

UNIVERSIDAD  
**NACIONAL**  
DE COLOMBIA

*Proyecto cultural y colectivo de nación*

Sede Medellín

# Taller de modelado: Arquitectura de soluciones

*Sede Medellín*

Monitor:  
Sebastián Figueroa Cañas  
Ingeniero Electrónico

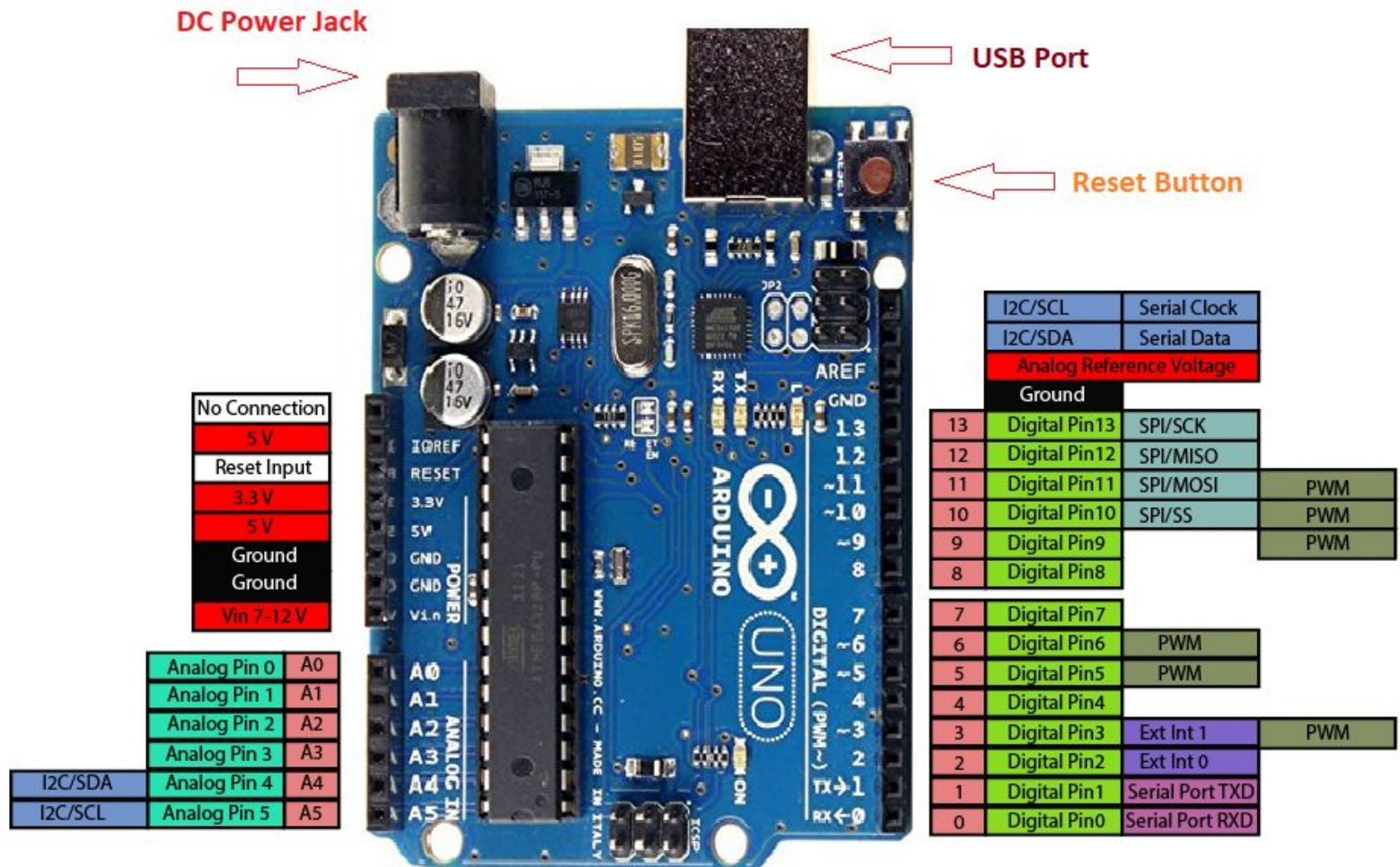


UNIVERSIDAD  
**NACIONAL**  
DE COLOMBIA

Número	Fecha	Titulo
1	2/10/2019	Introducción Arduino - Hola mundo electrónico
2	9/10/2019	Introducción Electrónica
3	16/10/2019	Uso de sensores con Arduino I
4	23/10/2019	Uso de sensores con Arduino II
5	30/10/2019	Actuadores con Arduino (Motores y Relay)
6	6/11/2019	IR+Bluetooth + Sensores + Actuadores
7	13/11/2019	Prototipo 1 del proyecto
8	20/11/2019	Introducción Raspberry Pi
9	27/11/2019	Configuración Raspberry (Network, RTC-Clock, etc)
10	4/12/2019	Raspberry + Arduino
11	11/12/2019	Web +Database + Raspberry pi
12	18/12/2019	Web +Database + Raspberry pi
13	15/01/2020	Web +Database + Raspberry pi

