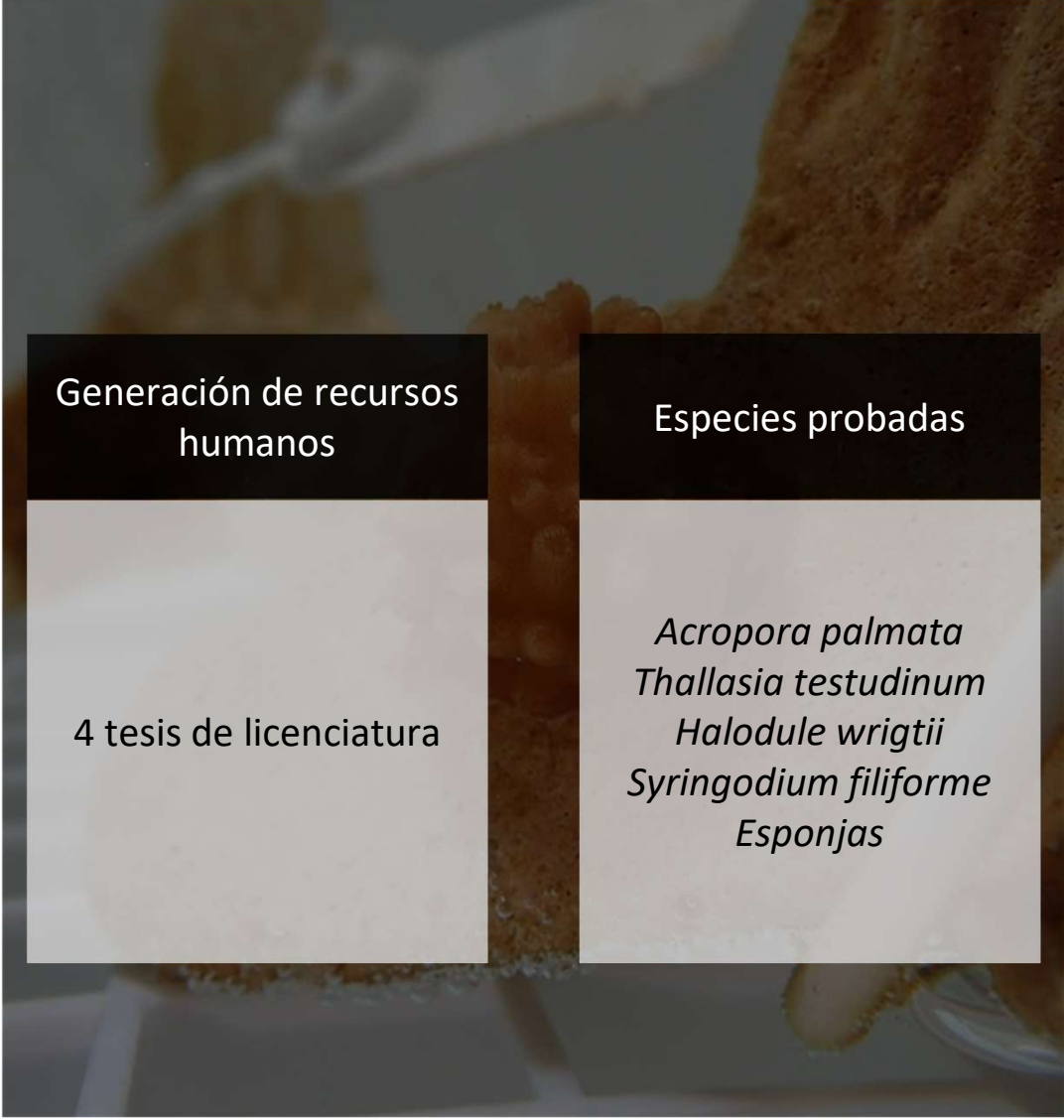




## **PROYECTO PAPIME PE207521**

# **Implementación e instrumentación de acuarios marinos como estrategia de enseñanza en la docencia e investigación**

**M. en C. Fernando Negrete, M. en C. Edgar Escalante, M. en I. Miguel Gomez, MTIA. Gustavo Villarreal, Dr. Edén Magaña, Dra. Anastazia Banaszack, Dra. Briggita I. van Tussenbroeck**




Generación de recursos humanos

4 tesis de licenciatura

Especies probadas

*Acropora palmata*  
*Thalassia testudinum*  
*Halodule wrightii*  
*Syringodium filiforme*  
Esponjas



Recursos generados

Desarrollo tecnológico de lámparas y monitoreo de acuarios

Divulgación para visitas guiadas

Curso/material de Arduino

Infraestructura

Cuarto de acuarios

Equipo de acuarios

Servidor

Día 1



# Curso Básico Instrumentación de acuarios con Arduino



PROYECTO PAPIME PE207521

## OBJETIVO DEL CURSO

QUE EL ESTUDIANTE APRENDA A CREAR SUS PROPIAS SOLUCIONES TECNOLÓGICAS CON LA AYUDA DE LA TECNOLOGÍA ARDUINO MEDIANTE EJEMPLOS PRACTICOS.

## ¿POR QUE SURGE ESTE CURSO?

Esta es la segunda edición del curso “Instrumentación con Arduino” impartido en el año 2019 por el M. en I. Miguel Ángel Gómez Raeli, solo que en esta edición nos enfocamos en la instrumentación de acuarios.

## ¿Qué son los acuarios?

Del latín “*aqua*” que significa agua y “*-rium*” que significa lugar o edificio.

En un inicio = “*vivarium*” → “cualquier encierro en el cual se mantienen animales vivos”.

Posteriormente = “*aqua-vivarium*” → confuso y difícil de pronunciar.

Finalmente = **acuario** → se extiende más allá del cultivo de plantas acuáticas y ahora abarca el mantenimiento tanto de organismos dulceacuícolas como marinos.



Sistema de acuarios experimental en Sisal, Yucatán



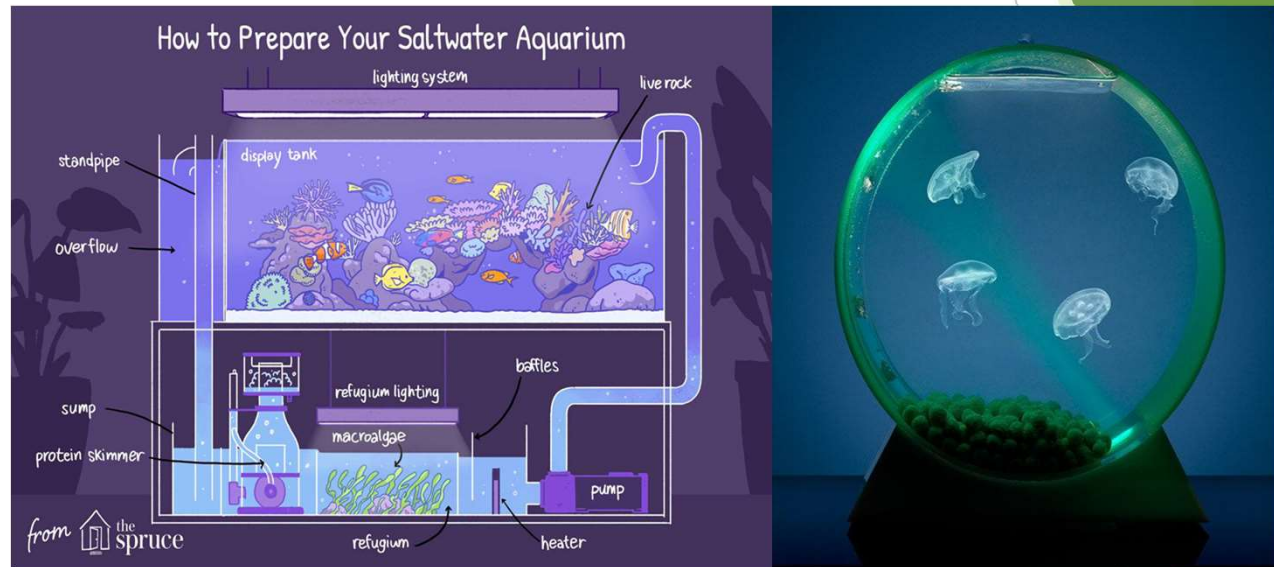
Acuario en Veracruz: recreativo, científico y conservación.



Instalaciones de la UASA.  
Enseñanza/educación

# Diseño de acuarios

- **Biología de la especie**
- Volumen
- Densidad
- Capacidad de carga
- Tipo de sistema
- Nutrición
- Enfermedades
- Equipos
- Objetivos del acuario
- Calidad del agua
- Sistemas de control
- Mantenimiento



# Sistemas de control en acuarios

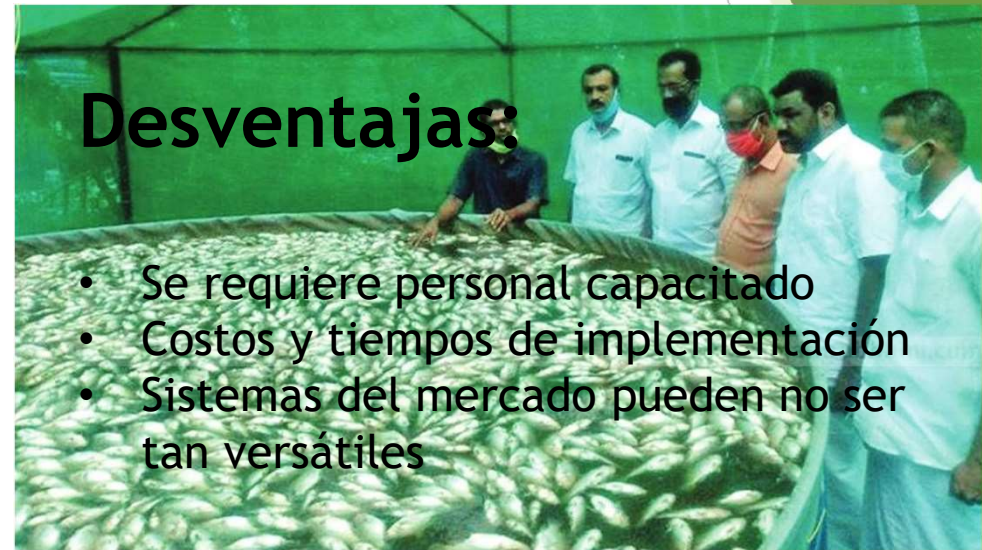
Los sistemas de control son: un conjunto de componentes interconectados formando un sistema de configuración que proveerá una respuesta deseada en el sistema.

## Ventajas

- Menor costo
  - Reducción de errores
  - Reducción de tiempo
  - Menor riesgo
  - Control de actividades
- Beneficios
- Evita errores del personal



DRAE15-1/  
DRAE15-2



## Desventajas

- Se requiere personal capacitado
- Costos y tiempos de implementación
- Sistemas del mercado pueden no ser tan versátiles

Mararikulam, India. Falla en oxigenación.

Censar la temperatura del agua y controlar un calentador o chiller. Pero, ¿podemos registrar su temperatura? ¿podemos hacer una rampa de temperatura? ¿podemos controlar múltiples equipos?

# Ejemplos de sistemas de control



Sistema de control de humedad y temperatura



Efecto de la temperatura en el sargazo pelágico





# Ejemplos de sistemas de control en el mercado



## PH/CO<sub>2</sub> CONTROLLER

Automate CO<sub>2</sub> dosing to ensure your plants get the right amount for respiration. Just plug a 115V solenoid valve (not included) into the socket supplied for dosing and your CO<sub>2</sub> supply to the solenoid. The included pH electrode measures 0 to 14 pH (± .2 pH accuracy) and has manual 2-point calibration. Controller has a set point range of 5.5 to 9.5 pH and a visual LED alarm that activates when the reading goes above the set point. Uses a 115V/60 Hz to 12VDC adapter (included) and measures 3 1/4" x 5 1/2" x 1 1/2". Two-year warranty on the base unit, six months on the probe. Ship weight is 2 lbs.

MODEL		EACH
MC122	PH/CO <sub>2</sub> SMART CONTROLLER	\$138.39
SM911	REPL. PH PROBE	44.09
CAL7	PH CALIBRATION SOLUTION 7.0	13.99
CAL4	PH CALIBRATION SOLUTION 4.01	13.99
CAL10	PH CALIBRATION SOLUTION 10.0	13.99



DRAE15-1/  
DRAE15-2



Switch de flujo

# Definición de una solución tecnológica

- Alcance
- Tiempo
- Entregables
- Presupuesto

Definición de un Objetivo



- Alcance
- Presupuesto
- Tiempo

Triangulo de Solución



- Soluciones de paga
- Desarrollo propio
- Desarrollos opensource

Estudio de mercado

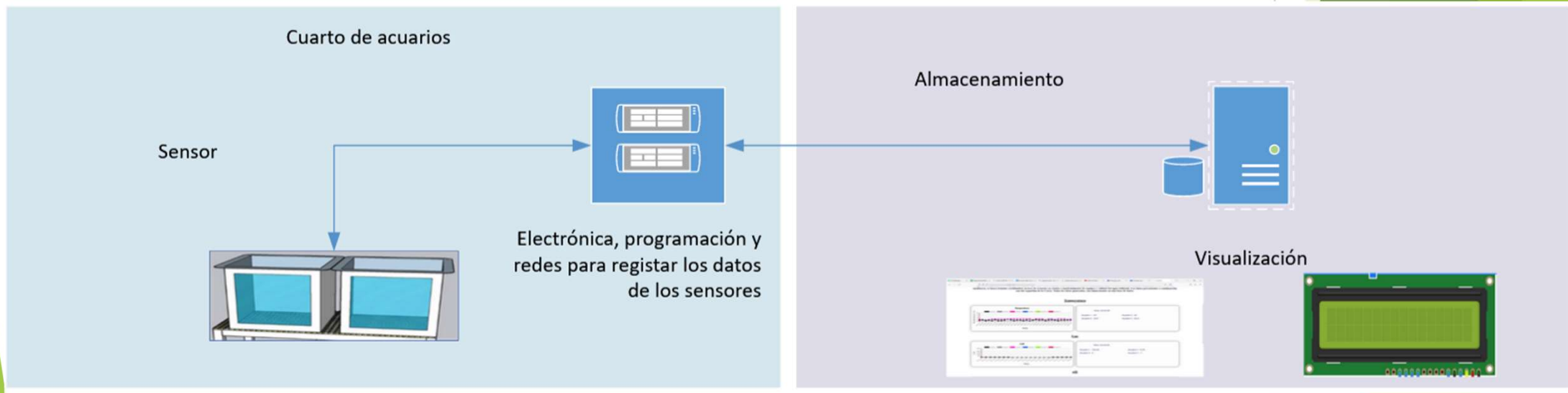


- Factibilidad Técnica
- Factibilidad Económica
- Mantenimiento a 3 años.

Definición de la alternativa



# Arquitectura



## Diseño de la solución

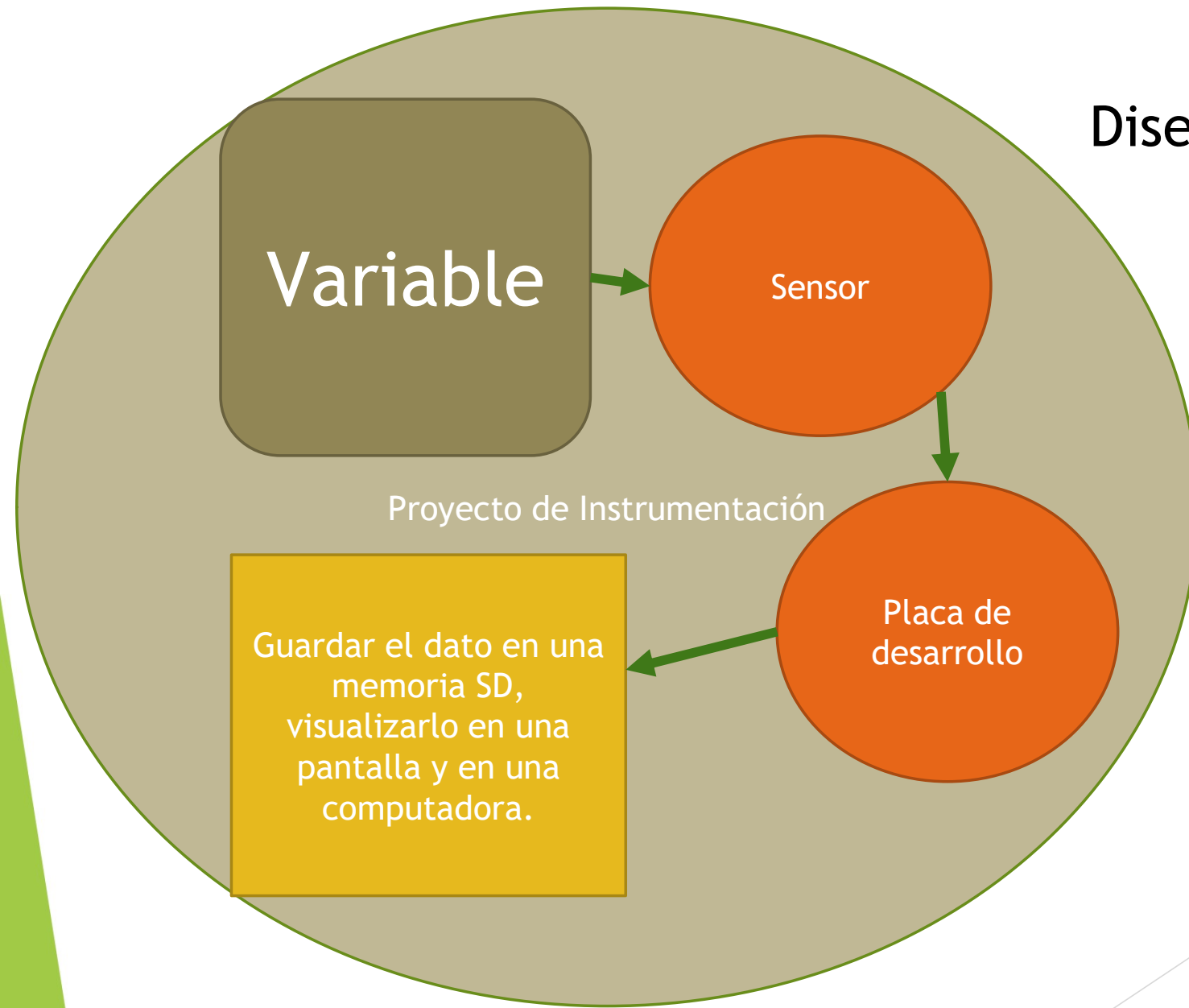
Variable

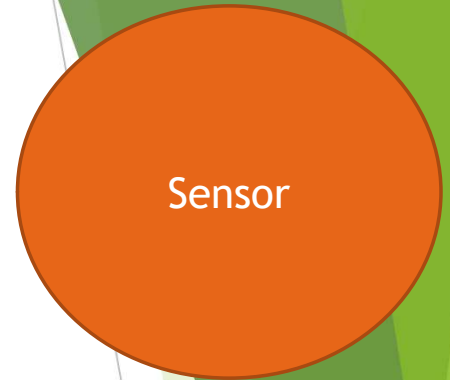
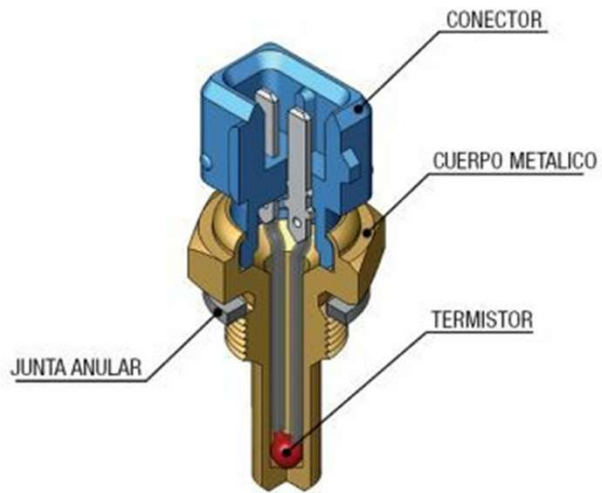
Sensor

Proyecto de Instrumentación

Placa de desarrollo

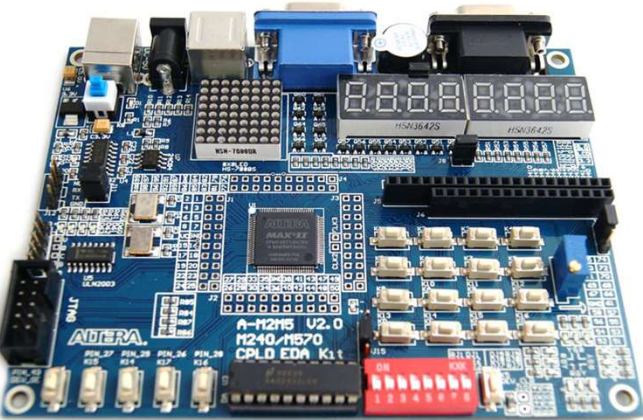
Guardar el dato en una memoria SD, visualizarlo en una pantalla y en una computadora.







Placas de desarrollo



El Arduino UNO según su fabricante es:

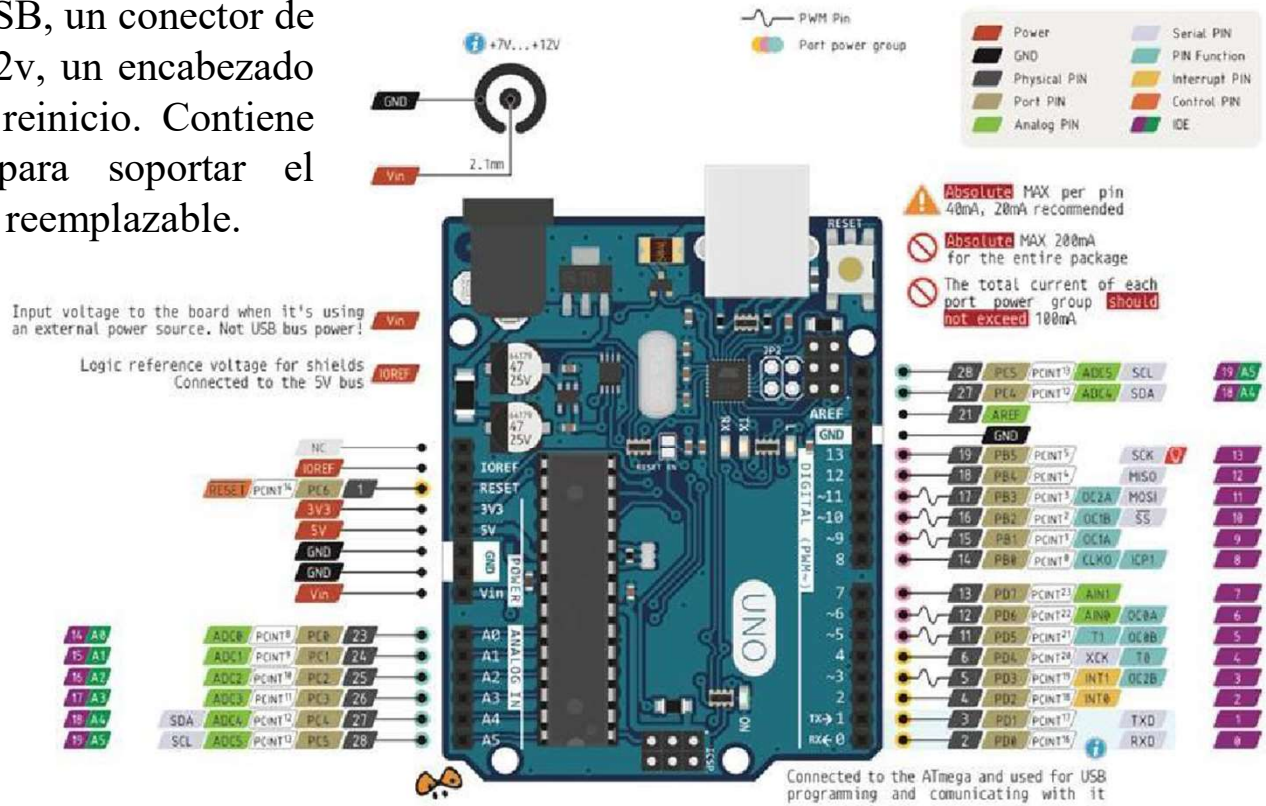
*“Es la mejor placa para comenzar con la electrónica y la codificación. Si esta es su primera experiencia jugando con la plataforma, UNO es la tarjeta más robusta con el que puede comenzar a jugar. La UNO es la tarjeta más utilizada y documentada de toda la familia Arduino.”*



