

# ORGA1: Arquitectura y Seguimiento

## Organización del Computador I

Verónica Coy

Departamento de Computación - FCEyN  
UBA

Primer Cuatrimestre 2019 - Turno Tarde

# Ciclo de Vida de un Programa

# Ciclo de Vida de un Programa

- 1 **Programación:** Escribir un algoritmo en **lenguaje ensamblador (o assembly)**. Esto es el **código fuente**.

# Ciclo de Vida de un Programa

- 1 **Programación:** Escribir un algoritmo en **lenguaje ensamblador (o assembly)**. Esto es el **código fuente**.
- 2 **Ensamblado:** Un programa llamado **Ensamblador** toma el **código fuente** y lo traduce a un **código máquina**.

# Ciclo de Vida de un Programa

- ① **Programación:** Escribir un algoritmo en **lenguaje ensamblador (o assembly)**. Esto es el **código fuente**.
- ② **Ensamblado:** Un programa llamado **Ensamblador** toma el **código fuente** y lo traduce a un **código máquina**.
  - ① Resuelve las directivas dirigidas al **Ensamblador**.
  - ② Calcula el tamaño de cada instrucciones, y resuelve las etiquetas.
  - ③ Traduce las instrucciones a 0s y 1s.

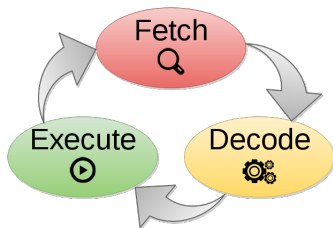
# Ciclo de Vida de un Programa

- ① **Programación:** Escribir un algoritmo en **lenguaje ensamblador (o assembly)**. Esto es el **código fuente**.
- ② **Ensamblado:** Un programa llamado **Ensamblador** toma el **código fuente** y lo traduce a un **código máquina**.
  - ① Resuelve las directivas dirigidas al **Ensamblador**.
  - ② Calcula el tamaño de cada instrucciones, y resuelve las etiquetas.
  - ③ Traduce las instrucciones a 0s y 1s.
- ③ **Carga:** Se le indica al **Ensamblador** una dirección inicial, y éste copia el **código máquina** en la **Memoria** desde esa posición en adelante. Luego, carga esa misma dirección en el **PC (Program Counter)**.

# Ciclo de Vida de un Programa

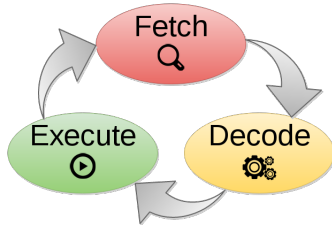
- ① **Programación:** Escribir un algoritmo en **lenguaje ensamblador (o assembly)**. Esto es el **código fuente**.
- ② **Ensamblado:** Un programa llamado **Ensamblador** toma el **código fuente** y lo traduce a un **código máquina**.
  - ① Resuelve las directivas dirigidas al **Ensamblador**.
  - ② Calcula el tamaño de cada instrucciones, y resuelve las etiquetas.
  - ③ Traduce las instrucciones a 0s y 1s.
- ③ **Carga:** Se le indica al **Ensamblador** una dirección inicial, y éste copia el **código máquina** en la **Memoria** desde esa posición en adelante. Luego, carga esa misma dirección en el **PC (Program Counter)**.
- ④ **Ejecución:** El **CPU** da inicio a su **ciclo de ejecución** comenzando por la posición indicada en el **PC**.

# Ciclo de Ejecución



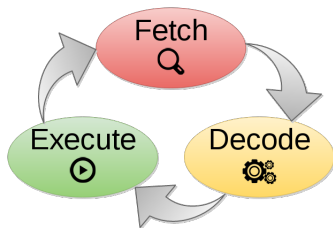


# Ciclo de Ejecución



**Fetch** La **UC (Unidad de Control)** obtiene una instrucción de la posición de **Memoria** a la que apunta el **PC** **y lo incrementa** (Si es necesario: busca más palabras de la instrucción usando el **PC** e incrementándolo cada vez.)

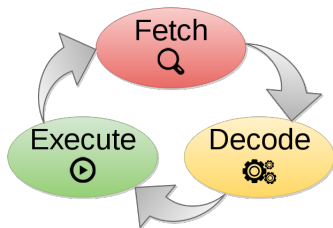
# Ciclo de Ejecución



**Fetch** La **UC (Unidad de Control)** obtiene una instrucción de la posición de **Memoria** a la que apunta el **PC** **y lo incrementa** (Si es necesario: busca más palabras de la instrucción usando el **PC** e incrementándolo cada vez.)

**Decode** La **UC** decodifica la instrucción.

# Ciclo de Ejecución



**Fetch** La **UC (Unidad de Control)** obtiene una instrucción de la posición de **Memoria** a la que apunta el **PC** **y lo incrementa** (Si es necesario: busca más palabras de la instrucción usando el **PC** e incrementándolo cada vez.)

**Decode** La **UC** decodifica la instrucción.

**Execute** La **UC** ejecuta la instrucción.

# Ejercicio 1

Codifiquemos el siguiente programa y carguémoslo desde la posición de memoria 0x0000:

```
main:  MOV R1,[[et1]]
        ADD [R1],0x6000
        JMP main
et1:    DW 0x0007
et2:    DW 0x0004
```

# Ejercicio 1

Codifiquemos el siguiente programa y carguémoslo desde la posición de memoria 0x0000:

```
main:  MOV R1,[[et1]]
        ADD [R1],0x6000
        JMP main
et1:    DW 0x0007
et2:    DW 0x0004
```

Tenemos que:

# Ejercicio 1

Codifiquemos el siguiente programa y carguémoslo desde la posición de memoria 0x0000:

```
main:  MOV R1,[[et1]]  
        ADD [R1],0x6000  
        JMP main  
  
et1:   DW 0x0007  
et2:   DW 0x0004
```

Tenemos que:

- ver cuántas palabras necesita cada instrucción

# Ejercicio 1

Codifiquemos el siguiente programa y carguémoslo desde la posición de memoria 0x0000:

```
main:  MOV R1,[[et1]]
        ADD [R1],0x6000
        JMP main
et1:    DW 0x0007
et2:    DW 0x0004
```

Tenemos que:

- ver cuántas palabras necesita cada instrucción
- calcular los valores de las etiquetas

# Ejercicio 1

Codifiquemos el siguiente programa y carguémoslo desde la posición de memoria 0x0000:

```
main:  MOV R1,[[et1]]
        ADD [R1],0x6000
        JMP main
et1:    DW 0x0007
et2:    DW 0x0004
```

Tenemos que:

- ver cuántas palabras necesita cada instrucción
- calcular los valores de las etiquetas

*DW (Define Word)*: directiva al ensamblador que provoca que en la posición de memoria que le corresponde, aparezca el valor indicado.



Codifiquemos el siguiente programa y carguémoslo desde la posición de memoria 0x0000:

dirección    ocupa

```
main:  MOV R1,[[et1]]
        ADD [R1],0x6000
        JMP main
et1:    DW 0x0007
et2:    DW 0x0004
```

Codifiquemos el siguiente programa y carguémoslo desde la posición de memoria 0x0000:

dirección    ocupa

```
main:  MOV R1,[[et1]]
        ADD [R1],0x6000
        JMP main

et1:    DW 0x0007
et2:    DW 0x0004
```

Codifiquemos el siguiente programa y carguémoslo desde la posición de memoria 0x0000:

```
main:  MOV R1,[[et1]]
        ADD [R1],0x6000
        JMP main
et1:    DW 0x0007
et2:    DW 0x0004
```

dirección    ocupa

0x0000

Codifiquemos el siguiente programa y carguémoslo desde la posición de memoria 0x0000:

```
main:  MOV R1,[[et1]]
        ADD [R1],0x6000
        JMP main

et1:   DW 0x0007
et2:   DW 0x0004
```

dirección	ocupa
0x0000	dos palabras

Codifiquemos el siguiente programa y carguémoslo desde la posición de memoria 0x0000:

```
main:  MOV R1,[[et1]]
        ADD [R1],0x6000
        JMP main
et1:    DW 0x0007
et2:    DW 0x0004
```

dirección	ocupa
0x0000	dos palabras
0x0002	

Codifiquemos el siguiente programa y carguémoslo desde la posición de memoria 0x0000:

```
main:  MOV R1,[[et1]]
        ADD [R1],0x6000
        JMP main
et1:    DW 0x0007
et2:    DW 0x0004
```

dirección	ocupa
0x0000	dos palabras
0x0002	dos palabras

Codifiquemos el siguiente programa y carguémoslo desde la posición de memoria 0x0000:

```
main:  MOV R1,[[et1]]
        ADD [R1],0x6000
        JMP main

et1:   DW 0x0007
et2:   DW 0x0004
```

dirección	ocupa
0x0000	dos palabras
0x0002	dos palabras
0x0004	

Codifiquemos el siguiente programa y carguémoslo desde la posición de memoria 0x0000:

```
main:  MOV R1,[[et1]]
        ADD [R1],0x6000
        JMP main
et1:    DW 0x0007
et2:    DW 0x0004
```

dirección	ocupa
0x0000	dos palabras
0x0002	dos palabras
0x0004	dos palabras



Codifiquemos el siguiente programa y carguémoslo desde la posición de memoria 0x0000:

```
main:  MOV R1,[[et1]]
        ADD [R1],0x6000
        JMP main
et1:    DW 0x0007
et2:    DW 0x0004
```

dirección	ocupa
0x0000	dos palabras
0x0002	dos palabras
0x0004	dos palabras
0x0006	

Codifiquemos el siguiente programa y carguémoslo desde la posición de memoria 0x0000:

```
main:  MOV R1,[[et1]]
        ADD [R1],0x6000
        JMP main
et1:    DW 0x0007
et2:    DW 0x0004
```

dirección	ocupa
0x0000	dos palabras
0x0002	dos palabras
0x0004	dos palabras
0x0006	una palabra

Codifiquemos el siguiente programa y carguémoslo desde la posición de memoria 0x0000:

```
main:  MOV R1,[[et1]]
        ADD [R1],0x6000
        JMP main
et1:    DW 0x0007
et2:    DW 0x0004
```

dirección	ocupa
0x0000	dos palabras
0x0002	dos palabras
0x0004	dos palabras
0x0006	una palabra
0x0007	

Codifiquemos el siguiente programa y carguémoslo desde la posición de memoria 0x0000:

```
main:  MOV R1,[[et1]]
        ADD [R1],0x6000
        JMP main
et1:    DW 0x0007
et2:    DW 0x0004
```

dirección	ocupa
0x0000	dos palabras
0x0002	dos palabras
0x0004	dos palabras
0x0006	una palabra
0x0007	una palabra

Codifiquemos el siguiente programa y carguémoslo desde la posición de memoria 0x0000:

```
main:  MOV R1,[[et1]]
        ADD [R1],0x6000
        JMP main
et1:    DW 0x0007
et2:    DW 0x0004
```

dirección	ocupa
0x0000	dos palabras
0x0002	dos palabras
0x0004	dos palabras
0x0006	una palabra
0x0007	una palabra

Codifiquemos el siguiente programa y carguémoslo desde la posición de memoria 0x0000:

```
main:  MOV R1,[[et1]]  
        ADD [R1],0x6000  
        JMP main  
et1:   DW 0x0007  
et2:   DW 0x0004
```

dirección	ocupa
0x0000	dos palabras
0x0002	dos palabras
0x0004	dos palabras
0x0006	una palabra
0x0007	una palabra

La etiqueta **main** corresponde a la dirección de memoria **0x0000**.

La etiqueta **et1** corresponde a la dirección de memoria **0x0006**.

La etiqueta **et2** corresponde a la dirección de memoria **0x0007**.

Reemplazando las etiquetas por valores de las direcciones estamos listos para codificar.

Codifiquemos el siguiente programa y carguémoslo desde la posición de memoria 0x0000:

```
MOV R1,[[0x0006]]
```

```
ADD [R1],0x6000
```

```
JMP 0x0000
```

```
DW 0x0007
```

```
DW 0x0004
```

dirección	ocupa
0x0000	dos palabras
0x0002	dos palabras
0x0004	dos palabras
0x0006	una palabra
0x0007	una palabra

Codifiquemos el siguiente programa y carguémoslo desde la posición de memoria 0x0000:

```
MOV R1,[[0x0006]]
```

```
ADD [R1],0x6000
```

```
JMP 0x0000
```

```
DW 0x0007
```

```
DW 0x0004
```

dirección	ocupa
0x0000	dos palabras
0x0002	dos palabras
0x0004	dos palabras
0x0006	una palabra
0x0007	una palabra

Traducidas las etiquetas lo codificamos y cargamos en memoria:

0000	0001	0002	0003	0004	0005	0006	0007

Y también cargamos la dirección inicial en el **PC**:

**PC**



Codifiquemos el siguiente programa y carguémoslo desde la posición de memoria 0x0000:

**MOV R1,[[0x0006]]**

ADD [R1],0x6000

JMP 0x0000

DW 0x0007

DW 0x0004

dirección	ocupa
0x0000	dos palabras
0x0002	dos palabras
0x0004	dos palabras
0x0006	una palabra
0x0007	una palabra

Traducidas las etiquetas lo codificamos y cargamos en memoria:

0000	0001	0002	0003	0004	0005	0006	0007

Y también cargamos la dirección inicial en el **PC**:

**PC**

Codifiquemos el siguiente programa y carguémoslo desde la posición de memoria 0x0000:

```
MOV R1,[[0x0006]]
```

```
ADD [R1],0x6000
```

```
JMP 0x0000
```

```
DW 0x0007
```

```
DW 0x0004
```

dirección	ocupa
0x0000	dos palabras
0x0002	dos palabras
0x0004	dos palabras
0x0006	una palabra
0x0007	una palabra

Traducidas las etiquetas lo codificamos y cargamos en memoria:

0000	0001	0002	0003	0004	0005	0006	0007
1858	0006						

Y también cargamos la dirección inicial en el PC:

PC	
----	--

Codifiquemos el siguiente programa y carguémoslo desde la posición de memoria 0x0000:

```
MOV R1,[[0x0006]]
```

```
ADD [R1],0x6000
```

```
JMP 0x0000
```

```
DW 0x0007
```

```
DW 0x0004
```

dirección	ocupa
0x0000	dos palabras
0x0002	dos palabras
0x0004	dos palabras
0x0006	una palabra
0x0007	una palabra

Traducidas las etiquetas lo codificamos y cargamos en memoria:

0000	0001	0002	0003	0004	0005	0006	0007
1858	0006						

Y también cargamos la dirección inicial en el PC:

PC

Codifiquemos el siguiente programa y carguémoslo desde la posición de memoria 0x0000:

```
MOV R1,[[0x0006]]
```

```
ADD [R1],0x6000
```

```
JMP 0x0000
```

```
DW 0x0007
```

```
DW 0x0004
```

dirección	ocupa
0x0000	dos palabras
0x0002	dos palabras
0x0004	dos palabras
0x0006	una palabra
0x0007	una palabra

Traducidas las etiquetas lo codificamos y cargamos en memoria:

0000	0001	0002	0003	0004	0005	0006	0007
1858	0006	2C40	6000				

Y también cargamos la dirección inicial en el PC:

PC

Codifiquemos el siguiente programa y carguémoslo desde la posición de memoria 0x0000:

```
MOV R1,[[0x0006]]
```

```
ADD [R1],0x6000
```

```
JMP 0x0000
```

```
DW 0x0007
```

```
DW 0x0004
```

dirección	ocupa
0x0000	dos palabras
0x0002	dos palabras
0x0004	dos palabras
0x0006	una palabra
0x0007	una palabra

Traducidas las etiquetas lo codificamos y cargamos en memoria:

0000	0001	0002	0003	0004	0005	0006	0007
1858	0006	2C40	6000				

Y también cargamos la dirección inicial en el PC:

PC

Codifiquemos el siguiente programa y carguémoslo desde la posición de memoria 0x0000:

```
MOV R1,[[0x0006]]
ADD [R1],0x6000
JMP 0x0000
DW 0x0007
DW 0x0004
```

dirección	ocupa
0x0000	dos palabras
0x0002	dos palabras
0x0004	dos palabras
0x0006	una palabra
0x0007	una palabra

Traducidas las etiquetas lo codificamos y cargamos en memoria:

0000	0001	0002	0003	0004	0005	0006	0007
1858	0006	2C40	6000	A000	0000		

Y también cargamos la dirección inicial en el PC:

PC	
----	--

Codifiquemos el siguiente programa y carguémoslo desde la posición de memoria 0x0000:

```
MOV R1,[[0x0006]]
```

```
ADD [R1],0x6000
```

```
JMP 0x0000
```

```
DW 0x0007
```

```
DW 0x0004
```

dirección	ocupa
0x0000	dos palabras
0x0002	dos palabras
0x0004	dos palabras
0x0006	una palabra
0x0007	una palabra

Traducidas las etiquetas lo codificamos y cargamos en memoria:

0000	0001	0002	0003	0004	0005	0006	0007
1858	0006	2C40	6000	A000	0000		

Y también cargamos la dirección inicial en el PC:

PC

Codifiquemos el siguiente programa y carguémoslo desde la posición de memoria 0x0000:

```
MOV R1,[[0x0006]]
```

```
ADD [R1],0x6000
```

```
JMP 0x0000
```

```
DW 0x0007
```

```
DW 0x0004
```

dirección	ocupa
0x0000	dos palabras
0x0002	dos palabras
0x0004	dos palabras
0x0006	una palabra
0x0007	una palabra

Traducidas las etiquetas lo codificamos y cargamos en memoria:

0000	0001	0002	0003	0004	0005	0006	0007
1858	0006	2C40	6000	A000	0000	0007	

Y también cargamos la dirección inicial en el PC:

PC



Codifiquemos el siguiente programa y carguémoslo desde la posición de memoria 0x0000:

```
MOV R1,[[0x0006]]
```

```
ADD [R1],0x6000
```

```
JMP 0x0000
```

```
DW 0x0007
```

```
DW 0x0004
```

dirección	ocupa
0x0000	dos palabras
0x0002	dos palabras
0x0004	dos palabras
0x0006	una palabra
0x0007	una palabra

Traducidas las etiquetas lo codificamos y cargamos en memoria:

0000	0001	0002	0003	0004	0005	0006	0007
1858	0006	2C40	6000	A000	0000	0007	

Y también cargamos la dirección inicial en el PC:

PC

Codifiquemos el siguiente programa y carguémoslo desde la posición de memoria 0x0000:

```
MOV R1,[[0x0006]]
```

```
ADD [R1],0x6000
```

```
JMP 0x0000
```

```
DW 0x0007
```

```
DW 0x0004
```

dirección	ocupa
0x0000	dos palabras
0x0002	dos palabras
0x0004	dos palabras
0x0006	una palabra
0x0007	una palabra

Traducidas las etiquetas lo codificamos y cargamos en memoria:

0000	0001	0002	0003	0004	0005	0006	0007
1858	0006	2C40	6000	A000	0000	0007	0004

Y también cargamos la dirección inicial en el PC:

PC

Codifiquemos el siguiente programa y carguémoslo desde la posición de memoria 0x0000:

```
MOV R1,[[0x0006]]
ADD [R1],0x6000
JMP 0x0000
DW 0x0007
DW 0x0004
```

dirección	ocupa
0x0000	dos palabras
0x0002	dos palabras
0x0004	dos palabras
0x0006	una palabra
0x0007	una palabra

Traducidas las etiquetas lo codificamos y cargamos en memoria:

0000	0001	0002	0003	0004	0005	0006	0007
1858	0006	2C40	6000	A000	0000	0007	0004

Y también cargamos la dirección inicial en el PC:

PC	0000
----	------

Empezamos a hacer el ciclo **Fetch-Decode-Execute** con el **PC** = 0x0000.

0000	0001	0002	0003	0004	0005	0006	0007
1858	0006	2C40	6000	A000	0000	0007	0004

## Ejercicio 1 - Seguimiento

Valores  
iniciales

PC	SP
0000	FFEF

R0	R1	R2	R3	R4	R5	R6	R7
0000	0000	0000	0000	0000	0000	0000	0000

Z	C	V	N
0	0	0	0

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0000	1858	0006	2C40	6000	A000	0000	0007	0004	0000	0000	0000	0000	0000	0000	0000	0000
0010	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

	PC	SP	[SP+1]	IR	Instrucción - 1er palabra	PC	2da palabra	PC	3ra palabra	PC	Instrucción decodificada				
1															
	Ejecución										Flags				
2															
	Ejecución										Flags				
3															
	Ejecución										Flags				

## Ejercicio 1 - Seguimiento

Valores  
iniciales

PC	SP
0000	FFEF

R0	R1	R2	R3	R4	R5	R6	R7
0000	0000	0000	0000	0000	0000	0000	0000

Z	C	V	N
0	0	0	0

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0000	1858	0006	2C40	6000	A000	0000	0007	0004	0000	0000	0000	0000	0000	0000	0000	0000
0010	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

	PC	SP	[SP+1]	IR	Instrucción - 1er palabra	PC	2da palabra	PC	3ra palabra	PC	Instrucción decodificada				
1	0000														
	Ejecución										Flags				
2															
	Ejecución										Flags				
3															
	Ejecución										Flags				

## Ejercicio 1 - Seguimiento

Valores  
iniciales

PC	SP
0000	FFEF

R0	R1	R2	R3	R4	R5	R6	R7
0000	0000	0000	0000	0000	0000	0000	0000

Z	C	V	N
0	0	0	0

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0000	1858	0006	2C40	6000	A000	0000	0007	0004	0000	0000	0000	0000	0000	0000	0000	0000
0010	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

	PC	SP	[SP+1]	IR	Instrucción - 1er palabra	PC	2da palabra	PC	3ra palabra	PC	Instrucción decodificada				
1	0000	FFEF	0000												
	Ejecución										Flags				
2															
	Ejecución										Flags				
3															
	Ejecución										Flags				

## Ejercicio 1 - Seguimiento

Valores  
iniciales

PC	SP
0000	FFEF

R0	R1	R2	R3	R4	R5	R6	R7
0000	0000	0000	0000	0000	0000	0000	0000

Z	C	V	N
0	0	0	0

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0000	1858	0006	2C40	6000	A000	0000	0007	0004	0000	0000	0000	0000	0000	0000	0000	0000
0010	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

	PC	SP	[SP+1]	IR	Instrucción - 1er palabra	PC	2da palabra	PC	3ra palabra	PC	Instrucción decodificada				
1	0000	FFEF	0000	1858											
	Ejecución										Flags				
2															
	Ejecución										Flags				
3															
	Ejecución										Flags				



## Ejercicio 1 - Seguimiento

Valores  
iniciales

PC	SP
0000	FFEF

R0	R1	R2	R3	R4	R5	R6	R7
0000	0000	0000	0000	0000	0000	0000	0000

Z	C	V	N
0	0	0	0

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0000	1858	0006	2C40	6000	A000	0000	0007	0004	0000	0000	0000	0000	0000	0000	0000	0000
0010	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

	PC	SP	[SP+1]	IR	Instrucción - 1er palabra	PC	2da palabra	PC	3ra palabra	PC	Instrucción decodificada				
1	0000	FFEF	0000	1858	0001 1000 0101 1000										
	Ejecución										Flags				
2															
	Ejecución										Flags				
3															
	Ejecución										Flags				

## Ejercicio 1 - Seguimiento

Valores  
iniciales

PC	SP
0000	FFEF

R0	R1	R2	R3	R4	R5	R6	R7
0000	0000	0000	0000	0000	0000	0000	0000

Z	C	V	N
0	0	0	0

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0000	1858	0006	2C40	6000	A000	0000	0007	0004	0000	0000	0000	0000	0000	0000	0000	0000
0010	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

	PC	SP	[SP+1]	IR	Instrucción - 1er palabra	PC	2da palabra	PC	3ra palabra	PC	Instrucción decodificada				
1	0000	FFEF	0000	1858	0001 1000 0101 1000	0001									
	Ejecución										Flags				
2															
	Ejecución										Flags				
3															
	Ejecución										Flags				

## Ejercicio 1 - Seguimiento

Valores  
iniciales

PC	SP
0000	FFEF

R0	R1	R2	R3	R4	R5	R6	R7
0000	0000	0000	0000	0000	0000	0000	0000

Z	C	V	N
0	0	0	0

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0000	1858	0006	2C40	6000	A000	0000	0007	0004	0000	0000	0000	0000	0000	0000	0000	0000
0010	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

	PC	SP	[SP+1]	IR	Instrucción - 1er palabra	PC	2da palabra	PC	3ra palabra	PC	Instrucción decodificada				
1	0000	FFEF	0000	1858	0001 1000 0101 1000	0001	0006	0002							
	Ejecución										Flags				
2															
	Ejecución										Flags				
3															
	Ejecución										Flags				

## Ejercicio 1 - Seguimiento

Valores  
iniciales

PC	SP
0000	FFEF

R0	R1	R2	R3	R4	R5	R6	R7
0000	0000	0000	0000	0000	0000	0000	0000

Z	C	V	N
0	0	0	0

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0000	1858	0006	2C40	6000	A000	0000	0007	0004	0000	0000	0000	0000	0000	0000	0000	0000
0010	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

	PC	SP	[SP+1]	IR	Instrucción - 1er palabra	PC	2da palabra	PC	3ra palabra	PC	Instrucción decodificada			
1	0000	FFEF	0000	1858	0001 1000 0101 1000	0001	0006	0002			MOV R1, [[0x0006]]			
	Ejecución										Flags			
2														
	Ejecución										Flags			
3														
	Ejecución										Flags			

## Ejercicio 1 - Seguimiento

Valores  
iniciales

PC	SP
0000	FFEF

R0	R1	R2	R3	R4	R5	R6	R7
0000	0000	0000	0000	0000	0000	0000	0000

Z	C	V	N
0	0	0	0

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0000	1858	0006	2C40	6000	A000	0000	0007	0004	0000	0000	0000	0000	0000	0000	0000	0000
0010	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

	PC	SP	[SP+1]	IR	Instrucción - 1er palabra	PC	2da palabra	PC	3ra palabra	PC	Instrucción decodificada				
1	0000	FFEF	0000	1858	0001 1000 0101 1000	0001	0006	0002			MOV R1, [[0x0006]]				
	Ejecución	R1=[[0x0006]]=[0x0007]=4									Flags				
2															
	Ejecución										Flags				
3															
	Ejecución										Flags				

## Ejercicio 1 - Seguimiento

Valores  
iniciales

PC	SP
0000	FFEF

R0	R1	R2	R3	R4	R5	R6	R7
0000	0000	0000	0000	0000	0000	0000	0000

Z	C	V	N
0	0	0	0

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0000	1858	0006	2C40	6000	A000	0000	0007	0004	0000	0000	0000	0000	0000	0000	0000	0000
0010	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

	PC	SP	[SP+1]	IR	Instrucción - 1er palabra	PC	2da palabra	PC	3ra palabra	PC	Instrucción decodificada				
1	0000	FFEF	0000	1858	0001 1000 0101 1000	0001	0006	0002			MOV R1, [[0x0006]]				
	Ejecución	R1=[[0x0006]]=[[0x0007]]=4									Flags				
2	0002														
	Ejecución										Flags				
3															
	Ejecución										Flags				

## Ejercicio 1 - Seguimiento

Valores  
iniciales

PC	SP
0000	FFEF

R0	R1	R2	R3	R4	R5	R6	R7
0000	0000	0000	0000	0000	0000	0000	0000

Z	C	V	N
0	0	0	0

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0000	1858	0006	2C40	6000	A000	0000	0007	0004	0000	0000	0000	0000	0000	0000	0000	0000
0010	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

	PC	SP	[SP+1]	IR	Instrucción - 1er palabra	PC	2da palabra	PC	3ra palabra	PC	Instrucción decodificada			
1	0000	FFEF	0000	1858	0001 1000 0101 1000	0001	0006	0002			MOV R1, [[0x0006]]			
	Ejecución	R1=[[0x0006]]=[[0x0007]]=4									Flags			
2	0002			2C40										
	Ejecución										Flags			
3														
	Ejecución										Flags			

