



MEMORIA Y DOCUMENTACIÓN

PRÁCTICA 1

RAFAEL DELGADO GARCÍA-VALDECASAS

PABLO RIENDA SÁNCHEZ

GRADO EN INGENIERÍA INFORMÁTICA

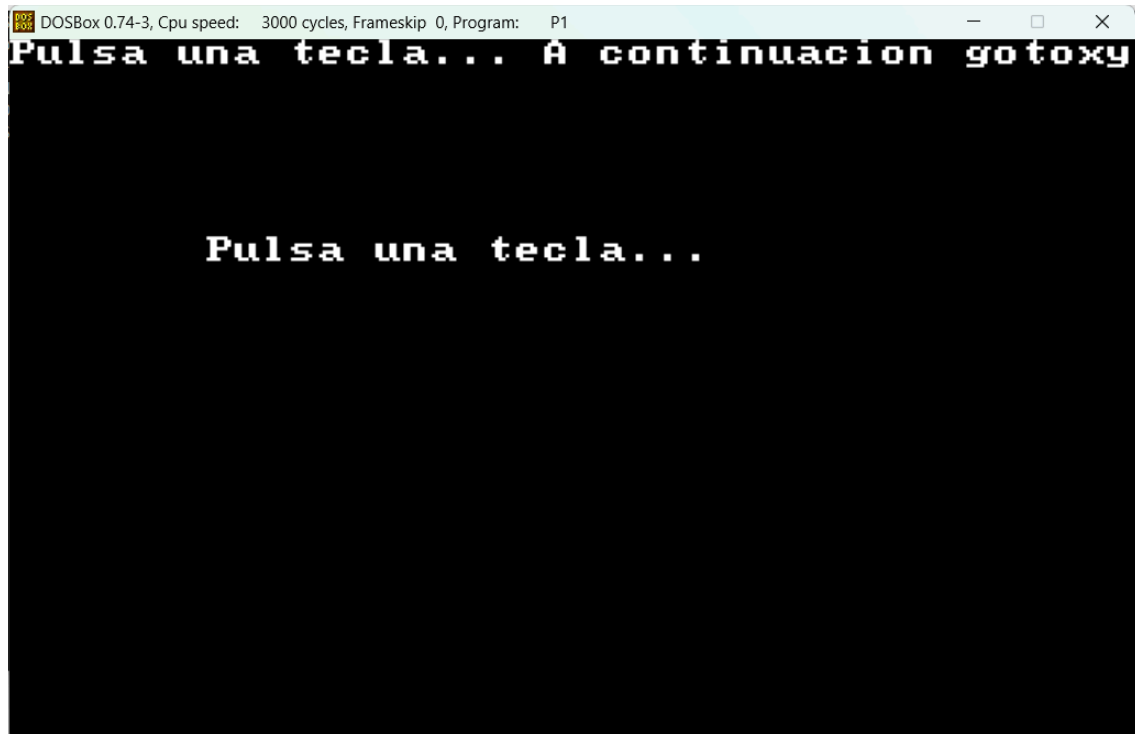
Curso 2023-2024

En esta práctica se nos pidió realizar varias funciones para trabajar en modo gráfico o texto en Dosbox. Para ellos programaremos en C.

La primera función que se nos pide consiste en colocar el cursor en una posición determinada. Se llamará gotoxy.

El código correspondiente a esta función es el siguiente:

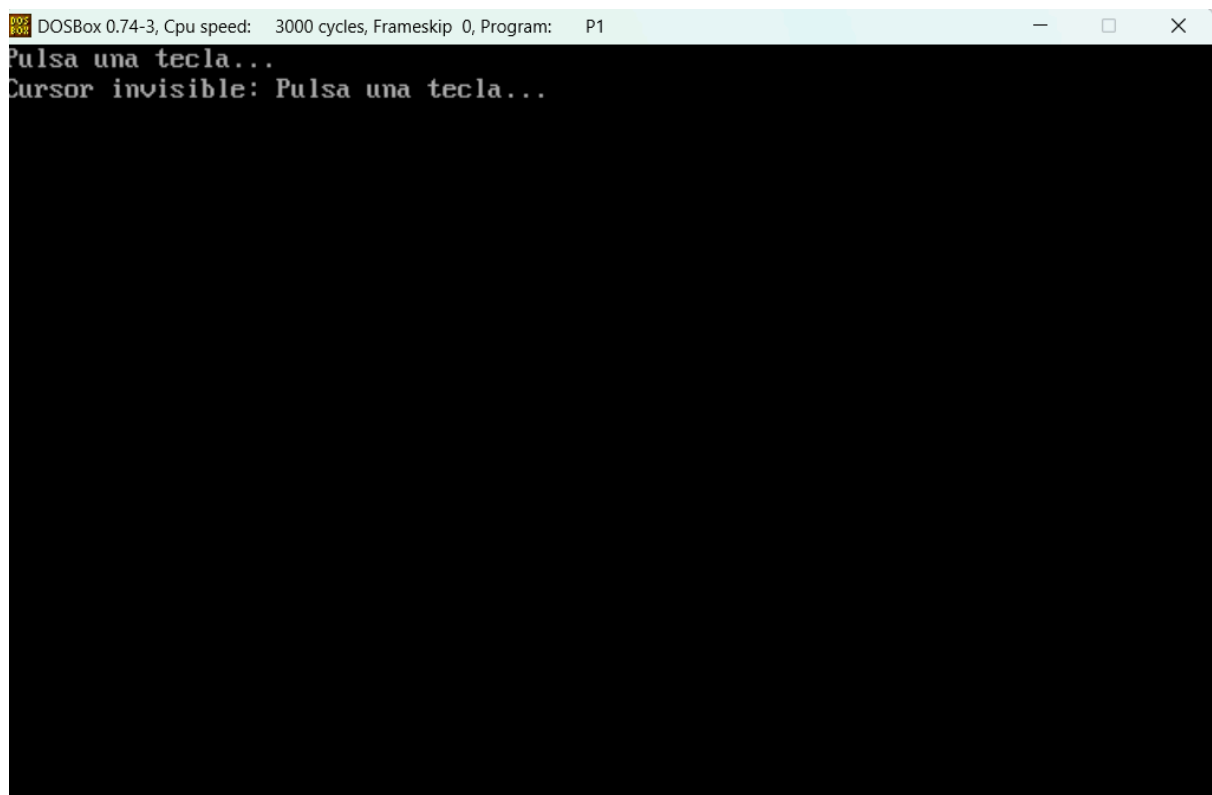
```
void gotoxy(int x, int y){  
    union REGS inregs, outregs;  
    inregs.h.ah = 0x02;  
    inregs.h.bh = 0x00;  
    inregs.h.dh = y;  
    inregs.h.dl = x;  
    int86(0x10,&inregs,&outregs);  
    return;  
}
```

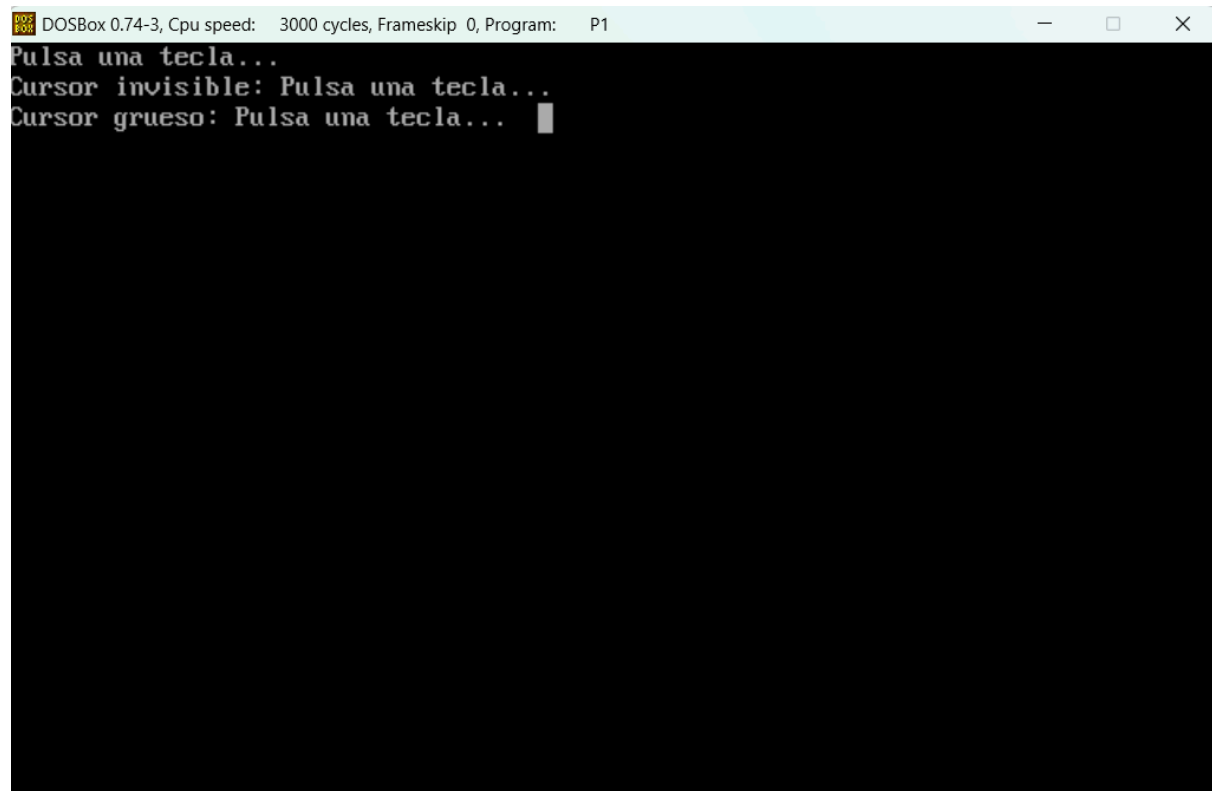


Como vemos, se desplazó el cursor a otro punto.

