



Treball de Final de Grau

GRAU EN ENGINYERIA INFORMÀTICA

Facultat de Matemàtiques i Informàtica

Universitat de Barcelona

**Comparativa i avaluació de sistemes
d'emulació en microconsoles amb
un Single-Board Computer**

Autor: Pere Ferré Medina

Director: Manel López De Miguel

Realitzat a: Departament de Matemàtiques i Informàtica

Barcelona, 15 de gener de 2025

Abstract

This Final Degree Project explores the feasibility of using a Single-Board Computer (SBC), specifically the Raspberry Pi 5, to create a microconsole capable of emulating video games from various classic and modern consoles.

Firstly, it examines the basic concepts of SBCs, microconsoles, and video game emulation, with an emphasis on their evolution and the technical challenges they involve. Next, it describes the setup and testing process of different platforms (RetroArch, Recalbox, Lakka), as well as the manual installation of multiple emulation cores to reproduce systems like Game Boy Advance, Nintendo DS, PSP, or GameCube. It also discusses optimization strategies and bottleneck diagnostics—both in hardware and software—including cooling techniques, shader configuration, overclocking, latency adjustments, and the selection of properly optimized ROMs.

The results show that, despite the inherent limitations of ARM architecture in more powerful 3D consoles, modern SBCs can deliver a satisfactory emulation experience for retro and 2D systems, and even some older 3D consoles. The importance of using lightweight operating systems and making specific tweaks for each emulator is highlighted. The conclusions suggest that, with a relatively low cost, it is possible to achieve a functional and stable microconsole, opening up new avenues for further enhancements in performance, compatibility, and ease of use.

Resumen

Este Trabajo de Fin de Grado estudia la viabilidad de utilizar un Single-Board Computer (SBC), concretamente la Raspberry Pi 5, para crear una microconsola capaz de emular videojuegos de diversas consolas clásicas y modernas.

En primer lugar, se analizan los conceptos básicos sobre los SBC, las microconsolas y la emulación de videojuegos, haciendo hincapié en su evolución y en los desafíos técnicos que plantea. A continuación, se describe el proceso de configuración y prueba de diferentes plataformas (RetroArch, Recalbox, Lakka), así como la instalación manual de varios núcleos de emulación para reproducir sistemas como Game Boy Advance, Nintendo DS, PSP o GameCube. También se exponen estrategias de optimización y diagnóstico de cuellos de botella —tanto a nivel de hardware como de software—, incluyendo técnicas de refrigeración, configuración de *shaders*, *overclocking*, ajustes de latencia y selección de ROMs adecuadamente optimizadas.

Los resultados indican que, a pesar de las limitaciones inherentes a la arquitectura ARM en consolas 3D más potentes, los SBC modernos pueden ofrecer una experiencia de emulación suficientemente satisfactoria para sistemas retro y 2D, así como para algunas consolas 3D de generaciones anteriores. Se destaca la importancia de emplear sistemas operativos ligeros y de realizar ajustes específicos para cada emulador. Las conclusiones apuntan que, con un coste relativamente bajo, es posible conseguir una microconsola funcional y estable, abriéndose diversas vías de futuro para seguir mejorando sus prestaciones, la compatibilidad y la facilidad de uso.

Resum

Aquest Treball de Final de Grau estudia la viabilitat d'utilitzar un Single-Board Computer (SBC), concretament la Raspberry Pi 5, per crear una microconsola capaç d'emular videojocs de diverses consoles clàssiques i modernes.

En primer lloc, s'hi analitzen els conceptes bàsics sobre els SBC, les microconsoles i l'emulació de videojocs, posant èmfasi en la seva evolució i en els reptes tècnics que suposa. A continuació, es descriu el procés de configuració i prova de diferents plataformes (RetroArch, Recalbox, Lakka), així com la instal·lació manual de diversos nuclis d'emulació per tal de reproduir sistemes com Game Boy Advance, Nintendo DS, PSP o GameCube. Es mostren també estratègies d'optimització i diagnosi de colls d'ampolla —tant en l'àmbit de maquinari com de programari—, incloent-hi tècniques de refrigeració, configuració de *shaders*, *overclocking*, ajustos de latència i selecció de ROMs adequadament optimitzades.

Els resultats indiquen que, malgrat les limitacions inherents de l'arquitectura ARM per a consoles 3D més potents, els SBC moderns poden arribar a oferir una experiència d'emulació prou satisfactòria per a sistemes retro i 2D, i algunes consoles 3D d'antiga generació. Destaca la importància d'emprar sistemes operatius lleugers i de realitzar ajustos específics per a cada emulador. Les conclusions apunten que, amb un cost relativament baix, es pot aconseguir una microconsola funcional i estable, i s'obren diverses línies de futur per millorar-ne encara més les prestacions, la compatibilitat i la facilitat d'ús.

Agraïments

Voldria expressar el meu agraïment a totes les persones que m'han ajudat al llarg d'aquest projecte, tant de manera directa com indirecta.

En primer lloc, a la meva parella, que m'ha encoratjat i m'ha donat suport en els moments de dubte, acompanyant-me en cada fase del procés i mostrant un interès incondicional.

En segon lloc, al meu tutor, el professor Manel López, per acceptar la meva proposta de treball, guiar-me en els moments de més incertesa i oferir-me consells valuosos per millorar tant el contingut tècnic com l'enfocament global del TFG.

Finalment, m'agradaria agrair a totes aquelles persones —amics, familiars i companys— que, amb els seus coneixements, idees o suport moral, han fet possible que aquest treball arribi a bon port.

Índex general

Introducció	10
1.1. Context i motivació	10
1.1.1. Context	10
1.1.2. Motivació	11
1.2. Objectius	11
1.2.1. Objectiu principal	11
1.2.2. Objectius secundaris	11
1.3. Planificació i costos	12
1.3.1. Planificació	12
1.3.2. Costos	13
1.4. Estructura de la memòria	14
 Estat de la qüestió	 16
2.1. Introducció als Single Board Computers	16
2.1.1. Exemples i usos	16
2.1.2. Comparativa entre SBC	19
2.2. Microconsoles i dispositius de streaming	23
2.2.1. Definicions i funcions	23
2.2.2. Exemples, usos i comparació	24
2.3. Emulació de videojocs	27
2.3.1. Definició i objectius de l'emulació	28
2.3.2. Principals emuladors i compatibilitats	28
2.3.3. Aspectes ètics de l'emulació	31
2.4. Relació entre els conceptes: SBC, microconsoles i emuladors	34
 Arquitectura de capes i proves inicials al SBC	 36
3.1. Capes en l'emulació: anàlisi i funcionament	36
3.1.1. Capes principals	37
3.2. Primeres proves amb diferents plataformes d'emulació	41
3.2.1. Selecció del SBC adequat	41
Comparació amb altres alternatives	41
3.2.2. Definicions de les plataformes i jocs a testear	42
3.2.3. Instal·lacions	44
3.2.4. Seguiment de la prova	45
Resultats	47
 Optimització del sistema	 49
4.1. Identificació dels colls d'ampolla	49

4.1.1. Tipus de colls d'ampolla	50
4.1.2. Eines i tècniques	50
4.2. Estratègies i proves per maximitzar el rendiment	52
4.2.1. Estratègies d'optimització	52
4.2.2. Proves per implementar estratègies d'optimització	53
Prova de refrigeració activa	53
Optimització de configuració de Lakka	55
Aplicació de shaders i configuració Run Ahead	61
Modes de recompilació d'un emulador	65
ROM Optimitzades vs. ROM No Optimitzades	69
Overclocking de la CPU i GPU	71
4.2.3. Anàlisi de les millores aplicades	73
Conclusions	76
5.1. Revisió d'objectius i correlació del treball fet	76
5.2. Problemes que han anat sorgint	77
5.3. Línies de futur	79
5.4. Reflexions finals	80
Referències	82
Glossari i acrònims	86
Apèndix A: Manual d'instal·lacions i configuracions	89
A.1. Instal·lació de Raspberry Pi OS	89
A.2. Instal·lació de Lakka i RetroArch	91
Lakka com a sistema operatiu independent	91
Recalbox com a sistema operatiu independent	91
RetroArch sobre Raspberry Pi OS	92
A.3. Manual d'usuari bàsic	92
Apèndix B: Scripts, configuracions i eines de suport	94
B.1. Eines emprades	94
B.2. Script d'instal·lació de Dolphin a RPi OS	97
B.3. Configuració de config.txt per fer overclocking	98

Índex de figures i taules

Índex de figures

Figura 1.1. Diagrama de Gantt que mostra la planificació del projecte i les franges de temps dedicades.	13
Figura 2.1. Esquema bàsic d'un Single-Board Computer i dels seus components principals.	19
Figura 2.2. Exemples de disseny físic de SBC: Raspberry Pi 5 i Odroid-H4.	23
Figura 2.3. Comparació visual d'una microconsola i un dispositiu de streaming.	27
Figura 2.4. Interfície gràfica principal de RetroArch mostrant diferents menús i opcions disponibles.	31
Figura 4.1. Diferents mesures de temperatura de la CPU de la Raspberry Pi 5 mentre s'executava Lakka amb el joc God of War.	54
Figura 4.2. Fent la comanda <code>systemctl list-unit-files --type=service</code> , veiem que després d'arrancar hi ha 106 fitxers amb processos llistats.	55
Figura 4.3. Fitxer creat dins la carpeta <code>/storage/.config/sysctl.d</code> , el qual s'executarà en l'arrencada.	57
Figura 4.4. Temps que triga Lakka en l'arrencada sense cap optimització aplicada i amb l'optimització aplicada.	58
Figura 4.5. Processos que triguen més en l'arrencada.	58
Figura 4.6. Interfície de top un cop aplicades les millores de configuració.	60
Figura 4.7. Imatge del joc sense shaders.	62
Figura 4.8. Imatge del joc amb el shader <code>crt-pi.glsfp</code> activat.	63
Figura 4.9. Imatge del joc amb el shader <code>hq2x</code> activat.	63
Figura 4.10. Configuració del Run Ahead aplicada.	65
Figura 4.11. Interfície de PuTTY mentre Lakka executa l'emulador Dolphin amb JITARM.	66
Figura 4.12. Interfície de PuTTY mentre Lakka executa Dolphin amb el mode Interpreter.	67

Figura 4.13. Interfície de PuTTY mentre Lakka executa l'emulador Dolphin amb recompilació mitjançant Cache Interpreter.	67
Figura 4.14. Interfície del PuTTY en executar la ROM més optimitzada.	70
Figura 4.15. Sortida de la comanda top sense overclocking actiu.	70
Figura 4.16. Sortida de la comanda top després d'haver activat l'overclocking.	72
Figura 4.17. Gràfic de dades reals comparant Metroid: Zero Mission (GBA) i The Legend of Zelda: The Wind Waker (GameCube).	72
Figura 4.18. Gràfic de barres representatiu dels canvis abans i després de les millores.	74
Figura A.1. Interfície gràfica de l'eina Raspberry Pi Imager, utilitzada per escriure la imatge del sistema en la targeta SD.	90
Figura B.1. Pantalla inicial de configuració de la connexió SSH a PuTTY per a Windows.	95
Figura B.2. Sessió de PuTTY un cop connectada a la Raspberry Pi, mostrant la línia d'ordres.	96
Figura B.3. Interfície de FileZilla, explorant fitxers locals i remots via SFTP per afegir ROMs i BIOS.	97

Índex de taules

Taula 2.1. Comparativa de característiques d'alguns SBC.	20
Taula 2.2. Comparativa de característiques d'alguns dispositius famosos.	25
Taula 2.3. Característiques d'alguns emuladors molt coneguts.	29
Taula 2.4. Comparativa de característiques d'algunes plataformes d'emulació.	30
Taula 3.1. Rendiment dels diferents jocs en cada plataforma	46
Taula 3.2. Comparació global de plataformes d'emulació en Raspberry Pi 5.	47
Taula 4.1. Millores en alguns aspectes del funcionament del sistema operatiu.	59
Taula 4.2. Comparativa dels modes de recompilació (JITARM64, Interpreter, Cached Interpreter) a Dolphin.	68
Taula 4.3. Comparació entre una ROM original poc optimitzada i una ROM més optimitzada (Super Mario 64).	69

Capítol 1.

Introducció

En aquest capítol es presenta el plantejament general del projecte de Treball de Final de Grau, el qual gira entorn de la comparativa i avaluació de sistemes d'emulació en microconsoles basades en un **Single-Board Computer** (SBC). L'objectiu principal és conèixer quins factors tècnics, de configuració i de planificació incideixen més en l'èxit d'una microconsola emuladora de videojocs, i com optimitzar-ne el rendiment amb un cost ajustat.

1.1. Context i motivació

1.1.1. Context

La pràctica de l'emulació de videojocs ha crescut exponencialment els darrers anys, impulsada per l'interès de preservar títols clàssics i de revitalitzar experiències retro en *hardware* (o maquinari) modern. Els Single-Board Computers s'han convertit en solucions atractives per materialitzar aquesta emulació gràcies al seu baix cost, petita mida i comunitat activa de desenvolupadors.

Les anomenades microconsoles aprofiten el potencial dels SBC per crear dispositius compactes, econòmics i òptims per a emuladors retro. La seva popularitat rau en la possibilitat de tenir, dins un únic aparell, un catàleg divers de videojocs de diferents èpoques i consoles, sense necessitat de disposar dels sistemes originals ni de llançar múltiples dispositius.

1.1.2. Motivació

L'emulació planteja un repte tècnic: aconseguir la màxima fluïdesa (FPS estables), qualitat d'imatge i so, i compatibilitat de jocs, tot limitant el consum de recursos. A més, apareixen factors com la temperatura (possible *thermal throttling*), l'arquitectura ARM, que pot dificultar l'execució d'alguns emuladors nascuts per a x86/x64, la necessitat de BIOS i ROM adequades, o la tria d'una interfície d'usuari intuïtiva que faciliti la selecció de jocs i la configuració.

Amb aquest projecte, es pretén, per una banda, donar una visió general de les diferents opcions que hi ha al mercat de dispositius i plataformes per a l'ús d'emuladors retro. Com a segon objectiu, es pretén maximitzar el rendiment de l'emulació en un SBC concret (en el cas pràctic, la Raspberry Pi 5) tot comparant diferents plataformes (Lakka, RetroArch, Recalbox) i ajustant elements com *shaders*, nuclis d'emulació, configuracions de *Run Ahead*, *overclocking* o sistemes de refrigeració. L'objectiu final és poder oferir unes recomanacions sòlides que permetin aprofitar tot el potencial d'un SBC i gaudir d'una experiència de joc òptima.

(vegeu el [glossari](#) per aprofundir en els termes)

1.2. Objectius

1.2.1. Objectiu principal

Comparar i avaluar el rendiment d'emuladors de videojocs en una microconsola basada en un **Single-Board Computer**, analitzant diferents configuracions i establint criteris per aconseguir una **experiència de joc òptima**.

1.2.2. Objectius secundaris

- **Proporcionar una visió general** de l'estat dels diferents emuladors que hi ha actualment a l'abast dels usuaris, proporcionant d'aquesta manera les bases a persones novelles per entrar en aquest món.
- **Identificar i seleccionar emuladors** compatibles amb el SBC per a diverses consoles de videojocs (Nintendo DS, PlayStation 2, GameCube/Wii, etc.), comprovant-ne l'estabilitat i la facilitat d'instal·lació.
- **Definir criteris d'avaluació** i mètriques de rendiment, com:
 - Velocitat d'emulació (FPS).
 - Temps de càrrega.
 - Compatibilitat de jocs.
 - Qualitat gràfica i d'àudio.
 - Eficiència energètica (recursos del SBC).
- **Dur a terme proves de rendiment** amb una **metodologia estandarditzada**, comprovant el funcionament de diversos títols (2D, 3D) i descripcions detallades de configuració.
- **Analitzar els resultats** per detectar els punts forts i febles de cada emulador en el SBC, comparant plataformes de forma sistemàtica.
- **Proposar recomanacions** concretes d'ús per a microconsoles: configuracions òptimes, paràmetres de nuclis, *shaders*, funcionament a llarg termini, refrigeració, etc.

1.3. Planificació i costs

1.3.1. Planificació

La planificació ha inclòs una sèrie de fites i tasques distribuïdes al llarg de diversos mesos durant la segona meitat de 2024 i inicis de 2025:

- **Primera meitat de juny:** Definició d'objectius i abast del TFG.
- **Finals de juny i inicis de juliol:** Planificació de la recerca bibliogràfica i estat de l'art.

- **Juliol i inici d'agost:** Recerca d'informació i selecció dels emuladors a analitzar. Revisió de plataformes (Lakka, RetroArch, etc.) i de versions compatibles.
- **Agost i setembre:** Aconseguir el material necessari (SBC, ventiladors, targetes SD, comandaments) i iniciar primeres proves de configuració.
- **Setembre al gener:** Desenvolupament de la placa i test d'emuladors, afinant configuracions, fent comparatives i recollint dades de rendiment (FPS, latència, temperatura).
- **Novembre al gener:** Redacció de la memòria i documentació de resultats. Ajust final de configuracions i conclusions.



Figura 1.1. Diagrama de Gantt que il·lustra els processos de desenvolupament d'aquest projecte i les franges de temps dedicades.

1.3.2. Costs

La planificació general d'aquest projecte ha tingut uns **4-5 mesos** d'abast, durant els quals s'ha treballat aproximadament **5 hores diàries** dedicades a tasques de recerca, configuració tècnica i proves. Atès que valorem aquest temps a uns **10 € l'hora**, el cost laboral total se situa entre:

- $4 \text{ mesos} \times 20 \text{ dies laborables/mes} = 80 \text{ dies} \times 5 \text{ hores/dia} = 400 \text{ hores} \times 10 \text{ €/hora} = 4.000 \text{ €}$
- $5 \text{ mesos} \times (20 \text{ dies laborables/mes}) = 100 \text{ dies} \times 5 \text{ hores/dia} = 500 \text{ hores} \times 10 \text{ €/hora} = 5.000 \text{ €}$

Per tant, la franja orientativa de costos laborals se situa entre 4.000 i 5.000 euros, en funció del temps exacte invertit.

A aquest import cal afegir-hi la despesa en materials, que en el meu cas inclou:

- Raspberry Pi 5 (8 GB): ~ 90 €
- Font d'alimentació: ~ 12 €
- Cable micro-HDMI a HDMI: ~ 6 €
- Carcassa amb ventilador: ~ 12 €
- Comandament: ~ 30 €
- Targetes SD / USB (suport emmagatzematge): ~ 10 € cadascuna
- Capturadora HDMI (opcional): ~ 8 €
- PC auxiliar (si es compta com a part del projecte): ~ 500 €

Sense comptar un PC extra, la despesa de maquinari pot rondar els **170-200 €** sumant tots els components. Així, el cost global del projecte (treball + materials) pot arribar a oscil·lar entre els **4.200 €** i els **5.200 €**, depenent del temps invertit i de si s'inclouen o no determinats components de maquinari.

El projecte suposa una inversió principalment orientada al temps de configuració i proves, a més d'un cost raonable en materials que permeten convertir el SBC en una microconsola efectiva. Tot i ser una xifra significativa, el resultat final ofereix una plataforma d'emulació potent, flexible i econòmica en comparació amb altres opcions comercials.

1.4. Estructura de la memòria

La memòria s'estructura en diversos capítols:

- **Capítol 1. Introducció:** El capítol actual, en el qual es contextualitza el TFG, s'explica la motivació del projecte, els objectius concrets, la planificació i els costos aproximats, així com l'organització general de la memòria.

- **Capítol 2. Estat de la qüestió:** S'hi inclou el marc conceptual i tecnològic sobre SBC, microconsoles i emulació de videojocs, aportant exemples, comparatives i referències.
- **Capítol 3. Arquitectura de capes i proves inicials al SBC:** Es donen a conèixer les diferents capes que actuen en el procés d'emulació (*hardware*, OS, nuclis, ROM...) i es descriuen els primers passos en la instal·lació i configuració del sistema en la SBC escollida, els resultats preliminars i els criteris per triar la plataforma d'emulació principal.
- **Capítol 4. Optimització del sistema:** S'hi mostra com identificar els colls d'ampolla i quines estratègies d'optimització i millores s'han aplicat a la plataforma anteriorment escollida.
- **Capítol 5. Conclusions:** Es revisen els objectius assolits, els problemes sorgits i les possibles línies de futur per a la millora i l'ampliació del projecte.

Amb aquesta estructura es busca donar coherència al projecte, passant d'una visió general i teòrica a la part pràctica de proves i comparatives, fins a arribar a les conclusions i propostes de treball futur.

Capítol 2.

Estat de la qüestió

En aquest capítol s'analitza el marc conceptual i tecnològic que sustenta el projecte. Es divideix en tres blocs principals: una introducció als Single Board Computers (SBC), una exploració de les microconsoles i dispositius de streaming, i una discussió sobre l'emulació de videojocs. Aquest enfocament permet establir les bases teòriques i pràctiques que guien el desenvolupament del projecte, proporcionant un context clar per a les decisions tècniques.

2.1. Introducció als Single Board Computers

Un Single Board Computer (SBC) és un ordinador complet construït sobre una sola placa de circuit imprès que inclou tots els components essencials, com el processador, la memòria RAM i els ports d'entrada/sortida. Els SBC s'utilitzen habitualment en aplicacions educatives, projectes personals, robòtica, IoT i com a plataformes per a sistemes integrats ([Single-board computer, n.d.](#)). A diferència dels ordinadors convencionals, els SBC tenen tots els components integrats, fet que redueix el cost i la mida, però també limita la seva flexibilitat en termes d'expansió.

Els SBC no només representen una solució compacta i econòmica, sinó que també constitueixen la base tecnològica per construir microconsoles i sistemes d'emulació eficients. En integrar processadors ARM i una arquitectura optimitzada, són capaços de suportar diferents tipus d'emuladors, sent això clau per aquest projecte.

2.1.1. Exemples i usos

Alguns dels SBC més destacats inclouen marques reconegudes a escala mundial pel seu compromís amb la innovació i la qualitat. Aquestes plataformes han guanyat

gran popularitat per la seva versatilitat i el suport de comunitats actives que faciliten el desenvolupament de projectes tant educatius com industrials:

- **Raspberry Pi:**

És la placa més popular, desenvolupada per Raspberry Foundation al Regne Unit. Inicialment concebuda per a l'ensenyament de la programació, s'ha expandit a camps com la robòtica, la domòtica i els jocs retro. Les versions modernes, com la Raspberry Pi 5, ofereixen suport per a sortida 4K i una àmplia compatibilitat amb emuladors gràcies als seus processadors i capacitat de memòria RAM (vegeu el [glossari](#)).

- Orientat a principalment a projectes de ciències de la computació a les escoles, és conegut pel seu baix cost i la seva gran comunitat.
- Usos comuns: domòtica, servidors multimèdia i emulació de videojocs retro.
- Exemple: la Raspberry Pi 5 amb un processador ARM Cortex-A76 i fins a 8 GB de RAM permet executar emuladors d'alta exigència com Dolphin.

- **Odroid:**

Orientada a usuaris avançats, aquesta gamma ofereix un alt rendiment gràcies a processadors ARM Cortex d'última generació i suport per a Linux i Android. És una opció molt bona per a centres multimèdia i emuladors gràfics intensius com Dolphin.

- Orientat a usuaris avançats, destaca pel seu alt rendiment.
- Exemple: l'Odroid-H4, amb un processador Intel N97 i 8 GB de RAM, és òptima per a aplicacions multimèdia exigents.

- **BeagleBone:**

Coneguda per la seva robustesa i aplicacions industrials, aquesta placa destaca en la robòtica i l'Internet de les Coses (IoT). Models com la BeagleBone Black són ideals

per a projectes de control en temps real, gràcies a la seva arquitectura ARM Cortex-A8 i una àmplia gamma de ports GPIO (vegeu el [glossari](#)).

- Dissenyat per a aplicacions industrials i educatives.
- Exemple: BeagleBone Black, robust i amb capacitat de gestió en temps real, és adequat per a projectes d'automatització i IoT.

- **ASUS Tinker Board:**

Dissenyada per competir amb la Raspberry Pi, aquesta placa destaca per la seva bona capacitat multimèdia i suport per a vídeo.

- Molt apreciada en projectes d'aprenentatge automàtic.
- Usos comuns: aplicacions multimèdia, reproducció de vídeo en 4K i projectes de realitat augmentada.
- Exemple: amb un processador Rockchip RK3288 i 2 GB de RAM, ofereix una excel·lent capacitat per gestionar projectes gràficament intensius.

- **96Boards:**

Una iniciativa que estandarditza el disseny de SBC per garantir compatibilitat entre fabricants. Són utilitzades en projectes d'investigació i desenvolupament d'aplicacions IoT.