## Ejercicio 1 - Seguimiento

Valores  
iniciales

PC	SP
0000	FFEF

R0	R1	R2	R3	R4	R5	R6	R7
0000	0000	0000	0000	0000	0000	0000	0000

Z	C	V	N
0	0	0	0

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0000	1858	0006	2C40	6000	A000	0000	0007	0004	0000	0000	0000	0000	0000	0000	0000	0000
0010	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

	PC	SP	[SP+1]	IR	Instrucción - 1er palabra	PC	2da palabra	PC	3ra palabra	PC	Instrucción decodificada			
1	0000	FFEF	0000	1858	0001 1000 0101 1000	0001	0006	0002			MOV R1, [[0x0006]]			
	Ejecución	R1=[[0x0006]]=[[0x0007]]=4									Flags			
2	0002			2C40	0010 1100 0100 0000									
	Ejecución										Flags			
3														
	Ejecución										Flags			



## Ejercicio 1 - Seguimiento

Valores  
iniciales

PC	SP
0000	FFEF

R0	R1	R2	R3	R4	R5	R6	R7
0000	0000	0000	0000	0000	0000	0000	0000

Z	C	V	N
0	0	0	0

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0000	1858	0006	2C40	6000	A000	0000	0007	0004	0000	0000	0000	0000	0000	0000	0000	0000
0010	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

	PC	SP	[SP+1]	IR	Instrucción - 1er palabra	PC	2da palabra	PC	3ra palabra	PC	Instrucción decodificada			
1	0000	FFEF	0000	1858	0001 1000 0101 1000	0001	0006	0002			MOV R1, [[0x0006]]			
	Ejecución	R1=[[0x0006]]=[[0x0007]]=4									Flags			
2	0002			2C40	0010 1100 0100 0000	0003								
	Ejecución										Flags			
3														
	Ejecución										Flags			

## Ejercicio 1 - Seguimiento

Valores  
iniciales

PC	SP
0000	FFEF

R0	R1	R2	R3	R4	R5	R6	R7
0000	0000	0000	0000	0000	0000	0000	0000

Z	C	V	N
0	0	0	0

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0000	1858	0006	2C40	6000	A000	0000	0007	0004	0000	0000	0000	0000	0000	0000	0000	0000
0010	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

	PC	SP	[SP+1]	IR	Instrucción - 1er palabra	PC	2da palabra	PC	3ra palabra	PC	Instrucción decodificada			
1	0000	FFEF	0000	1858	0001 1000 0101 1000	0001	0006	0002			MOV R1, [[0x0006]]			
	Ejecución	R1=[[0x0006]]=[[0x0007]]=4									Flags			
2	0002			2C40	0010 1100 0100 0000	0003	6000	0004						
	Ejecución										Flags			
3														
	Ejecución										Flags			

## Ejercicio 1 - Seguimiento

Valores  
iniciales

PC	SP
0000	FFEF

R0	R1	R2	R3	R4	R5	R6	R7
0000	0000	0000	0000	0000	0000	0000	0000

Z	C	V	N
0	0	0	0

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0000	1858	0006	2C40	6000	A000	0000	0007	0004	0000	0000	0000	0000	0000	0000	0000	0000
0010	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

	PC	SP	[SP+1]	IR	Instrucción - 1er palabra	PC	2da palabra	PC	3ra palabra	PC	Instrucción decodificada			
1	0000	FFEF	0000	1858	0001 1000 0101 1000	0001	0006	0002			MOV R1, [[0x0006]]			
	Ejecución	R1=[[0x0006]]=[[0x0007]]=4									Flags			
2	0002			2C40	0010 1100 0100 0000	0003	6000	0004			ADD [R1],0x6000			
	Ejecución										Flags			
3														
	Ejecución										Flags			

## Ejercicio 1 - Seguimiento

Valores  
iniciales

PC	SP
0000	FFEF

R0	R1	R2	R3	R4	R5	R6	R7
0000	0000	0000	0000	0000	0000	0000	0000

Z	C	V	N
0	0	0	0

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0000	1858	0006	2C40	6000	A000	0000	0007	0004	0000	0000	0000	0000	0000	0000	0000	0000
0010	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

	PC	SP	[SP+1]	IR	Instrucción - 1er palabra	PC	2da palabra	PC	3ra palabra	PC	Instrucción decodificada				
1	0000	FFEF	0000	1858	0001 1000 0101 1000	0001	0006	0002			MOV R1, [[0x0006]]				
	Ejecución	R1=[[0x0006]]=0x0007=4									Flags				
2	0002			2C40	0010 1100 0100 0000	0003	6000	0004			ADD [R1],0x6000				
	Ejecución	[0x0004]=0xA000+0x6000=0x0000									Flags				
3															
	Ejecución										Flags				

## Ejercicio 1 - Seguimiento

Valores  
iniciales

PC	SP
0000	FFEF

R0	R1	R2	R3	R4	R5	R6	R7
0000	0000	0000	0000	0000	0000	0000	0000

Z	C	V	N
0	0	0	0

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0000	1858	0006	2C40	6000	0000	0000	0007	0004	0000	0000	0000	0000	0000	0000	0000	0000
0010	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

	PC	SP	[SP+1]	IR	Instrucción - 1er palabra	PC	2da palabra	PC	3ra palabra	PC	Instrucción decodificada				
1	0000	FFEF	0000	1858	0001 1000 0101 1000	0001	0006	0002			MOV R1, [[0x0006]]				
	Ejecución	R1=[[0x0006]]=0x0007=4									Flags				
2	0002			2C40	0010 1100 0100 0000	0003	6000	0004			ADD [R1],0x6000				
	Ejecución	[0x0004]=0xA000+0x6000=0x0000									Flags	1	1	0	0
3															
	Ejecución										Flags				

## Ejercicio 1 - Seguimiento

Valores  
iniciales

PC	SP
0000	FFEF

R0	R1	R2	R3	R4	R5	R6	R7
0000	0000	0000	0000	0000	0000	0000	0000

Z	C	V	N
0	0	0	0

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0000	1858	0006	2C40	6000	0000	0000	0007	0004	0000	0000	0000	0000	0000	0000	0000	0000
0010	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

	PC	SP	[SP+1]	IR	Instrucción - 1er palabra	PC	2da palabra	PC	3ra palabra	PC	Instrucción decodificada				
1	0000	FFEF	0000	1858	0001 1000 0101 1000	0001	0006	0002			MOV R1, [[0x0006]]				
	Ejecución	R1=[[0x0006]]=0x0007=4									Flags				
2	0002			2C40	0010 1100 0100 0000	0003	6000	0004			ADD [R1],0x6000				
	Ejecución	[0x0004]=0xA000+0x6000=0x0000									Flags	1	1	0	0
3	0004														
	Ejecución										Flags				

## Ejercicio 1 - Seguimiento

Valores  
iniciales

PC	SP
0000	FFEF

R0	R1	R2	R3	R4	R5	R6	R7
0000	0000	0000	0000	0000	0000	0000	0000

Z	C	V	N
0	0	0	0

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0000	1858	0006	2C40	6000	0000	0000	0007	0004	0000	0000	0000	0000	0000	0000	0000	0000
0010	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

	PC	SP	[SP+1]	IR	Instrucción - 1er palabra	PC	2da palabra	PC	3ra palabra	PC	Instrucción decodificada				
1	0000	FFEF	0000	1858	0001 1000 0101 1000	0001	0006	0002			MOV R1, [[0x0006]]				
	Ejecución	R1=[[0x0006]]=0x0007=4									Flags				
2	0002			2C40	0010 1100 0100 0000	0003	6000	0004			ADD [R1],0x6000				
	Ejecución	[0x0004]=0xA000+0x6000=0x0000									Flags	1	1	0	0
3	0004			0000											
	Ejecución										Flags				

## Ejercicio 1 - Seguimiento

Valores  
iniciales

PC	SP
0000	FFEF

R0	R1	R2	R3	R4	R5	R6	R7
0000	0000	0000	0000	0000	0000	0000	0000

Z	C	V	N
0	0	0	0

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0000	1858	0006	2C40	6000	0000	0000	0007	0004	0000	0000	0000	0000	0000	0000	0000	0000
0010	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

	PC	SP	[SP+1]	IR	Instrucción - 1er palabra	PC	2da palabra	PC	3ra palabra	PC	Instrucción decodificada				
1	0000	FFEF	0000	1858	0001 1000 0101 1000	0001	0006	0002			MOV R1, [[0x0006]]				
	Ejecución	R1=[[0x0006]]=0x0007=4									Flags				
2	0002			2C40	0010 1100 0100 0000	0003	6000	0004			ADD [R1],0x6000				
	Ejecución	[0x0004]=0xA000+0x6000=0x0000									Flags	1	1	0	0
3	0004			0000	0000 0000 0000 0000										
	Ejecución										Flags				



## Ejercicio 1 - Seguimiento

Valores  
iniciales

PC	SP
0000	FFEF

R0	R1	R2	R3	R4	R5	R6	R7
0000	0000	0000	0000	0000	0000	0000	0000

Z	C	V	N
0	0	0	0

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0000	1858	0006	2C40	6000	0000	0000	0007	0004	0000	0000	0000	0000	0000	0000	0000	0000
0010	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

	PC	SP	[SP+1]	IR	Instrucción - 1er palabra	PC	2da palabra	PC	3ra palabra	PC	Instrucción decodificada				
1	0000	FFEF	0000	1858	0001 1000 0101 1000	0001	0006	0002			MOV R1, [[0x0006]]				
	Ejecución	R1=[[0x0006]]=0x0007=4									Flags				
2	0002			2C40	0010 1100 0100 0000	0003	6000	0004			ADD [R1],0x6000				
	Ejecución	[0x0004]=0xA000+0x6000=0x0000									Flags	1	1	0	0
3	0004			0000	0000 0000 0000 0000										
	Ejecución										Flags				

## Ejercicio 1 - Seguimiento

Valores  
iniciales

PC	SP
0000	FFEF

R0	R1	R2	R3	R4	R5	R6	R7
0000	0000	0000	0000	0000	0000	0000	0000

Z	C	V	N
0	0	0	0

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0000	1858	0006	2C40	6000	0000	0000	0007	0004	0000	0000	0000	0000	0000	0000	0000	0000
0010	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

	PC	SP	[SP+1]	IR	Instrucción - 1er palabra	PC	2da palabra	PC	3ra palabra	PC	Instrucción decodificada				
1	0000	FFEF	0000	1858	0001 1000 0101 1000	0001	0006	0002			MOV R1, [[0x0006]]				
	Ejecución	R1=[[0x0006]]=0x0007=4									Flags				
2	0002			2C40	0010 1100 0100 0000	0003	6000	0004			ADD [R1],0x6000				
	Ejecución	[0x0004]=0xA000+0x6000=0x0000									Flags	1	1	0	0
3	0004			0000	0000 0000 0000 0000	0005									
	Ejecución										Flags				

## Ejercicio 1 - Seguimiento

Valores  
iniciales

PC	SP
0000	FFEF

R0	R1	R2	R3	R4	R5	R6	R7
0000	0000	0000	0000	0000	0000	0000	0000

Z	C	V	N
0	0	0	0

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0000	1858	0006	2C40	6000	0000	0000	0007	0004	0000	0000	0000	0000	0000	0000	0000	0000
0010	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

	PC	SP	[SP+1]	IR	Instrucción - 1er palabra	PC	2da palabra	PC	3ra palabra	PC	Instrucción decodificada				
1	0000	FFEF	0000	1858	0001 1000 0101 1000	0001	0006	0002			MOV R1, [[0x0006]]				
	Ejecución	R1=[[0x0006]]=0x0007=4									Flags				
2	0002			2C40	0010 1100 0100 0000	0003	6000	0004			ADD [R1],0x6000				
	Ejecución	[0x0004]=0xA000+0x6000=0x0000									Flags	1	1	0	0
3	0004			0000	0000 0000 0000 0000	0005					Instrucción Inválida				
	Ejecución										Flags				

## Ejercicio 1 - Seguimiento

Valores  
iniciales

PC	SP
0000	FFEF

R0	R1	R2	R3	R4	R5	R6	R7
0000	0000	0000	0000	0000	0000	0000	0000

Z	C	V	N
0	0	0	0

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0000	1858	0006	2C40	6000	0000	0000	0007	0004	0000	0000	0000	0000	0000	0000	0000	0000
0010	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

	PC	SP	[SP+1]	IR	Instrucción - 1er palabra	PC	2da palabra	PC	3ra palabra	PC	Instrucción decodificada				
1	0000	FFEF	0000	1858	0001 1000 0101 1000	0001	0006	0002			MOV R1, [[0x0006]]				
	Ejecución	R1=[[0x0006]]=0x0007=4									Flags				
2	0002			2C40	0010 1100 0100 0000	0003	6000	0004			ADD [R1],0x6000				
	Ejecución	[0x0004]=0xA000+0x6000=0x0000									Flags	1	1	0	0
3	0004			0000	0000 0000 0000 0000	0005					Instrucción Inválida				
	Ejecución	FIN DE LA EJECUCIÓN									Flags				

## Ejercicio 1 - Seguimiento

## Ejercicio 1 - Solución

0000

Planilla de Seguimiento

Valores iniciales:	PC	SP													Z	C	V	N
	0000	FFEF													0	0	0	0
Memoria:		+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	
	0000	1858	0006	2C40	6000	A000	0000	0007	0004	0000	0000	0000	0000	0000	0000	0000	0000	
	0010	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	

	PC	SP	[SP+1]	IR	Instrucción - 1 <sup>er</sup> palabra (en bits)				PC	2 <sup>do</sup> palabra	PC	3 <sup>er</sup> a palabra	PC	Instrucción decodificada				
1	0000	FFEF	0000	1858	0001	1000	0101	1000	0001	0006	0002			MOV R1, [[0x0006]]				
	Ejecución		R1 = [[0x0006]] = [0x0007] = 4											Flags <sup>1</sup>	Z	C	V	N
2	0002			2C40	0010	1100	0100	0000	0003	6000	0004			ADD [R1], 0x6000				
	Ejecución		[0x0004] = 0xA000 + 0x6000 = 0x0000											Flags <sup>1</sup>	Z	C	V	N
3	0004			0000	0000	0000	0000	0000	0005					Instrucción inválida				
	Ejecución		FIN DE LA EJECUCIÓN											Flags <sup>1</sup>	Z	C	V	N

# Ejercicio 1 - Observaciones

**¿Qué pasó con el programa anterior?**

# Ejercicio 1 - Observaciones

**¿Qué pasó con el programa anterior?**

En la ejecución del programa cambió una posición de memoria donde estaba cargado... **el mismo programa!**

# Ejercicio 1 - Observaciones

## ¿Qué pasó con el programa anterior?

En la ejecución del programa cambió una posición de memoria donde estaba cargado... **el mismo programa!**

Eso hizo que nuestro programa se modificara, haciendo que la máquina llegase a una instrucción inválida.