La siguiente función que se nos pidió solicitaba cambiar el aspecto del cursor a invisible, grueso o normal.

```
void setcursortype(int tipo_cursor){
    union REGS inregs, outregs;
    inregs.h.ah = 0x01;
    switch(tipo_cursor){
        case 0: //invisible
            inregs.h.ch = 010;
            inregs.h.cl = 000;
            break;
        case 1: //normal
            inregs.h.ch = 010;
            inregs.h.cl = 010;
            break;
        case 2: //grueso
            inregs.h.ch = 000;
            inregs.h.cl = 010;
            break;
    }
    int86(0x10, &inregs, &outregs);
}
```

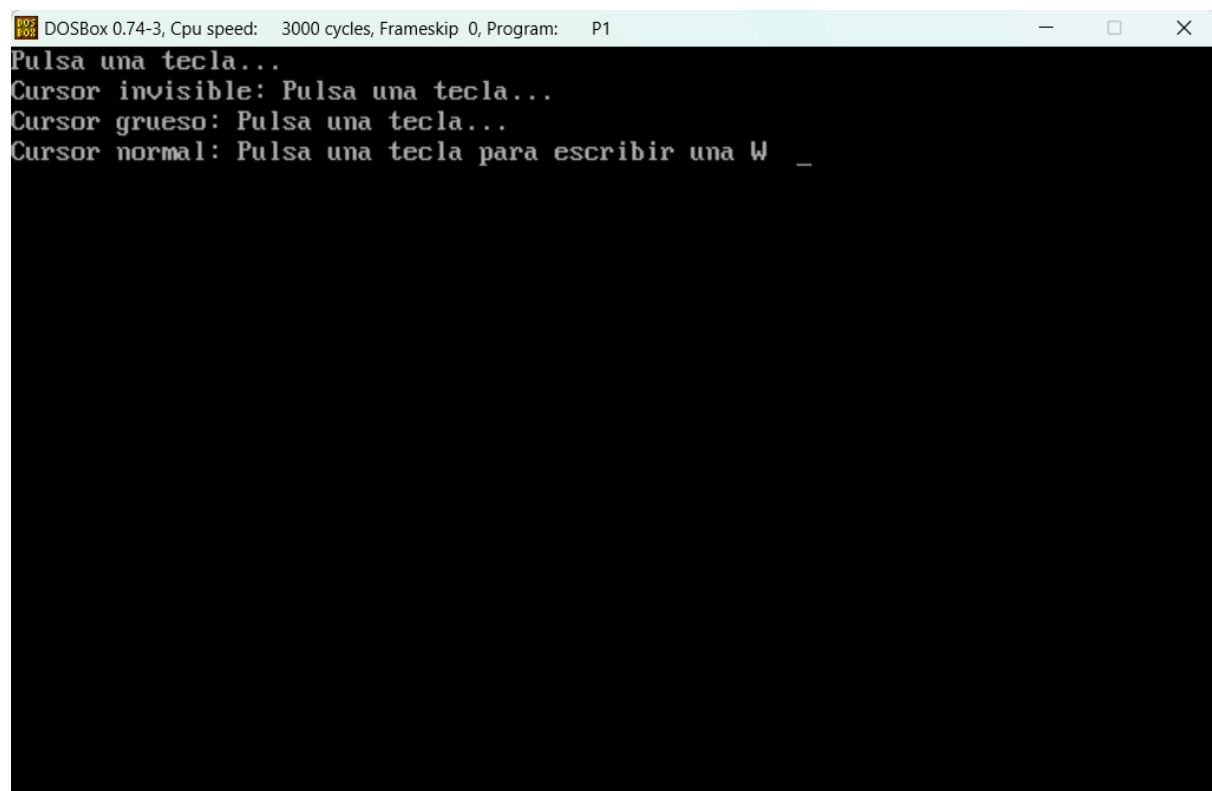




DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: P1

```
Pulsa una tecla...  
Cursor invisible: Pulsa una tecla...  
Cursor grueso: Pulsa una tecla... █
```

This screenshot shows a DOSBox window with a black background. The title bar at the top reads "DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: P1". The command prompt displays three lines of text: "Pulsa una tecla...", "Cursor invisible: Pulsa una tecla...", and "Cursor grueso: Pulsa una tecla...". A thick white cursor is visible at the end of the third line.



DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: P1

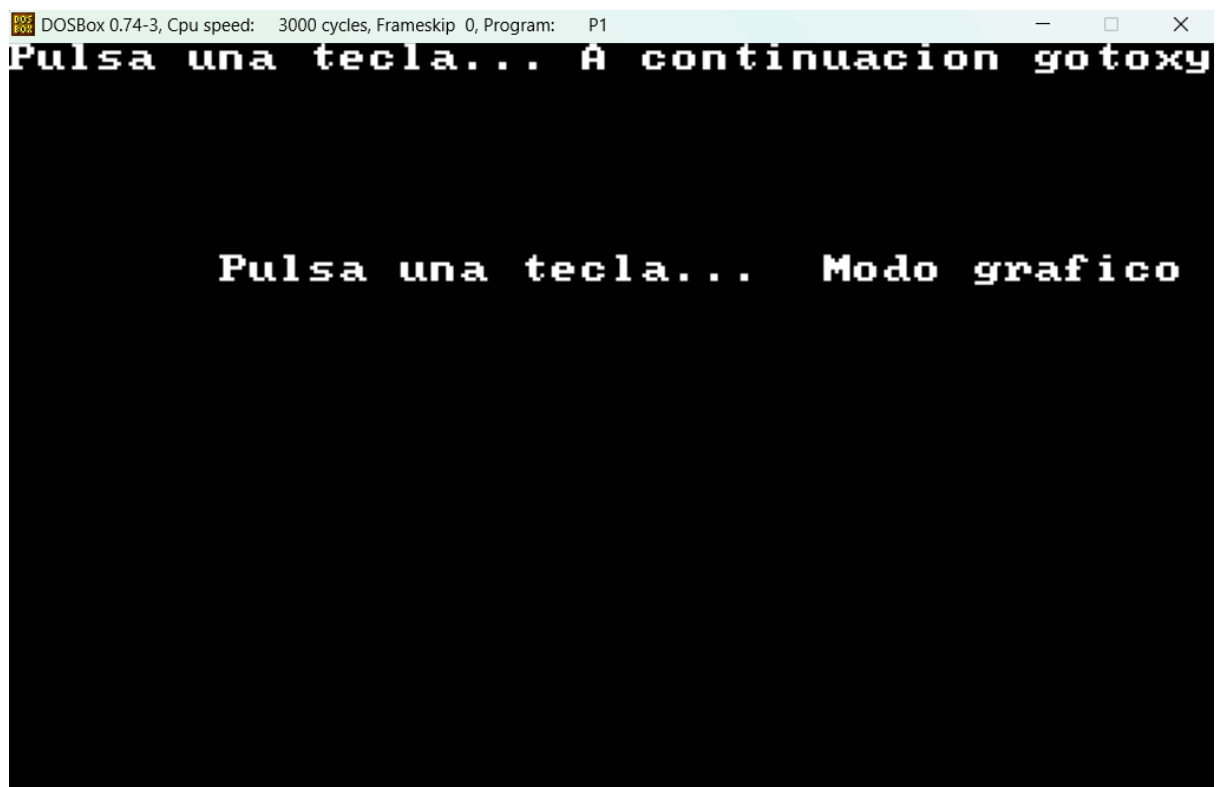
```
Pulsa una tecla...  
Cursor invisible: Pulsa una tecla...  
Cursor grueso: Pulsa una tecla...  
Cursor normal: Pulsa una tecla para escribir una W _
```

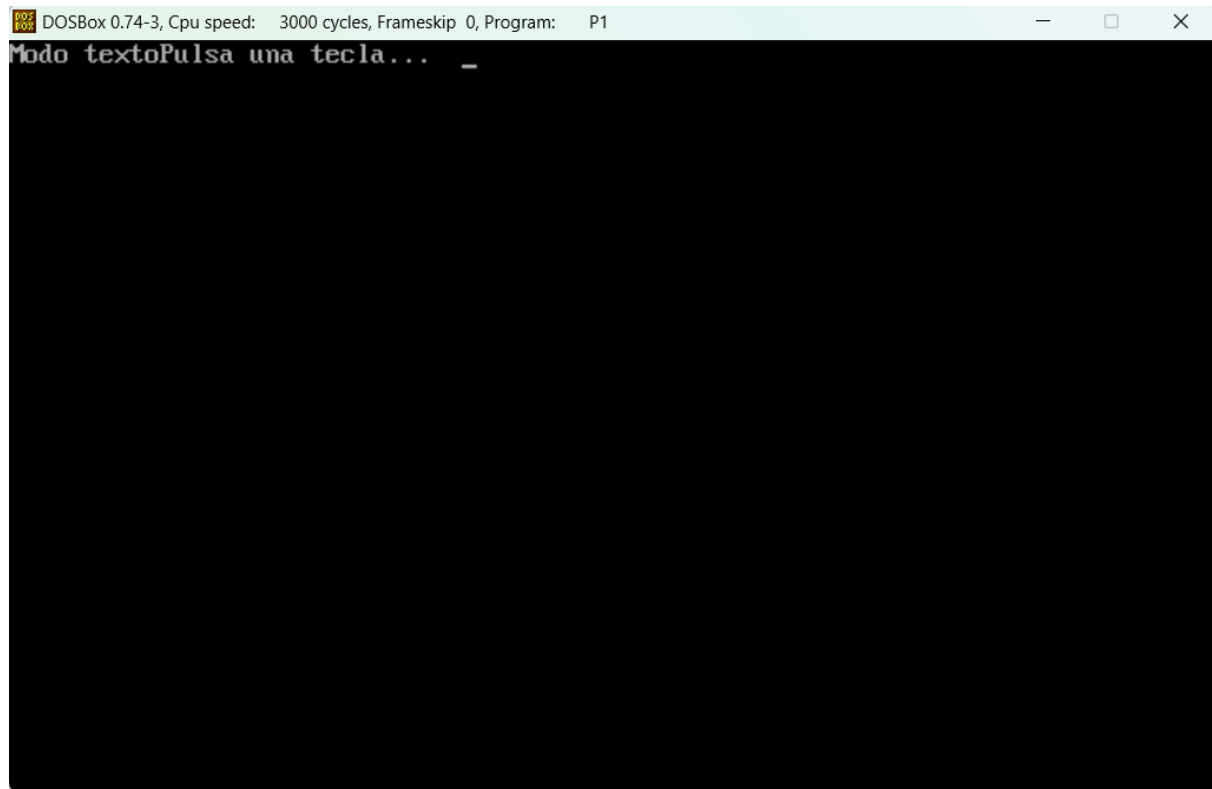
This screenshot shows a DOSBox window with a black background. The title bar at the top reads "DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: P1". The command prompt displays four lines of text: "Pulsa una tecla...", "Cursor invisible: Pulsa una tecla...", "Cursor grueso: Pulsa una tecla...", and "Cursor normal: Pulsa una tecla para escribir una W _". A thick white cursor is visible at the end of the fourth line.