- Usos comuns: investigació acadèmica, desenvolupament de IoT i aplicacions complexes de sistemes integrats.
- Exemple: el model HiKey960, amb un processador ARM Cortex-A73 i 4 GB de RAM, és una elecció habitual per a desenvolupadors Android.

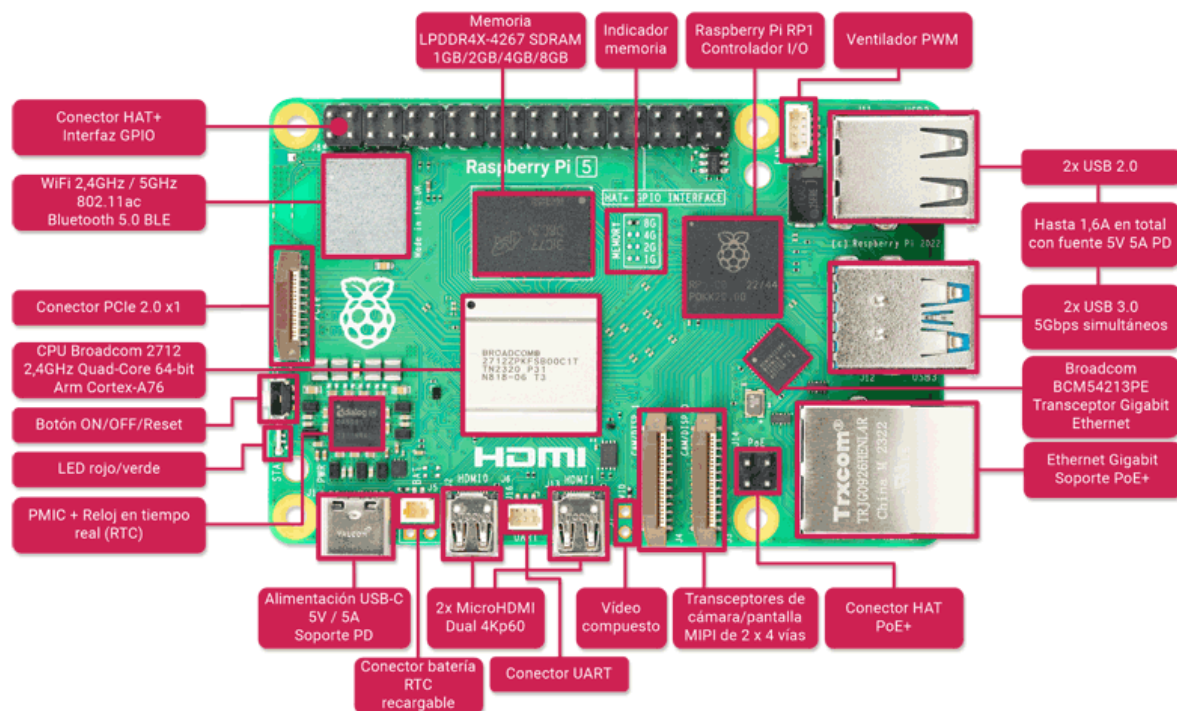


Figura 2.1. Esquema bàsic d'un Single-Board Computer (en aquest cas, d'una Raspberry Pi 5) i els seus components principals (en castellà).

2.1.2. Comparativa entre SBC

En funció de les necessitats del projecte, es presenten a continuació les característiques clau de diferents SBC, incloent-hi opcions més especialitzades com l'ASUS Tinker Board i la 96Boards a més de les opcions més conegudes com la Raspberry Pi, l'Odroid i la BeagleBone.

Nom	Processador	Memòria RAM	Emmagatzematge	Ports	Preu (€)
Raspberry Pi 5	ARM Cortex-A76, 4 nuclis	Fins a 8 GB	microSD, USB SSD	USB 3.0, HDMI	70 - 90 €
Odroid-H4	Intel N97, 4 nuclis	8 GB	eMMC, microSD	USB 3.0, HDMI	140 €
BeagleBone Black	ARM Cortex-A8, 1 nucli	512 MB	4 GB eMMC, microSD	USB, HDMI	45 €
ASUS Tinker Board	Rockchip RK3288, 4 nuclis	2 GB	microSD	USB 3.0, HDMI, GPIO	70-80 €
96Boards HiKey960	ARM Cortex-A73, 4 nuclis	4 GB	eMMC, microSD	USB 3.0, HDMI	200 €

Taula 2.1. Comparativa de característiques d'alguns SBC. Anàlisi tècnica basada en documentació oficial (vegeu el [glossari](#) per a aprofundir en termes més tècnics). Fonts: [[Raspberry Pi, n.d.](#); [BeagleBoard.org, n.d.](#); [ODROID Wiki, n.d.](#); [Asus, n.d.](#), [96boards, n.d.](#)].

1. Processador i potència de càlcul

El processador és un dels elements més crítics per a l'emulació de videojocs, ja que determina la capacitat de reproduir gràfics complexos i gestionar emuladors que requereixen recompilació dinàmica.

- **Raspberry Pi 5:** Amb el seu processador ARM Cortex-A76 de 4 nuclis a 2,4 GHz, ofereix un rendiment molt superior al seu predecessor, la Raspberry Pi 4, que utilitza un ARM Cortex-A72. Això la fa ideal per executar emuladors exigents com Dolphin (GameCube/Wii).
- **Odroid-H4:** El processador Intel N97, amb una freqüència de fins a 3,6 GHz, és el més potent de la comparativa, superant per un gran marge a les alternatives ARM. Aquesta opció és perfecta per a emulació de consoles modernes, però el seu cost és bastant més elevat.

- **ASUS Tinker Board 3:** El seu processador ARM Cortex-A55 de 4 nuclis és adequat per a emulació de consoles clàssiques i projectes multimèdia, però queda limitat per a jocs més exigents.
- **BeagleBone Black:** Amb un sol nucli ARM Cortex-A8, és una opció modesta, adequada només per a emuladors de consoles 2D com NES o Game Boy.

2. Memòria RAM

La quantitat de memòria RAM afecta la capacitat de l'SBC per carregar jocs grans i gestionar múltiples processos simultanis.

- **Raspberry Pi 5:** Amb opcions de 4 i 8 GB, és suficient per a emular consoles 3D i executar emuladors que requereixen més memòria, com PCSX2 per a PlayStation 2.
- **Odroid-H4:** Amb 8 GB de RAM, iguala la Raspberry Pi 5, però la seva arquitectura Intel li permet gestionar millor les aplicacions multitasca intensives.
- **ASUS Tinker Board 3:** Els seus 2-4 GB són suficients per a emulació retro, però poden limitar el rendiment en jocs més moderns.
- **BeagleBone Black:** Els 512 MB de RAM són clarament insuficients per a consoles 3D i limiten l'ús d'emuladors complexos.

3. Connexions i expansibilitat

Les connexions físiques i sense fils són crucials per integrar perifèrics com comandaments, pantalles o emmagatzematge extern.

- **Raspberry Pi 5 i Odroid-H4:** Amb ports USB 3.0, HDMI i Ethernet, són molt versàtils. La Raspberry Pi 5 és recolzada per una comunitat àmplia i característiques de gran valor, que facilita la compatibilitat amb múltiples perifèrics.
- **ASUS Tinker Board 3:** També ofereix una àmplia gamma de connexions, incloent-hi HDMI 4K i ports GPIO, però no té el mateix suport comunitari que les Raspberry Pi.
- **BeagleBone Black:** Disposa de connexions bàsiques com HDMI i USB, però queda enrere en termes de versatilitat.

4. Cost i suport comunitari

El cost és un factor determinant, especialment en projectes amb pressupost ajustat. A més, el suport comunitari pot ser un avantatge crucial en termes de documentació i resolució de problemes.

- **Raspberry Pi 5:** Combina un preu competitiu (70–90 €) amb el suport de la comunitat més gran d'usuaris d'SBC.
- **Odroid-H4:** Amb un preu de 140 €, és més cara, però ofereix prestacions úniques. Tanmateix, la comunitat és menys activa que la de Raspberry Pi.
- **ASUS Tinker Board 3:** El seu preu (70–80 €) és similar a la Raspberry Pi 5, però el seu suport comunitari és limitat.
- **BeagleBone Black:** És l'opció més barata (45 €), però les seves limitacions tècniques la fan menys adequada per a emulació avançada.

Aquestes característiques dels SBC els fan ideals com a base per desenvolupar una microconsola i per a altres tipus de projectes d'emulació de videojocs, gràcies al seu cost reduït, baix consum energètic i versatilitat.

La Raspberry Pi 5 destaca com la millor opció gràcies al seu equilibri entre cost, rendiment i suport comunitari. Encara que alternatives com l'Odroid-H4 ofereixen més potència bruta, el seu preu i la seva menor integració amb emuladors limiten la seva aplicació pràctica en aquest projecte. D'altra banda, opcions com l'ASUS Tinker Board 3 poden ser adequades per a projectes menys exigents gràficament, mentre que la BeagleBone Black només és viable en aplicacions educatives o amb requisits molt baixos.

Aquesta anàlisi reafirma que la **Raspberry Pi 5** és l'SBC més equilibrat per al propòsit d'aquest treball, oferint una experiència d'emulació fluida sense comprometre el pressupost.

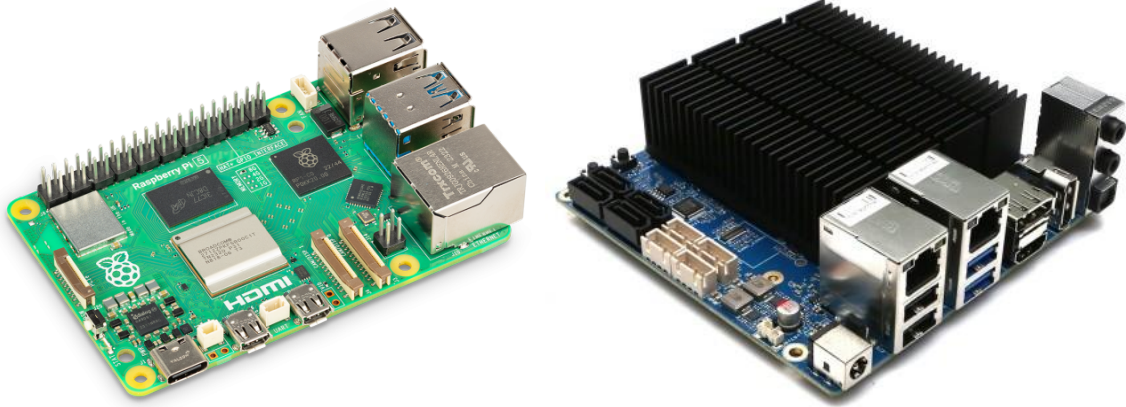


Figura 2.2. Exemples de disseny físic de SBC: (a) Raspberry Pi 5, (b) Odroid-H4.

2.2. Microconsoles i dispositius de streaming

Les microconsoles i els dispositius de streaming són tecnologies que combinen portabilitat i funcionalitat per oferir experiències multimèdia o de joc. Tot i compartir similituds, tenen finalitats i característiques diferents, que les fan adequades per a diferents tipus d'usos.

Les microconsoles són una aplicació directa dels SBC, que combinen potència computacional amb l'accessibilitat de les consoles tradicionals. Al mateix temps, els dispositius de streaming il·lustren com la funcionalitat multimèdia es pot combinar amb l'execució de jocs al núvol o via emuladors. Aquesta dualitat permet als SBC oferir solucions personalitzades segons les necessitats del projecte.

2.2.1. Definicions i funcions

- **Microconsoles:** Són consoles de videojocs de baix cost i dimensions reduïdes, dissenyades per a executar jocs digitals a través de descàrregues o streaming. Estan orientades a un públic casual i ofereixen compatibilitat amb

títols retro o independents, sovint gràcies a l'ús d'emuladors. Per exemple, la **Ouya**, la **Gamestick** i l'**Atari VCS** han estat creades específicament per a aquest propòsit.

- **Dispositius de streaming:** Són aparells dissenyats principalment per a la reproducció de contingut multimèdia (vídeos, música, etc.) a través d'Internet. Inclouen opcions com el **Google Chromecast**, l'**Amazon Fire TV Stick** i l'**Apple TV**, que també poden executar aplicacions de jocs via streaming, com Google Stadia o Amazon Luna.

Diferències clau:

- Les microconsoles estan optimitzades per al joc i, en molts casos, integren emuladors per accedir a títols clàssics.
- Els dispositius de streaming prioritzen el contingut multimèdia, amb funcions de joc limitades o depenent de serveis en el núvol.

En resum, les microconsoles són adequades per a aquells que busquen una experiència de joc accessible i econòmica, mentre que els dispositius de streaming són més versàtils però menys específics per al joc.

2.2.2. Exemples, usos i comparació

Per entendre millor les possibilitats que ofereixen aquests dispositius, es presenten les definicions i característiques d'alguns productes representatius.

Dispositiu	Tipus	Funcionalitats destacades	Preu (€)
NVIDIA Shield TV	Híbrid (streaming + jocs)	Jocs en el núvol, Android TV, suport per emuladors	150–200 €
Amazon Fire TV Stick	Streaming	Accés a serveis com Netflix, Spotify; compatibilitat amb Amazon Luna	40–60 €
Google Chromecast	Streaming	Reproducció de contingut multimèdia; suport limitat per a jocs	35–70 €
Ouya	Microconsola	Emulació de jocs retro, Android OS, baix cost	Discontinuada (50 € aprox.)
Atari VCS	Microconsola	Experiència retro moderna, compatibilitat amb emuladors	300 € aprox.

Taula 2.2. Comparativa de característiques d'alguns dispositius famosos. Fonts: [\[NVIDIA Shield TV Pro, n.d.\]](#); [\[Amazon Fire TV 4K, n.d.\]](#); [\[Ouya, n.d.\]](#); [\[Atari, n.d.\]](#)

1. Funcionalitat híbrida: NVIDIA Shield TV

El **NVIDIA Shield TV** sobresurt per combinar les millors característiques dels dispositius de streaming i les microconsoles. És capaç de reproduir contingut multimèdia en alta definició i permet jugar tant a jocs Android com a jocs en streaming amb GeForce Now. A més, té suport per a emuladors de consoles clàssiques. Això el converteix en una opció excel·lent per a usuaris que volen una solució completa.

2. Accessibilitat i preu: Amazon Fire TV Stick i Google Chromecast

Aquests dispositius són idonis per a consumidors que prioritzen la reproducció de contingut multimèdia. Encara que tenen capacitats limitades per als jocs, serveis com Amazon Luna o Google Stadia permeten jugar en el núvol sense la necessitat de hardware addicional. Tot i això, no són una opció sòlida per a emulació o jocs intensius gràfics. ([TomsGuide, n.d.](#))

3. Focalització en jocs retro: Ouya i Atari VCS

- **Ouya:** Va ser una pionera en el camp de les microconsoles amb un enfocament clar en jocs retro i independents gràcies a la seva base Android. Tot i que ja no es comercialitza, segueix sent un exemple del que poden oferir aquests dispositius.
- **Atari VCS:** Una opció moderna per a nostàlgics, amb un disseny atractiu i capacitat per a emular jocs clàssics. No obstant això, el seu preu elevat la situa fora de l'abast de molts consumidors casuals.

Comparació segons els usos

1. **Per a jocs retro:** Les microconsoles com la Ouya i l'Atari VCS són opcions més adequades, gràcies a la seva compatibilitat amb emuladors. Això les converteix en una opció atractiva per a aficionats als jocs clàssics.
2. **Per a contingut multimèdia:** Els dispositius de streaming com el Chromecast i el Fire TV Stick són la millor elecció, oferint accés senzill i econòmic a una àmplia varietat de serveis multimèdia.
3. **Opcions híbrides:** El NVIDIA Shield TV és l'opció més versàtil, combinant la funcionalitat de les dues categories amb un rendiment alt en totes les seves funcions.



Figura 2.3. Comparació visual d'una microconsola (Ouya) i un dispositiu de streaming (Amazon Fire TV)

Les microconsoles i els dispositius de streaming representen dues aproximacions diferents de l'entreteniment digital. En el context d'aquest projecte, el focus principal serà l'ús d'SBC per crear una solució personalitzada per a replicar la funcionalitat d'una microconsola, i que aprofiti els seus avantatges per als jocs. Aquesta tasca es durà a terme integrant la placa amb emuladors i plataformes com RetroArch.

2.3. Emulació de videojocs

L'emulació de videojocs és una pràctica que permet jugar a títols de consoles clàssiques o modernes en dispositius diferents dels seus sistemes originals. Aquesta tècnica ha evolucionat significativament, oferint noves funcionalitats i compatibilitats que no eren possibles en el maquinari original.

Els emuladors són el motor que transforma els SBC en microconsoles funcionals. Aquest projecte es basa en aprofitar el potencial d'aquests dos elements combinats

per recrear experiències de joc diverses. Mitjançant l'ús de plataformes multipropòsit, com RetroArch o Recalbox, els emuladors poden gestionar múltiples sistemes en un sol dispositiu, cosa que optimitza els processos tècnics, simplificant la gestió de jocs i oferint una solució completa en un dispositiu compacte; i alhora facilita la compatibilitat amb una àmplia varietat de consoles, maximitzant l'eficiència dels recursos disponibles.

2.3.1. Definició i objectius de l'emulació

L'emulació es basa en la reproducció via *software* del maquinari d'una consola per executar els seus jocs sense necessitat de disposar del sistema original. Els principals objectius són:

1. **Preservació de jocs antics:** Molts jocs clàssics ja no estan disponibles al mercat, i l'emulació permet accedir-hi preservant el patrimoni cultural dels videojocs.
2. **Millora gràfica i funcional:** Els emuladors moderns ofereixen millores com resolucions més altes, textures millorades, o funcions extra com estats de guardat ràpid i controls personalitzats.
3. **Accés i compatibilitat:** Permet jugar a títols de diverses consoles en un mateix dispositiu, eliminant la necessitat d'adquirir múltiples plataformes.

2.3.2. Principals emuladors i compatibilitats

Emuladors individuals

A continuació es detallen els emuladors més populars, les seves funcionalitats i compatibilitats:

Emulador	Consoles emulades	Compatibilitat amb els jocs de les pròpies consoles	Millores gràfiques	Característiques addicionals
Dolphin	GameCube, Wii	Molt alta	4K, <i>antialiasing</i>	Suport per a Wii Remotes i xarxes
PCSX2	PlayStation 2	Alta (amb problemes menors)	Resolució interna escalada	Memòries virtuals, estats de guardat
Citra	Nintendo 3DS	Moderada	Millores de resolució	Xarxes locals, estats de guardat
PPSSPP	PlayStation Portable (PSP)	Molt alta	Filtres de textures, resolució	Xat en línia, controls personalitzats
MAME	Màquines Arcade	Alta (per jocs arcade)	Fidelitat als originals	Suport per a una gran varietat de màquines arcade

Taula 2.3. Característiques d'alguns emuladors molt coneguts. Fonts: [[Dolphin Emulator, n.d.](#); [PCSX2, n.d.](#)]

Cada emulador té un nínxol específic, per exemple:

- **Dolphin i PCSX2** són excel·lents per a jocs 3D de consoles modernes (emulen GameCube/Wii i Play Station 2 respectivament).
- **MAME** és indispensable per als amants dels jocs arcade.
- **PPSSPP i Citra** són líders en emulació de consoles portàtils (emulen PSP i 3DS respectivament).

Plataformes d'emulació multipropòsit

A més dels emuladors individuals, hi ha plataformes que suporten múltiples emuladors i ofereixen una experiència integrada per gestionar diversos sistemes. Aquestes plataformes són especialment útils en projectes d'emulació en SBC.

Plataforma	Tipus de consoles suportades	Compatibilitat amb els jocs de les consoles	Millores gràfiques	Característiques destacades
RetroArch	Variada (NES, SNES, PlayStation, etc.)	Alta	Suport per a shaders, filtres i millores de resolució	Tot-en-un, suporta múltiples emuladors
Recalbox	Sobretot retro i arcade	Alta, però depèn de la consola	Millores de resolució	Interfície intuïtiva
Lakka	Similar a RetroArch	Alta	Idèntiques a RetroArch	Optimitzat per a dispositius de baix cost

Taula 2.4. Comparativa de característiques d'algunes plataformes d'emulació. Fonts: [[Lakka, n.d.](#); [Recalbox, n.d.](#)]

1. **RetroArch:** Aquesta plataforma centralitza múltiples emuladors gràcies al seu sistema de "nuclis", que permet descarregar i gestionar emuladors individuals dins d'una interfície integrada. A més de la seva versatilitat, RetroArch inclou funcionalitats avançades com shaders, filtres de resolució i opcions per reduir la latència.
2. **Recalbox:** Orientada a l'ús casual, Recalbox ofereix una interfície intuïtiva amb configuracions predeterminades que simplifiquen la posada en marxa. És

ideal per a projectes en SBC gràcies al seu enfocament plug-and-play, encara que no és tan personalitzable com RetroArch.

3. **Lakka:** Basada en RetroArch, però optimitzada per a dispositius de baix cost com les Raspberry Pi, Lakka redueix la càrrega del sistema operatiu i proporciona una experiència fluida fins i tot en maquinari limitat. La seva configuració minimalista la fa una opció robusta per a projectes que prioritzen el rendiment sobre la personalització.

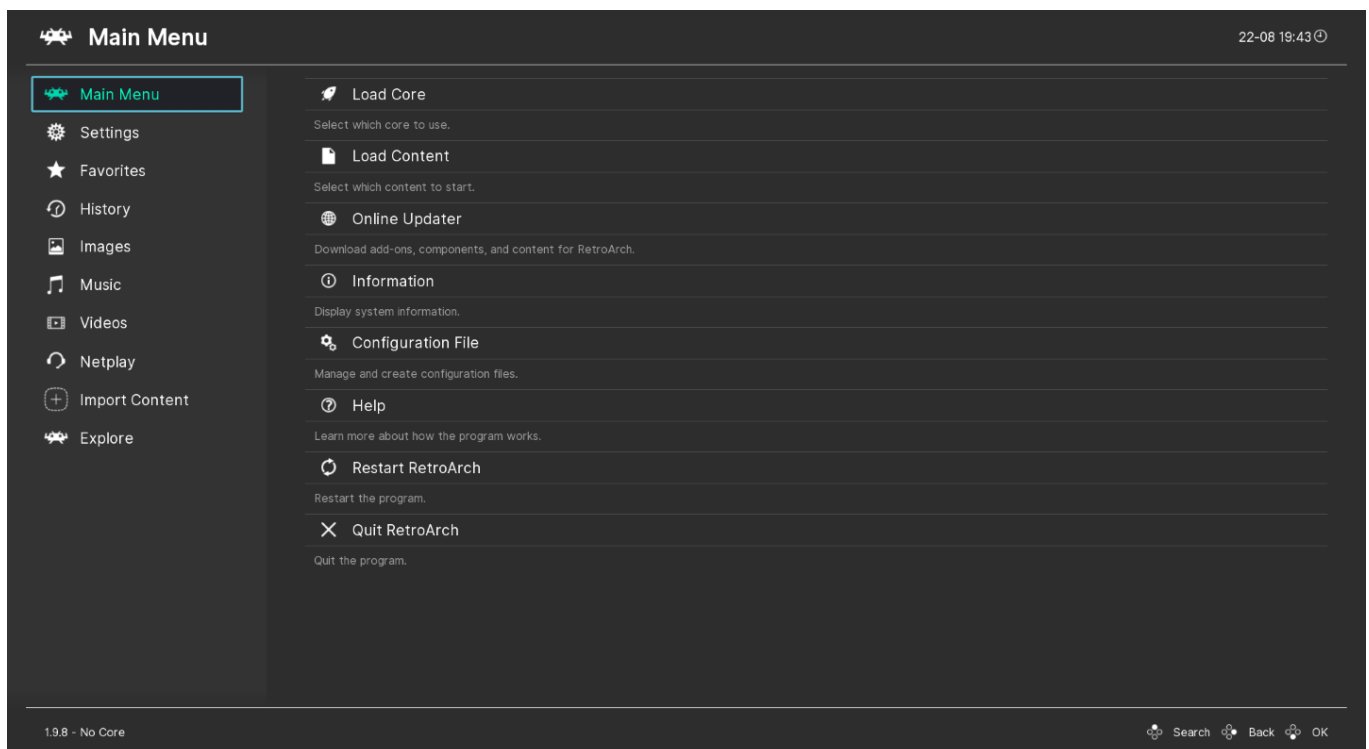


Figura 2.4. Interfície gràfica (GUI) principal de RetroArch mostrant diferents opcions disponibles.

2.3.3. Aspectes ètics de l'emulació

L'ús d'emuladors de videojocs ha estat objecte de debat en l'àmbit legal i ètic. Tot i que són eines potents per preservar i gaudir de jocs clàssics, el seu ús sovint es troba en un territori gris pel que fa als drets d'autor i la moralitat de la seva aplicació.