Placa de  
desarrollo  
Arduino UNO

El Arduino Uno es la tarjeta de desarrollo basada en el microcontrolador ATmega328P. Tiene 14 pines de entrada o salida digital, de los cuales 6 se pueden usar como salidas PWM, 6 entradas analógicas, un cristal de cuarzo de 16 MHz, una conexión USB, un conector de alimentación de 7 a 12v, un encabezado ICSP y un botón de reinicio. Contiene todo lo necesario para soportar el microcontrolador, y es reemplazable.

# Placa de desarrollo Arduino UNO

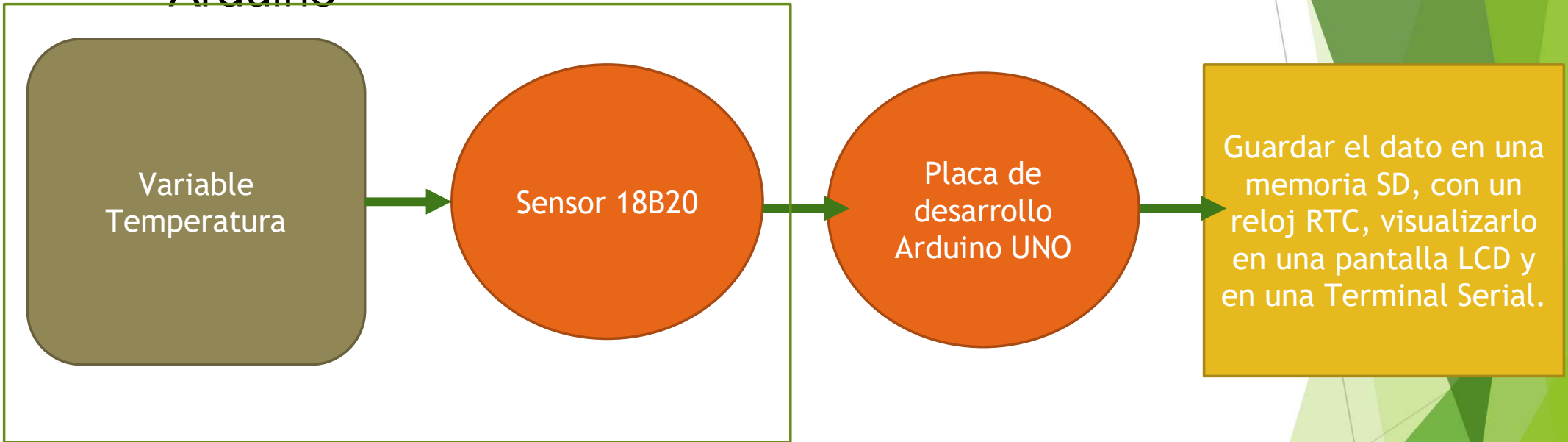




# Programación

- La **programación** es el proceso de crear un [conjunto de instrucciones](#) que le dicen a una computadora como realizar algún tipo de tarea. Pero no solo la acción de escribir un [código](#) para que la computadora o el [software](#) lo ejecute. Incluye, además, todas las tareas necesarias para que el código funcione correctamente y cumpla el objetivo para el cual se escribió

# Proyecto del Curso Básico de Instrumentación con Arduino



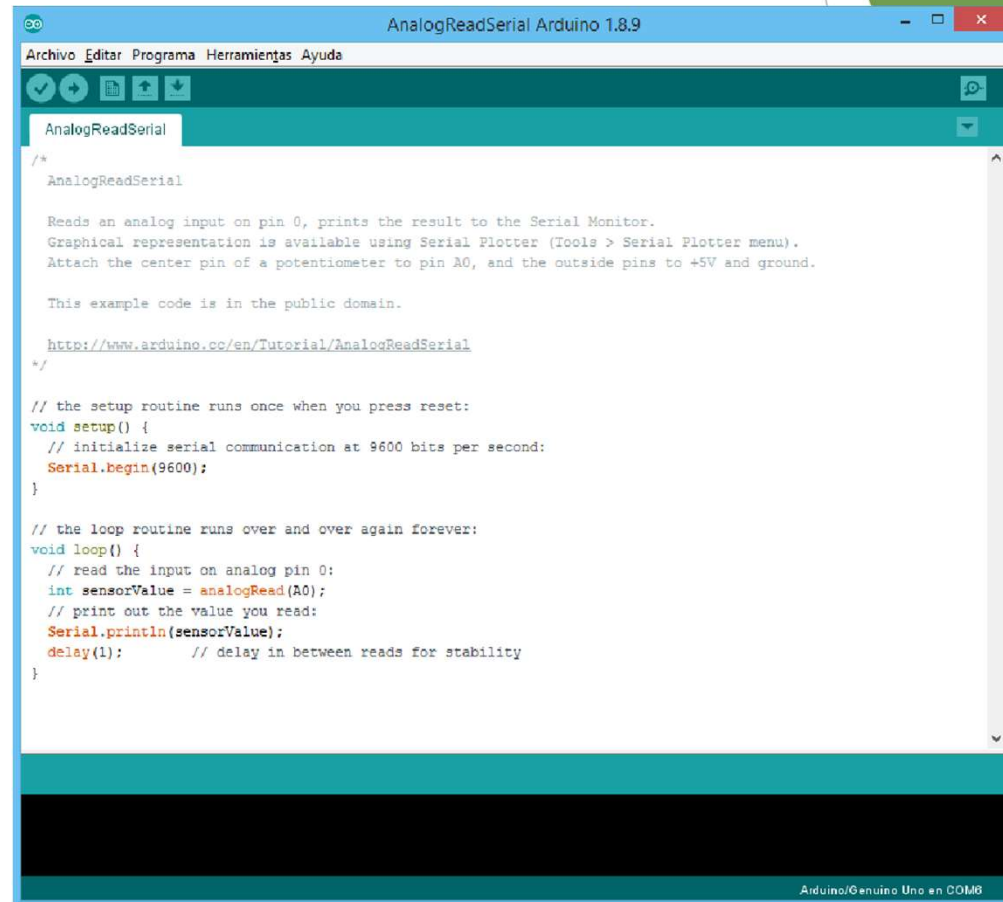
# Conociendo el Software.

Ejemplos: son los programas de ejemplo para utilizar las funciones, comandos básicos de Arduino así como sus librerías.

Librerías: son los programas que contienen instrucciones para utilizar en los programas principales que permiten hacer uso de otras funciones.

Monitor serial: es una interface mediante la cual permite enviar y recibir datos vía la comunicación USB.

Ayuda: permite consultar links acerca de los glosarios y términos utilizados por Arduino.

A screenshot of the Arduino IDE interface. The window title is 'AnalogReadSerial Arduino 1.8.9'. The menu bar includes 'Archivo', 'Editar', 'Programa', 'Herramientas', and 'Ayuda'. The toolbar shows icons for file operations and execution. The main text area contains the following code:

```
/*
AnalogReadSerial

Reads an analog input on pin 0, prints the result to the Serial Monitor.
Graphical representation is available using Serial Plotter (Tools > Serial Plotter menu).
Attach the center pin of a potentiometer to pin A0, and the outside pins to +5V and ground.

This example code is in the public domain.

http://www.arduino.cc/en/Tutorial/AnalogReadSerial
*/

// the setup routine runs once when you press reset:
void setup() {
  // initialize serial communication at 9600 bits per second:
  Serial.begin(9600);
}

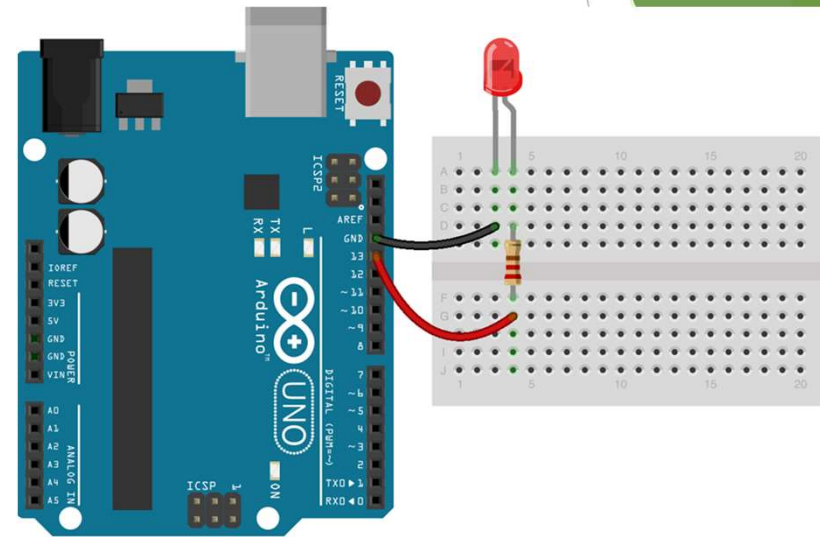
// the loop routine runs over and over again forever:
void loop() {
  // read the input on analog pin 0:
  int sensorValue = analogRead(A0);
  // print out the value you read:
  Serial.println(sensorValue);
  delay(1);        // delay in between reads for stability
}
```

The status bar at the bottom right indicates 'Arduino/Genuino Uno en COM8'.

Ejemplo blink

# Ejemplo: Blink

```
void setup() {  
  pinMode(LED_BUILTIN, OUTPUT);  
}  
  
void loop() {  
  digitalWrite(LED_BUILTIN, HIGH);  
  delay(1000);  
  digitalWrite(LED_BUILTIN, LOW);  
  delay(1000);  
}
```



Conectar con resistencia de 330 ohms

Hacer Ejemplo 2 o mas leds

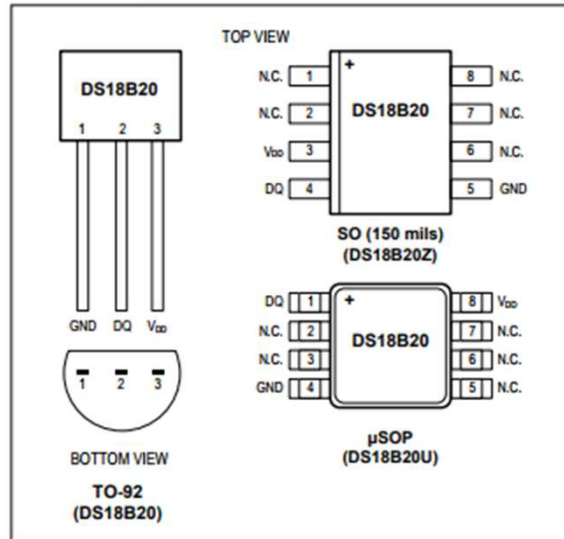
## Ejemplo: 1\_Serial

```
void setup() {  
  // start serial port at 9600 bps:  
  Serial.begin(9600);  
  while (!Serial) {  
    }  
}  
  
void loop() {  
  for(int i=0; i<10; i++){  
    Serial.print("Hola Mundo");  
    Serial.println(i);  
    delay(1000);  
  }  
}
```

Visualizarlo en una  
Terminal Serial.

Hacer Ejemplo serial\_basic

## Pin Configurations

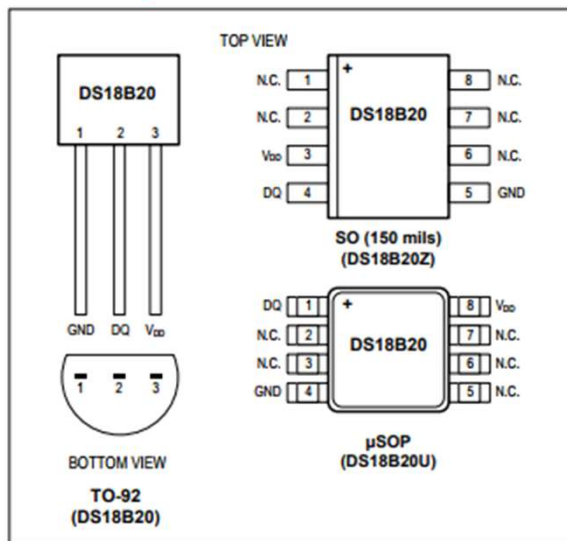


# Sensor de Temperatura

Sensor 18B20

- $V_{DD}$ : es la tensión de alimentación, podemos alimentar desde 3V a 5,5V.
- GND: es la toma de tierra. A este pin conectaremos la referencia 0V de nuestro circuito.
- DQ: es el pin de datos. Por este pin es por donde se recibirán todos los datos en el protocolo 1-Wire. Este protocolo tiene la ventaja para conectar varios sensores de temperatura DS18B20. Por lo tanto, solo utilizaremos 1 pin de Arduino para conectar múltiples sensores.

