# Arquitecturas y Entornos de desarrollo para Videoconsolas

# Introducción

Grado de Ingeniero en Informática
Escola Tècnica Superior d'Enginyeria Informàtica
Curso 2020/2021

# Objetivos

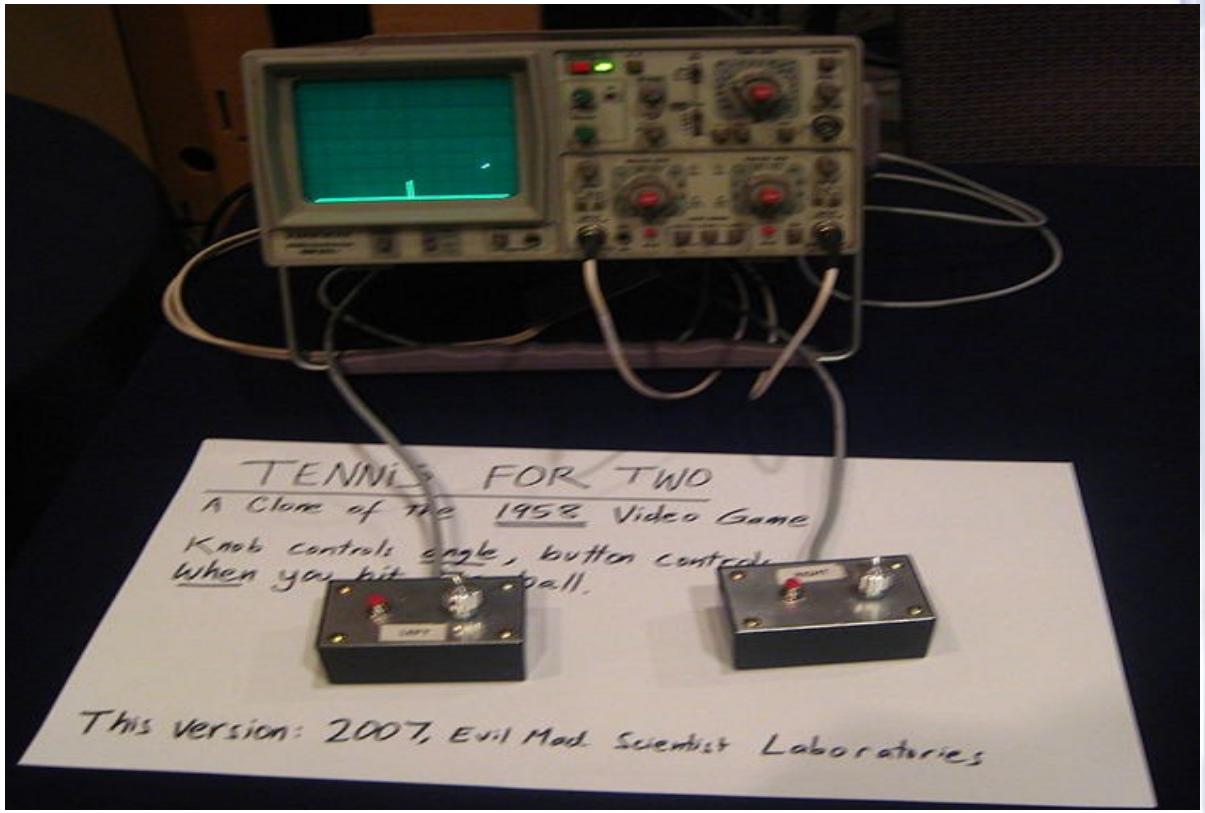
- Situar el conjunto de videoconsolas actuales en el desarrollo cronológico de estos dispositivos
- Caracterizar los dispositivos portables frentre a los fijos y definir posibles líneas de cambios en el sector
- Repasar el ciclo de desarrollo clásico de una aplicación y definir el concepto de desarrollo cruzado
- Instalar un entorno de desarrollo para una plataforma de videoconsola portable sobre GNU/Linux

# Índice

- Definición
- Perspectiva histórica
  - Línea de tiempos
  - Formatos existentes
  - Nuevas tendencias en consolas
- *Desarrollo para videoconsolas*
  - *Arquitectura y SDK*
- *Práctica asociada*
- Bibliografía

# Definición

- Juego electrónico (*electronic game*)
  - *Cathode-ray tube amusement device* (1948)
  - *Tennis for two* (1958)

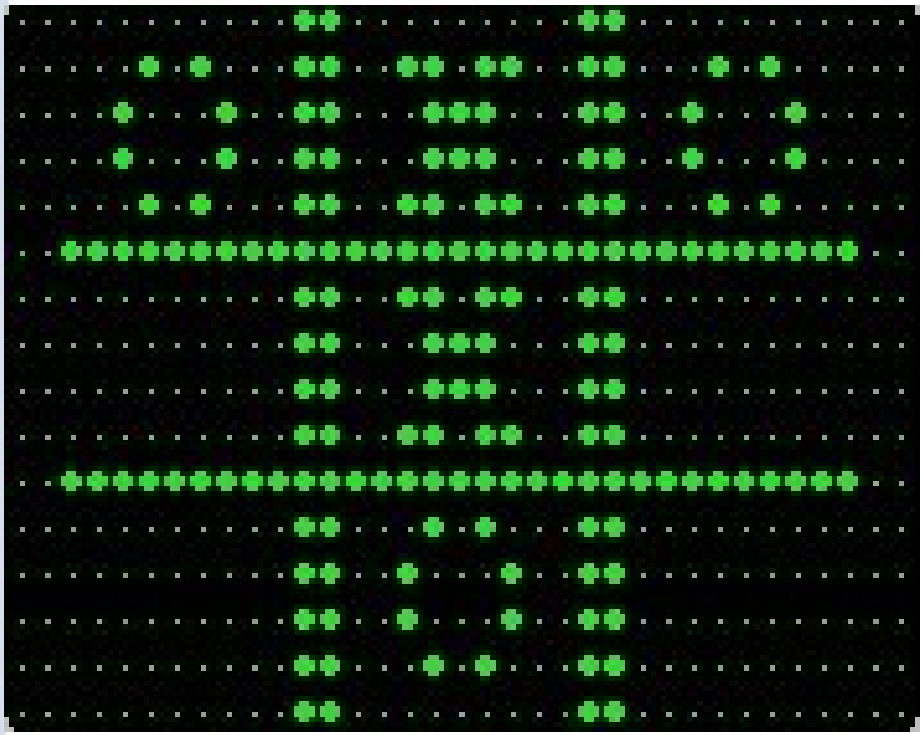


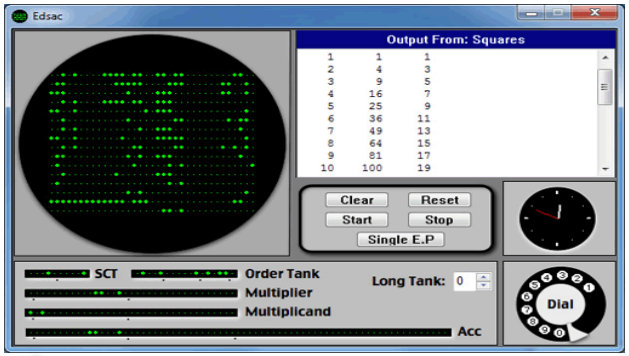


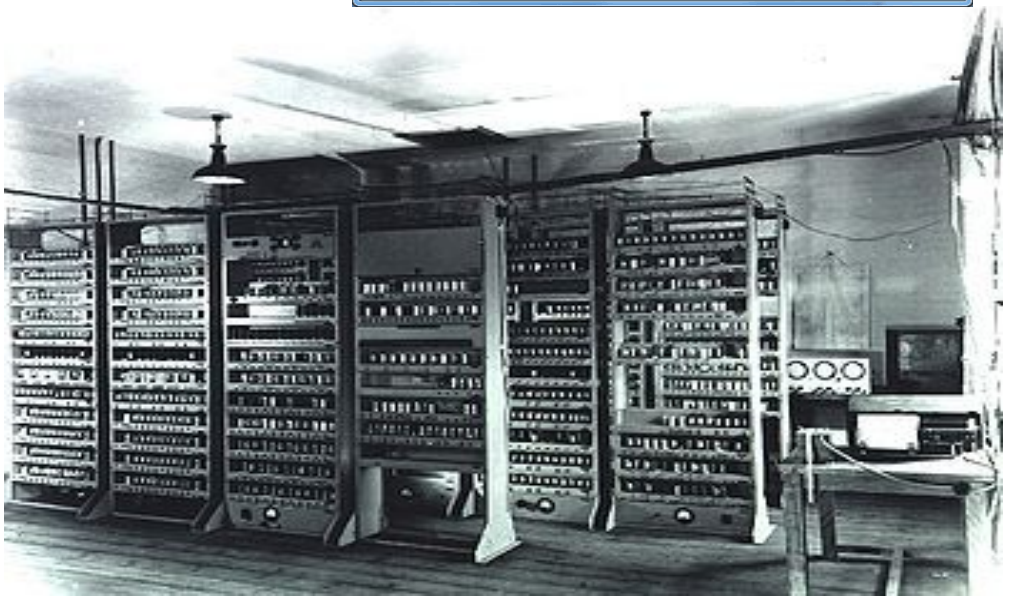
  - Simon (1978) ← Ralph Henry Baer y Howard J. Morrison