Legalitat i moralitat de l'emulació

L'emulació en si mateixa no és il·legal, ja que desenvolupar programes que imiten el funcionament d'una consola no viola necessàriament els drets d'autor. No obstant això, la legalitat pot variar segons com s'utilitzin els emuladors i, especialment, les ROM.

1. Emuladors:

- Els emuladors de codi obert solen estar dins dels límits legals sempre que no incloguin propietat intel·lectual protegida, com el *firmware* (vegeu el [glossari](#)) o BIOS propietari d'una consola.
- Per exemple, projectes com RetroArch i Dolphin utilitzen codis desenvolupats des de zero per evitar violacions de copyright.

2. ROM i BIOS:

- Les ROM són còpies digitals dels jocs originals, i el seu ús sense permís dels titulars dels drets d'autor és il·legal.
- Els usuaris només poden legalment utilitzar ROM i BIOS si són còpies de seguretat creades a partir dels jocs i consoles que ja posseeixen físicament.

Encara que els emuladors solen ser legals, la descàrrega de ROM o l'ús de BIOS sense permís planteja problemes ètics i legals importants. En aquest projecte, es farà ús exclusiu de ROM i BIOS generades com a còpies personals d'equipament original.

Impacte en la indústria dels videojocs

L'emulació pot tenir un impacte positiu i negatiu en la indústria:

1. Impactes positius:

- **Preservació cultural:** Els emuladors són essencials per conservar jocs antics que ja no estan disponibles al mercat, protegint-ne el patrimoni cultural. Això és especialment rellevant en consoles obsoletes, com NES o Sega Genesis.

- **Accés educatiu:** Permeten estudiar el disseny de videojocs antics i innovar en noves tecnologies gràcies a l'anàlisi de consoles clàssiques

2. Impactes negatius:

- **Pèrdues econòmiques:** Quan s'utilitzen per jugar a jocs recents o disponibles comercialment, els emuladors poden causar una reducció d'ingressos per als desenvolupadors i editors, especialment per a títols independents o petites empreses.
- **Desincentivació de reedicions:** Si una gran part del públic accedeix a jocs mitjançant emuladors, les empreses poden veure menys rendible invertir en remasteritzacions o llançaments oficials.

Reflexió ètica sobre l'ús de l'emulació

L'ús d'emuladors té una clara justificació ètica quan es fa per preservar jocs antics o per accedir a títols que ja no estan disponibles legalment. No obstant això, és fonamental assegurar que aquest ús no comprometi els interessos dels creadors, especialment en casos de jocs disponibles comercialment.

Bones pràctiques recomanades:

- Utilitzar únicament ROM i BIOS provinents de còpies personals de jocs i consoles adquirides legalment.
- Prioritzar l'ús d'emuladors de codi obert desenvolupats legalment.
- Reconèixer i respectar els esforços dels creadors adquirint jocs oficials sempre que sigui possible.

Els emuladors són una eina valuosa per a la preservació i l'estudi dels videojocs, però el seu ús ha de respectar els drets dels creadors i contribuir a un ecosistema de consum ètic i legal. Aquesta reflexió ètica ha guiat a la selecció de ROMs i BIOS fetes servir en aquest projecte que assegurin que totes compleixen amb les directrius legals i morals, prioritzant la preservació cultural i el respecte pels creadors. [[TimeExtension](#), n.d.; [PC Gamer](#), n.d.; [Irdeto](#), n.d.]

2.4. Relació entre els conceptes: SBC, microconsoles i emuladors

Aquest projecte integra tres elements clau – SBC, microconsoles i emuladors – per construir una solució funcional i optimitzada per a l'emulació de videojocs. Cada un d'aquests conceptes, desenvolupats en detall als apartats anteriors, s'interrelaciona de manera que l'objectiu general es pugui assolir amb eficàcia.

Els SBC serveixen com el cor tecnològic, proporcionant la base de hardware necessària per donar suport als emuladors. Les microconsoles aprofiten les capacitats d'aquests SBC per crear dispositius compactes i econòmics, orientats a l'emulació retro o l'entreteniment multimèdia. Finalment, els emuladors s'encarreguen d'adaptar jocs clàssics per executar-los en maquinari modern, integrant-los en plataformes com RetroArch o Lakka, que unifiquen aquesta experiència.

Per tant, l'usuari que volgués desenvolupar un projecte similar o simplement gaudir dels diferents emuladors i jocs que hi ha al mercat pot escollir entre dues opcions:

- SBC + Emuladors/Plataforma, ideal si es busca una solució econòmica i personalitzable, però cal invertir temps en la seva configuració i personalització, cosa que permet a més altres usos a més de l'emulació pròpiament dita. Les SBC són més versàtils, permetent connectar dispositius addicionals com controls, teclats o pantalles tàctils. Com a punts en contra podem dir que la corba d'aprenentatge és més lenta, ja que requereix coneixements de Linux, configuració de software i ús de la terminal.
- La microconsola és millor si es busca una solució senzilla, plug and play i sense massa complicació. Ideal per aquelles persones que només volen jugar, sense preocupar-se de la configuració del sistema. Són bastant més cares que les SBC tot i tenir un hardware molt més limitat pel que fa a versatilitat.

Donat el caire de recerca del projecte, estudiarem en detall el primer cas: SBC + Emuladors/Plataformes, buscant les millors condicions per emular (i jugar) al major nombre de jocs possibles.

En el capítol següent, es desenvoluparan les proves inicials realitzades sobre un SBC concret. Aquestes proves tenen l'objectiu de validar l'eficàcia dels emuladors seleccionats i analitzar com el hardware i el software treballen conjuntament per oferir una experiència de joc fluida i fiable. Això permetrà establir una metodologia que guiarà les fases posteriors del projecte, assegurant que els conceptes teòrics es tradueixin en resultats pràctics.

Capítol 3.

Arquitectura de capes i proves inicials al SBC

En aquest capítol es descriu l'arquitectura de capes necessària per emular videojocs en un Single-Board Computer, exposant els conceptes principals de les 5 capes explorades del sistema. A més, es justifica l'elecció de la placa que es farà servir, i es presenten les primeres proves i comparatives pràctiques realitzades en les plataformes com Recalbox, RetroArch i Lakka, tot valorant el seu rendiment i estabilitat. Finalment, es justifica quina plataforma esdevé la més adequada en abordar una optimització a fons en el capítol següent.

L'objectiu principal és seleccionar el sistema que ofereix millors garanties per a la posterior fase d'optimització.

3.1. Capes en l'emulació: anàlisi i funcionament

L'emulació de videojocs en un SBC és un procés complex que requereix la col·laboració de diverses capes tecnològiques per recrear el comportament d'una consola original en un entorn diferent. Aquestes capes inclouen el maquinari que executa l'emulador, el sistema operatiu que gestiona els recursos disponibles, la plataforma d'emulació que organitza els mateixos emuladors i els nuclis específics per a cada consola, així com les ROM que reproduïxen el contingut dels jocs.

Per entendre el funcionament i el rendiment d'un emulador, és essencial analitzar aquestes cinc capes que intervenen en el procés.

3.1.1. Capes principals

1. Hardware (o maquinari)

El hardware és la capa base sobre la qual es construeix tot el sistema d'emulació. És responsable de proporcionar la potència de càlcul i els recursos físics necessaris per executar els emuladors i gestionar les dades dels jocs.

En aquesta capa, la CPU és l'encarregada de traduir i executar les instruccions dels nuclis d'emulació, mentre que la GPU s'encarrega de les tasques gràfiques, com renderitzar textures i efectes visuals. La memòria RAM actua com a espai temporal per emmagatzemar dades mentre els emuladors i jocs s'executen. L'eficiència d'aquest maquinari és crucial, ja que qualsevol limitació en aquesta capa pot convertir-se en un coll d'ampolla per a les capes superiors, afectant el rendiment general.

Pel que fa al rendiment tèrmic, és important mantenir el sistema a temperatures òptimes. Sense refrigeració activa, la CPU pot superar els 80 °C durant càrregues intenses, activant el *thermal throttling* i reduint el rendiment. Per evitar-ho, és recomanable utilitzar ventiladors o dissipadors de calor per mantenir temperatures entre 35 i 45 °C durant les sessions de joc (vegeu el [glossari](#) per a més informació sobre termes més tècnics).

- **Impacte en el rendiment:** El processador i la GPU permeten una emulació fluida de jocs 2D i 3D.
- **Compatibilitat ARM:** Molts emuladors moderns estan optimitzats per arquitectura ARM, però molts altres (com PCSX2 per PS2) requereixen arquitectures x86 per oferir un rendiment òptim o simplement per a poder executar-se.
- **Temperatura:** És important controlar la temperatura i implementar sistemes de refrigeració per evitar la reducció automàtica de la velocitat del processador.

2. Sistema Operatiu

Un sistema operatiu gestiona els recursos del hardware, com la CPU, la memòria i els dispositius d'entrada/sortida, per garantir una execució eficient dels processos. Aquesta gestió implica assignar recursos, sincronitzar processos i protegir la memòria per evitar interferències. A més, el nucli del sistema operatiu utilitza tècniques de planificació, com el canvi de context, per maximitzar l'eficiència del sistema i equilibrar l'ús de recursos entre múltiples processos.

En entorns amb recursos limitats, com els SBC, és clau utilitzar distribucions lleugeres i optimitzacions del nucli per reduir la competència pels recursos. Això inclou eliminar processos innecessaris i ajustar el funcionament del nucli per millorar la latència i estabilitat. Aquestes optimitzacions són especialment importants per a aplicacions com l'emulació de videojocs, on la càrrega sobre el sistema pot ser elevada.

- **Impacte en el rendiment:** Lakka redueix la càrrega del sistema operatiu, eliminant processos innecessaris i centrant-se en l'emulació, cosa que millora el rendiment.
- **Configurabilitat:** Tots dos, però sobretot Raspberry Pi OS Lite, ofereixen eines per ajustar rendiment gràfic i latència.
- **Recursos:** Les distribucions lleugeres, com Lakka, en ajudar a reduir la competència pels recursos entre processos, milloren l'estabilitat i la latència de l'emulació.

3. Plataforma d'emulació

Les plataformes d'emulació com Lakka i Recalbox són essencials per gestionar els emuladors i proporcionar una experiència d'usuari personalitzable i optimitzada. Aquestes plataformes permeten ajustar configuracions avançades, com ara els shaders, que milloren l'aspecte visual dels jocs simulant efectes com pantalles CRT o aplicant filtres *antialiasing* (vegeu el [glossari](#)), tot i que això pot augmentar la càrrega de la GPU. A més, funcions com la sincronització de vídeo garanteixen que els fotogrames per segon (FPS) siguin consistents amb la freqüència de refresc del monitor, evitant problemes de dessincronització. El mode *Run Ahead* redueix la

latència percebuda avançant la simulació d'inputs, millorant la resposta als controls en jocs ràpids.

- **Impacte en el rendiment:** Configuracions com els shaders poden millorar l'aspecte visual dels jocs, aplicant filtres com *antialiasing* o efectes CRT per simular pantalles antigues, però també augmenten la càrrega de la GPU.
- **Sincronització de vídeo:** Ajusta els *frames* per segon per evitar dessincronitzacions amb el monitor.
- **Run Ahead:** Redueix la latència percebuda avançant la simulació d'inputs.

4. Nuclis d'emulació

Els nuclis d'emulació (en anglès, *cores*, i sent una altra denominació per a emuladors), tal com s'ha vist en l'apartat [2.3](#), són programes especialitzats que reproduïxen el funcionament del maquinari d'una consola específica. Aquests nuclis són fonamentals per al procés d'emulació, ja que tradueixen el codi original dels jocs a instruccions que el maquinari modern pot executar. Per aconseguir-ho, fan ús de tècniques com la recompilació dinàmica (Dynarec) o la compilació Just-In-Time (JIT), que permeten executar el codi en temps real amb un rendiment superior en comparació amb els modes d'interpret tradicionals. Això és crucial per a jocs 3D complexos, que requereixen càlculs intensius.

- **Recompilació dinàmica (Dynarec):** Genera i executa codi optimitzat en temps real, millorant el rendiment.
- **Compilació JIT (Just-In-Time):** Converteix codi natiu del joc a instruccions per al hardware modern mentre el joc s'executa, equilibrant rendiment i compatibilitat.
- **Modes d'interpret:** Opció més lenta però més precisa, útil per a depuració o compatibilitat amb jocs problemàtics.
- **Optimització per arquitectura:** Alguns nuclis estan optimitzats per arquitectures ARM (com Raspberry Pi), mentre que altres necessiten x86 per rendir al màxim.

- **Impacte en la compatibilitat:** Els nuclis ben optimitzats poden reproduir una gran varietat de jocs sense errors, però això depèn de la configuració específica i de la potència del hardware.

5. Jocs (ROM)

Les ROM són els fitxers digitals que contenen el codi original dels jocs i permeten a l'emulador recrear l'experiència de joc de la consola. A més de les ROM, algunes consoles requereixen BIOS per funcionar correctament, ja que aquestes actuen com a *firmware* essencial per inicialitzar el sistema. La qualitat i optimització de les ROM són factors clau per garantir un bon rendiment en l'emulació, perquè els fitxers grans o mal optimitzats poden generar problemes com errors gràfics, caigudes de FPS o inestabilitat general.

- **Tipologia dels jocs:**
 - **Jocs 2D:** Consumeixen menys recursos de CPU i memòria, generalment més fàcils de reproduir.
 - **Jocs 3D:** Requereixen més capacitat de processament, especialment en consoles com PS2 o GameCube.
- **Requisits de BIOS:** Algunes consoles necessiten fitxers de BIOS per carregar les ROM correctament (PS2, GameCube...). Altres, com NES o GBA, no els necessiten perquè el nucli emula el *firmware*.
- **Impacte de l'optimització:**
 - Les ROM optimitzades redueixen la càrrega en el sistema, minimitzant errors i millorant el rendiment.
 - Les ROM de baixa qualitat poden augmentar la latència o presentar errors visuals i auditius.
- **Talla i estructura:** Els fitxers de grans dimensions poden requerir més memòria i alentir l'accés als recursos, especialment en dispositius amb targetes SD més lentes.

3.2. Primeres proves amb diferents plataformes d'emulació

Un cop establerta l'arquitectura de capes i seleccionada la placa SBC adequada, és essencial dur a terme proves preliminars amb les diferents plataformes d'emulació disponibles. Aquestes proves inicials permeten avaluar la facilitat d'instal·lació, la compatibilitat amb els emuladors seleccionats, així com el rendiment bàsic en l'execució de diversos títols de videojocs representatius. A més, aquestes proves ajuden a identificar possibles problemes de configuració o incompatibilitat que podrien afectar el rendiment i l'estabilitat del sistema. En aquesta secció, es detalla el procés de selecció i instal·lació de les plataformes, així com les primeres observacions derivades de l'execució dels jocs de prova en cada entorn, sense cap modificació de rendiment.

3.2.1. Selecció del SBC adequat

D'entre les opcions d'SBC explorats al capítol 2, el millor per als meus interessos és la **Raspberry Pi 5 de 8 GB de RAM**. Aquesta elecció es fonamenta en la seva capacitat superior de processament i la memòria RAM ampliada, que la fa ideal per a un projecte d'emulació de videojocs que exigeix un rendiment constant. Tot i que la versió de 4 GB de la Raspberry Pi 5 ofereix una excel·lent relació qualitat/preu, la variant de 8 GB proporciona més flexibilitat per gestionar emuladors que requereixen més memòria, especialment quan es tracta de consoles modernes o quan s'executen múltiples emuladors alhora. Aquesta característica fa que la Raspberry Pi 5 de 8 GB sigui la millor opció per assegurar un rendiment fluid i sostingut en diversos contextos.

Comparació amb altres alternatives

En una anàlisi comparativa, la Raspberry Pi 5 de 8 GB sobresurt davant altres opcions com l'Odroid H4 i l'ASUS Tinker Board 3, entre altres:

- **Odroid H4:** Tot i oferir un rendiment comparable, el seu cost significativament més alt (al voltant de 140 €) fa que no sigui tan competitiva per aquest tipus de projecte.
- **ASUS Tinker Board 3:** Té un preu similar, però no iguala el suport comunitari ni la disponibilitat de recursos de programari optimitzats que caracteritzen la Raspberry Pi 5. Això pot augmentar la complexitat i el cost de manteniment del projecte.

La Raspberry Pi 5 de 8 GB ofereix una molt bona combinació de cost, rendiment i flexibilitat, convertint-la en la millor opció per a un projecte d'emulació de videojocs com el que vull dur a terme. Les seves especificacions tècniques, el seu ecosistema de programari i el suport comunitari robust garanteixen un funcionament òptim tant a curt com a llarg termini.

3.2.2. Definicions de les plataformes i jocs a testejar

Per tal de dur a terme una comparació representativa del rendiment en cada plataforma, s'han seleccionat quatre títols que cobreixen tant escenaris en 2D com en 3D. Aquests jocs es caracteritzen per la seva popularitat i per la variabilitat en els requisits de potència gràfica i computacional:

1. **Metroid: Zero Mission (GBA):** Un joc d'aventures i acció, amb un estil clàssic de plataformes i exploració. Aquest permet comprovar la fluïdesa i la compatibilitat bàsica en jocs 2D i, en principi, oferir un bon test de velocitat de fotogrames (60 FPS idealment).
2. **Mario Kart DS (Nintendo DS):** És un joc de curses de karts en 3D simple, amb dissenys poligonals senzills però moviment ràpid. Aquest és un joc útil per avaluar la precisió del *framerate* en un joc de velocitat i la gestió de la latència en controls ràpids.
3. **God of War: Chains of Olympus (PSP)**
Joc de la PSP destacat per la seva acció en tercera persona i un important desgast de recursos 3D, que inclou escenes cinematogràfiques.

Aquest joc posa a prova la càrrega de la CPU i la GPU, i també la gestió de la memòria en cinemàtiques.

4. ***The Legend of Zelda: The Wind Waker (GameCube)***

És un joc d'aventura 3D amb un món obert moderadament ampli, que permet avaluar fins a quin punt la plataforma pot assumir jocs 3D exigents. És un dels títols més representatius de GameCube.

Amb aquests quatre títols, es cobreix un ventall de casos d'ús que va des de jocs retro en 2D de baixa exigència fins a títols 3D més complexos. D'aquesta manera, és possible avaluar la consistència del rendiment en diferents emuladors que presentaré a continuació i comprovar quins d'ells ofereixen una experiència òptima de joc, tant pel que fa a FPS com a temps de càrrega, qualitat gràfica o latència en els controls.

Per a aquests emuladors, es van seleccionar els següents pel seu rendiment i compatibilitat amb la llista de jocs prèviament mencionada:

- **mGBA:** Per a Game Boy Advance, és conegut per la seva alta compatibilitat amb jocs i per ser lleuger en recursos.
- **DeSmuME:** Especialitzat en Nintendo DS, amb opcions gràfiques personalitzables.
- **Dolphin:** Emulador per a Nintendo GameCube i Wii, capaç de millorar significativament la qualitat gràfica en comparació amb el maquinari original.
- **PPSSPP:** Per a PlayStation Portable (PSP), amb millores en textures i resolució.

Per a l'ús integrat d'aquests emuladors individuals, es van explorar plataformes que centralitzen múltiples emuladors per simplificar la gestió i configuració:

1. **RetroArch:** Plataforma tot-en-un que suporta diversos emuladors a través dels nuclis. Inclou funcionalitats avançades com shaders i filtres gràfics.
2. **Recalbox:** Sistema operatiu dedicat a l'emulació retro, amb una interfície intuïtiva i suport per a una àmplia gamma d'emuladors.
3. **Lakka:** Basada en RetroArch, optimitzada per a dispositius de baix cost com la Raspberry Pi.

4. **RetroPie:** Orientada als jocs retro. Actualment, no hi ha una imatge oficial per a Raspberry Pi 5, però es va poder instal·lar sobre Raspberry Pi OS Lite.

3.2.3. Instal·lacions

(Vegeu l'[apèndix A](#) per a més informació sobre les instal·lacions)

Recalbox: La instal·lació de Recalbox és senzilla i consisteix a gravar la imatge del sistema operatiu en una targeta SD. La interfície gràfica intuïtiva permet configurar la placa sense complicacions. No obstant això, dels emuladors provats, només mGBA (Game Boy Advance) ve preconfigurat. Per a consoles més exigents, com Nintendo DS, PSP o GameCube, cal configurar manualment els nuclis o afegir altres emuladors, la qual cosa pot ser una limitació per a usuaris menys experimentats.

RetroArch (sobre Raspberry Pi OS): La instal·lació es va realitzar sobre Raspberry Pi OS de 32 bits, ja que la versió de 64 bits presenta problemes de compilació amb els nuclis. RetroArch destaca per la seva flexibilitat i permet descarregar nuclis des de la seva interfície. Tot i això, cal afegir manualment BIOS i ajustar configuracions per aconseguir que consoles més avançades, com GameCube, Wii o Nintendo DS, funcionin correctament.

Lakka: Igual que Recalbox, s'instal·la com un sistema operatiu complet en una targeta SD. Es basa en RetroArch i ofereix una configuració inicial senzilla i estable. A diferència de Recalbox, venen bastants emuladors ja instal·lats i preconfigurats per defecte, podent servir fàcilment per a usuaris més casuals.

RetroPie (sobre Raspberry Pi OS Lite): Aquesta es va instal·lar sobre Raspberry Pi OS Lite, ja que actualment no hi ha una imatge oficial per a Raspberry Pi 5. Tot i que ofereix una interfície gràfica (EmulationStation) intuïtiva, les versions no oficials presenten problemes d'estabilitat, cosa que va impedir avaluar-la.

3.2.4. Seguiment de la prova

Les proves es van realitzar utilitzant les configuracions predeterminades dels emuladors, assegurant una comparació justa. Els paràmetres analitzats van ser:

1. **Ràtio de fotogrames per segon (FPS):** Estabilitat a 60 FPS, especialment en escenes exigents.
2. **Temps de càrrega:** Temps requerit des de l'inici de l'emulador fins que el joc és jugable, mesurat a ull nu.
3. **Latència d'entrada (*Input lag*):** Resposta dels controls per detectar possibles retards, mesurat també a ull nu.
4. **Qualitat gràfica:** Presència d'errors gràfics o artefactes visuals que puguin ser inconvenients per a l'experiència gràfica del joc.

En la taula de la pàgina següent es poden veure els resultats dels quatre paràmetres esmentats en els diferents jocs i plataformes.

Observacions principals

- **RetroPie:** No va ser viable per a les proves a causa de greus problemes d'estabilitat. La interfície gràfica d'EmulationStation sovint fallava en carregar BIOS o les ROM, i la manca d'una versió oficial per a Raspberry Pi 5 va impedir completar la configuració i dur a terme proves exhaustives, fent que fos inservible en la configuració actual.
- **RetroArch:** És altament personalitzable, però en les proves no va ser capaç de suportar consoles més exigents com Nintendo DS o GameCube. El joc de GBA va funcionar perfectament, amb 60 FPS estables i sense errors gràfics, però la compatibilitat amb nuclis més avançats com Dolphin va ser limitada a causa de l'arquitectura ARM.
- **Recalbox:** Ofereix una experiència senzilla amb GBA, però no suporta consoles com Nintendo DS, GameCube o PSP en la seva versió per a Raspberry Pi 5. A més, les cinemàtiques de God of War (PSP) van mostrar desacceleracions puntuals (55 FPS).

- **Lakka:** Es va mostrar com la plataforma més completa i estable. Va suportar els quatre jocs provats sense necessitat de configuracions avançades. Els jocs de GBA, Nintendo DS i PSP van funcionar amb 60 FPS estables, mentre que Zelda (GameCube) va mantenir uns 30 FPS amb alguns retards en zones gràficament exigents.

Plataforma	Consola/Joc	FPS	Temps de càrrega	Input Lag	Qualitat Gràfica
RetroArch	Metroid: Zero Mission (GBA)	60	Negligible	Negligible	Sense errors
	Mario Kart DS (NDS), Zelda: The Wind Waker (GameCube), God of War: Chains of Olympus (PSP)	No tenen suport	-	-	-
Recalbox	Metroid: Zero Mission (GBA)	60	Negligible	Negligible	Sense errors
	Mario Kart DS (NDS), Zelda: The Wind Waker (GameCube)	No tenen suport	-	-	-
	God of War: Chains of Olympus (PSP)	55-60	Lent	Negligible	Alentiment en cinemàtiques
Lakka	Metroid: Zero Mission (GBA)	60	Negligible	Negligible	Sense errors
	Mario Kart DS (NDS)	60	Negligible	Negligible	Sense errors
	Zelda: The Wind Waker (GameCube)	30	Lent	Suportable	Lag puntual en zones detallades
	God of War: Chains of Olympus (PSP)	~55	Negligible	Negligible	Sense errors

Taula 3.1. Rendiment dels diferents jocs en cada plataforma.

