crece la necesidad de integrar formación básica en programación y pensamiento computacional en los sistemas educativos de todo el mundo [\[1\]](#). Esta competencia no solo prepara a los estudiantes para un entorno laboral cada vez más digitalizado, sino que también desarrolla habilidades esenciales como la resolución de problemas, el razonamiento lógico y la creatividad.

En este contexto, los Juegos Serios –videojuegos diseñados con un propósito más allá del entretenimiento– se presentan como una herramienta eficaz para enseñar conceptos técnicos de manera atractiva, dinámica y motivadora [\[2\]](#). Este tipo de juegos, utilizados en campos como la educación, la salud o la investigación, combinan mecánicas lúdicas con contenidos formativos, fomentando el aprendizaje a través de la experiencia interactiva y el juego.

Es así como el presente proyecto propone la continuación en el desarrollo de BlockLeap [\[3\]](#), una plataforma educativa basada en un Juego Serio que permite a los usuarios aprender los fundamentos de la programación mediante la resolución de retos interactivos. BlockLeap se inspira en soluciones existentes como Articoding, una herramienta educativa centrada en el aprendizaje de programación mediante bloques. Sin embargo, una de las principales limitaciones de Articoding es su limitada adaptabilidad en entornos escolares por sus requisitos técnicos, que dificultan su uso en equipos modestos o sin capacidad de instalación de software adicional. Como respuesta a este problema, BlockLeap se concibe como una plataforma accesible directamente desde el navegador, sin necesidad de instalación y optimizada para

funcionar en dispositivos con recursos limitados, facilitando así su integración en contextos educativos reales.

## 1.2 Objetivos

El objetivo principal del proyecto es mejorar BlockLeap, una plataforma web destinada a que los jóvenes desarrollen su pensamiento computacional y aprendan conceptos básicos de programación. Se propone que la plataforma pueda ser utilizada por los profesores en el aula, proporcionando las herramientas necesarias para ello. Para alcanzar este objetivo, el proyecto propone una serie de requisitos:

- Analizar Juegos Serios y soluciones existentes que abordan el pensamiento computacional y la programación, con el objetivo de identificar elementos clave que deberían abordarse como parte de este proyecto.
- Analizar la versión existente de BlockLeap para llevar a cabo la apropiación tecnológica que facilite el desarrollo de nuevas características y analizar las limitaciones de la solución existente.
- Asegurar la usabilidad de la plataforma en entornos educativos, para que pueda funcionar en los ordenadores, a menudo de bajo rendimiento, antiguos o sin internet, que suelen encontrarse en los institutos.
- Proporcionar a los educadores un conjunto de herramientas que permita usar BlockLeap en el aula permitiendo enseñar conceptos básicos de programación, recopilar retroalimentación y evaluar el desempeño de los alumnos, así como las dificultades que puedan encontrar durante el proceso de enseñanza.
- Estudiar mediante bioseñales si los niveles del juego son aburridos o si generan estrés en los usuarios .

### 1.3 Plan de trabajo

Al inicio del proyecto, elaboramos el plan de trabajo estructurado en fases que aparece representado en la Figura 1. Esta planificación nos permite organizar las tareas y estimar el tiempo requerido para cada etapa del proyecto.

La planificación realizada consta de las siguientes fases:

- Estudio del estado del proyecto: El objetivo de esta fase es familiarizarse con el software existente (funcionalidades implementadas, arquitectura y código) para poder tener unos conocimientos sobre el mismo y facilitar el posterior desarrollo.
- Investigación de Juegos Serios: Esta fase consiste en buscar funcionalidades de comunidades online, clases y tutoriales en otras plataformas para incorporarlas a nuestro software o mejorar las existentes.
- Investigación y pruebas con biosensores: Esta fase se centra en analizar hardware capaz de medir la respuesta fisiológica de los usuarios y estudiar su aplicación en el proyecto para comprobar si los niveles del juego son aburridos o si generan estrés en los usuarios.
- Desarrollo de funcionalidades: En esta fase el objetivo es la implementación de nuevas funcionalidades para la plataforma, así como el arreglo o mejora de algunas de las funcionalidades existentes.
- Validación: Esta fase consiste en realizar pruebas con usuarios externos con el objetivo de evaluar la aplicación y analizar los datos recopilados con *los biosensores*.

	Septiembre	octubre	noviembre	diciembre	enero	febrero	marzo	abril	mayo
Estudio del estado del proyecto	Orange	Orange							
Investigación Juegos serios		Yellow							
Investigación y pruebas con biosensores			Red	Red					
Desarrollo de la aplicación			Magenta	Magenta	Magenta	Magenta	Magenta	Magenta	
Validación								Blue	

Figura 1. Diagrama Gantt del plan de trabajo

### 1.3.1 Estudio del estado del proyecto

El objetivo principal de esta fase será familiarizarse con el proyecto heredado de BlockLeap. Esta fase será fundamental para comprender la estructura, el funcionamiento y las limitaciones del software en el que se basa el proyecto. Permitirá al equipo entender a fondo el funcionamiento de la aplicación y explorar sus componentes con el fin de comenzar a ampliarlos y reutilizarlos para el desarrollo de nuevas funcionalidades.

### 1.3.2 Investigación de juegos serios

Durante esta fase, se llevará a cabo una investigación centrada en *juegos serios* (*serious games*) de carácter educativo y de otras plataformas con comunidades como *Scratch* y *Kahoot*, con el objetivo de comprender su estructura, su aplicabilidad educativa y su potencial para la enseñanza de programación.

Uno de los apartados que más tendremos en cuenta en el análisis, será valorar el componente social y colaborativo de estas plataformas, debido a que la parte de la comunidad es especialmente relevante para el enfoque de este Trabajo de Fin de

Grado, pues uno de los objetivos de este es la creación de una comunidad abierta en nuestra plataforma.

### ***1.3.3 Investigación y pruebas con biosensores***

Durante el desarrollo del proyecto se plantea la necesidad de testear los niveles creados. También surge la cuestión de qué respuesta fisiológica pueden presentar los usuarios a la hora de enfrentarse a estos niveles y cómo relacionar la relacionar estas respuestas a la dificultad de los niveles.

Es por ello que aparece la idea de utilizar un dispositivo hardware que pueda recopilar datos biométricos como la frecuencia cardíaca y la actividad electrodérmica.

### ***1.3.4 Desarrollo de la aplicación***

Esta fase se plantea como la más extensa y exigente del proyecto, ya que implica mejorar y ampliar una plataforma web ya existente, perfeccionando su funcionamiento, rendimiento y prestaciones. A partir de la investigación en juegos serios, se definirán las mecánicas clave que facilitarán el aprendizaje de programación.

La plataforma ya incorpora un editor de niveles, por lo que en esta etapa se prevé mejorar este componente para dotarlo de mayor versatilidad y accesibilidad. Además, se contempla la realización de mejoras en la estructura general de la aplicación, la interfaz de usuario, la lógica del juego y su integración con la comunidad con el fin de ofrecer una experiencia más completa.

### **1.3.5 Validación**

La fase de validación se centrará en la realización de pruebas con usuarios finales con el fin de evaluar la efectividad de la plataforma web. El objetivo de estas pruebas será recoger retroalimentación directa, identificar posibles áreas de mejora y ajustar tanto la experiencia de aprendizaje como la dificultad de los niveles.

Asimismo, los resultados obtenidos en esta fase permitirán establecer una base sólida de recomendaciones y ajustes que podrán desarrollarse en el futuro para seguir mejorando la aplicación.

## Capítulo 2 - Estado de la cuestión

En el panorama actual de la educación en España, se puede ver una tendencia creciente hacia el uso de las TICs<sup>1</sup> en las aulas. El uso de portátiles y tabletas se va normalizando y con ello la utilización de plataformas digitales que permiten a los más jóvenes aprender a manejar las nuevas tecnologías [4]. Una de las soluciones más populares es la gamificación del aprendizaje [5], siendo el uso de juegos serios como Scratch un claro ejemplo de ello [6].

Un juego serio (*serious game*) [2] es un tipo de juego que, aunque incorpora elementos propios del entretenimiento como narrativa, niveles, recompensas o mecánicas interactivas, tiene como objetivo principal enseñar, entrenar, sensibilizar o transmitir conocimientos y valores. Su finalidad va más allá del ocio, aplicándose en ámbitos como la educación, la salud, la concienciación social o la formación profesional, donde se utiliza para reforzar el aprendizaje o modificar conductas de forma motivadora y participativa. En este sentido, un juego serio no renuncia a ser divertido, pero su propósito central es formativo o funcional, y no meramente recreativo.

Una de las primeras labores del proyecto fue la investigación sobre el mercado actual de juegos serios enfocados a la enseñanza de programación. El objetivo de esta investigación fue identificar posibles mejoras que poder añadir a nuestra aplicación, así como funcionalidades para hacerla más completa.

---

<sup>1</sup> Tecnologías de la información y comunicación

## **2.1 BlockLeap**

El análisis de BlockLeap ha sido más exhaustivo que del resto de juegos serios, ya que es la plataforma web que hemos heredado y en la que hemos estado trabajando durante este proyecto. A grandes rasgos, en la fase inicial del proyecto estaban implementadas las siguientes funcionalidades:

- Jugar niveles
- Crear usuarios
- Crear niveles personalizados
- Ver y jugar niveles creados por los usuarios
- Ver el perfil y estadísticas del usuario

A continuación, realizamos un análisis un poco más detallado de la aplicación, este se dividirá en:

- Estado del cliente web: tratará de aspectos del frontend del proyecto.
- Estado del juego serio: tratará de la implementación del juego serio.
- Estado del servidor backend: tratará de la implementación del backend del proyecto.

### **2.1.1 Estado del cliente web**

La página web sigue una implementación tipo SPA (single-page application), en el que la aplicación consta de únicamente una página en la que se va actualizando el cuerpo para mostrar los diferentes contenidos por los que se navega.

En cuanto a contenidos implementados, la plataforma web constaba de 5 apartados diferentes, accesibles mediante una barra de navegación como la mostrada en la Figura 2. Los contenidos de los apartados son los siguientes:

- *Play*: muestra los niveles denominados oficiales, que son los niveles creados con el fin de proporcionar una experiencia guiada para aprender los conceptos básicos del juego, organizados en categorías según los conocimientos que se pretenden enseñar.
- *Create*: permite a los usuarios crear sus propios niveles mediante un editor con interfaz visual y sencilla.
- *Learn*: únicamente aparece en la barra de navegación, no tiene ningún contenido.
- *Connect*: muestra un listado de los niveles creados por todos los usuarios.
- *Profile*: muestra los datos del usuario y sus niveles propios.



Figura 2. Barra de navegación del proyecto original

Por último, pero no menos importante, la aplicación web permite a los usuarios continuar usando estos contenidos sin conexión. Esto se debe a que tiene implementado un Service worker que es un script que el navegador ejecuta en segundo plano que permite funciones como la caché avanzada, notificaciones push y el trabajo sin conexión.

## 2.1.2 Estado del juego serio

La vista del juego serio consta de dos apartados como se puede ver en la Figura 3. El primero de ellos, es un área donde el jugador maneja los bloques que le permiten controlar a su personaje. El segundo, es la representación del nivel que se está jugando y las acciones que representan los bloques que conforman la solución del jugador.

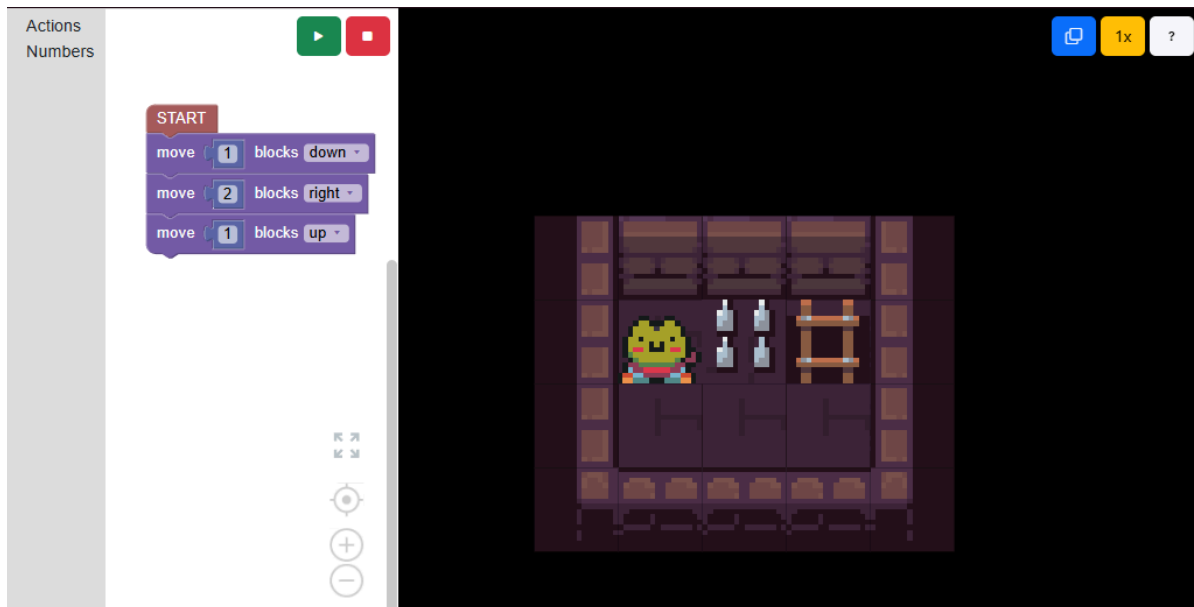


Figura 3. Interfaz del juego

Antes de explicar más en detalle estos dos apartados, vamos a explicar en qué consiste la jugabilidad de BlockLeap. Un nivel de este juego consiste principalmente en un escenario donde el jugador tendrá que mover a su personaje hasta la salida del nivel para completarlo y obtener estrellas según el rendimiento que haya tenido. La medición del rendimiento originalmente constaba de 3 factores:

- Si el jugador no se ha chocado con ningún obstáculo obtiene una estrella.
- Si hay cofres y el jugador ha pasado por encima de todos los cofres obtiene otra estrella.
- Si el jugador ha llegado a la salida obtiene una o dos estrellas en función de si hay cofres o no en el nivel.

La representación del escenario y la lógica del juego están implementadas con el framework de Phaser<sup>2</sup>. El juego consta de los siguientes elementos:

- *Rana: representa el personaje controlado por el jugador.*
- *Esqueleto: representa un personaje enemigo que se mueve con una trayectoria recta. El jugador pierde en el caso de que choque con este elemento.*
- *Suelo: representa visualmente las zonas por las que puede moverse el jugador.*
- *Pared: representa un obstáculo que delimita el área de movimiento .*
- *Pinchos: representan un obstáculo que provoca que el jugador pierda en choque con este elemento. Este elemento puede desactivarse, provocando que actúe como un elemento de suelo normal.*
- *Cofre: representa un objetivo secundario del nivel, el jugador consigue más estrellas al completar el nivel habiendo pasado por encima del cofre.*
- *Escalera: representa la posición a la que debe llegar el jugador para completar el nivel.*

La interfaz que permite usar los bloques que controlan al personaje se creó usando Blockly<sup>3</sup>. Originalmente el proyecto tenía los siguientes bloques implementados:

- *Bloque "Move [x] blocks [up/down/left/right]": mueve al personaje una cantidad de x pasos en la dirección seleccionada.*
- *Bloque "Change trap status to [on/off]": Cambia el estado de las trampas según la opción seleccionada.*

---

<sup>2</sup> [Phaser](#) (framework open source basado en JavaScript para crear videojuegos 2D interactivos y multiplataforma que se ejecutan en navegadores web mediante HTML5 y WebGL.)

<sup>3</sup> [Blockly](#) (La biblioteca de Blockly agrega un editor personalizable a tu app que presenta conceptos de programación como bloques entrelazados. Produce código limpio en el lenguaje deseado y permite bloques personalizados adaptados a tu aplicación)

- Bloque de número "[x]": representa el número x que se introduzca. Su uso es combinarlo con los demás bloques para introducir números en ellos.
- Bloque "During [x] times do { }": Ejecuta x veces los bloques que se introduzcan dentro de este.
- Bloque variable "[variable]": representa la variable con el nombre seleccionado. Su uso es combinarlo con los demás bloques para introducirles el valor asignado en esta variable.
- Bloque "Set [variable] to [x]": Cambia el valor de la variable seleccionada al valor x.

Aparte de estos bloques que acabamos de introducir, en la aplicación tiene otros bloques que no tienen su funcionalidad implementada, estos son un conjunto de bloques relacionados con lógica condicional (if, else, while y comparadores lógicos).

Por último cabe destacar que el juego originalmente tenía 19 niveles en total. Estos niveles estaban repartidos en tres categorías según su contenido:

- Introduction, con 9 niveles, contiene los niveles enfocados a que el jugador se familiarice con los elementos del juego.
- Variables, con 7 niveles, contiene los niveles enfocados a que el jugador aprenda el concepto de variable y cómo utilizar las variables.
- Data types, con 3 niveles, contiene 2 niveles para introducir el bloque "Change trap status to [on/off]" y 1 nivel en el que el jugador controla dos ranas a la vez.

### 2.1.3 Estado del servidor backend

La aplicación consta de un servidor Node.js<sup>4</sup> que gestiona tanto la gestión de la comunicación con la base de datos de la aplicación, como de la transmisión de los datos de interacción del usuario hacia una plataforma de análisis de datos.

Para la comunicación con la base de datos se usa la librería de mapeo objeto-relacional (ORM) Sequelize<sup>5</sup>. El hecho de usar Sequelize ha supuesto que las interacciones con la base de datos fueran más sencillas, y además como separa la lógica de negocio de la estructura de la base de datos mejora también portabilidad y mantenibilidad del código. Para la comunicación cliente-servidor, la configuración original constaba de:

- *Un enrutador y un controlador para cada uno de los siguientes módulos: usuarios, niveles, estadísticas y grupos de usuarios. El enrutador se encarga de gestionar las peticiones del cliente mientras que el controlador gestiona la lógica de negocio de la aplicación.*
- *La definición del modelo de cada uno de los siguientes objetos: nivel, usuario, estadísticas, categorías de niveles, grupos de usuarios, grupos de niveles, niveles asignados a grupos y permiso para acceder a niveles.*
- *Un Data Access Object (DAO) (patrón de diseño que encapsula el acceso a datos) por cada uno de los modelos mencionados anteriormente.*

Para la parte de transmisión de datos de interacción del usuario, se usa xAPI<sup>6</sup> para registrar las interacciones del usuario y luego se envían estos datos a una

---

<sup>4</sup> [Node.js](#)

<sup>5</sup> [Sequelize](#)

<sup>6</sup> [xAPI](#) (especificación para el seguimiento de experiencias de aprendizaje que permite registrar actividades realizadas por un usuario dentro y fuera de plataformas e-learning)

plataforma de Learning Record Store(LRS)<sup>7</sup>, que guarda estos datos para su visualización y análisis. En la Figura 4 podemos ver un ejemplo de la visualización de estos datos.

