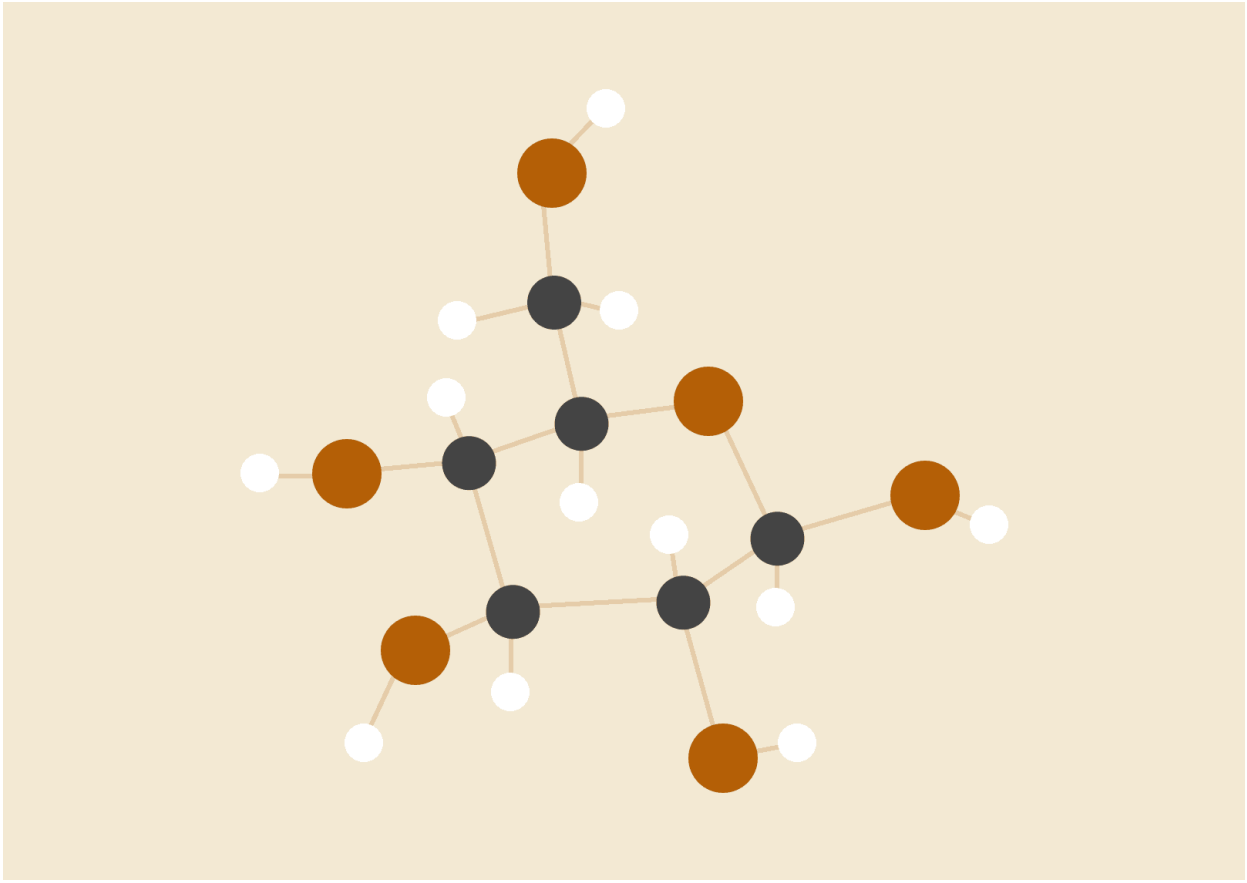
# MEMORAMA



**Pedro Alonso Díez**

16/05/2019

COMPUTADORES II, Ingeniería Informática

# INTRODUCCIÓN

En este trabajo tendremos que desarrollar el juego de Memorama en ensamblador. A continuación se describe el código y su funcionamiento. Iremos desde las funciones más básicas a las más complejas

# CÓDIGO FUENTE

Para empezar vamos a mostrar las subrutinas básicas que utilizaremos durante todo el programa, que son: “imprime_cadena”, “limpia_pantalla”, “salto_de_carro” e “imprime_tablero”.