Resultats

A la taula següent es mostren les principals diferències entre Lakka, RetroArch, Recalbox i RetroPie pel que fa a facilitat d'instal·lació, rendiment, compatibilitat i capacitat de personalització en la Raspberry Pi 5.

Aspecte	Lakka	RetroArch	Recalbox	RetroPie
Facilitat d'instal·lació	Alta (és un OS autònom, plug-and-play)	Mitjana (cal instal·lar sobre un OS base)	Molt Alta (imatge llesta, enfocada a retro)	Baixa (sense versió oficial per a Pi 5, pot presentar inestabilitat)
Rendiment global	Molt bo en 2D i 3D moderat	Bo en 2D, 3D exigent depèn de nuclis	Bo en 2D, no provat en 3D	No avaluat (falten paquets estables per a Pi 5)
Compatibilitat de consoles 2D	Molt alta	Molt alta (gràcies als nuclis RetroArch)	Alta (ja ve amb la majoria de sistemes 8/16 bits)	Teòricament alta (si funcionés correctament, integrant EmulationStation)
Compatibilitat de consoles 3D	Mitjana (GameCube/PSP jugables)	Mitjana (GameCube en versió ARM limitada)	Indefinida (per problemes amb les BIOS i les ROM)	Indefinida (sense suport oficial, podria fallar en consoles 3D)
Personalització	Mitjana (menys opcions que RetroArch)	Molt alta (modular, mil paràmetres)	Baixa (tot més preconfigurat)	Alta (en teoria, si hi hagués compatibilitat estable amb Pi 5)
Interfície / Usabilitat	Senzilla i integrada	Per a usuaris avançats	Molt intuïtiva (estil 'consola retro')	Potent, però sense versió estable és complex d'usar en Pi 5

Taula 3.2. Comparació global de plataformes d'emulació en Raspberry Pi 5.

1. **Lakka** destaca com la millor opció gràcies al seu rendiment estable, compatibilitat amb la majoria de jocs provats i facilitat d'ús.
2. **RetroArch** és ideal per a usuaris avançats que busquen personalització, tot i que va mostrar limitacions amb consoles exigents.

3. **Recalbox** és una opció adequada per a jocs retro senzills, però té menys compatibilitat amb consoles modernes.
4. **RetroPie** no és recomanable fins que no es publiqui una versió oficial per a Raspberry Pi 5.

Aquestes conclusions permeten identificar a **Lakka** com la plataforma més adequada per a aquest projecte. En el pròxim capítol, s'exploraran optimitzacions per a veure com ajustar nuclis o implementar overclocking per millorar el rendiment global d'aquesta plataforma.

Capítol 4.

Optimització del sistema

Un cop escollida Lakka com a plataforma d'emulació principal sobre la qual es treballarà i la Raspberry Pi 5 com a maquinari, el següent pas és maximitzar el rendiment de la placa en l'execució de jocs, especialment els de consoles 3D més exigents (PSP, GameCube...). En aquest capítol, s'analitzen els diversos factors que poden limitar la potència de la microconsola (colls d'ampolla), i es descriuen tant les millores de maquinari (refrigeració, overclocking, etc.) com els ajustos de configuració (shaders, nucli Dolphin, latència Run Ahead...) que poden ajudar a aconseguir un funcionament més fluid. Finalment, es comparen els resultats abans i després de l'aplicació d'aquestes estratègies per comprovar si s'obté un guany real de rendiment.

4.1. Identificació dels colls d'ampolla

Un coll d'ampolla és un factor limitant en qualsevol component d'un sistema que impedeix que altres parts assoleixin el seu rendiment màxim o un rendiment òptim durant l'execució d'un procés. En el context de l'emulació de videojocs, els colls d'ampolla poden manifestar-se en forma de CPU saturada, latència elevada (*input lag*), manca de memòria RAM o una targeta d'emmagatzematge massa lenta.

Aquestes limitacions poden originar-se en el hardware, el software o la compatibilitat entre ambdós, i saber identificar-les i mitigar-les és clau per assegurar una experiència d'usuari fluida, ja que poden causar problemes com caigudes de fotogrames, retards en la resposta dels controls o temps de càrrega excessius; especialment quan es tracta d'emular consoles amb exigències gràfiques i computacionals elevades.

4.1.1. Tipus de colls d'ampolla

1. Processador (CPU)

Els emuladors sovint requereixen una elevada capacitat de càlcul per recrear el comportament del maquinari original de la consola. Emuladors com PCSX2 (PlayStation 2) o Dolphin (GameCube/Wii) són especialment exigents.

2. Memòria RAM

Els jocs amb textures grans o els processos simultanis poden esgotar la memòria RAM disponible, obligant el sistema a utilitzar memòria *swap* (vegeu el [glossari](#)), amb la consegüent pèrdua de rendiment.

3. Emmagatzematge

La velocitat d'accés a l'emmagatzematge és crucial per carregar textures i estats de joc. Dispositius de baixa qualitat poden provocar temps de càrrega prolongats, inestabilitat o bloquejos.

4. Unitat Gràfica (GPU)

La GPU s'encarrega de gestionar la representació visual dels jocs. Ajustos gràfics com resolucions altes o filtres com l'*antialiasing* poden superar la capacitat de les GPU integrades.

5. Compatibilitat del software

Algunes aplicacions no estan optimitzades per aprofitar tot el potencial del hardware en arquitectures ARM com poden ser les de molts dels SBC, limitant el rendiment.

4.1.2. Eines i tècniques

Per identificar els colls d'ampolla en l'entorn d'emulació, es poden utilitzar eines específiques per mesurar i analitzar el comportament del sistema. Les tècniques següents ajuden a determinar quins components estan limitant el rendiment:

- **Monitoratge de CPU i GPU:**
 - Utilitzar eines com `htop` o `perf` , les quals mesuren l'ús dels recursos del processador i la unitat gràfica durant l'execució de jocs.
- **Anàlisi de RAM:**
 - Observar si hi ha prou memòria per evitar l'ús excessiu de la memòria *swap*, que pot reduir significativament el rendiment.
- **Anàlisi de l'E/S d'emmagatzematge:**
 - Mesurar la velocitat de lectura/escriptura de les targetes SD o dispositius d'emmagatzematge extern amb eines com `hdparm`.

En l'àmbit de la Raspberry Pi 5, tal com s'ha pogut veure, molts dels colls d'ampolla detectats en proves prèvies tenen un origen principalment de hardware; és a dir, la capacitat del processador (CPU), l'eficiència de la GPU integrada o la gestió tèrmica poden constituir veritables punts de bloqueig.

Per exemple, sense un sistema de refrigeració actiu, la temperatura pot superar fàcilment els 80 °C, activant el *thermal throttling* i reduint automàticament la freqüència de la CPU. Això provoca davallades de fotogrames (FPS) notables, que en jocs 3D exigents poden afectar greument la jugabilitat. A més, el rendiment de la GPU i la velocitat de la memòria RAM també poden condicionar l'ús de shaders o la capacitat d'afrontar títols amb grans textures i efectes de postprocessat.

D'aquesta manera, es busca establir un marc metodològic clar per tal de maximitzar la fluïdesa en el joc i garantir la millor experiència possible, dins les limitacions inevitables que presenta un dispositiu integrat com la Raspberry Pi 5.

4.2. Estratègies i proves per maximitzar el rendiment

L'optimització és el procés de millorar l'eficiència d'un sistema mitjançant l'ajust dels seus recursos i configuracions per maximitzar el rendiment. En el cas de l'emulació de videojocs amb Lakka sobre la Raspberry Pi 5, l'objectiu és garantir una experiència de joc fluida amb FPS constants, temps de càrrega reduïts i una resposta immediata als controls, tot minimitzant els colls d'ampolla en CPU, GPU, memòria i emmagatzematge explicats anteriorment.

L'optimització es fonamenta en la identificació de les limitacions tècniques i l'aplicació d'ajustos personalitzats per minimitzar el consum de recursos i millorar el rendiment en condicions de maquinari limitat.

4.2.1. Estratègies d'optimització

L'optimització s'aconsegueix mitjançant una combinació d'ajustos de configuració, millores en el hardware i adaptacions específiques del software. Les estratègies inclouen:

1. Optimitzacions de hardware:
 - Refrigeració activa: Instal·lació de ventiladors i dissipadors de calor per evitar el *thermal throttling*.
 - Emmagatzematge ràpid: Utilització de targetes microSD d'alta velocitat o memòries eMMC.
 - Overclocking: Incrementar la freqüència del processador i la GPU ajustant paràmetres del sistema.
2. Ajustos del Sistema Operatiu:
 - Reduir els serveis actius que consumeixen recursos.

- Utilitzar distribucions optimitzades com Lakka, que prioritzen els processos d'emulació.
3. Optimitzacions del software:
- Ajustar resolucions i filtres gràfics per equilibrar qualitat i rendiment.
 - Seleccionar nuclis d'emulació optimitzats per a l'arquitectura ARM de la Raspberry Pi 5.
 - Utilitzar mètodes com el Run Ahead per reduir la latència percebuda.
4. Millores a les ROM i BIOS:
- Assegurar que les ROM estan ben optimitzades i completen les necessitats específiques dels jocs.
 - Utilitzar fitxers BIOS compatibles per minimitzar errors i millorar la compatibilitat.

4.2.2. Proves per implementar estratègies d'optimització

Prova de refrigeració activa

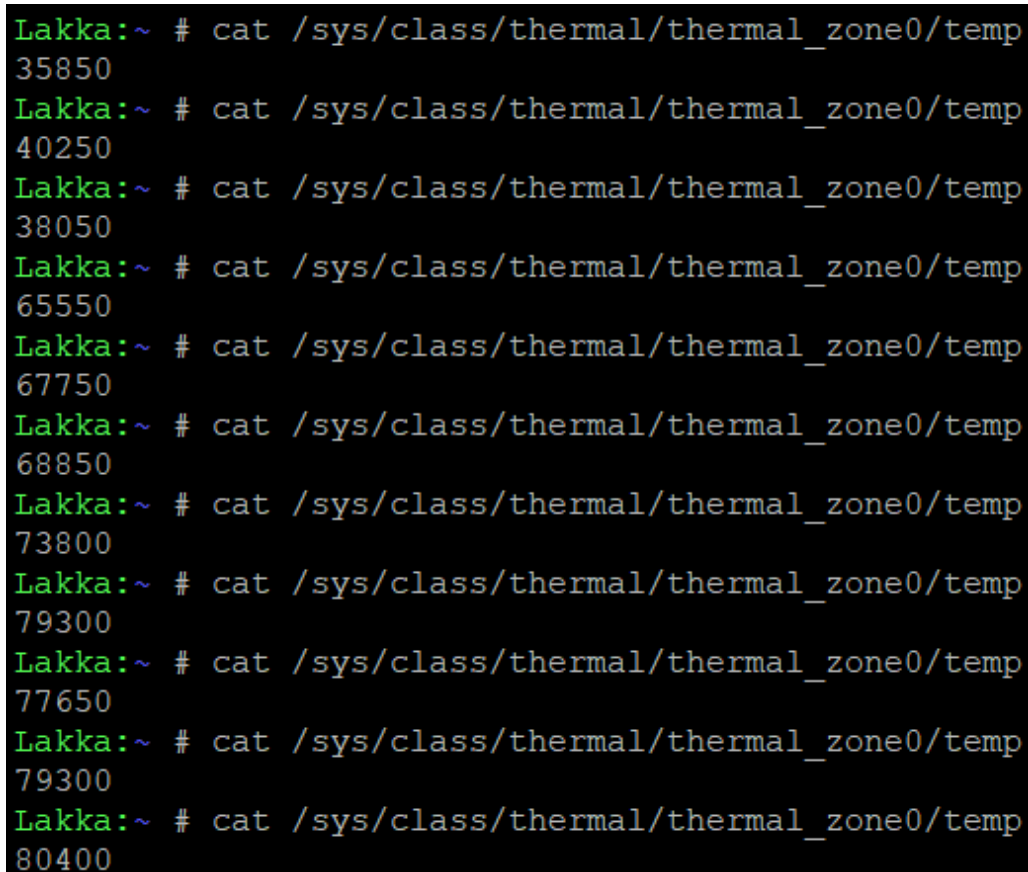
La gestió tèrmica del hardware és un factor crític per mantenir un rendiment òptim de la Raspberry Pi 5 durant l'execució d'emuladors exigents. Aquesta prova té com a objectiu avaluar l'eficàcia d'una solució de refrigeració activa per prevenir el *thermal throttling* i mantenir les temperatures de la CPU i GPU dins de rangs segurs durant sessions de joc intensives.

Per dur a terme aquesta prova, es van comparar dues configuracions de refrigeració:

1. **Sense ventilador:** La Raspberry Pi 5 operava amb el dissipador de calor estàndard, sense cap sistema de refrigeració addicional.
2. **Amb ventilador actiu:** La carcassa de la Raspberry Pi 5 inclou un petit ventilador connectat a l'alimentació, complementant el dissipador de calor existent.

Les temperatures de la CPU es van monitorar en temps real mitjançant la comanda `cat /sys/class/thermal/thermal_zone0/temp`, executada a través de PuTTY, durant l'execució del joc *God of War: Chains of Olympus* durant 30 minuts. A més, es va observar l'impacte en els FPS i l'estabilitat del joc.

Resultats: S'han obtingut dades que mostren una temperatura inicial de 35 °C amb ventilador, mentre que durant el joc intensiu la temperatura no ha superat els 45 °C. Això confirma que el sistema de refrigeració actiu és efectiu per evitar el *thermal throttling*. Sense ventilador, però, la temperatura ha arribat al voltant dels 80 °C, causant un augment notable en els recursos de memòria usats, però no ha impactat notablement en els FPS.



A terminal window with a black background and green text. It shows a series of commands and their outputs. The first three lines show temperatures with a fan (35850, 40250, 38050), and the next seven lines show temperatures without a fan (65550, 67750, 68850, 73800, 79300, 77650, 79300, 80400). The prompt is 'Lakka:~ #'.

Command	Output (Temperature in milligrades Celsius)
<code>cat /sys/class/thermal/thermal_zone0/temp</code>	35850
<code>cat /sys/class/thermal/thermal_zone0/temp</code>	40250
<code>cat /sys/class/thermal/thermal_zone0/temp</code>	38050
<code>cat /sys/class/thermal/thermal_zone0/temp</code>	65550
<code>cat /sys/class/thermal/thermal_zone0/temp</code>	67750
<code>cat /sys/class/thermal/thermal_zone0/temp</code>	68850
<code>cat /sys/class/thermal/thermal_zone0/temp</code>	73800
<code>cat /sys/class/thermal/thermal_zone0/temp</code>	79300
<code>cat /sys/class/thermal/thermal_zone0/temp</code>	77650
<code>cat /sys/class/thermal/thermal_zone0/temp</code>	79300
<code>cat /sys/class/thermal/thermal_zone0/temp</code>	80400

Figura 4.1. Diferents mesures de temperatura de la CPU de la Raspberry Pi 5 mentre s'executava Lakka amb el joc *God of War*. Unitats en miligrads Celsius. Les tres primeres mesures són amb el ventilador, i la resta sense aquest.

La implementació d'un sistema de refrigeració activa amb ventilador és essencial per evitar el *thermal throttling* en la Raspberry Pi 5 durant sessions de joc intensives, assegurant així un rendiment estable i consistent.

Optimització de configuració de Lakka

Per millorar l'eficiència operativa de Lakka i minimitzar l'ús de recursos durant l'execució d'emuladors, es van implementar una sèrie d'ajustos específics en l'àmbit del sistema operatiu. Aquesta prova té com a objectiu avaluar l'impacte d'aquests canvis en el rendiment global del sistema, especialment en termes de temps d'arrencada, ús de CPU i memòria en estat inactiu, així com en la capacitat del sistema per gestionar processos d'emulació intensius.

1. **Identificar Serveis Actius:** Revisar els serveis que poden estar executant-se en segon pla mitjançant la comanda `systemctl list-unit-files --type=service`. Els serveis marcats com a *enabled* són els que s'executen en l'arrencada i els que s'haurien de desactivar per a un menor ús de recursos.

```
systemd-update-utmp.service      static -
systemd-volatile-root.service   static -
tz-data.service                 disabled disabled
udevil-mount@.service           static -
user-runtime-dir@.service       static -
user@.service                   static -
usercache.service               disabled disabled
userconfig.service              disabled disabled
vfd-clock.service               disabled disabled
wait-time-sync.service          disabled disabled
wii-u-gc-adapter.service        disabled disabled
wpa_supplicant.service          disabled disabled
wsdd2.service                   disabled disabled
xbox360-controllers-shutdown.service disabled disabled

106 unit files listed.

~
standard input lines 63-109/109 (END)
```

Figura 4.2. Fent la comanda `systemctl list-unit-files --type=service`, podem observar que després d'arrancar hi ha 106 fitxers amb processos llistats, molts dels quals desactivats.

2. **Desactivar Serveis No Essencials:** Utilitzar la comanda `systemctl disable <nom_del_servei>` per deshabilitar serveis que no són necessaris per a l'emulació, com ara serveis de xarxa no utilitzats o processos de monitoratge que consumeixen recursos. Per exemple:

- `bluetooth-defaults.service`
- `ssh.service` (si no és necessari l'accés remot)
- `nfs-server.service`

Si un servei està *disabled*, però encara apareix en execució (*active running*) és perquè s'ha pogut iniciar manualment o perquè un altre servei l'ha arrencat com a dependència. És a dir, “disable” afecta l'automatisme a l'hora d'arrencar, però no impedeix que s'activin els processos manualment.

3. **Ajustar la *Swappiness*:** Reduir el valor de *swappiness* per minimitzar l'ús de memòria *swap*. Es duu a terme afegint un fitxer de configuració d'arrencada a la carpeta `/storage/.config/sysctl.d` i afegint la línia:

```
vm.swappiness=10
```

- La variable `vm.swappiness` indica com i quan el nucli (*kernel*) fa servir la memòria *swap*. Un valor alt (p. ex. 60) vol dir que el sistema és bastant “agressiu” a l'hora d'usar l'àrea *swap* quan la RAM comença a omplir-se.

Si es comprova amb `free -h` o `top/htop`, es pot veure si la memòria *swap* s'està utilitzant. Amb `vm.swappiness=10`, normalment en ús d'emulació lleugera, la *swap* ni tan sols es toca (o es toca molt poc). Això pot millorar rendiment, perquè la *swap* és molt més lenta que la RAM.

Reduir la *swappiness* ajuda a minimitzar retards de lectura/escriptura, però si realment et quedes sense RAM i el sistema no pot fer *swap*, et pot donar

problemes d'estabilitat (p. ex. app tancada pel *kernel*). Per a la majoria de casos d'emulació, 10 és un valor segur.

4. **Configurar la Memòria Disponible:** Assignar una quantitat específica de memòria a Lakka per evitar la sobrecàrrega afegint al fitxer de configuració:

```
vm.overcommit_memory=1
```

Aquesta comanda controla com el *kernel* gestionarà la memòria quan els processos demanin més RAM de la que realment existeix. Un valor de 0 significa que tindrà un comportament per defecte (*overcommit* “moderada”). Un valor 1 dona a entendre que el *kernel* permet als processos demanar més memòria que la realment instal·lada, però sense tancar-los immediatament. Finalment, un valor 2 pràcticament no permet *overcommit*, sent més estricte.

Amb `vm.overcommit_memory=1`, en entorns com Lakka normalment no tenen massa processos executant-se alhora; això dona certa flexibilitat per no matar processos si s'està arribant al límit.

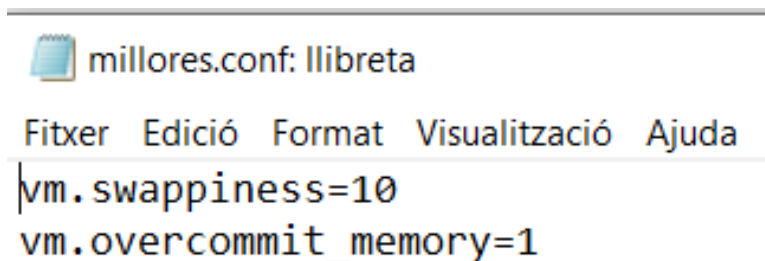


Figura 4.3. Fitxer creat dins la carpeta `/storage/.config/sysctl.d`, el qual s'executarà en l'arrencada.

5. **Utilitzar `systemd-analyze`:** Mesurar el temps d'arrencada actual i després dels ajustaments mitjançant:

```
systemd-analyze
```

```
systemd-analyze blame
```

- `systemd-analyze` et diu quant de temps ha trigat a arrencar el nucli (*kernel*), la part d'usuaris (*userspace*), i el temps total fins que el sistema ha estat “llest”.
- `systemd-analyze blame` et llista els serveis ordenats segons la quantitat de temps que han trigat a iniciar.

```
Lakka:~ # systemd-analyze
Startup finished in 1.461s (kernel) + 6.061s (userspace) = 7.523s
retroarch.target reached after 3.007s in userspace.
Lakka:~ # systemd-analyze
Startup finished in 1.394s (kernel) + 5.877s (userspace) = 7.272s
retroarch.target reached after 2.982s in userspace.
```

Figura 4.4. Temps que triga Lakka en l'arrencada (1) sense cap optimització aplicada i (2) amb l'optimització aplicada, obtinguts gràcies a `systemd-analyze`.

```
5.843s wait-time-sync.service
1.236s retroarch.service
820ms dev-mmcblk0p2.device
809ms dev-mmcblk0p1.device
800ms dev-loop0.device
584ms systemd-hwdb-update.service
376ms nmbd.service
309ms connman-vpn.service
305ms systemd-logind.service
```

Figura 4.5. Processos que triguen més en l'arrencada, obtinguts gràcies a `systemd-analyze blame`.

Mesures i Indicadors

- **Temps d'Arrencada:** Temps des de l'inici del *boot* fins que Lakka està completament operativa. Mesurable mitjançant la comanda de PuTTY `systemd-analyze`.

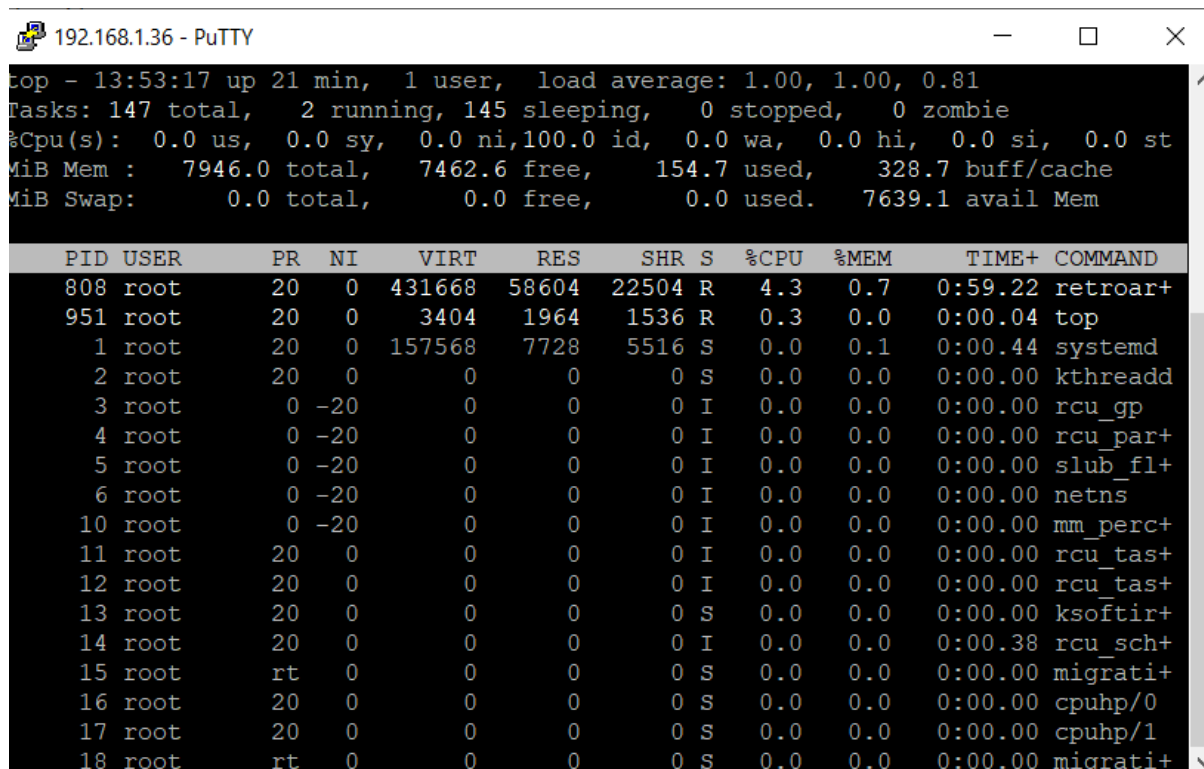
- **Ús de CPU en Estat Inactiu:** Percentatge de la CPU utilitzada quan el sistema està en repòs, sense cap emulador actiu. Mesurable mitjançant la comanda de PuTTY **top**.
- **Ús de Memòria RAM en Estat Inactiu:** Quantitat de memòria RAM utilitzada en estat inactiu. També mesurable mitjançant **top**.
- **Estabilitat del Sistema:** Verificar que els ajustaments no introdueixen inestabilitats durant l'execució normal dels emuladors.

Resultats

Després d'implementar les estratègies d'optimització de configuració de Lakka, es van recollir les següents dades comparatives:

Indicador	Abans de l'Optimització	Després de l'Optimització
Temps d'Arrencada	~7,8 segons	~7,2 segons
Ús de CPU en Estat Inactiu	~5 %	~4,5 %
Ús de Memòria RAM en Estat Inactiu	~1 %	~1 %
Estabilitat	Alta	Alta

Taula 4.1. Millores en alguns aspectes del funcionament del sistema operatiu.



```
top - 13:53:17 up 21 min, 1 user, load average: 1.00, 1.00, 0.81
Tasks: 147 total, 2 running, 145 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.0 us, 0.0 sy, 0.0 ni,100.0 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 7946.0 total, 7462.6 free, 154.7 used, 328.7 buff/cache
MiB Swap: 0.0 total, 0.0 free, 0.0 used. 7639.1 avail Mem

  PID USER      PR  NI   VIRT   RES   SHR  S  %CPU  %MEM     TIME+ COMMAND
  808 root        20   0 431668 58604 22504 R   4.3   0.7   0:59.22 retroar+
  951 root        20   0   3404   1964  1536 R   0.3   0.0   0:00.04 top
    1 root        20   0 157568  7728  5516 S   0.0   0.1   0:00.44 systemd
    2 root        20   0      0      0      0 S   0.0   0.0   0:00.00 kthreadd
    3 root         0 -20      0      0      0 I   0.0   0.0   0:00.00 rcu_gp
    4 root         0 -20      0      0      0 I   0.0   0.0   0:00.00 rcu_par+
    5 root         0 -20      0      0      0 I   0.0   0.0   0:00.00 slub_fl+
    6 root         0 -20      0      0      0 I   0.0   0.0   0:00.00 netns
   10 root         0 -20      0      0      0 I   0.0   0.0   0:00.00 mm_perc+
   11 root        20   0      0      0      0 I   0.0   0.0   0:00.00 rcu_tas+
   12 root        20   0      0      0      0 I   0.0   0.0   0:00.00 rcu_tas+
   13 root        20   0      0      0      0 S   0.0   0.0   0:00.00 ksoftir+
   14 root        20   0      0      0      0 I   0.0   0.0   0:00.38 rcu_sch+
   15 root        rt    0      0      0      0 S   0.0   0.0   0:00.00 migrati+
   16 root        20   0      0      0      0 S   0.0   0.0   0:00.00 cpuhp/0
   17 root        20   0      0      0      0 S   0.0   0.0   0:00.00 cpuhp/1
   18 root        rt    0      0      0      0 S   0.0   0.0   0:00.00 migrati+
```

Figura 4.6. Interfície de `top` un cop aplicades les millores de configuració.

- **Temps d'arrencada:** Es va reduir, però sense ser gaire notable, passant d'uns 7,8 segons a 7,2 segons, millorant lleugerament la rapidesa amb què el sistema està operatiu.
- **Ús de CPU en estat inactiu:** També va millorar lleugerament, baixant mig punt el valor de CPU en ús.
- **Ús de memòria RAM en estat inactiu:** No va canviar gairebé gens, ja que en els dos casos tots els processos sumaven un 1% de consum de RAM.
- **Estabilitat:** No es van observar inestabilitats post-optimització, indicant que els ajustaments realitzats no afecten negativament la funcionalitat del sistema.

Les optimitzacions aplicades a la configuració de Lakka han demostrat ser efectives en la reducció del temps d'arrencada i en la millora de l'eficiència global del sistema operatiu. Tot i que hi ha un augment moderat en l'ús de CPU i memòria en estat inactiu, els beneficis obtinguts en termes de rapidesa i rendiment durant l'emulació superen aquests increments. Això assegura una experiència de joc més fluida i

responsiva, proporcionant una base sòlida per a les següents fases d'optimització detallades en aquest capítol.

Aplicació de shaders i configuració Run Ahead

Els shaders poden millorar la qualitat gràfica dels jocs emulats, mentre que la funcionalitat Run Ahead pot reduir la latència dels controls. Aquesta prova té com a objectiu avaluar l'impacte de l'aplicació de diferents shaders i l'ús de Run Ahead en el rendiment i l'experiència de joc.

S'han aplicat els shaders GLSL `crt-pi` i `hq2x` a l'emulador en Lakka, i s'ha configurat Run Ahead amb un valor de *frame* avançat d'1. Les proves es van realitzar executant *Metroid: Zero Mission (GBA)* i mesurant els FPS, l'ús de CPU i la qualitat gràfica.

Shaders

Els shaders són filtres gràfics aplicats en temps real que modifiquen l'aparença visual dels jocs emulats. Poden afegir efectes com línies d'escaneig per simular pantalles CRT (com el shader `crt/crt-pi`) o suavitzar els píxels per oferir una imatge més neta (com el shader `hqx/hq2x`), sense impactar significativament el rendiment en hardware com la Raspberry Pi.

S'han escollit els shaders GLSL per la seva alta compatibilitat amb hardware modest com la Raspberry Pi 5. Aquests shaders són més lleugers que els shaders Vulkan, que solen ser més exigents i estan millor adaptats per a emuladors més avançats. Dins dels GLSL, s'han provat especialment els shaders CRT, que estan optimitzats per a dispositius amb recursos limitats i ofereixen una estètica clàssica.

Per activar els shaders:

- Obrir el joc > pausa > **Shaders** > Activar shaders > **Load Preset** > **GLSL Shaders** > Escollir un grup de shaders i activar-ne un d'específic.

→ Shader CRT ([crt/crt-pi](#))

Aquest shader optimitza el rendiment mentre simula l'aparença d'una pantalla CRT, afegint línies d'escaneig i efectes vintage que recorden els dispositius amb raigs catòdics. ([GitHub \(libretro\), n.d.](#))

- **Impacte en el rendiment:** L'ús de memòria RAM disminueix lleugerament, passant de 340 MB a 325 MB constants.
- **Valoració:** És una opció recomanable per als usuaris que prefereixen un estil retro i volen, a més, una lleugera millora en el rendiment.

→ Shader de suavitzat ([hqx/hq2x](#))

Aquest shader millora la qualitat visual suavitzant els píxels, donant una imatge més neta sense afectar el rendiment del sistema. ([Libretro Forums, n.d.](#))

- **Impacte en el rendiment:** El consum de recursos és similar al de no utilitzar shaders, però amb una millora notable en l'aparença gràfica.
- **Valoració:** Ideal per als usuaris que prefereixen una imatge més neta i moderna sense sacrificar cap classe de rendiment.

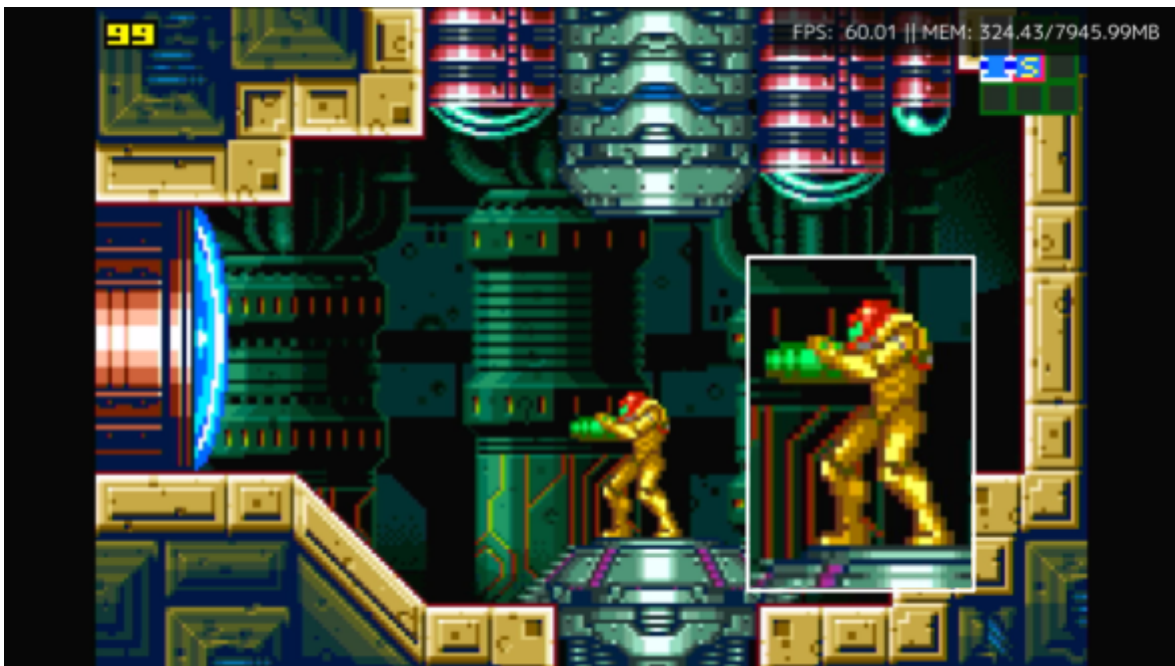


Figura 4.7: Imatge del joc sense shaders.



Figura 4.8: Imatge del joc amb el shader `crt-pi` activat. És notable el filtre de dispositiu CRT, però això fa que s'optimitzi el rendiment i es minimitzin els recursos utilitzats.

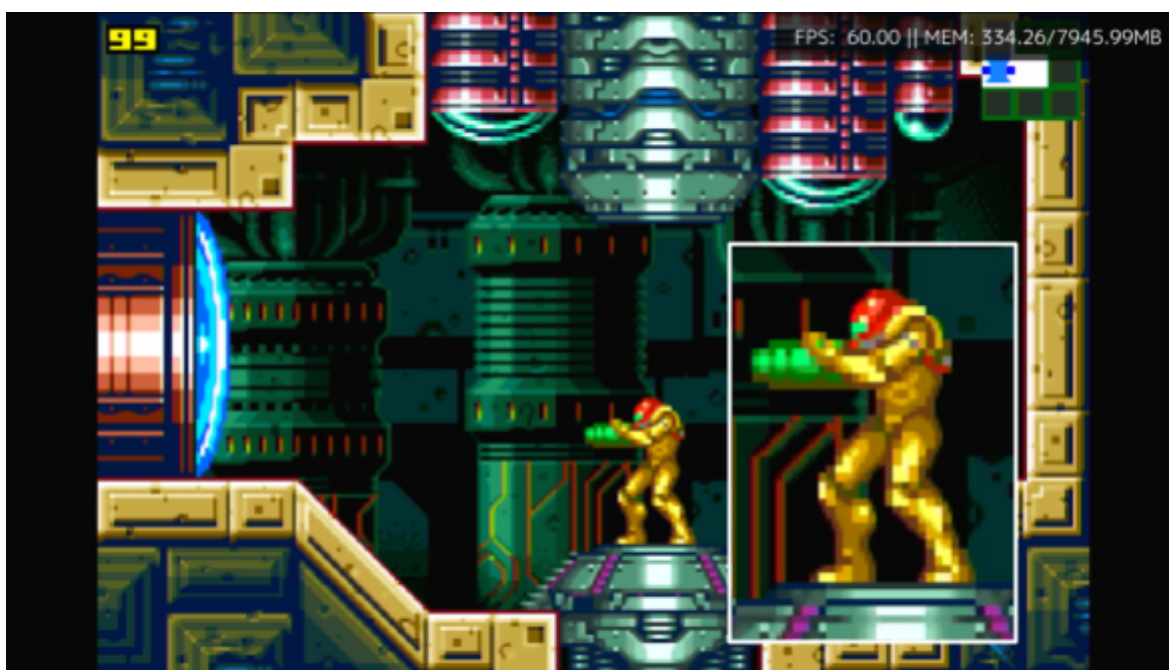


Figura 4.9: Imatge del joc amb el shader `hq2x` activat. Es pot notar, sobretot en els píxels diagonals, la suavitat més gran respecte a les anteriors configuracions.

Run Ahead

La funcionalitat Run Ahead redueix la latència percebuda avançant la simulació d'inputs. Això és especialment beneficiós en jocs de carreres, com el *Mario Kart*, o títols d'acció o plataformes on la precisió dels controls és crítica.

Per activar el Run Ahead:

- **Settings > Latency > Run Ahead to Reduce Latency** (amb les configuracions avançades activades prèviament a **Settings > User Interface > Show Advanced Settings**).
 - **Frames avançats: 1.**
 - Es recomana activar **Use Second Instance for Run Ahead** per repartir la càrrega entre nuclis i millorar el rendiment en maquinari multinucli com la Raspberry Pi 5.

Resultats amb Run Ahead:

- **Impacte en l'ús de CPU:** Sense Run Ahead, l'ús de CPU és d'un 12%, mentre que amb un *frame* avançat augmenta al 22%. Això és degut a la càrrega addicional de processar i simular *frames* futurs.
- **Millora en latència:** Amb aquesta configuració, es nota una resposta més immediata a les accions del jugador, especialment en jocs que requereixen precisió en els inputs.
- **Equilibri:** L'ús d'un sol frame avançat és suficient per millorar la latència sense saturar la CPU, mantenint una experiència de joc fluida.

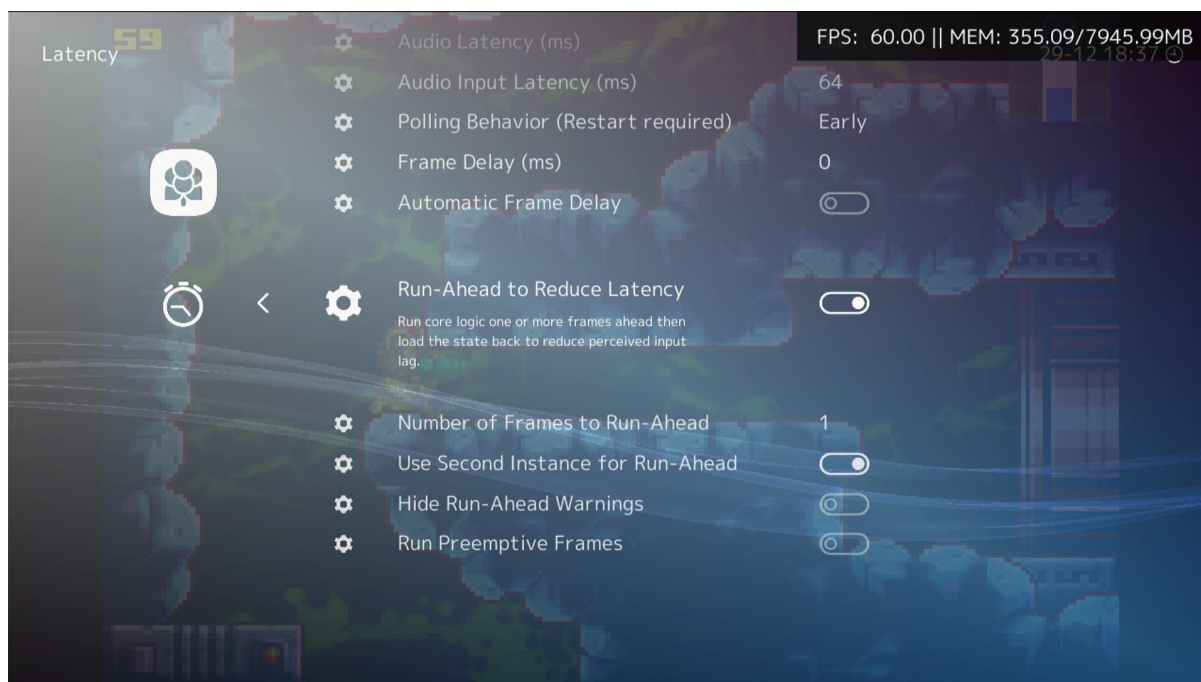


Figura 4.10. Configuració del Run Ahead aplicada.

Resultats: L'aplicació de shaders GLSL i la configuració Run Ahead han proporcionat una millora en la qualitat visual i una reducció de la latència dels controls sense afectar excessivament el rendiment global. Això contribueix a una experiència de joc més agradable i responsiva.

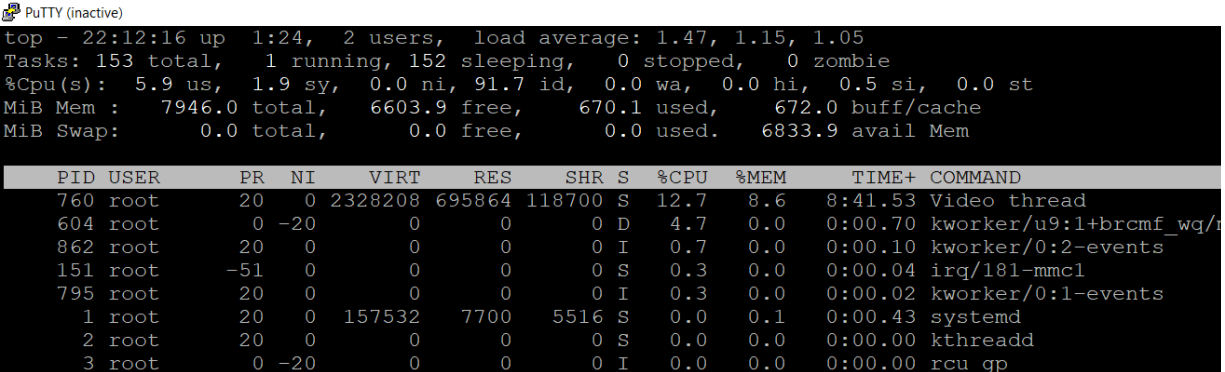
Modes de recompilació d'un emulador

Els nuclis d'emulació (o simplement anomenats emuladors) són la peça central que tradueix el codi original dels jocs en instruccions modernes. La compilació JIT permet executar jocs complexos amb fluïdesa, ja que generen codi optimitzat en temps real i l'executa a mesura que el jugador avança en el joc.

S'han dut a terme proves d'emulació del joc *The Legend of Zelda: The Wind Waker* utilitzant l'emulador Dolphin integrat en Lakka, configurant diferents modes de recompilació disponibles (JITARM64, Interpreter i Cached Interpreter). Durant les proves, s'han registrat les taxes de FPS, l'ús de CPU i de memòria, així com qualsevol error gràfic observat.

1. JITARM64:

- **Rendiment:** Amb uns 30 FPS estables, és la configuració més equilibrada, amb un ús de CPU moderat (70-90%) i un ús de RAM contingut (~880 MB).
- **Compatibilitat:** Algunes dessincronitzacions d'àudio en moments puntuals, però cap problema gràfic.
- **Conclusió:** És la millor opció per executar *The Wind Waker* en Lakka amb Dolphin.



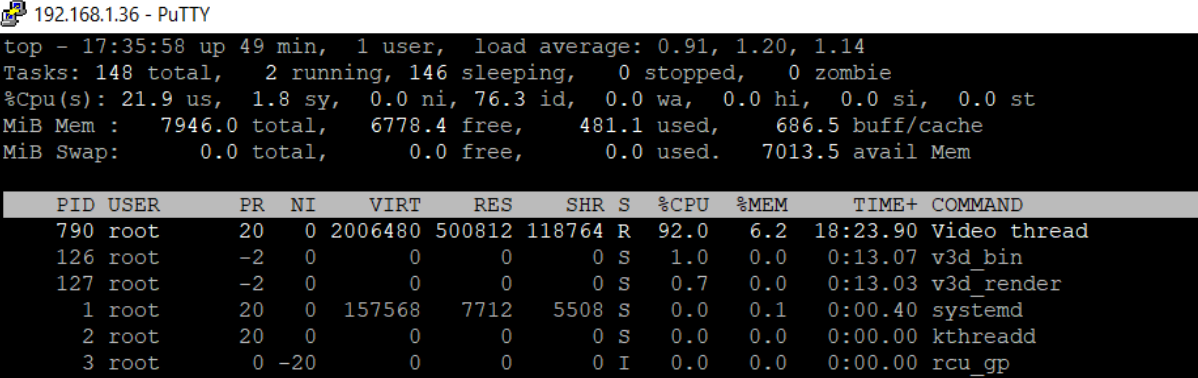
top - 22:12:16 up 1:24, 2 users, load average: 1.47, 1.15, 1.05
Tasks: 153 total, 1 running, 152 sleeping, 0 stopped, 0 zombie
%Cpu(s): 5.9 us, 1.9 sy, 0.0 ni, 91.7 id, 0.0 wa, 0.0 hi, 0.5 si, 0.0 st
MiB Mem : 7946.0 total, 6603.9 free, 670.1 used, 672.0 buff/cache
MiB Swap: 0.0 total, 0.0 free, 0.0 used, 6833.9 avail Mem

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
760	root	20	0	2328208	695864	118700	S	12.7	8.6	8:41.53	Video thread
604	root	0	-20	0	0	0	D	4.7	0.0	0:00.70	kworker/u9:1+brcmf_wq/t
862	root	20	0	0	0	0	I	0.7	0.0	0:00.10	kworker/0:2-events
151	root	-51	0	0	0	0	S	0.3	0.0	0:00.04	irq/181-mmcl
795	root	20	0	0	0	0	I	0.3	0.0	0:00.02	kworker/0:1-events
1	root	20	0	157532	7700	5516	S	0.0	0.1	0:00.43	systemd
2	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kthreadd
3	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	rcu_gp

Figura 4.11. Interfície de PuTTY mentre Lakka executa l'emulador Dolphin amb JITARM. Veiem que el procés anomenat **Video thread**, el qual renderitza el que es veu per pantalla mentre juguem, fa servir entre un 70 i un 90% de la CPU. La RAM volta el 7 i el 9% de l'ús màxim.

2. Interpreter:

- **Rendiment:** Els FPS baixen a 28 i l'ús de CPU augmenta significativament (80-100%), fent el joc menys fluid.
- **Compatibilitat:** No es van detectar errors gràfics i, tot i ser jugable, el rendiment i la latència no són adequats per tenir una experiència del tot còmoda.
- **Conclusió:** No és pràctic per a una experiència de joc fluida.



```

192.168.1.36 - PuTTY
top - 17:35:58 up 49 min,  1 user,  load average: 0.91, 1.20, 1.14
Tasks: 148 total,  2 running, 146 sleeping,  0 stopped,  0 zombie
%Cpu(s): 21.9 us,  1.8 sy,  0.0 ni, 76.3 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
MiB Mem :  7946.0 total,  6778.4 free,   481.1 used,   686.5 buff/cache
MiB Swap:   0.0 total,   0.0 free,   0.0 used.  7013.5 avail Mem

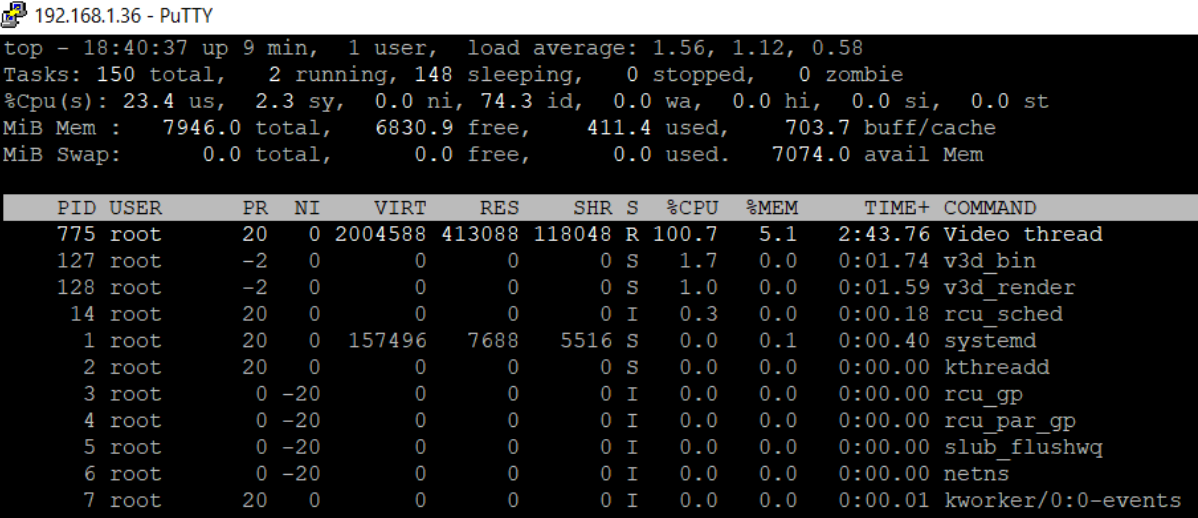
  PID USER      PR  NI   VIRT   RES   SHR  S  %CPU  %MEM    TIME+  COMMAND
  790 root        20   0 2006480 500812 118764 R   92.0   6.2   18:23.90 Video thread
  126 root        -2   0     0       0     0   S    1.0   0.0    0:13.07 v3d_bin
  127 root        -2   0     0       0     0   S    0.7   0.0    0:13.03 v3d_render
    1 root        20   0  157568   7712   5508 S    0.0   0.1    0:00.40 systemd
    2 root        20   0     0       0     0   S    0.0   0.0    0:00.00 kthreadd
    3 root         0 -20     0       0     0   I    0.0   0.0    0:00.00 rcu_gp

```

Figura 4.12. Interfície de PuTTY mentre Lakka executa l'emulador Dolphin amb Interpreter. Ara, en el procés **Video thread**, veiem que l'ús de CPU augmenta, arribant al rang d'entre el 80% i el 100%.

