UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

FACULTAD DE INGENIERIA

ARQUITECTURA DE COMPUTADORES Y ENSAMBLADORES 1.

TUTOR: OSCAR CUELLAR

PRACTICA 5

MANUAL DE TECNICO

JUAN PABLO OSUNA DE LEON

201503911

juanpabloosuna1997@gmail.com

REQUERIMIENTOS DEL SISTEMA:

- WINDOWS 32/64 BIT
- INSTALACION DE DOSBOX
- INSTALACION DE COMPILADOR MASM

MODO VIDEO

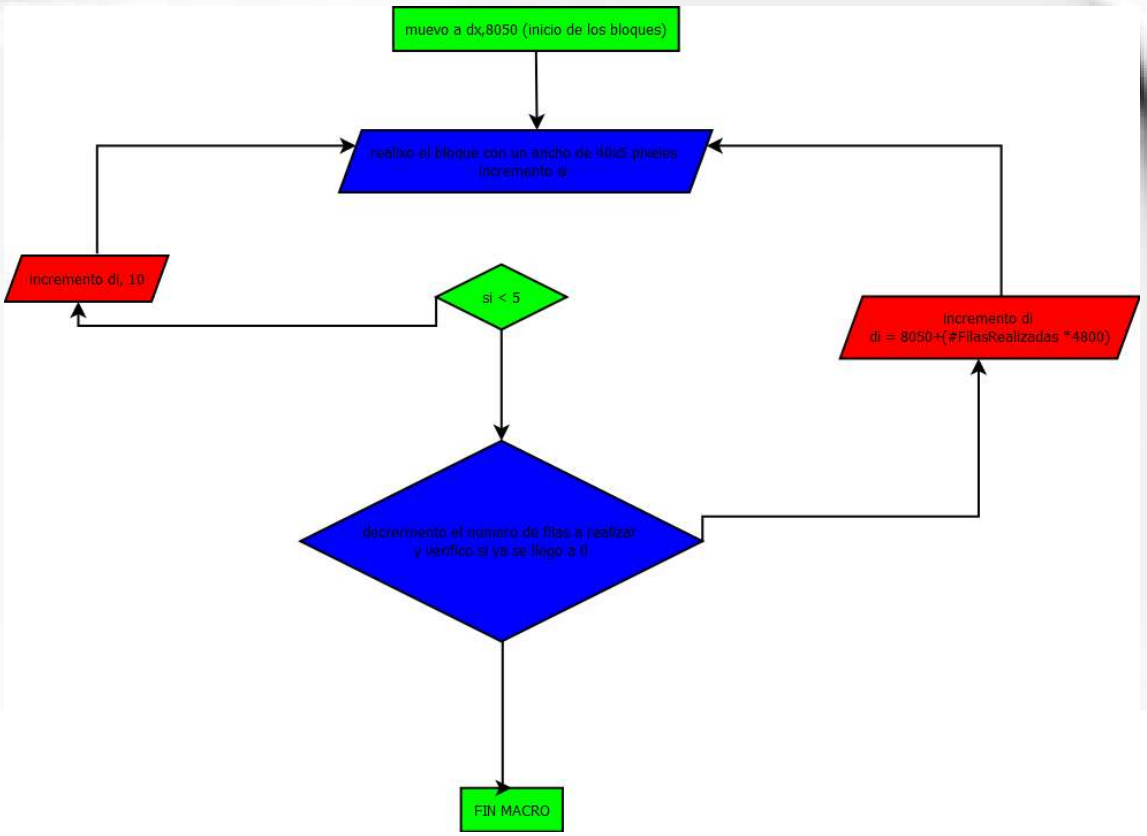
Macro utilizado para establecer en el segmento de dato extra la posición donde se iniciara a graficar.

```
ModoVideo macro
    mov ah,00h
    mov al,13h
    int 10h
    mov ax, 0A000h
    mov es, ax  ; ES = A000h (memoria de graficos).
endm
```

PINTAR BLOQUES:

Se realizará un macro el cual dibujara 5 bloques por fila con un ancho de 5 pixeles, cada bloque estará separado por 10 pixeles entre sí, se recibirá como parámetro la cantidad de filas que se realizaran, esto dependerá del nivel en el que se encuentre.

