

# DURANGO: PYTHON EN UN ORDENADOR RETRO

VÍCTOR SUÁREZ GARCÍA  
PYCONES 2023





# ÍNDICE



- INTRODUCCIÓN
- EL 6502
- EL PROYECTO DURANGO
- SOFTWARE RETRO
- ¿PYTHON RETRO?
- NANOPYTHON (POC)





# 1. INTRODUCCIÓN

¿A QUÉ VIENE AQUÍ ESTE GODO?  
SOY VÍCTOR SUÁREZ  
PROGRAMADOR INFORMÁTICO Y  
EX CANARIO. CREO QUE JAVA ES  
EL MEJOR LENGUAJE DEL  
MUNDO.





# 1. INTRODUCCIÓN

¿A QUÉ VIENE AQUÍ ESTE GODO?  
SOY VÍCTOR SUÁREZ  
~~PROGRAMADOR INFORMÁTICO~~  
PROFESOR EDUCACIÓN  
SECUNDARIA, EX CANARIO Y HE  
VENIDO A HABLAR DE COSAS  
RETRO.





# 1. INTRODUCCIÓN



Y SI! SOY UN EX CANARIO... ADICTO AL CLIPPER Y JAVERO!!!





# 1. INTRODUCCIÓN





## 2. EL 6502



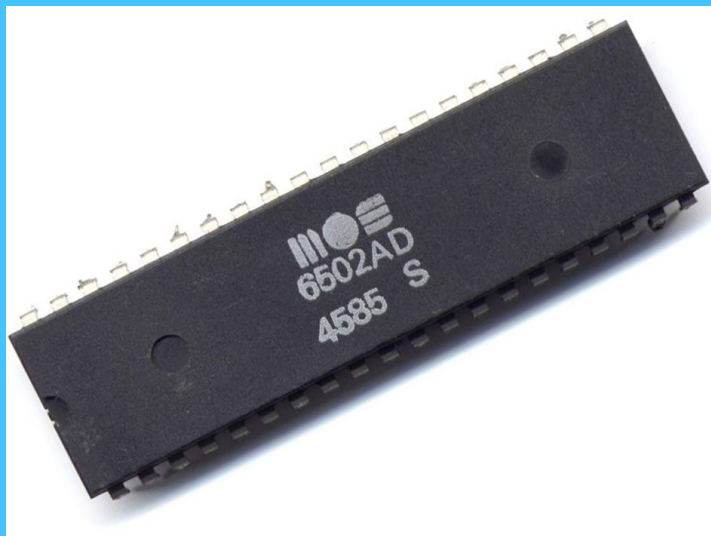
EL 6502, ES UN PROCESADOR DE 8 BITS CON DIRECCIONAMIENTO DE 16 BITS. SE UTILIZÓ EN VARIOS DISPOSITIVOS:

- APPLE II
- COMMODORE 64
- NES
- ATARI





## 2. EL 6502







## 2. EL 6502



PERO QUÉ OCURRE SI USAMOS ESTE PROCESADOR Y  
CREAMOS UN ORDENADOR CON LAS HERRAMIENTAS E  
IDEAS DEL SIGLO XXI





# 3. DURANGO

EN PYCONES 2022, SE  
PRESENTÓ ESTE PROYECTO  
COMO LIGHTING TALK... PERO  
¿QUÉ ES DURANGO?

