

PLANTAR CLAVELES CON ARDUINO

Objetivo

Tamagotchi vegetal, cuidar algo verdaderamente vivo.

Usaremos arduino, por pantalla nuestra planta nos pedirá cosas según lo que detecten nuestros sensores.

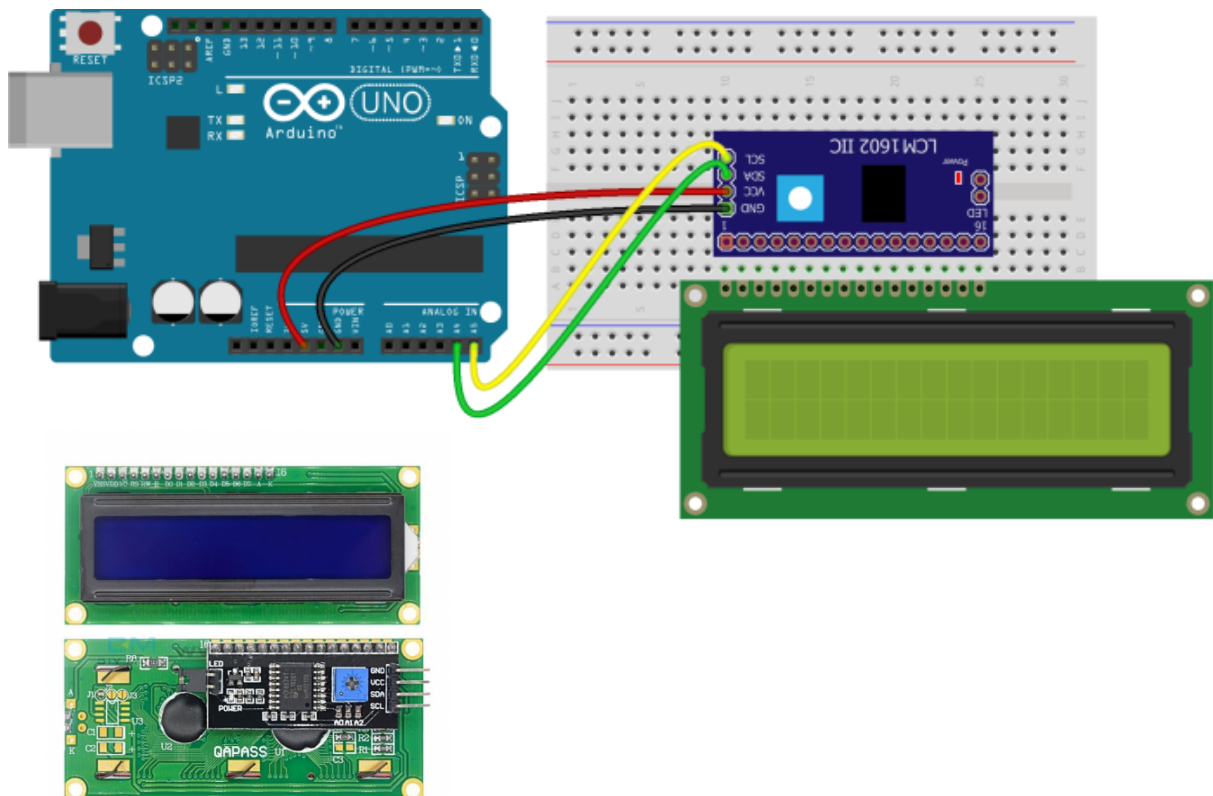
Usar:

- pantalla lcd
 - zumbador
 - detector de humedad
-

¿Cómo funciona la pantalla LCD?

Así se conecta.

Hay que tener en cuenta que la pantalla que estoy usando tiene incorporado un **adaptador I2C** para que la conexión sea más sencilla y así quedaría.



