

# **TRABAJO COMPUTADORES II**

## **CONECTA 4**

Andrés Fernández Sánchez **DNI: 70901956V**  
Luis Blázquez Miñambres **DNI:70910465Q**

# **\*\*\*Índice\*\*\***

- 1. Comando de ensamblaje**
- 2. Función de los ficheros**
- 3. Representación gráfica del juego**

**\*\*\*Índice\*\*\***

# Comando de ensamblaje

Para el enlazado y ensamblaje de nuestro programa, la orden de ensamblaje seria:

ensambla main cadenas conecta tablero menu\_instrucciones menu\_principal tituloASCII

```
$ cd MODULOS
$ cd Escritorio
$ cd Escritorio
/Escritorio$ cd MODULOS
/Escritorio/MODULOS$ ensambla main cadenas conecta tablero menu_instrucciones menu_principal tituloASCII
```

## Ficheros

En total hemos hecho uso de 7 ficheros, cada uno con su correspondiente función:

### ★ main.asm

- **Función:** Fichero principal del programa que contiene el salto a las subrutinas que desarrollan el juego y contiene las siguientes funciones:
  - **Turno\_jugador** → indica el turno del jugador al que le toca el turno correspondiente en cada iteración.
    - **bucle\_turno** → En cada turno indica al jugador si desea salir de la partida introduciendo una 's' o 'S'
    - **continuar\_turno** → Indica a qué jugador le toca introducir la ficha en cada momento.
    - **empate** → saca un mensaje indicando empate en caso de que uno de los dos jugadores haya alcanzado 21 turnos(es decir que todas las columnas están llenas(21 fichas x 2 jugadores=42 posiciones)).

- **pedir\_columna** → Pide de teclado la columna al jugador correspondiente y la pasa a decimal guardando la columna en "pos\_ficha".
- **comprobacion\_rango** → realiza las comprobaciones oportunas para asegurar que el jugador introduzca la ficha de su turno en una de las posibles columnas (1-7). En caso contrario se indicará con un mensaje y se volverá a pedir que introduzca una columna correcta.
- **pos\_erronea** → indica si la columna introducida es errónea.
- **guardar\_ficha** → carga el vector\_comprobacion que contiene 7 bytes en orden representando las 7 columnas inicializados a 6(número de posiciones libres de cada columna o número de filas del tablero). Por cada ficha introducida en una columna se decrementa ese byte y si uno de los 7 byte llega a 0 significa que la columna que corresponde a ese byte está llena. El valor de ese byte se guarda en "pos\_ocupada" que será el número de filas que hay que saltar hasta llegar a la posición deseada.
- **columna\_llena** → saca un mensaje por pantalla en caso de que se llene una columna.
- **bucle\_ficha** → itera tantas veces como columnas tenga que saltar(contador "pos\_ocupada") y va sumando 7 a b en representación de posiciones por fila.
- **salir\_bucle\_ficha** → suma el número de filas saltadas al número de posiciones que se han saltado en B y te da la posición efectiva donde se introducirá la ficha y se saltará a la subrutinas de comprobaciones e imprimir tablero.

### SIMULACIÓN FICHERO

Introducir columna: 5

Vector comprobación: byte 6 byte 6 byte 6 byte 6 **byte 6** byte 6 byte 6

Vector comprobación: byte 6 byte 6 byte 6 byte 6 **byte 5** byte 6 byte 6

1      2      3      4      5      6      7

pos\_ocupada=5

Bucle:  $7+7+7+7+7=35+5=40$

decb (porque son  $n-1$  posiciones en el tablero)

posición efectiva=39

[illegible]

```
* 0 1 2 3 4 5 6*
* 7 8 9 10 11 12 13*
*14 15 16 17 18 19 20*
*21 22 23 24 25 26 27*
*28 29 30 31 32 33 34*
*35 36 37 38 39 40 41*
*****
*1a | 2a | 3a | 4a | 5a | 6a |
;
```

- **Pasos que realiza el módulo:**

1. Imprime el título del juego a través del fichero “*títuloASCII*”.
2. Muestra el menú a través del fichero “*menú\_principal*”.
3. Dependiendo de la opción que se teclee podremos, por un lado, empezar el juego, por otro, mostrar las instrucciones a través del fichero “*menu\_instrucciones*”, y por otro, salir del juego.
4. Comprueba, por un lado, los turnos del jugador, por otro, en qué columna se introduce la ficha y si esta posición es válida o no, y, por último si hay un empate.
5. Guarda la ficha en el tablero.
6. Imprime el tablero por pantalla a través del fichero “*tablero*”.
7. Comprueba si hay un ganador a través del fichero “*conecta*”.

---

### ★ menu\_Instrucciones.asm

- **Función:** Contiene la subrutina que realiza la visualización por pantalla de las instrucciones del juego al introducir la opción 2) en el menú del juego.
- **Pasos que realiza el módulo:**
  1. Carga en el registro “X” el texto.
  2. Llama al fichero que contiene la subrutina “imprimir\_mensaje”.

---

### ★ cadenas.asm

- **Función:** Contiene la subrutina que imprime cualquier cadena ASCII que se le pase a través del registro X y la muestra por pantalla.
- **Pasos que realiza el módulo:**
  1. Inicia un bucle cargando en el registro “a” el contenido del registro x y éste se incrementará para la siguiente iteración.
  2. Si el contenido del registro “x” equivale a “o” saldrá del bucle y terminará la función.
  3. Sino imprimirá por pantalla el carácter almacenado en el registro “a”.
  4. Retorna al inicio del bucle.