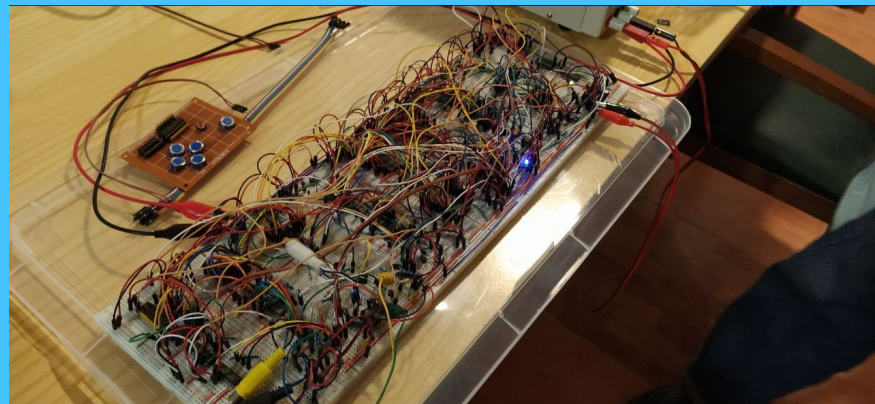




# 3. DURANGO

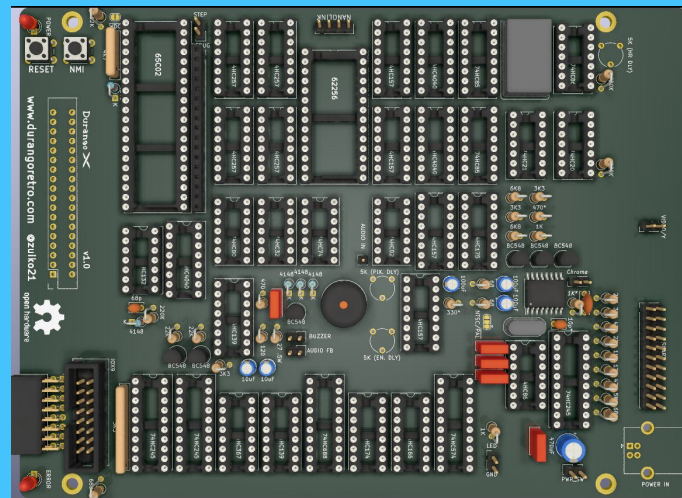


DURANGO ES UN ORDENADOR  
CREADO CON HERRAMIENTAS  
MODERNAS, PERO USANDO UN  
PROCESADOR RETRO...  
CONCRETAMENTE EL 6502



# 3. DURANGO

- PROCESADOR 6502 A 1,5MHZ.
- 32KB RAM ESTÁTICA
- 32 KB ROM.
- VÍDEO A 16 COLORES 128x128 DE RESOLUCIÓN O 256x256 EN B/N.
- CONEXIÓN DE HASTA 16 DISPOSITIVOS.

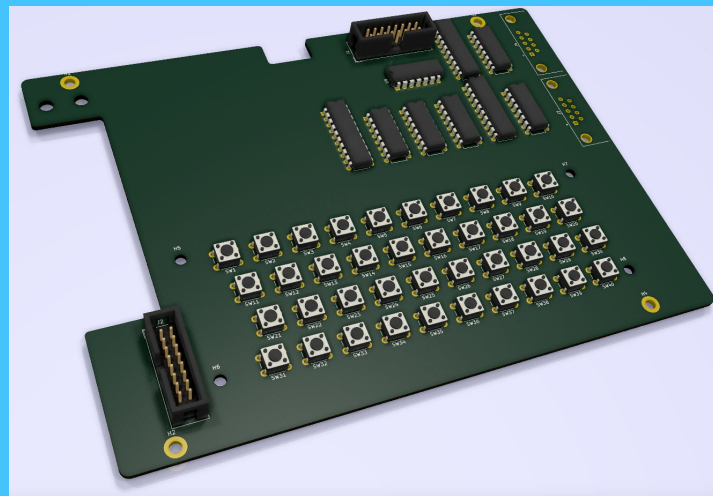




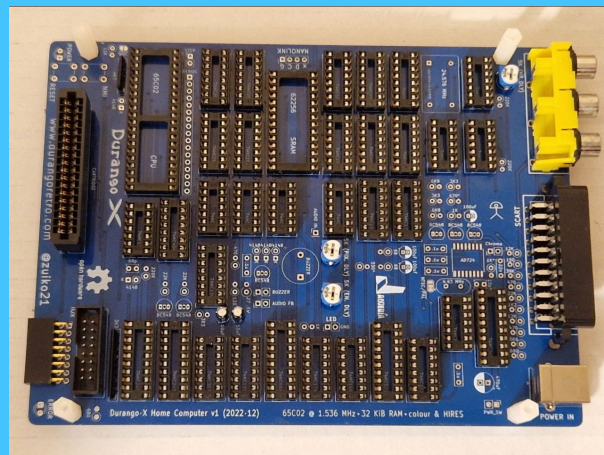
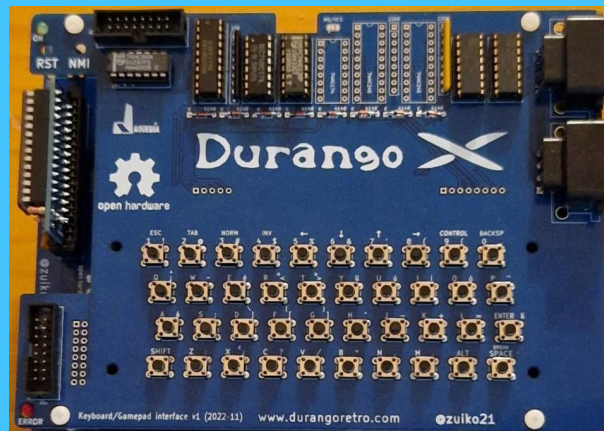
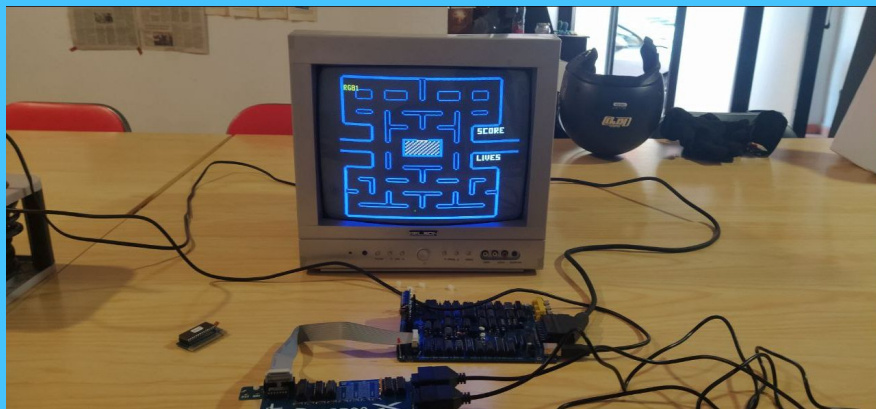
# 3. DURANGO



- TECLADO
- CONEXIÓN MANDOS NES/ MD O ATARI.
- RAM AMPLIABLE (USANDO CARTUCHO ESPECIAL).
- SONIDO MEJORADO



# 3. DURANGO



### 3. DURANGO

- ROM A TRAVÉS DE RANURA DE CARTUCHO CON MÚLTIPLES POSIBILIDADES.





# 3. DURANGO



Y LO MÁS IMPORTANTE; TODO ES HARDWARE LIBRE







# 4. SOFTWARE PARA DURANGO



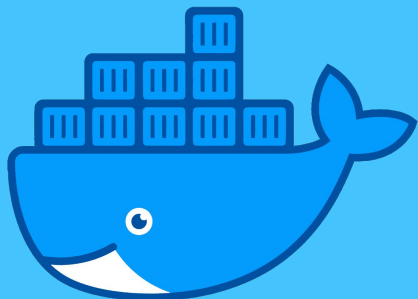
LOS CABLES Y LOS CHISES SON MU BONITOS PERO... ¿CÓMO SE CACHARREA?

- EMULADOR (PERDITA)
- HERRAMIENTAS DE DESARROLLO (DURANGO LIB)
- IMAGEN DOCKER PARA DESARROLLO

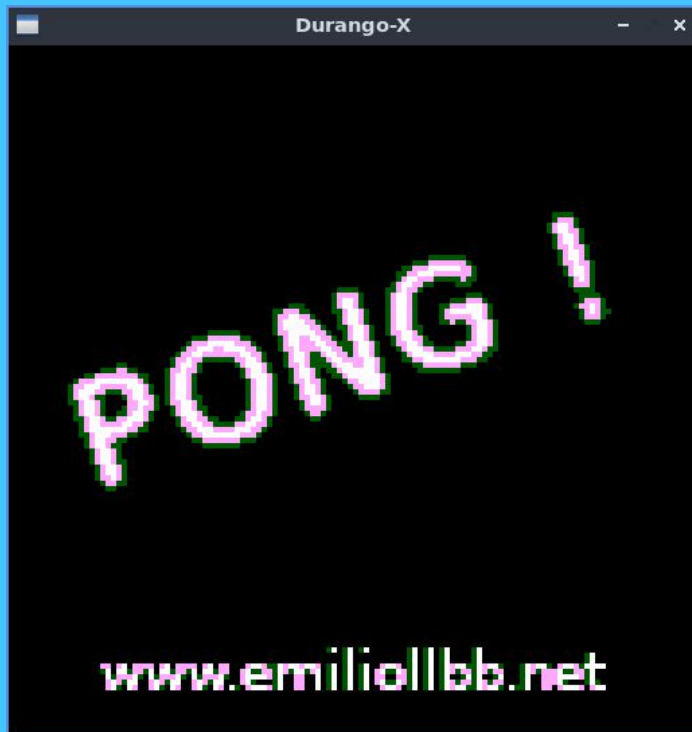




# 4. SOFTWARE PARA DURANGO



docker





# 5. SOFTWARE RETRO



PERO... A VER... ¿QUÉ TIENE QUE VER CON PYTHON?

PARA PODER VER ESTO... VEAMOS CÓMO SE PUEDE  
DESARROLLAR EN ESTE PROCESADOR...





# 5. SOFTWARE RETRO



DESARROLLAR PARA 6502:

- ASSEMBLER 6502
- BASIC
- C
- ¿PYTHON?





# 5. SOFTWARE RETRO

ENSAMBLADOR??

NOOOOOOOOOOO...MEJOR

USAR JAVA CON

ECLIPSE....

```
.export _drawFullScreen
.export _drawPixel
.export _drawRect
.export _drawFillRect
.export _drawLine
.export _drawCircle

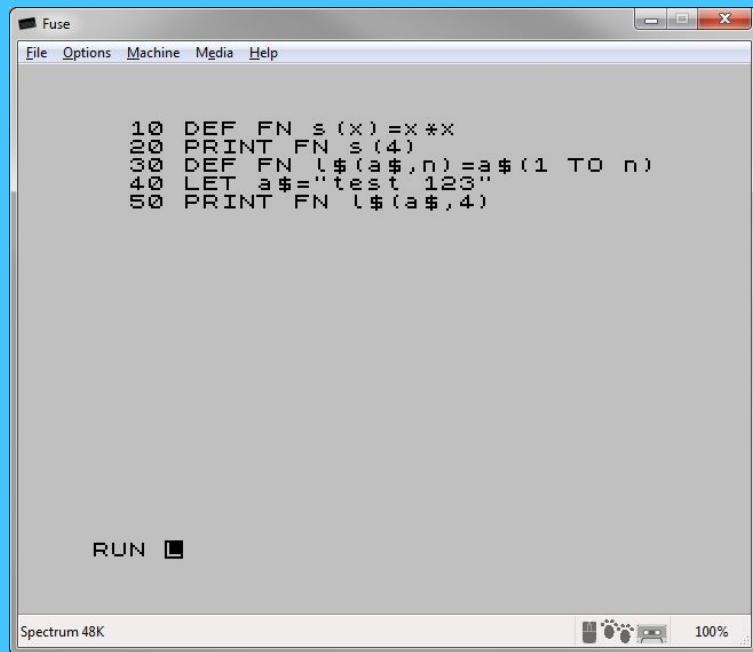
.proc _drawFullScreen: near
    LDX #>SCREEN_3
    STX VMEM_POINTER+1
    LDY #<SCREEN_3
    STY VMEM_POINTER
loop:
    STA (VMEM_POINTER), Y
    INY
    BNE loop
        INC VMEM_POINTER+1
    BPL loop
    RTS
.endproc
```



# 5. SOFTWARE RETRO

BASIC

EL PRIMER LENGUAJE DE  
PROGRAMACIÓN PARA  
MUCHOS...



```
10 DEF FN s(x)=x*x
20 PRINT FN s(4)
30 DEF FN l$(a$,n)=a$(1 TO n)
40 LET a$="test 123"
50 PRINT FN l$(a$,4)
```

RUN


Spectrum 48K 100%



# 5. SOFTWARE RETRO

C


POR SUERTE PODEMOS...  
SUBIR LA ABSTRACCIÓN



```
#include <durango.h>

int main(){
    while(1) {
        // Fill screen
        drawFullScreen(GREEN);
        // Wait 10 frames
        waitFrames(50);
        // Fill screen
        drawFullScreen(YELLOW);
        // Wait 10 frames
        waitFrames(30);
        // Fill screen
        drawFullScreen(RED);
        // Wait 10 frames
        waitFrames(50);
    }

    return 0;
}
```



# 5. SOFTWARE RETRO

¿PYTHON?

PERO... ¿NO ESTAMOS  
EN LA PYCONES? DONDE  
ESTA EL PYTHON

```
Durango-X - nanoPython POC
@zuiko21
>>> print 5*3+12
27
=>>> a=4
>>> b=3*7
>>> c=a+b
>>> d=255-c+b
>>> print a+b+c+d
45
>>> print a
4
>>> print b
21
>>> print c
25
>>> print d
251
>>> quit()

Thanks for using nanoPython on
the 65C02-powered Durango-X!
```



# 6. PYTHON EN 6502





## 6. PYTHON EN 6502



- NO HAY UNA “IMPLEMENTACIÓN” COMO TAL PARA 6502...
- MICROPYTHON SE BASA EN LA ARQUITECTURA DE LOS MICROCONTROLADORES Y ES COMPLICADO PORTARLA A ESTA.

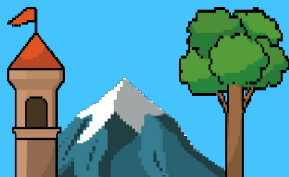




# 6. PYTHON EN 6502



¿ENTONCES?





# 7. NANOPYTHON



SE HA OPTADO POR REALIZAR UNA PRUEBA DE  
CONCEPTO...

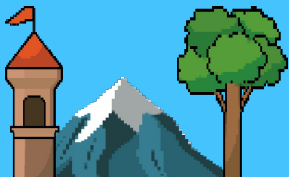




# 7. NANOPYTHON



SE HA TRATADO DE IMPLEMENTAR ALGUNOS  
ASPECTOS SENCILLOS PARA CREAR UN INTERPRETE  
PARECIDO A REPL.





# 7. NANOPYTHON



IMPLEMENTADO:

- INTÉRPRETE DE EXPRESIONES
- PRINT (UNA VERSIÓN AUN PRIMITIVA)
- QUIT()





# 7. NANOPYTHON



SE HA IMPLEMENTADO ESTA PRIMERA VERSIÓN EN  
ENSAMBLADOR PERO SE ESTÁ REVISANDO EN  
PORTARLO A OTROS LENGUAJES O HERRAMIENTAS.





# 7. NANOPYTHON



SE HA IMPLEMENTADO ESTA PRIMERA VERSIÓN EN  
ENSAMBLADOR PERO SE ESTÁ REVISANDO EN  
PORTARLO A OTROS LENGUAJES O HERRAMIENTAS.







## 8. REFERENCIAS



- [HTTPS://DURANGORETRO.COM](https://durangoretro.com)
- [HTTP://6502.ORG/](http://6502.org/)
- [HTTPS://ES.WIKIPEDIA.ORG/WIKI/MOS\\_6502](https://es.wikipedia.org/wiki/MOS_6502)





# 9. BIBA JABA!!

POR FAVOR... QUE SE ACERQUE UN REPRESENTANTE  
DE PYTHON CANARIAS...

