Manual Tecnico

```
Menu:
    limpiar
    imprimir enc1
    imprimir enc2
    imprimir salto
    getTexto bufname
    cmp bufname[0],'1'
        je prueba
    cmp bufname[0],'2'
        je LeerArchivo
    cmp bufname[0],'3'
        je Salir
    jmp Menu
```

En la imagen se puede observar el segmento de código para el menú dependiendo las opciones que se de este entrara a las etiquetas que correspondan.

```
prueba:
    imprimir salto
    imprimir matriz8
    imprimir sep
    imprimir matriz7
    imprimir sep
    imprimir matriz6
    imprimir sep
    imprimir matriz5
    imprimir sep
    imprimir matriz4
    imprimir sep
    imprimir matriz3
    imprimir sep
    imprimir matriz2
    imprimir sep
    imprimir matriz1
    imprimir salto
    imprimir letras
    imprimir salto



    cmp cx , 0
    je TurnoNegras
    cmp cx, 1
    je TurnoBlancas
```

Donde el método prueba es el que nos muestra las posiciones en el tablero en consola donde imprimir es una etiqueta para imprimir variables

```
imprimir macro string
    mov ah,09
    mov dx,offset string
    int 21h
endm
```

```
TurnoNegras:
    mov cx , 1 ; para el turno
    imprimir turnoN
    imprimir salto
    imprimir pedirX
    imprimir saltop
    getTexto bufname
    cmp bufname[0],'A'
        je validarF8
    cmp bufname[0],'B'
        je validarF7
    cmp bufname[0],'C'
        je validarF6
    cmp bufname[0],'D'
        je validarF5
    cmp bufname[0],'E'
        je validarF4
    cmp bufname[0],'F'
        je validarF3
    cmp bufname[0],'G'
        je validarF2
    cmp bufname[0],'H'
        je validarF1
    cmp bufname[0],'S'; para el show
        je CrearHtml
    cmp bufname[0],'s'; para el save
        je CrearReport
    cmp bufname[0],'P'
        je pasarTurno
    ; --------Comando EXIT---------------------
```

Donde la etiqueta turno negras y blancas es donde se valida los movimientos que pueden tener las fichas

```
validarF8:
    imprimir salto
    imprimir pediry
    imprimir saltop
    escribir
    cmp al, '8'
    je guardarMovF8
    cmp al, '7'
    je guardarMovF7
    cmp al, '6'
    je guardarMovF6
    cmp al, '5'
    je guardarMovF5
    cmp al, '4'
    je guardarMovF4
    cmp al, '3'
    je guardarMovF3
    cmp al, '2'
    je guardarMovF2
    cmp al, '1'
    je guardarMovF1

    imprimir salto
    imprimir msjerror2

    jmp prueba
; ---------------------------------------Todos los movimientos de A----------------------------
guardarMovF8:
```

donde el validar valida fila por fila los movimientos que se realizan

```
3
4   getChar macro
5       mov ah,0dh
6       int 21h
7       mov ah,01h
8       int 21h
9   endm
0
```

Macro para el salto de línea.

```
getTexto macro buffer
    LOCAL INICIO,FIN
    push si
    xor si,si
    INICIO:
        getChar
        cmp al,0dh
        je FIN
        mov buffer[si],al
        inc si
        jmp INICIO
    FIN:
        mov buffer[si],'$'
        pop si
endm
```

Este macro nos permite escribir en la consola hasta que presionemos enter

```
limpiar macro

    mov ax,0600h ;limpiar pantalla
    mov bh,0fh ;0 color de fondo negro, f color de letra blanco
    mov cx,0000h
    mov dx,184Fh
    int 10h

    mov ah,02h
    mov bh,00
    mov dh,00
    mov dl,00
    int 10h

endm
```

Este macro nos permite limpiar la pantalla

```
;---------------------FILA 7-----------------------
SC1 macro h
    LOCAL pos1, pol11,pol22,PDD
    pos1:
        cmp fila7[0] , 49
        je pol11
        cmp fila7[0],50
        je pol22
        jmp PDD
    pol11:
    contarElementos ficha1
    escribirF h , di , ficha1
    jmp PDD
    pol22:
    contarElementos ficha2
    escribirF h , di , ficha2
    PDD:
endm
```

Este macro nos permite escribir en un archivo el html

```
;------------------------------PARA FICH
abrirF macro ruta, handle
    mov ah,3dh
    mov al,010b
    lea dx,ruta
    int 21h
    jc ErrorAbrir
    mov handle,ax
endm
```

Este macro nos permite abrir un fichero

```
crearF macro ruta, handle
    mov ah,3ch
    mov cx,00h
    lea dx, ruta
    int 21h
    jc ErrorCrear
    mov handle,ax
endm
```

Macro que nos permite crear un fichero

```
escribirF macro handle, numBytes, buffer
    mov ah,40h
    mov bx,handle
    mov cx,numBytes
    lea dx,buffer
    int 21h
    jc ErrorEscribir
endm
```

