



# Institut Puig Castellar

Santa Coloma de Gramenet

## RETROPIXEL

(Projecte de desenvolupament)

CFGM Sistemes Microinformàtics i Xarxes



Alejandro Escobar Garcia  
SMX2D



Aquesta obra està subjecta a una llicència de [Reconeixement-NoComercial-CompartirIgual 3.0 Espanya de Creative Common](https://creativecommons.org/licenses/by-nc-sa/3.0/es/)

## **B) GNU Free Documentation License (GNU FDL)**

Copyright © ANY Alejandro Escobar García

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.

A copy of the license is included in the section entitled "GNU Free Documentation License".

## **C) Copyright**

© (l'autor/a)

Reservats tots els drets. Està prohibit la reproducció total o parcial d'aquesta obra per qualsevol mitjà o procediment, compresos la impressió, la reprografia, el microfilm, el tractament informàtic o qualsevol altre sistema, així com la distribució d'exemplars mitjançant lloguer i préstec, sense l'autorització escrita de l'autor o dels límits que autoritzi la Llei de Propietat Intel·lectual.

## **Resum del projecte (màxim 250 paraules):**

Este proyecto consiste en el diseño y desarrollo de RetroPixel, una máquina recreativa en miniatura basada en una Raspberry Pi 5 que permite jugar tanto a videojuegos clásicos mediante emulación como a juegos originales desarrollados específicamente para el proyecto.

El objetivo principal era construir un dispositivo físico completamente funcional que combinara la construcción de la recreativa, la configuración del software de emulación y el desarrollo de videojuegos propios, manteniendo en todo momento una estética y experiencia fiel al espíritu de las recreativas clásicas.

Para conseguirlo, el proyecto se dividió en cuatro fases principales. En primer lugar se diseñó y construyó la recreativa desde cero utilizando madera y metacrilato, integrando todos los componentes electrónicos necesarios como la pantalla, la botonera y los altavoces. A continuación se instaló y configuró Batocera como sistema operativo y gestor de emulación sobre la Raspberry Pi 5. Paralelamente se desarrollaron dos videojuegos originales con Godot Engine 4 y GDScript, siguiendo una progresión de complejidad creciente. Finalmente, los juegos fueron exportados como ejecutables Linux ARM64 e integrados en Batocera para ser accesibles desde el menú principal.

El proyecto ha resultado en una recreativa completamente funcional, cumpliendo los objetivos planteados inicialmente. A lo largo del desarrollo se han adquirido conocimientos en construcción física, configuración de sistemas Linux, desarrollo de videojuegos y resolución de problemas técnicos reales, lo que ha convertido este proyecto en una experiencia de aprendizaje completa y significativa.

## **Paraules clau (entre 4 i 8):**

Retropixel

Retro

Jugador

Estètica

Màquina

Recreativa

**Abstract (in English, 250 words or less):**

This project involves the design and development of RetroPixel, a miniature arcade machine based on a Raspberry Pi 5 that allows users to play both classic video games via emulation and original games developed specifically for the project.

The main objective was to build a fully functional physical device that combined the construction of the cabinet, the configuration of the emulation software and the development of original video games, whilst maintaining an aesthetic and experience faithful to the spirit of classic arcade machines at all times.

To achieve this, the project was divided into four main phases. Firstly, the cabinet was designed and built from scratch using wood and acrylic, integrating all the necessary electronic components such as the screen, the button panel and the speakers. Next, Batocera was installed and configured as the operating system and emulation manager on the Raspberry Pi 5. In parallel, two original video games were developed using Godot Engine 4 and GDScript, following a progression of increasing complexity. Finally, the games were exported as Linux ARM64 executables and integrated into Batocera so they could be accessed from the main menu.

The project has resulted in a fully functional arcade machine, meeting the objectives set out at the outset. Throughout the development process, knowledge has been gained in hardware construction, Linux system configuration, video game development and the resolution of real-world technical problems, making this project a comprehensive and meaningful learning experience.

Translated with DeepL.com (free version)

**Keywords (entre 4 i 8):**

Retropixel

Retro

Gamer

Aesthetics

Arcade

Machine

# Índex

<b>1</b>	<b>Introducción</b>	<b>1</b>
1.1	Contexto	1
1.2	Justificación	1
1.3	Objetivos	2
1.4	Estrategia y planificación del proyecto	2
1.5	Metodología de trabajo	2
1.6	Estudio económico y presupuestario	3
<b>2</b>	<b>Descripción del proyecto</b>	<b>4</b>
2.1	Análisis de requisitos	4
2.2	Tecnologías	5
2.3	Estructura del proyecto	6
2.4	Descripción de los componentes	6
2.5	Definición de las funcionalidades	9
<b>3.</b>	<b>Diseño y construcción de la estructura</b>	<b>11</b>
3.1	Diseño previo y planificación	11
3.2	Materiales escogidos y alternativas consideradas	12
3.3	Proceso de corte y lijado	12
3.4	Pintar la recreativa	17
3.5	Construcción de la botonera	18
3.6	Sustitución de pantalla	19
3.7	Cambios en la electrónica de la estructura	19
3.8	Cambio de botones y resultado físico final	21
<b>4.</b>	<b>Instalacion y configuracion de Batocera</b>	<b>23</b>
4.1	Instalación de Batocera en la tarjeta microSD	23
4.2	Configuración de los controles	24
4.3	Configuración de los emuladores y carga de ROMs	24
<b>5.</b>	<b>Juegos en Godot</b>	<b>25</b>
5.1	Entorno de desarrollo y aprendizaje	25
5.2	Shooter invaders	25
5.3	Shooter espacial	27
<b>6.</b>	<b>Integración de los juegos en Batocera</b>	<b>30</b>
6.1	Exportación desde Godot Engine	30
6.2	Transferencia a la Raspberry Pi	30
<b>7.</b>	<b>Conclusions</b>	<b>32</b>
7.1	Conclusions generals del projecte	32
7.2	Consecució dels objectius	32
7.3	Valoració de la metodologia i planificació	32
7.4	Visió de futur	32
<b>8.</b>	<b>Glossari</b>	<b>33</b>

<b>9. Webgrafia</b>	<b>34</b>
<b>10. Anexos</b>	<b>35</b>

# 1 Introducción

Este proyecto consiste en el diseño y la creación de RetroPixel, una máquina recreativa en miniatura pensada para revivir la experiencia de los videojuegos clásicos. La idea principal es construir un dispositivo físico funcional que permita jugar tanto a títulos retro mediante emulación como a juegos propios desarrollados durante el proyecto. Para conseguirlo, se combinarán conocimientos de hardware y software, utilizando herramientas como la Raspberry Pi, Batocera y Godot Engine. El objetivo es lograr un resultado sencillo, funcional y fiel al espíritu de las recreativas de antes.

## 1.1 Contexto

Actualmente, los videojuegos han evolucionado hacia sistemas cada vez más complejos, con gráficos avanzados, conexión en línea y múltiples funcionalidades. Sin embargo, existe una comunidad interesada en el retrogaming, que busca recuperar la simplicidad y la diversión de los juegos clásicos. Gracias a esta comunidad, se ha popularizado un movimiento que se basa en la creación de dispositivos electrónicos personalizados, combinando hardware de bajo coste como la Raspberry Pi con software libre para dar lugar a proyectos innovadores y accesibles. El interés personal que tengo por esta cultura ha sido un factor motivador clave para el desarrollo de este proyecto, que representa una oportunidad para poner en práctica conocimientos técnicos en un contexto creativo y significativo.

## 1.2 Justificación

Este proyecto es relevante porque combina aprendizaje técnico con un enfoque práctico y creativo. No se trata únicamente de construir una máquina, sino de comprender y aprender el funcionamiento del hardware y el software implicados en cada fase del desarrollo.

En cuanto a las herramientas escogidas, he seleccionado la Raspberry Pi5 por su amplia comunidad, toda la documentación disponible y su capacidad suficiente para ejecutar emuladores de sistemas clásicos a un coste reducido, frente a alternativas como Arduino, que no dispone de la potencia necesaria, o minimáquinas más caras que no aportan ventajas significativas para este proyecto. En el tema del sistema de emulación, he preferido utilizar Batocera en vez de otras opciones como RetroPie por la simple razón de la compatibilidad con la Pi5 que tengo.

Finalmente, Godot Engine se ha elegido para el desarrollo de los juegos propios por ser un motor de código abierto, ligero, con soporte para exportación a Linux y con una curva de aprendizaje adecuada para el nivel del proyecto.

## 1.3 Objetivos

### 1.3.1 Objetivo general

Crear una maquina recreativa en miniatura funcional que permita jugar a videojuegos tanto clásicos como propios, manteniendo una estética y experiencia retro.

### 1.3.2 Objetivos específicos

- Diseñar la estructura física de la máquina recreativa.
- Montar el hardware utilizando una Raspberry Pi.
- Instalar y configurar Batocera para la emulación de sistemas clásicos.
- Desarrollar videojuegos originales con Godot Engine, adaptados a los controles de la recreativa.
- Integrar todos los elementos en un sistema funcional y coherente.

## 1.4 Estrategia y planificación del proyecto

Para llevar a cabo el proyecto se ha definido una estrategia dividida en cinco fases secuenciales que permiten abordar el desarrollo de manera ordenada y controlada.