# Definición

- Juego de computador (*computer game)*
- OXO. ← A. S. “Sandy” Douglas (1952)
  - EDSAC
  - Sin animación



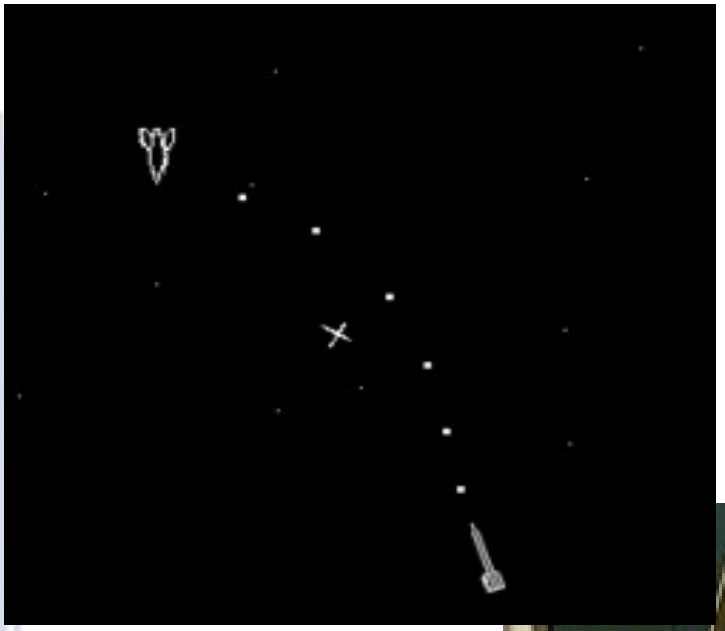


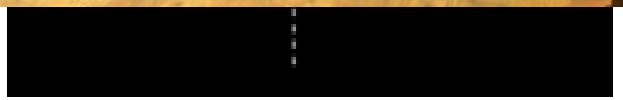



Fuente: <http://es.wikipedia.org/wiki/OXO>, <http://www.dcs.warwick.ac.uk/~edsac/>

# Definición

- SpaceWar (1961) ← PDP-1
  - Martin Graetz, Alan Kotok y Steve Russell

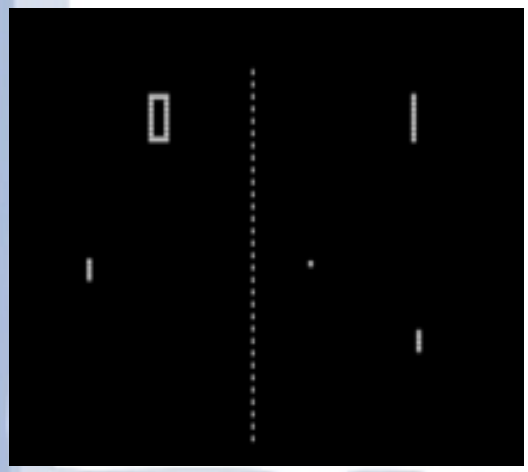




Fuente: <https://en.wikipedia.org/wiki/PC_game#History>, <https://en.wikipedia.org/wiki/Spacewar_%28video_game%29>, <https://en.wikipedia.org/wiki/PDP-1>

# Definición

- Videoconsola
  - Dispositivo doméstico dedicado
    - Salida vídeo Analógico o Digital
      ↔ pantallas vectoriales (juegos de comp.)
    - Controles especializados
    - Aplicaciones lúdicas
  - Tennis Game / Pong (1972)
    - Ralph Baer (Magnavox Odyssey)
    - Allan Alcorn ← Nolan Bushnell
      y Ted Dabney (Atari)



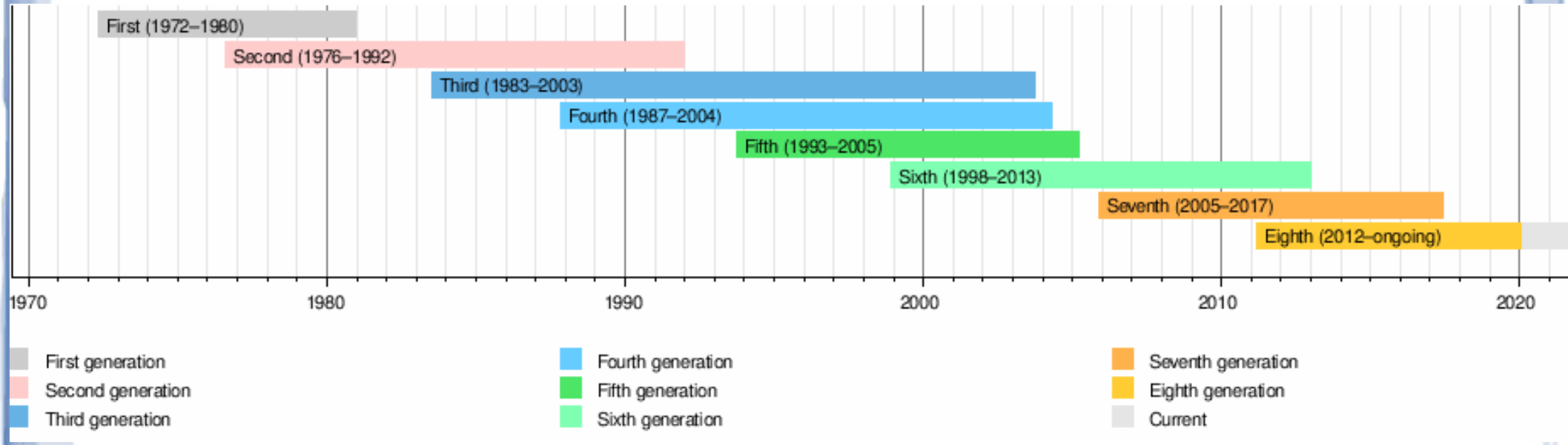

Fuente: <http://en.wikipedia.org/wiki/PONG>, <https://es.wikipedia.org/wiki/Magnavox_Odyssey>

# Perspectiva histórica

- Línea de tiempos
- Formatos existentes
  - Consolas portables
  - Consolas fijas
- Tendencias en consolas
  - Expandir funcionalidad
  - Flexibilidad hard. & soft.
    - Arquitecturas abiertas
    - Sistema operativo Android