### Pin Configurations



## Sensor de Temperatura



Sensor 18B20

### RESOLUCIÓN

### TEMPERATURA

9-bit

0,5°C

10-bit

0,25°C

11-bit

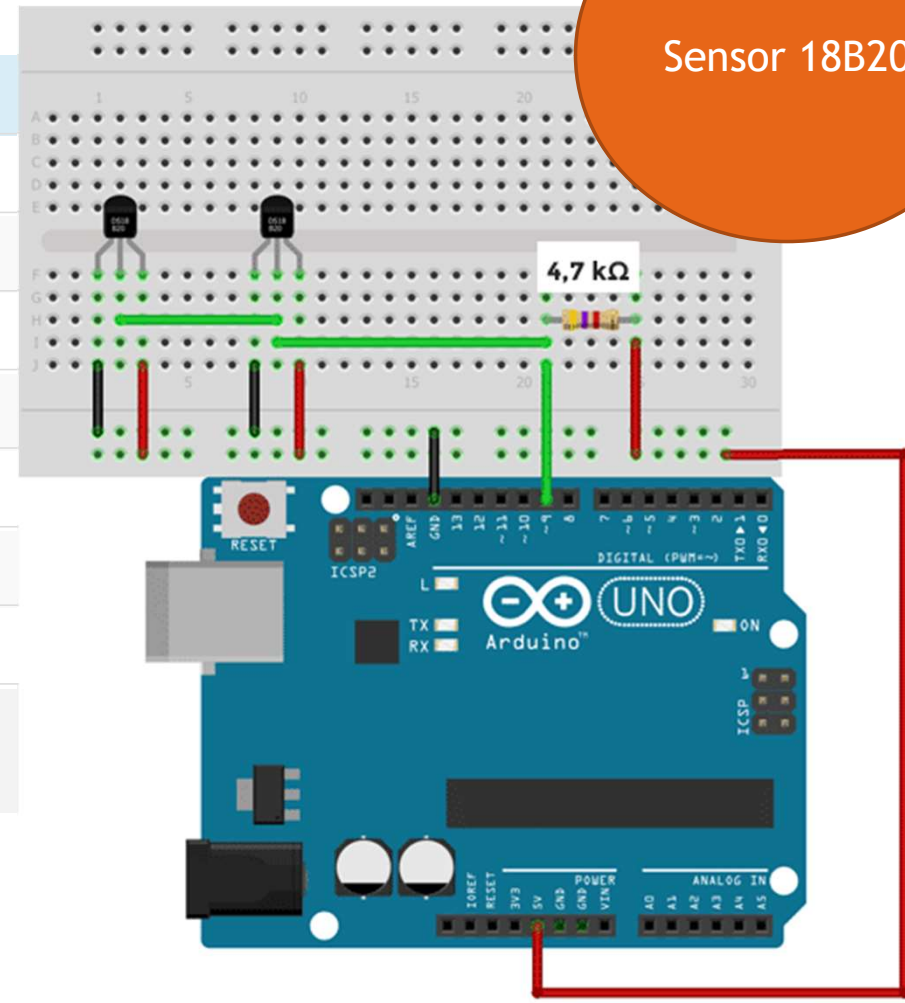
0,125°C

12-bit

0,0625°C

## Modo alimentación por el pin $V_{DD}$

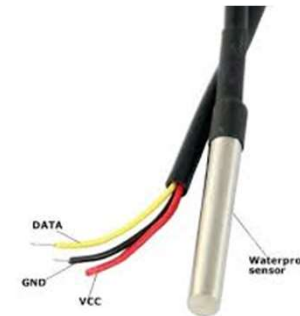
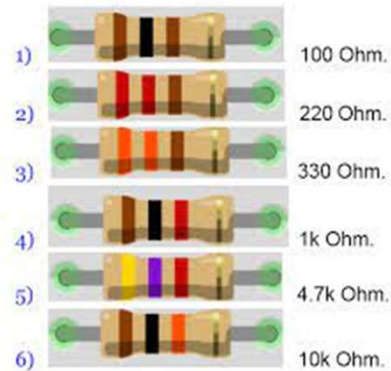
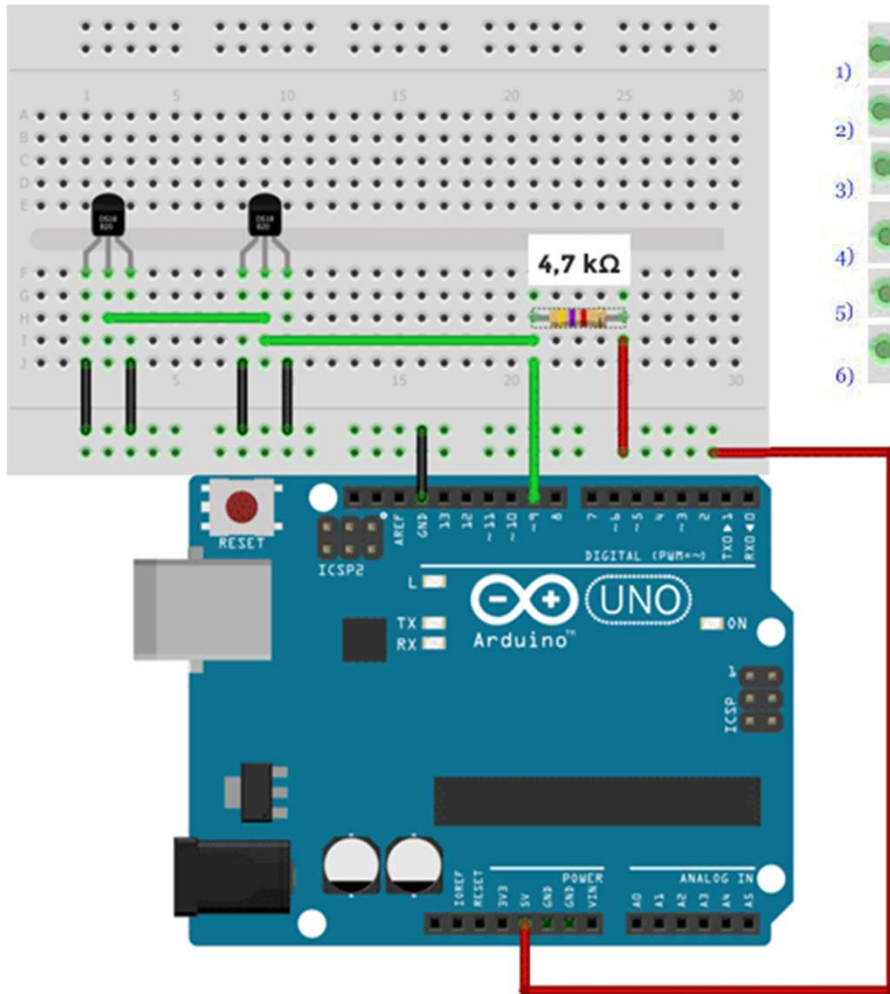
CARACTERÍSTICA	VALOR
Voltaje de alimentación	3V a 5,5V
VDD	voltaje de alimentación
GND	Tierra
DQ	Datos
Rango de temperaturas	-55°C a 125°C
Error (-10°C a 85°C)	$\pm 0,5^\circ\text{C}$
Error (-55°C a 125°C)	$\pm 2^\circ\text{C}$
Resolución programable	9-bit, 10-bit, 11-bit o 12-bit (default)





## Modo alimentación parásito por el pin DQ

Sensor 18B20



**RESISTENCIA  
PULL-UP**

**DISTANCIA DEL  
CABLE (METROS)**

4,7 kΩ

De 0 m a 5 m

3,3 kΩ

De 5 m a 10 m

2,2 kΩ

De 10 m a 20 m

1,2 kΩ

De 20 m a 50 m

## Ejemplo: 2\_Temperatura

```
#include <OneWire.h>
#include <DallasTemperature.h>
const int pinDatosDQ = 9;
int numeroSensoresConectados =0;
OneWire oneWireObjeto(pinDatosDQ);
DallasTemperature sensorDS18B20(&oneWireObjeto);
void setup() {
    Serial.begin(9600);
    sensorDS18B20.begin();
    Serial.println("Buscando dispositivos...");
    Serial.println("Encontrados: ");
    numeroSensoresConectados =sensorDS18B20.getDeviceCount();
    Serial.print(numeroSensoresConectados);
    Serial.println(" sensores");
}
```

