2022

Manual Técnico

ARQUITECTURA DE COMPUTADORAS Y ENSAMBLADORES

MARVIN DANIEL RODRIGUEZ FELIX

PROYECTO 2 | 201709450

Introducción

Es un programa en la cual la escuela de Ciencias y Sistemas de la facultad de ingeniería de la Universidad de San Carlos de Guatemala, ha propuesto a los estudiantes el desarrollo del juego de "galaga" el cual contara con varios niveles además de poder registrar usuarios con la finalidad de que exista un ranking del top 10 de jugadores por puntos obtenidos y mayor tiempo en el juego, contara también con un reporte de barras, donde tendrá opción de ordenamientos ya sea descendente o ascendente y velocidad de ordenamiento, se podrá registrar nuevos usuarios y el administrador podrá ascender o descender y desbloquear a usuarios.

Requerimientos

El programa se realizó en DOS Box y el lenguaje ensamblador utilizado es MASM.

- Los sistemas operativos, recomendados son: Windows 10, Windows 7, Mac con Mac OS X, Oracle Linux 10.9, Oracle Linux 6.x (32 bits), 6.x (64 bits)
- En este caso se Utilizó Windows 10 Home, procesador Intel(R) Core (TM) i5-4200 y 8 GB de memoria RAM.

Conceptos de Ensamblador MASM:

1. Macros:

Es un conjunto de instrucciones la cual funciona como método o funciones, las macros permiten la automatización de tareas repetitivas, permite un código de programación más compacto, flexibilidad en la programación al permitir uso de parámetros, la desventaja es que el ejecutable se vuelve más grande con cada llamada de macros.

2. Etiquetas:

Las etiquetas son variables dentro del código de programación, donde son llamados para realizar instrucciones, también son utilizados como solución de algunos registros de datos condicionales, ayuda para poder dar una solución a la misma, también es útil para tener un mejor control de lo que estemos programando.

3. Int 21h:

Está compuesta por un grupo de funciones, cuando se accede a la interrupción se debe de indicar el número de función que queremos ejecutar, se le denomina DOS-API: DOS- APPLICATION-PROGRAM-INTERFAC, debido a que la mayoría de las funciones de servicios o funciones del sistema operativo más-dos se obtienen de la interrupción 21h.

Registro de datos:

El CPU tiene 14 registros internos básicos, algunos son realmente de 32 bits, pero por ahora se utiliza el modo real que es compatible con el procesador 8086, algunos registros:

- Registros de uso General: AX acumulador, BX registro base, CX Registro contador y DX registro de datos
- Registros de segmento: DS Registro segmento de datos, ES Registro segmento de datos extra, SS Registro segmento de datos de pila y CS Registro segmento de datos de código.
- Registros Punteros: BP registro de apuntadores base, SI registro índice fuente y DI registro índice destino.

Saltos Incondicionales y Condicionales:

Es el cambio de flujo de un programa en forma incondicional o bajo una condición, lo que realiza en el programa es que salta a un determinado punto si se cumple la condición, algunos saltos son:

- JMP:
  En un salto incondicional, debido a que donde es invocado realiza un salto de líneas de código hacia el lugar que se le indique.
- JE:
  Salta si está prendiendo el bit Zero del registro de banderas, salta si la última comparación realizada da igual.
- JA o JNBE:
  Salta el bit carry CF o el bit Zero ZF del registro bandera esta desactivado, salta si la última comparación realizada con números naturales da mayor.
- JB o JNAE:
  Salta si el bit carry CF esta activa, salta si la última comparación realizada con números naturales da menor
- JNE
  Salta si está prendiendo el bit Zero del registro de banderas, salta si la última comparación realizada no da igual.
- LOOP:
  un salto condicional que no necesita la instrucción CMP, decrementa CX en 1 y salta si CX es distinto de cero y ZF este prendido.