# Perspectiva histórica (II)

- Línea de tiempos

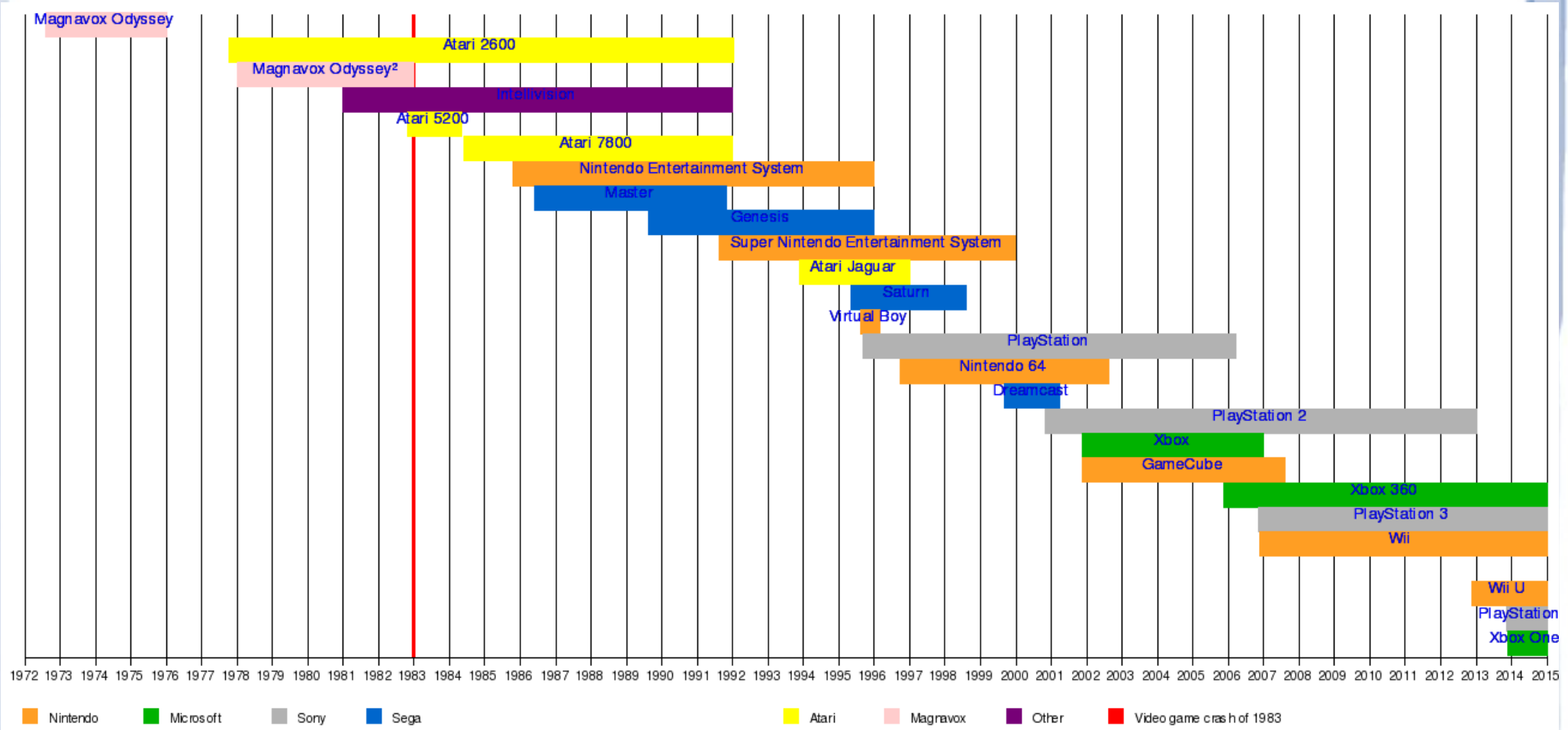



  - Acceso a Internet
    - Xbox Live (2002), Steam (2003)
    - Virtual Console (2006), Sony's PlayStation Network (PSN, 2006),
    - Nintendo Network (3DS y Wii/U2012), Nintendo Switch Online (2018)

Fuente: <http://en.wikipedia.org/wiki/Timeline_of_video_game_console_releases_in_North_America>

# Perspectiva histórica (III)

- Línea de tiempos




Fuente: <http://en.wikipedia.org/wiki/Timeline_of_video_game_console_releases_in_North_America>

# Perspectiva histórica (y IV)

- Línea de tiempos
- Formatos existentes
  - Consolas portables




# Historia

- Línea de tiempos
- Formatos:
  - Consolas portables
  - Consolas fijas



Imágenes tomadas de la Wikipedia

# Tendencias en consolas (II)

- Ampliar funcionalidades
  - *Home cinema / Home theather*
    - *Servidor de contenidos*
    - *Reproductor de formatos actualizable y lector de soportes*
    - *Más opciones: configuración de listas, obtención de información (letras, reparto de actores, ...)*
    - *Acceso a otras aplicaciones (Web, correo, videoconferencia, …) y control domótico*
  - Realidad Aumentada
- Flexibilidad en *hardware* y *software*

# Tendencias en consolas (III)

- Minicomputadores
  - OpenPandora
    - 2010
    - Linux – PC
    - 499,00 €


  - Pyra <https://pyra-handheld.com>
    - 2008 .. 2021, >= 600€

# Tendencias en consolas (IV)

- *Ouya*
  - 2012
  - Android 4.1

  - 2015 → *Razer*

https://es.wikipedia.org/wiki/Ouya

# Tendencias en consolas (V)


Fuente: http://shield.nvidia.com/

- NVIDIA – Shield
- Especificaciones
  - Controles
    - Dual analog joysticks,
      D-pad.
      Left/right analog triggers, Left/right bumpers, A/B/X/Y buttons, Volume control
    - Android Home and Back buttons, Start button, NVIDIA power/multi-function button
  - Audio
    - Integrated Stereo Speakers with Built-in Microphone
    - 3.5mm stereo headphone jack with microphone support
  - Display
    - 5 inch 1280x720 (294 ppi) Multi-Touch Retinal Quality Display

# Tendencias en consolas (VI)

- NVIDIA – Shield
- Especificaciones (cont.)


Fuente: http://shield.nvidia.com/

  - Procesador
    - NVIDIA Tegra 4 Quad Core Mobile Processor with 2GB RAM
  - Almacenamiento
    - 16GB Flash Memory
  - Conectividad
    - Wireless 802.11n 2x2 Mimo 2.4 GHz and 5 GHz Wi-Fi, Bluetooth 3.0, GPS, Mini-HDMI, output, Micro-USB 2.0 y MicroSD storage slot
  - Sensores
    - Movimiento (giroscopio de 3 ejes) y posición (acelerómetro de 3 ejes)

# Tendencias en consolas (VII)

- *Archos GamePad*

# Presente y ¿futuro?

- Plataformas híbridas
  - N. *Switch (*NX)
- Plataformas streaming
  - Google *Stadia*



Fuente: <https://www.nintendo.es/Nintendo-Switch/Nintendo-Switch-1148779.html>, <*https://labo.nintendo.com/kits/variety-kit/*>, <https://store.google.com/product/stadia?>

# *Desarrollo para videoconsolas*

- No piratería
- *Simon van de Berg.*
  *Running Nintendo DS homebrew*


  - Pirating of software is something I do not approve of.
    - Pirating is a term used for running official games you do not own, or do own, but are not allowed to play in some way by law.
    - Pirating is often associated with homebrew.
    - Homebrew is creating and sharing programs made by yourself and/or others for free. This means that no business is attached to the software. Please keep in mind that this also means that there is no support for the software, unless someone will provide it for you.
    - Homebrewing is a really great development. Just look at what great applications the DS has been given by the homebrew community! The capabilities of the Nintendo DS are explored in many ways. Some great applications I frequently use are programs such as DSlinux, and DSorganize. But there are also great Homebrew games like ScummVM, Omalone, etcetera.

# *Desarrollo para videoconsolas (II)*

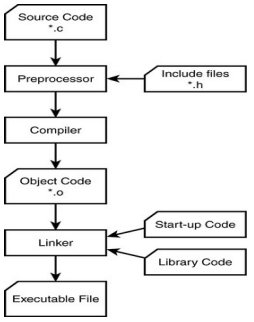
- SDK

  ¿genera ejecutable?

