

# Preservación y recreación de videojuegos retro con

# Software Libre

Erik Girón

<https://kreigiron.github.io/flisol-2023>

# ¿Para que preservar?

## Videojuegos como expresión artística

- Soundtrack
- Vídeo
- Historia
- Todo combinado

**FINAL FANTASY**  
©  
ファイナルファンタジー

# ¿Para qué preservar?

## Videojuegos como logro técnico

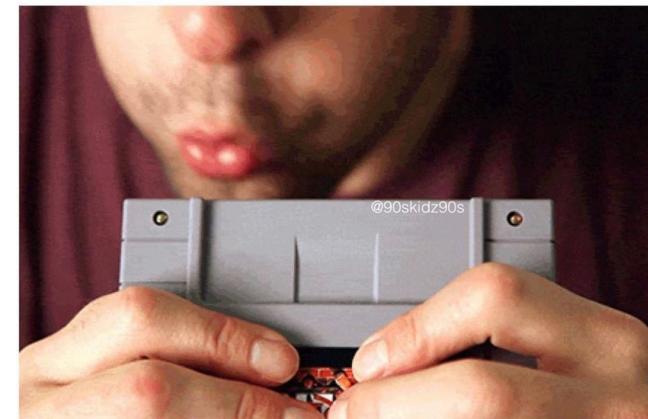


- Piezas unicas escritas a mano.
  - Efectos visuales, Musica, Interaccion.
- Ejecucion en hardware obsoleto o fuera de produccion.
  - Sensible al "ritmo" o velocidad de hardware para una correcta reproduccion.
- Plataformas o tiendas en linea privativas del fabricante.

# ¿Para qué preservar?

Nostalgia

Me trying to fix a problem in my life



# Cómo preservarlo?



## Hardware (Consolas y dispositivos físicos):

- Ingenieria inversa del hardware.
- Recreacion por el uso uso de especificaciones abiertas.

# Cómo preservarlo?

## Software (Juegos)

- Medios de almacenamiento vulnerables
  - cartucho
    - Hardware de lectura no disponible u obsoleto
  - diskettes
    - Deterioro magnético
  - CD y medios ópticos
    - Deterioro ambiental



# Cómo preservarlo?

## Proyectos de preservación

- Redump project - <http://redump.org/>
  - Guias y herramientas para
    - extraer - dump
    - comparar 'checksum'
    - compartir - share

## **Grupos de preservación de imágenes de juego**

- Gaming Alexandria
  - <https://www.gamingalexandria.com/wp/>
- Videogame preservation collective
  - <https://www.preservegames.org/>
- Archive.org

# Como recrearlos?

## Recreación por hardware

- Hardware original y clones
  - CRT
    - Upscalers por hardware
      - Retrothink
      - Framemeister
    - Open source
      - OSSC
  - Repro carts



# Como recrearlos?

## Recreación por Software: port FOSS

- Port multiplataforma por liberacion de codigo del fabricante liberados bajo una licencia FOSS



# Como recrearlos?

Recreación por Software:  
port FOSS

Doom

- <https://github.com/id-Software/DOOM>

# Como recrearlos?

## Recreación por Software: port FOSS

### Quake

- Quake <https://ioquake3.org/>
  - OpenArena  
<https://github.com/OpenArena/>



# Recreación por Software

## Ingeniería inversa

- Port multiplataforma por reverse engineering liberados bajo una licencia FOSS
  - <https://www.retroreversing.com/>



## Recreación por Software

### Ingeniería inversa

- Super Mario 64 -  
<https://github.com/n64decomp/sm64> - Licencia CC0

# **Recreación por Software**

## **Ingeniería inversa**

- Zelda Ocarina of Time  
<https://github.com/HarbourMasters/Shipwright>

# Recreación por Software

## Ingeniería inversa

- Otros ports
  - Super Metroid - <https://github.com/snesrev/sm> - Licencia MIT
  - Zelda - A link to the past <https://github.com/snesrev/zelda3>
  - Diablo - <https://github.com/diasurgical/devilutionX> Licencia SUL
  - entre otros

# Recreación por Software

## Emulación

- LLE para consolas antiguas.
  - carecen de OS regularmente
  - Programadas a bajo nivel
- HLE para consolas de 5ta + generacion
  - Tienen OS
  - Programadas a alto nivel con SDKs proveidos