Número Sesión	Fecha	Semana Semestre	Estado	Titulo
1	2/10/2019	2	OK	Introducción Arduino - Hola mundo electrónico
2	9/10/2019	3	OK	Introducción Electrónica
3	16/10/2019	4	OK	Uso de sensores con Arduino I
4	23/10/2019	5	OK	Uso de sensores con Arduino II
		6	OK	No se realizó
5	29/01/2020	7	OK	Sesión de repaso
6	5/02/2020	8		Actuadores con Arduino (Motores y Relay)
7	12/02/2020	9		Protocolos de Comunicación, I2C, SPI, Serial Bluetooth
8	19/02/2020	10		Prototipo 1 del proyecto
9	26/02/2020	11		Introducción Raspberry Pi
10	4/03/2020	12		Configuración Raspberry (Network, RTC-Clock)
12	11/03/2020	13		Raspberry + Arduino
13	18/03/2020	14		Web +Database + Raspberry pi
12	25/03/2020	15		Web +Database + Raspberry pi

# Revisión de conceptos



**Arduino Uno Pinout**

# Sensores

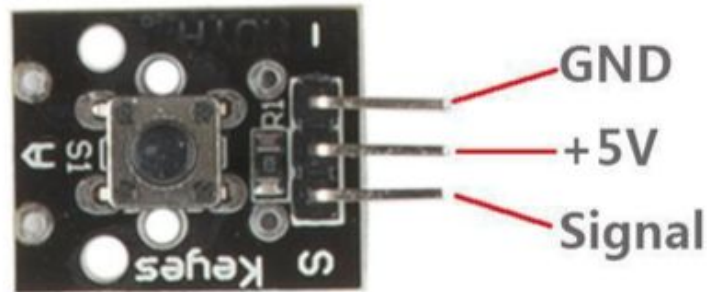
*Sede Medellín*



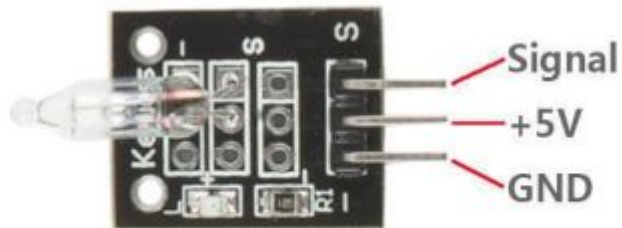
UNIVERSIDAD  
**NACIONAL**  
DE COLOMBIA

## Sensores binarios

Pulsador



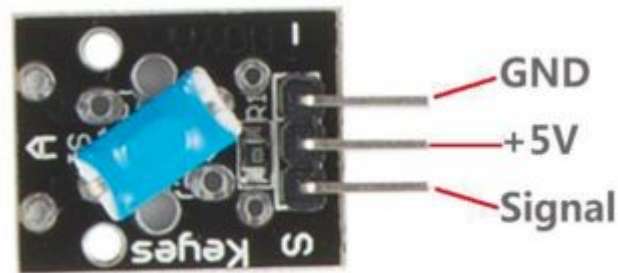
Sensor de Mercurio



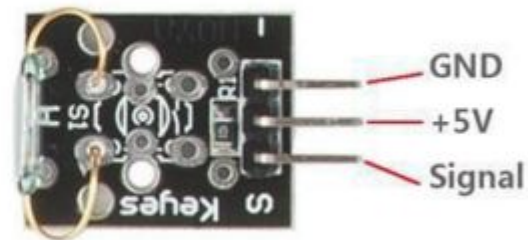


## Sensores binarios

Tilt Switch / Suiche de inclinación



**Reed Switch**



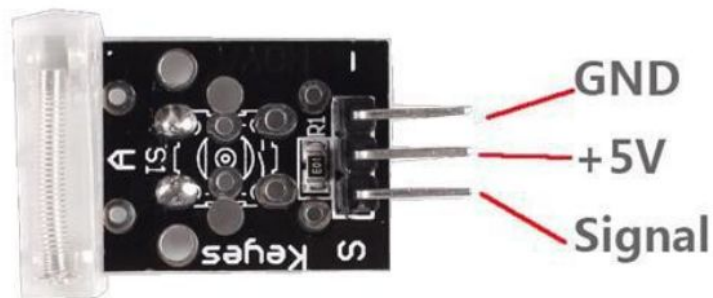
**Mini Reed**

## Sensores binarios

Sensor de flama

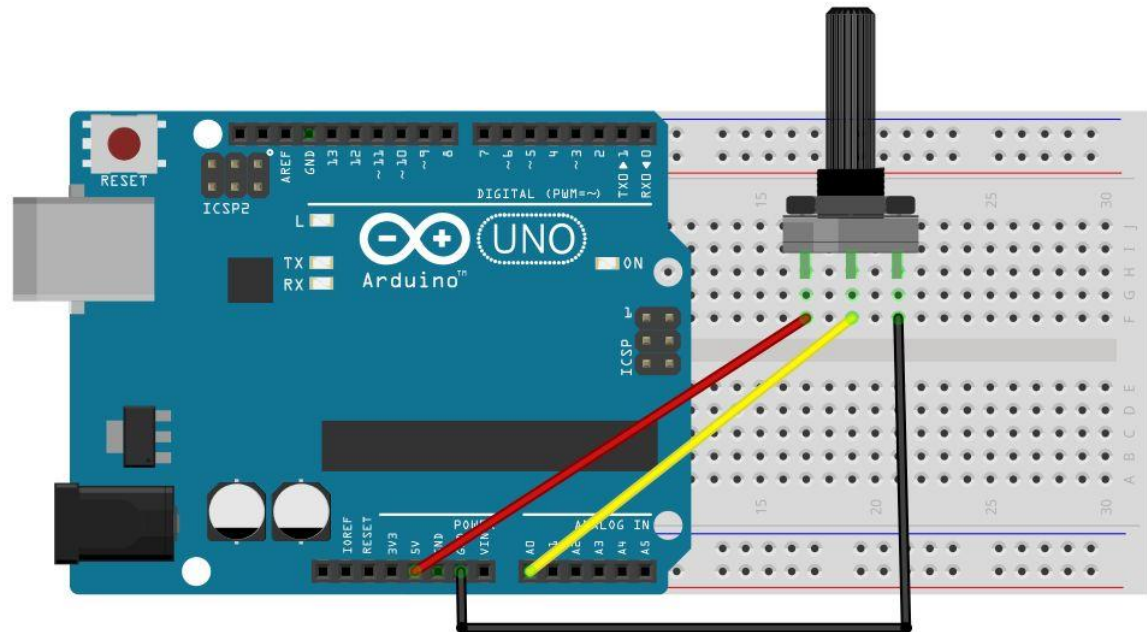


Sensor de golpe



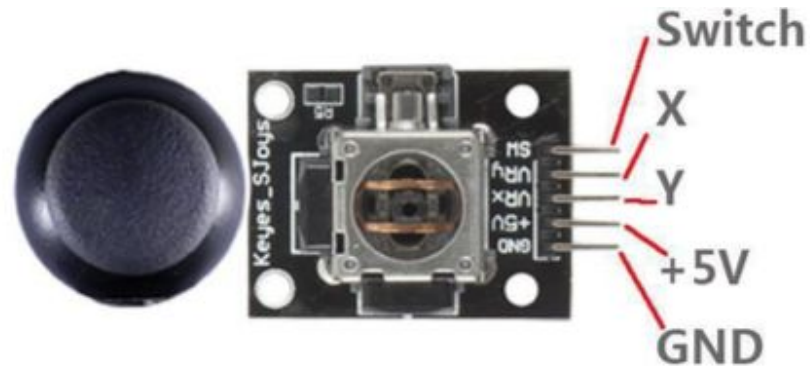
## Sensores análogos

Potenciómetro



## Sensores análogos

Joystick



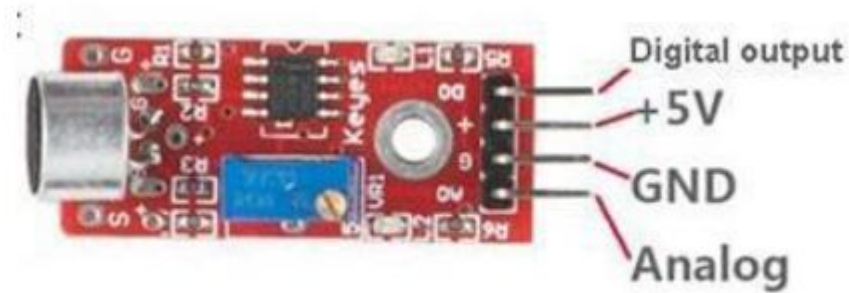
Sensor magnético



Linear Hall

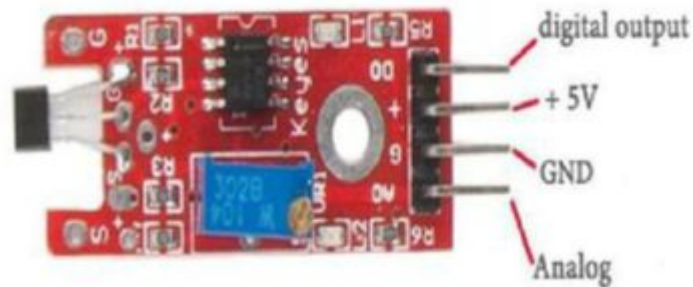
## Sensores análogos

Micrófono



High-sensitive Voice Sensor

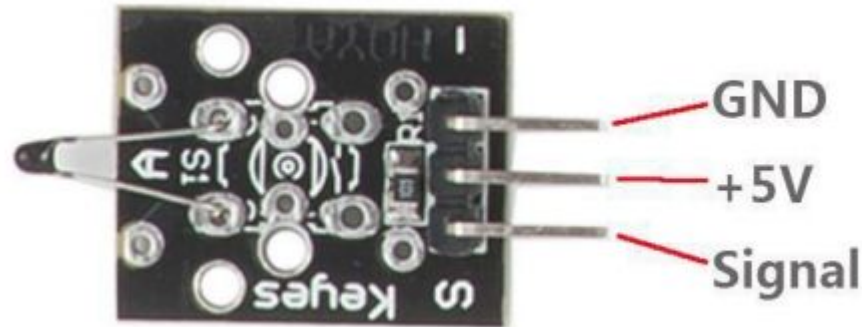
Sensor magnético



Linear Hall

