



# TRABAJO FIN DE GRADO

Magikal Kombat - TFG CFGS DAM

Ángel vaquero  
angelvaquerolarioja@gmail.com

## Índice

<b>Capítulo 1 - Introducción.....</b>	<b>3</b>
Resumen .....	3
Abstract.....	5
<b>Capítulo 2 - Objetivos Técnicos y su justificación.....</b>	<b>7</b>
<b>Capítulo 3 - Metodologías Ágiles y Planificación .....</b>	<b>8</b>
Scrum .....	8
Justificación del uso de Scrum .....	8
Planificación.....	9
Diagrama de Gantt .....	9
<b>Capítulo 4 - Estudio de la situación actual.....</b>	<b>10</b>
Evaluación de alternativas y justificación de su elección .....	10
Tecnologías utilizadas .....	10
Unity .....	10
Elegoo Uno y Arduino .....	10
Arduino IDE.....	10
<b>Capítulo 5 - Fases del proyecto.....</b>	<b>11</b>
Análisis .....	11
Requisitos Funcionales.....	11
Actores y sus acciones.....	11
Diagrama de caso de uso .....	12
Presupuesto .....	13
Diseño .....	14
Nombre del proyecto .....	14
Logo .....	14
Personajes .....	15
Habilidades de los personajes.....	15
Enemigos .....	16
Jefes.....	16
Cartas .....	17
Extras.....	18
Fuente de los textos.....	19
Diseño de las interfaces .....	20
Diseño de la base de datos .....	29
Implementación.....	30
Básica.....	30
Prefab .....	30
Scriptable Object .....	30
Audio .....	31
Elegoo Uno y su implementación en Unity .....	32
Juego .....	33
Personajes.....	33

Habilidades.....	33
Enemigos .....	33
Turnos.....	34
Cartas .....	34
Mazo.....	34
Cartas en mano .....	34
Animaciones .....	34
Pruebas .....	35
Unitarias .....	35
Pruebas de Caja Negra .....	35
<b>Capítulo 6 - Manual, Ampliaciones y Problemas y sus Soluciones .....</b>	<b>38</b>
Manual.....	38
Ampliaciones.....	38
Problemas encontrados y sus soluciones.....	38
<b>Capítulo 7 - Final.....</b>	<b>39</b>
Conclusión .....	39
Bibliografía.....	39

## Índice de Ilustraciones

Ilustración 1-Tabla de tipos.....	4
Ilustración 2-Type Table .....	6
Ilustración 3-Diagrama de Gantt.....	10
Ilustración 4-Diagrama de caso de uso Jugar .....	13
Ilustración 5-Diagrama de caso de uso Combate.....	13
Ilustración 6-Logo del juego .....	15
Ilustración 7-Personajes del juego .....	16
Ilustración 8-Habilidades de los personajes.....	16
Ilustración 9-Enemigos del juego .....	17
Ilustración 10-Jefes del juego.....	17
Ilustración 11-Cartas del juego.....	18
Ilustración 12-Extras del juego 1 .....	19
Ilustración 13-Extras del juego 2 .....	19
Ilustración 14-Fuente del juego .....	20
Ilustración 15-Menú inicial.....	21
Ilustración 16-Ventana de como jugar.....	22
Ilustración 17-Ventana de ajustes.....	23
Ilustración 18-Ventana de selección de personaje vacía .....	24
Ilustración 19-Ventana de selección de personaje funcional .....	24
Ilustración 20-Error en la selección de personaje.....	25
Ilustración 21-Error en la conexión del Elegoo .....	26
Ilustración 22-Ventana de batalla .....	26
Ilustración 23-Ventana de batalla al usar tu habilidad .....	27
Ilustración 24-Pausa en la batalla .....	27
Ilustración 25-Ganaste la batalla.....	28
Ilustración 26-Perdiste la batalla.....	28
Ilustración 27-Unión del Arduino y el lector .....	32

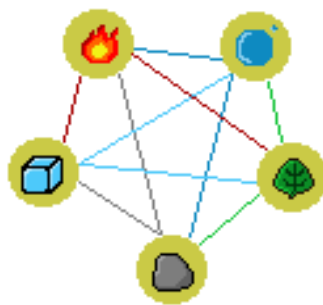
## Capítulo 1 - Introducción

### Resumen

- Este proyecto es un videojuego de cartas desarrollado con una base de piedra, papel, tijera, pero añadiendo dos fases más, en mi caso se hace uso de los siguientes elementos:
  - Roca
  - Fuego
  - Agua
  - Hielo
  - Planta
- De cada tipo de cartas existen 3 valores (2, 4, 6) que son cartas de ataque.
- La mecánica de lucha es la siguiente:

- SISTEMA DE COMBATE:

PLANTA:	GANA - AGUA / ROCA	PIERDE - FUEGO / HIELO
AGUA:	GANA - ROCA / FUEGO	PIERDE - HIELO / PLANTA
ROCA:	GANA - FUEGO / HIELO	PIERDE - PLANTA / AGUA
FUEGO:	GANA - HIELO / PLANTA	PIERDE - AGUA / ROCA
HIELO:	GANA - PLANTA / AGUA	PIERDE - ROCA / FUEGO



*Ilustración 1-Tabla de tipos*

1º SE PREPARAN LOS ELEMENTOS (x1.5 GANADOR, x1 EMPATE, x0.5 PERDEDOR).

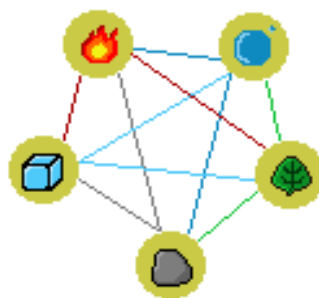
2º LAS CARTAS ATACAN.