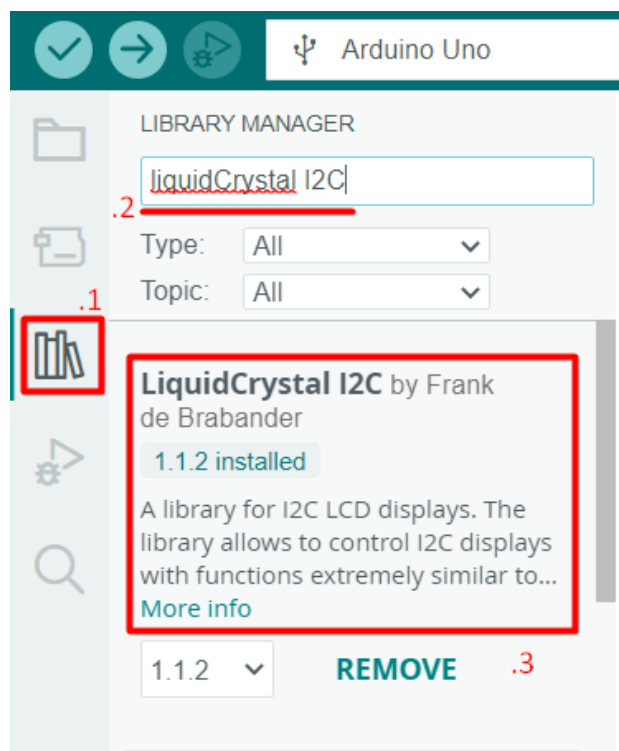
Por si queréis saber más acerca del adaptador:



(partes del adaptador) aquí lo explica mejor→
<https://www.youtube.com/watch?v=kuLgPLrg-cY>

PROGRAMACIÓN PANTALLA LCD

Necesitamos la aplicación de **ARDUINO IDE** lo primero de todo 😊
[Software | Arduino](#)



Dentro de la aplicación de arduino instalamos esta librería (**LiquidCrystal I2C**):

y nos ponemos a picar código:

Es sencillo:

- Al principio del código, llamamos a la librería `#include <LiquidCrystal_I2C.h>`
- creamos una pantalla:
`LiquidCrystal_I2C nombrepantalla(0x27, 16, 2);`
- en el setup iniciamos la pantalla con:
`lcd.init();`
`lcd.backlight();`
`lcd.clear();`
- para colocar el cursor donde queremos que empiece el texto a escribir:
`lcd.setCursor(x,y);`
- El código se lee de arriba abajo y de izquierda a derecha. Parece tontería pero **ES IMPORTANTE**

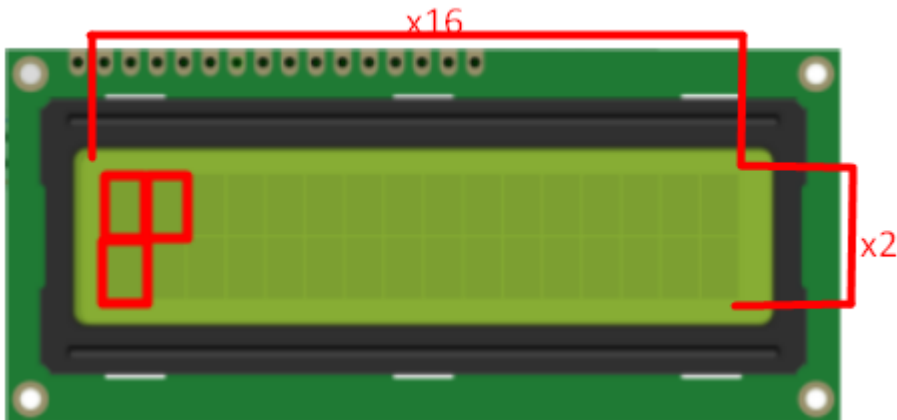
*

cuando nombramos la pantalla: `LiquidCrystal_I2C nombrepantalla(0x27, 16, 2);`
ese (0x27, 16, 2); significa:

el 0x27 es por lo que tengo entendido un protocolo de la pantalla, según la pantalla el protocolo puede cambiar entre:

- 0x27 <https://www.youtube.com/watch?v=Dbp66tmwBYQ> ← Este men explica bn 😊
- 0x20
- 0x30
- 0x3f

el 16, 2 son los caracteres que contiene la pantalla, mi pantalla sería una tabla de 16x2 pero puede cambiar según lo grande que sea la pantalla que usemos



EJEMPLO DE CÓDIGO:

```
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27, 16, 2); // Inicia el LCD en la dirección
0x27 con 16 caracteres y 2 líneas

void setup() {
    lcd.init();           //Iniciar pantalla
    lcd.backlight();      // Activar luz de fondo
    lcd.clear();          // Borrar todo texto en pantalla

    lcd.setCursor(3,0);    // coordenadas LCD (x,y)
    lcd.print("HOLA");     // Mensaje en fila de arriba
    lcd.setCursor(2,1);    // coordenadas LCD (x,y)
    lcd.print("MUNDO");    // Mensaje en fila de abajo
}

void loop() {
}
```

esto se vería:

			H	O	L	A									
		M	U	N	D	O									

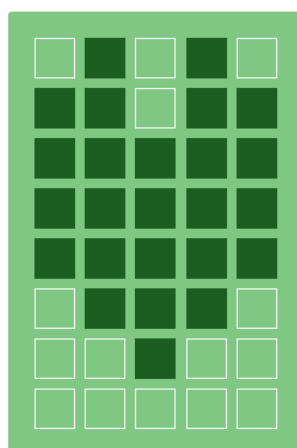
Caracteres personalizados

Aparte de solo letras, números o caracteres especiales, podemos crear nuestros propios caracteres.

<https://maxpromer.github.io/LCD-Character-Creator/> ← ese link es para crear caracteres personalizados

LCD Custom Character Generator

Support character lcd and create code for Arduino.



Clear

Invert

Link

- [Arduino LCD Circuit](#)
- [Arduino LCD I2C Circuit](#)
- [Arduino LCD I2C library](#)

Color

☒ Green

☐ Blue

Microcontroller

☒ Arduino

Interfacing

☐ Parallel

☒ I2C

Data Type

☒ Binary

☐ Hex

Code

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

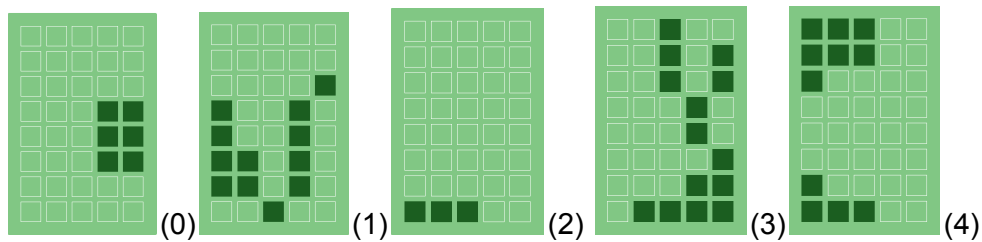
// Set the LCD address to 0x27 in PCF8574
LiquidCrystal_I2C lcd(0x3F, 16, 2);

byte customChar[] = {
  B01010,
  B11011,
  B11111,
  B11111,
  B11111,
  B11111,
  B01110,
  B00100,
  B00000
};
```

0	1	0	1	0
1	1	0	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
0	1	1	1	0
0	0	1	0	0
0	0	0	0	0

- Se crea la variable byte nombrecaracter [] = {}
- Dentro del setup llamamos al carácter: `lcd.createChar(0, nombrecaracter);`
- Si lo queremos imprimir por pantalla: `lcd.write(byte(0));`

Para hacer un dibujo más complejo podemos implicar varios caracteres personalizados:



Decimos la posición en la que empezamos a escribir
 el print es para texto y el write es para caracteres especiales
 y se pone en orden según lo que queremos que salga.

```
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27, 16, 2);

//caracteres plantita
#pragma region flor
byte flor1[8] = { B00000, B00000, B00000, B00011, B00011, B00011,
B00000, B00000};
byte flor2[8] = { B00000, B00000, B00001, B10010, B10010, B11010,
B11010, B00100};
byte flor3[8] = { B00000, B00000, B00000, B00000, B00000, B00000,
B00000, B11100};
byte flor4[8] = { B00100, B00101, B00101, B00010, B00010, B00001,
B00011, B01111};
byte flor5[8] = { B11100, B11100, B10000, B00000, B00000, B00000,
B10000, B11100};
#pragma endregion

void setup(){
  lcd.createChar(0, flor1);
  lcd.createChar(1, flor2);
  lcd.createChar(2, flor3);
  lcd.createChar(3, flor4);
  lcd.createChar(4, flor5);

  #pragma region Filal
  //caracteres en primera fila
  lcd.setCursor(0,0);
  lcd.write(byte(0));
  lcd.write(byte(1));
  lcd.write(byte(2));
```



```
#pragma endregion
#pragma region Fila2
//caracteres en segunda fila
lcd.setCursor(0,1);
lcd.print (" ");
lcd.write(byte(3));
lcd.write(byte(4));
#pragma endregion
}
```

ANIMACIÓN POR PANTALLA

Es muy parecida a escribir, la única diferencia es que metemos el código dentro de loop(); en vez de setup();

- creamos dos dibujos de las plantas y vamos alternando con un delay(1000);

```
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27, 16, 2); // Inicia el LCD en la dirección
0x27 con 16 caracteres y 2 líneas

//caracteres plantita
#pragma region floraniamacion1
byte flor1[8] = { B00000, B00000, B00000, B00011, B00011, B00011,
B00000, B00000};
byte flor2[8] = { B00000, B00000, B00001, B10010, B10010, B11010,
B11010, B00100};
byte flor3[8] = { B00000, B00000, B00000, B00000, B00000, B00000,
B00000, B11100};
byte flor4[8] = { B00100, B00101, B00101, B00010, B00010, B00001,
B00011, B01111};
byte flor5[8] = { B11100, B11100, B10000, B00000, B00000, B00000,
B10000, B11100};
#pragma endregion
#pragma region floraniamacion2
byte flore1[8] = { B00000, B00000, B00000, B00001, B00001, B00001,
B00000, B00000};
byte flore2[8] = { B00000, B00000, B00010, B11001, B11001, B11101,
B01101, B00010};
```

```

byte flore3[8] = { B00000, B00000, B00000, B00000, B00000, B00000,
B01110, B01110};
byte flore4[8] = { B00010, B00010, B00011, B00001, B00001, B00001,
B00011, B01111};
byte flore5[8] = { B11110, B11000, B00000, B00000, B00000, B00000,
B10000, B11100};
#pragma endregion

void setup() {
    lcd.init();
    lcd.backlight();
    lcd.clear();

    lcd.setCursor(6, 0);
    lcd.print("NO HAY");

    lcd.setCursor(5, 1);
    lcd.print("CLAVELES");
}

void loop() {
    /*lcd.setCursor(9, 1);
    lcd.print(millis() / 2000);*/
    animacionPlanta();
}

void animacionPlanta(){
    //frame 1
    #pragma region llamar animacion 1          //llamar caracteres primer
frame
    lcd.createChar(0, flor1);
    lcd.createChar(1, flor2);
    lcd.createChar(2, flor3);
    lcd.createChar(3, flor4);
    lcd.createChar(4, flor5);
    #pragma endregion

    #pragma region Fila11                      //escribir en pantalla primer
frame
    lcd.setCursor(0,0);
    lcd.write(byte(0));
    lcd.write(byte(1));

```



```

    lcd.write(byte(2));
    #pragma endregion
    #pragma region Fila21                                //escribir en pantalla
segundo frame
    lcd.setCursor(0,1);
    lcd.print (" ");
    lcd.write(byte(3));
    lcd.write(byte(4));
    #pragma endregion
    delay(1000);

    //frame 2
    #pragma region llamar animacion 2                    //llamar caracteres segunda
frame
    lcd.createChar(5, flore1);
    lcd.createChar(6, flore2);
    lcd.createChar(7, flore3);
    lcd.createChar(8, flore4);
    lcd.createChar(9, flore5);
    #pragma endregion