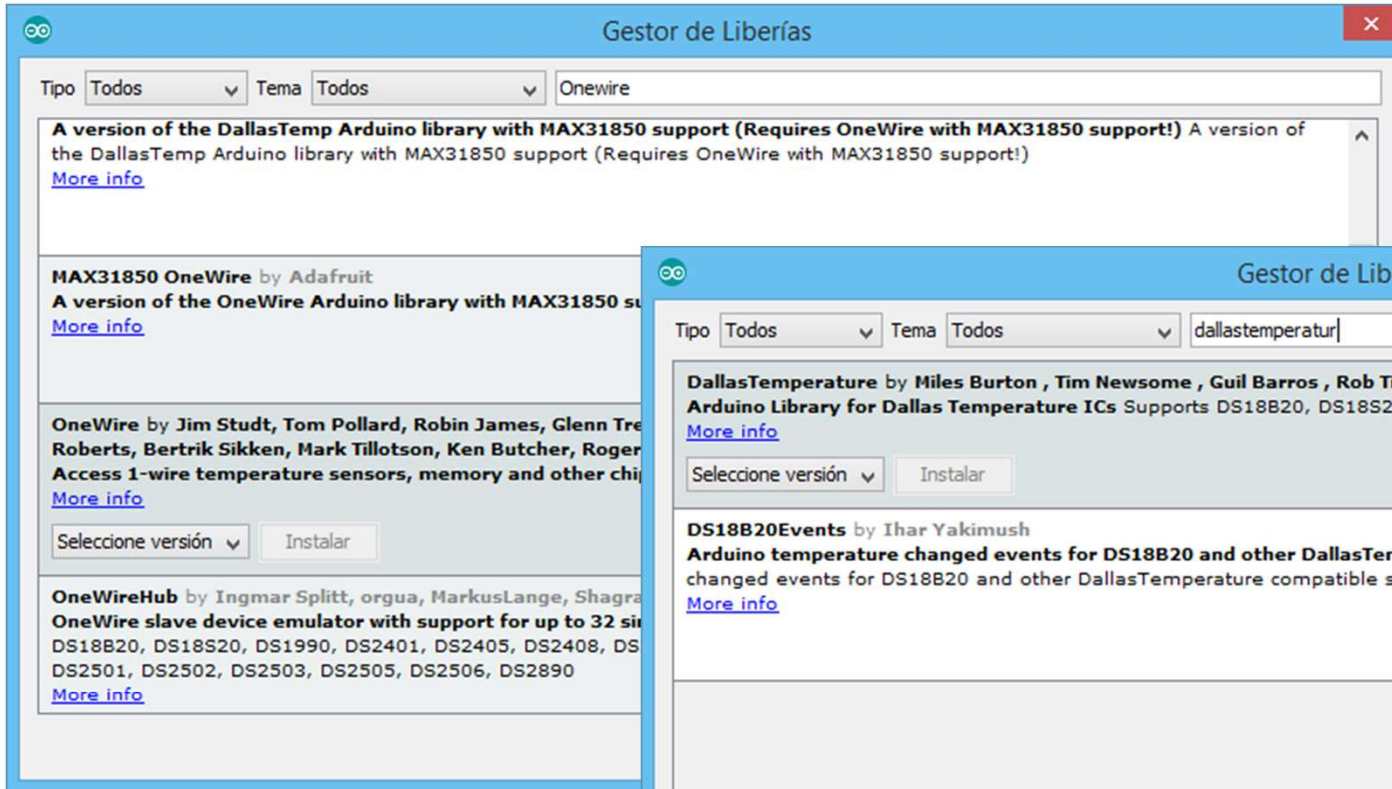


Sensor 18B20

Hacer Ejemplo:  
Temperatura\_D18B20

# Incluir librerías

Sensor 18B20



Gestor de Liberías

Tipo Todos Tema Todos Onewire

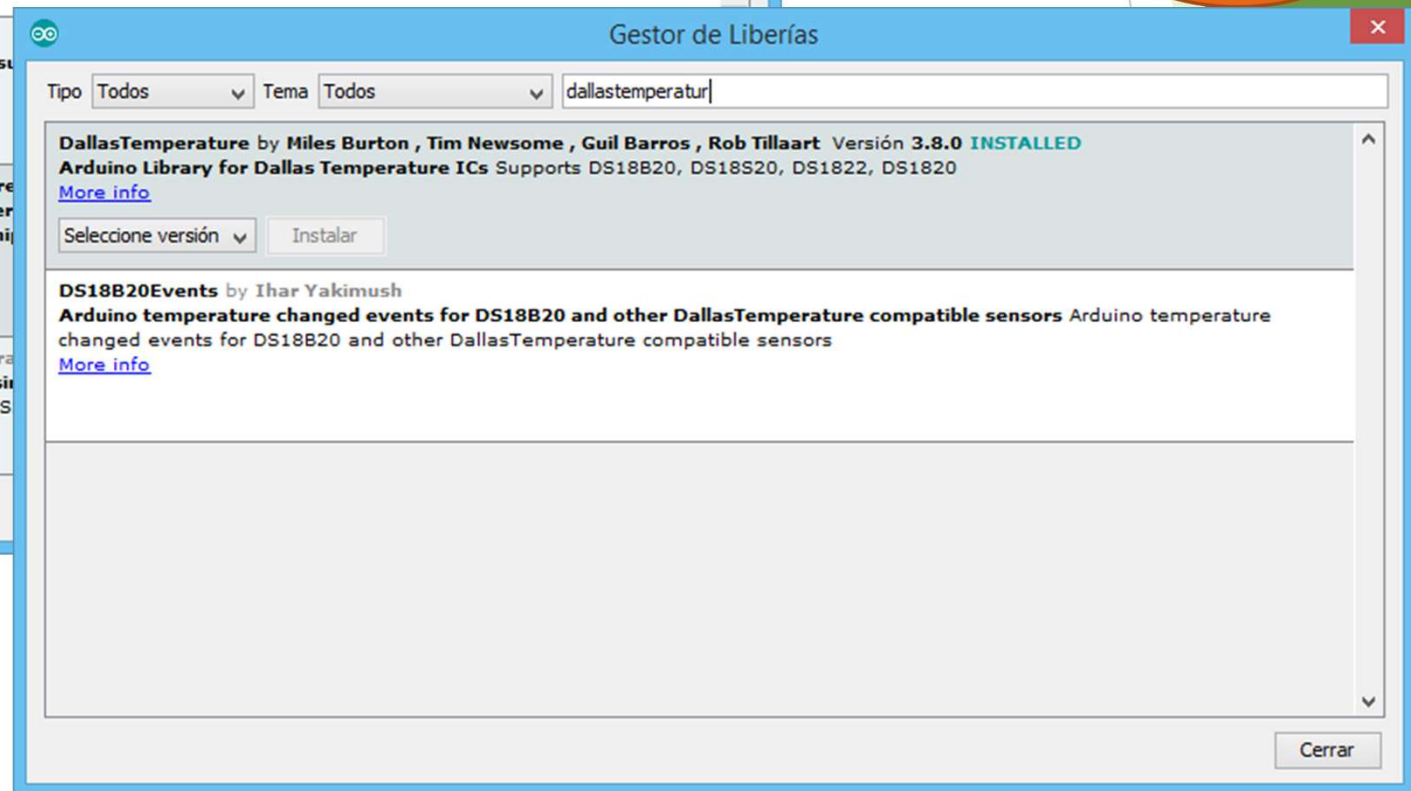
A version of the DallasTemp Arduino library with MAX31850 support (Requires OneWire with MAX31850 support!) A version of the DallasTemp Arduino library with MAX31850 support (Requires OneWire with MAX31850 support!)  
[More info](#)

**MAX31850 OneWire** by Adafruit  
A version of the OneWire Arduino library with MAX31850 support  
[More info](#)

**OneWire** by Jim Studt, Tom Pollard, Robin James, Glenn Trew Roberts, Bertrik Sikken, Mark Tillotson, Ken Butcher, Roger Cook  
Access 1-wire temperature sensors, memory and other chips  
[More info](#)

Seleccione versión Instalar

**OneWireHub** by Ingmar Splitt, orgua, MarkusLange, Shagrat  
OneWire slave device emulator with support for up to 32 sensors  
DS18B20, DS18S20, DS1990, DS2401, DS2405, DS2408, DS24C01, DS24C02, DS24C04, DS24C08, DS24E01, DS2501, DS2502, DS2503, DS2505, DS2506, DS2890  
[More info](#)



Gestor de Liberías

Tipo Todos Tema Todos dallastemperatur

**DallasTemperature** by Miles Burton, Tim Newsome, Guil Barros, Rob Tillaart Versión 3.8.0 **INSTALLED**  
Arduino Library for Dallas Temperature ICs Supports DS18B20, DS18S20, DS1822, DS1820  
[More info](#)

Seleccione versión Instalar

**DS18B20Events** by Ihar Yakimush  
Arduino temperature changed events for DS18B20 and other DallasTemperature compatible sensors Arduino temperature changed events for DS18B20 and other DallasTemperature compatible sensors  
[More info](#)