Las siguientes funciones se encargan, una, de cambiar a modo gráfico o modo texto, y la otra, indicar en qué modo estamos.

```
void setvideomode(BYTE modo){
    union REGS inregs, outregs;
    inregs.h.al = modo;
    inregs.h.ah = 0x00;
    int86(0x10, &inregs, &outregs);
}
```

```
BYTE getvideomode(){
    union REGS inregs, outregs;
    BYTE modo;
    int num_col;
    inregs.h.ah = 0x0F;
    int86(0x10, &inregs, &outregs);
    modo=outregs.h.al;
    return modo;
}
```





De las 5 funciones restantes, 4 están estrechamente relacionadas, así que hablaremos primero de la otra.

Concretamente nos pedían la implementación de una función que borrara toda la pantalla.

```
void clrscr(){
    union REGS inregs, outregs;
    inregs.h.al = 0;
    inregs.h.ah = 0x06;
    inregs.h.bh = 0x00;
    inregs.h.ch = 0x00;
    inregs.h.cl = 0x00;
    inregs.h.dh = 0x25;
    inregs.h.dl = 0x80;
    int86(0x10, &inregs, &outregs);
    pausa();
    setvideomode(MODOTEXTO);
}
```

```
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: P1
Pulsa una tecla...
Cursor invisible: Pulsa una tecla...
Cursor grueso: Pulsa una tecla...
Cursor normal: Pulsa una tecla para escribir una W
W
Introduzca un caracter para mostrar
Z
Pulsa una tecla para borrar pantalla_
```

```
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: P1
```

Por último, nos pedían 4 funciones estrechamente relacionadas.

Las 2 primeras se encargarán de cambiar el color del texto y del fondo de los caracteres escritos a continuación.

```
void textcolor(BYTE color_texto){
    ctexto=color_texto;
}
```

```
void textbackground(BYTE color_fondo){
    cfondo=color_fondo;
}
```