# Ejercicio 1 - Observaciones

## ¿Qué pasó con el programa anterior?

En la ejecución del programa cambió una posición de memoria donde estaba cargado... **el mismo programa!**

Eso hizo que nuestro programa se modificara, haciendo que la máquina llegase a una instrucción inválida.

**Moraleja:**



# Ejercicio 2

Realizar el seguimiento de la ejecución del siguiente programa cargado en memoria a partir de la dirección 0x0000.

0000	0001	0002	0003	0004	0005	0006	0007
19C8	0004	B000	0005	0FE0	39A7	C000	0000

## Ejercicio 2 - Seguimiento

## Ejercicio 2 - Seguimiento

Valores	PC	SP	R0	R1	R2	R3	R4	R5	R6	R7	Z	C	V	N
iniciales	0000	FFEF	0000	0000	0000	0000	0000	0000	0000	0000	0	0	0	0

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0000	19CB	0004	B000	0005	0FE0	39A7	C000	0000	0000	0000	0000	0000	0000	0000	0000	0000
0010	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

	PC	SP	[SP+1]	IR	Instrucción - 1er palabra	PC	2da palabra	PC	3ra palabra	PC	Instrucción decodificada				
1															
	Ejecución										Flags				
2															
	Ejecución										Flags				
3															
	Ejecución										Flags				
4															
	Ejecución										Flags				
5															
	Ejecución										Flags				

## Ejercicio 2 - Seguimiento

## Ejercicio 2 - Seguimiento

Valores	PC	SP	R0	R1	R2	R3	R4	R5	R6	R7	Z	C	V	N
iniciales	0000	FFEF	0000	0000	0000	0000	0000	0000	0000	0000	0	0	0	0

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0000	19CB	0004	B000	0005	0FE0	39A7	C000	0000	0000	0000	0000	0000	0000	0000	0000	0000
0010	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

	PC	SP	[SP+1]	IR	Instrucción - 1er palabra	PC	2da palabra	PC	3ra palabra	PC	Instrucción decodificada
1	0000										
	Ejecución										Flags
2											
	Ejecución										Flags
3											
	Ejecución										Flags
4											
	Ejecución										Flags
5											
	Ejecución										Flags

## Ejercicio 2 - Seguimiento

## Ejercicio 2 - Seguimiento

Valores	PC	SP	R0	R1	R2	R3	R4	R5	R6	R7	Z	C	V	N
iniciales	0000	FFEF	0000	0000	0000	0000	0000	0000	0000	0000	0	0	0	0

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0000	19CB	0004	B000	0005	0FE0	39A7	C000	0000	0000	0000	0000	0000	0000	0000	0000	0000
0010	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

	PC	SP	[SP+1]	IR	Instrucción - 1er palabra	PC	2da palabra	PC	3ra palabra	PC	Instrucción decodificada				
1	0000	FFEF	0000												
	Ejecución										Flags				
2															
	Ejecución										Flags				
3															
	Ejecución										Flags				
4															
	Ejecución										Flags				
5															
	Ejecución										Flags				

## Ejercicio 2 - Seguimiento

## Ejercicio 2 - Seguimiento

Valores	PC	SP	R0	R1	R2	R3	R4	R5	R6	R7	Z	C	V	N
iniciales	0000	FFEF	0000	0000	0000	0000	0000	0000	0000	0000	0	0	0	0

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0000	19CB	0004	B000	0005	0FE0	39A7	C000	0000	0000	0000	0000	0000	0000	0000	0000	0000
0010	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

	PC	SP	[SP+1]	IR	Instrucción - 1er palabra	PC	2da palabra	PC	3ra palabra	PC	Instrucción decodificada				
1	0000	FFEF	0000	19CB											
	Ejecución										Flags				
2															
	Ejecución										Flags				
3															
	Ejecución										Flags				
4															
	Ejecución										Flags				
5															
	Ejecución										Flags				

## Ejercicio 2 - Seguimiento

## Ejercicio 2 - Seguimiento

Valores	PC	SP	R0	R1	R2	R3	R4	R5	R6	R7	Z	C	V	N
iniciales	0000	FFEF	0000	0000	0000	0000	0000	0000	0000	0000	0	0	0	0

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0000	19CB	0004	B000	0005	0FE0	39A7	C000	0000	0000	0000	0000	0000	0000	0000	0000	0000
0010	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

	PC	SP	[SP+1]	IR	Instrucción - 1er palabra	PC	2da palabra	PC	3ra palabra	PC	Instrucción decodificada				
1	0000	FFEF	0000	19CB	0001 1001 1100 1000										
	Ejecución										Flags				
2															
	Ejecución										Flags				
3															
	Ejecución										Flags				
4															
	Ejecución										Flags				
5															
	Ejecución										Flags				

## Ejercicio 2 - Seguimiento

## Ejercicio 2 - Seguimiento

Valores	PC	SP	R0	R1	R2	R3	R4	R5	R6	R7	Z	C	V	N
iniciales	0000	FFEF	0000	0000	0000	0000	0000	0000	0000	0000	0	0	0	0

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0000	19CB	0004	B000	0005	0FE0	39A7	C000	0000	0000	0000	0000	0000	0000	0000	0000	0000
0010	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

	PC	SP	[SP+1]	IR	Instrucción - 1er palabra	PC	2da palabra	PC	3ra palabra	PC	Instrucción decodificada				
1	0000	FFEF	0000	19CB	0001 1001 1100 1000	0001									
	Ejecución										Flags				
2															
	Ejecución										Flags				
3															
	Ejecución										Flags				
4															
	Ejecución										Flags				
5															
	Ejecución										Flags				



## Ejercicio 2 - Seguimiento

## Ejercicio 2 - Seguimiento

Valores	PC	SP	R0	R1	R2	R3	R4	R5	R6	R7	Z	C	V	N
iniciales	0000	FFEF	0000	0000	0000	0000	0000	0000	0000	0000	0	0	0	0

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0000	19CB	0004	B000	0005	0FE0	39A7	C000	0000	0000	0000	0000	0000	0000	0000	0000	0000
0010	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

	PC	SP	[SP+1]	IR	Instrucción - 1er palabra	PC	2da palabra	PC	3ra palabra	PC	Instrucción decodificada				
1	0000	FFEF	0000	19CB	0001 1001 1100 1000	0001	0004	0002							
	Ejecución										Flags				
2															
	Ejecución										Flags				
3															
	Ejecución										Flags				
4															
	Ejecución										Flags				
5															
	Ejecución										Flags				

## Ejercicio 2 - Seguimiento

## Ejercicio 2 - Seguimiento

Valores	PC	SP	R0	R1	R2	R3	R4	R5	R6	R7	Z	C	V	N
iniciales	0000	FFEF	0000	0000	0000	0000	0000	0000	0000	0000	0	0	0	0

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0000	19CB	0004	B000	0005	0FE0	39A7	C000	0000	0000	0000	0000	0000	0000	0000	0000	0000
0010	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

	PC	SP	[SP+1]	IR	Instrucción - 1er palabra	PC	2da palabra	PC	3ra palabra	PC	Instrucción decodificada				
1	0000	FFEF	0000	19CB	0001 1001 1100 1000	0001	0004	0002			MOV R7, [0x0004]				
	Ejecución										Flags				
2															
	Ejecución										Flags				
3															
	Ejecución										Flags				
4															
	Ejecución										Flags				
5															
	Ejecución										Flags				

## Ejercicio 2 - Seguimiento

## Ejercicio 2 - Seguimiento

Valores	PC	SP	R0	R1	R2	R3	R4	R5	R6	R7	Z	C	V	N
iniciales	0000	FFEF	0000	0000	0000	0000	0000	0000	0000	0000	0	0	0	0

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0000	19CB	0004	B000	0005	0FE0	39A7	C000	0000	0000	0000	0000	0000	0000	0000	0000	0000
0010	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

	PC	SP	[SP+1]	IR	Instrucción - 1er palabra	PC	2da palabra	PC	3ra palabra	PC	Instrucción decodificada				
1	0000	FFEF	0000	19CB	0001 1001 1100 1000	0001	0004	0002			MOV R7, [0x0004]				
	Ejecución	R7=[0x0004]=0x0FE0									Flags				
2															
	Ejecución										Flags				
3															
	Ejecución										Flags				
4															
	Ejecución										Flags				
5															
	Ejecución										Flags				

## Ejercicio 2 - Seguimiento

## Ejercicio 2 - Seguimiento

Valores	PC	SP	R0	R1	R2	R3	R4	R5	R6	R7	Z	C	V	N
iniciales	0000	FFEF	0000	0000	0000	0000	0000	0000	0000	0000	0	0	0	0

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0000	19CB	0004	B000	0005	0FE0	39A7	C000	0000	0000	0000	0000	0000	0000	0000	0000	0000
0010	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

	PC	SP	[SP+1]	IR	Instrucción - 1er palabra	PC	2da palabra	PC	3ra palabra	PC	Instrucción decodificada				
1	0000	FFEF	0000	19CB	0001 1001 1100 1000	0001	0004	0002			MOV R7, [0x0004]				
	Ejecución	R7=[0x0004]=0x0FE0									Flags				
2	0002														
	Ejecución										Flags				
3															
	Ejecución										Flags				
4															
	Ejecución										Flags				
5															
	Ejecución										Flags				

## Ejercicio 2 - Seguimiento

## Ejercicio 2 - Seguimiento

Valores	PC	SP	R0	R1	R2	R3	R4	R5	R6	R7	Z	C	V	N
iniciales	0000	FFEF	0000	0000	0000	0000	0000	0000	0000	0000	0	0	0	0

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0000	19CB	0004	B000	0005	0FE0	39A7	C000	0000	0000	0000	0000	0000	0000	0000	0000	0000
0010	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

	PC	SP	[SP+1]	IR	Instrucción - 1er palabra	PC	2da palabra	PC	3ra palabra	PC	Instrucción decodificada				
1	0000	FFEF	0000	19CB	0001 1001 1100 1000	0001	0004	0002			MOV R7, [0x0004]				
	Ejecución	R7=[0x0004]=0x0FE0									Flags				
2	0002			B000											
	Ejecución										Flags				
3															
	Ejecución										Flags				
4															
	Ejecución										Flags				
5															
	Ejecución										Flags				

## Ejercicio 2 - Seguimiento

## Ejercicio 2 - Seguimiento

Valores	PC	SP	R0	R1	R2	R3	R4	R5	R6	R7	Z	C	V	N
iniciales	0000	FFEF	0000	0000	0000	0000	0000	0000	0000	0000	0	0	0	0

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0000	19CB	0004	B000	0005	0FE0	39A7	C000	0000	0000	0000	0000	0000	0000	0000	0000	0000
0010	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

	PC	SP	[SP+1]	IR	Instrucción - 1er palabra	PC	2da palabra	PC	3ra palabra	PC	Instrucción decodificada				
1	0000	FFEF	0000	19CB	0001 1001 1100 1000	0001	0004	0002			MOV R7, [0x0004]				
	Ejecución	R7=[0x0004]=0x0FE0									Flags				
2	0002			B000	1011 0000 0000 0000										
	Ejecución										Flags				
3															
	Ejecución										Flags				
4															
	Ejecución										Flags				
5															
	Ejecución										Flags				

## Ejercicio 2 - Seguimiento

## Ejercicio 2 - Seguimiento

Valores	PC	SP	R0	R1	R2	R3	R4	R5	R6	R7	Z	C	V	N
iniciales	0000	FFEF	0000	0000	0000	0000	0000	0000	0000	0000	0	0	0	0

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0000	19CB	0004	B000	0005	0FE0	39A7	C000	0000	0000	0000	0000	0000	0000	0000	0000	0000
0010	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

	PC	SP	[SP+1]	IR	Instrucción - 1er palabra	PC	2da palabra	PC	3ra palabra	PC	Instrucción decodificada				
1	0000	FFEF	0000	19CB	0001 1001 1100 1000	0001	0004	0002			MOV R7, [0x0004]				
	Ejecución	R7=[0x0004]=0x0FE0									Flags				
2	0002			B000	1011 0000 0000 0000	0003									
	Ejecución										Flags				
3															
	Ejecución										Flags				
4															
	Ejecución										Flags				
5															
	Ejecución										Flags				

## Ejercicio 2 - Seguimiento

## Ejercicio 2 - Seguimiento

Valores	PC	SP	R0	R1	R2	R3	R4	R5	R6	R7	Z	C	V	N
iniciales	0000	FFEF	0000	0000	0000	0000	0000	0000	0000	0000	0	0	0	0

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0000	19CB	0004	B000	0005	0FE0	39A7	C000	0000	0000	0000	0000	0000	0000	0000	0000	0000
0010	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

	PC	SP	[SP+1]	IR	Instrucción - 1er palabra	PC	2da palabra	PC	3ra palabra	PC	Instrucción decodificada				
1	0000	FFEF	0000	19CB	0001 1001 1100 1000	0001	0004	0002			MOV R7, [0x0004]				
	Ejecución	R7=[0x0004]=0x0FE0									Flags				
2	0002			B000	1011 0000 0000 0000	0003	0005	0004							
	Ejecución										Flags				
3															
	Ejecución										Flags				
4															
	Ejecución										Flags				
5															
	Ejecución										Flags				