La fase 1 corresponde a la investigación y el diseño, donde se definen los requisitos del sistema, se eligen los componentes y se planifica la estructura física y lógica del proyecto. La fase 2 es la construcción de la recreativa, que incluye el corte y montaje de los materiales, así como la integración de los elementos electrónicos como la pantalla, los botones y el joystick. La fase 3 se centra en la configuración del software, con la instalación de Batocera en la Raspberry Pi y el ajuste de los controles y la interfaz. La fase 4 corresponde al desarrollo de los juegos propios con Godot Engine, desde el diseño hasta la exportación e integración en el sistema. Por último, la fase 5 es la de integración y pruebas, donde se verifica el funcionamiento global del sistema y se corrigen las incidencias detectadas.

## 1.5 Metodología de trabajo

La metodología de trabajo se basará en una combinación de trabajo práctico dividido por entornos, organizando las tareas según dónde se realicen. Por un lado, en casa se llevará a cabo el desarrollo de los videojuegos utilizando Godot Engine, así como el diseño y construcción de la carcasa de la máquina y la integración de la botonera. Por otro lado, en clase se trabajará principalmente con la Raspberry Pi, centrandó el esfuerzo en la instalación y configuración de Batocera y la realización de pruebas de funcionamiento.

Para el seguimiento de las tareas se elaborará un diagrama de Gantt dividido en tareas y semanas. En función del tiempo necesario para completar cada tarea se asignarán más o menos semanas, lo que permitirá visualizar el progreso del proyecto y detectar posibles desviaciones respecto a la planificación inicial.

## 1.6 Estudio económico y presupuestario

El proyecto requiere una serie de componentes como la Raspberry Pi, pantalla, botones, joystick y materiales para la carcasa. El coste total será relativamente bajo en comparación con una recreativa tradicional, lo que lo hace accesible. Además, el mantenimiento es mínimo, ya que se trata de un sistema sencillo. Aunque no está pensado como un producto comercial, podría tener potencial en ese sentido si se mejora y se produce en mayor escala.

Componente	Descripcion	Precio aproximado
Raspberry Pi5 8GB	Unidad central de procesamiento	0€
Pantalla LCD	Monitor	0€
Botones, joystick y placa de control	Controladores de entrada	30€
Madera y metacrilato	Material para la estructura	0€
Pintura	Decoraciones	10€
Cables y conectores	Conexiones internas	0€
Tornillos y tuercas	Fijaciones para el montaje	0€
Fuente de alimentacion	Suministro electrico	0€
Adaptador VGA/HDMI	Adaptador para el monitor	7€
<b>Precio total estimado</b>		<b>47€</b>

Los precios que se ubican en 0€ es porque esos materiales han sido reutilizados de otros proyectos o sistemas.

## 2 Descripción del proyecto

### 2.1 Análisis de requisitos

#### 2.1.1 Requisitos funcionales

Los requisitos funcionales definen qué debe ser capaz de hacer el sistema. Para que RetroPixel cumpla con su propósito, se han establecido los siguientes:

- La máquina debe encenderse y funcionar de forma estable sin interrupciones.
- Debe permitir jugar a videojuegos clásicos mediante emulación a través de Batocera.
- Debe permitir ejecutar los juegos propios desarrollados con Godot Engine.
- Los controles (joystick y botones) deben responder correctamente en todo momento.
- Debe mostrar imagen en una pantalla externa conectada mediante HDMI.
- Debe reproducir sonido a través de los altavoces integrados.
- Debe contar con un sistema de navegación sencillo e intuitivo mediante menú.

#### 2.1.2 Requisitos no funcionales

- El sistema debe ser fácil de usar para cualquier persona sin conocimientos técnicos.
- La estructura debe ser resistente gracias al uso de madera como material principal.
- El diseño debe ser visualmente atractivo y mantener una estética retro coherente.
- El sistema debe ser estable y no presentar fallos frecuentes durante el uso.
- El montaje interno debe ser seguro, sin cables sueltos ni conexiones peligrosas.
- El coste total del proyecto debe mantenerse en un rango económicamente accesible.

## 2.2 Tecnologías

### 2.2.1 Comparativa de las tecnologías valoradas

Antes de escoger las herramientas definitivas, valore varias alternativas tanto para el sistema de emulación como para el desarrollo de los juegos propios.

En cuanto al motor de los videojuegos, tuve en cuenta GDevelop y Construct 3 por su facilidad de uso, pero ambos se basan en un sistema de programación visual por bloques que limita el control de la lógica del juego y no permite trabajar directamente con código. GameMakerStudio ofrece mayor potencia y programación en código propio, pero su complejidad lo hacía menos adecuado para los objetivos del proyecto.

Respecto al sistema de emulación, mire RetroPie como alternativa a Batocera. RetroPie es la opción más conocida y documentada, pero presenta una limitación importante: su compatibilidad oficial se extiende únicamente hasta la Raspberry Pi 4, lo que lo descarta directamente para este proyecto al utilizar una Raspberry Pi 5. Batocera se destacó como la única opción que combinaba compatibilidad nativa con la Raspberry Pi 5, una interfaz cuidada y una configuración sencilla de controles y sistemas.

### 2.2.2 Tecnologías escogidas

Se escogió Godot Engine como motor de desarrollo de los juegos propios por ser una herramienta gratuita y de código abierto, lo que facilita el acceso y la flexibilidad en el desarrollo. Su enfoque en 2D permite crear juegos arcade de forma eficiente, mientras que su lenguaje GDScript es sencillo y similar a otros lenguajes. Además, ofrece control completo sobre la lógica y las colisiones, y permite exportar los juegos fácilmente para ejecutarlos en la Raspberry Pi.

Se escogió Batocera como sistema de emulación por ofrecer una experiencia más pulida que sus alternativas, con una interfaz intuitiva, configuración sencilla de controles y soporte nativo para una gran variedad de sistemas clásicos, todo ello sin necesidad de conocimientos técnicos avanzados.

## 2.3 Estructura del proyecto

RetroPixel está formado por varias partes que trabajan conjuntamente. Por un lado, se encuentra la estructura física de la recreativa, construida principalmente con madera para la estructura principal y metacrilato en la zona de la botonera para un mejor acabado visual.

Dentro de esta estructura está la Raspberry Pi 5, que actúa como el cerebro del sistema. Esta se conecta a la pantalla mediante un cable HDMI con adaptador a VGA ya que el monitor que se utiliza únicamente dispone de entrada VGA. Los controles (joystick y botones) se conectan a la Raspberry Pi a través de una placa controladora que traduce las pulsaciones en señales reconocibles por el sistema. Los altavoces se conectan directamente a la salida de audio del adaptador. Todo el sistema excepto la Raspberry Pi 5 se alimenta mediante una fuente de alimentación, mientras que, la Raspberry Pi5 se conecta a la corriente a través de su propia fuente.

A nivel de software, Batocera gestiona tanto la emulación de sistemas clásicos como la ejecución de los juegos propios. Estos juegos, desarrollados con Godot Engine, se integran en Batocera como si fueran títulos de un sistema personalizado y son accesibles directamente desde el menú principal junto al resto de emuladores.

El desarrollo de los juegos propios sigue una progresión deliberada en cuanto a complejidad, empezando por mecánicas simples y directas propias de los arcades más básicos, e incrementando gradualmente la profundidad de la lógica, el diseño y la interacción con el jugador a lo largo de los tres videojuegos.

Esta progresión permite que el proyecto no solo cumpla un objetivo técnico, sino que también refleje una evolución real en el aprendizaje del motor de desarrollo.

## 2.4 Descripción de los componentes

En esta sección se repasará todos y cada uno de los componentes utilizados en la creación de RetroPixel y sus juegos.

### 2.4.1 ESTRUCTURA (MADERA Y METACRILATO)

Es la parte física de la recreativa. La madera forma la estructura principal y le aporta resistencia y rigidez al conjunto. El metacrilato se utiliza en la zona de la botonera para conseguir un acabado más limpio y visualmente atractivo. Las diferentes piezas se fijan entre sí mediante tornillos, tuercas y pegamento, garantizando una estructura sólida y desmontable si fuera necesario. El exterior se trata con pintura para darle una estética retro acorde con el proyecto y proteger la madera.

## 2.4.2 RASPBERRY PI 5

Es el componente central de todo el sistema. Se encarga de ejecutar Batocera con los juegos de emulación y los juegos propios, actuando como la unidad de procesamiento principal.

Dispone de un procesador quad-core Cortex-A76 a 2,4 GHz y 8GB de RAM, lo que garantiza un rendimiento más que suficiente para los requisitos del proyecto.



## 2.4.3 CONTROLES (BOTONES Y JOYSTICK)

El panel de control incluye 6 botones de juego, un joystick, un botón de Start y un botón de Coin. Permiten interactuar con los juegos de forma similar a una recreativa clásica y están diseñados para soportar un uso continuado.

## 2.4.4 PLACA CONTROLADORA

Es la encargada de conectar los botones y el joystick con la Raspberry Pi, traduciendo las pulsaciones físicas en señales de entrada que el sistema operativo reconoce como si fuera un mando o teclado USB estándar.

## 2.4.5 PANTALLA / MONITOR

Muestra la imagen del sistema y los juegos. Se conecta a la Raspberry Pi mediante un cable HDMI con adaptador a VGA, adaptándose así al tipo de entrada disponible en el monitor utilizado.

