

# Práctica 1

Marcos Rico Guerra

## Ejercicios obligatorios

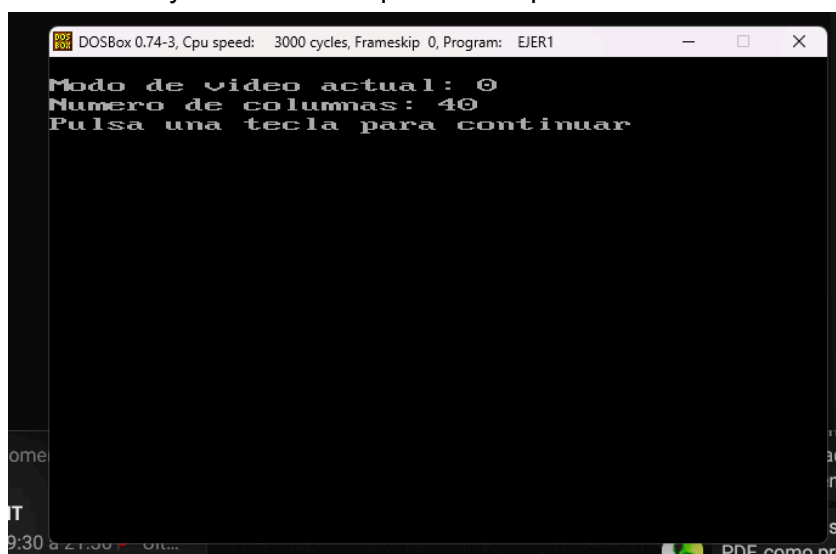
Para la parte obligatoria, se ha realizado un programa llamado ejer1.c en el que vamos llamando poco a poco a las distintas funciones que se piden en la práctica.

A continuación, una breve explicación de las funciones, en el mismo orden en el que son llamadas en el programa:

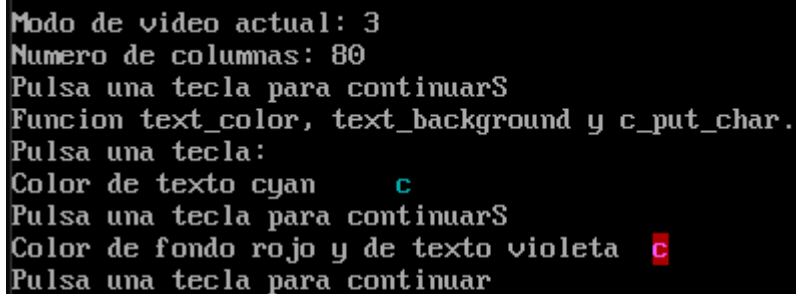
**Gotoxy(x,y):** Usamos la interrupción 10 y la función 2 para situar el cursor en la posición x,y de la pantalla(o del terminal)

**SetCursorType(x):** Usamos la función 1 de la interrupción 10 para cambiar el grosor del cursor. Permitimos 3 tipos: x=0 pone un cursor invisible, x=1 es el cursor normal(un guión bajo) y x=2 es un cursor grueso(un rectángulo que ocupa como un carácter entero)

**SetVideoMode(x) y GetVideoMode():** Usamos la interrupción 10 y las funciones 0 y F. La primera establece la forma en la que vemos la pantalla(si es modo texto o gráfico, las líneas y columnas de la pantalla...) Con la segunda simplemente obtenemos el modo de video actual y las columnas que tiene la pantalla.



TextColor(x) , TextBackground(x) y CPutChar(c): En el programa tenemos dos variables globales ctexto y cfondo. Cuando llamamos a CPutChar pone el carácter c en pantalla con color ctexto y color de fondo cfondo(Por defecto ctexto=15 blanco y cfondo=0 negro)  
Las funciones TextColor y TextBackground cambian los valores de ctexto y background respectivamente.



```
Modo de video actual: 3
Numero de columnas: 80
Pulsa una tecla para continuar
Funcion text_color, text_background y c_put_char.
Pulsa una tecla:
Color de texto cyan      c
Pulsa una tecla para continuar
Color de fondo rojo y de texto violeta  c
Pulsa una tecla para continuar
```

ClearScreen(): Borra toda la pantalla. Mientras hacía la práctica, me di cuenta que cuando se cambia el modo de vídeo la pantalla se borra entera, así que decidí aprovechar esto; por lo que la función llama primero a GetVideoMode(). Guarda el modo de vídeo actual y hace SetVideoMode con ese mismo modo de video. Es decir, estamos cambiando el modo de video al mismo que teníamos, así conseguimos borrar la pantalla sin tener que cambiar el modo de video.