- El juego se basa en un roguelike (Un roguelike es un subgénero de videojuegos de rol caracterizado por niveles generados proceduralmente, muerte permanente y una jugabilidad basada en turnos.), al iniciar el juego deberás de seleccionar el personaje mediante posicionar la figura del personaje deseado sobre el lector RFID o de forma aleatoria, ese personaje te acompañará a lo largo de la partida.
- Los personajes son 3:
  - Kassy, que su objeto es la moneda (50% de hacer 20 ptos. de daño al enemigo y un 50% de hacerte 20 ptos. de daño a ti mismo), tiene 60 puntos de vida.
  - Cecik, que su objeto es la cruz invertida (Le quita 10 puntos de vida al enemigo), tiene 50 puntos de vida.
  - KJ, que su objeto es el anillo de oro (Se cura 10 ptos. de vida), tiene 70 puntos de vida.
- La estructura de la partida se basa en una lucha contra un enemigo normal o un jefe, los cuales son seleccionados de forma aleatoria.
- Los enemigos tendrán entre 55-65 ptos. de vida, los jefes tendrán entre 70-85 ptos. de vida, y sus jugadas funcionarán mediante IA.
- Durante las batallas tu mano tendrá siempre un mínimo y un máximo de 3 cartas, podrás usar un máximo de 2 cartas por ronda, las batallas llegan a su fin cuando tu enemigo o tu morís.
- Al finalizar la ronda recibes en tu mano tantas cartas como te hagan falta para llegar a 3.
- Tu mazo de cartas se basa en los 5 tipos de cartas multiplicado por 3 que son los 3 valores de cartas que hay. El mazo tendrá 15 cartas no reutilizables, una vez las acabas de jugar se borran de tu mazo, si acabas tu mazo llegarán las cartas especiales, las cartas malditas, de las cuales hay una por cada tipo de carta sus valores solo son de 2, el punto de las malditas es que si las usas te quitan 1 de vida por cada uso.

## Abstract

- This project is a card game developed with a rock-paper-scissors base, but adding two more phases. In my case, the following elements are used:
  - Rock
  - Fire
  - Water
  - Ice
  - Plant
- There are 3 values (2, 4, 6) for each type of card, which are attack cards.
- The combat mechanics are as follows:
  - COMBAT SYSTEM:
 

PLANT:	WINS - WATER / ROCK	LOSES - FIRE / ICE
WATER:	WINS - ROCK / FIRE	LOSES - ICE / PLANT
ROCK:	WINS - FIRE / ICE	LOSES - PLANT / WATER
FIRE:	WINS - ICE / PLANT	LOSES - WATER / ROCK
ICE:	WINS - PLANT / WATER	LOSES - ROCK / FIRE



*Ilustración 2-Type Table*

1st ELEMENTS ARE PREPARED (x1.5 WINNER, x1 DRAW, x0.5 LOSER).

2nd CARDS ATTACK.

- The game is based on a roguelike (A roguelike is a subgenre of role-playing games characterized by procedurally generated levels, permanent death, and turn-based gameplay.). At the start of the game, you must select the character by positioning the desired character figure on the RFID reader or randomly. That character will accompany you throughout the game.
- There are 3 characters:
  - Kassy, whose object is the coin (50% chance of dealing 20 damage to the enemy and a 50% chance of dealing 20 damage to yourself), has 60 health points.
  - Cecik, whose object is the inverted cross (Removes 10 health points from the enemy), has 50 health points.
  - KJ, whose object is the gold ring (Heals 10 health points), has 70 health points.
  - The structure of the game consists of battles against a normal enemy or a boss, which are randomly selected.
- Enemies will have between 55-65 health points, bosses will have between 70-85 health points, and their moves will operate using AI.
- During battles, your hand will always have a minimum and maximum of 3 cards. You can use a maximum of 2 cards per round. Battles end when either your enemy or you die.
- At the end of the round, you receive in your hand as many cards as you need to reach 3.
- Your deck of cards is based on the 5 types of cards multiplied by 3, which are the 3 card values available. The deck will have 15 non-reusable cards. Once you finish playing them, they are removed from your deck. If you run out of cards in your deck, special cards will come into play, the cursed cards, of which there is one for each type of card. Their values are only 2, and the point of the cursed cards is that if you use them, they take away 1 life point for each use.



## Capítulo 2 - Objetivos Técnicos y su justificación

- Uno de mis objetivos primordiales en el proyecto es aprender Unity y a usar correctamente Elegoo Uno, lo priorizo ante todo. Luego también sería llegar a realizar un proyecto que me sienta orgulloso de haber realizado.
- Este proyecto lo estoy realizando ya que quiero realizar un videojuego que toque dos de mis pasiones, el desarrollo de aplicaciones y la tecnología como viene siendo el uso del Elegoo Uno con el lector RFID, también me introduje a este proyecto ya que es un nuevo desafío, por el motivo de que nunca he programado haciendo uso de Unity, ni nunca hice uso de Elegoo Uno.

## Capítulo 3 - Metodologías Ágiles y Planificación

### Scrum

- Scrum es un marco de administración que los equipos utilizan para organizarse por cuenta propia y trabajar en aras de alcanzar un objetivo común. Describe un conjunto de reuniones, herramientas y funciones para entregar proyectos de forma eficiente. Al igual que un equipo deportivo que practica para un importante partido, las prácticas de Scrum permiten a los equipos de trabajo gestionarse por cuenta propia, aprender a partir de la experiencia y adaptarse al cambio. Los equipos de software utilizan Scrum para resolver problemas complejos de forma rentable y sostenible.

### Justificación del uso de Scrum

- Decidí trabajar con Scrum y sprints semanales en mi proyecto por varias razones clave:
  1. **Estructura y enfoque:** Me gusta cómo Scrum proporciona una estructura clara para mi proyecto. Los sprints semanales me ayudan a mantenerme enfocado y productivo al dividir mi trabajo en bloques manejables y alcanzables en un período de tiempo definido.
  2. **Adaptabilidad y rapidez:** Como soy el único responsable del proyecto, es crucial para mí poder adaptarme rápidamente a los cambios y prioridades emergentes. Scrum me permite hacerlo al ofrecer ciclos cortos de desarrollo y la capacidad de ajustar mi planificación en cada sprint.
  3. **Autogestión:** Scrum fomenta la autogestión y la toma de decisiones. Como soy el único miembro del equipo, tengo la libertad de organizar mi trabajo y tomar decisiones sobre cómo abordar cada sprint para alcanzar mis objetivos.
  4. **Retroalimentación continua:** Los sprints semanales me brindan la oportunidad de obtener retroalimentación temprana y continua sobre mi trabajo. Esto me permite ajustar mi enfoque según sea necesario y asegurarme de que estoy en el camino correcto hacia mis metas.
  5. **Mejora personal:** Con cada sprint, tengo la oportunidad de reflexionar sobre mi desempeño y buscar formas de mejorar. Esta mentalidad de mejora continua es fundamental para mi crecimiento profesional y el éxito a largo plazo de mi proyecto.
- En resumen, trabajar con Scrum y sprints semanales en mi proyecto me permite mantenerme organizado, adaptable y enfocado en mis objetivos. Estas prácticas me ayudan a maximizar la productividad, mejorar mi rendimiento y alcanzar mis metas de manera eficiente como único propietario del proyecto.

Planificación

- Para la planificación del proyecto se ha tenido en cuenta que la jornada laboral será de 4 horas diarias.
- Trabajé en el proyecto con sprints semanales donde el master era mi tutor de TFG.

Diagrama de Gantt

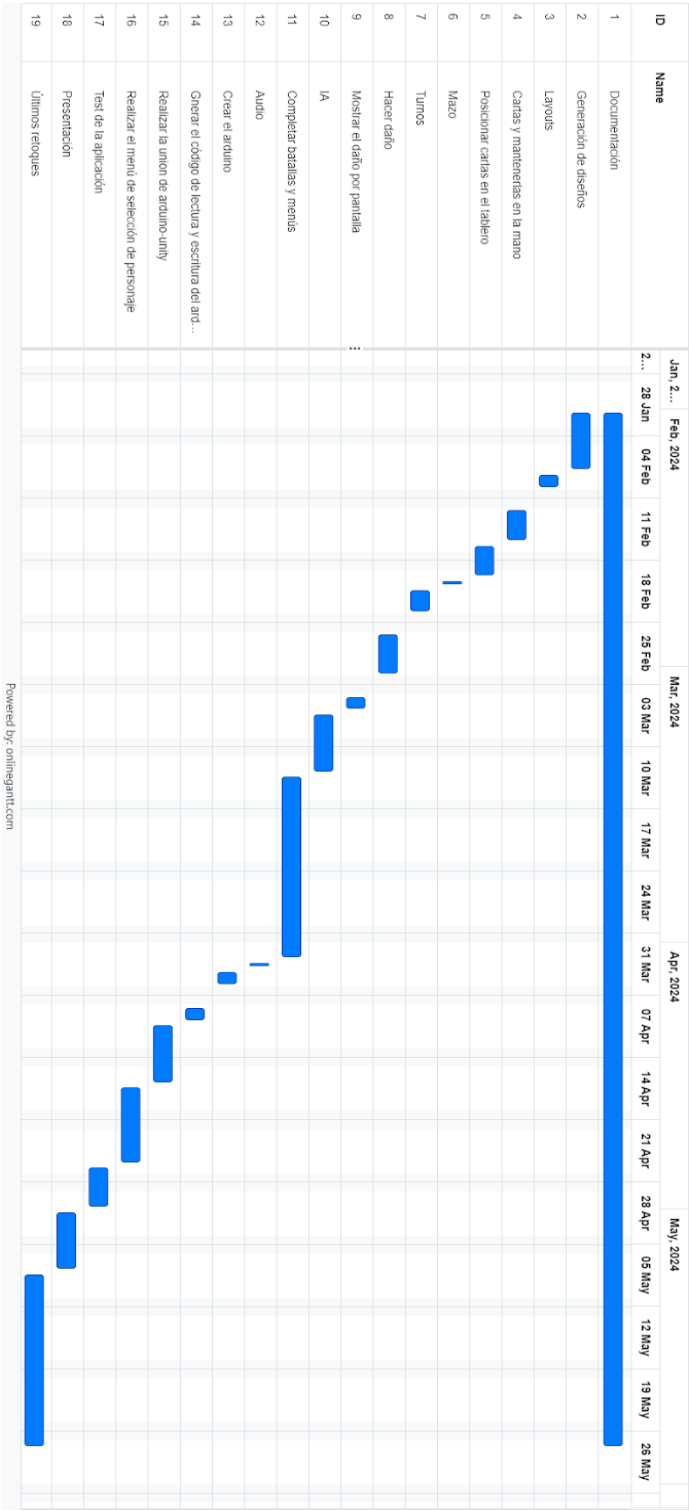


Ilustración 3-Diagrama de Gantt

## Capítulo 4 - Estudio de la situación actual

### Evaluación de alternativas y justificación de su elección

- En este proyecto, podría haber utilizado distintas tecnologías para desarrollar la sección del videojuego. Por ejemplo, consideré Unreal Engine 5, que ofrece mayores capacidades gráficas y de desarrollo. Sin embargo, opté por Unity debido a su extensa documentación y la abundancia de cursos explicativos que me facilitan el aprendizaje y el desarrollo del proyecto. En cuanto al dispositivo para la lectura de tarjetas NFC, elegí usar un Elegoo Uno en lugar de un Arduino convencional. La razón principal fue la significativa diferencia de precios, ya que ambos tienen las mismas características funcionales y pueden programarse con Arduino IDE. No encontré justificación para gastar más en un producto que ofrece las mismas prestaciones.

### Tecnologías utilizadas

#### Unity

- Unity es un motor de videojuego multiplataforma creado por Unity Technologies, anunciado y lanzado por primera vez en junio de 2005 en la Conferencia Mundial de Desarrolladores de Apple como motor de juegos para Mac OS X. Unity está disponible como plataforma de desarrollo para Microsoft Windows, Mac OS, Linux. La plataforma de desarrollo tiene soporte de compilación con diferentes tipos de plataformas.

#### Elegoo Uno y Arduino

- El Elegoo Uno es una placa de desarrollo basada en el microcontrolador ATmega328P de la familia Arduino. Arduino es una plataforma de hardware y software de código abierto que facilita la creación de proyectos electrónicos interactivos. El Elegoo Uno es una de las muchas variantes compatibles con Arduino Uno, ofreciendo una alternativa generalmente más asequible.

#### Arduino IDE

- El entorno de desarrollo integrado (IDE) de Arduino es una aplicación multiplataforma (para Windows, macOS, Linux) que está escrita en el lenguaje de programación Java. Se utiliza para escribir y cargar programas en placas compatibles con Arduino, pero también, con la ayuda de núcleos de terceros, se puede usar con placas de desarrollo de otros proveedores. El IDE de Arduino admite los lenguajes C y C++ utilizando reglas especiales de estructuración de códigos.

## Capítulo 5 - Fases del proyecto

**Análisis****Requisitos Funcionales**

Código	Nombre Requisito
R1	Selección de personaje
R1.1	Selección: <ul style="list-style-type: none"> <li>- El usuario deberá de seleccionar un personaje mediante la posicionamiento de un una tarjeta en la base RFID</li> </ul>
R2	Combate
R2.1	Selección de carta
R2.1.1	Elección de cartas: <ul style="list-style-type: none"> <li>- El usuario deberá de seleccionar una carta o dos cartas que se posicionarán en su debido lugar. (Existen dos tipos de cartas normales y las especiales)</li> <li>- El enemigo/jefe selecciona carta mediante una IA</li> </ul>
R2.2	Usar habilid
R2.2.1	Utilizar habilidades: <ul style="list-style-type: none"> <li>- El usuario tendrá la posibilidad de usar la habilidad especial de su personaje en el turno</li> </ul>
R2.3	Finalizar turno
R2.3.1	Finalizar: <ul style="list-style-type: none"> <li>- El usuario tendrá la posibilidad de finalizar el turno</li> </ul>

## Actores y sus acciones

- Los actores son el **usuario**, **enemigo** y **jefe** (Enemigo y Jefe son muy similares pero la diferencia es que, el jefe tiene mayor puntuación de vida y por lo tanto decidí que fueran actores distintos.).