  - Oficial → *Your adventure … awaits* 
  - Homebrew (no oficial) → devkitPro 
- *Middelware*
  - *Unity3D* 
  - *Unreal* 
  - GameMaker Studio 2 

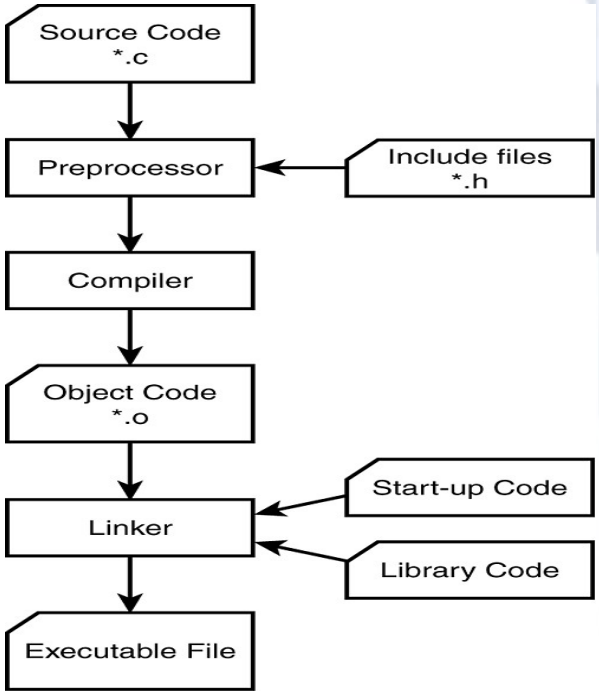
# *Desarrollo para videoconsolas (III)*

- No piratería
- *Desarrollo para computadores, en general*
  - *Arquitectura de la plataforma*
    - *Características*
    - *Herramientas*
      - *Compilador, enlazador, ensamblador, depurador*
  - *Software Developmet Kit (SDK)*
    - *Oficial*
    - *¿Alternativas?*
- Desarrollo *Homebrew*



# Desarrollo para videoconsolas (VI)

- Desarrollo para computadores, en general
  - Del código fuente al ejecutable
    - Dependencia
      - Hardware destino
      - Modelo de ejecución:
        - Compilado
        - Interpretado




- Soporte de librerías

# *Desarrollo para videoconsolas (V)*

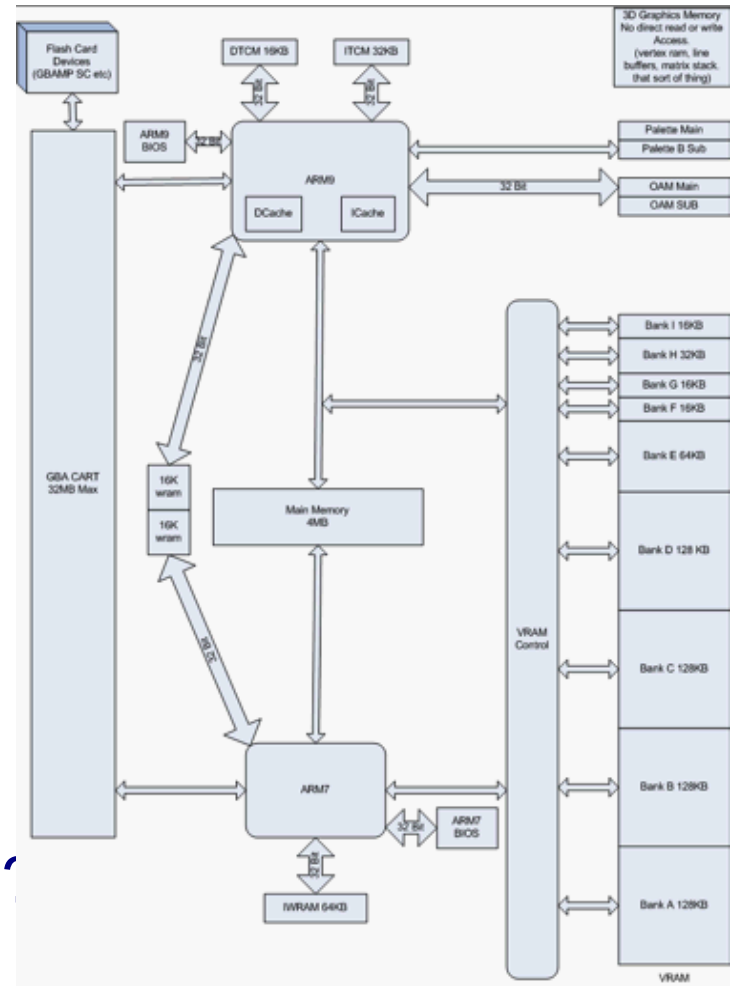
- *No piratería*
- Desarrollo *Homebrew*
  - Acuñado en videojuegos (consolas, …)
  - Informal y misterioso
- *devkitPro*
  - *Toolchain C/C++*
    +
  - *SDK*

# Desarrollo para videoconsolas (VI)

- *devkitPro*
  - *Toolchain C/C++*
    +
  - *SDK*
- *Administradores*
  - *Michael Noland (joat)*
  - *Jason Rogers (dovoto)*
    - *http://jason.drunkencoders.com/*
  - *Dave Murphy (WinterMute)*
    - *https://plus.google.com/102196840057650751944/about*
  - *Jordan (smealum)* <*http://smealum.net*>

# *devkitPro: Toolchain*

- Arquitectura
  - PSP
  - Wii
  - y NDS, GBA, GP32
- Ejecutable
  - 2 procesadores
    - Ejecutables: ¿uno, dos?
  - ¿Cómo se obtiene?



# *devkitPro: Toolchain (II)*

- Arquitectura Hardware NDS (ARM)
- Ejecutable: ¿cómo se obtiene?
  - *Toolchain*
    - C/C++, fpc4nds, MicroLuaDS, DSPhyton, DSBasic
    - Compilador / Enlazador
    - Ensamblador
    - Depurador
  - SDK
  - Emulador
  - Instalación

# *devkitPro: Toolchain (III)*

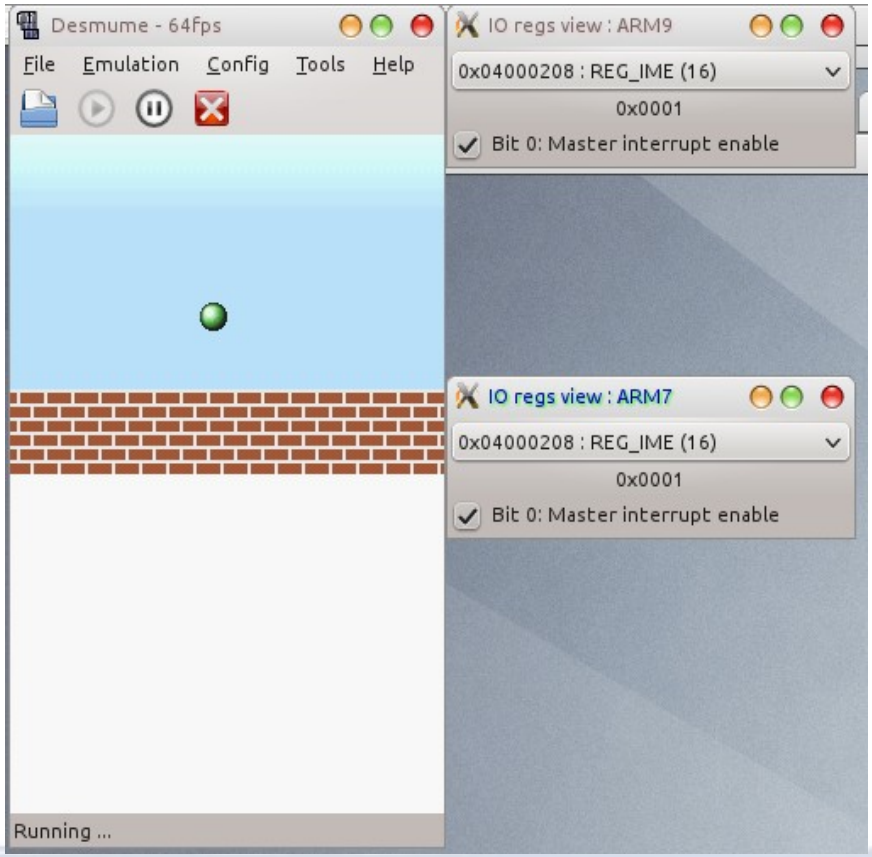
- Ejecutable: ¿cómo se obtiene?
  - *Toolchain*
  - SDK
    - *devkitARM*
      - *NDS → libnds (libgba, libmirko), libfilesystem, libfat, libdswifi, libmm*
        - ~~*PALib,*~~ *Nflibm, dsgmLib*
      - *N3DS → libctru*
    - *devkitPSP → libpsp**
    - *devkitPPC → libogc*
  - *Emulador*
  - *Instalación*

