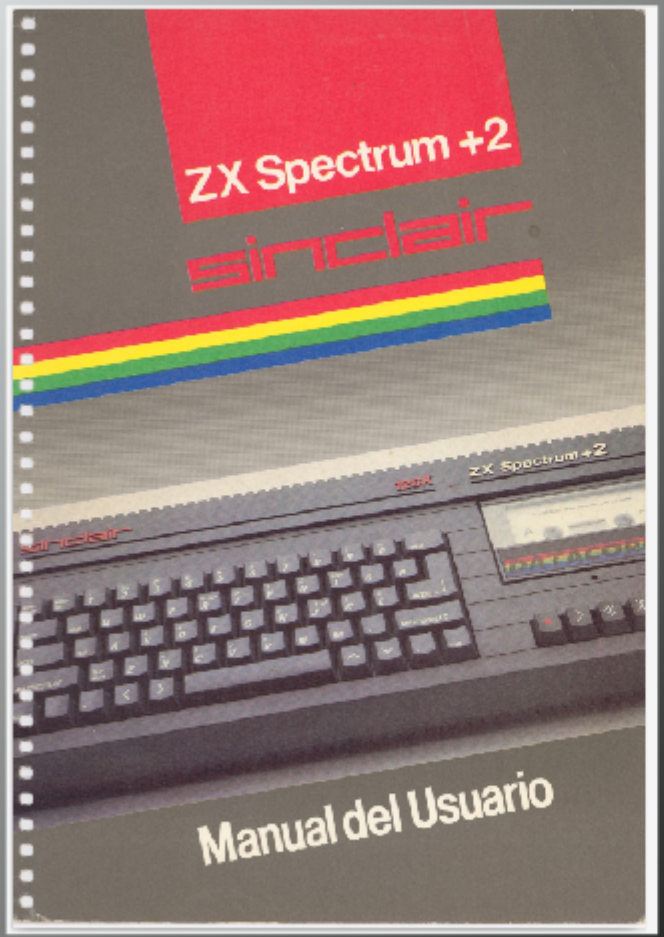


Las cosas no son tan nuevas como crees....

Los FAQ y o Knowledge base, ya estaban inventados.

¿Algún problema?

Si ha conseguido sintonizar el televisor satisfactoriamente, puede pasar a la siguiente sección.

Si no es capaz de sintonizarlo, la siguiente lista de comprobación puede ayudarle a determinar dónde reside el problema y qué remedio aplicarle.

1. Problema: **No se enciende el piloto rojo de alimentación.**

   Remedios:
   - Compruebe que la fuente de alimentación está conectada al ordenador.
   - Compruebe que la clavija de corriente alterna de la fuente de alimentación se encuentra conectada a la toma de corriente.
   - Compruebe las conexiones dentro de la toma de corriente.

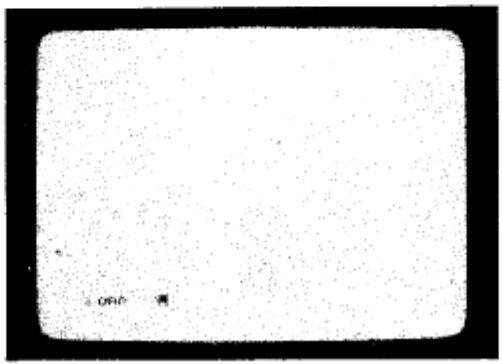
2. Problema: **El piloto de alimentación se enciende, pero no es posible sintonizar ninguna señal en el televisor.**

   Remedios:
   - Compruebe que el televisor está conectado y que funciona correctamente.
   - Compruebe que el televisor es del tipo UHF estándar (para color o para blanco y negro).
   - Compruebe que el cable de antena (suministrado con el ordenador) está bien conectado al zócalo de antena del televisor y al del ordenador.

No existia el tag <kbd> de html pero ya teniamos teclas en los manuales.

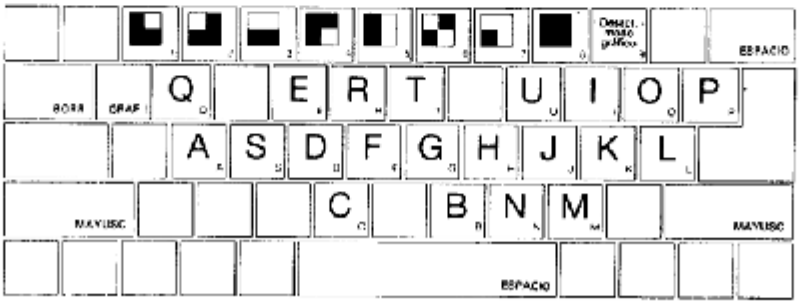
nar una opción del menú, consulte el capítulo 2.)

3. El menú de presentación desaparece y en la parte inferior de la pantalla se muestra el mensaje '©1982 Amstrad'. Ahora pulse la tecla [J] una vez y dos veces la tecla ["] (comillas). El aspecto de la pantalla debe ser el siguiente:



Teniamos caracteres gráficos para hacer videojuegos simples como ahora con los motores de videojuegos como Godot.

Al aplicar MAYUSC en modo G, se invierten los gráficos de mosaico (es decir, el color de la tinta pasa a ser el color del papel, y vice versa). En consecuencia, la siguiente pulsación será interpretada de esta manera:



El teclado con MAYUSC en el modo G

Ya teniamos herramientas de linteo de código, como el renumerar en Basic.

Las opciones del menú de edición se seleccionan de la misma manera que las del menú de presentación (utilizando las teclas del cursor y la tecla [INTRO]).

Veamos esas opciones una por una.