# Ejemplos sobresalientes de emuladores FOSS

- Zsnes (obsoleto)
  - snes
  - Programado en ASM x86
  - Liberado en 2000 bajo la GPL
    - <https://sourceforge.net/projects/zsnes/>
    - <https://github.com/xyproto/zsnes> (fork)



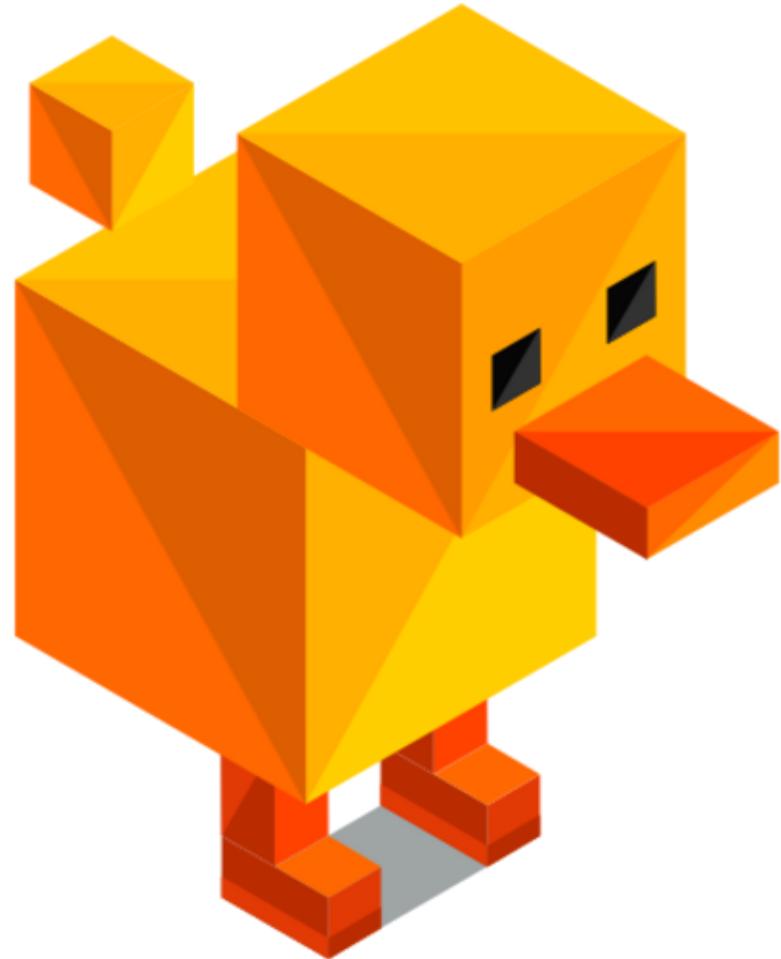
## Ejemplos sobresalientes de emuladores FOSS



- BSnes/Higan
  - snes y multiconsola (higan)
  - C/C++
  - Enfocado en exactitud
  - Licencia GPLv3
    - <https://github.com/bsnes-emu/bsnes>
    - <https://github.com/higan-emu/higan>

# Ejemplos sobresalientes de emuladores FOSS

- Duckstation
  - PSX
  - C++
  - Mejoras tecnicas a la consola
    - anti alias
    - z buffer emulation
  - Licencia GPLv3
    - <https://github.com/stenzek/duckstation>



# Ejemplos sobresalientes de emuladores FOSS

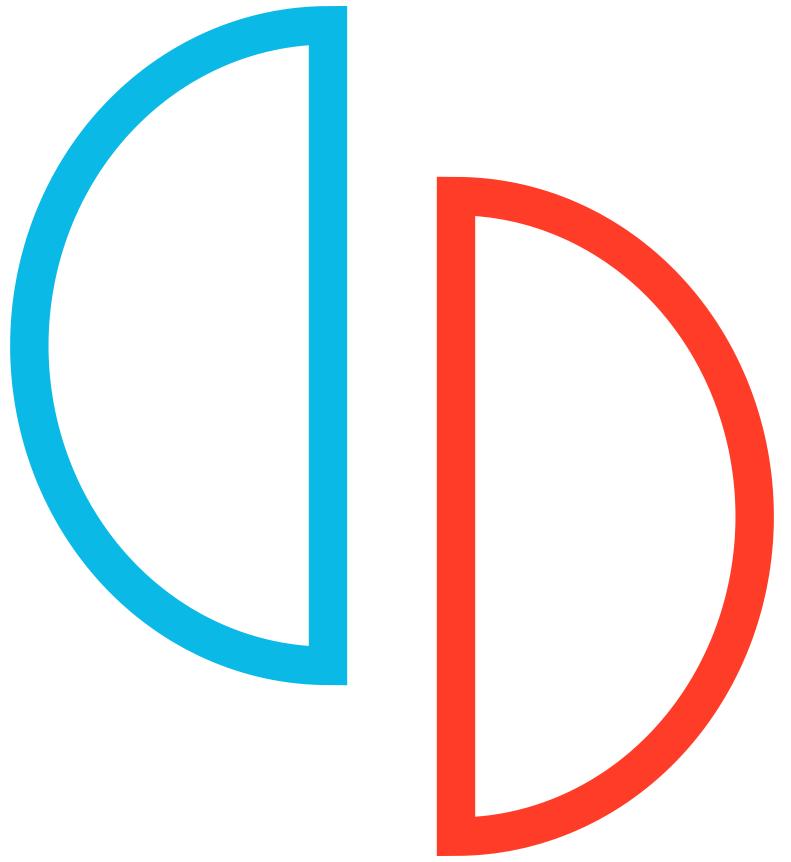


- Dolphin
  - GameCube/Wii
  - C++
  - Mejoras tecnicas a la consola
    - HD, UHD
  - Con mucha actividad de desarrollo
  - Licencia GPLv2+
    - <https://github.com/dolphin-emu/dolphin>

# Ejemplos sobresalientes de emuladores FOSS

- RPSC3
  - PS3
  - C++
  - 60%+ de compatibilidad
  - Licencia GPLv2
    - <https://github.com/RPCS3/rpcs3>



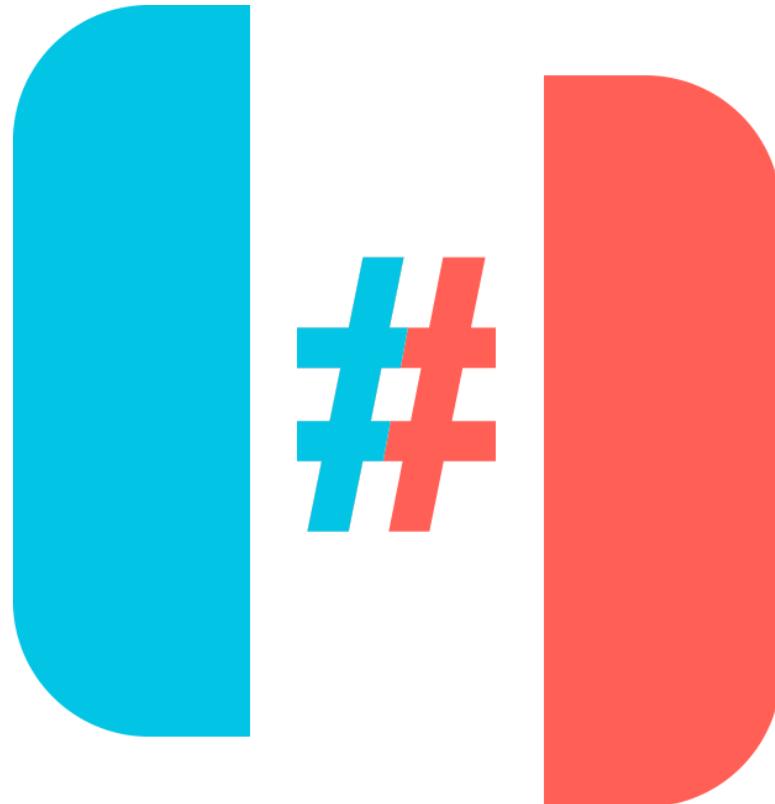


## Ejemplos sobresalientes de emuladores FOSS

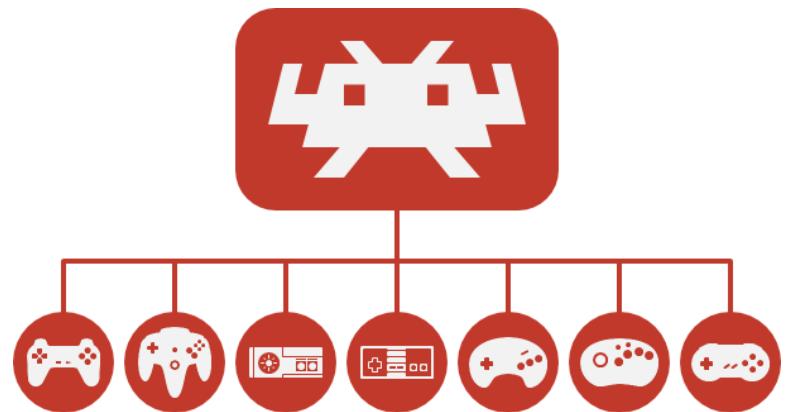
- Yuzu
  - Switch
  - C
  - Corre en SteamDeck!
  - GPLv3
    - <https://github.com/yuzu-emu/yuzu>

# Ejemplos sobresalientes de emuladores FOSS

- Ryujinx
  - Switch
  - C# (!)
  - Alternativa a Yuzu
  - MIT
    - <https://github.com/Ryujinx/Ryujinx>



## Ejemplos sobresalientes de emuladores FOSS

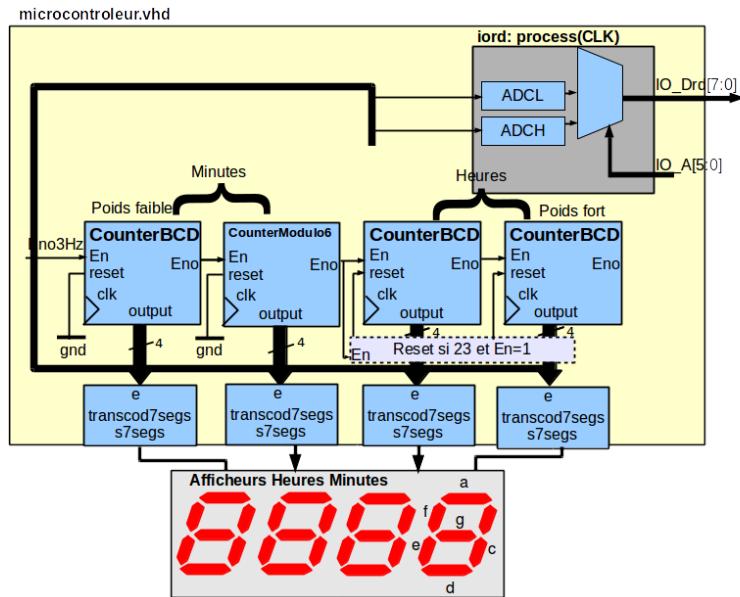


- Plataformas libretró
  - Retroarch
  - Lakka <https://www.lakka.tv/>
  - Retropie
  - Recalbox
  - Emuelec

# Recreacion por FPGA:

- Que es un FPGA?
  - Hardware programable
    - Industria
    - Medicina
    - Videojuegos





## Recreacion de consolas por FPGA

- No es Emulacion, es recreacion del hardware en el FPGA
- Nucleos intercambiables segun el hardware.

# Proyectos de recreacion FOSS por FPGA

- MisT
  - Cyclone EP3C25 FPGA
  - nucleos desde Atari ST hasta Amiga
- MisTER
  - FPGA 'DE10-Nano' modular
  - nucleos desde PDP-11 hasta PSX
  - Activo



# Conclusiones

- 4 principios del software libre
  - Como manera de recrear y ejecutar aplicaciones y juegos de plataformas obsoletas
    - Plataforma de ejecucion de dominio publico
  - Como medio de aprendizaje de tecnicas y optimizacion a bajo nivel
    - Desarrollo colaborativo
  - Como plataforma de distribucion de las que antes eran tecnologias privativas
    - e.g. MiSTer project
  - Como manera de mejorarlo continuamente
    - e.g. upscalers(ossC), overclocking

## **Dudas, comentarios**

# Gracias

Created by Erik Giron ([@kreig](#))