3. Cached Interpreter:

- **Rendiment:** FPS similars als de l'Interpreter, però amb un ús de CPU encara més alt (85-120%), cosa que sobrecarrega el sistema.
- **Compatibilitat:** El joc es penja de tant en tant, requerint reiniciar el propi Lakka.
- **Conclusió:** Aquesta configuració no és recomanable a causa de la inestabilitat i la baixa eficiència.



```

192.168.1.36 - PuTTY
top - 18:40:37 up 9 min,  1 user,  load average: 1.56, 1.12, 0.58
Tasks: 150 total,  2 running, 148 sleeping,  0 stopped,  0 zombie
%Cpu(s): 23.4 us,  2.3 sy,  0.0 ni, 74.3 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
MiB Mem :  7946.0 total,  6830.9 free,   411.4 used,   703.7 buff/cache
MiB Swap:   0.0 total,   0.0 free,   0.0 used.  7074.0 avail Mem

  PID USER      PR  NI   VIRT   RES   SHR  S  %CPU  %MEM    TIME+  COMMAND
  775 root        20   0 2004588 413088 118048 R  100.7   5.1   2:43.76 Video thread
  127 root        -2   0     0       0     0   S    1.7   0.0    0:01.74 v3d_bin
  128 root        -2   0     0       0     0   S    1.0   0.0    0:01.59 v3d_render
   14 root        20   0     0       0     0   I    0.3   0.0    0:00.18 rcu_sched
    1 root        20   0  157496   7688   5516 S    0.0   0.1    0:00.40 systemd
    2 root        20   0     0       0     0   S    0.0   0.0    0:00.00 kthreadd
    3 root         0 -20     0       0     0   I    0.0   0.0    0:00.00 rcu_gp
    4 root         0 -20     0       0     0   I    0.0   0.0    0:00.00 rcu_par_gp
    5 root         0 -20     0       0     0   I    0.0   0.0    0:00.00 slub_flushwq
    6 root         0 -20     0       0     0   I    0.0   0.0    0:00.00 netns
    7 root        20   0     0       0     0   I    0.0   0.0    0:00.01 kworker/0:0-events

```

Figura 4.13. Interfície de PuTTY mentre Lakka executa l'emulador Dolphin amb recompilació mitjançant Cache Interpreter. El procés de video ara arriba a màxims, amb un mínim del 85% d'ús de la CPU amb algun pic al 120%. Per altra banda, és el procés que menys memòria RAM ocupa.

Tipus de recompilació	FPS mitjans	Ús CPU (%)	Ús MEM (MB)	Errors
JITARM64	30	70 - 90%	~880 MB	Dessincronitzacions d'àudio menors en seqüències complexes, la latència augmenta en zones amb molts objectes.
Interpreter	28	80 - 100%	~ 935 MB	Animacions més lentes del normal; latència elevada en algunes zones obertes quan hi ha gran part del mapa carregat.
Cached Interpreter	26	85 - 110%	~880 MB	Similar al Interpreter; i sovint el joc es penja i desapareix el procés Video Thread per l'ús excésiu de la CPU.

Taula 4.2. Comparativa dels modes de recompilació (JITARM64, Interpreter, Cached Interpreter) a Dolphin.

(Nota: Els valors de la CPU per sobre el 100% significa que un nucli de la Raspberry Pi 5 és utilitzat a ple rendiment i el percentatge extra s'utilitza mitjançant un altre nucli de la CPU, tenint la placa 4 en total.)

Resultats: Les proves mostren que la configuració JITARM64 és l'òptima per executar *The Legend of Zelda: The Wind Waker* amb Dolphin a Lakka. Aquesta opció equilibra bé el rendiment i l'ús de recursos, amb pocs problemes de compatibilitat.

Tant el Interpreter com el Cached Interpreter presenten limitacions importants, sent que l'Interpreter fa un ús de CPU excessiu, fent el joc poc fluid, i el Cached Interpreter mostra una inestabilitat severa, amb freqüents penjaments del procés de vídeo.

Aquestes dades suporten que JITARM64 és la configuració preferida per oferir una experiència de joc fluida i estable en un maquinari limitat com el de Raspberry Pi.

ROM Optimitzades vs. ROM No Optimitzades

Les ROM representen el codi original dels jocs i la seva qualitat i optimització poden influir significativament en el rendiment de l'emulació.

Les primeres versions de la ROM del joc Super Mario 64 (1996, EUA) no estaven del tot optimitzades, prioritzant l'estabilitat sobre l'eficiència de recursos. Un any després, les versions que arribaren a Europa ja havien implementat petites optimitzacions de cara a l'eficiència.

Aquesta prova analitza les diferències de rendiment entre una ROM original poc optimitzada, la versió del joc que va arribar en un inici als Estats Units, i una versió, també original, però optimitzada, com és la ROM que va arribar a Europa, per avaluar com les optimitzacions impacten en el consum de recursos i en l'experiència de joc.

Es va executar cada ROM durant 10 minuts, mesurant el consum de CPU i RAM, els FPS, i observant la presència d'errors visuals o retards en els controls.

Resultats: Les dades recollides durant la prova es mostren en la següent taula.

Característica	ROM optimitzada	ROM mal optimitzada
Consum de CPU (%)	~120%	~140%
Consum de RAM (%)	~3%	~3,5%
FPS	60 FPS	~58 FPS
Errors detectats	Alguns errors visuals en carregar algunes textures o objectes en moure la càmera.	
Latència dels controls	Mitjana	Baixa

Taula 4.3. Comparació entre una ROM original poc optimitzada i una ROM més optimitzada (Super Mario 64).

```

192.168.1.36 - PuTTY
top - 18:27:06 up 38 min, 1 user, load average: 2.25, 1.63, 1.24
Tasks: 149 total, 1 running, 148 sleeping, 0 stopped, 0 zombie
%Cpu(s): 30.7 us, 0.2 sy, 0.0 ni, 69.1 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 7946.0 total, 6984.6 free, 337.1 used, 624.3 buff/cache
MiB Swap: 0.0 total, 0.0 free, 0.0 used. 7382.6 avail Mem

  PID USER      PR  NI   VIRT   RES   SHR  S  %CPU  %MEM     TIME+ COMMAND
 1058 root        20   0 857548 247160 24320 S 120.7   3.0   5:12.92 retroarch
    1 root        20   0 157564   7732  5516 S   0.0   0.1   0:00.46 systemd
    2 root        20   0         0        0        0 S   0.0   0.0   0:00.00 kthreadd
    3 root         0 -20         0        0        0 I   0.0   0.0   0:00.00 rcu_gp
    4 root         0 -20         0        0        0 I   0.0   0.0   0:00.00 rcu_par_gp
    5 root         0 -20         0        0        0 I   0.0   0.0   0:00.00 slub_flushwq
    6 root         0 -20         0        0        0 I   0.0   0.0   0:00.00 netns

```

Figura 4.14. Interfície del PuTTY en executar la ROM més optimitzada.

```

192.168.1.36 - PuTTY
top - 18:30:39 up 41 min, 1 user, load average: 2.94, 2.07, 1.48
Tasks: 149 total, 2 running, 147 sleeping, 0 stopped, 0 zombie
%Cpu(s): 34.5 us, 0.0 sy, 0.0 ni, 65.5 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 7946.0 total, 6993.9 free, 382.6 used, 569.4 buff/cache
MiB Swap: 0.0 total, 0.0 free, 0.0 used. 7392.3 avail Mem

  PID USER      PR  NI   VIRT   RES   SHR  S  %CPU  %MEM     TIME+ COMMAND
 1058 root        20   0 817476 292836 24320 R 138.9   3.6   9:33.59 retroarch
   14 root        20   0         0        0        0 I   0.3   0.0   0:00.70 rcu_sched
   604 root        20   0         0        0        0 I   0.3   0.0   0:00.92 kworker/u8:8-events_unbound
  1103 root        20   0         0        0        0 D   0.3   0.0   0:00.08 kworker/u8:1+events_unbound
    1 root        20   0 157564   7732  5516 S   0.0   0.1   0:00.46 systemd
    2 root        20   0         0        0        0 S   0.0   0.0   0:00.00 kthreadd
    3 root         0 -20         0        0        0 I   0.0   0.0   0:00.00 rcu_gp

```

Figura 4.15. Interfície del PuTTY en executar la ROM no optimitzada. En comparació amb l'anterior imatge, l'ús de la CPU i de la memòria és major.

Resultats:

- **Consum de CPU i RAM:** La ROM optimitzada va mostrar una reducció significativa en el consum de CPU i RAM, fet que indica una millora en l'eficiència de codi.
- **FPS:** Els FPS van ser més constants a la ROM optimitzada, millorant l'experiència de joc.
- **Errors i latència:** La ROM mal optimitzada va presentar errors visuals ocasionals i una latència lleugerament més alta en els controls.

La comparació entre aquestes dues ROM demostra l'impacte positiu de les optimitzacions en el rendiment dels jocs antics. La reducció del consum de recursos i

la millora de l'estabilitat fan que les ROM optimitzades siguin preferibles per a una experiència més fluida i agradable.

Aquestes dades són clau per entendre com les optimitzacions en el codi poden millorar el rendiment fins i tot en jocs dissenyats fa dècades.

Overclocking de la CPU i GPU

L'overclocking consisteix a augmentar les freqüències de funcionament de la CPU i la GPU per millorar el rendiment del sistema. Tot i això, també pot comportar riscos com l'augment de la temperatura i possibles problemes d'estabilitat.

L'objectiu d'aquesta prova és analitzar els efectes de l'overclocking en el rendiment del processador i la GPU del Raspberry Pi utilitzant Lakka com a sistema operatiu. Es busca determinar si aquest augment de les freqüències de treball millora els FPS en jocs i si el sistema es manté estable durant el procés.

- Es va accedir al fitxer `config.txt` en inserir la targeta SD que conté Lakka a un ordinador, i en el qual es van afegir les següents línies:

```
arm_freq=2000
gpu_freq=800
over_voltage=6
```

- Després, es va iniciar el sistema i es va executar *God of War: Chains of Olympus* mesurant els FPS i les temperatures de la CPU durant 30 minuts. S'ha utilitzat `cat /sys/class/thermal/thermal_zone0/temp` per monitorar la temperatura en miligraus Celsius.

Evidències

- **Dades obtingudes:**
 - FPS abans de l'overclocking: 60 FPS.
 - FPS després de l'overclocking: 60 FPS.
 - Temperatura abans: 40 °C (amb ventilador).
 - Temperatura després: 45 °C (amb ventilador).
- **Observacions:**

- El sistema es va mantenir estable durant la prova.
- No es van detectar errors gràfics ni bloquejos del sistema.
- Es va poder afegir un *shader* durant la sessió d'overclocking sense canviar gairebé l'impacte sobre el rendiment, sent aquest shader l'anteriorment explorat *hq2x*.

192.168.1.36 - PuTTY

```
top - 21:50:26 up 43 min, 1 user, load average: 1.80, 1.69, 1.42
Tasks: 150 total, 2 running, 148 sleeping, 0 stopped, 0 zombie
%Cpu(s): 13.6 us, 1.3 sy, 0.0 ni, 85.2 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 7946.0 total, 7114.8 free, 310.4 used, 520.8 buff/cache
MiB Swap: 0.0 total, 0.0 free, 0.0 used. 7404.5 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
771	root	20	0	1356336	264596	68668	R	59.8	3.3	9:53.44	retroarch
129	root	-2	0	0	0	0	S	1.0	0.0	0:07.28	v3d_render
128	root	-2	0	0	0	0	S	0.3	0.0	0:04.78	v3d_bin
866	root	20	0	0	0	0	D	0.3	0.0	0:00.03	kworker/u8:3+events_unbound
1	root	20	0	157536	7700	5516	S	0.0	0.1	0:00.40	systemd
2	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kthreadd
3	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	rcu_gp
4	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	rcu_par_gp

Figura 4.16. Comanda `top` després d'haver jugat una estona sense tenir l'overclocking actiu.

192.168.1.36 - PuTTY

```
top - 22:23:39 up 27 min, 1 user, load average: 1.78, 1.68, 1.36
Tasks: 149 total, 2 running, 147 sleeping, 0 stopped, 0 zombie
%Cpu(s): 17.2 us, 0.9 sy, 0.0 ni, 81.9 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 7946.0 total, 7148.8 free, 290.5 used, 506.7 buff/cache
MiB Swap: 0.0 total, 0.0 free, 0.0 used. 7419.3 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
784	root	20	0	1326528	235976	68480	R	72.4	2.9	14:36.12	retroarch
126	root	-2	0	0	0	0	S	1.0	0.0	0:08.43	v3d_bin
127	root	-2	0	0	0	0	S	1.0	0.0	0:10.42	v3d_render
968	root	20	0	3412	1968	1536	R	0.3	0.0	0:01.02	top
976	root	20	0	0	0	0	D	0.3	0.0	0:00.13	kworker/u8:0+events_unbound
1	root	20	0	157496	7692	5516	S	0.0	0.1	0:00.44	systemd
2	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kthreadd

Figura 4.17. Comanda `top` després d'haver jugat una estona amb l'overclocking actiu.

Resultats

L'overclocking, almenys en el joc de *God of War*, no va tenir gran rellevància en veure un augment significatiu dels FPS ni millorant la fluïdesa del joc. Amb tot i això, caldria vigilar la temperatura per evitar sobreescalfaments en sessions prolongades, ja que en executar aquesta tècnica es recomana implementar sistemes de refrigeració addicional per garantir la seguretat del maquinari.

4.2.3. Anàlisi de les millores aplicades

Després de dur a terme totes les proves i aplicar les estratègies d'optimització descrites (refrigeració activa, ajustos de Lakka, ús de shaders i Run Ahead, selecció de modes de recompilació adequats, ROM optimitzades i overclocking), s'han observat millores notables en diversos paràmetres clau:

1. Temperatura

- Sense cap ventilador, la temperatura podia arribar fàcilment als 80 °C quan es feia ús intensiu de la CPU/GPU, provocant risc de *thermal throttling*.
- Amb refrigeració activa, la temperatura màxima es va situar al voltant dels 45 °C en el mateix escenari de càrrega, mantenint un rendiment òptim sense haver de reduir automàticament la freqüència del processador.

2. Ús de CPU en estat inactiu i arrencada de Lakka

- Després de desactivar serveis innecessaris i ajustar paràmetres com la *swappiness*, l'ús de CPU en repòs va baixar lleugerament (d'un ~5% a ~4,5%) i el temps d'arrencada de Lakka es va reduir d'uns 7,8 s a 7,2 s.

3. Rendiment i latència en jocs 2D

- L'aplicació de *shaders* lleugers i del *Run Ahead* a 1 frame va proporcionar una millora visual (efecte CRT o suavitzat de píxels) i va disminuir la latència percebuda, sense impactar greument en l'ús de CPU.

4. Modes de recompilació (Dolphin)

- El mode JITARM64 va oferir el millor balanç entre rendiment (FPS) i estabilitat, mentre que els modes *Interpreter* o *Cached Interpreter* produïen més ús de CPU i, fins i tot, penjaments en casos puntuals.

5. ROM optimitzades

- Les versions optimitzades de certes ROM (com *Super Mario 64* en la seva versió més polida) van exigir menys CPU i van mantenir els FPS més estables, evitant errors gràfics o retards sobtats.

6. Overclocking

- Tot i aconseguir freqüències més elevades en CPU/GPU, en el cas estudiat (*God of War: Chains of Olympus*), no es va apreciar un increment significatiu dels FPS (es va mantenir en uns 60 FPS). Això sí, la temperatura va pujar fins als $\sim 45^{\circ}\text{C}$ (amb ventilador) i cal vigilar possibles inestabilitats en sessions prolongades.

A la següent figura es mostra un gràfic de barres que sintetitza algunes de les dades comparatives més representatives abans i després de les millores principals (temperatura sota càrrega, ús de CPU en repòs i temps d'arrencada de Lakka):

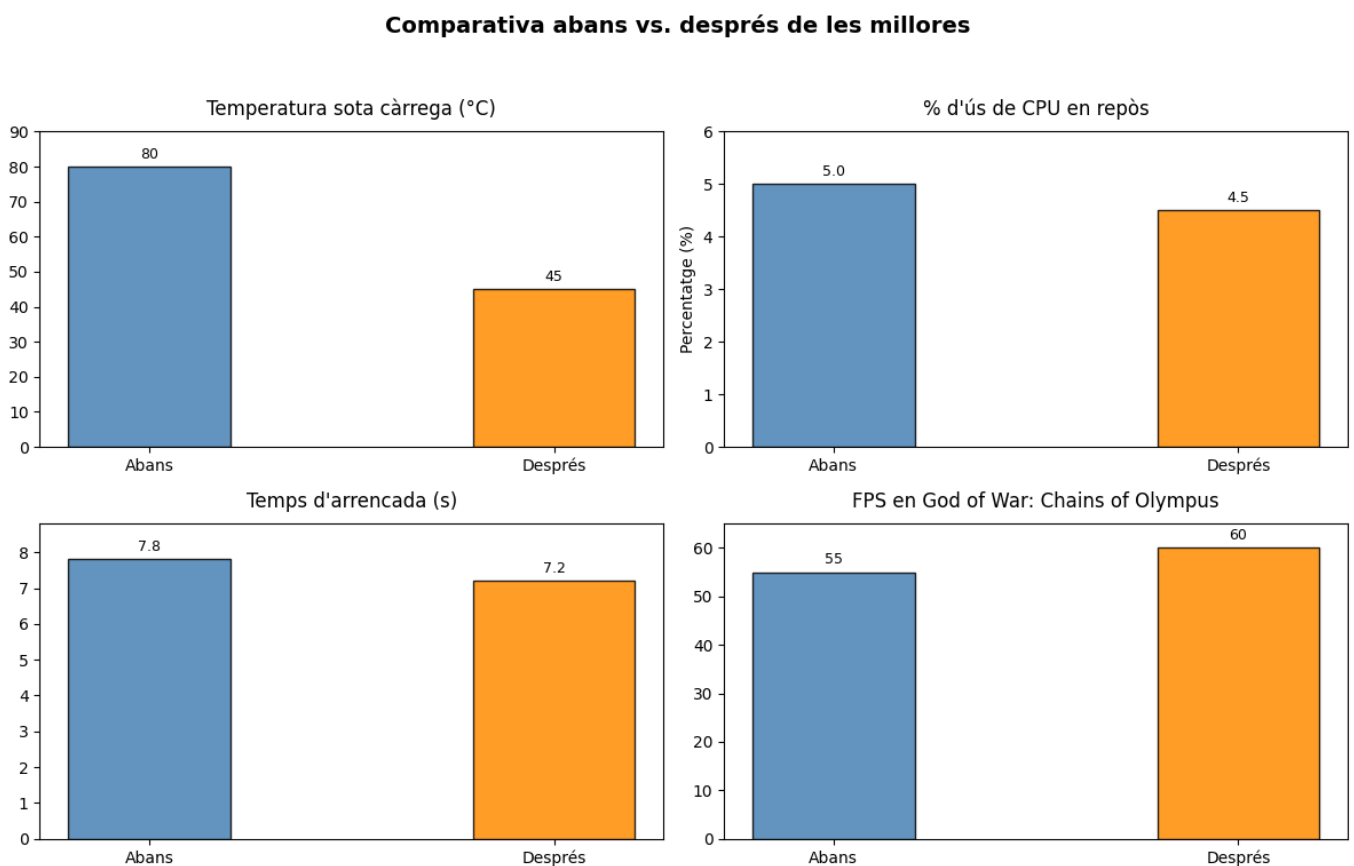


Figura 4.18. Gràfic de barres representatiu dels canvis abans i després de les millores.

En aquest gràfic s'aprecia clarament la reducció de la temperatura (de 80 °C a 45 °C), la lleu disminució de l'ús de CPU en estat inactiu (d'un 5% a un 4,5%) i el guany moderat en el temps d'arrencada (de 7,8 s a 7,2 s). Tot plegat contribueix a una experiència de joc més estable i satisfactòria en la Raspberry Pi 5 amb Lakka.

En conjunt, la mesura que més canvi ha experimentat durant les proves, fent-la rellevant a l'hora d'implementar-la ha estat la refrigeració activa, ja que ha evitat la pèrdua de rendiment derivada del *thermal throttling*. Juntament amb els ajustos de sistema operatiu i la selecció acurada de nuclis i ROM, s'ha aconseguit mantenir uns FPS estables, un temps d'arrencada una mica inferior i una reducció significativa de la temperatura.

Capítol 5.

Conclusions

En aquest capítol es presenten les reflexions finals del projecte, on es revisen els objectius plantejats i s'avalua fins a quin punt han estat assolits. A més, s'hi descriuen els problemes i reptes que han sorgit durant la realització del treball i es proposen línies de futur per a la investigació o la millora de la microconsola. Finalment, es tanquen les conclusions amb una valoració global de l'experiència i del potencial d'aquest tipus de solucions basades en SBC.

5.1. Revisió d'objectius i correlació del treball fet

L'objectiu principal d'aquest projecte era comparar i avaluar el rendiment d'emuladors de videojocs en una microconsola basada en un Single-Board Computer, per tal de determinar quines configuracions ofereixen la millor experiència de joc. Al llarg d'aquesta memòria s'han anat assolint les fites plantejades al punt [1.2](#):

- **Identificar i seleccionar emuladors compatibles amb el SBC:** S'ha fet una recerca detallada de plataformes i emuladors i s'ha validat la compatibilitat amb la Raspberry Pi 5.
- **Definir criteris d'avaluació i mètriques de rendiment:** S'han establert indicadors com els FPS, temps de càrrega, latència d'entrada, ús de recursos (CPU/RAM) i qualitat gràfica.
- **Realitzar proves de rendiment:** S'han dut a terme tests amb diferents consoles i jocs (Game Boy Advance, Nintendo DS, PSP, GameCube) per observar com cada plataforma i cada nucli d'emulació responien en condicions reals.

- **Analitzar i comparar els resultats:** S'ha comprovat que Lakka és la plataforma més estable i polivalent, mentre que RetroArch, tot i ser molt flexible, presentava limitacions amb consoles 3D exigents. Recalbox queda més orientat a sistemes retro senzills i RetroPie encara no disposa de versió oficial per a Raspberry Pi 5.
- **Proposar recomanacions per a l'ús òptim d'emuladors:** Les proves han confirmat la importància d'una bona refrigeració i de l'ús de shaders o ajustos com el Run Ahead per millorar l'experiència de joc. També s'ha comprovat que l'overclocking té un impacte discret en alguns títols, però pot ser rellevant en altres, sempre que es controli la temperatura.

D'aquesta manera, tots els objectius definits inicialment han trobat resposta en la recerca, l'experimentació i l'anàlisi d'aquest TFG.

5.2. Problemes que han anat sorgint

En el transcurs del projecte, han aparegut diverses dificultats que han impactat en el desenvolupament i la presa de decisions. A continuació es descriuen de manera ordenada:

1. Definició inicial del projecte i dubtes d'abast

Durant la fase inicial, es van plantejar diverses opcions pel TFG: des de la simple comparació d'emuladors fins al desenvolupament de codi propi o la creació de noves funcionalitats. Aquest gran ventall de possibilitats va generar incerteses sobre com enfocar el treball i quin era l'abast més realista dins el temps disponible. Finalment, es va optar per centrar-se en la comparativa i l'optimització d'emuladors en un SBC concret, descartant la part més extensa de programació o desenvolupament de nous mòduls.

2. Limitacions de hardware en arquitectura ARM

Les Single-Board Computers, tot i ser molt versàtils, treballen amb arquitectura ARM. Molts emuladors —especialment els de consoles modernes— estan concebuts per funcionar amb arquitectures x86/x64. Aquest fet ha impedit, per exemple, fer córrer PCSX2 (emulador de PlayStation 2) de manera nativa a la Raspberry Pi 5; o ha requerit modificacions i versions experimentals d'altres emuladors (com Dolphin). Això suposa un gran repte a l'hora d'emular jocs 3D molt exigents.

