

- **Ejercicio:** activar el modo de texto 40x25 y mostrar una cadena de texto para comprobar su funcionamiento. Antes de terminar hacer una pausa esperando la pulsación de una tecla y retornar al modo de video por defecto 80x25.

```
pila segment stack 'stack'
    dw 100h dup (?)      ; tamaño de la pila
pila ends

datos segment 'data'
    cadena db 10,13,'Cadena de prueba$'
datos ends

codigo segment 'code'
    assume cs:codigo, ds:datos, ss:pila
    main PROC
        mov ax,datos
        mov ds,ax

        mov al,1        ; MODO = 1 para resolución 40x25 color
        mov ah,0        ; función para poner modo texto o gráfico deseado
        int 10h

        mov dx,OFFSET cadena
        mov ah,9        ; guarda en ah el servicio 9 (imprimir por pantalla)
        int 21h        ; interrupción 21 para el servicio 9

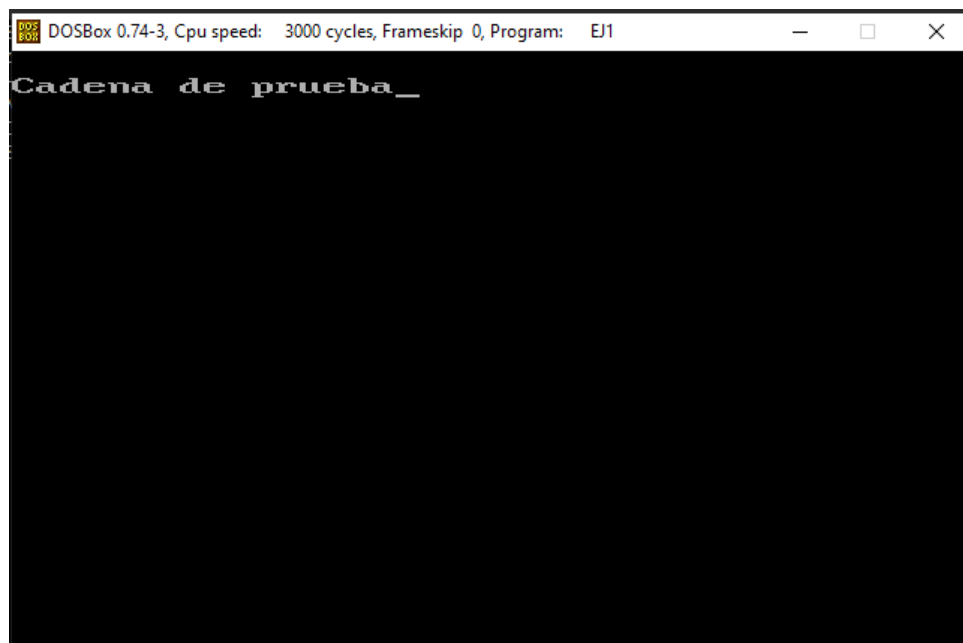
        mov ah,08h      ; función para leer una tecla
        int 21h        ; interrupción DOS para teclado (en AL devuelve el carácter tecleado)

        mov al,3        ; MODO = 3 para resolución 80x25 color (DEFAULT)
        mov ah,0        ; función para poner modo texto o gráfico deseado
        int 10h

        mov ax,4C00h    ; guarda en ax el servicio 4C00h (return)
        int 21h        ; interrupción 21 para el servicio 4C00h
    main ENDP
codigo ends

END main
```

Tras su compilación y enlazado con TASM/TLINK respectivamente y su ejecución el resultado por terminal sería el que se muestra a continuación:



- **Ejercicio:** activar el modo gráfico 320x200 a 4 colores y dibujar un rectángulo blanco. Finalmente mostrar varios pixeles de distintos colores dentro del rectángulo.

```

BLANCO EQU 3
ROSA EQU 2
VERDE EQU 1

pila segment stack 'stack'
    dw 100h dup (?) ; tamaño de la pila
pila ends

datos segment 'data'
datos ends

codigo segment 'code'
    assume cs:codigo, ds:datos, ss:pila
    main PROC
        mov ax,datos
        mov ds,ax

        mov al,4 ; MODO = 4 para modo grafico 320x200 4 colores CGA
        mov ah,0 ; función para poner modo texto o gráfico deseado
        int 10h

        ; PINTAR RECTANGULO
        mov cx, 10 ; columna
        mov dx, 10 ; fila

        bucle:
            mov al,BLANCO
            mov ah,0Ch
            int 10h
            inc cx ; incrementamos columna
            cmp cx,100 ; comparamos con 50
            jne bucle ; si no es igual a 50 volvemos a iterar
    
```

```

sigLinea:
    mov cx, 10 ; restauramos columna
    inc dx ; incrementamos fila
    cmp dx, 50 ; comparamos con 25
    jne bucle;

    ; Pintar puntos dentro del rectangulo
    mov cx,50 ;columna
    mov dx,25 ;fila
    mov al,VERDE
    mov ah,0Ch
    int 10h

    mov cx,70 ;columna
    mov dx,35 ;fila
    mov al,ROSA
    mov ah,0Ch
    int 10h

    ; Pausa para poder ver lo pintado
    mov ah,08h ; función para leer una tecla
    int 21h ; interrupción DOS para teclado (en AL devuelve el carácter tecleado)

    mov al,3 ; MODO = 3 para resolucio 80x25 color (DEFAULT)
    mov ah,0 ; función para poner modo texto o gráfico deseado
    int 10h

    mov ax,4C00h ; guarda en ax el servicio 4C00h (return)
    int 21h ; interrupcion 21 para el servicio 4C00h

    main ENDP
    codigo ends
    
```

Al igual que antes tras su compilación y enlazado con TASM/TLINK respectivamente y su ejecución el resultado por terminal sería el que se muestra a continuación:

