

# ZX BASIC

Programar nunca fue tan sencillo

<http://zxbasic.net>

@boriel



boriel  
ZX  
BASIC

A diagonal rainbow stripe runs from the bottom right corner towards the center of the logo.

¡Jajaja! 😂 ...por aquí 🙄 🖕

¡Puede ser realmente complejo!

- Programa compilado (código máquina)
- Tipos de datos
- Lenguaje BASIC

# ¿Pero qué es ZX BASIC? 🤔

- Es un *compilador cruzado* (lo usas en tu PC)
- Es software libre (...y gratis)
- Acepta un BASIC muy similar al del ZX Spectrum
  - Hay limitaciones (VAL, INPUT...)
- Acepta un BASIC más moderno, con funciones, etc.
- Traduce a código máquina (superveloaz 🚀 )

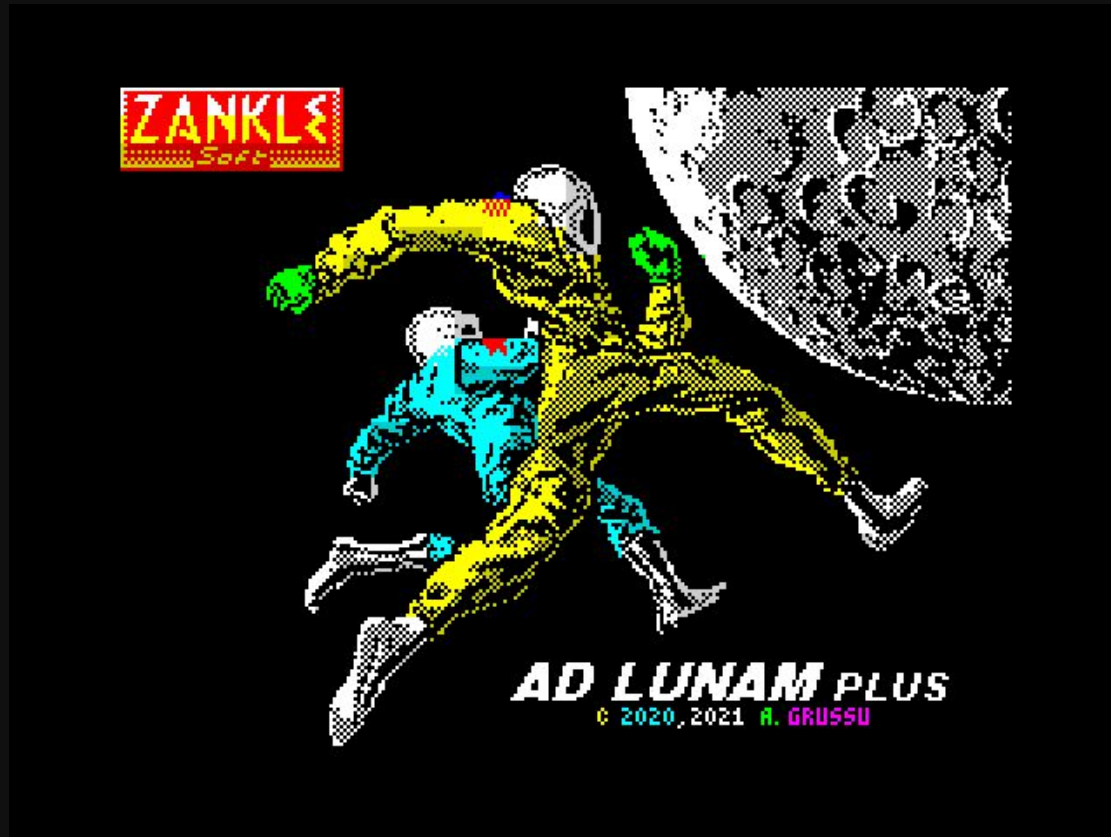
boriel  
**ZX**  
**BASIC**



# Algunas características

- Es un *compilador cruzado* (lo usas en tu PC)
- Es software libre (...y gratis)
- Acepta un BASIC muy similar al del ZX Spectrum
  - Hay limitaciones (VAL, INPUT...)
- Además de un BASIC más moderno, con funciones, etc. Ensamblador *inline*.
- Traduce a código máquina (superveloaz 🚀 )

# Juegos



# Juegos



boriel  
**ZX**  
**BASIC**

# Juegos



boriel  
**ZX**  
BASIC

# Juegos





# Juegos



boriel  
ZX  
BASIC

# Juegos



boriel  
**ZX**  
BASIC

# Juegos



boriel  
**ZX**  
BASIC

# ZX Basic en una peli!



ELIGE O MUERE

boriel  
**ZX**  
BASIC

# ZX Basic en una peli!

## UNIT MEDICS

ELTON ALEXANDER · JULIA WILLIAMS · SARAH BOSWORTH  
ROB BROADWAY · SIMON O'BRIEN · MARK CHARLESWORTH

CURS>R GAME DESIGN & GRAPHICS JONATHAN HOWELLS

ZX SPECTRUM CONSULTANT JOSE RODRIGUEZ 

GAME CHARACTER 3D ARTIST ADRIÁN PUJOL

NUKE COMPOSITOR MAHMOUD JABER

GHOSTER CREATOR THOM BURGESS

GHOSTER ARTWORK DAVID ROMERO

SOUND SOFTWARE IZOTOPE, INC.

24 FRAME PLAYBACK TECHNICIAN TED MITCHELL

24 FRAME PLAYBACK SOFTWARE CHRIS GIBBONS

24 FRAME PLAYBACK SUPERVISOR

boriel  
**ZX**  
**BASIC**



# ZX Basic en la universidad!

The screenshot shows a web browser window displaying a course page from the University of La Laguna. The page is titled 'Doctorado y posgrado 21/22' and features a sidebar with navigation options like 'Participantes', 'Insignias', and 'Calificaciones'. The main content area is titled 'Máquinas de 8 bits - ZX Spectrum - Boriel Basic' and lists various resources and activities, including a website, an interview, a Telegram channel, and a compiler file. It also includes sections for 'Práctica en grupo' and 'Ejemplos' with links to 'D1', 'D6', 'Vaus', and 'TAP de Vaus'.

Curso: Aprendizaje y Enseñanza

https://campusdoctoradoyposgrado2122.ulles/course/view.php?id=2122110473

Universidad de La Laguna

Servicios telemáticos | Contacto

Doctorado y posgrado 21/22

Entornos virtuales | Soporte UDV | Francisco Javier Martínez García

2122125771099

Participantes

Insignias

Calificaciones

General

OTROS RECURSOS

Tema 1. La Formación Profesional.

EJERCICIOS Y TAREAS

Tema 2. Modelos de Enseñanza.

Tema 3. La Evaluación

Segundo cuatrimestre (módulo David Abreu)

Introducción a POV-Ray

Introducción a la enseñanza de la programación

Máquinas de 8 bits - ZX Spectrum - Boriel Basic

Currículo del ciclo formativo de Grado Superior de Técnico Superior en Desarrollo de Aplicaciones Multiplataforma.

Ficha modelo de las actividades en las unidades de trabajo

Tabla plantilla para la definición de las unidades didácticas y las sesiones dentro de estas (ESO y Bachillerato)

Unidad de Trabajo

### Máquinas de 8 bits - ZX Spectrum - Boriel Basic

- Página web de Boriel Basic
- Entrevista a Boriel
- Canal de Telegram de Boriel Basic
- Archivo .bat para compilar y ejecutar (FUSE)

**Práctica en grupo**

- Esqueleto para hacer la práctica en grupo
- Entrega de la práctica en grupo

**Ejemplos**

- D1
- D6
- Vaus
- TAP de Vaus

Universidad de La Laguna  
Pabellón de Gobierno, C/ Padre Herrera s/n | 38200 | Apartado Postal 456 | San Cristóbal de La Laguna | España | (+34) 922 31 90 00

boriel  
**ZX  
BASIC**

# Tipos de datos (I)

- **float** - El punto flotante de toda la vida. Lento, 5 bytes. Bastante precisión. Se usa la ROM.
- **fixed** - Punto fijo. Más rápido, 4 bytes. Decimales de poca precisión entre -32767 y 32768 aprox.
- **string** - Cadenas de caracteres.

# Tipos de datos (II)

Números enteros de 8 bits: poca memoria,

- **byte** - Entero de 8 bits con signo:  
-128...127
- **ubyte** - Entero de 8 bits sin signo:  
0 .. 255



# ¡Cuidado, trampa!


```
DIM x,y AS Ubyte
```

```
FOR x = 0 TO 255: REM Esto nunca termina  
  FOR y = 0 TO 192  
    PLOT x,y  
  NEXT y  
NEXT x
```

*"FOR termina cuando la variable supera el límite".*

# Solución

```
DIM x,y AS UInteger
```

```
FOR x = 0 TO 255: REM Ahora sí   
  FOR y = 0 TO 192  
    PLOT x,y  
  NEXT y  
NEXT x
```

# Tipos de datos (III)

## Números enteros:

- `integer` - Entero de 16 bits con signo:  
-32768...32767
- `uinteger` - Entero de 16 bits sin signo:  
0 .. 65535

Cuidado (nuevamente) con los desbordamientos.

# Tipos de datos (IV)

Números enteros:

- `long` - Entero de 32 bits con signo:  
-2,147,483,647.. 2,147,483,648
- `ulong` - Entero de 32 bits sin signo:  
0 .. 4,294,967,295

Más lentos y más memoria.

# IF ... ELSE... 🤔

IF admite varias sintaxis y permite encapsular bloques

IF <cond> THEN ...

IF <cond> THEN ... : ELSE ...

La segunda forma puede ser difícil de leer:

```
IF a > 0 THEN PRINT: GOTO 10: ELSE GOTO 20
```

# IF ... ELSE... 🤔

## IF con bloque de sentencias

```
IF <cond> THEN
```

```
...
```

```
END IF
```

```
IF <cond> THEN
```

```
...
```

```
ELSE
```

```
...
```

```
END IF
```

# IF ... ELSE...



```
IF a < 5 THEN
    PRINT "SI"
    GOTO 10
ELSE
    PRINT "NO"
    GOTO 20
END IF
```

# IF ... ELSEIF ... ELSE



```
IF a = 1 THEN PRINT "HAY UNO"  
IF a = 2 THEN PRINT "HAY DOS"  
IF a = 3 THEN PRINT "HAY TRES"  
IF a > 3 THEN PRINT "HAY MUCHOS"
```

Ineficiente: se evalúan todos los IFs



# IF ... ELSEIF... ELSE



```
IF a = 1 THEN
    PRINT "HAY UNO"
ELSEIF a = 2 THEN
    PRINT "HAY DOS"
ELSEIF a = 3 THEN
    PRINT "HAY TRES"
ELSE
    PRINT "HAY MUCHOS"
END IF
```

# TABLAS DE SALTO



Cosas como:

```
900 LET a = 0
1000 LET a = a + 1
1010 GOTO 2000 + a * 100
...
2100 PRINT "IR NORTE": ... : GOTO 1000
2200 PRINT "IR SUR": ... : GOTO 1000
2300 PRINT "IR ESTE": ... : GOTO 1000
...
```

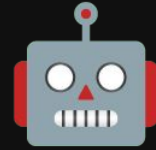
# ON ... GOTO



En ZX BASIC:

```
900 LET a = 0
1000 LET a = a + 1
1010 ON a - 1 GOTO 2100, 2200, 2300
...
2100 PRINT "IR NORTE": ... : GOTO 1000
2200 PRINT "IR SUR": ... : GOTO 1000
2300 PRINT "IR ESTE": ... : GOTO 1000
...
```

# ON ... GOSUB



También con GOSUB

```
900 LET a = 0
1000 LET a = a + 1
1010 ON a - 1 GOSUB 2100, 2200, 2300
1020 GOTO 1000
2100 PRINT "IR NORTE": ... : RETURN
2200 PRINT "IR SUR": ... : RETURN
2300 PRINT "IR ESTE": ... : RETURN
```

...

# ARRAYS

Copiar arrays de forma "tradicional":

```
DIM a(10, 20) AS byte
```

```
DIM b(10, 20) AS byte
```

```
FOR i = 1 TO 10
```

```
  FOR j = 1 TO 20
```

```
    LET a(i, j) = b(i, j): REM 🐢 😴
```

```
  NEXT j
```

```
NEXT i
```

# ARRAYS

Copiar arrays de forma eficiente:

```
DIM a(10, 20) AS byte
```

```
DIM b(10, 20) AS byte
```

...

```
a = b : REM copia b en a 
```

# FUTURAS MEJORAS (I)

- Modulos (librerías)

```
MODULE scroll
PUBLIC PROCEDURE up(pixels AS ubyte)
...
END PROCEDURE
...
END MODULE
-
IMPORT scroll
scroll.up(8)
```

# FUTURAS MEJORAS (II)

- Structs (clases)

```
TYPE Persona
  nombre AS string
  edad AS uinteger
  email AS string
END TYPE

DIM usuario AS Persona
```



# FUTURAS MEJORAS (III)

- Evaluación en circuito corto (short circuit evaluation): AndAlso, OrElse
- Uso de memoria extendida (bancos de memoria) => Linker
- Soporte para otras plataformas (a medio plazo, Z80 - MSX, CPC...)
- TCO - Tail Call Optimization
- Clases y herencia
- Tipos Enumerados (constantes)

# LA COMUNIDAD



- Foro
- Canal oficial en Telegram

GRACIAS



2022