



Nombre de la práctica	CONTADOR			No.	1
Asignatura:	Arquitectura de computadoras	Carrera:	Ingeniería en Sistemas Computacionales	Duración de la práctica (Horas)	12 horas

Nombre: Abimael Gabriel Carlos

Grupo:312

I. Competencia(s) específica(s):

II. Lugar de realización de la práctica (laboratorio, taller, aula u otro):

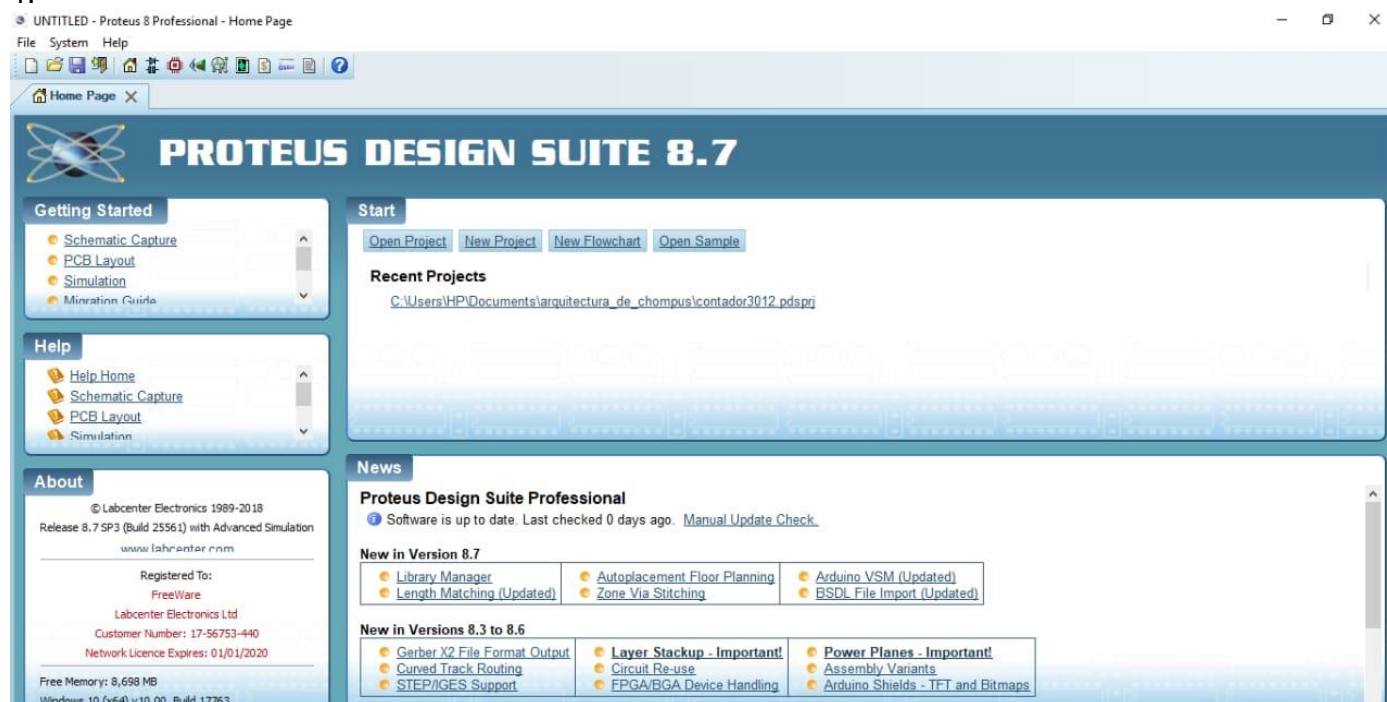
Aula de clase

III. Material empleado:

- 1 tabla Protoboard
- 1 Pic 16F84A
- 1 Cristal oscilador de 4 MHZ
- 1 Resistencia de 220 OMS
- 2 Capacitores cerámicos
- 2 Picofaradios
- 1 Cátodo de 7 segmentos
- 2 Metros de cable UTP

IV. Desarrollo de la práctica: Lo que realizamos para llevar acabó esta práctica fue desarrollar el prototipo de la tabla Protoboar ya con todas las conexiones y accesorios para que esta pueda funcionar en PROTEUS DESIGN SUITE 8.7

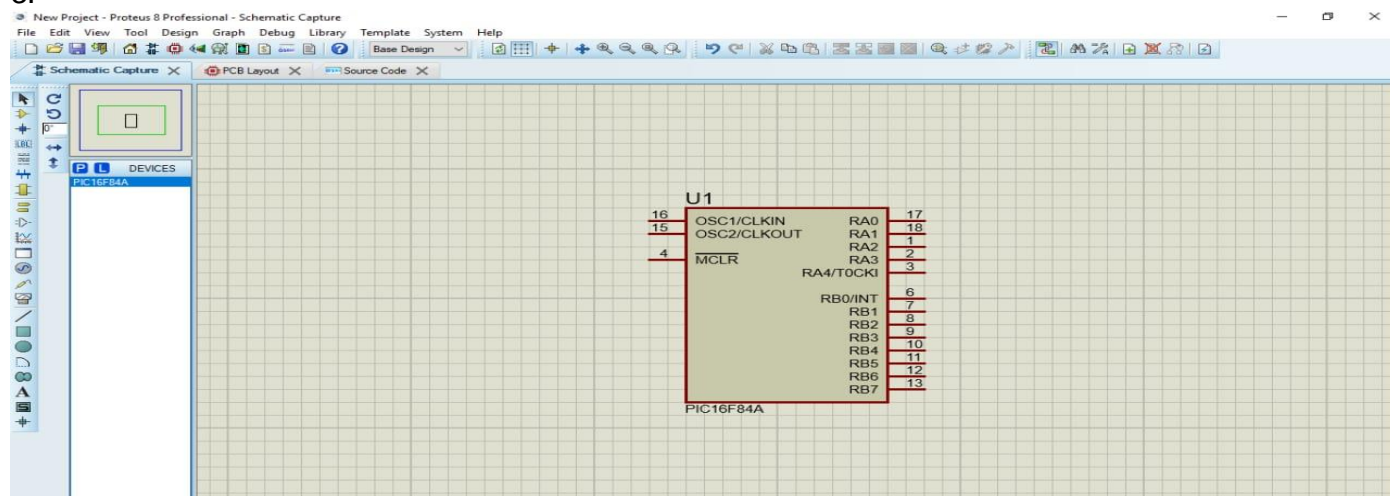
1.-



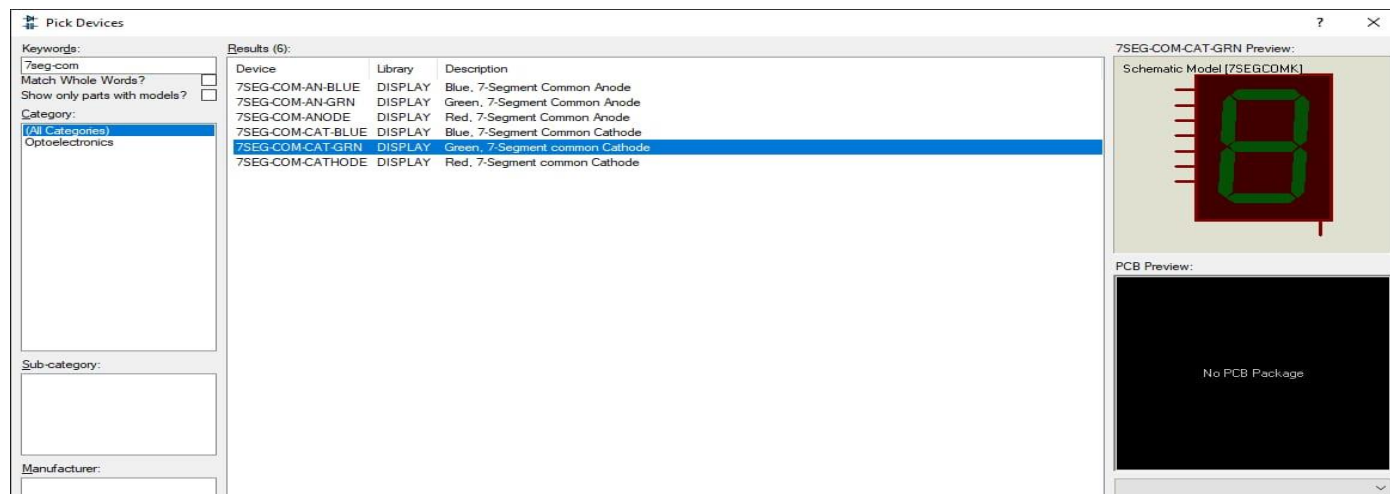
2.-



3.-

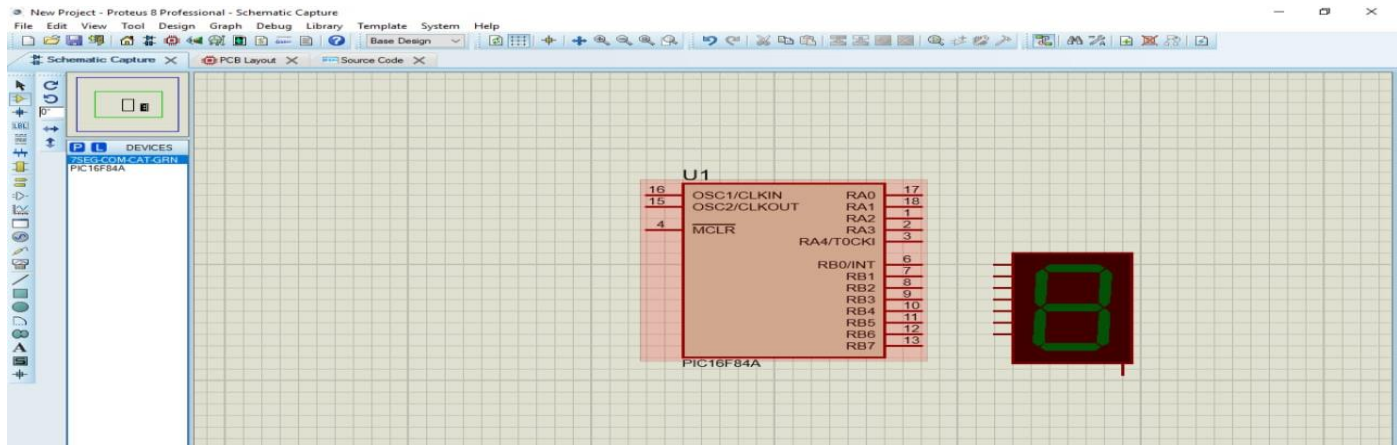


4.-

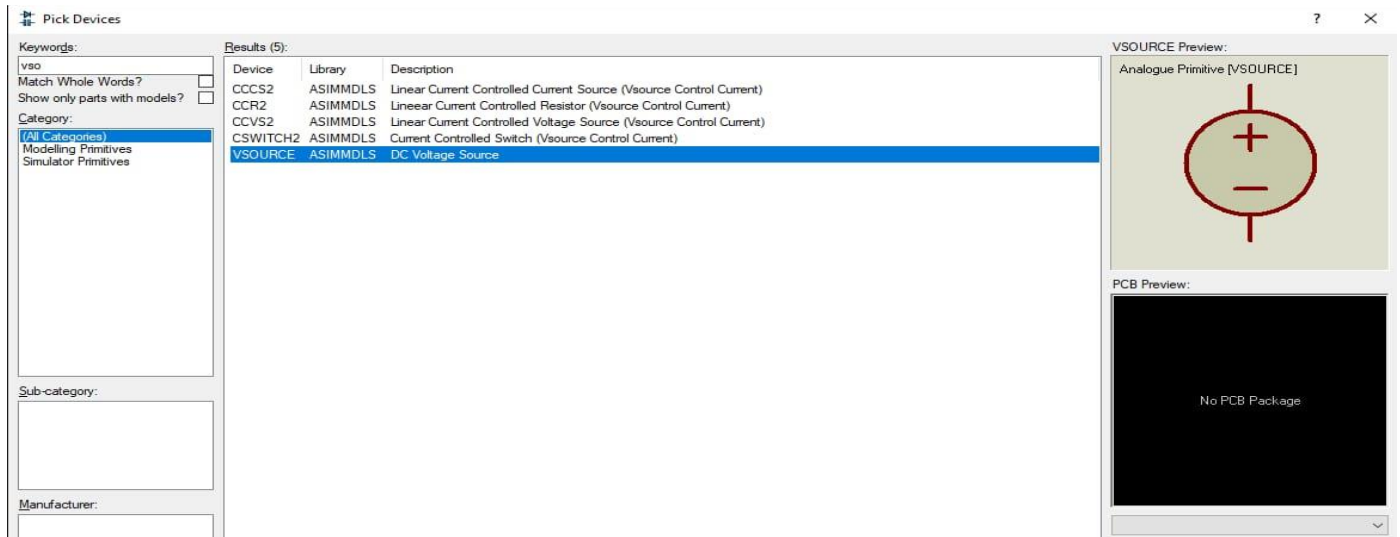




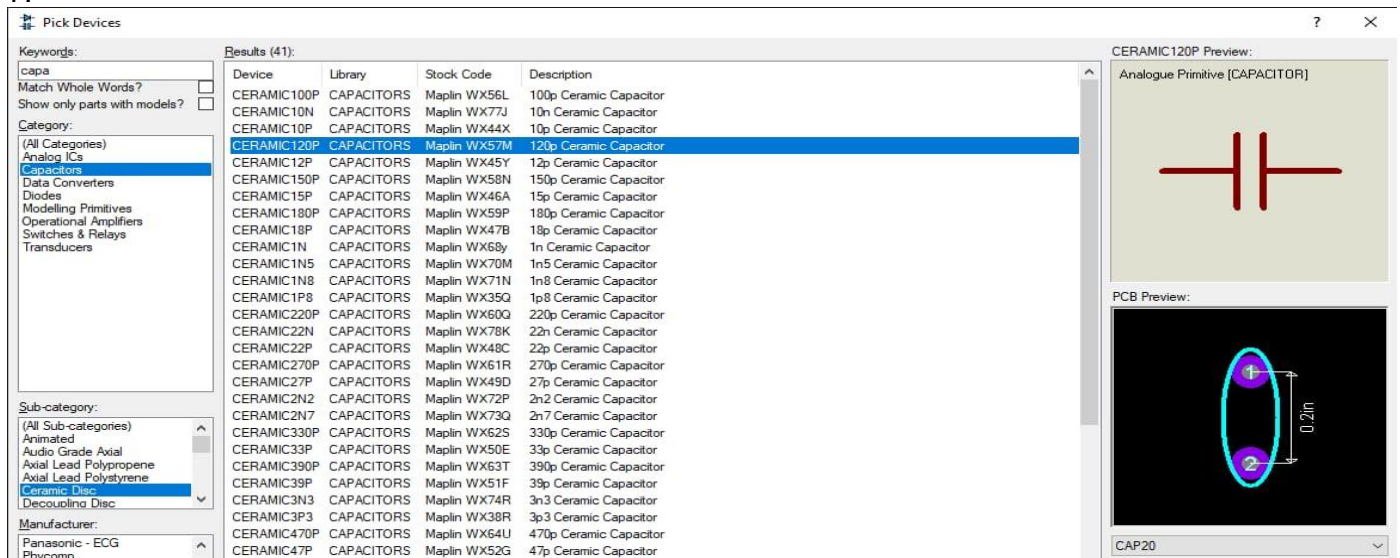
## 5.-



## 6.-



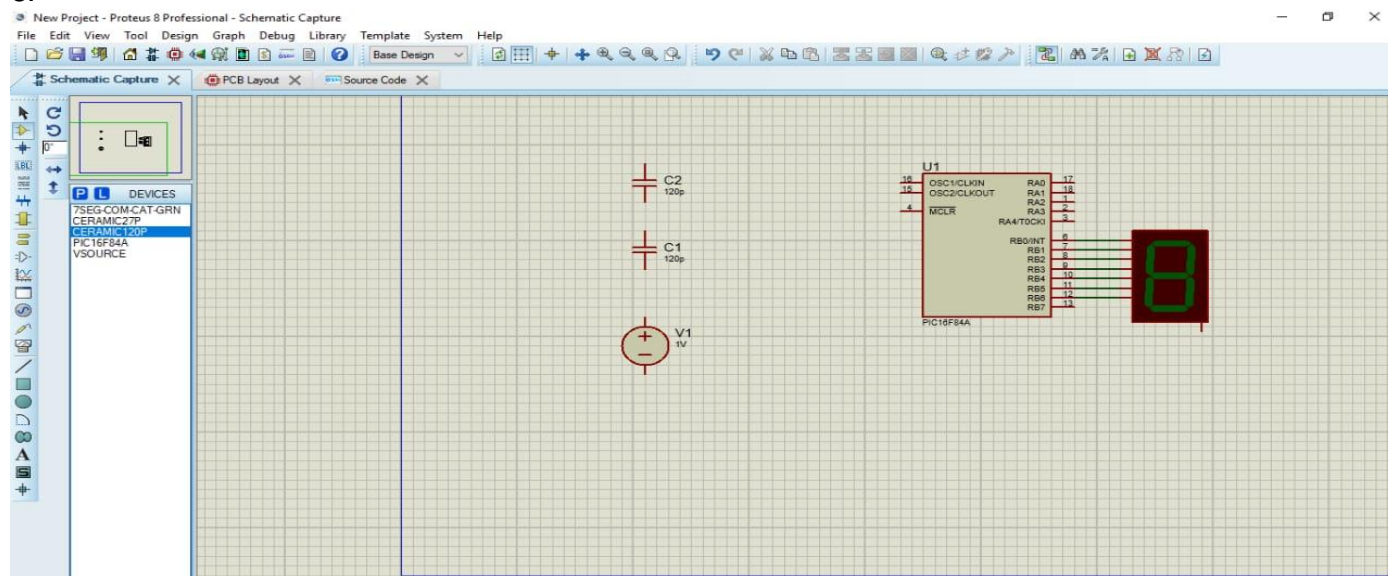
## 7.-



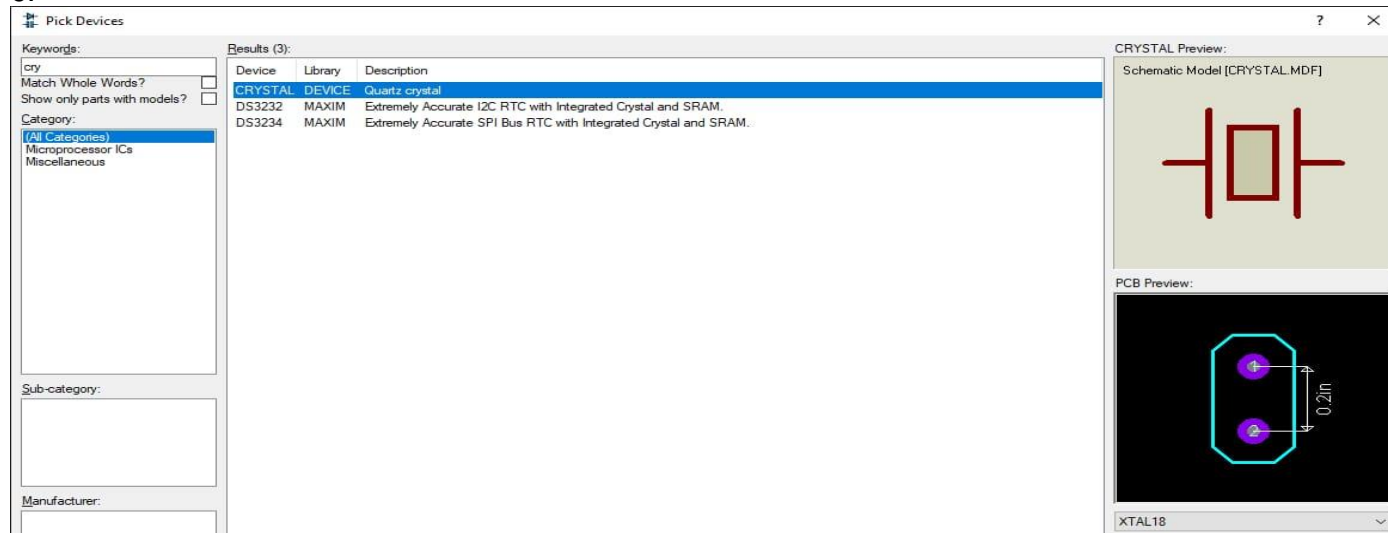




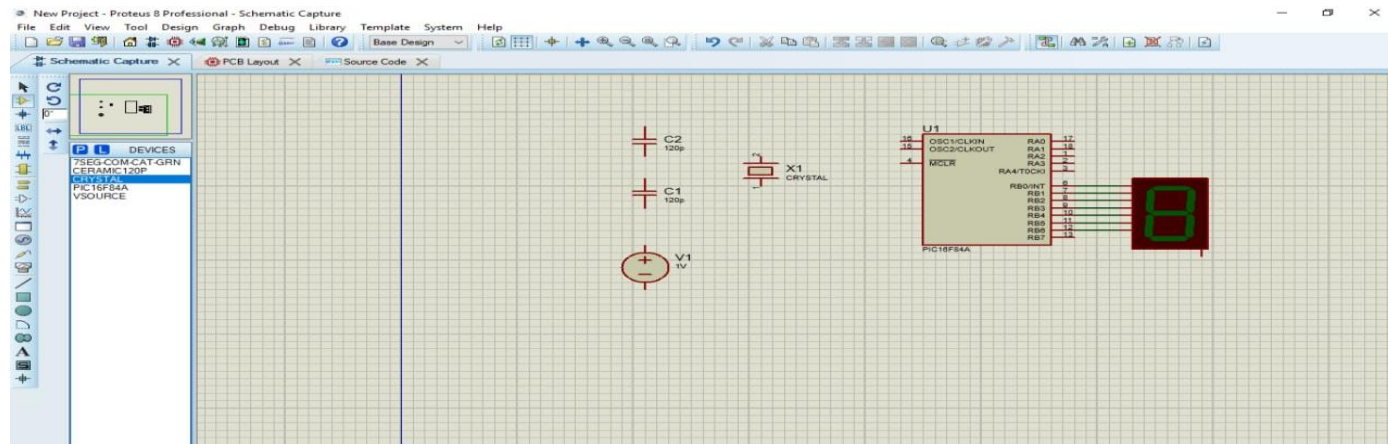
## 8.-



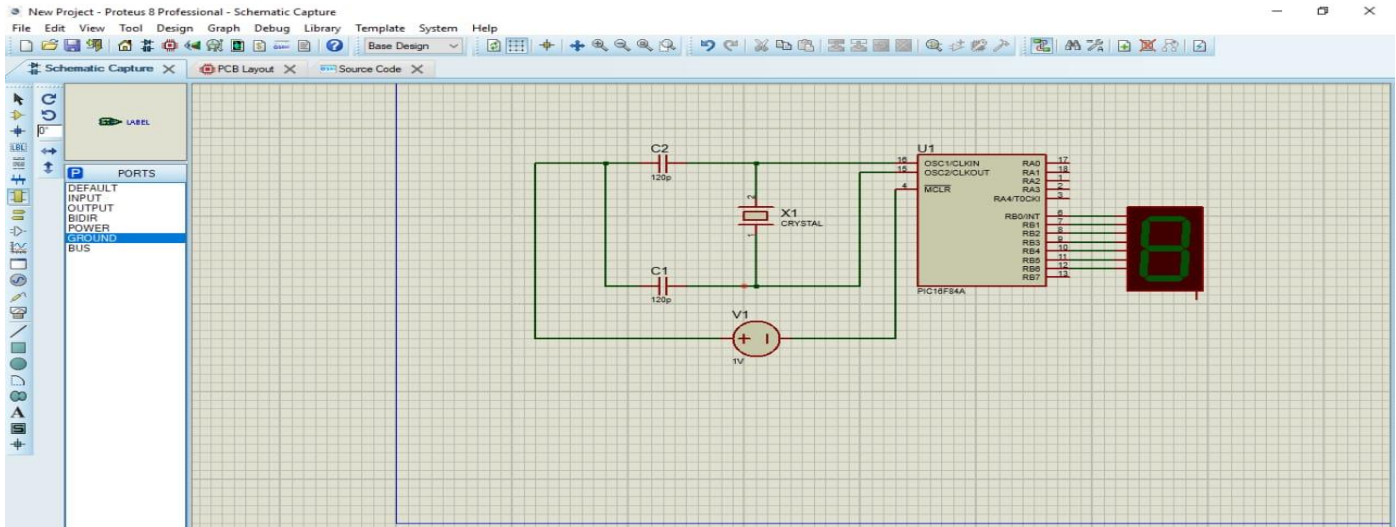
## 9.-



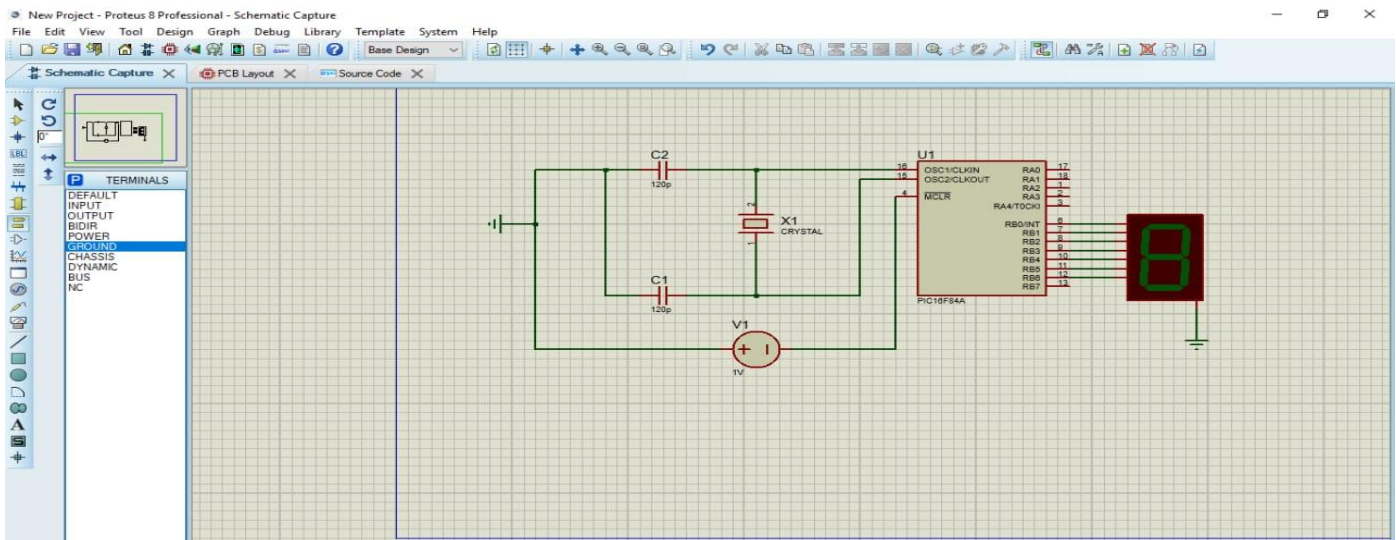
## 10.-



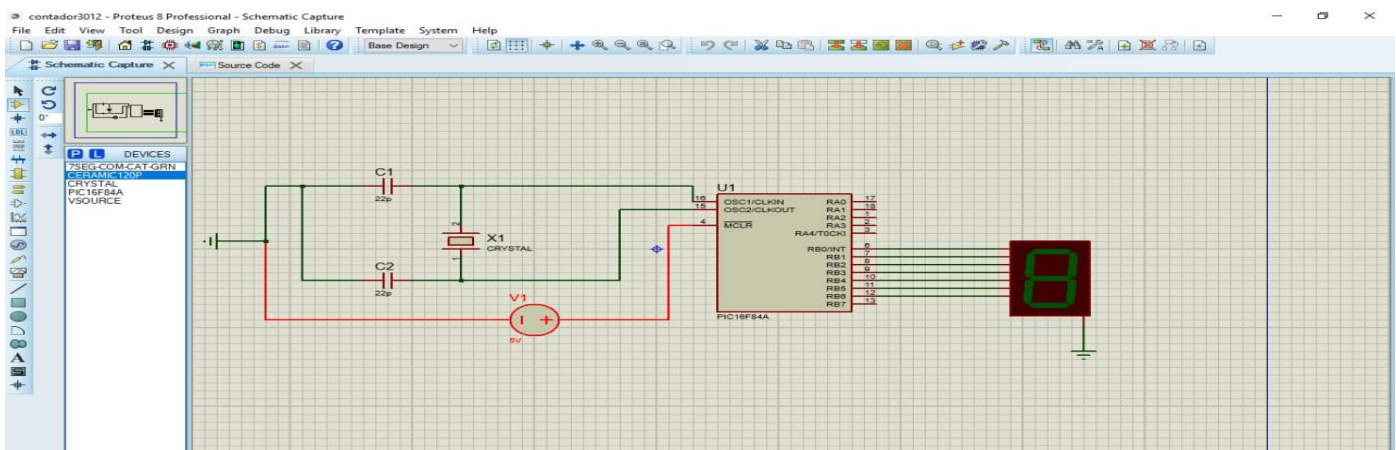
11.-



12.-

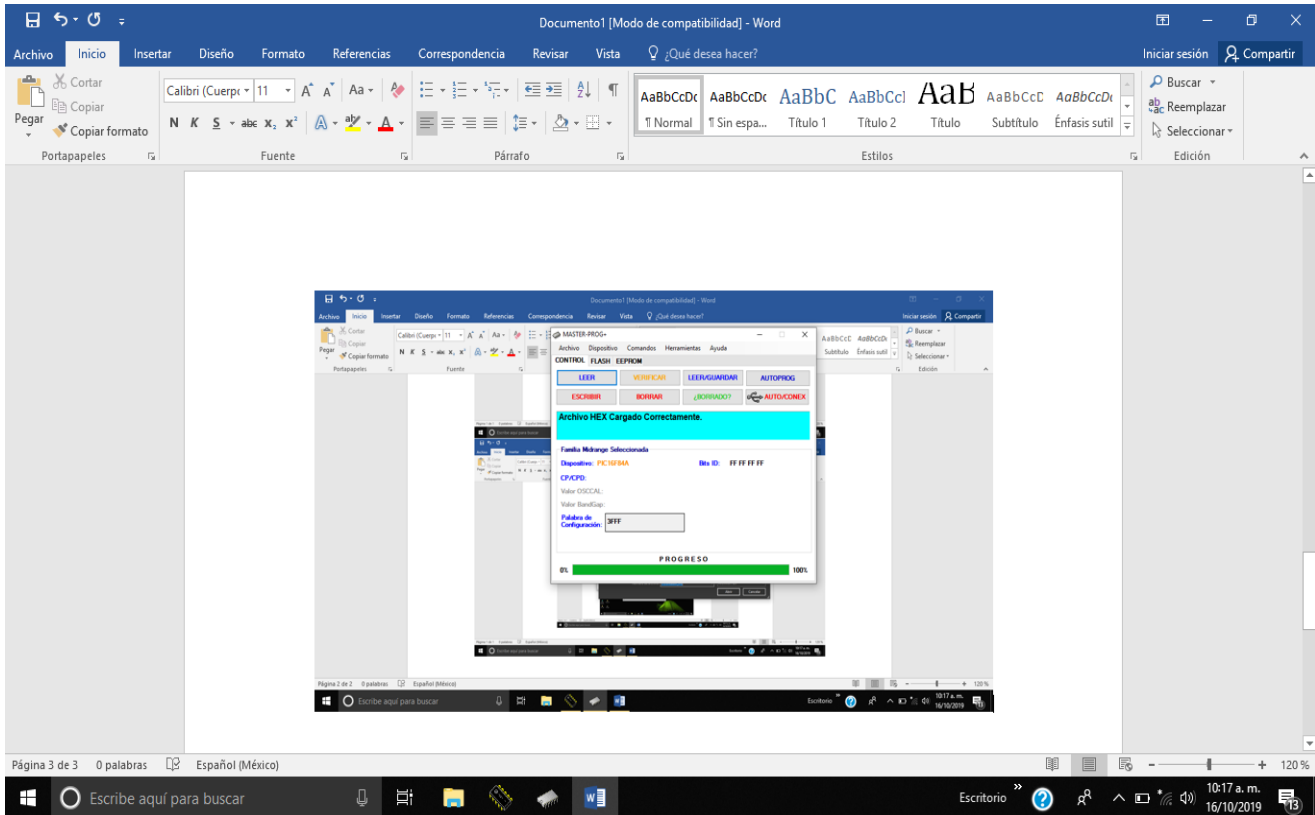


13.-

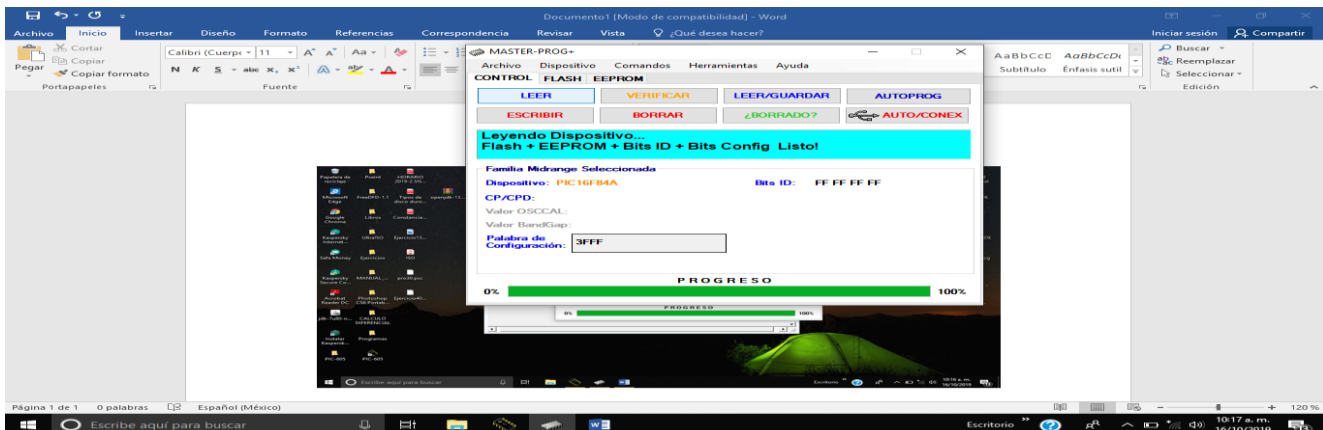


- Después de que terminamos de hacer el prototipo digital de la tabla Protoboar le metimos el código del Pic 16F84A
- Programa el Pic con el código compilado realizado en clase.

1.-

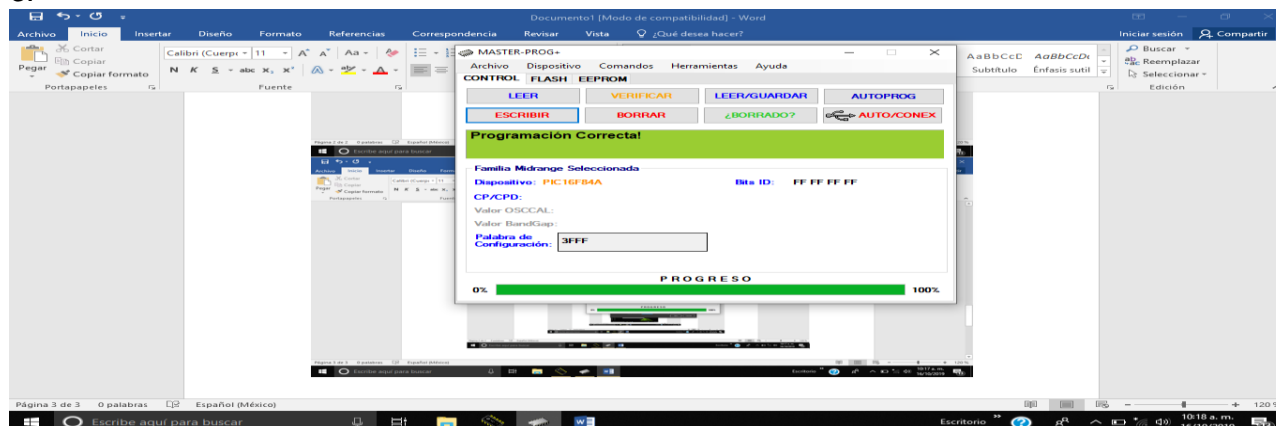


2.-

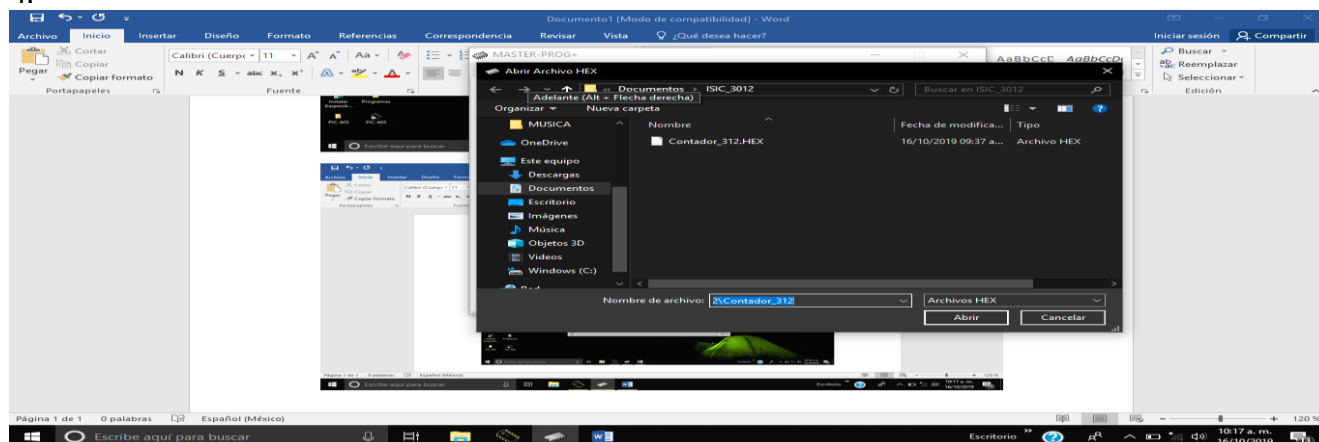




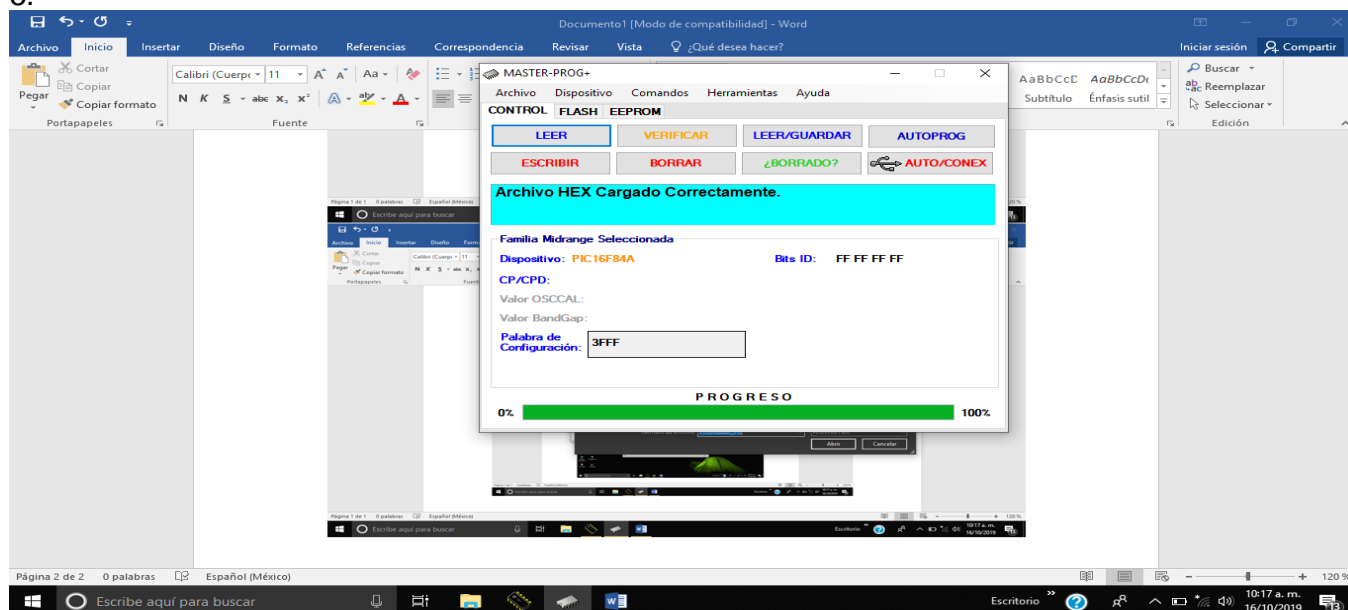
3.-



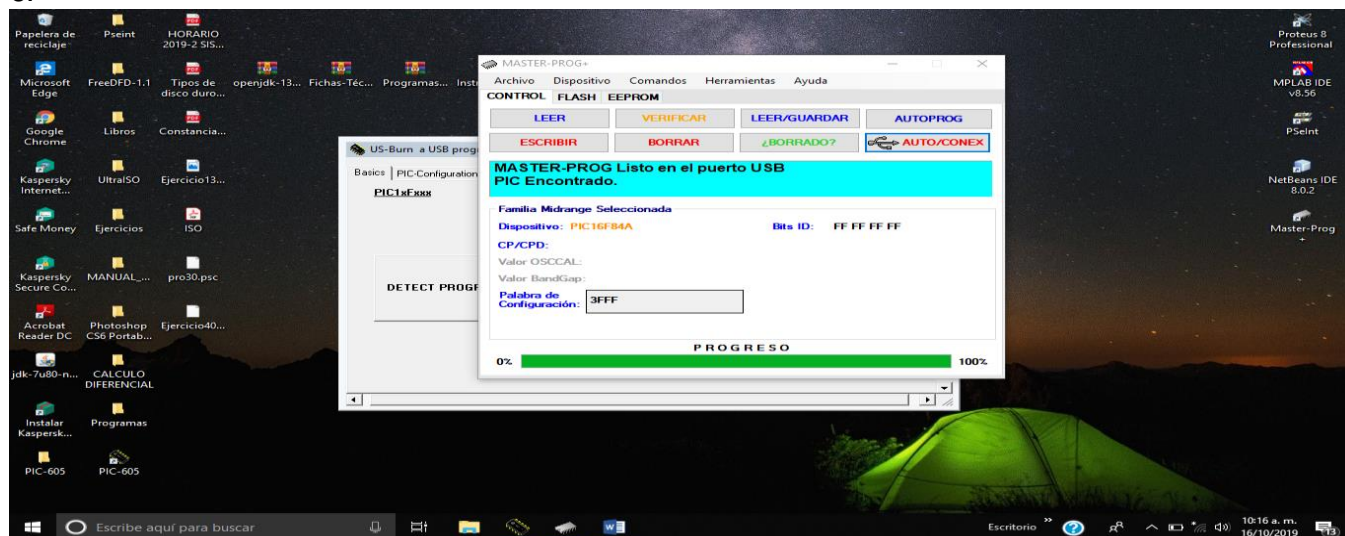
4.-



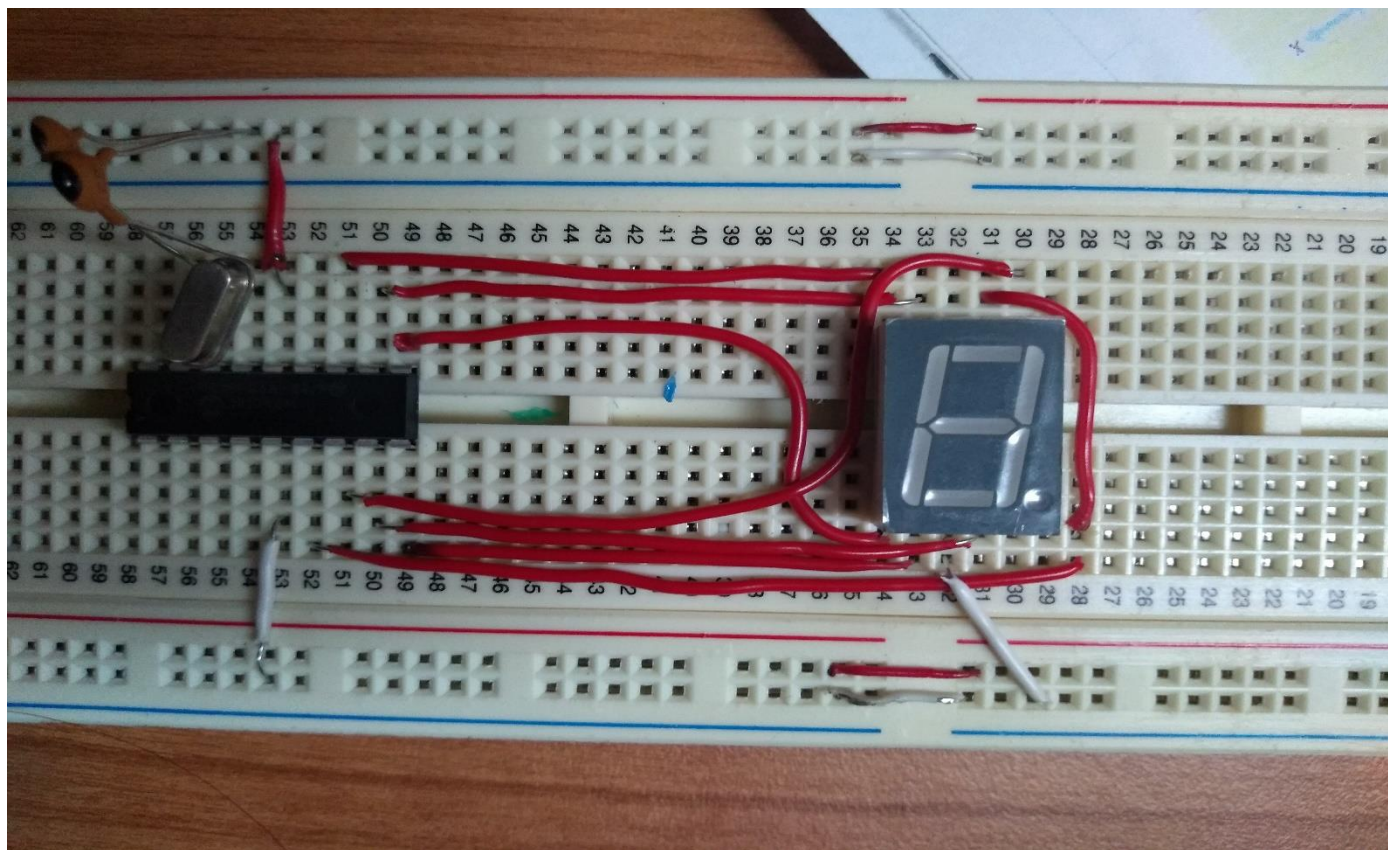
5.-



6.-

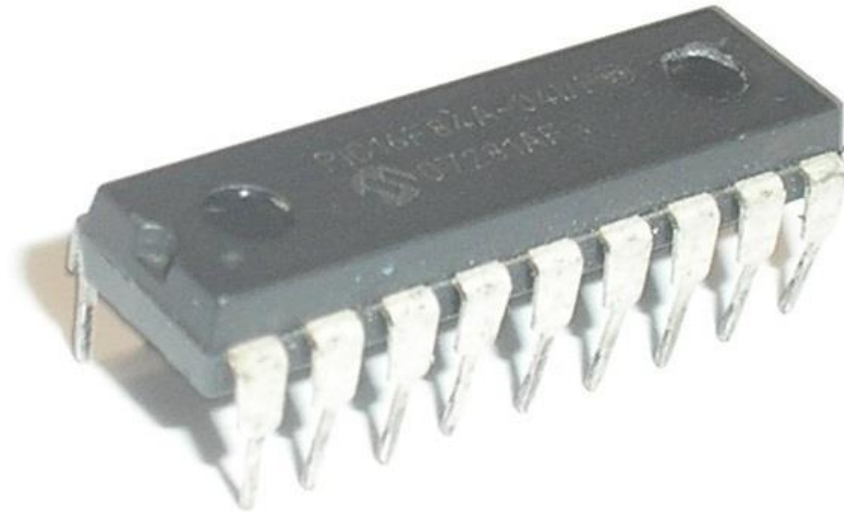


- Cuando ya hayas programado tu Pic 16F84A realiza tu tabla Portoboar ya con las conexiones, como lo avías realizado en el programa PROTEUS DESIGN SUITE 8.7





# MICROCONTROLADOR PIC 16F84A



El PIC16F84A es un microcontrolador gama media de 8 bits que físicamente consta de 18 pines, su memoria de programa es de tipo *flash*, lo que nos permitirá grabarlo hasta unas 10000 veces. Es realmente sencillo de manejar, por eso será nuestro punto de partida para el estudio del lenguaje ensamblador y microcontroladores.

Sus características principales son:

- Set de solo 35 instrucciones
- Memoria de programa de 1 KB (equivale a 1024 instrucciones)
- Máxima velocidad de operación: 20 MHz
- 68 Bytes de RAM
- 64 Bytes de EEPROM
- 4 fuentes de interrupción
- 2 puertos de salida
- 13 Líneas de I/O configurables individualmente
- 25 mA de corriente por pin

## Descripción de los pines

El PIC 16F84A contiene 13 líneas I/O (entrada o salida), 5 pertenecen al puerto A y 8 pertenecen al puerto B.

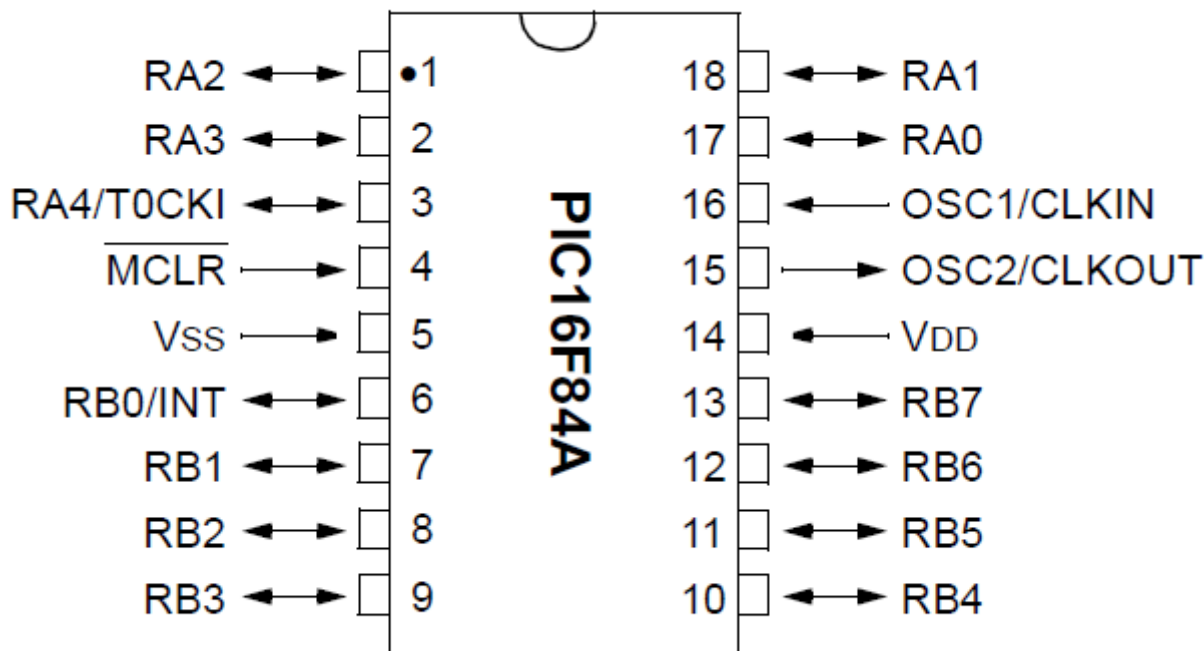


fig 2.2

Distribución de pines del PIC 16F84A

- **VDD:** Alimentación positiva (5 voltios)
- **VSS:** Tierra (0 voltios)
- **MCLR:** pin de resetear
- **OSC1, OSC2:** Conexión de oscilador
- **RA0, RA1, RA2, RA3, RA4:** líneas I/O del puerto A
- **RB0, RB1, RB2, RB3, RB4, RB5, RB6, RB7:** líneas I/O del puerto B

### Conexión de los pines

Es importante tener muy claro la forma en la que se deben conectar los pines del microcontrolador aún cuando estos no se usen.

1. **La alimentación:** El voltaje de alimentación debe estar comprendido entre 2.0 y 5.5 voltios (preferiblemente usar una alimentación regulada de 5 voltios). Para mejorar su desempeño se puede adicionar un capacitor cerámico de 100 nF en paralelo con la fuente de voltaje y ubicado lo más cerca posible a los pines VDD y VSS.

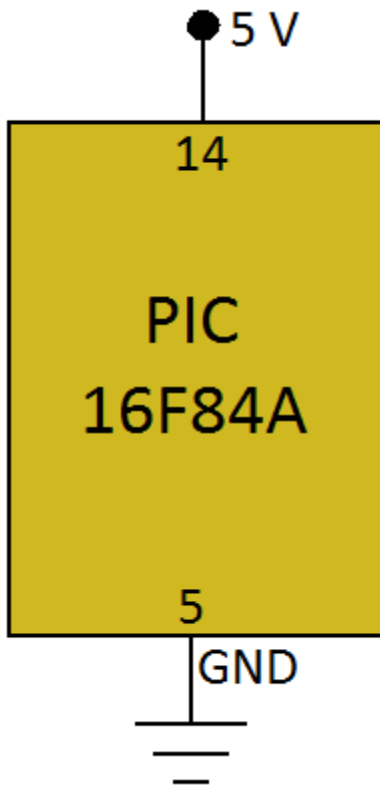


fig 2.3 Conexión de los pines de alimentación del PIC 16F84A

2. **El reset (MCLR):** A este pin se le conectarán dos componentes: un resistor de 10 k $\Omega$  a VDD (5 voltios) y un pulsador a GND (tierra).

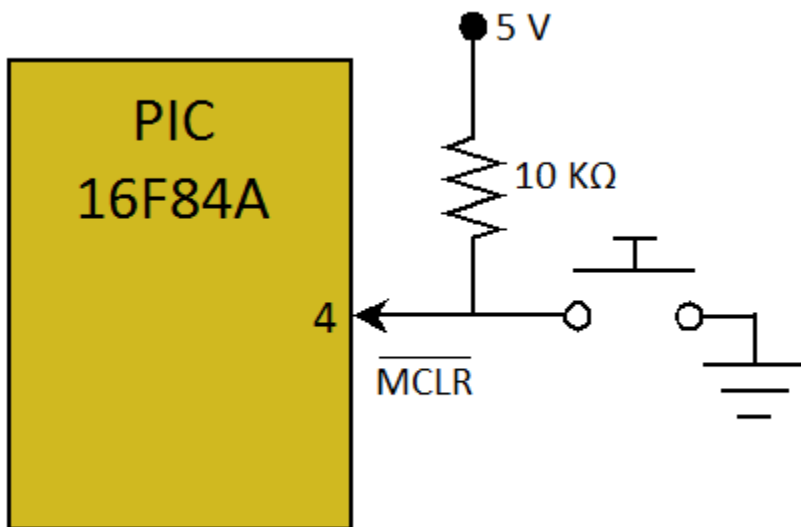


fig 2.4 Configuración del pin RESET



3. **El oscilador:** Por ahora, conectaremos un cristal de cuarzo de 4 MHz entre los pines OSC1 y OSC2, también conectaremos un capacitor cerámico de 22 pF entre OSC1 y tierra y otro capacitor igual entre OSC2 y tierra. Más adelante se detallarán las diversas formas de configurar estos pines.

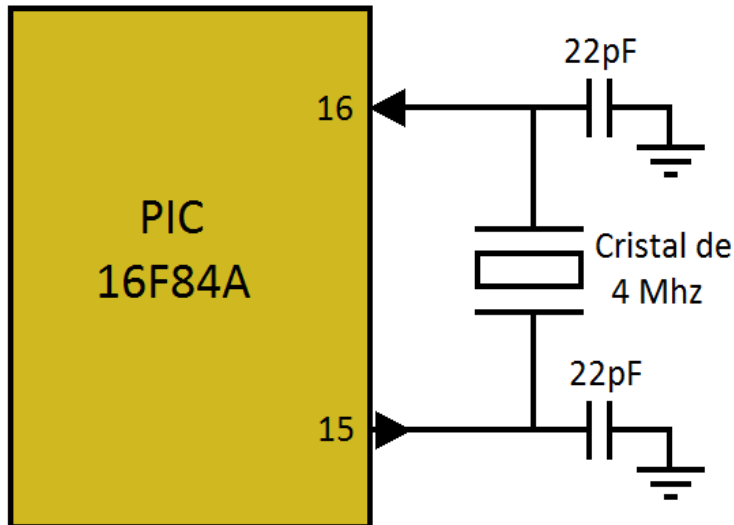


fig 2.5 Configuración del oscilador de

4MHz

4. **Las líneas de salida:** Si lo que queremos es solo testear el comportamiento del microcontrolador, basta con poner en cada salida un LED en serie con una resistencia.

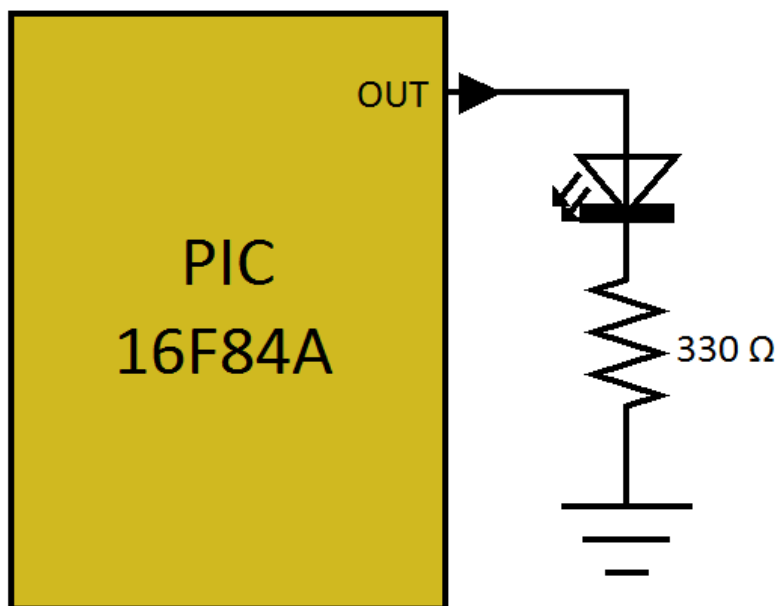
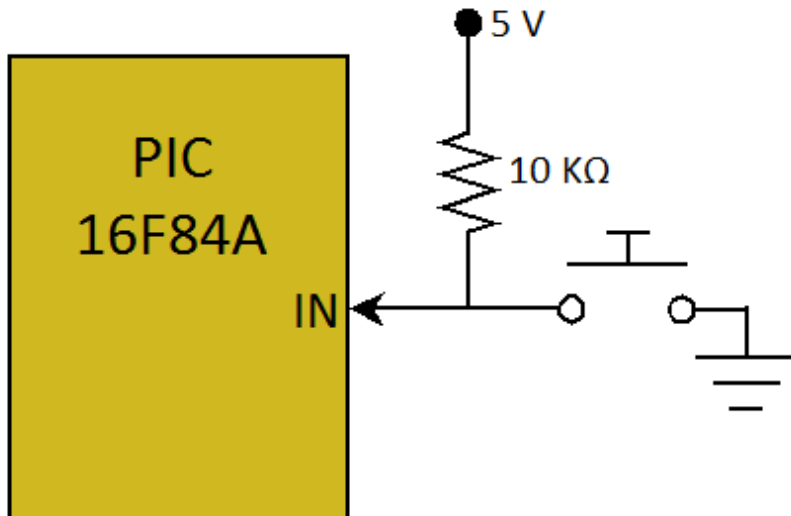


fig 2.6 Conexión de un LED en la salida

del PIC

5. **Las líneas de entrada:** Si la entrada se da a través de un interruptor será obligatorio conectar un resistor de 10 k $\Omega$  a VDD sin importar si la línea se usará o no, una solución sencilla para evitar la conexión de tantos resistores es configurar como salidas las líneas que no vayamos a usar en nuestro proyecto.



### Las resistencias de pull-up

La electrónica digital maneja dos estados lógicos para su funcionamiento: el «1» y el «0». Supongamos que un circuito digital tiene una entrada y conectamos un interruptor entre dicha entrada y VSS (tierra), analizando veremos que el cero ya está definido por el interruptor cuando este está cerrado, pero si el interruptor estuviera abierto no hay un estado definido porque la entrada no tiene contacto con ningún cero ni con ningún uno lógico (sería un pin sin conexión), esto se llama estado flotante y puede tener consecuencias como el mal funcionamiento del circuito. Esto se soluciona poniendo una resistencia entre el pin de entrada y VDD llamada resistencia de **pull-up**, su objetivo es asegurar un «1» lógico cuando el interruptor esté abierto. Si lo que se desea es asegurar un «0» lógico, la resistencia se conecta a VSS y recibe el nombre de resistencia de **pull-down**.

**Nota:** Se aconseja conectar cualquier línea de entrada a VDD mediante una resistencia de 10 k $\Omega$ , omitir este paso podría causar daños en el PIC.