Código de la aplicación

- **Todas las variables declaradas**

```
.model small
.stack
.data
    encabezado db "Universidad de San Carlos de Guatemala",10, 13,
"Facultad de Ingenieria",10, 13, "Escuela de Ciencias y Sistemas",10, 13,
"Arquitectura de Compiladores y ensambladores 1",10, 13, "Seccion A", 10,
13, "Marvin Daniel Rodriguez Felix", 10, 13, "201709450", 10, 13, "$"
    menuP db 0ah, 0dh, 'MENU', 0ah, 0dh, '$'
    opciones db 0ah, 0dh, 'F1. Iniciar Sesion', 0ah, 0dh, 'F5.
Registrar', 0ah, 0dh, 'F9. Salir', 0ah, 0dh, '$'
    salto db 10, 13, "$"
    var db 0, '$'
    msgErrorGeneral db 0ah,0dh,'Se ha cometido algun error de
archivo','$'

    ;============ MENU PRINCIPAL =============

    msglogin db 0ah, 0dh, 'INICIAR SESION', 0ah, 0dh, '$'
    msgLine db '=============================================', 0ah,
0dh, '$'
    msgRegister db 0ah, 0dh, 'REGISTRO', 0ah, 0dh, '$'

    msgUsername db 0ah, 0dh, 'Username: ', '$'
    msgPassword db 0ah, 0dh, 'Password: ', '$'

    bufferU db 30 dup('$'), '$'
    bufferP db 30 dup('$'), '$'

    contadorSI dw 0, '$'

    ArchivoInformacion db 250 dup('$')
    ArchivoRutaUsuarios db 'users.gal', 00h, '$'
    ArchivoRutaReporte db 'lastsort.rep', 00h, '$'
    ArchivoHandler dw ?, '$'

    users db 1000 dup('$'),'$'

    msgNoExisteUser db 0ah, 0dh, '>>  El Usuario no esta registrado',
0ah, 0dh, '$'
    msgPassError db 0ah, 0dh, '>>  Password incorrecta', 0ah, 0dh, '$'

    msgAdminLogin db 0ah, 0dh, '>>  Bienvenido Admin General', 0ah, 0dh,
'$'
    msgAdmin1 db 0ah, 0dh, '>>  Bienvenido Admin ', 0ah, 0dh, '$'
    msgLoginCorrecto db 0ah, 0dh, '>>  Bienvenido Usuario ', 0ah, 0dh,
'$'

    msgUsuarioBloqueado db 0ah, 0dh, '>>  Usuario Bloqueado, Contacte con
Administrador', 0ah, 0dh, '$'
    msg3intentos db 0ah, 0dh, '>>  Hubo 3 intentos fallidos', '$'
```

```
    msgAdminPassE db 0ah, 0dh, '>> Porfavor espere: ', '$'
    msgAdminPassE2 db 's e intente nuevamente', 0ah, 0dh,  '$'


    msgValidaciones db 0ah, 0dh, '>>                        Accion
Rechazada                                <<', '$'
    msgValidacion   db 0ah, 0dh, '>>
<<', '$'
    msgRequisito    db 0ah, 0dh, '>> Requisitos olvidados:
<<', '$'

    msgVU1          db 0ah, 0dh, '>> No se puede empezar por numero el
usuario                          <<', '$'
    msgVU2          db 0ah, 0dh, '>> El usuario tiene que tener entre 8 y
15 caracteres                  <<', '$'
    msgVU3          db 0ah, 0dh, '>> El usuario ya existe
<<', '$'
    msgVU4          db 0ah, 0dh, '>> El usuario solo puede contener (-
)(_)(.) como caracter especial    <<', '$'

    msgP1           db 0ah, 0dh, '>> La contrasenia debe de tener al
menos una mayuscula                <<', '$'
    msgP2           db 0ah, 0dh, '>> La contrasenia debe de tener al
menos un numero                    <<', '$'
    msgP3           db 0ah, 0dh, '>> Se debe de tener al menos un
caracter especial: (!)(>)(%)(;)(*)    <<', '$'
    msgP4           db 0ah, 0dh, '>> La contrasenia debe contener entre
16 y 20 caracteres              <<', '$'

    banderaU1 db 00h, '$'
    banderaU2 db 00h, '$'
    banderaU3 db 00h, '$'
    banderaU4 db 00h, '$'

    banderaP1 db 00h, '$'
    banderaP2 db 00h, '$'
    banderaP3 db 00h, '$'
    banderaP4 db 00h, '$'

    banderaG db 00h, '$'
    conteo db 0, '$'

    msgRegistroCorrecto db 0ah, 0dh, '>>    Usuario registado
correctamente ', 0ah, 0dh, '$'

    intentos db 0, '$'
    guardarDI dw 0, '$'
    temp db 0, '$'

    ;================== MENU USUARIO ==============