Diagrama de flujo del mismo:



Programación de macro:

```
PintarBloques macro filas       ;recibira como parametro las filas, esto dependera de cada nivel
    LOCAL anchoBloque,Bloque,Fin,sumo,sigo

    mov dl,14   ;color de los bloques
    xor di,di   ;inicio de bloques
    xor cx,cx   ;ancho de bloque
    xor si,si   ;cantidad de bloques realizados
    xor bx,bx   ;cantidad de filas realizadas
    mov bx,filas
    xor ax,ax   ;numero de filas realizadas

    ;margen empieza en la fila 20, loa bloques empezaran en la linea 25 y columna 50
    mov di,8050 ;(25*320)+50 = 8050 inicio de los bloques

    anchoBloque:
        mov cx,40   ;ancho del bloque
    Bloque:
        mov es:[di],dl ;color de los cubos
        mov es:[di+320],dl
        mov es:[di+640],dl
        mov es:[di+960],dl
        mov es:[di+1280],dl
        inc di  ;pintara de ancho de 3 filas cada bloque
        loop Bloque

    ;cuando termine de dibujar un bloque verifico si a esa fila le caben mas bloques
    inc si
    add di,0ah  ;una separacion de 10 pixeles horizontalmente
    cmp si,05h  ;si ya complete los 5 bloques por fila paso a la siguiente filaa
    jne anchoBloque

    inc ax          ;incremento las filas realizadas
    cmp bx,1
    je Fin          ;comparo si es la ultima linea de lo contrario mando a hacer otra linea
    dec bx
    mov di,8050     ;regreso a la posicion inicial y debo bajar: el tamaño del bloque = 5 mas 10 filas de separacion

    push ax     ;guardo la cantidad de filas que llevo
    ;AJUSTO LA NUEVA POSICION EN LA LINEA DE ABAJO
    sumo:
        ;sumo 4800 por cada linea que llevo
        cmp ax,0
        je sigo
        add di,4800     ;le sumo las 5 filas ocupadas por el bloque anterior mas 10 FILAS de esapacio vertical
        dec ax
        jmp sumo

    sigo:
        pop ax  ;retomo el valor de lineas que llevo
        xor si,si
        jmp anchoBloque     ;inicio a dibujar la nueva fila

    Fin:
endm
```

