

Durango Computer

Durango Computer Team - La Jaquería
Eslibre 2023

Índice

1. Introducción
2. Home PC Revolution
3. Procesador 6502
4. Durango
5. Hardware
6. Software
7. Development
8. Referencias

1. Introducción

¿Quién soy? ¿Qué hago a más de 750 KM de mi casa?

Soy Víctor Suárez García. Ingeniero en informática por la UAL y miembro activo de la asociación La Jaquería.

Siempre estoy aprendiendo cosas y tratando de dar charlas como esta.

Además de colaborar en proyectos libres como en este caso el proyecto Durango.



2. Home PC Revolution

Un poco de historia...

En los años 60/70, los ordenadores eran tan grandes como habitaciones y su precio solo los hacía estar disponibles para grandes empresas.



Sin embargo, llegó la “Home Computer Revolution” con algunos de los primeros pcs personales.



2. Home PC Revolution

- Zx Spectrum.
- Atari 400/800.
- Commodore 64.
- Apple II.



2. Home PC Revolution

- Zx Spectrum.
- Atari 400/800.
- Commodore 64.
- Apple II.



3. Procesador 6502

El procesador 6502, fue diseñado por el mismo equipo que el motorola 6800 que abandonó motorola.

Es un procesador de 8 bits, con un bus de dirección de 16 bits. Además de haber sido utilizado en muchos dispositivos de la época (Apple II, Atari 2600, NES,C64).

En la actualidad se siguen utilizando para su utilización en marcapasos.



4. Durango

Y si....

Usamos el 6502 y creamos un ordenador hoy en día?

Totalmente libre.

Con herramientas modernas.

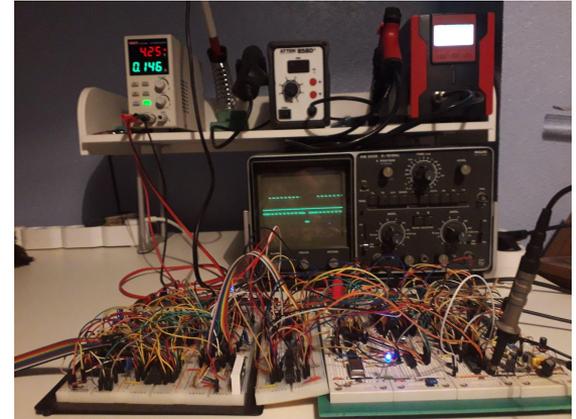
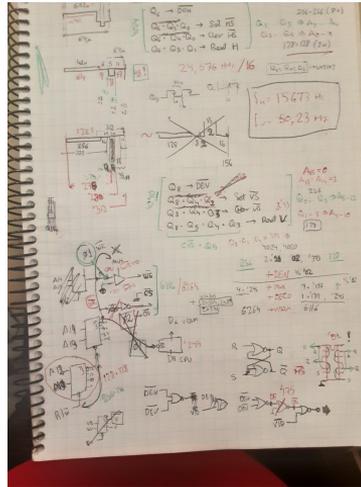
Que pueda usarse tanto en aplicaciones de hoy en día como antiguamente.

4. Durango

Un poco de historia...

Durango es un proyecto personal creado por mi compañero Carlos Santisteban ([@zuiko21](#)).

Durante los últimos años ha ido diseñando tanto arquitectura como el diseño de algunos componentes.