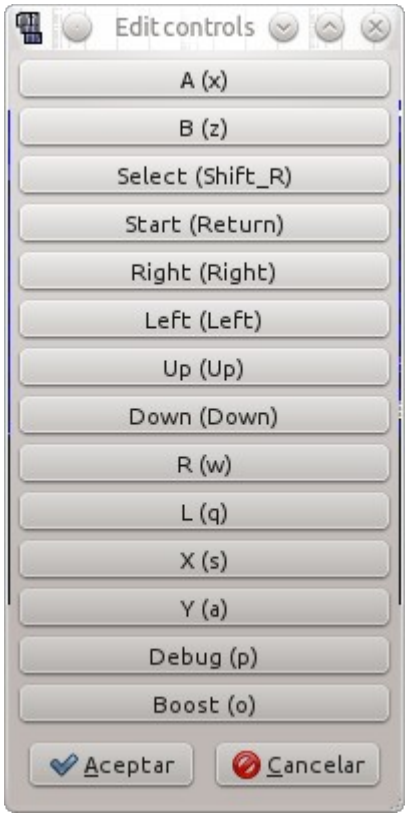
# *devkitPro: Toolchain (VI)*

- Arquitectura Hardware
- ¿Ejecutable?
  - *Toolchain*
  - SDK
  - *Emulador*
    - *NDS: DesMuME, no$gba, IdeaS, Dualis, …*
    - *3DS: Citra, 3dmoo, TronDS, 3DSEmulator, ...*
  - *Instalación*

# Ejecutar en emulador NDS

- Ejecutar
  - $ desmume bouncing.nds



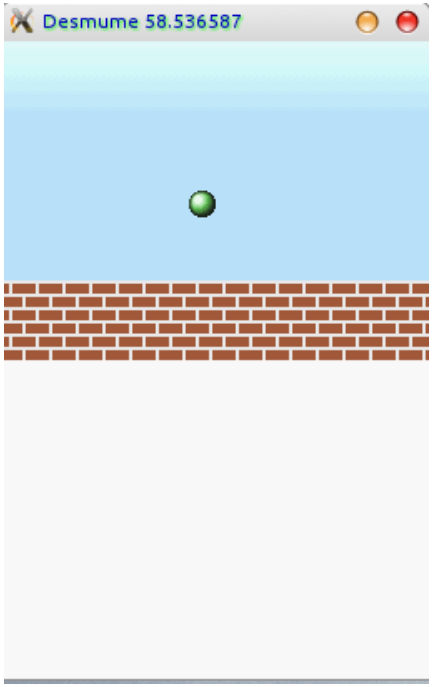
# Ejecutar en emulador NDS (II)

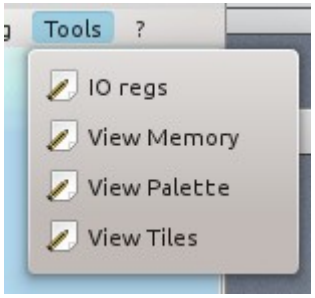
- Depurar con GDB
  - $ desmume-cli --arm9gdb=55555 file.nds
  - $ arm-eabi-gdb file-arm9.elf

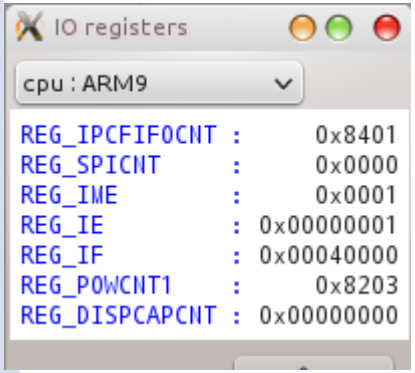
```
(gdb) target remote localhost:55555
Remote debugging using localhost:55555
0x02000000 in _start ()

(gdb) b main
```

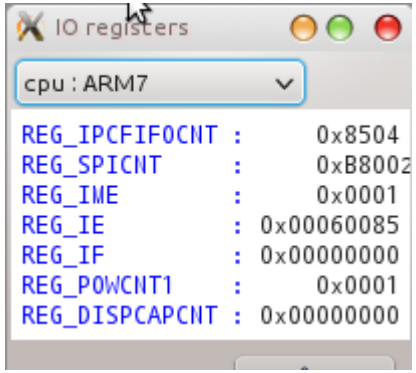
# Ejecutar en emulador NDS (III)

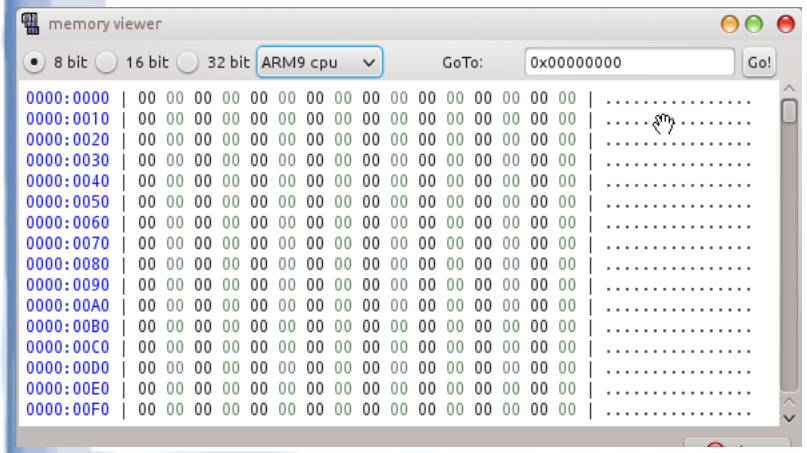
- $ desmume-glade
  - “Tools”