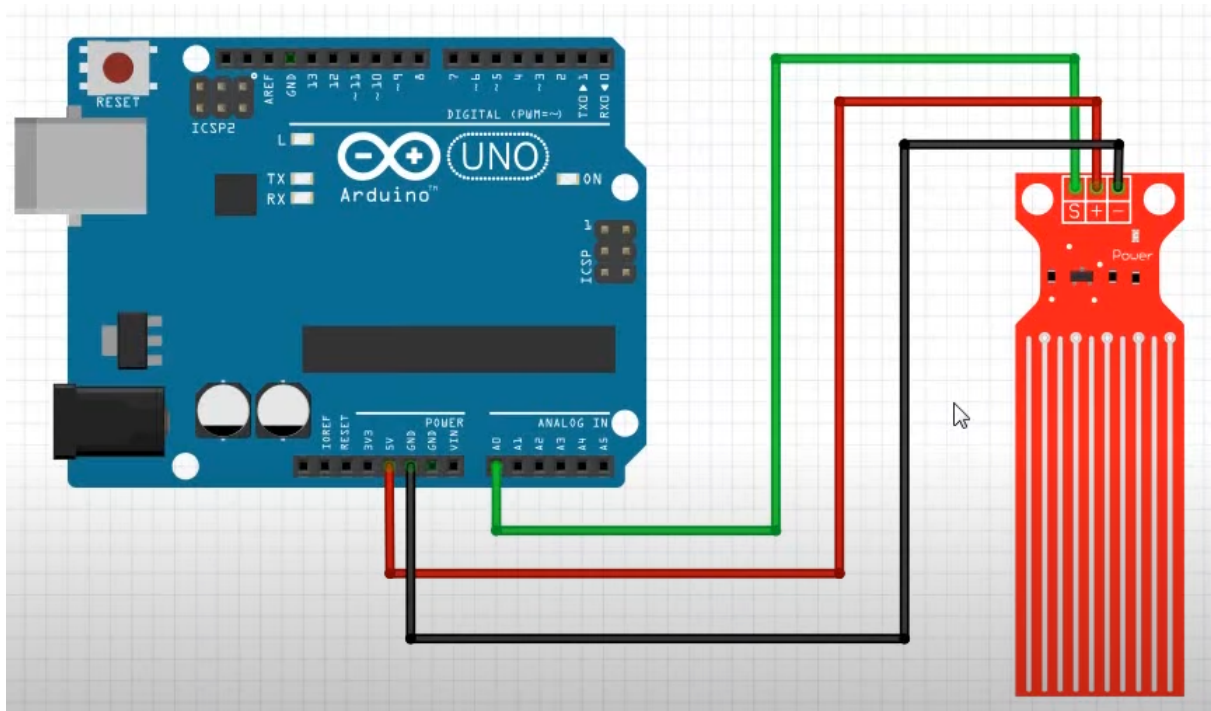
    #pragma region Fila12
    lcd.setCursor(0,0);
    lcd.write(byte(5));
    lcd.write(byte(6));
    lcd.write(byte(7));
    #pragma endregion
    #pragma region Fila22
    lcd.setCursor(0,1);
    lcd.print (" ");
    lcd.write(byte(8));
    lcd.write(byte(9));
    #pragma endregion
    delay(1000);
}

```

SENSOR DE HUMEDAD

Así se conecta.

(si quieres conectar la pantalla y el sensor a la vez, uno de los dos tiene que ir el pin que le da los 5 voltios en otro lado, en mi caso he usado el PIN13)



Poco que decir porque entender entiendo poco xD (sinceridad lo primero)

```
int SENSOR; //variable que recoge datos del sensor

void setup() {
  Serial.begin(9600);
  pinMode (13, OUTPUT); //activar el PIN13
}

void loop() {
  digitalWrite (13,1);
  SENSOR = analogRead(A0); //llama al sensor
  Serial.println(SENSOR); //escribe x consola valores de SENSOR
  delay(1000);
}
```

EL CÓDIGO CON PANTALLA Y SENSOR

Aquí están mezclados pantalla y sensor

```
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27, 16, 2); // Inicia el LCD en la dirección
0x27 con 16 caracteres y 2 líneas
int SENSOR;
//caracteres plantita
#pragma region floranimacion1
byte flor1[8] = { B00000, B00000, B00000, B00011, B00011, B00011,
B00000, B00000};
byte flor2[8] = { B00000, B00000, B00001, B10010, B10010, B11010,
B11010, B00100};
byte flor3[8] = { B00000, B00000, B00000, B00000, B00000, B00000,
B00000, B11100};
byte flor4[8] = { B00100, B00101, B00101, B00010, B00010, B00001,
B00011, B01111};
byte flor5[8] = { B11100, B11100, B10000, B00000, B00000, B00000,
B10000, B11100};
#pragma endregion
#pragma region floranimacion2
byte flore1[8] = { B00000, B00000, B00000, B00001, B00001, B00001,
B00000, B00000};
byte flore2[8] = { B00000, B00000, B00010, B11001, B11001, B11101,
B01101, B00010};
byte flore3[8] = { B00000, B00000, B00000, B00000, B00000, B00000,
B01110, B01110};
byte flore4[8] = { B00010, B00010, B00011, B00001, B00001, B00001,
B00011, B01111};
byte flore5[8] = { B11110, B11000, B00000, B00000, B00000, B00000,
B10000, B11100};
#pragma endregion

void setup() {
  lcd.init();
  lcd.backlight();
  lcd.clear();
  Serial.begin(9600); //esto es de la pantalla, que ni puta idea
  pinMode (13, OUTPUT); //activa el pin 13
```