## Ejercicio 2 - Seguimiento

## Ejercicio 2 - Seguimiento

Valores	PC	SP	R0	R1	R2	R3	R4	R5	R6	R7	Z	C	V	N
iniciales	0000	FFEF	0000	0000	0000	0000	0000	0000	0000	0000	0	0	0	0

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0000	19CB	0004	B000	0005	0FE0	39A7	C000	0000	0000	0000	0000	0000	0000	0000	0000	0000
0010	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

	PC	SP	[SP+1]	IR	Instrucción - 1er palabra	PC	2da palabra	PC	3ra palabra	PC	Instrucción decodificada				
1	0000	FFEF	0000	19CB	0001 1001 1100 1000	0001	0004	0002			MOV R7, [0x0004]				
	Ejecución	R7=[0x0004]=0x0FE0									Flags				
2	0002			B000	1011 0000 0000 0000	0003	0005	0004			CALL 0x0005				
	Ejecución										Flags				
3															
	Ejecución										Flags				
4															
	Ejecución										Flags				
5															
	Ejecución										Flags				

## Ejercicio 2 - Seguimiento

## Ejercicio 2 - Seguimiento

Valores	PC	SP	R0	R1	R2	R3	R4	R5	R6	R7	Z	C	V	N
iniciales	0000	FFEF	0000	0000	0000	0000	0000	0000	0000	0000	0	0	0	0

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0000	19CB	0004	B000	0005	0FE0	39A7	C000	0000	0000	0000	0000	0000	0000	0000	0000	0000
0010	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

	PC	SP	[SP+1]	IR	Instrucción - 1er palabra	PC	2da palabra	PC	3ra palabra	PC	Instrucción decodificada				
1	0000	FFEF	0000	19CB	0001 1001 1100 1000	0001	0004	0002			MOV R7, [0x0004]				
	Ejecución	R7=[0x0004]=0x0FE0									Flags				
2	0002			B000	1011 0000 0000 0000	0003	0005	0004			CALL 0x0005				
	Ejecución	[0xFFEF]=0x0004 SP=0xFFEE PC=0x0005									Flags				
3															
	Ejecución										Flags				
4															
	Ejecución										Flags				
5															
	Ejecución										Flags				

## Ejercicio 2 - Seguimiento

## Ejercicio 2 - Seguimiento

Valores	PC	SP	R0	R1	R2	R3	R4	R5	R6	R7	Z	C	V	N
iniciales	0000	FFEF	0000	0000	0000	0000	0000	0000	0000	0000	0	0	0	0

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0000	19CB	0004	B000	0005	0FE0	39A7	C000	0000	0000	0000	0000	0000	0000	0000	0000	0000
0010	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

	PC	SP	[SP+1]	IR	Instrucción - 1er palabra	PC	2da palabra	PC	3ra palabra	PC	Instrucción decodificada				
1	0000	FFEF	0000	19CB	0001 1001 1100 1000	0001	0004	0002			MOV R7, [0x0004]				
	Ejecución	R7=[0x0004]=0x0FE0									Flags				
2	0002			B000	1011 0000 0000 0000	0003	0005	0004			CALL 0x0005				
	Ejecución	[0xFFEF]=0x0004 SP=0xFFEE PC=0x0005									Flags				
3	0005														
	Ejecución										Flags				
4															
	Ejecución										Flags				
5															
	Ejecución										Flags				

## Ejercicio 2 - Seguimiento

## Ejercicio 2 - Seguimiento

Valores	PC	SP	R0	R1	R2	R3	R4	R5	R6	R7	Z	C	V	N
iniciales	0000	FFEF	0000	0000	0000	0000	0000	0000	0000	0000	0	0	0	0

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0000	19CB	0004	B000	0005	0FE0	39A7	C000	0000	0000	0000	0000	0000	0000	0000	0000	0000
0010	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

	PC	SP	[SP+1]	IR	Instrucción - 1er palabra	PC	2da palabra	PC	3ra palabra	PC	Instrucción decodificada				
1	0000	FFEF	0000	19CB	0001 1001 1100 1000	0001	0004	0002			MOV R7, [0x0004]				
	Ejecución	R7=[0x0004]=0x0FE0									Flags				
2	0002			B000	1011 0000 0000 0000	0003	0005	0004			CALL 0x0005				
	Ejecución	[0xFFEF]=0x0004 SP=0xFFEE PC=0x0005									Flags				
3	0005	FFEE													
	Ejecución										Flags				
4															
	Ejecución										Flags				
5															
	Ejecución										Flags				

## Ejercicio 2 - Seguimiento

## Ejercicio 2 - Seguimiento

Valores	PC	SP	R0	R1	R2	R3	R4	R5	R6	R7	Z	C	V	N
iniciales	0000	FFEF	0000	0000	0000	0000	0000	0000	0000	0000	0	0	0	0

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0000	19CB	0004	B000	0005	0FE0	39A7	C000	0000	0000	0000	0000	0000	0000	0000	0000	0000
0010	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

	PC	SP	[SP+1]	IR	Instrucción - 1er palabra	PC	2da palabra	PC	3ra palabra	PC	Instrucción decodificada				
1	0000	FFEF	0000	19CB	0001 1001 1100 1000	0001	0004	0002			MOV R7, [0x0004]				
	Ejecución	R7=[0x0004]=0x0FE0									Flags				
2	0002			B000	1011 0000 0000 0000	0003	0005	0004			CALL 0x0005				
	Ejecución	[0xFFEF]=0x0004 SP=0xFFEE PC=0x0005									Flags				
3	0005	FFEE	0004	39A7											
	Ejecución										Flags				
4															
	Ejecución										Flags				
5															
	Ejecución										Flags				

## Ejercicio 2 - Seguimiento

## Ejercicio 2 - Seguimiento

Valores	PC	SP	R0	R1	R2	R3	R4	R5	R6	R7	Z	C	V	N
iniciales	0000	FFEF	0000	0000	0000	0000	0000	0000	0000	0000	0	0	0	0

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0000	19CB	0004	B000	0005	0FE0	39A7	C000	0000	0000	0000	0000	0000	0000	0000	0000	0000
0010	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

	PC	SP	[SP+1]	IR	Instrucción - 1er palabra	PC	2da palabra	PC	3ra palabra	PC	Instrucción decodificada				
1	0000	FFEF	0000	19CB	0001 1001 1100 1000	0001	0004	0002			MOV R7, [0x0004]				
	Ejecución	R7=[0x0004]=0x0FE0									Flags				
2	0002			B000	1011 0000 0000 0000	0003	0005	0004			CALL 0x0005				
	Ejecución	[0xFFEF]=0x0004 SP=0xFFEE PC=0x0005									Flags				
3	0005	FFEE	0004	39A7	0011 1001 1010 0111										
	Ejecución										Flags				
4															
	Ejecución										Flags				
5															
	Ejecución										Flags				

## Ejercicio 2 - Seguimiento

## Ejercicio 2 - Seguimiento

Valores	PC	SP	R0	R1	R2	R3	R4	R5	R6	R7	Z	C	V	N
iniciales	0000	FFEF	0000	0000	0000	0000	0000	0000	0000	0000	0	0	0	0

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0000	19CB	0004	B000	0005	0FE0	39A7	C000	0000	0000	0000	0000	0000	0000	0000	0000	0000
0010	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

	PC	SP	[SP+1]	IR	Instrucción - 1er palabra	PC	2da palabra	PC	3ra palabra	PC	Instrucción decodificada				
1	0000	FFEF	0000	19CB	0001 1001 1100 1000	0001	0004	0002			MOV R7, [0x0004]				
	Ejecución	R7=[0x0004]=0x0FE0									Flags				
2	0002			B000	1011 0000 0000 0000	0003	0005	0004			CALL 0x0005				
	Ejecución	[0xFFEF]=0x0004 SP=0xFFEE PC=0x0005									Flags				
3	0005	FFEE	0004	39A7	0011 1001 1010 0111	0006									
	Ejecución										Flags				
4															
	Ejecución										Flags				
5															
	Ejecución										Flags				

## Ejercicio 2 - Seguimiento

## Ejercicio 2 - Seguimiento

Valores	PC	SP	R0	R1	R2	R3	R4	R5	R6	R7	Z	C	V	N
iniciales	0000	FFEF	0000	0000	0000	0000	0000	0000	0000	0000	0	0	0	0

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0000	19CB	0004	B000	0005	0FE0	39A7	C000	0000	0000	0000	0000	0000	0000	0000	0000	0000
0010	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

	PC	SP	[SP+1]	IR	Instrucción - 1er palabra	PC	2da palabra	PC	3ra palabra	PC	Instrucción decodificada				
1	0000	FFEF	0000	19CB	0001 1001 1100 1000	0001	0004	0002			MOV R7, [0x0004]				
	Ejecución	R7=[0x0004]=0x0FE0									Flags				
2	0002			B000	1011 0000 0000 0000	0003	0005	0004			CALL 0x0005				
	Ejecución	[0xFFEF]=0x0004 SP=0xFFEE PC=0x0005									Flags				
3	0005	FFEE	0004	39A7	0011 1001 1010 0111	0006					SUB R6, R7				
	Ejecución										Flags				
4															
	Ejecución										Flags				
5															
	Ejecución										Flags				



## Ejercicio 2 - Seguimiento

## Ejercicio 2 - Seguimiento

Valores	PC	SP	R0	R1	R2	R3	R4	R5	R6	R7	Z	C	V	N
iniciales	0000	FFEF	0000	0000	0000	0000	0000	0000	0000	0000	0	0	0	0

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0000	19CB	0004	B000	0005	0FE0	39A7	C000	0000	0000	0000	0000	0000	0000	0000	0000	0000
0010	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

	PC	SP	[SP+1]	IR	Instrucción - 1er palabra	PC	2da palabra	PC	3ra palabra	PC	Instrucción decodificada				
1	0000	FFEF	0000	19CB	0001 1001 1100 1000	0001	0004	0002			MOV R7, [0x0004]				
	Ejecución	R7=[0x0004]=0x0FE0									Flags				
2	0002			B000	1011 0000 0000 0000	0003	0005	0004			CALL 0x0005				
	Ejecución	[0xFFEF]=0x0004 SP=0xFFEE PC=0x0005									Flags				
3	0005	FFEE	0004	39A7	0011 1001 1010 0111	0006					SUB R6, R7				
	Ejecución	R6=0x0000-0x0FE0=0xF020									Flags				
4															
	Ejecución										Flags				
5															
	Ejecución										Flags				

## Ejercicio 2 - Seguimiento

## Ejercicio 2 - Seguimiento

Valores	PC	SP	R0	R1	R2	R3	R4	R5	R6	R7	Z	C	V	N
iniciales	0000	FFEF	0000	0000	0000	0000	0000	0000	F020	0000	0	0	0	0

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0000	19CB	0004	B000	0005	0FE0	39A7	C000	0000	0000	0000	0000	0000	0000	0000	0000	0000
0010	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

	PC	SP	[SP+1]	IR	Instrucción - 1er palabra	PC	2da palabra	PC	3ra palabra	PC	Instrucción decodificada				
1	0000	FFEF	0000	19CB	0001 1001 1100 1000	0001	0004	0002			MOV R7, [0x0004]				
	Ejecución	R7=[0x0004]=0x0FE0									Flags				
2	0002			B000	1011 0000 0000 0000	0003	0005	0004			CALL 0x0005				
	Ejecución	[0xFFEF]=0x0004 SP=0xFFEE PC=0x0005									Flags				
3	0005	FFEE	0004	39A7	0011 1001 1010 0111	0006					SUB R6, R7				
	Ejecución	R6=0x0000-0x0FE0=0xF020									Flags	0	1	0	1
4															
	Ejecución										Flags				
5															
	Ejecución										Flags				

## Ejercicio 2 - Seguimiento

## Ejercicio 2 - Seguimiento

Valores	PC	SP	R0	R1	R2	R3	R4	R5	R6	R7	Z	C	V	N
iniciales	0000	FFEF	0000	0000	0000	0000	0000	0000	F020	0000	0	0	0	0

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0000	19CB	0004	B000	0005	0FE0	39A7	C000	0000	0000	0000	0000	0000	0000	0000	0000	0000
0010	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

	PC	SP	[SP+1]	IR	Instrucción - 1er palabra	PC	2da palabra	PC	3ra palabra	PC	Instrucción decodificada				
1	0000	FFEF	0000	19CB	0001 1001 1100 1000	0001	0004	0002			MOV R7, [0x0004]				
	Ejecución	R7=[0x0004]=0x0FE0									Flags				
2	0002			B000	1011 0000 0000 0000	0003	0005	0004			CALL 0x0005				
	Ejecución	[0xFFEF]=0x0004 SP=0xFFEE PC=0x0005									Flags				
3	0005	FFEE	0004	39A7	0011 1001 1010 0111	0006					SUB R6, R7				
	Ejecución	R6=0x0000-0x0FE0=0xF020									Flags	0	1	0	1
4	0006														
	Ejecución										Flags				
5															
	Ejecución										Flags				

## Ejercicio 2 - Seguimiento

## Ejercicio 2 - Seguimiento

Valores	PC	SP	R0	R1	R2	R3	R4	R5	R6	R7	Z	C	V	N
iniciales	0000	FFEF	0000	0000	0000	0000	0000	0000	F020	0000	0	0	0	0

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0000	19CB	0004	B000	0005	0FE0	39A7	C000	0000	0000	0000	0000	0000	0000	0000	0000	0000
0010	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

	PC	SP	[SP+1]	IR	Instrucción - 1er palabra	PC	2da palabra	PC	3ra palabra	PC	Instrucción decodificada				
1	0000	FFEF	0000	19CB	0001 1001 1100 1000	0001	0004	0002			MOV R7, [0x0004]				
	Ejecución	R7=[0x0004]=0x0FE0									Flags				
2	0002			B000	1011 0000 0000 0000	0003	0005	0004			CALL 0x0005				
	Ejecución	[0xFFEF]=0x0004 SP=0xFFEE PC=0x0005									Flags				
3	0005	FFEE	0004	39A7	0011 1001 1010 0111	0006					SUB R6, R7				
	Ejecución	R6=0x0000-0x0FE0=0xF020									Flags	0	1	0	1
4	0006			C000											
	Ejecución										Flags				
5															
	Ejecución										Flags				

