

Universidad Nacional Autónoma de México

Facultad de Ingeniería

Estructura y Programación de Computadoras

Grupo 05

SEGUNDO PROYECTO: SERPIENTES Y ESCALERAS

Nombre: Aguilar Maya Daniel
Profesor: Pedro Ignacio Rincón Gómez
Semestre 2023-2

Instrucciones de uso: Ejecutar el programa en THRSim11 y seguir la siguiente metodología:

1. Abrir puerto serial
2. Escribir la palabra DADOS
3. Observar que el jugador 1 cambie su posición de acuerdo con lo que marque la TIRADA de los dados (DADO 1 Y DADO 2).
4. Escribir la palabra DADOS
5. Observar que el jugador 2 cambie su posición de acuerdo a lo que marque la TIRADA de los dados (DADO 1 Y DADO 2)
6. Repetir paso 2,3,4,5 hasta que algún jugador llegue a la posición 100

NOTA: En los casos de alguna serpiente o escalera, o exceso de las 100 unidades, se hará el ajuste adecuado y se actualizará automáticamente la posición.

Lista de cotejo:

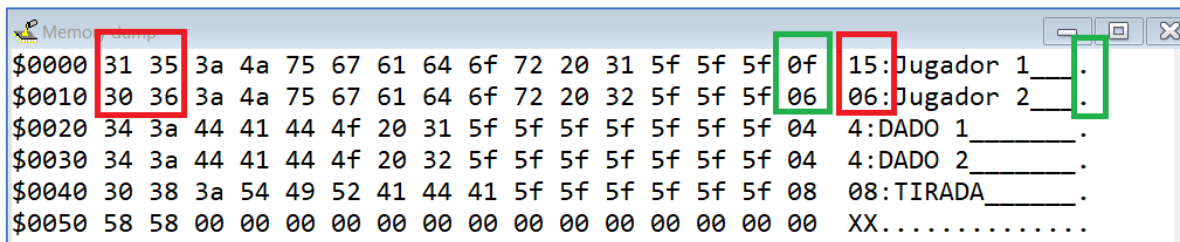
1. Correcto despliegue de la casilla en la que está cada jugador en decimal

CUMPLE

En las localidades \$0000 \$0001 se despliega la casilla de cada jugador en decimal y en la localidad \$000F se despliega el ASCII del valor hexadecimal.

Para el segundo jugador se hace lo propio en las localidades \$0010, \$0011 y \$001F.

Si hay ganador entonces se ocupará el carácter ':' para mostrar el 100.



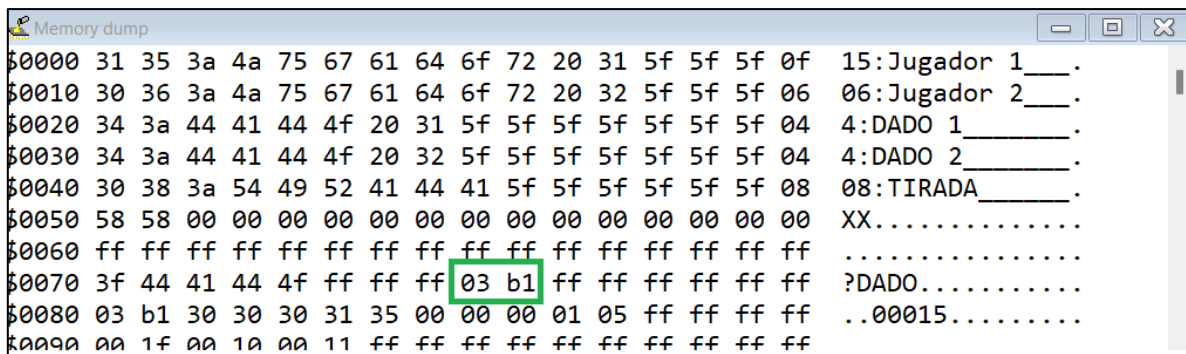
```
Memor...
$0000 31 35 3a 4a 75 67 61 64 6f 72 20 31 5f 5f 5f 0f 15: Jugador 1___.
$0010 30 36 3a 4a 75 67 61 64 6f 72 20 32 5f 5f 5f 06 06: Jugador 2___.
$0020 34 3a 44 41 44 4f 20 31 5f 5f 5f 5f 5f 5f 5f 04 4: DADO 1___.
$0030 34 3a 44 41 44 4f 20 32 5f 5f 5f 5f 5f 5f 5f 04 4: DADO 2___.
$0040 30 38 3a 54 49 52 41 44 41 5f 5f 5f 5f 5f 5f 08 08: TIRADA___.
$0050 58 58 00 00 00 00 00 00 00 00 00 00 00 00 00 00 XX.....
```

2. Correcta generación del número aleatorio de 16 bits

CUMPLE

En la localidad de memoria \$0078 y \$0079 se guardan los valores del número aleatorio de 16 bits (variable ALEATORIO). La generación de los dados se hace tomando cada byte y dividiendo entre 6 para tomar el residuo y sumar 1 (dado que al dividir por 6 el residuo es del 0 al 5).

Por ejemplo: En el ejemplo siguiente se tiene el número aleatorio 03 b1: El primer byte es 03 (en decimal 3) que al dividirlo por 6 hará que el residuo sea 3 y al sumarle 1, 4. Para el segundo byte se tiene b1 (33 en decimal) que al dividirlo por 6 se tiene un residuo de 3 unidades y al sumar 1 se tiene 4. Lo cual concuerda con los valores de los dados, que también se ve visualmente



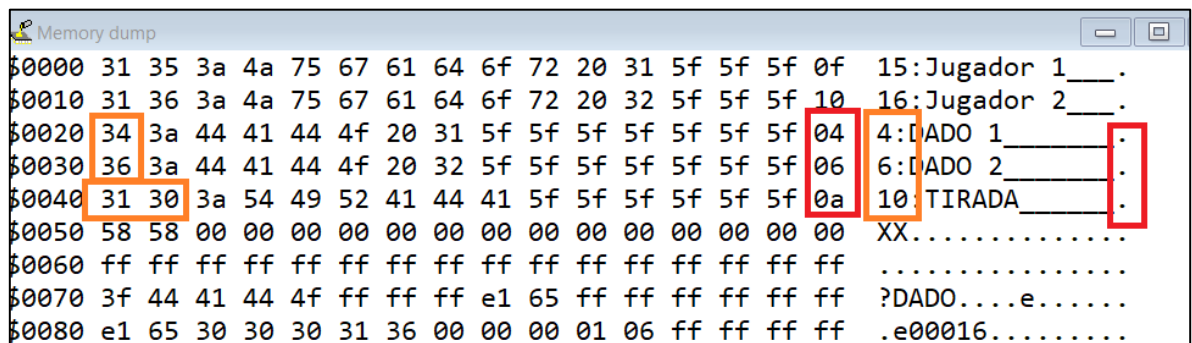
```
Memory dump
$0000 31 35 3a 4a 75 67 61 64 6f 72 20 31 5f 5f 5f 0f 15:Jugador 1___.
$0010 30 36 3a 4a 75 67 61 64 6f 72 20 32 5f 5f 5f 06 06:Jugador 2___.
$0020 34 3a 44 41 44 4f 20 31 5f 5f 5f 5f 5f 5f 5f 04 4:DADO 1___.
$0030 34 3a 44 41 44 4f 20 32 5f 5f 5f 5f 5f 5f 5f 04 4:DADO 2___.
$0040 30 38 3a 54 49 52 41 44 41 5f 5f 5f 5f 5f 5f 08 08:TIRADA___.
$0050 58 58 00 00 00 00 00 00 00 00 00 00 00 00 00 00 XX.....
$0060 ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff .....
$0070 3f 44 41 44 4f ff ff ff ff 03 b1 ff ff ff ff ff ff ?DADO.....
$0080 03 b1 30 30 30 31 35 00 00 00 01 05 ff ff ff ff ff ..00015.....
$0090 00 1f 00 10 00 11 ff ff ff ff ff ff ff ff ff ff
```

3. Correcto despliegue del valor de cada dado y de su suma en decimal

CUMPLE

El valor de cada dado en decimal se ve en el recuadro naranja de la parte derecha. Los recuadros naranjas de la parte izquierda son los valores en hexadecimal cuyo ASCII es el decimal deseado.

En rojo, se tienen los valores reales (numérico) hexadecimales del dado 1, dado 2 y la suma (tirada)

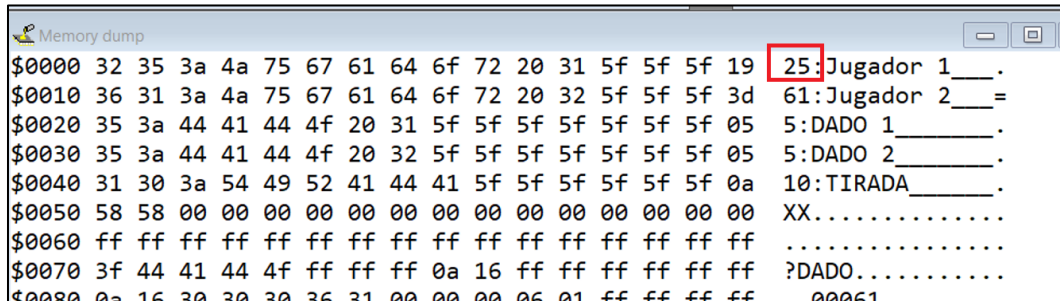


```
Memory dump
$0000 31 35 3a 4a 75 67 61 64 6f 72 20 31 5f 5f 5f 0f 15:Jugador 1___.
$0010 31 36 3a 4a 75 67 61 64 6f 72 20 32 5f 5f 5f 10 16:Jugador 2___.
$0020 34 3a 44 41 44 4f 20 31 5f 5f 5f 5f 5f 5f 5f 04 4:DADO 1___.
$0030 36 3a 44 41 44 4f 20 32 5f 5f 5f 5f 5f 5f 5f 06 6:DADO 2___.
$0040 31 30 3a 54 49 52 41 44 41 5f 5f 5f 5f 5f 5f 0a 10:TIRADA___.
$0050 58 58 00 00 00 00 00 00 00 00 00 00 00 00 00 00 XX.....
$0060 ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff .....
$0070 3f 44 41 44 4f ff ff ff ff e1 65 ff ff ff ff ff ff ?DADO....e.....
$0080 e1 65 30 30 30 31 36 00 00 00 01 06 ff ff ff ff ff .e00016.....
```

4. Correcto funcionamiento del juego respetando las escaleras.

CUMPLE

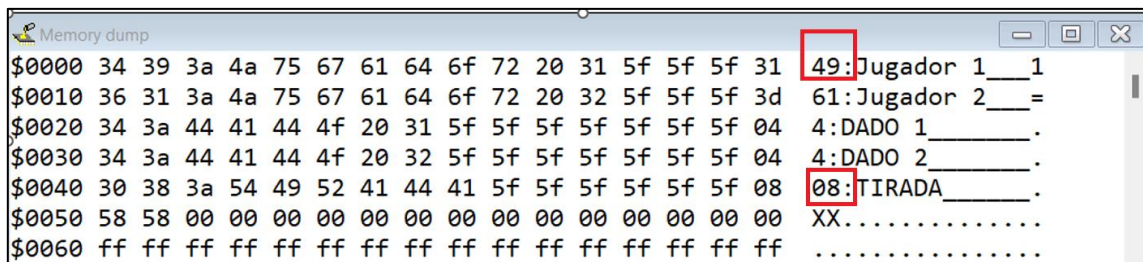
Al hacer una tirada de dados, el jugador se mueve y entonces se verifica si está en la parte baja de la escalera, para poder aumentar el valor de su posición y estar mas cerca de la meta:



A screenshot of a memory dump window titled "Memory dump". It displays a list of memory addresses and their corresponding hex values. On the right side, there are labels for game variables. The label "25:Jugador 1__." is highlighted with a red box, indicating the current position of Player 1.

Address	Hex Data	Label
\$0000	32 35 3a 4a 75 67 61 64 6f 72 20 31 5f 5f 5f 19	25:Jugador 1__.
\$0010	36 31 3a 4a 75 67 61 64 6f 72 20 32 5f 5f 5f 3d	61:Jugador 2__=
\$0020	35 3a 44 41 44 4f 20 31 5f 5f 5f 5f 5f 5f 05	5:DADO 1__.
\$0030	35 3a 44 41 44 4f 20 32 5f 5f 5f 5f 5f 5f 05	5:DADO 2__.
\$0040	31 30 3a 54 49 52 41 44 41 5f 5f 5f 5f 5f 0a	10:TIRADA__.
\$0050	58 58 00 00 00 00 00 00 00 00 00 00 00 00 00	XX.....
\$0060	ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff
\$0070	3f 44 41 44 4f ff ff ff 0a 16 ff ff ff ff ff ff	?DADO.....
\$0080	0a 16 30 30 30 36 31 00 00 00 06 01 ff ff ff ff	00061

Antes: Posición del jugador 1 en 25



A screenshot of a memory dump window titled "Memory dump". It displays a list of memory addresses and their corresponding hex values. On the right side, there are labels for game variables. The label "49:Jugador 1__1" is highlighted with a red box, indicating the current position of Player 1 after a move.

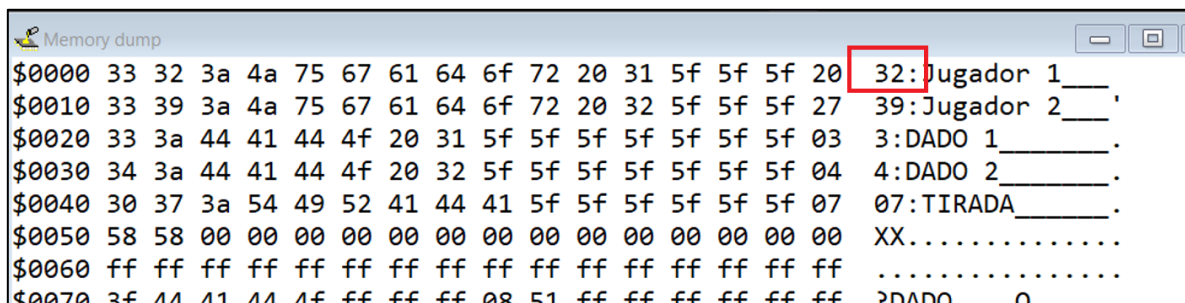
Address	Hex Data	Label
\$0000	34 39 3a 4a 75 67 61 64 6f 72 20 31 5f 5f 5f 31	49:Jugador 1__1
\$0010	36 31 3a 4a 75 67 61 64 6f 72 20 32 5f 5f 5f 3d	61:Jugador 2__=
\$0020	34 3a 44 41 44 4f 20 31 5f 5f 5f 5f 5f 5f 04	4:DADO 1__.
\$0030	34 3a 44 41 44 4f 20 32 5f 5f 5f 5f 5f 5f 04	4:DADO 2__.
\$0040	30 38 3a 54 49 52 41 44 41 5f 5f 5f 5f 5f 08	08:TIRADA__.
\$0050	58 58 00 00 00 00 00 00 00 00 00 00 00 00 00	XX.....
\$0060	ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff

Después: Jugador 1 con tirada de 8 (25 más 8 es 33). En 33 hay una escalera a 49

5. Correcto funcionamiento del juego respetando las serpientes

CUMPLE

Al hacer una tirada de dados, el jugador se mueve y entonces se verifica si está en la cabeza de la serpiente. En cuyo caso, a la posición actual se le restan unidades para tener la nueva posición del jugador y estar más alejado de la meta.



A screenshot of a memory dump window titled "Memory dump". It displays a list of memory addresses and their corresponding hex values. On the right side, there are labels for game variables. The label "32:Jugador 1__" is highlighted with a red box, indicating the current position of Player 1.

Address	Hex Data	Label
\$0000	33 32 3a 4a 75 67 61 64 6f 72 20 31 5f 5f 5f 20	32:Jugador 1__
\$0010	33 39 3a 4a 75 67 61 64 6f 72 20 32 5f 5f 5f 27	39:Jugador 2__
\$0020	33 3a 44 41 44 4f 20 31 5f 5f 5f 5f 5f 5f 03	3:DADO 1__.
\$0030	34 3a 44 41 44 4f 20 32 5f 5f 5f 5f 5f 5f 04	4:DADO 2__.
\$0040	30 37 3a 54 49 52 41 44 41 5f 5f 5f 5f 5f 07	07:TIRADA__.
\$0050	58 58 00 00 00 00 00 00 00 00 00 00 00 00 00	XX.....
\$0060	ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff
\$0070	3f 44 41 44 4f ff ff ff 08 51 ff ff ff ff ff ff	?DADO 0

