

```

.*****
;
; TP 1 - Comparación de números y encendido de Leds según el resultado
; Este programa compara 2 números (n1 y n2)
; *Si N1 es mayor o igual a N2, enciende un led conectado al puerto RB1 Pic
; *Si N1 es menor, enciende un led conectado al puerto RB2 del Pic
; Alumno Luciano Moliterno
; DNI 38.705.112
.*****
;

```

```

#include <P16F628A.INC> ; Incluimos el archivo que contiene los registros Pic

```

```

NUM1 Equ 0x20 ; asigno las posicon de memoria 0x20 como variable a la que llamare
NUM1

```

```

NUM2 Equ 0x21 ; asigno las posicon de memoria 0x20 como variable a la que llamare
NUM2

```

```

org 0 ; Indica que el programa comienza desde la posicon 0 de la
memoria

```

```

Inicio:

```

```

;//////////////////////////////// SETEO DE LED //////////////////////////////////

```

```

SetLed ;Seccion de seteo de salidas

```

```

BSF STATUS,RP0 ;Selecciono el banco de memoria 1
BCF TRISB,1 ;Configuro el pin RB1 como salida
BCF TRISB,2 ;Configuro el pin RB1 como salida
BCF STATUS,RP0 ;Volvemos al banco 0
BCF PORTB,1 ;PONGO A 0 EL BIT 1 DEL PORTB
BCF PORTB,2 ;PONGO A 0 EL BIT 2 DEL PORTB

```

```

;//////////////////////////////// CARGA DE NUMEROS //////////////////////////////////

```

```

NLoad1 ; NumberLoad, carga de numeros

```

```

MOVLW 0x250 ;Numero1 en este caso en valores Hexadecimales
;se puede alterar el resultado del programa cambiando
este numero
MOVWF NUM1 ;Se lo asigno a la variable Num1

```

```

CLRW ;Luego de la carga de N1 limpio el registro W

```

```

NLoad2 ; NumberLoad, carga de numeros

```

```

MOVLW 0x250 ;Numero2 en este caso en valores Hexadecimales,
;se puede alterar el resultado del programa cambiando este
numero
MOVWF NUM2 ;Se lo asigno a la variable Num2

```

```

CLRW ;Luego de la carga de N1 limpio el registro W

```

```

;//////////////////////////////// COMPARACION DE NUMEROS //////////////////////////////////

```

NComp ;comparacion de numeros

```
movf  NUM1,W ; muevo el valor de NUM1 al registro W
subwf NUM2,W ; resto N2 al registro W que contiene el valor de NUM1
```

```
btfsc STATUS,c ; comprueba el flag C del registro STATUS
goto led2      ;si es mayor o igual, la resta sera positiva
               ; por lo tanto el bit C sera 1 y saltara a Led2
```

```
goto led1      ;si es menor, la resta dara sera negativa
               ;por lo tanto el bit C sera 0 y saltara a Led2
```

;///////////////////////////////// ENCENDIDO DE LEDS ///////////////////////////////////
Led1 ;NUM2 ES MENOR A NUM1

```
bsf  PORTB,1  ;Enciendo el Led 1
goto $        ;Mantiene el led encendido
```

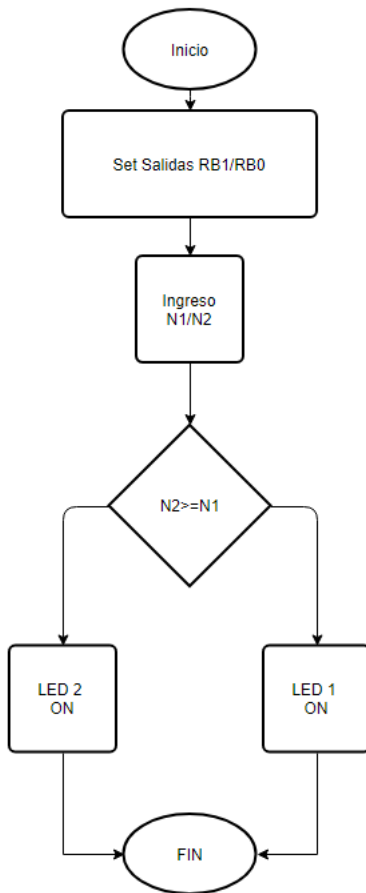
Led2 ;NUM2 ES Igual o mayor A NUM1

```
bsf  PORTB,2  ;Enciendo el Led 2
goto $        ;Mantiene el led encendido
```

end

.*****
;

Diagrama de flujo del Programa

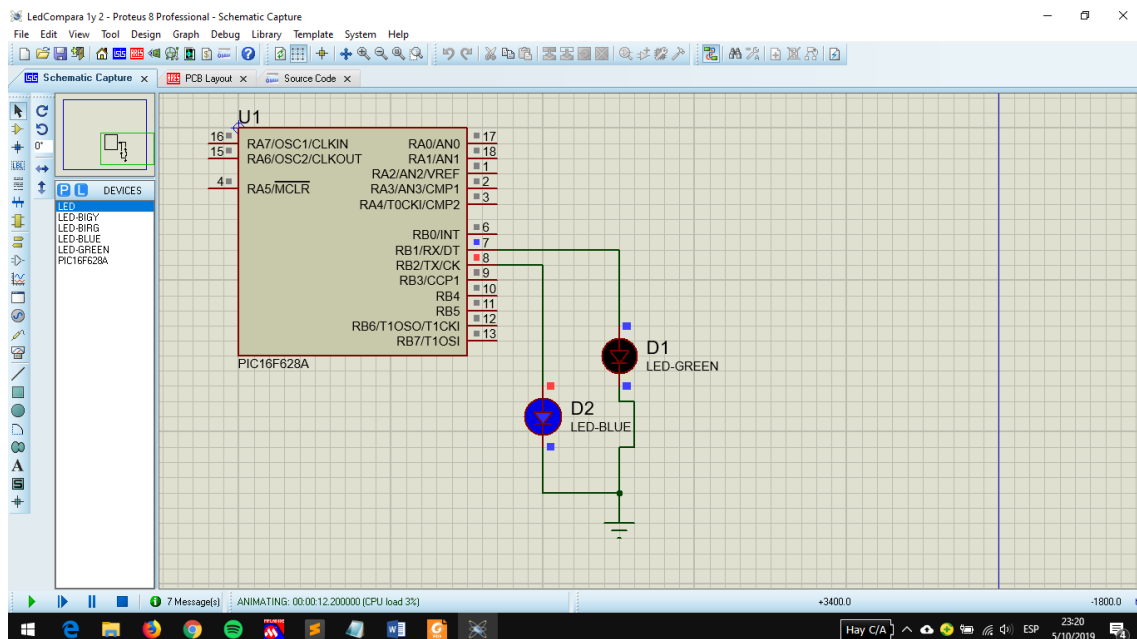


Explicacion del diagrama:

1. Inicio del programa
2. Configurar como salidas los puertos RB1 y RB2
3. Se ingresan 2 valores llamados Num1 y Num2 en el programa, por una cuestión de espacio en el diagrama les puse N1 N2
4. Si N2 es mayor o igual que N1 se encenderá el LED N2.
5. En caso contrario se encenderá el LED N1
6. Final del programa, dejando el led encendido

Prueba en proteus

Numero mayor o igual



Numero menor

