



Manual Técnico

Para que el programa funcione correctamente, es necesario utilizar dos archivos utilizados para el guardar la información de los jugadores y de las partidas jugadas. El archivo que almacena a los jugadores es “**players.txt**” y tiene que estar colocado en el directorio raíz de donde se encuentre la aplicación. El archivo que contiene todos los resúmenes de partidas tiene que estar colocado en el mismo directorio de donde se encuentre el ejecutable, y tiene el nombre de “**top.txt**”. La notación que utiliza cada archivo es la siguiente:

```
players: Bloc de notas
Archivo Edición Formato Ver Ayuda
admin,1234$;edwin,12345;user1,pass1/$$$$$$$$$$$$$$
```

```
TOP: Bloc de notas
Archivo Edición Formato Ver Ayuda
00000,0,0,0;edwin,1,009,001;edwin,3,049,031;edwin,
```

Macros Utilizadas

PrintChar

Macro utilizada para mostrar un carácter en el modo texto.

```
printChar macro char
    LOCAL Start
    Start:
        mov ah,02h
        mov dl,char
        int 21h
endm
```

PrintArray

Macro utilizada para mostrar un arreglo en el modo texto.

```
printArray macro array
    LOCAL Start
    Start:
        mov ah,09h
        mov dx,@data
        mov ds,dx
        mov dx,offset array
        int 21h
endm
```

GetLine

Macro utilizada para obtener una línea de texto y almacenarla en un arreglo.

```
getLine macro buffer
    LOCAL Start,Finish
    xor si,si
    Start:
        getChar
        cmp al,0dh
        je Finish
        mov buffer[si],al
        inc si
        jmp Start
    Finish:
        mov buffer[si],'$'
endm
```

GetChar

Macro utilizada para obtener un carácter y almacenarlo en un registro.

```
getChar macro
    mov ah,0dh
    int 21h
    mov ah,01h
    int 21h
endm
```

Clean

Macro utilizada para llenar de un caracter en especifico un determinado arreglo.

```
clean macro buffer,numbytes,caracter
    local CleanLoop
    xor bx,bx
    xor cx,cx
    mov cx,numbytes
CleanLoop:
    mov buffer[bx],caracter
    inc bx
    Loop CleanLoop
endm
```

Delay

Macro utilizada para agregar un retardo de tiempo de ejecución.

```
delay macro constant
    local Delay1,Delay2,EndDelay
    push si
    push di
    xor si,si
    xor di,di
    mov si,constant
Delay1:
    dec si
    jz EndDelay
    mov di,constant
Delay2:
    dec di
    jnz Delay2
    jmp Delay1
EndDelay:
    pop di
    pop si
endm
```

DrawBall

Macro utilizada para dibujar una pelota en la memoria de video.

```
drawBall macro position, color
    push dx
    mov di,position
    mov dl,color
    mov es:[di+0],dl
    mov es:[di+1],dl
    mov es:[di+2],dl
    mov es:[di+320],dl
    mov es:[di+321],dl
    mov es:[di+322],dl
    mov es:[di+640],dl
    mov es:[di+641],dl
    mov es:[di+642],dl
    pop dx
endm
```

DrawBar

Macro utilizada para dibujar una barra en la memoria de video.

```
drawBar macro position, color
local DrawBarLoop
    push dx
    push cx
    xor cx,cx
    mov cx,position
    add cx,32
    mov di,position
    mov dl,color
    DrawBarLoop:
        mov es:[di+0],dl
        mov es:[di+320],dl
        mov es:[di+640],dl
        inc di
        cmp di,cx
        jne DrawBarLoop
    pop cx
    pop dx
endm
```

DrawBlock

Macro utilizada para dibujar un bloque en la memoria de video.

```
drawBlock macro position, color
local DrawBlockLoop
    push dx
    push cx
    xor cx,cx
    mov cx,position
    add cx,48
    mov di,position
    mov dl,color
    DrawBlockLoop:
        mov es:[di+0],dl
        mov es:[di+320],dl
        mov es:[di+640],dl
        mov es:[di+960],dl
        mov es:[di+1280],dl
        mov es:[di+1600],dl
        inc di
        cmp di,cx
        jne DrawBlockLoop
    pop cx
    pop dx
endm
```

Variables Utilizadas

```
message1 db 0ah,0dh,0ah,0dh,09h,'#### BALLS BRICKS BREAKER ####',0ah,0dh,0ah,0dh,09h,'1. LOGIN',0ah,0dh,
message2 db 0ah,0dh,'$'
message3 db 0ah,0dh,0ah,0dh,09h,'===== SIGNIN =====',0ah,0dh,0ah,0dh,09h,'>>NICKNAME: ','$'
message4 db 0ah,0dh,09h,'>>PASSWORD: ','$'
message5 db 0ah,0dh,09h,'**** ERROR ABRIENDO ARCHIVO ****','$'
message6 db 0ah,0dh,09h,'**** ERROR LEYENDO ARCHIVO ****','$'
message7 db 0ah,0dh,09h,'**** ERROR CREANDO ARCHIVO ****','$'
message8 db 0ah,0dh,09h,'*** ERROR ESCRIBIENDO ARCHIVO **','$'
message9 db 0ah,0dh,09h,'**** ERROR CERRANDO ARCHIVO ****','$'
message10 db 0ah,0dh,0ah,0dh,09h,'===== LOGIN =====',0ah,0dh,0ah,0dh,09h,'>>NICKNAME: ','$'
message11 db 0ah,0dh,0ah,0dh,09h,'>>>>>>>>> ADMIN <<<<<<<<<<<',0ah,0dh,0ah,0dh,09h,'1. TOP 10 SCORES'
message12 db 0ah,0dh,0ah,0dh,09h,'::: TOP10 SCORES :::','$'
message13 db 0ah,0dh,0ah,0dh,09h,'::: TOP10 TIMES :::','$'
```

```
nickname db 5 dup('$')
password db 5 dup('$')
levelone db 12 dup('$')
leveltwo db 24 dup('$')
levelthree db 36 dup('$')
playerspath db 16 dup('$')
rankingpath db 16 dup('$')
info db 6 dup('$')
top1 db 8 dup('$')
top2 db 8 dup('$')
top3 db 8 dup('$')
top4 db 8 dup('$')
top5 db 8 dup('$')
top6 db 8 dup('$')
top7 db 8 dup('$')
top8 db 8 dup('$')
top9 db 8 dup('$')
top10 db 8 dup('$')
top11 db 8 dup('$')
topaux db 8 dup('$')
players db 240 dup('$')
filehandle dw ?
```

```
finalscore db 0
finaltime db 0
currentlevel db 0
typeofsort db 0
delaygameplay db 0
quadrantfirst db 0
quadrantsecond db 0
quadrantthird db 0
switchball db 0
thirdball dw 0
secondball dw 0
firstball dw 0
currenttime dw 0
startlevel db 0
time dw 0
```


Código Principal

```
main proc ...
main endp

login proc ...
login endp

AdminControlPanel proc ...
AdminControlPanel endp

showTop10 proc ...
showTop10 endp

itsNumber proc ...
itsNumber endp

fillRankingPath proc ...
fillRankingPath endp

sortTop proc ...
sortTop endp

cleanTops proc ...
cleanTops endp

addResume proc ...
addResume endp
```

```
signin proc ...
signin endp

fillPlayersPath proc ...
fillPlayersPath endp

graphicMode proc ...
graphicMode endp

drawMargin proc ...
drawMargin endp

gamePlay1 proc ...
gamePlay1 endp

gamePlay2 proc ...
gamePlay2 endp

gamePlay3 proc ...
gamePlay3 endp

textMode proc ...
textMode endp
```

```
showNumber proc ...
showNumber endp

OpenError:
    printArray message5
    jmp main
ReadError:
    printArray message6
    jmp main
CreateError:
    printArray message7
    jmp main
WriteError:
    printArray message8
    jmp main
CloseError:
    printArray message9
    jmp main
```

El funcionamiento del programa comienza en el bloque **main**, en donde se muestra el menú principal y se solicita la instrucción a ejecutar. Para poder iniciar sesión se usa el bloque **login** y para registrar el bloque **signin**. Para poder acceder a los archivos que contienen a los jugadores y resúmenes de partidas son **fillRankingPath** y **fillPlayersPath**. El bloque **addResume** es utilizado para agregar un nuevo resumen en el archivo que contiene a los mismos. El bloque **graphicMode** pasa a modo video la aplicación y el bloque **textMode** pasa a modo texto la aplicación. Los bloques **gamePlay1**, **gamePlay2** y **gamePlay3** corresponden a los niveles de juego. El bloque para ordenar el ranking de jugadores es **sortTop** y para mostrarlos el bloque **showTop10**. El bloque **drawMargin** se encarga de pintar los delimitadores del tablero. El método **showNumber** es utilizado para castear valores numéricos y poder mostrarlos en consola. El bloque **cleanTops** se encarga de limpiar el ranking inicialmente. Las etiquetas restantes son utilizadas para mostrar mensajes de error en las operaciones a las que hacen alusión.