Entre cada función usamos una función de pausa para poder apreciar el comportamiento de cada función. Esta está diseñada de forma que se espera una pulsación de teclado para avanzar, usando la interrupción 21h, función 1.

Usando la función de pausa también hacemos la función GetChe(), ya que esta función requiere que pulsemos el teclado y que la pulsación se refleje en pantalla, que es precisamente lo que hace la función de pausa.

## Dibujar cuadrado

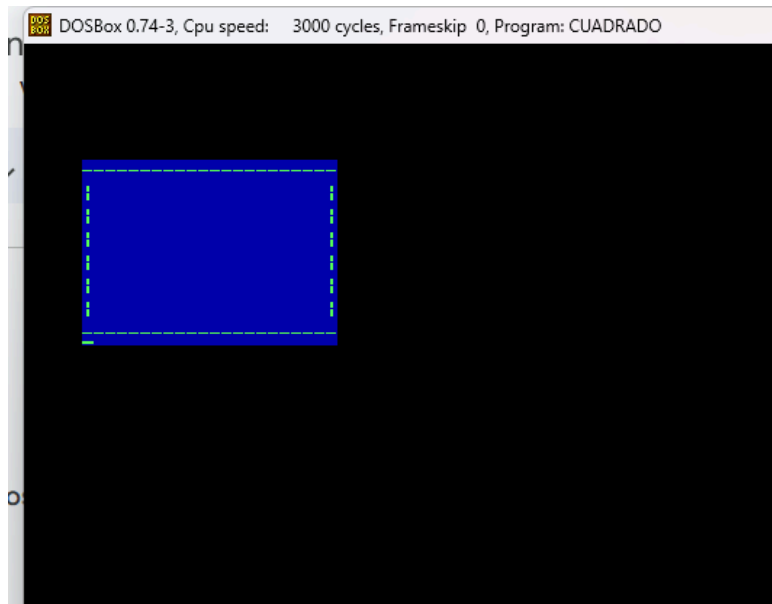
La siguiente programa es capaz de construir un rectángulo dadas 4 coordenadas, las 2 de la esquina superior izquierda y las de la esquina inferior derecha usando 'I' para las paredes verticales y '-' para las horizontales.

La función además recibe dos colores, que serán usados para cambiar el color del texto y del background usando las funciones creadas anteriormente para este fin.

Para pintar el cuadrado, simplemente usamos la función Gotoxy(x,y) para situar el cursor en la esquina superior izquierda del cuadrado. Usamos cputchar de '-' tantas veces como anchura tenga el cuadrado, posteriormente se usa un bucle donde aumentamos x en 1

hasta llegar al x establecido por las segundas coordenadas. En cada iteración, se pinta un '|' seguido de tantos espacios como anchura - 2 tenga el cuadrado, y se vuelve a pintar un '|'.

Por último, se repite lo mismo que al principio, salvo que con  $x = x$  de la segunda coordenada

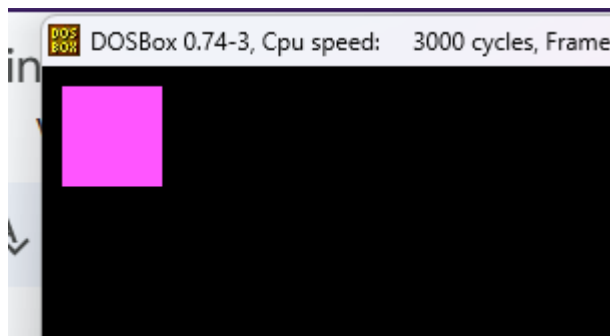


## Modo Gráfico

Este programa es parecido al anterior, solo que dibuja un cuadrado en modo gráfico en vez de en modo texto.

Primero cambiamos a modo gráfico con `SetVideoMode(mode)` indicando que `mode=4` que es un modo gráfico con 4 colores y resolución 320x200.

Usamos después la función `Pixel(x,y,c)` que usa la función `0x0C` de la interrupción 10 para pintar un pixel en la posición  $x,y$  de color  $c$ . Usamos dos bucles anidados para pintar un cuadrado de color magenta.



# Arte Ascii

Este programa está enfocado en usar distintos símbolos o letras para poder “dibujar” ciertas cosas simples en modo texto. En mi caso, el programa pinta una pequeña rana.

Para ello, primero hacemos un clear a la pantalla para que no molesten posibles textos anteriores, y posteriormente hacemos uso del GotoXY(x,y) para ir situando el cursor en distintos sitios de la pantalla, haciendo prints de las cadenas que formarán nuestra rana. Vemos el resultado final en la siguiente imagen