## Ejercicio 2 - Seguimiento

## Ejercicio 2 - Seguimiento

Valores	PC	SP	R0	R1	R2	R3	R4	R5	R6	R7	Z	C	V	N
iniciales	0000	FFEF	0000	0000	0000	0000	0000	0000	F020	0000	0	0	0	0

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0000	19CB	0004	B000	0005	0FE0	39A7	C000	0000	0000	0000	0000	0000	0000	0000	0000	0000
0010	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

	PC	SP	[SP+1]	IR	Instrucción - 1er palabra	PC	2da palabra	PC	3ra palabra	PC	Instrucción decodificada				
1	0000	FFEF	0000	19CB	0001 1001 1100 1000	0001	0004	0002			MOV R7, [0x0004]				
	Ejecución	R7=[0x0004]=0x0FE0									Flags				
2	0002			B000	1011 0000 0000 0000	0003	0005	0004			CALL 0x0005				
	Ejecución	[0xFFEF]=0x0004 SP=0xFFEE PC=0x0005									Flags				
3	0005	FFEE	0004	39A7	0011 1001 1010 0111	0006					SUB R6, R7				
	Ejecución	R6=0x0000-0x0FE0=0xF020									Flags	0	1	0	1
4	0006			C000	1100 0000 0000 0000										
	Ejecución										Flags				
5															
	Ejecución										Flags				

## Ejercicio 2 - Seguimiento

## Ejercicio 2 - Seguimiento

Valores	PC	SP	R0	R1	R2	R3	R4	R5	R6	R7	Z	C	V	N
iniciales	0000	FFEF	0000	0000	0000	0000	0000	0000	F020	0000	0	0	0	0

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0000	19CB	0004	B000	0005	0FE0	39A7	C000	0000	0000	0000	0000	0000	0000	0000	0000	0000
0010	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

	PC	SP	[SP+1]	IR	Instrucción - 1er palabra	PC	2da palabra	PC	3ra palabra	PC	Instrucción decodificada				
1	0000	FFEF	0000	19CB	0001 1001 1100 1000	0001	0004	0002			MOV R7, [0x0004]				
	Ejecución	R7=[0x0004]=0x0FE0									Flags				
2	0002			B000	1011 0000 0000 0000	0003	0005	0004			CALL 0x0005				
	Ejecución	[0xFFEF]=0x0004 SP=0xFFEE PC=0x0005									Flags				
3	0005	FFEE	0004	39A7	0011 1001 1010 0111	0006					SUB R6, R7				
	Ejecución	R6=0x0000-0x0FE0=0xF020									Flags	0	1	0	1
4	0006			C000	1100 0000 0000 0000	0007									
	Ejecución										Flags				
5															
	Ejecución										Flags				

## Ejercicio 2 - Seguimiento

## Ejercicio 2 - Seguimiento

Valores	PC	SP	R0	R1	R2	R3	R4	R5	R6	R7	Z	C	V	N
iniciales	0000	FFEF	0000	0000	0000	0000	0000	0000	F020	0000	0	0	0	0

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0000	19CB	0004	B000	0005	0FE0	39A7	C000	0000	0000	0000	0000	0000	0000	0000	0000	0000
0010	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

	PC	SP	[SP+1]	IR	Instrucción - 1er palabra	PC	2da palabra	PC	3ra palabra	PC	Instrucción decodificada				
1	0000	FFEF	0000	19CB	0001 1001 1100 1000	0001	0004	0002			MOV R7, [0x0004]				
	Ejecución	R7=[0x0004]=0x0FE0									Flags				
2	0002			B000	1011 0000 0000 0000	0003	0005	0004			CALL 0x0005				
	Ejecución	[0xFFEF]=0x0004 SP=0xFFEE PC=0x0005									Flags				
3	0005	FFEE	0004	39A7	0011 1001 1010 0111	0006					SUB R6, R7				
	Ejecución	R6=0x0000-0x0FE0=0xF020									Flags	0	1	0	1
4	0006			C000	1100 0000 0000 0000	0007					RET				
	Ejecución										Flags				
5															
	Ejecución										Flags				

## Ejercicio 2 - Seguimiento

## Ejercicio 2 - Seguimiento

Valores	PC	SP	R0	R1	R2	R3	R4	R5	R6	R7	Z	C	V	N
iniciales	0000	FFEF	0000	0000	0000	0000	0000	0000	F020	0000	0	0	0	0

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0000	19CB	0004	B000	0005	0FE0	39A7	C000	0000	0000	0000	0000	0000	0000	0000	0000	0000
0010	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

	PC	SP	[SP+1]	IR	Instrucción - 1er palabra	PC	2da palabra	PC	3ra palabra	PC	Instrucción decodificada				
1	0000	FFEF	0000	19CB	0001 1001 1100 1000	0001	0004	0002			MOV R7, [0x0004]				
	Ejecución	R7=[0x0004]=0x0FE0									Flags				
2	0002			B000	1011 0000 0000 0000	0003	0005	0004			CALL 0x0005				
	Ejecución	[0xFFEF]=0x0004 SP=0xFFEE PC=0x0005									Flags				
3	0005	FFEE	0004	39A7	0011 1001 1010 0111	0006					SUB R6, R7				
	Ejecución	R6=0x0000-0x0FE0=0xF020									Flags	0	1	0	1
4	0006			C000	1100 0000 0000 0000	0007					RET				
	Ejecución	PC=[0xFFEF]=0x0004 SP=0xFFEF									Flags				
5															
	Ejecución										Flags				



## Ejercicio 2 - Seguimiento

## Ejercicio 2 - Seguimiento

Valores	PC	SP	R0	R1	R2	R3	R4	R5	R6	R7	Z	C	V	N
iniciales	0000	FFEF	0000	0000	0000	0000	0000	0000	F020	0000	0	0	0	0

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0000	19CB	0004	B000	0005	0FE0	39A7	C000	0000	0000	0000	0000	0000	0000	0000	0000	0000
0010	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

	PC	SP	[SP+1]	IR	Instrucción - 1er palabra	PC	2da palabra	PC	3ra palabra	PC	Instrucción decodificada				
1	0000	FFEF	0000	19CB	0001 1001 1100 1000	0001	0004	0002			MOV R7, [0x0004]				
	Ejecución	R7=[0x0004]=0x0FE0									Flags				
2	0002			B000	1011 0000 0000 0000	0003	0005	0004			CALL 0x0005				
	Ejecución	[0xFFEF]=0x0004 SP=0xFFEE PC=0x0005									Flags				
3	0005	FFEE	0004	39A7	0011 1001 1010 0111	0006					SUB R6, R7				
	Ejecución	R6=0x0000-0x0FE0=0xF020									Flags	0	1	0	1
4	0006			C000	1100 0000 0000 0000	0007					RET				
	Ejecución	PC=[0xFFEF]=0x0004 SP=0xFFEF									Flags				
5	0004														
	Ejecución										Flags				

## Ejercicio 2 - Seguimiento

## Ejercicio 2 - Seguimiento

Valores	PC	SP	R0	R1	R2	R3	R4	R5	R6	R7	Z	C	V	N
iniciales	0000	FFEF	0000	0000	0000	0000	0000	0000	F020	0000	0	0	0	0

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0000	19CB	0004	B000	0005	0FE0	39A7	C000	0000	0000	0000	0000	0000	0000	0000	0000	0000
0010	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

	PC	SP	[SP+1]	IR	Instrucción - 1er palabra	PC	2da palabra	PC	3ra palabra	PC	Instrucción decodificada				
1	0000	FFEF	0000	19CB	0001 1001 1100 1000	0001	0004	0002			MOV R7, [0x0004]				
	Ejecución	R7=[0x0004]=0x0FE0									Flags				
2	0002			B000	1011 0000 0000 0000	0003	0005	0004			CALL 0x0005				
	Ejecución	[0xFFEF]=0x0004 SP=0xFFEE PC=0x0005									Flags				
3	0005	FFEE	0004	39A7	0011 1001 1010 0111	0006					SUB R6, R7				
	Ejecución	R6=0x0000-0x0FE0=0xF020									Flags	0	1	0	1
4	0006			C000	1100 0000 0000 0000	0007					RET				
	Ejecución	PC=[0xFFEF]=0x0004 SP=0xFFEF									Flags				
5	0004	FFEF	0000												
	Ejecución										Flags				

## Ejercicio 2 - Seguimiento

## Ejercicio 2 - Seguimiento

Valores	PC	SP	R0	R1	R2	R3	R4	R5	R6	R7	Z	C	V	N
iniciales	0000	FFEF	0000	0000	0000	0000	0000	0000	F020	0000	0	0	0	0

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0000	19CB	0004	B000	0005	0FE0	39A7	C000	0000	0000	0000	0000	0000	0000	0000	0000	0000
0010	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

	PC	SP	[SP+1]	IR	Instrucción - 1er palabra	PC	2da palabra	PC	3ra palabra	PC	Instrucción decodificada				
1	0000	FFEF	0000	19CB	0001 1001 1100 1000	0001	0004	0002			MOV R7, [0x0004]				
	Ejecución	R7=[0x0004]=0x0FE0									Flags				
2	0002			B000	1011 0000 0000 0000	0003	0005	0004			CALL 0x0005				
	Ejecución	[0xFFEF]=0x0004 SP=0xFFEE PC=0x0005									Flags				
3	0005	FFEE	0004	39A7	0011 1001 1010 0111	0006					SUB R6, R7				
	Ejecución	R6=0x0000-0x0FE0=0xF020									Flags	0	1	0	1
4	0006			C000	1100 0000 0000 0000	0007					RET				
	Ejecución	PC=[0xFFEF]=0x0004 SP=0xFFEF									Flags				
5	0004	FFEF	0000	0FE0											
	Ejecución										Flags				

## Ejercicio 2 - Seguimiento

## Ejercicio 2 - Seguimiento

Valores	PC	SP	R0	R1	R2	R3	R4	R5	R6	R7	Z	C	V	N
iniciales	0000	FFEF	0000	0000	0000	0000	0000	0000	F020	0000	0	0	0	0

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0000	19CB	0004	B000	0005	0FE0	39A7	C000	0000	0000	0000	0000	0000	0000	0000	0000	0000
0010	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

	PC	SP	[SP+1]	IR	Instrucción - 1er palabra	PC	2da palabra	PC	3ra palabra	PC	Instrucción decodificada				
1	0000	FFEF	0000	19CB	0001 1001 1100 1000	0001	0004	0002			MOV R7, [0x0004]				
	Ejecución	R7=[0x0004]=0x0FE0									Flags				
2	0002			B000	1011 0000 0000 0000	0003	0005	0004			CALL 0x0005				
	Ejecución	[0xFFEF]=0x0004 SP=0xFFEE PC=0x0005									Flags				
3	0005	FFEE	0004	39A7	0011 1001 1010 0111	0006					SUB R6, R7				
	Ejecución	R6=0x0000-0x0FE0=0xF020									Flags	0	1	0	1
4	0006			C000	1100 0000 0000 0000	0007					RET				
	Ejecución	PC=[0xFFEF]=0x0004 SP=0xFFEF									Flags				
5	0004	FFEF	0000	0FE0	0000 1111 1110 0000										
	Ejecución										Flags				

## Ejercicio 2 - Seguimiento

## Ejercicio 2 - Seguimiento

Valores	PC	SP	R0	R1	R2	R3	R4	R5	R6	R7	Z	C	V	N
iniciales	0000	FFEF	0000	0000	0000	0000	0000	0000	F020	0000	0	0	0	0

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0000	19CB	0004	B000	0005	0FE0	39A7	C000	0000	0000	0000	0000	0000	0000	0000	0000	0000
0010	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

	PC	SP	[SP+1]	IR	Instrucción - 1er palabra	PC	2da palabra	PC	3ra palabra	PC	Instrucción decodificada				
1	0000	FFEF	0000	19CB	0001 1001 1100 1000	0001	0004	0002			MOV R7, [0x0004]				
	Ejecución	R7=[0x0004]=0x0FE0									Flags				
2	0002			B000	1011 0000 0000 0000	0003	0005	0004			CALL 0x0005				
	Ejecución	[0xFFEF]=0x0004 SP=0xFFEE PC=0x0005									Flags				
3	0005	FFEE	0004	39A7	0011 1001 1010 0111	0006					SUB R6, R7				
	Ejecución	R6=0x0000-0x0FE0=0xF020									Flags	0	1	0	1
4	0006			C000	1100 0000 0000 0000	0007					RET				
	Ejecución	PC=[0xFFEF]=0x0004 SP=0xFFEF									Flags				
5	0004	FFEF	0000	0FE0	0000 1111 1110 0000	0005									
	Ejecución										Flags				

## Ejercicio 2 - Seguimiento

## Ejercicio 2 - Seguimiento

Valores	PC	SP	R0	R1	R2	R3	R4	R5	R6	R7	Z	C	V	N
iniciales	0000	FFEF	0000	0000	0000	0000	0000	0000	F020	0000	0	0	0	0

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0000	19CB	0004	B000	0005	0FE0	39A7	C000	0000	0000	0000	0000	0000	0000	0000	0000	0000
0010	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

	PC	SP	[SP+1]	IR	Instrucción - 1er palabra	PC	2da palabra	PC	3ra palabra	PC	Instrucción decodificada				
1	0000	FFEF	0000	19CB	0001 1001 1100 1000	0001	0004	0002			MOV R7, [0x0004]				
	Ejecución	R7=[0x0004]=0x0FE0									Flags				
2	0002			B000	1011 0000 0000 0000	0003	0005	0004			CALL 0x0005				
	Ejecución	[0xFFEF]=0x0004 SP=0xFFEE PC=0x0005									Flags				
3	0005	FFEE	0004	39A7	0011 1001 1010 0111	0006					SUB R6, R7				
	Ejecución	R6=0x0000-0x0FE0=0xF020									Flags	0	1	0	1
4	0006			C000	1100 0000 0000 0000	0007					RET				
	Ejecución	PC=[0xFFEF]=0x0004 SP=0xFFEF									Flags				
5	0004	FFEF	0000	0FE0	0000 1111 1110 0000	0005					Instrucción Inválida				
	Ejecución										Flags				

## Ejercicio 2 - Seguimiento

## Ejercicio 2 - Seguimiento

Valores	PC	SP	R0	R1	R2	R3	R4	R5	R6	R7	Z	C	V	N
iniciales	0000	FFEF	0000	0000	0000	0000	0000	0000	F020	0000	0	0	0	0

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0000	19CB	0004	B000	0005	0FE0	39A7	C000	0000	0000	0000	0000	0000	0000	0000	0000	0000
0010	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

	PC	SP	[SP+1]	IR	Instrucción - 1er palabra	PC	2da palabra	PC	3ra palabra	PC	Instrucción decodificada				
1	0000	FFEF	0000	19CB	0001 1001 1100 1000	0001	0004	0002			MOV R7, [0x0004]				
	Ejecución	R7=[0x0004]=0x0FE0									Flags				
2	0002			B000	1011 0000 0000 0000	0003	0005	0004			CALL 0x0005				
	Ejecución	[0xFFEF]=0x0004 SP=0xFFEE PC=0x0005									Flags				
3	0005	FFEE	0004	39A7	0011 1001 1010 0111	0006					SUB R6, R7				
	Ejecución	R6=0x0000-0x0FE0=0xF020									Flags	0	1	0	1
4	0006			C000	1100 0000 0000 0000	0007					RET				
	Ejecución	PC=[0xFFEF]=0x0004 SP=0xFFEF									Flags				
5	0004	FFEF	0000	0FE0	0000 1111 1110 0000	0005					Instrucción Inválida				
	Ejecución	FIN DE LA EJECUCIÓN									Flags				

## Ejercicio 2 - Seguimiento

## Ejercicio 2 - Solución

Planilla de Seguimiento														F020							
Valores iniciales:		PC	SP																		
		0000	FFEF																		
Memoria:		+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F				
		0000	19C8	0004	8000	0005	0FE0	39A7	C000	0000	0000	0000	0000	0000	0000	0000	0000	0000			
		0001	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000			
	PC	SP	[SP+1]	IR	Instrucción - 1 <sup>er</sup> palabra (en bits)				PC	2 <sup>do</sup> palabra	PC	3 <sup>er</sup> a palabra	PC	Instrucción decodificada							
1	0000	FFEF	0000	19C8	0001	1001	1100	1000	0001	0004	0002			MOV R7, [0x0004]							
	Ejecución		R7 = [0x0004] = 0x0FE0														Flags <sup>1</sup>	Z	C	V	N
2	0002			B000	1011	0000	0000	0000	0003	0005	0004			CALL 0x0005							
	Ejecución		[0xFFEF] = 0x0004    SP = 0xFFEE    PC = 0x0005														Flags <sup>1</sup>	Z	C	V	N
3	0005	FFEE	0004	39A7	0011	1001	1010	0111	0006					SUB R6, R7							
	Ejecución		R6 = 0x0000 - 0x0FE0 = 0xF020														Flags <sup>1</sup>	Z	C	V	N
4	0006			C000	1100	0000	0000	0000	0007					RET							
	Ejecución		PC = [0xFFEF] = 0x0004    SP = 0xFFEF														Flags <sup>1</sup>	Z	C	V	N
5	0004	FFEF	0000	0FE0	0000	1111	1110	0000	0005					Instrucción Inválida							
	Ejecución		FIN DE LA EJECUCIÓN.														Flags <sup>1</sup>	Z	C	V	N



## Ejercicio 2 - Observaciones

**¿Qué pasó con el programa anterior?**

## Ejercicio 2 - Observaciones

### ¿Qué pasó con el programa anterior?

- Utilizamos la instrucción CALL.
- El valor del PC de retorno se almacena en la pila durante de la ejecución de la instrucción.
- Recuerden que la pila crece en el sentido de las direcciones de memoria menores.
- Al regresar con RET, se “saca” la dirección de retorno de allí.

# Ejercicio 3

Codifiquemos el siguiente código y carguémoslo desde la posición de memoria 0x0000:

```
main:  MOV R1, 0x0000
loop:  CMP R1, 0x0001
       JL sumoUno
       JGE fin
sumoUno: ADD R1, 0x0001
       JNE loop
fin:    RET
```

# Ejercicio 3

Codifiquemos el siguiente código y carguémoslo desde la posición de memoria 0x0000:

```
main:  MOV R1, 0x0000
      loop:  CMP R1, 0x0001
              JL sumoUno
              JGE fin
sumoUno:  ADD R1, 0x0001
              JNE loop
      fin:  RET
```

dirección	ocupa
0x0000	

# Ejercicio 3

Codifiquemos el siguiente código y carguémoslo desde la posición de memoria 0x0000:

```
main:  MOV R1, 0x0000
      loop:  CMP R1, 0x0001
              JL sumoUno
              JGE fin
sumoUno:  ADD R1, 0x0001
              JNE loop
      fin:  RET
```

dirección	ocupa
0x0000	dos palabras

# Ejercicio 3

Codifiquemos el siguiente código y carguémoslo desde la posición de memoria 0x0000:

```
main:  MOV R1, 0x0000
```

```
loop:  CMP R1, 0x0001
```

```
        JL sumoUno
```

```
        JGE fin
```

```
sumoUno:  ADD R1, 0x0001
```

```
        JNE loop
```

```
fin:     RET
```

dirección	ocupa
0x0000	dos palabras
0x0002	

# Ejercicio 3

Codifiquemos el siguiente código y carguémoslo desde la posición de memoria 0x0000:

```
main:  MOV R1, 0x0000
```

```
loop:  CMP R1, 0x0001
```

```
      JL sumoUno
```

```
      JGE fin
```

```
sumoUno:  ADD R1, 0x0001
```

```
      JNE loop
```

```
fin:    RET
```

dirección	ocupa
0x0000	dos palabras
0x0002	dos palabras

# Ejercicio 3

Codifiquemos el siguiente código y carguémoslo desde la posición de memoria 0x0000:

```
main:  MOV R1, 0x0000
loop:  CMP R1, 0x0001
        JL sumoUno
        JGE fin
sumoUno: ADD R1, 0x0001
        JNE loop
fin:    RET
```

dirección	ocupa
0x0000	dos palabras
0x0002	dos palabras
0x0004	



# Ejercicio 3

Codifiquemos el siguiente código y carguémoslo desde la posición de memoria 0x0000:

```
main:  MOV R1, 0x0000
```

```
loop:  CMP R1, 0x0001
```

```
      JL sumoUno
```

```
      JGE fin
```

```
sumoUno: ADD R1, 0x0001
```

```
      JNE loop
```

```
fin:   RET
```

dirección	ocupa
0x0000	dos palabras
0x0002	dos palabras
0x0004	una palabra

# Ejercicio 3

Codifiquemos el siguiente código y carguémoslo desde la posición de memoria 0x0000:

```
main:  MOV R1, 0x0000
loop:  CMP R1, 0x0001
        JL sumoUno
        JGE fin
sumoUno: ADD R1, 0x0001
        JNE loop
fin:    RET
```

dirección	ocupa
0x0000	dos palabras
0x0002	dos palabras
0x0004	una palabra
0x0005	

# Ejercicio 3

Codifiquemos el siguiente código y carguémoslo desde la posición de memoria 0x0000:

```
main:  MOV R1, 0x0000
loop:  CMP R1, 0x0001
        JL sumoUno
        JGE fin
sumoUno: ADD R1, 0x0001
        JNE loop
fin:    RET
```

dirección	ocupa
0x0000	dos palabras
0x0002	dos palabras
0x0004	una palabra
0x0005	una palabra

# Ejercicio 3

Codifiquemos el siguiente código y carguémoslo desde la posición de memoria 0x0000:

```
main:  MOV R1, 0x0000
loop:  CMP R1, 0x0001
        JL sumoUno
        JGE fin
sumoUno: ADD R1, 0x0001
        JNE loop
fin:    RET
```

dirección	ocupa
0x0000	dos palabras
0x0002	dos palabras
0x0004	una palabra
0x0005	una palabra
0x0006	

# Ejercicio 3

Codifiquemos el siguiente código y carguémoslo desde la posición de memoria 0x0000:

```
main:  MOV R1, 0x0000
loop:  CMP R1, 0x0001
        JL sumoUno
        JGE fin
sumoUno: ADD R1, 0x0001
        JNE loop
fin:    RET
```

dirección	ocupa
0x0000	dos palabras
0x0002	dos palabras
0x0004	una palabra
0x0005	una palabra
0x0006	dos palabras

# Ejercicio 3

Codifiquemos el siguiente código y carguémoslo desde la posición de memoria 0x0000:

```
main:  MOV R1, 0x0000
loop:  CMP R1, 0x0001
        JL sumoUno
        JGE fin
sumoUno: ADD R1, 0x0001
        JNE loop
fin:    RET
```

dirección	ocupa
0x0000	dos palabras
0x0002	dos palabras
0x0004	una palabra
0x0005	una palabra
0x0006	dos palabras
0x0008	

# Ejercicio 3

Codifiquemos el siguiente código y carguémoslo desde la posición de memoria 0x0000:

```
main:  MOV R1, 0x0000
loop:  CMP R1, 0x0001
        JL sumoUno
        JGE fin
sumoUno: ADD R1, 0x0001
        JNE loop
fin:    RET
```

dirección	ocupa
0x0000	dos palabras
0x0002	dos palabras
0x0004	una palabra
0x0005	una palabra
0x0006	dos palabras
0x0008	una palabra

# Ejercicio 3

Codifiquemos el siguiente código y carguémoslo desde la posición de memoria 0x0000:

	dirección	ocupa
main: MOV R1, 0x0000	0x0000	dos palabras
loop: CMP R1, 0x0001	0x0002	dos palabras
JL sumoUno	0x0004	una palabra
JGE fin	0x0005	una palabra
sumoUno: ADD R1, 0x0001	0x0006	dos palabras
JNE loop	0x0008	una palabra
fin: RET	0x0009	



# Ejercicio 3

Codifiquemos el siguiente código y carguémoslo desde la posición de memoria 0x0000:

	dirección	ocupa
main: MOV R1, 0x0000	0x0000	dos palabras
loop: CMP R1, 0x0001	0x0002	dos palabras
JL sumoUno	0x0004	una palabra
JGE fin	0x0005	una palabra
sumoUno: ADD R1, 0x0001	0x0006	dos palabras
JNE loop	0x0008	una palabra
fin: RET	0x0009	una palabra

# Ejercicio 3

Codifiquemos el siguiente código y carguémoslo desde la posición de memoria 0x0000:

**main:** MOV R1, 0x0000

**loop:** CMP R1, 0x0001

JL sumoUno

JGE fin

**sumoUno:** ADD R1, 0x0001

JNE loop

**fin:** RET

dirección	ocupa
0x0000	dos palabras
0x0002	dos palabras
0x0004	una palabra
0x0005	una palabra
0x0006	dos palabras
0x0008	una palabra
0x0009	una palabra

La etiqueta **main** corresponde a la dirección de memoria **0x0000**.

La etiqueta **loop** corresponde a la dirección de memoria **0x0002**.

La etiqueta **sumoUno** corresponde a la dirección de memoria **0x0006**.

Buenísimo, tenemos las direcciones de las etiquetas, pero... *los saltos condicionales son relativos.*

Buenísimo, tenemos las direcciones de las etiquetas, pero... *los saltos condicionales son relativos*.

- Esto significa que las etiquetas de los saltos son reemplazadas por el desplazamiento necesario para “llegar” desde la dirección en la que estamos parados, hasta la de la etiqueta de destino.

Buenísimo, tenemos las direcciones de las etiquetas, pero... *los saltos condicionales son relativos*.

- Esto significa que las etiquetas de los saltos son reemplazadas por el desplazamiento necesario para “llegar” desde la dirección en la que estamos parados, hasta la de la etiqueta de destino.
- Vamos a tener que calcular estos desplazamientos.

0x0000: MOV R1, 0x0000

0x0002: CMP R1, 0x0001

0x0004: JL sumoUno

0x0005: JGE fin

0x0006: ADD R1, 0x0001

0x0008: JNE loop

0x0009: RET

$$\text{Despl} = \text{Dir\_Etiqueta} - \text{Dir\_Instr}_{\text{postfetch}}$$

La etiqueta **loop** corresponde a la dirección de memoria **0x0002**.

La etiqueta **sumoUno** corresponde a la dirección de memoria **0x0006**.

La etiqueta **fin** corresponde a la dirección de memoria **0x0009**.

```
0x0000:  MOV R1, 0x0000
0x0002:  CMP R1, 0x0001
0x0004:  JL  sumoUno
0x0005:  JGE fin
0x0006:  ADD R1, 0x0001
0x0008:  JNE loop
0x0009:  RET
```

$\text{Despl} = \text{Dir\_Etiqueta} - \text{Dir\_Instr}_{\text{postfetch}}$

desplazamiento **sumoUno**:

La etiqueta **loop** corresponde a la dirección de memoria **0x0002**.

La etiqueta **sumoUno** corresponde a la dirección de memoria **0x0006**.

La etiqueta **fin** corresponde a la dirección de memoria **0x0009**.

0x0000: MOV R1, 0x0000

0x0002: CMP R1, 0x0001

0x0004: JL *sumoUno*

0x0005: JGE fin

0x0006: ADD R1, 0x0001

0x0008: JNE loop

0x0009: RET

$\text{Despl} = \text{Dir\_Etiqueta} - \text{Dir\_Instr}_{\text{postfetch}}$

desplazamiento *sumoUno*:

La etiqueta *loop* corresponde a la dirección de memoria 0x0002.

La etiqueta *sumoUno* corresponde a la dirección de memoria 0x0006.

La etiqueta *fin* corresponde a la dirección de memoria 0x0009.