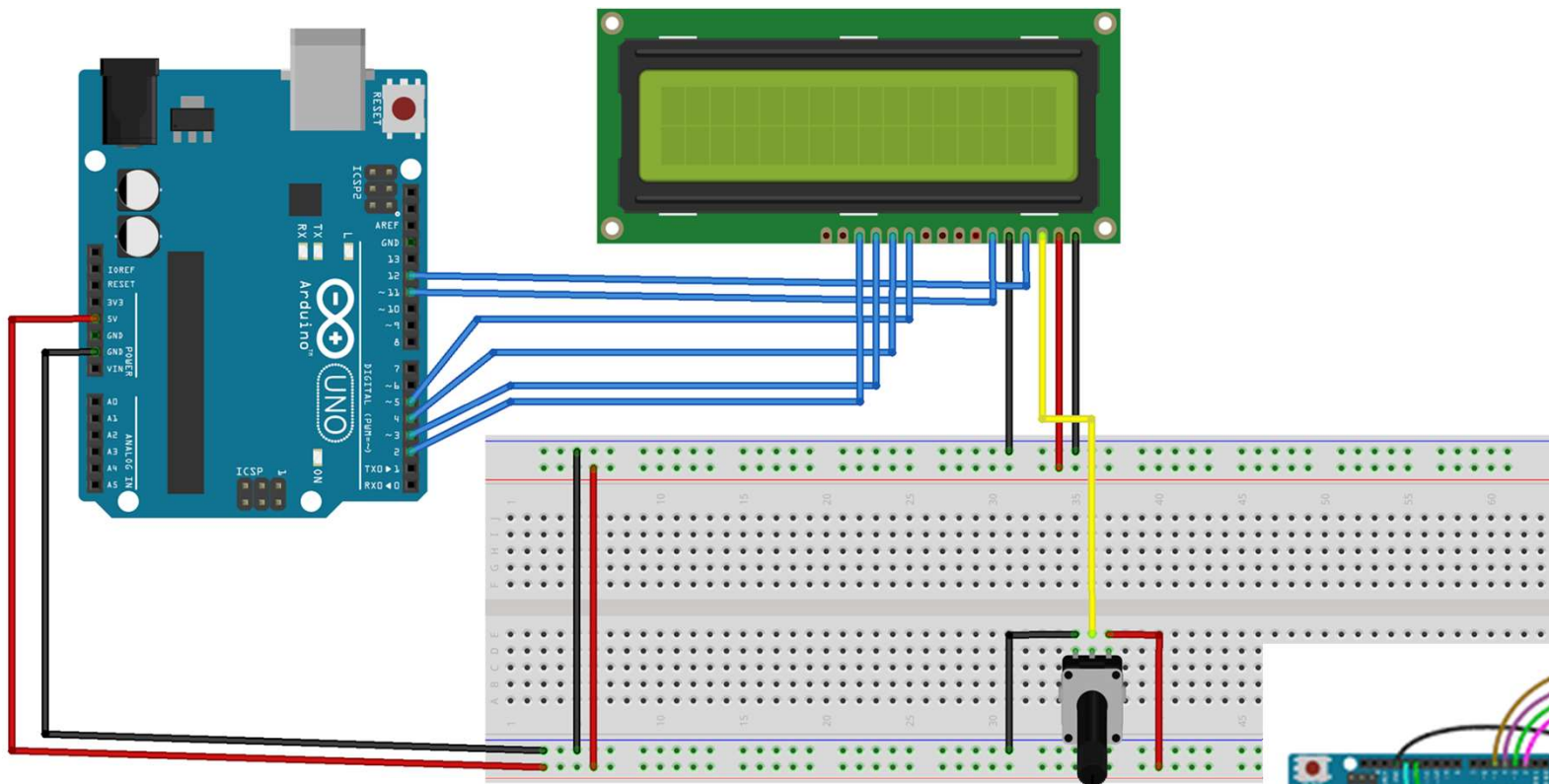
Cerrar

## Pantalla LCD

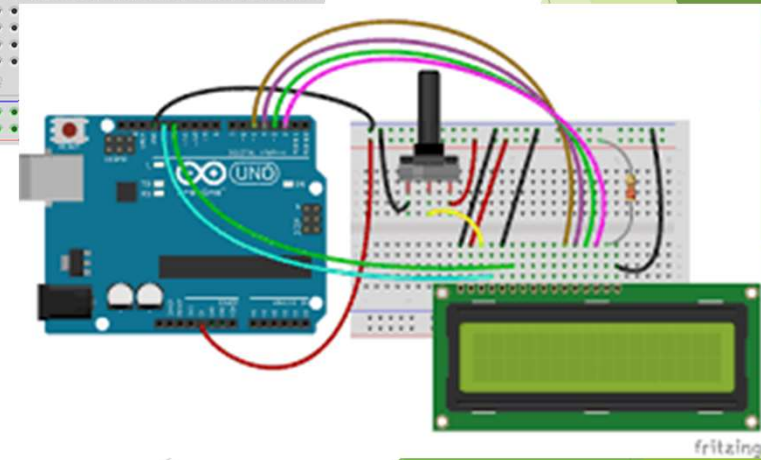
Por sus siglas del inglés Liquid Cristal Display, para el ejemplo se tiene un pantalla del tipo 16x1

No.	Symbol	Level	Function
1	V <sub>ss</sub>	--	Power Supply
2	V <sub>dd</sub>	--	
3	V <sub>0</sub>	--	
4	RS	H/L	Register Select: H:Data Input L:Instruction Input
5	R/W	H/L	H--Read L--Write
6	E	H.H-L	Enable Signal
7	DB0	H/L	Data bus used in 8 bit transfer
8	DB1	H/L	
9	DB2	H/L	
10	DB3	H/L	
11	DB4	H/L	Data bus for both 4 and 8 bit transfer
12	DB5	H/L	
13	DB6	H/L	
14	DB7	H/L	
15	BLA	--	BLACKLIGHT +5V
16	BLK	--	BLACKLIGHT 0V-

Guardar el dato en una memoria SD, con un reloj RTC, visualizarlo en una pantalla LCD y en una Terminal Serial.



Guardar el dato en una memoria SD, con un reloj RTC, visualizarlo en una pantalla LCD y en una Terminal Serial.



fritzing

## Ejemplo:3\_ LCD

```
#include <LiquidCrystal.h>
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
void setup() {
  lcd.begin(8,2);
  lcd.setCursor(0, 0);
  lcd.print(" hello");
}

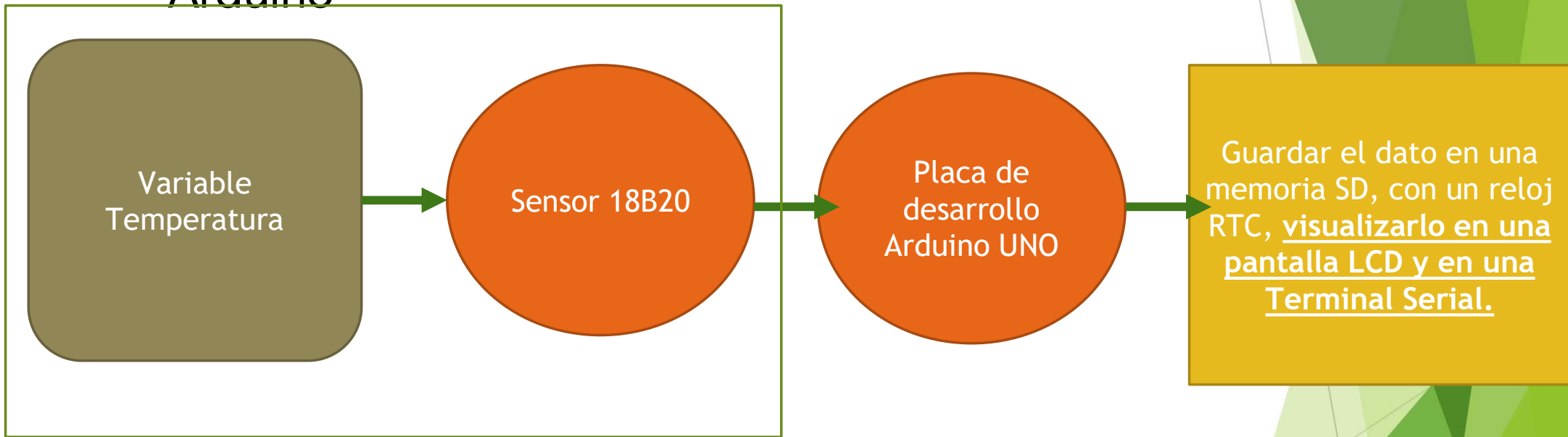
void loop() {
  lcd.setCursor(0, 1);
  lcd.print(millis() / 1000);
}
```

Guardar el dato en una memoria SD, con un reloj RTC, visualizarlo en una pantalla LCD y en una Terminal Serial.

Incluir librería desde el administrador de bibliotecas  
LiquidCrystal.h

Hacer Ejemplo: LCD\_1\_16

# Proyecto del Curso Básico de Instrumentación con Arduino



Ver ejemplo protoboard

PROYECTO PAPIME **PE207521**



Gracias!!!