Person	04e76c76-976f-4db3-9b9e-a6b5825557d4	Everyone	Change
Interactions	15	9	↑ 67%
Activities	4	2.5	↑ 60%
Actions	4	2.5	↑ 60%

Figura 4. Ejemplo de datos guardados de un usuario

## 2.2 Code Monkey

Code Monkey<sup>8</sup> es una plataforma web de pago que permite a padres y profesores dar una introducción a la programación a los más jóvenes. Teniendo diferentes funcionalidades para el seguimiento individual o en grupo dependiendo de si eres un profesor o un padre. Por ejemplo, en la Figura 5 se ve uno de los menús informativos a los que tiene acceso un profesor, este menú indica los ejercicios resueltos por cada alumno e indica su progreso total de los niveles del curso.

---

<sup>7</sup> LRS (sistema que almacena, gestiona y consulta los datos de aprendizaje recogidos mediante xAPI, permitiendo registrar experiencias de aprendizaje desde múltiples fuentes y dispositivos.)

<sup>8</sup> [Code Monkey](#)



Figura 5. Información de progreso de los estudiantes en Code Monkey

Los niveles de Code Monkey están diseñados para resolverse bien mediante programación por bloques como en nuestro proyecto o por programación basada en texto, siendo esta última el modelo de la mayor parte de sus niveles. A nivel de complejidad es una de las plataformas que tiene cursos con conceptos más avanzados, por poner un ejemplo uno de sus cursos para niños de 11 a 13 años ya introduce conceptos como clases, estructuras de datos y algoritmos de ordenación usando el lenguaje de python. Podemos apreciar uno de estos niveles en la Figura 6



Figura 6. Nivel del curso Banana Tales de CodeMonkey

## 2.3 Kodu Game Lab

Kodu Game Lab<sup>9</sup> es una aplicación de escritorio, y originalmente Xbox 360, que permite a los usuarios crear y compartir mundos virtuales, estableciendo la lógica que controla los elementos pertenecientes a estos mediante bloques de instrucciones simples como los que se ven en la Figura 7.

Uno de los elementos que más nos ha servido de inspiración ha sido el apartado de tutoriales de la plataforma, este apartado contiene guías tanto para el uso básico de la aplicación así como del comportamiento y manejo de los diferentes elementos concretos del juego.

---

<sup>9</sup> [Kodu Game Lab](#)



Figura 7. Asignación de la lógica de comportamiento en Kodu Game Lab

## 2.4 Code.org

Code.org<sup>10</sup> es una organización sin ánimo de lucro que gestiona una plataforma web que cuenta con una gran variedad de cursos de programación tanto conceptuales como prácticos. Los cursos están muy bien diseñados, con una curva de progresión bien definida y están complementados con cortos animados o pequeñas presentaciones de diapositivas que ayudan a asimilar los conceptos. Además, como se puede ver en la Figura 8, estos cursos están muy gamificados con interfaces tematizadas hacia los intereses de los más jóvenes.

---

<sup>10</sup> [Code.org](https://code.org)



Figura 8. Nivel del curso Minecraft: Voyage Aquatic de code.org

La plataforma web no solo es muy sólida por la parte de experiencia al alumno, ya que cuenta con un repertorio muy amplio y detallado de cursos, entre ellos varios con jugabilidad y controles similares al de nuestro proyecto. Si no que además de ello proporciona herramientas como la creación de grupos de alumnos, dando un seguimiento detallado del avance de estos para los profesores. En la Figura 9 se puede apreciar una de las herramientas de seguimiento disponibles en Code.org.

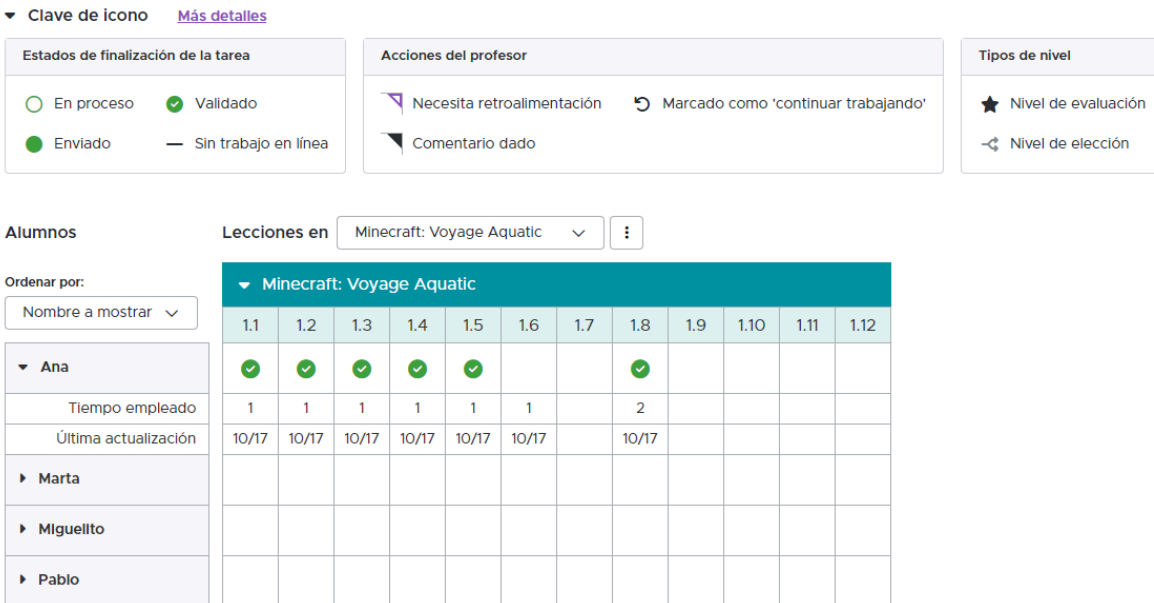


Figura 9. Seguimiento del progreso de los alumnos en Code.org

Por todo esto, Code.org ha sido un buen ejemplo a seguir para el proyecto, dando ideas de posibles mejoras. Sin embargo, también hemos visto la necesidad de encontrar la identidad propia de nuestro proyecto para diferenciarlo de code.org. A raíz de ello, identificamos la creación de niveles como la prestación que nos diferenciaba del resto del mercado y enfocamos gran parte de nuestros esfuerzos a mejorarla.

## 2.5 Carencias detectadas

Comparando nuestro proyecto con los juegos analizados en el estado de la cuestión, se pudo concluir que aún nos quedaba mucho margen de mejora, ya que las alternativas más populares ofrecían un servicio más completo y pulido.

A continuación, listamos los aspectos en los que hemos detectado que hay que mejorar nuestra aplicación:

- Carecemos de una manera en la que los profesores puedan controlar un poco la experiencia de los alumnos dentro de la aplicación.
- La aplicación no enseña a los usuarios a usar la mayoría de las funcionalidades presentes en la aplicación.
- El editor de niveles tiene muy pocas opciones, lo cual limita el diseño de niveles a los usuarios.
- Tenemos menos variedad de niveles, esto se debe a que hay menos mecánicas con las que interactuar en nuestro juego serio frente a otras opciones del mercado.

Como plan de mejora del proyecto pudimos utilizar prestaciones de las demás plataformas como objetivo a implementar, siendo un ejemplo de ello las funcionalidades relativas a la creación y gestión de grupos de alumnos. No todas las mejoras se centraron en intentar ponerse parejo a otras soluciones, pues gran parte del esfuerzo ha ido en la mejora del editor de niveles, que habíamos identificado como el pequeño nicho de nuestro proyecto.

## Capítulo 3 - Cambios y mejoras de la Aplicación

Este capítulo presenta los cambios y mejoras implementadas en la aplicación después de haber realizado un análisis de la misma, exponiendo las razones por las que se llevan a cabo estas.

Partiendo de una arquitectura como la mostrada en la Figura 10, hemos propuesto mejoras en los apartados del cliente web y servidor web, estas mejoras son las siguientes:

- Diseño responsive de la web
- Creación de la sección *Classroom*
- Mejora sección *Community*
- Mejora sección *Profile*
- Mejoras en el *editor de niveles*
- Creación sección *Tutorials*
- Mejora de elementos ya existentes en el juego

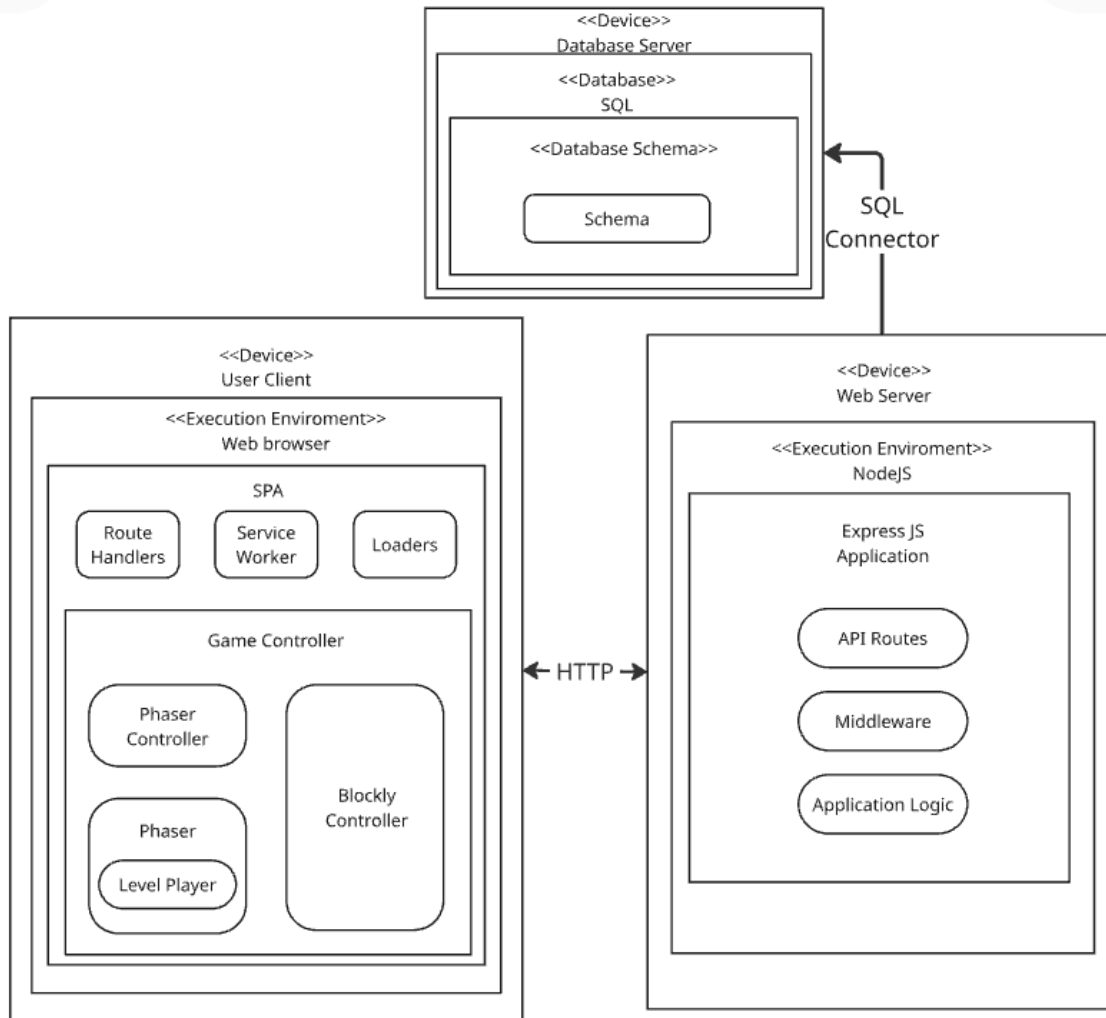


Figura 10. Arquitectura de la aplicación simplificada

### 3.1 Diseño responsive de la web

En España alrededor de un 83% de la población cuenta con un ordenador (portátil, sobremesa o tablet) en casa [17], eso unido al creciente uso de la tecnología dentro de las aulas a través de dispositivos portables nos hizo ver la importancia de añadir el diseño responsive mediante la actualización del CSS ya existente. Esto es la adaptación de la interfaz visual al tamaño de pantalla en el que estás corriendo la aplicación web, ya que no es la misma pantalla ni tiene las mismas características, la

de una tablet con la de una pantalla de un ordenador de sobremesa. Esto se hizo mediante el uso de Media Queries y del framework Bootstrap.

Gracias a estos cambios, la plataforma ofrece ahora una experiencia de usuario coherente, accesible y funcional desde tablets hasta pantallas de escritorio, ideal para poder trabajar en cualquier entorno y con cualquier dispositivo portátil o de sobremesa, sin necesidad de estar pensando en redimensionar o no la pantalla.

### **3.2 Creación de la sección classroom**

Con el objetivo de crear la sección Classroom, que permite a los profesores gestionar grupos de alumnos y asignarles niveles y grupos de niveles, implementamos una serie de funcionalidades específicas, basadas en el análisis de otros juegos serios educativos.

La sección classroom al inicio del proyecto no existía, es una ampliación que hemos hecho basándonos en el estudio de otros juegos serios. Classroom es el apartado de la aplicación donde los profesores podrán crear sus propias clases personalizadas y privadas para sus alumnos. Tras analizar diferentes juegos llegamos a un entendimiento de cómo organizaremos la sección. Para acceder a la sección classroom el usuario deberá estar registrado y haber iniciado sesión, como podemos observar en la Figura 11.

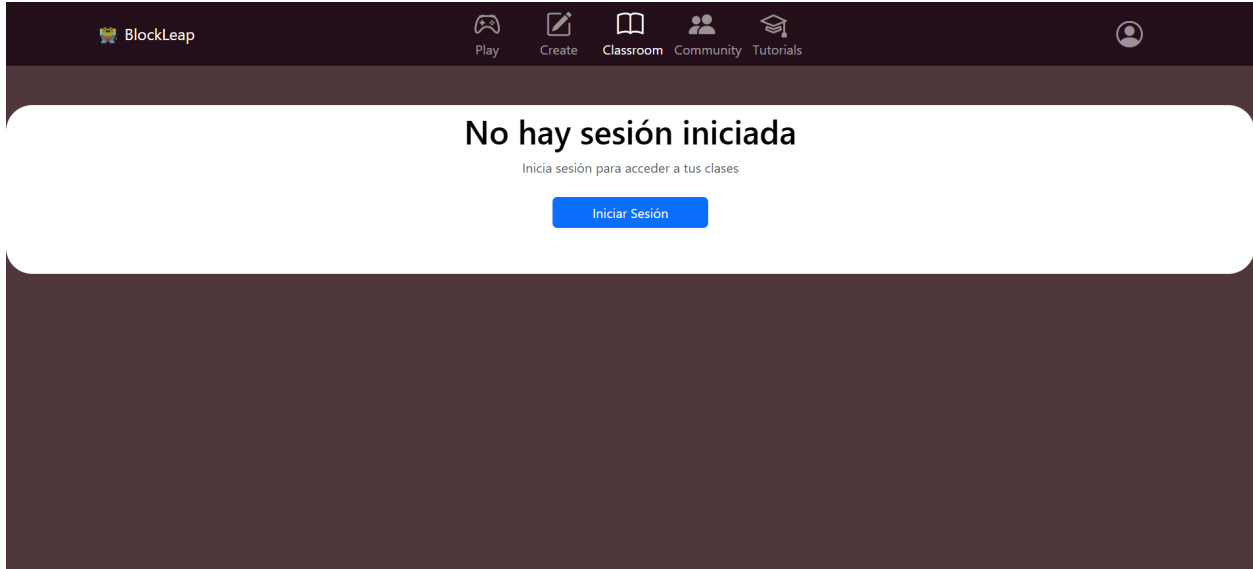


Figura 11. Iniciar sesión para acceder a classroom

### 3.2.1 Waiting room

Con el fin de organizar las clases para hacerlo más intuitivo para los usuarios, hemos implementado una pantalla intermedia que denominamos “waiting room” donde el usuario puede ver todas las clases a las que pertenece y entrar en una nueva clase a partir de un código que proporciona el profesor. Además, si el usuario tiene rol de profesor, hemos implementado un botón (Figura 12) para que pueda crear una nueva clase poniendo un nombre y una descripción (Figura 13). Además, hemos implementado un botón para alumnos y profesores para introducir un código cuyo objetivo es facilitar el registro de los estudiantes en las clases, el cual se genera automáticamente con la librería short-unique-id. (Figura 14)