## Sensores análogos

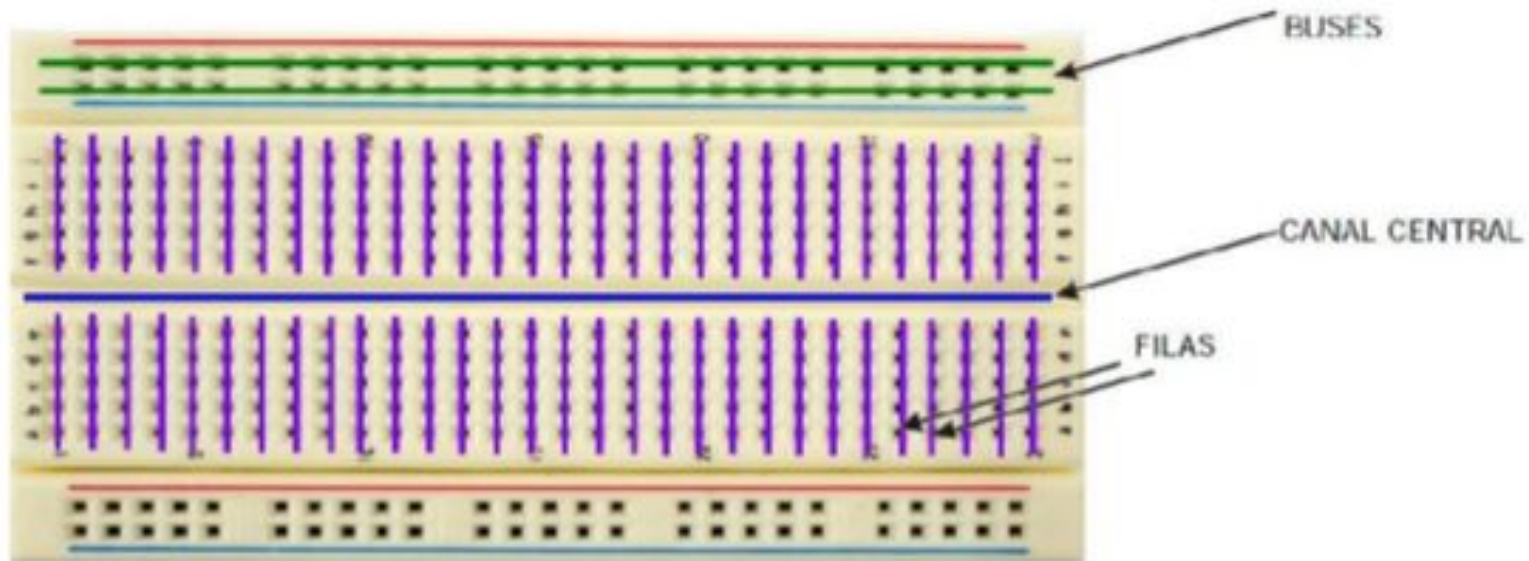
Sensor de Temperatura  
análogo



Sensor magnético



## La protoboard



## Proyecto cultural y colectivo de nación

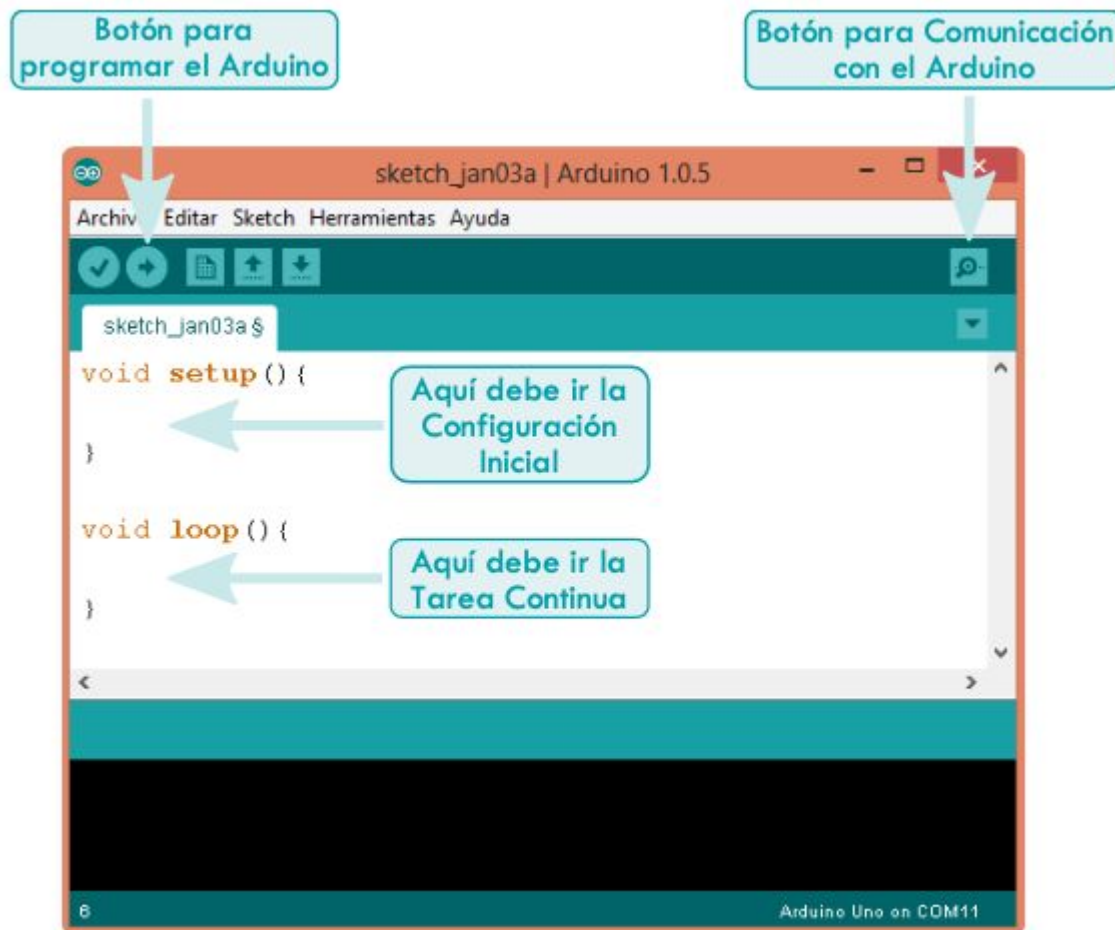


Figura 2.4: Interfaz de Programación Arduino



Instrucción	Descripción
<code>pinMode (nPin, Mod)</code>	nPin indica el numero un pin que queramos configurar. Mod indica el modo de Trabajo, si es pin de entrada en vez Mod escribimos <b>INPUT</b> y como pin de Salida escribimos <b>OUTPUT</b> .
<b>Ejemplo:</b> <code>pinMode (13, OUTPUT)</code>	Así se configura el pin número 13 como pin de salida.
<code>digitalWrite (nPin, Valor)</code>	Escribe un Valor <b>HIGH</b> (1) o <b>LOW</b> (0) en el pin indicado por nPin.
<b>Ejemplo:</b> <code>digitalWrite (2, HIGH)</code>	En este ejemplo se le da un valor de <b>HIGH</b> al pin 2.
<code>analogWrite (nPin, Valor);</code>	Escribe un Valor entre 0 y 255 en el pin indicado por nPin.
<b>Ejemplo:</b> <code>analogWrite (3, 150);</code>	En este ejemplo se le da un valor de <b>150</b> al pin 3.
<code>digitalRead (nPin);</code>	Lee el valor digital del pin indicado por npin. Esta instrucción nos dice si el pin se encuentra en <b>HIGH</b> o en <b>LOW</b> nPin.
<b>Ejemplo:</b> <code>digitalRead (8);</code>	En este ejemplo se lee el valor del pin 8.

<code>analogRead(A0);</code>	Lee el valor analogo del pin indicado por <code>npin</code> . Esta instrucción entrega un valor entre 0 y 1024 de acuerdo al voltaje del pin.
<b>Ejemplo:</b> <code>analogRead(A0);</code>	En este ejemplo se lee el valor del pin A0.
<code>delay(miliseecs);</code>	Es un retardo que detiene la ejecución del programa por el numero de milisegundos indicados por <code>miliseecs</code> .
<b>Ejemplo:</b> <code>delay(1000);</code>	En este ejemplo se hace un ratardo de 1 segundo (1000 milisegundos).
<code>Serial.begin(9600);</code>	Sirve para iniciar una comunicación entre el arduino y otros dispositivos como el Computador. El valor de 9600 es la velocidad de la comunicación.
<code>int Var= 1;</code>	Crea una variable llamada <code>Var</code> y se le da un valor (en este caso "1"). Todas las partes en las que aparezca <code>Var</code> tomaran el valor asignado.
<b>Ejemplo:</b> <code>int Val=2;</code>	En este ejemplo se crea una variable llamada <code>Val</code> y se le asigna un valor de 2.

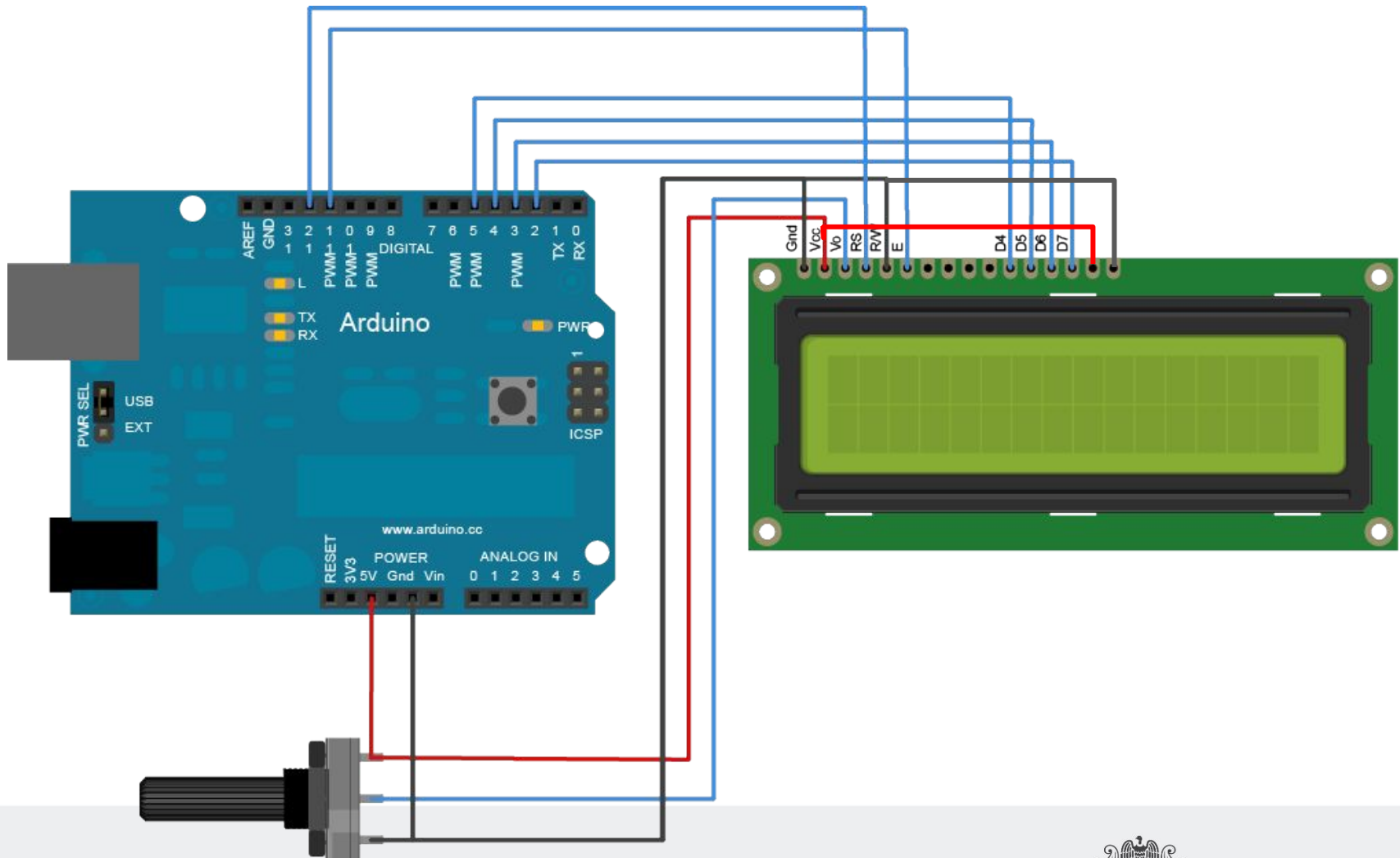
Tabla 2.1: Table caption

```
//Function to read serial depending on events
void serialEvent(){
  while(Serial.available()){
    char rec = Serial.read();
    if(rec == '\n'){
      //We got a line
      if(BUFF[buffCount-1]=='\r'){
        BUFF[buffCount -1]=0;
      }
      BUFF[buffCount] = 0; //En String
      OnLineReceived(String(BUFF));
      buffCount = 0;
    }else{
      if(buffCount< 100){
        BUFF[buffCount++] = rec;
      }
    }
  }
}
```

Revisar el repositorio:

[https://github.com/sefigueroacUNAL/TModeladoUNAL/blob/master/Sesiones/P06\\_REPASO/SimpleSerialRead/SimpleSerialRead.ino](https://github.com/sefigueroacUNAL/TModeladoUNAL/blob/master/Sesiones/P06_REPASO/SimpleSerialRead/SimpleSerialRead.ino)

# Ensamble LCD



Abrir los  
ejemplos de  
Arduino

## LiquidCrystal Library

This library allows an Arduino board to control LiquidCrystal displays (LCDs) based on the Hitachi HD44780 (or a compatible) chipset, which is found on most text-based LCDs. The library works with in either 4- or 8-bit mode (i.e. using 4 or 8 data lines in addition to the rs, enable, and, optionally, the rw control lines).

To use this library

```
#include <LiquidCrystal.h>
```

## Examples

- [Autoscroll](#): Shift text right and left.
- [Blink](#): Control of the block-style cursor.
- [Cursor](#): Control of the underscore-style cursor.
- [Display](#): Quickly blank the display without losing what's on it.
- [Hello World](#): Displays "hello world!" and the seconds since reset.
- [Scroll](#): Scroll text left and right.
- [Serial Display](#): Accepts serial input, displays it.
- [Set Cursor](#): Set the cursor position.
- [Text Direction](#): Control which way text flows from the cursor.

### Function

- [LiquidCrystal\(\)](#)
- [begin\(\)](#)
- [clear\(\)](#)
- [home\(\)](#)
- [setCursor\(\)](#)
- [write\(\)](#)
- [print\(\)](#)
- [cursor\(\)](#)
- [noCursor\(\)](#)
- [blink\(\)](#)
- [noBlink\(\)](#)
- [display\(\)](#)
- [noDisplay\(\)](#)
- [scrollDisplayLeft\(\)](#)
- [scrollDisplayRight\(\)](#)
- [autoscroll\(\)](#)
- [noAutoscroll\(\)](#)
- [leftToRight\(\)](#)
- [rightToLeft\(\)](#)
- [createChar\(\)](#)

1. Escribir un programa para Arduino que permita hacer lo siguiente:
  - a. Escribir un string desde la consola serial y mostrarlo en el LCD
  - b. Limpiar el LCD a través de un comando desde la consola serial.
  - c. Mostrar una cara feliz al enviar el string :)
  - d. Mostrar una cara triste al enviar el string :(



1. Escribir un programa para Arduino que permita hacer lo siguiente:
  - a. Ingresar y editar un texto utilizando en joystick.

Escribir un programa para Arduino que permita hacer lo siguiente:

- a. Visualizar mediante el Lcd la temperatura, la humedad el nivel de luz y el valor del sensor de gases.
- b. Notificar por consola serial con una letra cuando se presione cada uno de 5 pulsadores. La letra de cada pulsador debe ser diferente.



**GRACIAS**

[medellin.unal.edu.co](http://medellin.unal.edu.co)

*Sede Medellín*



UNIVERSIDAD  
**NACIONAL**  
DE COLOMBIA