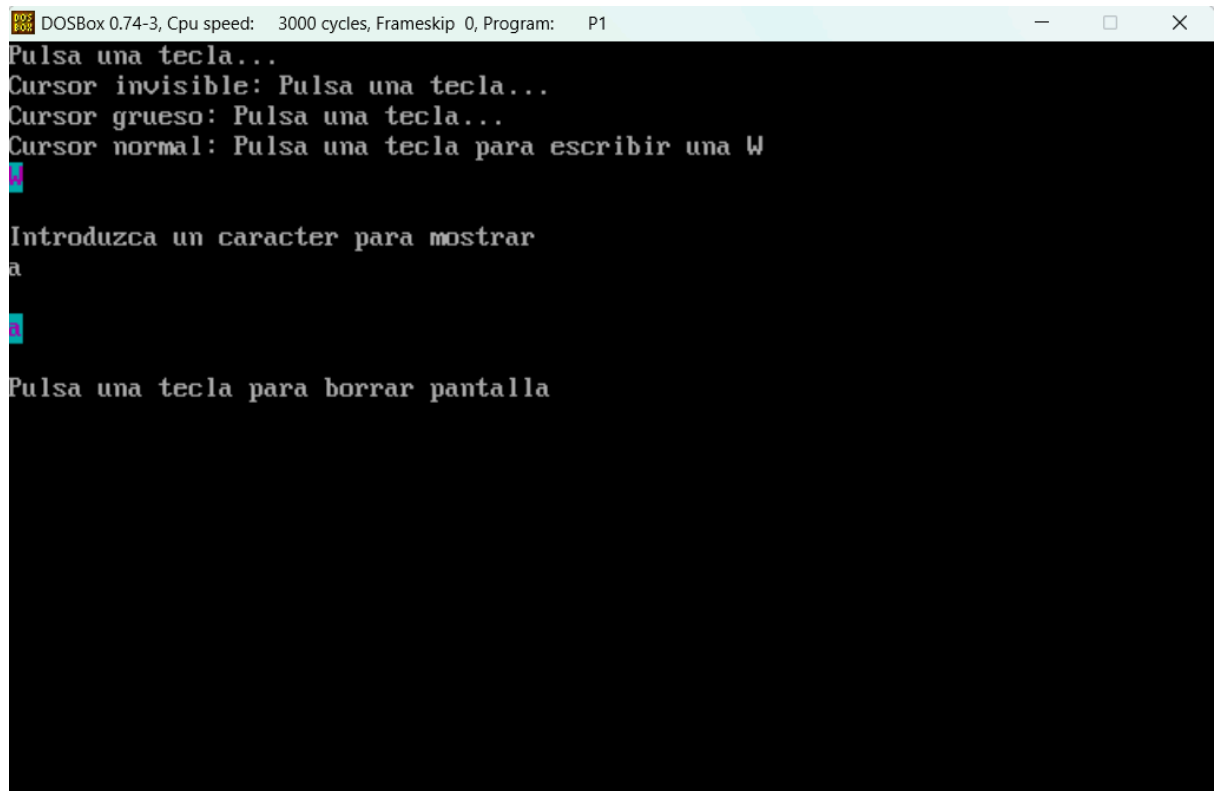
La tercera función imprime un caracter en pantalla con estos colores (en nuestro caso una W) y la última pide un caracter al usuario y lo muestra por pantalla, también con los colores predefinidos.

```
void cputchar(char caracter){
    union REGS inregs, outregs;
    inregs.h.ah = 0x09;
    inregs.h.al = caracter;
    inregs.h.bl = cfondo << 4 | ctexto;
    inregs.h.bh = 0x00;
    inregs.x.cx = 1;
    int86(0x10,&inregs,&outregs);
}
```

```
void getche(){
    union REGS inregs, outregs;
    int caracter;

    inregs.h.ah = 1;
    int86(0x21, &inregs, &outregs);

    caracter = outregs.h.al;
    printf("\n\n");
    cputchar(caracter);
}
```

```
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: P1
Pulsa una tecla...
Cursor invisible: Pulsa una tecla...
Cursor grueso: Pulsa una tecla...
Cursor normal: Pulsa una tecla para escribir una W
a
Introduzca un caracter para mostrar
a
Pulsa una tecla para borrar pantalla
```

Como se puede apreciar, se ha modificado el color del texto y del fondo y se ha mostrado por pantalla lo requerido utilizando ambas funciones.

Para todo el desarrollo de la práctica hemos utilizado algunas variables globales que fomentan un código más limpio y legible.

Éstas son:

BYTE MODOGRAFICO, BYTE MODOTEXTO, BYTE ctexto y BYTE cfondo.

Además, también hemos realizado los ejercicios adicionales. En el primero se nos pedía dibujar un recuadro obteniendo las coordenadas de las esquinas y los colores.

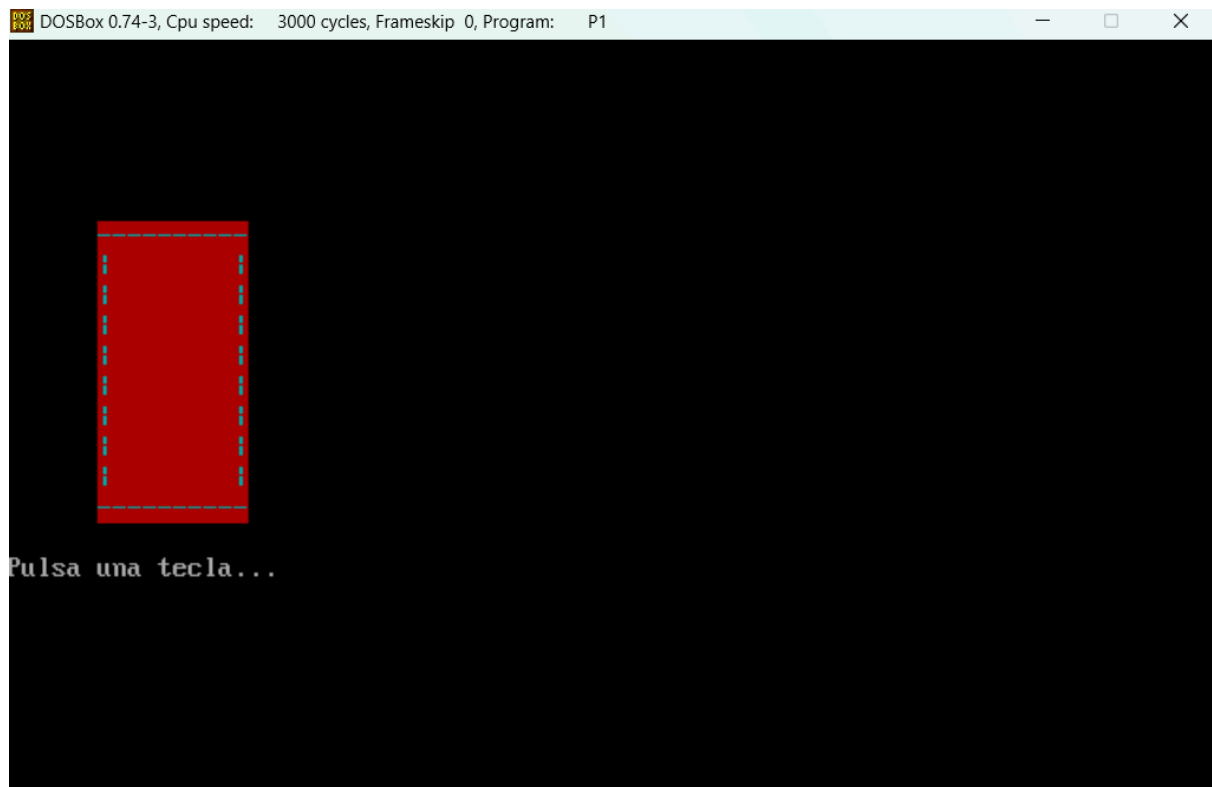
La función que hemos desarrollado dibujará el recuadro haciendo uso de los caracteres - y |