3. Dificultats per emular consoles 3D de generacions avançades

Tot i que Dolphin ha funcionat amb jocs de GameCube i Wii, s'han reportat “crashes” puntuals i caigudes en el rendiment, sobretot amb títols més exigents. La PS2, directament, no ha pogut ser emulada de manera estable a la Raspberry Pi 5 per la manca de versions ARM amb prou maduresa. A més, l'arquitectura ARM sol requerir recompilacions dinàmiques molt ben optimitzades per aconseguir un rendiment mínimament fluid.

4. RetroPie: inexistència d'una versió oficial i errors d'interfície

RetroPie és una de les plataformes més populars per a l'emulació en Raspberry Pi, però actualment no disposa d'una imatge oficial per a la Raspberry Pi 5. Els paquets “no oficials” sovint pateixen errors en la interfície (EmulationStation) i en la càrrega de ROMs i BIOS. Aquests problemes de compatibilitat han fet inviable el seu ús en aquest projecte, ja que es requeria un entorn estable per realitzar proves comparatives.

5. RetroArch en 64 bits: problemes de compilació

En intentar instal·lar RetroArch sobre Raspberry Pi OS de 64 bits, van aparèixer errors de compilació en alguns nuclis d'emulació. Tot i que la versió de 32 bits funcionava correctament, la de 64 bits no estava prou madura en el moment de les proves. Això va limitar l'ús de RetroArch en un entorn òptim de 64 bits i va forçar a emprar altres solucions, com Lakka, que ve ja precompilada i adaptada.

6. Requeriments de BIOS i ROMs no incloses de sèrie

Algunes plataformes (Recalbox, Lakka) no porten de sèrie les BIOS necessàries per a certes consoles (p. ex. PlayStation, PSP o GameCube). A més, l'usuari ha d'afegir manualment fitxers i carpetes específics per a cada nucli d'emulació. Això ha comportat molta feina de configuració i d'assaig-error, i

en ocasions ha estat font d'errors “fantasma” (com bloquejos o gràfics corromputs) derivats de BIOS mal ubicades o ROMs incompletes.

7. **Altres dificultats de compatibilitat amb software**

La manca de versions estables o documentació actualitzada per a Raspberry Pi 5 va fer que s'haguessin d'usar repositoris externs, scripts d'instal·lació no oficials o, fins i tot, compilar a mà certs components. Això incrementa el risc de tenir inconsistències entre llibreries i versions de l'emulador, amb la consegüent inestabilitat o menor rendiment.

8. **Dubtes sobre la part de programació**

Inicialment, no quedava clar si el projecte requeria desenvolupar codi font per a nous emuladors, modificar-ne de ja existents o limitar-se a provar-los i configurar-los. Finalment, es va descartar la part de programació intensiva, ja que s'havia optat per prioritzar la investigació comparativa i l'anàlisi de rendiment. No obstant això, en un futur, podria ser interessant endinsar-se en la modificació de nuclis o en la creació de paràmetres de compilació més avançats per a l'arquitectura ARM.

En conjunt, aquests punts han anat configurant els reptes que s'han hagut d'abordar o esquivar durant el projecte. Tot i això, la majoria s'han pogut resoldre amb investigació prèvia, versions alternatives del software i la configuració manual dels emuladors que sí que tenien prou suport per a la Raspberry Pi 5.

5.3. Línies de futur

A partir de l'experiència adquirida, es destaquen diverses línies de treball futur possibles:

1. **Optimització de nuclis i ROMs**

- **Exploració del codi font** d'alguns emuladors individuals o ROMs per millorar-ne l'eficiència i reduir-ne la latència.

- Aprofundir en la compilació de nuclis específics per a ARM, per treure el màxim profit de la Raspberry Pi 5.

2. Més suport a consoles 3D avançades

- Investigar la viabilitat d'emular PS2, GameCube o Wii en condicions òptimes, aprofitant millores de software i la creixent potència dels nous models de SBC.
- Explorar versions “experimentals” de PCSX2 o Dolphin adaptades a ARM que puguin aparèixer en el futur.

3. Plataformes més integrades

- Crear un “pack” personalitzat (com una imatge de Lakka o RetroPie) preconfigurat, amb BIOS, shaders i paràmetres òptims, per reduir la corba d'aprenentatge dels usuaris finals.
- Es podria plantejar crear desde zero o a partir del codi d'alguna d'aquestes plataformes, una plataforma personalitzada.
- Aprofundir en la interfície d'usuari per facilitar la selecció de jocs, l'aplicació de filtres i la gestió de comandaments.

4. Més proves de rendiment

- Realitzar benchmarks extensos amb jocs més complexos i consoles diverses, per extreure mètriques comparatives més completes.
- Avaluar la despesa energètica i la temperatura en sessions prolongades, així com l'estabilitat en mode overclocking intensiu.

5.4. Reflexions finals

Aquest TFG posa de manifest la capacitat dels Single-Board Computers moderns, com la Raspberry Pi 5, per convertir-se en microconsoles solvents per a l'emulació de videojocs, especialment en el terreny 2D i en consoles 3D de primeres etapes (p. ex. Nintendo DS o PSP). El suport comunitari i la flexibilitat de plataformes com Lakka han estat elements clau per aconseguir uns resultats satisfactoris.

Pel que fa a l'impacte pràctic, el treball dut a terme pot ser de gran utilitat tant en entorns educatius —per introduir la programació i l'arquitectura de sistemes— com

en aplicacions més immediates, com en projectes de lleure, esdevenir part d'arcades personals o funcionar com a solució de *streaming*, la qual cosa multiplica les seves potencials aplicacions futures amb unes poques modificacions addicionals.

Tanmateix, cal tenir present que l'arquitectura ARM continua presentant limitacions per a consoles més modernes (PS2, GameCube, Wii), i requereix optimitzacions avançades dels emuladors o configuracions de hardware molt afinades. A més, la gestió tèrmica és un factor crític: sense un bon sistema de refrigeració, el *thermal throttling* pot deteriorar el rendiment i afectar la fluïdesa de joc.

El projecte demostra que amb un pressupost relativament baix i un SBC potent, és factible construir una microconsola que ofereixi una bona experiència retro i, amb certs ajustos, fins i tot s'apropi a l'emulació de jocs 3D exigents. Això obre les portes a nous desenvolupaments futurs que puguin aprofundir encara més en l'optimització de software, la integració de hardware i l'ampliació de la compatibilitat amb consoles de generacions més recents, tot contribuint a la preservació i l'accessibilitat del patrimoni cultural dels videojocs.

Referències

A continuació, es mostren les referències en format APA. Com que la majoria d'aquests llocs web no inclouen una data de publicació explícita, s'utilitza "(n.d.)" i s'especifica "Recuperat/da el [data] de [URL]" per indicar la data de consulta.

1. **Single-board computer.** (n.d.). *Viquipèdia*. Recuperat el 7 de juliol de 2024, de https://en.wikipedia.org/wiki/Single-board_computer
2. **Avnet.** (n.d.). *Single-board computers*. Recuperat el 7 de juliol de 2024, de <https://www.avnet.com/shop/us/c/embedded-boards-systems/single-board-computers/>
3. **Mouser.** (n.d.). *Single-board computers*. Recuperat el 7 de juliol de 2024, de <https://www.mouser.es/c/embedded-solutions/computing/single-board-computers/#>
4. **Raspberry Pi.** (n.d.). *Raspberry Pi*. Recuperat el 7 de juliol de 2024, de <https://www.raspberrypi.com/>
5. **Raspberry Pi.** (n.d.). *Viquipèdia*. Recuperat el 7 de juliol de 2024, de https://ca.wikipedia.org/wiki/Raspberry_Pi
6. **BeagleBoard.org.** (n.d.). *BeagleBoard Documentation*. Recuperat el 19 de juliol de 2024, de <https://docs.beagleboard.org/latest/>
7. **BeagleBoard.** (n.d.). *Viquipèdia (en anglès)*. Recuperat el 19 de juliol de 2024, de <https://en.wikipedia.org/wiki/BeagleBoard>
8. **ODROID Wiki.** (n.d.). Recuperat el 19 de juliol de 2024, de <https://wiki.odroid.com/>
9. **ODROID.** (n.d.). *Viquipèdia (en anglès)*. Recuperat el 19 de juliol de 2024, de <https://en.wikipedia.org/wiki/ODROID>
10. **Asus.** (n.d.). *Tinker Board Series*. Recuperat el 19 de juliol de 2024, de <https://tinker-board.asus.com/product/series.html>
11. **Asus Tinker Board.** (n.d.). *Viquipèdia (en anglès)*. Recuperat el 19 de juliol de 2024, de https://en.wikipedia.org/wiki/Asus_Tinker_Board
12. **96Boards.** (n.d.). *About*. Recuperat el 19 de juliol de 2024, de <https://www.96boards.org/about>
13. **Linaro Community Board Group.** (n.d.). *Viquipèdia (en anglès)*. Recuperat el 19 de juliol de 2024, de https://en.wikipedia.org/wiki/Linaro#Linaro_Community_Board_Group
14. **ODROID-XU4.** (n.d.). Recuperat el 20 de juliol de 2024, de <https://www.odroid.co.uk/odroid-xu4>
15. **Raspberry Pi 4 Model B (Reseller type).** (n.d.). Recuperat el 20 de juliol de 2024, de <https://www.raspberrypi.com/products/raspberry-pi-4-model-b/?resellerType=industry>
16. **Raspberry Pi 4 Model B (Specifications).** (n.d.). Recuperat el 20 de juliol de 2024, de <https://www.raspberrypi.com/products/raspberry-pi-4-model-b/specifications/>
17. **HiKey960.** (n.d.). *96Boards*. Recuperat el 20 de juliol de 2024, de <https://www.96boards.org/product/hikey960/>
18. **Asus.** (n.d.). *Tinker Board 3 Series*. Recuperat el 20 de juliol de 2024, de <https://tinker-board.asus.com/series/tinker-board-3.html>
19. **BeagleBoard.org.** (n.d.). *Boards*. Recuperat el 20 de juliol de 2024, de <https://www.beagleboard.org/boards>
20. **Mouser.** (n.d.). *BeagleBoard Beagle y AI Computer*. Recuperat el 22 de juliol de 2024, de <https://www.mouser.es/new/beagleboardorg/beagleboard-beagle-y-ai-computer/>

21. **Raspberry Pi 5 (8GB).** (n.d.). Recuperat el 22 de juliol de 2024, de <https://www.raspberrypi.com/products/raspberry-pi-5/?variant=raspberry-pi-5-8gb>
22. **Raspberry Pi Pico (WH).** (n.d.). Recuperat el 22 de juliol de 2024, de <https://www.raspberrypi.com/products/raspberry-pi-pico/?variant=raspberry-pi-pico-wh>
23. **Hardkernel.** (n.d.). *ODROID-H4*. Recuperat el 22 de juliol de 2024, de <https://www.hardkernel.com/shop/odroid-h4/>
24. **96Boards.** (n.d.). *DragonBoard 410c*. Recuperat el 22 de juliol de 2024, de <https://www.96boards.org/product/dragonboard410c/>
25. **TechPowerUp.** (n.d.). *CPU Specs: Processor N97*. Recuperat el 22 de juliol de 2024, de <https://www.techpowerup.com/cpu-specs/processor-n97.c3009>
26. **Kubii.** (n.d.). *Tarjeta ODROID-H4*. Recuperat el 22 de juliol de 2024, de [https://www.kubii.com/es/microordenadores/4306-2062-tarjeta-odroid-h4-h4-h4-ultra-3272496318649.html#/version odroid h4 h4 h4 ultra-h4](https://www.kubii.com/es/microordenadores/4306-2062-tarjeta-odroid-h4-h4-h4-ultra-3272496318649.html#/version%20odroid%20h4%20h4%20h4%20ultra%20h4)
30. **Microconsole.** (n.d.). *Viquipèdia (en anglès)*. Recuperat el 10 d'agost de 2024, de <https://en.wikipedia.org/wiki/Microconsole>
31. **Microconsolas Android.** (n.d.). *Xataka Android*. Recuperat el 13 d'agost de 2024, de <https://www.xatakandroid.com/gadgets-android/microconsolas-android-quienes-son-y-cual-es-su-futuro>
32. **TomsGuide.** (n.d.). *Best streaming players*. Recuperat el 13 d'agost de 2024, de <https://www.tomsguide.com/us/best-streaming-players.review-2140.html>
33. **Atari.** (n.d.). *Atari VCS*. Recuperat el 13 d'agost de 2024, de <https://atari.com/pages/atari-vcs>
34. **GameStick.** (n.d.). *Xataka*. Recuperat el 13 d'agost de 2024, de <https://www.xataka.com/videojuegos/gamestick-es-la-consola-embebida-en-un-controlador-bajo-un-diseno-minimalista>
35. **Xataka.** (n.d.). *Mejores Smart TV Box*. Recuperat el 13 d'agost de 2024, de <https://www.xataka.com/seleccion/mejores-smart-tv-box-cual-comprar-12-set-top-box-recomendados-34-160-euros>
36. **NVIDIA Shield TV Pro.** (n.d.). *TechRadar*. Recuperat el 13 d'agost de 2024, de <https://www.techradar.com/reviews/nvidia-shield-tv-pro>
37. **Amazon Fire TV 4K.** (n.d.). *TechRadar*. Recuperat el 13 d'agost de 2024, de <https://www.techradar.com/televisions/streaming-devices/amazon-fire-tv-4k-review>
38. **Ouya.** (n.d.). *Viquipèdia (en castellà)*. Recuperat el 13 d'agost de 2024, de <https://es.wikipedia.org/wiki/Ouya>
39. **GameStick vs OUYA.** (n.d.). *TechSpot*. Recuperat el 13 d'agost de 2024, de <https://www.techspot.com/news/51212-ouya-console-competitor-gamestick-launches-kickstarter-campaign.html>
40. **Emulador.** (n.d.). *Viquipèdia (en català)*. Recuperat el 14 d'agost de 2024, de https://ca.wikipedia.org/wiki/Emulador_de_codi
41. **PCSX2.** (n.d.). Recuperat el 14 d'agost de 2024, de <https://pcsx2.net/>
42. **XDA-Developers.** (n.d.). *Emulate games on Raspberry Pi*. Recuperat el 14 d'agost de 2024, de <https://www.xda-developers.com/emulate-games-raspberry-pi/>
43. **Dolphin Emulator.** (n.d.). Recuperat el 14 d'agost de 2024, de <https://es.dolphin-emu.org/?cr=es>
44. **Lakka.** (n.d.). Recuperat el 14 d'agost de 2024, de <https://www.lakka.tv/>
45. **Recalbox.** (n.d.). Recuperat el 14 d'agost de 2024, de <https://www.recalbox.com/es/>
46. **TimeExtension.** (n.d.). *Where do you stand on ethical emulation*. Recuperat el 14 d'agost de 2024, de <https://www.timeextension.com/features/talking-point-where-do-you-stand-on-ethical-emulation>
47. **PC Gamer.** (n.d.). *The ethics of emulation*. Recuperat el 14 d'agost de 2024, de <https://www.pcgamer.com/the-ethics-of-emulation-how-creators-the-community-and-the-law-view-console-emulators/>

48. **Irdeto.** (n.d.). *The dark side of game emulation*. Recuperat el 14 d'agost de 2024, de <https://irdeto.com/blog/the-dark-side-of-game-emulation-how-its-hurting-the-industry>
49. **YouTube (Modern Vintage Gamer).** (n.d.). *The ethics of emulation*. Recuperat el 14 d'agost de 2024, de https://www.youtube.com/watch?v=Du4LK_Dq1FA
50. **Pocket-lint.** (n.d.). *Raspberry Pi 5 vs 4*. Recuperat el 17 d'agost de 2024, de <https://www.pocket-lint.com/raspberry-pi-5-vs-4/>
51. **Raspberry Pi Documentation.** (n.d.). *Getting started*. Recuperat el 4 de setembre de 2024, de <https://www.raspberrypi.com/documentation/computers/getting-started.html>
52. **Recalbox.** (n.d.). *Descargar e instalar Recalbox*. Recuperat el 10 de setembre de 2024, de <https://www.recalbox.com/es/>
53. **Lakka.** (n.d.). *Descargar e instalar Lakka*. Recuperat el 10 de setembre de 2024, de <https://www.lakka.tv/>
54. **Raspberry Pi Forums.** (n.d.). *Descargar e instalar mgba*. Recuperat el 10 de setembre de 2024, de <https://forums.raspberrypi.com/viewtopic.php?t=366696>
55. **XDA-Developers.** (n.d.). *How I turned my Raspberry Pi into an emulation beast*. Recuperat el 26 de setembre de 2024, de <https://www.xda-developers.com/how-i-turned-my-raspberry-pi-into-an-emulation-beast/>
56. **Reddit (r/raspberry_pi).** (n.d.). *Problem compiling RetroArch (64-bit)*. Recuperat el 26 de setembre de 2024, de https://www.reddit.com/r/raspberry_pi/comments/spxeif/problem_compiling_retroarch/?r_dt=47934
57. **Reddit (r/RetroArch).** (n.d.). *Compile for 64bit Raspberry Pi*. Recuperat el 26 de setembre de 2024, de https://www.reddit.com/r/RetroArch/comments/18mhgh5/anyone_manage_to_compile_it_for_64bit_raspberry_pi/
58. **GitHub (danielfreer).** (n.d.). *raspberrypi5-retroPie-setup*. Recuperat el 26 de setembre de 2024, de <https://github.com/danielfreer/raspberrypi5-retroPie-setup>
59. **Pi-Apps.** (n.d.). *Install PPSSPP (PSP emulator) on Raspberry Pi*. Recuperat el 26 de setembre de 2024, de <https://pi-apps.io/install-app/install-ppsspp-psp-emulator-on-raspberry-pi/>
60. **RetroPie Forums.** (n.d.). *Dolphin GameCube/Wii performance on Pi 5*. Recuperat el 4 d'octubre de 2024, de <https://retropie.org.uk/forum/topic/35203/dolphin-gamecube-wiiware-wii-compatibility-on-pi-5>
61. **Libretro Forums.** (n.d.). *Lakka no audio problem*. Recuperat el 12 d'octubre de 2024, de <https://forums.libretro.com/t/lakka-no-audio-problem/35147/2>
62. **Recalbox Wiki.** (n.d.). *Emulator compatibility*. Recuperat el 12 d'octubre de 2024, de <https://wiki.recalbox.com/en/hardware-compatibility/emulators-compatibility>
63. **RetroPie Documentation.** (n.d.). Recuperat el 10 de novembre de 2024, de <https://github.com/RetroPie/RetroPie-Docs/>
64. **Reddit (r/emulation).** (n.d.). *What is a BIOS?* Recuperat el 10 de novembre de 2024, de https://www.reddit.com/r/emulation/comments/1xuduj/what_is_a_bios/
65. **GitHub (archtaurus).** (n.d.). *RetroPieBIOS*. Recuperat el 10 de novembre de 2024, de <https://github.com/archtaurus/RetroPieBIOS>
66. **YouTube (ChaseBrogan).** (n.d.). *Instal·lar BIOS en RetroPie*. Recuperat el 10 de novembre de 2024, de <https://www.youtube.com/watch?v=xbNaDD4pHa8>
67. **Lakka.** (n.d.). *BIOSes*. Recuperat el 16 de novembre de 2024, de <https://www.lakka.tv/doc/BIOSes/>
68. **GitHub (Abdess).** (n.d.). *retroarch_system (releases)*. Recuperat l'1 de desembre de 2024, de https://github.com/Abdess/retroarch_system/releases/
69. **Internet Archive.** (n.d.). *Retroarch System Files*. Recuperat l'1 de desembre de 2024, de <https://archive.org/download/RetroarchSystemFiles/>

70. **Internet Archive.** (n.d.). *Retro ROMs (best set)*. Recuperat l'1 de desembre de 2024, de <https://archive.org/details/retro-roms-best-set>
71. **Switcher.gg.** (n.d.). *How to setup the RetroArch PS2 core (PCSX2 lrps2)*. Recuperat l'1 de desembre de 2024, de <https://switcher.gg/s/emulation/how-to-setup-the-retroarch-ps2-core-pcsx2-lrps2/>
72. **Lakka.** (n.d.). *Audio settings*. Recuperat l'1 de desembre de 2024, de <https://www.lakka.tv/doc/Audio-settings/>
73. **Raspberry Pi Documentation.** (n.d.). *Getting started (RPI5 info)*. Recuperat l'11 de desembre de 2024, de <https://www.raspberrypi.com/documentation/computers/getting-started.html>
74. **TodoRaspberryPi.** (n.d.). *Raspberry Pi 5*. Recuperat el 22 de desembre de 2024, de <https://todoraspberrypi.com/articulos/raspberry-pi-5>
75. **OmniRetro.** (n.d.). *Raspberry Pi 5 es una realidad*. Recuperat el 22 de desembre de 2024, de <https://omniretro.com/blog/tecnologia/ya-llego-ya-esta-aqui-raspberry-pi-5-es-una-realidad-y-estos-son-sus-detalles>
76. **Raspberry Pi.** (n.d.). *Heating and cooling Raspberry Pi 5*. Recuperat el 22 de desembre de 2024, de <https://www.raspberrypi.com/news/heating-and-cooling-raspberry-pi-5>
77. **How-To Geek.** (n.d.). *Does your Raspberry Pi need an active cooler?*. Recuperat el 22 de desembre de 2024, de <https://www.howtogeek.com/does-your-raspberry-pi-need-an-active-cooler-how-to-know-its-too-hot>
78. **UNED.** (n.d.). *Tema 16. Sistemas operatius (gestió de processos)*. Recuperat el 22 de desembre de 2024, de <https://www.studocu.com/es/document/uned/sistemas-operativos/tema-16-sistemas-operatius-gestio-de-processos/14163034>
79. **IOC XTEC.** (n.d.). *Sistemas operatius: gestió de processos*. Recuperat el 22 de desembre de 2024, de https://ioc.xtec.cat/materials/FP/Recursos/fp_asx_mo6_/web/fp_asx_mo6_htmlindex/WebContent/u1/a1/continguts.html
80. **GitHub (libretro).** (n.d.). *crt-pi.glsl (CRT shader)*. Recuperat el 25 de desembre de 2024, de <https://github.com/libretro/glsl-shaders/blob/master/crt/shaders/crt-pi.glsl>
81. **Libretro Forums.** (n.d.). *Recommend shaders for the Lakka documentation*. Recuperat el 25 de desembre de 2024, de <https://forums.libretro.com/t/recommend-shaders-for-the-lakka-documentation/5243>

Glossari i acrònims

A continuació es presenten alguns termes clau i acrònims d'aquest treball que poden necessitar una definició deguts a ser més tècnics.

- ***Antialiasing***

Tècnica de suavitzat de les vores en gràfics 2D o 3D, per evitar l'efecte d'esglaonament en línies diagonals.

- ***ARM (Advanced RISC Machine)***

Arquitectura de processador molt comuna en dispositius mòbils i SBC, dissenyada per consumir pocs recursos i baixa potència.

- ***Arquitectura x86/x64***

Tipus d'arquitectura freqüent en ordinadors d'escriptori i portàtils, generalment més potent però més consumidora d'energia.

- ***BIOS (Basic Input/Output System)***

Firmware bàsic que inicialitza el hardware abans d'arrencar el sistema operatiu o consola. En emulació, cal afegir-lo per a sistemes com PlayStation o PSP.

- ***CPU (Central Processing Unit)***

Unitat de processament principal d'un dispositiu, encarregada de dur a terme la majoria dels càlculs i instruccions generals.

- ***CRT (Cathode Ray Tube)***

Tecnologia de pantalles antigues basades en tubs de raigs catòdics. S'emula amb shaders per recrear l'efecte retro.

- ***Domòtica***

Automatització i control intel·ligent de llars, sovint basades en SBC i dispositius connectats.

- ***eMMC (Embedded MultiMediaCard)***

Tipus d'emmagatzematge sòlid integrat en alguns SBC, més ràpid que la targeta SD.

- ***Firmware (o microprogramari)***

Programari de baix nivell que enllaça el hardware amb el sistema operatiu (p. ex. BIOS).

- **FPS (*Frames Per Second*)**

Indicador de quants fotogrames es renderitzen per segon en un joc. Més FPS implica més fluïdesa visual.

- **Frames**

Fotogrames individuals que el sistema mostra per segon (relacionat directament amb FPS).

- **GB (Gigabytes)**

Unitat de mesura de la capacitat de memòria o emmagatzematge.

- **GPIO (*General-Purpose Input/Output*)**

Pins de connexió de dades en un SBC, usats per connectar sensors, LEDs o altres dispositius.

- **GPU (*Graphics Processing Unit*)**

Processador especialitzat en càlculs gràfics, encarregat de renderitzar imatges, textures i efectes visualment complexos.

- **Hardware (o maquinari)**

Conjunt físic d'elements que formen un dispositiu: CPU, GPU, RAM, etc.

