

Seminario 1. Uso e instalación de un entorno para la programación de interrupciones en MS-DOS.

Ejercicios:

1. Instala el software “DosBox” en tu PC. Estudia la forma en que puedes ejecutar aplicaciones en MS-DOS como juegos clásicos.

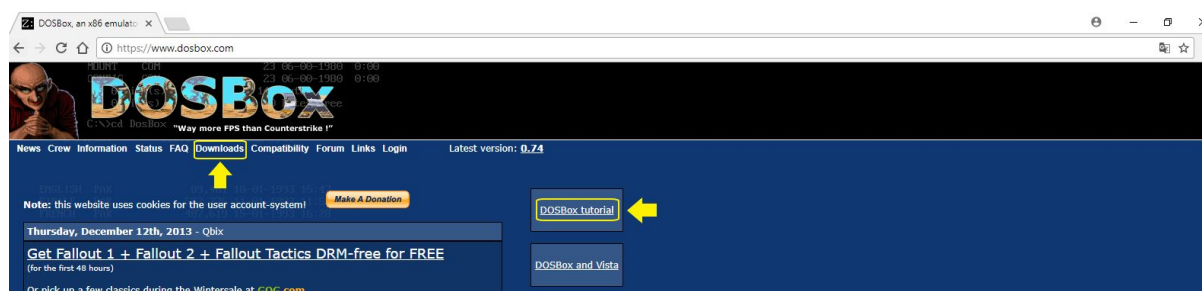
Para resolver este ejercicio vamos a la **página oficial de DosBox** (<http://www.dosbox.com>), pinchamos en la **pestaña “Downloads”** y descargamos la versión correspondiente de DosBox, dependiendo del SO que queramos usar.

