

UNIVERSIDAD DE SONSONATE

Arquitectura de Computadoras

Guía 3 – LCD

Instructor: Ricardo González



Facultad de Ingeniería y Ciencias Naturales



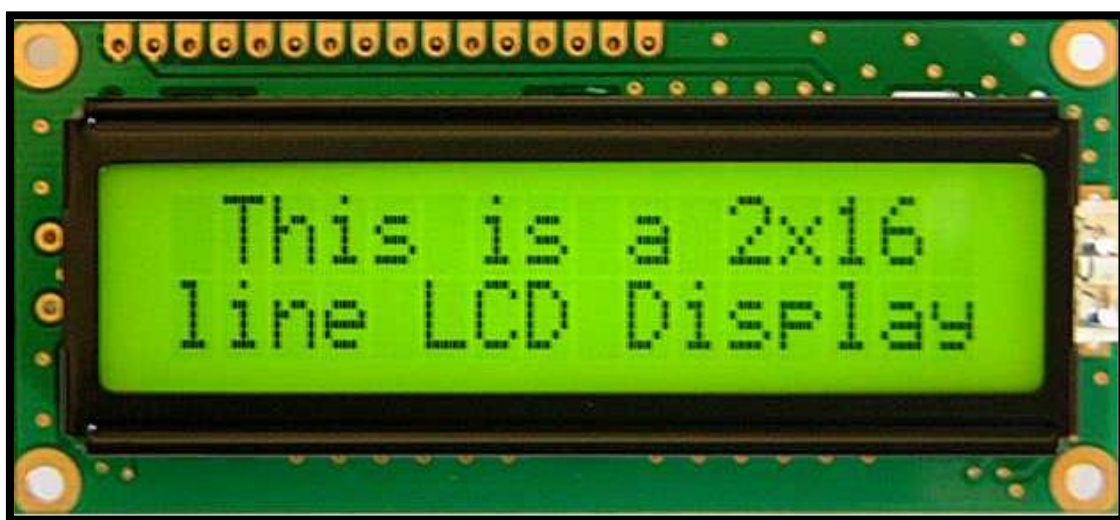
Contenido

¿Qué es un LCD?	3
Programación en C.....	4
Archivos de Cabecera.....	4
EJERCICIOS.....	10

¿Qué es un LCD?

El LCD en sus siglas en ingles **Liquid Crystal Display (Pantalla de Cristal Líquido)** es un dispositivo muy versátil que nos permite mostrar datos, configuraciones o desplegar determinada información donde el LED o un display de 7 segmentos no es suficiente.

Cuando trabajamos con LCD's, se acostumbra a utilizar un LCD con un driver de tipo **HD44780**, con un número de líneas variable y un número de caracteres por línea también variables (por ejemplo podemos obtener una configuración 2 x16, que trabaja con 2 líneas de 16 caracteres cada una). A continuación se muestra la imagen de un LCD de configuración 2 x 16:



Cuando se manipulan dichos dispositivos es muy importante verificar la hoja del fabricante (**DataSheet**), ya que existe distintas configuraciones para este tipo de dispositivo un ejemplo de ello podría ser configuraciones tales como: 1 x 16, 2 X 8, 1 x30, etc. Para un LCD con driver de tipo **HD44780**, la configuración de pines seria la siguiente:

Pin no.	Symbol	Function
1	GND	Power supply ground
2	VCC	+5V supply
3	VEE	Contrast adjustment voltage
4	RS	Register select (H: data, L: instruction)
5	R/W	Read/Write data (H: LCD -> μ C, L: μ C -> LCD)
6	E	Enable pulse
7	D0	Data bit 0
8	D1	Data bit 1
9	D2	Data bit 2
10	D3	Data bit 3
11	D4	Data bit 4
12	D5	Data bit 5
13	D6	Data bit 6
14	D7	Data bit 7
15	A	Anode of backlight LED
16	K	Cathode of backlight LED



La conexión que se establece entre el microcontrolador y el LCD puede ser de 8 bits o de 4 bits, la diferencia entre ellos está en cómo se envía los datos o comandos al LCD. Cuando se envía los datos a través del modo de 8bits, se envían 8 bits de datos y los comandos se envían a través de las líneas D0 a la D7; mientras tanto cuando utilizamos el envío de datos utilizando 4bits se envía de forma secuencial a través de la línea de datos D4 a la D7. Al Utilizar solamente 4bits en nivel de procesamiento de la información es más lento que una configuración de 8bits, como los LCD son dispositivos de baja velocidad la diferencia entre estos dos modos no es significativa, ya que recordemos que el microcontrolador trabaja a una velocidad de Mhz y estamos visualizando el LCD con nuestros ojos, la persistencia de nuestro ojos no nota la diferencia entre ellos.

Para poder manipular las configuraciones de 8bits y de 4bits sería necesario conocer completamente como trabaja el LCD para escribir un programa para nuestro PIC. Para simplificar dicho trabajo utilizaremos un archivo de cabecera que incluye el funcionamiento de una configuración de 4bits creado por el sitio: <http://electrosome.com/>. Pero antes recordaremos que son los archivos de cabecera.

Programación en C

Cuando trabajemos firmware con nuestro PIC16F84A, en muchas ocasiones no será de utilidad separar los métodos y funciones para mantener limpio y ordenado nuestro código de nuestro archivo principal, utilizaremos los archivos de cabecera.

Archivos de Cabecera.

Un header file (".h") contiene, normalmente, una declaración directa de clases, subrutinas, variables, u otros identificadores. Cuando deseamos declarar identificadores estándares en más de un archivo fuente pueden colocar esos identificadores en un único header file, que se incluirá cuando el código que contiene sea requerido por otros archivos.

En la mayoría de lenguajes de programación modernos, podemos dividir los programas en componentes de menor tamaño (como pueden ser clases y subrutinas) y distribuir esos componentes entre muchas unidades por traducir (típicamente en forma de archivos), que el sistema puede compilar de forma autónoma. Podemos definir una función en un archivo de cabecera y utilizarlo en el archivo principal como en el siguiente ejemplo:

Archivo1.h

```
int suma(int val1, val2){  
    return val1+val2;  
}
```

Main.c

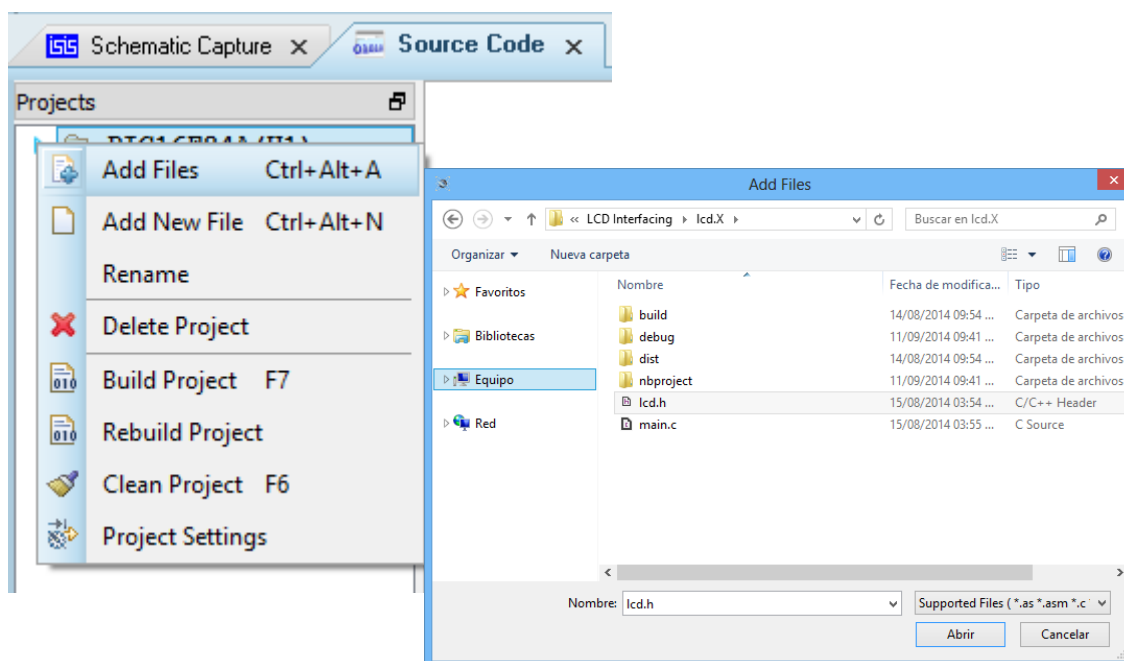
```
#include "Archivo1.h";  
void main(void){  
    suma(1,2);  
}
```



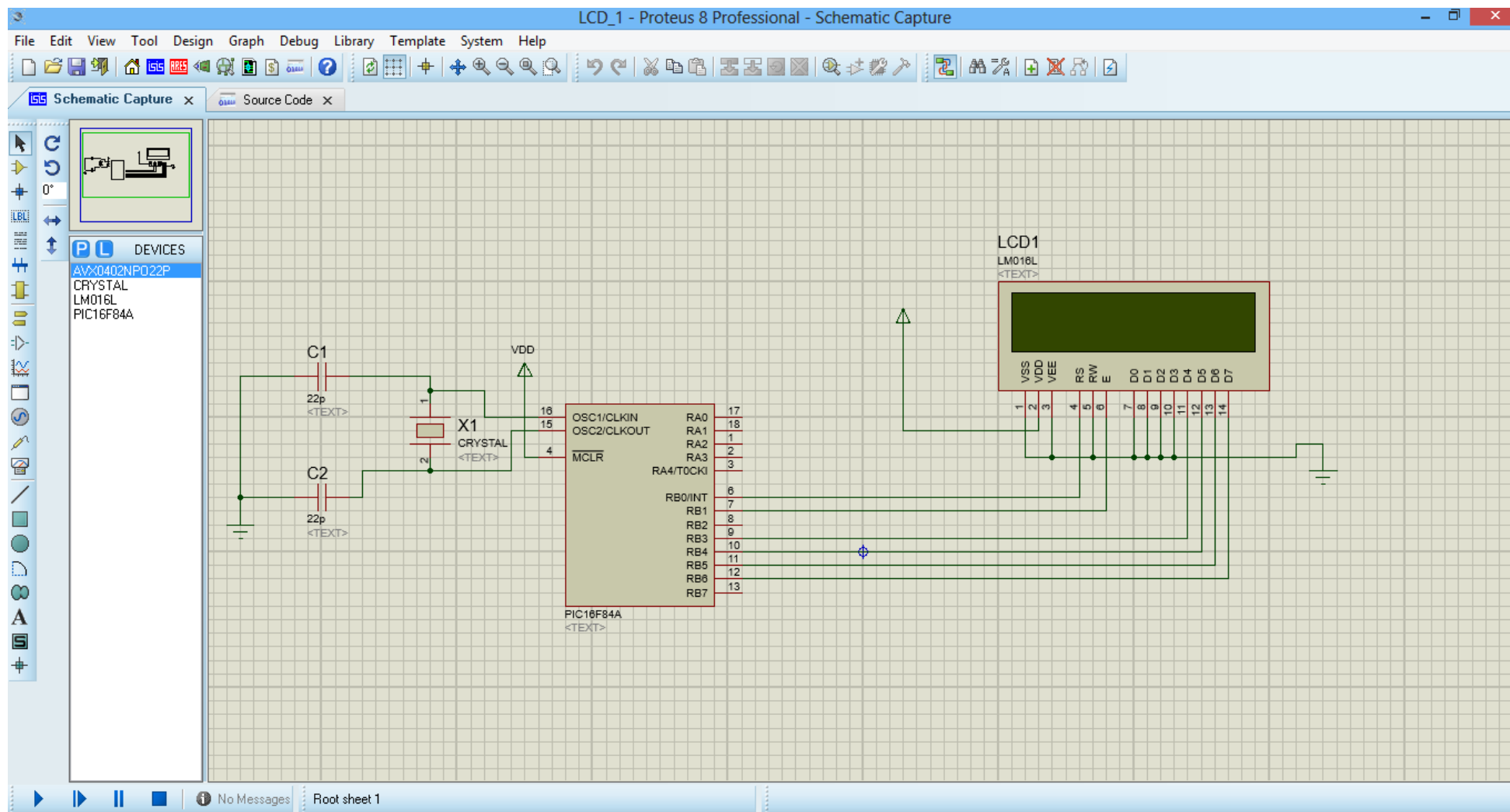
Ejemplo: Para nuestro primer ejemplo utilizando el dispositivo LCD, crearemos un circuito un circuito básico para la manipulación de dicho dispositivo y mostraremos en pantalla el clásico “*Hola Mundo*”, para ello necesitamos los siguientes componentes:

- ✓ 1- PIC16F84A.
- ✓ 1- Cristal de Cuarzo configurado a 4Mhz: **CRYSTAL**.
- ✓ 2- Capacitores: **AVX0402NPO22P**.
- ✓ 1- Display LCD: **LM016L**.

Cuando creemos nuestro proyecto agregamos el archivo de cabecera que nos permitirá manipular nuestro display LCD, de una forma más amigable, para ello cuando tengamos cargado nuestro proyecto nos trasladamos a la pestaña “**source**”, y en la carpeta donde aparece el nombre de nuestro PIC (PIC16F84A), daremos click derecho y seleccionaremos la opción “**Add Files**”, luego se los desplegará una venta para que busquemos nuestro archivo a incluir en este caso será el archivo “**lcd.h**”.



El esquema de nuestro se muestra en la siguiente página, del cual se hace uso de los componentes que se mencionó en la lista anterior.



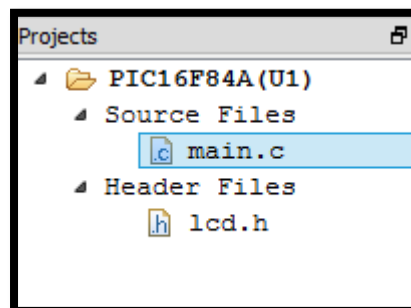
Para codificar nuestro primer ejemplo de **“Hola Mundo”**, colocaremos lo siguiente en nuestro firmware:

```

1  //DEFINIMOS LA FRECUENCIA
2  #define _XTAL_FREQ 4000000
3  //DEFINIMOS LOS PUERTOS
4  //=====
5  #define RS RB0
6  #define EN RB1
7  #define D4 RB3
8  #define D5 RB4
9  #define D6 RB5
10 #define D7 RB6
11 //=====
12
13 #include <xc.h>
14 #include "lcd.h";
15
16
17 void main(void)
18 {
19     //Configuramos los puertos
20     TRISB = 0; //Como Salida
21     Lcd_Init(); //Inicializamos la pantalla
22     while (1){
23         Lcd_Clear(); //Limpiamos la pantalla
24         Lcd_Set_Cursor(1,1); //Establecemos el cursor en la 1 fila y 1 primera columna
25         Lcd_Write_String("Hola Mundo"); //Escribimos el mensaje en la pantalla
26         __delay_ms(40000);
27     };
28 }

```

Hay que recordar que para poder utilizar los métodos **Lcd_Clear()**, **Lcd_Set_Cursor()** y **Lc_Write_String()**, es necesario haber incluido el archivo de cabecera **“lcd.h”**.



Algunos de los métodos que encontraremos definidas en este archivo son las siguientes:

Método	Descripción
Lcd_Clear()	<i>Limpiamos el display LCD.</i>
Lcd_Set_Cursor(int row, int column)	<i>Método utilizado para establecer la posición en la cual se colocara el cursor para mostrar un texto.</i>
Lcd_Write_Char(char)	<i>Escribe un carácter en la posición actual del cursor.</i>
Lcd_Write_String(char *String)	<i>Escribe una cadena de caracteres en la posición actual del cursor.</i>
Lcd_Shift_Right()	<i>Permite desplazar un carácter a la derecha el texto mostrado en el display sin borrar la memoria del LCD.</i>
Lcd_Shift_Left()	<i>Permite desplazar un carácter a la izquierda</i>



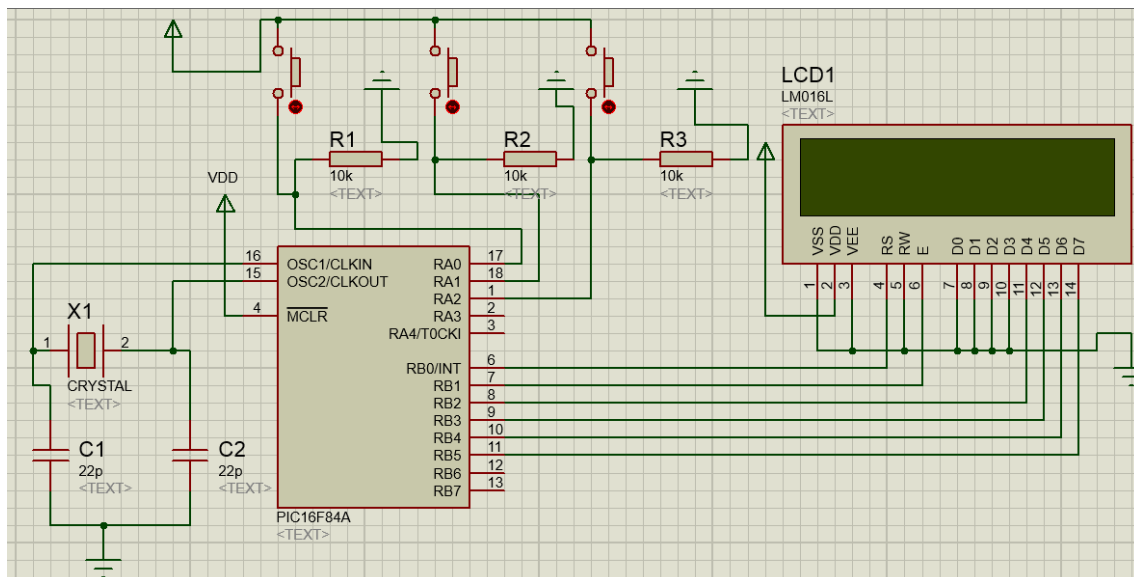
el texto mostrado en el display sin borrar la memoria del LCD.

Como se mencionaba en los apartados anteriores la idea de trabajar con 4bits, es para poder tener libre más pines de nuestro PIC, a continuación veremos un ejemplo aprovechando esta ventaja que ofrece trabajar con 4bits.

Ejemplo: Para nuestro ejemplo trabajaremos en el circuito base que se mostró en el primer ejemplo, la diferencia que este circuito poseerá un firmware que nos permitirá seleccionar la carrera a la cual pertenecemos y mostrar el resultado de la selección en el display LCD. Para este ejemplo solo necesitaremos de los siguientes componentes extras:

- ✓ 3 Botones: **Button**.
- ✓ 3 Resistencia de 10k: **RESISTOR**.

El circuito nos deberá quedar de la siguiente manera:



El código para nuestro firmware es el que se muestra en la siguiente página, recordemos que para utilizar las funciones del LCD es necesario haber incluido el archivo de cabecera **"lcd.h"**.



```
1 //DEFINIMOS LA FRECUENCIA
2 #define _XTAL_FREQ 4000000
3 //DEFINIMOS LOS PUERTOS
4 //=====
5 #define RS RB0
6 #define EN RB1
7 #define D4 RB2
8 #define D5 RB3
9 #define D6 RB4
10 #define D7 RB5
11 //=====
12 #include <xc.h>
13 #include "lcd.h";
14
15 //Definicion de prototipos de funciones
16 unsigned int carrera();
17
18 void main(void)
19 {
20     //Configuramos los puertos
21     TRISA = 1; //Configuramos el puerto como entrada
22     TRISB = 0; //Configuramos el puerto como salida
23     PORTA = 0; //Inicializamos el puerto
24     Lcd_Init();
25     Lcd_Clear();
26     Lcd_Set_Cursor(1,1);
27     Lcd_Write_String("Tu Carrera?");
28     __delay_ms(300);
29     while (1){
30         if(carrera()){
31             Lcd_Clear();
32             Lcd_Clear();
33             Lcd_Set_Cursor(1,1);
34             Lcd_Write_String("Tu Carrera?");
35             __delay_ms(300);
36         }
37     };
38 }
39
40 unsigned int carrera(){
41     int flag = 0;
42     if(PORTAbits.RA0){
43         Lcd_Clear();
44         Lcd_Set_Cursor(1,1);
45         Lcd_Write_String("Su carrera es");
46         Lcd_Set_Cursor(2,1);
47         Lcd_Write_String("Ingenieria en Sistemas");
48         //Realizamos el desplazamiento a la Izquierda
49         for(int i= 0;i<10; i++){
50             __delay_ms(300);
51             Lcd_Shift_Left();
52         }
53         flag = 1;
54     }
55     if(PORTAbits.RA1){
56         Lcd_Clear();
57         Lcd_Set_Cursor(1,1);
58         Lcd_Write_String("Su carrera es");
59         Lcd_Set_Cursor(2,1);
60         Lcd_Write_String("Ingenieria Electrica");
61         //Realizamos el desplazamiento a la Izquierda
62         for(int i= 0;i<10; i++){
63             __delay_ms(300);
64             Lcd_Shift_Left();
65         }
66         flag = 1;
67     }
68     if(PORTAbits.RA2){
69         Lcd_Clear();
70         Lcd_Set_Cursor(1,1);
71         Lcd_Write_String("Su carrera es");
72         Lcd_Set_Cursor(2,1);
73         Lcd_Write_String("Ingenieria Industrial");
74         //Realizamos el desplazamiento a la Izquierda
75         for(int i= 0;i<10; i++){
76             __delay_ms(300);
77             Lcd_Shift_Left();
78         }
79         flag = 1;
80     }
81     return flag;
82 }
83 }
```



EJERCICIOS.

Para cada uno de los ítems se tiene que realizar en distintos proyectos utilizando el PIC16F84A.

- 1) Realice la configuración de un circuito utilizando la pantalla LCD, que permita visualizar las veces que se ha presionado un botón, para llevar a cabo este ejercicio agregamos 3 botones al circuito y en la pantalla LCD se debe de mostrar cuantas veces se ha presionado determinado botón. Por ejemplo:
 - a. Si presiono el botón que se encuentra configurado en el PIN0 del PUERTO A, se debe de visualizar en la pantalla: **PUERTO A-PINO: 1.**
- 2) Realizar el ejercicio de la guía 2 de la calculadora solo que en este caso en vez de utilizar un display de 7 segmentos utilizaremos el display LCD para mostrar la información.