Arquitecturas y Entornos de desarrollo para Videoconsolas

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

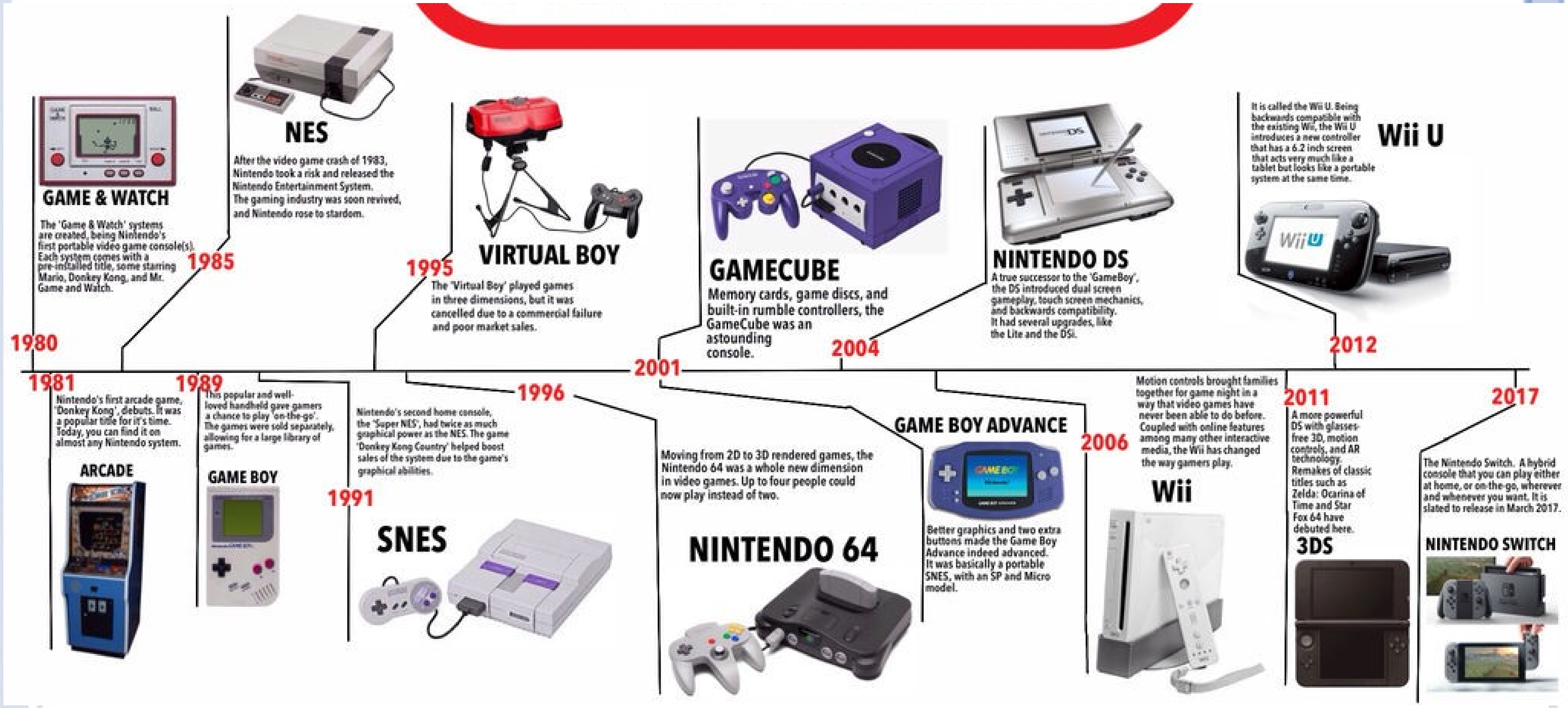
Objetivos

- Conocer la arquitectura hardware de la plataforma NDS
  - Identificar las rutas de datos e instrucciones
  - Reconocer el uso de los componentes de la arquitectura de los que se hace uso en las aplicaciones existentes
- Conocer la arquitectura hardware de las plataformas 3DS y Switch
- Aplicar las estrategias existentes a los propios desarrollos y los emuladores disponibles
- Conocer y saber dimensionar una aplicación para esta plataforma

Índice

- Introducción
  - Características *hardware* de las familias N
  - Desarrollo y ejecución de aplicaciones propias
- Arquitectura de la plataforma NDS
  - Arq. hardware de NDS y arq. software
  - Estructura de una aplicación NDS
- Arquitectura de la plataforma 3DS
  - Arq. hardware de 3DS y arq. software
  - Estructura de una aplicación 3DS
- Arquitectura de la plataforma *Switch*
  - Arq. hardware de NDS y arq. software
  - Estructura de una aplicación NDS

Características Hw de la familia Nintendo



1980
GAME & WATCH
The 'Game & Watch' systems are created, being Nintendo's first portable video game console(s). Each system comes with a pre-installed title, some starring Mario, Donkey Kong, and Mr. Game and Watch.

1981
Nintendo's first arcade game, 'Donkey Kong', debuts. It was a popular title for it's time. Today, you can find it on almost any Nintendo system.
ARCADE

1985
NES
After the video game crash of 1983, Nintendo took a risk and released the Nintendo Entertainment System. The gaming industry was soon revived, and Nintendo rose to stardom.

1989
This popular and well-loved handheld gave gamers a chance to play 'on-the-go'. The games were sold separately, allowing for a large library of games.
GAME BOY

1991
Nintendo's second home console, the 'Super NES', had twice as much graphical power as the NES. The game 'Donkey Kong Country' helped boost sales of the system due to the game's graphical abilities.
SNES

1995
VIRTUAL BOY
The 'Virtual Boy' played games in three dimensions, but it was cancelled due to a commercial failure and poor market sales.

1996
Moving from 2D to 3D rendered games, the Nintendo 64 was a whole new dimension in video games. Up to four people could now play instead of two.
NINTENDO 64

2001
GAMECUBE
Memory cards, game discs, and built-in rumble controllers, the GameCube was an astounding console.

GAME BOY ADVANCE
Better graphics and two extra buttons made the Game Boy Advance indeed advanced. It was basically a portable SNES, with an SP and Micro model.

2004
NINTENDO DS
A true successor to the 'GameBoy', the DS introduced dual screen gameplay, touch screen mechanics, and backwards compatibility. It had several upgrades, like the Lite and the DSi.

2006
Motion controls brought families together for game night in a way that video games have never been able to do before. Coupled with online features among many other interactive media, the Wii has changed the way gamers play.
Wii

2011
A more powerful DS with glasses-free 3D, motion controls, and AR technology. Remakes of classic titles such as Zelda: Ocarina of Time and Star Fox 64 have debuted here.
3DS

2012
It is called the Wii U. Being backwards compatible with the existing Wii, the Wii U introduces a new controller that has a 6.2 inch screen that acts very much like a tablet but looks like a portable system at the same time.
Wii U

2017
The Nintendo Switch. A hybrid console that you can play either at home, or on-the-go, wherever and whenever you want. It is slated to release in March 2017.
NINTENDO SWITCH

Imagen de la VincentWeir, "Nintendo Timeline"<https://www.deviantart.com/vincentweir/art/Nintendo-Timeline-650928953>

Características Hw de la familia Nintendo (II)

- Game Boy
  - *8-bit Sharp LR35902 (compat. Z80), 4,19MHz*

- *Game Boy Advance* (GBA)
  - ARM7TDMI, de 32 bits + Z80, soporte a la GameBoy clásica

- *Nintendo DS* (NDS)
  - Z80 → ARM7 33Mhz + ARM9 67Mhz ↑↑2D y motor 3D

- *GamePark 32* (GP32)
  - ARM920T + Tarjetas SD.

Imágenes obtenidas de la Wikipedia: <http://www.wikipedia.org/>

Características Hw de la familia Nintendo (III)

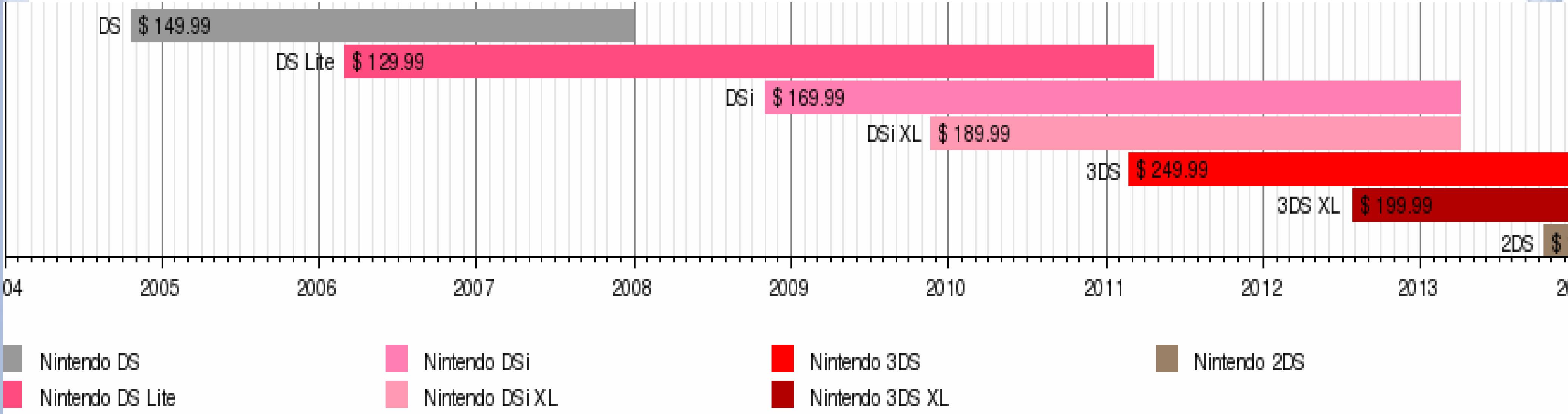
- *NDS* / *NDS Lite*
  - "*Lite*" y 4 niveles de brillo
- NDSi / NDSi XL
  - 2 cámaras
  - ↑↑prestaciones , tarjetas SD, ↓Slot2.
- 3DS / 3DS XL
  - Pantalla 3D
  - Acelerómetro, giróscopo, IR
- 2DS
- VC

Virtual Console™

Imágenes obtenidas de la Wikipedia: <http://www.wikipedia.org/> y del sitio web de Nintendo <http://www.nintendo.com/>

Características Hw de la familia Nintendo (VI)

- Disponibilidad actual
  - No se fabrican DS/DS Lite



  - Comprobar cambios en la arquitectura para fijar objetivos

Imágenes obtenidas de la Wikipedia: <http://www.wikipedia.org/> y del sitio web de Nintendo <http://www.nintendo.com/>

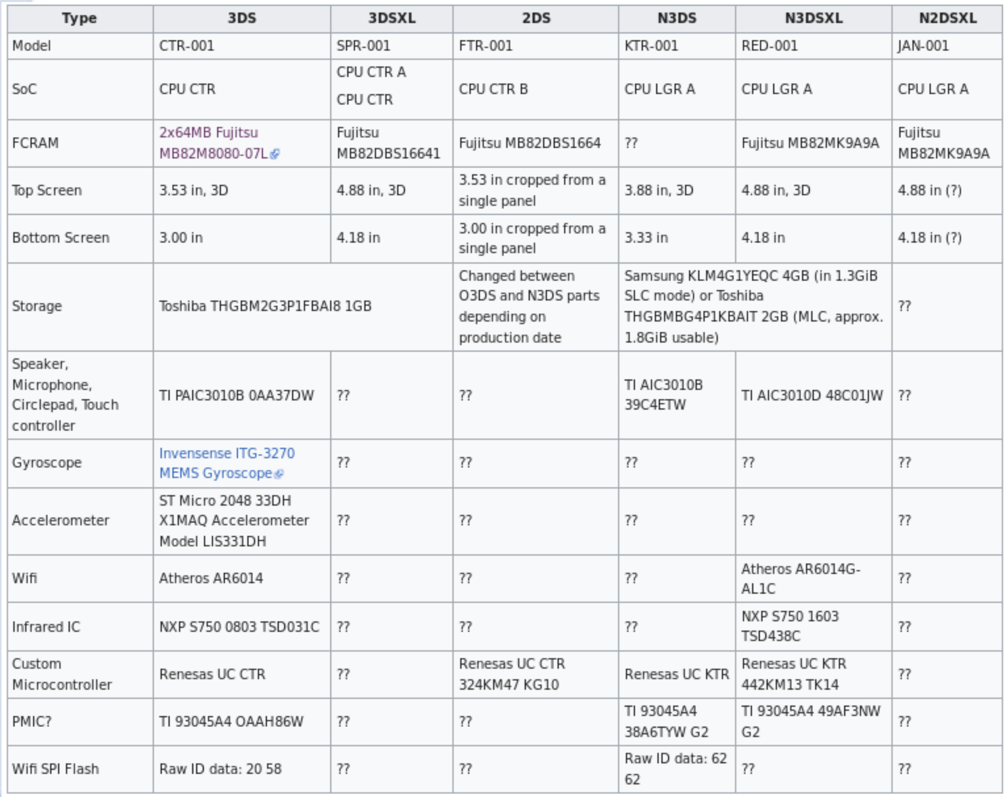
Características Hw de la familia Nintendo (V)

- *NDS*
  - *256x192px, 262,144 (2^{18}?) colores, 2 niveles brillo, 67 MHz ARM946E-S + 33 MHz ARM7TDMI, 4MB, 802.11 (legacy mode)*
- *NDS Lite*
  - *4 niveles brillo*
- NDSi / NDSi XL
  - 133 MHz ARM9 + 33 MHz ARM7, 16MB, 802.11b/g, 2x0.3 Mpx cámaras

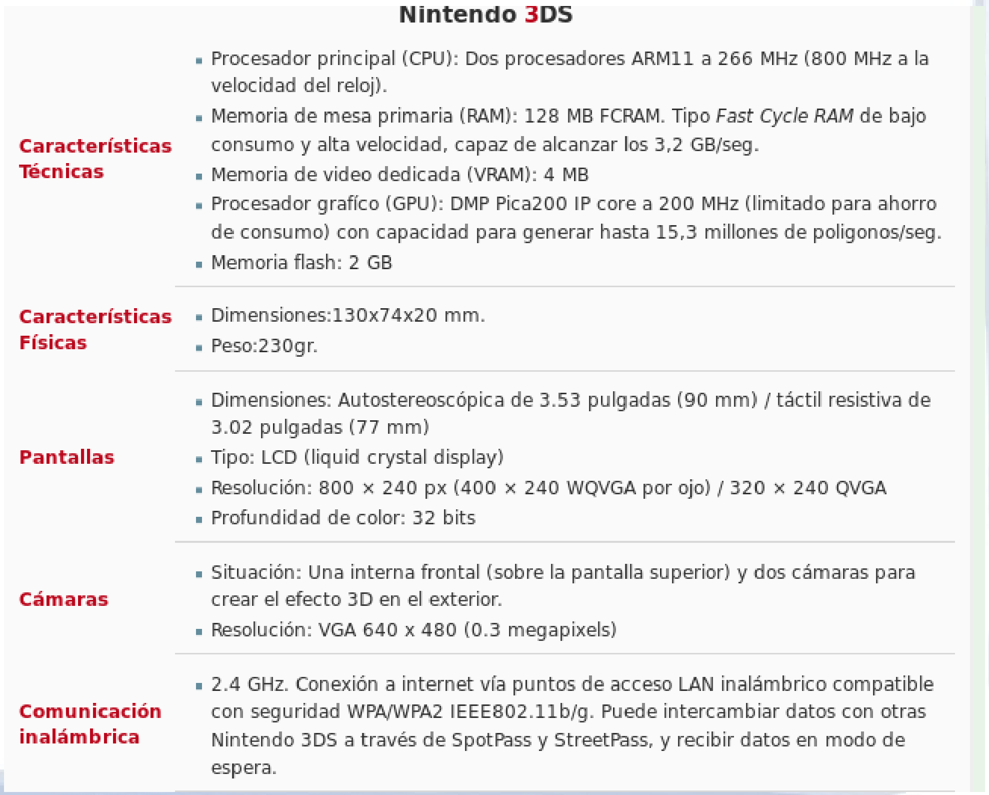
Imágenes obtenidas de la Wikipedia: <http://www.wikipedia.org/> y del sitio web de Nintendo <http://www.nintendo.com/>

Características Hw de la familia Nintendo (VI)

- 3DS / 3DS XL
  - 800×240px + 320× 240px, 16,7Mcolores, 5 niveles brillo, dual/quad-core ARM11 + ARM9 + GPU PICA200 + 128MB, 3x0,3Mpx cámaras, acelerómetro, giróscopo,



Type	3DS	3DSXL	2DS	N3DS	N3DSXL	N2DSXL
Model	CTR-001	SPR-001	FTR-001	KTR-001	RED-001	JAN-001
SoC	CPU CTR	CPU CTR A CPU CTR	CPU CTR B	CPU LGR A	CPU LGR A	CPU LGR A
FCRAM	2x64MB Fujitsu MB82M8080-07L	Fujitsu MB82DBS16641	Fujitsu MB82DBS1664	??	Fujitsu MB82MK9A9A	Fujitsu MB82MK9A9A
Top Screen	3.53 in, 3D	4.88 in, 3D	3.53 in cropped from a single panel	3.88 in, 3D	4.88 in, 3D	4.88 in (?)
Bottom Screen	3.00 in	4.18 in	3.00 in cropped from a single panel	3.33 in	4.18 in	4.18 in (?)
Storage	Toshiba THGBM2G3P1FBAI8 1GB		Changed between O3DS and N3DS parts depending on production date	Samsung KLM4G1YEQC 4GB (in 1.3GiB SLC mode) or Toshiba THGBMBG4P1KBAIT 2GB (MLC, approx. 1.8GiB usable)		??
Speaker, Microphone, Circlepad, Touch controller	TI PAIC3010B 0AA37DW	??	??	TI AIC3010B 39C4ETW	TI AIC3010D 48C01JW	??
Gyroscope	Invensense ITG-3270 MEMS Gyroscope	??	??	??	??	??
Accelerometer	ST Micro 2048 33DH X1MAQ Accelerometer Model LIS331DH	??	??	??	??	??
Wifi	Atheros AR6014	??	??	??	Atheros AR6014G-AL1C	??
Infrared IC	NXP S750 0803 TSD031C	??	??	??	NXP S750 1603 TSD438C	??
Custom Microcontroller	Renesas UC CTR	??	Renesas UC CTR 324KM47 KG10	Renesas UC KTR	Renesas UC KTR 442KM13 TK14	??
PMIC?	TI 93045A4 OAAH86W	??	??	TI 93045A4 38A6TYW G2	TI 93045A4 49AF3NW G2	??
Wifi SPI Flash	Raw ID data: 20 58	??	??	Raw ID data: 62 62	??	??



Nintendo 3DS

Características Técnicas	▪ Procesador principal (CPU): Dos procesadores ARM11 a 266 MHz (800 MHz a la velocidad del reloj). ▪ Memoria de mesa primaria (RAM): 128 MB FCRAM. Tipo *Fast Cycle RAM* de bajo consumo y alta velocidad, capaz de alcanzar los 3,2 GB/seg. ▪ Memoria de video dedicada (VRAM): 4 MB ▪ Procesador grafíco (GPU): DMP Pica200 IP core a 200 MHz (limitado para ahorro de consumo) con capacidad para generar hasta 15,3 millones de poligonos/seg. ▪ Memoria flash: 2 GB
Características Físicas	▪ Dimensiones:130x74x20 mm. ▪ Peso:230gr.
Pantallas	▪ Dimensiones: Autostereoscópica de 3.53 pulgadas (90 mm) / táctil resistiva de 3.02 pulgadas (77 mm) ▪ Tipo: LCD (liquid crystal display) ▪ Resolución: 800 × 240 px (400 × 240 WQVGA por ojo) / 320 × 240 QVGA ▪ Profundidad de color: 32 bits
Cámaras	▪ Situación: Una interna frontal (sobre la pantalla superior) y dos cámaras para crear el efecto 3D en el exterior. ▪ Resolución: VGA 640 x 480 (0.3 megapixels)
Comunicación inalámbrica	▪ 2.4 GHz. Conexión a internet vía puntos de acceso LAN inalámbrico compatible con seguridad WPA/WPA2 IEEE802.11b/g. Puede intercambiar datos con otras Nintendo 3DS a través de SpotPass y StreetPass, y recibir datos en modo de espera.

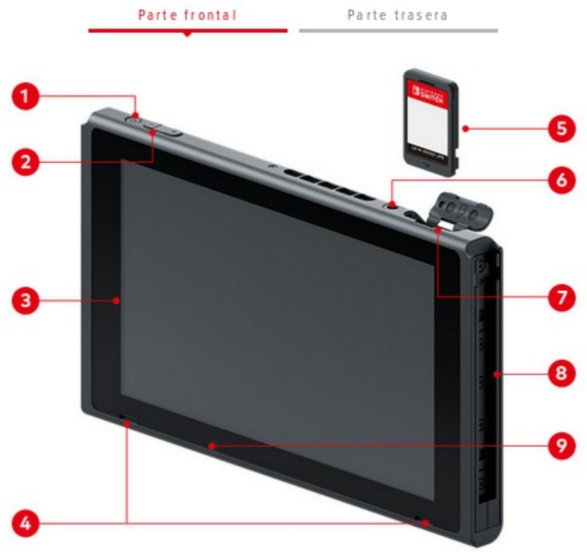
Imágenes de Wikipedia: <http://www.wikipedia.org/>, 3dbrew <https://3dbrew.org/wiki/Hardware > y Nitntedo 3DS – ElOtroLado <https://www.elotrolado.net/wiki/Nintendo

Características Hw de la familia Nintendo (y V)

- *Switch (NX)*
  - *Consola: Pantalla Multitáctil de 6.2” (15.75 cm), 1280 x 720, NVIDIA Custom Tegra, WiFi, Bluetooth 4.1, acelerómetro, giroscopio y sensor de brillo*
  - *Controles Joy-Con:* Bluetooth 3.0, NFC, acelerómetro, giroscopio, cámara infrarroja de movimiento, vibración HD

NINTENDO SWITCH
Fecha de lanzamiento: 03/03/2017
Nintendo Switch
Hardware de Nintendo Switch
Nintendo Switch - Juega cuando quieras

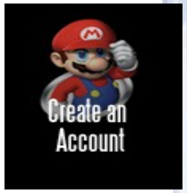
Contenido de la caja
Especificaciones técnicas
Accesorios
Control parental de Nintendo Switch
Servicio en línea



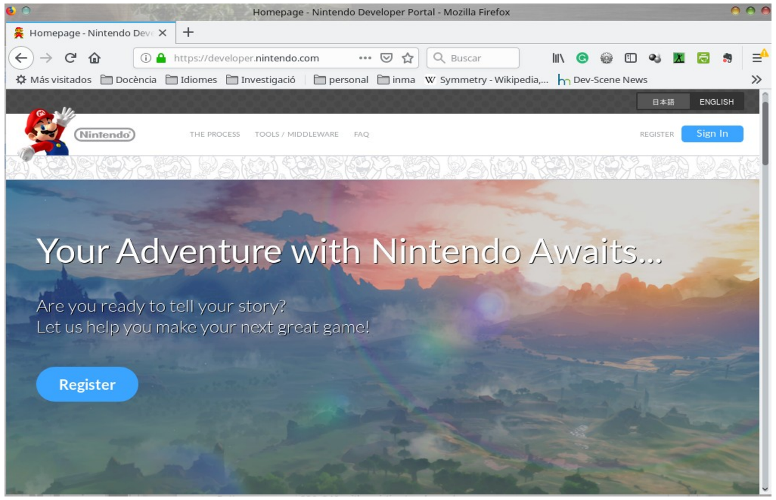
Especificaciones oficiales <https://www.nintendo.com/es_LA/switch/features/tech-specs/>
Imágenes obtenidas de Nintendo <https://www.nintendo.es/Nintendo-Switch/Nintendo-Switch-1148779.html>,
<https://www.nintendo.es/Nintendo-Switch/Nintendo-Switch-1148779.html>

Desarrollo de aplicaciones

- *Desarrollo para esta plataforma*
  - Con el SDK “oficial”
    - Pre - 2016



    - 2016


Homepage - Nintendo Developer Portal - Mozilla Firefox
https://developer.nintendo.com
Nintendo
THE PROCESS
TOOLS / MIDDLEWARE
FAQ
REGISTER
Sign In
ENGLISH
Register

Your Adventure with Nintendo Awaits...

Are you ready to tell your story?
Let us help you make your next great game!

  - Desarrollo no oficial → *Homebrew*
- *Ejecución del distribuible resultante*

Imágenes obtenidas del sitio web de SDSG: <http://www.warioworld.com/> y de NDP <https://developer.nintendo.com/>,

Desarrollo de aplicaciones (II)

- *SDK "oficial" (pre – 2016)*

  - *"Software Development Suport Group" (SDSG)*
  - *Gestiona el acceso a la documentación y SDK:*
    - *"To become an Authorized Developer for Nintendo game platforms".*

Imagen obtenida del sitio web de SDSG (2k14/2k15):

<http://www.warioworld.com/>,

NINTENDO
NON-DISCLOSURE AGREEMENT

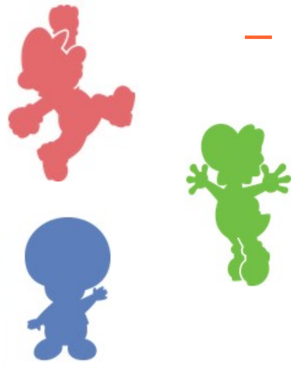
[illegible]

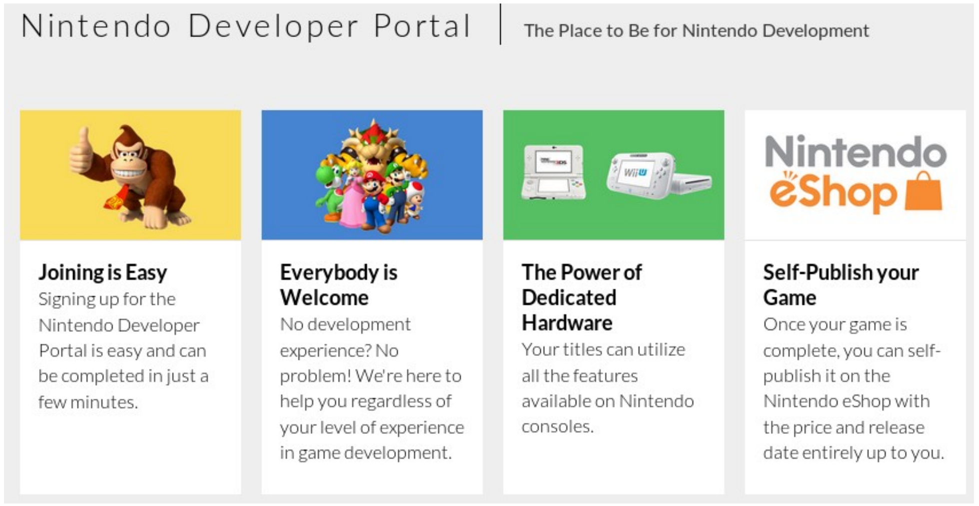
    - *"To apply for Wii U, Nintendo 3DS, Wii/WiiWare, or Nintendo DSi/DSiWare, you will need to complete an updated Developer Application for your company."*

Imágenes obtenidas del sitio web de SDSG: <http://www.warioworld.com/>.

Desarrollo de aplicaciones (III)

- *¿Novedades en el SDK oficial?*
  - *Nintendo Developer Portal <https://developer.nintendo.com/ >*
    - *Nintendo Developer Program*





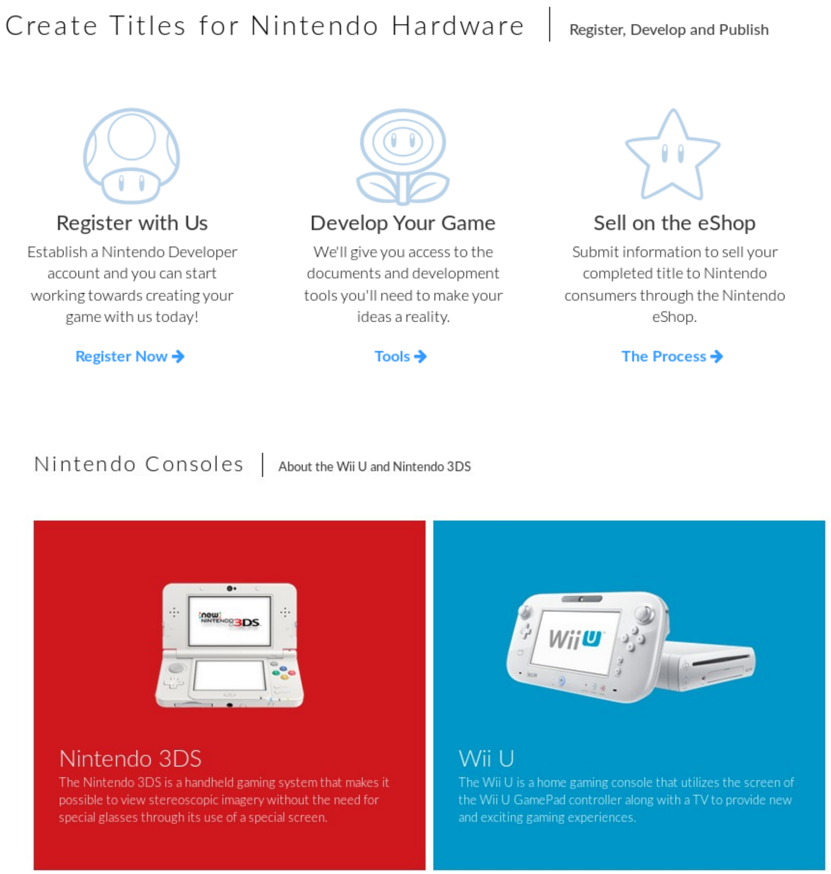
Nintendo Developer Portal | The Place to Be for Nintendo Development

Joining is Easy
Signing up for the Nintendo Developer Portal is easy and can be completed in just a few minutes.

Everybody is Welcome
No development experience? No problem! We're here to help you regardless of your level of experience in game development.

The Power of Dedicated Hardware
Your titles can utilize all the features available on Nintendo consoles.

Self-Publish your Game
Once your game is complete, you can self-publish it on the Nintendo eShop with the price and release date entirely up to you.



Create Titles for Nintendo Hardware | Register, Develop and Publish

Register with Us
Establish a Nintendo Developer account and you can start working towards creating your game with us today!
Register Now →

Develop Your Game
We'll give you access to the documents and development tools you'll need to make your ideas a reality.
Tools →

Sell on the eShop
Submit information to sell your completed title to Nintendo consumers through the Nintendo eShop.
The Process →

Nintendo Consoles | About the Wii U and Nintendo 3DS

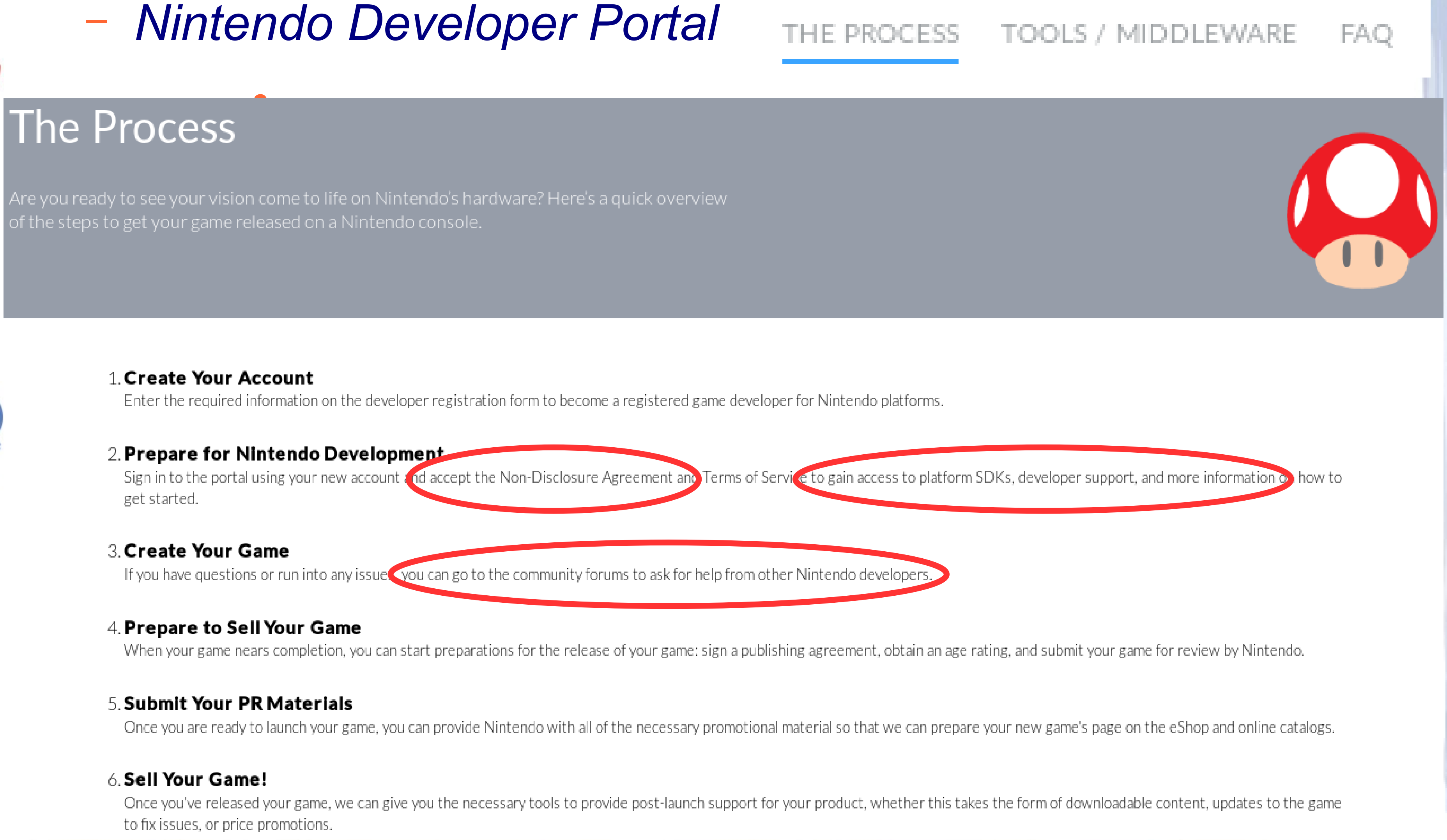
Nintendo 3DS
The Nintendo 3DS is a handheld gaming system that makes it possible to view stereoscopic imagery without the need for special glasses through its use of a special screen.

Wii U
The Wii U is a home gaming console that utilizes the screen of the Wii U GamePad controller along with a TV to provide new and exciting gaming experiences.

Imágenes de <http://listas.20minutos.es/lista/personajes-de-nintendo-280983/ > y de <https://developer.nintendo.com/>

Desarrollo de aplicaciones (IV)

- *¿Novedades en el SDK oficial?*
  - *Nintendo Developer Portal*



The Process

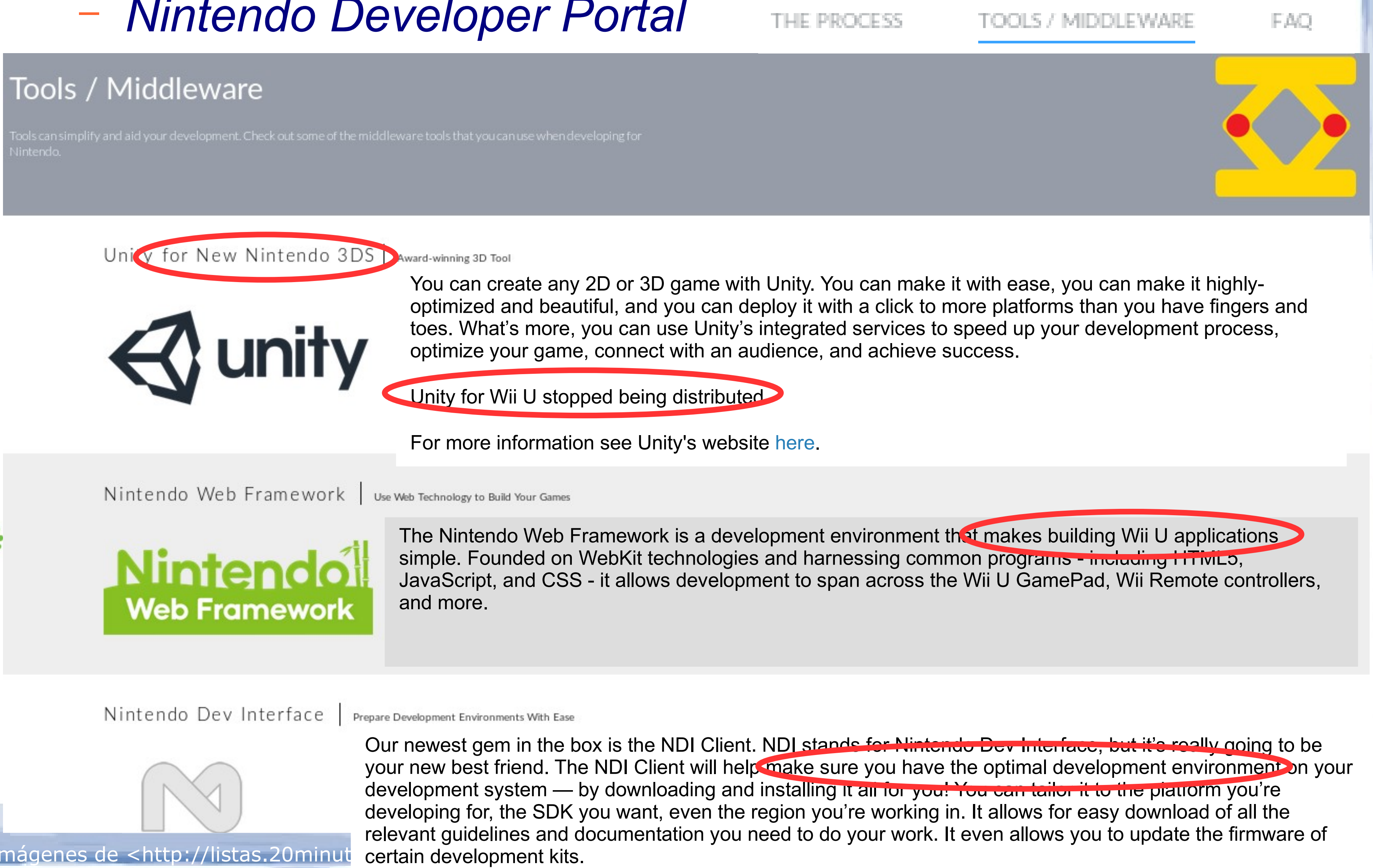
Are you ready to see your vision come to life on Nintendo's hardware? Here's a quick overview of the steps to get your game released on a Nintendo console.

1. **Create Your Account**
   Enter the required information on the developer registration form to become a registered game developer for Nintendo platforms.
2. **Prepare for Nintendo Development**
   Sign in to the portal using your new account and accept the Non-Disclosure Agreement and Terms of Service to gain access to platform SDKs, developer support, and more information on how to get started.
3. **Create Your Game**
   If you have questions or run into any issues, you can go to the community forums to ask for help from other Nintendo developers.
4. **Prepare to Sell Your Game**
   When your game nears completion, you can start preparations for the release of your game: sign a publishing agreement, obtain an age rating, and submit your game for review by Nintendo.
5. **Submit Your PR Materials**
   Once you are ready to launch your game, you can provide Nintendo with all of the necessary promotional material so that we can prepare your new game's page on the eShop and online catalogs.
6. **Sell Your Game!**
   Once you've released your game, we can give you the necessary tools to provide post-launch support for your product, whether this takes the form of downloadable content, updates to the game to fix issues, or price promotions.

Imágenes de <http://listas.20minutos.es/lista/personajes-de-nintendo-280983/ > y de <https://developer.nintendo.com/>

Desarrollo de aplicaciones (V)

- *¿Novedades en el SDK oficial?*
  - *Nintendo Developer Portal*



THE PROCESS TOOLS / MIDDLEWARE FAQ

Tools / Middleware

Tools can simplify and aid your development. Check out some of the middleware tools that you can use when developing for Nintendo.

Unity for New Nintendo 3DS | Award-winning 3D Tool

You can create any 2D or 3D game with Unity. You can make it with ease, you can make it highly-optimized and beautiful, and you can deploy it with a click to more platforms than you have fingers and toes. What's more, you can use Unity's integrated services to speed up your development process, optimize your game, connect with an audience, and achieve success.

Unity for Wii U stopped being distributed

For more information see Unity's website here.

Nintendo Web Framework | Use Web Technology to Build Your Games

The Nintendo Web Framework is a development environment that makes building Wii U applications simple. Founded on WebKit technologies and harnessing common programs - including HTML5, JavaScript, and CSS - it allows development to span across the Wii U GamePad, Wii Remote controllers, and more.

Nintendo Dev Interface | Prepare Development Environments With Ease

Our newest gem in the box is the NDI Client. NDI stands for Nintendo Dev Interface, but it's really going to be your new best friend. The NDI Client will help make sure you have the optimal development environment on your development system — by downloading and installing it all for you! You can tailor it to the platform you're developing for, the SDK you want, even the region you're working in. It allows for easy download of all the relevant guidelines and documentation you need to do your work. It even allows you to update the firmware of certain development kits.

Imágenes de <http://listas.20minut

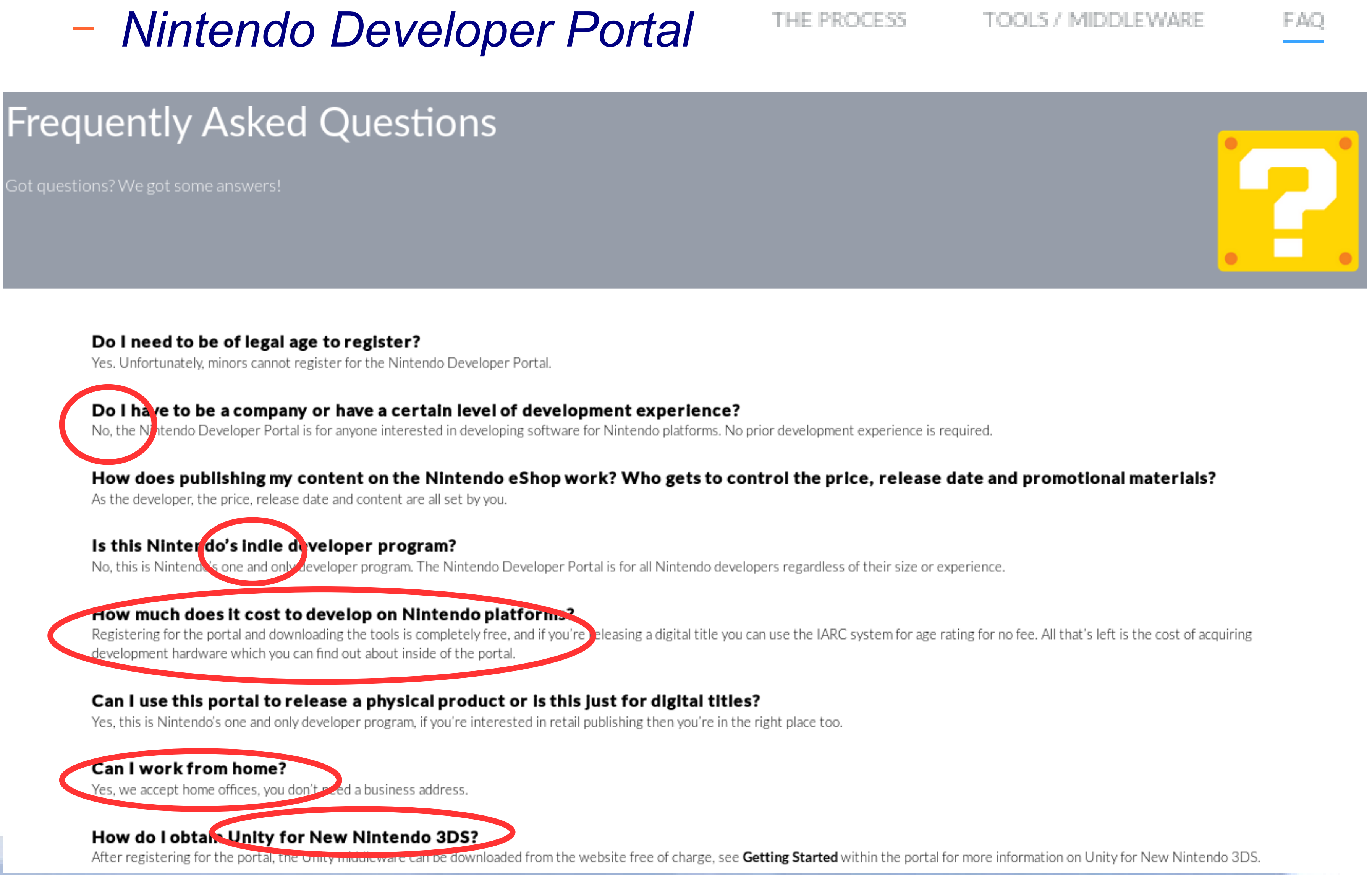
Desarrollo de aplicaciones (VI)

- *Nintendo Developer Portal*

  - *Middleware <https://developer.nintendo.com/game-developers>*
    - *Middleware can simplify and aid your game development, providing services as diverse as game engines, compressions tools, video codecs, sound players, and more. In addition to a large array of 3rd-Party middleware that's available for our systems, Nintendo has gone the extra step and licensed some middleware to our developers for free or reduced cost. Want to use Unity for Wii U to build your Wii U game? You can do that. With the Nintendo Web Framework, you can build your game or app in HTML 5, as well. From Havok Wii U XS to PUX for both Wii U and Nintendo 3DS, we've got quite the selection to meet your needs.*
    - *Havox <http://www.havok.com/>*
      - *is an Irish computer software company that provides interactive software and services for digital media creators in the video game and movie industries: physics, destruction, cloth, AI, ...*
        - *Microsft (2015) ← Intel (2007)*
    - PUX <http://pux.co.jp/en/casestudy/> (Panasonic, Nintendo 27% en 2013)
      - Codec (3GPP / 3GPP2)
      - LiteSpeech / LiteSpeech Advance (SR / TTS) → brainTraining; OCR
      - SoftSensor Image Recognition Software
        - Handwriting, face recognition, scene recognition, object (hand / finger) recognition, gesture recognition

Desarrollo de aplicaciones (VII)

- *¿Novedades en el SDK oficial?*
  - *Nintendo Developer Portal*


THE PROCESS TOOLS / MIDDLEWARE FAQ

Frequently Asked Questions

Got questions? We got some answers!

Do I need to be of legal age to register?
Yes. Unfortunately, minors cannot register for the Nintendo Developer Portal.

Do I have to be a company or have a certain level of development experience?
No, the Nintendo Developer Portal is for anyone interested in developing software for Nintendo platforms. No prior development experience is required.

How does publishing my content on the Nintendo eShop work? Who gets to control the price, release date and promotional materials?
As the developer, the price, release date and content are all set by you.

Is this Nintendo's Indie developer program?
No, this is Nintendo's one and only developer program. The Nintendo Developer Portal is for all Nintendo developers regardless of their size or experience.

How much does it cost to develop on Nintendo platforms?
Registering for the portal and downloading the tools is completely free, and if you're releasing a digital title you can use the IARC system for age rating for no fee. All that's left is the cost of acquiring development hardware which you can find out about inside of the portal.

Can I use this portal to release a physical product or is this just for digital titles?
Yes, this is Nintendo's one and only developer program, if you're interested in retail publishing then you're in the right place too.