## 2.4.6 ALTAVOCES

Permiten reproducir el sonido de los juegos. Se conectan directamente a la salida de audio de la Raspberry Pi y van integrados dentro de la recreativa para mantener una apariencia compacta y ordenada.

## 2.4.7 FUENTE DE ALIMENTACIÓN

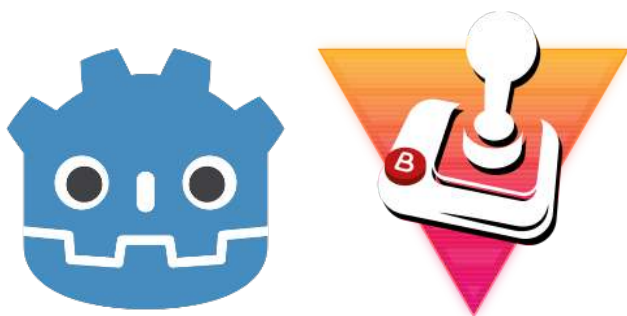
La fuente de alimentación utilizada es una fuente de PC estándar, encargada de suministrar energía a todo el conjunto del sistema, garantizando un funcionamiento estable y continuo de todos los componentes integrados en la recreativa.

## 2.4.8 CABLES Y CONEXIONES

Incluyen el cable HDMI con adaptador VGA para la pantalla, los cables de conexión entre la placa controladora y los botones, y el resto de conexiones internas necesarias para que todos los componentes funcionen de forma conjunta y ordenada.

## 2.4.9 SOFTWARE (BATOCERA Y GODOT)

Batocera se encarga de la emulación de videojuegos clásicos y actúa como sistema operativo principal de la máquina, gestionando tanto los emuladores como los juegos propios. Godot Engine se utiliza para desarrollar los juegos originales del proyecto, que una vez exportados se integran en Batocera y se ejecutan directamente desde el menú principal.



## 2.4.10 TARJETA MICROSD

Es el elemento de almacenamiento principal del sistema. En ella se instala Batocera como sistema operativo, y también se almacenan las ROMs de los juegos clásicos y los ejecutables de los juegos propios desarrollados con Godot. Su capacidad debe ser suficiente para albergar todos estos contenidos sin problemas de espacio.

## 2.4.11 ADAPTADOR HDMI/VGA

Es el elemento que permite conectar la Raspberry Pi 5 con el monitor de la recreativa. Dado que la Raspberry Pi únicamente dispone de salida HDMI y el monitor utilizado solo acepta entrada VGA, este adaptador actúa como puente entre ambos, convirtiendo la señal digital HDMI en una señal analógica VGA sin pérdida de calidad visual apreciable.



## 2.5 Definición de las funcionalidades

En este apartado se describe conceptualmente cómo se implementan las funcionalidades del sistema, detallando qué permite hacer cada una, el proceso seguido para llevarlo a cabo y su estado de implementación actual.

### 2.5.1 Emulación de videojuegos clásicos

El sistema permite cargar y ejecutar ROMs de videojuegos clásicos de diferentes plataformas a través de Batocera. Este se instala en una tarjeta microSD y arranca directamente desde la Raspberry Pi. Una vez iniciado, detecta automáticamente las ROMs almacenadas en la carpeta donde se almacenan y las muestra organizadas por sistema en el menú principal, permitiendo al jugador navegar con el joystick y seleccionar el juego con los botones.

### 2.5.2 Shooter Espacial

Juego de tipo arcade inspirado en Space Invaders. Una vez desarrollado en Godot Engine, se exporta como ejecutable y se integra en Batocera como título de un sistema personalizado, siendo accesible desde el menú principal.

#### **2.5.2.1 Movimiento del jugador**

El jugador puede desplazar su nave horizontalmente a lo largo de la parte inferior de la pantalla. El sistema lee continuamente la entrada del joystick y traduce el movimiento horizontal en un desplazamiento de la nave dentro de los límites de la pantalla, impidiendo que salga del área de juego.

### **2.5.2.2 Sistema de disparo**

El jugador puede disparar proyectiles hacia arriba pulsando el botón de disparo. Al pulsarlo, el sistema genera un proyectil en la posición actual de la nave que se desplaza hacia arriba a velocidad constante, desapareciendo al salir de la pantalla o al impactar con un enemigo.

### **2.5.2.3 Eliminación de enemigos**

Cuando un proyectil impacta con un enemigo, el sistema detecta la colisión y elimina ambos elementos de la escena. Si todos los enemigos son eliminados, la partida finaliza con éxito.

## **2.5.3 Shooter vertical**

Juego de tipo arcade con mayor complejidad que el anterior. El jugador puede moverse libremente por toda la pantalla en un entorno de scroll vertical mientras los enemigos reaparecen de forma continua. Se exporta e integra en Batocera siguiendo el mismo proceso que el juego anterior.

### **2.5.3.1 Movimiento libre del jugador**

El jugador puede desplazar su personaje en cualquier dirección dentro de los límites de la pantalla. El sistema lee la entrada del joystick en ambos ejes y aplica el desplazamiento correspondiente, comprobando en todo momento que el jugador no sobrepase los bordes de la pantalla.

### **2.5.3.2 Sistema de disparo**

Al pulsar el botón de disparo, el sistema genera un proyectil en la posición actual del jugador que se desplaza en una dirección fija, desapareciendo al salir de la pantalla o al chocar contra un enemigo.

### **2.5.3.3 Reparición continua de enemigos**

El sistema genera enemigos periódicamente en posiciones aleatorias de la parte superior de la pantalla. Estos se desplazan hacia abajo y, al ser eliminados o salir de la pantalla, el sistema continúa generando nuevos, manteniendo una presión constante sobre el jugador durante toda la partida.

## 3. Diseño y construcción de la estructura

### 3.1 Diseño previo y planificación

El diseño de RetroPie se hizo desde cero, adaptando las dimensiones y la forma a los componentes disponibles y al espacio necesario para integrarlos correctamente. Antes de comenzar con el corte de la madera, se elaboró un boceto en papel con las medidas principales de cada pieza, definiendo la forma general de la recreativa, la posición de la pantalla, la zona de la botonera y el espacio interior destinado a alojar la Raspberry Pi, los cables y la fuente de alimentación.

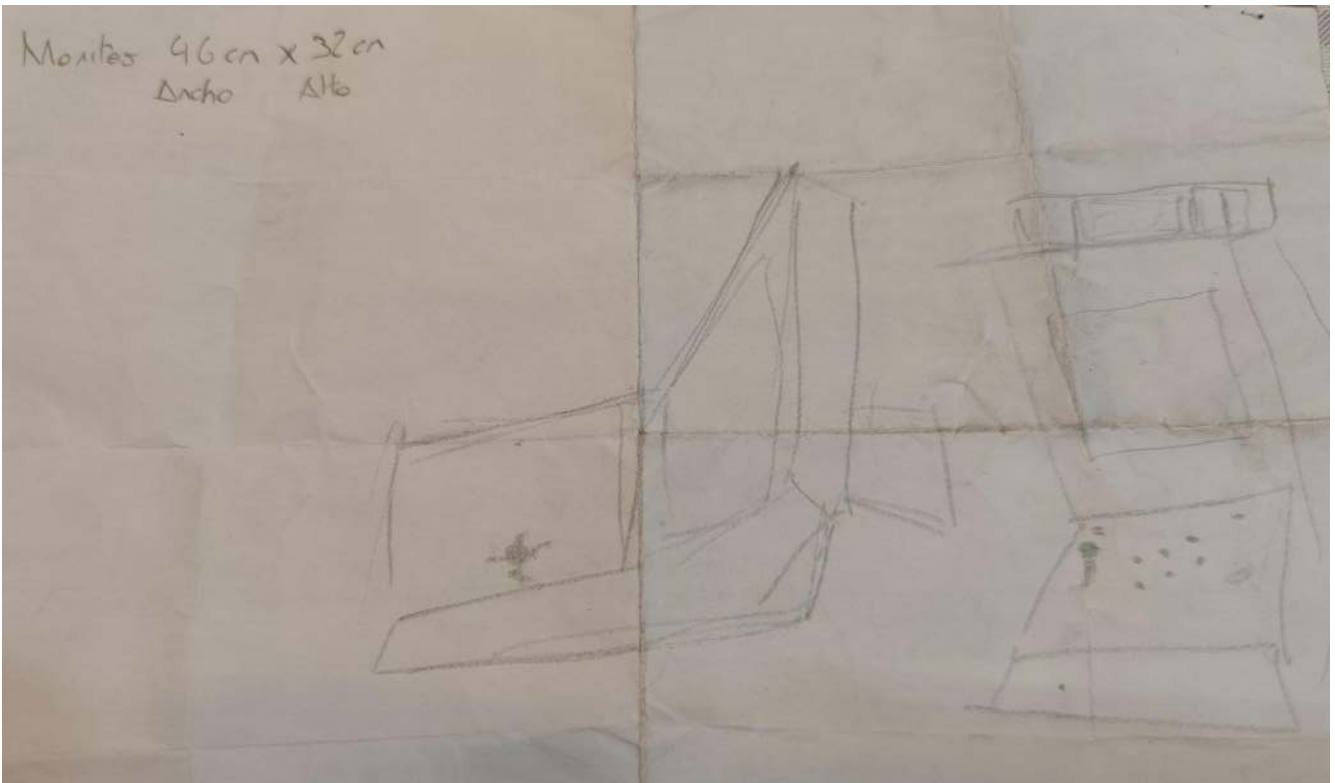


Imagen del boceto inicial con las medidas del monitor original en la parte superior izquierda

Este paso fue muy importante para tener en cuenta las dimensiones del material que necesitaria y tener una imagen clara de cómo sería en un futuro la recreativa. Me inspire en la estética de las recreativas bartop, que son las que son más pequeñas que una clásica.