---

### ★ menu\_principal.asm

- **Función:** Contiene la subrutina que imprime por pantalla el menú principal del juego y el mensaje que le indica al usuario a que opción quiere acceder en el menú.
- **Pasos que realiza el módulo:**
  1. Carga en el registro “x” el texto.
  2. Llama al fichero que contiene la subrutina “imprimir\_mensaje”.

---

### ★ tituloASCII.asm

- **Función:** Contiene la subrutina que imprime por pantalla el título que da nombre al juego a través de varias cadenas ASCII.
  - **Pasos que realiza el módulo:**
    1. Carga en el registro “x” el texto.
    2. Llama al fichero que contiene la subrutina “imprimir\_mensaje”.
- 

### ★ conecta.asm

- **Función:** Fichero que contiene la subrutina que realiza las comprobaciones principales del juego, vertical ,diagonal y horizontal, así como en caso de que se diera alguno de los 3 casos, sacar por pantalla un mensaje indicando el ganador de la partida.Las funciones horizontal y diagonal siguen la siguiente estructura:

```

.....
a
ima posicion introducida= X
Horizontal      Diagonal Izquierda      Diagonal Derecha
      -                      -
      - X -          - X          X -
      -                      -
.....

```

La horizontal y diagonal llevan ambas un contador y los recorridos se llevan a cabo desde la posición de la última ficha que se introdujo.

**La horizontal** empieza recorriendo posiciones hacia delante y si no encuentra 3 fichas sin contar la última que se ha introducido vuelve a la última posición introducida y recorre hacia atrás con el contador intacto.

En el caso de **la diagonal** recorre hacia un lado y hacia el otro de manera similar, primero recorre las posiciones hacia arriba a la izquierda y derecha con un contador y si no encuentra

otras 3 fichas del mismo tipo vuelve a la posición inicial y recorre las posiciones hacia abajo en diagonal.

En ambos casos, al ser las posiciones del tablero posiciones contiguas le daba la victoria al jugador de la horizontal nada más detectaba 4 fichas en horizontal aunque estuvieran en distintas filas. Para ello hemos implementado un contador que en el momento en el que detecte que no hay 4 fichas contiguas (en la misma fila) pone el contador de la comprobación a 0 y sale del bucle.

- **Pasos que realiza el módulo:**
  1. Comprueba el turno del jugador.
  2. Verifica si hay conectado 4 fichas en vertical, horizontal y diagonal.
  3. En caso de ser cierto termina el juego.
  4. Sino, seguirá el turno del otro contrincante.

---

### ★ tablero.asm

- **Función:** Fichero que contiene la subrutina que saca por pantalla el contorno del tablero, así como el contenido del mismo en cada turno que se realiza.
- **Pasos que realiza el módulo:**
  1. Presenta por pantalla la tabla de fila en fila imprimiendo:
    - a. Primero, un “\*” simulando el contorno del tablero.
    - b. Después, el interior del tablero con las correspondientes fichas y huecos de cada una de las 6 filas.
    - c. Por último, otro “\*” para terminar el contorno de una fila.
  3. (Lo anterior se repite 6 veces por haber 6 filas)
  4. Imprime la base del tablero con “\*”.
  5. Imprime los números de las columnas debajo de la base.



# Representación gráfica del juego

```

/$$$$$$ /$$$$$ /$$ /$$ /$$$$$$$ /$$$$$ /$$$$$$$ /$$$$$ /$$ /$$
/$$ _ $$ /$$ _ $$ $$$ | $$ $$$ _ /$$ _ $$ _ $$ _ /$$ _ $$
| $$ \_ / | $$ \_ | $$$ | $$$ | $$$ \_ / | $$ \_ | $$ \_ | $$$ \_ | | | | | | | | | | | | | | | | | | | | |
| $$ | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| $$ | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| $$$ $ | | | | | | | | | | | | | | | | | | | | | | | | | |
| $$$$$$ / | $$$$$$ / | | | | | | | | | | | | | | | | | | |
\_ \_ / \_ \_ / \_ / \_ / \_ \_ / \_ \_ / \_ \_ / \_ \_ / \_ \_ /

MENU PRINCIPAL DEL JUEGO
1) Jugar
2) Instrucciones
3) Salir

Seleccione una opcion:

```

```

MENU PRINCIPAL DEL JUEGO
1) Jugar
2) Instrucciones
3) Salir

Seleccione una opcion:
2
OBJETIVO: Alinear cuatro fichas en vertical/diagonal u horizontal
JUGADORES: 2 con 21 fichas X y O
COLUMNA: se introducirán las fichas en las columnas que se quieran:
-----> No se puede introducir una ficha en una columna llena
-----> Caerá en la posición más baja
-----> Si todas las columnas están llenas significará EMPATE

```

```

Jugador 1 (O):
1
* _ _ _ _ *
* _ _ _ _ *
* _ _ _ _ *
* _ _ _ _ *
* _ _ _ _ *
* O _ _ _ *
*****
*1234567*
2 Jugador 2 (X):
2
* _ _ _ _ *
* _ _ _ _ *
* _ _ _ _ *
* _ _ _ _ *
* _ _ _ _ *
* O X _ _ *
*****
*1234567*

```

```

1234567
Jugador 2 (X):
3
* _ _ _ _ *
* _ _ _ _ *
* _ X _ _ *
* O X _ _ *
* O O X _ _ *
* O X X O _ _ *
*****
*1234567*
!!ENHORABUENA Jugador 2(X)!! Has ganado la partida
main.s19 : 74189 cycles, 65847 ms
luisblazquezm@luisblazquezm-S550CM:~/Escritorio/MODULOS$

```