

Nintendo

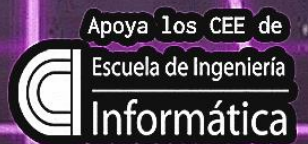
GAME BOY®

**¡Aprenda homebrew y haga su juego 100% retro!**

**Martes 3 Octubre (Parte 1) - 12:15 hrs**  
**Miercoles 4 Octubre (Parte 2) - 14:00 hrs**  
**Escuela Ingeniería Informática PUCV (Piso 2)**  
**PUCV Edificio Isabel Brown Caces**  
**Av. Brasil 2241, Valparaíso**



**Videojuegos retro en**  
**GAME BOY**  
**... de verdad!**



**Videojuegos retro en  
Gameboy ... de verdad!**

**(Parte 1)**

**Enzo Barbaguelatta D.**



# Hola!

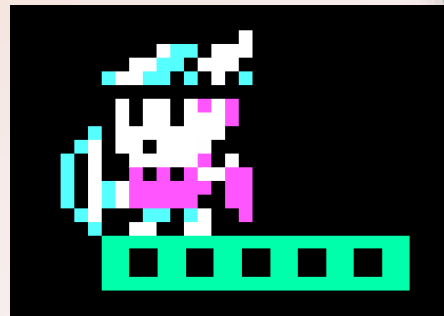
- Primero, gracias por venir! :D
- Ojala estén disfrutando la semana de escuela.
- Ojala disfruten esta charla/minicworkshop/queseyo.
- Mi primera vez presentando algo por gusto propio... Si algo no sale como debe, les pido que sean pacientes 😊
- Asi también, el feedback es importante. Si crees que puedo mejorar algo, o esté cometiendo algún error, dimelo!
- Toda esta presentación estará online, asi que no es necesario anotar mas que los links!

La pregunta:

**¿Que experiencia han tenido con las consolas de la vieja escuela?**

# Yo

- **Enzo Barbaguelatta.** (En internet, **EN.I**)
- Ingeniero Civil Informático PUCV (2010-2016)
- Bugueo cosas, gusto por los videojuegos old-school, la imaginación, las pantallas de error extrañas y tecnologías de los años 90.
- Mi primera consola: **Nintendo 64**
- Gusto en consolas: Herramientas, hack-roms, homebrew, Zophar Domain
- BitbitJAM, Hilda (SuperNintendo)



- Mi primer juego de Game Boy: Uno temático como competencia de “saludo original de cumpleaños” para el aniversario de cierta tiendita de juegos...
- No me gane la Nintendo Switch, pero saqué tercer lugar y feliz con un libro de la Nintendo64 😊



<https://twitter.com/elsemieni/status/841064462341799936>

# ¿Que haremos hoy?

- No podemos ir a programar de lleno, si no sabemos el transfondo de la Game Boy con **el mundo homebrew** y sus particularidades!
- Esto se dividirá en 2 partes. La primera donde introduciré el concepto del **homebrew** y instalaremos las herramientas, y en la segunda cocinaremos un juego. (No se esperen gran cosa, son solo 2 días!).
- Saber **C** y imaginación bastará. (Si estas cursando Fundamentos de programación también puedes intentar hacer algo... si no sabes, puedes quedarte a esta primera parte que es cultura general 😊 ).
- (Algo de arquitectura de hardware puede ayudar también ;))

**¿Empecemos?**

# Videojuegos

- Si no sabes que son los videojuegos, te pediría amablemente que abandones la sala... (broma!)
- Todos hemos jugado alguna vez alguno. Practicamente es ya parte de nuestro diario vivir.



# Vamos específicamente a los años 80-90





- Nintendo hizo furor con su consola, la **Nintendo Entertainment System** (o cariñosamente mas conocida como la NES).



- Se venían los años 90, y la pelea por las consolas y computadoras multimedia se hacía **muy, pero muy intensa!**

- Hasta ese entonces, era un sueño muy remoto poder jugar esos videojuegos geniales en la palma de tu mano, sin tener que enchufar la consola de turno a la tele (previo permiso a tus papás) y poner el canal 2-3 para jugar algo. Olvídate de los smartphones, de la Nintendo 3DS... esas cosas no existían. Las diversiones eran otras.



- Salió el SuperNintendo, el Sega Genesis (o Megadrive, como quieran llamarla), y un sinfín de consolas raras de dudoso éxito que querían subirse cuanto antes al tren de lo cool y “*bacán*” (léase con acento de termino pasado de moda) del CD-ROM y los gráficos en 3D.







Hasta que llegó ese ladrillito feo de color gris

(un caaaaro ladrillito)

<https://www.youtube.com/watch?v=Di68PCqOIXE>



- La GameBoy, pequeña consola portátil monocromática a pilas que permitía jugar juegos de alta calidad (para ese entonces), como el **Tetris** (el juego mas vendido para la consola) ,o el **SuperMario Land**. (Pokémon salió después...)
  - A lo largo de su historia, se vendieron cerca de **120 millones** de GameBoy!!! Era el sueño de muchos, y presentes en muchas listas del viejo pascuero (cuando en la triste realidad, te llegaban calcetines).
- Si tenias la suerte de tener tu una GB real (no, usar emuladores no te hacían cool) en la escuela/colegio, te volvías instantáneamente la persona mas popular de la clase.



- Un majestuoso procesador tipo Z80 (el mismo que usaba el arcade del Pac-Man! ) a unos majestuosos 4.19 MHz (hey, eran los 90!)
- 16 kb de RAM en total! Mas que suficiente! ;D (8 para los programas, otros 8 para video)
- Pantallita de 2.6 pulgadas de matriz de LEDs (de verdad, no de esas TV de ahora) a 160x144 (144p)! Podía mostrar 4 tonos del verde claro al negro.
- 4 canales de sonido en completo estéreo, con posibilidad de conectar tus audífonos preferidos.
- Requería 6V DC para funcionar (con 4 pilas AA la haces), consumía 0.4 watt, mas que suficiente para unas ratonas 4 horas xD
- Cuando salio costaba 89 dólares (ahora unas 169 dólares... sus 100 lukitas... si le agregas los costos de importación mas demases cosas, sus 200 luquitas).
- Era barata aunque no crean... La gameGear de sega costaba 150 USD -> 257 USD -> 160.000 -> 320 lukas

**LOWER PRICES, BIGGER SELECTION...GUARANTEED!**



984701



984728



984663



984698



984671

**Game Boy Pocket**  
Improved clarity, and now 30% smaller and slimmer...to fit right in your pocket! Batteries included.

**59<sup>99</sup>**

**Play It Loud Game Boy**

Nintendo power in hot colors! Batteries included. Game cartridges sold separately.

**Nintendo**

**YOUR CHOICE**

**49<sup>99</sup>**

**GAME BOY**

**Nintendo Game Boy Carrying Case 9.99**

97612



**19<sup>99</sup>**

576743



**34<sup>99</sup>**

576778



**19<sup>99</sup>**

777080



**29<sup>99</sup>**

572713



**29<sup>99</sup>**

581143



**19<sup>99</sup>**

579947



**19<sup>99</sup>**

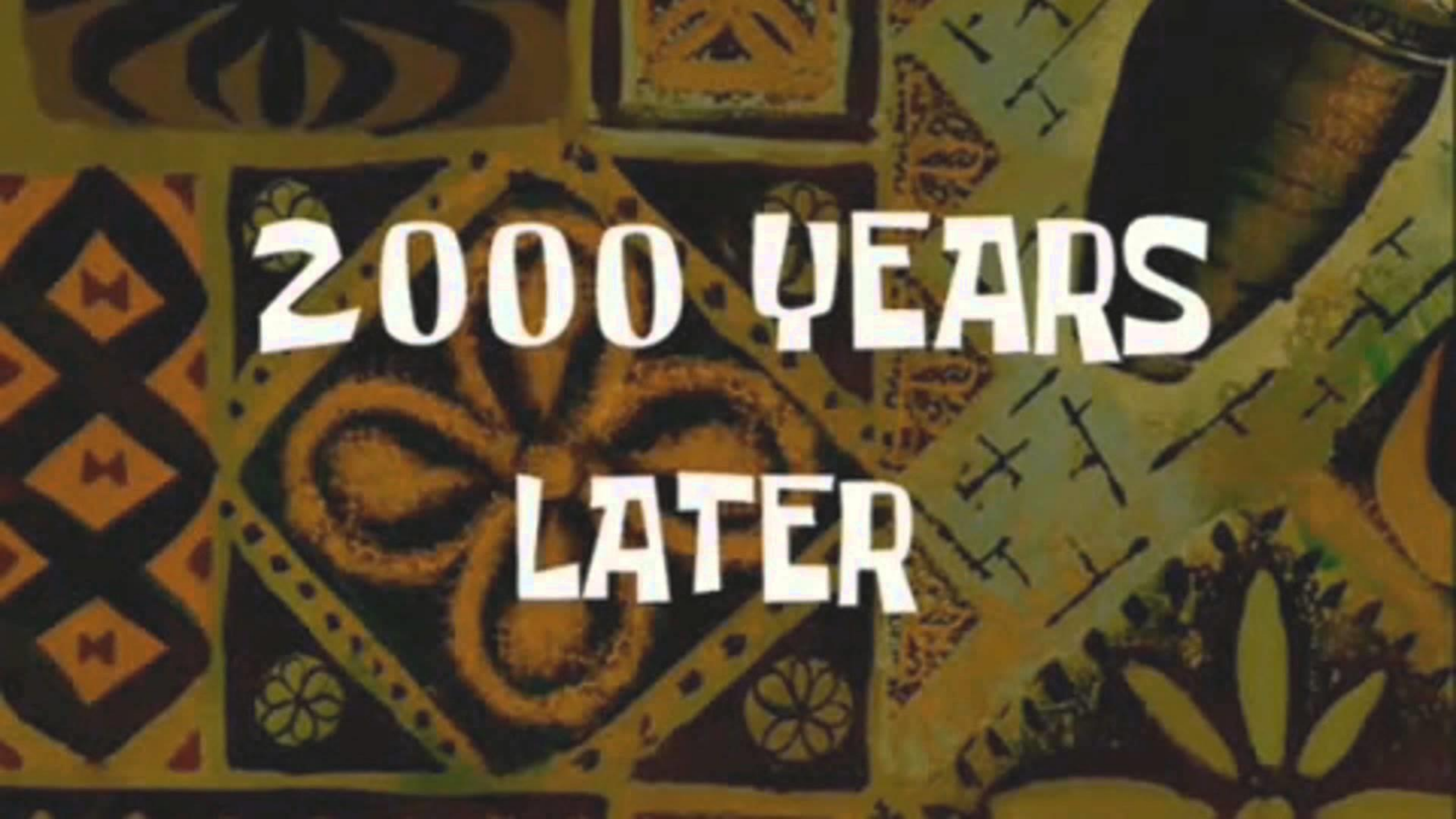
876410



**19<sup>99</sup>**

974269



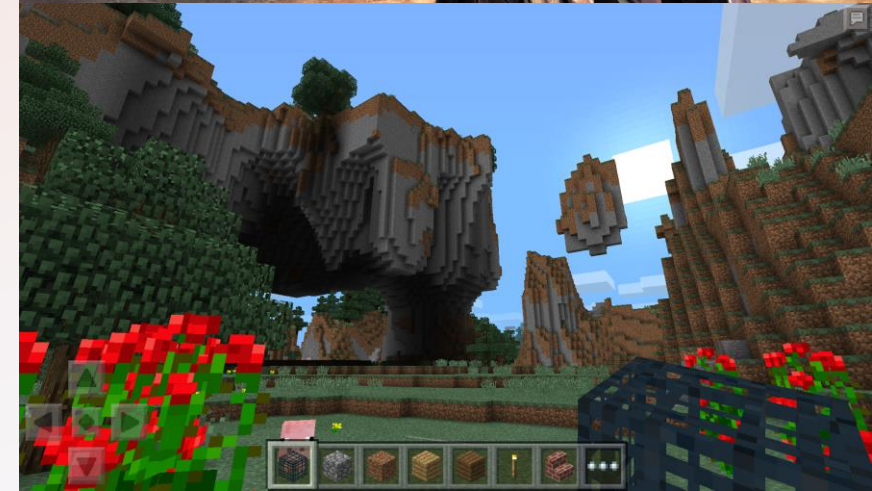


**2000 YEARS  
LATER**



# Estamos en un mundo totalmente cambiado desde los años 90:

- Una industria muy solida para el mercado de los videojuegos.
- Consolas y computadoras de ultra requete archicontra performance para jugar.
- Títulos AAA repletos de gráficos 16k, niños rata y DLC's
- Y la U sin estadio, el santuario de Teresa de los Andes inconcluso, y aun ni anuncian Half-Life 3



Lo portátil también cambió bastante...

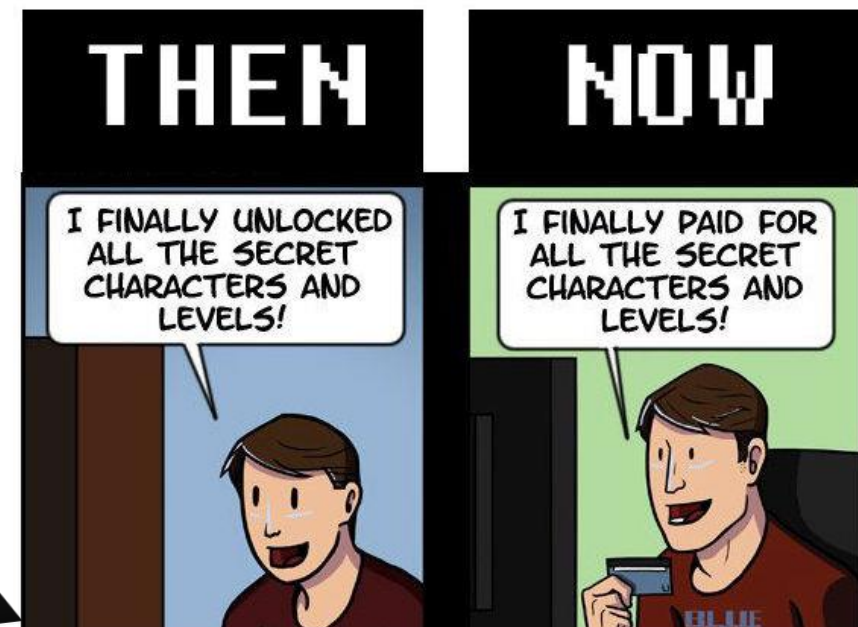
De la Gameboy nos saltamos a consolas touch de tamaño bastante portátil de variadas marcas, con videojuegos que nada tienen que envidiar a los de las consolas grandes (bueno... casi).

Y hasta los celulares se subieron al tren: Ahora se usan para casi cualquier cosa (menos hacer llamadas), y eso ha permitido que naciera un poderoso mercado para un montón de tipos de videojuegos modernos.

... Si, **todo un mercado.**



# — GAMING —





\$ Un ojo a la cara.00

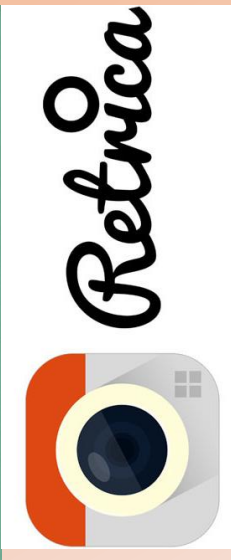
Pero... ¿porqué cresta seguimos jugando cosas “tipo” retro?

**QUE CRESTA**

**PASO AQUÍ!?**

Quieranlo o no. Esto ocurre porque...

# *Retro/vintage/como Quieran llamar a la#\$%& está de maldita Moda!*



... Aunque bien al peo.

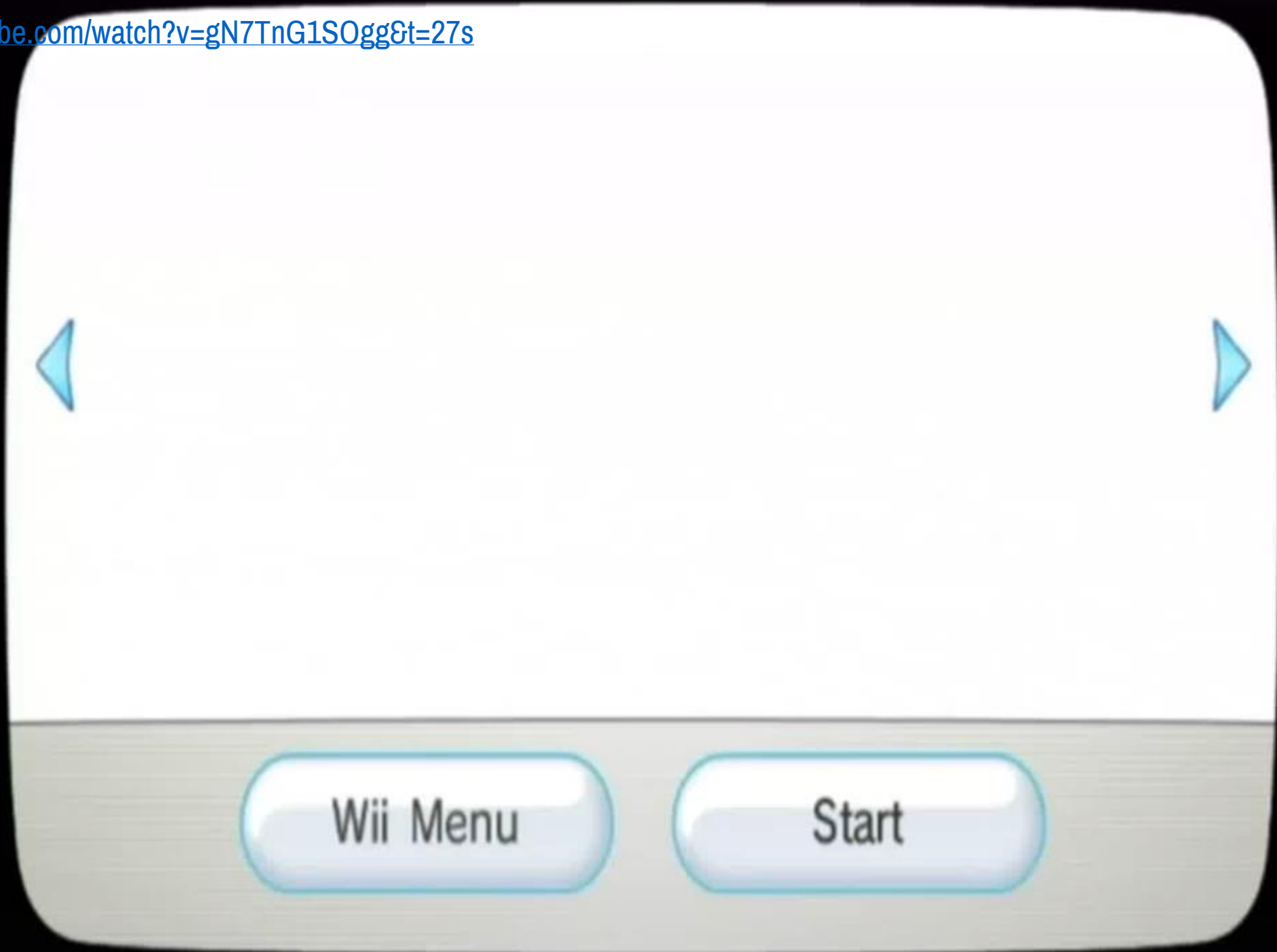
# Y como todo elemento... esto también afecta al mundo de los videojuegos.



- Mucha gente le encanta lo retro... solo porque es retro! Y esta a la moda! Eso ya es suficientemente cool.
- De hecho, si a algo le pones que es 8 bits (sin saber que cresta significa 8 bits), con música de pitidos surtidos sin armonía y dibujitos feos de Megaman, ya puedes ser ultra famoso sin mayor esfuerzo (bares temáticos de videojuegos por allí, ex novias de youtubers en decaimiento, y un sinfín de etc.)



Pero como siempre detrás de todas las modas y artes, hay gente escondida en el **underground** que va mas allá: emergentes, políticamente incorrectos, super incomprendidos y quizá desconocidos, pero a los que le debemos muchísimo... **Si, hasta en la informática y los videojuegos este movimiento existe!** Solo miren! 😊



# La escena homebrew

- Si lo traducimos directamente al español tenemos “*fermentación casera*” (wut?)
- El origen del termino, como sugiere el nombre provino del mundillo de la elaboración de las **cervezas caseras** (y se sigue utilizando).
- No, no se entusiasmen. Esta no es una charla de cervezas caseras (y tampoco hay degustación).
- Es un pseudo-movimiento interesante, el cual consiste básicamente en mirar mas allá de lo que ofrece la industria y probar algo hecho con tus propias manos.
- Y como se pueden imaginar, aquí no estamos hablando solo de cervezas.





En el mundillo computacional...

...muchos años...



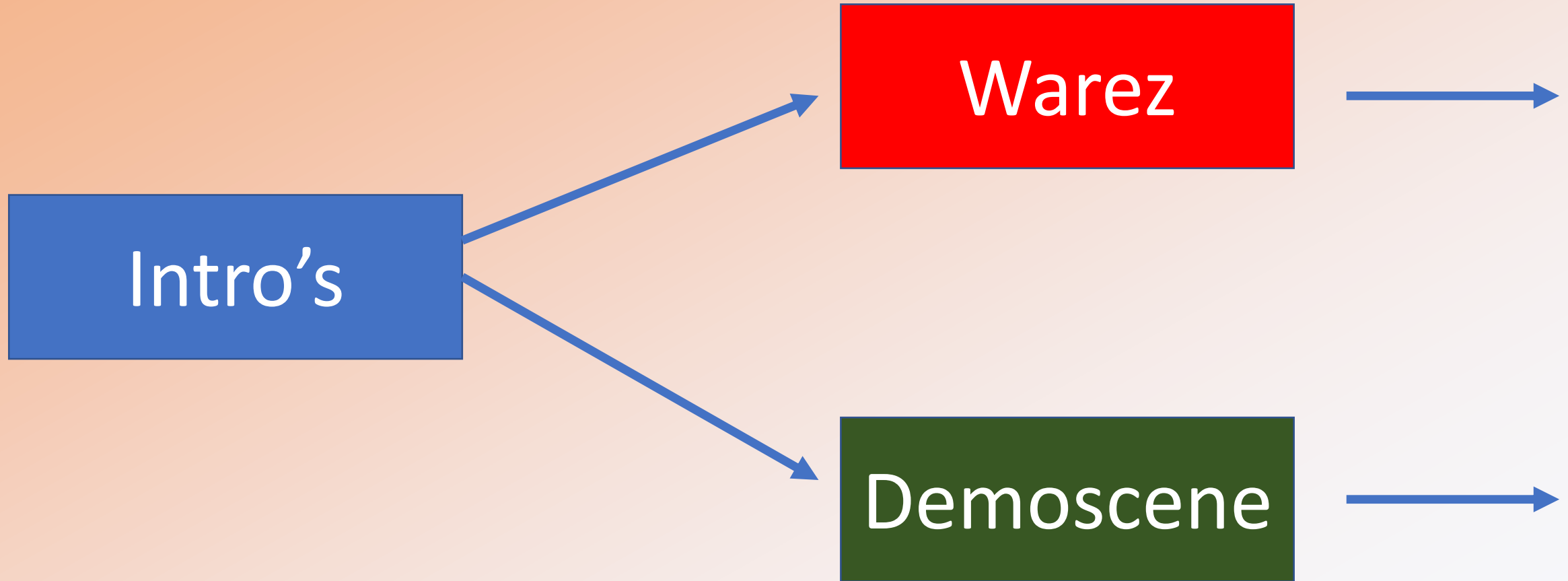
```
***** COMMODORE 64 BASIC V2 *****  
64K RAM SYSTEM 38911 BASIC BYTES FREE  
READY.  
LOAD"IKARI-02.PRG",8,1:  
SEARCHING FOR IKARI-02.PRG  
LOADING  
READY.  
RUN
```

# El principio del mal: **Las INTRO's**

(Mas conocidos como *la musiquita de los cracks*)

# Desde allí, esto tomó 2 caminos.

(Uno con futuro, otro... no tanto...)



<https://www.youtube.com/watch?v=YiYhnCVbmKo>

El oscuro mundo del **Warez** ... en casi cualquier consola

(Hay que reconocerlo... algunas musicas eran pegajosas.)

<https://www.youtube.com/watch?v=I8gUyMLyAMU>

(Pero a opinión propia, las ahora conocidas como “*medicinas*” perdieron la gracia ya...)

(Psssst, medicina gratis pal nero 6, +10 y a favoritos xDDD)

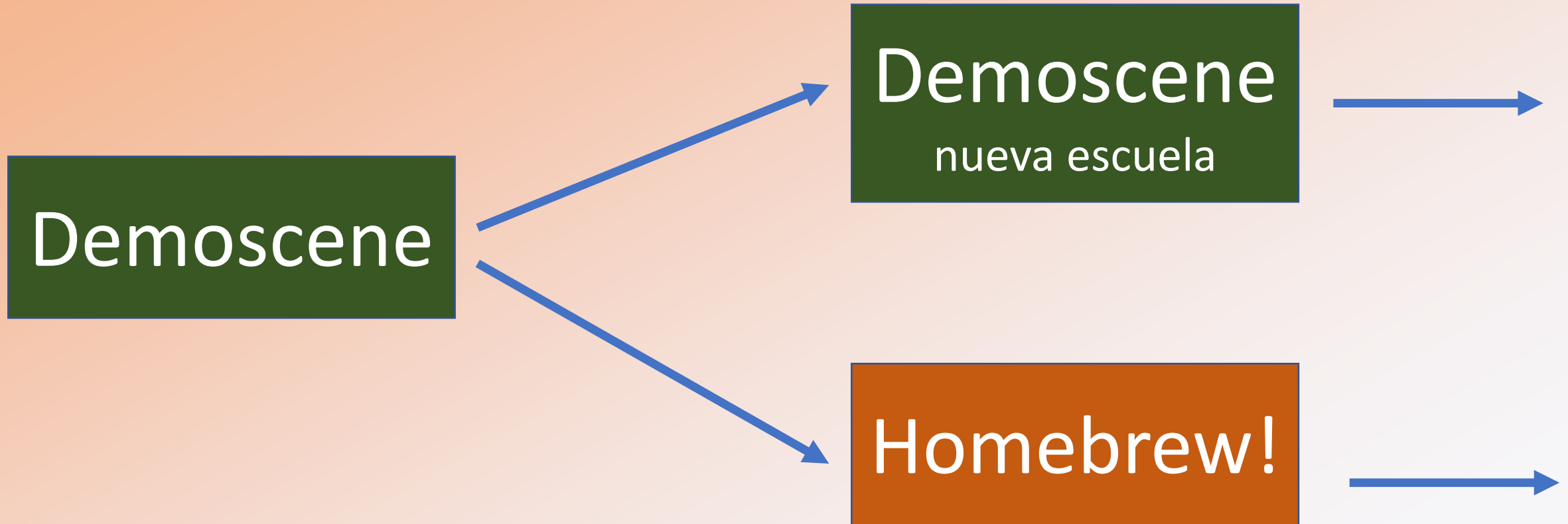
<https://www.youtube.com/watch?v=89wq5EoXy-0>

# El otro mundo es el Demoscene

(Gente que le gustaba hacer intro's, pero no por cracks sino por gusto y hacer explotar las consolas)

# Desde allí, esto tomó otros 2 caminos.

(Uno con futuro, otro... igual tiene futuro...)