0x0000: MOV R1, 0x0000

0x0002: CMP R1, 0x0001

0x0004: JL *sumoUno*

0x0005: JGE fin

0x0006: ADD R1, 0x0001

0x0008: JNE loop

0x0009: RET

$\text{Despl} = \text{Dir\_Etiqueta} - \text{Dir\_Instr}_{\text{postfetch}}$

desplazamiento *sumoUno*:

0x0006 - 0x0005

La etiqueta *loop* corresponde a la dirección de memoria 0x0002.

La etiqueta *sumoUno* corresponde a la dirección de memoria 0x0006.

La etiqueta *fin* corresponde a la dirección de memoria 0x0009.

0x0000: MOV R1, 0x0000

0x0002: CMP R1, 0x0001

0x0004: JL *sumoUno*

0x0005: JGE fin

0x0006: ADD R1, 0x0001

0x0008: JNE loop

0x0009: RET

$\text{Despl} = \text{Dir\_Etiqueta} - \text{Dir\_Instr}_{\text{postfetch}}$

desplazamiento *sumoUno*:

$0x0006 - 0x0005 = 1$

La etiqueta *loop* corresponde a la dirección de memoria 0x0002.

La etiqueta *sumoUno* corresponde a la dirección de memoria 0x0006.

La etiqueta *fin* corresponde a la dirección de memoria 0x0009.

0x0000: MOV R1, 0x0000

0x0002: CMP R1, 0x0001

0x0004: JL *sumoUno*

0x0005: JGE fin

0x0006: ADD R1, 0x0001

0x0008: JNE loop

0x0009: RET

$\text{Despl} = \text{Dir\_Etiqueta} - \text{Dir\_Instr}_{\text{postfetch}}$

desplazamiento *sumoUno*:

$0x0006 - 0x0005 = 1 = 0x01$

La etiqueta *loop* corresponde a la dirección de memoria 0x0002.

La etiqueta *sumoUno* corresponde a la dirección de memoria 0x0006.

La etiqueta *fin* corresponde a la dirección de memoria 0x0009.

```
0x0000:  MOV R1, 0x0000
0x0002:  CMP R1, 0x0001
0x0004:  JL  sumoUno
0x0005:  JGE fin
0x0006:  ADD R1, 0x0001
0x0008:  JNE loop
0x0009:  RET
```

$\text{Despl} = \text{Dir\_Etiqueta} - \text{Dir\_Instr}_{\text{postfetch}}$

desplazamiento sumoUno:

$0x0006 - 0x0005 = 1 = 0x01$

desplazamiento fin:

La etiqueta **loop** corresponde a la dirección de memoria **0x0002**.

La etiqueta **sumoUno** corresponde a la dirección de memoria **0x0006**.

La etiqueta **fin** corresponde a la dirección de memoria **0x0009**.

```
0x0000:  MOV R1, 0x0000
0x0002:  CMP R1, 0x0001
0x0004:  JL  sumoUno
0x0005:  JGE fin
0x0006:  ADD R1, 0x0001
0x0008:  JNE loop
0x0009:  RET
```

Despl =  $\text{Dir\_Etiqueta} - \text{Dir\_Instr}_{\text{postfetch}}$

desplazamiento sumoUno:

$0x0006 - 0x0005 = 1 = 0x01$

desplazamiento fin:

La etiqueta **loop** corresponde a la dirección de memoria **0x0002**.

La etiqueta **sumoUno** corresponde a la dirección de memoria **0x0006**.

La etiqueta **fin** corresponde a la dirección de memoria **0x0009**.

```
0x0000:  MOV R1, 0x0000
0x0002:  CMP R1, 0x0001
0x0004:  JL sumoUno
0x0005:  JGE fin
0x0006:  ADD R1, 0x0001
0x0008:  JNE loop
0x0009:  RET
```

Despl =  $\text{Dir\_Etiqueta} - \text{Dir\_Instr}_{\text{postfetch}}$

desplazamiento sumoUno:

$0x0006 - 0x0005 = 1 = 0x01$

desplazamiento fin:

$0x0009 - 0x0006$

La etiqueta **loop** corresponde a la dirección de memoria **0x0002**.

La etiqueta **sumoUno** corresponde a la dirección de memoria **0x0006**.

La etiqueta **fin** corresponde a la dirección de memoria **0x0009**.

```
0x0000:  MOV R1, 0x0000
0x0002:  CMP R1, 0x0001
0x0004:  JL sumoUno
0x0005:  JGE fin
0x0006:  ADD R1, 0x0001
0x0008:  JNE loop
0x0009:  RET
```

Despl =  $\text{Dir\_Etiqueta} - \text{Dir\_Instr}_{\text{postfetch}}$

desplazamiento sumoUno:

$$0x0006 - 0x0005 = 1 = 0x01$$

desplazamiento fin:

$$0x0009 - 0x0006 = 3$$

La etiqueta `loop` corresponde a la dirección de memoria `0x0002`.

La etiqueta `sumoUno` corresponde a la dirección de memoria `0x0006`.

La etiqueta `fin` corresponde a la dirección de memoria `0x0009`.

```
0x0000:  MOV R1, 0x0000
0x0002:  CMP R1, 0x0001
0x0004:  JL sumoUno
0x0005:  JGE fin
0x0006:  ADD R1, 0x0001
0x0008:  JNE loop
0x0009:  RET
```

$\text{Despl} = \text{Dir\_Etiqueta} - \text{Dir\_Instr}_{\text{postfetch}}$

desplazamiento sumoUno:

$$0x0006 - 0x0005 = 1 = 0x01$$

desplazamiento fin:

$$0x0009 - 0x0006 = 3 = 0x03$$

La etiqueta **loop** corresponde a la dirección de memoria **0x0002**.

La etiqueta **sumoUno** corresponde a la dirección de memoria **0x0006**.

La etiqueta **fin** corresponde a la dirección de memoria **0x0009**.



```
0x0000:  MOV R1, 0x0000
0x0002:  CMP R1, 0x0001
0x0004:  JL sumoUno
0x0005:  JGE fin
0x0006:  ADD R1, 0x0001
0x0008:  JNE loop
0x0009:  RET
```

Despl =  $\text{Dir\_Etiqueta} - \text{Dir\_Instr}_{\text{postfetch}}$

desplazamiento sumoUno:

$$0x0006 - 0x0005 = 1 = 0x01$$

desplazamiento fin:

$$0x0009 - 0x0006 = 3 = 0x03$$

desplazamiento loop:

La etiqueta **loop** corresponde a la dirección de memoria **0x0002**.

La etiqueta **sumoUno** corresponde a la dirección de memoria **0x0006**.

La etiqueta **fin** corresponde a la dirección de memoria **0x0009**.

```
0x0000:  MOV R1, 0x0000
0x0002:  CMP R1, 0x0001
0x0004:  JL  sumoUno
0x0005:  JGE fin
0x0006:  ADD R1, 0x0001
0x0008:  JNE loop
0x0009:  RET
```

Despl =  $\text{Dir\_Etiqueta} - \text{Dir\_Instr}_{\text{postfetch}}$

desplazamiento sumoUno:

$$0x0006 - 0x0005 = 1 = 0x01$$

desplazamiento fin:

$$0x0009 - 0x0006 = 3 = 0x03$$

desplazamiento loop:

La etiqueta **loop** corresponde a la dirección de memoria **0x0002**.

La etiqueta **sumoUno** corresponde a la dirección de memoria **0x0006**.

La etiqueta **fin** corresponde a la dirección de memoria **0x0009**.

```
0x0000: MOV R1, 0x0000
0x0002: CMP R1, 0x0001
0x0004: JL sumoUno
0x0005: JGE fin
0x0006: ADD R1, 0x0001
0x0008: JNE loop
0x0009: RET
```

Despl =  $\text{Dir\_Etiqueta} - \text{Dir\_Instr}_{\text{postfetch}}$

desplazamiento sumoUno:

$$0x0006 - 0x0005 = 1 = 0x01$$

desplazamiento fin:

$$0x0009 - 0x0006 = 3 = 0x03$$

desplazamiento loop:

$$0x0002 - 0x0009$$

La etiqueta **loop** corresponde a la dirección de memoria **0x0002**.

La etiqueta **sumoUno** corresponde a la dirección de memoria **0x0006**.

La etiqueta **fin** corresponde a la dirección de memoria **0x0009**.

```
0x0000: MOV R1, 0x0000
0x0002: CMP R1, 0x0001
0x0004: JL sumoUno
0x0005: JGE fin
0x0006: ADD R1, 0x0001
0x0008: JNE loop
0x0009: RET
```

Despl =  $\text{Dir\_Etiqueta} - \text{Dir\_Instr}_{\text{postfetch}}$

desplazamiento sumoUno:

$$0x0006 - 0x0005 = 1 = 0x01$$

desplazamiento fin:

$$0x0009 - 0x0006 = 3 = 0x03$$

desplazamiento loop:

$$0x0002 - 0x0009 = -7$$

La etiqueta `loop` corresponde a la dirección de memoria `0x0002`.

La etiqueta `sumoUno` corresponde a la dirección de memoria `0x0006`.

La etiqueta `fin` corresponde a la dirección de memoria `0x0009`.

```
0x0000: MOV R1, 0x0000
0x0002: CMP R1, 0x0001
0x0004: JL sumoUno
0x0005: JGE fin
0x0006: ADD R1, 0x0001
0x0008: JNE loop
0x0009: RET
```

Despl =  $\text{Dir\_Etiqueta} - \text{Dir\_Instr}_{\text{postfetch}}$

desplazamiento sumoUno:

$$0x0006 - 0x0005 = 1 = 0x01$$

desplazamiento fin:

$$0x0009 - 0x0006 = 3 = 0x03$$

desplazamiento loop:

$$0x0002 - 0x0009 = -7 = 0xF9$$

La etiqueta **loop** corresponde a la dirección de memoria **0x0002**.

La etiqueta **sumoUno** corresponde a la dirección de memoria **0x0006**.

La etiqueta **fin** corresponde a la dirección de memoria **0x0009**.

```
0x0000: MOV R1, 0x0000
0x0002: CMP R1, 0x0001
0x0004: JL sumoUno
0x0005: JGE fin
0x0006: ADD R1, 0x0001
0x0008: JNE loop
0x0009: RET
```

Despl =  $\text{Dir\_Etiqueta} - \text{Dir\_Instr}_{\text{postfetch}}$

desplazamiento sumoUno:

$$0x0006 - 0x0005 = 1 = 0x01$$

desplazamiento fin:

$$0x0009 - 0x0006 = 3 = 0x03$$

desplazamiento loop:

$$0x0002 - 0x0009 = -7 = 0xF9$$

La etiqueta **loop** corresponde a la dirección de memoria **0x0002**.

La etiqueta **sumoUno** corresponde a la dirección de memoria **0x0006**.

La etiqueta **fin** corresponde a la dirección de memoria **0x0009**.

0x0000: MOV R1, 0x0000

0x0002: CMP R1, 0x0001

0x0004: JL 0x01

0x0005: JGE 0x03

0x0006: ADD R1, 0x0001

0x0008: JNE 0xF9

0x0009: RET

Ahora sí: nuestro programa  
esta listo para ser ensamblado!

Tarea: ensamblado y segui-  
miento!

## Ejercicio 3 - Observaciones

Moraleja del ejercicio anterior:



## Ejercicio 3 - Observaciones

Moraleja del ejercicio anterior:

- Los desplazamientos de los saltos condicionales *siempre son relativos*.

## Ejercicio 3 - Observaciones

Moraleja del ejercicio anterior:

- Los desplazamientos de los saltos condicionales *siempre son relativos*.
- Siempre van a ser traducidas las etiquetas primero, y luego calculados los desplazamientos.

## Ejercicio 3 - Observaciones

Moraleja del ejercicio anterior:

- Los desplazamientos de los saltos condicionales *siempre son relativos*.
- Siempre van a ser traducidas las etiquetas primero, y luego calculados los desplazamientos.
- Para calcular los desplazamientos, tengan en cuenta que el salto se ejecuta *después* de incrementar el PC.

## Ejercicio 3 - Observaciones

Moraleja del ejercicio anterior:

- Los desplazamientos de los saltos condicionales *siempre son relativos*.
- Siempre van a ser traducidas las etiquetas primero, y luego calculados los desplazamientos.
- Para calcular los desplazamientos, tengan en cuenta que el salto se ejecuta *después* de incrementar el PC.
- Los desplazamientos se representan en complemento a dos.

## Ejercicio 3 - Observaciones

Moraleja del ejercicio anterior:

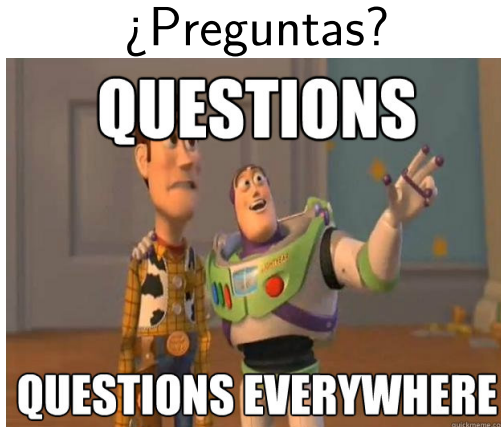
- Los desplazamientos de los saltos condicionales *siempre son relativos*.
- Siempre van a ser traducidas las etiquetas primero, y luego calculados los desplazamientos.
- Para calcular los desplazamientos, tengan en cuenta que el salto se ejecuta *después* de incrementar el PC.
- Los desplazamientos se representan en complemento a dos.
- Siempre extender los desplazamientos *con signo* a la hora de aplicarlos.

# Resumen

Hoy vimos:

- Máquina Orga 1 y su arquitectura
  - ISA: formato de instrucción, modos de direccionamiento
  - ciclo de ejecución
- Ciclo de vida de un programa
  - uso de etiquetas
  - directivas
- Ensamblamos programas
- Seguimos programas dentro de la máquina Orga1

# ¿Preguntas?



# Avisos

Martes 8/5: Taller de Ciclo de Instrucción en los labos.  
**IMPORTANTE:** Traer Cartilla Orga 1 y planillas de Seguimiento



¡Eso es todo amigos!