    msgUsermenu db 0ah, 0dh, 'MENU DE USUARIO      USER: ', '$'
    msgOpcionesUser db 0ah, 0dh, 'F2. Jugar ', 0ah, 0dh, 'F3. Mostrar el
marcador de los 10 primeros', 0ah, 0dh, 'F5. Mostrar mi marcador de los
10 primeros', 0ah, 0dh, 'F9. Cerrar Sesion', 0ah, 0dh, '$'

    ;==================MENU ADMIN==================
```

```
    msgAdminMenu db 0ah, 0dh, 'MENU ADMINISTRADOR      USER: ', '$'
    msgOpcionesAG db 0ah, 0dh, 'F1. Debloquear Usuario ', 0ah, 0dh, 'F2.
Ascender usuario a admin', 0ah, 0dh, 'F3. Descender admin a usuario',
0ah, 0dh, 'F5. Ordenamiento de burbuja', 0ah, 0dh, 'F6. Ordenamiento por
monticulos', 0ah, 0dh, 'F7. TimSort', 0ah, 0dh, 'F9. Cerrar Sesion', 0ah,
0dh, '$'
    msgOpcionesA db 0ah, 0dh, 'F1. Debloquear Usuario ', 0ah, 0dh, 'F2.
Mostrar el marcador de los 10 primeros', 0ah, 0dh, 'F3. Mostrar mi
marcador de los 10 primeros', 0ah, 0dh, 'F4. Jugar', 0ah, 0dh, 'F5.
Ordenamiento de burbuja', 0ah, 0dh, 'F6. Ordenamiento por monticulos',
0ah, 0dh, 'F7. TimSort', 0ah, 0dh, 'F9. Cerrar Sesion', 0ah, 0dh, '$'

    msgCerrarSession db 0ah, 0dh, '>> Sesion Cerrada con exito.... ',
0ah, 0dh, '$'


    msgAdminUnlock db 0ah, 0dh, 'Desbloquear Usuario    USER: ', '$'
    msgUnlock db 0ah, 0dh, 'Usuario a desbloquear: ', '$'

    msgUnlock1 db 0ah, 0dh, '>> Usuario desbloqueado correctamente
<<', '$'
    msgUnlock2 db 0ah, 0dh, '>> ERROR, El Usuario no estaba bloqueado
<<', '$'

    bufferUnlock db 30 dup('$'), '$'
    bufferBan db 30 dup('$'), '$'

    msgAscender  db 0ah, 0dh, 'Ascender a Admin    USER: ', '$'
    msgAscender1 db 0ah, 0dh, 'Usuario para Ascender: ', '$'

    msgAscender2 db 0ah, 0dh, '>> Usuario Ascendido correctamente
<<', '$'
    msgAscender3 db 0ah, 0dh, '>> ERROR, El Usuario ya es Admin
<<', '$'

    msgDescender  db 0ah, 0dh, 'Descender a Usuario    USER: ', '$'
    msgDescender1 db 0ah, 0dh, 'Usuario a descender: ', '$'

    msgDescender2 db 0ah, 0dh, '>> Usuario Desendido correctamente
<<', '$'
    msgDescender3 db 0ah, 0dh, '>> ERROR, El Usuario no es Admin
<<', '$'

    cont db 0, '$'
    resultado db 5 dup('$')

    ;================JUEGO==============
    msgjuego1 db '-Usuario:', '$'
    msgjuego2 db '-Nivel:', '$'
    msgjuego3 db '-Punteo:', '$'
    msgjuego4 db '-Tiempo:', '$'
    msgjuego5 db '-Vidas:', '$'


    level db 49, '$'
    score db 48,48,48, '$'
    time db 48,48,58,48,48,58,48,48, '$'
```

```
lives dw 3, '$'

auxcora  dw 0, '$'
auxcoro1 dw 0, '$'
auxcoro2 dw 0, '$'

contador1 dw 0, '$'
contador2 dw 0, '$'

auxnave dw 0, '$'
auxnave2 dw 0, '$'
contador3 dw 0, '$'
contador4 dw 0, '$'
contador5 dw 0, '$'

nx dw 0, '$'
ny dw 0, '$'

cIzqx dw 0, '$'
cIzqy dw 0, '$'
cCenx dw 0, '$'
cCeny dw 0, '$'
cDerx dw 0, '$'
cDery dw 0, '$'

contaux dw 0, '$'
contaux1 dw 0, '$'
contaux2 dw 0, '$'
oaux dw 0, '$'
oaux2 dw 0, '$'
oaux3 dw 0, '$'

banderaTerminaJuego db 00h, '$'

msg_esc1   db 'Esc Pausa', '$'
msg_esc2   db 'Esc Salir', '$'
msg_start1 db 'Spa Empieza', '$'
limpia1    db '           ', '$'
msg_start2 db 'Spa Continua', '$'
finjuego   db 'Fin Juego', '$'

enemigos db 126 dup('$'), '$'

posicion db 126 dup('$'), '$'

otravez dw 0, '$'

cote1 dw 0, '$'
cote2 dw 0, '$'

suplente dw 0, '$'
siaux dw 0, '$'

balas db 9 dup('$'), '$'

auxbala dw 0, '$'
auxbala1 dw 0, '$'
```

```
    auxbala2 dw 0, '$'

    ab dw 0, '$'
    auxcolision  dw 0, '$'
    auxcolision2 dw 0, '$'

    decsuma dw 0, '$'

    auxvida dw 0, '$'
    templlenado db 0, '$'
    retardo dw 0, '$'

    ;==================reporte scores==================
    ArchivoRutaScore db 'scores.gal', 00h, '$'
    ArchivoHandlerScore dw ?, '$'
    scores db 1000 dup('$'),'$'
    scoresOrdenados db 1000 dup('$'),'$'

    msgMarcador1 db 0ah, 0dh, 'Top 10 Marcador
USER: ', '$'
    msgLine2     db
'============================================================', 0ah,
0dh, '$'
    msgMarcador11 db 0ah, 0dh, 'Mi Top 10 Marcador
USER: ', '$'
    msgMarcador2 db 0ah, 0dh, 'Rango     Jugador              N    Puntos
Tiempo ', 0ah, 0dh, '$'

    msgline_     db 0ah, 0dh,
'____________________________________________________________', 0ah,
0dh, '$'

    msgenterCo   db 0ah, 0dh, '>>             PRESIONA ENTER PARA VOLVER
A MENU           <<', 0ah, 0dh, '$'

    auxaux db 50 dup('$'), '$'

    i db 0, '$'
    j db 0, '$'

    conteo_scores dw 0, '$'
    cont_caracteres dw 0 ,'$'

    conteo_auxiliar dw 0, '$'
    guardar_inicio_i dw 0, '$'
    guardar_inicio_j dw 0, '$'

    uaxb  db 0,0,0, '$'
    auxw  dw 0, '$'
    auxwi dw 0, '$'
    auxwj dw 0, '$'

    sd db 0, '$'
    caracterP db 0, '$'
    rank db 48, '$'
    sangria1 db 32,32,32,32,32,32,32,32,32, '$'
    sangria2 db 32,32,32,32,32, '$'
```

```
    ;================ordenamientos===============
    msg01 db 0ah, 0dh, 'Ordenamiento de burbuja                USER:', '$'
    msg02 db 0ah, 0dh, 'Ordenamiento por monticulos            USER:', '$'
    msg03 db 0ah, 0dh, 'TimSort                                USER:', '$'


    msgsubMenu  db 0ah, 0dh, 'F1. Ascendente', 0ah, 0dh, 'F2.
Descendente', 0ah, 0dh, 'F9. Regresar', 0ah, 0dh, '$'

    msgSubMenu2 db 0ah, 0dh, 'F1. Puntos', 0ah, 0dh, 'F2. Tiempo', 0ah,
0dh, 'F9. Regresar', 0ah, 0dh, '$'

    msgSubMenu3 db 0ah, 0dh, 'F1. 0', 0ah, 0dh, 'F2. 1', 0ah, 0dh, 'F3.
2', 0ah, 0dh,'F4. 3', 0ah, 0dh,'F5. 4', 0ah, 0dh,'F6. 5', 0ah, 0dh,'F7.
6', 0ah, 0dh,'F8. 7', 0ah, 0dh, 'F9. Regresar', 0ah, 0dh, '$'

    banderaPuntos db 00h, '$'
    banderaTiempo db 00h, '$'
    banderaAscendente db 00h, '$'
    banderaDescendente db 00h, '$'

    velocidad dw 0, '$'
    veloci db 0, '$'

    punteos db 100 dup('$'), '$'
    tiempos dw 55 dup('$'), '$'
    temporall db 0,0,0, '$'
    temporatt dw ?, '$'

    obtengo dw 48,48,48, '$'
    obte db 0, '$'
    pe db 0, '$'

    teste db 10 dup('$'), '$'
    auxiliar_burbuja1 db 0, '$'
    auxiliar_burbuja2 db 0, '$'

    buble db 'Burbuja', '$'
    vel db 'Velocidad:', '$'
    pun db 'Punteo', '$'
    home db 'Presionar HOME para iniciar', '$'
    endd db 'Presionar  END  para  salir', '$'

    posicion_grafica db 0, '$'
    posicion_pixeles dw 0, '$'
    posicio_div db 0, '$'
    posicion_final dw 0, '$'

    banderaTerminaOrdenamiento db 00h, '$'

    arreglo_limpiar db 118 dup('$'), '$'
    limpa db 20h, '$'
    ll db 0, '$'
    iax dw 0, '$'
    jax dw 0, '$'
    orp db 0, '$'
```

```
    ;==========reporte==========

    nombre db '           Marvin Daniel Rodriguez Felix', 0ah, 0dh, '$'
    carnet db '                      201709450', 0ah, 0dh, '$'
    encabezado2 db "Universidad de San Carlos de Guatemala",10, 13,
"Facultad de Ingenieria",10, 13, "Escuela de Ciencias y Sistemas",10, 13,
"Arquitectura de Compiladores y ensambladores 1",10, 13, "Seccion A", 10,
13, "$"
    lineg db  '-----------------------------------------------', 0ah,
0dh, '$'

    msgtipo db 'Tipo: ', '$'
    msgsentido db 'Sentido: ', '$'
    mshfecha db 'Fecha: ', '$'
    msghora db 'Hora: ', '$'
    msgAscendente db 'Ascendente', 0ah, 0dh,  '$'
    msgDescendente db 'Descendente', 0ah, 0dh, '$'

    Hora db 8 dup('$'),'$'
    Fecha db 5 dup('$'),'$'
    Anio db '/2022', '$'
```

- **En este macro utilizamos para solicitar credenciales**

```
Credenciales MACRO principio
Local pord, iniuser, user, inipass, pass, sale
    pord:
        mov al, principio
        cmp al, 01h
        je inipass
    ;=============incio preguntando usuario==================
    iniuser:
        LimpiarArreglo bufferU
        print msgUsername
        xor si, si
    user:
        getChar
        cmp al, 13
        je inipass
        mov bufferU[si], al
        inc si
        jmp user
    ;==================preguntando password==================
    inipass:
        ;print bufferU
        LimpiarArreglo bufferP
        print msgPassword
        xor si, si
    pass:
        getpass
        cmp al, 13
        je sale
        mov bufferP[si], al
        aste
        inc si
        jmp pass
```

```
    sale:
        ;print bufferP
        print salto
ENDM
```

- **En esta macro la utilizo para verificar a los usuarios**

```
ComprobarUsuario MACRO
LOCAL Inicio, contador30,ur, final, u1, u2, u3, u4, u5, u6, u7, u8, u9,
u10, u11
    Inicio:
        xor di, di
        mov guardarDI, 0
        mov intentos, 0
        mov cont, 31
    ;======== compruebo el nombre de usuario ============
    u0:
        xor si, si
    u1:
        mov al, users[di]
        cmp al, 44
        je ur ; comprobar pass
        cmp al, bufferU[si]
        jne u4
        inc si
        inc di
        jmp u1
    ur:
        mov guardarDI, di
    ;==================compruebo la password del usuario==========
    u2:
        xor si, si
    u3:
        inc di
        mov al, users[di]
        cmp al, 44
        je u5
        cmp al, bufferP[si]
        jne u6
        inc si
        jmp u3
    ;============== busco el siguiente usuario ============
    u4:
        mov al, users[di]
        inc di
```

```
        cmp al, 13
        je u0
        cmp al, 36
        je u7
        jmp u4
    ;==============coinciden las credenciales============
    u5:
        inc di
        mov al, users[di]
        cmp al, 49 ; si el usuario esta bloqueado
        je u8
        inc di
        inc di
        mov al, users[di]
        cmp al, 48 ; si es usuario normal
        je u9
        cmp al, 49; si es admin
        je u10
        cmp al, 50; si es admin general
        je u11
        jmp final
    ;============existe el usuario pero no esta correcta la
password=========
    u6:
        limpiar
        print msgPassError
        print salto
        add intentos, 1
        mov al, intentos

        xor di, di
        mov di, guardarDI
        cmp al, 3
        je ure
        jmp u6c
    ;========= verifico que tipo de usuario es el que fallo 3 veces
=======
    ure:
        inc di
        mov al, users[di]
        cmp al, 13
        je eh

        jmp ure
    eh:
```

```
        dec di
        mov al, users[di]
        cmp al, 48
        je normal
        jne contador30
    ; ================ si es admin solo hago un delay de 30s==========
    contador30:
        sub cont, 1
        mov al, cont
        mov bl,al
        mov ax,bx
        ConvertirString resultado
        Delay2 1000
        limpiar
        cmp cont, 0
        je ee
        print msg3intentos
        print msgAdminPassE
        print resultado
        print msgAdminPassE2
        jmp contador30
    ee:
        jmp u6c
    ; ============= si es normal bloqueo el usuario en el
arreglo==========
    normal:
        print msg3intentos
        xor di, di
        mov di, guardarDI
    cloop:
        inc di
        mov al, users[di]
        cmp al, 44
        je cambiar
        jmp cloop
    ; ================ escribo en el archivo de users.gal==========
    cambiar:
        inc di
        mov users[di], 49
        ActualizarUsers
        jmp u8
    ; ==============continua con el intento de ingresar la pass========
    u6c:
        print msglogin
        print msgLine
```

```
        print msgUsername
        print bufferU
        print salto
        Credenciales 01h
        xor di, di
        mov di, guardarDI
        jmp u2
    ;================no existe el usuario que ingreso==============
    u7:
        print msgNoExisteUser
        getChar
        limpiar
        jmp final
    ;================ el usuario esta bloqueado ========
    u8:
        print msgUsuarioBloqueado
        getChar
        limpiar
        jmp final
    ;==============Ingreso al menu del Usuario=========
    u9:
        limpiar
        MenUsuario
        jmp final
    ;==============Ingreso al menu de Admin ==============
    u10:
        limpiar
        MenuAdmin
        jmp final
    ;============Ingreso al menu de Admin General==========
    u11:
        limpiar
        MenuAdminGeneral
        jmp final
    final:
        print salto
ENDM
```

- **En estas macros pinto las figuras del juego**

```
PintarNave MACRO x, abajo, color
LOCAL e2, ciclo1, e1
    push si
    push cx

    xor cx, cx
```

```
    mov cx, abajo
    mov contador4, cx

    xor si, si
    mov si, x

    mov auxnave2, si

    mov auxnave, si
    add auxnave, 15

    mov contador3, 0
    mov contador5, 0

    e2:
        inc contador3
        e1:
            inc contador5
            mov si, auxnave2
            ciclo1:
                pintar_pixel contador4, si, color
                inc si
                cmp si, auxnave
            jne ciclo1
            dec contador4
            cmp contador5, 3
        jne e1

        mov contador5, 0

        add auxnave2, 2
        mov si, auxnave2

        sub auxnave, 2

        cmp contador3, 4
    jne e2

    pop cx
    pop si
ENDM

PintarCanon MACRO x, abajo, despintar
LOCAL e0, e1
    push si
    push cx

    xor cx, cx
    mov cx, abajo
    mov auxnave2, cx

    mov al, despintar

    cmp al, 01h
    je e0

    pintar_pixel auxnave2, x, 9d
```

```
    dec auxnave2
    pintar_pixel auxnave2, x, 7d
    dec auxnave2
    pintar_pixel auxnave2, x, 7d
    dec auxnave2
    pintar_pixel auxnave2, x, 6d
    dec auxnave2

    jmp e1

    e0:
        pintar_pixel auxnave2, x, 0d
        dec auxnave2
        pintar_pixel auxnave2, x, 0d
        dec auxnave2
        pintar_pixel auxnave2, x, 0d
        dec auxnave2
        pintar_pixel auxnave2, x, 0d
        dec auxnave2


    e1:
    pop cx
    pop si

ENDM

Dibujar_Bala MACRO x, y, color
LOCAL e0, e1
    push cx

    xor cx, cx
    mov cx, y
    mov auxbala, cx

    pintar_pixel auxbala, x, color
    dec auxbala
    pintar_pixel auxbala, x, color
    dec auxbala
    pintar_pixel auxbala, x, color

    pop cx
ENDM


Dibujar_enemigo MACRO x, y, color
LOCAL e2, ciclo1, e1
    push si
    push cx
    xor cx, cx
    mov cx, y
    mov contador4, cx
    xor si, si
    mov si, x
    mov auxnave2, si
    mov auxnave, si
    add auxnave, 8
```

```
    mov contador3, 0
    mov contador5, 0
    e2:
        inc contador3
        e1:
            inc contador5
            mov si, auxnave2
            ciclo1:
                pintar_pixel contador4, si, color
                inc si
                cmp si, auxnave
            jne ciclo1
            inc contador4
            cmp contador5, 2
        jne e1
        mov contador5, 0
        add auxnave2, 1
        mov si, auxnave2
        sub auxnave, 1
        cmp contador3, 4
    jne e2
    xor cx, cx
    xor si, si
    mov cx, y
    mov oaux, cx
    mov si, x
    mov oaux2, si
    add oaux, 2
    add oaux2, 2
    pintar_pixel oaux, oaux2, 0d
    add oaux2, 1
    pintar_pixel oaux, oaux2, color
    add oaux2, 1
    pintar_pixel oaux, oaux2, color
    add oaux, 1
    mov oaux2, si
    add oaux2, 2
    pintar_pixel oaux, oaux2, 0d
    add oaux2, 1
    pintar_pixel oaux, oaux2, color
    add oaux2, 1
    pintar_pixel oaux, oaux2, color
    add oaux, 2
    mov oaux2, si
    add oaux2, 2
    pintar_pixel oaux, oaux2, color
    add oaux2, 2
    pintar_pixel oaux, oaux2, 0d
    add oaux2, 1
    pintar_pixel oaux, oaux2, color
    pop cx
    pop si
ENDM
```

- **En esta macro esta mi ordenamiento burbuja, para la parte grafica**

```
Ordena_Burbuja MACRO
Local for1, for2, regfor1, regfor2, res, h, desc
    push ax
    push bx
    xor bx, bx
    xor ax, ax
    mov i, 0
    mov j, 0
    mov auxiliar_burbuja1, 0
    mov auxiliar_burbuja2, 0
    for1:
        mov al, i
        cmp ax, conteo_scores
        jae res ; mayor o igual

        add al, 1
        mov j, al
        for2:
            mov al, j
            cmp ax, conteo_scores
            jae regfor1 ; mayor o igual

            mov bl, i
            mov al, punteos[bx]
            mov auxiliar_burbuja1, al

            mov bl, j
            mov bl, punteos[bx]
            mov auxiliar_burbuja2, bl

            cmp orp, 01h
            je desc

            cmp al, bl
            jbe regfor2
            jmp h

            desc:
            cmp al, bl
            jae regfor2

            h:
            push ax
            push bx
            LimpiarPantalla
            pintar_S_linea 35, 13d, 180d, 8
            pintar_linea_ho 12d, 180d, 3d
            xor ax, ax
            xor bx, bx
            mov bl, i
            mov al, j
            mov iax, bx
            mov jax, ax
            PantallaArreglo iax, jax
```

```
            delay velocidad
            pop bx
            pop ax

            mov bl, i
            mov al, auxiliar_burbuja2 ; i = j
            mov punteos[bx], al

            mov bl, j
            mov al, auxiliar_burbuja1
            mov punteos[bx], al

            push ax
            push bx
            LimpiarPantalla
            pintar_S_linea 35, 13d, 180d, 8
            pintar_linea_ho 12d, 180d, 3d
            xor ax, ax
            xor bx, bx
            mov bl, i
            mov al, j
            mov iax, bx
            mov jax, ax
            PantallaArreglo iax, jax
            delay velocidad
            pop bx
            pop ax

            regfor2:
                mov al, j
                add al, 1
                mov j, al
        jmp for2

        regfor1:
            mov al, i
            add al, 1
            mov i, al
    jmp for1

    res:

    LimpiarPantalla
    pintar_S_linea 35, 13d, 180d, 8
    pintar_linea_ho 12d, 180d, 3d
    PantallaArreglo 50, 50
    ImprimirPantalla 23, 46, endd
    e1:
        getTecla
        cmp ah, 4fh
    jne e1


    pop bx
    pop ax
ENDM
```