128 BASIC Esta opción cancela el menú de edición y restituye el cursor. Aunque no le parezca muy útil, permite volver al programa actual, sin deteriorarlo, cuando se pulsa [EDIT] accidentalmente.

Renumerar Los programas de BASIC utilizan números de línea para determinar el orden en el que las instrucciones han de ser ejecutadas. Estos números (que pueden ser cualquier entero desde el 1 al 9999) debe usted introducirlos al principio de cada línea de programa que escriba. Al seleccionar la opción

Se podia imprimir en papel el código fuente...¿Hola software libre?

en cualquier momento durante la edición), seleccione de nuevo la opción **Pantalla** del menú de edición.

Imprimir Si hay una impresora conectada, esta opción imprimirá un listado del programa actual. Cuando concluya este listado, el menú desaparecerá y volverá el cursor. Si por alguna razón el ordenador no puede imprimir (por ejemplo, porque la impresora se encuentra fuera de línea o está desconectada), pulsando dos veces la tecla BREAK se vuelve al editor.

Salida Esta opción conduce de nuevo al menú de presentación (el +2 retiene en la memoria el programa con el que se estuviera trabajando). Si desea volver al programa, seleccione la opción **128 BASIC** del menú de presen-

Ya se podían deconstruir variables como ahora en js o python.

Parte 6
Datos en los programas

Temas tratados:

READ, DATA, RESTORE

En algunos de los programas anteriores hemos visto que la información (es decir, los datos) puede ser captada por el programa mediante la sentencia INPUT. En ocasiones esto resulta muy tedioso, especialmente cuando buena parte de la información que hay que introducir es la misma en todas las ejecuciones del programa. Se puede ahorrar un tiempo considerable utilizando las sentencias READ, DATA y RESTORE. Veamos un ejemplo:

```
10 READ a,b,c
20 PRINT a,b,c
30 DATA 1,2,3
```

La programación funcional ya existia...a su manera no en plan Haskell.

Una función se define poniendo una sentencia DEF en algún lugar del programa. Por ejemplo, la siguiente sentencia define la función FN c, cuyo resultado es el cuadrado del argumento:

```
10 DEF FN c(x)=x*x: REM el cuadrado de x
```

La letra c que sigue a DEF FN es el nombre que hemos elegido para la función. La x entre paréntesis es el nombre por el cual nos referiremos al argumento de la función. Para esto se puede utilizar una letra cualquiera (pero sólo una), o bien, si el argumento

Embeber una imagen en un base64 en un html...ya existía a su manera.

Pues bien, aunque no haya conseguido seguirnos en el proceso que acabamos de explicar, pruebe el siguiente programa, que realiza la definición del carácter:

```
10 FOR n=0 to 7
20 READ fila: POKE USR "P"+n,fila
30 NEXT n
40 DATA BIN 00000000
50 DATA BIN 00000000
60 DATA BIN 00000010
70 DATA BIN 00111100
80 DATA BIN 01010100
90 DATA BIN 00010100
100 DATA BIN 00010100
110 DATA BIN 00000000
```

La sentencia POKE almacena directamente un número en una posición de memoria,

Los merges de SVN, Git, HG ya había algo parecido.

SAVE	• Almacena en la cinta el programa y las variables.
VERIFY	• Compara el programa y las variables leídos en la cinta con los de la memoria del ordenador.
LOAD	• Borra el programa y todas las variables del ordenador y los sustituye por los nuevos que ha leído en la cinta.
MERGE	• Similar a LOAD, pero no borra las líneas ni las variables del programa antiguo a menos que tenga que hacerlo (por tener el mismo número o nombre en los dos programas).

En todas estas órdenes la palabra clave va seguida por una cadena. Para la orden SA-

Ya existian "las capturas de pantalla".

la sentencia LOAD.

CODE 16384,6912 es un área de memoria (imagen de pantalla) tan útil, que BASIC dispone de una función especial, SCREEN$, para representarla. Así, para grabar y cargar la pantalla se puede usar las órdenes:

```
SAVE "imagen" SCREEN$
```

y

```
LOAD "imagen" SCREEN$
```

GNU/Linux tiene el maravilloso ramfs para crear discos duros virtuales en RAM pero el Spectrum ya tenía ese invento.

Cualquier operación que podamos hacer en la cinta con SAVE, LOAD o MERGE puede ser realizada también con el *disco de silicio* (que está incorporado al +2 y que actúa como si fuese una cinta, con un par de órdenes adicionales). La diferencia entre la cinta

y el disco de silicio consiste en que éste tiene una capacidad de almacenamiento de unos 64K y un tiempo de acceso muy pequeño; la desventaja es que el contenido del disco de silicio se pierde cuando se apaga o reinicializa el ordenador (sin embargo, «sobrevive» a la orden NEW). Todas las órdenes se utilizan exactamente igual que con el magnetófono, pero intercalando un signo de admiración ! entre la palabra clave y la cadena asociada. Así, en lugar de grabar en la cinta con

Ya existían las variables de sistema como en GNU/Linux y otros Unix.

Parte 25
Variables de sistema

Temas tratados:

POKE, PEEK

Los bytes de memoria del 23296 al 23733 están reservados para usos específicos del siste-
ma. Hay también unas cuantas rutinas (usadas para mantener en orden la paginación

Ensamblador embebido en C ...ya teniamos código máquina embebido en Basic.

reserva un espacio de 100 bytes a partir de la dirección 65268. Para introducir el programa de código de máquina, podríamos usar un programa de BASIC de este estilo:

```
10 LET a=65268
20 READ n: POKE a,n
30 LET a=a+1: GO TO 20
40 DATA 1,99,0,201
```

(Este programa se detiene y emite el mensaje E Out of DATA ('Datos agotados') cuando