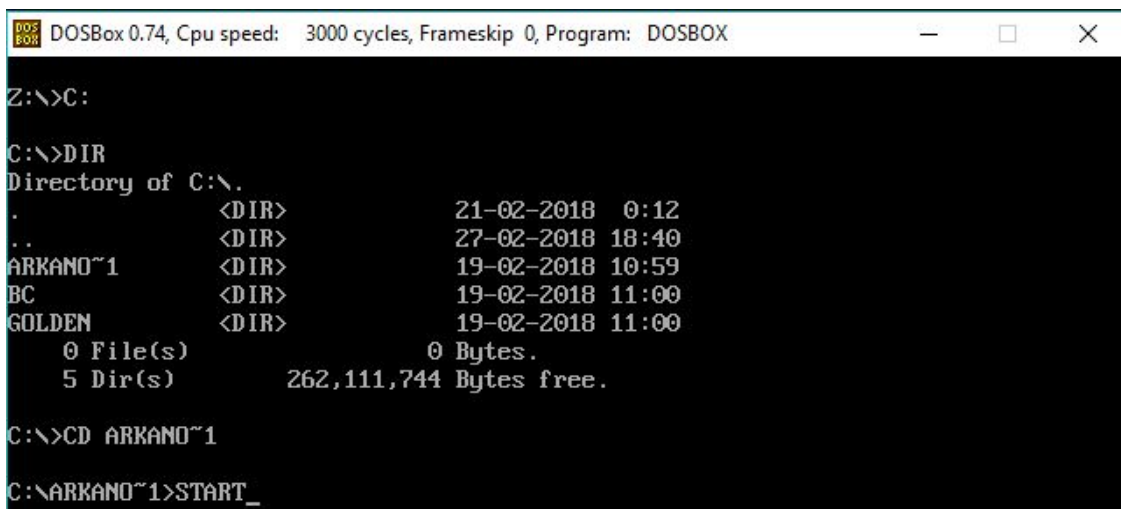
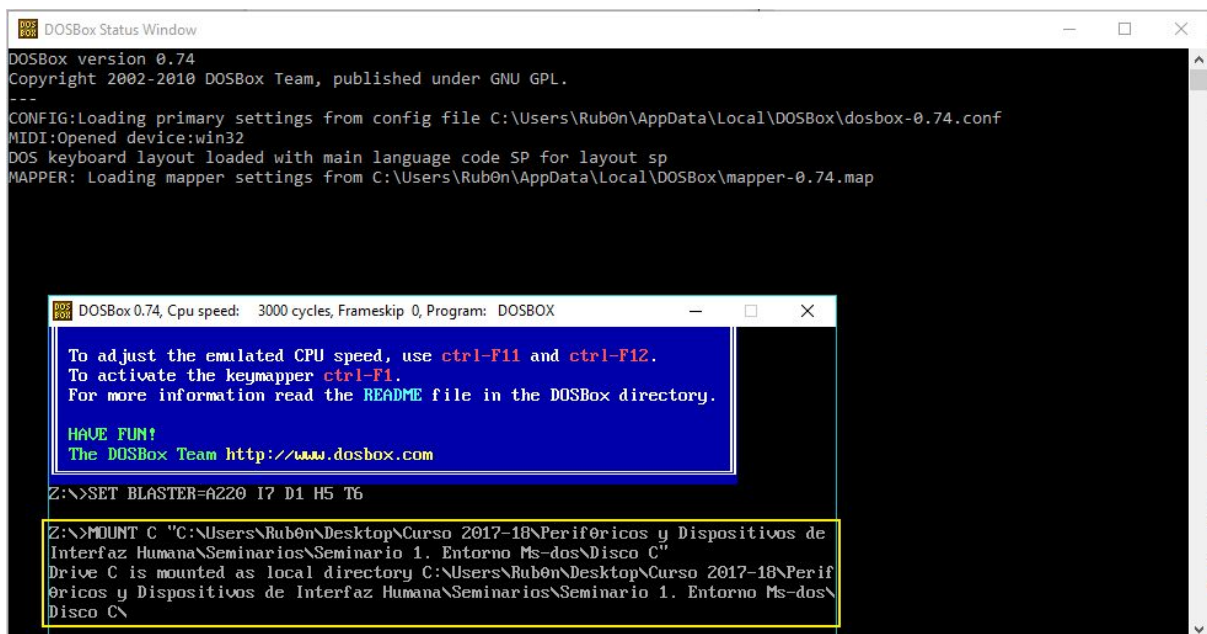
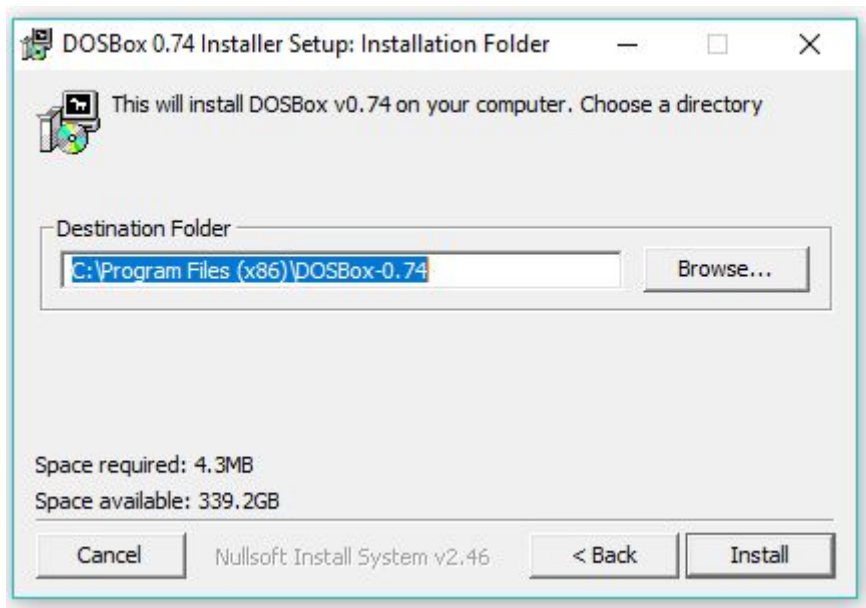
A continuación, ejecutamos el **instalador de DosBox**. Ya hemos completado la primera parte del ejercicio.