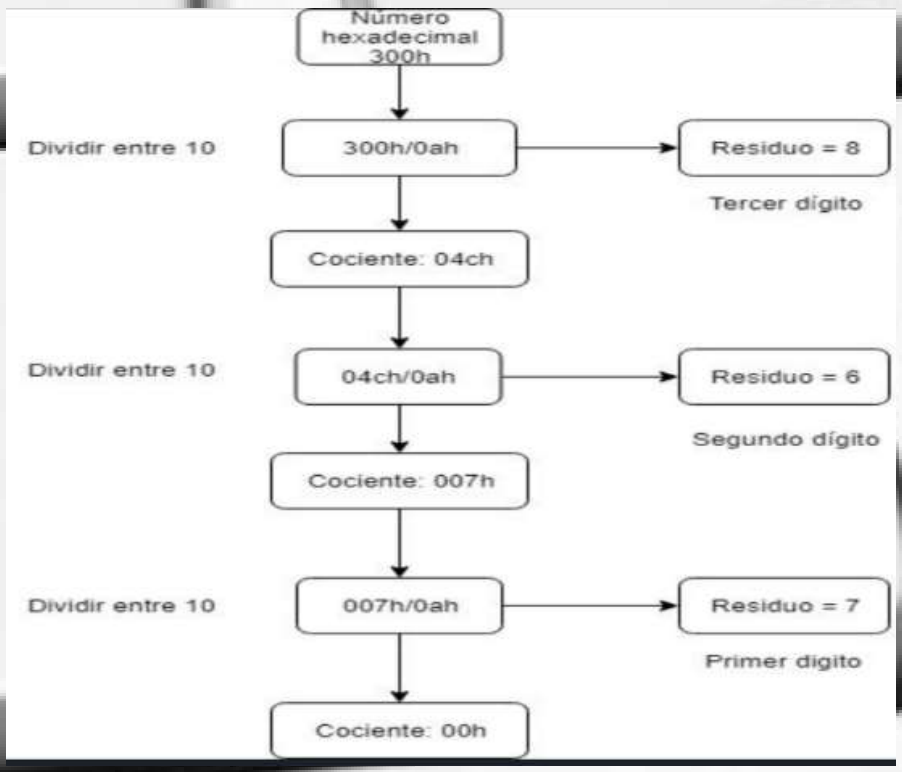
MOSTRAR ENCABEZADO

Se mostrara el nombre del usuario en sesión activa, seguido del nivel que juega, seguido de su puntaje y su tiempo.

```
MostrarEncabezado macro
    LOCAL Seguir,Otro,SaltoLineaar,SaltoLineaar2,pri,noes,niv1,niv2,imprimir

    ImpresionCaracter 32    ;imprimimos un espacio
    printGrafico name_      ;nombre de usuario actual

    ImpresionCaracter 32 ;imprimimos un espacio
    ImpresionCaracter 32 ;imprimimos un espacio

    ImpresionCaracter 78 ;Letra N de Nivel
    push ax
    xor ax,ax
    mov ax,NivelActual
    add ax,30h
    ImpresionCaracter al ;numero de nivel
    pop ax

    ImpresionCaracter 32 ;imprimimos un espacio
    ImpresionCaracter 32 ;imprimimos un espacio

    ConvertirBCD punetoActual

    ;ImpresionCaracter centena
    ImpresionCaracter decena
    ImpresionCaracter unidad

    ImpresionCaracter 32 ;imprimimos un espacio
    ImpresionCaracter 32 ;imprimimos un espacio

    ;tiempo----
    ;capturo el tiempo actual
    push ax
    push cx
    push bx
    push dx

    mov ah,2ch
    int 21h

    mov segundosActual,dh  ;dh =segundos
    mov minutosActual,cl    ;cl=minutos
    xor ax,ax
    xor bx,bx
    mov al,minutosActual
    mov bx,60
    mul bx
    xor bx,bx
    mov bl,segundosActual
    add ax,bx
    sub ax,segundostotalesInicio    ;resto el tiempo en el que inicie asi encuantro la diferencia de segundos
    mov tiempoTotalSegundos,ax
    mov timepoParaRegistro,ax   ;voy guardando en el registro el tiempo acumulado

    ConvertirBCD tiempoTotalSegundos
    ImpresionCaracter centena
    ImpresionCaracter decena
    ImpresionCaracter unidad

    ImpresionCaracter 13    ;retorno de carro para que siempre me escriba en la misma linea
    pop dx
    pop bx
    pop cx
    pop ax
endm
```

HEXADECIMAL A DECIMAL

Siguiendo el algoritmo:



El resultado se almacenara en 3 variables, centena, decena, unidad

```
ConvertirBCD macro numero
    LOCAL REALIZAR,HEXA_DECIMAL,BCDCOMPLETO
    push di
    push ax
    push dx
    push cx

    mov ax,numero
    REALIZAR:
        xor dx,dx
        ; ImpresionCaracter al
        ; mov bl,ah
        ; ImpresionCaracter bl
        ; ;leercaracter

        ;limpio las variables
         mov centena,0
         mov decena,0
         mov unidad,0

         xor cx,cx
         mov cx,0ah
         HEXA_DECIMAL:

            div cx  ;RESIDUO DX Y COCIENTE EN AX
            mov unidad,dl

            cmp ax,00h
            je BCDCOMPLETO

            xor dx,dx

            div cx  ;RESIDUO DX Y COCIENTE EN AX
            mov decena,dl

            cmp ax,00h
            je BCDCOMPLETO

            xor dx,dx

            div cx  ;RESIDUO DX Y COCIENTE EN AX
            mov centena,dl

            cmp ax,00h
            je BCDCOMPLETO

            xor dx,dx

            BCDCOMPLETO:
                 add centena,30h    ;le sumamos 30h para mostrar ek numero real en pantalla
                 add decena,30h
                 add unidad,30h

                 pop cx
                 pop dx
                 pop ax
                 pop di
endm
```

LEER TECLA

Para leer la tecla en tiempo de ejecución de utilizar la interrupción 16h, la cual primero se leerá el estado del buffer, es decir si hay disponible una tecla para ser leída, si fuera que no continua con la jugabilidad normal, en caso de que haya una tecla en espera, este comprara para saber que tecla se oprimió y hará las acciones correspondientes.

```
;==============================JUEGO=======================
leertecla macro
    LOCAL pausa,off,VerificarTecla,Derecha,Izquierda,Nosepuede,Nosepuede1,pausa,Nivel2_,Nivel3_,estoyNiv3D,estoyNiv3I,yes,rep,rep1
    push ax
    ;xor ax,ax
    mov ah,01h
    int 16h ;verificar si hay tecla lista para ser leida
    jz off
    mov ah,00h
    int 16h ;leer la tecla

    cmp ah,1 ;si es el boton ESC
    jne VerificarTecla

    pausa:
        mov ah,00h  ;haccemos que espere una tecla por ende el juego quedara congelado
        int 16h

        cmp ah,1 ;si es otra ESC reanuda el juego
        je off

        cmp ah,57 ;espacio corresponde a 9 = 57, si viene espacio se regresara al menu principal
        je INSERTARUSUARIO  ;como se sale del juego mandamos a insertar la data que tenemos

        cmp al,50 ;a nivel 2
        je PASAMOSNIVEL2

        cmp al,51 ;a nivel 3
        je PASAMOSNIVEL3

    VerificarTecla:

        cmp ah,77
        je Derecha  ;la flecha derecha responde a la letra M

        cmp ah,75
        je Izquierda    ;la flecha derecha responde a la letra K

        cmp ah,57   ;espacio
        je INSERTARUSUARIO
```