- **HDMI (*High-Definition Multimedia Interface*)**

Interfície de vídeo i àudio digital utilitzada per connectar pantalles o televisors.

- **IoT (*Internet of Things*)**

Xarxa d'objectes quotidians connectats a Internet per interactuar i compartir dades.

- **Memòria RAM**

Memòria de treball d'accés ràpid on es carreguen el sistema operatiu i els programes durant la seva execució.

- **Memòria swap**

Àrea d'emmagatzematge secundari que el sistema operatiu utilitza quan s'esgota la RAM, encara que amb un accés molt més lent.

- **Overclocking**

Pràctica d'augmentar la freqüència de CPU/GPU més enllà de la configuració oficial, guanyant rendiment però incrementant temperatura i consum.

- **Pantalles CRT**

(Sinònim de CRT, vegeu “[CRT](#)” més amunt)

- ***Plug and play***

Sistema de reconeixement i configuració automàtica de perifèrics, sense intervenció manual avançada.

- **RAM (*Random Access Memory*)**

(vegeu “[Memòria RAM](#)”)

- **ROM (*Read-Only Memory*)**

En emulació, fa referència als fitxers que contenen la informació d'un joc original.

- ***Run Ahead***

Opció de RetroArch/Lakka que redueix la latència percebuda avançant la simulació un o més fotogrames endavant.

- **SBC (*Single-Board Computer*)**

Ordinador complet integrat en una sola placa (CPU, RAM, etc.). Ex. Raspberry Pi, Odroid, etc.

- ***Software* (o programari)**

Conjunt de programes, aplicacions i sistemes que s'executen sobre el hardware, com el sistema operatiu o els emuladors.

- **SSD (*Solid-State Drive*)**

Unitat d'emmagatzematge basat en xips de memòria flaix, més ràpid que els discos durs tradicionals.

- ***Thermal throttling***

Reducció automàtica de la velocitat de CPU/GPU en superar cert llindar de temperatura, per evitar danys o sobreescalfament.

- **USB (*Universal Serial Bus*)**

Estàndard de connexió per a perifèrics, emmagatzematge extern, etc.

Apèndix A

Manual d'instal·lacions i configuracions

L'objectiu d'aquest apèndix és oferir una guia de referència perquè qualsevol persona pugui reproduir el procés de configuració de la Raspberry Pi 5 i instal·lar-hi les eines necessàries per fer-la servir com a microconsola. S'hi inclou:

- Instal·lació de Raspberry Pi OS (en modalitat Lite o completa).
- Instal·lació i configuració de Lakka, un sistema minimalista dedicat a l'emulació.
- Instal·lació manual de RetroArch sobre Raspberry Pi OS, en cas de voler una plataforma d'emulació modular.

Finalment, al subapartat A.3, s'hi troba una guia d'ús per a l'usuari final.

A.1. Instal·lació de Raspberry Pi OS

1. Descarregar i preparar el sistema operatiu

- Baixa l'eina oficial **Raspberry Pi Imager** des de <https://www.raspberrypi.com/software/>.
- Insereix la targeta microSD (o unitat USB) al PC i selecciona la versió de Raspberry Pi OS que vulguis (Lite o completa).

2. Escriure la imatge

- A **Raspberry Pi Imager**, selecciona *Choose OS > Raspberry Pi OS (32 o 64 bits)* o *Raspberry Pi OS Lite (32 o 64 bits)*.
- *Choose Storage* > escull la targeta/USB.
- (Opcional) A “*Advanced settings*” pots establir nom d'usuari, contrasenya, Wi-Fi, etc.

- Fes *Write* i espera que acabi el procés.

3. Configuració inicial

- Connecta la Raspberry Pi 5 al monitor via micro-HDMI i a la xarxa Ethernet (o Wi-Fi).
- Un cop arrencada, executa `sudo apt update && sudo apt upgrade -y`.

Verificar la instal·lació

- Reinicia la Raspberry Pi i comprova que el sistema arrenca correctament.

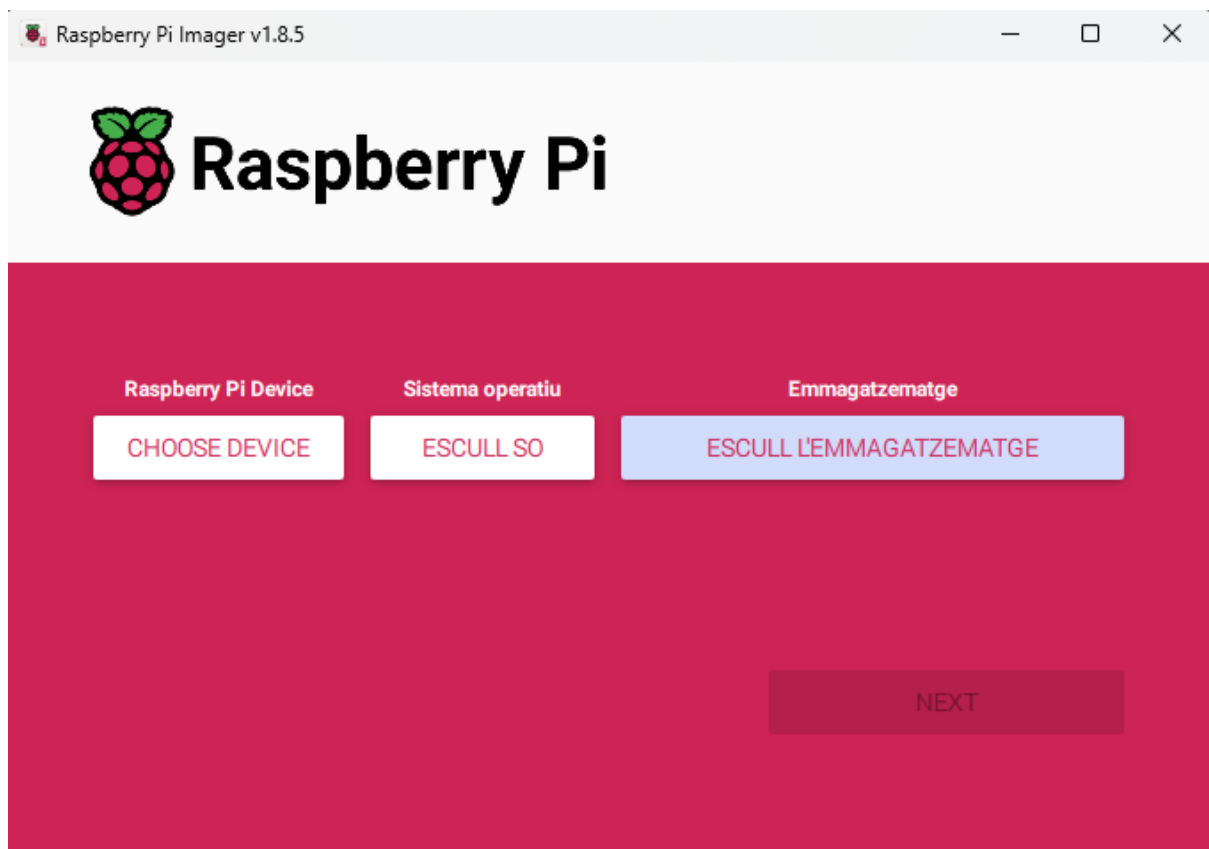


Figura A.1. Interfície gràfica del Raspberry Pi Imager.

A.2. Instal·lació de Lakka i RetroArch

Lakka com a sistema operatiu independent

1. **Descarregar la imatge** de Lakka des de <https://www.lakka.tv/get/windows/rpi/> (la versió específica per a Raspberry Pi 5, si n'hi ha de disponibles).
2. **Escriure la imatge** a la targeta SD/USB amb l'eina Raspberry Pi Imager (mateix procediment que amb Raspberry Pi OS a l'apartat A.1).
3. **Encendre la Pi**: Lakka arrencarà directament a la interfície basada en RetroArch.
4. **Accedir via SSH** (usuari: **root**, contrasenya: **root**) per si cal transferir ROMs, BIOS o editar configuracions avançades.

Recalbox com a sistema operatiu independent

1. **Escriure la imatge** a la targeta SD/USB amb l'eina Raspberry Pi Imager (mateix procediment que amb Raspberry Pi OS). En l'apartat per a escollir SO, cal baixar fins l'apartat 'Emulation and Game OS', dins del qual estarà la instància de Recalbox. Si no hi és, provar a tornar a baixar l'Imager des de l'enllaç de Recalbox:
https://www.recalbox.com/es/download/stable/rpi/rpi5_64/
2. **Encendre la Pi**: Lakka arrencarà directament a la interfície basada en RetroArch.

RetroArch sobre Raspberry Pi OS

1. Instal·lar dependències:

```
sudo apt install -y build-essential libudev-dev libegl-dev libgles-dev  
libx11-xcb-dev libpulse-dev libasound2-dev libavcodec-dev libavdevice-dev  
libavformat-dev libfreetype6-dev libgbm-dev libsdl2-dev libswresample-dev  
libswscale-dev zlib1g-dev
```

2. Clonar RetroArch

```
git clone https://github.com/libretro/RetroArch -b v1.15.0  
cd RetroArch  
./configure  
make -j4  
sudo make install
```

3. Executar RetroArch

```
retroarch
```

4. **Descarregar nuclis** des de la mateixa interfície (Online Updater > Core Updater).
Afegir manuals, BIOS, etc., segons la consola a emular.

A.3. Manual d'usuari bàsic

Aquest apartat s'adreça a l'usuari final que vulgui jugar directament sense haver de tocar configuracions complexes.

1. Connexió inicial

- Connecta l'alimentació, el cable HDMI al televisor/monitor i, si escau, un cable Ethernet (o Wi-Fi) per a actualitzacions.
- Encén la Raspberry Pi i espera que aparegui la interfície (Lakka o RetroArch).

2. Afegir jocs

- Les ROM i BIOS no venen de sèrie. Per afegir-les:
 - Si fas servir Lakka, pots accedir via xarxa o SSH (sftp) i pujar-les a la carpeta `/storage/roms/`.
 - Amb RetroArch sobre Raspberry Pi OS, pots fer-ho via FileZilla, posant-les dins `~/.config/retroarch/roms/`, o la ubicació que hagis definit.

3. Selecció de jocs

- Un cop les ROM estan a la carpeta corresponent, Lakka o RetroArch les detectaran i les mostraran en la interfície. Tria la consola desitjada i escull el joc.

4. Controladors i comandaments

- En connectar un comandament USB, la majoria de vegades es configura directament mitjançant el plug and play (vegeu el [glossari](#)).
- Pots reassignar botons des de *Settings > Input*.

5. Ajustos d'emulació

- Si notes retard en la resposta, activa el *Run Ahead* (RetroArch) o revisa la configuració de *frameskip*.
- Si vols un efecte retro, aplica shaders (p. ex. [crt-pi.glsl](#)).

6. Problemes freqüents

- **No arrenca?** Revisa l'alimentació i la targeta SD.
- **No detecta el comandament?** Prova a canviar de port USB, o configura manualment els botons mitjançant un teclat.
- **No carrega la ROM?** Assegura't que la BIOS estigui correctament ubicada (algunes consoles ho requereixen).
- **No funciona l'àudio i la configuració no mostra totes les opcions?** Canvia el cable HDMI de port en la placa perquè s'actualitzi.

Amb aquests passos, l'usuari pot començar a jugar sense complicacions. Per aspectes més avançats (instal·lació de nuclis, configuració profunda, etc.), vegeu l'apèndix B.

Apèndix B

Scripts, configuracions i eines de suport

En aquest apèndix es recullen exemples de codi, scripts i instruccions addicionals per a usuaris que vulguin personalitzar o compilar components avançats. A més, inclou la secció on s'explica l'ús de PuTTY i FileZilla per connectar-se remotament a la Raspberry Pi i gestionar arxius.

B.1. Eines emprades

PuTTY és una eina gratuïta per connectar-te via SSH a la Raspberry Pi des d'un ordinador amb Windows.

- **Instal·lació:** Descarrega el fitxer executiu de <https://www.PuTTY.org/> i instal·la'l.
- **Configuració:**
 - Obre PuTTY i a "Host Name (or IP address)" escriu la IP de la Raspberry Pi (en el meu cas era `192.168.1.36` per a Lakka) i el port 22 (per defecte).
 - Clica Open i, quan surti la consola, introdueix usuari i contrasenya (p. ex. `root` / `root` a Lakka).
- **Ús bàsic:**
 - Un cop dins, pots executar comandes Linux, navegar per carpetes (`cd /home/pi`), editar fitxers, etc.
 - Ideal per comprovar el consum de CPU amb `top`, i controlar la temperatura amb `cat /sys/class/thermal/thermal_zone0/temp`.

Això facilita l'administració remota sense necessitat de tenir teclat i monitor físics connectats a la Raspberry Pi.

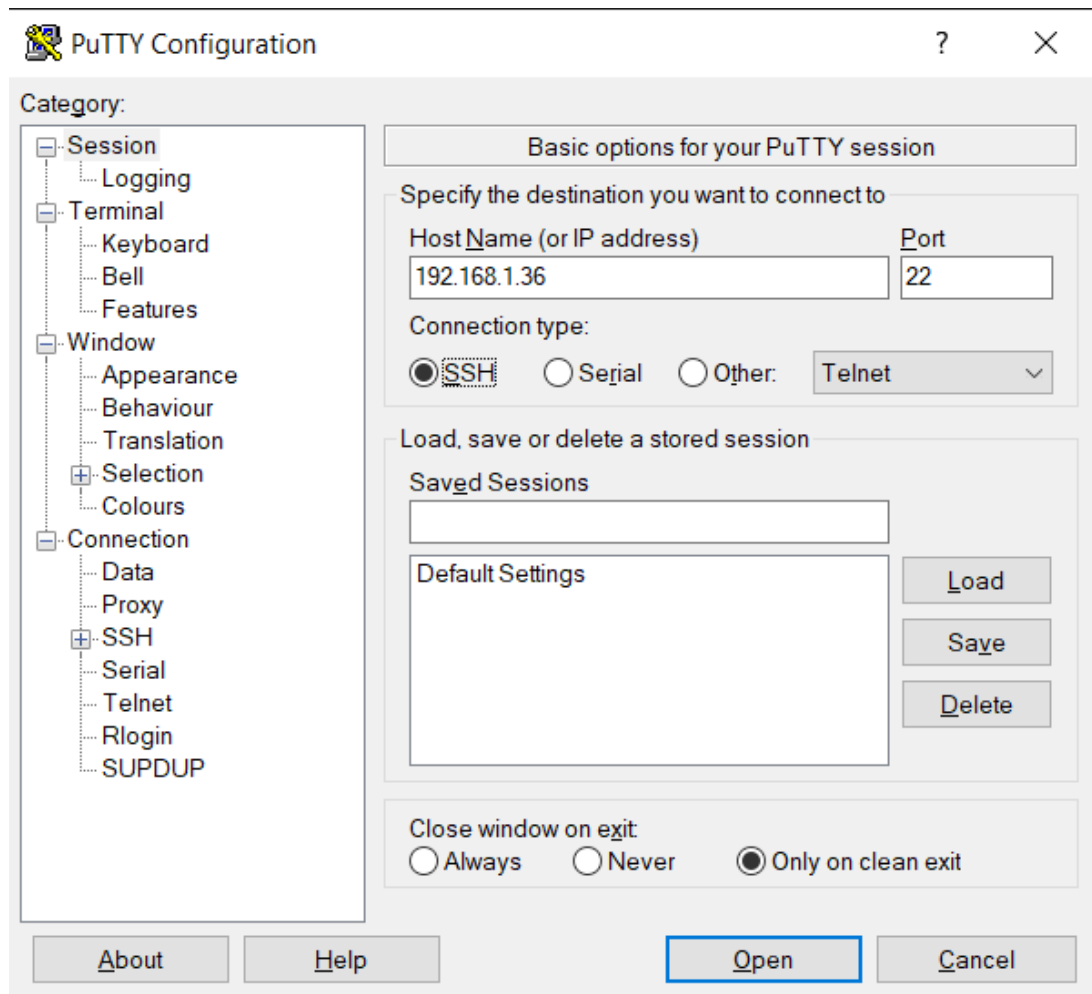


Figura B.1. Interfície de l'eina PuTTY per a Windows en la pantalla d'inici, on es configura la connexió.

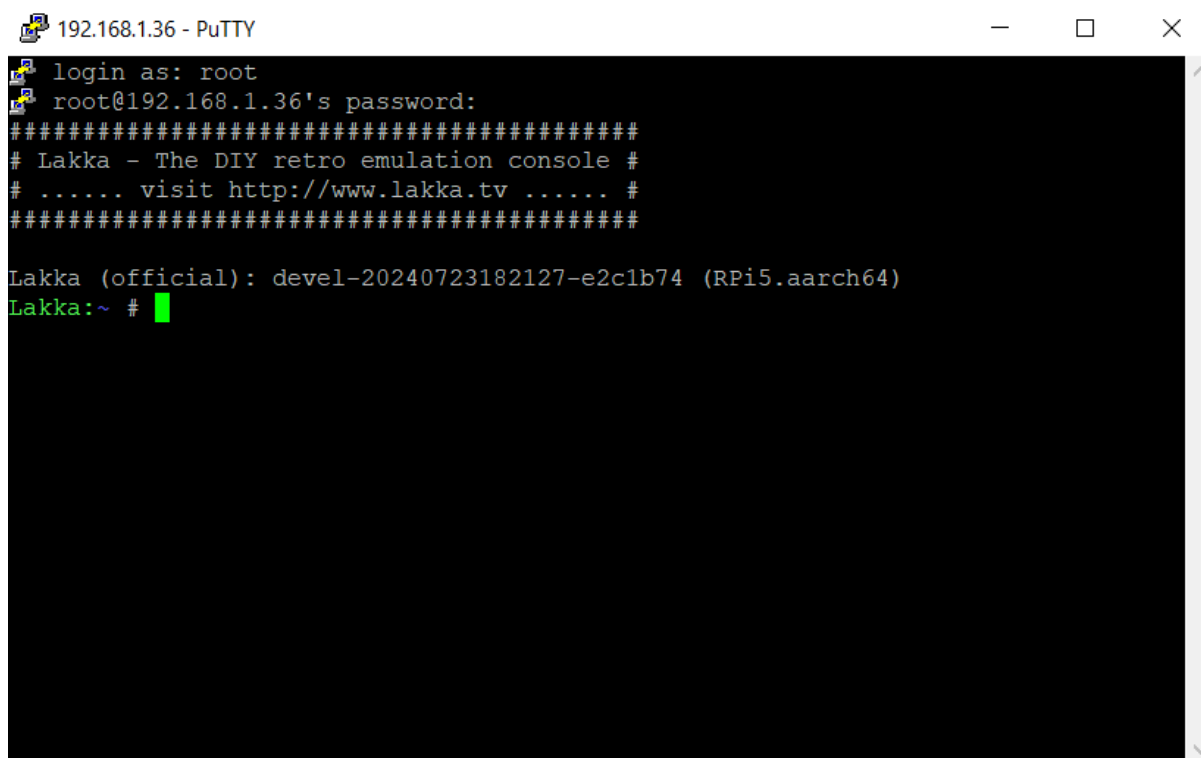
A screenshot of a PuTTY terminal window titled "192.168.1.36 - PuTTY". The terminal shows a login sequence: "login as: root", "root@192.168.1.36's password:", followed by a series of hash characters. Then, it displays the Lakka banner: "# Lakka - The DIY retro emulation console #", "# visit http://www.lakka.tv #", and another series of hash characters. Below this, it shows the Lakka version and architecture: "Lakka (official): devel-20240723182127-e2c1b74 (RPi5.aarch64)". The prompt "Lakka:~ # " is visible with a green cursor.

Figura B.2. Interfície de l'eina PuTTY per a Windows, en la CLI on s'introdueixen les comandes.

FileZilla permet pujar i baixar fitxers (ROMs, BIOS, captures) fàcilment mitjançant SFTP.

- **Instal·lació:** Descarrega'l de <https://FileZilla-project.org/>.
- **Configuració:**
 - A "Site Manager" (Gestor de llocs), crea un perfil nou amb:
 - **Protocol:** SFTP – SSH File Transfer Protocol
 - **Host:** 192.168.1.45 (exemple)
 - **Port:** 22
 - **Usuari/Contrasenya:** els mateixos que per a PuTTY
 - Desa i connecta't.
- **Ús bàsic:**
 - A la part esquerra veus els fitxers del teu PC, a la dreta els de la Raspberry Pi.
 - Arrossega ROMs o BIOS cap a la carpeta corresponent:
 - Lakka: /storage/roms

■ RetroArch (Pi OS): ~/.config/retroarch/roms

Aquesta eina és molt útil per gestionar grans quantitats de ROMs sense haver de fer servir comandes manualment.

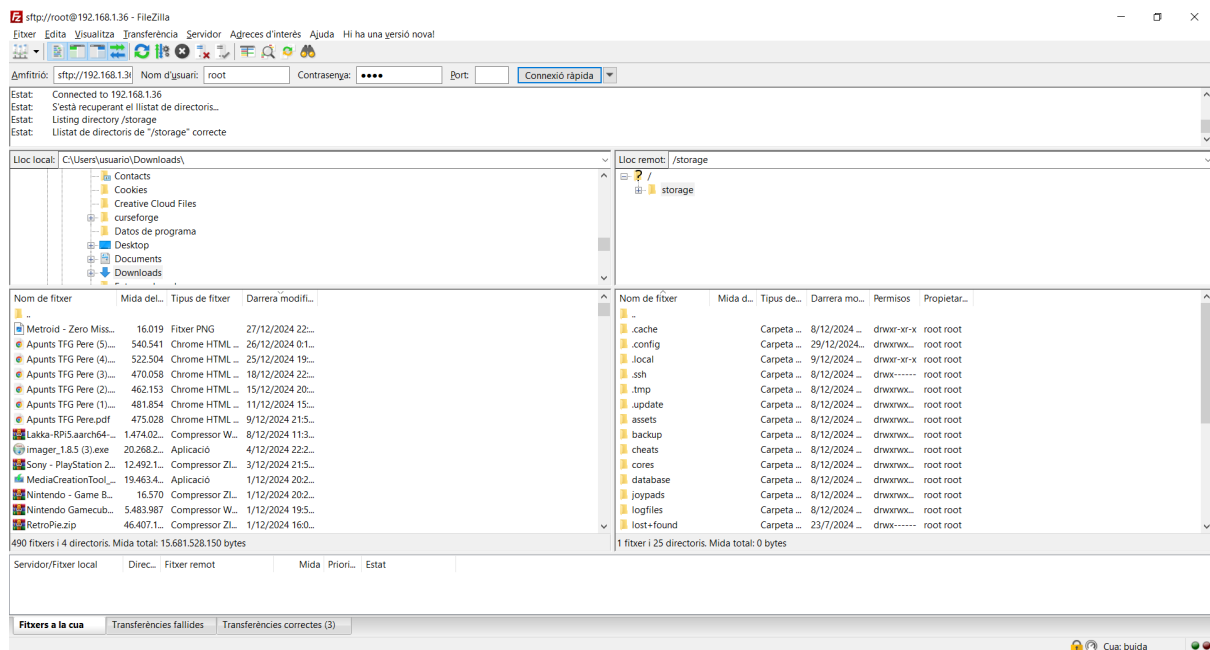


Figura B.3. Interfície de l'eina FileZilla per a Windows, la qual té una interfície de connexió i de navegació junta.

B.2. Script d'instal·lació de Dolphin a RPi OS

```
#!/bin/bash
# Comandes per compilar Dolphin a RPiOS

sudo apt update
sudo apt install -y build-essential cmake git libevdev-dev \
libudev-dev libxrandr-dev libwayland-dev libegl1-mesa-dev \
libgles2-mesa-dev libavcodec-dev libavformat-dev \
libswscale-dev
git clone https://github.com/dolphin-emu/dolphin.git
cd dolphin
mkdir build && cd build
cmake .. -DCMAKE_BUILD_TYPE=Release
make -j$(nproc)
sudo make install
```

Executant aquest script s'instal·larà Dolphin a `/usr/local/bin/dolphin-emu`, llest per ser utilitzat mitjançant la comanda `dolphin-emu`.

B.3. Configuració de `config.txt` per fer overclocking

(Advertència: L'overclocking pot invalidar la garantia i causar inestabilitat si no es refrigera bé.)

El següent és el codi afegit a l'arxiu `config.txt` per a aplicar l'overclocking utilitzat:

```
arm_freq=2000
gpu_freq=800
over_voltage=6
```

Amb això, la CPU i GPU aniran a freqüències més altes que les de fàbrica. Cal monitorar la temperatura amb `vcgencmd measure_temp` o, si no funciona, amb `cat /sys/class/thermal/thermal_zone0/temp` (aquest segon funciona bé al PuTTY).