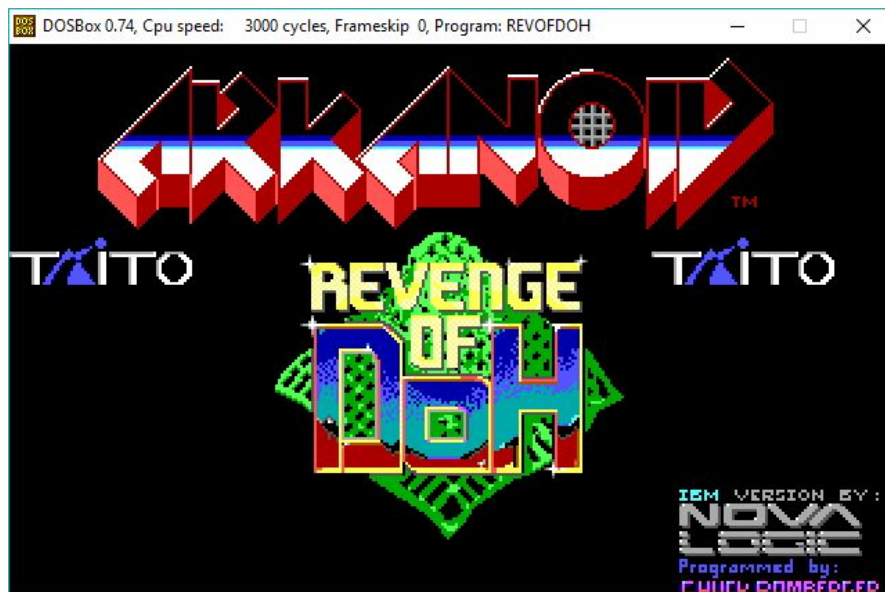
Para resolver la segunda parte, **volvemos** a la **página oficial de DosBox** y pinchamos en el enlace situado a la derecha con el texto “**DosBox tutorial**” subrayado. Una vez en la página, vemos un **apartado** que se titula “**Running a game**”. En él se explican los pasos necesarios para ejecutar un juego en DosBox. Éstos son los siguientes:

- El primer paso es **crear un directorio** donde poner todos nuestros juegos o aplicaciones que queramos ejecutar en DosBox.
- A continuación, **ejecutamos DosBox** y mediante la **orden “Mount”** montamos una nueva unidad sobre el directorio creado anteriormente. La orden “Mount” usa los siguientes parámetros:
 - MOUNT [nombre_unidad] [ruta_absoluta]
- Una vez montada la nueva unidad podemos seleccionarla **escribiendo el nombre** que le hayamos asignado. Desde ese momento estaremos en la carpeta sobre la que está montada la unidad y **podremos ejecutar los juegos o aplicaciones** que se encuentren en ella.

Vamos a ver algunas **capturas** sobre los pasos a seguir:

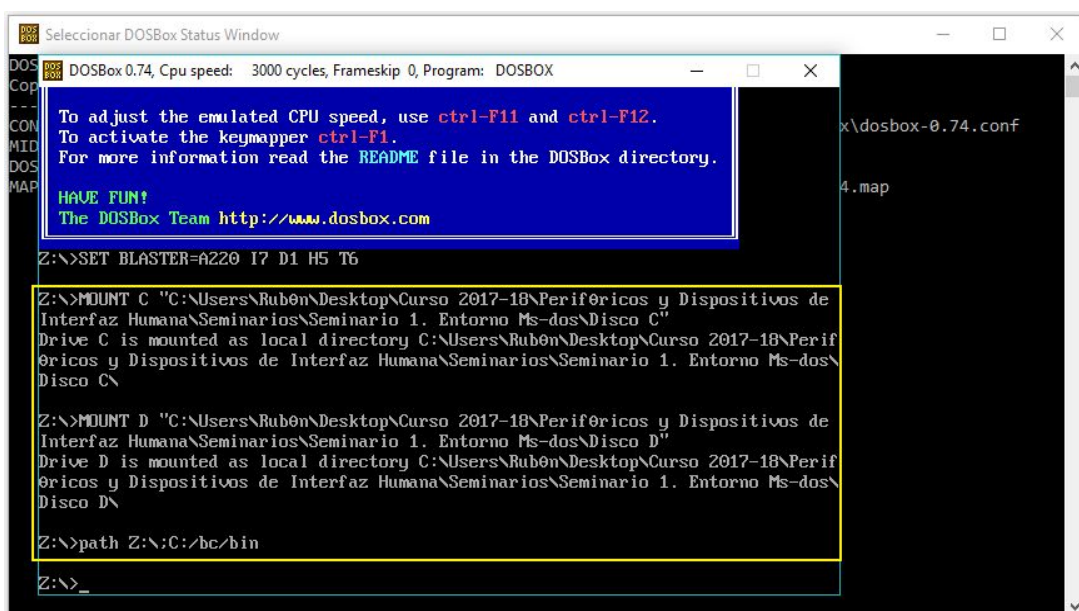
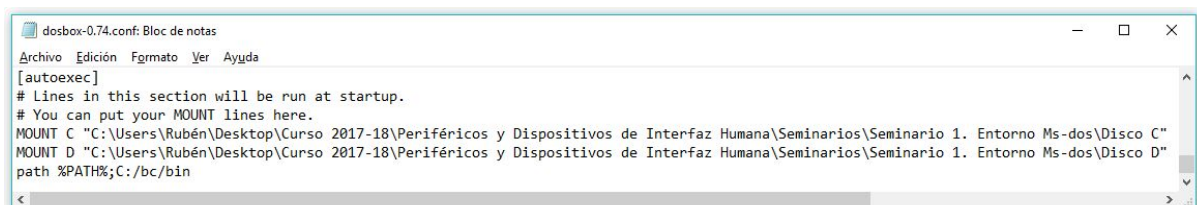