Antes: Jugador 1 en la posición 32

\$0000	31	38	3a	4a	75	67	61	64	6f	72	20	31	5f	5f	5f	12	18:Jugador 1__.
\$0010	33	39	3a	4a	75	67	61	64	6f	72	20	32	5f	5f	5f	27	39:Jugador 2__.
\$0020	35	3a	44	41	44	4f	20	31	5f	5f	5f	5f	5f	5f	5f	05	5:DADO 1__.
\$0030	36	3a	44	41	44	4f	20	32	5f	5f	5f	5f	5f	5f	5f	06	6:DADO 2__.
\$0040	31	31	3a	54	49	52	41	44	41	5f	5f	5f	5f	5f	5f	0b	11:TIRADA__.
\$0050	58	58	00	00	00	00	00	00	00	00	00	00	00	00	00	00	XX.....
\$0060	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff
\$0070	3f	44	41	44	4f	ff	ff	ff	ff	0a	29	ff	ff	ff	ff	ff	?DADO....)

Después: Jugador 1 con tirada de 11 (Posición inicial de 32 + 11 son 43). En 43 hay una serpiente cuya cola está en la posición 18

6. Correcto funcionamiento del juego cuando sobrepasa el puntaje de 100

CUMPLE

El programa se diseño para que, antes de verificar si hay escalera o serpiente, se verifique siempre si se ha excedido la suma en el valor de 100. En caso de ser verdadero, ese excedente se le resta a 100 y será el valor final

En el siguiente ejemplo, la posición del jugador 2 es 98. Cuando le vuelve a tocar, se tiene un valor de 7 en la tirada, por lo que la posición debería ser 105. Sin embargo ese 5 de excedente se restan, dando un total de 95

\$0000	37	32	3a	4a	75	67	61	64	6f	72	20	31	5f	5f	5f	48	72:Jugador 1__H
\$0010	39	38	3a	4a	75	67	61	64	6f	72	20	32	5f	5f	5f	62	98:Jugador 2__b
\$0020	31	3a	44	41	44	4f	20	31	5f	5f	5f	5f	5f	5f	5f	01	1:DADO 1__.
\$0030	32	3a	44	41	44	4f	20	32	5f	5f	5f	5f	5f	5f	5f	02	2:DADO 2__.
\$0040	30	33	3a	54	49	52	41	44	41	5f	5f	5f	5f	5f	5f	03	03:TIRADA__.
\$0050	58	58	00	00	00	00	00	00	00	00	00	00	00	00	00	00	XX.....
\$0060	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff
\$0070	3f	44	41	44	4f	ff	ff	ff	ff	0c	1f	ff	ff	ff	ff	ff	?DADO.....

Antes: 98 en jugador 2

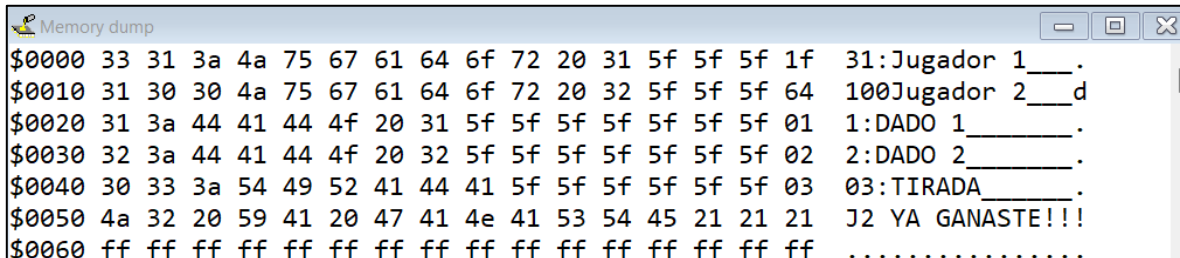
\$0000	38	30	3a	4a	75	67	61	64	6f	72	20	31	5f	5f	5f	50	80:Jugador 1__P
\$0010	39	35	3a	4a	75	67	61	64	6f	72	20	32	5f	5f	5f	5f	95:Jugador 2__
\$0020	36	3a	44	41	44	4f	20	31	5f	5f	5f	5f	5f	5f	5f	06	6:DADO 1__.
\$0030	31	3a	44	41	44	4f	20	32	5f	5f	5f	5f	5f	5f	5f	01	1:DADO 2__.
\$0040	30	37	3a	54	49	52	41	44	41	5f	5f	5f	5f	5f	5f	07	07:TIRADA__.
\$0050	58	58	00	00	00	00	00	00	00	00	00	00	00	00	00	00	XX.....
\$0060	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff

Después: 95 en jugador 2 con tirada de 7 (hubo un excedente)

7. ¡¡¡Correcto despliegue del mensaje GANASTE!!!