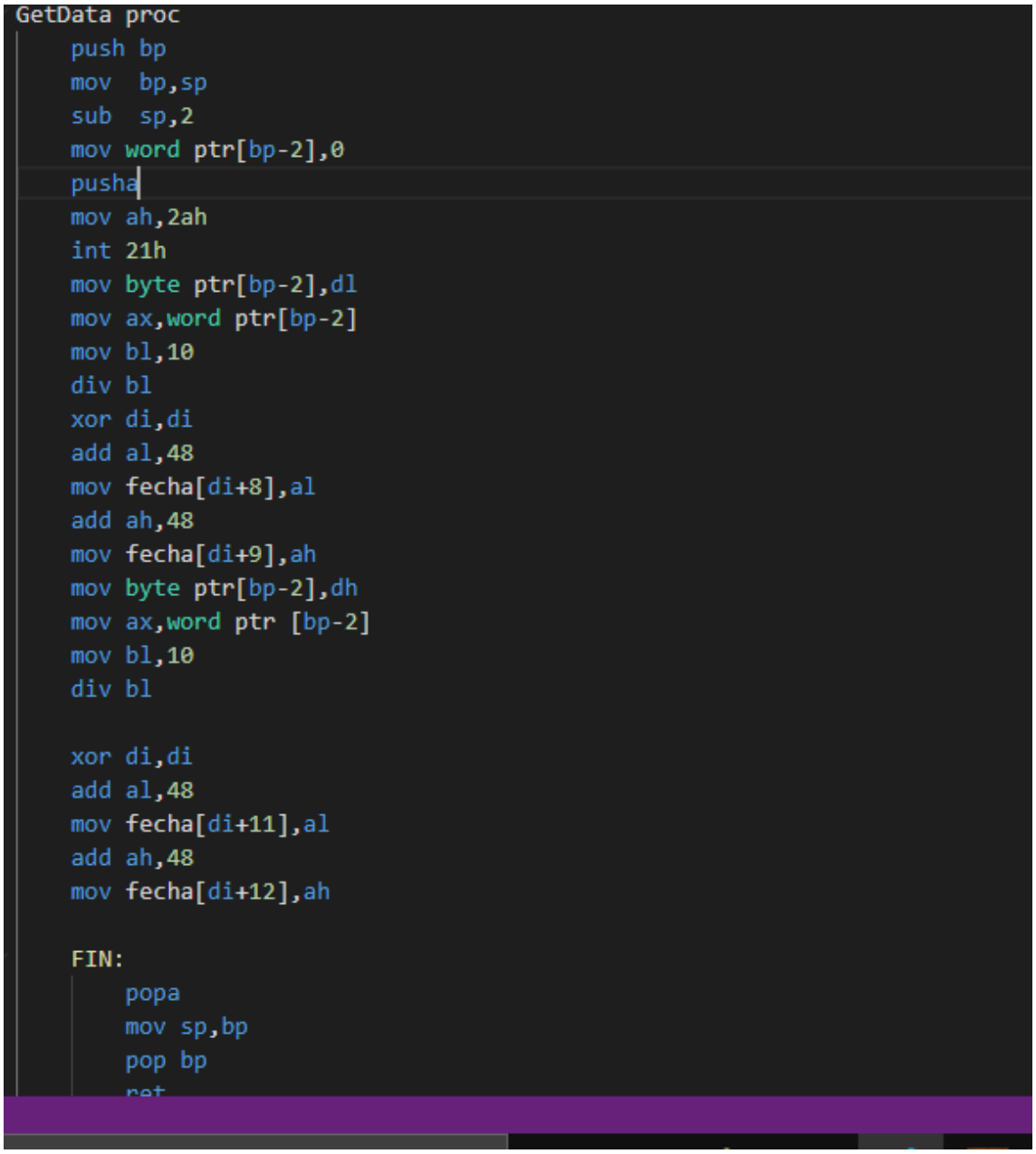
macro que nos permite abrir un fichero

```
leerF macro handle, numBytes, buffer
        mov ah,3fh
        mov bx,handle
        mov cx,numBytes
        lea dx, buffer
        int 21h
        jc ErrorLeer

endm

cerrarF macro handle
    mov ah,3eh
    mov bx,handle
    int 21h
endm
```

Estos macros nos permiten leer un fichero y cerrarlo por que si no se cierra un fichero seguiría escribiendo en el.



```
GetData proc
    push bp
    mov  bp,sp
    sub  sp,2
    mov word ptr[bp-2],0
    pusha
    mov ah,2ah
    int 21h
    mov byte ptr[bp-2],dl
    mov ax,word ptr[bp-2]
    mov bl,10
    div bl
    xor di,di
    add al,48
    mov fecha[di+8],al
    add ah,48
    mov fecha[di+9],ah
    mov byte ptr[bp-2],dh
    mov ax,word ptr [bp-2]
    mov bl,10
    div bl

    xor di,di
    add al,48
    mov fecha[di+11],al
    add ah,48
    mov fecha[di+12],ah

    FIN:
        popa
        mov sp,bp
        pop bp
        ret
```

Procedimiento para obtener la fiche y mostrarla en un html