```

    lcd.setCursor(3, 0);
    lcd.print("HAY AGUA?");
}

void loop() {
    digitalWrite (13,1);
    animacionPlanta();
    SENSOR = analogRead(A0);
    comprobacion();
}

void comprobacion(){
    if (SENSOR <= 30){
        Serial.println(SENSOR);
        lcd.setCursor(3, 1);
        lcd.print(" AGUAAAA!");
    }
    else if (SENSOR >30){
        Serial.println(SENSOR);
        lcd.setCursor(3, 1);
        lcd.print("  IS WET ");
    }
}

void animacionPlanta(){
    //frame 1
    #pragma region llamar animacion 1      //llamar caracteres primer
frame
    lcd.createChar(0, flor1);
    lcd.createChar(1, flor2);
    lcd.createChar(2, flor3);
    lcd.createChar(3, flor4);
    lcd.createChar(4, flor5);
    #pragma endregion

    #pragma region Fila11                  //escribir en pantalla primer
frame
    lcd.setCursor(0,0);
    lcd.write(byte(0));
    lcd.write(byte(1));
    lcd.write(byte(2));
    #pragma endregion
    #pragma region Fila21                  //escribir en pantalla

```

```

segundo frame
    lcd.setCursor(0,1);
    lcd.print (" ");
    lcd.write(byte(3));
    lcd.write(byte(4));
    #pragma endregion

    #pragma region Fila11                                //escribir en pantalla primer
frame
    lcd.setCursor(13,0);
    lcd.write(byte(0));
    lcd.write(byte(1));
    lcd.write(byte(2));
    #pragma endregion
    #pragma region Fila21                                //escribir en pantalla
segundo frame
    lcd.setCursor(13,1);
    lcd.print (" ");
    lcd.write(byte(3));
    lcd.write(byte(4));
    #pragma endregion
    delay(1000);

    //frame 2
    #pragma region llamar animacion 2                    //llamar caracteres segunda
frame
    lcd.createChar(5, flore1);
    lcd.createChar(6, flore2);
    lcd.createChar(7, flore3);
    lcd.createChar(8, flore4);
    lcd.createChar(9, flore5);
    #pragma endregion

    #pragma region Fila12
    lcd.setCursor(0,0);
    lcd.write(byte(5));
    lcd.write(byte(6));
    lcd.write(byte(7));
    #pragma endregion
    #pragma region Fila22
    lcd.setCursor(0,1);
    lcd.print (" ");

```

```
lcd.write(byte(8));  
lcd.write(byte(9));  
#pragma endregion  
  
#pragma region Fila12  
lcd.setCursor(13,0);  
lcd.write(byte(5));  
lcd.write(byte(6));  
lcd.write(byte(7));  
#pragma endregion  
#pragma region Fila22  
lcd.setCursor(13,1);  
lcd.print (" ");  
lcd.write(byte(8));  
lcd.write(byte(9));  
#pragma endregion  
delay(1000);  
}
```

BIBLIOGRAFÍA

<https://docs.arduino.cc/built-in-examples/> (creo que viendo los ejemplos que te trae la propia web de arduino te haces el curso. Eso sí, el cómo funciona la pantalla olvidate 😊)
<https://www.youtube.com/watch?v=bV7rujD-9NU&list=PLEzmH7aN82FEh2JjYuNCvFu6wFolHai32> (esto es un curso para aprender desde 0)

(Ejemplos de código)

<https://github.com/johannpereze/cursoArduinoJohannPerezE>
<https://github.com/bitwiseAr/Curso-Arduino-desde-cero/>

(animación en pantalla código)

- https://github.com/creatividadcodificada/ProyectosArduino/blob/master/LCd_arduino_dinosaurio_completo.ino
- https://www.youtube.com/watch?v=_yDPX4DIbA