

## UD1: INTRODUCCIÓN A LA PROGRAMACIÓN

3

### PROGRAMANDO EN BASIC

#### Introducción

Quizás no tenga la épica de los años 50 o 60, cuando brillantes ingenieros inventaron el transistor y el mundo se volvió digital, cuando se desarrollaron las primeras computadoras comerciales y los primeros lenguajes de programación de alto nivel.

Ni el glamour de los años 70, con la aparición de espectaculares supercomputadores vectoriales como el Cray-1, el sistema operativo UNIX o el lenguaje C.

Tampoco tiene la dimensión de los años 90, donde la explosión de Internet cambia para siempre la forma en que nos relacionamos y nos convierte en ciudadanos globales.

Aún así, los años 80 fueron un momento mágico en la historia de la informática. Y no por los grandes desarrollos, descubrimientos o avances tecnológicos que cambian el mundo. Fueron mágicos porque, por primera vez, en esa todavía incipiente historia de la informática, las computadoras personales entraron en los hogares de millones de personas en todo el mundo, en las vidas de millones de niños y niñas a quienes se abrió un mundo completamente nuevo para explorar: la programación.

De la noche a la mañana, pequeños ordenadores basados en microprocesadores de 8 bits, como el Zilog Z80 o el MOS 6510, pasaron a formar parte del panorama doméstico, al igual que décadas antes los aparatos de radio o televisión.

El Commodore Vic-20 y el Commodore 64, que con más de 12 millones de unidades fue el tercer ordenador personal más vendido de la historia tras el IBM-PC y el Apple Macintosh (hoy superado por la Raspberry Pi), los ZX81 y Spectrum de Sinclair, los CPC 464 o 6128 de Amstrad, los MSX de las compañías japonesas Canon, Sony o Toshiba, y un larguísimo etcétera, abarrotaban a los expositores en las secciones de “Imagen y Sonido” de los centros comerciales, conectados a los ya extintos televisores de tubo de rayos catódicos, aprovechando al máximo sus capacidades gráficas y sonoras, a imagen y semejanza de lo que ocurre hoy en día con los portátiles y los smartphones.

Esta práctica sirve para rendirles un pequeño y modesto homenaje.

#### Dos líneas cambiaron el mundo

```
10 PRINT "HOLA, MUNDO! ";  
20 GO TO 10
```

Desde que Brian Kernighan escribió el primer programa "hola mundo" como parte de la documentación del lenguaje BCPL de Martin Richards, millones de programadores en todo el

mundo han aprendido a aprender lenguajes de programación con ese mensaje icónico. Y con los ordenadores personales de los años 80 no fue diferente.

A diferencia de lo que ocurre hoy en día, cuando se iniciaba un ordenador de aquella época, no había entornos gráficos y multitud de herramientas ejecutándose en nuestro sistema operativo multitarea. Estos ordenadores, al carecer de un sistema operativo tal y como lo entendemos actualmente, sólo contaban con una pequeña memoria ROM que incluía un intérprete BASIC que se cargaba al iniciar el ordenador. Entonces, después de presionar el botón de encendido, todo lo que encontró fue una pantalla en blanco y un breve "OK" o "READY", que le advertía que la computadora estaba lista para aceptar comandos en BASIC.

Y entonces, tus dedos volaron sobre el teclado y, en medio de una vorágine creativa, líneas de código hacían brotar de la pantalla una explosión de formas, colores y sonidos... mmmm... Bueno, no tanto... En realidad empezaste escribiendo ese "hola mundo", ese programita que habías visto escribir antes a algún compañero, o que alguien te había contado en secreto, y te quedaste cinco minutos absorto viendo cómo esos caracteres rellenaban la pantalla...

Y mientras tu imaginación volaba pensando en todos esos programas que querías hacer algún día, abriste el "Manual del Programador" que acompañaba a tu ZX Spectrum o Commodore 64 hasta la primera página y empezaste a leer...