# La Demoscene sigue en lo mismo

(Pero con mas cerveza, mas GTX 1080, y mas capacidad para hacer explotar mentes)

<https://www.youtube.com/watch?v=vUMVtJmC01c>





Tastatur und Maus

COMING UP NOW  
OPENING CEREMONY



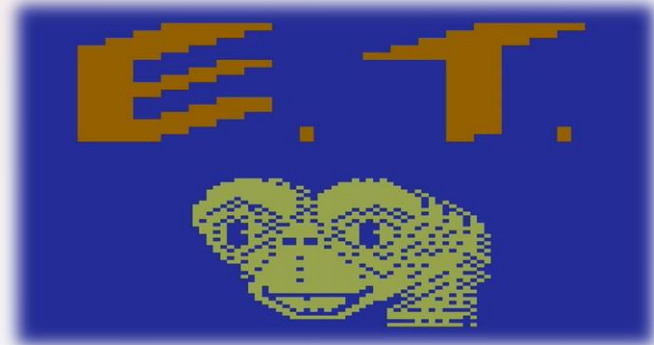
Y tenemos  
al fin el  
Homebrew!

<https://www.youtube.com/watch?v=59KW8xT74eo>

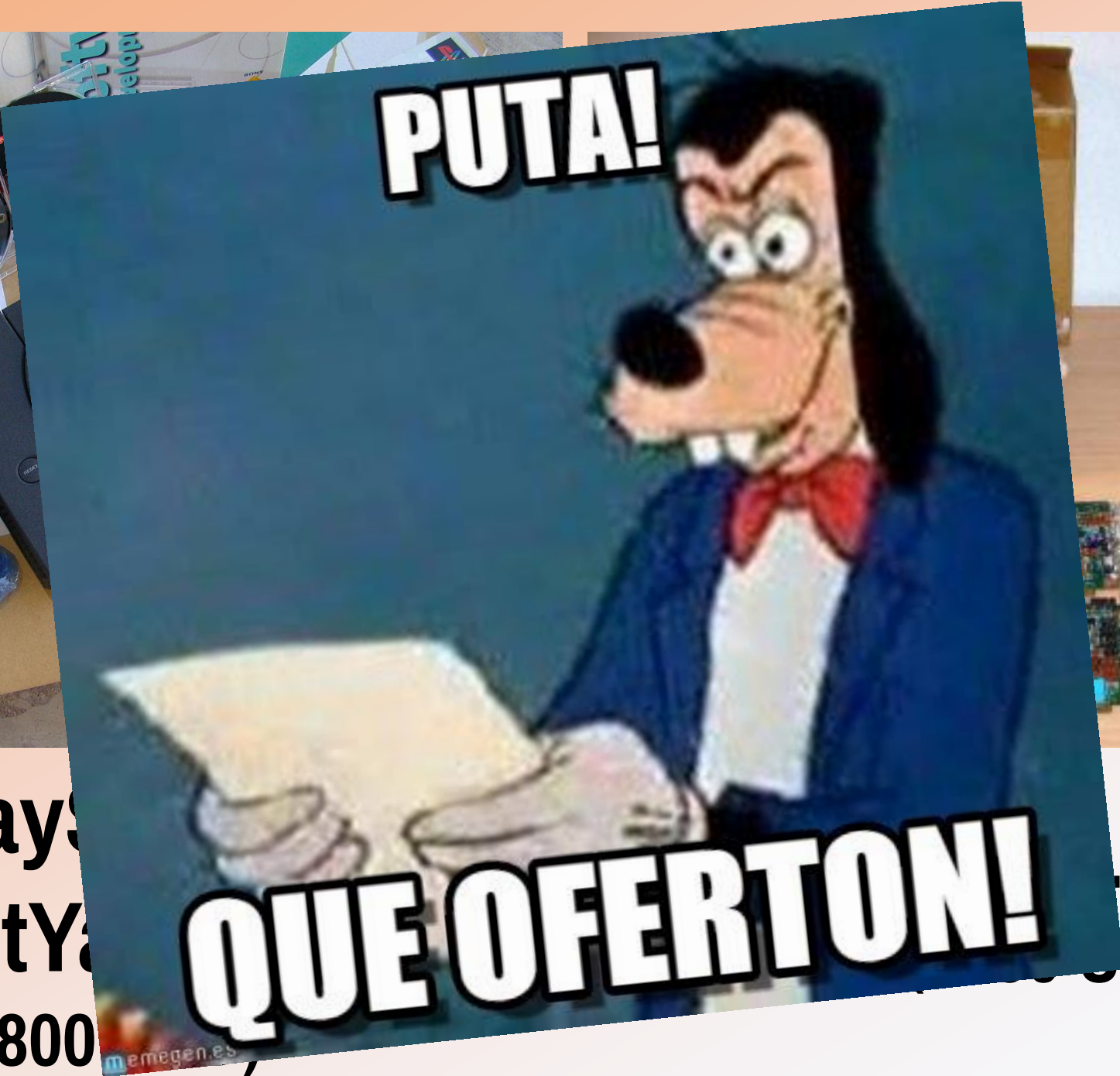
# Muy básicamente: Homebrew es **hacer software no licenciado de forma casera para una consola!**

Espera... ¿como eso así de no licenciado?

- Cuando tu publicas un juego para una consola, primero debes contactarte con un Publisher, o con la compañía de la consola misma.
- Si eres un don nadie lo mas probable es que se te rían en la cara.
- Si pasas el primer filtro, comprar un kit de desarrollo (carisimos).
- Desarrollar un juego en una tecnología bastante bizarra y no morir en en el intento (o irte en bancarrota).
- Mandar todo eso al Publisher y a la compañía y esperar que te den el visto bueno (similar a las App stores, pero mucho mas tradicional y \$%&/()# para aprobarlos... nadie se salva, ni siquiera las grandes).
- Y finalmente mandar a hacer copias en masa, y cruzar los dedos para que tu jueguito no se transforme en un E.T.



- No todo estaba perdido. Erase los años 90 y aparecían los primeros computadores multimedia con Windows 95 (y Macintosh se hacían bastante populares también).
- ...Y con ello la gente con ganas de modificar cosas e ir más allá.
- Aparecían kits truchos de desarrollo, generalmente por chinos, rusos y otras nacionalidades de dudosa procedencia.
- El mercado de la gameboy era muy prometedor. Aparecían kits de desarrollo caseros “alternativos” (RGBDS, GBDK, entre otros ). Te ahorrabas el kit, pero igual tenias que hacer todo el proceso de publicación.
- Sony también apostó por ese mundo, lanzando una rareza llamada NetYaroze. Básicamente una consola “a bajo costo” para permitir jugar y hacer juegos (pero muy limitados). **Los mejores eran publicados en las revistas de PlayStation** como “juegos de regalo” que probablemente compraste alguna vez en tu kiosco y probaste como “¿y este juego que onda?”.



PlayS  
NetYa  
(800

-H2010  
(USD)



Este juego de NetYaroze venia de regalo en una revista de play...s, había uno de estos "juegos de regalo" llamado "Manta". Se hizo super famoso.

# Y el homebrew estalló. La industria le gustaba...

(Por lo tanto evolucionó)

Homebrew!



*(Pero eso es otra historia, pregúntenles  
A los del taller de Unity ... )*

Escena  
indie

**FINAL FELIZ?**



**NO**

**O**



- Primero, si termino mi presentación acá, me tirarán vivo a la hoguera. Vinieron acá para aprender de Gameboy, no para ver memes surtidos unos cuantos años añejos xD
- Segundo, la industria indie, si bien permite a la gente emprendedora hacer sus juegos, deja COMPLETAMENTE olvidado las consolas antiguas.
- ¿Recuerdan que lo retro estaba de moda? Si sisisi... puedes hacer un juego en Unity estilo 8 bits, pero no es la idea (si no entiendes porque, tendremos que volver un par de PPT's atrás).
- Entonces el homebrew sigue vivo en 2 formatos! Como en consolas sin usar los kits autorizados para desarrollar juegos (ahem... Luma 3DS? *Homebrew Channel?*) y en las consolas abandonadas...  
**como la GAMEBOY!**



[www.youtube.com/watch?v=milyBS3pTmE](http://www.youtube.com/watch?v=milyBS3pTmE)

¿Y que ha habido de bueno en homebrew para la Gameboy?



¿Y que ha habido de bueno en homebrew para la Gameboy?



<https://www.youtube.com/watch?v=iXLZ0Ns6s04>

¿Y que ha habido de bueno en homebrew para la Gameboy?

Yayayaya muy bonito y  
todo eso, pero yo quiero  
hacer juegos de gameboy  
por la cresta!

# Muy bien, entonces hagamos homebrew para gameboy...

Necesitaremos:

- Una Gameboy (funcionando)
- Pilas pa la gameboy
- Una flashcart pa meterle juegos
- Un kit de desarrollo para Gameboy
- Mas pilas pa la gameboy (escencial)
- Un partner y una licencia de Nintendo
- Mucho dinero
- Un computador multimedia con Windows 95
- Una maquina del tiempo
- Un par de bálticas
- Ah, y por supuesto, mas pilas pa la gameboy.



# No, espera... no necesitamos todo eso

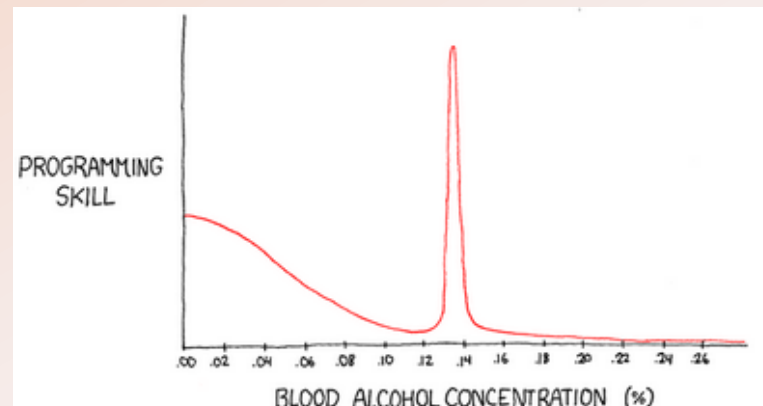
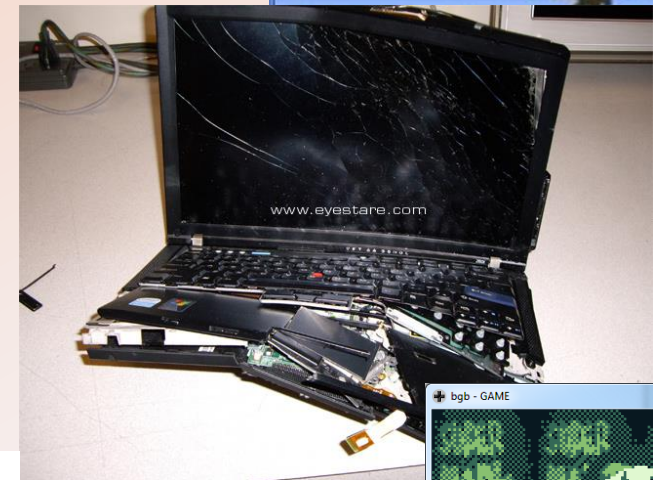
Podemos hacer juegos por mucho menos cosas.

- Podemos usar perfectamente emuladores, así que ahí nos ahorramos la gameboy, las pilas, y la flashcart. (Ahora si tienes todo eso, pulento! :D )
- Estamos haciendo **homebrew!** Somos chic@s rebeldes punkies! No necesitamos al imperi... digo la licencia de Nintendo (aparte que hace 17 años que ya no dan licencias para gameboy)
- Usaremos kits de desarrollo bien mantenidos de la comunidad, así que no tienes que contactar a Rafael Garay para ello, ni preocuparnos por AESTHETICS para Windows95 ni la maquina del tiempo.
- Por favor, dejen la báltica pa la loco cabros...