2. Configurar el inicio de DOSBox para que monte la unidad C: en el directorio donde se encuentra el entorno de programación Borland C. Añadir a la variable "PATH" de inicio el directorio "bin" donde se encuentra el ejecutable "bc.exe".

De nuevo accedemos al enlace titulado "DosBox Tutorial". En él, encontramos una sección titulada "Automatically Mounting Your DOS Games folder" que nos explica cómo podemos hacer que DOSBox monte automáticamente nuestros directorios. Lo que tenemos que hacer es modificar el archivo dosbox.conf del siguiente modo:



3. Configurar el inicio de DOSBox para que monte en la unidad C: el directorio donde se irán almacenando los subdirectorios con los proyectos en C que se vayan generando.

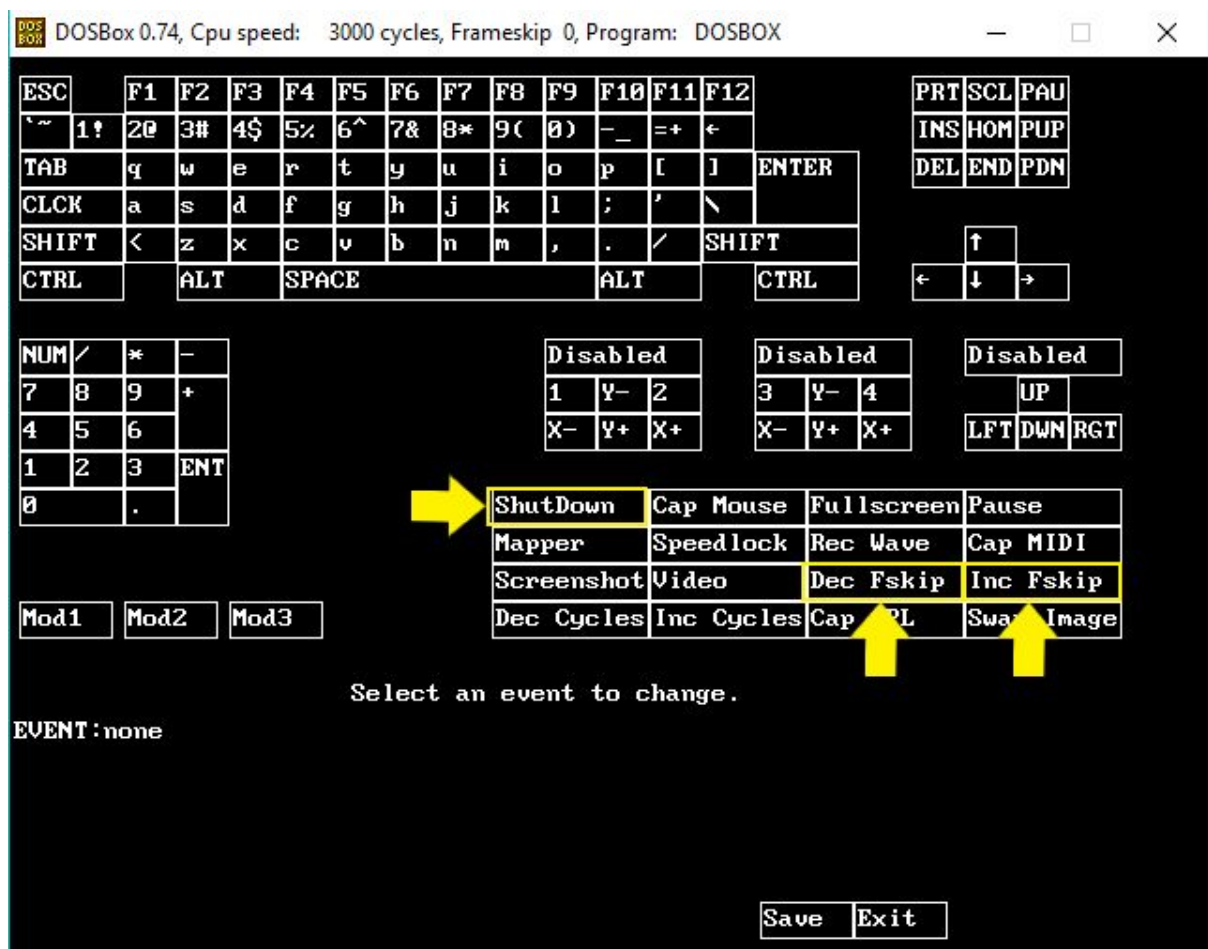
Este ejercicio lo hemos realizado junto con el anterior. En la captura de pantalla anterior se puede apreciar la modificación realizada para asignar a la unidad D la carpeta donde se guardarán los proyectos desarrollados.

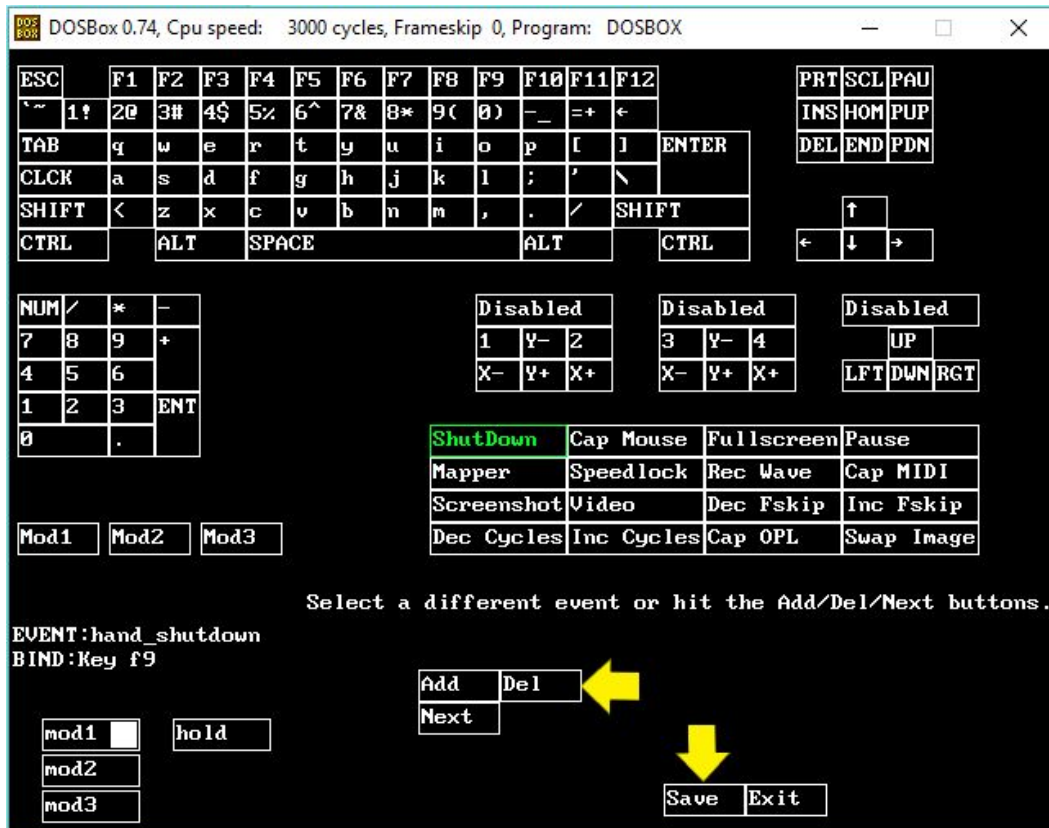
4. Configurar el “keymapper” de DOSBox para que el programa Borland C no se cierre al pulsar CTRL+F9 y para que funcionen correctamente las teclas F7 y F8 para el modo de depuración paso a paso de Borland C.