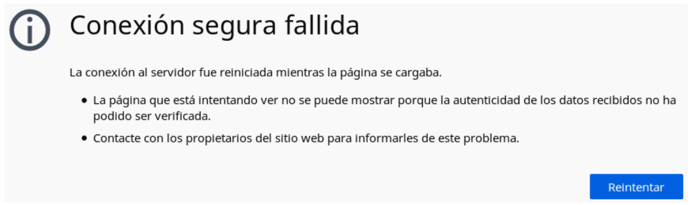
Can I work from home?
Yes, we accept home offices, you don't need a business address.

How do I obtain Unity for New Nintendo 3DS?
After registering for the portal, the Unity middleware can be downloaded from the website free of charge, see **Getting Started** within the portal for more information on Unity for New Nintendo 3DS.

Imágenes de <http://listas.20minutos.es/lista/personajes-de-nintendo-280983/ > y de <https://developer.nintendo.com/>

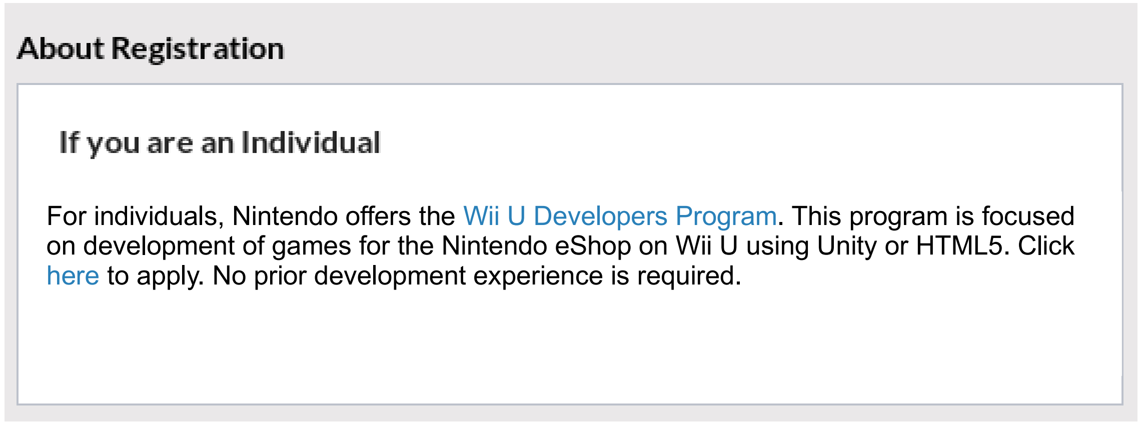
Desarrollo de aplicaciones (VIII)

- *¿Novedades en el SDK oficial?*
  - *Nintendo Developer Portal <https://developer.nintendo.com/ >*
    - *Nintendo Developer Program*
      - *Register as a Nintendo Developer*


Conexión segura fallida

La conexión al servidor fue reiniciada mientras la página se cargaba.

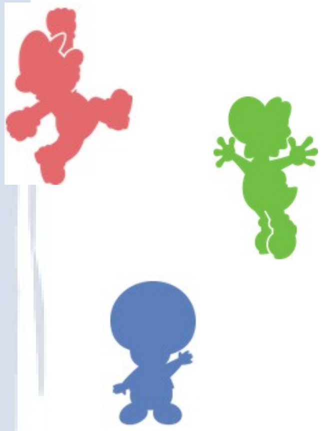
- La página que está intentando ver no se puede mostrar porque la autenticidad de los datos recibidos no ha podido ser verificada.
- Contacte con los propietarios del sitio web para informarles de este problema.


About Registration

If you are an Individual

For individuals, Nintendo offers the Wii U Developers Program. This program is focused on development of games for the Nintendo eShop on Wii U using Unity or HTML5. Click here to apply. No prior development experience is required.

      - *¿NX?*



Desarrollo: conclusión (y IX)

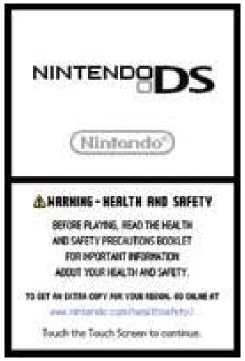
- Entonces, ¿es cierto? ¡Existe!
  - Y es gratis … o casi
  - No se puede publicar: *Non Disclossure Agrement!*
  - Limitaciones
    - En la versión de Unity 3D
    - En el acceso a plataformas:
      - *WiiU* se ha quedado fuera
      - ¿Dónde está el soporte para *Switch*?

Desarrollo de aplicaciones: ejecución

- *Cómo se ejecuta el distribuible resultante*
  - *Firmware* nativo: caso de estudio NDS
  - *FlashCards / FlashCarts*
  - *Homebre Launcher*

Desarrollo de aplicaciones: ejecución (II)

- *Firmware* de la NDS
  - *Nintendo's own firmware boots the system.*
    - *A health and safety warning is displayed first, then the main menu is loaded.*
    - *The main menu presents the player with four main options to select: play a DS game [3], use PictoChat [4], initiate DS Download Play [12], or play a Game Boy Advance game [6] y [10].*
      - *The main menu also has some secondary options such as date and time, GBA screen, and touchscreen calibration.*
      - *Also features an alarm clock, several options for customization ..., and the ability to input user information and preferences (such as name, birthday, favorite color, etc.) that can be used in games.*


NINTENDO DS

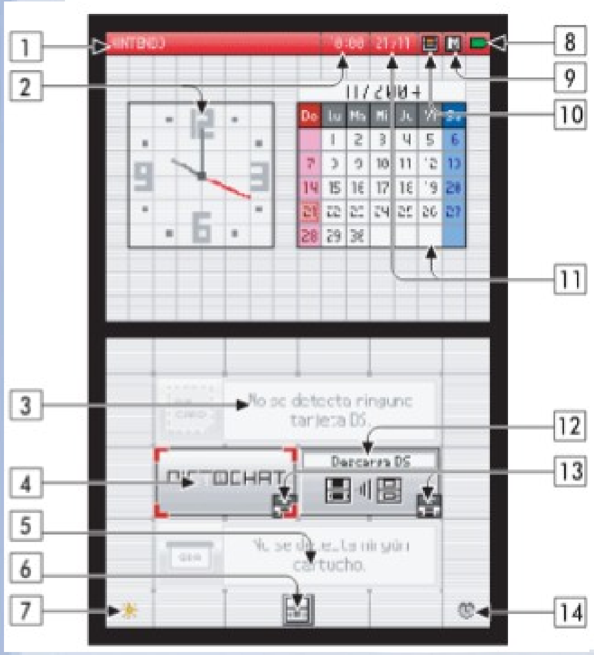
Nintendo

⚠WARNING - HEALTH AND SAFETY

BEFORE PLAYING, READ THE HEALTH AND SAFETY PRECAUTIONS BOOKLET FOR IMPORTANT INFORMATION ABOUT YOUR HEALTH AND SAFETY.

TO GET AN EXTRA COPY FOR YOUR REGION, GO ONLINE AT www.nintendo.com/healthsafety/

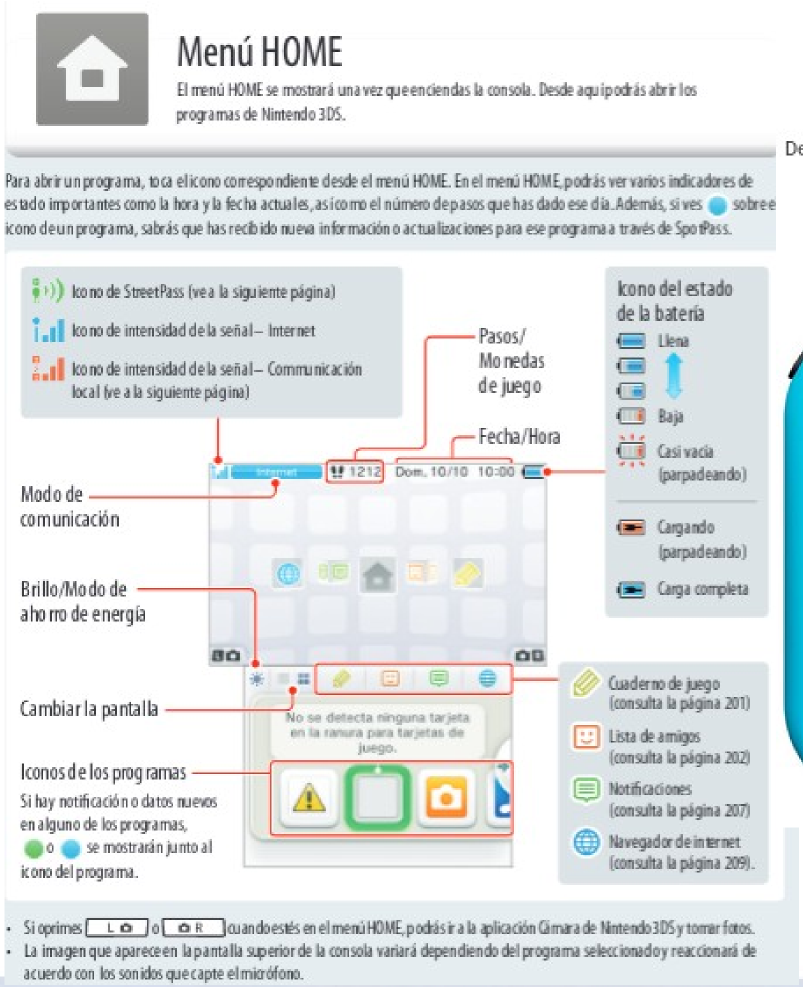
Touch the Touch Screen to continue.



Imágenes de "Manuales descargables" <https://es-americas-support.nintendo.com/app/answers/detail/a_id/23171>.

Desarrollo de aplicaciones: ejecución (III)

- *3DS y Switch*



Menú HOME

El menú HOME se mostrará una vez que enciendas la consola. Desde aquí podrás abrir los programas de Nintendo 3DS.

Para abrir un programa, toca el icono correspondiente desde el menú HOME. En el menú HOME, podrás ver varios indicadores de estado importantes como la hora y la fecha actuales, así como el número de pasos que has dado ese día. Además, si ves ● sobre el icono de un programa, sabrás que has recibido nueva información o actualizaciones para ese programa a través de SpotPass.

- Icono de StreetPass (vea la siguiente página)
- Icono de intensidad de la señal– Internet
- Icono de intensidad de la señal– Communicación local (ve a la siguiente página)

Icono del estado de la batería

- Llena
- Baja
- Casi vacía (parpadeando)
- Cargando (parpadeando)
- Carga completa

Iconos de los programas

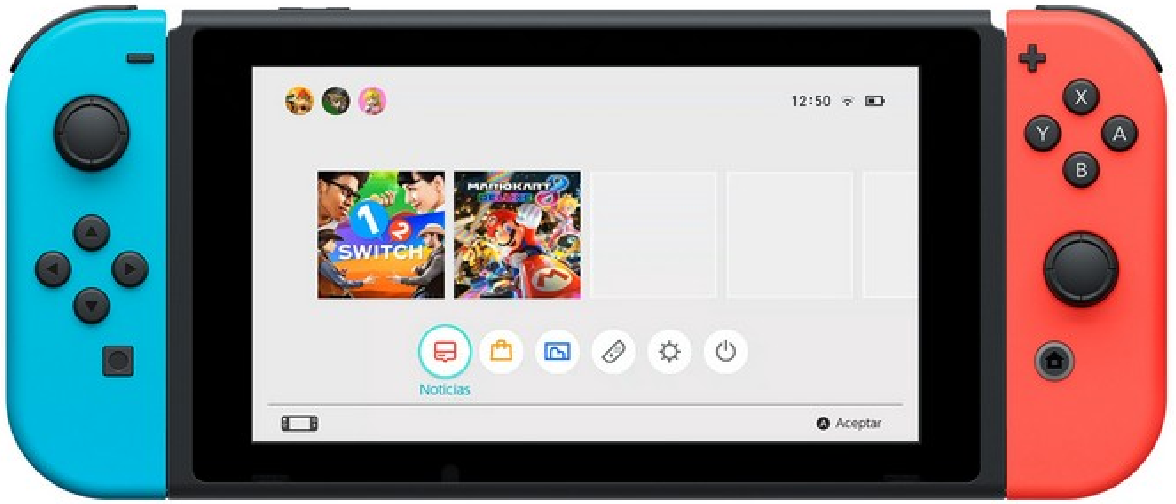
Si hay notificación o datos nuevos en alguno de los programas, ● o ● se mostrarán junto al icono del programa.

- Cuaderno de juego (consulta la página 201)
- Lista de amigos (consulta la página 202)
- Notificaciones (consulta la página 207)
- Navegador de internet (consulta la página 209).

- Si oprimes L o R cuando estés en el menú HOME, podrás ir a la aplicación Cámara de Nintendo 3DS y tomar fotos.
- La imagen que aparece en la pantalla superior de la consola variará dependiendo del programa seleccionado y reaccionará de acuerdo con los sonidos que capte el micrófono.

Menú HOME

Desde el menú HOME puedes iniciar juegos y seleccionar ajustes. Aquí, además de gestionar tu lista de amigos y cuentas de usuario, también puedes cambiar los ajustes del sistema.



Imágenes de "CTR_ops_sp.pdf" <https://www.nintendo.com/consumer/downloads/CTR_ops_sp.pdf>.

Desarrollo de aplicaciones: ejecución (III)

- *FlahsCards / FlashCarts*
  - *Cargar aplicaciones desde el FlashCard*
    - *Ejecución de juegos oficiales*
    - *Copias de seguridad*
    - *Ejecución de aplicaciones no oficiales*
      - *SDK no oficial o "homebrew"*
  - Simon van de Berg. 2006. Running Nintendo DS homebrew
    - Pirating of software is something I do not approve of.
    - Pirating is often associated with homebrew. 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.
    - Homebrew is creating and sharing programs made by yourself and/or others for free. This means that no business is attached to the software. ... no support ...
- *Tipos: DS Slot vs* GBA Slot

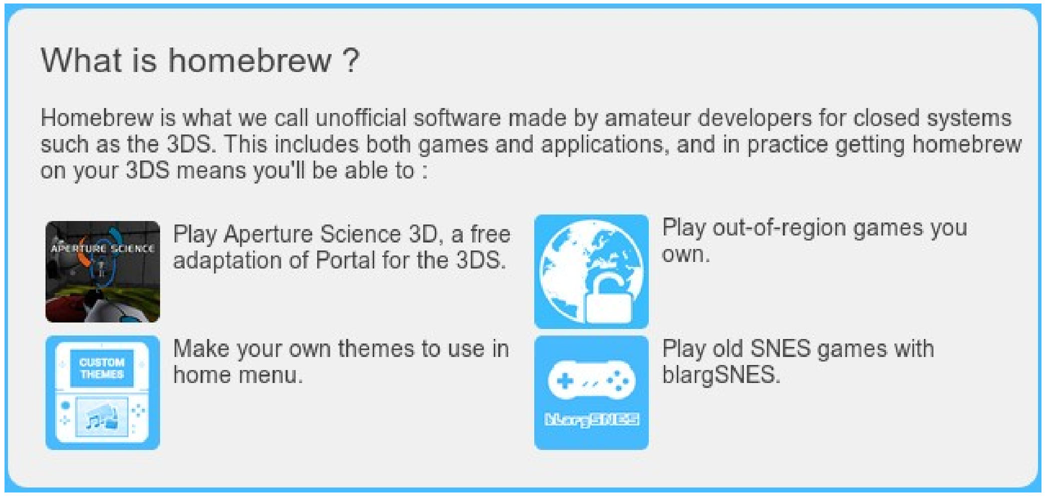
Desarrollo de aplicaciones: ejecución (IV)

- *Flash cartridges*
  - Dispositivos de almacenamiento NDS
    - *32 MiB block of rewritable flash memory directly-accessible by both CPUs of the Nintendo DS.*
- Tipos
  - En función del puerto o *slot*
    - Slot-1: DS Slot
    - Slot-2: GBA Slot
      - Desaparece en DSi

Imagen obtenida de la Wikipedia: <http://en.wikipedia.org/wiki/Nintendo_DS_homebrew>,

Desarrollo de aplicaciones: ejecución (yV)

- *Homebrew Loader / Launcher*
  - *The Homebrew Launcher (hbmenu for short) is a fairly simple (and beautiful) menu that lists homebrew applications in the 3DSX format and lets you run them.*
  - **No como opción para piratear**



What is homebrew ?

Homebrew is what we call unofficial software made by amateur developers for closed systems such as the 3DS. This includes both games and applications, and in practice getting homebrew on your 3DS means you'll be able to :

Play Aperture Science 3D, a free adaptation of Portal for the 3DS.

Play out-of-region games you own.

Make your own themes to use in home menu.

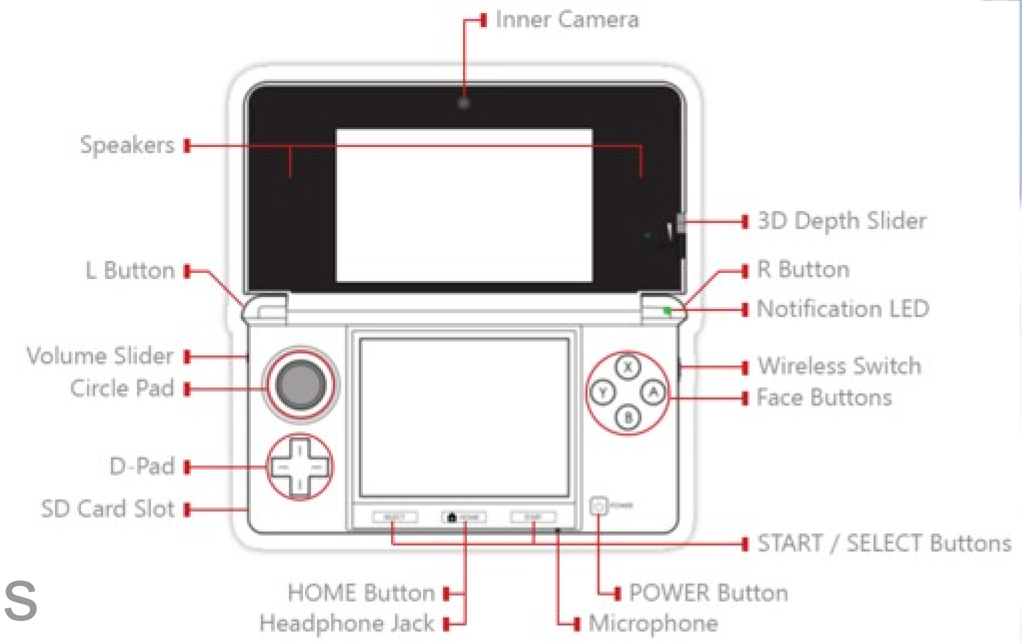
Play old SNES games with blargSNES.

- Carga ejecutables a través de *WiFi*
  - *ndslink/ftpd, 3dslink/ftPony o nxlink*

Imágenes obtenidas de "The Homebrew Launcher 3DS" <http://smealum.github.io/3ds/> y <https://github.com/switchbrew/nx-hbmenu/rele

Arquitectura de la NDS

- Hardware
  - Componentes
    - Compatibilidad hacia atrás
    - Tendencias



  - Esquema de bloques
- Software
  - SDK

Imágenes obtenidas de Jason Fitzpatrick on Lifehacker, 7/23/10 9:00am, "Supercharge Your Homebrew-Hacked Nintendo DS" < http://lifehacker.com/5594640/supercharge-your-homebrew+hacked-nintendo-ds> y de Nintendo <http://www.nintendo.com>.

Componentes

- Vista trasera

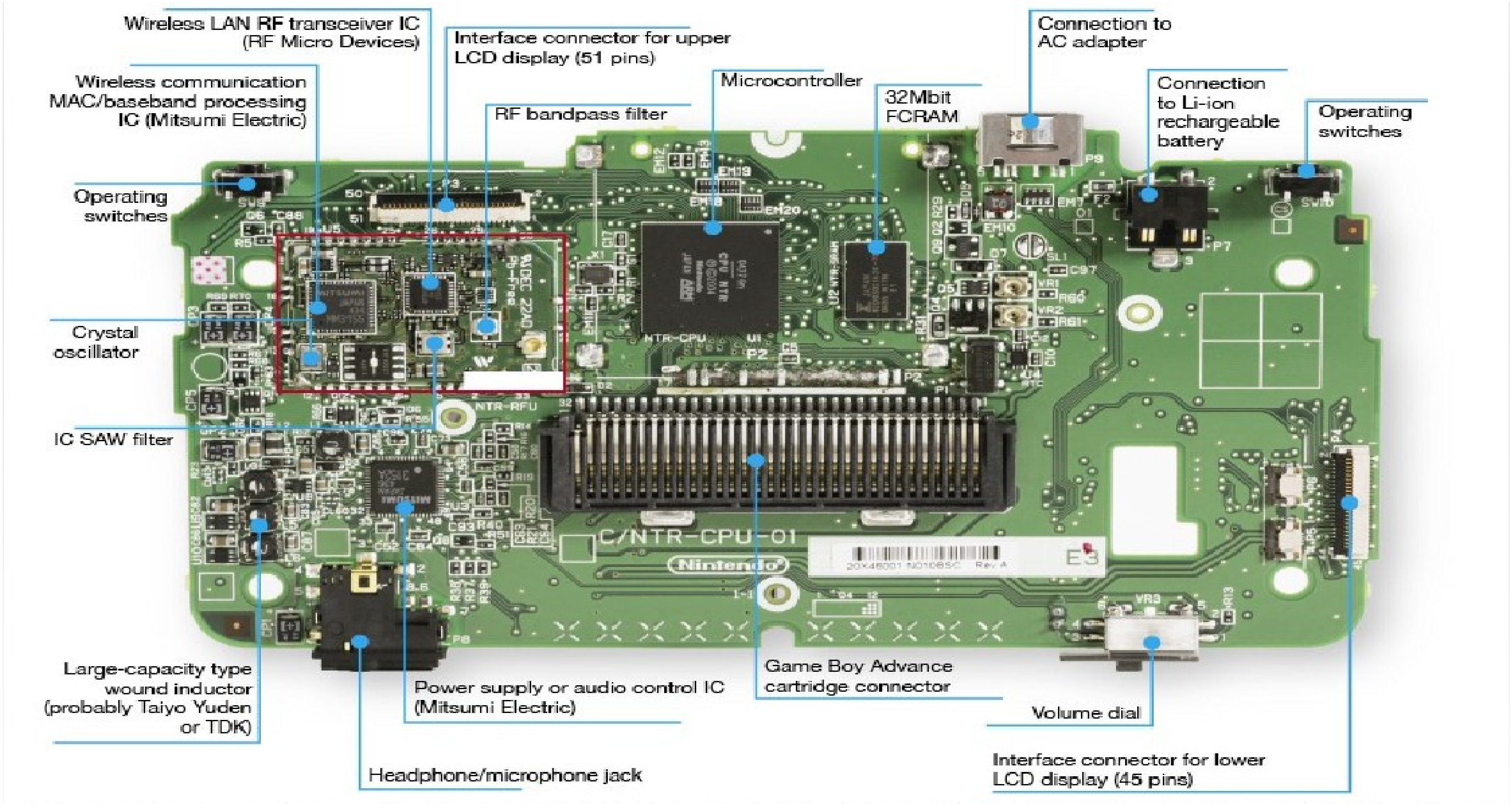

Wireless LAN RF transceiver IC (RF Micro Devices)
Interface connector for upper LCD display (51 pins)
Connection to AC adapter
Wireless communication MAC/baseband processing IC (Mitsumi Electric)
Microcontroller
32Mbit FCRAM
Connection to Li-ion rechargeable battery
Operating switches
RF bandpass filter
Operating switches
Crystal oscillator
IC SAW filter
Large-capacity type wound inductor (probably Taiyo Yuden or TDK)
Power supply or audio control IC (Mitsumi Electric)
Game Boy Advance cartridge connector
Volume dial
Interface connector for lower LCD display (45 pins)
Headphone/microphone jack

Fig 1 Key Circuitry Clumped on One Side The photo is close to the actual size – 141 x 76mm (longest area). The wireless module and microcontroller were covered by a metal shielding plate.