# Por ende, ahora necesitaremos solamente lo siguiente...


- Un emulador de gameboy (BGB - VisualBoyAdvance)
- Computador con Windows (con un 7 tamos dados).
- Kit de desarrollo de gameboy alternativo (GBDK-ZGB)
- Conocer lenguaje C
- Algún editor de C (Notepad++, SublimeText, VisualStudio, blablabla).
- Muchas ganas (cerveza es opcional)
- Pilas pa la gameboy



# GBDK-ZGB ... kisaw3@?

**GBDK** (*Game Boy Development Kit*) Es un kit de desarrollo bastante comprensible para Gameboy, compuesto de librerías para hacer cosas con la gameboy, ensambladores (Assembler), un compilador de C, y un linkeador para generar ROMs :D Es usualmente mantenido, y funciona “afortunadamente” en computadores modernos (da gracias xD)

**ZGB** es un framework BBSD (bueno bonito simpático y diminuto) hecho hace un par de años por un desarrollador español llamado *Zal0*. Con este kit se pueden crear juegos relativamente fácil, sin lidiar taaaaaanto con la parte oscura de hacer juegos de consola. Viene con varias herramientas que nos hará la vida más fácil!



(Zal0 sabe de la existencia de esta charla y les manda saludos! :D)



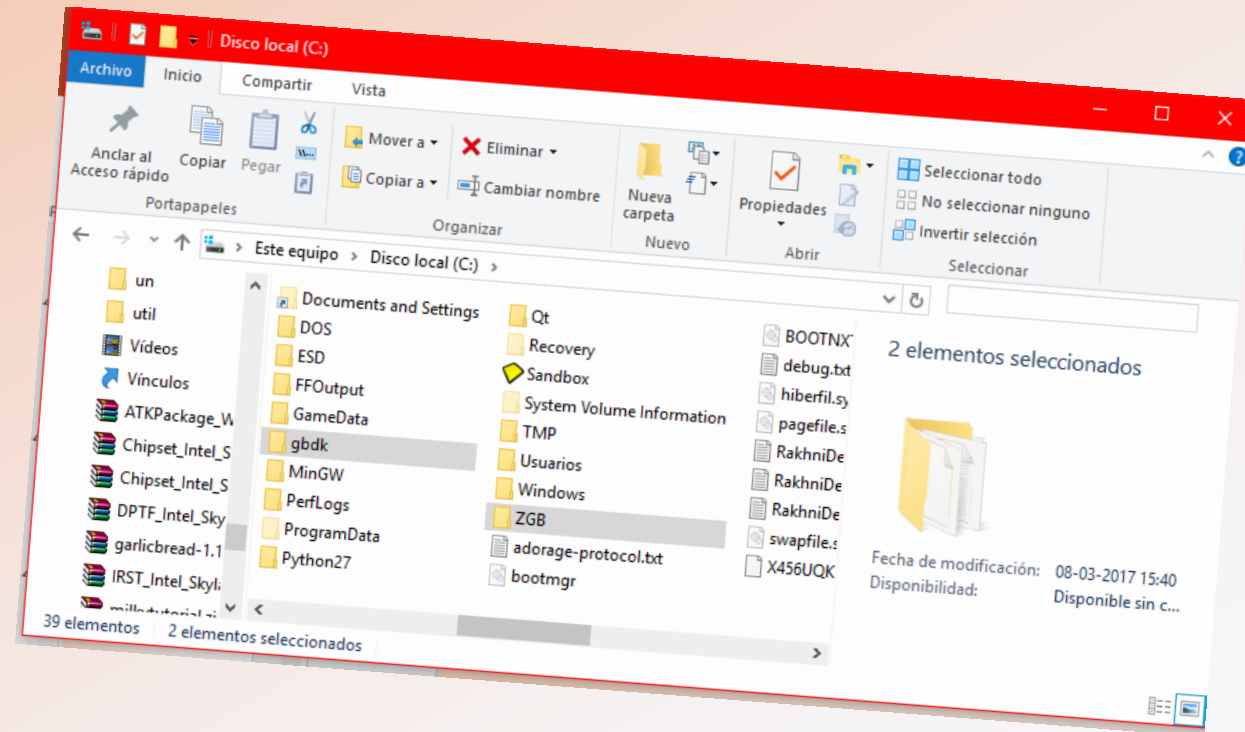


# ¿Que cresta necesito descargar entonces?

**Un montón de programas (SDCC, GBDK, ZGB, MSYS, Make, GBMB, GBTD, BGB, OpenMPT, GBT Player, etc, etc, etc ... ), pero ya que estamos en la PUCV-PRO y el ancho de banda no es muy bueno, solo baja este zip que ya viene todo incluido:**

**[http://elsemieni.net/gb/gb\\_devkit.zip](http://elsemieni.net/gb/gb_devkit.zip)**

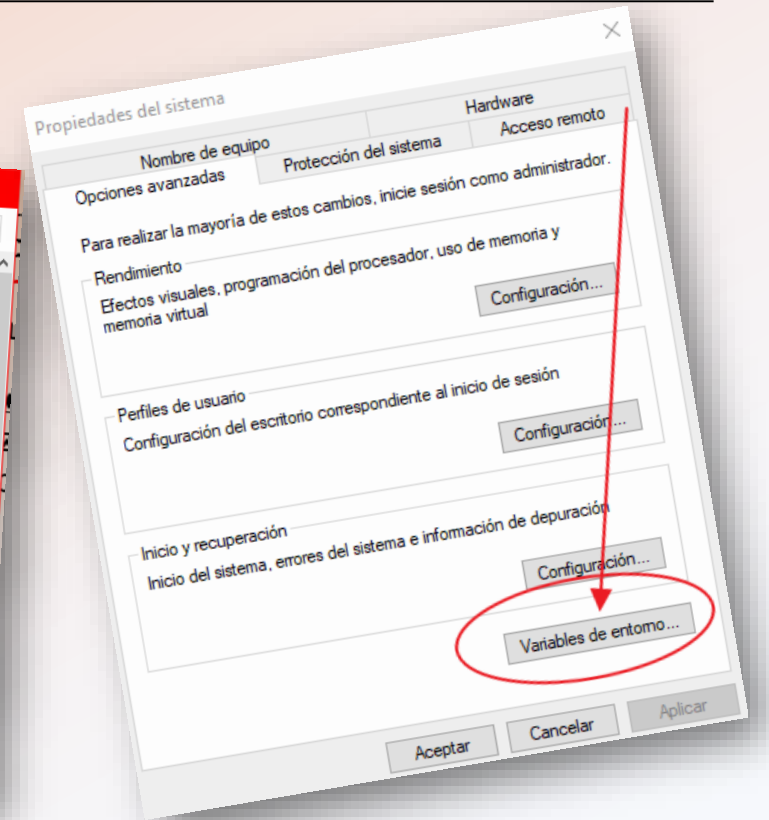
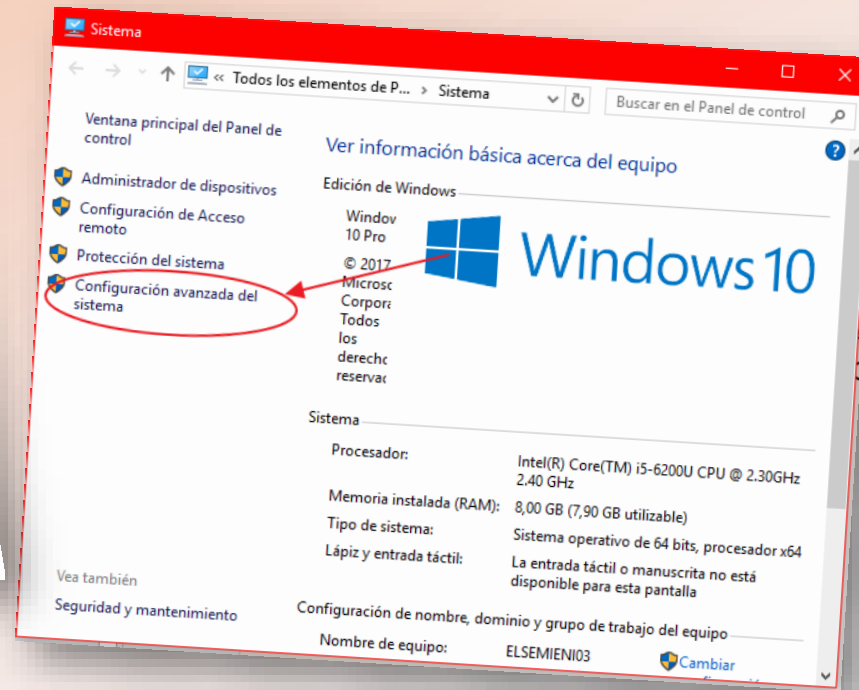
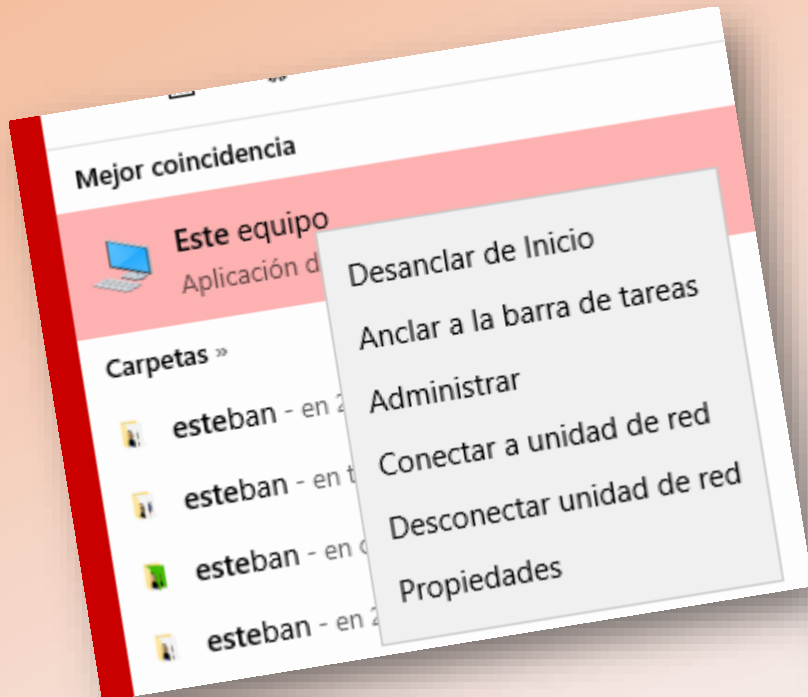
- Que me estas mirando fe@? ¡Descomprímelo por ahí! (Recuerda tener WinRAR/7zip). Tendremos una salpicadera de archivos y carpetas varias.
- Instala el **SDCC**! (compilador para chips raros y aparatos bizarros). (Al terminar activa la opción de añadir SDCC al path).
- Ahora pesca las carpetas **GBDK** y **ZGB**, y déjalas en tu disco duro **C:/**

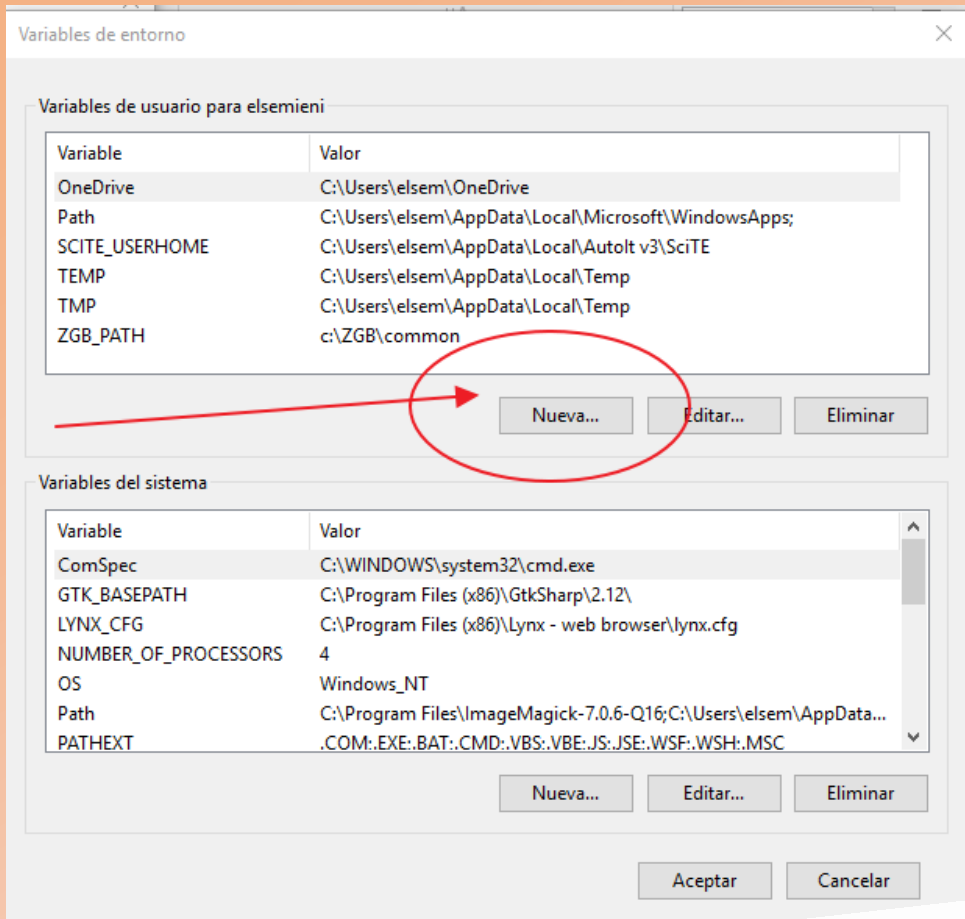


# Y ahora?

NOTA: Si te pide permisos de administrador/privilegios/vender tu alma al demonio, di que sí!

- Ahora, para que los compiladores sepan en que carpeta están las cosas, agregaremos algunas variables de entorno.
- ¿Que cresta es una **variable de entorno**? Frases “comodines” del sistema operativo que si las escribes se cambian por valores. (Por ejemplo si escribes **%windir%** inmediatamente cambia a **C:/Windows**)
- Para ello anda a **Propiedades del sistema** (Mi PC/Equipo/Este Equipo, click derecho y Propiedades, o Panel de Control y Sistema) Y a la izquierda click en **configuración avanzada del sistema**. Una vez allí, abajo haces click en **variables de entorno**.





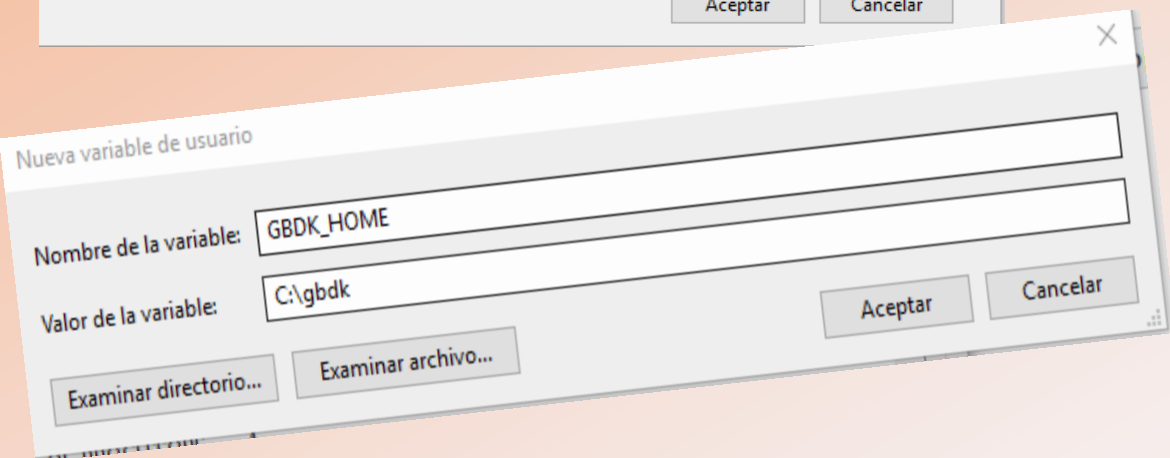
Ahora que estamos aquí,  
agreguemos 2 variables de entorno  
**de usuario** (no de sistema, de usuario)

Para eso vas a la sección Variables de usuario  
para XXXXXX (tu user) y haces click en **nueva**.

Agrega estas 2 variables de usuario (nombre y  
valor):

**GBDK\_HOME**  
**C:\gbdk**

**ZGB\_PATH**  
**C:\ZGB\common**





# Pero no te vayas todaviaaaaa!

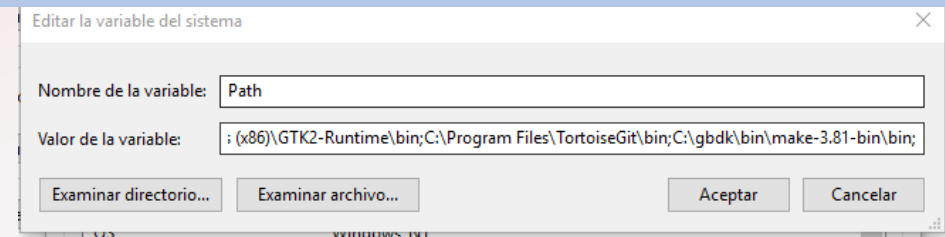
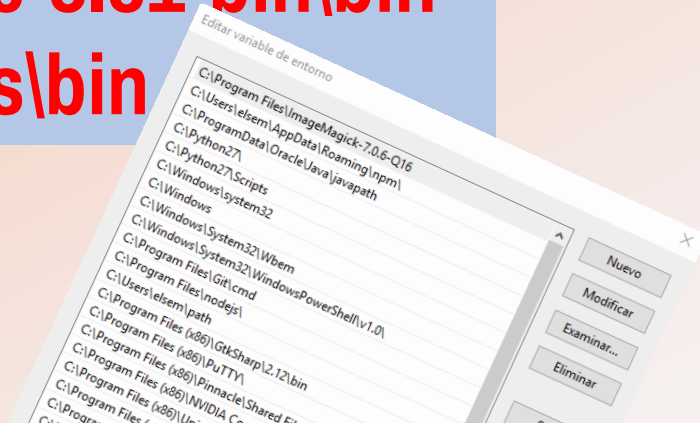
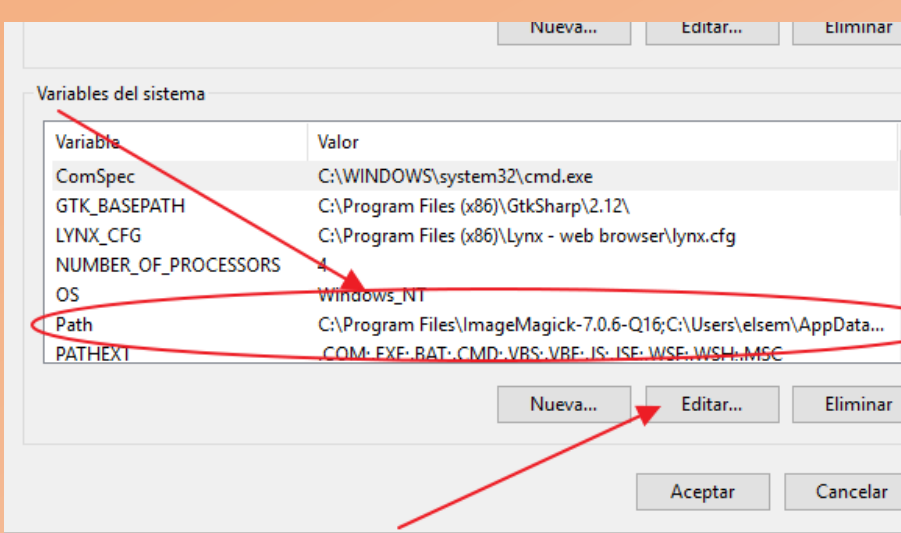
Necesitamos modificar la variable de entorno del sistema **PATH**. (Las variables del sistema están debajo de las de usuario) Así que búscala y pon Editar.

Si te aparece una lista, simplemente agrega estas entradas nuevas que digan cada una:

**C:\gbdk\bin\make-3.81-bin\bin**  
**C:\gbdk\bin\msys\bin**

Si te aparece para ingresar nombre-valor como los anteriores, agrega al final estas nuevas entradas, separados por ; (punto-coma).

**C:\gbdk\bin\make-3.81-bin\bin**  
**C:\gbdk\bin\msys\bin**



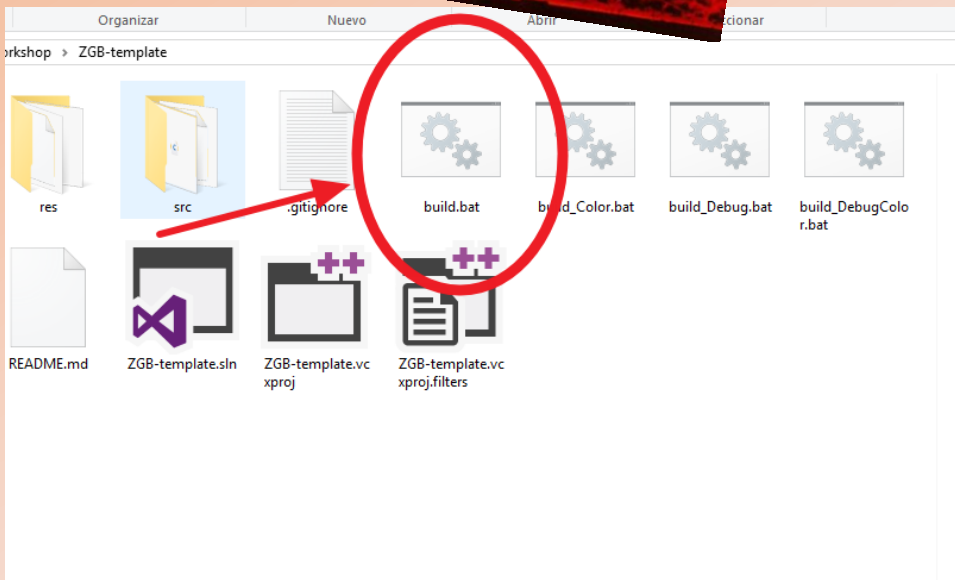
لا تجميع هذه الملفات!

# Llego la hora de probar



Vamos a ver si lo que acabamos de hacer cuaja o explota de forma hollywoodense en nuestras caras. Compilemos una ROM de Gameboy básica.

- Copia la carpeta **zgb-template** a cualquier lugar.
- Entra a la carpeta.
- Abre el archivo **build.bat**
- Disfruta el espectáculo 😊 . **Se abrirá un emulador (BGB) con tu flamante juego, o bien presenciaremos una explosión de dimensiones épicas.**





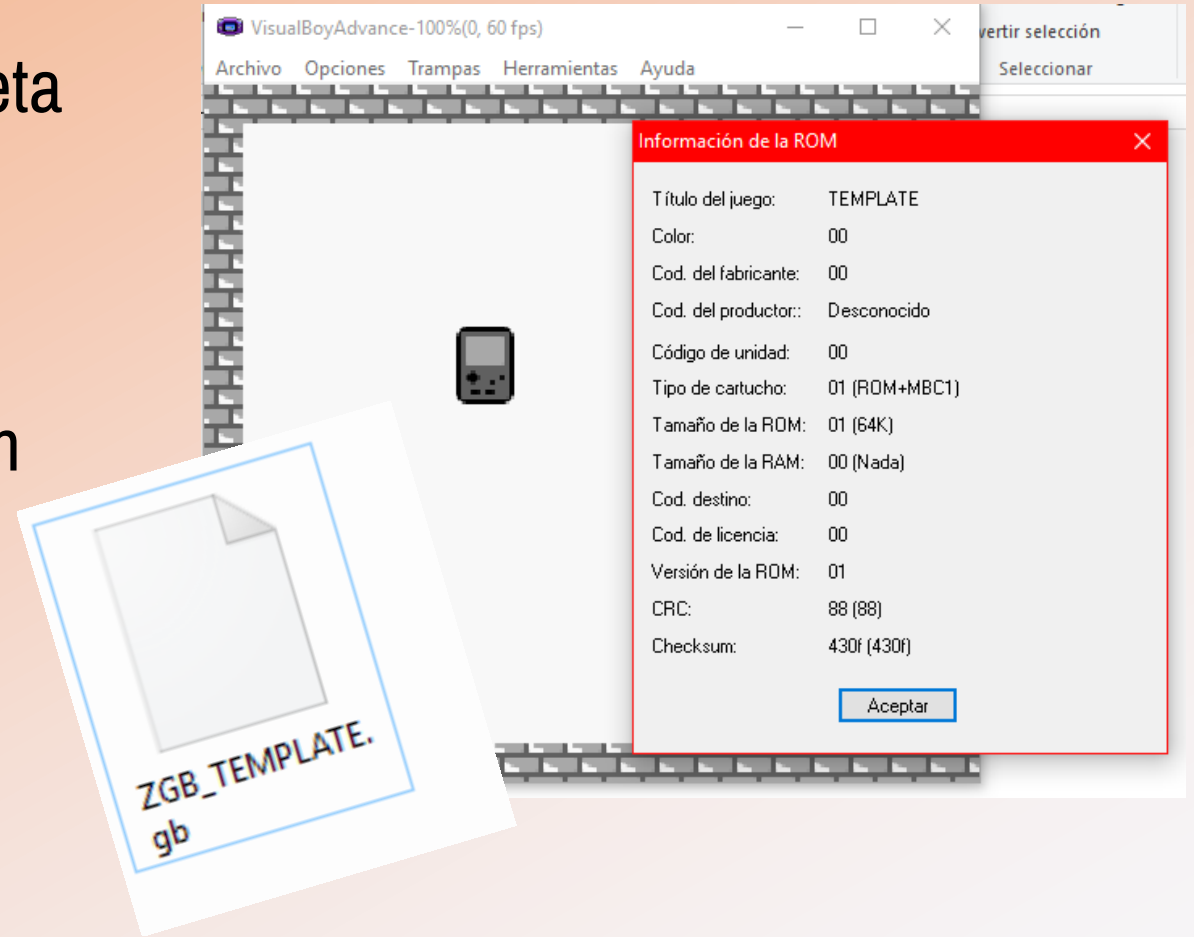
# Mamá! Mira! Mi primer hola mundo en Gameboy!

\*\_\*



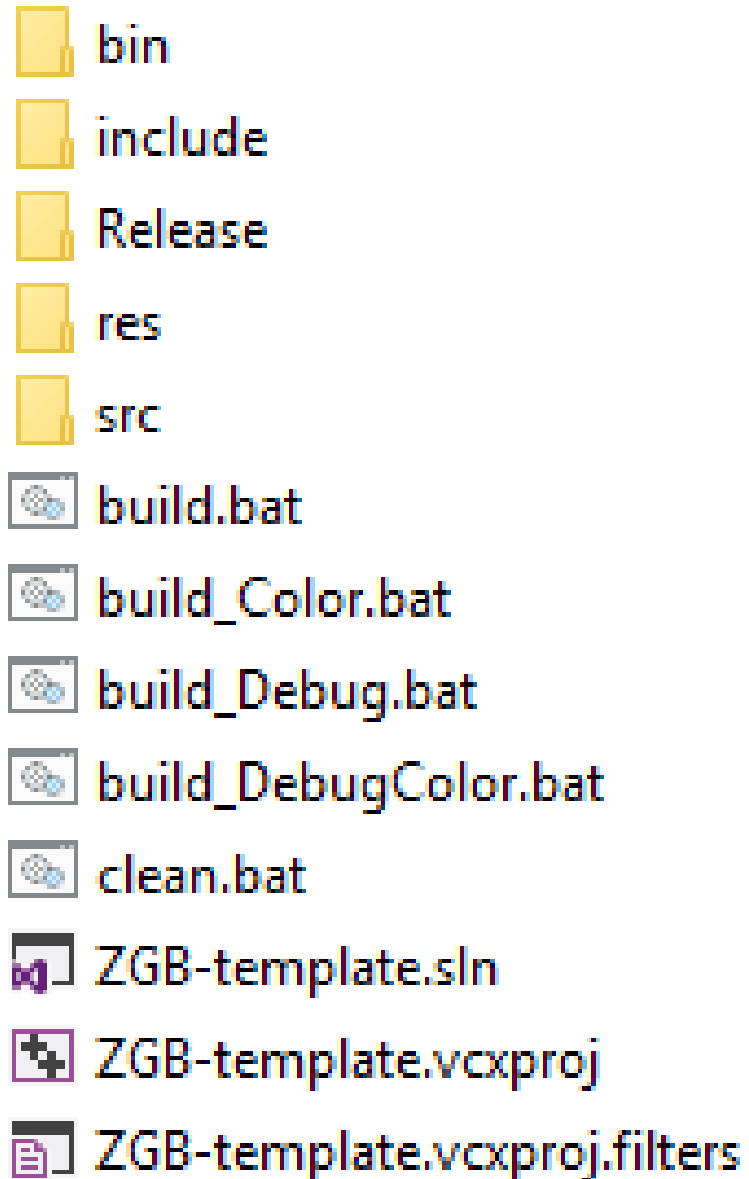
Si te explotó la cara al compilar, mira 14 veces si seguiste los pasos o llama al Enzo pa que te ayude.

- Nuestro Hola Mundo quedó en la carpeta BIN del proyecto.
- Yep! Es una ROM auténtica, fresca y recién salida del horno.
- Y como toda ROM puedes abrirla en un emulador, o volcarla en un cartucho y jugarla en una Gameboy de carne y hueso! :D



Pero ... ¿Como se hacen mas cosas?

# Veamos que hay en la carpeta de nuestro proyecto.



- **Include:** Aquí hay archivos .h (En proyectos de C, para que se sepa que cosas hay, constantes, blabla).
  - **Res:** Aquí van nuestros assets, no compilados y compilados!
  - **Src:** El código fuente de nuestro juego!
  - **Build\_???:** Los programas para compilar nuestra ROM. Los hay en diferentes sabores.
  - **Clean:** Programita para limpiar todas las cosas compiladas. Útil por si hay cosas que explotan y no sabes porque.
  - **ZGB-template.sln/vcxproj/filters:** Por si se te llega a ocurrir usar VisualStudio .
- Ojo: Estas carpetas no estarán si no has compilado nada:**
- **Bin:** Carpeta donde esta nuestra ROM lista
  - **Release:** Cosas pre-compiladas al peo.



ZGB\_TEMPLATE.  
gb

/Bin/

SpritePlayer.h

StateGame.h

ZGBMain.h

/Include/

Makefile

SpritePlayer.c

StateGame.c

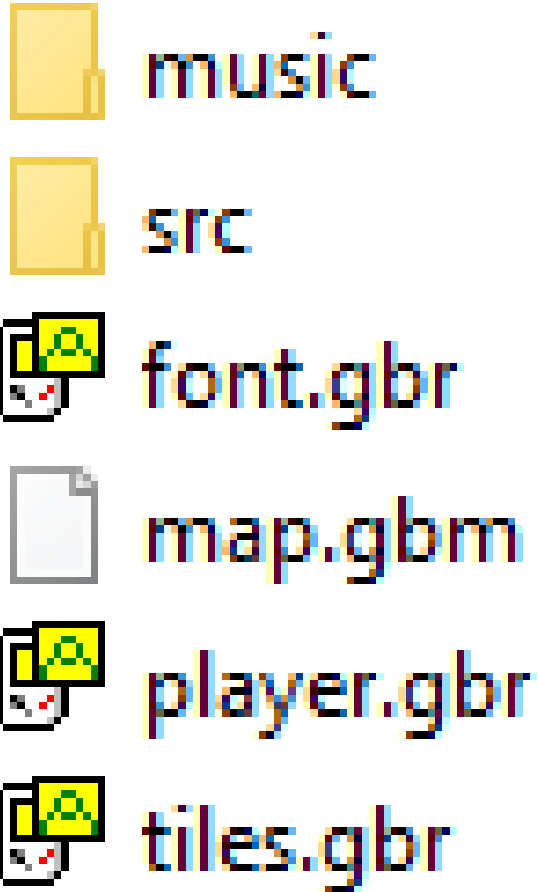
ZGBMain.c

ZGBMain\_Init.c

/src/

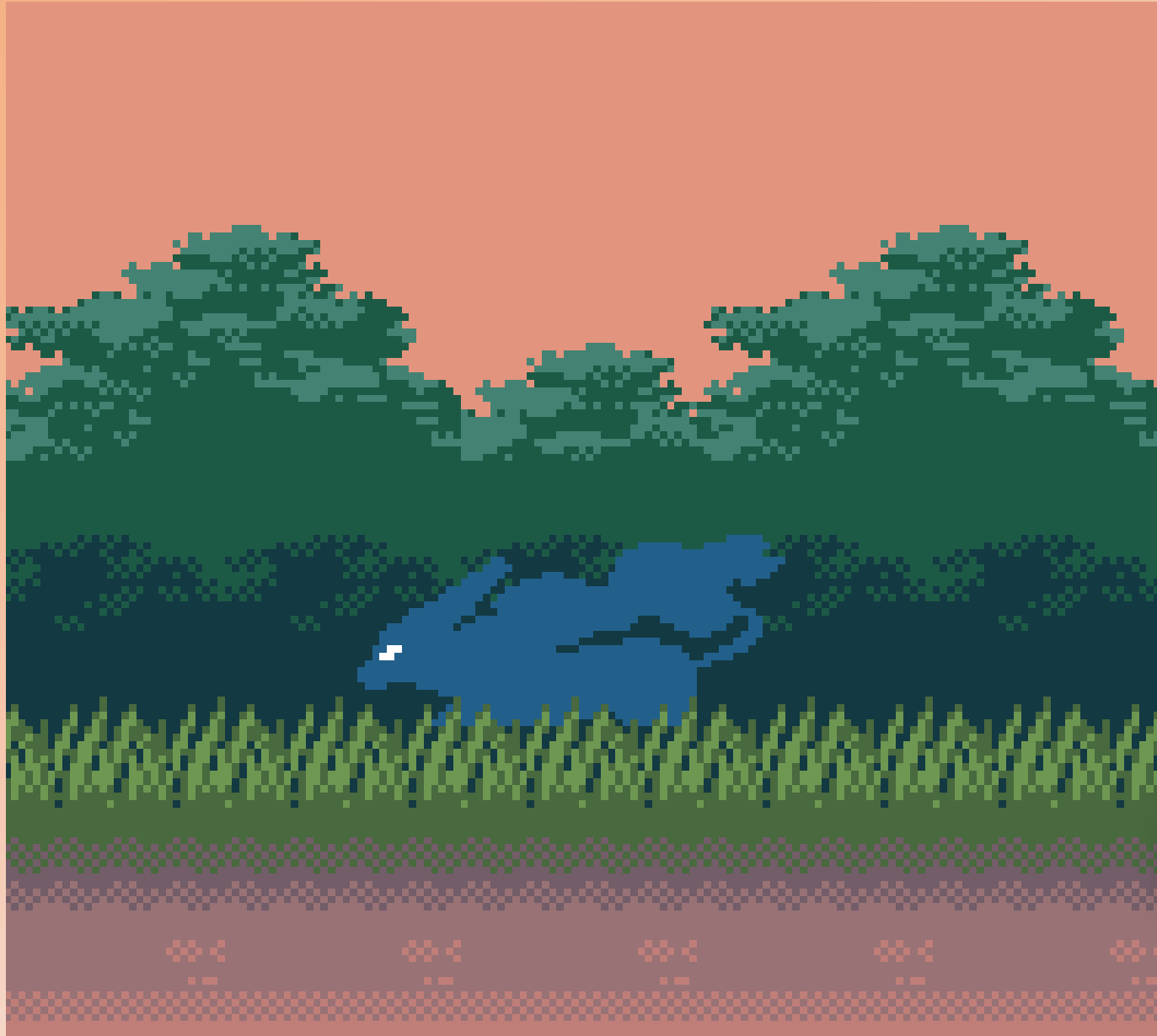
**El contenido de bin, incluye y src, si bien es importante su contenido, a nivel de estructura no tiene mucha gracia.**

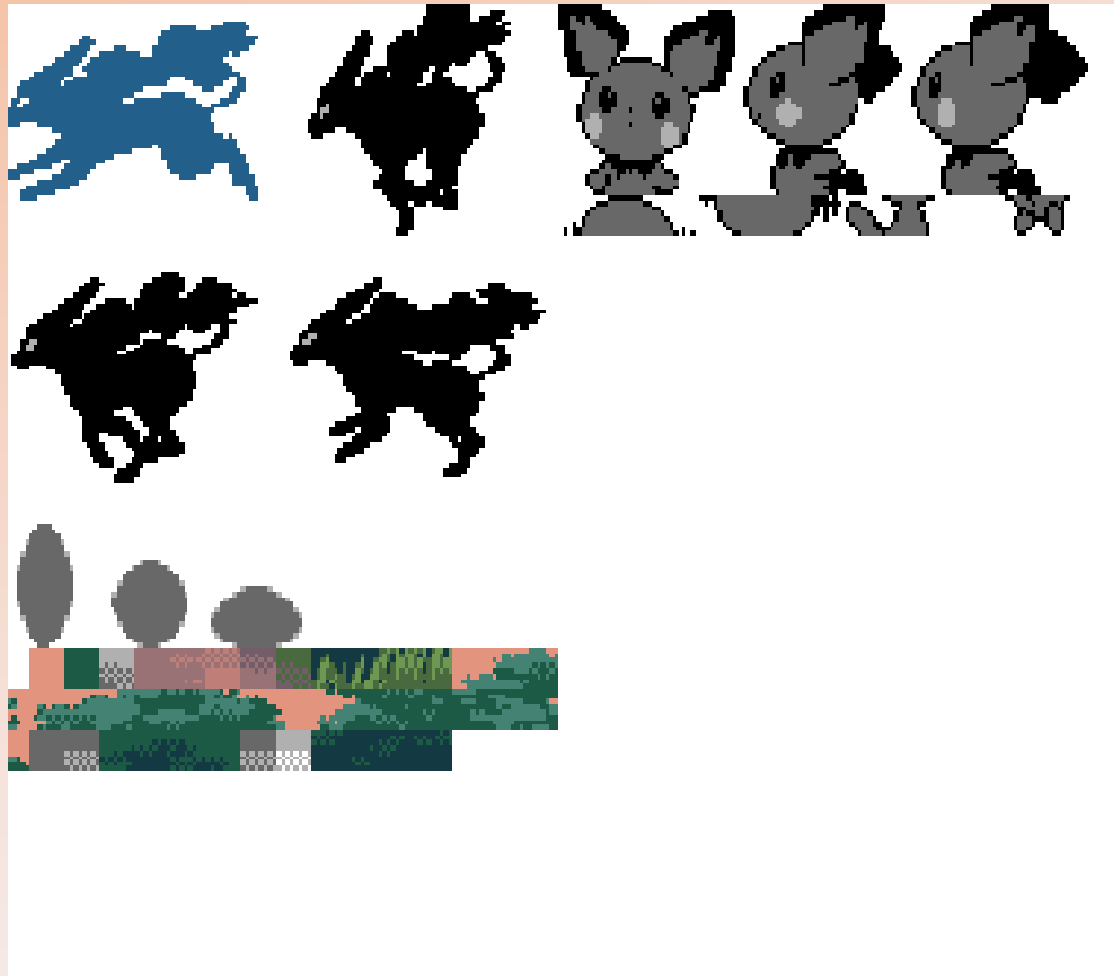
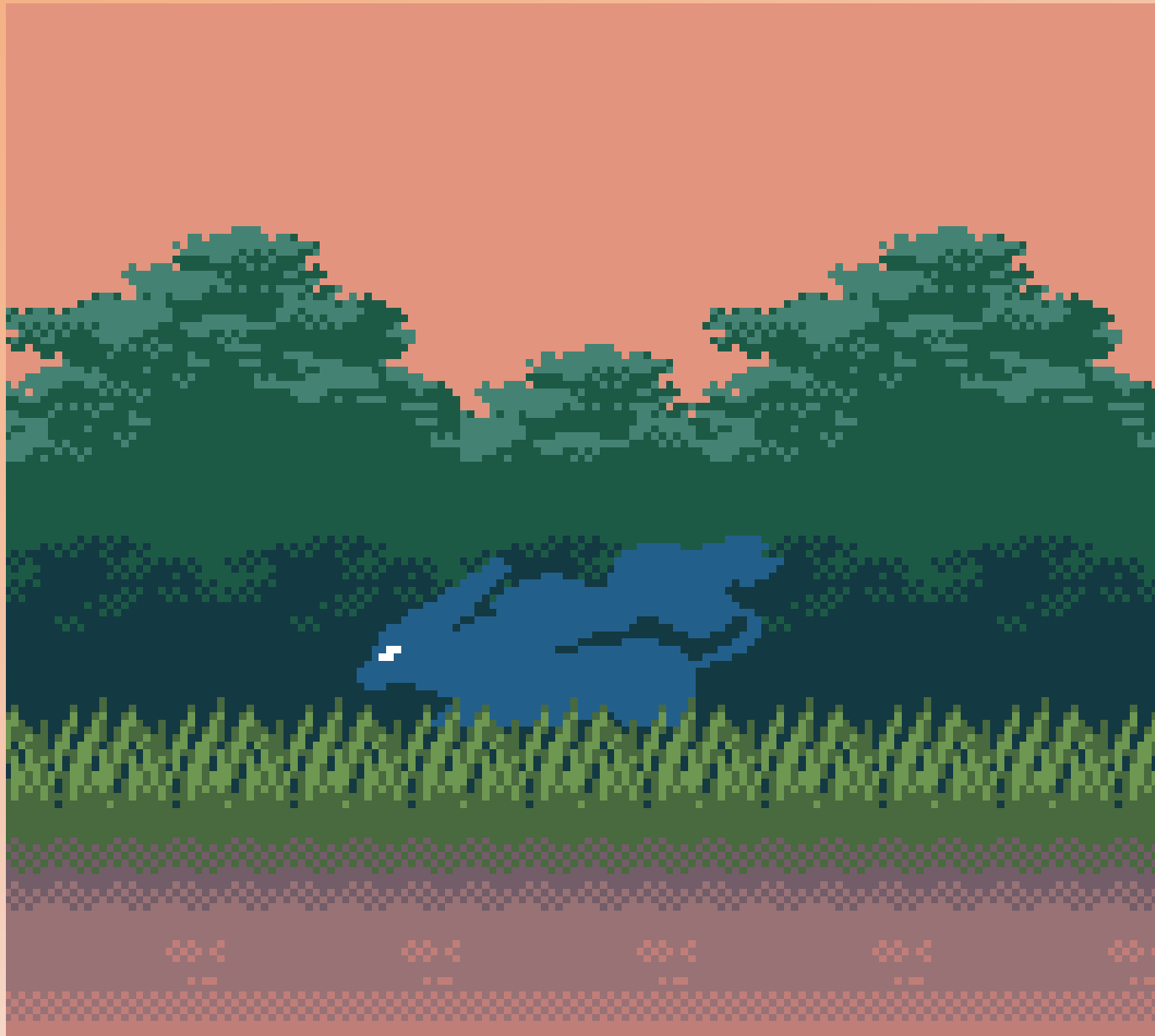
# En /res/ la cosa es mas interesante.



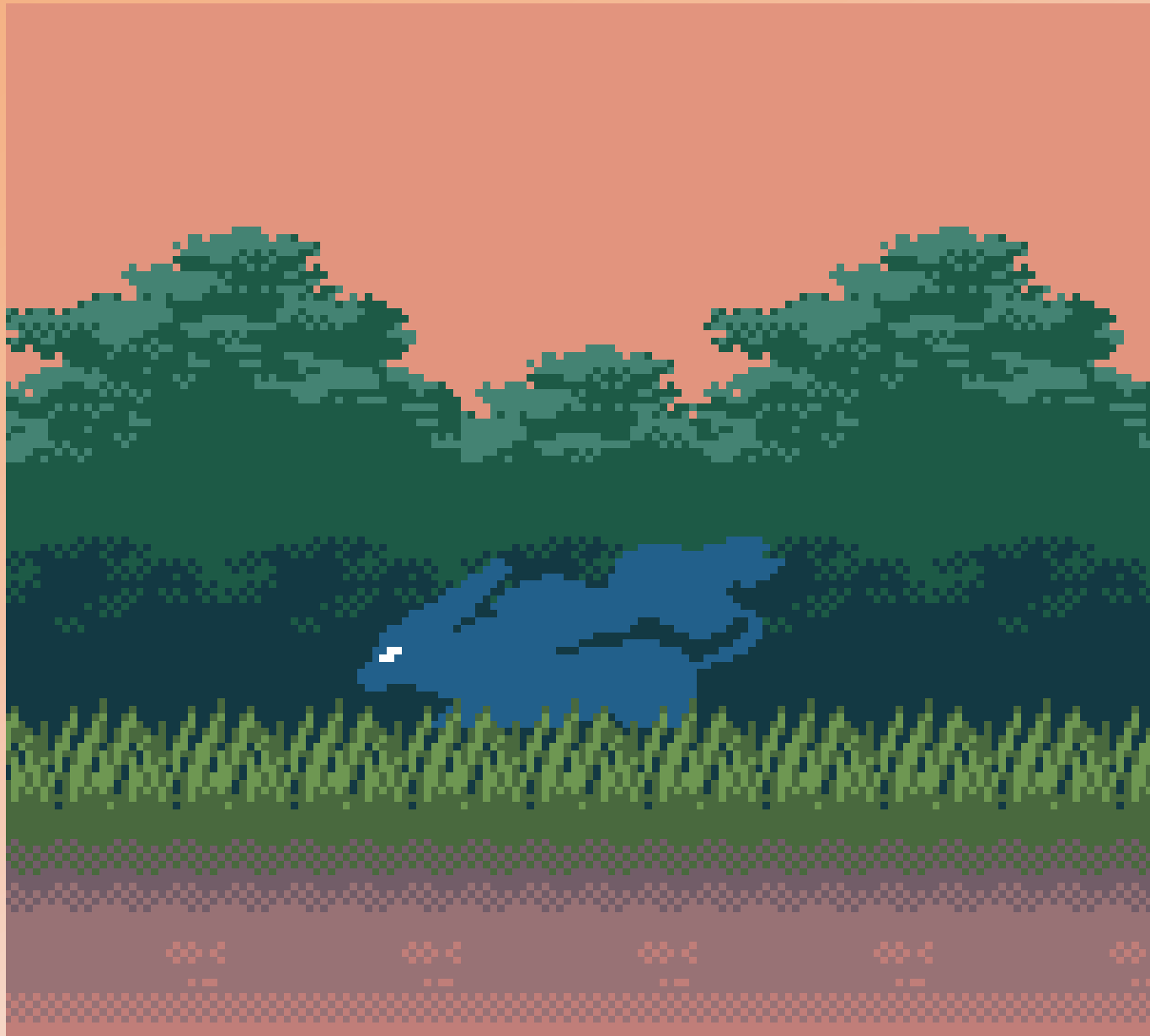
- Nos topamos con archivos en formatos que ni Santa Isabel conoce, y una carpeta llamada `/src/` y `/music/`
- ¡Estos archivos son assets del juego sin compilar!
- Los `.gbr` son graficos/sprites/texturas.
- Los `.gbm` son “pantallas”, tales como imágenes y/o escenarios.
- En `/res/src/` están los graficos compilados.
- En `music` está la música... obvio no. En ...formatos no tan obvios!
- Tranquil@... ¡En el pack dejé los editores para todos esos tipos de archivos!



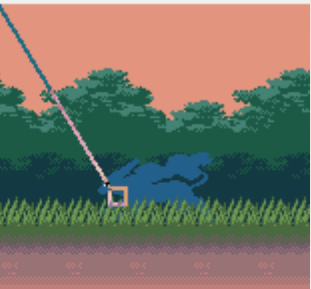




**Gráficos (VRAM/tiles)**



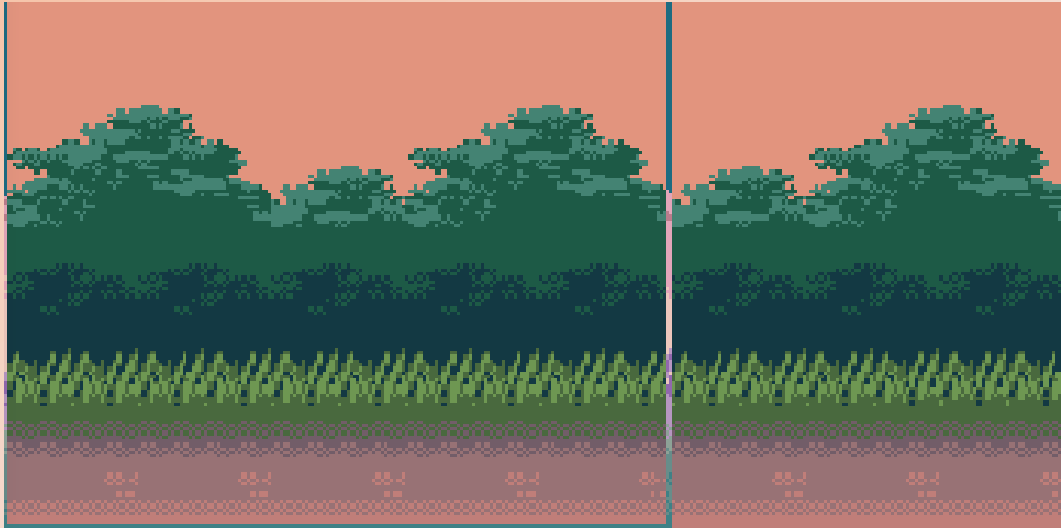
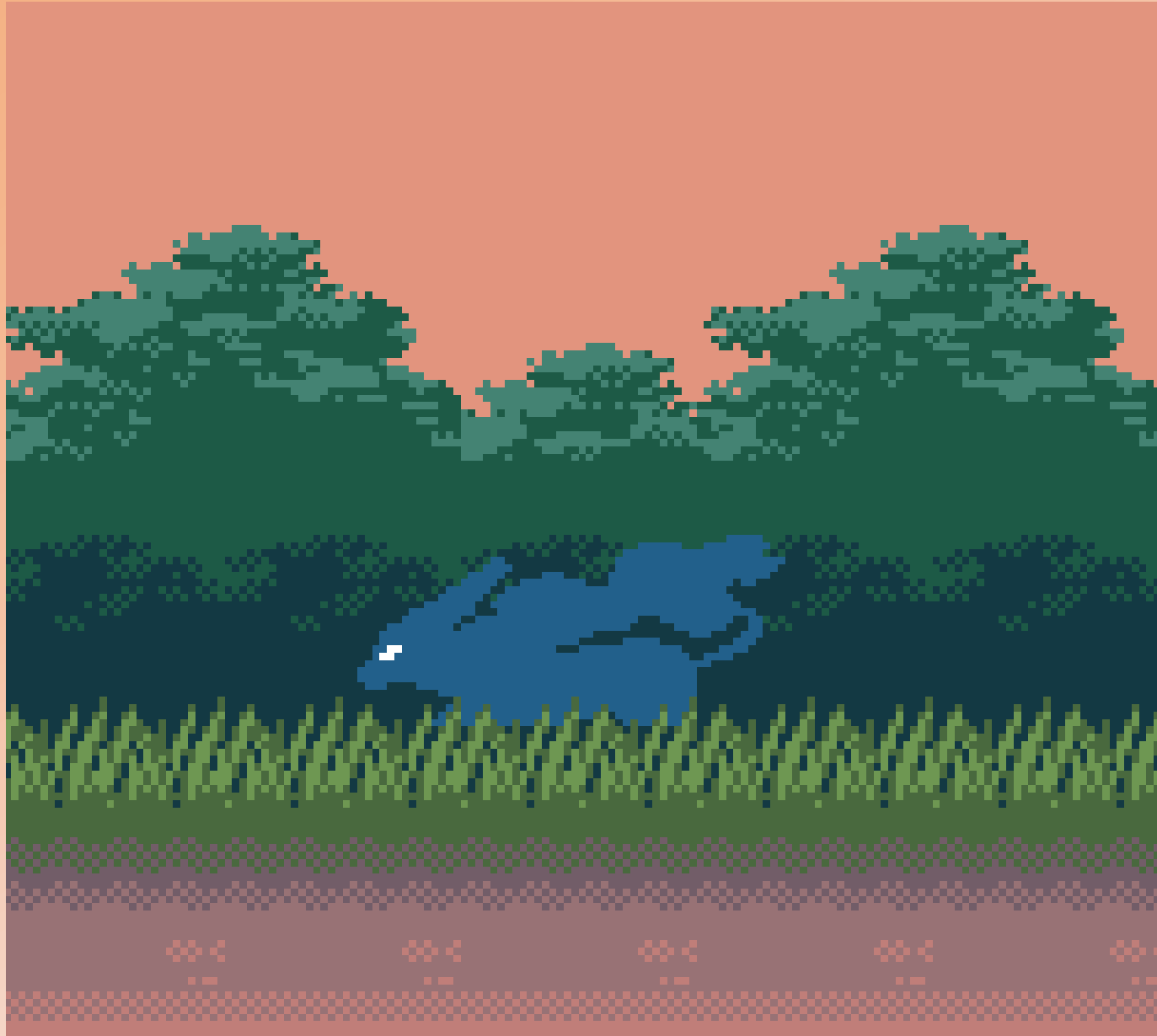
54 60 05 00	54 68 06 00	54 70 07 00	5C 40 11 00	5C 48 12 00	5C 50 13 00	5C 58 14 00	5C 60 15 00
5C 68 16 00	5C 70 17 00	64 38 20 00	64 40 21 00	64 48 22 00	64 50 23 00	64 58 24 00	64 60 25 00
64 68 26 00	64 70 27 00	6C 38 30 00	6C 40 31 00	6C 48 32 00	6C 50 33 00	6C 58 34 00	6C 60 35 00
6C 68 36 00	74 38 40 00	74 40 41 00	74 48 42 00	74 50 43 00	74 58 44 00	74 60 45 00	74 68 46 00
74 70 47	7C 38 50	7C 40 51	7C 70 57	00 00 00	00 00 00	00 00 00	00 00 00



Details

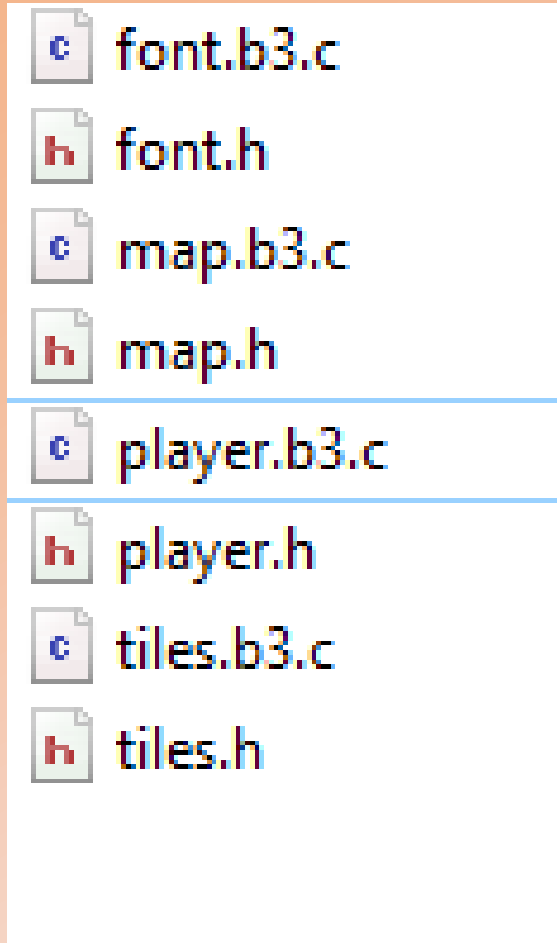
40	X-loc	31	Tile No
6C	Y-loc	00	Attribute
FE4C	OAM addr		
0:8310	Tile Address		
<input type="checkbox"/>	X-flip	OBJ 0	Palette
<input type="checkbox"/>	Y-flip	<input type="checkbox"/>	Priority

## Gráficos (OAM/Sprites)



**Fondos (BG/escenarios)**





# En /res/src están los recursos compilados...




```
/* Start of tile array. */  
const unsigned char player[] =  
{  
    0x1F,0x1F,0x20,0x3F,0x2F,0x30,0x2F,0x30,  
    0x2F,0x30,0x2F,0x30,0x2F,0x30,0x2F,0x30,  
    0x20,0x3F,0x24,0x3F,0x2E,0x3F,0x24,0x3F,  
    0x20,0x3F,0x26,0x3F,0x20,0x3F,0x1F,0x1F,  
    0xF8,0xF8,0x04,0xFC,0xF4,0x0C,0xF4,0x0C,  
    0xF4,0x0C,0xF4,0x0C,0xF4,0x0C,0xF4,0x0C,  
    0x04,0xFC,0x04,0xFC,0x14,0xFC,0x44,0xFC,  
    0x04,0xFC,0xC4,0xFC,0x04,0xFC,0xF8,0xF8,  
    0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,  
    0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,  
    0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,  
    0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,  
    0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,  
    0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,  
    0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,  
    0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,  
};  
/* End of SRC/PLAYER.B3.C */
```

... Si! Como arreglos de C! :D




NOTA: Las músicas se compilan al momento de compilar el proyecto así que no te preocupes de ello por ahora.

 font.b3.c  
 font.h  
 map.b3.c  
 map.h  

---

 player.b3.c  

---

 player.h  
 tiles.b3.c  
 tiles.h

**¿Oye... porque  
los archivos  
estos terminan  
en .b3.c?**



¿Te acuerdas cuando digimos que la GameBoy tenia poquita RAM?

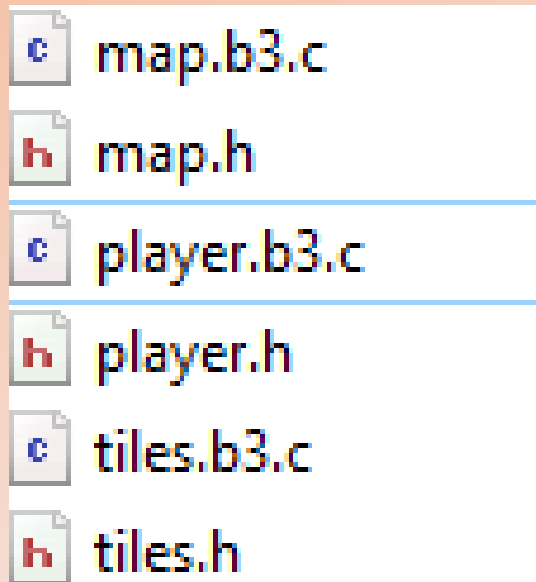
- Los **ROMs** son técnicamente mas grandes de la RAM.
- Para cargar un juego, hay que cargar la **ROM** en la RAM.
- Si, lo de arriba no tiene #\$\$%&/ sentido. No podemos hacerla encajar.
- Como cresta hacemos caer el **Pokemon Crystal** (Una ROM **de 2 megas**) en los malditos 16kb de RAM ?



¿Solución?

# Dividir la ROM en pedazos

Los esclav... digo ingenieros en Nintendo pensaron que seria una excelente idea dividir las ROM en pedazos de 16 kb cada uno. Esos pedazos se llaman **Rom Banks**.



- Entonces, ese **b3** significa que ese grafico compilado se ubicará automáticamente en el **Rom Bank numero 3**, o sea ...si empezamos de cero contando de 16kb, estará aproximadamente entre el kilobyte numero 48 y el kilobyte 64 del cartucho.



La GameBoy solo puede cargar **2 ROM Banks** a la vez.

- La ROM Bank 0 (no, no la puedes cambiar)
- La ROM Bank X (Elije tu que banco quieres).

**Puedes cargar un bank en el segundo slot** cuando se te antoje (cuando necesites cargar los recursos en verdad) dichos bancos, cambiando este cuando sea necesario y necesites otras cosas. La 0 no, así que ahí va lo mas interesante y esencial del juego, tal como un engine o algún mecanismo de “alta performance” (para la gameboy, claro).



# Una noticia buena y 2 malas

**La buena.** Nuestro querido ZGB se encargará del manejo de las ROM Banks :D

**La mala 1.** Eso no nos salva de ubicar nuestros recursos de forma ordenadita e inteligente para aprovechar nuestros bancos (en los 90 las ROM eran caras!)

**La mala 2.** GBDK usa el **banco 0**. ZGB y sus componentes como el audio usan el **banco 1**. El código de tu juego usara casi todo el **banco 2**. Nos queda parte del 3. Nuestro cartucho es de **4 bancos**.

# Bueno... 1 noticia buena mas

Estamos en el 2017. Y tenemos 500 GB en nuestros discos duros para gastarla en ROMs gigantes (o bien, 31.250.000 de bancos) . Por el módico precio de **GRATIS**

Así que si rellenas el ROM Bank 3, llegaremos al límite de nuestro cartucho. Vayamos a editar `/src/Makefile` y agrandemos el tamaño del cartucho **en potencia de 2** (ahora es 4 así que agrandémoslo en **8...** después **16, 32,** y así).

```
PROJECT_NAME = ZGB_TEMPLATE  
  
all: build_gb  
  
N_BANKS = 4  
  
include $(ZGB_PATH)/src/MakefileCommon
```

No en realidad mentía con los 31 millones de bancos. 64 o 128 (color) es la cantidad límite de bancos que un cartucho puede tener. (O sea, 1 o 2 megas)

# Bueno... 1 noticia buena mas

Estamos en el 2017. Y tenemos 500 GB en nuestros discos duros para gastarla en ROMs gigantes (o bien, 31.250.000 de bancos) . Por el módico precio de **GRATIS**

Asi que si rellenas el ROM Bank 3, llegaremos al limite de nuestro cartucho. Vayamos a editar `/src/Makefile` y agrandemos el tamaño del cartucho **en potencia de 2** (ahora es 4 así que agrandémoslo en **8...** después **16, 32,** y así).

```
PROJECT_NAME = ZGB_TEMPLATE  
  
all: build_gb  
  
N_BANKS = 4  
  
include $(ZGB_PATH)/src/MakefileCommon
```

No en realidad mentía con los 31 millones de bancos. 64 o 128 (color) es la cantidad limite de bancos que un cartucho puede tener. (O sea, 1 o 2 megas)

# Bueno... 1 noticia buena mas

Estamos en el 2017. Y tenemos 500 GB en nuestros discos duros para gastarla en ROMs gigantes (o bien, 31.250.000 de bancos). Por el módico precio de **GRATIS**

Asi que si rellenas el ROM Bank 3, llegaremos al limite de nuestro cartucho. Vayamos a editar `/src/Makefile` y agrandemos el tamaño del cartucho en potencia de 2 (ahora es 4 así que agrandémoslo en 8... después 16, 32, y así).

```
PROJECT_NAME = Z80_TEMPLATE  
...  
...  
...  
... /src/Makefile ...
```



... limite de bancos que un cartucho puede tener. (O sea, 1 o 2 megas)

# Esta historia continuará!

- Nos introducimos en la **GameBoy**, en el mundillo **homebrew** y la escena underground de la informática.
- Conocimos el oscuro mundo del **desarrollo de videojuegos** en los viejos tiempos.
- Conocimos el **GBDK**, y el **ZGB**, herramientas que nos harán la vida teóricamente mas fácil a la hora de hacer juegos.
- Instalamos un **kit de desarrollo** en nuestros computadores con “casi” todos los programas que necesitaremos (pilas no incluidas).
- Hicimos **nuestro primer hola mundo**, teniendo suficiente material para presumir en nuestras redes sociales :D
- **Nos quedamos con ganas de como cresta hacer juegos**, pero tendremos que esperar hasta mañana.

# Mañana



- Comprar pilas
- Limitaciones propias de la GameBoy!
- Aprenderemos como esta compuesto un proyecto de ZGB
- Haremos un Sprite que se mueva
- Un escenario donde pueda ese Sprite moverse
- Comprar mas pilas!
- Mas pilas porque no fueron suficientes
- Enemigos!
- Música!
- Estados y menús
- Conquistar el mundo
- Bailar la macarena (?)
- Después de eso, comprar aún mas pilas!

# ¿Aprender mas?

- <http://gbdev.gg8.se> Blog y Wiki de desarrollo de Gameboy
- <https://github.com/Zal0/ZGB> Repositorio y documentación oficial de ZGB
- <http://gbdk.sourceforge.net> Documentación oficial de GBDK
- <https://github.com/avivace/awesome-gbdev> Recopilacion de xuxerias para desarrollar en Gameboy y ser feliz
- <https://www.youtube.com/watch?v=RZUDEaLa5Nw> Gameboy autopsy. Video interesante que explica todo de la gameboy
- <https://www.youtube.com/watch?v=HyzD8pNlpwl> The ultimate gameboy talk!
- <http://www.littlesounddj.com/lzd/> LSDj, por si quieres ser DJ con GameBoy!
- [https://www.google.cl/?gws\\_rd=ssl](https://www.google.cl/?gws_rd=ssl) Google es tu amigo!
- <http://procatinator.com> Hay que relajarse digo yo de vez en cuando, no?



# ¿Dudas antes que me escape con los millones de dólares?

Las PPT y el devkit los puedes encontrar aquí en  
<http://elsemieni.net/gb/>

**Nintendo**  
**GAME BOY**<sup>®</sup>

**¡Aprenda homebrew y haga su juego 100% retro!**

**Martes 3 Octubre (Parte 1) - 12:15 hrs**  
**Miércoles 4 Octubre (Parte 2) - 14:00 hrs**  
**Escuela Ingeniería Informática PUCV (Piso 2)**  
**PUCV Edificio Isabel Brown Caces**  
**Av. Brasil 2241, Valparaíso**

**Videojuegos retro en**  
**GAME BOY**  
**... de verdad!**



**Nos vemos mañana!**

**Videojuegos retro en**  
**Gameboy ... de verdad!**  
**(Parte 1)**

**Enzo Barbaguelatta D.**