

ROBIN ALFONSO BLANCO CAÑON



UNIVERSIDAD  
NACIONAL  
DE COLOMBIA

# PRESENTACIÓN DE LABORATORIO N.º 1 MICROCONTROLADORES

Maria Lucia Arias Ortiz

Juan Antonio Rojas Cobos

1005281015

1002341393

mariasor@unal.edu.co

juariojasco@unal.edu.co

FACULTAD DE INGENIERÍA  
DEPARTAMENTO DE MECÁNICA Y MECATRÓNICA

# PSEUDOCÓDIGO EJERCICIOS

## Punto 1

1. Cargar el valor 7 en el registro W
2. Almacenar el valor del registro W en la variable aux1
3. Sumar la constante 3 al valor de la variable aux1
4. Almacenar el resultado de la suma en la variable aux4

## Punto 2

1. Cargar el valor 8 en el registro W
2. Almacenar el valor del registro W en la variable aux1
3. Cargar el valor 10 en el registro W
4. Almacenar el valor del registro W en la variable aux2
5. Cargar el valor de la variable aux1 en el registro W
6. Sumar el contenido del registro W con el valor de la variable aux2
7. Almacenar el resultado de la suma en la variable aux4

## Punto 1

1. Cargar el valor 5 en el registro W
2. Almacenar el valor del registro W en la variable aux1
3. Cargar el valor de la variable aux1 en el registro W
4. Restar el valor constante 9 al valor en el registro W
5. Almacenar el resultado de la resta en la variable aux4

## Punto 4

1. Cargar el valor 6 en el registro W
2. Almacenar el valor del registro W en la variable aux1
3. Cargar el valor 4 en el registro W
4. Almacenar el valor del registro W en la variable aux2
5. Cargar el valor de la variable aux2 en el registro W
6. Restar el contenido del registro W al valor de la variable aux1
7. Almacenar el resultado de la resta en la variable aux4

# PSEUDOCÓDIGO EJERCICIOS

## Punto 5

1. Cargar el valor 5 en el registro W
2. Almacenar el valor del registro W en la variable aux1
3. Cargar el valor de la variable aux1 en el registro W
4. Multiplicar el contenido del registro W por la constante 4
5. Mover el producto al registro de bajo orden (PRODL)
6. Almacenar el contenido del registro en la variable aux4

## Punto 6

1. Cargar el valor 12 en el registro W
2. Almacenar el valor del registro W en la variable aux1
3. Cargar el valor 15 en el registro W
4. Almacenar el valor del registro W en la variable aux2
5. Cargar el valor de la variable aux1 en el registro W
6. Multiplicar el contenido del registro W por el valor de la variable aux2
7. Mover el producto al registro de bajo orden (PRODL)
8. Almacenar el contenido del registro en la variable aux4

## Punto 7

1. Cargar el valor 12 en el registro W
2. Almacenar el valor del registro W en la variable aux1
3. Realizar el complemento a 1 de la variable aux1

## Punto 8

1. Cargar el valor 12 en el registro W
2. Almacenar el valor del registro W en la variable aux1
3. Realizar complemento a 2 de la variable aux1

# PSEUDOCÓDIGO EJERCICIOS

## Punto 9

1. Cargar el valor 35 en el registro W
2. Almacenar el valor del registro W en la variable aux1
3. Cargar el valor de la variable aux1 en el registro W
4. Realizar la operación OR con la constante 7
5. Almacenar el resultado en la variable aux4

## Punto 10

1. Cargar el valor 20 en el registro W
2. Almacenar el valor del registro W en la variable aux1
3. Cargar el valor 56 en el registro W
4. Almacenar el valor del registro W en la variable aux2
5. Cargar el valor de la variable aux1 en el registro W
6. Realizar la operación OR entre el contenido de la variable aux1 y la variable aux2
7. Almacenar el resultado en la variable aux4

## Punto 11

1. Cargar el valor 62 en el registro W
2. Almacenar el valor del registro W en la variable aux1
3. Cargar el valor de la variable aux1 en el registro W
4. Realizar la operación AND con la constante 15
5. Almacenar el resultado en la variable aux4

## Punto 12

1. Cargar el valor 100 en el registro W
2. Almacenar el valor del registro W en la variable aux1
3. Cargar el valor 45 en el registro W
4. Almacenar el valor del registro W en la variable aux2
5. Cargar el valor de la variable aux1 en el registro W
6. Realizar la operación AND entre el contenido de la variable aux1 y la variable aux2
7. Almacenar el resultado en la variable aux4

# PSEUDOCÓDIGO EJERCICIOS

## Punto 13

1. Cargar el valor 120 en el registro W
2. Almacenar el valor del registro W en la variable aux1
3. Cargar el valor de la variable aux1 en el registro W
4. Realizar la operación XOR con la constante 1
5. Almacenar el resultado en la variable aux4

## Punto 14

1. Cargar el valor 17 en el registro W
2. Almacenar el valor del registro W en la variable aux1
3. Cargar el valor 90 en el registro W
4. Almacenar el valor del registro W en la variable aux2
5. Cargar el valor de la variable aux1 en el registro W
6. Realizar la operación XOR entre el contenido de la variable aux1 y la variable aux2
7. Almacenar el resultado en la variable aux4

## Punto 15

1. Cargar el valor 25 en el registro W
2. Almacenar el valor del registro W en la variable aux1
3. Cargar el valor 40 en el registro W
4. Almacenar el valor del registro W en la variable aux2
5. Cargar el valor 103 en el registro W
6. Almacenar el valor del registro W en la variable aux3
7. Cargar el valor de la variable aux3 en el registro W
8. Realizar la operación XOR con la constante 0xD0
9. Almacenar el resultado en la variable aux4
10. Cargar el valor de la variable aux1 en el registro W
11. Realizar la operación OR con el contenido de la variable aux2
12. Almacenar el resultado en el registro W
13. Realizar la operación AND con el contenido de la variable aux4
14. Almacenar el resultado en la variable aux4

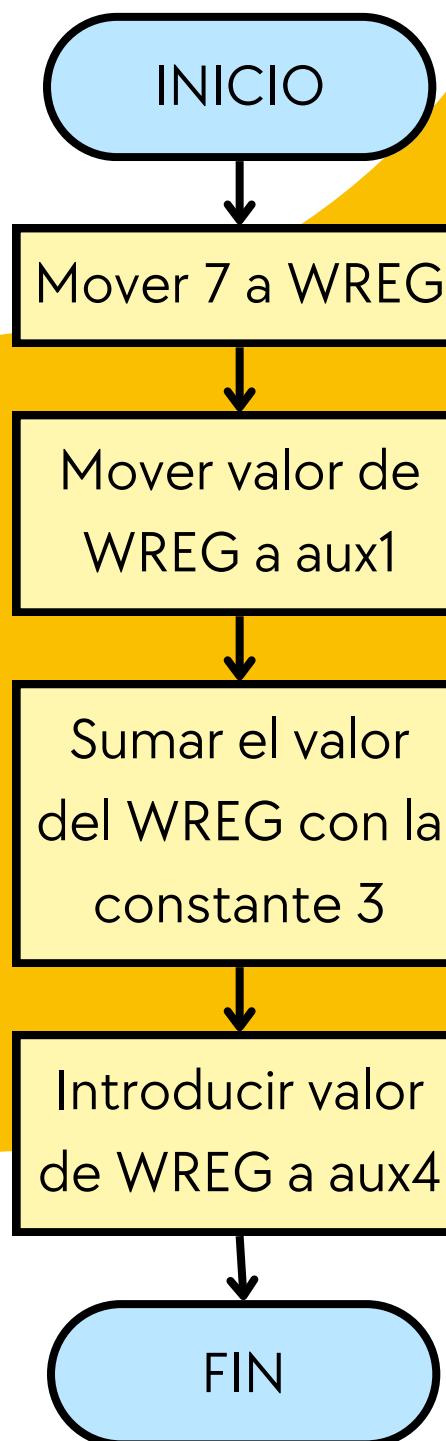
# PSEUDOCÓDIGO EJERCICIOS

## Punto 16

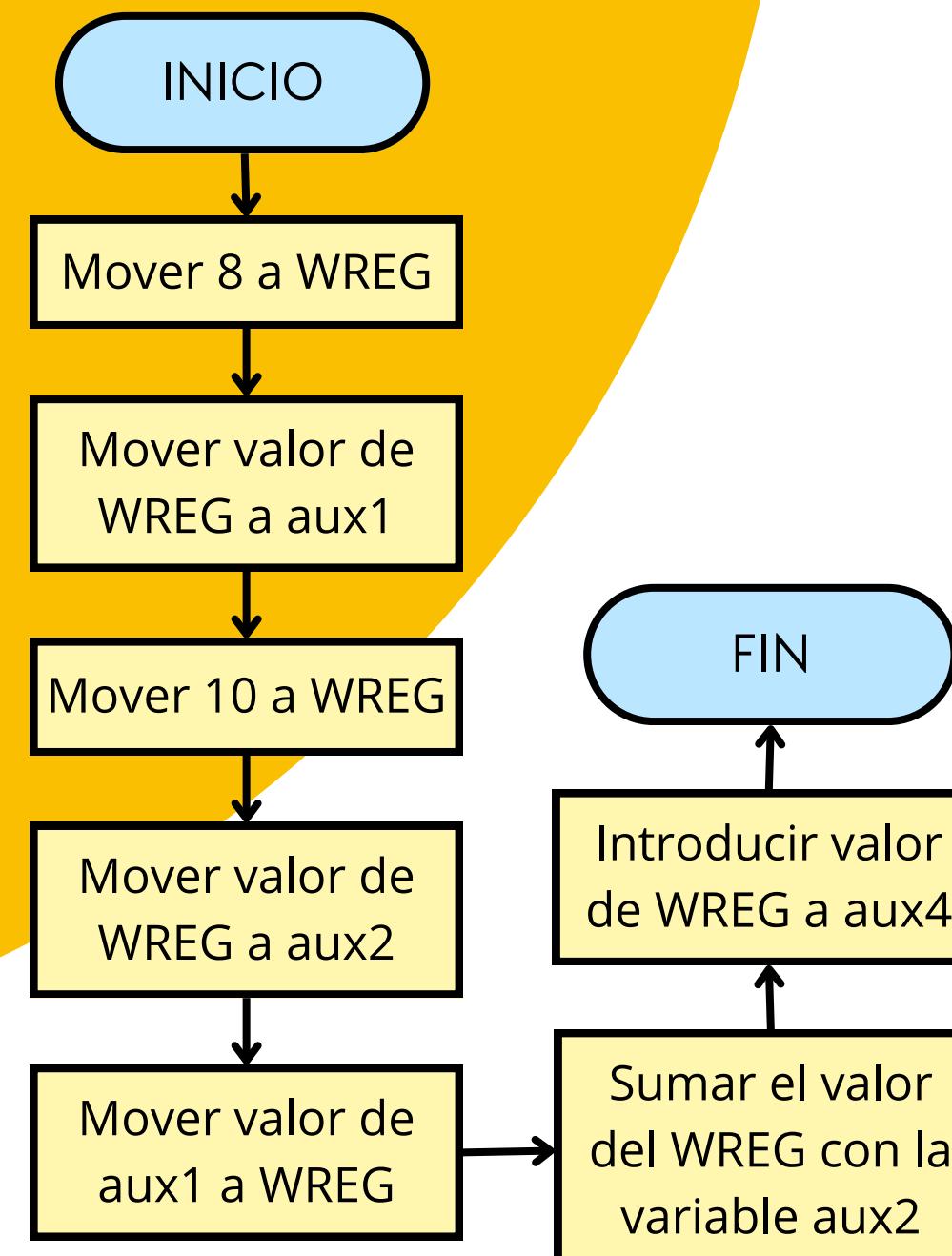
1. Cargar el valor 18 en el registro W
2. Almacenar el valor del registro W en la variable aux1
3. Cargar el valor 60 en el registro W
4. Almacenar el valor del registro W en la variable aux2
5. Cargar el valor 16 en el registro W
6. Almacenar el valor del registro W en la variable aux3
7. Restar la constante 26 (en binario: b'11010') al valor en el registro W
8. Almacenar el resultado en el registro W
9. Multiplicar el contenido del registro W por la constante 3
10. Mover el producto al registro de bajo orden (PRODL)
11. Almacenar el contenido del registro en la variable aux4
12. Mover el valor de la variable aux1 al registro W
13. Sumar el contenido de la variable aux2 al registro W
14. Sumar el contenido de la variable aux4 al registro W
15. Almacenar el resultado en la variable aux4

# DIAGRAMAS DE FLUJO

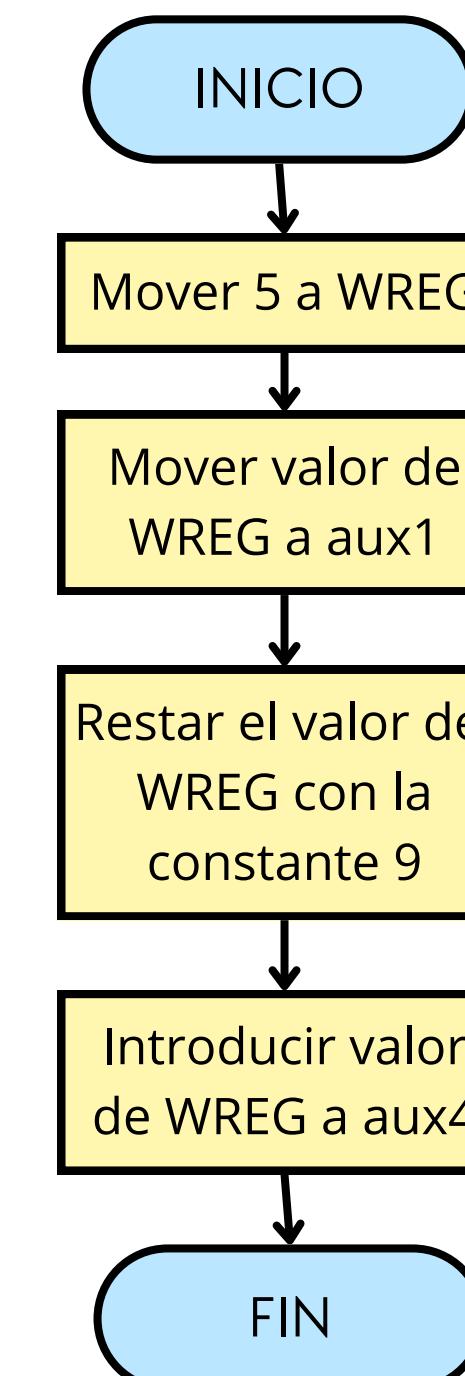
## Punto 1



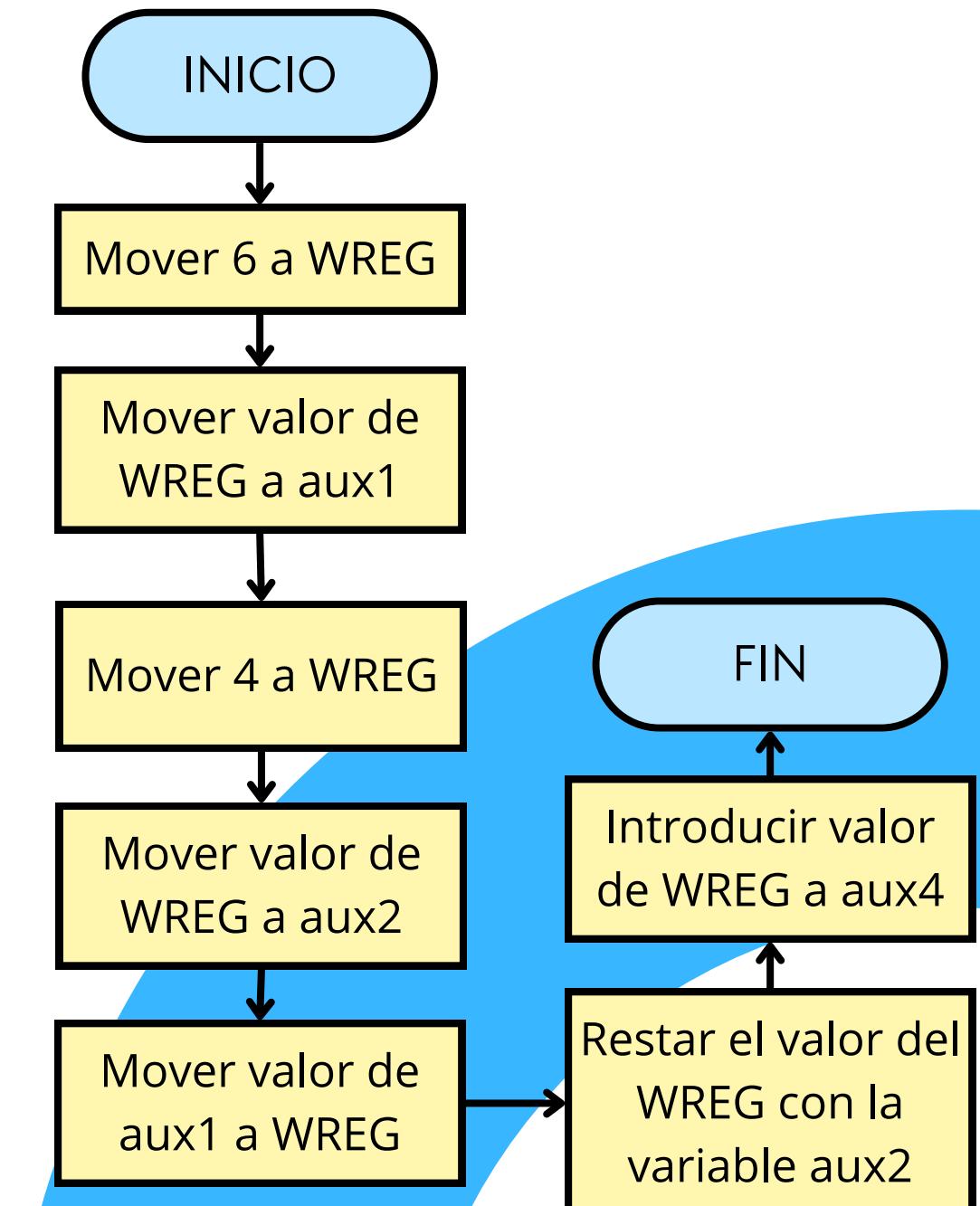
## Punto 2



## Punto 3

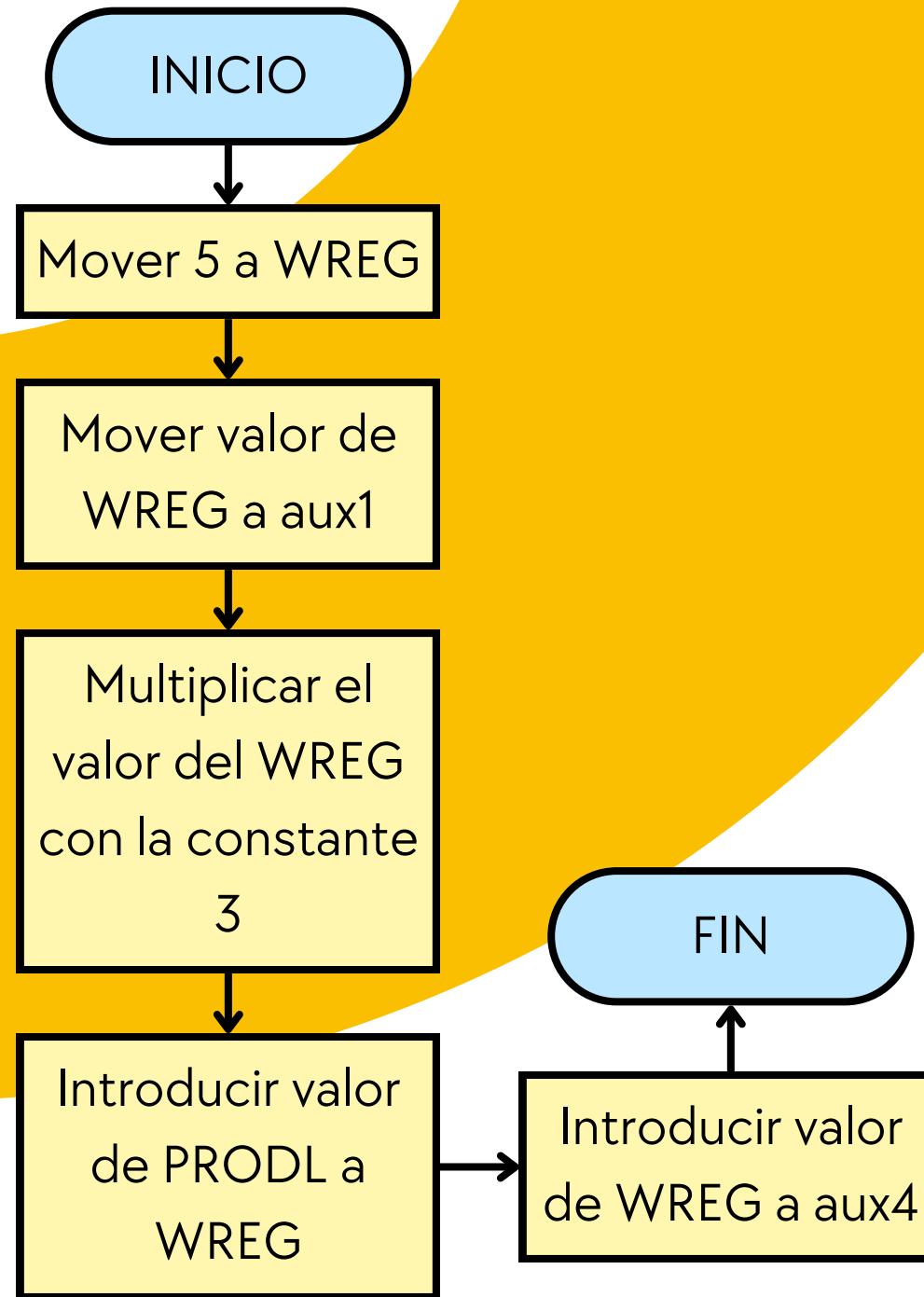


## Punto 4

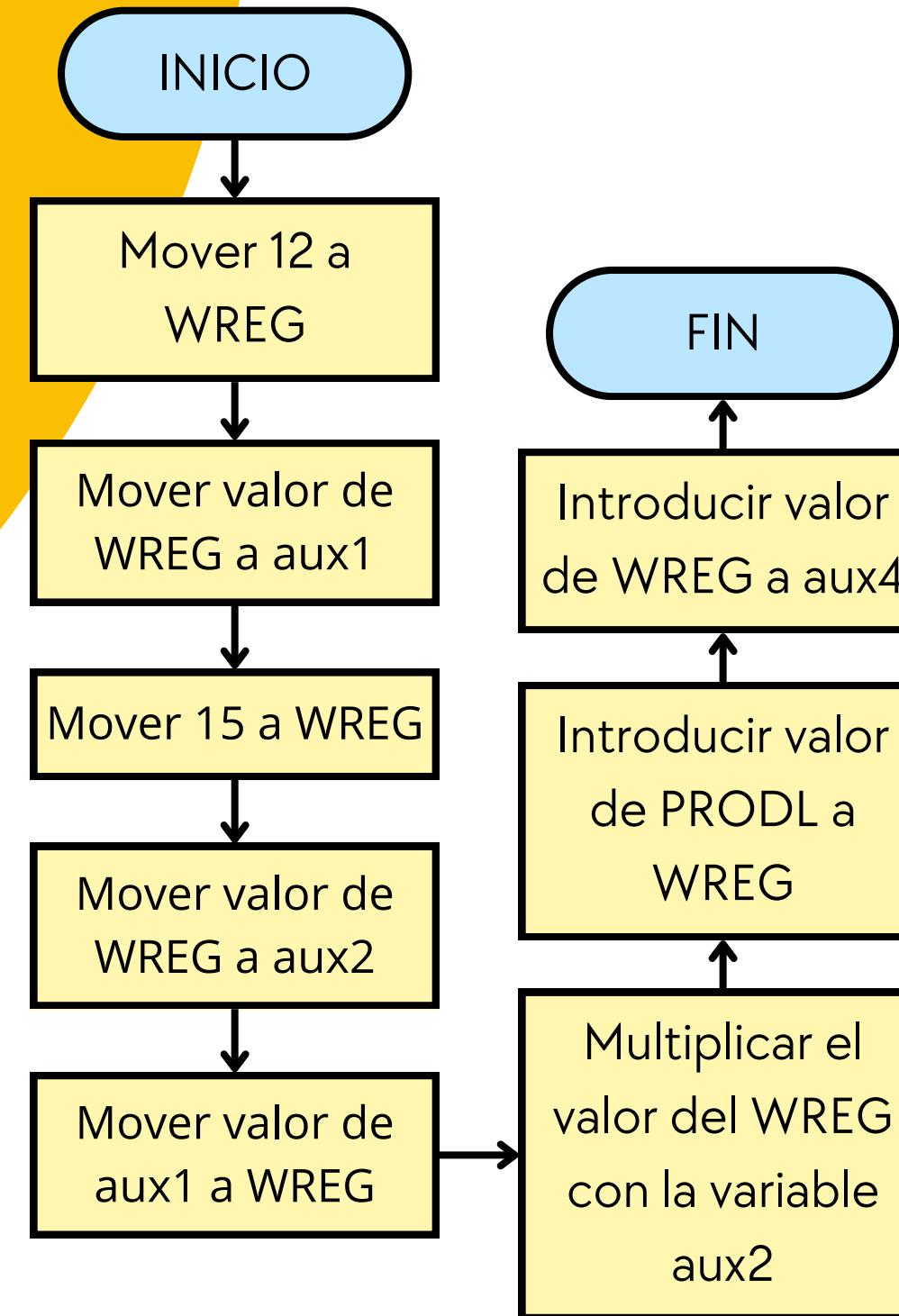


# DIAGRAMAS DE FLUJO

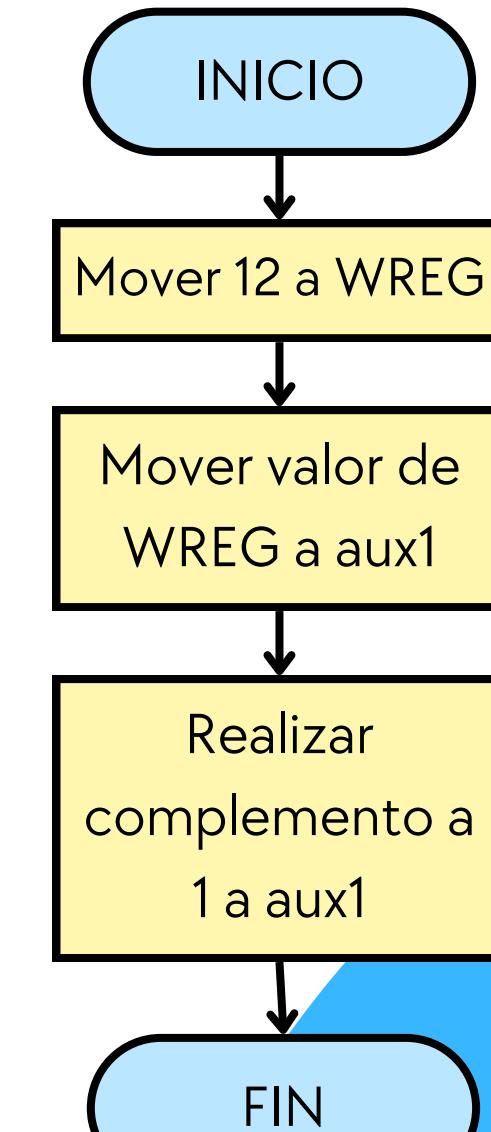
## Punto 5



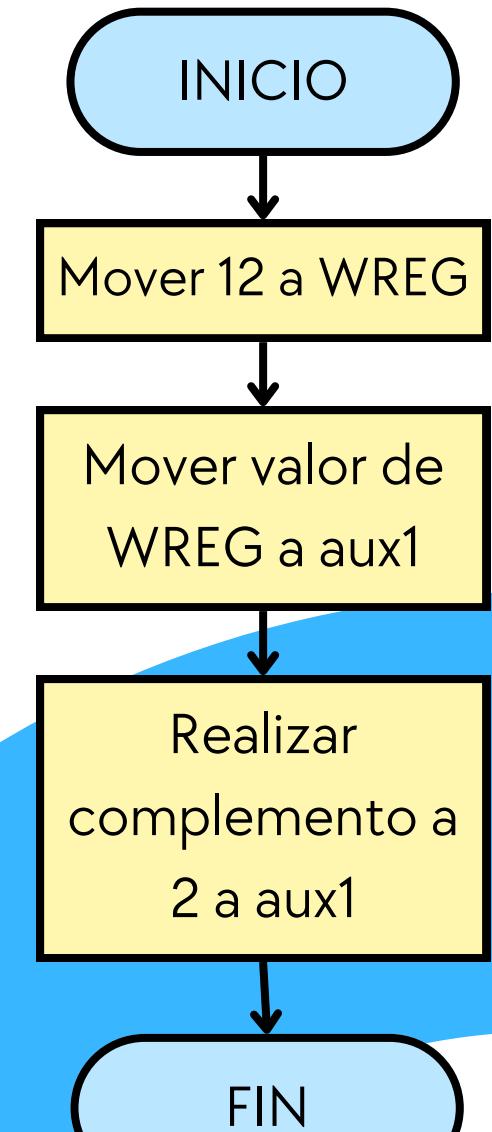
## Punto 6



## Punto 7

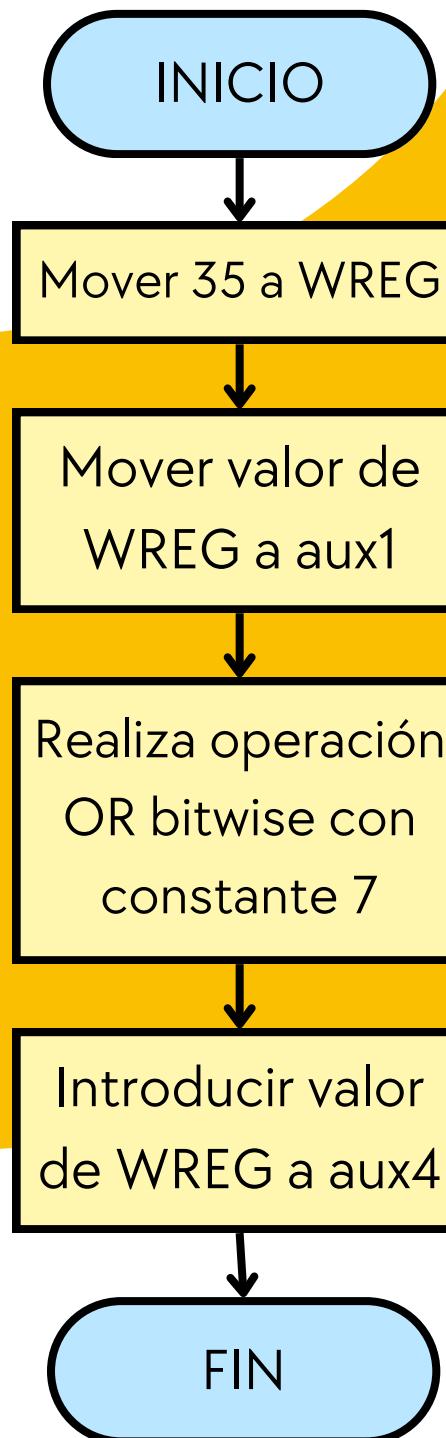


## Punto 8

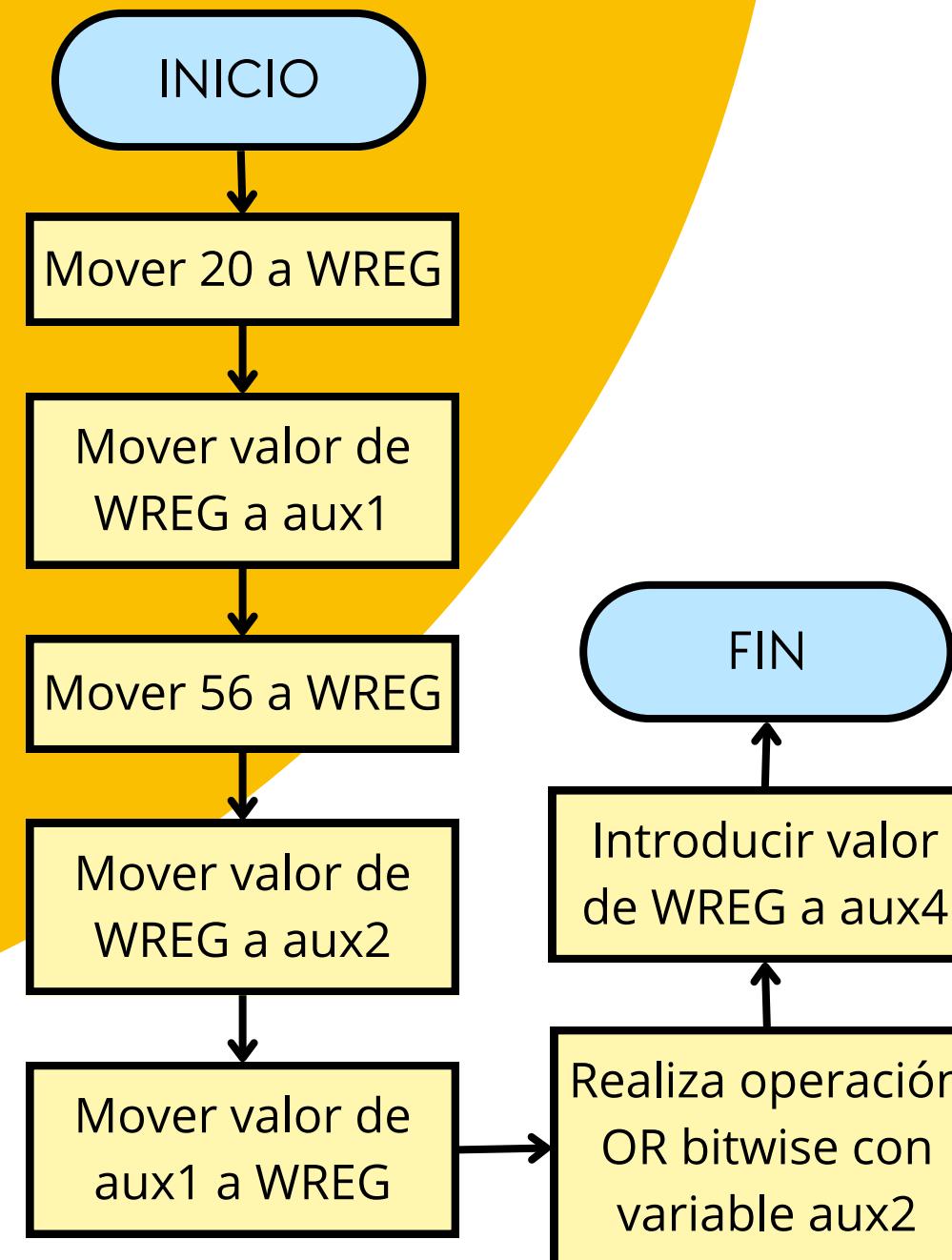


# DIAGRAMAS DE FLUJO

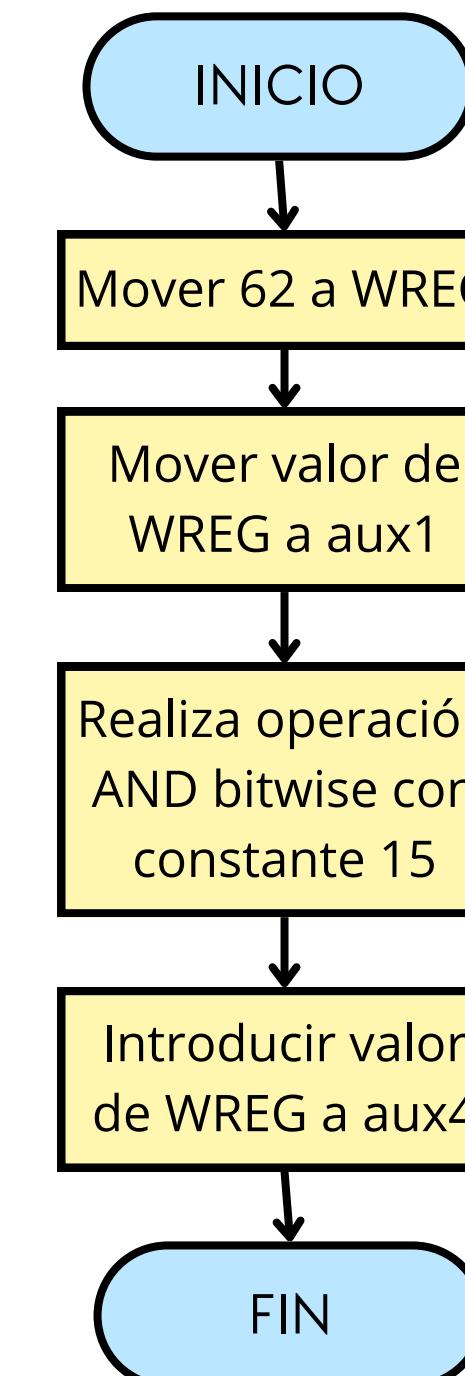
Punto 9



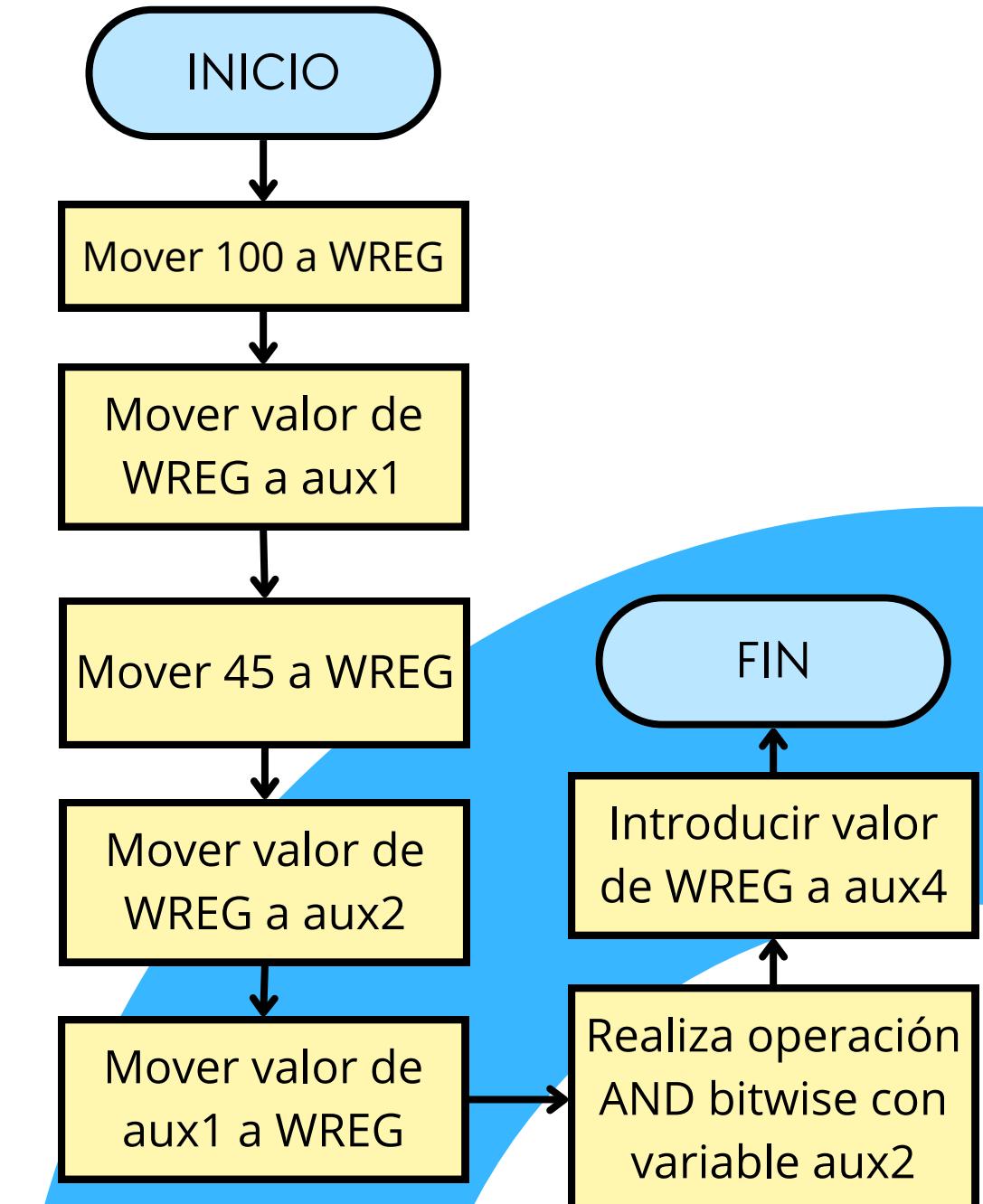
Punto 10



Punto 11

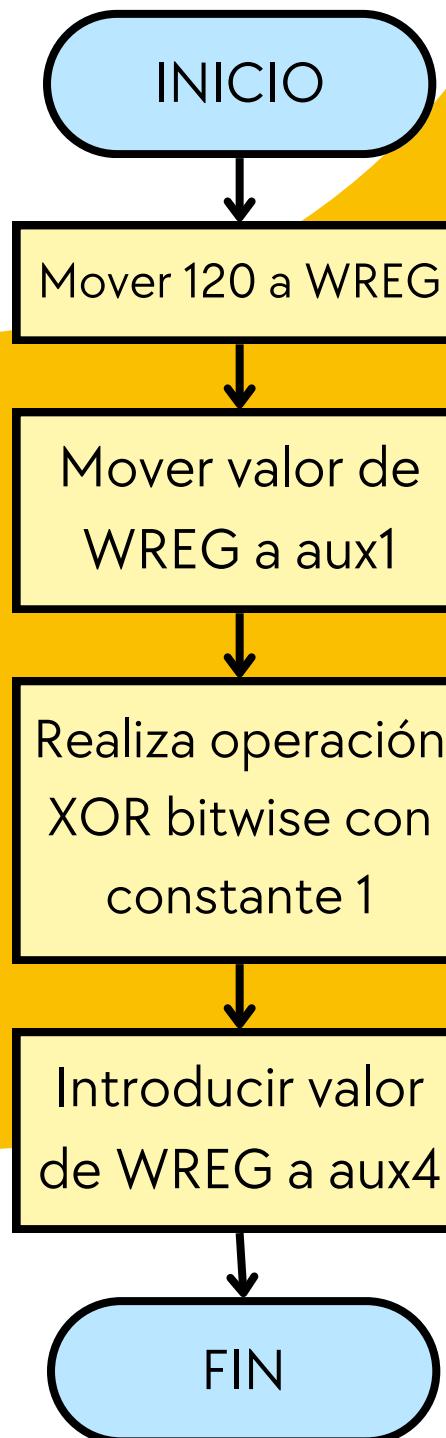


Punto 12

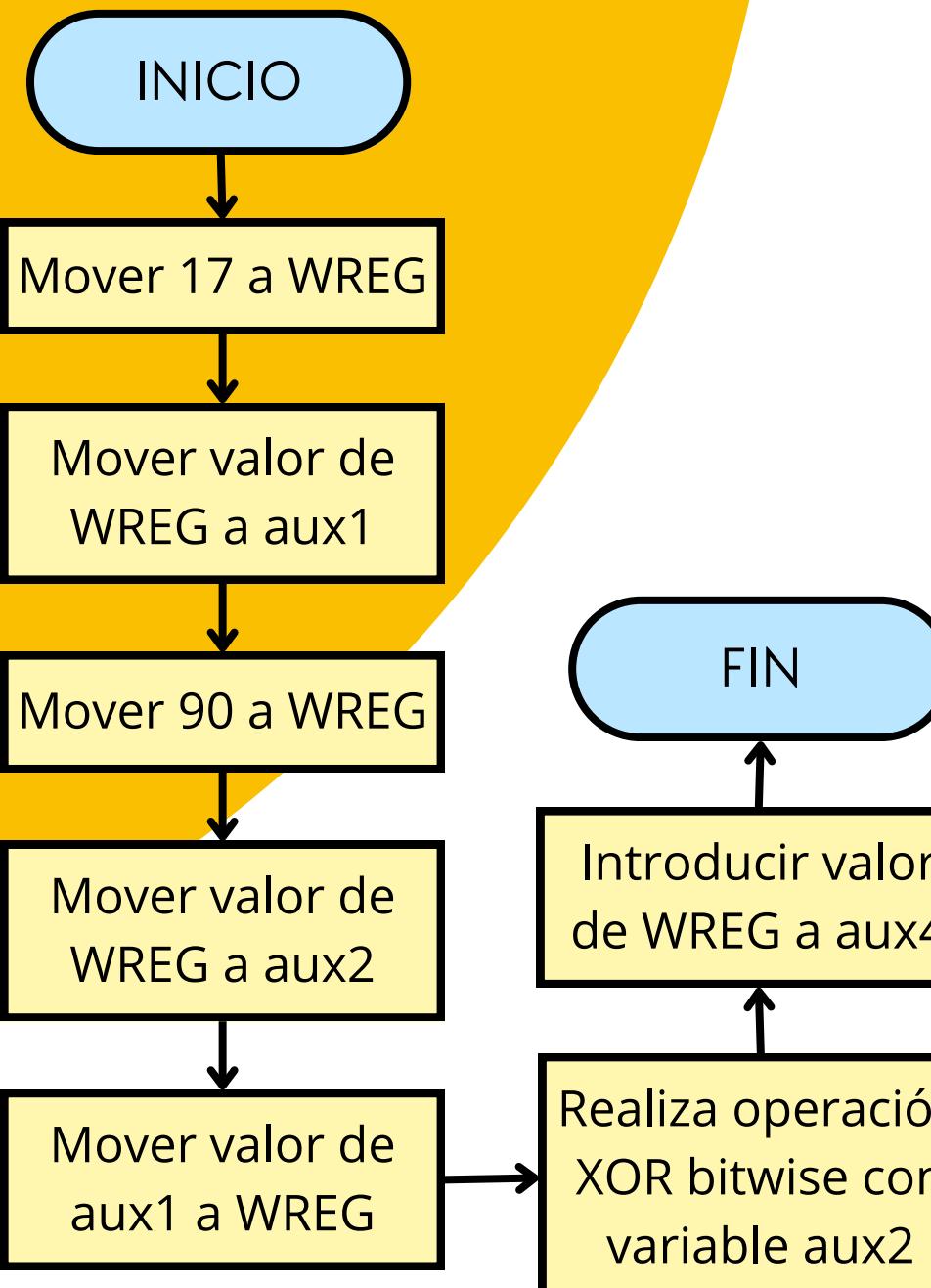


# DIAGRAMAS DE FLUJO

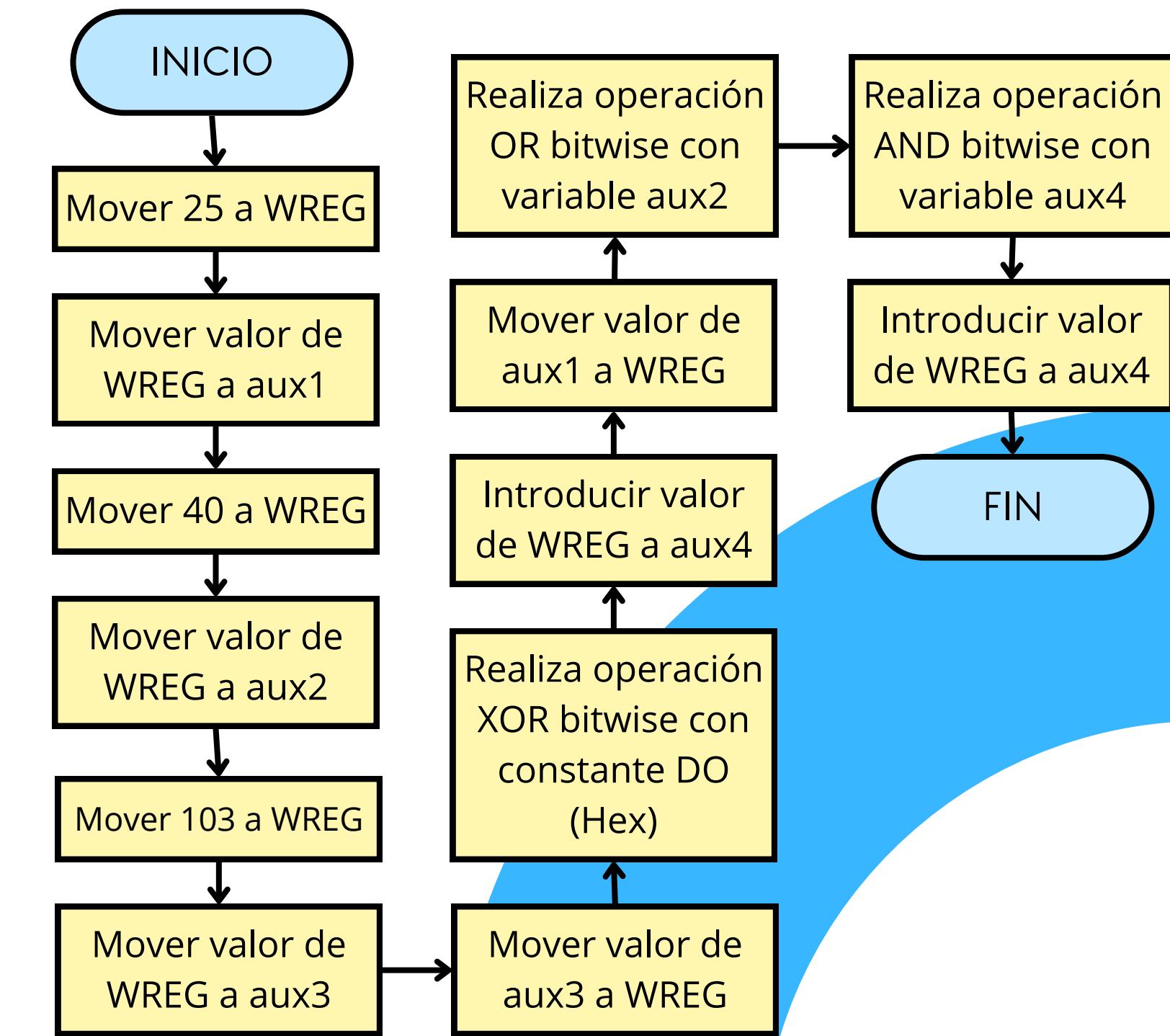
Punto 13



Punto 14

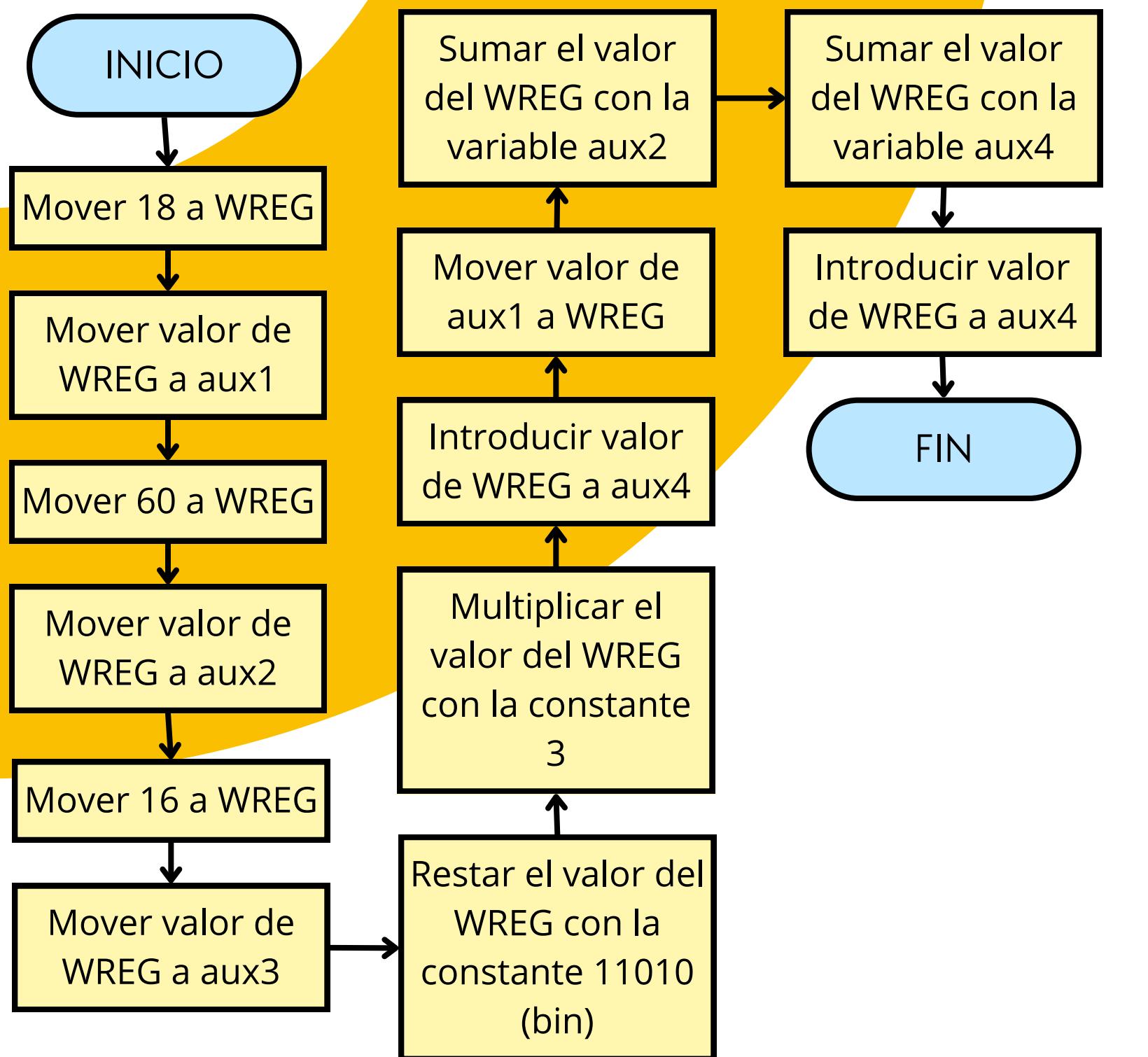


Punto 15



# DIAGRAMAS DE FLUJO

## Punto 16



# USO DEL SIMULADOR

## Declaración de variables

The screenshot shows the Microchip MPLAB X IDE interface. The main window displays assembly code for a PIC18F4550 microcontroller. The code is organized into several sections:

- Primer Sección (Librerías):** include P18F4550.inc
- Segunda Sección (Configuración del Microcontrolador, solo para programar):** aux1 equ 0h, aux2 equ 1h, aux3 equ 2h, aux4 equ 3h
- Tercera Sección (Definición de variables, assembly solo hay un tipo):** movlw .7, movwf aux1, addlw .3 ; Sumar la constante 3 al valor de la variable aux1, movwf aux4; Se carga la suma al valor de la variable aux4
- Cuarta Sección (Inicialización de variables):** movlw .8, movwf aux1, movlw .10

The left sidebar shows the project structure under the 'Projects' tab, with files like ClaseLab2, Lab1Micros, and Lab1Micros2. The bottom-left pane shows the 'Lab1Micros2 - Dashboard' with device configuration (PIC18F4550), compiler toolchain (MPASMWIN v5.67), and memory usage. The bottom-right pane shows the 'Output' window with build logs, indicating a successful build and loading of the program.

```
1 ;Primera Sección (Librerias)
2 include P18F4550.inc
3
4 ;Segunda Sección (Configuracion del Microcontrolador, solo para programar)
5
6 ;Tercera Sección (Definicion de variables, assembly solo hay un tipo)
7 aux1 equ 0h
8 aux2 equ 1h
9 aux3 equ 2h
10 aux4 equ 3h
11
12 ;Cuarta Sección (Inicializacion de variables)
13 Punto1
14     movlw .7
15     movwf aux1
16     addlw .3 ;Sumar la constante 3 al valor de la variable aux1
17     movwf aux4;Se carga la suma al valor de la variable aux4
18
19 Punto2
20     movlw .8
21     movwf aux1
22
23     movlw .10
```

Output x

Simulator x Lab1Micros2 (Build, Load, ...) #2 x Debugger Console x Lab1Micros2 (Clean, Build, ...) x

```
MP2HEX 5.08, COFF to HEX File Converter
Copyright (c) 1998-2011 Microchip Technology Inc.
Errors : 0

make[2]: Leaving directory 'C:/Users/maria/Documents/Unal/Semestre8/Microcontroladores/Lab1Micros2.X'
make[1]: Leaving directory 'C:/Users/maria/Documents/Unal/Semestre8/Microcontroladores/Lab1Micros2.X'

BUILD SUCCESSFUL (total time: 1s)
Loading code from C:/Users/maria/Documents/Unal/Semestre8/Microcontroladores/Lab1Micros2.X/dist/default/production/Lab1Micros2.X.production.hex...
Loading completed
```

# USO DEL SIMULADOR

## Punto 1

The screenshot displays the MPLAB X IDE interface for a project named 'Lab1Micros2'. The assembly source code for 'Lab1Micros2.asm' is shown in the main editor window. The code includes sections for configuration, variable definition, and initialization. The 'Variables' window at the bottom right shows the current state of variables aux1, aux2, aux3, aux4, and WREG.

```
include P18F4550.inc

;Segunda Seccion (Configuracion del Microcontrolador, solo para programar)

;Tercera Seccion (Definicion de variables, assembly solo hay un tipo)
aux1 equ 0h
aux2 equ 1h
aux3 equ 2h
aux4 equ 3h

;Cuarto Seccion (Inicializacion de variables)
Punto1
    movlw .7
    movwf aux1
    addlw .3 ;Sumar la constante 3 al valor de la variable aux1
    movwf aux4;Se carga la suma al valor de la variable aux4

Punto2
    movlw .8
    movwf aux1
    movlw .10
    movwf aux2
```

Name	Address	Hexadecimal	Decimal	Binary
aux1	0x0	0x07	7	00000111
aux2	0x1	0x3C	60	00111100
aux4	0x3	0x6C	108	01101100
WREG	0xFE8	0x0A	10	00001010
aux3	0x2	0x10	16	00010000

# USO DEL SIMULADOR

## Punto 2

The screenshot displays the MPLAB X IDE interface for a PIC18F4550 project named "Lab1Micros2".

**Projects View:** Shows the project structure with three main components: ClaseLab2, Lab1Micros, and Lab1Micros2.

**Assembly Source View:** The main window displays assembly code:

```
19    Punto2
20        movlw .8
21        movwf aux1
22
23        movlw .10
24        movwf aux2
25
26        movf aux1,w ;Llevar el valor de la variable aux1 al registro w
27        addwf aux2,w ;Sumar la constante aux2 al valor guardado en registro w
28        movwf aux4;Se carga la suma al valor de la variable aux4
29
30    Punto3
31        movlw .5
32        movwf aux1
33
34        movf aux1,w ;Llevar el valor de la variable aux1 al sistema
35        sublw .9 ;Restar aux1 al valor constante 9
36        movwf aux4;Se carga la resta al valor de la variable aux4
37
38    Punto4
39        movlw .6
40        movwf aux1
41
```

**Output View:** Shows memory dump tables for variables:

Name	Address	Hexadecimal	Decimal	Binary
aux1	0x0	0x08	8	0001000
aux2	0x1	0x0A	10	0001010
aux4	0x3	0x12	18	00010010
WREG	0xFE8	0x12	18	00010010
aux3	0x2	0x10	16	00010000

# USO DEL SIMULADOR

## Punto 3

The screenshot shows the MPASM IDE interface with the following details:

- Projects:** Shows three projects: ClaseLab2, Lab1Micros, and Lab1Micros2.
- Lab1Micros2.asm - Asm Source:** Displays assembly code for four points:
  - Punto3:** movlw .5  
movwf aux1  
  
movf aux1,w ;Llevar el valor de la variable aux1 al registro w  
addwf aux2,w ;Sumar la constante aux2 al valor guardado en registro w  
movwf aux4;Se carga la suma al valor de la variable aux4
  - Punto4:** movlw .6  
movwf aux1  
  
movf aux1,w ;Llevar el valor de la variable aux1 al sistema  
sublw .9 ;Restar aux1 al valor constante 9  
movwf aux4;Se carga la resta al valor de la variable aux4
- Lab1Micros2 - Dashboard:** Shows project type: Application - Configuration: default, device: PIC18F4550, compiler toolchain: MPASMIN (v5.67), and memory usage: Data 2048 (0x800) bytes (0%), Program 32768 (0x8000) bytes (1%).
- Output - Variables:** A table showing variable values:

Name	Address	Hexadecimal	Decimal	Binary
aux1	0x0	0x05	5	00000101
aux2	0x1	0x0A	10	00001010
aux4	0x3	0x04	4	00000100
WREG	0xFE8	0x04	4	00000100
aux3	0x2	0x10	16	00010000

# USO DEL SIMULADOR

## Punto 4

The screenshot shows the MPASM IDE interface with the following details:

- Projects:** Lab1Micros2, ClaseLab2, Lab1Micros, Lab1Micros2.
- Assembly Source:** Lab1Micros2.asm
- Code:**

```
32    movwf aux1
33
34    movf aux1,w ;Llevar el valor de la variable aux1 al sistema
35    sublw .9 ;Restar aux1 al valor constante 9
36    movwf aux4;Se carga la resta al valor de la variable aux4
37
38 Punto4
39    movlw .6
40    movwf aux1
41
42    movlw .4
43    movwf aux2
44
45    movf aux2,w ;Llevar el valor de la variable aux2(la menor) al registro w
46    subwf aux1,w ;Restar la constante aux1(la mayor) al valor guardado en registro w
47    movwf aux4 ;Se carga la resta al valor de la variable aux4
48
49 Punto5
50    movlw .5
51    movwf aux1
52
53    movf aux1,w ;Llevar el valor de la variable aux1 al registro w
54    mullw .4 ;multiplicar el contenido del registro por la constante 4
```
- Output:** Variables table showing memory dump:

Name	Address	Hexadecimal	Decimal	Binary
aux1	0x0	0x06	6	00000110
aux2	0x1	0x04	4	00000100
aux4	0x3	0x02	2	00000010
WREG	0xFE8	0x02	2	00000010
aux3	0x2	0x10	16	00010000
- Project Summary:** Lab1Micros2 - Dashboard  
Project Type: Application - Configuration: default  
Device: PIC18F4550  
Checksum: Debug Image  
Compiler Toolchain: MPASMWIN (v5.67) [C:\Program Files (x86)\]  
Debug Image: COFF: Optimization  
Memory: Data 2048 (0x800) bytes (0%)  
Program 32768 (0x8000) bytes (1%)  
Debug Tool: Simulator  
Debug Resources: Program BP Used: 1 Free: 999  
Data BP Used: 1 Free: 999  
Data Capture BP: No Support

# USO DEL SIMULADOR

## Punto 5

The screenshot shows the MPASM IDE interface with the following details:

- Projects View:** Shows the project structure with files like ClaseLab2, Lab1Micros, and Lab1Micros2.
- Assembly Source View:** Displays the assembly code for Punto 5 and Punto 6.

```
41      movlw .4
42      movwf aux2
43
44      movf aux2,w ;Llevar el valor de la variable aux2(la menor) al registro w
45      subwf aux1,w ;Restar la constante aux1(la mayor) al valor guardado en registro w
46      movwf aux4 ;Se carga la resta al valor de la variable aux4
47
48      Punto5
49      movlw .5
50      movwf aux1
51
52      movf aux1,w ;Llevar el valor de la variable aux1 al registro w
53      mullw .4 ;multiplicar el contenido del registro por la constante 4
54      movf PRODL,w ;mover el producto al registro
55      movwf aux4 ;mover el contenido del registro a la variable
56
57      Punto6
58      movlw .12
59      movwf aux1
60
61      movlw .15
62      movwf aux2
63
```
- Output View:** Shows the memory usage and debug resources. Memory usage: Data 2048 (0x800) bytes (0%), Program 32768 (0x8000) bytes (1%). Debug resources: Program BP Used: 1 Free: 999, Data BP Used: 1 Free: 999, Data Capture BP: No Support.
- Variables View:** A table showing variable names, addresses, and values.

Name	Address	Hexadecimal	Decimal	Binary
aux1	0x0	0x05	5	0000101
aux2	0x1	0x04	4	0000100
aux4	0x3	0x14	20	00010100
WREG	0xFE8	0x14	20	00010100
aux3	0x2	0x10	16	00010000

# USO DEL SIMULADOR

## Punto 6

The screenshot shows the MPASM IDE interface with the following details:

- Projects:** Lab1Micros2
- Files:** Lab1Micros2.asm
- Code View:** Assembly source code for Punto 6 and Punto 7.
- Punto 6 Code:**

```
53    movf aux1,w ;Llevar el valor de la variable aux1 al registro w
54    mullw .4 ;multiplicar el contenido del registro por la constante 4
55    movf PRODL,w ;mover el producto al registro
56    movwf aux4 ;mover el contenido del registro a la variable

57
58 Punto6
59    movlw .12
60    movwf aux1

61
62    movlw .15
63    movwf aux2

64
65    movf aux1,w ;Llevar el valor de la variable aux1 al registro w
66    mulwf aux2 ;multiplicar el contenido del registro por aux2
67    movf PRODL,w ;mover el producto al registro
68    movwf aux4 ;mover el contenido del registro a la variable
```
- Punto 7 Code:**

```
69
70 Punto7
71    movlw .12
72    movwf aux1

73
74    comf aux1 ;Realizar complemento al registro
```
- Output View:** Shows memory usage and variables.
- Variables View:** Displays the values of variables aux1, aux2, aux4, WREG, aux3, and aux5.

Name	Address	Hexadecimal	Decimal	Binary
aux1	0x0	0x0C	12	00001100
aux2	0x1	0x0F	15	00001111
aux4	0x3	0xB4	180	10110100
WREG	0xFE8	0xB4	180	10110100
aux3	0x2	0x10	16	00010000

# USO DEL SIMULADOR

## Punto 7

The screenshot shows the MPASM IDE interface with the following details:

- Projects:** Shows three projects: ClaseLab2, Lab1Micros, and Lab1Micros2.
- Lab1Micros2.asm - Asm Source:** Displays assembly code for Punto7, Punto8, and Punto9. The code includes instructions like movlw, movwf, mulwf, movf, comf, and negf.
- Lab1Micros2 - Dashboard:** Shows project configuration (Application - Configuration: default), device (PIC18F4550), compiler toolchain (MPASMWIN v5.67), memory usage (Data 2048 (0x800) bytes, Program 32768 (0x8000) bytes), and debug tools (Simulator).
- Output - Variables:** A table showing variable values for aux1, aux2, aux4, WREG, and aux3. aux1 is 243 (0xF3), aux2 is 15 (0x0F), aux4 is 180 (0xB4), WREG is 12 (0x0C), and aux3 is 16 (0x10).

Name	Address	Hexadecimal	Decimal	Binary
aux1	0x0	0xF3	243	11110011
aux2	0x1	0x0F	15	00001111
aux4	0x3	0xB4	180	10110100
WREG	0xFE8	0x0C	12	00001100
aux3	0x2	0x10	16	00010000
<Enter new watch>				

# USO DEL SIMULADOR

## Punto 8

The screenshot shows the MPLAB X IDE interface with the following components:

- Projects View:** Shows the project structure with three main folders: ClaseLab2, Lab1Micros, and Lab1Micros2.
- Code Editor:** Displays assembly code for Lab1Micros2.asm. The code includes comments in Spanish explaining the operations:
  - Lines 65-69: movf aux1,w ;Llevar el valor de la variable aux1 al registro w  
mulwf aux2 ;multiplicar el contenido del registro por aux2  
movf PRODL,w ;mover el producto al registro  
movwf aux4 ;mover el contenido del registro a la variable
  - Line 70: Punto7
  - Lines 71-72: movlw .12  
movwf aux1
  - Line 74: comf aux1 ;Realizar complemento al registro
  - Line 76: Punto8
  - Lines 77-78: movlw .12  
movwf aux1
  - Line 80: negf aux1 ;Realizar complemento a 2 al registro
  - Line 82: Punto9
  - Lines 83-84: movlw .35  
movwf aux1
  - Line 86: movf aux1,w ;Llevar el valor de la variable aux1 al registro w  
iorlw .7 ;Realizar OR
- Project Explorer:** Shows the project configuration for Lab1Micros2, including Device (PIC18F4550), Compiler Toolchain (MPASMWIN v5.67), and Memory usage.
- Output View:** Displays a table of variables and their values:

Name	Address	Hexadecimal	Decimal	Binary
aux1	0x0	0xF4	244	11110100
aux2	0x1	0x0F	15	00001111
aux4	0x3	0xB4	180	10110100
WREG	0xFE8	0x0C	12	00001100
aux3	0x2	0x10	16	00010000

# USO DEL SIMULADOR

## Punto 9

The screenshot shows the MPASM IDE interface with the following details:

- Projects View:** Shows three projects: ClaseLab2, Lab1Micros, and Lab1Micros2. Lab1Micros2 is currently selected.
- Assembly Source (Lab1Micros2.asm):** Displays assembly code with color-coded labels:
  - Line 73: `comf aux1 ;Realizar complemento al registro`
  - Line 76: `Punto8`
  - Line 77: `movlw .12`
  - Line 78: `movwf aux1`
  - Line 80: `negf aux1 ;Realizar complemento a 2 al registro`
  - Line 82: `Punto9`
  - Line 83: `movlw .35`
  - Line 84: `movwf aux1`
  - Line 86: `movf aux1,w ;Llevar el valor de la variable aux1 al registro w`
  - Line 87: `iorlw .7 ;Realizar OR`
  - Line 88: `movwf aux4 ;mover el contenido del registro a la variable`
  - Line 90: `Punto10`
  - Line 91: `movlw .20`
  - Line 92: `movwf aux1`
  - Line 94: `movlw .56`
  - Line 95: `movwf aux2`
- Project Configuration (Lab1Micros2 - Dashboard):** Shows the project type is Application - Configuration: default. It includes sections for Device (PIC18F4550), Compiler Toolchain (MPASMWIN v5.67), and Memory (Data 2048 (0x800) bytes, Program 32768 (0x8000) bytes).
- Output Window:** Displays memory dump and variable watch information. The memory dump shows Data 2048 (0x800) bytes starting at address 0x0 with value 0%. The variable watch window shows the following table:

Name	Address	Hexadecimal	Decimal	Binary
aux1	0x0	0x23	35	00100111
aux2	0x1	0x0F	15	00001111
aux4	0x3	0x27	39	00100111
WREG	0xFE8	0x27	39	00100111
aux3	0x2	0x10	16	00010000

# USO DEL SIMULADOR

## Punto 10

The screenshot displays the MPLAB X IDE interface with the following details:

- Projects View:** Shows the project structure with folders for ClaseLab2, Lab1Micros, and Lab1Micros2.
- Assembly Source (Lab1Micros2.asm):**

```
84    movwf aux1
85
86    movf aux1,w ;Llevar el valor de la variable aux1 al registro w
87    iorlw .7 ;Realizar OR
88    movwf aux4 ;mover el contenido del registro a la variable
89
90 Punto10
91    movlw .20
92    movwf aux1
93
94    movlw .56
95    movwf aux2
96
97    movf aux1,w ;Llevar el valor de la variable aux1 al registro w
98    iorwf aux2,w ;Realizar OR
99    movwf aux4 ;mover el contenido del registro a la variable
100
101 Punto11
102    movlw .62
103    movwf aux1
104
105    movf aux1,w ;Llevar el valor de la variable aux1 al registro w
106    andlw .15 ;Realizar AND
```
- Output Window:** Shows the state of variables:

Name	Address	Hexadecimal	Decimal	Binary
aux1	0x0	0x14	20	00010100
aux2	0x1	0x38	56	00111000
aux4	0x3	0x3C	60	00111100
WREG	0xFE8	0x3C	60	00111100
aux3	0x2	0x10	16	00010000
<Enter new watch>				
- Device Manager:** Shows the selected device as PIC18F4550.

# USO DEL SIMULADOR

## Punto 11

The screenshot shows the MPASM IDE interface with the following components:

- Projects View:** Shows the project structure with ClaseLab2, Lab1Micros, and Lab1Micros2.
- Assembly Source View:** Displays the assembly code for Lab1Micros2.asm. The code includes instructions like movwf, movlw, iorwf, and andlw, along with comments in Spanish explaining the purpose of each section.
- Dashboard View:** Shows the project configuration for Lab1Micros2, including the device (PIC18F4550), compiler toolchain (MPASMWIN v5.67), and memory usage details.
- Output View:** A table showing the current values of variables used in the assembly code. The variables and their values are:

Name	Address	Hexadecimal	Decimal	Binary
aux1	0x0	0x3E	62	00111110
aux2	0x1	0x38	56	00111000
aux4	0x3	0x0E	14	00001110
WREG	0xFE8	0x0E	14	00001110
aux3	0x2	0x10	16	00010000

# USO DEL SIMULADOR

## Punto 12

The screenshot shows the MPASM IDE interface with the following details:

- Projects:** Shows the project structure with three main sections: ClaseLab2, Lab1Micros, and Lab1Micros2.
- Assembly Source:** The current file is Lab1Micros2.asm, containing assembly code for points 12 through 14. The code includes instructions like movwf, movlw, and xorlw.
- Project Configuration:** The Project Type is Application - Configuration: default. The Device is set to PIC18F4550. The Compiler Toolchain is MPASMWIN (v5.67) and the Memory is Data 2048 (0x800) bytes.
- Output Window:** Displays a memory dump for registers aux1, aux2, aux4, WREG, and aux3. The values are as follows:

Name	Address	Hexadecimal	Decimal	Binary
aux1	0x0	0x64	100	01100100
aux2	0x1	0x20	45	00101101
aux4	0x3	0x24	36	00100100
WREG	0xFE8	0x24	36	00100100
aux3	0x2	0x10	16	00010000

# USO DEL SIMULADOR

## Punto 13

The screenshot shows the MPLAB X IDE interface with the following components:

- Projects View:** Shows the project structure with ClaseLab2, Lab1Micros, and Lab1Micros2.
- Code Editor:** Displays the assembly code for Lab1Micros2.asm. The code includes instructions like movwf, movlw, and movf, along with comments in Spanish explaining the purpose of each section.
- Memory View:** Shows the memory usage for the PIC18F4550 device, indicating 2048 bytes used and 32768 bytes free.
- Variables View:** A table showing the current values of variables aux1, aux2, aux4, WREG, aux3, and aux5. The values are listed in Hexadecimal, Decimal, and Binary formats.

Name	Address	Hexadecimal	Decimal	Binary
aux1	0x0	0x78	120	01111000
aux2	0x1	0x20	45	00101101
aux4	0x3	0x79	121	01111001
WREG	0xFE8	0x79	121	01111001
aux3	0x2	0x10	16	00010000

# USO DEL SIMULADOR

## Punto 14

The screenshot shows the MPLAB X IDE interface with the following components:

- Projects View:** Shows three projects: ClaseLab2, Lab1Micros, and Lab1Micros2. The Lab1Micros2 project is selected.
- Assembly Source View:** Displays the assembly code for Lab1Micros2.asm. The code includes instructions for moving values between registers (aux1, aux2, aux4) and performing XOR operations. It also contains sections labeled Punto14 and Punto15.
- Memory View:** Shows the memory usage for the Lab1Micros2 project. It displays Data (0x800 bytes used) and Program (32768 bytes used) sections.
- Variables View:** A table showing the current values of variables: aux1 (0x11, 17, 00010001), aux2 (0x5A, 90, 01011010), aux4 (0x4B, 75, 01001011), WREG (0x4B, 75, 01001011), and aux3 (0x10, 16, 00010000).

Key assembly code snippets from the source view:

```
122 movwf aux1
123
124 movf aux1,w ;Llevar el valor de la variable aux1 al registro w
125 xorlw .1 ;Realizar XOR
126 movwf aux4 ;mover el contenido del registro a la variable
127
128 Punto14
129 movlw .17
130 movwf aux1
131
132 movlw .90
133 movwf aux2
134
135 movf aux1,w ;Llevar el valor de la variable aux1 al registro w
136 xorwf aux2,w ;Realizar XOR
137 movwf aux4 ;mover el contenido del registro a la variable
138
139 Punto15
140 movlw .25
141 movwf aux1
142
143 movlw .40
144 movwf aux2
```

# USO DEL SIMULADOR

## Punto 15

The screenshot shows the MPLAB X IDE interface with the following components:

- Projects View:** Shows the project structure with ClaseLab2, Lab1Micros, and Lab1Micros2.
- Assembly Source View:** Displays the assembly code for Punto 15 and Punto 16. The code includes instructions like movlw, movwf, xorlw, and andwf, along with comments explaining the operations.
- Memory Dump View:** A table showing memory dump information for variables aux1, aux2, aux4, WREG, aux3, and aux5. The table includes columns for Name, Address, Hexadecimal, Decimal, and Binary.
- Output View:** Shows build, load, and debugger status.

**Assembly Source (Lab1Micros2.asm):**

```
138
139 Punto15
140     movlw .25
141     movwf aux1
142
143     movlw .40
144     movwf aux2
145
146     movlw .103
147     movwf aux3
148
149     movf aux3,w ;Llevar el valor de la variable aux1 al registro w
150     xorlw 0xD0 ;Realizar XOR
151     movwf aux4 ;mover el contenido del registro a la variable
152
153     movf aux1,w ;Llevar el valor de la variable aux1 al registro w
154     iorwf aux2,w ;Realizar OR
155     andwf aux4,w ;Realizar AND
156     movwf aux4 ;mover el contenido del registro a la variable
157
158 Punto16
159     movlw .18
160     movwf aux1
```

**Memory Dump:**

Name	Address	Hexadecimal	Decimal	Binary
aux1	0x0	0x19	25	00011001
aux2	0x1	0x28	40	00101000
aux4	0x3	0x31	49	00110001
WREG	0xFE8	0x31	49	00110001
aux3	0x2	0x67	103	01100111

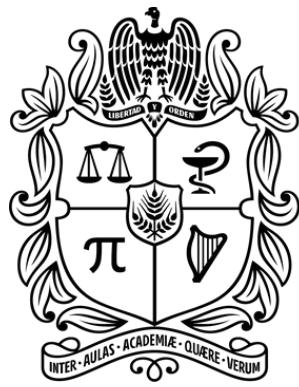
# USO DEL SIMULADOR

## Punto 16

The screenshot shows the MPASM IDE interface with the following details:

- Projects:** The project tree on the left shows three main projects: ClaseLab2, Lab1Micros, and Lab1Micros2. Under Lab1Micros2, there is a file named Lab1Micros.asm.
- Assembly Source:** The main window displays assembly code for "Punto 16". The code includes:
  - Initialization: `movlw .18` and `movwf aux1`.
  - Setting aux2: `movlw .60` and `movwf aux2`.
  - Setting aux3: `movlw .16` and `movwf aux3`.
  - A subtraction operation: `sublw b'11010'` followed by a note "multiplicar el contenido del registro por aux2".
  - Multiplication: `mullw .3; multiplicar el contenido del registro por aux2`.
  - Product storage: `movf PRODL,w ;mover el producto al registro`.
  - Temporary storage: `movwf aux4`.
  - Intermediate calculations: `movf aux1, w`, `addwf aux2,w`, and `addwf aux4,w`.
  - Final output: `goto Punto1` and `end ; Fin del Código`.
- Output:** The bottom pane shows a memory dump table with the following data:

Name	Address	Hexadecimal	Decimal	Binary
aux1	0x0	0x12	18	00010010
aux2	0x1	0x3C	60	00111100
aux4	0x3	0x6C	108	01101100
WREG	0xFE8	0x6C	108	01101100
aux3	0x2	0x10	16	00010000
- Dashboard:** The left sidebar provides project configuration information, including the device (PIC18F4550), compiler (MPASMWIN v5.67), and memory usage details.



UNIVERSIDAD  
**NACIONAL**  
DE COLOMBIA

INTER-AULAS-ACADEMIA-QUIERE-VERUM

# GRACIAS

FACULTAD DE INGENIERÍA  
DEPARTAMENTO DE MECÁNICA Y MECATRÓNICA