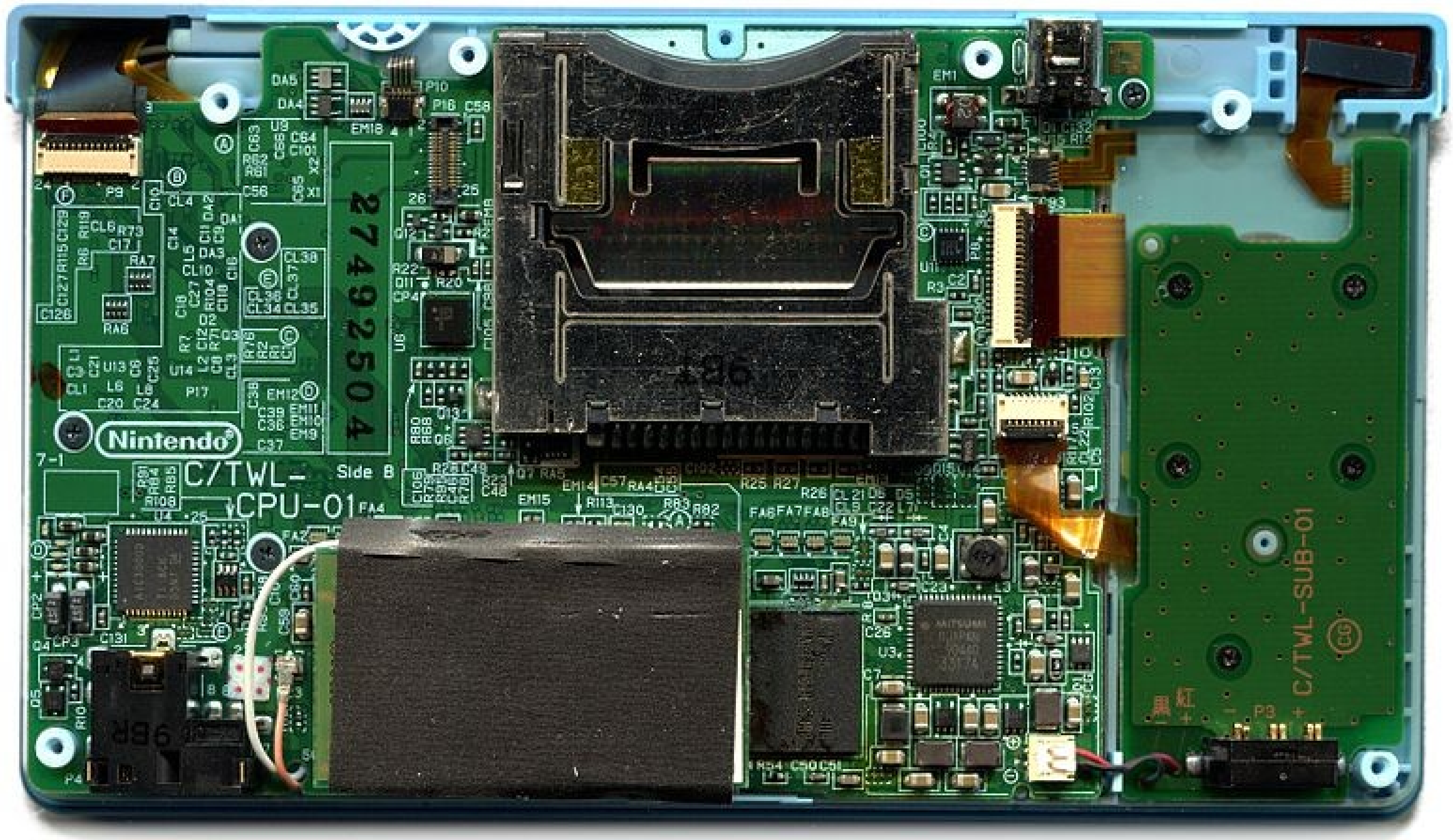
Imagen obtenida de

“CS4803 Design and Programming of Game Consoles, Spring 2011” <http://www.cc.gatech.edu/~hyesoon/spr11/index.html>

<http://techon.nikkeibp.co.jp>

Componentes (y II)

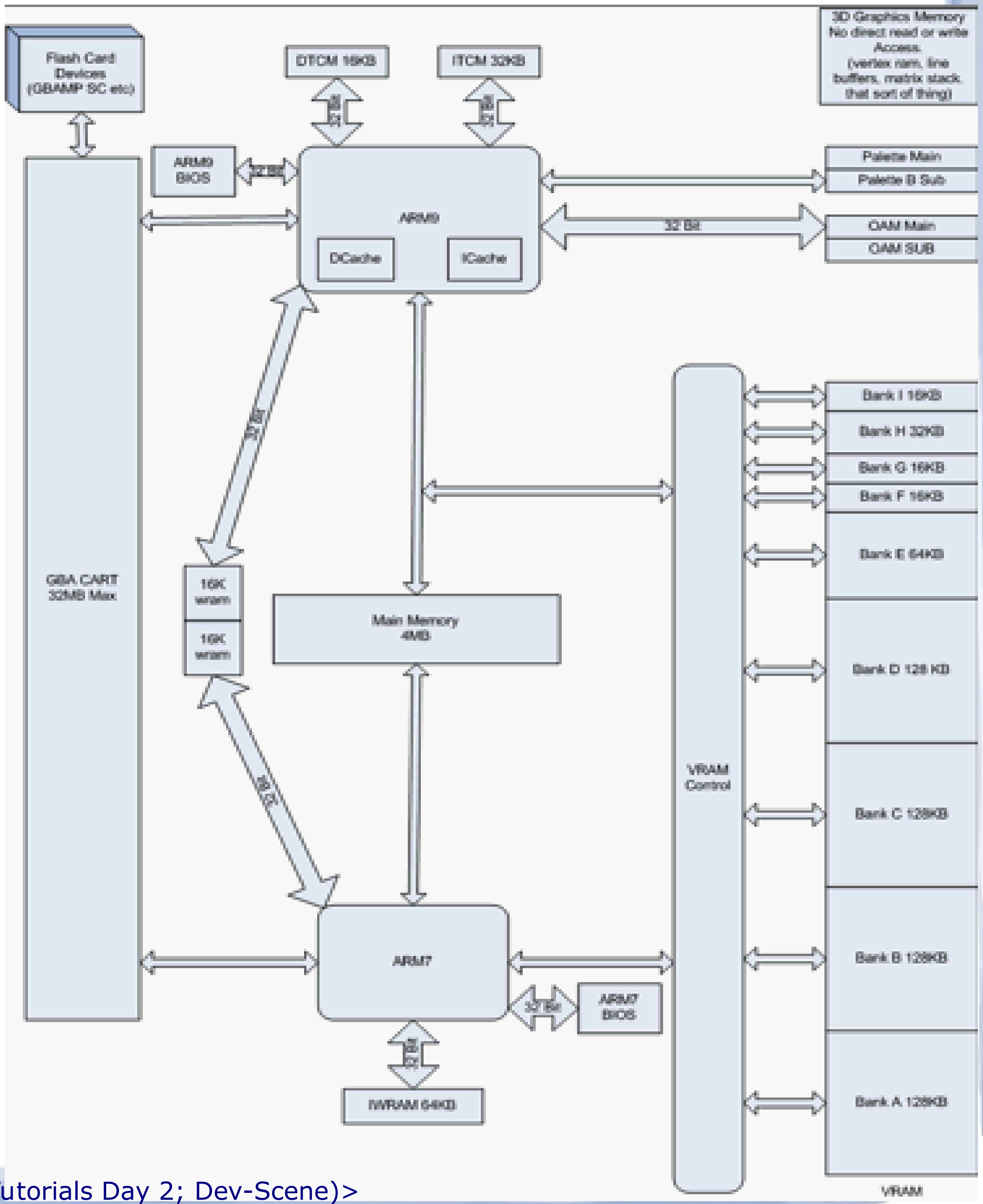
- Vista frontal (NDSi)



Micah Elizabeth Scott from Mountain View, CA <http://www.flickr.com/photos/micahdowty/3778907003>

Esquema de bloques

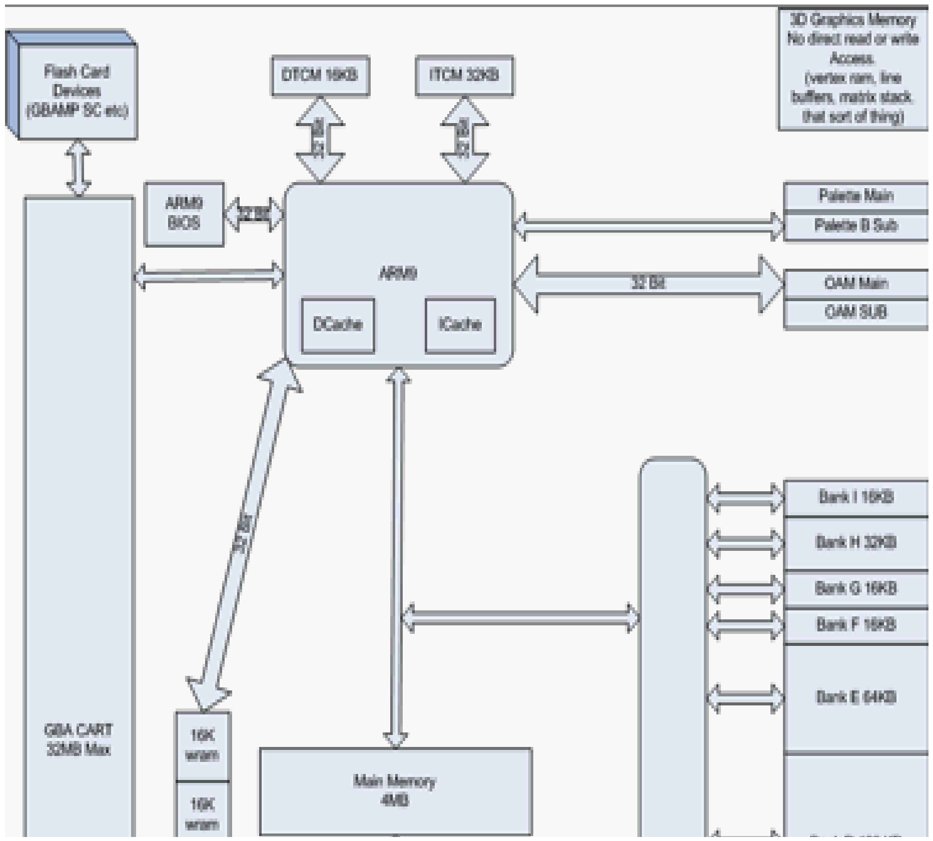
- Conexión entre elementos
  - Procesadores
  - Gestión de memoria
  - Subsistemas
    - Audio
    - “Aceleradores” gráficos



Memory_Layout (NDS/Tutorials Day 2; Dev-Scene)>

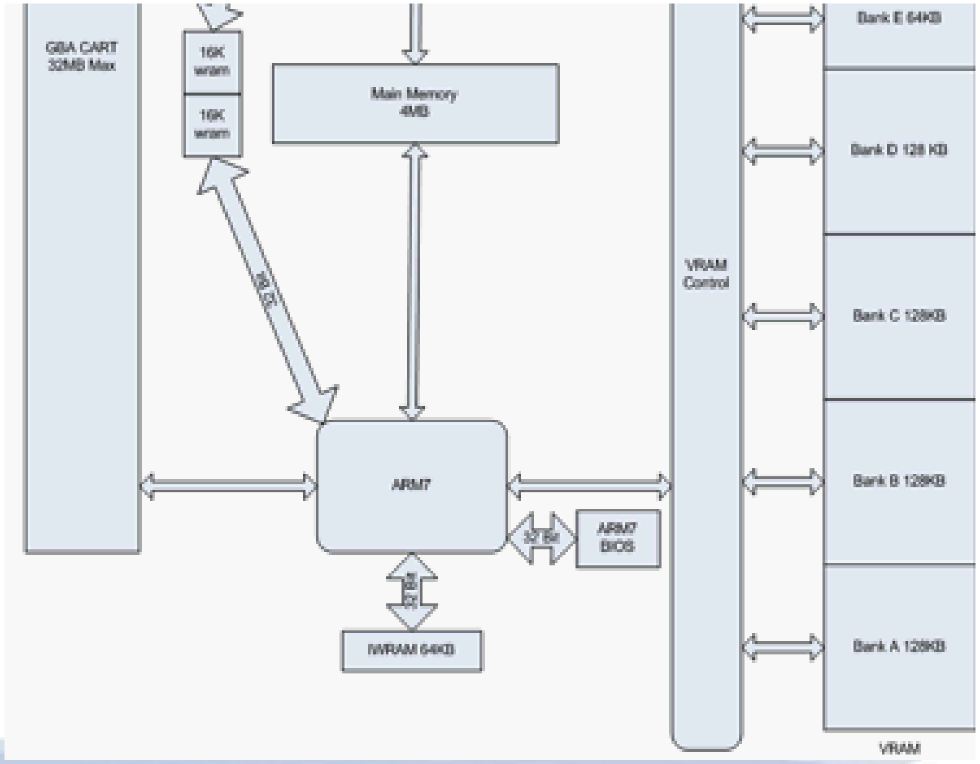
Esquema de bloques (II)

- Elementos
  - ARM9 (66Mhz)
    - Lógica principal del programa



Esquema de bloques (III)

- Elementos
  - ARM7 (33Mhz)
    - Audio
    - Red inalámbrica (*WiFi*)
    - Teclado

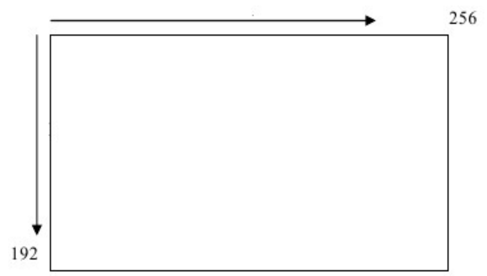


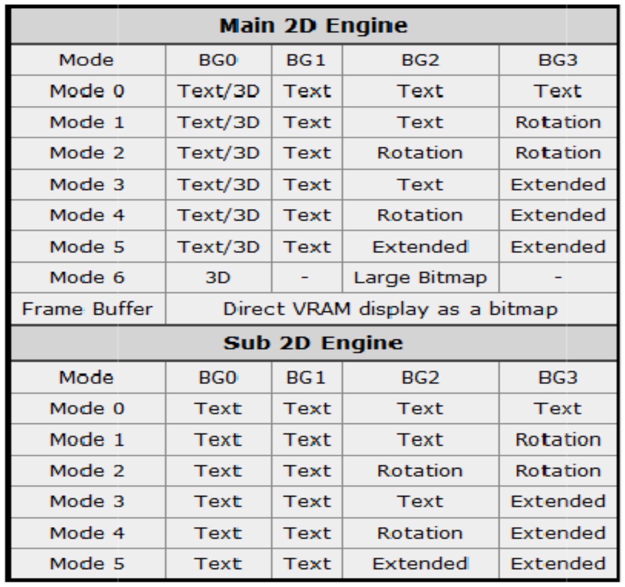
Esquema de bloques (IV)

- RAM de 4 MB
  - Ejecutable para el ARM9
  - Datos globales de la aplicación
  - ARM9 y ARM7 pueden acceder
    - Si hay colisión, prioridad para el ARM7
- ¿MMU? ¿Contigua / Dispersa? ¿Páginas, marcos?
  - Asignación de bancos de memoria
    - Gestionada por la aplicación
    - *Ej.: DevkitPro* → código del ARM7
      - WRAM + IWRAM (96Kb = 32 + 46)

Esquema de bloques (y V)

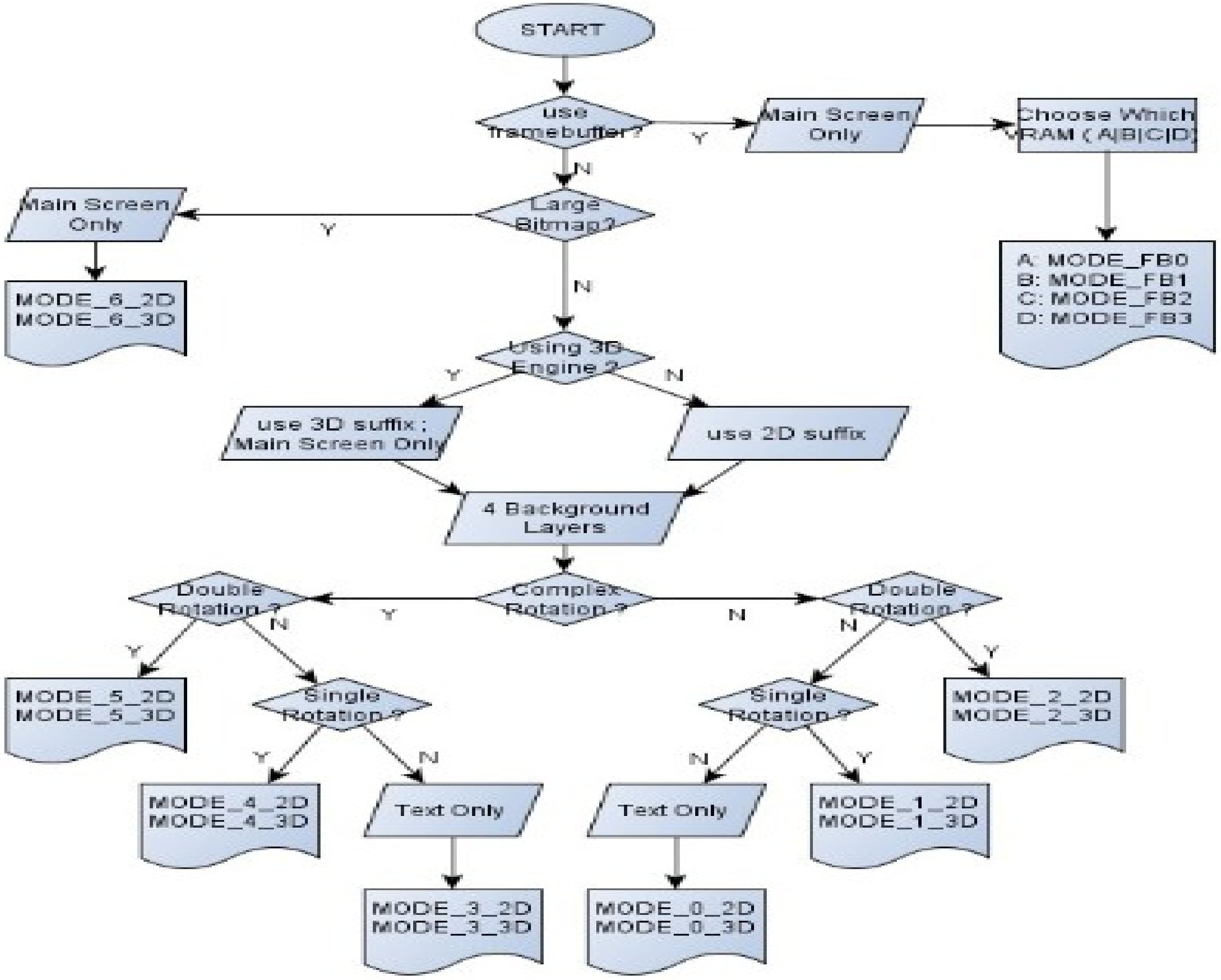
- Modos gráficos → “aceleradores”
  - 2 motores gráficos 2D + 1 motor 3D
  - 2 pantallas: LCD + táctil
    - Resolución




Graphics Modes

Main 2D Engine				
Mode	BG0	BG1	BG2	BG3
Mode 0	Text/3D	Text	Text	Text
Mode 1	Text/3D	Text	Text	Rotation
Mode 2	Text/3D	Text	Rotation	Rotation
Mode 3	Text/3D	Text	Text	Extended
Mode 4	Text/3D	Text	Rotation	Extended
Mode 5	Text/3D	Text	Extended	Extended
Mode 6	3D	-	Large Bitmap	-
Frame Buffer	Direct VRAM display as a bitmap			
Sub 2D Engine				
Mode	BG0	BG1	BG2	BG3
Mode 0	Text	Text	Text	Text
Mode 1	Text	Text	Text	Rotation
Mode 2	Text	Text	Rotation	Rotation
Mode 3	Text	Text	Text	Extended
Mode 4	Text	Text	Rotation	Extended
Mode 5	Text	Text	Extended	Extended

- Otros:
  - 16 canales de sonido, micrófono y altavoces estéro,
  - 2x4 temporizadores, aceleradores para las operaciones de división y raíz cuadrada, etc.



operaciones de división y raíz cuadrada, etc.

Arquitectura de la plataforma: el arranque

- El arranque de la consola
  - BIOS de la consola
    - Código de arranque de los dos procesadores
    - Lee una memoria flash de 256KiB
      - *Firmware* (cifrado) con el menú inicial y la aplicación *pictochat*
      - Lo copia (desencriptado) a la memoria principal y le da el control.
  - *Firmware* comprueba si el cartucho insertado es válido y lee las primeras posiciones de la memoria del cartucho.
    - Características del juego + posiciones códigos ARM9 y ARM7 y dónde se deben copiar.
    - Una vez copiados a memoria principal, el *firmware* les da el control.

Arquitectura de la plataforma: el arranque

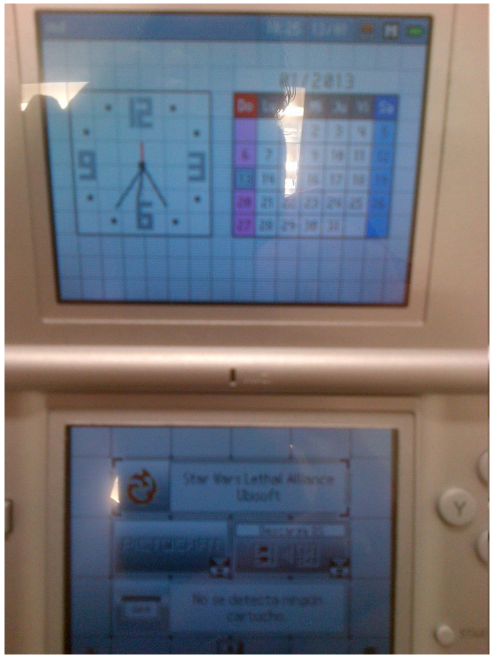
- El arranque de la consola (cont.)
  - BIOS de la consola, *firmware* … cede el control al código del cartucho
  - El *firmware* no ejecutará programas que no estén debidamente firmados
- Técnicas para poder ejecutar soft. no oficial
  - *Pass-through*
    - Evitar tener que entender el mecanismo de cifrado.
    - Estas técnicas requieren un cartucho de Slot1 que simula el comportamiento de un cartucho de juego comercial.
  - *Flashme*
    - Reemplazar el *firmware* por una versión que no comprueba ningún tipo de firma.