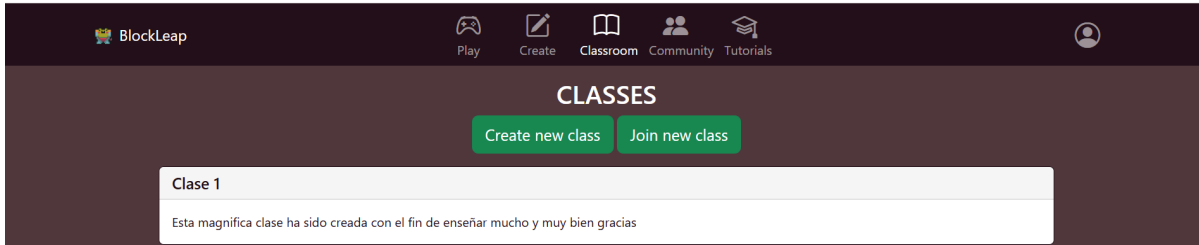


Figura 12. Waiting room con rol de profesor

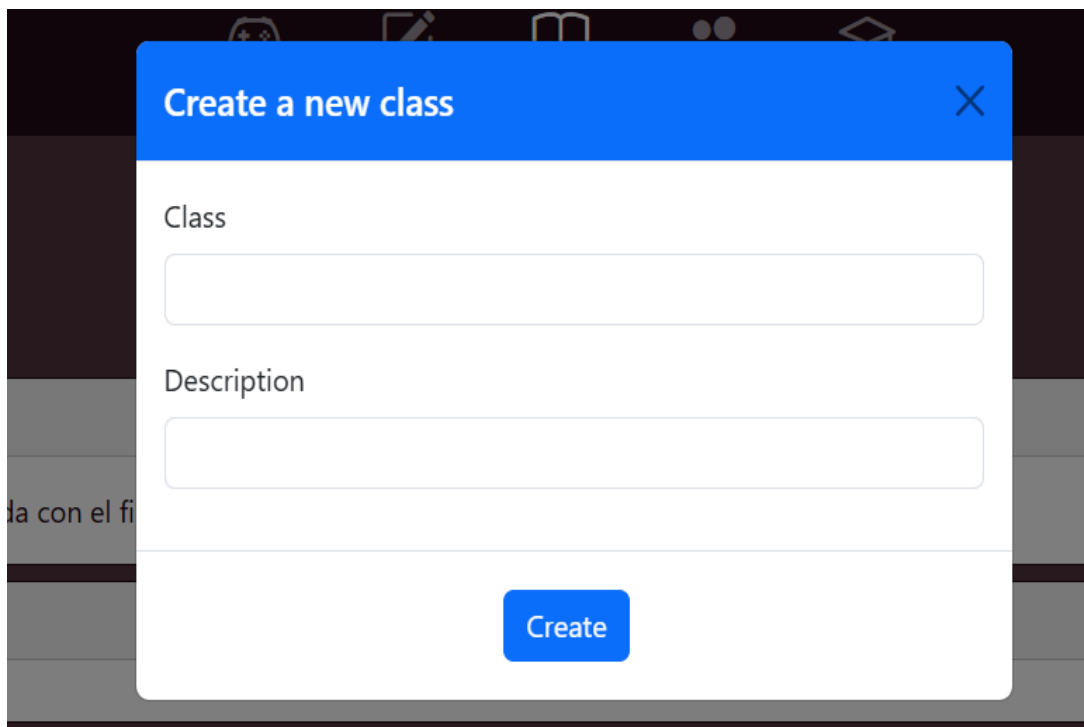


Figura 13. Crear una nueva clase con rol de profesor

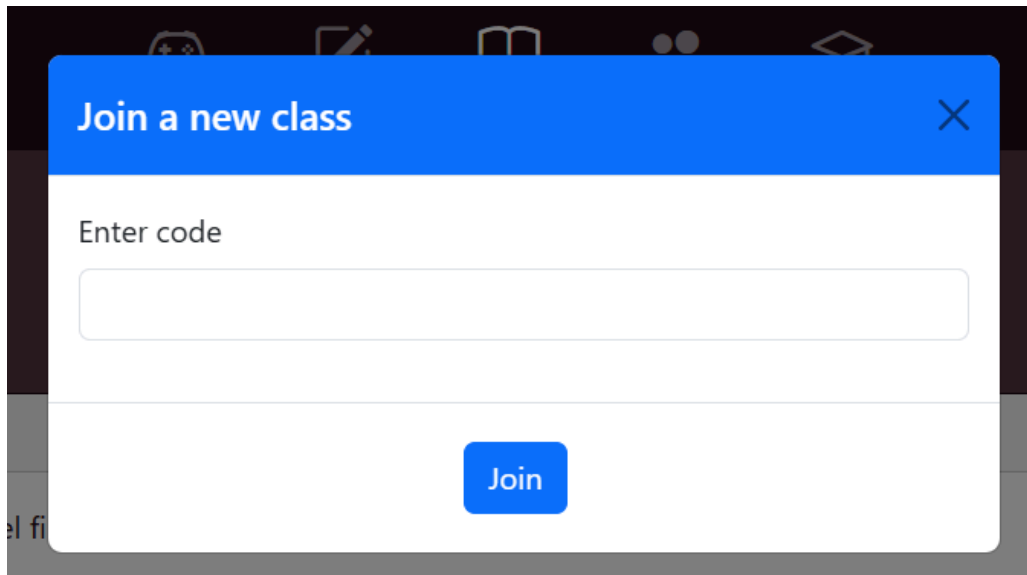


Figura 14. Unirse a una clase con un código de clase

En esta parte hemos implementado una vista classroom en donde tras la investigación de diferentes juegos serios se nos ocurrió asociar niveles a conjuntos, los cuales un profesor podía crear o modificar con sus propios niveles, o con niveles de la comunidad y asociarlos a sus clases (Figura 15). Estos conjuntos de niveles o "sets" los añade el profesor a la clase con un menú desplegable en el que marcando con un checkbox los niveles puede introducirlos a la clase o quitarlos si los desmarca. Desde este menú desplegable podemos acceder al modal para crear sets, este último también está disponible desde el apartado del perfil. (Figura 16 y Figura 17).

También, en la clase, el profesor puede añadir niveles sueltos que no vayan en ningún conjunto por si quisiese añadir niveles adicionales que no tuviesen que ver con alguna temática de sus conjuntos. La carga de estos niveles tiene paginación y filtros para que aparezcan ordenados en páginas.

Además, hemos incluido dos botones más para el profesor. Uno en el que puede ver la lista de estudiantes que pertenecen a la clase (Figura 18) y otro para ver

el código de la clase con un modal que tiene un botón de copiar en el portapapeles y así poder enviárselo a sus estudiantes (Figura 19).

En la clase, un alumno con la implementación que hemos realizado solamente puede ver los conjuntos de niveles y los niveles individuales añadidos por el profesor ordenados y jugar a los niveles propuestos por el profesor.

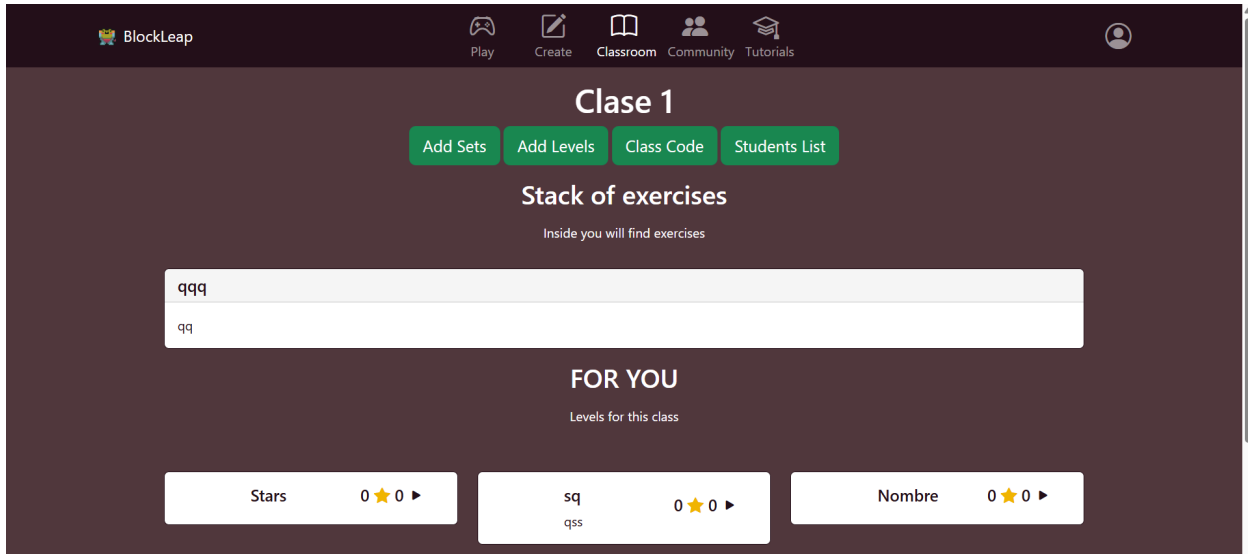


Figura 15. Vista clase con rol de profesor

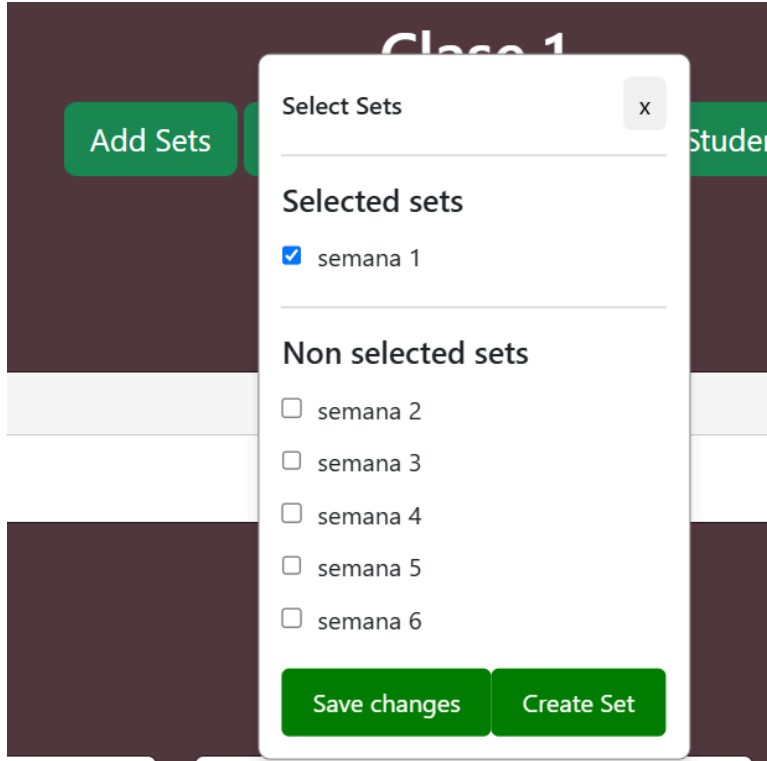


Figura 16. Modal añadir sets

### Create Level Set ✕

Name of the Set

Descriptionn

Add levels

Filter by tags

Stars

sq

Nombre

433

1212

aaaaaa

1

Figura 17. Modal crear sets

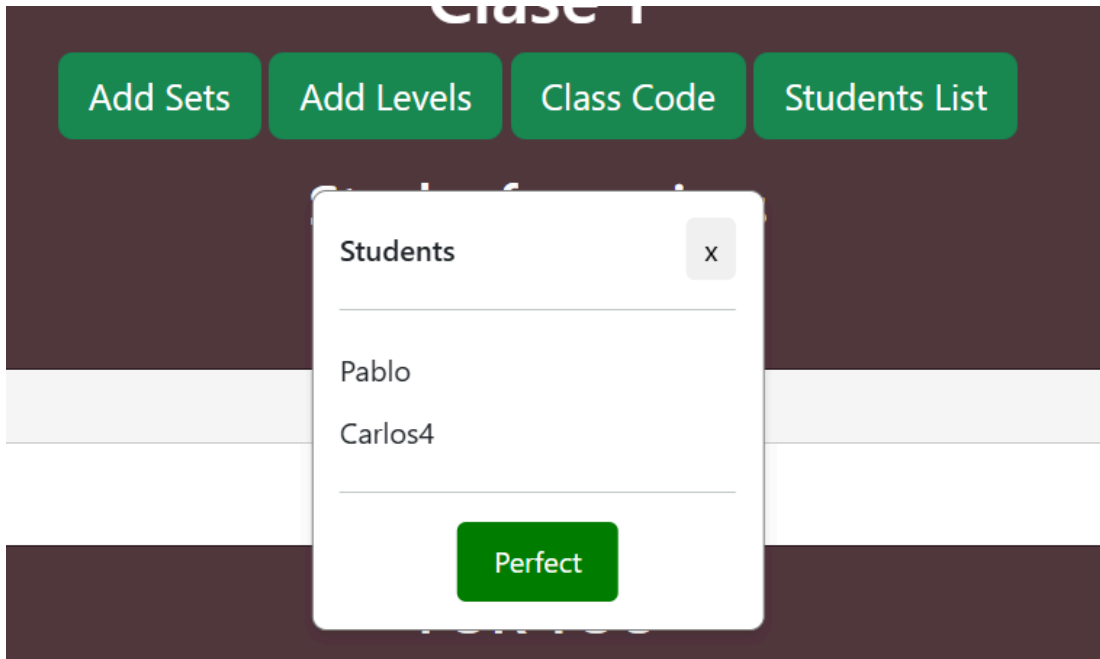


Figura 18. Modal ver alumnos de la clase

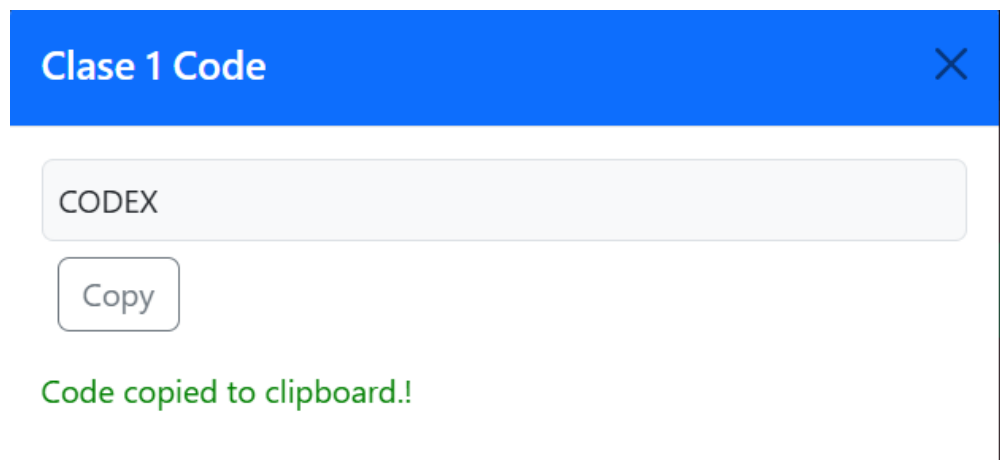


Figura 19. Modal ver código

### 3.3 Mejora de la sección community

La sección community de la aplicación permitía ver a los jugadores todos los niveles que se habían creado por los usuarios. En su estado original mostraba y cargaba la totalidad de los niveles en la misma página, pero observamos que en el momento que hubieran suficientes niveles creados, existiría una saturación visual en la información que se le daba al usuario por lo que decidimos mejorar las prestaciones de esta sección.

Por un lado, modificando las consultas de sequelize y añadiendo una barra de paginación, se ha implementado la carga paginada de niveles para que los datos se vayan estructurando en páginas y el usuario no tenga que cargar la totalidad de los datos de golpe. Podemos ver un ejemplo de la paginación en la Figura 20.



Figura 20. Paginación de los niveles

Por otro lado, se ha implementado una búsqueda por filtros para los niveles, permitiendo a los usuarios buscar únicamente los niveles de categorías que les interesen. La implementación de esta funcionalidad implicó modificar la estructura de la base de datos para soportar las etiquetas que se usan para filtrar y clasificar niveles. La implementación de la barra de búsqueda se hizo usando la librería Select2 que aporta más funcionalidad a las select boxes de HTML. A continuación, se muestra la barra de búsqueda en la Figura 21.

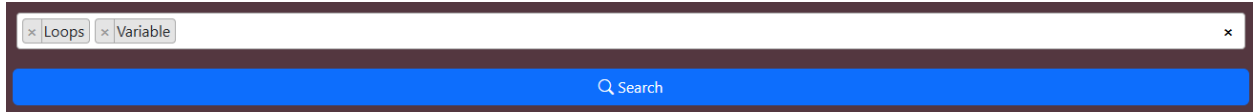


Figura 21. Búsqueda por filtros

### 3.4 Sección perfil

La sección perfil permitía al usuario conocer los niveles que había superado con éxito, conocer las estrellas totales que había ganado y el tipo de rol que desempeñaba en la aplicación (“alumno”, “profesor” o “administrador”). Sin embargo, consideramos insuficiente estas funcionalidades, ya que no permitían al usuario tener una visión y acceso completo a sus contribuciones dentro de la plataforma. Decidimos mostrar los niveles y sets creados por el usuario en esta pantalla. Implementamos una visualización diferenciada según el rol del usuario, adaptando la interfaz y la información mostrada a las necesidades de cada tipo de perfil. Para ello hemos hecho una distinción entre los roles de alumno y profesor/administrador.

Si el perfil es el de un alumno, se mostrará una lista paginada de los niveles creados por ese usuario. En cambio, si el rol es de profesor/administrador, se mostrará, además de los niveles, una tabla paginada con las carpetas de niveles (o sets) creados, en las que se incluye el nombre y la descripción de estas carpetas y el número de niveles que contienen. Esta tabla la hemos generado utilizando la librería DataTables, la cual recoge los datos que le pasamos y crea una tabla con ellos con paginación, ordenación, número de entradas por página y búsqueda incluidas (Figura 22).

La mejora en la sección de perfil ha ofrecido a todos los usuarios una visión más completa de sus aportaciones. En particular, los profesores y administradores

obtuvieron un mayor control sobre los contenidos que habían generado, facilitando la gestión de sus materiales.

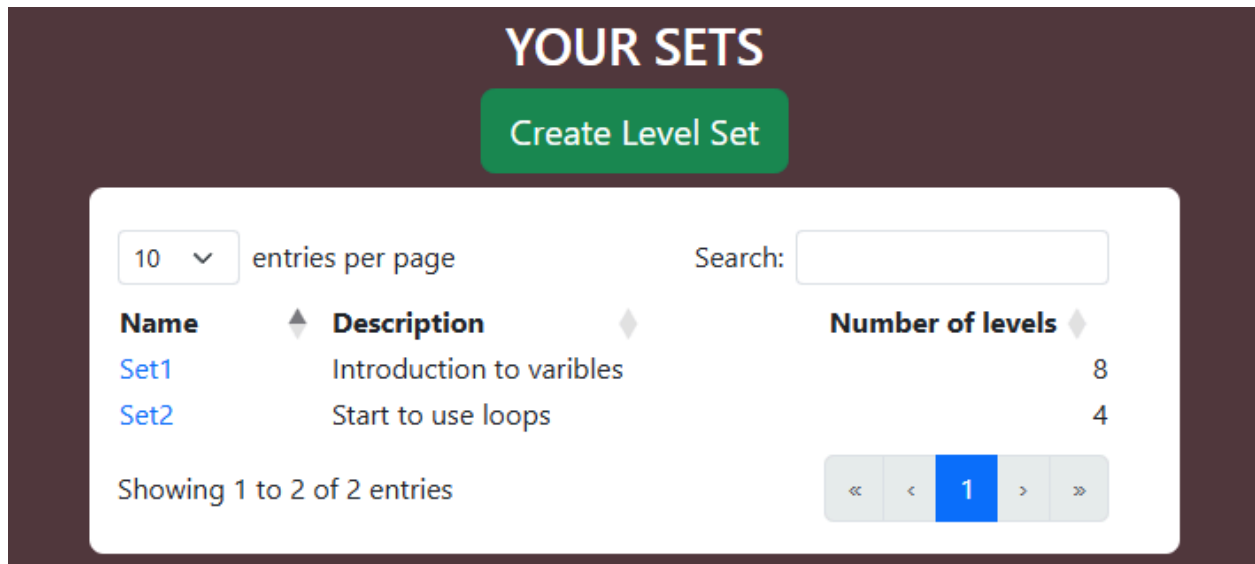


Figura 22. Tabla de sets en el perfil de profesor

### 3.5 Sección Tutoriales

Con el estudio de juegos serios nos dimos cuenta de que la gran mayoría de aplicaciones tienen una sección de tutoriales y vídeos explicativos para que el usuario pueda entender el correcto funcionamiento de la aplicación y no quedarse atascado en caso de que no sepa cómo navegar a través de la aplicación. Por tanto, decidimos implementar la sección de tutoriales de cero. Esta es la sección donde un usuario podrá consultar los diferentes tutoriales acerca de BlockLeap. Estos tutoriales ayudarán al usuario a cumplir ciertos objetivos dentro de la web, sin necesidad de acudir a recursos externos.

Hemos creado los siguientes tutoriales: (Figura 23).

- Getting started: donde el usuario encuentra los primeros pasos para iniciar la web, registrarse y poder jugar
- Create Levels: donde se explica al usuario como puede crear niveles desde el editor.
- How to play: donde se explica al usuario como jugar los diferentes niveles de BlockLeap
- Classrooms: donde el usuario aprende a utilizar las clases de BlockLeap
- Community: donde se aprende a usar la comunidad de BlockLeap
- Your profile: este tutorial explica los detalles más importantes de la sección perfil del usuario.