4. Durango

Durango Computer Team



Carlos Santisteban



Emilio Lopez



Víctor Suárez

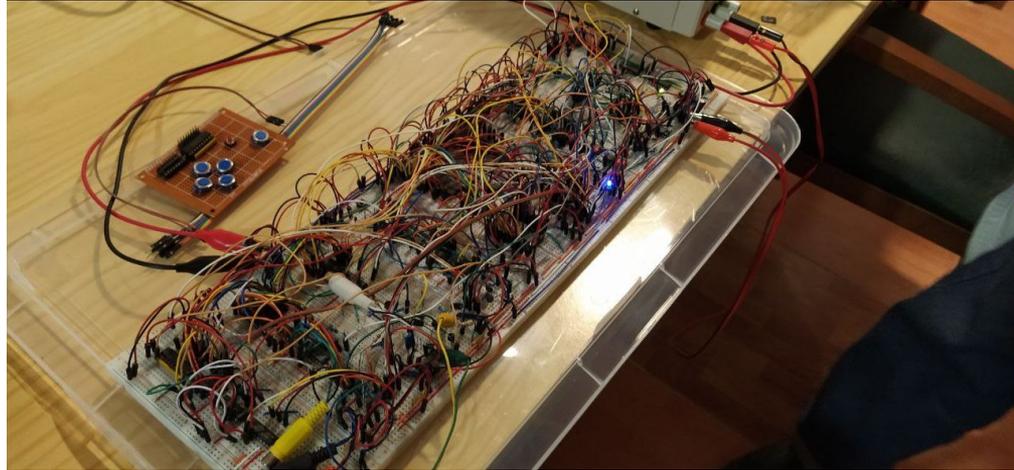


Tú

Durango-X

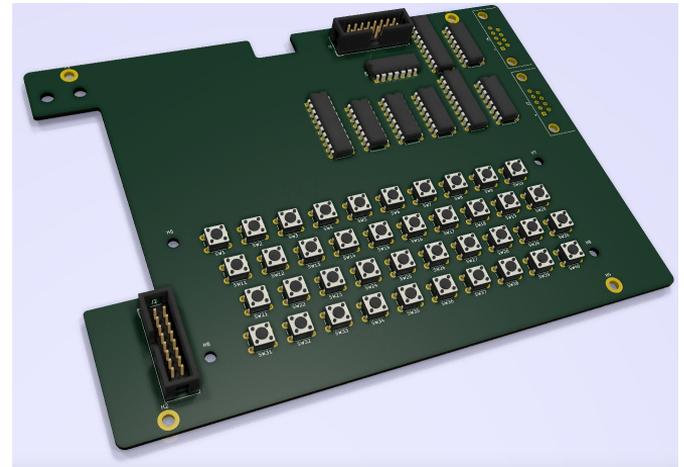
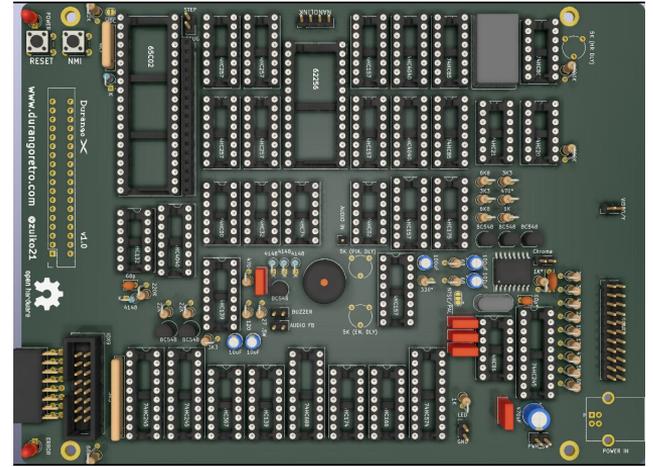
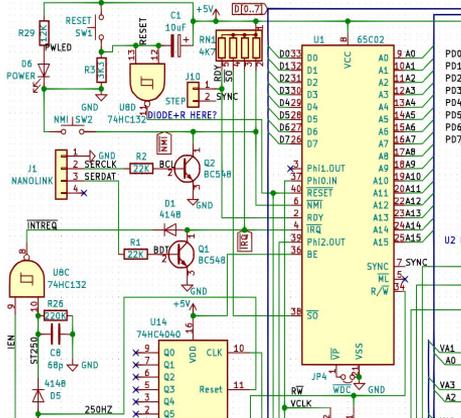
Se trata de la última revisión de la arquitectura y utiliza diferentes configuraciones y componentes.

- Permite mostrar 16 colores a una resolución de 128x128 píxeles.
- Permite una versión “High-Res” monocromática de 256x256 píxeles.