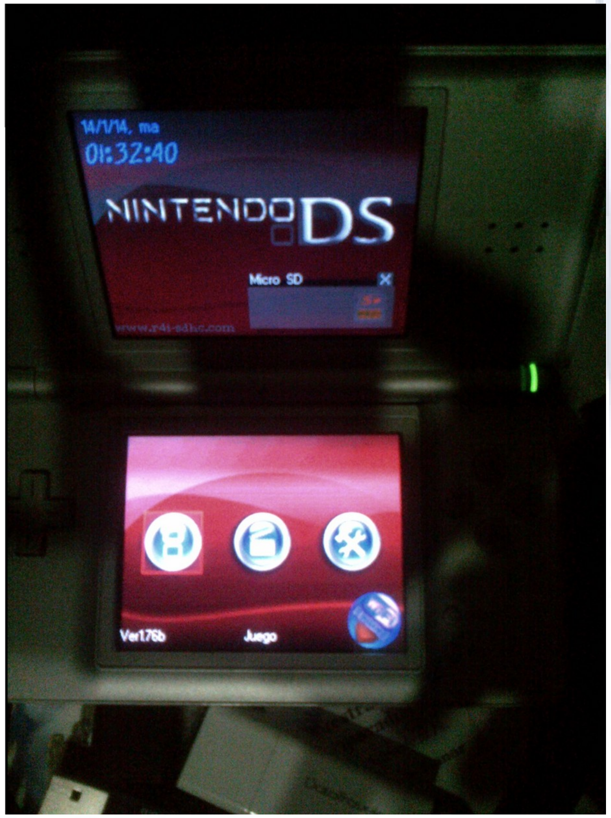
Arquitectura de la plataforma: el arranque

- Técnicas para poder ejecutar soft. no oficial
  - *Pass-through o flashme*
  - Los dispositivos (cartuchos)
    - Programa interno (navegador) que se comporta a su vez como cargador para otros programas.
    - El proceso de carga se produce de forma muy similar a como lo hace el *firmware*
      - El programa se lee de una tarjeta de memoria Flash utilizando un protocolo propio de cada fabricante

Cartuchos disponibles en prácticas

- R4 en Slot-1 NDS
  - R4i-SDHC se identifica como "Star Wars Lethal Alliance Ubisoft"

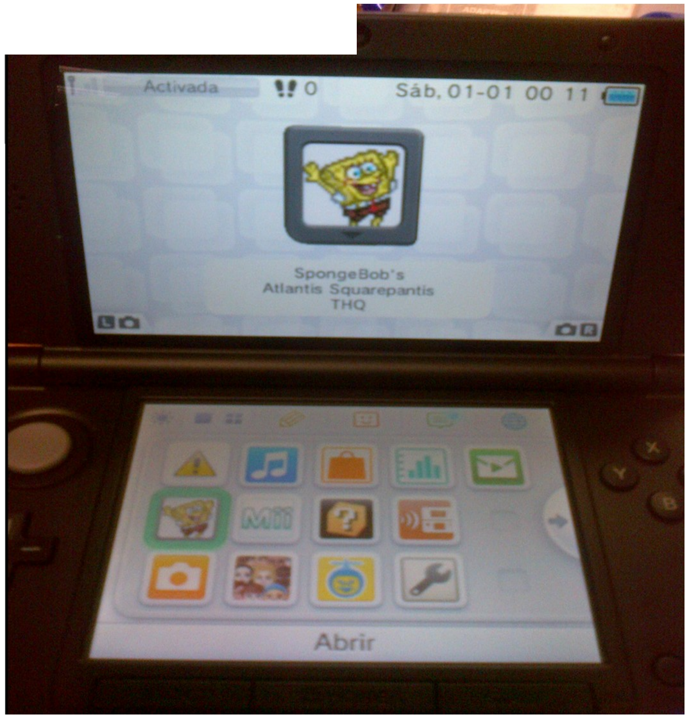




Cartuchos disponibles en prácticas (y II)

- R4 en Slot-1 NDS 3D

  - *R4 Gold Pro* se identifica como “Sponge Bob's Atlantis Squarepantis THQ”



Arquitectura software: DevkitPro + libnds

- SDK no oficial
  - Programar la consola ← “ROM” para el dispositivo de carga
    - Crear uno de esos archivos (.nds) que emulan el contenido de la memoria ROM de un cartucho comercial.
  - API básico (*libnds*) + Herramientas (*DevkitPro*)
    - ¿Herramientas? Para generar el ejecutable para la NDS
    - Si lenguaje C *arm-eabi-gcc / arm-eabi-gcc*++
      - Generación de código objeto para el ARM7 y ARM9:
    - Archivo ejecutable en un formato estándar: ELF
    - *Arm-eabi-objcopy* → generar los ejecutables reducidos *.arm7* y *.arm9*
    - *nds-tool* → cabecera + .arm7 y .arm9 + otros datos (p.ej. Gráficos) = *.nds*

Arquitectura software: DevkitPro + libnds

- Entorno de desarrollo
  - DevkitPro
    - Conjunto de librerías, compiladores y utilidades que nos permitirán el desarrollo de aplicaciones para varios tipos de consolas, incluida la NDS.
    - Incluye *libnds*, una librería que se encarga de adaptar el código C al hardware específico de la NDS.
      - Michael Noland (joat), Jason Rogers (dovoto) y Dave Murphy (WinterMute)
  - Editor de textos: donde se escribirá el código de la aplicación.
  - Emulador, depurador, ...

Arquitectura software: DevkitPro + libnds

- Estructura "general" de un desarrollo en *DevkitPro + libnds*
  - Directorio base del proyecto
    - *arm9* + *arm7* ó *source* + *Makefile*
    - *data / audio* (XM, S3M, MOD), music (IT),
    - *data / gfx* (mapas de bits, instrucciones *grit*)
    - *build* (binarios intermedios ← *mmutil*, *bin2o*, *grit*)
    - *Makefile*
  - arm?
    - source
      - *main.c* ó *main.cpp*
    - *Makefile*

Arquitectura software: DevkitPro + libnds

- Ejemplo de desarrollo
  - Escoger una plantilla

```
$cp -r ${DEVKITPRO}/examples/templates/arm9 hola
$ cd hola
```

  - Editar el fichero *main.c*

```
#include <nds.h>
 #include <stdio.h>
 void main() {
        consoleDemoInit();
        printf("\n Hola, mundo\n");
 }
```

  - Compilación, prueba (emulación) y ejecución

```
$ make
$ desmume hola.nds
```

Arquitectura software: DevkitPro + libnds

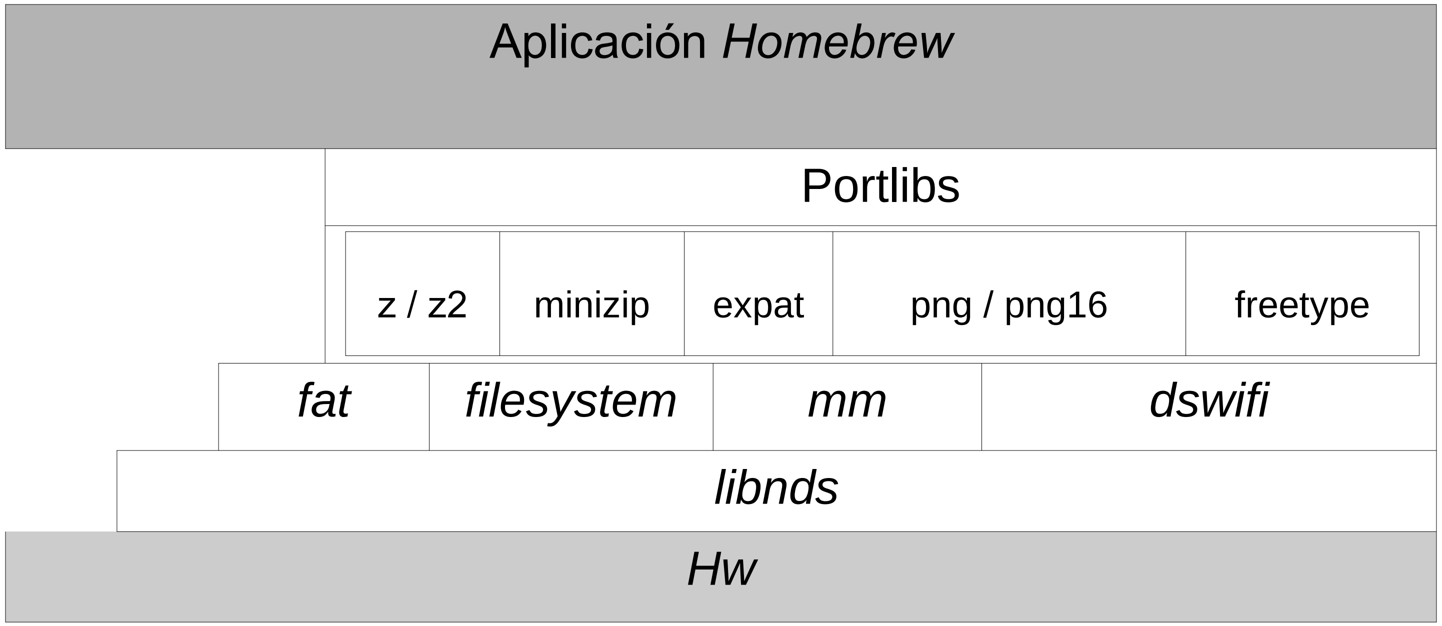
- Algunas cuestiones de bajo nivel
  - Libnds. → ndstypes.h
    - double, float → float64, float32, int16, ...
    - byte, u8, int8, s8, vs8, ...
- Declaraciones y definiciones
  - define
  - aligned vs packed
  - const vs static
  - volatile vs register
-

Arquitectura software: DevkitPro + libnds

- API libnds:
  - Secciones en la documentación
    - *2D engine API → Background + Sprites*
    - *3D engine API → OpenGL*
    - *Audio API → Simple Sound Engine +* ***MaxMod***
    - *Memory → DMA*
    - *System → Hw + Int*
    - *User Input/ouput → keyboard + touch*
    - *Utility → PCX + decompression*
    - *Custom Peripherals*
    - *Debugging → asserts + dbg messages*
  - *Doxygen* → gestión de la documentación
    - Local y en red <http://libnds.devkitpro.org/index.html>

Arquitectura software: DevkitPro + libnds

- Arquitectura de una aplicación DevkitPro para NDS: librerías de sistema y *portlibs*



Estructura de una aplicación

- Estructura mínima
  - Dos versiones del ¡Hola, mundo! dentro de los ejemplos de *libnds*
    - ${DEVKITPRO}/examples/nds/templates/arm9/source/main.c
    - ${DEVKITPRO}/examples/nds/hello_world/source/main.cpp

main.c

```
/*---------------------------------------------------------------------------------
	Basic template code for starting a DS app
---------------------------------------------------------------------------------*/
#include <nds.h>
#include <stdio.h>
//---------------------------------------------------------------------------------
int main(void) {
//---------------------------------------------------------------------------------
	consoleDemoInit();
	iprintf("Hello World!");
	while(1) {
		swiWaitForVBlank();
	}

}
```

main.cpp

```
/*---------------------------------------------------------------------------------
	$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;
}
```

Estructura de una aplicación

- Estructura mínima: v1

main.c

```
/*---------------------------------------------------------------------------------
	Basic template code for starting a DS app
---------------------------------------------------------------------------------*/
#include <nds.h>
#include <stdio.h>
//---------------------------------------------------------------------------------
int main(void) {
//---------------------------------------------------------------------------------
	consoleDemoInit();
	iprintf("Hello World!");
	while(1) {
		swiWaitForVBlank();
	}

}
```

Estructura de una aplicación

- Estructura mínima: v2

main.cpp

```
/*---------------------------------------
----------------------------------------
        $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++;
}

,,,
```

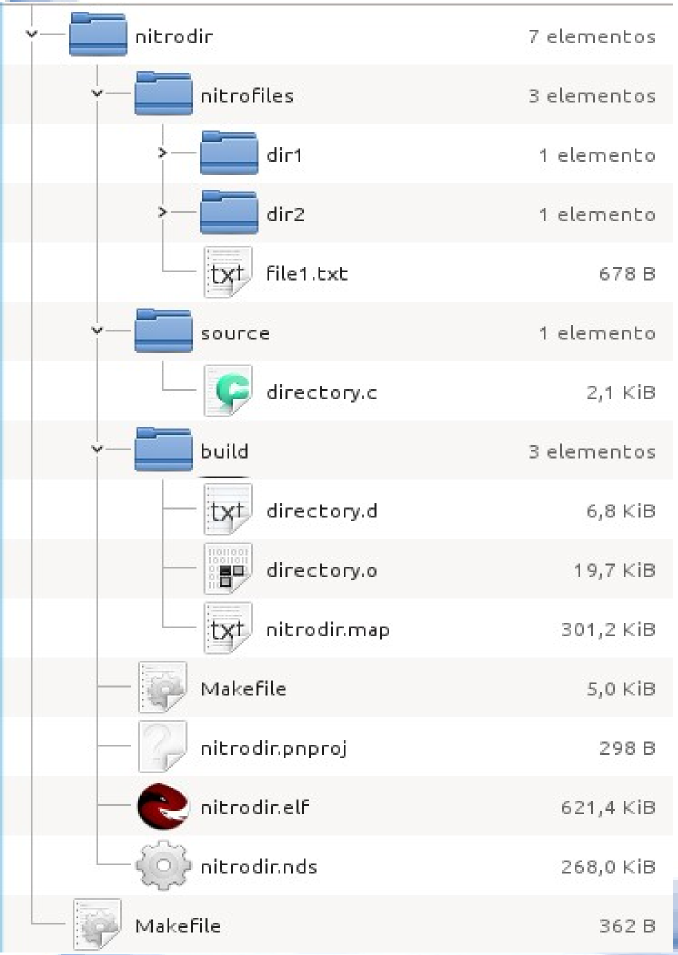
```
...
//---------------------------------------------------------------------------------
int main(void) {
//---------------------------------------------------------------------------------
        touchPosition touchXY;

        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;
}
```

Acceso al sistema de ficheros

- NitroFS
  - Sistema de solo lectura “incrustado” en el NDS
  - No es visible por el usuario del NDS
- FAT
  - Acceso r/w al sistema de ficheros de la tarjeta de memoria del cartucho (*flashcard*)
  - Versiones FAT: 12, 16, 32

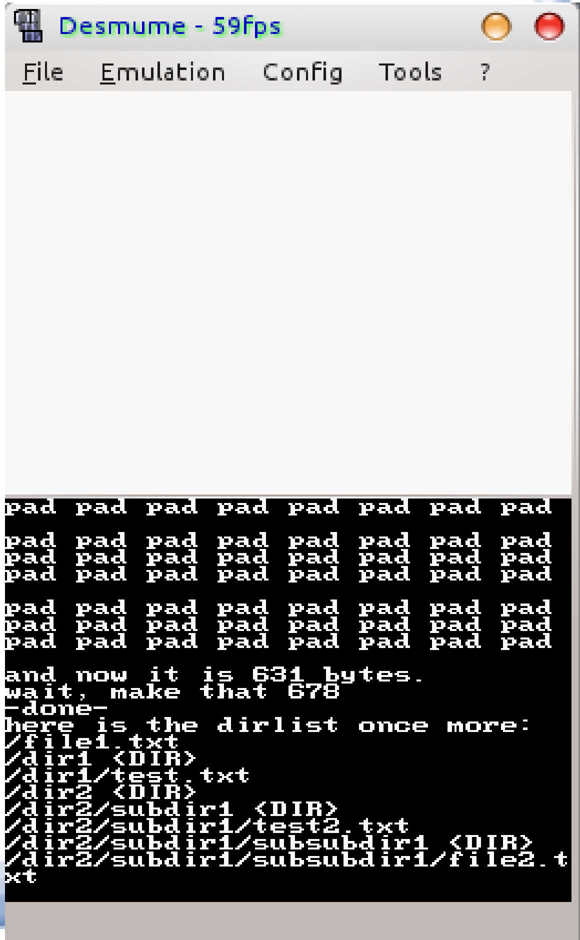
Acceso al sistema de ficheros

- NitroFS
  - ${DEVKITPRO}/examples/nds/filesystem/nitrofs



nitrodir	7 elementos
nitrofiles	3 elementos
dir1	1 elemento
dir2	1 elemento
file1.txt	678 B
source	1 elemento
directory.c	2,1 KiB
build	3 elementos
directory.d	6,8 KiB
directory.o	19,7 KiB
nitrodir.map	301,2 KiB
Makefile	5,0 KiB
nitrodir.pnproj	298 B
nitrodir.elf	621,4 KiB
nitrodir.nds	268,0 KiB
Makefile	362 B

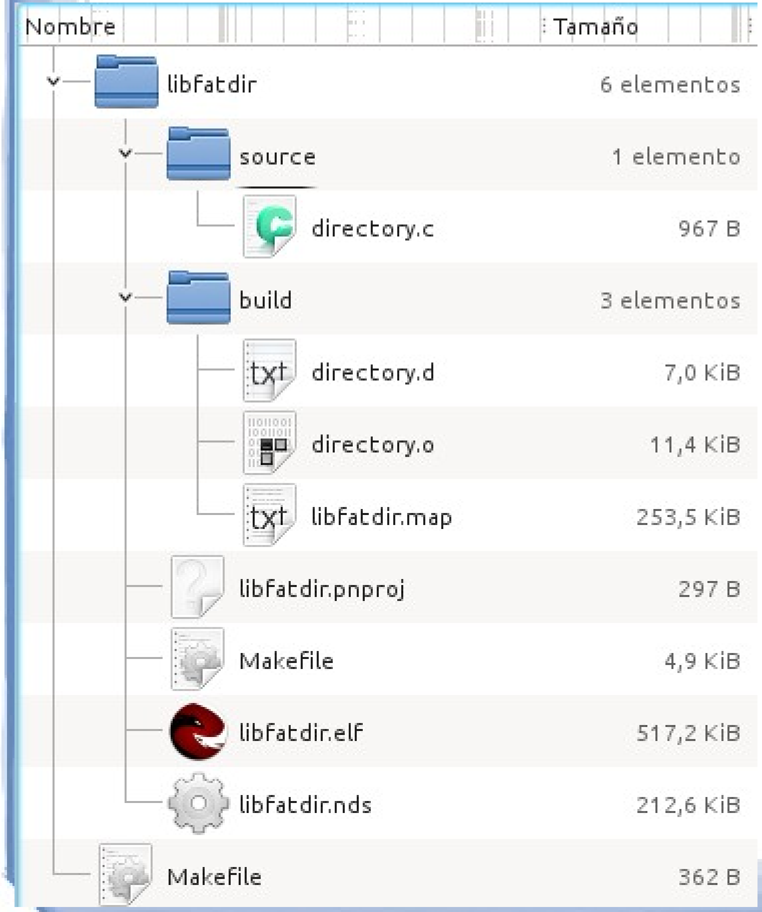
```
int main(int argc, char **argv) {
        // Initialise the console, required for printf
        consoleDemoInit();

        if (nitroFSInit(NULL)) {

          dirlist("/");
…
           FILE* inf = fopen("file1.txt","rb");
…
             fseek(inf,0,SEEK_END);
…
             if(fread(entireFile,1,len,inf) != len)
…
             fclose(inf);
        …
        } else {
                iprintf("nitroFSInit failure: terminating\n");
        }

        while(1) {
                swiWaitForVBlank();
                scanKeys();
                if(keysDown()&KEY_START) break;
        }

        return 0;
}
```



```
pad pad pad pad pad pad pad pad

pad pad pad pad pad pad pad pad
pad pad pad pad pad pad pad pad
pad pad pad pad pad pad pad pad

pad pad pad pad pad pad pad pad
pad pad pad pad pad pad pad pad
pad pad pad pad pad pad pad pad

and now it is 631 bytes.
wait, make that 678
-done-
here is the dirlist once more:
/file1.txt
/dir1 <DIR>
/dir1/test.txt
/dir2 <DIR>
/dir2/subdir1 <DIR>
/dir2/subdir1/test2.txt
/dir2/subdir1/subsubdir1 <DIR>
/dir2/subdir1/subsubdir1/file2.t
xt
```

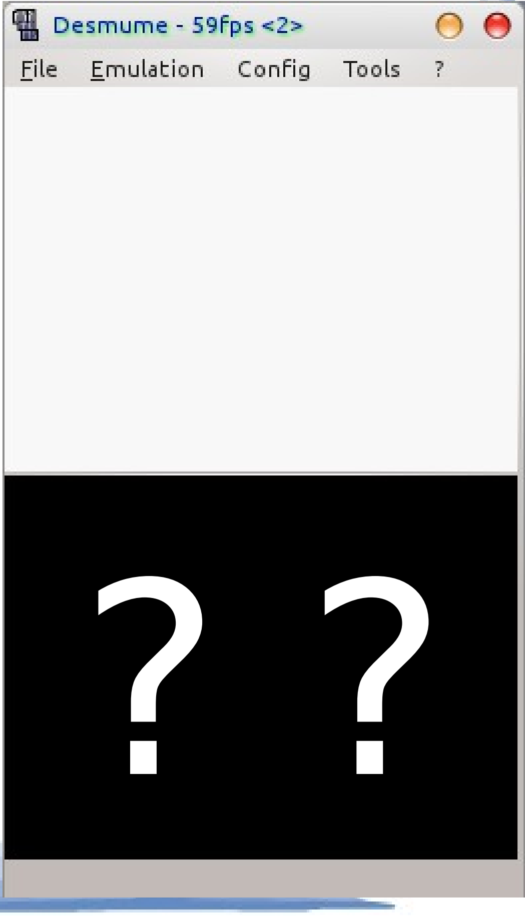
Acceso al sistema de ficheros

- FAT
  - ${DEVKITPRO}/examples/nds/filesystem/libfat



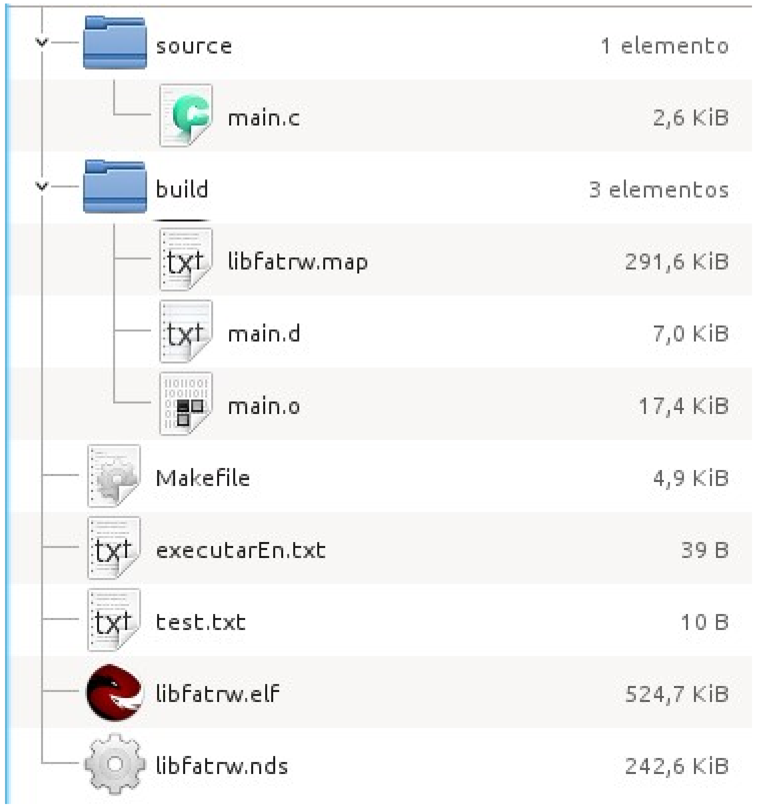
Nombre	Tamaño
libfatdir	6 elementos
source	1 elemento
directory.c	967 B
build	3 elementos
directory.d	7,0 KiB
directory.o	11,4 KiB
libfatdir.map	253,5 KiB
libfatdir.pnproj	297 B
Makefile	4,9 KiB
libfatdir.elf	517,2 KiB
libfatdir.nds	212,6 KiB
Makefile	362 B

```
int main(int argc, char **argv) {
    // Initialise the console, required for printf
     consoleDemoInit();

     if (fatInitDefault()) {
                DIR *pdir;
                struct dirent *pent;
                pdir=opendir("/");
                if (pdir){
                        while ((pent=readdir(pdir))!=NULL) {
                        if(strcmp(".", pent->d_name) == 0 ||
                         strcmp("..", pent->d_name) == 0)
                        continue;
                        if(pent->d_type == DT_DIR)
                        iprintf("[%s]\n", pent->d_name);
                        else
                        iprintf("%s\n", pent->d_name);
                        }
                        closedir(pdir);
                } else {
                        iprintf ("opendir() failure; terminating\n");
                }

        } else {
                iprintf("fatInitDefault failure: terminating\n");
        }

        while(1) {  swiWaitForVBlank();  }
        return 0;
}
```



Acceso al sistema de ficheros

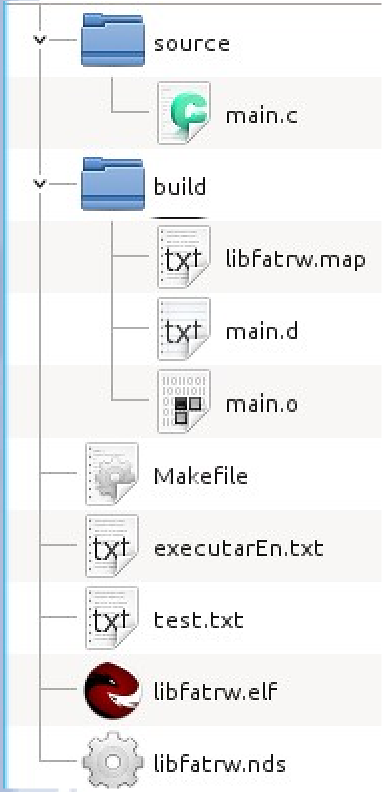
- Caso de ejemplo sobre DeSmuME
  - Contenido de un directorio en disco (…./libfatrw)



source	1 elemento
main.c	2,6 KiB
build	3 elementos
libfatrw.map	291,6 KiB
main.d	7,0 KiB
main.o	17,4 KiB
Makefile	4,9 KiB
executarEn.txt	39 B
test.txt	10 B
libfatrw.elf	524,7 KiB
libfatrw.nds	242,6 KiB

Acceso al sistema de ficheros

- Caso de ejemplo sobre DeSmuME



```
$ desmume --cflash-path=./ libfatrw.nds  → IMPTE.: --cflash-path=./
    Nintendo DS rom tool 1.50.1 - Jun 19 2012
    …
    Using CFlash directory: ./                 → crea un sistema de archivos temporal
    cflash added main.original
    cflash added libfatrw.nds
    cflash added Makefile
    cflash added source                        → entra recursivamente en directorios
    cflash added main.c
    cflash added ..                            → vuelve al directorio padre y continua
    …
    cflash added test.txt
    ...
    Trying with 1 sectors/cluster:             → decide el tipo de sistema de archivos
    FAT12: #clu=73266, fatlen=215, maxclu=4080, limit=4080
    FAT12: too much clusters
    FAT16: #clu=73124, fatlen=286, maxclu=65520, limit=65520
    FAT16: too much clusters
    FAT16: would be misdetected as FAT12
    FAT32: #clu=72562, fatlen=567, maxclu=72576, limit=268435440
    Using sector 6 as backup boot sector (0 = none)
    ...
    DeSmuME .dsv save file not found. Trying to load an old raw .sav file.
    Missing save file /home/magusti/.config/desmume/libfatrw.dsv

    $.
```

Acceso al sistema de ficheros

- Caso de ejemplo sobre DeSmuME
  - Código fuente

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

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <dirent.h>

int main(int argc, char **argv) {
   FILE *fp;
   u8 c;
   consoleDemoInit();
   printf("Init FAT: fatInitDefault!\n");
   if(!fatInitDefault())
               printf("Init FAT: Error!\n");
   else
  {
     printf("Init FAT: conseguit!\n");
     // Ejemplo de acceso en modo texto: lectura
     printf("Operaciones en modo texto\nLlegint fat:/test.txt\n");
     fp = fopen("fat:/test.txt", "r");
      if(!fp)
            printf("Llegint fat:/test.txt: Error!\n");
     else
       {
                while (!feof( fp )) {
                  c = fgetc(fp);
                  printf("%c", c);
        }
       printf("\n");
       fclose(fp);
   ...
```

```
…
// Ejemplo de acceso en modo texto: escritura
 printf("Escribint fat:/test.txt\n");
 fp = fopen("fat:/test.txt", "w");
 if(!fp)
           printf("Escribint fat:/test.txt: Error!\n");
  else
  {
          fprintf(fp, "%s\n", "Manolo" );
          printf("\n");
          fclose(fp);
  }

         // Comprobación
         // Recordar que al salir de desmume no permanecen los cambios
         fp = fopen("fat:/test.txt", "r");
         if(!fp)
           printf("Tornant a llegir fat:/test.txt: Error!\n");
         else
         {
          while (!feof( fp )) {
            c = fgetc(fp);
            printf("%c", c);
          }
          printf("\n");
          fclose(fp);
         }
...
```

Acceso al sistema de ficheros

- Caso de ejemplo sobre DeSmuME
  - Código fuente (y II)

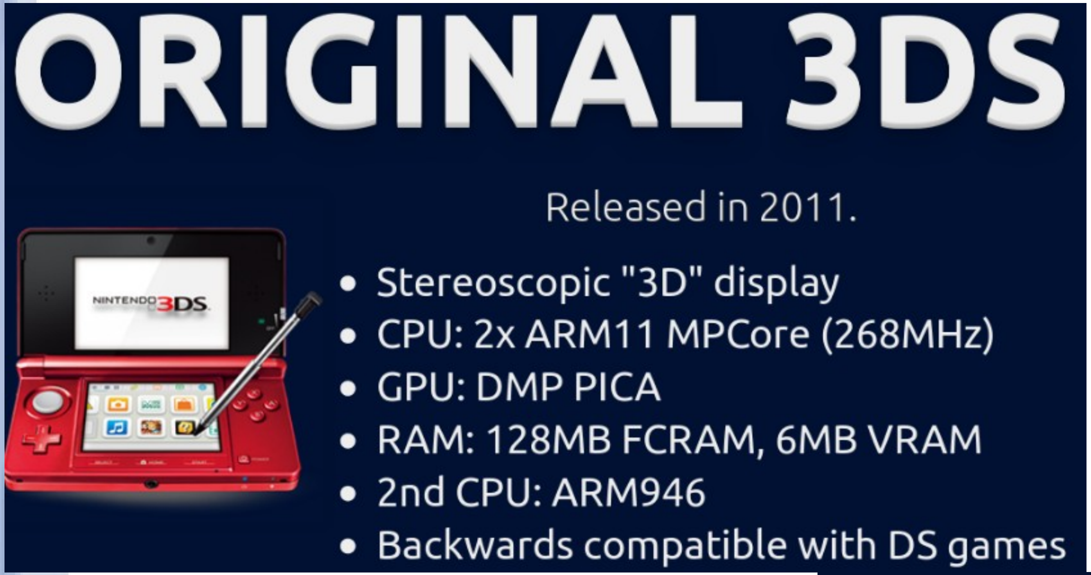
```
     printf("Init FAT: fatInitDefault!\n");

        if(!fatInitDefault())
          printf("Init FAT: Error!\n");
        else
        {
         …
          // Acceso en modo binario
          fp = fopen("fat:/Tetris.sav", "wb");
          if(!fp) printf("Writing fat:/Tetris.sav: Error!\n");
                            else
                            {
                              printf("Write something to fat:/Tetris.sav\n");
                              fputc(0x78, fp);
                              fclose(fp);
                            }

         fp = fopen("fat:/Tetris.sav", "rb");
          if(!fp) printf("Reading fat:/Tetris.sav: Error!\n");
                            else
                            {
                              u8 c = fgetc(fp);
                              fclose(fp);
                              printf("Char read: 0x%02X\n", c);
                            }