Esta función cuenta con precondiciones, todos los valores de la primera coordenada deben ser menores que los de la segunda.

El código resultante es el siguiente:

```
//Precondiciones, x1 debe ser menor que x2. y1 debe ser menor que y2.
void recuadro_texto(int x1, int y1, int x2, int y2, int color_texto, int color_fondo){
    int i=0;
    int j=0;
    textcolor(color_texto);
    textbackground(color_fondo);
    setvideomode(MODOTEXTO);
    gotoxy(x1,y1);
    // Dibujar el rectángulo
    for (i = y1; i < y2+1; i++) {
        for (j = x1; j < x2+1; j++) {
            gotoxy(j,i);
            if((i==y1)||i==y2){
                cputchar('-');
            }
            else if((j==x1)||j==x2){
                cputchar('|');
            }
            else{
                cputchar(' ');
            }
        }
    }
}
```

El resultado final es este:



Por otro lado, el último ejercicio consistía en, mediante el modo gráfico, realizar un pequeño dibujo. Nosotros hemos dibujado una cara sacando la lengua.

En esta ocasión hemos necesitado una función auxiliar llamada pixel.

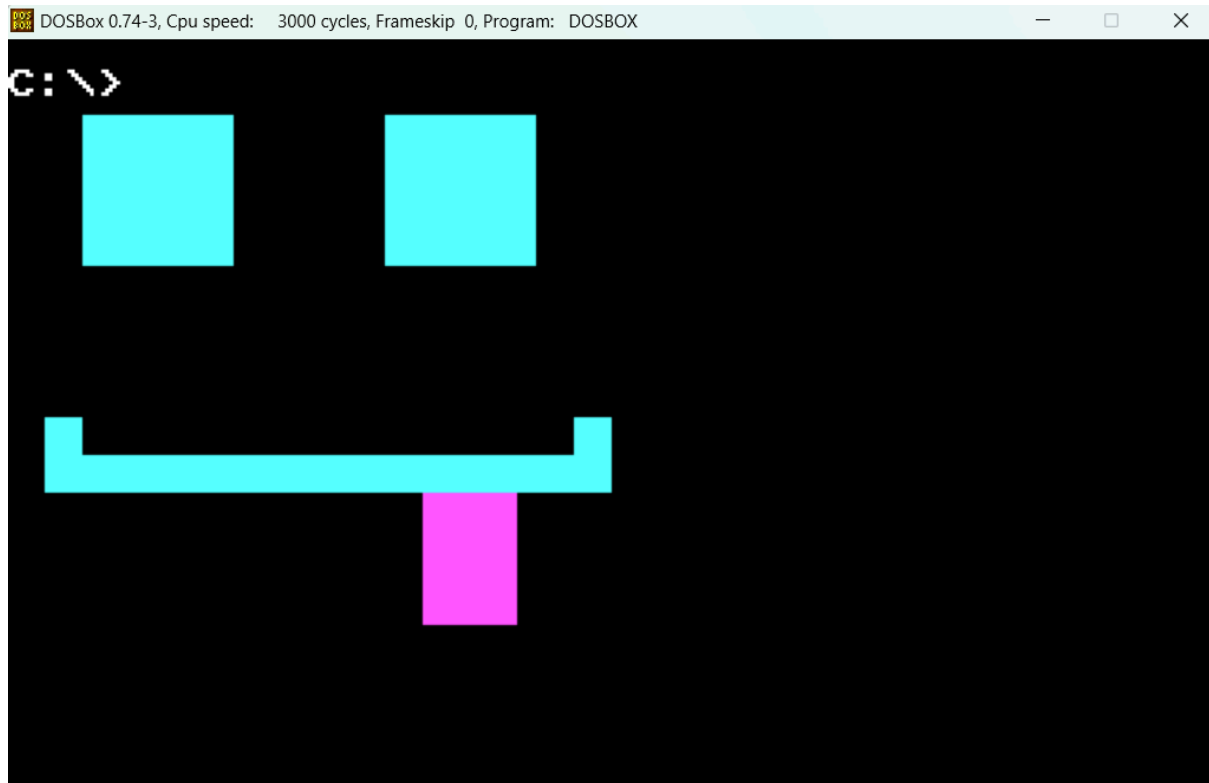
La función auxiliar es esta:

```
void pixel(int x, int y, BYTE C){
    union REGS inregs, outregs;
    inregs.x.cx = x;
    inregs.x.dx = y;
    inregs.h.al = C;
    inregs.h.ah = 0x0C;
    int86(0x10, &inregs, &outregs);
}
```

La función que realiza el dibujo:

```
void dibujografico(){
    int i=0;
    int j=0;
    setvideomode(MODOGRAFICO);
    //ojo izq
    for(i=20;i<60;i++){
        for(j=20;j<60;j++){
            pixel(i,j,1);
        }
    }
    //ojo der
    for(i=100;i<140;i++){
        for(j=20;j<60;j++){
            pixel(i,j,1);
        }
    }
    //comisura izq
    for(i=10;i<20;i++){
        for(j=100;j<120;j++){
            pixel(i,j,1);
        }
    }
    //comisura der
    for(i=150;i<160;i++){
        for(j=100;j<120;j++){
            pixel(i,j,1);
        }
    }
    //boca
    for(i=10;i<160;i++){
        for(j=110;j<120;j++){
            pixel(i,j,1);
        }
    }
    //lengua
    for(i=110;i<135;i++){
        for(j=120;j<155;j++){
            pixel(i,j,2);
        }
    }
}
```

El resultado de la ejecución:



Para la realización de las funciones de la práctica hemos utilizado interrupciones.

Concretamente, las interrupciones usadas son dos:

- 10: Peticiones de vídeo DOS. Esta interrupción se usa cuando se quiere cambiar el modo de vídeo (setvideomode) u obtenerlo (getvideomode), cuando se cambia el tipo de cursor (setcursortype) o posición (gotoxy), cambiar la disposición de la pantalla (clrscr), escribir un carácter (cputchar) o escribir un píxel (pixel). En definitiva, se usa para funciones que modifican el modo de vídeo o sus posibles atributos.
- 21: Peticiones de carácter general. Esta interrupción se usa para recibir un carácter a escribir (getche). Pese a que, en este caso, se usa poco, al ser la más general de todas las interrupciones es la más usada.

Por último, la lista de pruebas que hemos utilizado para comprobar el correcto funcionamiento de las funciones es el siguiente main:

```
int main(){
    BYTE modo;
    setvideomode(MODOGRAFICO);
    printf("Pulsa una tecla... A continuacion gotoxy ");
    pausa();
    gotoxy(7,7);
    printf("Pulsa una tecla... ");
    pausa();
    modo=getvideomode();
    if((modo<=3&&modo>=0)||modo==7){
        printf("Modo texto");
    }
    else{
        printf("Modo grafico");
    }
    pausa();
    setvideomode(MODOTEXTO);
    printf("Pulsa una tecla... ");
    pausa();
    printf("\nCursor invisible: ");
    setcursortype(0);
    printf("Pulsa una tecla... ");
    pausa();
    printf("\nCursor grueso: ");
    setcursortype(2);
    printf("Pulsa una tecla... ");
    pausa();
    printf("\nCursor normal: ");
    setcursortype(1);
    printf("Pulsa una tecla para escribir una W ");
    pausa();
    printf("\n");
    textbackground(3);
    textcolor(5);
    cputchar("W");
    printf("\n\nIntroduzca un caracter para mostrar");
    printf("\n");
    getche();
    printf("\n\nPulsa una tecla para borrar pantalla");
    pausa();
    clrscr();
    printf("Pulsa una tecla... ");
    pausa();
    setvideomode(MODOGRAFICO);
    modo=getvideomode();
```

```

if((modo<=3&&modo>=0)||modo==7){
    printf("Modo texto");
}
else{
    printf("Modo grafico");
}
printf("Pulsa una tecla... ");
pausa();
setvideomode(MODOTEXTO);
modo=getvideomode();
if((modo<=3&&modo>=0)||modo==7){
    printf("Modo texto");
}
else{
    printf("Modo grafico");
}
printf("Pulsa una tecla... ");
pausa();
recuadro_texto(6,6,15,15,3,4);
printf("\n\nPulsa una tecla... ");
pausa();
dibujografico();
return 0;
}

```