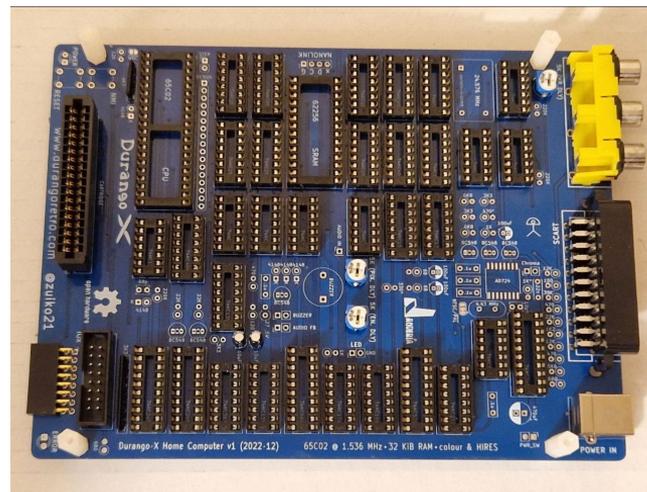
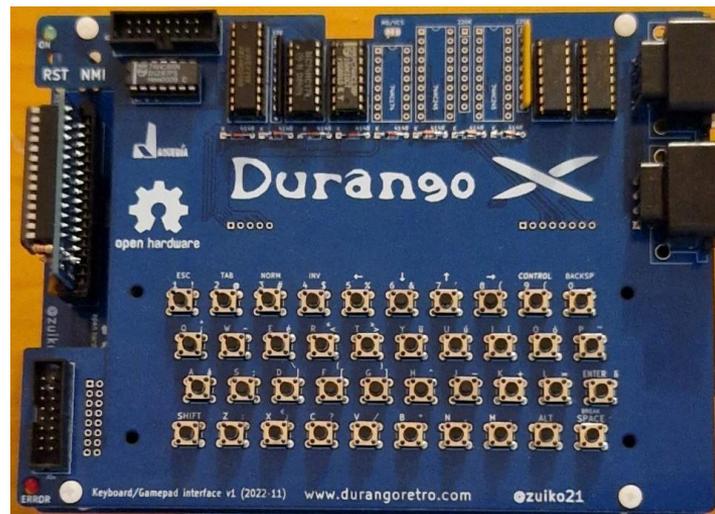
Durango-X

El proyecto ha evolucionado de tal forma que ya se tienen las primeras unidades funcionales; y además, se está trabajando en crear herramientas y software para el mismo.



Durango-X

El proyecto ha evolucionado de tal forma que ya se tienen las primeras unidades funcionales; y además, se está trabajando en crear herramientas y software para el mismo.



5. Hardware

Durango X

Especificaciones:

- Procesador CMOS 6502 a 1,5Mhz
- 32KB SRAM (ampliable por cartucho devcart).
- 32KB ROM.
- Conexión de hasta 16 dispositivos usando el Puerto paralelo de 8 bits
- Conexión con Raspberry Pi usando el puerto *NanoLink*.
- Modos de Vídeo:
 - 16 colores a 128x128 píxeles
 - modo “high-res” monocromático 256x256 píxeles
- Conexiones de vídeo:
 - RGB usando Euroconector
 - Sync on Green usando RCA
 - Modo por componentes (próximamente).
 - Opción vídeo por PAL (experimental)



5. Hardware

KeyBoard and Gamepad Interface

Permite conectar un teclado y hasta dos controladores.

Especificaciones:

- Compatible eléctricamente (tienes hasta los conectores) con teclado de ZX Spectrum.
- Posibilidad de dos opciones:
 - Modo NES: Permite conectar hasta 2 controladores de NES.
 - Modo MD/Atari: Permite conectar hasta 2 controladores de Mega Drive o Atari.



5. Hardware

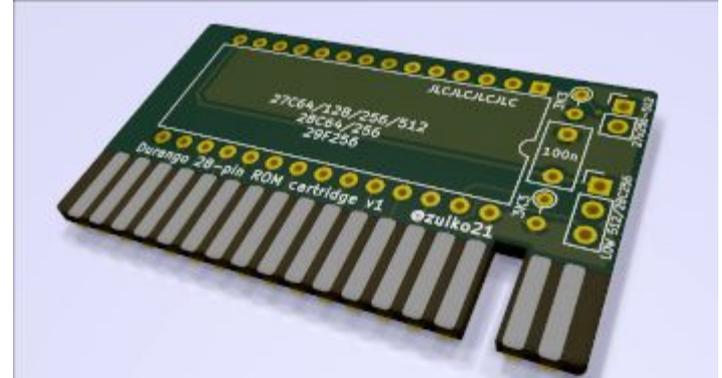
Cartucho ROM

Permite almacenar una ROM de hasta 32 KB*.

Especificaciones:

- Almacena una EPROM/EEPROM de hasta 32KB*.
- Tiene diferentes modos y podemos almacenar el programa que necesitamos para durango.

* Permite ampliar la información usando bankSwitch (usando el botón NMI para cambiar de ROM).



5. Hardware

Cartucho Dev Cart

Permite almacenar información en una SDcard y además ampliar la RAM con hasta 32KB adicionales (usando shadow ROM).

Especificaciones:

- Almacena una EPROM/EEPROM de hasta 32KB.
- Añade 32KB de SRAM adicionales.
- Permite cargar ROMS y guardar información* en una tarjeta SD.
- Permite recibir programas e información a partir de una Raspberry Pi usando la conexión *nanolink*.

* Aún en proceso de desarrollo.



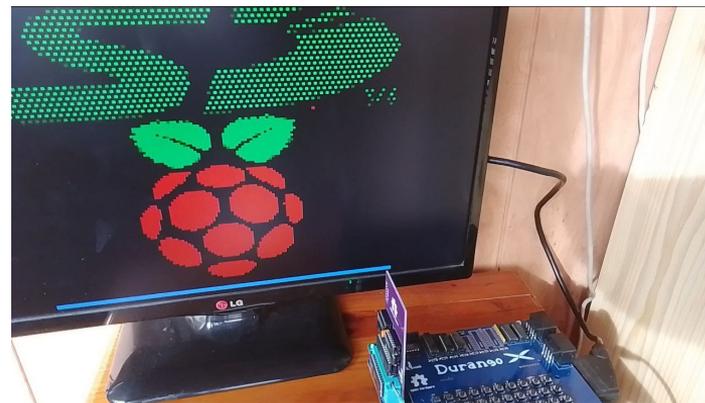
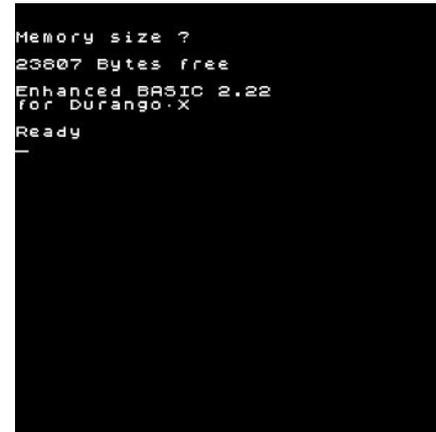
6. Software

“Pero el hardware sin software... sólo son chises y cables...”

Por ello, se ha desarrollado una serie de aplicaciones y juegos que pueden ampliarse:

- Aplicaciones:
 - *ehbasic*: Intérprete BASIC con comandos propios incluidos..
 - *NanoBoot*: Permite cargar programas desde una Raspberry Pi usando la interfaz NanoLink.
 - *FullTest*: Utilidades de Test de Durango (usa BankSwitch para cambiar de test (botón NMI)).
 - *NanoPython*: Intérprete para el lenguaje Python (en desarrollo).

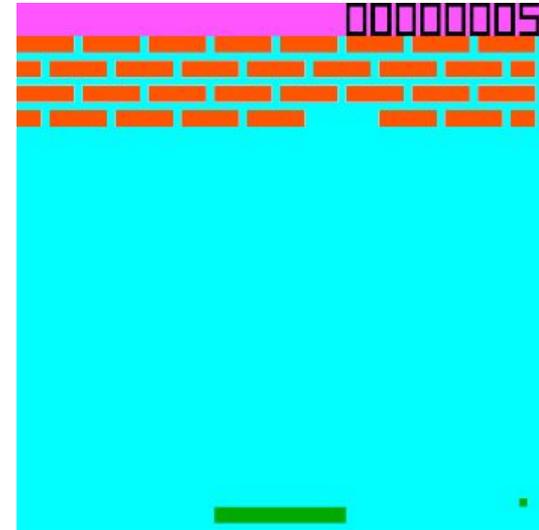
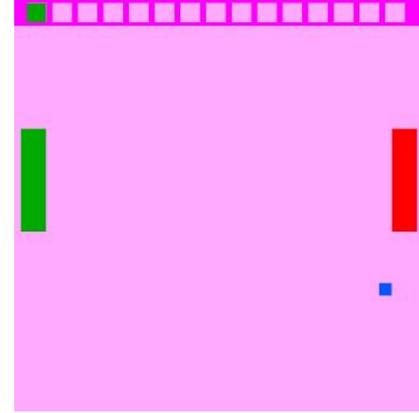
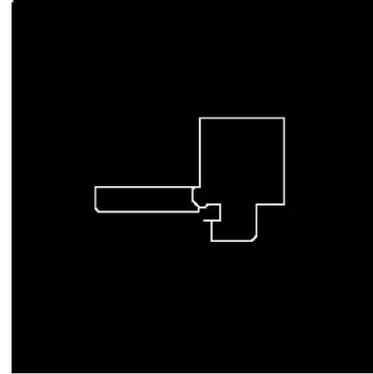
Y si! todos totalmente libres.



6. Software

“Pero el hardware sin software... sólo son chises y cables...”

- Juegos:
 - Brick/arkanoid: Juego clásico para romper bloques.
 - Pong: Juego clásico de 2 jugadores.
 - Telesketch: Juego para poder dibujar por pantalla usando las direcciones.



7. Desarrollo

Pero lo importante del proyecto... Su ecosistema

Durango es un proyecto libre que se centra en que cualquiera pueda aportar y crear su propio contenido.

NO es necesario construir un Durango ya que se disponen de herramientas de desarrollo modernas para crear tanto aplicaciones como juegos.

- Emulador (Perdita)
- Herramientas de desarrollo
 - CC65
 - Rescomp
- Librería propia para el lenguaje C.
- Imágenes para Docker con todas las herramientas publicadas en DockerHub (también compatibles para podman).

Por supuesto, todas las herramientas (mientras no se indique lo contrario), son libres.

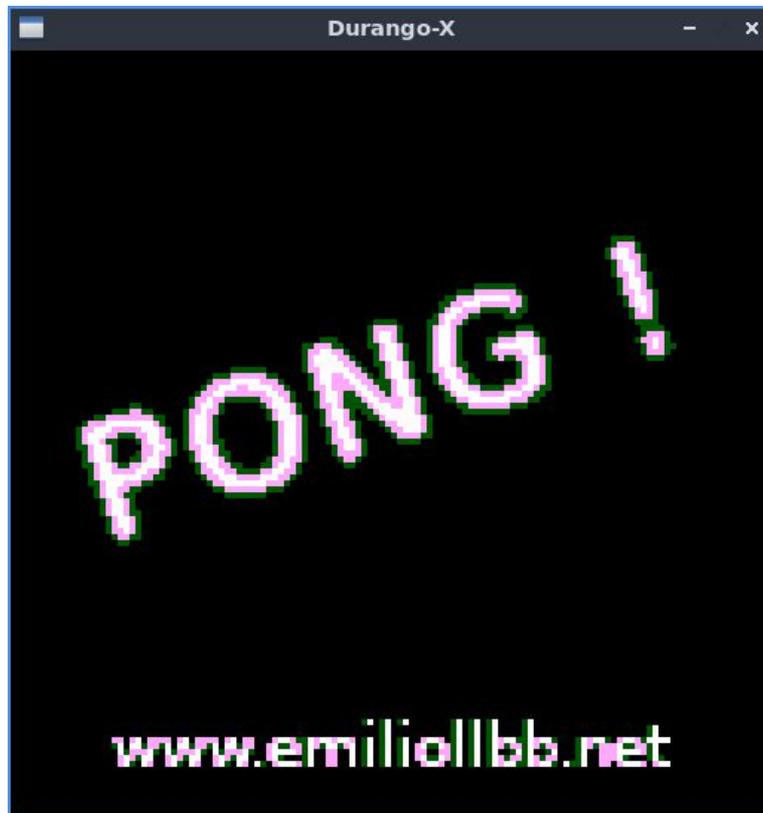
7. Desarrollo

Perdita

Perdita es un emulador para Durango-X escrito en C y utiliza la librería SDL2 para mostrar la pantalla.

- Soporte para teclado y gamepads
- Soporte para PSV (Puerto serie Virtual) para enviar o recibir información del exterior.

Por supuesto; código fuente libre.



7. Desarrollo

Durango Lib

Librería escrita en C y assembler para crear software de forma amigable para Durango.

- Permite crear pequeños programas tanto de forma gráfica como utilizando distintos periféricos (teclado, gamepad, etc...).

Totalmente libre.

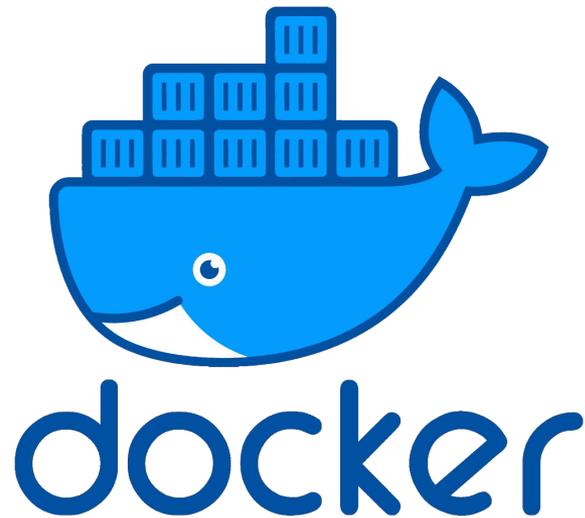
```
1 /* Copyright 2022 Durango Computer Team
2
3 Use of this source code is governed by an MIT-style
4 license that can be found in the LICENSE file or at
5 https://opensource.org/licenses/MIT.
6 */
7
8 /** @file durango.h Main Header File */
9
10 #ifndef _H_DURANGO
11 #define _H_DURANGO
12
13 /* TYPE DEFINITIONS */
14 typedef unsigned char byte;
15 typedef unsigned char u8;
16
17 /* PROCEDURES */
18 extern void __fastcall__ setHiRes(unsigned char);
19
20 /**
21  * Fill the entire Screen of one color
22  * @param color one of the 16 colors to print. Check video.h to see the 16 colors Macros.
23  */
24 extern void __fastcall__ fillScreen(unsigned char);
25
26 /**
27  * Draw a Pixel on Screen
28  * @param x: X Coord in pixels. The x coordinate is from left to Right.
29  * @param y: Y Coord in pixels. The Y coordinate is from up to Down.
30  * @param color: color to use.
31  */
32 extern void __cdecl__ drawPixel(unsigned char x, unsigned char y, unsigned char color);
33
```

7. Desarrollo

Docker

Se ha creado una imagen docker con todas las herramientas necesarias para desarrollar para Durango.

- Librería Durango Llb
- Herramienta CC65 (incluyendo todo el software).
- Herramienta Rescomp basada en Java.
- Compatible con x86_64 y ARMv7
- Se está trabajando para que también pueda usarse con Podman.



<https://hub.docker.com/r/zerasul/durangodevkit>

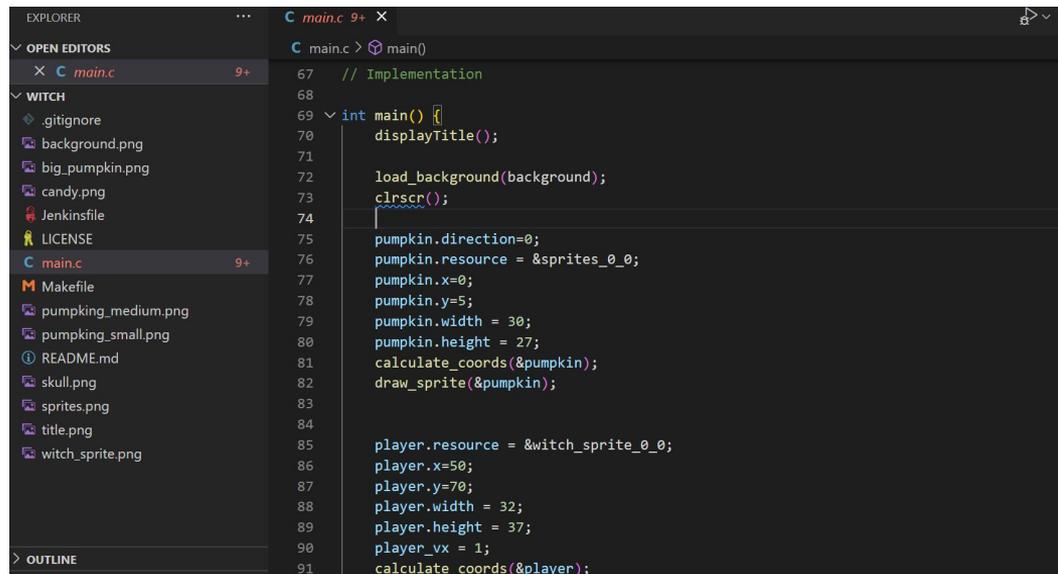
7. Desarrollo

Durango Code (En desarrollo)

Extensión para Visual Studio Code/codium, que ayuda a la hora de crear software para Durango.

- Permite compilar/crear proyectos para Durango.
- Permite enviar y recibir información tanto a Durango como a Perdita.

NOTA: Esta extensión aún no está disponible.



```
EXPLORER
OPEN EDITORS
  C main.c 9+
WITCH
  .gitignore
  background.png
  big_pumpkin.png
  candy.png
  Jenkinsfile
  LICENSE
  C main.c 9+
  Makefile
  pumpking_medium.png
  pumpking_small.png
  README.md
  skull.png
  sprites.png
  title.png
  witch_sprite.png
OUTLINE

C main.c 9+ X
C main.c > main()
67 // Implementation
68
69 int main() {
70     displayTitle();
71
72     load_background(background);
73     clrscr();
74
75     pumpkin.direction=0;
76     pumpkin.resource = &sprites_0_0;
77     pumpkin.x=0;
78     pumpkin.y=5;
79     pumpkin.width = 30;
80     pumpkin.height = 27;
81     calculate_coords(&pumpkin);
82     draw_sprite(&pumpkin);
83
84
85     player.resource = &witch_sprite_0_0;
86     player.x=50;
87     player.y=70;
88     player.width = 32;
89     player.height = 37;
90     player_vx = 1;
91     calculate_coords(&player);
```

8. Referencias

- <https://www.durangoretro.com>
- <https://lajaqueria.org>
- <https://github.com/durangoretro/>
- https://es.wikipedia.org/wiki/Sinclair_ZX81
- https://es.wikipedia.org/wiki/MOS_6502
- https://es.wikipedia.org/wiki/Commodore_64
- https://es.wikipedia.org/wiki/Apple_II
- <https://es.wikipedia.org/wiki/Furby>
- <https://hub.docker.com/r/zerasul/durangodevkit>