Esta es sin duda la diferencia más relevante entre la informática personal de aquella época y la actual. Equipos infinitamente menos capaces que cualquier ordenador moderno pero en los que la iniciación fue desde la perspectiva de la programación y no la de meros consumidores de tecnología. Es particularmente sintomático cómo personas de esa generación como Eben Upton, ante la desconexión con los "fundamentos" de la informática que presentaban los nuevos estudiantes de las escuelas de ingeniería informática, idease un proyecto como Raspberry Pi, para que las nuevas generaciones de niños se iniciasen en el mundo de la informática de una forma similar a lo que ocurrió en los años 80. Con equipos sencillos y económicos, y con una clara orientación hacia la programación y la electrónica. No es casualidad, por tanto, que hoy sea una de las plataformas estrella de las comunidades Maker y del fenómeno DIY (Do-It-Yourself).

## BASIC

El pequeño programa de la sección anterior ilustra varias de las características del lenguaje BASIC. La primera, y fundamental, es la sencillez del lenguaje. Diseñado para que estudiantes de carreras no tecnológicas puedan iniciarse en la programación informática, se caracteriza por una rápida curva de aprendizaje.

Otra de sus características son los números de línea. Cada declaración en BASIC tiene un número de línea, lo que facilitó la entrada y edición de programas desde los antiguos editores de línea. Además, era el mecanismo utilizado para cambiar el flujo del programa a través de la tan denostada instrucción "GO TO", otra de las características del lenguaje. Aunque existen cientos de dialectos de BASIC, en origen era un lenguaje no estructurado. El uso de la instrucción "GO TO" para controlar el flujo del programa complicó enormemente el desarrollo y mantenimiento de grandes aplicaciones.

Aun así, su sencillez y pequeño tamaño facilitaron su expansión de la mano de los microcomputadores de los años 80.

En esta tarea vamos a abordar la programación BASIC en uno de esos ordenadores “legendarios” de los años 80: el británico Sinclair ZX Spectrum. Para ello utilizaremos el simulador online ZX Spectrum.

Nuestro primer "experimento" será presentar y ejecutar nuestro programa "Hola mundo".

#### Algunas consideraciones:

- En el ZX Spectrum, **los comandos se escribían haciendo clic en la tecla correspondiente, que estaba superpuesta a ellos**. Esto permitiría un uso más eficiente del almacenamiento de programas, pues se grababan los códigos de control asociados a la pulsación y no el comando en sí. El Anexo I muestra la distribución del teclado y el acceso a los comandos en el simulador desde un teclado estándar.
- Una vez introducido el programa **lo ejecutaremos con el comando RUN**. Para detenerlo usaremos la tecla BARRA ESPACIADORA.
- Para eliminar una línea, escribimos su número y pulsamos RETORNO
- Para editar una línea podemos escribirla nuevamente o posicionarnos con el cursor > (ARRIBA, ABAJO) y presionar ESCAPE para recuperarla y editarla.

#### ZX Spectrum (<http://torinak.com/qaop>)

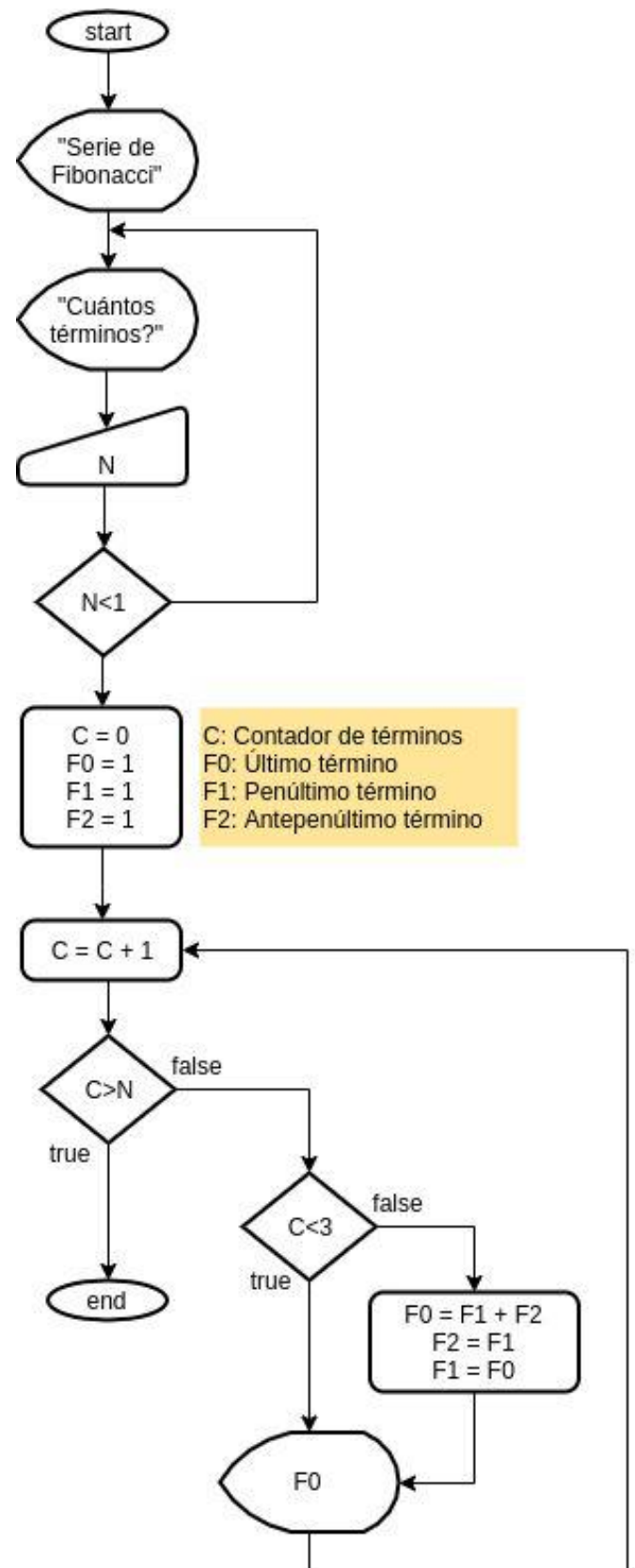
- o Basado en el microprocesador Zilog Z80A de 3,5 MHZ. Bus de datos de 8 bits y direcciones de 16 bits (podría direccionar hasta 64 kB)
- o RAM de 16 kB o 48 kB
- o ROM de 16 kB para intérprete Sinclair BASIC, juego de caracteres ASCII y memoria de vídeo
- o Resolución de 256x192 píxeles 16 colores
- o El almacenamiento se realizó en cintas de casete de audio convencionales.
- o Contaba con una ranura de expansión para la interconexión de periféricos
- o Los comandos se escribían haciendo clic en la tecla correspondiente, que los tenía superpuestos (Anexo I). Esto permitió un uso más eficiente del almacenamiento de programas, ya que se registraban los códigos de control asociados a la pulsación y no el comando en sí.

#### Ejemplo

El siguiente programa implementa un algoritmo para obtener términos de la serie de Fibonacci. Esta serie está formada por una secuencia de números naturales donde cada término (a partir del tercero) es el resultado de la suma de los dos términos anteriores: 1, 1, 2, 3, 5, 8, 13, 21, 34, ..

A continuación se muestra el pseudocódigo y diagrama de flujo de dicho programa:

Inicio  
Mostrar "Serie de Fibonacci"  
*loop1* :  
Mostrar "Cuántos términos?"  
Leer N  
Si  $N < 1$   
    Ir a *loop1*  
 $C \leftarrow 0$   
 $F0 \leftarrow 1$   
 $F1 \leftarrow 1$   
 $F2 \leftarrow 1$   
*loop2*:  
 $C \leftarrow C + 1$   
Si  $C < N$  Hacer  
    Si  $C > 2$  Hacer  
         $F0 \leftarrow F1 + F2$   
         $F2 \leftarrow F1$   
         $F1 \leftarrow F0$   
    Mostrar F0  
    Ir a *loop2*  
Fin



## Ejercicios:

1. Introducir y ejecutar un programa "Hola mundo".
2. Traducir el pseudocódigo anterior a instrucciones BASIC y probar en el simulador.
3. Intenta hacer un programa que resuelva ecuaciones de segundo grado:

$$\text{RECUERDA: } ax^2 + bx + c = 0 \rightarrow x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

