#include <LiquidCrystal\_I2C.h>

LiquidCrystal\_I2C lcd (0x27, 20, 4);

const int Trigger = 8;   //Pin digital 8 para el Trigger del sensor

const int Echo = 9;   //Pin digital 9 para el Echo del sensor

void setup() {

  lcd.init();

  lcd.backlight();

  pinMode(Trigger, OUTPUT); //pin como salida

  pinMode(Echo, INPUT);  //pin como entrada

  digitalWrite(Trigger, LOW);//Inicializamos el pin con 0 logico

}

void loop()

{

  lcd.setCursor(0,0);

  long t; //timepo que demora en llegar el eco

  long d; //distancia en centímetros

  digitalWrite(Trigger, HIGH);

  delayMicroseconds(10);          //Enviamos un pulso de 10us

  digitalWrite(Trigger, LOW);

  t = pulseIn(Echo, HIGH); //obtenemos el ancho del pulso

  d = t/59;             //escalamos el tiempo a una distancia en cm

  lcd.print("Distancia: ");

  lcd.print(d);      //Enviamos serialmente el valor de la distancia

  lcd.print("cm  ");

  delay(100);          //Hacemos una pausa de 100ms

}

Se declara como siempre la librería LCD para el objeto y poder imprimir la distancia en este caso.

Se definen los pines para el echo y el trigger del sensor ultrasónico.  
En el setup se inicializa todo, la lcd, entra y salida de datos por los pines correspondientes (ECHO entrada y TRIGGER salida).

En el setup se necesita inicializar el cursor siempre en 0,0 para que la lcd siempre imprima en la misma posición sin tener que limpiar la pantalla, se declaran variables para la distancia y el tiempo, y se activa y desactiva el trigger con un intervalo de 10 segundos, después se lee el pulso del ECHO, el cual se almacena en la variable tiempo debido a que pulseIN(), te devulve el tiempo del estado que se solicito de un determinado pin , en este caso devuelve el tiempo en microsegundos de cuanto tiempo el pin ECHO estuvo en estado bajo o 0 logico.  
Después por medio de una formula se despeja distancia y tenemos que es igual a t/59, ya finalmente se imprime el valor obtenido en cm.