1 - “Imprime_cadena” - Va recorriendo el registro x y mostrándolo por pantalla.

```
471 imprime_cadena:
472
473         pshs a
474
475 sgte:    lda ,x+
476         lbeq ret_imprime_cadena
477         sta pantalla
478         bra sgte
479 ret_imprime_cadena:
480         puls a
481         rts
482
```

2 - “imprime_tablero” - Primero llama a una función que muestra el marcador que veremos después y carga el tablero1 (tablero de las X) y va contando con el contador hasta 4 y entonces imprime un \n.

```
442 imprime_tablero:
443         jsr     show_marcador
444         ldy     #tablero1
445         clrb
446         ldb     #'0
447         stb     contador_char
448 sgte_c:
449         ldb     contador_char
450
451         cmpb    #'4
452         lbeq    sgte_linea
453         incb
454         stb     contador_char
455         clra
456         lda     ,y+
457         lbeq    ret_imprime_tb
458         sta     pantalla
459         bra     sgte_c
460
461 sgte_linea:
462         ldb     #'0
463         stb     contador_char
464         jsr     salto_de_carro
465         lbra    sgte_c
466
467 ret_imprime_tb:
468         rts
469
```

3 - “Limpia_pantalla” y “salto_de_carro” - Funciones que como su nombre indica limpian pantalla e imprimen un \n

```
53 ; UTILIDADES
54 clear_screen:    .asciz  "\33[2J \33[1;1H"
55 retorno_carro:   .asciz  "\n"
```

```
483 limpia_pantalla:
484
485         ldx     #clear_screen
486         jsr     imprime_cadena
487         rts
488
489 salto_de_carro:
490
491         ldx     #retorno_carro
492         jsr     imprime_cadena
493         rts
```

Lo primero que tenemos que hacer es inicializar las pilas y cargaremos el menú. A su vez tendremos la función de acabar que finalizará el programa.

```
 91 ; INICIO PROGRAMA
 92
 93 programa:
 94         ldu     #pilaU
 95         lds     #pilaS
 96         lbsr    pantalla_menu
 97
 98 acabar:
 99         clra
100         sta fin
101
```

Función pantalla_menu: Esta es la función más sencilla, lo que hará es imprimir el menú, esperar a que el usuario teclee una opción y según su opción se irá a las pantallas.

```
102 pantalla_menu:
103         ;jsr    salto_de_carro
104         jsr     limpia_pantalla
105         ldx     #show_menu
106         jsr     imprime_cadena
107
108 teclado_opcion:
109
110         ldb     teclado
111         cmpb    #'1
112         lbeq    pantalla_juego
113
114         cmpb    #'2
115         lbeq    pantalla_inst
116
117         cmpb    #'3
118         lbeq    acabar
119
120         bne     default
121
122 default:
123
124         jsr     limpia_pantalla
125         ldx     #show_opc_inc
126         jsr     imprime_cadena
127         lbra    pantalla_menu
128
```

Como vemos, la opción 3 llama a “acabar” que hemos visto antes y el programa finaliza.

La opción 2 llama a “pantalla_inst” que lo único que hace es mostrar por pantalla las instrucciones..

```
150 pantalla_inst:
151         jsr     limpia_pantalla
152
153         ldx     #show_inst
154         jsr     imprime_cadena
155         lda     teclado
156         lbra    pantalla_menu
157
158         rts
159
```

La opción 3 llama a “pantalla_de_juego” que inicializa la base del juego. Carga en el registro y el tablero con las soluciones. Imprime el tablero, y llama a pedir_coordenadas, al final compara una variable que controla los aciertos, si es igual a 8 pasa al siguiente nivel.

```
130 pantalla_juego:
131         ldx     #solucion1
132 nivel:
133         clra
134         clrb
135         ldb     #'0
136         stb     contador_par
137         jsr     limpia_pantalla
138
139         jsr     imprime_tablero
140
141         jsr     pedir_coordenadas
142
143         lda     contador_a
144         cmpa    #8
145         lbeq    nivel_superado
146
147         lbra    nivel
148
149
```

Después llamamos a “pedir_coordenadas”, que llama a las subrutinas para pedir fila y pedir columna que veremos a continuación, una vez que tenemos las coordenadas calculamos las veces que nos tenemos que desplazar para localizar las coordenadas y saber que letras cambiar, con la subrutina “calcular_desplazamiento”

```
159 pedir_coordenadas:
160
161         jsr     pedir_fila
162         jsr     pedir_col
163
164         jsr     calcular_desplazamiento
165
166         rts
167
```

Lo que hacemos aquí es pedir un número por teclado y lo comparamos para que si es menor que 1 o mayor que 4, se vuelva a solicitar la introducción del número, también comprobamos que si introduce una ‘X’ volvemos al menú. Al final tenemos que restarle 1 (con deca) y le restamos el código ascii del ‘0’ y lo guardamos en las variables “nfila” y “ncol”.

```
173 pedir_fila:
174         jsr     salto_de_carro
175         ldx     #show_pide_fila
176         jsr     imprime_cadena
177
178         lda     teclado
179
180         cmpa    #'1
181         lblo    pedir_fila
182
183         cmpa    #'X
184         lbeq    volver_menu
185
186         cmpa    #'4
187         lbhi    pedir_fila
188
189         deca
190         suba    #'0
191         sta     nfila
192         rts
193 pedir_col:
194         jsr     salto_de_carro
195
196         ldx     #show_pide_col
197         jsr     imprime_cadena
198
199         lda     teclado
200
201         cmpa    #'1
202         lblo    pedir_col
203
204         cmpa    #'X
205         lbeq    volver_menu
206
207         cmpa    #'4
208         lbhi    pedir_col
209
210         deca
211         suba    #'0
212         sta     ncol
213
214         rts
215
```

Una vez tenemos guardadas la fila y la columna ya sabemos la coordenada, por lo que tendremos que hallar la posición que ocupa en los tableros, esto se hace a través de “calcular_desplazamiento”. ((fila-1)*ancho + (col-1)). Con lsla nos desplazamos un bit hacia la izquierda, por lo que dos veces lsla es como multiplicar por 4. A este resultado le sumamos “ncol” que hemos hallado antes y lo guardamos en “aux_despla” esta variable nos sirve para movernos por el tablero que contiene las “X” ya que va de 0 a 16. Después multiplicamos “ntablero” que es el nivel en el que estamos por 16, desplazando los bit, así

ya podemos situarnos en el tablero en el que estamos, y a esto le sumamos aux_despla para ubicarnos en la columna. Guardamos en carta la letra que corresponde a la X de la coordenada y posteriormente la sustituimos en el tablero de las X, comprobando antes si está destapada o no. Para saber si estamos pidiendo la primera coordenada o la segunda utilizo un booleano que llama a métodos distintos dependiendo de su valor.

```
294 calcular_desplazamiento:
295
296         lda     nfila
297
298         lsla
299         lsla
300
301         adda    ncol
302         sta     aux_despla
303
304         lda     ntablero
305         lsla
306         lsla
307         lsla
308         lsla
309
310         adda    aux_despla
311         sta     desplazamiento
312
313         ldx     #tablero1
314         ldy     #solucion1
315
316         ldb     desplazamiento
317         lda     b,y
318         sta     carta
319
320         ldb     aux_despla
321         lda     b,x
322         cmpa    #'X
323         lbne    destapada
324
325         lda     carta
326         sta     b,x
327
328
329         lda     bool_1coor
330         cmpa    #0
331         lbeq    primera_carta
332         cmpa    #1
333         lbeq    segunda_carta
334
335         rts
336 destapada:
337         jsr     salto_de_carro
338         ldx     #show_error
339         jsr     imprime_cadena
340         ldx     #show_pulsa_t
341         jsr     imprime_cadena
```

Papelera

Los métodos de “primera_carta” y “segunda_carta” sirven para guardar las letras destapadas y poder compararlas. Como se ve, solo se va a llamar a “comparar_cartas” cuando se solicita correctamente la segunda coordenada.

```
primera_carta:
        lda     carta
        sta     carta1
        ldb     #1
        stb     bool_1coor

        ldb     #1
        stb     bool_2coor

        ldb     aux_despla
        stb     mov1
        rts

segunda_carta:
        lda     carta
        sta     carta2
        ldb     #0
        stb     bool_1coor

        ldb     #0
        stb     bool_2coor

        ldb     aux_despla
        stb     mov2
        jsr     comparar_cartas
        rts

```

El método comparar cartas, compara las cartas guardadas en la subrutina anterior y si son iguales, incrementará el número de aciertos y el número de intentos.

```
comparar_cartas:
        lda     carta1
        cmpa    carta2

        lbeq    iguales

        lda     carta1
        cmpa    carta2
        lbne    distintas

        rts

iguales:
        lda     contador_a
        inca
        sta     contador_a

        lda     nintentos
        inca
        sta     nintentos

        rts

```

Sin embargo, si son distintas, incrementará el número de intentos, esperará a que el usuario teclee una tecla para volver a cubrir las letras. Como tenemos guardada la posición de cada carta en ‘mov1’ y ‘mov2’, simplemente cargamos una ‘X’ en el registro a, y sustituimos por la ‘X’ cargada en el tablero de las X.

```
268 distintas:
269
270         lda     nintentos
271         inca
272         sta     nintentos
273
274         jsr     limpia_pantalla
275         jsr     imprime_tablero
276
277         ldx     #show_sig
278         jsr     imprime_cadena
279         lda     teclado
280
281         ldx     #tablero1
282         lda     #'X
283         ldb     mov1
284
285         sta     b,x
286
287         ldb     mov2
288         sta     b,x
289
290
291         rts
292
```

También utilicé un método para mostrar el número de intentos y el número de tablero, lo único que hay que tener en cuenta es restarle el código ascii del ‘0’ a los números para que se muestren correctamente.

```
349  show_marcador:
350         ldx     #show_intentos
351         jsr     imprime_cadena
352
353         lda     nintentos
354         adda    #'0
355         sta     pantalla
356
357         jsr     salto_de_carro
358
359         ldx     #show_tablero
360         jsr     imprime_cadena
361
362         lda     ntablero
363         adda    #'0
364         sta     pantalla
365
366         jsr     salto_de_carro
367         jsr     salto_de_carro
368
369         rts
370
```

Ahora echaremos un vistazo a las rutinas que sirven para cambiar el nivel, necesitaremos, lo primero devolver el tablero de las ‘X’ ahora descubierto por completo, a su estado inicial con todo ‘X’, también tendremos que devolver a 0 el número de intentos y aciertos, por último deberemos incrementar el número de tablero y compararlo con 8 para saber si hemos acabado el juego.

Para devolver el tablero con ‘X’ otra vez se utiliza “nivel_superado”. Lo que hace el código es cargar en a ‘X’, cargar en b un contador que empieza en 0 y se va incrementando hasta 16, para recorrer todo el tablero e ir sustituyendo por X, en el momento que el contador llegue a 16, se llamará a “menu_bra” y este llamará a “sig_nivel” que inicializa otra vez los intentos y los aciertos a 0 e incrementa el número de tablero, comparándolo con 8 para comprobar si hemos acabado el juego. Mostrará un mensaje de enhorabuena.

```
380 nivel_superado:
381         jsr     limpia_pantalla
382         jsr     imprime_tablero
383         lda     teclado
384         ldb     #0
385         ldx     #tablero1
386
387 seguir_menu:
388         lda     #'X
389         sta     b,x
390         incb
391         cmpb    #16
392         lbeq    menu_bra
393         lbra    seguir_menu
394
395 menu_bra:
396         jsr     sig_nivel
397         ldx     salto_de_carro
398         jsr     imprime_cadena
399         ldx     #show_win
400         jsr     imprime_cadena
401         lda     teclado
402         lbra    pantalla_juego
403
404 sig_nivel:
405         lda     #0
406         sta     nintentos
407
408         sta     contador_a
409
410         lda     ntablero
411         inca
412         sta     ntablero
413
414         lda     ntablero
415         cmpa    #8
416         lbeq            juego_finalizado
417
418         rts
```

Cuando se llegue al tablero número 8, se llama a “juego_finalizado” que inicializa a 0 los intentos y los aciertos y aquí también se inicializa a 0 el tablero, para que el juego vuelva a empezar.

```
420 juego_finalizado:
421
422         jsr     limpia_pantalla
423         jsr     imprime_tablero
424
425         ldx     #show_fin
426         jsr     imprime_cadena
427
428         lda     teclado
429
430         lda     #0
431         sta     nintentos
432         sta     contador_a
433         sta     ntablero
434
435         lbra    pantalla_menu
436
```