        }

        while(1) {  swiWaitForVBlank();  }
        return 0;
}
```

Arquitectura de la 3DS

- Hardware
  - Componentes
  - Esquema de bloques
- Software
  - SDK

Imágenes obtenidas de Jason Fitzpatrick on Lifehacker, 7/23/10 9:00am, “Supercharge Your Homebrew-Hacked Nintendo DS” < http://lifehacker.com/5594640/supercharge-your-homebrew+hacked-nintendo-ds> y de Nintendo <http://www.nintendo.com>.

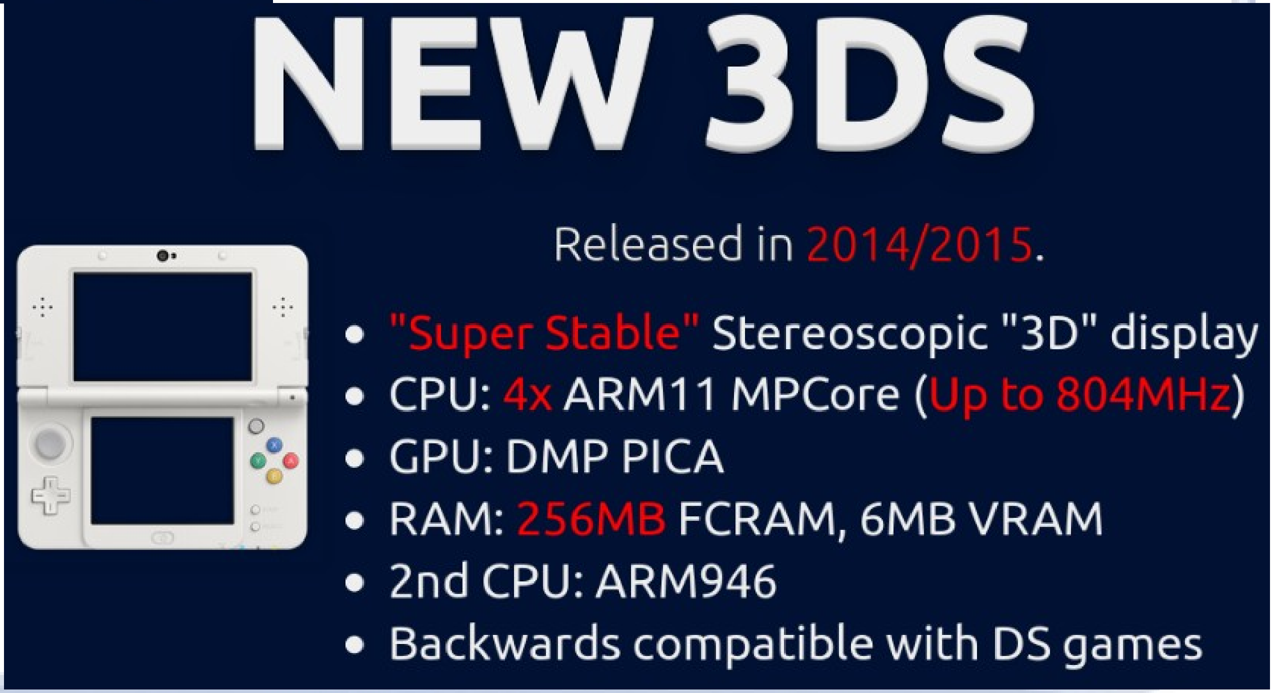
Descripción oficial de componentes



ORIGINAL 3DS

Released in 2011.

- Stereoscopic "3D" display
- CPU: 2x ARM11 MPCore (268MHz)
- GPU: DMP PICA
- RAM: 128MB FCRAM, 6MB VRAM
- 2nd CPU: ARM946
- Backwards compatible with DS games



NEW 3DS

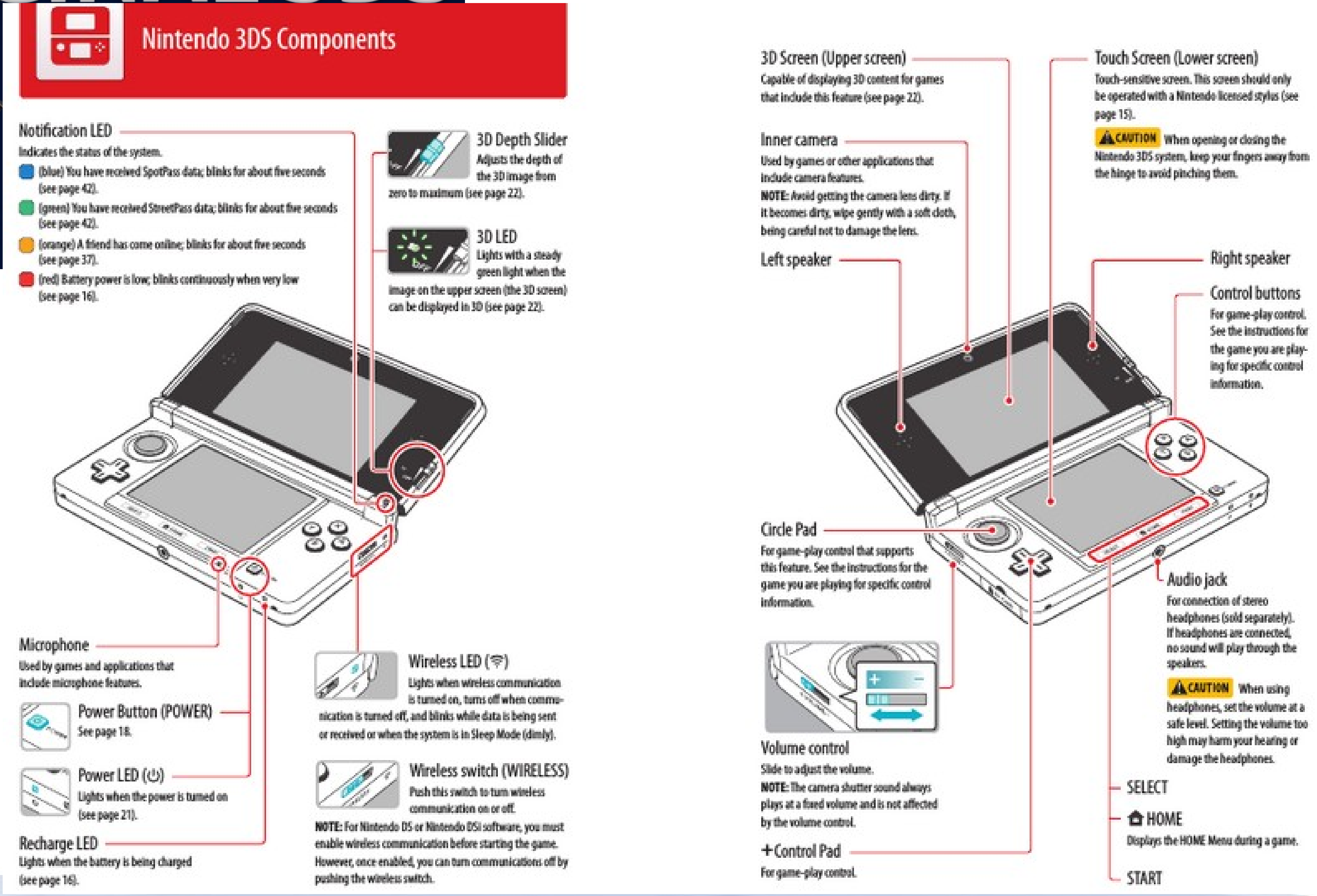
Released in 2014/2015.

- "Super Stable" Stereoscopic "3D" display
- CPU: 4x ARM11 MPCore (Up to 804MHz)
- GPU: DMP PICA
- RAM: 256MB FCRAM, 6MB VRAM
- 2nd CPU: ARM946
- Backwards compatible with DS games

Imágenes obtenidas de "Breaking the 3DS security system": <https://smealum.github.io/3ds/32c3/#/91>

Descripción oficial de componentes

ORIGINAL 3DS



Nintendo 3DS Components

Notification LED
Indicates the status of the system.
- 🟦 (blue) You have received SpotPass data; blinks for about five seconds (see page 42).
- 🟩 (green) You have received StreetPass data; blinks for about five seconds (see page 42).
- 🟧 (orange) A friend has come online; blinks for about five seconds (see page 37).
- 🟥 (red) Battery power is low; blinks continuously when very low (see page 16).

3D Depth Slider
Adjusts the depth of the 3D image from zero to maximum (see page 22).

3D LED
Lights with a steady green light when the image on the upper screen (the 3D screen) can be displayed in 3D (see page 22).

Microphone
Used by games and applications that include microphone features.

Power Button (POWER)
See page 18.

Power LED (⏻)
Lights when the power is turned on (see page 21).

Recharge LED
Lights when the battery is being charged (see page 16).

Wireless LED (ᯤ)
Lights when wireless communication is turned on, turns off when communication is turned off, and blinks while data is being sent or received or when the system is in Sleep Mode (dimly).

Wireless switch (WIRELESS)
Push this switch to turn wireless communication on or off.
NOTE: For Nintendo DS or Nintendo DSi software, you must enable wireless communication before starting the game. However, once enabled, you can turn communications off by pushing the wireless switch.

3D Screen (Upper screen)
Capable of displaying 3D content for games that include this feature (see page 22).

Inner camera
Used by games or other applications that include camera features.
NOTE: Avoid getting the camera lens dirty. If it becomes dirty, wipe gently with a soft cloth, being careful not to damage the lens.

Left speaker

Touch Screen (Lower screen)
Touch-sensitive screen. This screen should only be operated with a Nintendo licensed stylus (see page 15).
⚠ CAUTION When opening or closing the Nintendo 3DS system, keep your fingers away from the hinge to avoid pinching them.

Right speaker

Control buttons
For game-play control. See the instructions for the game you are playing for specific control information.

Circle Pad
For game-play control that supports this feature. See the instructions for the game you are playing for specific control information.

Volume control
Slide to adjust the volume.
NOTE: The camera shutter sound always plays at a fixed volume and is not affected by the volume control.

+Control Pad
For game-play control.

Audio jack
For connection of stereo headphones (sold separately). If headphones are connected, no sound will play through the speakers.
⚠ CAUTION When using headphones, set the volume at a safe level. Setting the volume too high may harm your hearing or damage the headphones.

SELECT

⌂ HOME
Displays the HOME Menu during a game.

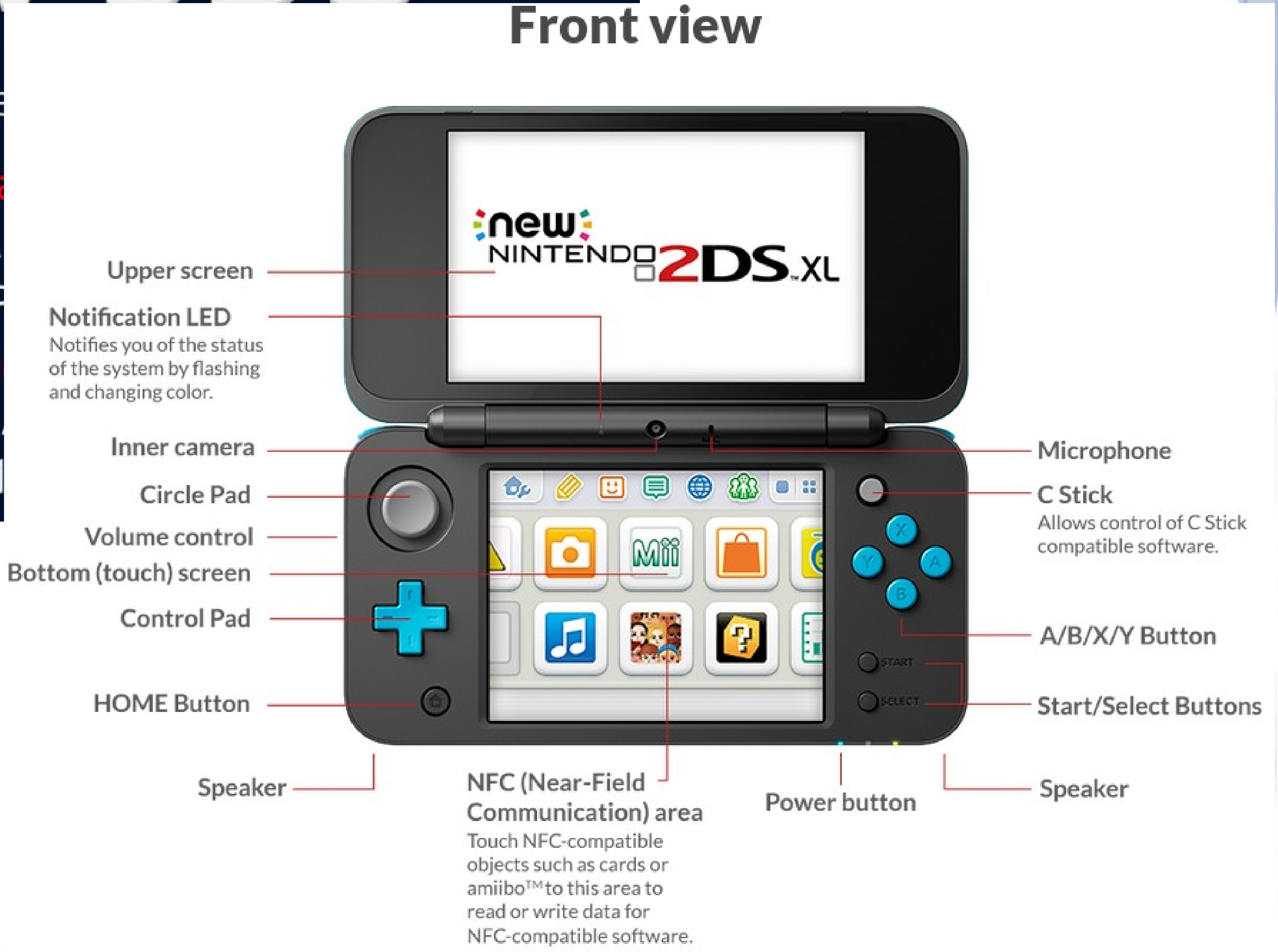
START

Imágenes obtenidas de "Breaking the 3DS security system": <https://smealum.github.io/3ds/32c3/#/91>

NEW 3DS

Re

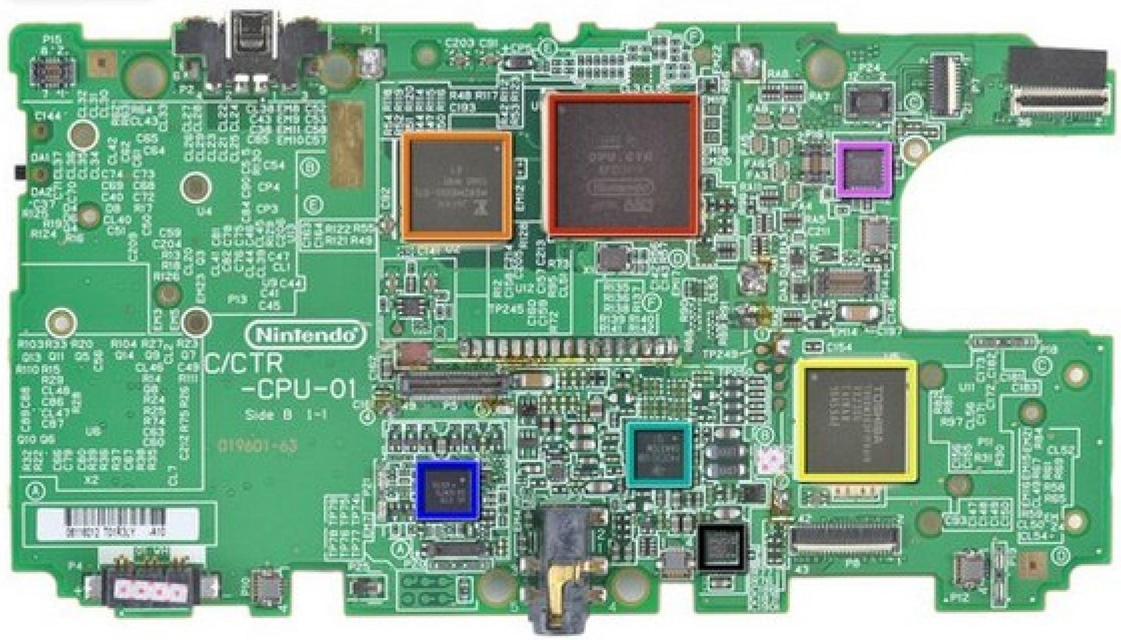
- "Super Sta
- CPU: 4x A
- GPU: DMF
- RAM: 256
- 2nd CPU: /
- Backward


Front view
new NINTENDO 2DS XL
Upper screen
Notification LED
Notifies you of the status of the system by flashing and changing color.
Inner camera
Circle Pad
Volume control
Bottom (touch) screen
Control Pad
HOME Button
Speaker
NFC (Near-Field Communication) area
Touch NFC-compatible objects such as cards or amiibo™ to this area to read or write data for NFC-compatible software.
Power button
Speaker
Microphone
C Stick
Allows control of C Stick compatible software.
A/B/X/Y Button
Start/Select Buttons
START
SELECT

Imágenes de Imagen de "Features – Nintendo 3DS <https://www.nintendo.com/3ds/features>
y "Breaking the 3DS security system": <https://smealum.github.io/3ds/32c3/#/91>

Arquitectura de la 3DS

- Hardware
  - Componentes

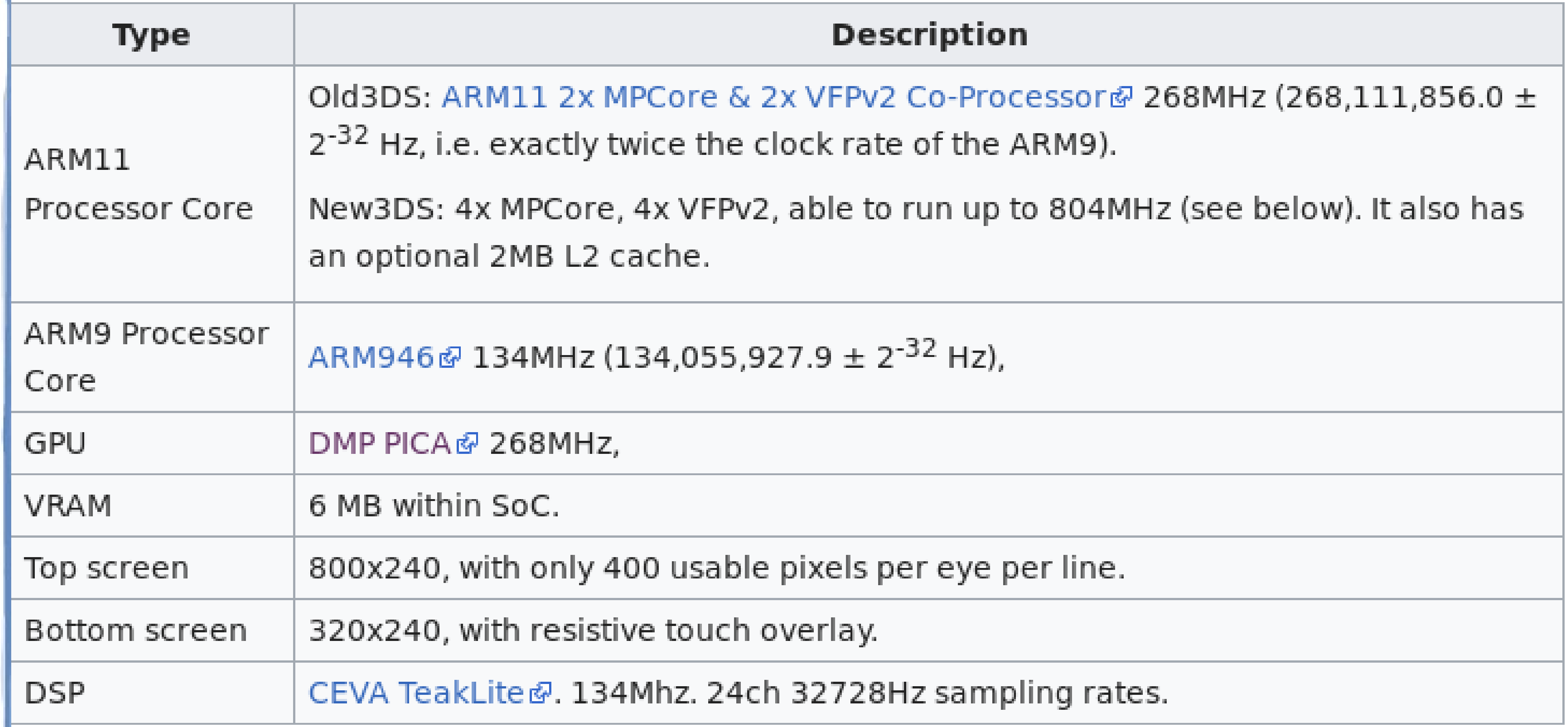


- Chips we've found inside the 3DS motherboard (click here for high-res version):
  - 🔴 Nintendo 1048 0H ARM CPU
  - 🟠 Fujitsu MB82M8080-07L 128MB FC-RAM
  - 🟡 Toshiba THGBM2G3P1FBAI8 2 GB NAND Flash
  - 🟢 Texas Instruments PAIC3010B 0AA37DW
  - 🔵 UC CTR 041KM73 KG10
  - 🟣 Invensense ITG-3270 MEMS Gyroscope
  - ⚫ ST Micro 2048 33DH X1MAQ Accelerometer Model LIS331DH

Imágenes obtenidas Ifixit – Nitendo 3DS teardown <https://es.ifixit.com/Desmontaje/Nintendo+3DS+Teardown/5029>.

Arquitectura de la 3DS

- Hardware
  - Componentes



Type	Description
ARM11 Processor Core	Old3DS: ARM11 2x MPCore & 2x VFPv2 Co-Processor 268MHz (268,111,856.0 ± 2^{-32} Hz, i.e. exactly twice the clock rate of the ARM9). New3DS: 4x MPCore, 4x VFPv2, able to run up to 804MHz (see below). It also has an optional 2MB L2 cache.
ARM9 Processor Core	ARM946 134MHz (134,055,927.9 ± 2^{-32} Hz),
GPU	DMP PICA 268MHz,
VRAM	6 MB within SoC.
Top screen	800x240, with only 400 usable pixels per eye per line.
Bottom screen	320x240, with resistive touch overlay.
DSP	CEVA TeakLite. 134Mhz. 24ch 32728Hz sampling rates.

Imágenes obtenidas 3DBrew<http://www.3dbrew.org>.

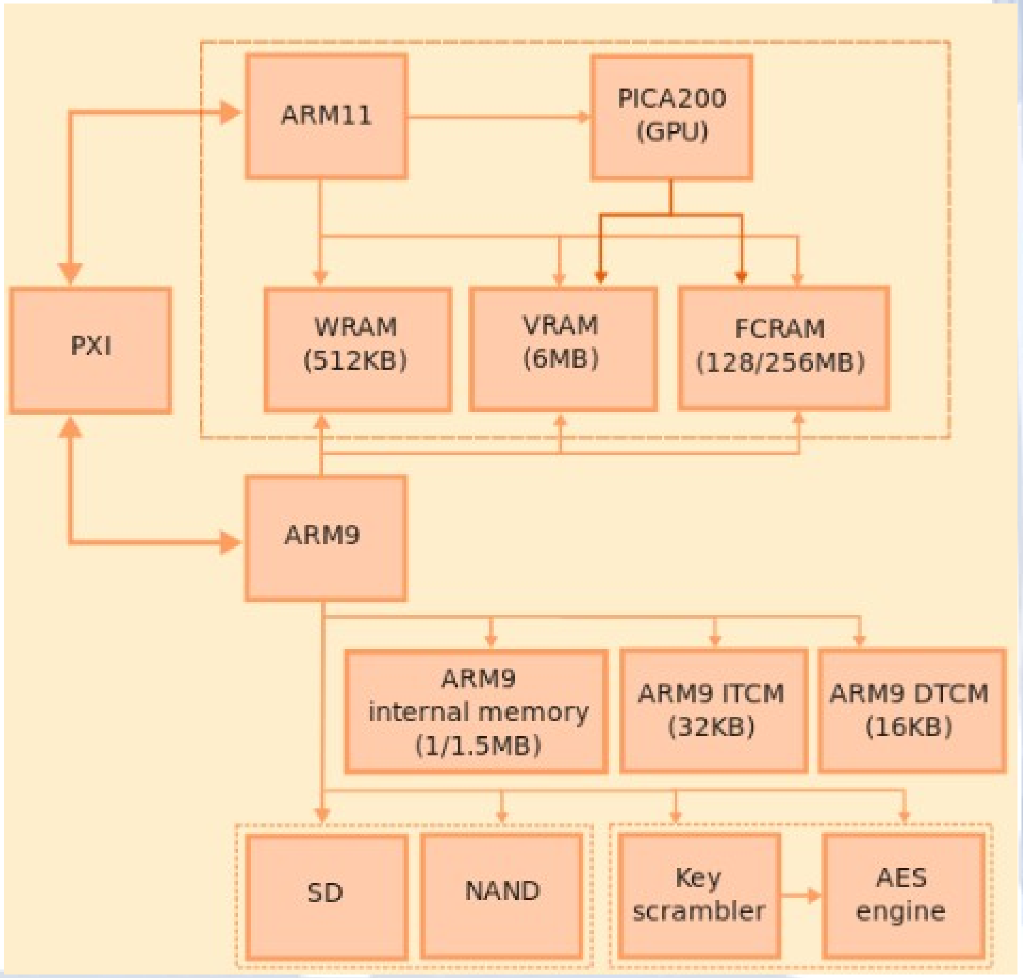
Esquema de bloques

- Conexión entre elementos
  - Procesadores
    - *ARM11 (2-cores) (quad-core en N3DS)*
      - *"syscore" & "appcore" → multiprocessor OS*
    - *ARM9 → 1 proceso (process9)*
  - Gestión de memoria
    - ARM11
      - WRAM (512 KB, ARM11 kernel, cache?)
      - VRAM (6MB, vídeo)
      - FCRAM (128/256 MB,
        - app., system (applets), base (system modules & tables)
    - ARM9
      - Internal, memory (1/1.5MB)+ ITCM (32KB) + DTCM (16KB)
      - SD, NAND, KeyScrambler, AES Engine
  - Subsistemas
    - Audio
    - "Aceleradores" gráficos

Imágenes obtenidas de "Breaking the 3DS security system": <https://smealum.github.io/3ds/32c3/>

Arquitectura de la 3DS

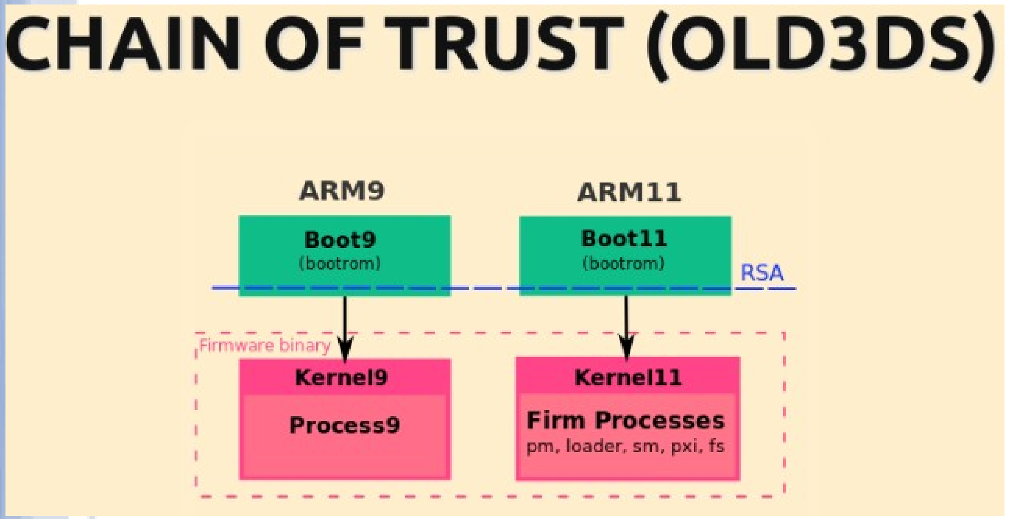
- Esquema de bloques del *Hardware*
  - *ARM11*
    - *Multitask OS: "syscore" & "appcore"*
  - *ARM9*
    - *"Process9"*
    - *Storage media*
    - *Crypto ops.*

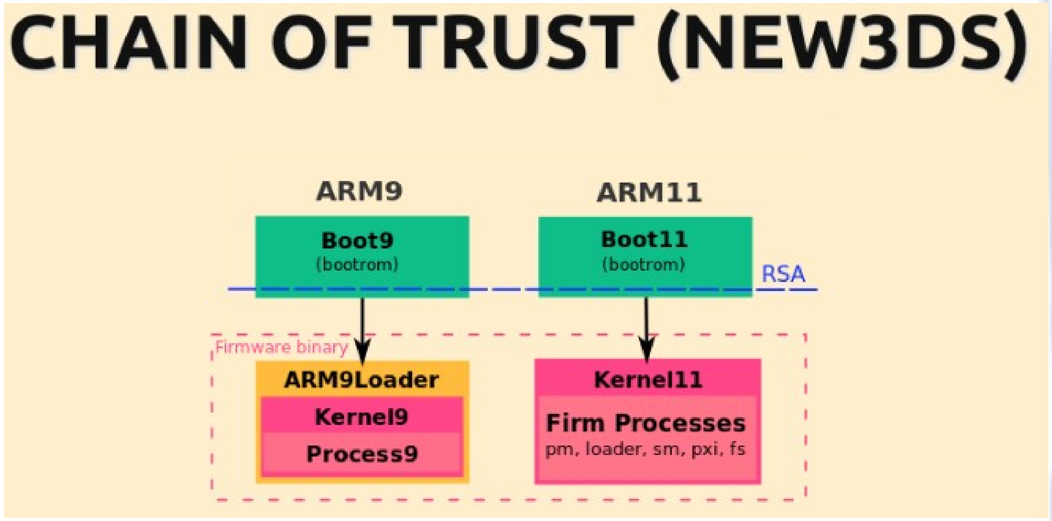


Imágenes obtenidas de "Breaking the 3DS security system": <https://smealum.github.io/3ds/32c3/#/5>

Arquitectura de la plataforma: el arranque

- El arranque de la consola 3DS / N3DS
  - BIOS
    - Arranque de los dos procesadores (encriptado)
  - *Firmware*

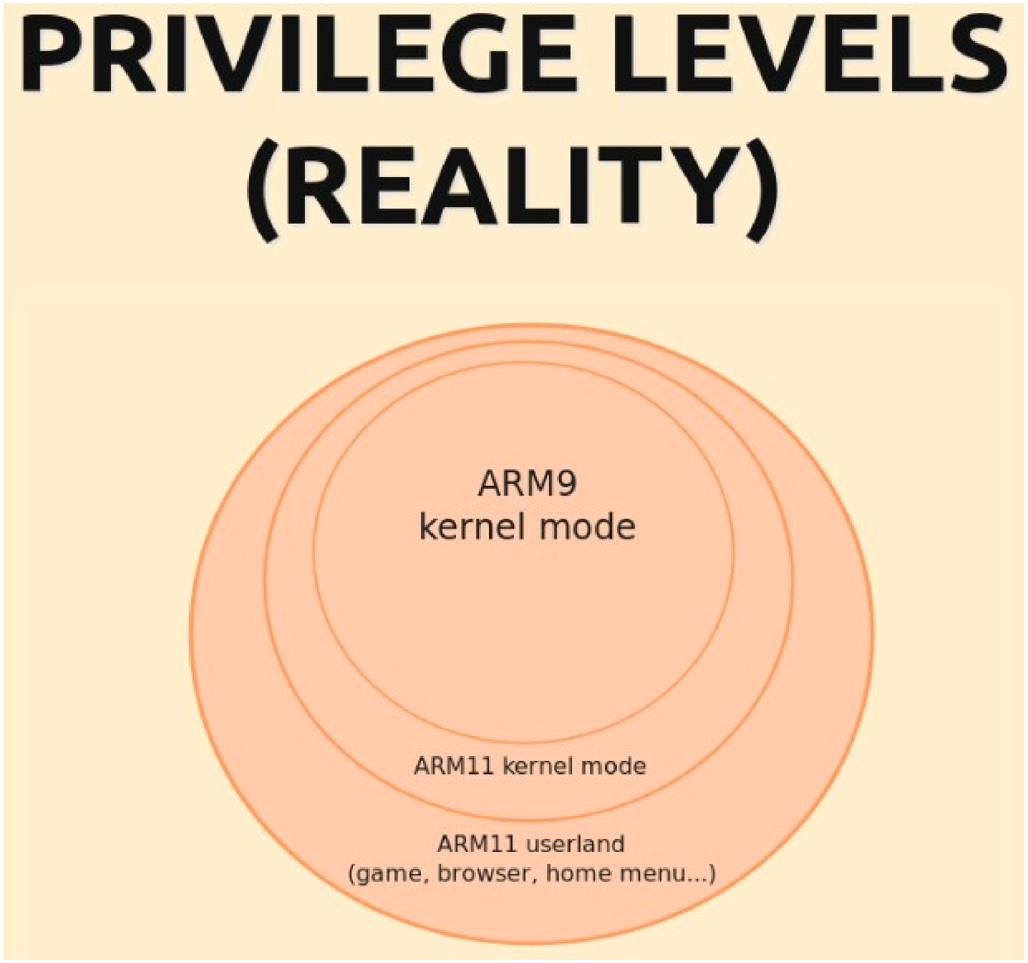




Imágenes obtenidas de "Breaking the 3DS security system": <https://smealum.github.io/3ds/32c3/#/91>

Arquitectura de la plataforma: el arranque

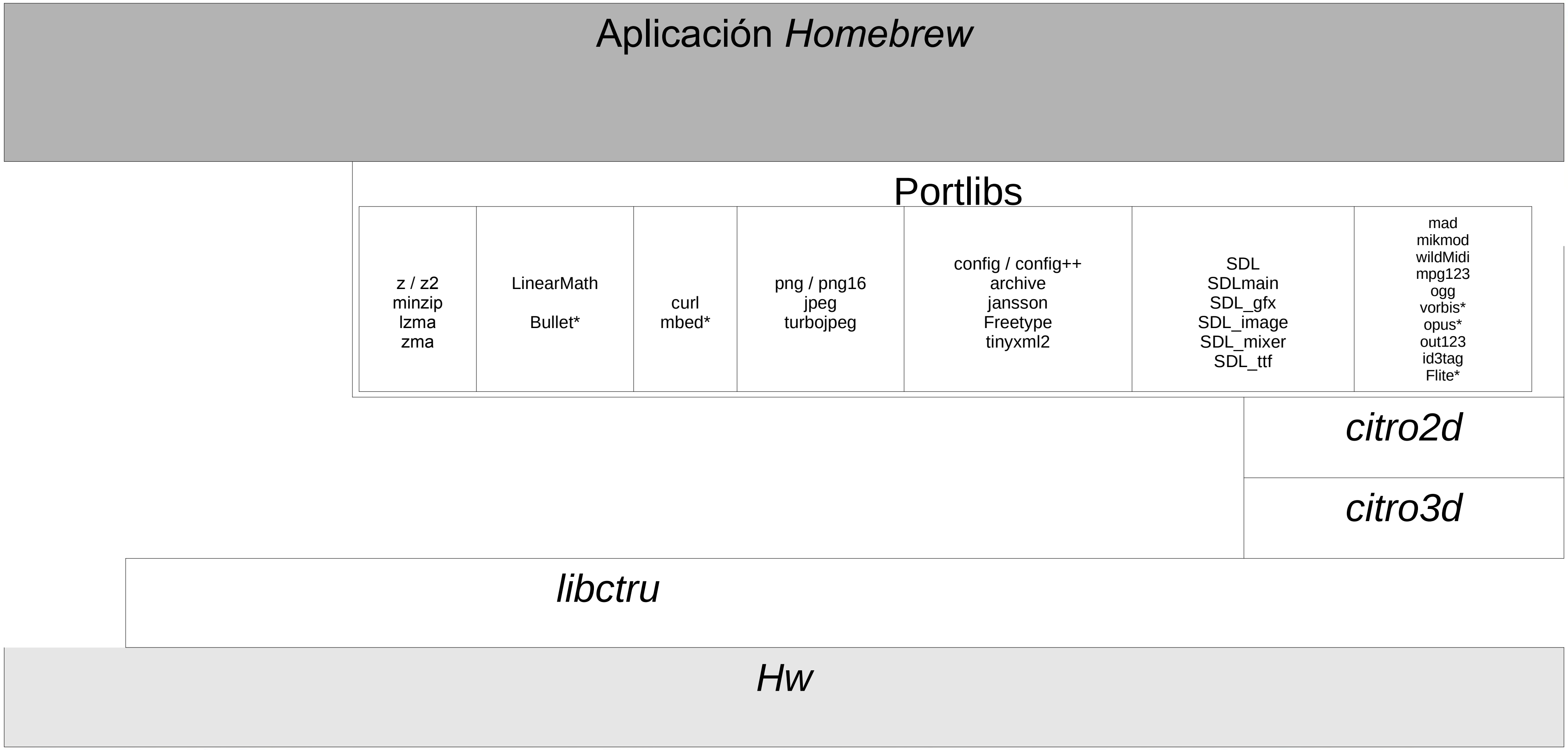
- El arranque de la consola 3DS / N3DS


PRIVILEGE LEVELS (REALITY)
ARM9 kernel mode
ARM11 kernel mode
ARM11 userland (game, browser, home menu...)

Imágenes obtenidas de "Breaking the 3DS security system": <https://smealum.github.io/3ds/32c3/#/19>

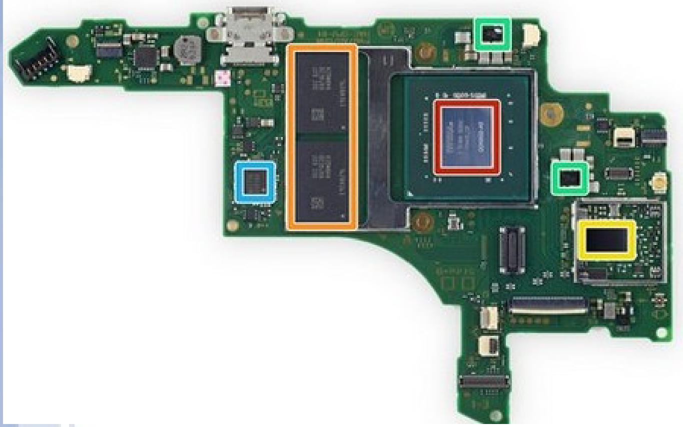
Arquitectura software: devkitARM + libctru

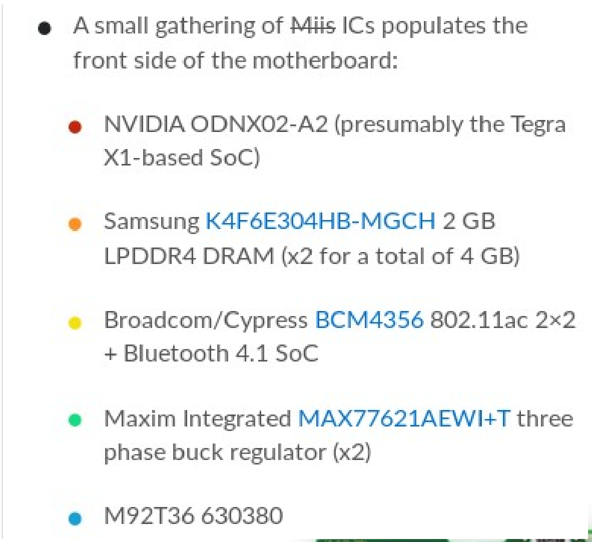
- Para 3DS: librerías de sistema y *portlibs*



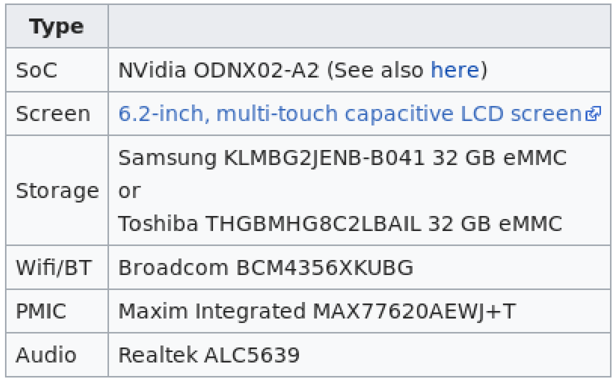
Arquitectura de la *Switch*

- Hardware
  - Componentes:
    It... looks like a computer! Battery, heatpipe, thermal paste, fan. It's all there.

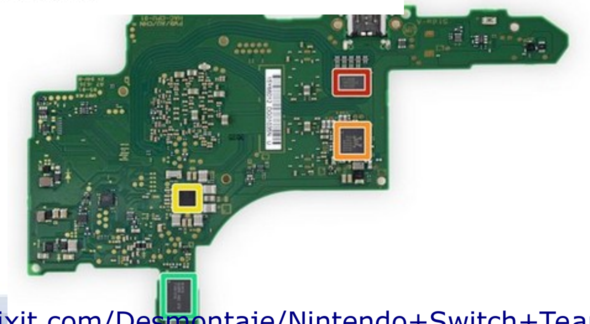


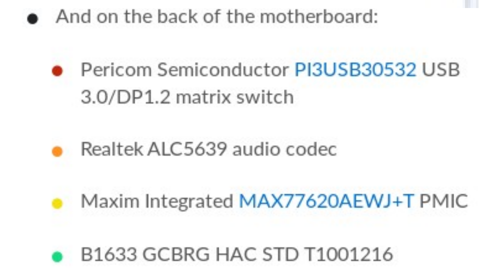


- A small gathering of ~~Miis~~ ICs populates the front side of the motherboard:
  - ● NVIDIA ODNX02-A2 (presumably the Tegra X1-based SoC)
  - ● Samsung K4F6E304HB-MGCH 2 GB LPDDR4 DRAM (x2 for a total of 4 GB)
  - ● Broadcom/Cypress BCM4356 802.11ac 2×2 + Bluetooth 4.1 SoC
  - ● Maxim Integrated MAX77621AEWI+T three phase buck regulator (x2)
  - ● M92T36 630380



Type	
SoC	NVidia ODNX02-A2 (See also here)
Screen	6.2-inch, multi-touch capacitive LCD screen
Storage	Samsung KLMBG2JENB-B041 32 GB eMMC or Toshiba THGBMHG8C2LBAIL 32 GB eMMC
Wifi/BT	Broadcom BCM4356XKUBG
PMIC	Maxim Integrated MAX77620AEWJ+T
Audio	Realtek ALC5639

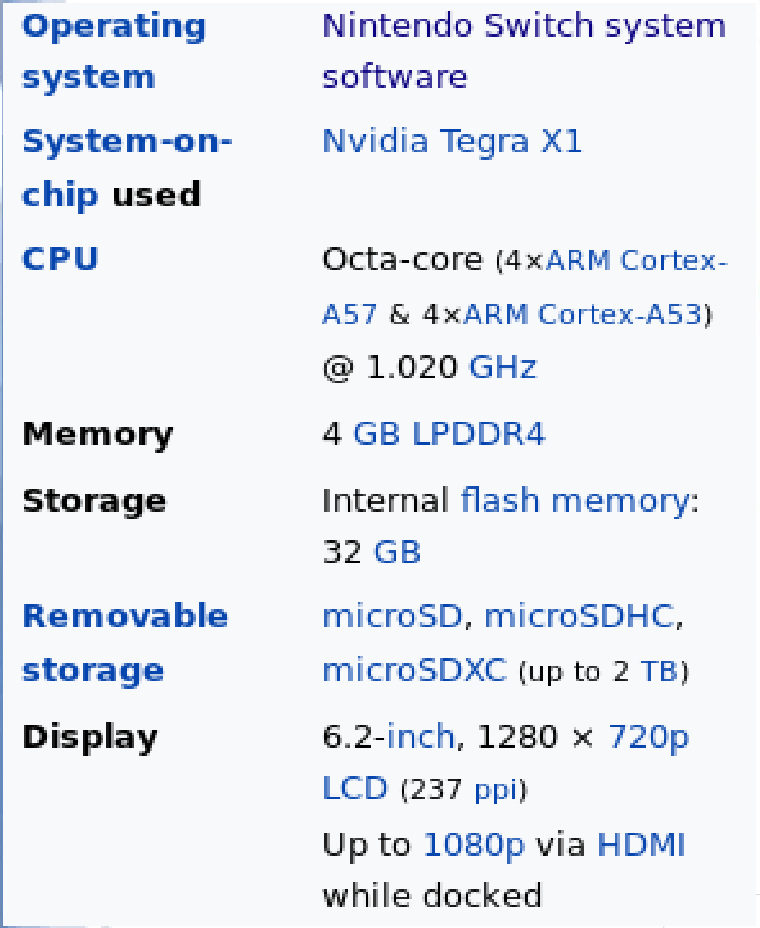




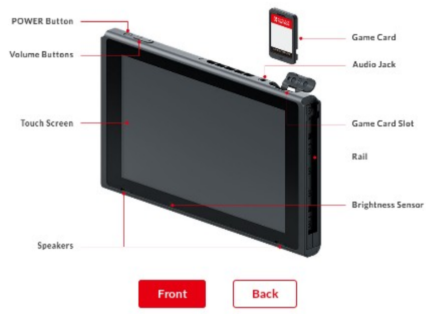
- And on the back of the motherboard:
  - ● Pericom Semiconductor PI3USB30532 USB 3.0/DP1.2 matrix switch
  - ● Realtek ALC5639 audio codec
  - ● Maxim Integrated MAX77620AEWJ+T PMIC
  - ● B1633 GCBRG HAC STD T1001216

Imágenes obtenidas Ifixit – Nitendo Switch teardown <https://es.ifixit.com/Desmontaje/Nintendo+Switch+Teardown/78263? > y switchbrew.org <https://switchbrew.org/wiki/Hardware>.

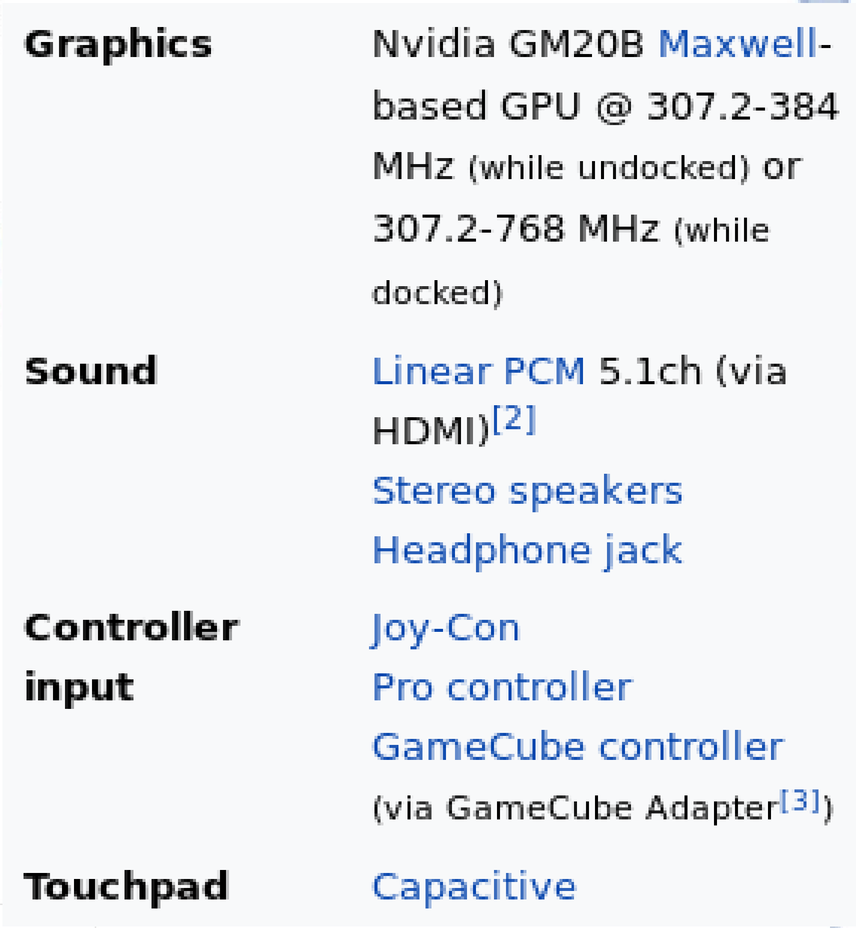
Arquitectura de la *Switch* (III)



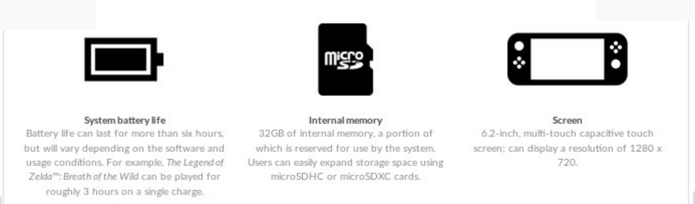
Operating system	Nintendo Switch system software
System-on-chip used	Nvidia Tegra X1
CPU	Octa-core (4×ARM Cortex-A57 & 4×ARM Cortex-A53) @ 1.020 GHz
Memory	4 GB LPDDR4
Storage	Internal flash memory: 32 GB
Removable storage	microSD, microSDHC, microSDXC (up to 2 TB)
Display	6.2-inch, 1280 × 720p LCD (237 ppi) Up to 1080p via HDMI while docked



System details



Graphics	Nvidia GM20B Maxwell-based GPU @ 307.2-384 MHz (while undocked) or 307.2-768 MHz (while docked)
Sound	Linear PCM 5.1ch (via HDMI)[2] Stereo speakers Headphone jack
Controller input	Joy-Con Pro controller GameCube controller (via GameCube Adapter[3])
Touchpad	Capacitive



System battery life
Battery life can last for more than six hours, but will vary depending on the software and usage conditions. For example, *The Legend of Zelda™: Breath of the Wild* can be played for roughly 3 hours on a single charge.

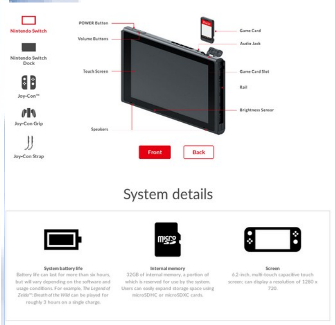
Internal memory
32GB of internal memory, a portion of which is reserved for use by the system. Users can easily expand storage space using microSDHC or microSDXC cards.

Screen
6.2-inch, multi-touch capacitive touch screen; can display a resolution of 1280 x 720.

Imágenes obtenidas de Wikipedia – Nitendo Switch <https://en.wikipedia.org/wiki/Nintendo_switch>.

Arquitectura de la *Switch* (IV)

- Hardware
  - Consola: multitáctil 6.2”, 1280 x 720, NVIDIA Custom Tegra processor, Wi-Fi, Bluetooth 4.1, acelerómetro, giróscopo y sensor de brillo
  - Joy-Con: Bluetooth 3.0/NFC, acelorómetro, giróscopo. *Motion IR Camera*, HD Rumble
  - Complementos
- Software
  - SDK

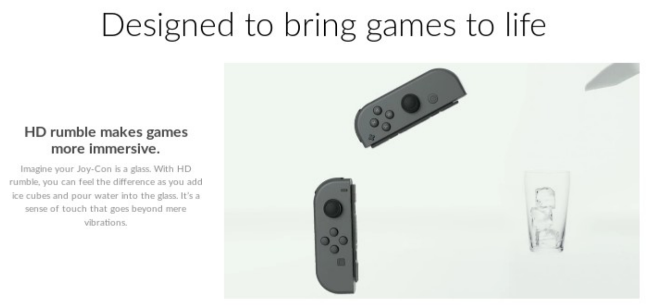


System details

System battery life
Battery life can last for more than six hours, but will vary depending on the software and usage conditions. For example, *The Legend of Zelda™: Breath of the Wild* can be played for roughly 3 hours on a single charge.

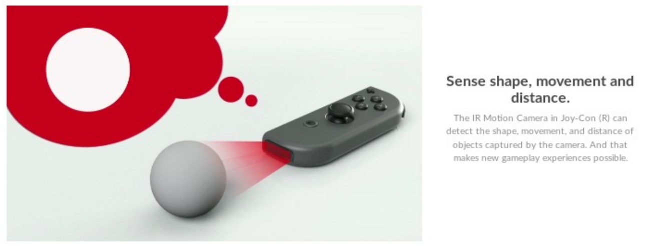
Internal memory
32GB of internal memory, a portion of which is reserved for use by the system. Users can easily expand storage space using microSDHC or microSDXC cards.

Screen
6.2-inch, multi-touch capacitive touch screen; can display a resolution of 1280 x 720.



Designed to bring games to life

HD rumble makes games more immersive.
Imagine your Joy-Con is a glass. With HD rumble, you can feel the difference as you add ice cubes and pour water into the glass. It's a sense of touch that goes beyond mere vibrations.

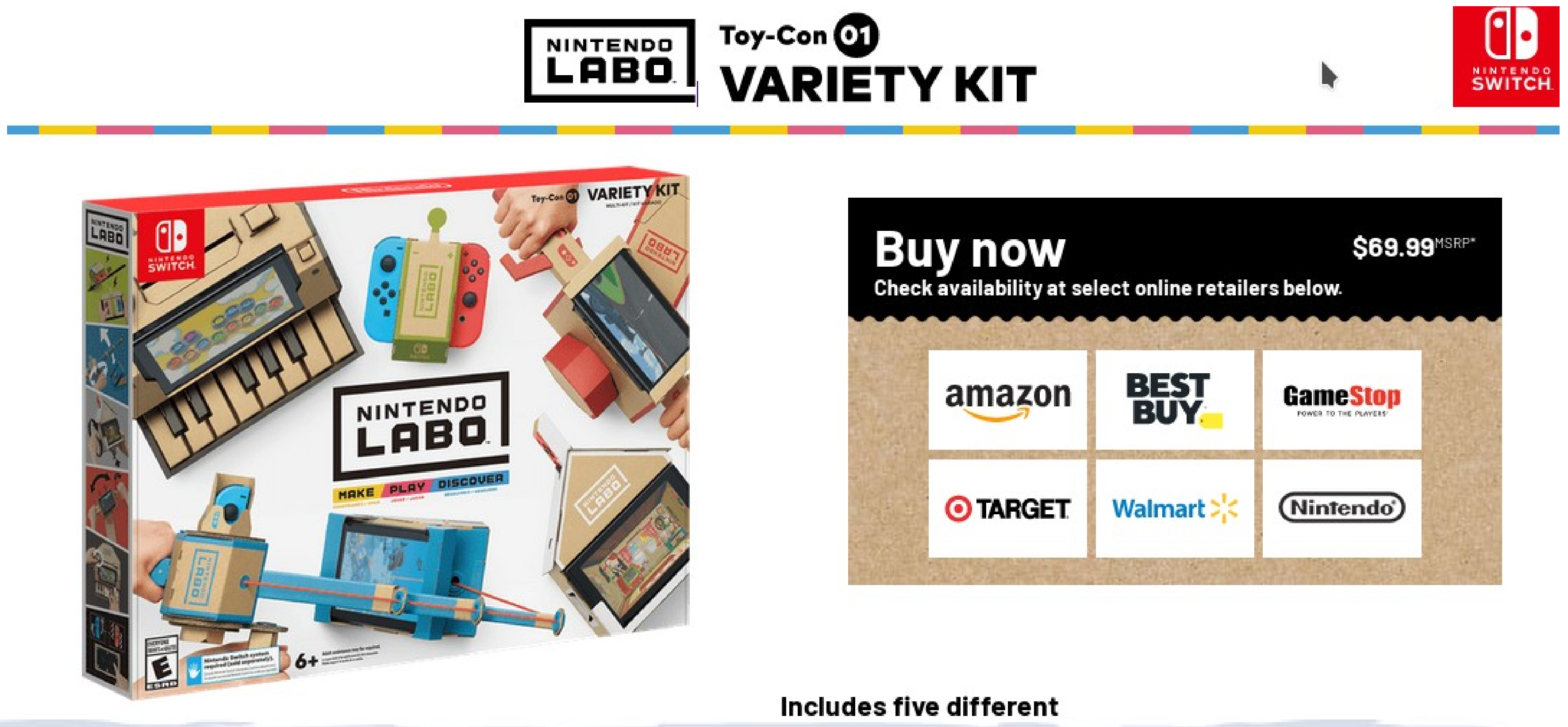


Sense shape, movement and distance.
The IR Motion Camera in Joy-Con (R) can detect the shape, movement, and distance of objects captured by the camera. And that makes new gameplay experiences possible.

Imágenes obtenidas Ifixit – Nitendo Switch teardown <https://es.ifixit.com/Desmontaje/Nintendo+Switch+Teardown/78263? > y switchbrew.org <https://switchbrew.org/wiki/Hardware>.

Arquitectura de la *Switch (y V)*

- Complementos
  - Nintendo Labo < https://labo.nintendo.com/kits/variety-kit/>
    - Toy-Con 01 Variety kit



Buy now $69.99MSRP*

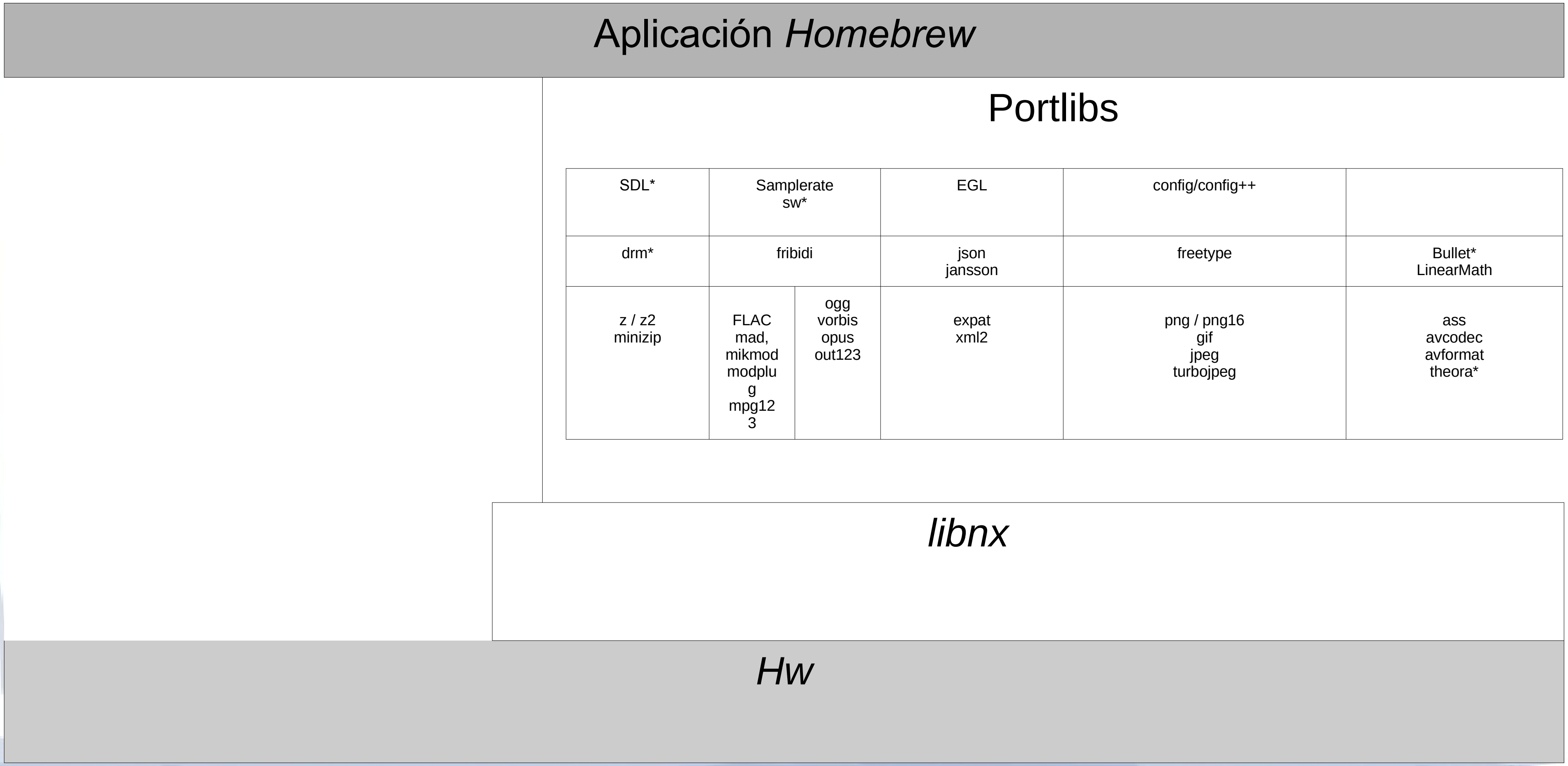
Check availability at select online retailers below.

Includes five different

Imágenes obtenidas Ifixit – Nitendo Switch teardown <https://es.ifixit.com/Desmontaje/Nintendo+Switch+Teardown/78263? > y switchbrew.org <https://switchbrew.org/wiki/Hardware>.

Arquitectura software: devkitA64 + libnx

- Para *switch*: librerías de sistema y *portlibs*



SDL*	Samplerate sw*		EGL	config/config++	
drm*	fribidi		json jansson	freetype	Bullet* LinearMath
z / z2 minizip	FLAC mad, mikmod modplu g mpg12 3	ogg vorbis opus out123	expat xml2	png / png16 gif jpeg turbojpeg	ass avcodec avformat theora*

En prácticas

- NDS → How to make a bouncing ball game
- 3DS
  - Escribir en consola
  - Acceso a botonera y pantalla táctil
  - Acceso a fecha y hora
  - *Personal data structure*
  - Sistema de archivo: romfs vs fat

Bibliografía

- DevkitPro
  - devkitARM/devkitA64, libnds, libctru, libnx
  - <https://devkitpro.org/>, <https://github.com/devkitPro>
- Wikipedia - Nintendo DS homebrew
  - <https://en.wikipedia.org/wiki/Homebrew_(video_games)>
- Foros de desarrollo
  - “Scene”
    - NDS / 3DS <https://nds.scenebeta.com/>
    - Switch <https://switchscene.org/ >
  - *Drunken Coders <http://drunkencoders.com>, <>https://github.com/drunken-coders>*
  - jdriselvato/NDS-Development *<https://github.com/jdriselvato/NDS-Development >*
  - 3DBrew <http://3dbrew.org/wiki>
  - Smealum. Breaking the 3DS security system

    <https://smealum.github.io/3ds/32c3/#/91>

Bibliografía (II)

- Jaeden Amero (Patater). 2010. Introduction to Nintendo DS Programming
  - <http://www.patater.com>
- 3DGuy. 3DS Development Hardware
- Neimod y Martin Korth. 2013. DSTek: Nintendo DS Technical Information
- GbaTek. Technical information from no$gba
- Märten Tonissoo. 2010. Nintendo DS game console.
  - <http://www.martentonissoo.com/documents/Nintendo_DS_game_console.pdf>
- ThomasWorld. Baby Steps In Nintendo DS Homebrew Hacking

Bibliografía (III)

- F. García Bernal. 2008. Desarrollo de Videojuego 3D Para La Videoconsola Nintendo DS.
  - Dpto. Lenguajes y Ciencias de la Computación. ETS ING.INF. Universidad de Málaga.
- Puyover. 2010. Programación en Libnds y ensamblador ARM. Una introducción a la programación en NDS
  - <http://www.martentonissoo.com/documents/Nintendo_DS_game_console.pdf>
- F. Sivianes, J. Barros y A. Martín. 2009. Entorno Desarrollo para NDS.
  - Periféricos e Interfaces. 3º Ingeniería Técnica en Informática, especialidad en Sistemas Físicos
  - <http://www.dte.us.es/tec_inf/itis/peri_int/EvolucionInicio/Trabajo_NDS.pdf>

Bibliografía (y IV)

- M. Banahan, D. Brady y M. Doran. 1991.The C Book. Addison Wesley.
  - En línea <http://publications.gbdirect.co.uk/c_book/chapter8/declarations_and_definitions.html>
- B, W. Kernighan y R. Pike. 1983. The Unix Programming Environment. Prentice-Hall Software Series.

  by Brian W. Kernighan (Author) , Rob Pike
- B. W. Kernighan y D. M. Ritchie. 1998. C Programming Language (2nd Edition)