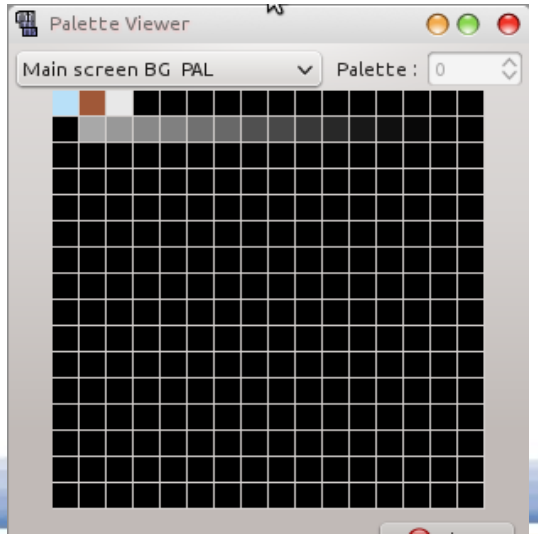


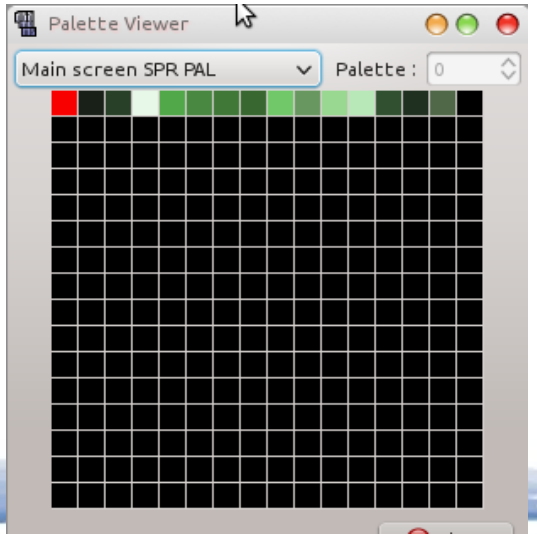



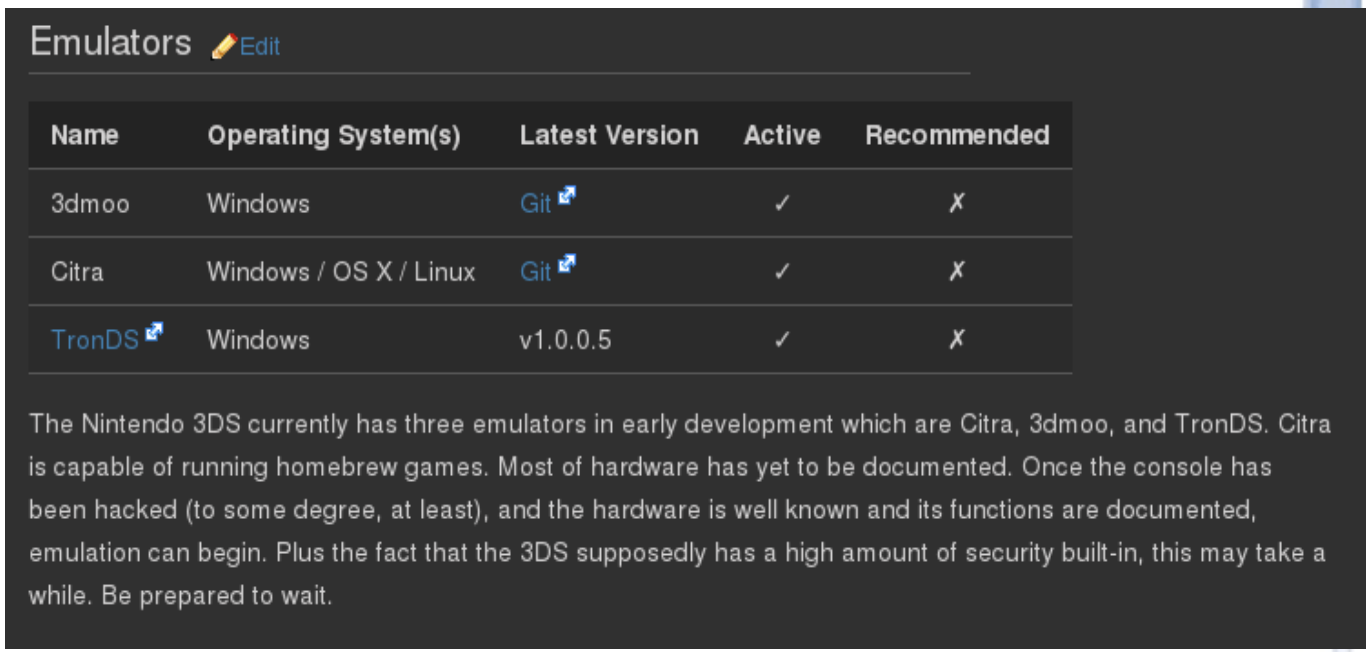



# Ejecutar en emulador N3DS

- Posibilidades



Emulators Edit

| Name | Operating System(s) | Latest Version | Active | Recommended |
|---|---|---|---|---|
| 3dmoo | Windows | Git | ✓ | ✗ |
| Citra | Windows / OS X / Linux | Git | ✓ | ✗ |
| TronDS | Windows | v1.0.0.5 | ✓ | ✗ |

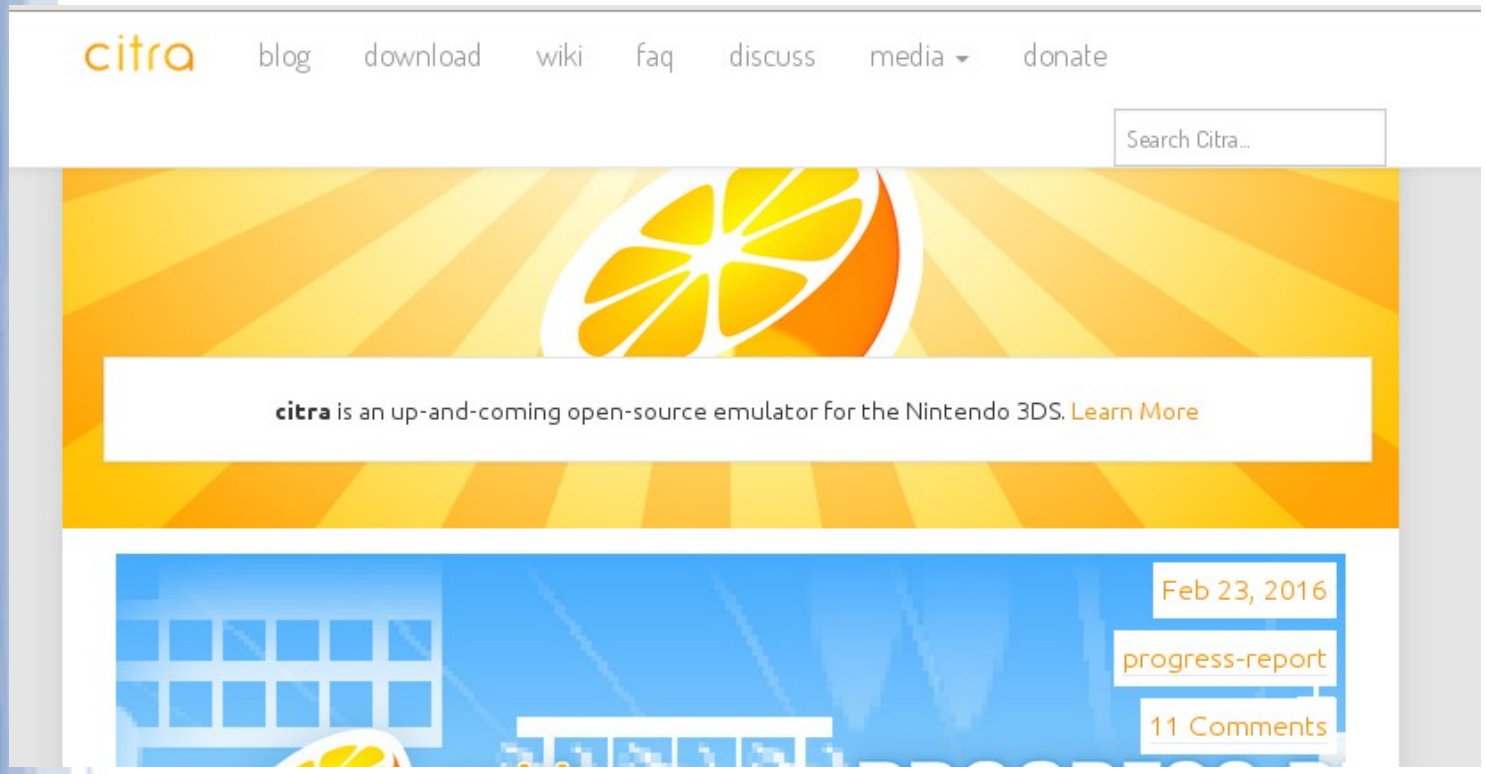
The Nintendo 3DS currently has three emulators in early development which are Citra, 3dmoo, and TronDS. Citra is capable of running homebrew games. Most of hardware has yet to be documented. Once the console has been hacked (to some degree, at least), and the hardware is well known and its functions are documented, emulation can begin. Plus the fact that the 3DS supposedly has a high amount of security built-in, this may take a while. Be prepared to wait.

  - 3dmoo <https://github.com/plutooo/3dmoo/>
  - Citra <https://citra-emu.org/>
  - TronDS <http://trondsemu.byethost15.com/>
  - 3DS Emulator (e3DSx) <http://3dsemulatorx.net/>
    - “PC, Android and Mac”
  - ...

Imagen obtenida de <http://emulation-general.wikia.com/wiki/Nintendo_3DS_emulators>

# Ejecutar en emulador N3DS (II)

- Citra



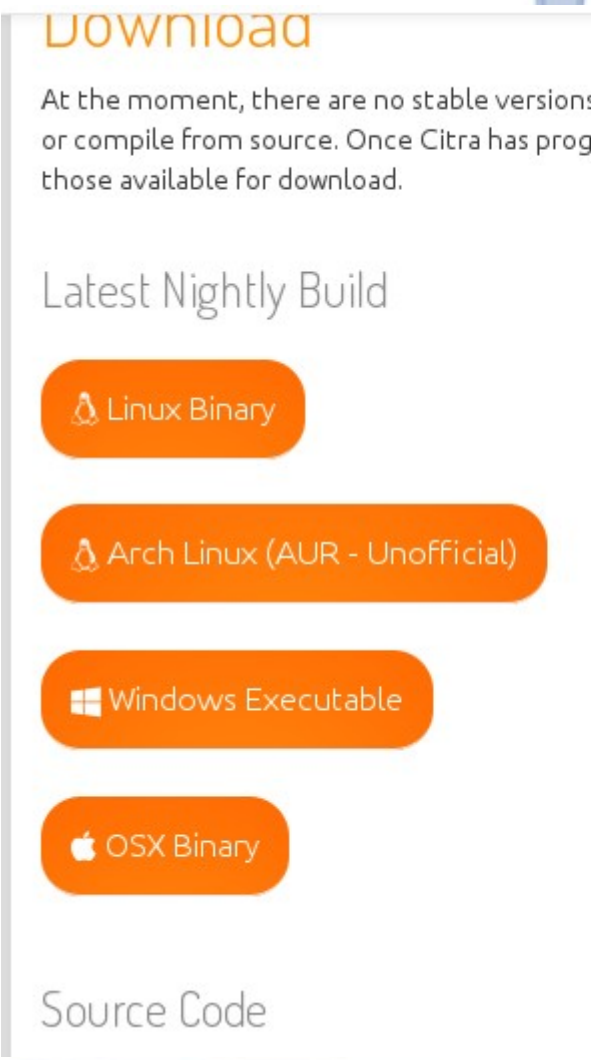
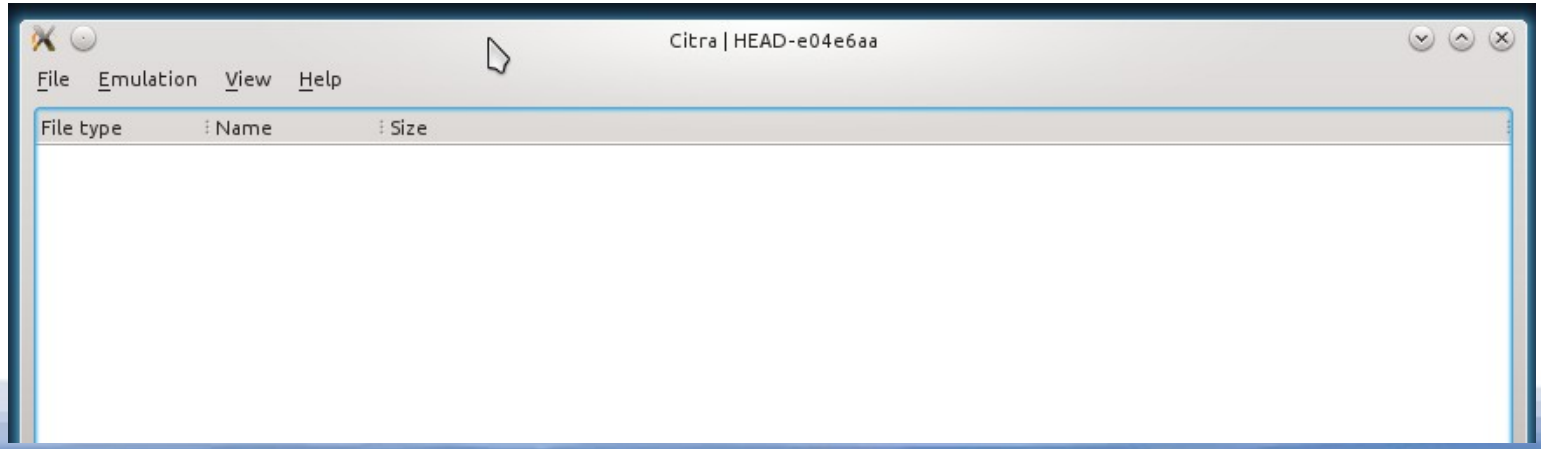




Imagen obtenida de <https://citra-emu.org/>

# Ejecutar en emulador N3DS (III)

- Citra
  - citra-latest-linux-amd64.tar.xz

```
$ tar -xf citra-latest-linux-amd64.tar.xz

$ ls -l citra-20160224-e04e6aa-linux-amd64
total 12M
-rwxr-xr-x 1 magustim disca-upvnet 6.6M Feb 25 03:18 citra-qt
-rwxr-xr-x 1 magustim disca-upvnet 5.4M Feb 25 03:18 citra

$ ./citra-qt
```



# Ejecutar en emulador N3DS (VI)

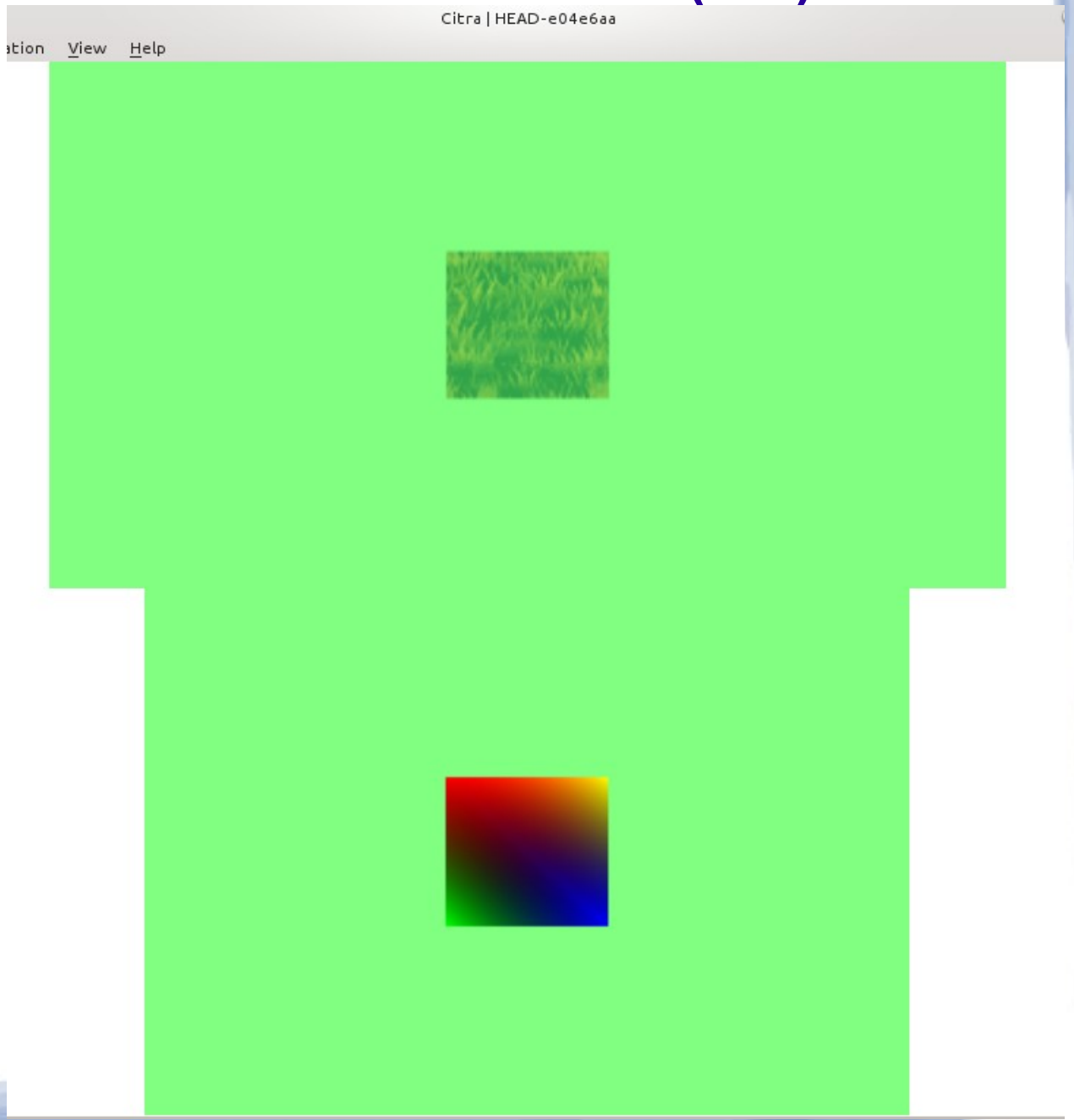
- 3ds-examples-master/ ← GitHub
  - templates
  - graphics (gpu, printing, bitmap)
  - audio
  - input
  - camera (image, video) y qtm
  - sdmc, romfs
  - **nfc**
  - **network** (http, udp, ...)
  - app_launch, time, get_system_language, threads, libapplet_launch

# Ejecutar en emulador N3DS (V)

- both-screen-text.3dsx

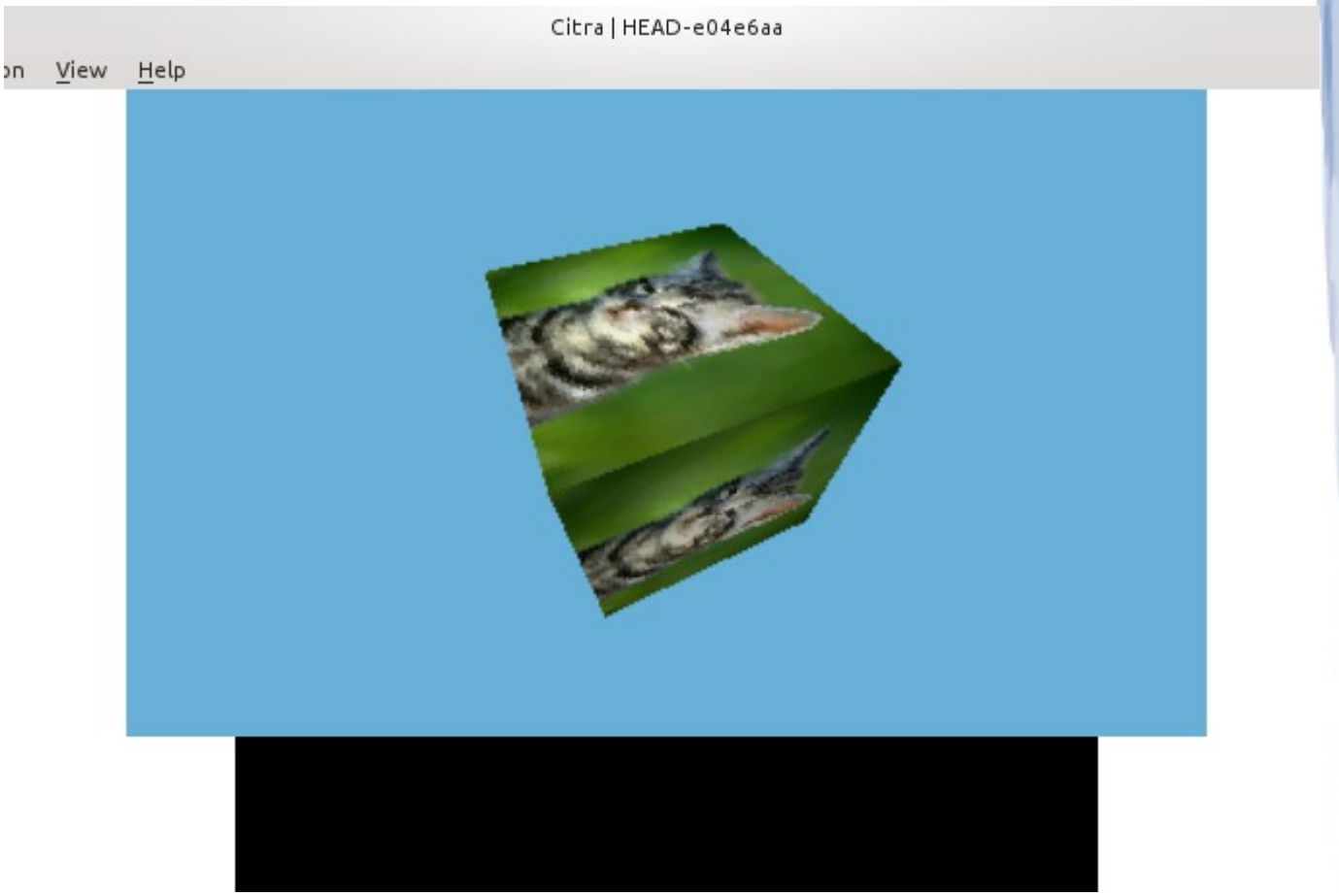
# Ejecutar en emulador N3DS (VI)

- gputest.3dsx



# Ejecutar en emulador N3DS (y VII)

- textured_cube.3dsx



# *Práctica*

- Instalación
  - devkitPro
  - libnds
- Compilación ejemplos *libnds*
- Ejecución
  - Emuladores
  - Equipos reales

# Hola Mundo!!!! en NDS

```
#include <nds.h>
#include <stdio.h>

void main() {

    consoleDemoInit();

    printf("\n Hola, mundo\n");

}
```

# devkitpro/examples/nds/hello_world/

```
/*---------------------------------------------------------------------------
 $Id: main.cpp,v 1.13 2008-12-02 20:21:20 dovoto Exp $
 Simple console print demo
 -- dovoto
*/
#include <nds.h>
#include <stdio.h>
volatile int frame = 0;

void Vblank() {
 frame++;
}
```

```
//---------------------------------------------------------------------------
int main(void) {
//---------------------------------------------------------------------------
 touchPosition touchXY;
 irqSet(IRQ_VBLANK, Vblank);
 consoleDemoInit();
 iprintf(" Hello DS dev'rs\n");
 iprintf(" \x1b[32mwww.devkitpro.org\n");
 iprintf(" \x1b[32;1mwww.drunkencoders.com\x1b[39m");

 while(1) {

 swiWaitForVBlank();
 touchRead(&touchXY);
 // print at using ansi escape sequence
  // \x1b[line;columnH
 iprintf("\x1b[10;0HFrame = %d",frame);
 iprintf("\x1b[16;0HTouch x = %04X, %04X\n",
              touchXY.rawx, touchXY.px);
 iprintf("Touch y = %04X, %04X\n", touchXY.rawy, touchXY.py);

 }
 return 0;
}
```

# devkitpro/examples/nds/hello_world/

```
/*---------------------------------------------------------------------------
 $Id: main.cpp,v 1.13 2008-12-02 20:21:20 dovoto Exp $
 Simple console print demo
 -- dovoto
*/
#include <nds.h>
#include <stdio.h>
volatile int frame = 0;

void Vblank() {
 frame++;
}
```

```
//---------------------------------------------------------------------------
int main(void) {
//---------------------------------------------------------------------------
 touchPosition touchXY;
 irqSet(IRQ_VBLANK, Vblank);
 consoleDemoInit();
 iprintf(" Hello DS dev'rs\n");
 iprintf(" \x1b[32mwww.devkitpro.org\n");
 iprintf(" \x1b[32;1mwww.drunkencoders.com\x1b[39m");

 while(1) {

 swiWaitForVBlank();
 touchRead(&touchXY);
 // print at using ansi escape sequence
  // \x1b[line;columnH
 iprintf("\x1b[10;0HFrame = %d",frame);
 iprintf("\x1b[16;0HTouch x = %04X, %04X\n",
              touchXY.rawx, touchXY.px);
 iprintf("Touch y = %04X, %04X\n", touchXY.rawy, touchXY.py);

 }
 return 0;
}
```

# $(DEVKITPRO)/examples/3ds/graphics/printing/hello-world/

```
/* Hello World example made by Aurelio Mannara for ctrulib ( 12/12/2014 21:00 UTC+1) */
#include <3ds.h>
#include <stdio.h>

int main(int argc, char **argv)
{
	gfxInitDefault();

	//Initialize console on top screen. Using NULL as the second argument tells the
	//console library to use the internal console structure as current one
	consoleInit(GFX_TOP, NULL);

	//To move the cursor you have to print "\x1b[r;cH"
	printf("\x1b[15;19HHello World!");

	printf("\x1b[29;15HPress Start to exit.");


...
	gfxExit();
	return 0;
}
```

# $(DEVKITPRO)/examples/3ds/graphics/printing/hello-world/

```
...
	printf("\x1b[29;15HPress Start to exit.");

	// Main loop
	while (aptMainLoop())
	{
		//Scan all the inputs. This should be done once for each frame
		hidScanInput();

		//hidKeysDown returns information about which buttons have been just pressed
		// (and they weren't in the previous frame)
		u32 kDown = hidKeysDown();

		if (kDown & KEY_START) break; // break in order to return to hbmenu

		// Flush and swap framebuffers
		gfxFlushBuffers();
		gfxSwapBuffers();

		//Wait for VBlank
		gspWaitForVBlank();
	}
	gfxExit();
...
```

# Bibliografía


- *History of video games* <http://en.wikipedia.org/wiki/History_of_video_games>
- devkitPro <http://devkitpro.org>
- Nintendo DS homebrew - Wikipedia
- DevScene <http://www.dev-scene.com/Main_Page>
- Marten Tonissoo. ThomasWorld. Baby Steps In Nintendo DS Homebrew Hacking
- 3DBrew <http://3dbrew.org/wiki>