CUMPLE

Cuando uno de los dos jugadores llega a 100, se muestra un mensaje en la ventana DUMP, evidenciando si ganó el jugador 1 (J1) o el jugador 2 (J2)



```
$0000 33 31 3a 4a 75 67 61 64 6f 72 20 31 5f 5f 5f 1f 31:Jugador 1___.
$0010 31 30 30 4a 75 67 61 64 6f 72 20 32 5f 5f 5f 64 100Jugador 2__d
$0020 31 3a 44 41 44 4f 20 31 5f 5f 5f 5f 5f 5f 5f 01 1:DADO 1___.
$0030 32 3a 44 41 44 4f 20 32 5f 5f 5f 5f 5f 5f 5f 02 2:DADO 2___.
$0040 30 33 3a 54 49 52 41 44 41 5f 5f 5f 5f 5f 5f 03 03:TIRADA___.
$0050 4a 32 20 59 41 20 47 41 4e 41 53 54 45 21 21 21 J2 YA GANASTE!!!
$0060 ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff .....
```

8. Empleo correcto de la cadena START para comenzar desde cero

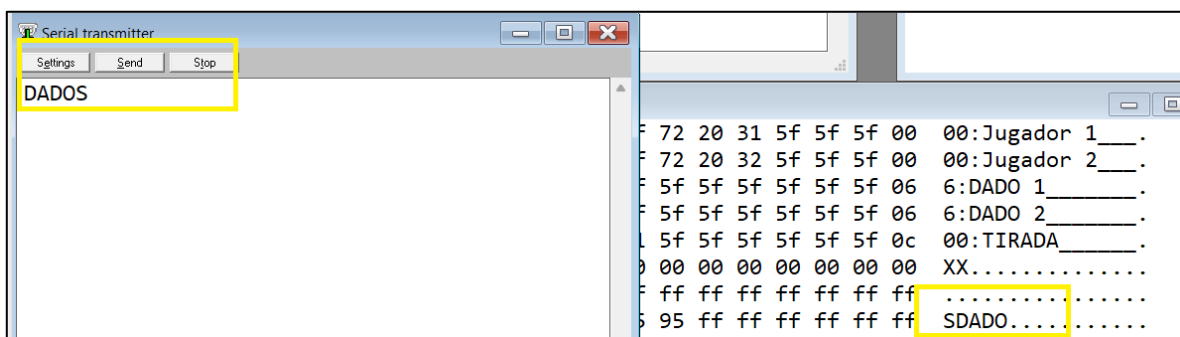
No CUMPLE

9. Uso del puerto serial con la cadena DADOS para hacer un nuevo tiro

CUMPLE

Se hace uso del puerto serial para escribir la cadena 'DADOS' y el programa genere un número aleatorio para poder generar a los dos primeros dados y a su suma (nuevo tiro).

Se partió del programa start visto en clase para usar la palabra DADOS



```
Serial transmitter
Settings Send Stop
DADOS

72 20 31 5f 5f 5f 00 00:Jugador 1___.
72 20 32 5f 5f 5f 00 00:Jugador 2___.
5f 5f 5f 5f 5f 5f 06 6:DADO 1___.
5f 5f 5f 5f 5f 5f 06 6:DADO 2___.
5f 5f 5f 5f 5f 5f 0c 00:TIRADA___.
00 00 00 00 00 00 00 XX.....
ff ff ff ff ff ff ff .....
95 ff ff ff ff ff ff ff SDADO.....
```

10. Uso de interrupción del puerto serial asíncrono en el programa

CUMPLE

Mientras el programa se cicla, se espera una interrupción para escribir la palabra DADOS y así el programa pueda seguir su flujo natural hasta hallar al ganador del juego.

* VECTOR INTERRUPCION SERIAL

ORG \$FFD6
FCB \$F1,\$00

*RESET

ORG \$FFFE
RESET FCB \$80,\$00

ORG \$82b0
EXIT03:
END \$8000