## 3.2 Materiales escogidos y alternativas consideradas

Para la estructura principal se escogió madera por su facilidad de corte, su superficie lisa que facilita el acabado con pintura y su rigidez suficiente para una recreativa de estas dimensiones. Se valoró también el uso de contrachapado, que ofrece mayor resistencia a la humedad, pero se descartó por ser más difícil de trabajar y tener un acabado superficial menos uniforme.

Para la zona de la botonera se utilice metacrilato, que permite un acabado más limpio y visualmente atractivo en la zona de mayor interacción con el jugador. El ensamblaje de las piezas se realizó mediante tornillos y tuercas, combine esta solución junto al pegamento en depende que zonas, así podía elegir que partes estarán permanentemente juntas y que otras podrían separarse y/o desmontarse.

## 3.3 Proceso de corte y lijado

Teniendo en cuenta el boceto como referencia y con el uso de un lápiz marque el cómo debía de cortar cada una de las piezas empezando por las paredes laterales, que estarán sujetando toda la estructura.

Seguido a esto procedí a cortar las dos paredes de RetroPixel con una sierra



eléctrica para empezar a darle forma a todo. Para que fuera algo más estético,

opte por darle un poco de curvatura donde se alojaría la pantalla para que no se sienta un diseño tan rígido.



Una vez acabadas las dos paredes las tuve que lijar para quitar imperfecciones, dejarlo más limpio y además en la parte frontal quitarle el pico tan agresivo que tiene.



Una vez acabado el lijado de las paredes, la madera queda con una superficie más suave y uniforme, lo que facilita tanto su manipulación para los pasos posteriores. Seguido a esto con varios trozos de madera se recortaran y pegaran en la madera para utilizarlos como soporte del monitor, de los altavoces y para union para la madera inferior. Ya puesto el pegamento y la madera se sujeta con unos sargentos.



Mientras se seca el pegamento se procede a empezar a hacer el marco, una parte muy importante ya que delimita los bordes del monitor para hacer como si el monitor se fusiona con la recreativa, para esto se coge una lámina de madera, le marque un rectángulo en el centro junto a dos agujeros en la parte superior que es donde estarían los altavoces para que pueda salir el audio sin problemas.



Una vez terminado de cortar todo pudimos empezar el montaje de todas las maderas para tener el primer vistazo inicial a una fase muy preliminar de lo que sería en un futuro RetroPixel, con todas las piezas que tenía las empecé a juntar con tornillos y pegamento para que quede todo bien fijado y sin ningún tipo de problema.



Como vemos en la imagen esta es la primera fase de RetroPixel, ahora teníamos que continuar con la pintura y luego para terminar haríamos la botonera y toda la alimentación de nuestra recreativa.

### 3.4 Pintar la recreativa

Ahora seguía el proceso de pintar la maquina para darle un acabado colorido y con estetica retro, al principio pensamos en darle un aspecto colorido para cumplir con nuestras expectativas originales, solo que acabamos pintando las piezas con un color negro para que quede limpio.

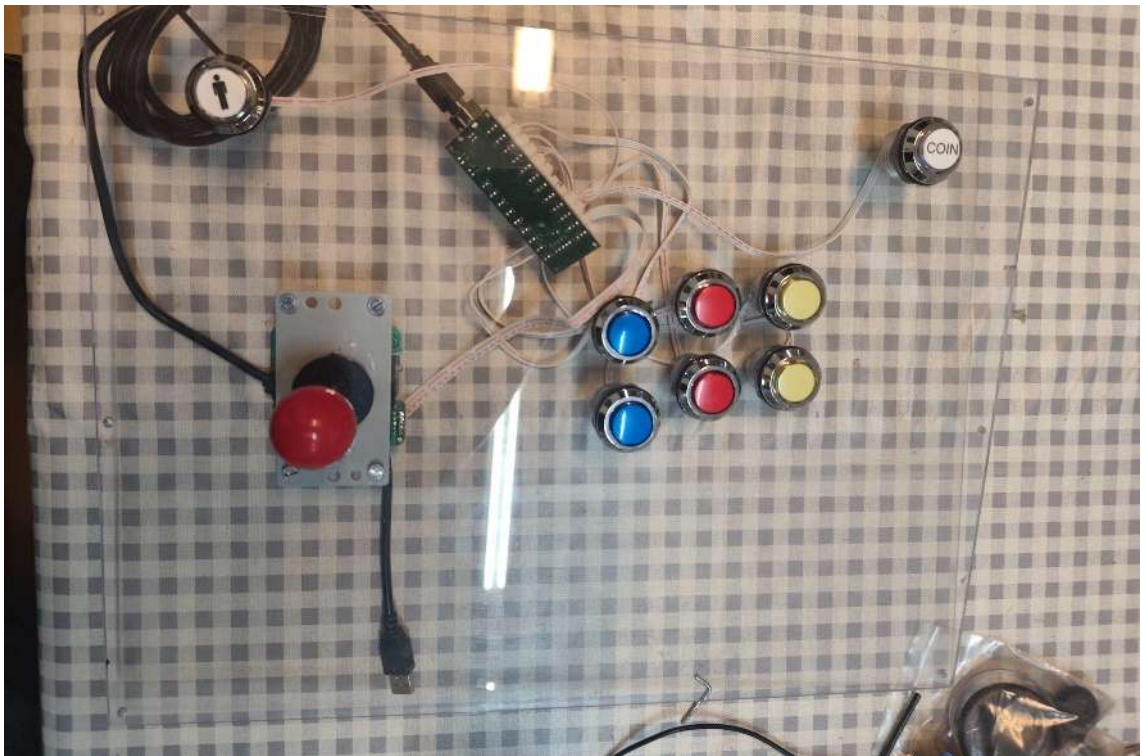


### 3.5 Construcción de la botonera

Con los botones ya comprados empecé a planificar la distribución de la botonera para que quede bien y que a la hora de jugar sea cómodo para el jugador, al basarme en las recreativas bartop, hice un diseño poniendo el joystick a la izquierda, 6 botones en la derecha y en la parte superior los botones de select y start.



Aquí tuvimos que hacer varios agujeros y pasarlos, después los atornille en el metacrilato, conecte cada uno de los botones y el joystick a la placa de control, seguido a eso la placa la conecte a la Raspberry y estaría en funcionamiento.



### 3.6 Sustitución de pantalla

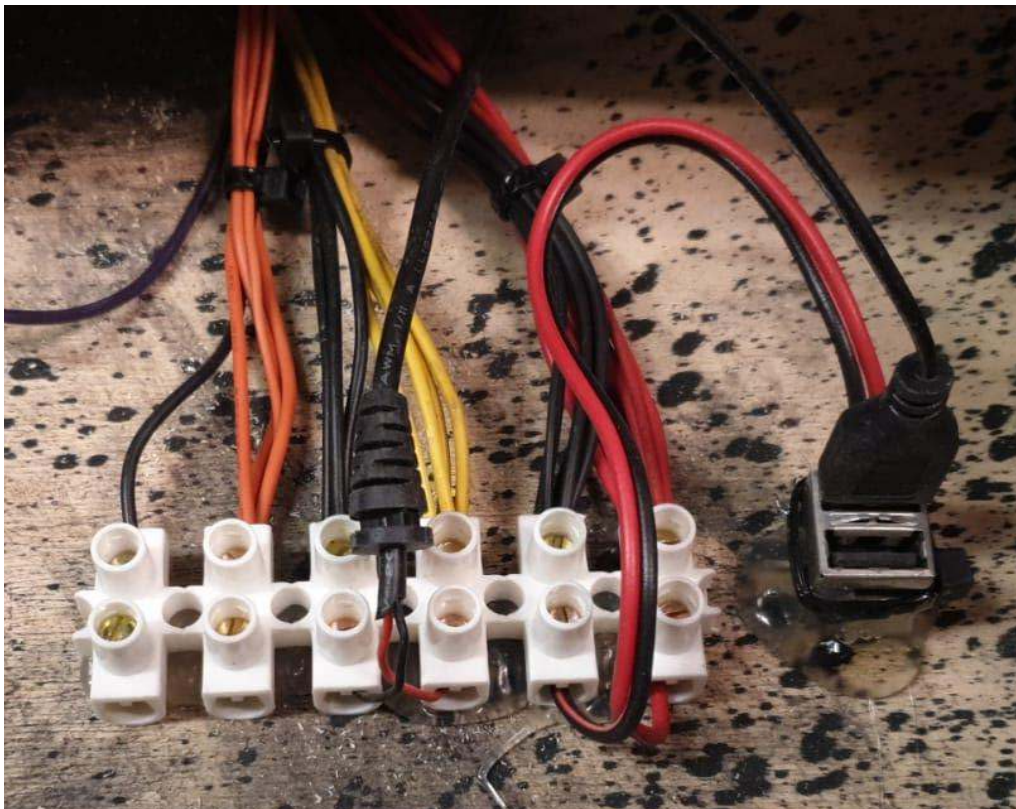
En el momento de haber terminado todo y empezar a hacer pruebas para comprobar de que todo funcionaba correctamente tuvimos un inconveniente bastante grande, el monitor que estábamos utilizando no tenía la misma frecuencia que daba la Raspberry Pi 5, por ende, el monitor era completamente inútil para este proyecto, por suerte tuvimos suerte, dio la casualidad que mi padre tenía varios monitores que se podían utilizar de diferentes tamaños, probamos varios y algunos tenían el mismo problema, otros daban señal pero le quitaban el sonido a los altavoces. Al final mi padre y yo nos acabamos decantando por un monitor que era más ancho que el original que estábamos utilizando, pero no nos daba ningún problema, lo único que tuvimos que hacer, fue cortar un poco más las paredes de la recreativa para poder ajustar la pantalla.

Este es el resultado final de la pantalla.



### 3.7 Cambios en la electrónica de la estructura

Aprovechando que tenía una fuente de alimentación de PC, se decidió utilizarla por completo para alimentar a todos los componentes de la máquina, con el uso de unas regletas, un puerto USB y silicona para adherir todo a la máquina, se ingenio un sistema que ahorra espacio y tiempo, además de dejarlo con un toque más “profesional”.



También se hizo una última modificación a la madera que consistía en hacer un agujero en el lateral de una de las paredes para que pueda pasar el cable de alimentación a la fuente sin complicaciones y en la pared trasera hacer otro agujero en la zona de la fuente para que el ventilador respire correctamente.



### 3.8 Cambio de botones y resultado físico final

Acabando todo tuvimos una última complicación, uno de los botones no respondía correctamente y no daba señal la gran parte de las veces que se pulsaba, afortunadamente en el pack de botones que se compro venian 8 botones de juego y no los 6 que se utilizaron, asi que se cambiaron los botones rojos por los verdes, lo que al final del dia quedaba incluso mas estetico, ya que los botones seguian el orden de “azul-rojo-amarillo” y ahora cambiaba a “azul-verde-amarillo” dando un efecto mas limpio con la paleta de colores, siendo un cambio de color mas tranquilo y no tan abrupto.

Habiendo cambiado esto, RetroPixel había finalizado toda su parte física teniendo como resultado el siguiente.



## 4. Instalacion y configuracion de Batocera

### 4.1 Instalación de Batocera en la tarjeta microSD

El primer paso para poner en marcha el sistema fue instalar Batocera en la tarjeta microSD que actúa como almacenamiento principal de la Raspberry Pi 5. Para ello se descargó la imagen de Batocera correspondiente a la arquitectura de la Raspberry Pi 5 desde la página oficial de esta.

Para grabar la imagen en la tarjeta microSD utilice Raspberry Pi Imager, una herramienta oficial de Raspberry Pi que simplifica considerablemente el proceso. Se seleccionó la imagen descargada de Batocera, se escogió la tarjeta microSD como destino y se inició el proceso de escritura. Una vez finalizado, la tarjeta quedó lista para arrancar el sistema.

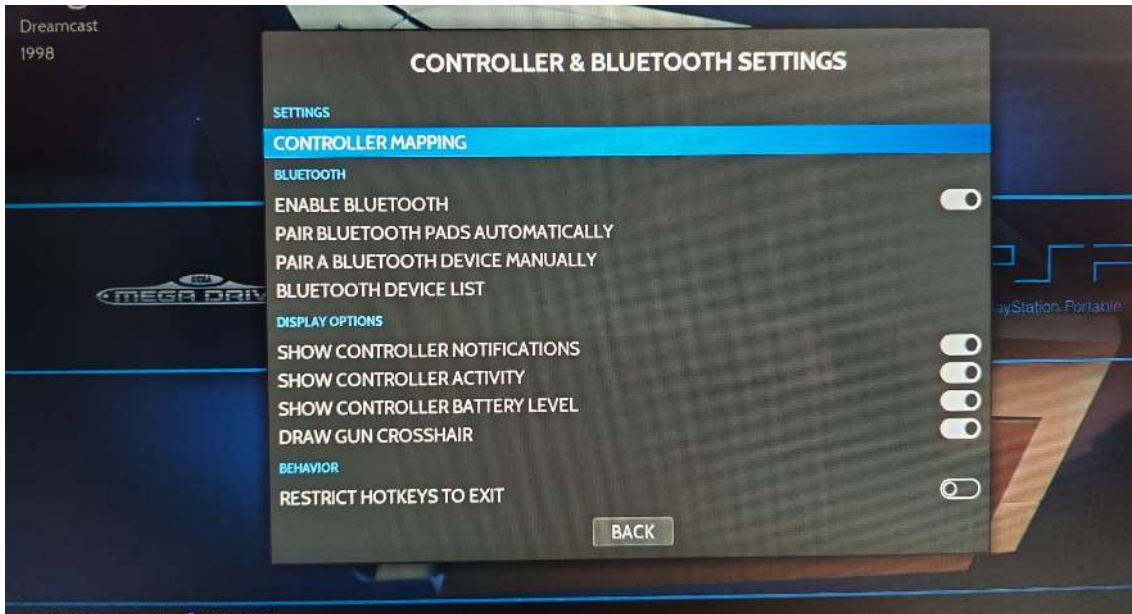


Con la tarjeta microSD puesta en la Raspberry Pi 5 y está conectada a la pantalla, se procedió al primer arranque del sistema. Batocera se inició correctamente y mostró su interfaz principal, confirmando que la instalación se había conseguido con éxito.



## 4.2 Configuración de los controles

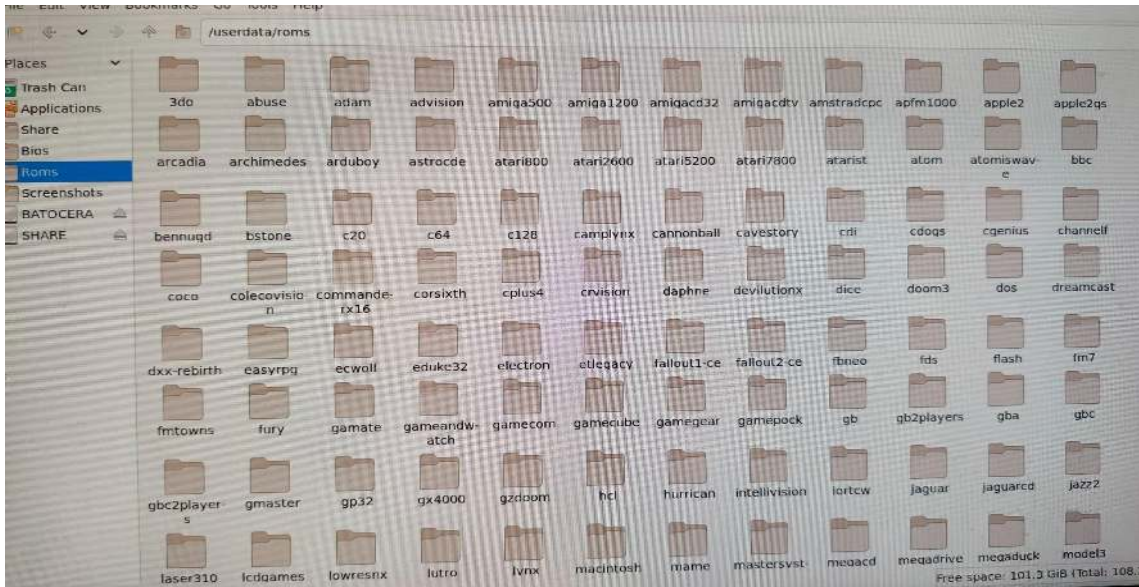
Una vez iniciado el sistema, fue necesario configurar manualmente los controles de la recreativa, ya que la placa controladora no fue reconocida de forma automática con la asignación de botones correcta. Este programam dispone de un asistente de configuración de mandos accesible desde el menú principal, que solicita al jugador pulsar cada botón y dirección del joystick uno a uno para registrar su asignación en el sistema.



Se configuraron todas las entradas disponibles: las cuatro direcciones del joystick, los 6 botones de juego, el botón de Start y el botón de Coin. Una vez completado el proceso, los controles respondieron correctamente tanto en el menú de navegación como dentro de los juegos, confirmando que la configuración había quedado registrada correctamente.

## 4.3 Configuración de los emuladores y carga de ROMs

Con los controles configurados, se procedió a cargar las ROMs de los juegos clásicos en el sistema. Batocera reserva una partición específica en la tarjeta microSD destinada al almacenamiento de ROMs, organizada en carpetas por sistema (NES, SNES, Arcade, etc.). Las ROMs se cargaron en el sistema mediante USB, primero se descargó la ROM, se pasó al USB y por último se pasó a su respectiva carpeta.



Una vez transferidas, se reinició el sistema y Batocera detectó automáticamente los juegos, mostrándolos organizados por sistema en el menú principal. Se realizaron pruebas de funcionamiento con varios títulos para verificar que los emuladores funcionaban correctamente y que los controles respondían de forma adecuada en cada sistema.

## 5. Juegos en Godot

### 5.1 Entorno de desarrollo y aprendizaje

Para el desarrollo de los juegos utilice Godot Engine 4, descargado gratuitamente desde la página oficial del proyecto. Al tratarse de una herramienta nueva para el desarrollo de este proyecto, el aprendizaje se basó en el seguimiento de tutoriales y la consulta de la documentación oficial de Godot, que está ampliamente detallada y cubre desde los conceptos más básicos hasta funcionalidades avanzadas.

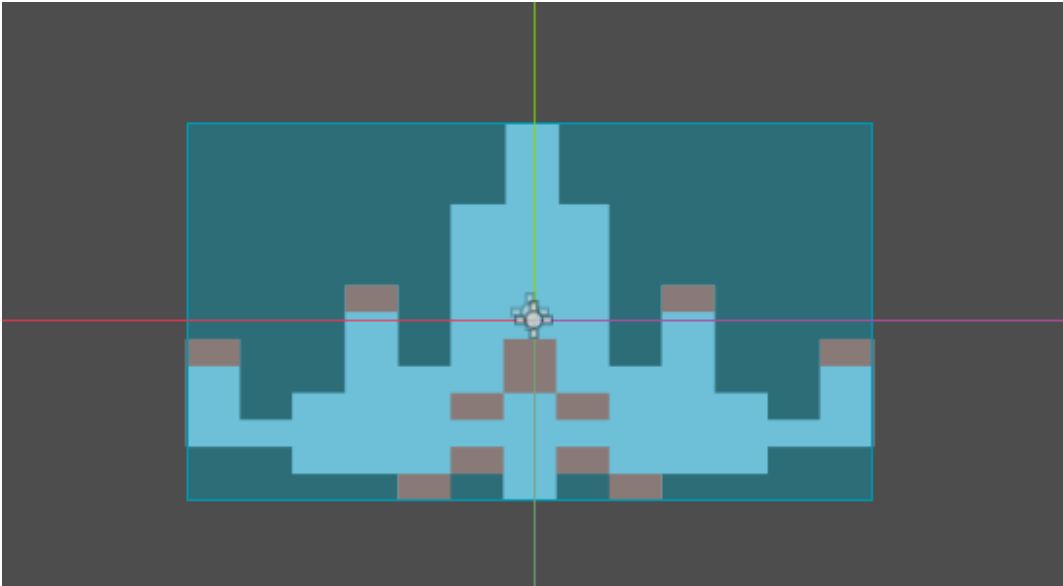
El lenguaje de programación utilizado fue GDScript, el lenguaje propio de Godot, de sintaxis similar a Python y especialmente diseñado para trabajar con el motor. Su integración nativa con el editor facilita el trabajo con nodos, escenas y señales, que son los elementos fundamentales sobre los que se construye cualquier proyecto en Godot.

### 5.2 Shooter invaders

#### 5.2.1 Descripción y diseño

El jugador controla una nave que se desplaza horizontalmente a lo largo de la parte inferior de la pantalla. El movimiento se gestiona leyendo la entrada del joystick y aplicando el desplazamiento al nodo del jugador, limitando su posición dentro de los bordes de la pantalla para que no pueda salirse del área de juego.

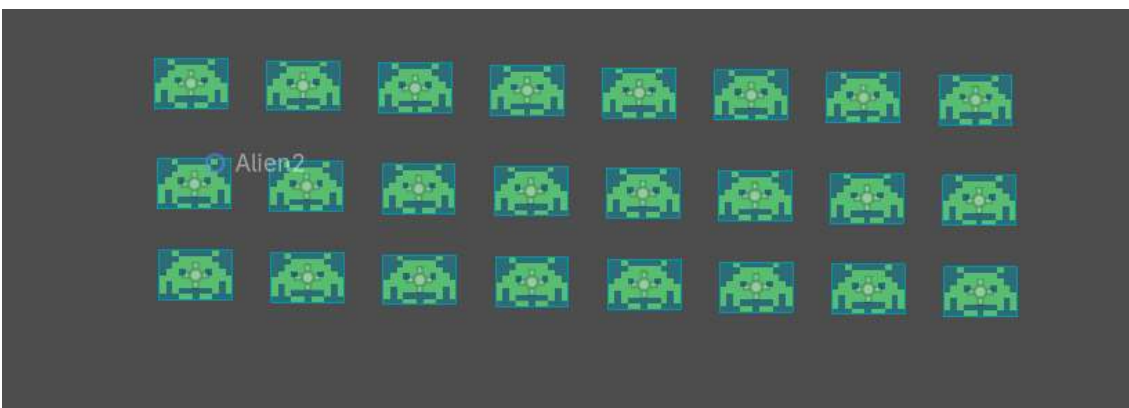
El jugador puede disparar proyectiles hacia arriba pulsando el botón de disparo. Cada pulsación genera un proyectil en la posición actual de la nave, permitiendo disparar tantas veces como el jugador pulse el botón. El jugador dispone de tres puntos de vida, de modo que necesita recibir tres impactos de los proyectiles enemigos para que la partida finalice.



### 5.2.3 Los enemigos

Los enemigos son naves del mismo tipo distribuidas en filas horizontales en la parte superior de la pantalla, replicando la formación clásica de Space Invaders. El conjunto de enemigos se desplaza progresivamente hacia abajo durante la partida y, de forma aleatoria, algunos de ellos disparan proyectiles hacia el jugador, aumentando la dificultad a medida que avanza la partida.

Cuando un proyectil del jugador impacta con un enemigo, el sistema detecta la colisión mediante las áreas de colisión de Godot y elimina ambos nodos de la escena. Si todos los enemigos son eliminados, la partida finaliza con victoria. Si algún enemigo alcanza al jugador o sus proyectiles agotan los tres puntos de vida, la partida termina en derrota. Actualmente no hay una pantalla de victoria ni de derrota implementada, por lo que la partida simplemente se detiene al cumplirse cualquiera de estas condiciones.



### 5.2.2.1 Sistema de disparo

El juego cuenta con dos tipos de proyectiles: el del jugador y el de los enemigos. El proyectil del jugador se genera en su posición actual cada vez que pulsa el botón de disparo y se desplaza hacia arriba a velocidad constante, desapareciendo al salir de la pantalla o al impactar con un enemigo. Los enemigos, por su parte, disparan sus propios proyectiles hacia abajo de forma aleatoria durante la partida. Cuando uno de estos impacta con el jugador, el sistema detecta la colisión, descuenta un punto de vida y elimina el proyectil de la escena.

## 5.3 Shooter espacial

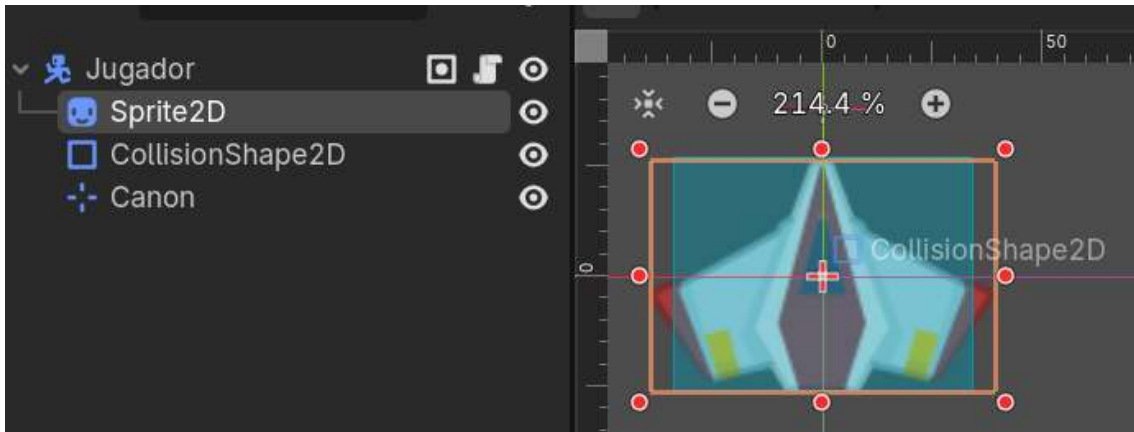
### 5.3.1 Descripción y diseño

El segundo juego amplía bastante las mecánicas del primero, introduciendo libertad de movimiento total, generación continua de enemigos y un sistema de puntuación completo. El objetivo no es eliminar una cantidad fija de enemigos sino sobrevivir el mayor tiempo posible y conseguir la puntuación más alta. La partida es infinita y solo termina cuando el jugador es eliminado.



### 5.3.2 El jugador

A diferencia del primer juego, el jugador puede desplazarse libremente en las cuatro direcciones por toda la pantalla. El sistema lee las entradas horizontal y vertical del joystick de forma simultánea y aplica el desplazamiento correspondiente, comprobando en todo momento que el jugador no sobrepase los bordes de la pantalla. En este juego el jugador muere de un solo impacto con un enemigo.



### 5.3.3 Sistema de disparos

En este juego solo dispara el jugador. Al pulsar el botón de disparo se genera un proyectil en la posición actual de la nave que se desplaza hacia arriba a velocidad constante, desapareciendo al salir de la pantalla o al impactar con un enemigo. A diferencia del juego anterior no hay proyectiles enemigos, siendo el contacto directo entre el jugador y un enemigo la única forma de recibir daño.