MOVER PELOTA

Se manejara estados en el cual la pelota debe moverse, el estado incremento derecha = 1, decremento derecha = 3, incremento izquierda = 2, decremento izquierda = 4

```
            sub dx,319
    IncrementoDerecha:
            push dx
            MostrarEncabezado
            pop dx
            leertecla
            pintarPelota dx, 0
            sub dx,319
```

Para estado incremento derecha se reducirá 319 pixeles y se pintara la pelota, antes se debe borrar la pelota dibujada anteriormente

Para estado decremento derecha se sumara 321pixeles y se pintara la pelota, antes se debe borrar la pelota dibujada anteriormente.

```
DecrementoDerecha:
        push dx
        MostrarEncabezado
        pop dx
        leertecla
        pintarPelota dx, 0
        add dx,321
```

Para estado decremento izquierda se sumara 319 pixeles y se pintara la pelota, antes se debe borrar la pelota dibujada anteriormente.

```
DecrementoIzquierda:
        push dx
        MostrarEncabezado
        pop dx
        leertecla
        pintarPelota dx, 0
        add dx,319
```

Para estado incremento izquierda se reducirá 321 pixeles y se pintara la pelota, antes se debe borrar la pelota dibujada anteriormente

```
    IncrementoIzquierda:
            push dx
            MostrarEncabezado
            pop dx
            leertecla
            pintarPelota dx, 0
            sub dx,321
```

VALIDAR CHOQUE

Para validar choque se analizaran todo los bloques que existen desde la posición inicial, le utilizara las misma metodología de pintar bloque, pero la diferencia que en lugar de pintar el bloque se analizara si hay un pixel del color de la pelota de serlo el bloque se pinta de negro y se sumara un punto.

```
RevisoBloque:
    mov dl,es:[di-320] ;arriba
    cmp dl,2
    je Choco

    mov dl,es:[di-1] ;fila 0 izquierda
    cmp dl,2
    je Choco

    mov dl,es:[di+1] ;fila 0 derecha
    cmp dl,2
    je Choco

    mov dl,es:[di+319] ;fila 1 izquierda
    cmp dl,2
    je Choco

    mov dl,es:[di+321] ;fila 1 derecha
    cmp dl,2
    je Choco

    mov dl,es:[di+639] ;fila 2 izquierda
    cmp dl,2
    je Choco

    mov dl,es:[di+641] ;fila 2 derecha
    cmp dl,2
    je Choco

    mov dl,es:[di+959] ;fila 3 izquierda
    cmp dl,2
    je Choco

    mov dl,es:[di+961] ;fila 3 derecha
    cmp dl,2
    je Choco

    mov dl,es:[di+1279] ;fila 4 izquierda
    cmp dl,2
    je Choco

    mov dl,es:[di+1281] ;fila 4 derecha
    cmp dl,2
    je Choco

    mov dl,es:[di+1600] ;fila 5 abajo
    cmp dl,2
    je Choco
```

Limpieza de bloque:

```
Choco:
    ;printGrafico aqui
    pop di      ;inicio del bloque que colisiono
    mov dl,0 ;muevo el color negro
    mov cx,40   ;anccho del bloque
    limpiaar:
        mov es:[di],dl
        mov es:[di+320],dl
        mov es:[di+640],dl
        mov es:[di+960],dl
        mov es:[di+1280],dl
        inc di
        Loop limpiaar
```

SUMADOR DE PUNTOS

Se sumara un punto por cada colisión que se detecte, y se verificara si el punteo ya es 10, pasa al segundo nivel, si es 25 pasa al 3er nivel y si es 45 gana.

```
add punetoActual,01h        ;si el punteo llega a 10,

cmp punetoActual,17
je activoPelota2

cmp punetoActual,0ah        ;nivel1 se gana con 10 pu
je Gano1erNivel

cmp punetoActual,19h        ;nivel 2 se gana con 25 p
je Gano2doNivel

cmp punetoActual,2dh        ;nivel 3 se gana con 45 p
je Gano3erNivel

jmp finval2

activoPelota2:
    mov estadop2,1
    add vidas,01h
    jmp finval2

Gano1erNivel:
    ;NIVEL2
    ;push ax
    ;mov ax,tiempoTotalSegundos
    ;mov tiempoNivel1,ax
    ;pop ax
    jmp PASAMOSNIVEL2

Gano2doNivel:
    ;NIVEL2
    ;push ax
    ;mov ax,tiempoTotalSegundos
    ;mov tiempoNivel1,ax
    ;pop ax
jmp PASAMOSNIVEL3

Gano3erNivel:
    ;NIVEL2
    ;push ax
    ;mov ax,tiempoTotalSegundos
    ;mov tiempoNivel1,ax
    ;pop ax
jmp INSERTARUSUARIO
```