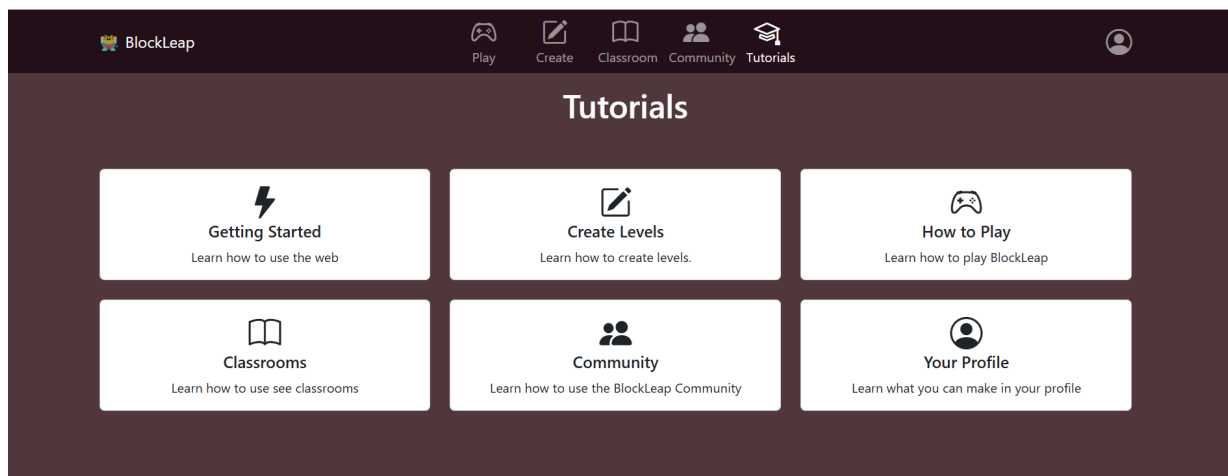


Figura 23. Sección tutoriales

Dentro de la sección de tutoriales hemos implementado dos tipos de vistas, una para el administrador de la web y otra para el resto de usuarios ya sean alumnos o profesores.

El administrador puede escribir tutoriales en formato Markdown<sup>11</sup> desde un editor de la propia aplicación, este texto se transforma a HTML para su visualización mediante

---

<sup>11</sup> Markdown: lenguaje simple que se utiliza para crear texto enriquecido (por ejemplo, HTML) con un editor de texto sin formato.

el uso de la librería `Marked`<sup>12</sup>. En la Figura 24 podemos ver la vista del administrador dentro de uno de los tutoriales.

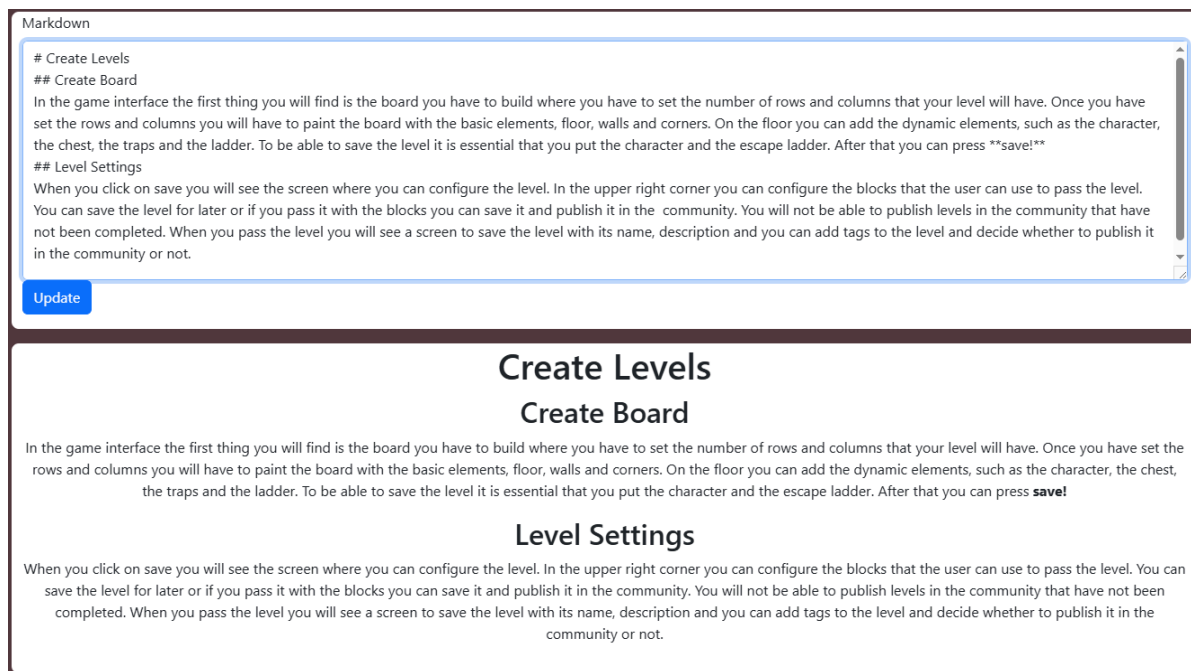


Figura 24. Vista de administrador en una sección de tutorial

El resto de usuarios simplemente ve los tutoriales y no puede interactuar con ellos (Figura 25).

---

<sup>12</sup> [Marked](#): librería de javascript que analiza y transforma texto de formato markdown a html.

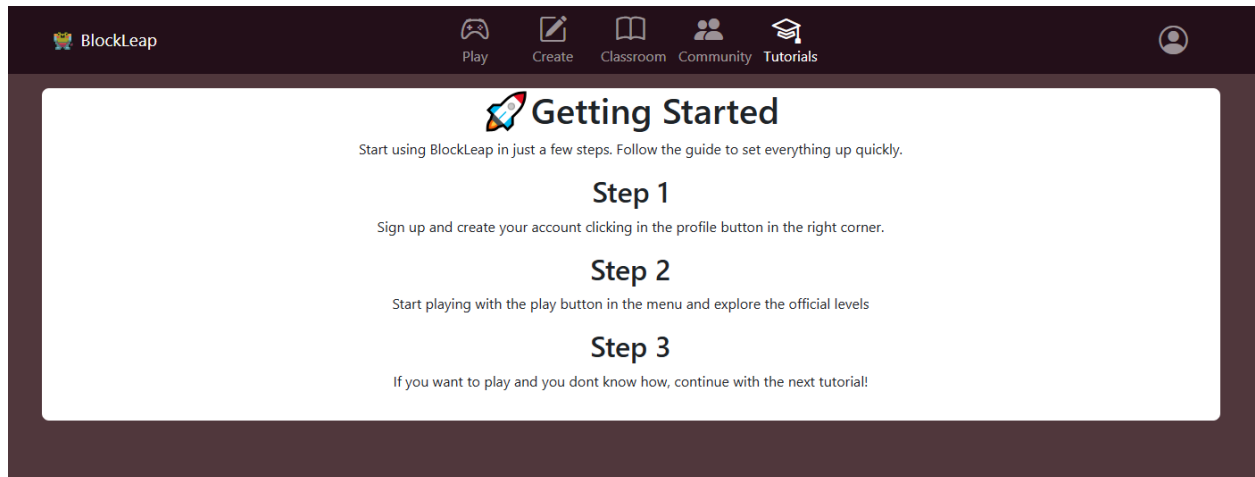


Figura 25. Ejemplo de tutorial

## 3.6 Mejoras en el editor de niveles

Una de las conclusiones que obtuvimos de la investigación de juegos serios fue la necesidad de mejorar el editor de niveles de BlockLeap, ya que lo identificamos como uno de los factores diferenciadores respecto a otros juegos serios educativos. Por ello durante el desarrollo hemos añadido nuevas funcionalidades para ofrecer más opciones al crear niveles. Los siguientes apartados, abordan las limitaciones identificadas y cómo las hemos solventado.

### 3.6.1 Opción de editar los bloques que aparecen en el nivel

Durante la fase de investigación y familiarización con la aplicación previa, identificamos que en algunos niveles ya existentes se había impuesto una limitación en cuanto al uso de los bloques de acción. Esta restricción había sido definida por el programador para ciertos niveles concretos. Teniendo en cuenta esta lógica de funcionamiento, decidimos implementarla en el editor, de modo que el propio usuario

podiera establecer límites tanto en los bloques de acción disponibles como en los que se pueden utilizar para resolverlos.

Implementamos un botón en la zona superior derecha del editor de niveles que despliega un menú en la parte lateral derecha de la pantalla. En este menú, aparecen cada uno de los tipos existentes de bloques, acompañados por un ejemplo visual de estos para facilitar su identificación.

Cada tipo de bloque incluye la opción "Enable" que se puede activar o no para permitir el uso de ese bloque en concreto al jugar el nivel. Además, añadimos una sección que permite limitar el número de veces que se puede usar el bloque durante el nivel (Figura 26).

La incorporación de esta funcionalidad ha permitido a los creadores de niveles mayor control sobre la complejidad y orientación que le quiera dar al nivel. Beneficia especialmente a los profesores, quienes pueden adaptar los niveles a objetivos y competencias concretas, mejorando así la utilidad educativa de la aplicación.

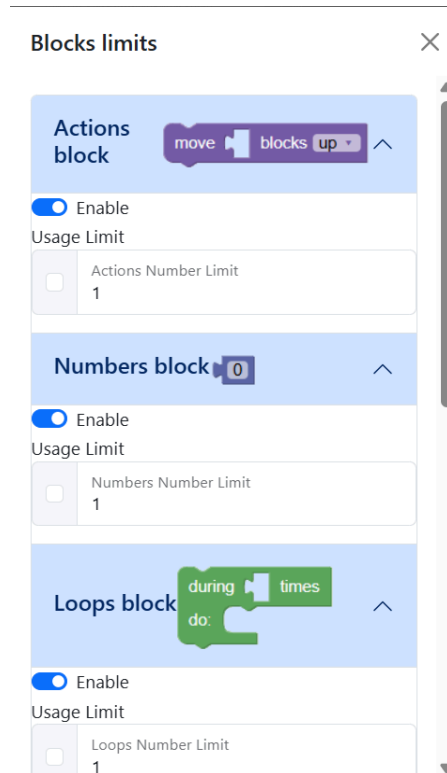


Figura 26. Limitador de bloques

### 3.6.2 Creación de un modal para guardar el nivel

En el proyecto inicial, para poder guardar un nivel editado o creado desde el editor, teníamos que resolverlo primero y este se guardaba con el nombre por defecto "Editor". No existía la posibilidad de editar esa información. Esto no resulta una buena práctica, puesto que, si te equivocabas o simplemente querías cambiar el nombre del nivel, no podías hacerlo.

Nuestra solución ha consistido en introducir un modal que se muestra al finalizar la creación o edición de un nivel, solicitando al usuario información adicional sobre este antes de proceder a guardarlo. Esto nos permite estructurar mejor los datos asociados a cada nivel.

El modal diseñado (Figura 27) solicita al usuario los siguientes campos:

- Un nombre de nivel.
- Una descripción opcional para contextualizar el contenido.
- Un selector de etiquetas para facilitar la categorización y búsqueda de niveles según temáticas o mecánicas específicas.
- Una casilla de visibilidad, que permite al usuario marcar el nivel como público o no. Si el nivel es marcado como público, se muestra en la sección *Community*, siendo visible y jugable para cualquier usuario. En caso contrario, sólo es accesible para el propio creador desde su perfil.
- Un selector de estrellas que define los criterios de evaluación del nivel. Hasta que no se complete la asignación en cada una de las estrellas, el botón "Submit" permanece desactivado, lo que hace obligatoria esta selección para finalizar el guardado. Especificaremos este tema más adelante.

Esta mejora ha optimizado la experiencia de creación de niveles significativamente. Los usuarios pueden nombrar, definir y clasificar mejor sus niveles, lo que facilita su gestión y la interacción con otros usuarios a través de la comunidad.

**Enter your level data** ✕

Name

Description

Tags

Clear this level to be able to publish

---

**Stars selector**

★      ★      ★

▼     ▼     ▼

---

Figura 27. Modal para guardado de nivel

### 3.6.3 Selector de estrellas al crear el nivel

En la primera versión del juego, la recompensa por resolver el nivel de la manera más eficaz o adecuada se reflejaba en un total de 3 estrellas que se podían ganar usando ciertos recursos al completar el nivel. Estas estrellas se asignaban simplemente por terminar el nivel o no. Esto nos pareció un problema que abordar, puesto que no dejaba claro el cómo se puntuaba el nivel, ni se realizaba correctamente. Por ello hemos implementado un sistema que, dependiendo de cómo se haya resuelto el nivel y los bloques que se han usado, se asignen cierta cantidad de estrellas como puntuación al completar el nivel.

El sistema de asignación de estrellas implementado cuenta con dos versiones. El sistema de condiciones de ambas versiones se basa en las siguientes acciones al completar el nivel:

- Completar el nivel
- No chocarse con los muros
- Utilizar los mismos o menos de los bloques utilizados al crear el nivel (lo más óptimo posible)
- Utilizar los mismos o menos de los bloques utilizados al crear el nivel + 2 (para dar pie a otras posibles soluciones en algunos niveles)
- Recoger todos los cofres
- Recoger un mínimo de cofres (dependiendo cuantos estén en el mapa)
- Utilizar bucles
- Utilizar variables

La primera versión era totalmente automática: se analizaba el desempeño del jugador al completar el nivel, y dependiendo del código y los bloques usados, se le otorgaban hasta 3 estrellas según los criterios impuestos. Sin embargo, esta solución era rígida y con el crecimiento de la aplicación, el código se volvería complejo puesto que habría que tener en cuenta muchos factores y realizar muchas comprobaciones, además de que no permitiría personalización.

La prioridad de asignación de estrellas de esta primera versión se regía por los siguientes criterios:

1. Bucles y/o variables.

Si al crear el nivel se habían utilizado ambos o alguno de estos dos bloques, al completar el nivel durante el juego se comprobaba si el

jugador los había utilizado también. En caso afirmativo se asigna una estrella al uso de cada uno de ellos obligatoriamente.

## 2. Cofres.

Dependiendo de las estrellas restantes por ganar y del número de cofres en el nivel, se asignaba una cantidad de estrellas determinada según las siguientes fórmulas: Si el número de estrellas restantes por asignar es igual o mayor al número de cofres en el mapa, se asigna una estrella a la recolección de cada uno de los cofres. En caso contrario, se sigue la siguiente fórmula:

$\text{Math.floor}((\text{cofresRecogidos} / \text{totalCofres}) * \text{nEstrellasRestantes})$ .

## 3. Número de bloques.

Si aún quedan estrellas sin asignar, se tiene en cuenta el número de bloques usados al crear el nivel y se asocia a una estrella. Este número se compara con el número de bloques que el jugador ha usado al resolver el nivel. Si ha usado los mismos o menos recibirá esta estrella.

Adicionalmente, hemos tenido en cuenta que, si aún quedan estrellas por conseguir, añadimos una segunda condición para otorgar estrellas por el número de bloques usados, esta vez con el límite aumentado en dos bloques, para dar un margen al jugador.

## 4. No chocarse con los muros.

Por último, en caso de quedar alguna estrella, se asigna a que el jugador no se choque con las paredes al resolver el nivel.

Aparte de complejo, este código suponía una actualización constante en caso de agregar elementos o funciones al código, además de que no mostraba la información de cómo se ganaba cada estrella.

La versión final implementa un selector manual de condiciones en el mismo modal explicado en el apartado 3.6.2, que se muestra al guardar un nivel una vez este es creado y completado. Este nuevo sistema permite al creador del nivel, dependiendo del código seleccionado para resolver el nivel y los objetos que aparecen en el mapa, decidir qué logros están asociados a cada una de las 3 estrellas. Para ello se añadieron selectores desplegables que permiten asociar una condición específica a cada una de las estrellas del nivel.

Estos selectores, de los que se muestra un ejemplo en la Figura 28, tienen por defecto seleccionada la opción "select a tag", que no cuenta como una selección. Para poder guardar el nivel se ha de haber seleccionado una opción en cada uno de los selectores, que es cuando el botón submit se activará.

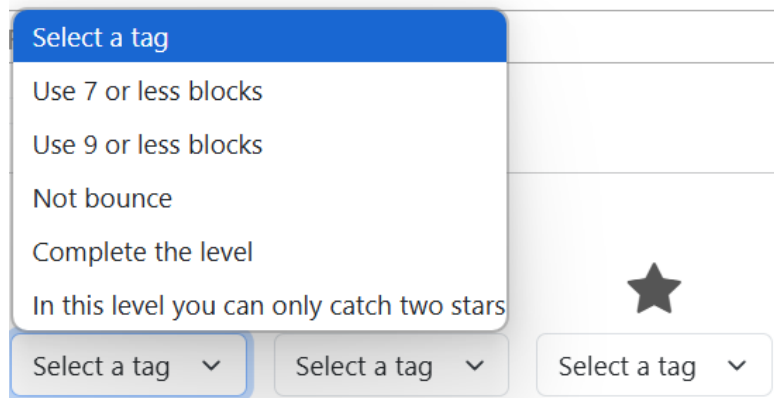


Figura 28. Selector de estrellas

Como mencionamos anteriormente, las opciones que aparecen en los selectores se toman dependiendo de los bloques de Blockly usados al solucionar el nivel durante su creación. Las condiciones para obtener estrellas son las siguientes:

- Usar [bucles/variables]: se puede establecer esta condición si el creador usó bucles o variables durante la creación del nivel.
- Recoger [todos/mitad de] los cofres: se puede establecer esta condición si el nivel contiene al menos un cofre.
- Usar [X] o menos bloques: disponible siempre, se puede establecer como el número de bloques utilizados durante la creación del nivel o una cantidad superior en dos.
- No chocar con las paredes: disponible siempre.
- Estrella no obtenible: disponible siempre. Limita la cantidad de estrellas disponibles en el nivel.
- Completar el nivel: disponible siempre.

Para evitar duplicados, estos desplegados están interconectados, por lo que, si se ha seleccionado una condición concreta en uno de los desplegados, esta ya no aparece en el resto.

Una vez se selecciona una opción en un desplegable, se colorea su estrella correspondiente indicando que está asignada a una condición. Cuando se han seleccionado condiciones para todas las estrellas, el modal queda como se muestra en la Figura 29, y se activa el botón de submit.

The image shows a mobile application modal titled "Enter your level data". It features a blue header with a close button (X). The form includes the following elements:

- Name:** A text input field.
- Description:** A text area with a small edit icon at the bottom right.
- Tags:** A text input field with the placeholder text "Filter by tags".
- Clear this level to be able to publish:** A checkbox.
- Stars selector:** A section with three yellow stars. Below each star is a dropdown menu:
  - Star 1: "Use 9 or less" (dropdown arrow)
  - Star 2: "Not bounce" (dropdown arrow)
  - Star 3: "Complete th" (dropdown arrow, highlighted with a blue border)
- SUBMIT:** A blue button at the bottom center.

Figura 29. Modal para guardado de nivel con estrellas seleccionadas

La información de las condiciones seleccionadas se guarda en la base de datos del nivel, y se muestra al jugador durante la partida en el lateral derecho de la pantalla, de forma que este sepa qué debe hacer para obtener cada estrella(Figura 30).

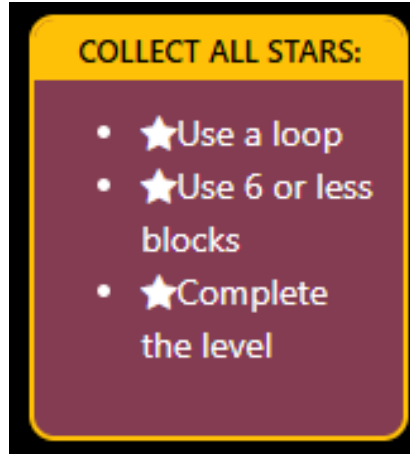


Figura 30. Información sobre obtención de estrellas en un nivel

Este nuevo sistema, mejora la experiencia del jugador y las posibilidades educativas del juego. Ahora el jugador tiene más claro los logros a alcanzar para obtener recompensas y mayor puntuación. Para los docentes y creadores de niveles aumenta el control de puntuación sobre los niveles, permitiendo lograr objetivos más concretos y adaptados a cada propósito.

Como punto adicional, al separar la lógica del juego y la de evaluación, es más fácil escalar y fácil de mantener estas partes del código.

#### **3.6.4 Guardado de código anterior**

En la versión inicial, al editar un nivel que ya estaba creado, se perdían los bloques usados como solución. Esto obligaba al usuario a reescribir todo el bloque de programación desde cero, incluso si el cambio realizado era mínimo. Esto ralentizaba el desarrollo de contenido y dificultaba iterar sobre el diseño de los niveles.

Para resolver este problema hemos implementado un sistema que guarda automáticamente el último código de bloques utilizado al crear o editar un nivel. De esta forma, cuando el usuario accede de nuevo a la edición del nivel, el código se

restaura exactamente como se había dejado, ofreciendo así continuidad en el trabajo.

Para conseguir esto, hemos utilizado una función de Blockly que permite serializar<sup>13</sup> el código en formato XML. Este código se guarda directamente en la base de datos asociada al nivel en concreto en el momento del guardado. Al volver a cargar el editor para modificar el nivel, se accede a este fragmento de código almacenado y se vuelca sobre el espacio de trabajo de Blockly. Los bloques se regeneran visualmente con la misma estructura, con conexiones y contenido que tenían en el momento del guardado.

Gracias a esta mejora, los usuarios pueden modificar un nivel sin perder el trabajo realizado previamente. Esto reduce el tiempo necesario en la edición de un nivel.

### **3.7 Mejora de elementos del juego ya existentes**

Tras probar e investigar el juego creado, realizamos una lista de mejoras a realizar en los elementos del juego que exponemos a continuación.

#### **3.7.1 Ajuste de elementos del juego ya existentes**

Tras la investigación y toma de contacto inicial con el juego, descubrimos una serie de factores que no quedaban del todo claros y que podíamos mejorar.

- Enemigo esqueleto:

Previamente al cambio, el enemigo actuaba de la siguiente manera: se movía hacia abajo en vertical y una vez chocaba con la pared desaparecía.

---

<sup>13</sup> **Serializar:** convertir un objeto, incluyendo sus datos y código en información (bytes) para su almacenamiento o transmisión.

Hemos cambiado este comportamiento, para que, al chocar con una pared, rebote, y el siguiente movimiento sea en sentido totalmente opuesto hasta que vuelva a chocar con otra pared.

El esqueleto sigue el mismo “ritmo” que sigue el resto del flujo de reglas programadas si se les cambia la velocidad de ejecución.

- Cofre de tesoro:

Hemos editado y añadido una imagen de cofre abierto siguiendo el estilo de arte al que había inicialmente para el cofre, para que cuando se diera la condición de que el personaje recoge el cofre, este no cambie solo su estado a nivel lógico sino también a nivel visual para que el usuario tenga feedback de que ha interactuado correctamente con el cofre.

### **3.7.2 Creación de nuevas funcionalidades y elementos del juego**

El diseño de niveles nuevos para BlockLeap, requirió la creación de nuevos bloques y objetos de nivel, permitiendo introducir nuevos elementos al jugador en cada nivel y manteniendo el interés. Uno de los bloques nuevos que se ha implementado es el bloque “Switch trap status”, que alterna los estados de activado y desactivado de las trampas de pinchos. El objetivo de añadir este bloque es dar más opciones de usar el elemento de las trampas en el juego. Por ejemplo, el nivel representado en la Figura 31 obliga al jugador a usar el bloque 3 veces ya que hay 3 trampas en la secuencia activada -desactivada-activada, por lo que cada vez que desactive y cruce una trampa se encontrará la siguiente activada. Cabe aclarar que esta interacción no se podría dar con el bloque de “Switch trap status to [on/off]” que ya estaba presente en el juego, ya que con solo un uso con el valor “off” seleccionado quedarían las tres trampas desactivadas.



Figura 31. Nivel que hace uso del bloque "Switch trap status"

En cuanto a los bloques relativos a la lógica condicional, al final no se han añadido, esto es debido a que se empezó este objetivo en una fase bastante tardía del proyecto, por lo que no se ha podido terminar a tiempo. Antes de introducir los bloques de lógica condicional, había que diseñar primero cuáles serían las condiciones ya que sin ellas no hay lógica condicional. En este punto el problema sería de diseño de juegos, sin olvidar el objetivo educativo del juego. Algunas de las ideas por las que iteramos fueron:

- Condiciones relativas a las trampas: una de las primeras ideas que tuvimos fue un condicional que compruebe si hay una trampa al lado del jugador. El defecto principal de esta condición es que el jugador no tiene la necesidad de comprobar esta condición porque puede ver las trampas y su estado antes de preparar la solución al problema. Se propusieron implementar zonas en las que el jugador no pudiera ver que hay dentro de ellas o trampas que se activan y desactivan aleatoriamente, forzando así a usar el condicional, pero para su futura implementación habría que tener en cuenta que como afectarían estos elementos a la claridad visual del juego.

- Condición "nivel no terminado": Esta condición se podría usar con un bloque lógico "while", ofreciendo una opción diferente a usar al bloque for x times.
- Condición "hay camino": Inspirada en implementaciones de otros juegos serios, esta condición comprobaría si hay un camino disponible en una dirección seleccionada. Aunque no es necesario, se podría complementar esta condición implementando un bloque que mueva al personaje en la dirección en la que mira y otro bloque que rotase al personaje hacia una dirección.

## Capítulo 4 - Familiarización con biosensores

Desde el principio del proyecto, una de las ideas de la propuesta era evaluar los niveles de nuestro juego mediante datos fisiológicos recogidos a través de algún hardware que tuviera esas prestaciones. El objetivo era analizar esos datos para saber si el usuario se frustraba o aburría con cada nivel del juego. Para ello se propuso utilizar la placa *EmotiBit*, puesto que teníamos un acceso fácil y gratuito a través de la universidad.

*EmotiBit* es un módulo sensor portátil para capturar datos emocionales, fisiológicos y de movimiento de alta calidad. Fácil de usar y validado científicamente, te permite disfrutar de la transmisión inalámbrica de datos a cualquier plataforma o la grabación directa de datos en una tarjeta SD integrada (Figura 32).

*EmotiBit* mide una gran cantidad de parámetros, actividad electrodérmica, pulsaciones del corazón, conductividad de la piel, tiene acelerómetro y giroscopio. Por todo ello es una herramienta perfecta para realizar el estudio que nos hemos propuesto.



Figura 32. Placa Emotibit

#### 4.1 Motivación para usar Emotibit

La decisión de utilizar *EmotiBit* en este proyecto radica en su capacidad para capturar –de manera precisa y no invasiva– datos fisiológicos que nos ayudan a comprender mejor las respuestas emocionales y físicas de los usuarios. Con esta herramienta, buscamos ir más allá de la simple observación del comportamiento en la plataforma; queremos medir la respuesta fisiológica de cada usuario mientras interactúa con los niveles y la navegación de BlockLeap.

El objetivo principal es realizar un análisis detallado y profundo sobre la experiencia del usuario. Gracias a las métricas que ofrece *EmotiBit*, como la frecuencia

cardíaca, la conductividad de la piel, la variabilidad de la frecuencia cardíaca y otros indicadores fisiológicos, podemos identificar si un nivel representa un desafío significativo o si pasa desapercibido como una tarea trivial que no genera una respuesta emocional o física relevante [8][9]. Esta información es esencial para:

1. Relacionar el estrés con la dificultad de los niveles planteados a los usuarios, analizando los datos biométricos recopilados durante pruebas con usuarios.
2. Optimizar la dificultad de los niveles: Si descubrimos que un nivel resulta demasiado sencillo o extremadamente complicado, podemos ajustar su diseño para proporcionar una mejor experiencia a los usuarios finales.
3. Fomentar el aprendizaje y el bienestar: Al entender cómo los retos afectan a los usuarios, podemos asegurarnos de que la plataforma no solo sea efectiva desde un punto de vista educativo, sino que también promueva el bienestar emocional de quienes la utilizan y evitar frustraciones y abandonos por estrés.

## **4.2 Variables**

Las variables que se propusieron medir son las propuestas a continuación.

**Pulsaciones (Frecuencia Cardíaca):** El ritmo de nuestro corazón refleja directamente nuestras emociones. Cuando sentimos nerviosismo o estrés, nuestro corazón late más rápido; en cambio, cuando estamos tranquilos, su ritmo se desacelera. Al medir las pulsaciones, podemos identificar cómo los usuarios responden ante situaciones de estrés o presión. Esto nos ayuda a entender su nivel de activación emocional y su capacidad para manejar la aplicación.

**Conductividad de la piel (Respuesta Galvánica):** Nuestra piel también nos da mucha información. Cuando experimentamos emociones intensas, como ansiedad o

nerviosismo, las glándulas sudoríparas producen pequeñas cantidades de sudor sin que nos demos cuenta. Esto altera la conductividad eléctrica de nuestra piel. Al medir este parámetro, podemos detectar cambios sutiles en el estado emocional de los estudiantes, lo que nos proporciona una visión más completa de cómo enfrentan los desafíos a nivel fisiológico.

Nuestro objetivo es utilizar estas dos variables para analizar, de manera objetiva y no invasiva, cómo los alumnos manejan el estrés y el nerviosismo. Estos datos son valiosos para entender el impacto emocional de las actividades propuestas a los usuarios del proyecto. Así que cruzando estos datos podemos determinar si una persona está sufriendo situaciones de estrés o no.

### **4.3 Estudio de los datos**

El objetivo del estudio de los datos recolectados es detectar momentos de estrés donde un usuario sufre dificultades para resolver un determinado ejercicio en BlockLeap o también para detectar frustración del usuario a la hora de navegar por la web en caso de que no sea lo suficientemente accesible y poder mejorarla de cara al futuro.

Para realizar el estudio de los datos recopilados hemos utilizado la aplicación Visual Studio Code y creado un script en python para analizar los datos. En estos datos hemos buscado y detectado anomalías. Estas anomalías son irregularidades en los datos que se desvían de la tendencia general. Hemos utilizado la librería de detección de anomalías *statsmodel*<sup>14</sup> así como las librerías *numpy*<sup>15</sup>, *pandas*<sup>16</sup> y *matplotlib*<sup>17</sup>.

---

<sup>14</sup> [Statmodels](#)

<sup>15</sup> [Numpy](#)

<sup>16</sup> [Pandas](#)

<sup>17</sup> [Matplotlib](#)

En el código lo que hacemos es cargar los datos generados por la placa *Emotibit* mediante un archivo .csv, estos datos son de pulsaciones (HR, heart rate) y actividad electrodérmica (EA, electrodermal activity). Para poder sacar conclusiones en claro tenemos que analizar los datos como conjunto y no individualmente por lo que hemos unido los datos en base a la hora de recolección. Como de EA se recolectaron 15 muestras por segundo y de HR 25 muestras por segundo, hemos eliminado las filas en donde los valores de HR no coincidían con una muestra de EA.

Tras esto creamos una función para eliminar *outliers* –que son valores distantes del resto de datos– que no tienen sentido dentro de nuestra muestra de datos, como pueden ser contaminaciones en los datos de HR como un pico de 180 cuando el usuario presentaba en realidad unas pulsaciones de 80.

De la librería *statsmodel* hemos utilizado la función *seasonal\_decompos*, que sirve para descomponer la serie temporal en: tendencia, estacionalidad y ruido. La tendencia es la evolución general de los datos a lo largo del tiempo; la estacionalidad son patrones que se repiten periódicamente; el ruido es lo que queda después de eliminar la tendencia y la estacionalidad, es decir, la parte irregular de la señal.

Una anomalía se detecta usando el ruido de la descomposición estacional del promedio de HR y EA y se considera una anomalía si su valor absoluto es mayor que un umbral que es 3 veces la desviación estándar de los residuos.

Una vez tenemos los datos preparados podemos empezar a analizarlos, agrupamos las anomalías detectadas que están muy cercanas en el tiempo como una única anomalía y ya estamos listos para graficar los resultados y ver los momentos de nerviosismo que buscamos, el algoritmo nos generará gráficas donde podremos ver en qué momento de la recolección ha habido una anomalía.

Formato de los datos recopilados con *EmotiBit* después de pasarlos por “Emotibit Parser” listos para analizar. (Figura 33)

	A	B	C	D	E	F	G	H
1	LocalTimestamp,	EmotiBitTimestamp,	PacketNumber,	DataLength,	TypeTag,	ProtocolVersion,	DataReliability,	EA
2	1731510874.163566,	114996.000,	16307,1,	EA,1,	100,4.	371040		
3	1731510874.230063,	115062.500,	16324,2,	EA,1,	100,4.	174344		
4	1731510874.296560,	115129.000,	16324,2,	EA,1,	100,4.	362796		
5	1731510874.363557,	115196.000,	16340,1,	EA,1,	100,4.	404223		
6	1731510874.430054,	115262.500,	16357,2,	EA,1,	100,4.	184372		
7	1731510874.496551,	115329.000,	16357,2,	EA,1,	100,4.	219914		
8	1731510874.563548,	115396.000,	16372,2,	EA,1,	100,4.	240485		
9	1731510874.630545,	115463.000,	16372,2,	EA,1,	100,4.	174344		

Figura 33. Datos capturados por *Emotibit*

#### 4.4 Primera recolección de datos y comprobaciones del algoritmo

Se realizó una primera recolección de datos para probar el prototipo del algoritmo. Al ser la primera vez, en los datos detectamos gran cantidad de ruido al no estar bien colocada la placa en el dedo y estar en el dedo utilizado para manejar el ratón del ordenador. Por lo que estos datos fueron descartados al probarlos con el algoritmo de detección de anomalías.

Las siguientes recolecciones se realizaron colocando mejor la placa *EmotiBit* pero con cantidad de estímulos externos, móvil, música y gente entrando en la estancia de la prueba. Al probar el algoritmo de detección de anomalías salieron datos muy dispares en momentos en los que sí que había surgido algún estímulo externo.

Como se puede observar en el gráfico de la Figura 34 detectamos anomalías en puntos donde podemos corroborar que ha habido estímulos durante la prueba. En la gráfica se pueden observar los datos combinados de HR y EA. Estos datos los mostramos en el gráfico unidos en su media combinada en función del tiempo,

teniendo en cuenta que los datos de EA se recogen con un retardo con respecto a los datos de HR.

En estos puntos detectados, ocurrieron situaciones como llamadas telefónicas, gente entrando en la sala de la prueba y mucho movimiento del usuario que realizó la prueba.

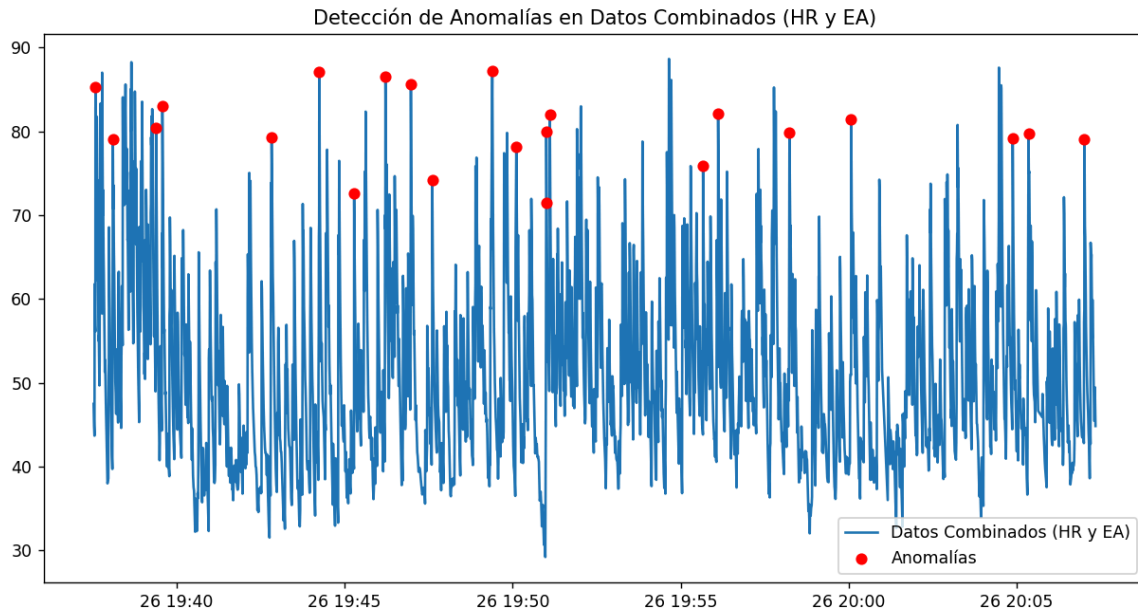


Figura 34. Primeros datos analizados de Emotibit, donde el eje X es el día la hora y el minuto y el eje Y la media combinada de HR y EA.



# Capítulo 5 - Pruebas con usuarios

## 5.1 Pruebas con usuarios *Emotibit*

El objetivo principal de esta etapa fue analizar la calidad, variabilidad y fiabilidad de los datos capturados durante la prueba, así como evaluar el comportamiento de la placa en entornos de uso real.

### 5.1.1 Recolección de datos usuarios

Para esta prueba se invitó a participar a diferentes usuarios, con perfiles variados en cuanto a edad y familiaridad con herramientas de programación. En el espectro de población para la prueba contamos con 4 usuarios de entre 24 y 55 años, 2 familiarizados con la programación y uso de webs y 2 con poca o nula familiaridad con la programación y uso de webs.

En esta fase de las pruebas utilizamos los niveles oficiales de BlockLeap: niveles introductorios como uso de variables y uso de bucles.

Los usuarios voluntarios que se presentaron para la prueba utilizaron también los niveles citados anteriormente.

Cada participante interactuó con el juego bajo las mismas condiciones, que son:

- Navegación por menús: para captar señales en momentos de baja carga cognitiva.
- Juego de niveles progresivos: desde niveles introductorios hasta niveles más complejos que requerían lógica, precisión y toma de decisiones.
- Momentos de pausa o reflexión: donde el jugador se detenía a pensar en cómo resolver un problema.

### 5.1.2 Análisis de los datos recopilados

Con todos estos datos recopilados hemos podido hacer un estudio para cada tipo de usuario centrándonos en la fase de juego –de los niveles citados– que pasamos a detallar a continuación.

- Usuarios familiarizados con la programación: Figura 35

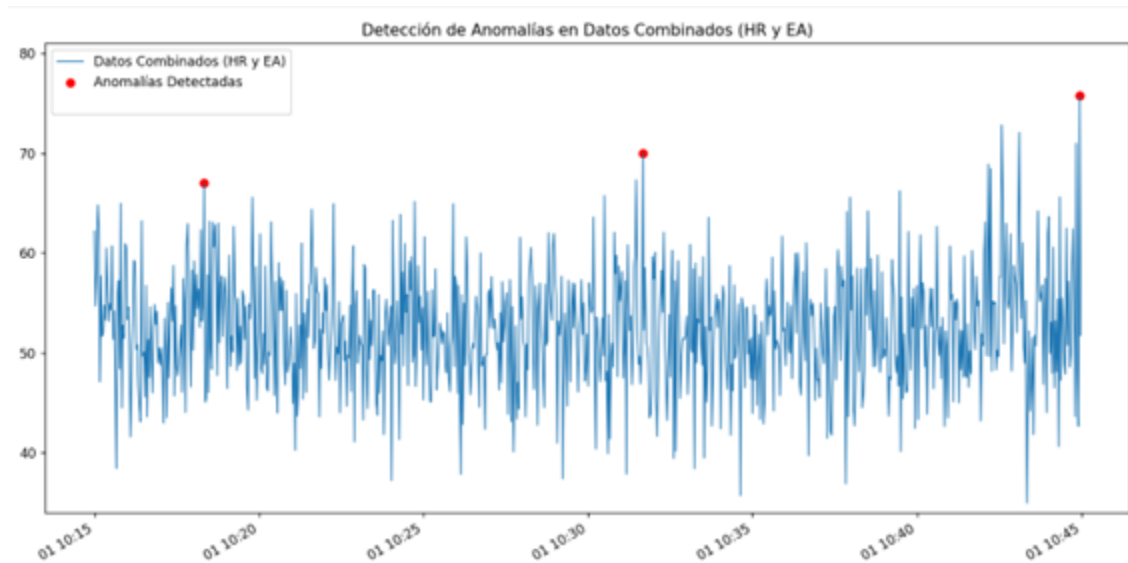


Figura 35. Gráfico usuario con buenas habilidades programando

Para un usuario con buenas habilidades programando hemos detectado que no se presentan anomalías de estrés más que unos pequeños momentos que podrían ser *outliers*, esto se pudo ver visualmente durante la prueba ya que los usuarios de este espectro de población no tuvieron ningún problema para resolver el conjunto total de los niveles planteados. La ausencia de anomalías puede estar indicándonos que este usuario está aburrido y que no tiene ninguna dificultad para resolver los problemas planteados.

- Usuario no familiarizado con la programación: Figura 36

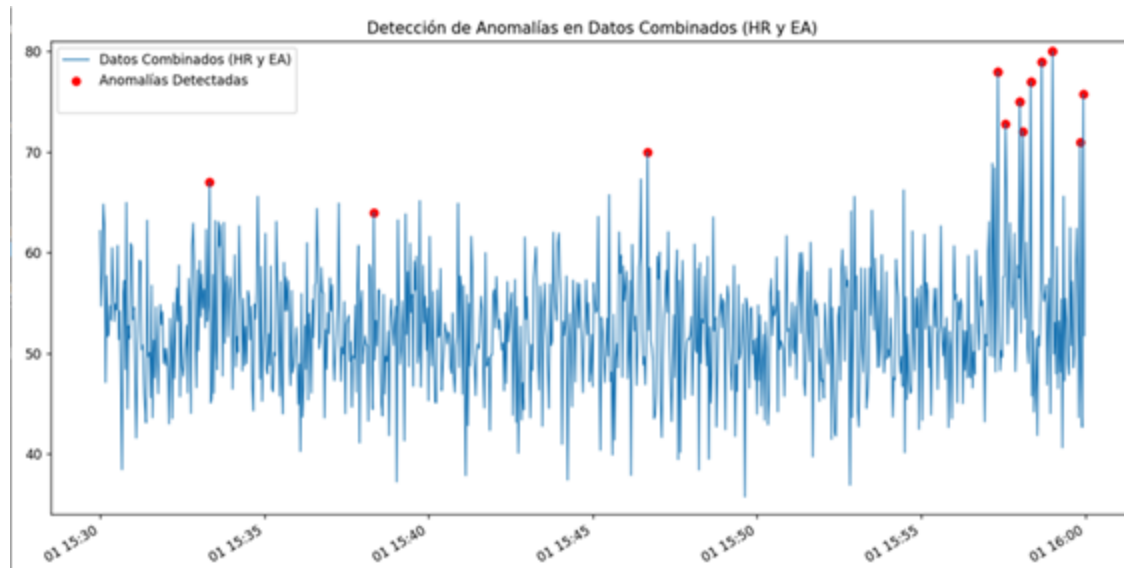


Figura 36. Gráfico usuario con ninguna habilidad programando.

Para un usuario con pocas o nulas habilidades programando hemos detectado que al principio de la prueba no tuvo muchos problemas a la hora de resolver los niveles planteados, más que unos posibles *outliers*, ya que desde un punto de vista visual no se apreció ninguna dificultad para resolver los primeros niveles introductorios. En el último nivel planteado en la categoría de uso de variables pudimos detectar visualmente que el usuario no pudo completar el nivel y pidió ayuda. Esto queda reflejado en el gráfico a partir de la hora 15:55 que fue cuando comenzó el usuario a resolver el problema.

Esto puede suponer que, para un usuario con poca familiaridad con la programación, la curva de aprendizaje planteada en esta fase de pruebas no ha sido la adecuada ya que no ha sido capaz de resolver todos los niveles. Por lo que habría que modificar la curva de aprendizaje de BlockLeap para usuarios más nuevos.

### 5.1.3 Conclusiones de la prueba

Podemos concluir que, los niveles no tienen una buena curva de aprendizaje para los usuarios con pocas o nulas habilidades de programación ya que para estos usuarios se ha detectado estrés a la hora de resolver ciertos problemas. Y para los usuarios más avanzados hemos detectado un pico de aburrimiento resolviendo los niveles.

Por último, podemos decir que este análisis ofrece una base sólida para crear un sistema adaptativo que, en el futuro, pueda modificar la dificultad del juego según el estado emocional o fisiológico del jugador. De esta manera, se promoverá un entorno de aprendizaje más personalizado y efectivo.

## **5.2 Pruebas con usuarios en BlockLeap**

Con estas pruebas, lo que buscábamos era información de usuarios reales que han probado a fondo cada una de las funcionalidades de la web BlockLeap. Estos usuarios han sido gente de la universidad y familiares. Ha habido un total de 14 personas que han probado la aplicación y rellenado el cuestionario. La forma en la que se han hecho ha sido la siguiente:

Se les pedía abrir la aplicación y jugar niveles desde el primer nivel oficial y hasta que no supieran cómo continuar, en ese caso se les indicaba, si aún no la habían visto, donde se situaba la pestaña de tutoriales.

También se les pidió crear un nivel con el editor, para el cual tenían que registrar un usuario previamente, desde la comunidad jugar a este mismo nivel creado, y volver a editarlo para hacer nuevos cambios.

También se les pidió entrar en una clase mediante un código que les facilitamos, y navegar por la misma clase, todo esto dándoles la menor información posible sobre cómo llevarlo a cabo para que sus respuestas sean lo más intuitivas posibles.

De acuerdo con los datos recogidos, hemos llegado a la conclusión sobre varios temas.

### **5.2.1 Usabilidad de la aplicación**

Hemos medido la usabilidad e intuitividad de la aplicación a partir de las respuestas e interacciones de los usuarios después de probar la aplicación. Según los

comentarios de los usuarios, la web es muy intuitiva, fácil de navegar, pero sí que se han propuesto mejoras respecto a esta. Comentan que sería una mejora indicar que no se ha podido iniciar sesión, al igual que señalar de alguna manera que está iniciada la sesión (con una foto de perfil, por ejemplo). Nos han comentado también que faltan indicaciones en la creación de niveles para saber el funcionamiento de algunos objetos o lo que representan.

Con el tema de las clases y acceso a estas no se muestran problemas (Figura 37 última sección). Los únicos comentarios recibidos sobre este apartado es la petición de añadir un ranking de alumnos dentro de la clase.

Cómo de acuerdo estás con las siguientes afirmaciones siendo 1 Muy en desacuerdo y 5 Muy de acuerdo.

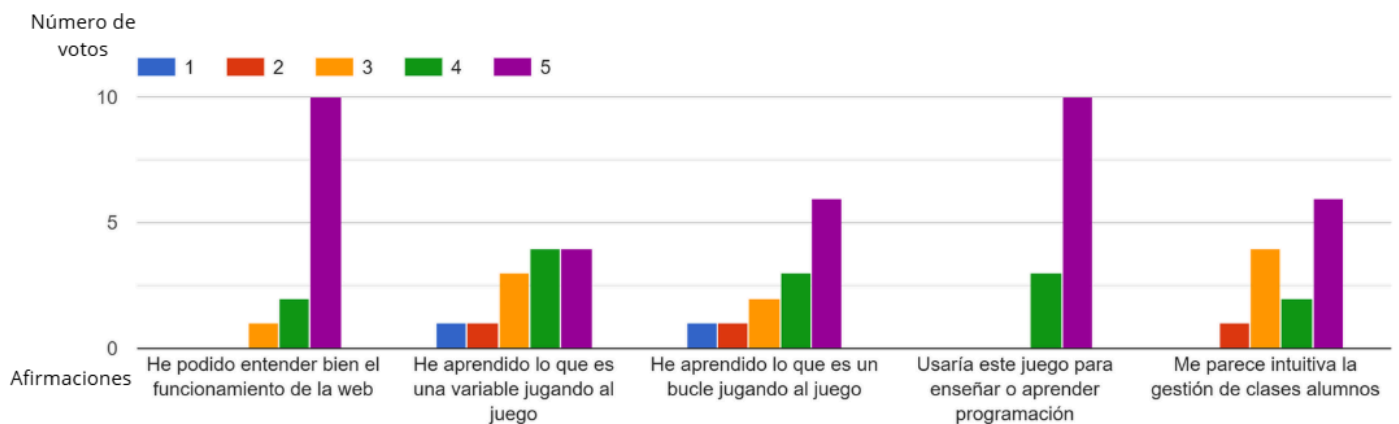


Figura 37. Encuesta usabilidad y aprendizaje del juego

### 5.2.2 Aprendizaje gracias a la aplicación

Hemos querido medir lo aprendido gracias a la aplicación, y hemos sacado los resultados explicados a continuación.

La mayoría de los probadores del juego tienen una experiencia en programación medio-alta (Figura 38), sólo una persona tiene experiencia baja, por lo que la mayoría de los probadores están familiarizados con los conceptos que se

enseñan en la aplicación.

Además, más de la mitad ha usado aplicaciones o juegos semejantes al que proponemos, lo cual lo hace más asequible e intuitivo (Figura 39).

Experiencia previa con la programación  
14 respuestas

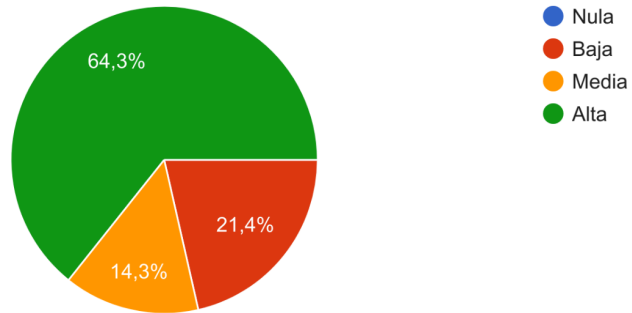


Figura 38. Experiencia de los usuarios

¿Has usado antes aplicaciones con lenguajes por bloques, como por ejemplo Scratch?  
14 respuestas

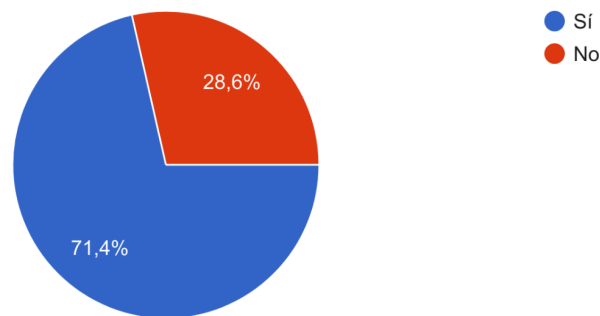


Figura 39. Uso de aplicaciones webs parecidas

En la encuesta realizada hemos preguntado sobre tipos de niveles y condiciones dadas en el juego para hacer un sondeo de si ha servido y cuánto. Respecto a la diversión del juego, tal cual se muestra en la Figura 40, nos muestra que los niveles con cofres y con bucles son los más entretenidos para nuestros jugadores. Esto puede deberse en parte a que como se ha comentado anteriormente, la mayoría de las personas que han probado el juego y realizado la encuesta, tienen experiencia previa con la programación, por lo que cuanto más simple, menos entretenido va a resultar.

El entretenimiento y nivel de diversión del juego es importante pues de ello depende que se interesen en seguir aprendiendo y usando el juego.

Valora el nivel de diversión del juego dependiendo de ciertas condiciones del 1 al 5 siendo 1 Poco Entretenido y 5 Muy Entretenido.

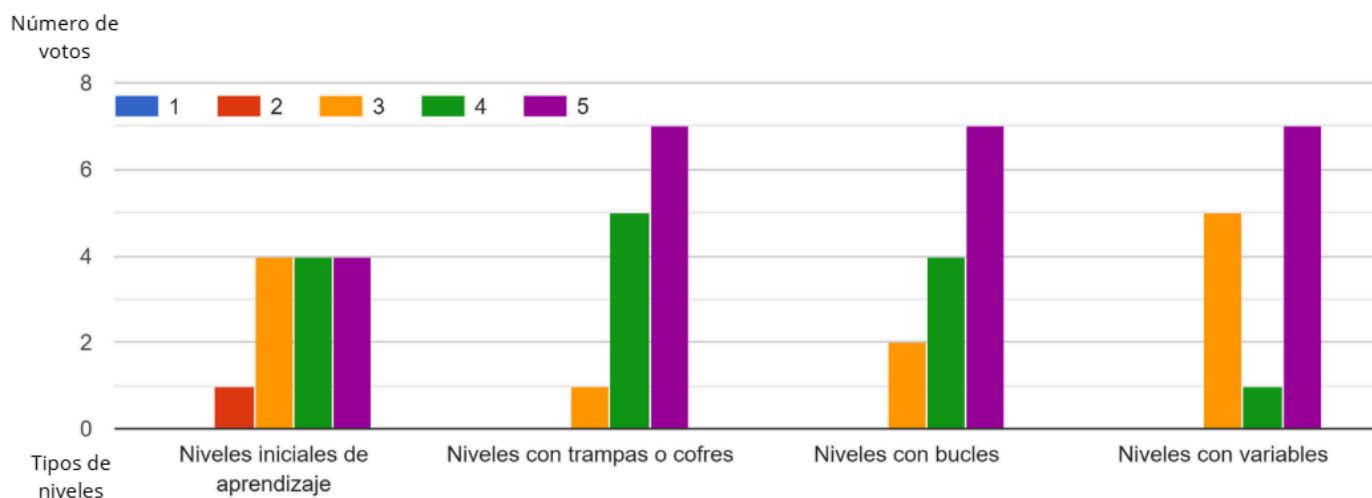


Figura 40. Nivel de entretenimiento del juego

Por otro lado, la mayoría de los niveles jugados no han presentado mucha dificultad para la mayoría de los jugadores, excepto los niveles específicos que requieren un conocimiento mínimo de conocimiento informático, como por ejemplo los niveles de resolución con bucles o variables obligatorias. También han supuesto

algunas dificultades el conseguir las 3 estrellas en algunos niveles, por los criterios de los propios niveles (Figura 41 y Figura 42).

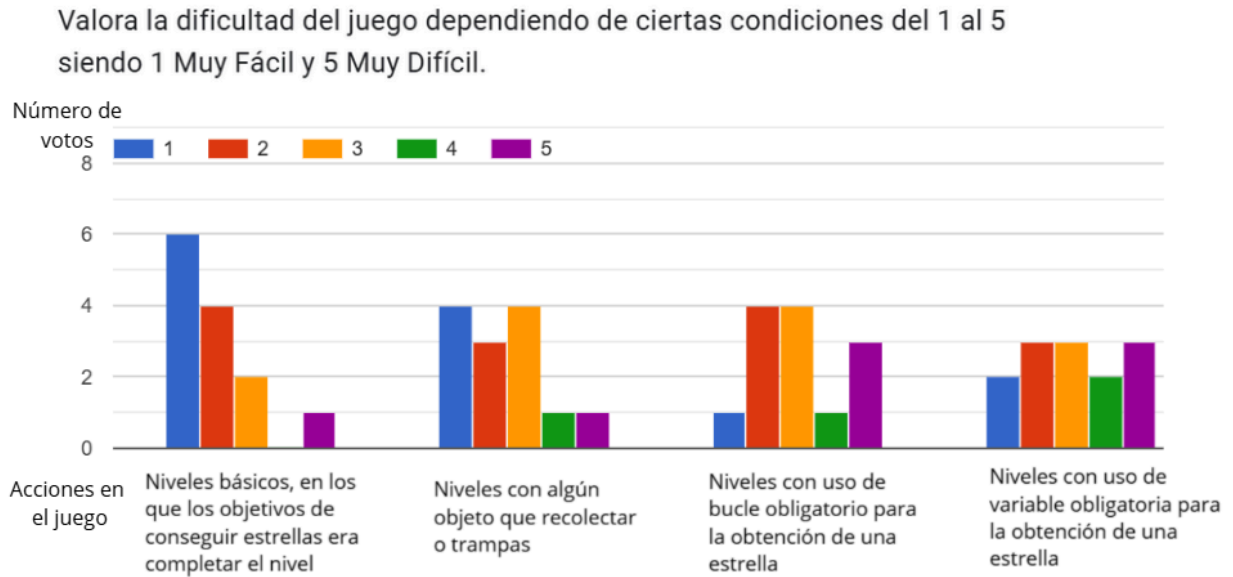


Figura 41. Dificultad del juego

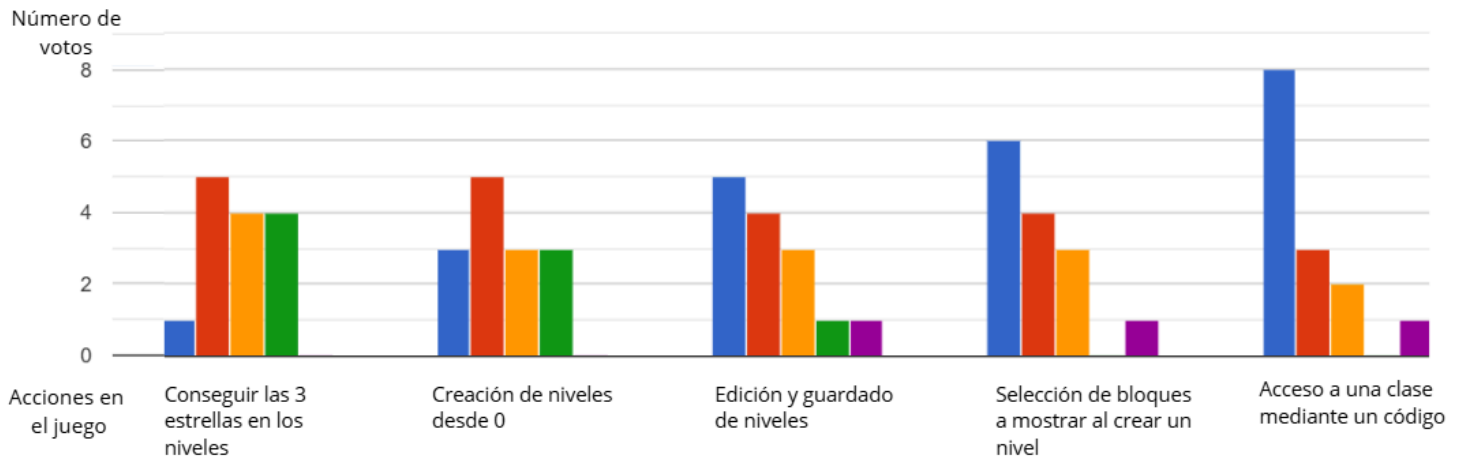


Figura 42. Dificultad del juego-2

### 5.2.3 Opiniones de la aplicación

Por último, hemos pedido opiniones generales de la aplicación y, en su mayoría, son opiniones positivas como se muestra en la Figura 43

En una escala del 1 al 5, siendo 1 Poco y 5 Mucho, cómo de entretenido te ha parecido el juego  
14 respuestas

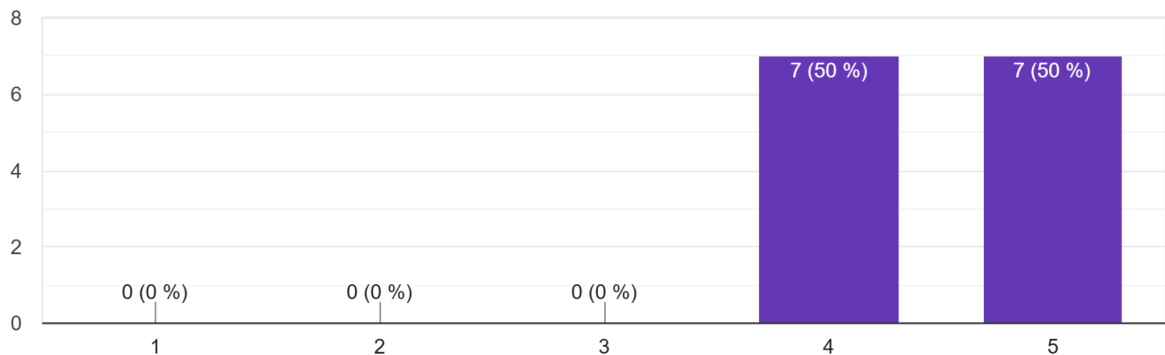


Figura 43. Puntuación de juego

Posteriormente hemos pedido que nos dejen sus opiniones y los cambios que realizarían en la aplicación. La mayoría de los jugadores coinciden en que el juego es muy intuitivo, entretenido y una buena herramienta educativa para introducir a los usuarios, especialmente aquellos con menos conocimientos, en el mundo de la programación.

Respecto a la dificultad, los usuarios principiantes la encuentran adecuada, pero agradecerían tutoriales sobre cómo utilizar variables y bucles.

Teniendo en cuenta su uso en el ámbito educativo, los usuarios que han probado el juego valoran positivamente que los alumnos puedan crear y compartir niveles, y lo ven visto como un incentivo adicional para el aprendizaje.

Algunas sugerencias de mejoras son:

- Añadir más explicaciones o tutoriales para niveles con bucles y variables.
- Mejorar la comunicación de estado, como mensajes al iniciar sesión o al introducir mal las credenciales.
- Ofrecer opciones de dificultad progresiva para usuarios más avanzados.
- Posibilidad de desbloquear niveles. En los niveles de aprendizaje, no permitir acceder a ciertos niveles hasta que se haya completado cierta formación inicial.

En conclusión, el juego ha sido recibido muy positivamente. Se considera una herramienta divertida y eficaz para el aprendizaje de programación, especialmente adecuada para usuarios principiantes o entornos educativos. Se valoran aspectos como la jugabilidad, la personalización de niveles y el enfoque pedagógico, aunque se sugieren pequeñas mejoras en la usabilidad y claridad de ciertas mecánicas.

## Capítulo 6 - Conclusiones y trabajo futuro

En este capítulo vamos a detallar las conclusiones resultantes del trabajo realizado durante el proyecto, además como detallaremos posibles mejoras a implementar en el futuro.

### 6.1 Conclusiones

Este proyecto no ha sido sencillo, ya que hay que tener en cuenta que el proyecto actual partió de la base del trabajo realizado por otros alumnos en el Trabajo de Fin de Grado “Tecnologías web para la creación de una herramienta de enseñanza de conceptos de programación a través de un videojuego” [\[3\]](#). Antes de empezar a implementar mejoras, hemos tenido que comprender la estructura del proyecto y aprender a usar las tecnologías implementadas en el proyecto, así como la forma en la que han sido conectadas estas mismas dentro de la aplicación por el grupo anterior de alumnos.

Consideramos que se han logrado mejoras en la aplicación en relación con los objetivos planteados inicialmente, que detallamos a continuación.

Nos hemos familiarizado con el proyecto de BlockLeap y hemos analizado otros juegos serios para identificar elementos que añadir o mejorar en la aplicación. Una de las consecuencias del análisis fue añadir más funcionalidad a uno de los puntos fuertes de la aplicación, el editor de niveles, lo cual conlleva a facilitar la creación de niveles y permitir que estos tengan más variedad y calidad.

Con el propósito de proporcionar herramientas para el uso de BlockLeap en las aulas, hemos implementado más herramientas en la aplicación que dan más

posibilidad a su uso en las aulas como la sección de classroom. Esta sección permite al profesor gestionar grupos de alumnos en clases y asignar niveles o conjuntos de niveles a las diferentes clases.

Con el objetivo de mejorar la usabilidad de la aplicación hemos implementado, por un lado, diseño responsive a la página web para ajustar la vista de la página a los diferentes dispositivos que nos podemos encontrar en las aulas (principalmente ordenadores de escritorio, portátiles y tablets) y, por otro lado, hemos creado un apartado de tutoriales para que los usuarios puedan aprender a utilizar los diferentes componentes de BlockLeap.

Por último, para la validación de la aplicación hemos podido usar el biosensor Emotibit para recolectar y analizar datos sobre el estrés y aburrimiento que pudieran aparecer mientras los usuarios juegan, pudiendo ajustar los niveles en consecuencia de estos datos y del feedback proporcionado directamente por los usuarios mediante encuestas. Aunque hemos comprobado que es factible usar el biosensor para la toma de datos, hay que matizar que las pruebas que hemos podido realizar han sido escasas. Se necesitan más muestras para validarlo correctamente, por lo que proponemos hacer más de estas pruebas como trabajo a futuro.

Por otra parte, también es cierto que no se han podido desarrollar todas las mejoras que se han identificado como necesarias a lo largo del proyecto. Por lo que en la siguiente sección detallaremos posibles ideas a implementar en el futuro.

## **6.2 Trabajo a futuro**

En esta sección vamos a detallar las posibles mejoras que se podrían implementar. El desglose de mejoras las hemos dividido en 3 subcategorías (BlockLeap, Datos biométricos y Accesibilidad digital) para organizarlo mejor.

## **6.2.1 Trabajo a futuro para BlockLeap**

En esta sección hablaremos de nuevos elementos que se podrían introducir a la plataforma web de BlockLeap, algunos de ellos son ideas que no dieron tiempo a acabar y otras son ideas que no fueron exploradas durante el desarrollo por dar prioridad a otras funcionalidades más relevantes.

### **6.2.1.1 Nuevos bloques de Blockly y elementos de juego**

Uno de los frentes por los que más se puede mejorar la plataforma es la introducción de nuevas mecánicas en el juego serio. Como ya se ha mencionado en el apartado de cambios y mejoras de la aplicación, la introducción de nuevos elementos permitirá enseñar más conceptos de programación y tener más diversidad en los niveles que se quieran crear.

Una de las ideas que se exploraron y que podría implementarse es la introducción de bloques condicionales en el controlador de Blockly. Estos bloques recibirían el estado de un elemento del juego y se usarían en conjunto con bloques lógicos de if y while.

### **6.2.1.2 Seguimiento de progreso de los estudiantes**

Una de las funcionalidades que se pueden contemplar para mejorar el uso de BlockLeap en las aulas es la creación de una sección en la que los profesores puedan ver el progreso de sus alumnos. Podría ser una vista similar a la que tienen Code.org o Code Monkey, permitiendo al profesor conocer información que le permita tener mayor control de la clase, como son los niveles completados por cada uno de sus estudiantes o en qué niveles fallan más.

### **6.2.1.3 Borrado de contenidos**

Actualmente la plataforma no consta de la funcionalidad de borrar ningún tipo de contenido desde el cliente web, con esto nos referimos a que los niveles, los sets de niveles, las clases de alumnos y las cuentas de usuario no se pueden borrar desde la aplicación. Sería oportuno las interfaces correspondientes al borrado de estos elementos para que el usuario pueda eliminarlos en caso de que lo desee.

#### **6.2.1.4 Herramientas de moderación**

Como una plataforma que cuenta con una comunidad, *BlockLeap* debería de contar con herramientas de moderación que permitieran a administradores eliminar contenidos indeseados de la plataforma, como podrían ser niveles con nombres, descripciones o diseños inapropiados para niños (y adultos).

Una posible propuesta sería implementar una interfaz que permitiese al administrador introducir palabras a una "lista negra", provocando que los niveles que contengan estas palabras en su nombre o descripción no puedan ser publicados.

#### **6.2.2 Integración de los datos biométricos**

La placa *EmotiBit* es un dispositivo muy versátil para capturar datos biométricos, y obtener un reflejo de lo que está experimentando una persona. Por ello sería una buena idea capturar datos en tiempo real con la placa y mostrarlos directamente en el juego *BlockLeap*. Sería interesante por ejemplo que los datos de pulsaciones para que el investigador pudiese verlos a la vez que juega. O, por otro lado, sería interesante capturar los datos y mostrar las gráficas generadas con el algoritmo de detección de anomalías una vez se haya completado el nivel acompañado de momentos críticos, por ejemplo, en algún movimiento del jugador o cuando se le da a play en el nivel. Cabe destacar que esto es posible únicamente a nivel de

investigación, ya que, en un escenario real, como es un aula, su implementación no es posible por la normativa vigente debido a la sensibilidad de los datos biométricos [\[10\]](#).

También es una buena idea aplicar una narrativa adaptativa en función del estrés del usuario. Por ejemplo, si el sistema nota un aumento considerable en el nivel de estrés, BlockLeap podría ofrecer un nivel más guiado, con ayudas visuales o instrucciones paso a paso, creando así un ambiente de aprendizaje más tranquilo. Por otro lado, si los datos muestran que el usuario está relajado y concentrado, se podrían desbloquear niveles con desafíos más grandes o introducir mecánicas de programación más complejas, fomentando así un aprendizaje más activo.

Además, se podría detectar movimiento con el giroscopio y con el acelerómetro, que podría detectar aspavientos de un usuario o si está demasiado quieto. E implementaciones más elaboradas con el acelerómetro u con el giroscopio sería implementar un sistema de movimiento con el cual el usuario pudiese manejar el juego con el movimiento de su mano. Por ejemplo, con una sacudida hacia la derecha, hacer que el personaje en el nivel se moviese más rápido y con una sacudida a la izquierda hacer que el personaje se moviese más lento.

### **6.2.3 Trabajo a futuro en Accesibilidad Digital**

Actualmente, la plataforma no cuenta con un sistema específico de accesibilidad para personas con discapacidad. Sin embargo, sería recomendable que en el futuro se incorporen mejoras para hacerla más inclusiva y accesible para todos los usuarios, independientemente de sus capacidades. Garantizar una experiencia accesible para personas con discapacidad visual, auditiva, física o intelectual podría mejorar significativamente la usabilidad y alcance de la plataforma.

Algunas posibles mejoras que se podrían implementar son:

1. Compatibilidad con lectores de pantalla: Se recomienda asegurar que el contenido de la plataforma sea completamente accesible para personas con discapacidad visual. Esto puede lograrse utilizando etiquetas *aria* (Accessible Rich Internet Applications) y una correcta estructura semántica de los elementos HTML, lo que permitiría una navegación eficaz mediante lectores de pantalla como NVDA o JAWS.
2. Controles de teclado mejorados: Sería beneficioso permitir la navegación completa por teclado, lo cual facilitaría su uso a personas con movilidad reducida.
3. Subtítulos y descripciones auditivas: Para hacer la plataforma más accesible para personas con discapacidad auditiva, se recomienda incluir subtítulos en los contenidos multimedia, así como descripciones auditivas para facilitar la comprensión del contenido a personas ciegas o con visión reducida.
4. Contraste y colores accesibles: Es recomendable mejorar el contraste entre el texto y el fondo, considerando especialmente las necesidades de personas con daltonismo o discapacidad visual. Además, se podría incluir una opción para que los usuarios personalicen la paleta de colores según sus necesidades visuales.

Para facilitar la implementación de estas mejoras, se pueden utilizar varias herramientas de evaluación de accesibilidad, tales como:

- WAVE (Web Accessibility Evaluation Tool): Esta herramienta puede ser útil para identificar problemas de accesibilidad en la web, mostrando errores y advertencias visualmente en la página.

- Colour Contrast Analyzer: Sería beneficioso usar esta herramienta para comprobar la relación de contraste entre el texto y el fondo, asegurando que cumpla con las directrices de accesibilidad.
- AXE: Utilizar esta extensión de navegador puede ser útil para realizar auditorías de accesibilidad y recibir sugerencias sobre cómo mejorar la conformidad con las pautas WCAG (Web Content Accessibility Guidelines).

Además, como referencia, el Libro Blanco de Accesibilidad para Desarrolladores de la Fundación ONCE podría ser consultado para obtener una guía detallada sobre cómo integrar la accesibilidad en el desarrollo web, ayudando a cumplir con las normativas legales y técnicas en materia de accesibilidad digital.

Incorporar estas recomendaciones permitiría que la plataforma sea más inclusiva y conforme con las normativas internacionales de accesibilidad, como las pautas WCAG 2.1, asegurando una experiencia óptima para todos los usuarios, independientemente de sus capacidades.



# Introduction

## Motivation

As technology becomes increasingly integrated into various areas of society, the need to incorporate basic training in programming and computational thinking into educational systems around the world is also growing [\[1\]](#). This competency not only prepares students for an increasingly digital work environment but also develops essential skills such as problem-solving, logical reasoning, and creativity.

In this context, Serious Games –video games designed with a purpose beyond entertainment– are presented as an effective tool to teach technical concepts in an attractive, dynamic and motivating way [\[2\]](#). These types of games, used in fields such as education, health or research, combine playful mechanics with educational contents, encouraging learning through interactive experience and the game.

Thus, the present project proposes the continuation of the development of BlockLeap [\[3\]](#), an educational platform based on a Serious Game that allows users to learn the basics of programming by solving interactive challenges. BlockLeap is inspired by existing solutions such as Articoding, an educational tool focused on learning programming through blocks. However, one of the main limitations of Articoding is its limited adaptability in school environments due to its technical requirements, which make it difficult to use on modest equipment or without the capacity to install additional software. In response to this problem, BlockLeap is conceived as a platform accessible directly from the browser, without the need for installation and optimised to work on devices with limited resources, thus facilitating its integration in real educational contexts.

## Objetives

The main objective of the project is to improve BlockLeap, a web platform for young people to develop their computational thinking and learn basic programming concepts. It is proposed that the platform can be used by teachers in the classroom, providing the necessary tools to do so. To achieve this goal, the project proposes a series of steps:

- Analyze existing Serious Games and solutions that addresses computational thinking and programming, with the aim of identifying key elements that should be addressed as part of this project.
- Analyze the existing version of BlockLeap to carry out technological appropriation to facilitate the development of new features and analyze the limitations of the existing solution.
- Ensure the usability of the platform in educational environments, so that it can run on the often low-performance, old or offline computers that are usually found in schools.
- Provide educators with a set of tools that allow them to use BlockLeap in the classroom to teach basic programming concepts, collect feedback and evaluate students' performance, as well as any difficulties they may find during the teaching process.
- Study through bio-signals whether the game levels are boring or if they cause stress in the users.

## Work plan

At the beginning of the project, we drew up a work plan structured in phases, as shown in Figure 1. This planning allows us to organise the tasks and estimate the time required for each stage of the project.

The planning consists of the following phases:

- Project status study: The aim of this phase is to become familiar with the existing software (implemented functionalities, architecture and code) in order to gain knowledge about it and to facilitate further development.
- Serious Games Research: This phase consists of looking for functionalities of online communities, classes and tutorials on other platforms to incorporate them into our software or improve the existing ones..
- Research and testing with biosensors: This phase focuses on analysing hardware capable of measuring the physiological response of users and studying its application in the project to check if the game levels are boring or if they generate stress in the users.
- Development of functionalities: In this phase the objective is the implementation of new functionalities for the platform, as well as fixing or improving some of the existing functionalities.
- Validation: This phase consists of testing with external users in order to evaluate the application and analyze the data collected with the biosensors.

	Septiembre	octubre	noviembre	diciembre	enero	febrero	marzo	abril	mayo
Estudio del estado del proyecto	Orange	Orange							
Investigacion Juegos serios		Yellow							
Investigación y pruebas con biosensores			Red	Red					
Desarrollo de la aplicación			Magenta	Magenta	Magenta	Magenta	Magenta	Magenta	
Validacion								Blue	

Figure 1. Gantt diagram of the work plan

### **Project status study**

The main objective of this phase will be to become familiar with the legacy BlockLeap project. This phase will be essential to understand the structure, operation and limitations of the software on which the project is based. It will allow the team to thoroughly understand how the application works and to explore its components in order to start extending and reusing them for the development of new functionalities.

### **Serious games research**

During this phase, research will focus on educational serious games and other community-based platforms such as Scratch and Kahoot, with the aim of understanding their structure, their educational applicability and their potential for teaching programming.

One of the sections that we will take most into account in the analysis will be to assess the social and collaborative component of these platforms, because the community part is especially relevant to the focus of this Final Degree Project, as one of the objectives of this is the creation of an open community on our platform.

### ***Research and testing with biosensors***

During the development of the project, the need to test the levels created became apparent. The question also arises as to what physiological response users may have when confronted with these levels and how to relate these responses to the difficulty of the levels.

That is why the idea of using a hardware device that can collect biometric data such as heart rate and electrodermal activity has emerged.

### ***Development of functionalities***

This phase is the most extensive and demanding of the project, as it involves improving and extending an existing web platform, perfecting its operation, performance and features. On the basis of the research of serious games, we will define the key mechanics that will facilitate the learning of programming.

The platform already incorporates a level editor, so for this stage we plan to improve this component to provide it with greater versatility and accessibility. In addition, improvements will be made to the general structure of the application, the user interface, the game logic and its integration with the community in order to offer a more complete experience.

### ***Validation***

The validation phase will focus on testing with end users in order to evaluate the effectiveness of the web platform. The objective of these tests will be to collect direct feedback, identify possible areas for improvement and adjust both the learning experience and the difficulty of the levels.

Furthermore, the results obtained in this phase will allow establishing a solid basis

for recommendations and adjustments that can be developed in the future to further improve the application.

# Conclusions and future work

In this chapter we will detail the conclusions resulting from the work carried out during the project, as well as detailing possible improvements to be implemented in the future.

## Conclusions

This project has not been easy, since it must be taken into account that the current project was based on the work done by other students in the Final Degree Project "Web technologies for the creation of a tool for teaching programming concepts through a video game"[\[3\]](#). Before starting to implement improvements, the project development team had to understand the structure of the project and learn how to use the technologies implemented in the project, as well as the way in which they have been connected within the application by the previous group of students.

We consider that improvements have been achieved in the application in relation to the objectives initially set out, which we detail below.

We have familiarized ourselves with the BlockLeap project and we have analyzed other serious games to identify elements to add or improve in the application. One of the consequences of the analysis was adding more functionality to one of the strengths of the application, the level editor, which facilitates the creation of levels and allows them to have more variety and quality.

With the aim of providing tools for the use of BlockLeap in classrooms, we have implemented more tools in the application that give more possibility to its use in classrooms such as the classroom section. This section allows the teacher to manage groups of students in classes and assign levels or sets of levels to different classes.

In order to improve the usability of the application, on the one hand, we have implemented responsive design to the website to adjust the view of the page to the different devices that we can find in the classroom (mainly desktops, laptops and tablets), and on the other hand, we have created a tutorial section so that users can learn how to use the different components of BlockLeap.

It is also true that it has not been possible to develop all the improvements that have emerged throughout the development. So in the following section we will detail possible ideas to implement in the future.

## **Future work**

In this section we will detail the possible improvements that could be implemented. The breakdown of improvements has been divided into 3 subcategories (BlockLeap, *Emotibit* and Digital Accessibility) for better organization.

### ***Future work for BlockLeap***

In this section we will talk about new elements that could be introduced to the BlockLeap web platform, some of them are ideas that we did not have time to finish and others are ideas that were not explored during development.

#### **New Blockly blocks and game elements**

One of the fronts on which the platform can be improved the most is the introduction of new mechanics in the serious game. As already mentioned in the section on changes and improvements to the application, the introduction of new elements will make it possible to teach more programming concepts and have more diversity in the levels you want to create.

One of the ideas that was explored and could be implemented is the introduction of conditional blocks in the Blockly controller. These blocks would receive the state of a game element and would be used in conjunction with if and while logic blocks.

### **Tracking student progress**

One of the functionalities that can be contemplated to improve the use of BlockLeap in classrooms is the creation of a section where teachers could see the progress statistics of their students. This view could be similar to the ones seen in Code.org or Code Monkey, allowing teachers to know information that allows him to have more control of the class, such as the levels completed by each of his students or in which levels they fail the most.

### **Content removal**

Currently the platform does not have the functionality to delete any type of content from the web client, by this we mean that levels, level sets, student classes and user accounts cannot be deleted from the application. It would be appropriate to have the corresponding interfaces to delete these elements so that the user can delete them in case he/she wishes to do so.

### **Moderation tools**

As a community-based platform, BlockLeap should have moderation tools that allow administrators to remove unwanted content from the platform, such as levels with names, descriptions or designs inappropriate for children (and adults).

One possible proposal would be to implement an interface that would allow the administrator to enter words to a "blacklist", causing levels containing these words in their name or description to be unpublishable.

## **Future work with Emotibit**

The *EmotiBit* badge is a very versatile device to capture biometric data, and get a reflection of what a person is experiencing. It would therefore be a good idea to capture real-time data with the board and display it directly in the BlockLeap game. It would be interesting for example to have the keystroke data so that the user could see it while playing the game. Or on the other hand, it would be interesting to capture the data and display the graphs generated with the anomaly detection algorithm once the level is completed accompanied by critical moments for example, in some movement of the player or when you hit play on the level. It should be noted that this is possible only at the research level, since, in a real scenario, such as a classroom, its implementation is not possible due to current regulations because of the sensitivity of biometric data [\[10\]](#).

It is also a good idea to apply an adaptive narrative based on the user's stress. For example, if the system notices a considerable increase in the stress level, BlockLeap could offer a more guided level, with visual aids or step-by-step instructions, thus creating a calmer learning environment. On the other hand, if the data shows that the user is relaxed and focused, more challenging levels could be unlocked or more complex programming mechanics could be introduced, encouraging more active learning.

In addition, movement could be detected with the gyroscope and the accelerometer, which could detect the user's movements or if the user is too still. And already more elaborate implementations with the accelerometer or the gyroscope would be to implement a motion system with which the user could operate the game with the movement of his hand. For example, with a shake to the right, make the

character in the level move faster and with a shake to the left make the character move slower.

### ***Future work in Digital Accessibility***

Currently, the platform does not have a specific accessibility system for people with disabilities. However, it would be highly recommended that in the future improvements are incorporated to make it more inclusive and accessible to all users, regardless of their abilities. Ensuring an accessible experience for people with visual, hearing, motor or cognitive disabilities could significantly improve the usability and reach of the platform.

Some possible improvements that could be implemented are:

1. Compatibility with screen readers: it is recommended to ensure that the platform content is fully accessible for visually impaired people. This can be achieved by using aria (Accessible Rich Internet Applications) tags and a correct semantic structure of the HTML elements, which would allow efficient navigation through screen readers such as NVDA or JAWS.
2. Improved keyboard controls: It would be beneficial to allow full keyboard navigation, which would facilitate its use by people with reduced mobility. In addition, it is suggested to include customizable options for keyboard navigation, adapting to the specific needs of each user.
3. Subtitles and audio descriptions: In order to make the platform more accessible for people with hearing disabilities, it is recommended to include subtitles in the multimedia contents, as well as audio descriptions to facilitate the understanding of the content for blind or visually impaired people.

4. Contrast and accessible colors: It is advisable to improve the contrast between text and background, especially considering the needs of people with color blindness or visual impairments. In addition, an option for users to customize the color palette according to their visual needs could be included.

To facilitate the implementation of these improvements, several accessibility evaluation tools can be used, such as:

- WAVE (Web Accessibility Evaluation Tool): This tool can be useful for identifying accessibility issues on the web by displaying errors and warnings visually on the page.
- Colour Contrast Analyzer: It would be beneficial to use this tool to check the contrast ratio between text and background, ensuring that it meets accessibility guidelines.
- AXE: Using this browser extension can be useful for performing accessibility audits and receiving suggestions on how to improve compliance with WCAG (Web Content Accessibility Guidelines).

Also, as a reference, the ONCE Foundation's White Paper on Accessibility for Developers could be consulted for detailed guidance on how to integrate accessibility into web development, helping to comply with legal and technical regulations on digital accessibility.

Incorporating these recommendations would make the platform more inclusive and compliant with international accessibility standards, such as the WCAG 2.1 guidelines, ensuring an optimal experience for all users, regardless of their capability

# Contribuciones Personales

Este proyecto ha sido desarrollado en equipo por cuatro integrantes, con una colaboración constante y activa por parte de todos. Desde el inicio, hemos trabajado de forma conjunta en las distintas fases del proyecto —planificación, diseño, desarrollo y documentación—, aportando ideas, resolviendo problemas y tomando decisiones en común. Si bien cada miembro ha tenido un grado de implicación diferente en determinadas tareas, la participación ha sido global y compartida en la mayoría de los aspectos del trabajo.

## Jie Qiang Guo

- Selección de bloques en niveles personalizados

Implementación de un menú dentro del editor de niveles que permite escoger qué bloques están disponibles, o en su defecto limitar el número de veces que se puede usar cada uno de los bloques.

- Ampliación de la base de datos

Adición de esquemas de datos relativos a las nuevas funcionalidades que se han implementado. Así como modificación de los esquemas existentes para acomodar las nuevas relaciones.

- Paginación de niveles

Ajuste de las consultas `sequelize` para permitir la carga paginada de los niveles de la comunidad, así como los controles en el cliente para navegar entre las páginas de niveles.

- Filtros de búsqueda

Implementación de etiquetas de categoría para los niveles de la comunidad, así como de una barra de búsqueda que permite filtrar por estas mismas etiquetas.

- Códigos de clase

Implementación de la generación de códigos únicos para unirse a una clase.

- Creación y modificación de DAOs, Controllers y routers

Mantenimiento de la estructura del servidor backend, añadiendo o modificando componentes para la implementación de nuevas funcionalidades.

- Diseño e implementación de nuevos elementos de juego

Conceptualización de mejoras y elementos nuevos en el juego serio. Así como la implementación de ciertos de estos elementos.

- Testing y arreglo de errores

Realización de pruebas manuales de la aplicación para la detección de errores y la implementación de su posterior corrección.

- Apoyo al resto del equipo

Guiado y ayuda a la implementación al resto del equipo a lo largo de todas las funcionalidades durante el desarrollo de la aplicación.

- Investigación de juegos serios

Estudio de las características de otras plataformas educativas con juegos serios como Code.org y CodeCombat.

- Carga y modificación de tutoriales

Transformación del apartado de tutoriales para permitir su carga y guardado en la base de datos de la aplicación. Así como de un editor que permite modificar el texto de los tutoriales.

- Exploración de herramientas

Estudio de las herramientas presentes originalmente en el proyecto (Blockly, Phaser...) como de las herramientas implementadas (o descartadas) durante el nuevo desarrollo del proyecto (Marked, Select2...)

## **Carlos Valera Sansano**

- Investigación juegos serios

Investigación de Kodu Game Lab, juego serio del que se extrajeron varias ideas para implementar en BlockLeap.

- Creación de bocetos Figma

Creación de bocetos con la aplicación web Figma en la que se crearon bocetos de alto nivel para tener una idea de cómo estructuraríamos la sección de classroom de BlockLeap.

- Estudio de BlockLeap

Fase inicial del proyecto en el que estuvimos un tiempo estudiando la estructura y el funcionamiento del cliente y del servidor.

- Sección classroom de BlockLeap

Desarrollo de las funcionalidades de classroom.

- Creación de waiting room tanto para alumno como para profesor, creación de clases y unirse a clase.
  - Creación de las clases, funcionalidades los botones de añadir sets y de añadir niveles, así como mostrar niveles y sets en la clase, sin la parte de paginación y los filtros de búsqueda.
- Crear y añadir niveles al set
    - Implementación de la funcionalidad para que el profesor pueda crear sets y añadirle niveles desde el perfil.
  - Soporte en la implementación de la sección perfil
    - Soporte en la sección de perfil, crear sets, visualizar niveles creados sin la parte de datatables y sin paginación ni filtro
  - Configuración *EmotiBit*
    - Primeros pasos con *Emotibit* configurado con el wifi de casa y capturando los primeros datos con el osciloscopio y con el data parser.
  - Creación de algoritmo *EmotiBit*
    - Creación de un código python para analizar los datos capturados con *Emotibit* y sacar gráficas de los datos
  - Pruebas *Emotibit*
    - Desarrollo de un entorno de pruebas con diferentes usuarios que probaron BlockLeap con la placa *Emotibit* para recolectar datos para su posterior análisis.
  - Estudio datos *EmotiBit*

Con los datos obtenidos en las pruebas pasar los datos por el algoritmo creado y generar las gráficas para su análisis

- Creación y modificación de DAOs, Controllers y routers

Implementación de las funciones necesarias para el correcto funcionamiento de la sección classroom.

- Sección tutoriales

Ayuda en la implementación de la sección de tutoriales, así como escribir textos explicativos para cada sección.

- Ampliación de la base de datos

Adición de esquemas de datos relativos a las nuevas funcionalidades que se han implementado. Así como modificación de los esquemas existentes para acomodar las nuevas relaciones.

- Memoria

Aportaciones a la memoria final del trabajo.

## **Victoria Martín Rojo**

- Investigación juegos serios

Investigación de Code Monkey, juego que permite aprender a programar y gestionar clases de alumnos desde un usuario profesor, de la que se extrajeron ideas para la gestión de clases del juego y la organización.

- Creación nuevos niveles en el juego

Inicialmente, creación de niveles en el apartado play para ampliar el juego y aprender a gestionar este.

- Estudio de BlockLeap

Estudio de la documentación del juego, tanto de la memoria de este como la documentación de blockly y phaser para entender el funcionamiento del código y la estructura de este.

- Aportación al figma (diseño de alto nivel)

Diseño final de figma con elementos añadidos de la web para simular el comportamiento ideal de esta

- Actualización del elemento chest del juego

Creación mediante una web externa de dibujo por pixeles de un cofre abierto e introducción de esta imagen en el código para simular la recolección de este.

- Gestión de las estrellas y la obtención de estas

Recopilación de la información aportada por estrellas y conexión de lo programado con estas. Programación de ambas versiones y solución de los problemas de estas

- Creación selector de estrellas

Modificación del modal para aportar el selector de estrellas y conexión con la base de datos del nivel.

- Aportación al perfil con una tabla de sets

Organización de las carpetas de niveles creadas en una tabla importando DataTables al programa.

- Guardado de código de niveles al crearlo

Investigación de las herramientas que ofrecía Blockly más exhaustivamente para gestionar estos bloques.

Ampliación del controlador de bloques para añadirle funciones dependiendo del modo en el que se llama a los bloques, si modo editor o modo jugador.

- Mejoras de diseño en la web
- Recolectar datos de usuarios para la memoria

Creación de encuesta y recolección de opiniones de personas que han probado el juego teniendo en cuenta las que lo han jugado con pulsera o no para analizar estos datos y opiniones y tenerlos en cuenta para mejoras y trabajo futuro

## **Pablo Lavandeira Poyato**

- Investigación juegos serios

Investigación de FreeCodeCamp, que posee una de las mayores comunidades abiertas donde los usuarios pueden compartir sus creaciones y logros

- Estudio de BlockLeap

Fase inicial del proyecto en el que estuvimos un tiempo estudiando la estructura y el funcionamiento del cliente y del servidor

- Creación de bocetos Figma

Creación de bocetos con la aplicación web Figma en la que se crearon bocetos de alto nivel para tener una idea de cómo estructurar la plataforma de BlockLeap en su conjunto.

- Soporte en la creación sección classroom de BlockLeap

Desarrollo de las funcionalidades de classroom.

- Creación de waiting room para profesor, creación de clases y unirse a clase.
  - Apoyo en la creación de las clases, funcionalidades de los botones de añadir sets y de añadir niveles, así como mostrar niveles y sets en la clase.
- Apoyo en crear y añadir niveles al set

Implementación de la funcionalidad para que el profesor pueda crear sets y añadirle niveles desde el perfil.

- Implementación de la sección perfil

Mejora de la sección de perfil, crear sets, visualizar niveles creados sin la parte de datatables y sin paginación ni filtro

- Apoyo en la creación y modificación de DAOs, Controllers y routers

Implementación de las funciones necesarias para el correcto funcionamiento de la sección classroom, perfil y tutoriales.

- Sección tutoriales

Ayuda en la implementación de la sección de tutoriales, así como escribir textos explicativos para cada sección.

- Mejora de la estructura HTML

Análisis y mejora de toda la estructura previa de HTML así como su ampliación y traducción para mantener un nivel de coherencia.

- Aportaciones a la memoria final del trabajo

## Bibliografía

- [1] Odilo. 2024. «Integración del Pensamiento Computacional, Programación y Robótica en el Currículo Educativo: ¿Por qué es crucial para el futuro?,» [En línea]. Available:  
<https://www.odilo.es/blog/integracion-del-pensamiento-computacional-programacion-y-robotica-en-el-curriculo-educativo-por-que-es-crucial-para-el-futuro/>
- [2] Gestionet. 2019. «Serious games: qué son, ejemplos y tipos,» [En línea]. Available:  
<https://gestionet.net/serious-games-blog/>
- [3] N. Espinosa, S. Moral, S. Parra, D. Álvarez. 2024. «Tecnologías web para la creación de una herramienta de enseñanza de conceptos de programación a través de un videojuego,» [En línea]. Available:  
<https://docta.ucm.es/entities/publication/86c3ccbd-da5e-4bc9-bc4b-44e8b648ee75>
- [4] F. Sánchez. 2021. «Aumento de tiempo de estudio en alumnos mediante la utilización de videojuegos y gamificación,» [En línea]. Available:  
<https://digibuo.uniovi.es/dspace/handle/10651/60337>
- [5] J. Koivisto, J. Hamari. 2021. «The rise of motivational information systems: A review of gamification research,» [En línea]. Available:  
<https://www.sciencedirect.com/science/article/pii/S0268401217305169>

- [6] M. L. Velasco, A. D. Otero. 2023 «El impacto del uso de Scratch para favorecer el pensamiento algorítmico en estudiantes de educación superior,» [En línea] Available: <https://dialnet.unirioja.es/servlet/articulo?codigo=9741384>
- [7] R. Martín. 2024. «El 83 % de los hogares españoles tiene un PC,» [En línea] Available: <https://newsbook.es/actualidad/el-83-de-los-hogares-espanoles-tiene-un-pc-20241118113349.htm>
- [8] Blockly, guías de estudio «Potente programación bloque a bloque,» [En línea]. Available: <https://developers.google.com/blockly/guides/create-custom-blocks/overview?hl=es-419>
- [9] Guías educativas de code «IniciacionCode.pdf,» [En línea]. Available: <https://github.com/lobotic/GuiasEducativas/blob/main/Code/IniciacionCode.pdf>
- [10] Editor Setemcat «Nuevos criterios para el tratamiento de datos biométricos,» [En línea]. Available: <https://www.setemcat.com/nuevos-criterios-para-el-tratamiento-de-datos-biometricos/>