### 5.3.4 Los enemigos

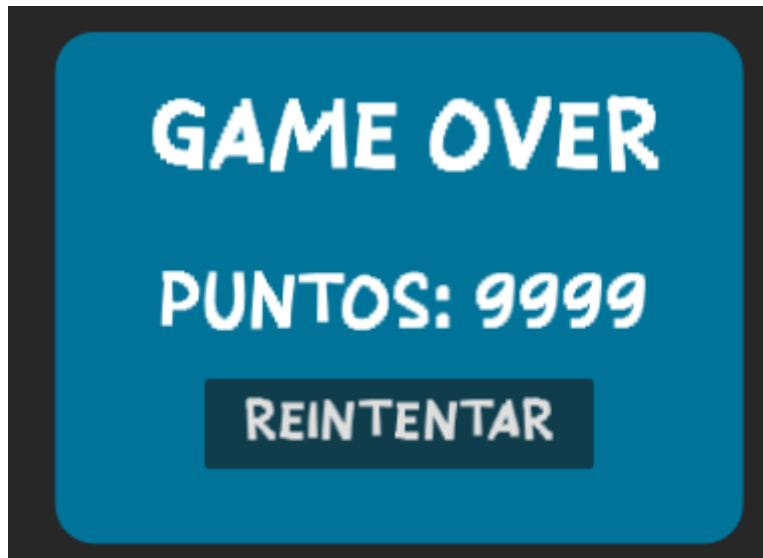
Los enemigos se generan de forma continua mediante un temporizador que crea nuevos nodos en posiciones aleatorias de la parte superior de la pantalla. Simplemente descienden hacia abajo a velocidad constante sin disparar, con la diferencia de que hay dos tipos de enemigos uno más lento y otro más rápido. Cuando un enemigo sale de la pantalla por la parte inferior sin ser eliminado desaparece de la escena y el sistema continúa generando nuevos enemigos sin interrupción. Si un enemigo impacta directamente con el jugador la partida termina.



### 5.3.5 Sistema de puntuación y game over

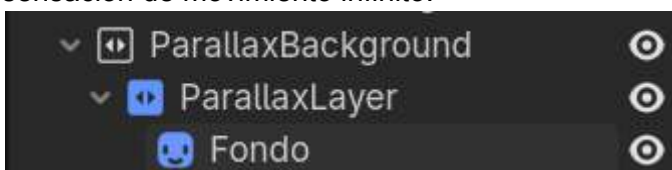
Este segundo juego incluye un sistema de puntuación completo que actúa como elemento central de la experiencia. Cada vez que el jugador elimina un enemigo, el sistema incrementa la puntuación mediante una variable global y actualiza el marcador visible en pantalla en tiempo real.

Cuando el jugador muere, el juego muestra una pantalla de game over dedicada con la puntuación final obtenida durante la partida, añadiendo un cierre claro a cada sesión de juego y reforzando el componente de rejugaridad del título.



### 5.3.6 Fondo del juego

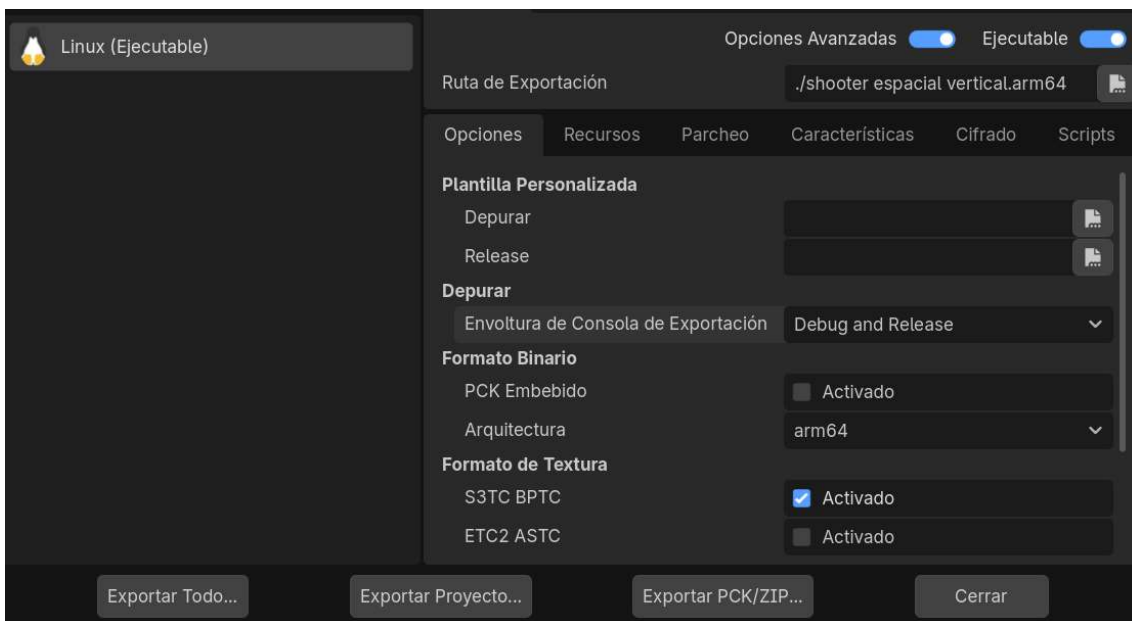
Para reforzar la ambientación espacial del juego se implementó un fondo de espacio estrellado en movimiento. Se trata de una imagen de un espacio con estrellas que se desplaza continuamente hacia abajo, repitiéndose cuando llega al final para crear la sensación de movimiento infinito.



## 6. Integración de los juegos en Batocera

### 6.1 Exportación desde Godot Engine

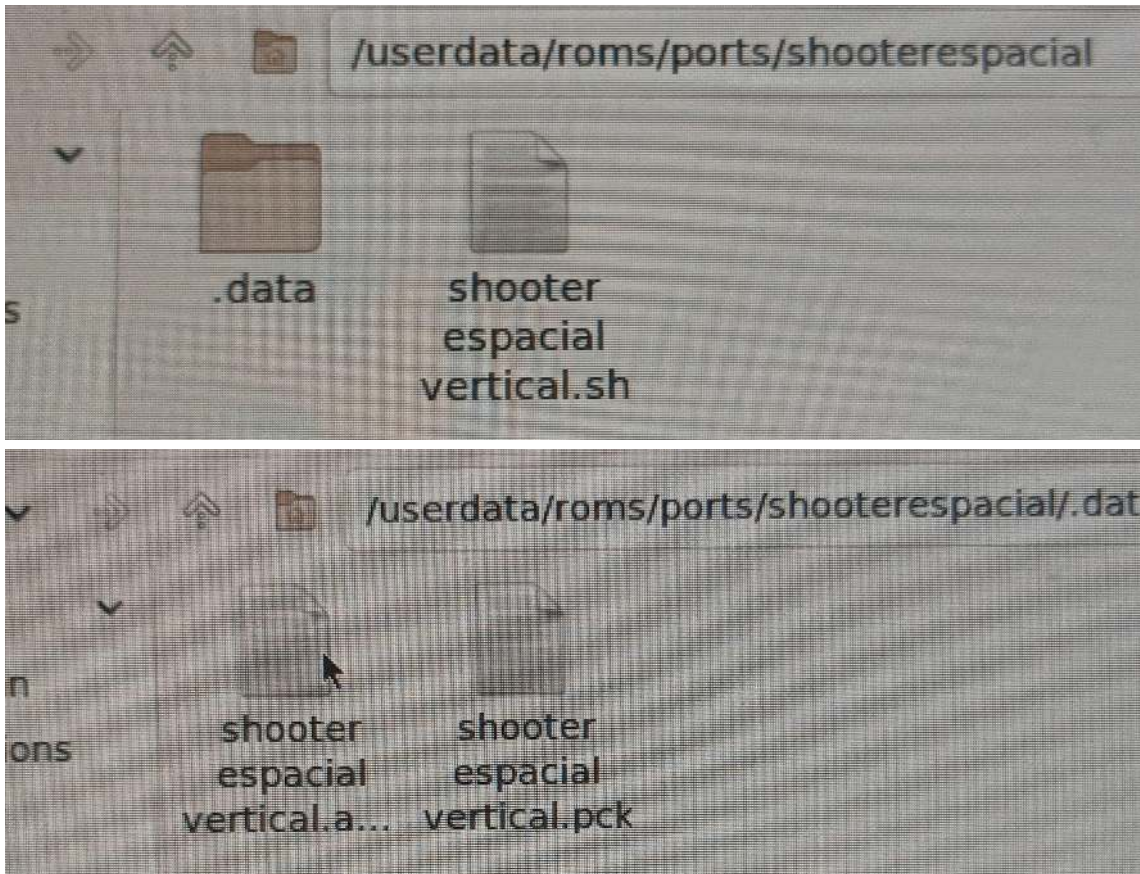
Para poder ejecutar los juegos desarrollados en Godot Engine directamente desde la Raspberry Pi 5, fue necesario exportarlos como ejecutables compatibles con su arquitectura. Godot permite exportar proyectos a múltiples plataformas, pero para ello requiere descargar previamente las plantillas de exportación oficiales correspondientes a cada sistema. Estas plantillas se descargaron desde el propio editor de Godot a través del gestor de plantillas de exportación, seleccionando la versión correspondiente a Linux ARM64, que es la arquitectura del procesador de la Raspberry Pi 5.



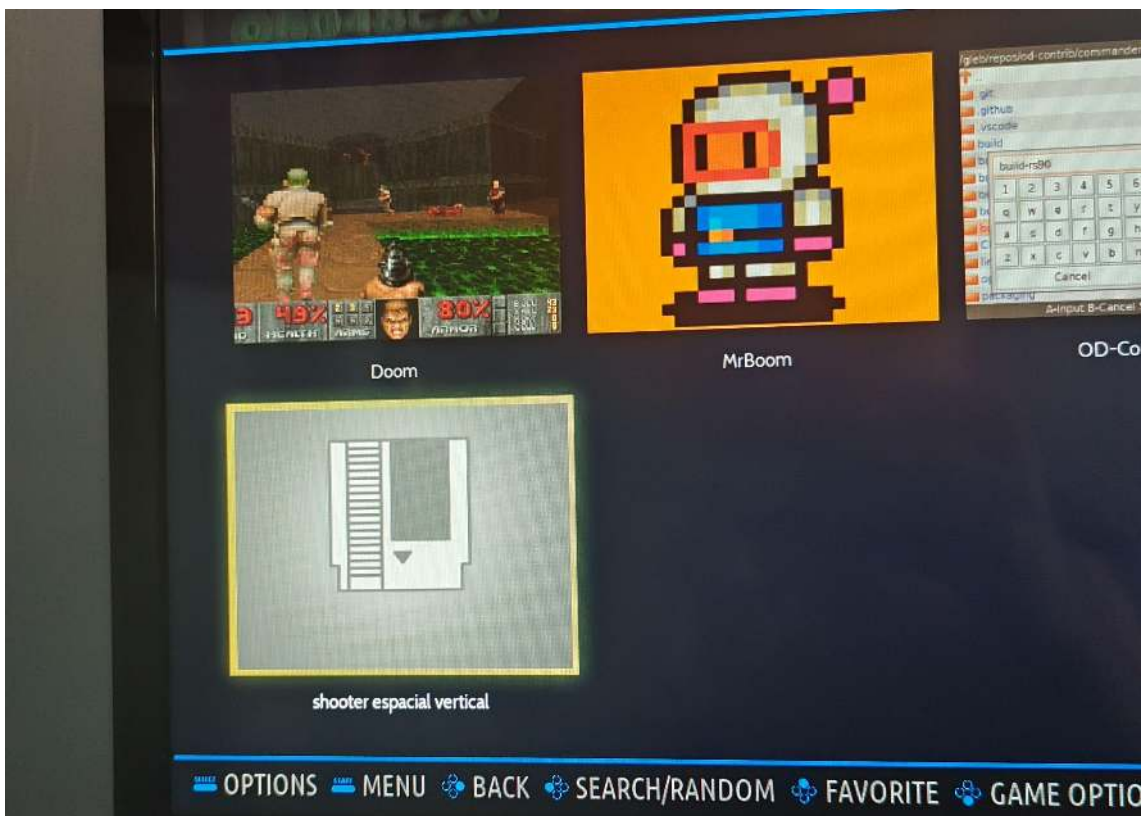
Una vez instaladas las plantillas, se procedió a exportar cada juego seleccionando Linux ARM64 como plataforma de destino. El proceso genera un archivo ejecutable que contiene todo lo necesario para que el juego funcione de forma autónoma en la Raspberry Pi sin necesidad de tener Godot instalado en ella.

### 6.2 Transferencia a la Raspberry Pi

Con los ejecutables generados, se empezó a pasarlos a la Raspberry Pi. Para ello se copiaron los archivos en un USB desde el portátil del desarrollo y después se conectó el USB a la Raspberry Pi para trasladar los ejecutables a su carpeta correspondiente dentro del sistema de archivos de Batocera.



Para acabar con la parte de los juegos se comprobó que el juego aparecía en la sección de "Ports" con su título.



## 7. Conclusions

### 7.1 Conclusions generals del projecte

En este proyecto he podido aprender varias cosas, entre ellas el funcionamiento interno de una máquina recreativa, desde el diseño y construcción física del estructura hasta la integración de todos los componentes electrónicos. También he conseguido conocimientos sobre la configuración de sistemas basados en Linux, concretamente Batocera, y he aprendido a desarrollar videojuegos desde cero utilizando Godot Engine 4 y GDScript. Además, el proyecto me ha permitido enfrentarme a problemas reales durante el desarrollo, como la incompatibilidad de la pantalla original con la Raspberry Pi 5, aprendiendo a buscar soluciones y tomar decisiones técnicas justificadas.

### 7.2 Consecució dels objectius

En general se han llegado a cumplir todos los objetivos iniciales excepto uno, originalmente quería hacer 3 juegos que fueran escalando en complejidad, finalmente acabé haciendo el primero que era el más sencillo de todos con muy poca complejidad y el segundo bastante más complejo con muchas más funciones. Todo esto fue debido a decisiones que tomé a medida que iba pasando el tiempo, porque el tercer juego iba a tener un tono muy diferente a lo que fueron los dos juegos anteriores, así que me decante a hacer más complejo el segundo juego.

### 7.3 Valoració de la metodologia i planificació

Este proyecto no se pudo seguir paso a paso cómo se planificó originalmente pero se pudo desarrollar de una manera completa, al final resultó que entre semana iba investigando y desarrollando los juegos y los fines de semana iba al taller a construir la recreativa, documentando en tiempos libres.

### 7.4 Visió de futur

Para empezar por la parte física, ya a futuro será dejar más limpio toda la recreativa, cubriendo las partes sobresalientes de la pantalla con más madera, además de poner alguna calcomanía a modo de decoración, y algún letrero con luces, siendo meramente estético todos estos cambios, por lo general estoy muy contento con el acabado de la recreativa.

Por la parte del desarrollo de los juegos, me gustaría hacer un tercer juego de otro tipo diferente (un juego de plataformas por ejemplo) para poder seguir profundizando en el desarrollo de juegos, ya que es un mundo bastante complejo pero cuando lo entiendes es muy interesante.

## 8. Glossari

**ARM64:** Arquitectura de procesador de 64 bits utilizada en dispositivos de bajo consumo como la Raspberry Pi. Determina el tipo de ejecutable compatible con el sistema.

**Batocera:** Sistema operativo basado en Linux especializado en la emulación de videojuegos clásicos. Gestiona los emuladores, los controles y la interfaz de usuario de RetroPixel.

**Emulación:** Proceso mediante el cual un sistema informático imita el comportamiento de otro sistema, permitiendo ejecutar software diseñado para hardware diferente.

**Escena:** En Godot Engine, unidad básica de organización del juego. Cada pantalla, personaje o elemento del juego se estructura como una escena formada por nodos.

**GDScript:** Lenguaje de programación propio de Godot Engine, de sintaxis similar a Python, diseñado específicamente para el desarrollo de videojuegos dentro del motor.

**Instanciar:** En Godot Engine, acción de crear una copia activa de una escena o nodo dentro del juego en tiempo de ejecución, como generar un proyectil al disparar.

**Nodo:** En Godot Engine, elemento básico de construcción de cualquier escena. Cada objeto del juego está representado por uno o varios nodos.

**Placa controladora:** Componente electrónico que actúa como intermediario entre los botones y el joystick de la recreativa y la Raspberry Pi, traduciendo las pulsaciones físicas en señales reconocibles por el sistema.

**ROM:** Archivo digital que contiene una copia de un videojuego clásico, utilizado por los emuladores para ejecutar el juego en hardware moderno.

**Scraping:** Proceso automático mediante el cual Batocera descarga carátulas, descripciones y metadatos de los juegos desde internet para mostrarlos en el menú.

**Señales:** En Godot Engine, sistema de comunicación entre nodos que permite que un nodo notifique a otro cuando ocurre un evento, como una colisión o la pulsación de un botón.

**Variable global:** Variable accesible desde cualquier parte del código del proyecto, utilizada para gestionar la puntuación del jugador de forma centralizada.

## 9. Webgrafia

Webs para desarrollar RetroPixel:

Web del software de Raspberry Pi: <https://www.raspberrypi.com/software/>

Web de Batocera: <https://batocera.org/>

Sitio web donde descargue las ROMS: <https://romsfun.com/roms/>

Enlace de compra del adaptador HDMI a VGA: <https://amzn.eu/d/0aqsju58>

Enlace de compra de los botones, joystick y placa de control:  
[https://www.amazon.es/dp/B08R9Z53RC?ref\\_=cm\\_sw\\_r\\_cp\\_ud\\_dp\\_AEZWGT65K9D7RC7HFGDX](https://www.amazon.es/dp/B08R9Z53RC?ref_=cm_sw_r_cp_ud_dp_AEZWGT65K9D7RC7HFGDX)

Webs para el desarrollo de los juegos:

Enlace a la documentación oficial de godot:  
<https://docs.godotengine.org/en/stable/>

Enlace a la web oficial de Godot: <https://godotengine.org/es/>

Sitio web donde descargue la fuente de letra: [Enlace](#)

Sitio web de Kenney donde se descargó la nave aliada:  
<https://kenney.nl/assets/space-shooter-remastered>

Sitio web de Kenney donde se descargó las naves alienígenas:  
<https://kenney.nl/assets/alien-ufo-pack>

Sitio web para el fondo del juego 2: [Enlace](#)

Sitio web de Kenney donde descargue los archivos de audio:  
<https://kenney.nl/assets/sci-fi-sounds>

## 10. Anexos

[Carpeta de imágenes de la construcción de RetroPixel](#)