El usuario es la persona que va iniciar la partida y quién marcará el tempo de la batalla. Cuando un usuario acaba su turno el enemigo selecciona una o dos cartas, además de ser el único actor con posibilidad de selección de personaje y uso de habilidad.

## Diagrama de caso de uso

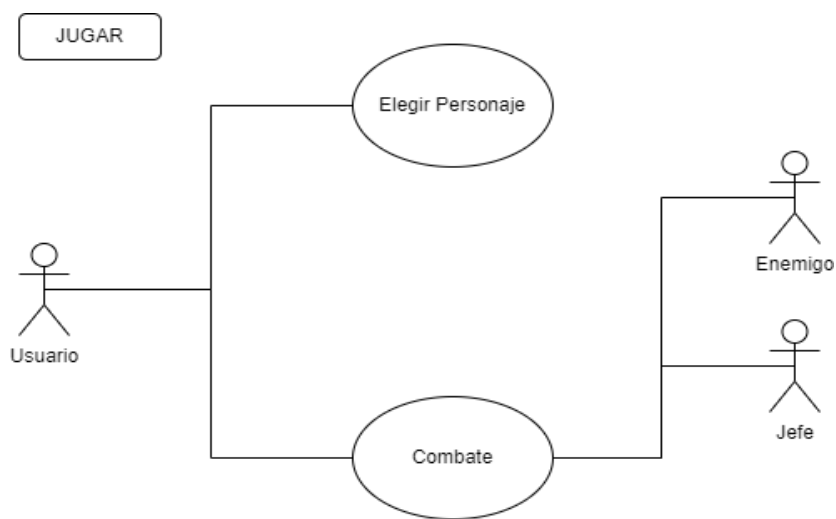


Ilustración 4-Diagrama de caso de uso Jugar

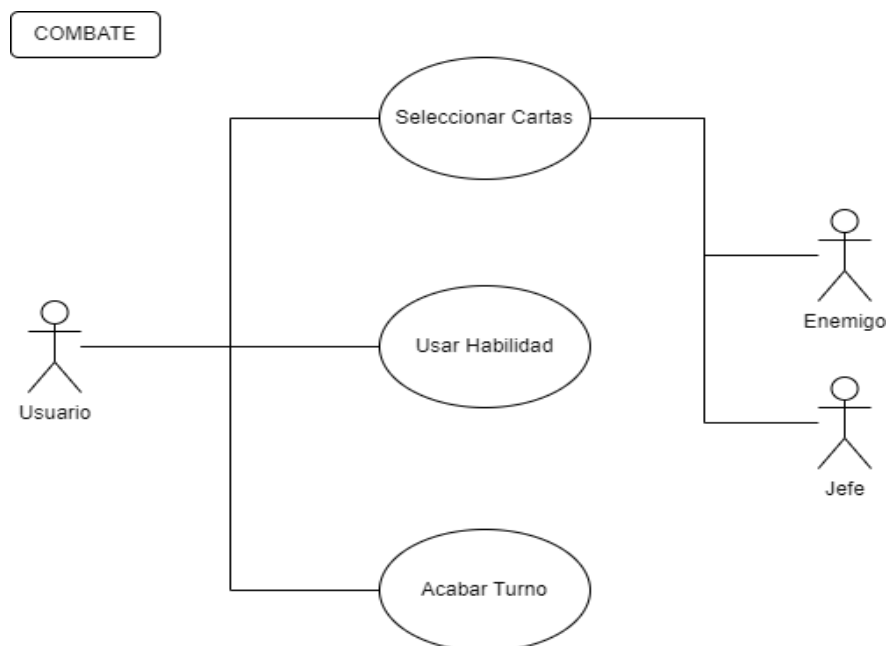


Ilustración 5-Diagrama de caso de uso Combate

## Presupuesto

- Tomando como referencia la planificación del proyecto, y haciendo una media de unas 4 horas por día de lunes a viernes, realicé 336 horas aproximadas, poniendo un salario justo al resultado obtenido cobraría a 15€/hora y a ese salario hay que sumarle el gasto del Elegoo Uno y de las pegatinas NFC lo que son aproximadamente 60€.

Item	Concepto	Coste Item
001	Desarrollo de la Aplicación	5.040,00€
002	Software e Instalación	150,00€
003	Hardware	60,00€
004	Otros Gastos	0,00€
<b>Subtotal</b>		5250,00€
<b>IVA(21%)</b>		1.102,50€
<b>TOTAL</b>		6.352,50€

## Diseño

### Nombre del proyecto

- El nombre de este proyecto viene a raíz del personaje principal del conocido juego "The Binding of Isaac" Isaac, ya que si lees su nombre al revés es "Caasi", lo cual no queda demasiado bien, así que decidí darle otra vuelta y decidí que Kassy quedaba mejor. El resto de personajes llevó la misma lógica de jugar con la 'K', por lo que el nombre del proyecto también tenía que contener la 'K' y como el proyecto tiene temática "mágica" y de lucha el nombre que más me encajó fue Magikal Kombat.

### Logo

- El logo representa un libro donde estarían los conjuros para crear las cartas mágicas y sus portadores:

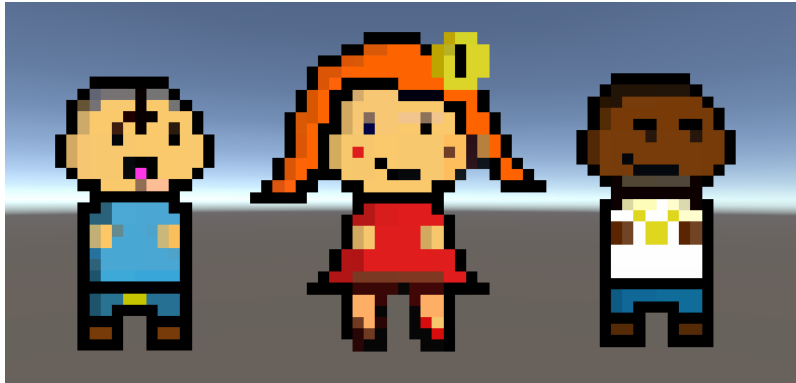


Ilustración 6-Logo del juego



## Personajes

- Los personajes son tres, pero el personaje principal del proyecto se podría considerar que es Kassy, aquí los personajes:

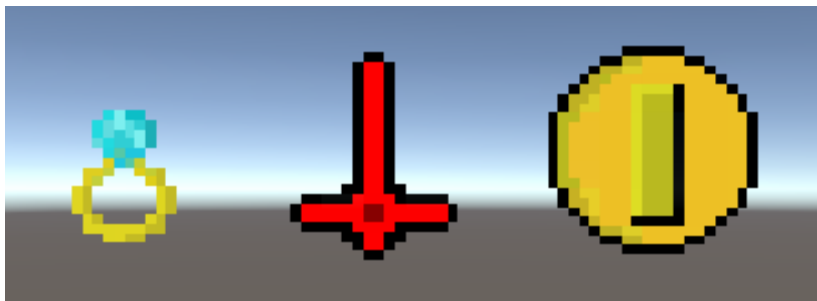


*Ilustración 7-Personajes del juego*

- En orden son Cecik, Kassy y KJ.

## Habilidades de los personajes

- Los personajes poseen habilidades únicas que el diseño de las mismas es el siguiente:



*Ilustración 8-Habilidades de los personajes*

- En orden son las habilidades de KJ, Cecik y Kassy.

## Enemigos

- Los enemigos son los siguientes:



*Ilustración 9-Enemigos del juego*

- Están inspirados en los monstruos clásicos de las películas de miedo, en orden son la momia, el fantasma, frankenstein el cual llame franky en el juego, el vampiro y un zombie.

## Jefes

- Los jefes son el diseño de los enemigos pero con toques dorados y con corona menos franky ya que él tenía un diseño demasiado grande como para añadirle una corona y son los siguientes:



*Ilustración 10-Jefes del juego*

## Cartas

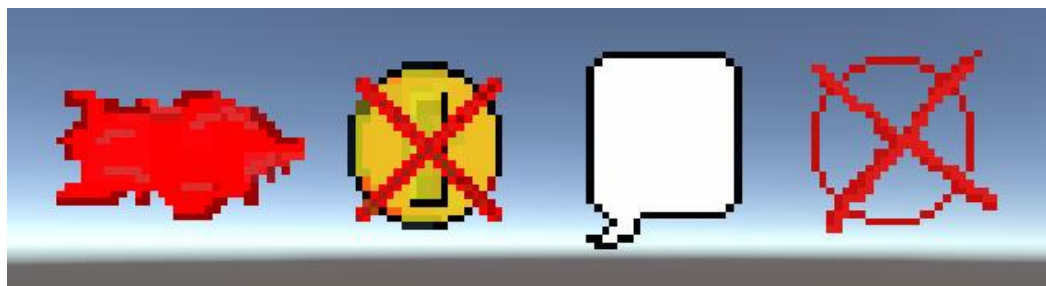
- Las cartas del juego son bastante reconocibles por su tipo pero las cartas malditas que son las cuales te quitan vida su punto especial es que tienen una calavera:



Ilustración 11-Cartas del juego

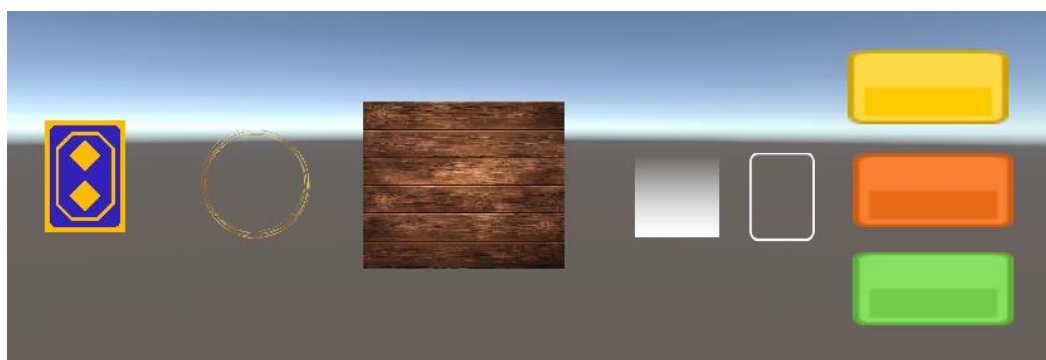
## Extras

- Aquí enseño el resto de diseños usados en el proyecto:



*Ilustración 12-Extras del juego 1*

- Aquí se enseña el diseño de los avisos que da el proyecto cuando algo no funciona o también usado para indicar el multiplicador de las cartas o si ganaste o perdiste tu batalla, luego va el diseño de la moneda con cruz usado para indicar que el volumen está desactivado, para indicar que está activado uso el diseño de la habilidad de Kassy que es una moneda, luego están los cuadros de diálogo que usan los personajes para explicar sus habilidades y por último el diseño del botón de cerrar ventana usado en la ventana de como jugar.
- Estos son el resto de extras:



*Ilustración 13-Extras del juego 2*

- El primer diseño es usado para la parte de atrás de las cartas y para el mazo desde donde se generan las cartas, lo siguiente es el aro que rodea la imagen del personaje en la pantalla de batalla, también usado para la habilidad y para el enemigo de la misma forma, continuamos con una imagen de madera la cual se usa para la mesa donde se juega, para los fondos de algunas escenas y para el letrero del menú donde está el título. Luego un degradado usado en la pantalla inicial para darle un degradado, luego los recuadros donde se colocan las cartas en la pantalla de batalla y para los enemigos solo cambié el color a uno más oscuro, y luego los tres diseños de botones del juego.

## Fuente de los textos

- La fuente usada para todos los textos de este proyecto ha sido:



*Ilustración 14-Fuente del juego*

- La fuente es propia de Unity.

## Diseño de las interfaces

- En este apartado voy a justificar el diseño implementado en el videojuego, primero empezaremos por el menú principal.



*Ilustración 15-Menú inicial*

- Aquí podemos ver el título del juego y 4 botones, los cuales en orden sirven para:
  - Comenzar: Te lleva a la ventana de selección de personajes.
  - Como Jugar: Te lleva a una ventana donde hay una explicación de cómo jugar.
  - Ajustes: Te lleva a la ventana de ajustes de sonido del juego.
  - Salir: Te saca del juego.

- Continuamos explicando la ventana de como jugar:



*Ilustración 16-Ventana de como jugar*

- Esta ventana posee 3 botones en pantalla, uno para cerrar esta ventana explicativa y otros dos uno a cada lado de la imagen explicativa para ir viendo explicación la explicación de cómo jugar mediante la imagen y el texto abajo de ella.

- Proseguimos con la ventana de ajustes:



*Ilustración 17-Ventana de ajustes*

- Nos abre la siguiente ventana en la cual podemos modificar el sonido del juego mediante tres opciones. General modifica el sonido general del juego como el nombre indica, música modifica el sonido de la música del juego y por último SFX, el cual modifica los efectos de sonido del juego.



- Por último, en referencia al menú el botón comenzar nos lleva a la siguiente ventana:



*Ilustración 18-Ventana de selección de personaje vacía*

- En la cual podemos ver que tenemos varios botones en pantalla. Qué son los siguientes:
  - Volver: Nos lleva nuevamente al menú.
  - Elegir: Confirma el personaje con el que quieres jugar.
  - Aleatorio: Elige un personaje de forma aleatoria.
  - Jugar: Nos lleva a la pantalla de juego.
- Aquí vemos como sería al hacer la selección de un personaje:



*Ilustración 19-Ventana de selección de personaje funcional*

- Se ve el personaje, su nombre, su habilidad y el da una explicación de la misma.

### *Notas sobre esta ventana*

- ❖ Desde esta ventana es donde se hace uso del Elegoo Uno y el RFID.
- ❖ Al hacer clic en elegir su color cambia a verde para confirmar su elección.
- ❖ Si no haces la selección del personaje de forma correcta sale un aviso visual:



*Ilustración 20-Error en la selección de personaje*

- ❖ Si no se ha conectado el Elegoo Uno o hubo un error con la conexión da un aviso:



Ilustración 21-Error en la conexión del Elegoo

- Continuamos enseñando la ventana de jugar:



Ilustración 22-Ventana de batalla

- En esta ventana tenemos varias opciones:
  - Habilidad: Podemos hacer click a la habilidad para usarla.
  - Usar nuestras cartas para poder luchar contra el enemigo para derrotarlo posicionando las mismas en las casillas blancas que es donde el usuario las coloca, ya que en las casillas negras las coloca el enemigo.
  - Acabar el turno: El botón acaba el turno y las cartas sobre la mesa atacan a las cartas que tenga enfrente o en ende al enemigo directamente si no hay carta.



## Notas sobre esta ventana

- ❖ Al usar la habilidad desaparece, ya que la has usado:



Ilustración 23-Ventana de batalla al usar tu habilidad

- ❖ Al finalizar el turno y que las cartas ya hayan atacado es cuando se te devuelven las cartas que necesitas para llegar a 3 en tu mano.
- ❖ Si presionas la tecla “ESC” de tu teclado pausas el juego y puedes reanudar la partida, ir a los ajustes de sonido o volver al menú, si vuelves a presionar la tecla “ESC” hace la misma función que el botón de reanudar:



Ilustración 24-Pausa en la batalla

- Si ganas la partida este es el menú:



Ilustración 25-Ganaste la batalla

- Y si pierdes es este otro:



Ilustración 26-Perdiste la batalla

## Diseño de la base de datos

Durante el diseño y desarrollo de mi trabajo se ha llevado a cabo una consideración cuidadosa sobre la arquitectura y los requisitos del sistema. Se ha tomado la decisión deliberada de omitir el uso de bases de datos, y a continuación se presenta una justificación fundamentada en las necesidades específicas del proyecto:

- **Naturaleza temporal de los datos:** El sistema aborda principalmente el procesamiento de datos efímeros y temporales, tales como datos en tiempo real o generados por procesos de cálculo y simulación. Dado que la persistencia a largo plazo de estos datos no es necesaria para los objetivos del sistema, la introducción de una base de datos sería redundante y no aportaría valor funcional.
- **Simplicidad y eficiencia del sistema:** La incorporación de una base de datos añadiría una capa adicional de complejidad y sobrecarga al sistema, así como requeriría recursos adicionales para su gestión y mantenimiento. Al evitar esta dependencia, se promueve la simplicidad y la eficiencia del sistema, permitiendo una implementación más ágil y un funcionamiento más fluido.
- **Portabilidad y facilidad de implementación:** La ausencia de una base de datos facilita la portabilidad del sistema y su implementación en una variedad de entornos informáticos. Al no requerir configuraciones específicas de bases de datos ni dependencias externas, se simplifica el proceso de despliegue y se reduce la complejidad de la infraestructura necesaria.
- **Inutilidad de una Base de Datos:** La decisión de no incluir una base de datos en el presente proyecto se basa en la inutilidad percibida de su implementación. Dado que no es necesario persistir información a largo plazo para los objetivos y requisitos del proyecto, se considera que la introducción de una base de datos sería innecesaria.

En conclusión, la omisión de bases de datos en el diseño y desarrollo del sistema se justifica por las necesidades específicas del proyecto, incluida la naturaleza temporal de los datos, la simplicidad y eficiencia del sistema. Esta decisión se alinea con los objetivos y requisitos establecidos para el TFG, asegurando un enfoque coherente y eficaz para alcanzar los resultados deseados.

## Implementación

### Básica

- Empezaremos hablando de las “clases” en Unity, que se llaman scripts y se aplican a los objetos en pantalla para que hagan ciertas acciones específicas según los métodos que tenga el script.
- Los métodos básicos dentro de un script de Unity son los siguientes:
  - **Start:** Utilizado para inicializar variables.
  - **Update:** Llamado en cada 'frame' (Imagen concreta dentro de una sucesión de imágenes en movimiento) cuando se ejecuta el juego.
  - **Awake:** Se ejecuta antes de todos los Start.
- Luego tú como usuario puedes crear los métodos que desees dentro del script para asignarlos a lo que quieras.

### Prefab

- El prefab actúa como una plantilla a partir de la cual se pueden crear nuevas instancias del objeto en la escena. Cualquier edición hecha a un prefab asset será inmediatamente reflejado en todas las instancias producidas por él, pero, también se puede anular “components” y ajustes para cada instancia individualmente.
- Uso los prefabs en botones y en cartas para ahorrarme tiempo en la creación de elementos que funcionan base una misma plantilla.

### Scriptable Object

- Es una clase que le permite a usted almacenar grandes cantidades de datos compartidos independientes de instancias de script.
- Uso este tipo de clase para la creación de cartas, habilidades, enemigos y personajes, ya que ayudan a mantener una organización y clasificación de los objetos usados.

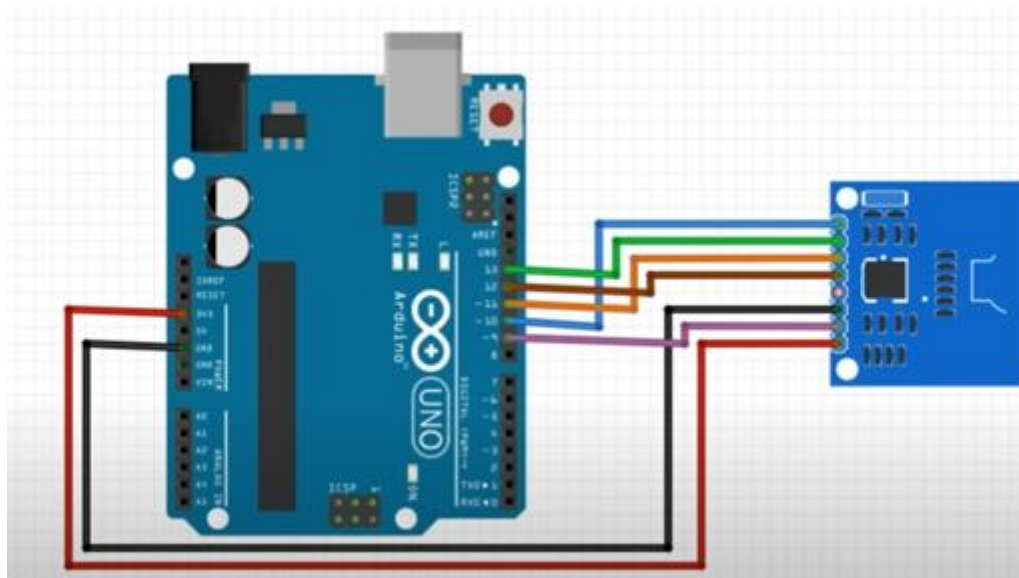
## Audio

- En el apartado del Audio hablaré de la implementación del mismo en el proyecto.
- Empezaré hablando del script encargado de controlar el audio general el **AudioController**, en resumen, es una clase central para la gestión del audio en un juego de Unity. Facilita la reproducción y gestión de diferentes tipos de audio, asegurando que la experiencia sonora del jugador sea fluida y coherente a lo largo del juego. El uso del patrón Singleton y el método DontDestroyOnLoad garantiza que el controlador de audio no se reinicie o duplique entre escenas, manteniendo la continuidad del audio.
- Prosigo hablando del script asociado a un objeto que está en todas las ventanas del proyecto para asegurar el audio, el **AMLoader**(Audio Manager Loader) es un componente que garantiza la existencia de un AudioController en el juego. Si no hay un AudioController presente, crea uno nuevo y se asegura de que persista a través de los cambios de escena, manteniendo la gestión del audio consistente durante todo el juego.
- Por último, hablaré del control del audio de ajustes, **AudioSettings** es el que controla el audio general, la música y los efectos de sonido del proyecto. Básicamente gestiona la inicialización de los valores de los sliders, aplica los ajustes de volumen a los grupos de audio correspondientes y actualiza visualmente el estado del slider para indicar si el audio está silenciado o no.



## Elegoo Uno y su implementación en Unity

- Aquí os hablaré un poco de cómo fue la construcción del Elegoo Uno y su implementación en Unity, lo primero seguí una página web para saber cómo montar el Elegoo Uno con el lector que fue la siguiente:



*Ilustración 27-Unión del Arduino y el lector*

- Luego tocaba programar la escritura de las tarjetas NFC que estaban en blanco, que lo realice con un código en C++ desarrollado en el IDE de arduino y el código interactúa con un lector RFID para escribir el valor 0, 1 ó 2 (Dependiendo del personaje que desees escribir en la tarjeta) en el bloque 5 de una tarjeta NFC y proporciona mensajes en el monitor serie sobre el éxito o fracaso de la operación.
- Y ya por último el código de lectura su función es leer datos del bloque 5 de una tarjeta RFID utilizando un lector MFRC522 y muestra los valores '0', '1', y '2' en el monitor serie. Además, maneja los errores de lectura e interrumpe la comunicación con la tarjeta después de la operación.
- Y la unión la hice abriendo en Unity en la ventana deseada las comunicaciones con el puerto serie necesario especificando el nombre del puerto y su puerto como tal, que en el Elegoo era COM5 y el puerto 9600 y para la selección uso la variable que está vinculada al Elegoo Uno y leo lo que envía por el puerto serie y lo uso para seleccionar el personaje según un array de personajes por eso se escribe y se envía 0, 1 ó 2.

## Juego

- La implementación del juego se puede dividir en 5 puntos básicos, personajes, la habilidad única de los personajes, los enemigos, los turnos de las batallas y las cartas.

### Personajes

- La implementación de los personajes empezó en la creación de sus **Scriptable Objects**, los cuales contienen la imagen del personaje y un Scriptable Object de su habilidad y su implementación es mediante un script que está en la pantalla de batalla que según el resultado pasado en la ventana de selección de personaje inicializa cada personaje según el nombre de su imagen.

### Habilidades

- Las habilidades solo contienen la imagen que usa su habilidad y una descripción de qué hacen, la implementación de que hacen está en un script que es llamado si le das clic al botón de la habilidad.

### Enemigos

- Los enemigos funcionan bajo la misma lógica de los personajes, pero con el cambio de que ellos son elegidos de forma aleatoria, desde su vida hasta su apariencia.

### Turnos

- Los turnos llevan el tempo que el usuario desee, cuando un usuario acabe de colocar sus cartas en el tablero le dará clic en el botón de acabar turno el cual le impedirá seleccionar nada hasta que no vuelva a ser su turno, explicar que como tal el enemigo no tiene turno. El “turno” del enemigo es una secuencia de acciones continuas hasta que todas las cartas hacen sus acciones y le vuelve a tocar a usuario poner sus cartas. El enemigo funciona con una “IA” que elige una o dos cartas de forma aleatoria. Al acabar todas las acciones el mazo de cartas te devuelve 1 o 2 cartas según hayas usado en tu turno.

### Cartas

- Las cartas también son Scriptable Objects, los cuales únicamente contienen la imagen de la carta, a la hora de la creación de la carta usa un **prefab** de la carta según sea una carta de un enemigo o un personaje. Las cartas se crean mediante una clase Card que la crea con su tipo de carta, su valor y si es maldita o no, además de que esa misma clase configura como funcionará la carta a nivel de movimiento por el escenario, pero la organización de las cartas en mano se lleva en otra clase.

### Mazo

- El mazo solo es una textura desde la cual se generan las cartas y se llevan a tu mano al acabar tu turno.

### Cartas en mano

- La mano de la partida se genera de forma aleatoria haciendo uso de la clase **HandController** con la cual te lleva las cartas a la mano al inicio de los turnos e indica cuántas cartas debes de tener.

### Animaciones

- La implementación de las animaciones fue mediante el mismo Unity que permite crear animaciones, lo único que tuve que hacer es crear un trigger/activador que ejecute esa animación cuando yo lo desee.

## Pruebas

- Establecí un plan de pruebas, para verificar que el proyecto satisface todos los requisitos descritos en el apartado de **Requisitos Funcionales**. Realizaremos una serie de pruebas del tipo “**Caja negra**”, en el que me fijaré en las entradas y salidas que produce el programa.

## Unitarias

- Mi proyecto de Unity depende en gran medida de componentes y sistemas complejos que involucran interacciones visuales y de físicas difíciles de simular en un entorno de pruebas unitarias. Dado este contexto, considero que las pruebas unitarias no serían eficaces para garantizar la calidad del juego.

## Pruebas de Caja Negra

- Empezaremos realizando las pruebas de caja negra:

Caso de Prueba: CP1.1	
Entrada	Resultado esperado
Seleccionar un personaje con el lector RFID	El usuario ha seleccionado un personaje
	Resultado obtenido
	Prueba correcta

Caso de Prueba: CP2.1.1	
Entrada	Resultado esperado
El Enemigo/Jefe selecciona una carta mediante IA	El Enemigo/Jefe seleccionó una carta mediante IA
	Resultado obtenido
	Prueba correcta

Caso de Prueba: CP2.1.2	
Entrada	Resultado esperado
El Enemigo/Jefe selecciona dos cartas mediante IA	El Enemigo/Jefe seleccionó dos cartas mediante IA
	Resultado obtenido
	Prueba correcta

Caso de Prueba: CP2.1.3	
Entrada	Resultado esperado
El Usuario selecciona una carta	El Usuario seleccionó una carta
	Resultado obtenido
	Prueba correcta

Caso de Prueba: CP2.1.4	
Entrada	Resultado esperado
El Usuario selecciona dos cartas	El Usuario seleccionó dos cartas
	Resultado obtenido
	Prueba correcta

Caso de Prueba: CP2.2.1	
Entrada	Resultado esperado
El Usuario usa la habilidad única del personaje seleccionado	El Usuario usa la habilidad única del personaje seleccionado
	Resultado obtenido
	Prueba correcta

Caso de Prueba: CP2.3.1	
Entrada	Resultado esperado
El Usuario finaliza su turno	El Usuario finaliza su turno
	Resultado obtenido
	Prueba correcta

## Capítulo 6 - Manual, Ampliaciones y Problemas y sus Soluciones

### Manual

- Dado que mi juego incluye un tutorial incorporado que guía a los usuarios a través de las mecánicas y la jugabilidad, opté por no incluir un manual separado en la memoria de mi TFG.

### Ampliaciones

- La verdad que este proyecto lo he disfrutado mucho, pero se me ha quedado un poco grande para completar todos los aspectos que quería cubrir, por ejemplo, me gustaría haber añadido más funciones al NFC para el juego, como habilidades u otro tipo de funciones. Además, me hubiera gustado poner más personajes con más habilidades, añadir también un sistema de logros personalizado dependiendo de con qué personaje ganaste a algún jefe. Otro punto que me hubiera gustado cubrir es el diseño e impresión de los personajes en 3D, quitando esos puntos creo que poco más puedo ampliar el proyecto.

### Problemas encontrados y sus soluciones

- En este apartado voy a comentar los problemas técnicos que he tenido a lo largo del proyecto.
  1. Uno de los problemas más grandes que he tenido ha sido el apartado del Elegoo Uno, ya que era mi primera vez utilizándolo y desconocía en qué lenguaje de programación, que resultó acabando siendo C + +.
  2. Un problema que tuve fue que el módulo de RFID que compré para el proyecto no vino soldado y tuve que pedir ayuda al profesor Javier Gutiérrez del departamento de informática del IES Comercio, para realizar la unión.
  3. Otro apartado que me costó relacionado con el Elegoo Uno fue la implementación del mismo en el sistema de Unity.
  4. Aprender Unity desde cero y se solventó leyendo mucha documentación y con un curso.
  5. Un apartado donde tuve problemas fue en la implementación del audio ya que Unity trabaja con escenas y al hacer el cambio entre ellas el objeto que controlaba el audio se borraba, pero con varios videos conseguí entender cómo hacer que ese objeto persista.
  6. Luego el resto de apartados con mayor o menor problema de implementación, pero fue de forma correcta en cuanto a la organización que llevaba.

## Capítulo 7 - Final

### Conclusión

- La conclusión que me puedo llevar de este proyecto es que he disfrutado mucho de aprender a utilizar Unity y a programar en C + + para el Elegoo Uno, que me ha servido mucho para aprender a desarrollar un proyecto real y aprender a organizar los tiempos de desarrollo de cada apartado. En resumen, estoy muy orgulloso de lo que he conseguido crear.

### Bibliografía

*Tutorial módulo Lector RFID RC522.* (s. f.). Naylamp Mechatronics - Perú.

[https://naylampmechatronics.com/blog/22\\_tutorial-modulo-lector-rfid-rc522.html](https://naylampmechatronics.com/blog/22_tutorial-modulo-lector-rfid-rc522.html)

Enjoy Mechatronics. (2022, 17 junio). *How to use RFID (RC522) with Arduino - Easy*

*Tutorial* [Vídeo]. YouTube. <https://www.youtube.com/watch?v=aIO6KdzITzU>

DBM. (2018, 30 noviembre). *Comunicando Unity y Arduino - #1 - Tutorial español 2019*

[Vídeo]. YouTube. <https://www.youtube.com/watch?v=0R3mECWm1E8>

colaboradores de Wikipedia. (s. f.). *Wikipedia, la enciclopedia libre.*

<https://es.wikipedia.org/wiki/Wikipedia:Portada>

Udemy. (s. f.). <https://www.udemy.com>.

<https://www.udemy.com/share/1071dE3@psV0QXSH9S8xedTbJYKJx2QnUcbNGoHpjy>

[phz4faYw5sdlYeNge3HIU9vr\\_TxqgHQA==/](https://www.udemy.com/share/1071dE3@psV0QXSH9S8xedTbJYKJx2QnUcbNGoHpjyphz4faYw5sdlYeNge3HIU9vr_TxqgHQA==/)

FiftySounds. (2020, 27 agosto). *Música sin Copyright Gratis para Uso Comercial |*

*FiftySounds.* <https://www.fiftysounds.com/es/>