Para realizar este ejercicio volvemos a consultar la documentación de DOSBox mediante el siguiente enlace: <https://www.dosbox.com/wiki/Keymapper>.

De este modo, sabemos que para configurar el “keymapper” de DOSBox debemos de iniciar la aplicación y pulsar CTRL+F1 para ver la configuración del “keymapper”.

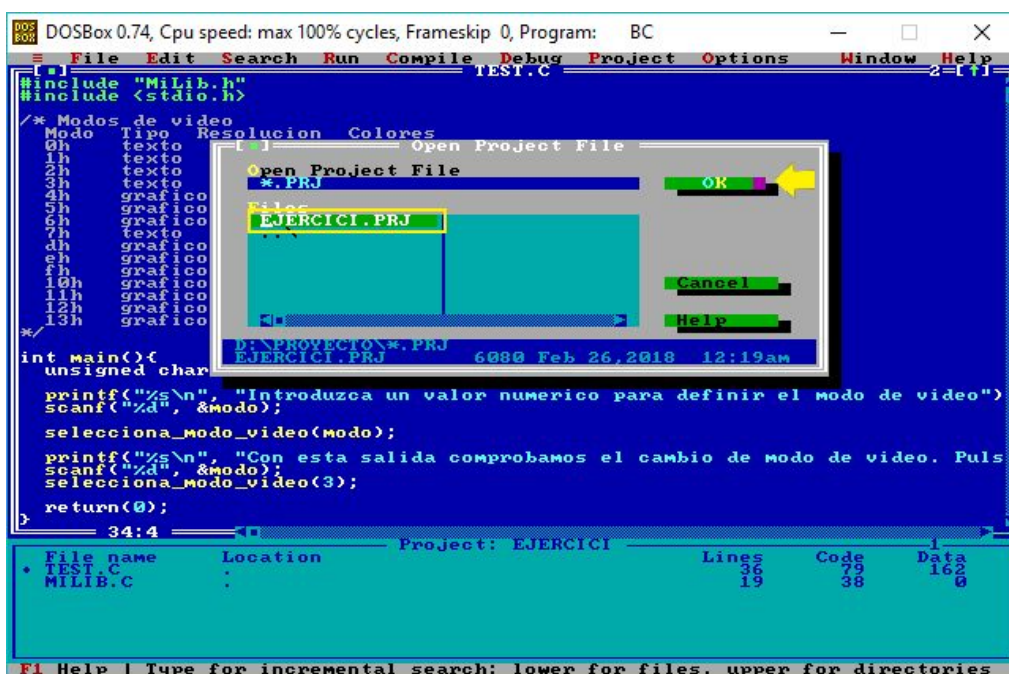
Una vez ahí, pulsamos sobre el evento “shutdown” y le decimos que ya no queremos que se active ese evento con la combinación de teclas CTRL+F9. Hacemos lo mismo con los eventos “Dec fskip” y “Inc fskip”, los cuales tienen asociados las teclas F7 y F8.





- Realizar un programa que llame a la función de cambio de modo de vídeo especificado en la página 8. Debe pedir un valor numérico al usuario y cambiar a dicho modo de vídeo. Guardar los ficheros de este ejercicio dentro de un proyecto (.PRJ).

En este caso, como se van a adjuntar los ficheros y el proyecto simplemente veremos las capturas para ver que la ejecución es correcta.



```
DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: BC
Introduzca un valor numerico para definir el modo de video
4_
```

```
DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: BC
Con esta salida comprobamos el cambio de
modo de video. Pulse cualquier tecla pa
ra salir
```

```
DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: BC
File Edit Search Run Compile Debug Project Options Window Help
#include "Milib.h"
#include <stdio.h>

/* Modos de video
Modo Tipo Resolucion Colores
0h texto 40x25 16 tonos de gris
1h texto 40x25 16 colores
2h texto 80x25 16 tonos de gris
3h texto 80x25 16 colores
4h grafico 320x200 4 colores
5h grafico 320x200 4 colores
6h grafico 640x200 2 colores
7h texto 80x25 monocromo
8h grafico 320x200 16 colores
9h grafico 640x200 16 colores
10h grafico 640x350 monocromo
11h grafico 640x480 2 colores
12h grafico 640x480 16 colores
13h grafico 320x200 256 colores
*/

int main()
{
    unsigned char modo;
    printf("%s\n", "Introduzca un valor numerico para definir el modo de video");
    scanf("%d", &modo);
    selecciona_modo_video(modo);
    printf("%s\n", "Con esta salida comprobamos el cambio de modo de video. Puls
    scanf("%d", &modo);
    selecciona_modo_video(3);
    return(0);
}

23:1
Project: EJERCICI
File name Location Lines Code Data
* TEST.C : 36 79 162
* MILIB.C : 19 38 0
F1 Help F7 Trace F8 Step F9 Make F10 Menu
```

```
DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: BC
File Edit Search Run Compile Debug Project Options Window Help
#include "Milib.h"
#include <stdio.h>

/* Modos de video
Modo Tipo Resolucion Colores
0h texto 40x25 16 tonos de gris
1h texto 40x25 16 colores
2h texto 80x25 16 tonos de gris
3h texto 80x25 16 colores
4h grafico 320x200 4 colores
5h grafico 320x200 4 colores
6h grafico 640x200 2 colores
7h texto 80x25 monocromo
8h grafico 320x200 16 colores
9h grafico 640x200 16 colores
10h grafico 640x350 monocromo
11h grafico 640x480 2 colores
12h grafico 640x480 16 colores
13h grafico 320x200 256 colores
*/

int main()
{
    unsigned char modo;
    printf("%s\n", "Introduzca un valor numerico para definir el modo de video");
    scanf("%d", &modo);
    selecciona_modo_video(modo);
    printf("%s\n", "Con esta salida comprobamos el cambio de modo de video. Puls
    scanf("%d", &modo);
    selecciona_modo_video(3);
    return(0);
}

26:1
Project: EJERCICI
File name Location Lines Code Data
* TEST.C : 36 79 162
* MILIB.C : 19 38 0
F1 Help F7 Trace F8 Step F9 Make F10 Menu
```

DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: BC

```

File Edit Search Run Compile Debug Project Options Window Help
TEST.C
#include "Milib.h"
#include <stdio.h>

/* Modos de video
Modo Tipo Resolucion Colores
0h texto 40x25 16 tonos de gris
1h texto 40x25 16 colores
2h texto 80x25 16 tonos de gris
3h texto 80x25 16 colores
4h grafico 320x200 4 colores
5h grafico 320x200 4 colores
6h grafico 640x200 2 colores
7h texto 80x25 monocromo
8h grafico 320x200 16 colores
9h grafico 640x200 16 colores
ah grafico 640x350 monocromo
bh grafico 640x350 16 colores
ch grafico 640x480 2 colores
dh grafico 640x480 16 colores
eh grafico 320x200 256 colores
*/

int main(){
    unsigned char modo;
    printf("%s\n", "Introduzca un valor numerico para definir el modo de video");
    scanf("%d", &modo);
    selecciona_modo_video(modo);
    printf("%s\n", "Con esta salida comprobamos el cambio de modo de video. Puls");
    scanf("%d", &modo);
    selecciona_modo_video(3);
    return(0);
}
35:1
Project: EJERCICI
File name Location Lines Code Data
TEST.C . 36 79 162
MILIB.C . 19 38 0
F1 Help F7 Trace F8 Step F9 Make F10 Menu

```