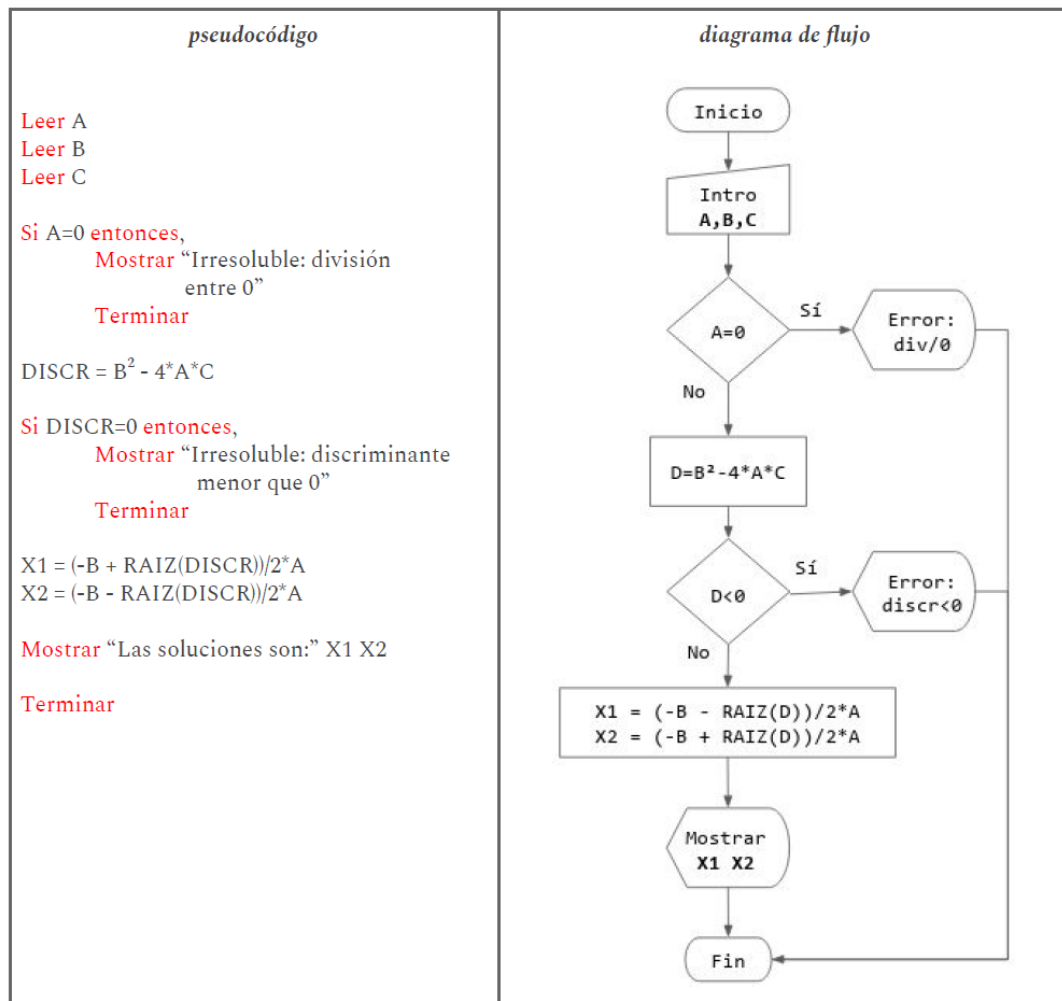
## Anexo I

### 1. AYUDA PARA RESOLVER LOS PROBLEMAS:

Comandos BASIC para la serie Fibonacci:

ZX Spectrum
<pre>10 PRINT "Serie de Fibonacci" 20 INPUT "Cuantos terminos? "; n 30 IF n&lt;1 THEN GO TO 20 40 LET c=0 50 LET f0=1 60 LET f1=1 70 LET f2=1 80 LET c=c+1 90 IF n&lt;c THEN GO TO 200 100 IF c&lt;3 THEN GO TO 140 110 LET f0=f1+f2 120 LET f2=f1 130 LET f1=f0 140 PRINT f0;" "; 150 GO TO 80</pre>

Pseudocódigo y diagrama de flujo para resolución de ecuaciones de 2º grado:



## 2. Distribución del teclado y comandos del ZX Spectrum:

Distribución del teclado del ZX Spectrum:



Acceso a los comandos desde el emulador:

