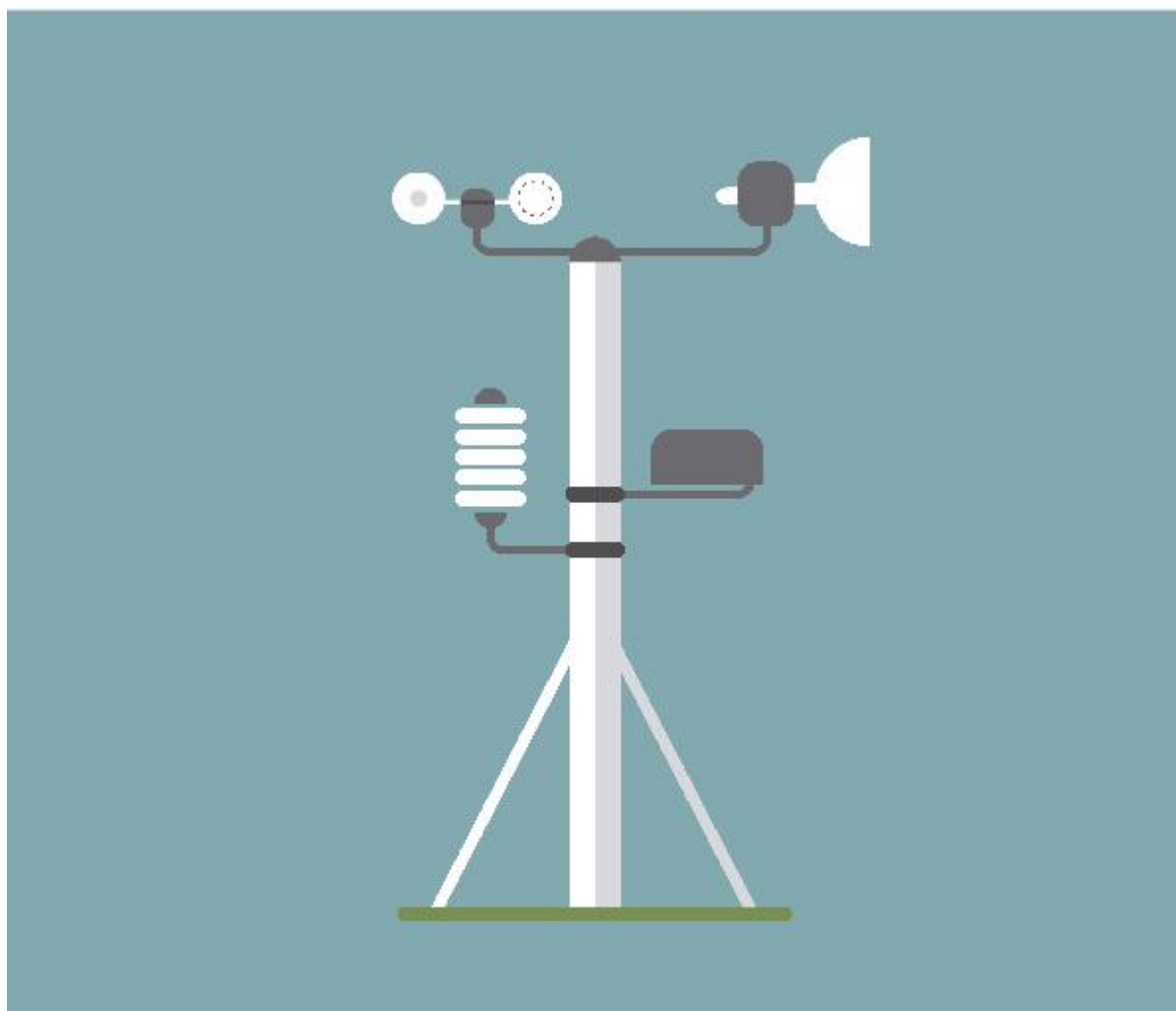


# SABERES DIGITALES

---



ESTACIÓN METEOROLÓGICA AUTOMATIZADA

---

# AUTORIDADES

**Presidente de la Nación**

Mauricio Macri

**Vicepresidenta de la Nación**

Marta Gabriela Michetti

**Jefe de Gabinete de Ministros**

Marcos Peña

**Ministro de Educación, Cultura, Ciencia y Tecnología**

Alejandro Finocchiaro

**Titular de la Unidad de Coordinación General del  
Ministerio de Educación, Cultura, Ciencia y Tecnología**

Manuel Vidal

**Subsecretario de Coordinación Administrativa**

Javier Mezzamico

**Director Ejecutivo INET**

Leandro Goroyesky

**Gerenta General de EDUCAR Sociedad del Estado**

Liliana Casaleggio

**Directora Nacional de Asuntos Federales**

María José Licio Rinaldi

**Director Nacional de Educación Técnico - Profesional**

Fabián Prieto

**Coordinador de Secundaria Técnica**

Alejandro Anchava

**Responsable de Formación Docente Inicial y Continua INET**

Judit Schneider

**Coordinador General En FoCo**

Pablo Trangone

## ESTACIÓN METEOROLÓGICA AUTOMATIZADA

### AUTORIDADES

¡Error! Marcador no definido.

### ESTACIÓN METEOROLÓGICA AUTOMATIZADA

5

Ficha técnica

5

Presentación

6

#### Desarrollo

7

##### Nivel Inicial

7

**Paso 1 - Primer etapa: Display y sensor DHT11**

7

**Paso 2 - Conectar el módulo del *display* LCD**

7

**Paso 3 - Programar la lectura del sensor DHT11 y visualizar los datos en el display**

8

**Paso 4 - Agregamos el sensor de lluvia**

14

##### Nivel Intermedio

17

**Paso 2 - Subir el código a través de Arduino IDE.**

18

**Paso 3 - Incluir librería Adafruit BMP085**

19

##### Nivel Avanzado

22

**Paso 1 - Introducción a Internet de las Cosas (IoT)**

22

**Paso 2 - Crear un Panel de Control**

23

**Paso 3 - Conectar el módulo OBLOQ**

29

**Paso 4 - Programación IoT.**

29

##### Cierre

33

##### Glosario

34

##### Reconocimientos

¡Error! Marcador no definido.



# ESTACIÓN METEOROLÓGICA AUTOMATIZADA

## Ficha técnica

<b>Nivel educativo</b>	Secundario. Ciclo Básico.
<b>Descripción general</b>	Diseño, construcción y programación de una estación meteorológica automatizada.
<b>Niveles de complejidad</b>	<p>Nivel inicial: Instalación de sensores de temperatura, humedad y presencia de lluvia junto a un <i>display</i> LCD.</p> <p>Nivel intermedio: Se agrega medición de la presión atmosférica.</p> <p>Nivel avanzado: Se incorpora Internet de las Cosas (IoT) para monitorear a distancia los datos registrados.</p>
<b>Insumos</b>	<ul style="list-style-type: none"><li>• 1 x Placa Arduino UNO</li><li>• 1 x Protoboard</li><li>• 1 x Cable USB tipo B</li><li>• 1 x Fuente de 9V 1A</li><li>• 1 x Módulo OBLOQ IoT</li><li>• 1 x Sensor de lluvia</li><li>• 1 x Sensor de temperatura y humedad DHT11</li><li>• 1 x Sensor Presión BMP180</li><li>• 1 x Display LCD 2x16</li><li>• 20 cables dupont hembra hembra</li><li>• 20 cables dupont macho hembra</li><li>• Filamento para impresora 3D PLA</li></ul>
<b>Herramientas</b>	<ul style="list-style-type: none"><li>• Soldador</li><li>• Estaño</li><li>• Alicata</li><li>• Pinza de punta</li><li>• Brusela</li></ul>

# Presentación

## **Descripción ampliada del proyecto**

Se propone la elaboración de una estación meteorológica automatizada mediante el uso de una placa Arduino y distintos sensores para medir las condiciones climáticas.

En el nivel inicial, se instalarán sensores de la temperatura del ambiente, de la humedad relativa y de la presencia de lluvia. Los datos obtenidos con los sensores se mostrarán en un *display* LCD.

En el nivel intermedio, se agrega la medición de la presión atmosférica. En el nivel más avanzado, se incorpora también Internet de la Cosas (IoT) para monitorear, a distancia, los datos medidos por los sensores.

Al final de esta guía se puede encontrar un glosario donde se provee la información técnica necesaria para poder poner el proyecto en funcionamiento. El mismo cuenta con aclaraciones sobre los diversos elementos electrónicos involucrados así como también conceptos claves.

## **Objetivos**

- Aproximarse al conocimiento y el manejo de distintos componentes electrónicos mediante la construcción de una estación meteorológica automatizada.
- Conocer los componentes de la interfaz de Arduino.
- Familiarizarse con el uso de diferentes sensores que miden las condiciones climáticas y analizar los datos obtenidos por ellos.
- Aproximarse al funcionamiento de las impresoras y programas de impresión 3D.
- Utilizar IoT (Internet de las Cosas) para registrar y monitorear los datos obtenidos por los sensores en la estación meteorológica.

# Desarrollo

## Nivel Inicial

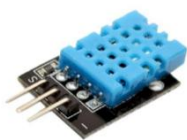
El Servicio Meteorológico Nacional realiza pronósticos y anuncia las condiciones climáticas para las diferentes las regiones del país. Esos datos son obtenidos en distintos lugares y son promediados para anunciar el pronóstico de áreas extensas. Por lo tanto, si quisiéramos obtener el estado del clima de una ubicación particular, como una casa, habría que buscar una forma propia para medir las condiciones climáticas. Los alumnos de una escuela de Santa Rosa, en La Pampa, se propusieron desarrollar una estación meteorológica en su escuela para registrar los datos de la humedad, de la temperatura y de la presencia de lluvia en el ambiente.

**En esta instancia del proyecto, se propone realizar una estación meteorológica conformada por sensores de humedad, de temperatura y de lluvia, conectados a una placa Arduino y monitoreados por un *display* LCD.**

## Paso 1 - Primer etapa: Display y sensor DHT11

Construiremos una estación meteorológica para monitorear la temperatura, la humedad y la lluvia, para lo cual se instalarán tres sensores correspondientes (Sensor de humedad y temperatura DHT11 y sensor de lluvia) conectados a una placa Arduino UNO y monitoreados mediante un *display* LCD.

Para realizar el prototipado y programación del sistema, vamos a realizar el montaje sobre una protoboard, ya que nos permite de forma rápida conectar los sensores antes de soldarlos de forma definitiva.



El DHT11 es un sensor que mide humedad y temperatura del aire. Este modelo de sensor posee electrónica interna que digitaliza los datos registrados y los reporta a el Arduino mediante una comunicación digital, por eso es que se conecta a un pin digital de la placa.

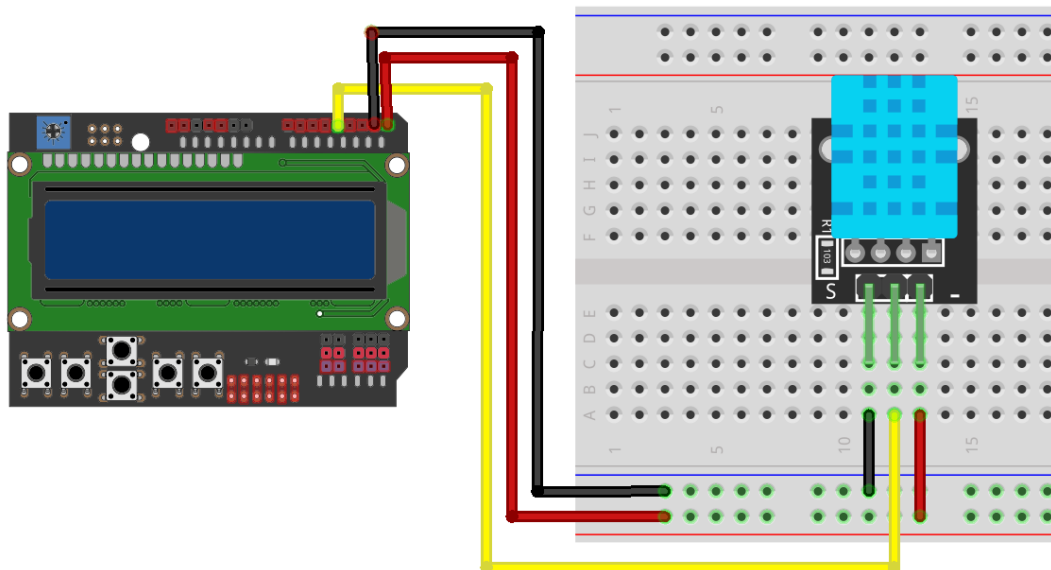
## Paso 2 - Conectar el módulo del *display* LCD

Dado que el display LCD con que trabajamos tiene formato tipo shield, se puede conectar al Arduino apilando y encastrando ambas placas. Es importante tener cuidado y no forzar las piezas, deben coincidir todos los pines perfectamente.



Los *shields* son placas de circuitos modulares que se montan unas encima de otras para agregar nuevas funcionalidades a la placa Arduino. Existen los que agregan funciones tales como comunicación, pantallas, sensores, interconexión, etc.

Conectamos el *display Shield* al Arduino. Luego, conectamos la señal del sensor de humedad y temperatura DHT11 al *pin 2* del *Shield*, con su correspondiente alimentación de energía (5V y GND), como indica el esquema.

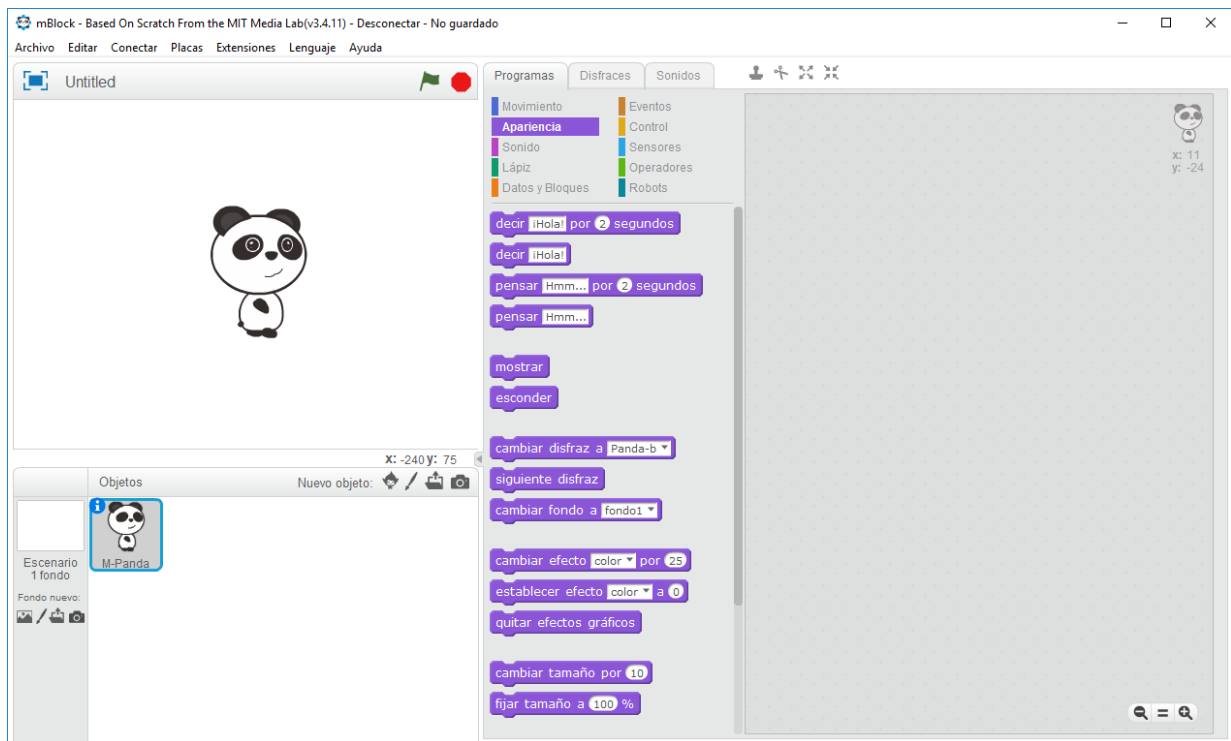


### Paso 3 - Programar la lectura del sensor DHT11 y visualizar los datos en el display

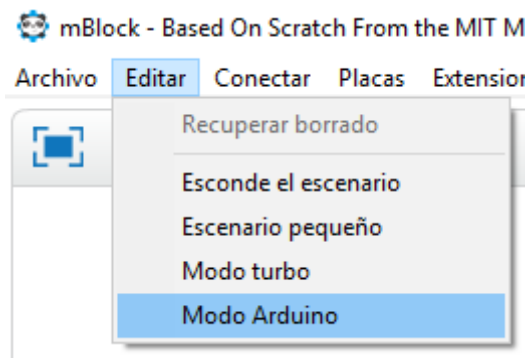
La programación la realizaremos con mBlock3, un entorno de programación basado en Scratch2 que permite programar proyectos de Arduino utilizando bloques. Se puede descargar siguiendo este enlace: <http://www.mblock.cc/software-1/mblock/mblock3/>

Cuando abrimos mBlock3, vemos una pantalla como la siguiente:

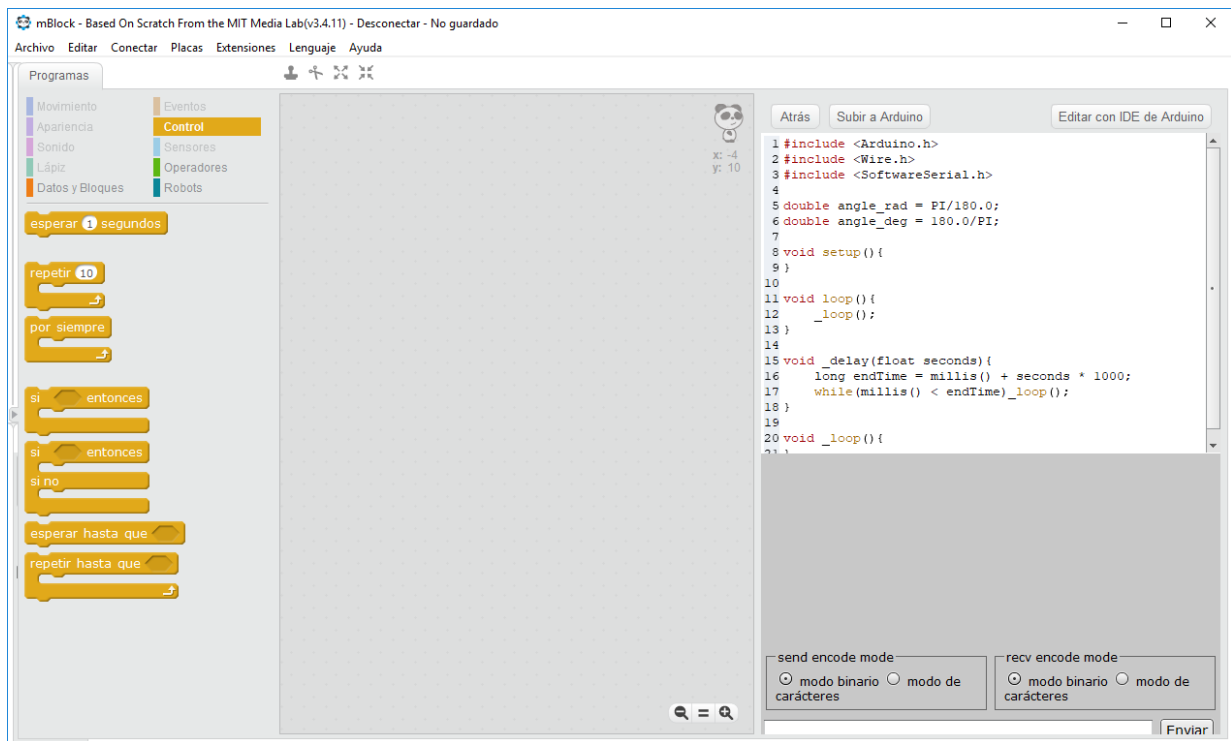




Para programar un proyecto de Arduino con mBlock3 debemos seleccionar el “Modo Arduino” desde el menú.



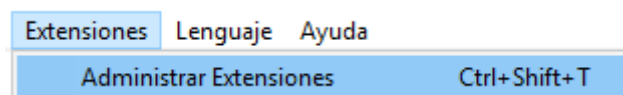
Al seleccionar este modo, el programa cambiará de aspecto. Se verá un área en el centro que es la que utilizaremos para programar con bloques. A la derecha se verá un campo donde aparecerá el código escrito que le corresponde a los bloques que están en el centro. Este código se irá escribiendo automáticamente a medida que se vaya armando el programa con los bloques.



Los bloques están agrupados por categorías. En este caso, se usarán bloques de las categorías **“Robots”**, **“Control”**, **“Operadores”** y **“Datos y Bloques”**. Cuando seleccionamos una de estas categorías, se pueden visualizar todos los bloques que pertenecen a ese grupo.



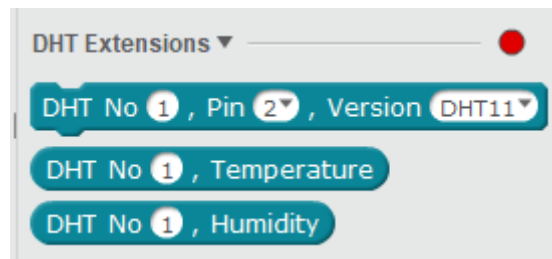
Para poder utilizar el sensor de temperatura y humedad, necesitamos instalar una extensión de mBlock3. Esto lo hacemos desde el menú:



Si en el buscador tipeamos **“dht”**, encontraremos el que precisamos entre los resultados, de esta manera: **“DHT Extensions”**.



Bajamos la librería, Veremos que contamos con nuevos bloques disponibles dentro de la categoría **“Robots”**.



Para programar el *display*, necesitamos instalar otra extensión. Como hicimos antes, buscamos la extensión en el administrador ingresando la palabra clave “lcd”. Encontraremos la extensión que necesitamos con el nombre **“LCD”**.



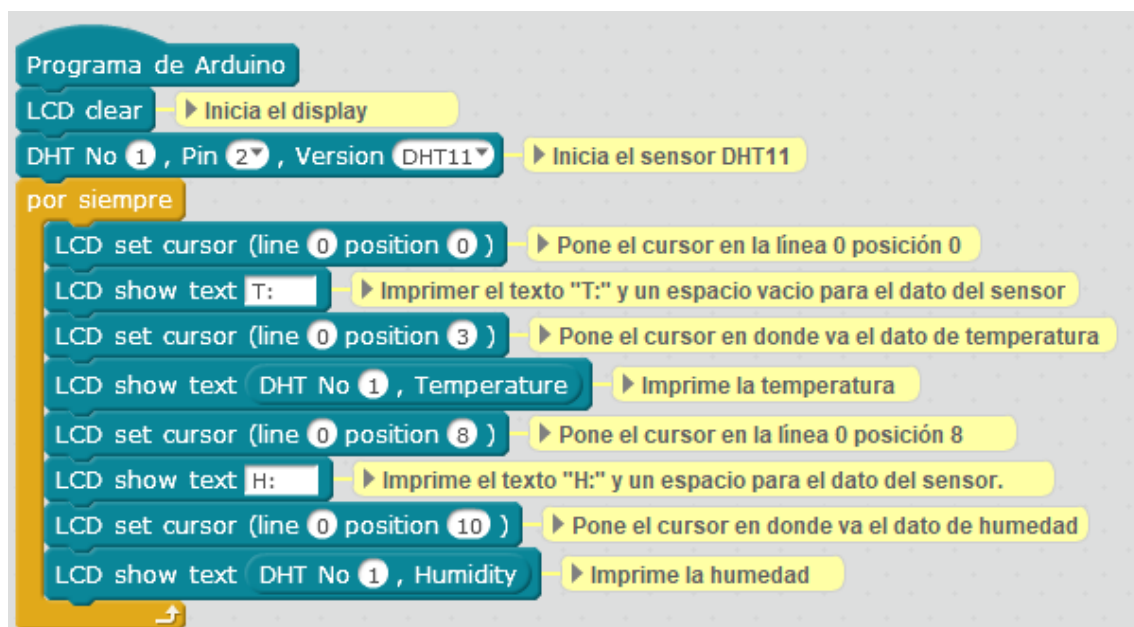
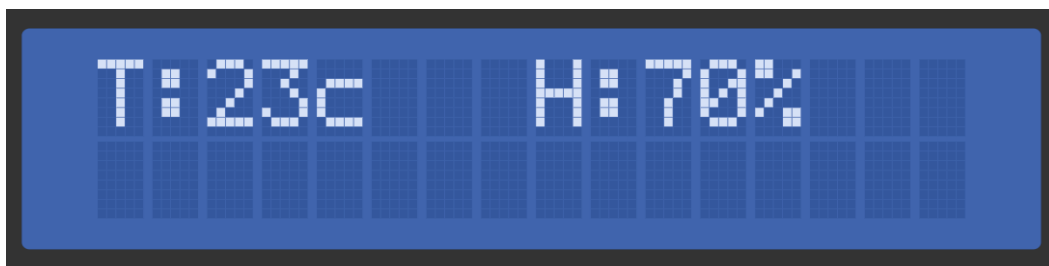
Luego de bajar la librería, veremos que contamos con nuevos bloques disponibles dentro de la categoría **“Robots”**.



Para visualizar en el *display* los datos obtenidos por el sensor, debemos armar el siguiente código de bloques.

Utilizaremos la primera fila del *display* para mostrar el valor de la temperatura y de la humedad. En el *display* se verá "T: " para la temperatura y "H:" para la humedad junto con el valor que le corresponde a cada una. Antes debemos instalar las extensiones para el sensor DHT11 y para nuestro display LCD.

El *display* se verá de esta manera:



Veremos que a la derecha se muestra el código escrito que corresponde a este programa.

```
#include <Arduino.h>
#include <Wire.h>
#include <SoftwareSerial.h>

#include "DHT.h"
#include <LiquidCrystal.h>
double angle_rad = PI/180.0;
double angle_deg = 180.0/PI;
LiquidCrystal lcd(8, 9, 4, 5, 6, 7);
DHT dht_1(2,11);

void setup(){
    lcd.begin(16, 2);
    lcd.clear();
}

void loop(){
    lcd.setCursor(0,0); //Posición de la temperatura en el display.
    lcd.print("T:    ");
    lcd.setCursor(3,0);
    lcd.print(dht_1.readTemperature()); //Mostramos el valor de la
temperatura en el display.
    lcd.setCursor(8,0); //Posición de la humedad en el display.
    lcd.print("H:    ");
    lcd.setCursor(10,0);
    lcd.print(dht_1.readHumidity()); //Mostramos el valor de la humedad
en el display.

    _loop();
}

void _delay(float seconds){
    long endTime = millis() + seconds * 1000;
    while(millis() < endTime)_loop();
}

void _loop(){
}
```

## Paso 4 - Agregamos el sensor de lluvia

Armaremos un programa para que el sensor detecte la presencia de lluvia y para que en la segunda fila del *display* se muestre "Lluvia: SI" o "Lluvia: NO", dado el caso correspondiente. Conectamos el sensor de lluvia como indica el esquema.

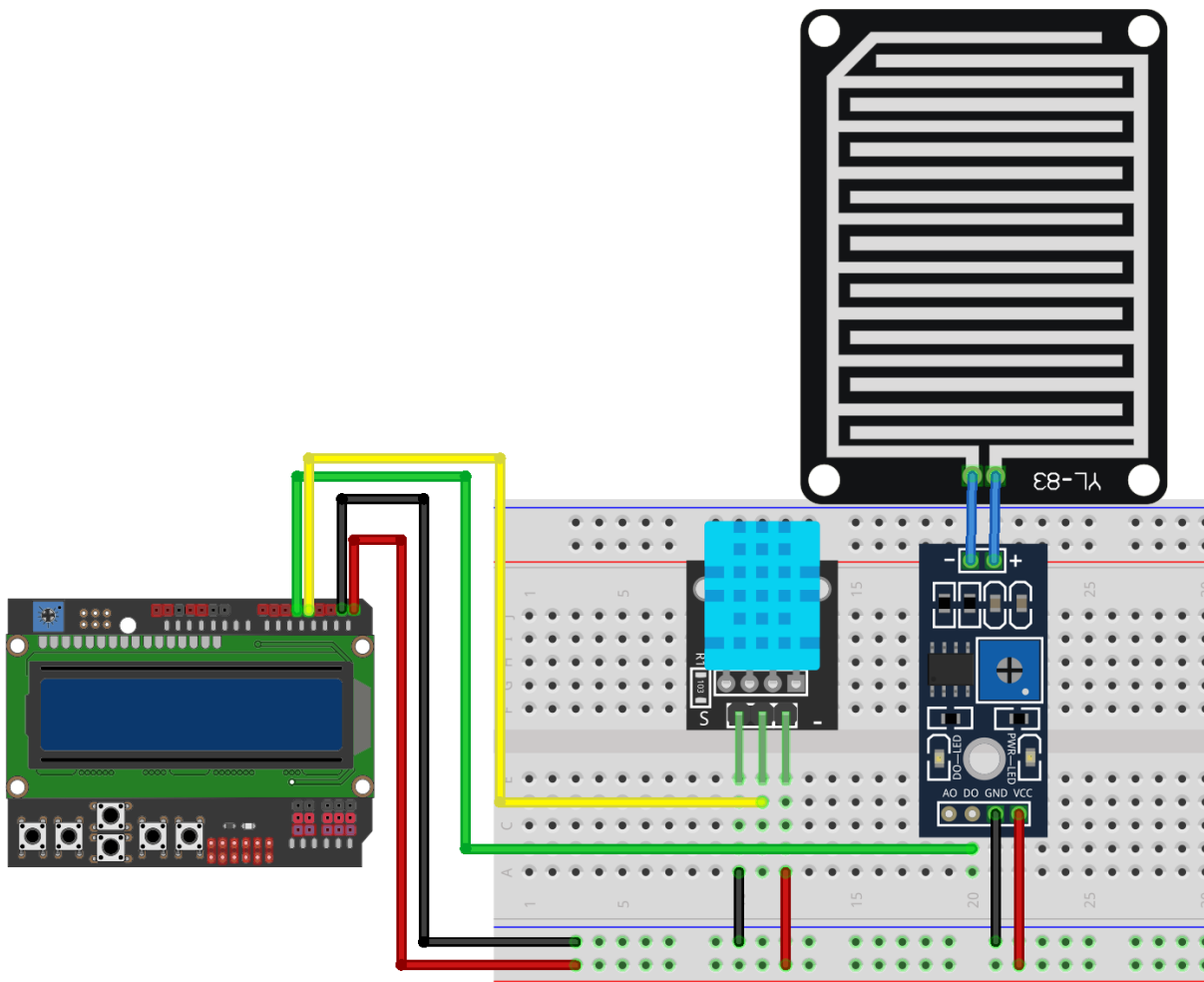


### Sensor de lluvia

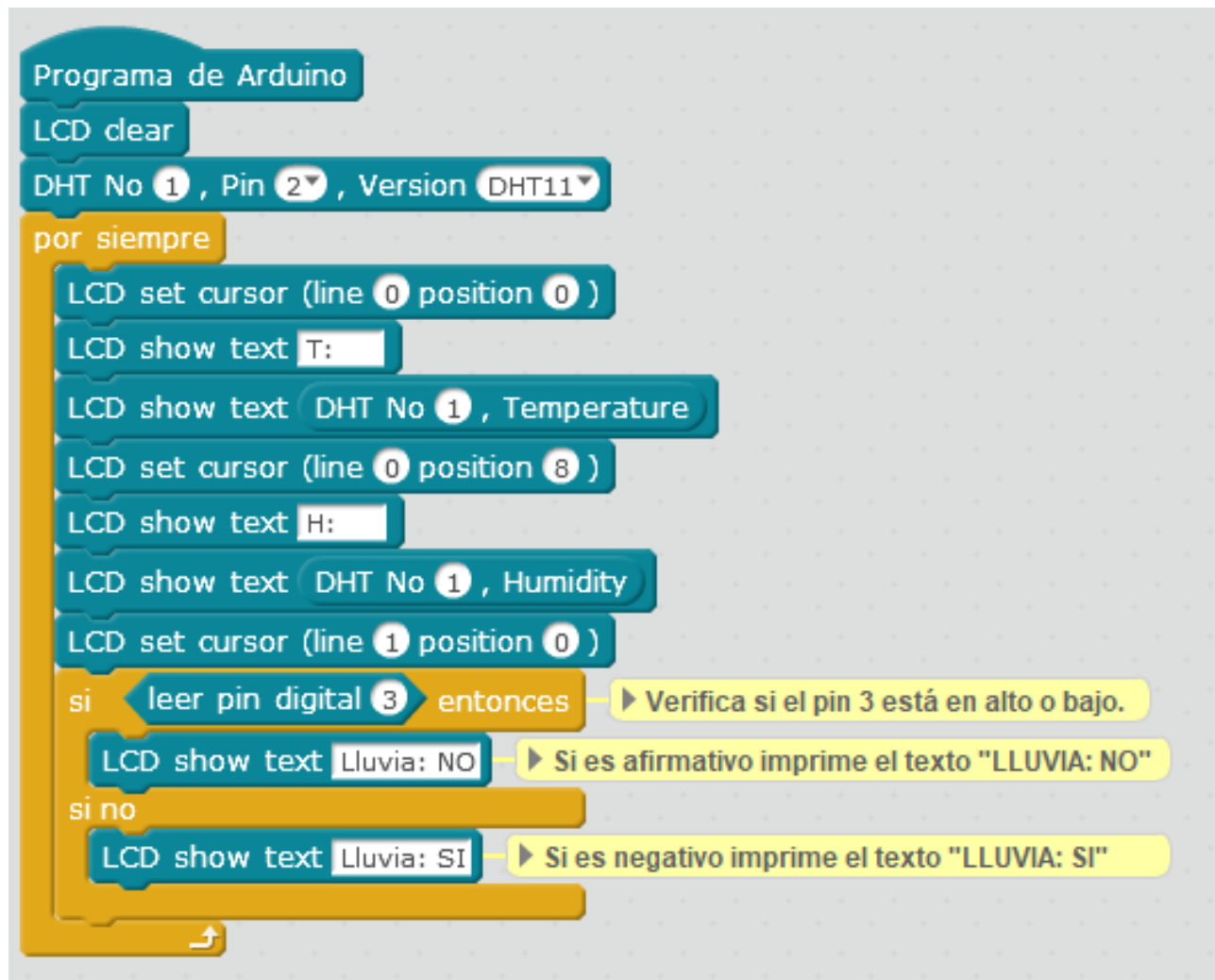
Los sensores de lluvia detectan la presencia de lluvia en el ambiente por la variación de conductividad que sufre el sensor. El mismo consta de dos cintas conductoras separadas que cuando se deposita agua se cierra el circuito y aumenta la conductividad.

Antes de realizar la lectura del sensor vamos a calibrarlo utilizando el potenciómetro de color azul que se encuentra en el circuito electrónico. Este se complementa al panel conductivo del sensor.

Para calibrarlo, hay que tener en cuenta que, al momento de detectar agua en el panel la salida digital D0 del sensor se debe encontrar en el estado BAJO (0V); y, cuando no se detecta lluvia, la salida digital del sensor se debe encontrar en el estado ALTO (5V).



Para visualizar en el display los datos obtenidos por el sensor de lluvia, debemos armar el siguiente código de bloques:



Veremos que a la derecha se muestra el código escrito que corresponde a este programa.

```
#include <Arduino.h>
#include <Wire.h>
#include <SoftwareSerial.h>

#include "DHT.h"
#include <LiquidCrystal.h>
double angle_rad = PI/180.0;
double angle_deg = 180.0/PI;
LiquidCrystal lcd(8, 9, 4, 5, 6, 7);
DHT dht_1(2,11);

void setup(){
  lcd.begin(16, 2);
```

```

    lcd.clear();
    pinMode(3,INPUT);
}

void loop(){
    lcd.setCursor(0,0);
    lcd.print("T:    ");
    lcd.print(dht_1.readTemperature());
    lcd.setCursor(8,0);
    lcd.print("H:    ");
    lcd.print(dht_1.readHumidity());
    lcd.setCursor(0,1);
    if(digitalRead(3)){
        lcd.print("Lluvia: NO");
    }else{
        lcd.print("Lluvia: SI");
    }
    _loop();
}

void _delay(float seconds){
    long endTime = millis() + seconds * 1000;
    while(millis() < endTime)_loop();
}

void _loop(){
}

```



## Nivel Intermedio

Uno de los parámetros importantes a la hora de monitorear el estado del tiempo es la presión atmosférica. Por lo general, las bajas presiones están asociadas a condiciones inestables del tiempo y, ocasionalmente, a lluvia y la presión alta se asocia con condiciones agradables.

Por eso, los chicos consideraron que era importante que la estación meteorológica de la escuela también mida la presión atmosférica.

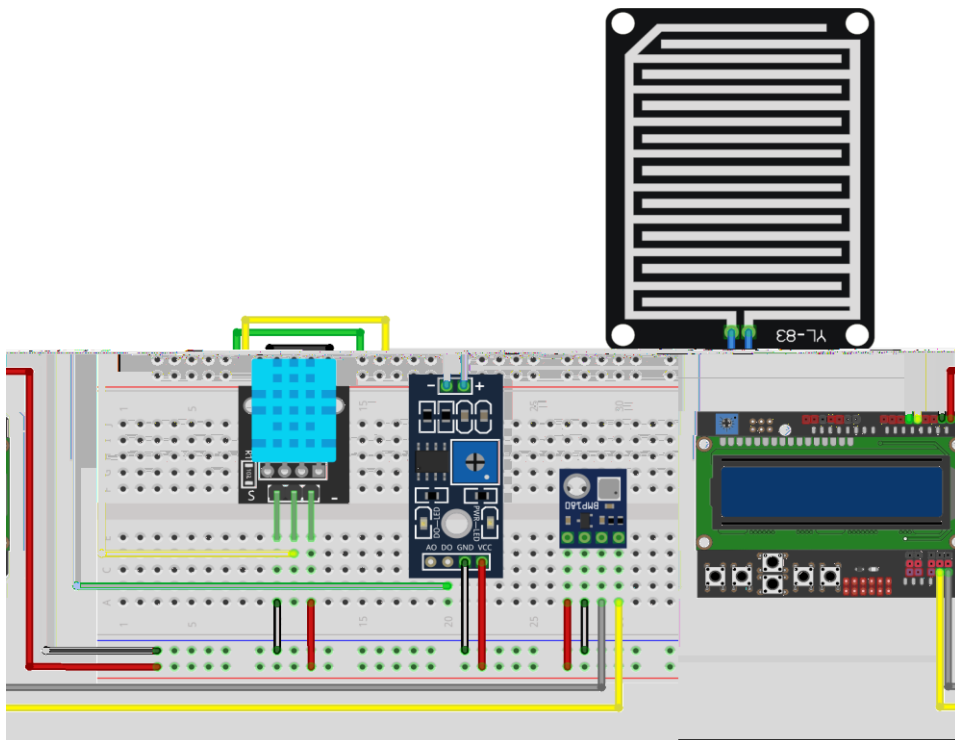
**En este nivel sumaremos a nuestra estación meteorológica un sensor que mide la presión atmosférica.**

### Paso 1 - Conexión del sensor de presión atmosférica BMP180

En el nivel inicial armamos una estación meteorológica que mide la temperatura, la humedad y la presencia de lluvia. Ahora le sumaremos un sensor que mide la presión atmosférica, el sensor BMP180. La conexión será mediante el I2C de la placa Arduino. Estos *pines* corresponden a los ubicados en el A4 (SDA) y A5 (SCL) y a su respectiva alimentación de energía (5V y GND), como indica el esquema:



El sensor de presión barométrica BMP180 está diseñado para leer la presión atmosférica. Se trata de la fuerza que ejerce el aire (atmósfera) sobre la superficie de la tierra.



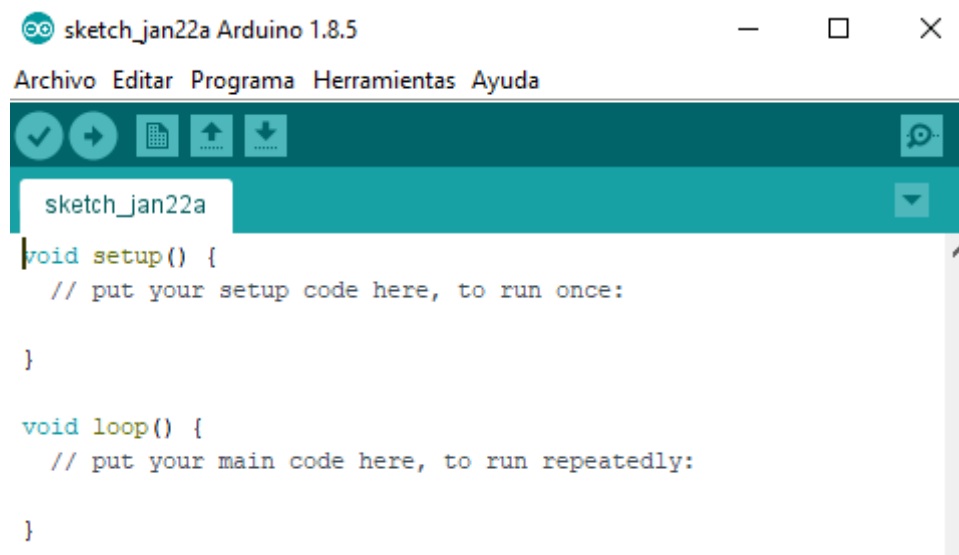
## Paso 2 - Subir el código a través de Arduino IDE.

La programación por bloques tiene sus ventajas desde un punto de vista didáctico pero cuando el programa crece en complejidad puede resultar poco práctico. A menudo podemos encontrarnos con el hecho de que ciertas operaciones no pueden resolverse utilizando bloques o que hacerlo con este método resulta más engorroso y difícil de interpretar que si se utilizara el código escrito.

Hasta ahora hemos visto cómo al realizar nuestra programación en bloques se generaba simultáneamente un código escrito en el área lateral derecha. Para esta sección de la actividad se propone trabajar directamente sobre el código, para ello vamos a recurrir a el entorno nativo de Arduino que llamamos “Arduino IDE” (IDE proviene la siga entorno de desarrollo integrado).

Para ello descarga el Arduino IDE desde el siguiente enlace y luego procede con la instalación del mismo: [www.enfoco.net.ar/sd](http://www.enfoco.net.ar/sd)

Veremos que se nos presenta la siguiente interfaz:



A continuación, se presenta una estructura mínima de un *sketch* (un programa) de Arduino:

```
void setup() {  
  // Código de inicialización. Se ejecuta una sola vez.  
}  
  
void loop() {  
  // Código principal. Se ejecuta repetidamente.  
}
```

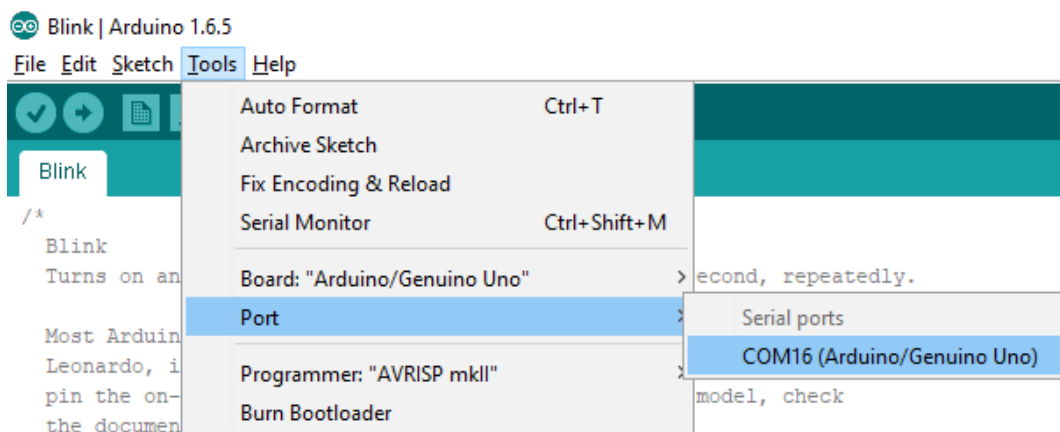
En líneas generales, un programa de Arduino es:

1. Un bloque de código que se ejecuta por única vez al inicializarse el dispositivo. Este bloque de código está contenido dentro de la función “setup” (se coloca dentro de `void setup() { y }` ).
2. Un bloque de código que se ejecuta repetidamente luego de la función “setup”. Este bloque de código está contenido dentro de la función “loop” (se coloca dentro de `void loop() { y }` ).

Después de `//` se incluyen comentarios para el lector que no tienen ningún efecto en el programa. Estos comentarios sirven para clarificar el código y que sea más fácil de interpretar para otras personas.

Los pasos para subir el código a través del Arduino IDE son similares a los que hemos visto para mBlock3:

1. Conectar la placa a la entrada USB.
2. Chequear que estén seleccionados la placa “Arduino/Genuino Uno” y el puerto serie al que está conectada la placa.



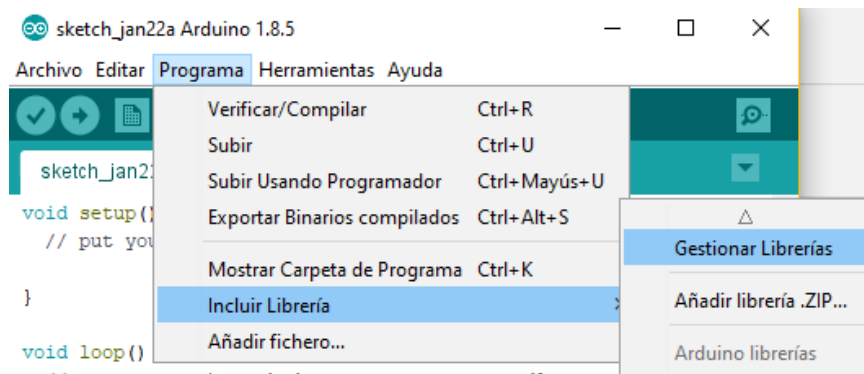
3. Clickear el botón de “Subir” .

Sabremos que nuestro código subió correctamente si en la barra de estado se escribe “Subido”.

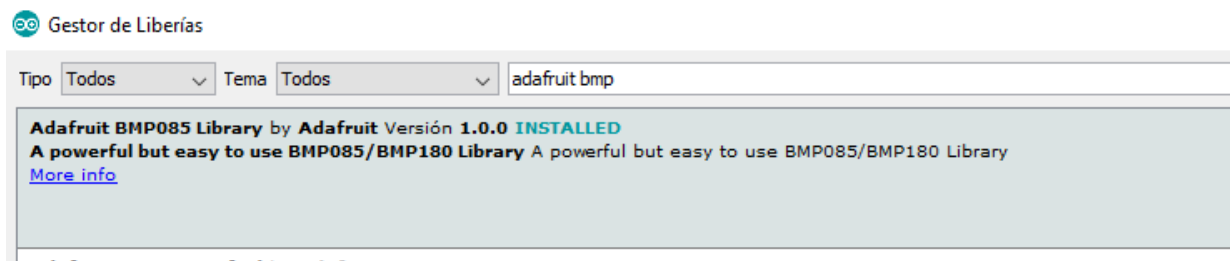
### Paso 3 - Incluir librería Adafruit BMP085

Es necesario instalar la librería “Adafruit BMP085” en nuestro programa Arduino. Para hacerlo, seleccionamos “Programa” en el menú. Luego, “Incluir librería”. Y, por último, “Gestionar librería”.

De este modo, vamos a buscar e instalar una nueva librería a nuestro Arduino; en este caso la librería “Adafruit BMP085”, que es necesaria para programar de forma sencilla nuestro sensor de presión.



Buscamos la librería “Adafruit BMP085” y seleccionamos “Instalar”.



La presión atmosférica se podrá ver junto al estado de la lluvia, identificada con la letra P. En nuestro caso, será “**P: 1023**”.



Nuestro código debería ser similar al siguiente modelo:

```
#include <Arduino.h>
#include <Wire.h>
#include <Adafruit_BMP085.h>
#include <SoftwareSerial.h>
#include <DHT.h>
#include <LiquidCrystal.h>

Adafruit_BMP085 bmp;
LiquidCrystal lcd(8, 9, 4, 5, 6, 7);
DHT dht_1(2,11);

void setup() {
```

```

Serial.begin(9600);
bmp.begin();
lcd.begin(16, 2);
lcd.clear();
pinMode(3, INPUT);

}

void loop() {
    bmp.readTemperature();
    long presion=bmp.readPressure()/100;//Medición de la presión.
    lcd.setCursor(0,0); //Posición para la temperatura en el display.
    lcd.print("T:");
    lcd.print(dht_1.readTemperature()); //Mostramos el valor de la
temperatura en el display.
    lcd.setCursor(8,0); //Posición de la humedad en el display.
    lcd.print("H:");
    lcd.print(dht_1.readHumidity()); //Mostramos el valor de la humedad en
display.
    lcd.setCursor(0,1); //Posición para lluvia en el display.
    if(digitalRead(3)){ //Condición del estado del sensor de lluvia.
        lcd.print("Lluvia:NO");
    }else{
        lcd.print("Lluvia:SI");
    }
    lcd.setCursor(10,1); //Posición para la presión en el display.
    lcd.print("P:");
    lcd.print(presion); //Mostramos el valor de la presión en el display.
}

```

## Nivel Avanzado

Ahora, los chicos quieren tener un registro histórico de los datos obtenidos en la estación meteorológica, para poder analizarlos y buscar regularidades y relaciones entre las variables del clima en su escuela. Para esto, decidieron instalar un sistema de Internet de las Cosas (IoT), que les permite registrar y enviar la información obtenida en su estación a un dispositivo móvil.

**En esta actividad se programará el envío de los datos obtenidos en la estación meteorológica a un dispositivo móvil a través de Internet de las Cosas (IoT).**

### Paso 1 - Introducción a Internet de las Cosas (IoT)

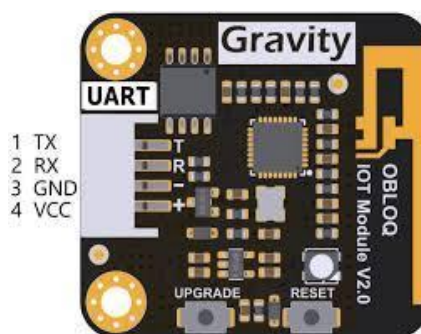
**Internet de las Cosas** (en inglés *Internet of Things*, abreviado IoT) es un concepto que refiere a la interconexión digital de objetos cotidianos con internet. Esta interconexión puede tener diversas funciones. Por ejemplo, puede utilizarse para monitorear la temperatura de un ambiente, enviando los datos obtenidos por un sensor a una central donde se recopile la información. De esta manera podría visualizarse en un dispositivo móvil la temperatura de un laboratorio, de un invernadero o de una sala de un hospital.

Para poder incorporar IoT a nuestro proyecto es necesario:

1. Un dispositivo capaz de conectarse a internet.
2. Un servidor que reciba y aloje los datos.

Existen diversas formas de lograr el cometido de registrar y almacenar los datos del sistema de tanques construido. En este caso, se detallará cómo hacerlo con un módulo OBLOQ de DFRobot, y con los servidores de Adafruit.

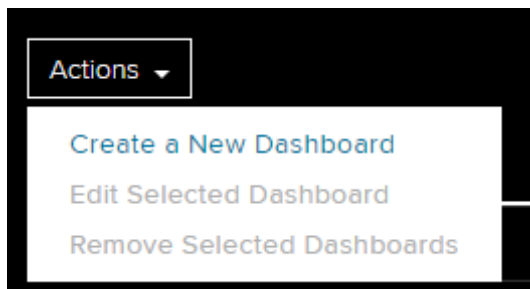
El módulo UART OBLOQ es un dispositivo WiFi a serie pensado para desarrolladores no profesionales. Permite enviar y recibir datos mediante los protocolos HTTP y MQTT.




## Paso 2 - Crear un Panel de Control


En primer lugar, se explicará cómo crear un Panel de Control en Adafruit. Luego, se verá cómo vincular los controles del Panel con los datos que se intercambian con el dispositivo.

Debemos crear una cuenta de usuario en [io.adafruit.com](https://io.adafruit.com). Una vez que ingresamos con nuestro usuario, creamos un nuevo panel haciendo click en "Create a New Dashboard".



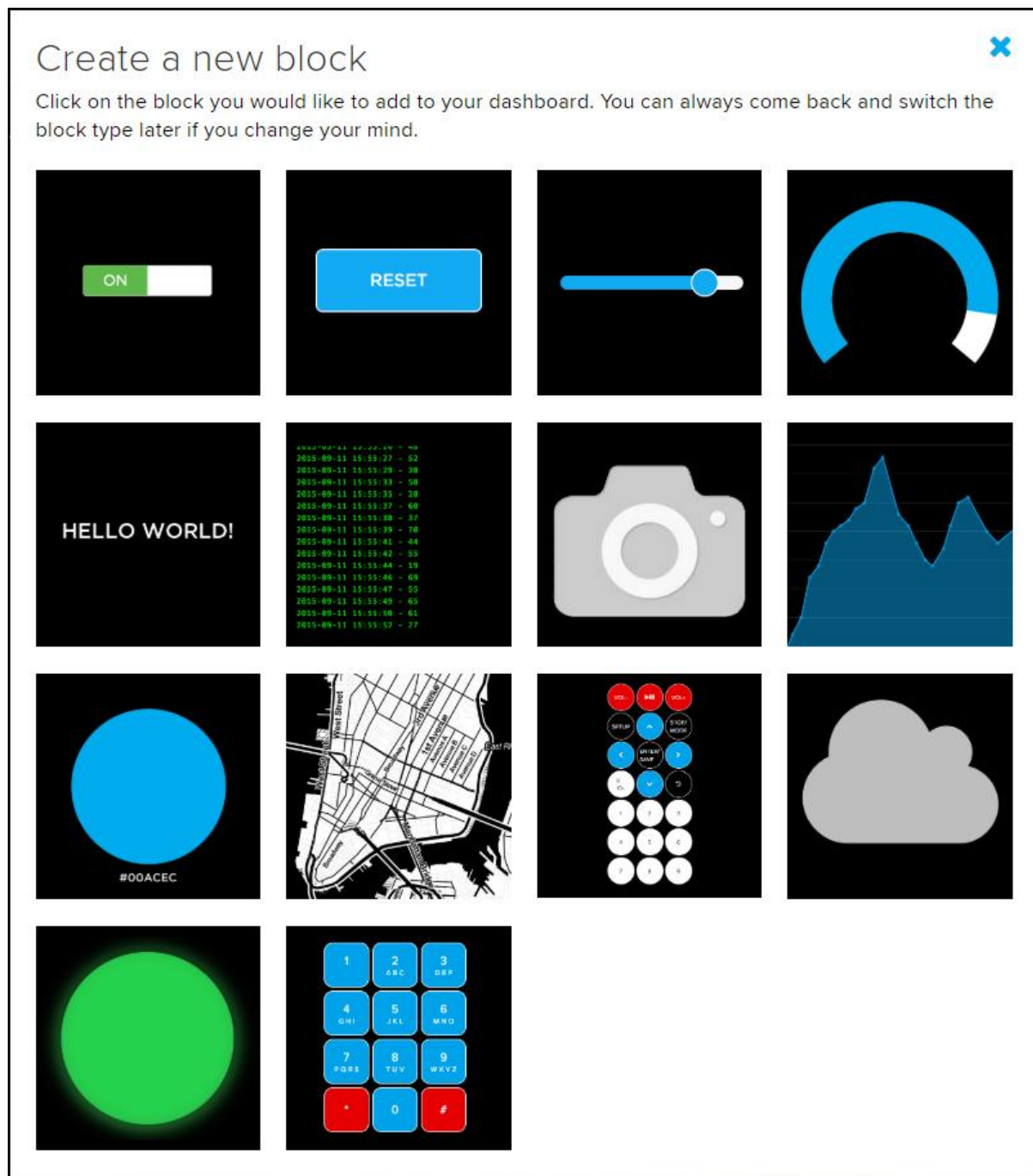
Creamos un nombre y una descripción.

A screenshot of a 'Create a new Dashboard' form. The title 'Create a new Dashboard' is at the top right with a blue close button (X). Below the title are two input fields. The first is labeled 'Name' and contains the text 'Estación Meteorológica'. The second is labeled 'Description' and contains the text 'Diseño y construcción de una estación meteorológica'. At the bottom right, there are two buttons: 'Cancel' (gray) and 'Create' (blue).

Hacemos click en el nuevo panel creado y veremos una pantalla vacía. Podemos comenzar a agregar bloques haciendo click en .



Veremos una serie de controles posibles, como en la siguiente imagen:



Para nuestra estación meteorológica, podríamos ubicar dos *Line Chart* (gráficos de línea) que nos permitan ver el historial de los cambios que surjan en la temperatura, la humedad y la presión atmosférica.





Cuando agregamos un control al panel debemos asociarlo a un “*feed*”.

Un *feed* es una fuente de datos en la que uno puede publicar así como también se puede suscribir para recibir los datos de cierto feed.

Llamamos al primer *feed* “temperatura”. En él publicaremos, desde nuestro dispositivo, el valor de la temperatura que lea el sensor DHT11.

## Choose up to 5 feeds



**Line Chart:** The line chart is used to graph one or more feeds.

If you have lot of feeds, you may want to use the search field. You can also create a feed quickly below.

Group / Feed	Last value	Recorded
<input type="checkbox"/> humedad	🔒	less than a minute
<input type="checkbox"/> presion	🔒	less than a minute
<input checked="" type="checkbox"/> temperatura	🔒	less than a minute

Luego de crearlo, hacemos click en “*Next step*” (paso siguiente) para configurar nuestro control y completamos los campos como se ve en la imagen a continuación:

## Block settings

In this final step, you can give your block a title and see a preview of how it will look. Customize the look and feel of your block with the remaining settings. When you are ready, click the "Create Block" button to send it to your dashboard.

Block Title (optional)

Show History

24 hours

X-Axis Label

Y-Axis Label

Y-Axis Minimum

Leave blank to automatically detect.

Y-Axis Maximum

Leave blank to automatically detect.

Decimal Places

Number of decimal places to display, defaults to 4.

Block Preview

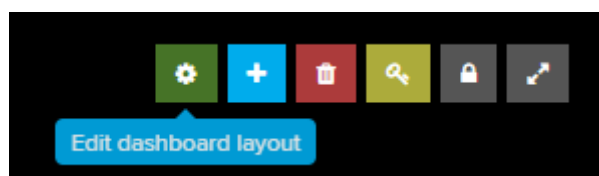
**Line Chart** The line chart is used to graph one or more feeds.

Previous step

Create block

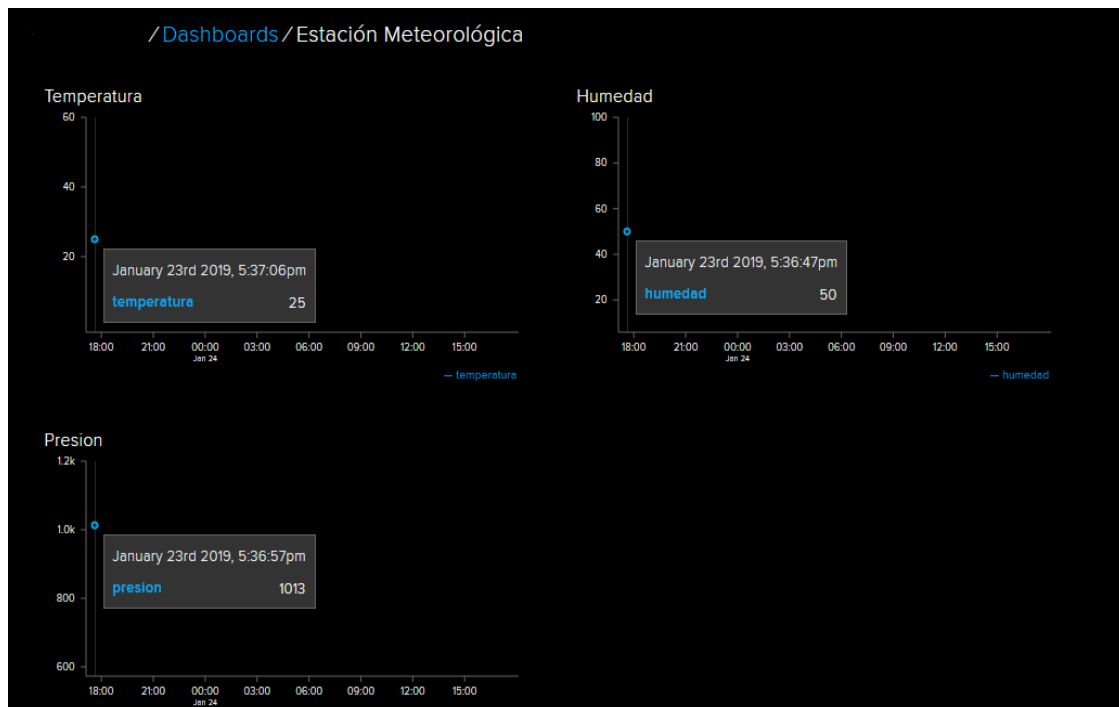
Hacemos click en "Create block" (crear bloque) para completar la operación.

Podemos modificar el tamaño y la ubicación de los bloques haciendo click en la "rueda de configuración".



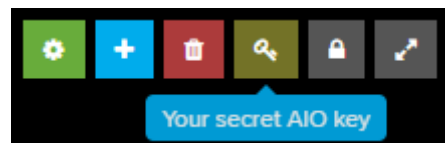
Repetimos este procedimiento para el *feed* "humedad" y "presión".

Deberíamos visualizar algo similar a lo siguiente, dado que aún no contamos con datos publicados:



Una vez realizado el Panel, programaremos nuestro sistema para que publique en él los datos que obtenga por medio de los sensores, lo que nos permitirá monitorearlos de manera remota.

Antes de salir, debemos copiar las credenciales de acceso para poder publicar en nuestros *feeds* “temperatura”, “humedad” y “presión”. Para ver nuestras credenciales, hacemos click en el ícono de la “llave”.



## YOUR AIO KEY

Your Adafruit IO key should be kept in a safe place and treated with the same care as your Adafruit username and password. People who have access to your AIO key can view all of your data, create new feeds for your account, and manipulate your active feeds.

If you need to regenerate a new AIO key, all of your existing programs and scripts will need to be manually changed to the new key.

Username

usuario\_aio

Active Key

1234cfdd29a244b6b049abb07727c117

REGENERATE AIO KEY

[Hide Code Samples](#)

Arduino

```
#define IO_USERNAME "usuario_aio"
#define IO_KEY "1234cfdd29a244b6b049abb07727c117"
```

Copiamos el código que nos ofrece para Arduino, con nuestro usuario y *key*. Más adelante se verá que estos datos aparecen en el código de la siguiente manera:

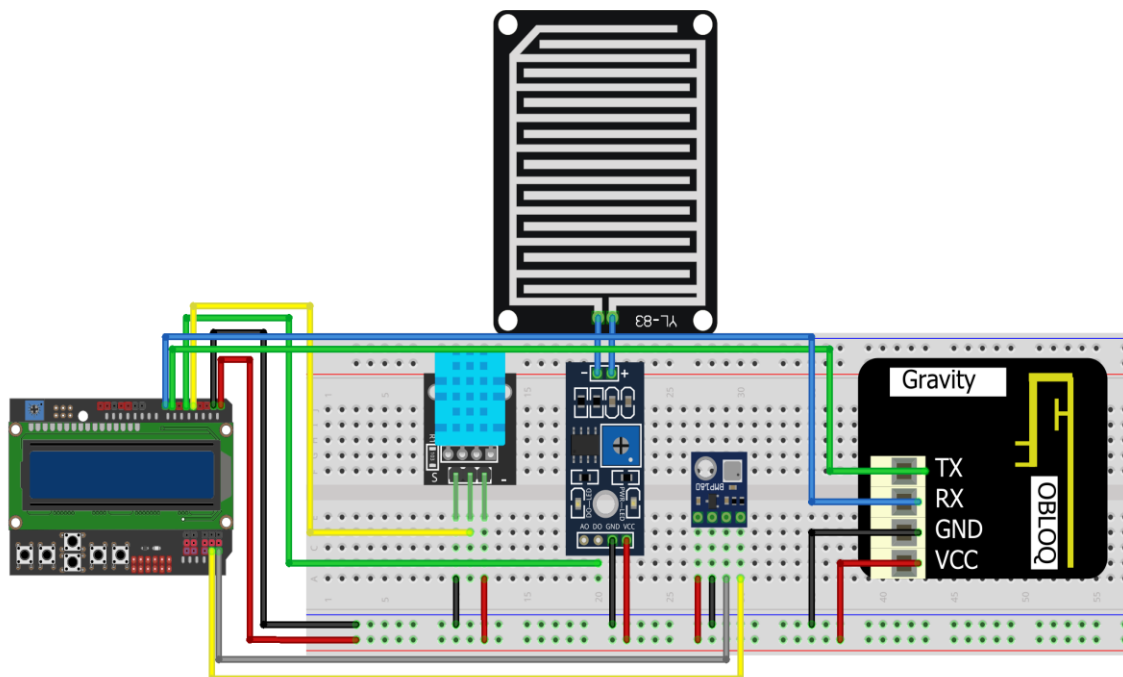
```
#define IO_USERNAME "usuario_adafruit"
#define IO_KEY "key_adafruit"
```

Se deberán reemplazar en esas dos líneas el usuario y *key* por los que se hayan obtenido en Adafruit. Por ejemplo:

```
#define IO_USERNAME "usuario_aio"
#define IO_KEY "1234cfdd29a244b6b049abb07727c117"
```

### Paso 3 - Conectar el módulo OBLOQ

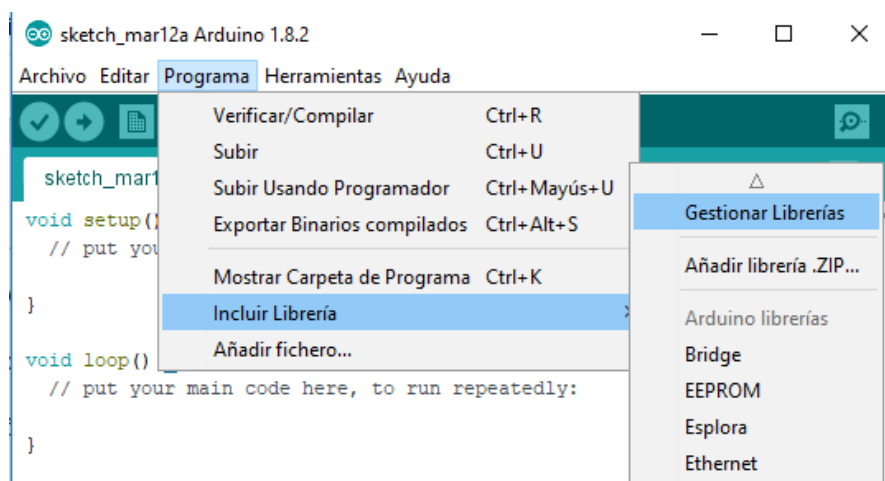
A continuación se muestra el esquema de conexión de Arduino y OBLOQ.



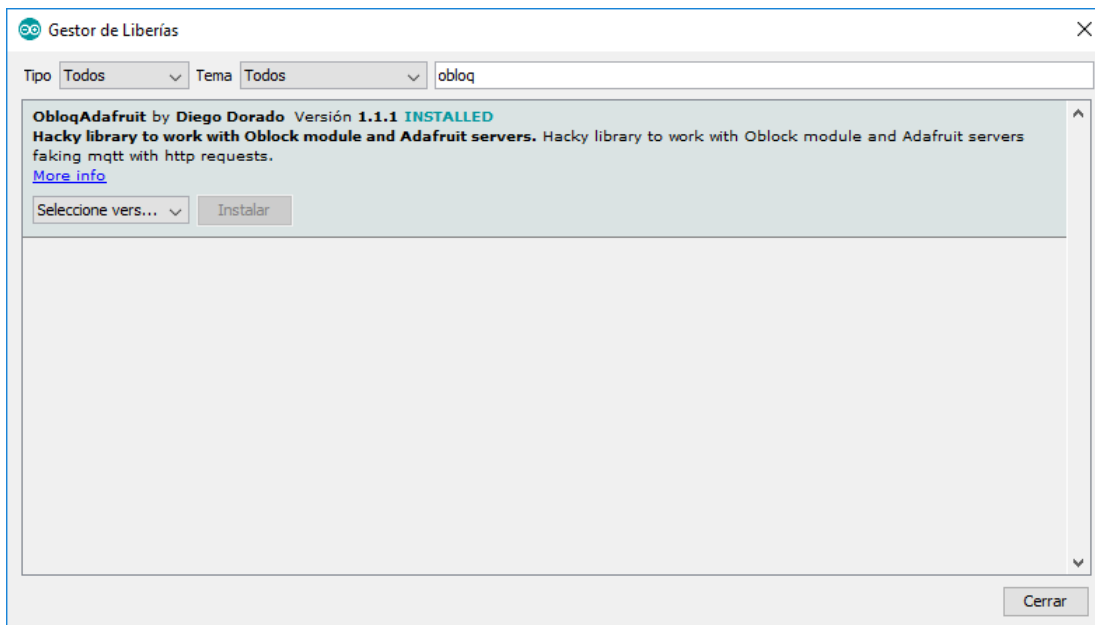
### Paso 4 - Programación IoT.

Utilizaremos la librería ObloqAdafruit para informar a Adafruit el estado de las variables de nuestra estación meteorológica. Podremos monitorearlo desde el Panel de Control que hemos creado.

En primer lugar debemos instalar la librería en el Arduino IDE. Para esto debemos ingresar al menú Programa > Incluir Librería > Gestionar Librerías.



Se abrirá una ventana con un buscador en margen superior. Debemos escribir Obloq, seleccionar la librería ObloqAdafruit y apretar el botón Instalar.



En general las librerías traen códigos de ejemplo como referencia. Abrimos el ejemplo “Publicar” ubicado en Archivo > Ejemplos > ObloqAdafruit > Publicar.

Debemos reemplazar el SSID de la WiFi, su password, el IO\_USERNAME e IO\_KEY por los que copiamos de Adafruit.

```
#include "SoftwareSerial.h"
#include "ObloqAdafruit.h"

// Indicamos conexión de wifi.
#define WIFI_SSID      "SSID_de_Wifi"
#define WIFI_PASSWORD  "PWD_de_WIFI"

// Copiamos las credenciales obtenidas anteriormente en Adafruit.
#define IO_USERNAME    "usuario_adafruit"
#define IO_KEY         "key_adafruit"

SoftwareSerial softSerial(12,13);
ObloqAdafruit olq(&softSerial,WIFI_SSID,WIFI_PASSWORD,IO_USERNAME,IO_KEY);
```

El *setup* debe incluir la línea de inicialización del softwareSerial:

```
void setup()
{
    softSerial.begin(9600);
}
```

Se debe agregar también la función de actualización o “update”: `olq.update()`. Por esto es importante que nuestro código no sea bloqueante.

```
void loop()
{
    olq.update();
    // ..
    // ..
}
```

Para publicar un *feed*, utilizaremos la función `publish` del objeto `olq` :

```
olq.publish("humedad", humedad); // Publicar el dato obtenido de humedad.
```

```
olq.publish("temperatura", temperatura); // Publicar el dato obtenido de
temperatura.
```

```
olq.publish("presion", presion); // Publicar el dato obtenido de presión.
```

Finalmente, nuestro programa de estación meteorológica con IoT debería quedar como se ve a continuación:

```
// Incluimos las librerías necesarias.
#include "SoftwareSerial.h"
#include "ObloqAdafruit.h"
#include <Arduino.h>
#include <Wire.h>
#include <Adafruit_BMP085.h>
#include <SoftwareSerial.h>
#include <DHT.h>
#include <LiquidCrystal.h>

// Indicamos conexión de wifi.
#define WIFI_SSID      "SSID_de_Wifi"
#define WIFI_PASSWORD  "PWD_de_WIFI"

// Copiamos las credenciales obtenidas anteriormente en Adafruit.
#define IO_USERNAME    "usuario_adafruit"
#define IO_KEY         "key_adafruit"

SoftwareSerial softSerial(12,13);
ObloqAdafruit olq(&softSerial,WIFI_SSID,WIFI_PASSWORD,IO_USERNAME,IO_KEY);

Adafruit_BMP085 bmp;
```

```

LiquidCrystal lcd(8, 9, 4, 5, 6, 7);
DHT dht(2,11);

unsigned long millisAnterior = 0;

void setup() {
  bmp.begin();
  lcd.begin(16, 2);
  lcd.clear();
  pinMode(3,INPUT);
  softSerial.begin(9600);
}

void loop() {
  //Toma la información de los sensores cada 10 segundos.
  if(millis() - millisAnterior > 10000)
  {
    bmp.readTemperature();
    long presion=bmp.readPressure()/100; //Presion en hP
    lcd.setCursor(0,0);
    lcd.print("T:");
    int temperatura=dht.readTemperature();
    lcd.print(temperatura);
    lcd.setCursor(8,0);
    lcd.print("H:");
    int humedad =dht.readHumidity();
    lcd.print(humedad);
    lcd.setCursor(0,1);
    if(digitalRead(3)){
      lcd.print("Lluvia:NO");
    }else{
      lcd.print("Lluvia:SI");
    }
    lcd.setCursor(10,1);
    lcd.print("P:");
    lcd.print(presion);
    olq.publish("humedad", humedad); // Publica Humedad.
    olq.publish("presion", presion); // Publica presion.
    olq.publish("temperatura", temperatura);// Publica temperatura.
    millisAnterior = millis();

  }
  // Llamamos a que la librería actualice lo que necesite.
  olq.update();
}

```



## Cierre

Una vez finalizado este proyecto, es posible extenderlo si se quiere continuar. Estas son algunas opciones sugeridas:

- Agregar un acelerómetro para monitorear la dirección del viento.
- Agregar un anemómetro para medir la velocidad del viento.
- Agregar un LDR (sensor lumínico) para medir la intensidad de la luz del Sol.

El proceso de resolución de problemas como los que se han planteado aquí permite la movilización y la integración de distintos saberes en la búsqueda de soluciones posibles a una situación dada. Si bien la información aquí fue presentada a modo de instructivo, se espera que sean los estudiantes organizados en pequeños grupos quienes vayan encontrando las mejores formas para construir los dispositivos.

Esto implica preparar los materiales para que cada grupo cuente con todo lo necesario para la construcción del proyecto. Además, al interior de cada grupo, los estudiantes deben distribuirse los roles y las tareas de acuerdo a las demandas que van teniendo en las actividades. Es importante que los docentes acompañen las producciones de cada grupo monitoreando los avances de todos los estudiantes y presentando la información que se considere necesaria para continuar la tarea.

Pero, al mismo tiempo, es necesario que habiliten espacios para que los alumnos realicen hipótesis, planteen interrogantes, indaguen, prueben y realicen ajustes de acuerdo a lo que ellos mismo van pensando sobre cómo llevar a cabo el proyecto.

En este sentido, registrar lo que se va haciendo, las preguntas de los alumnos, las pruebas, los errores y cómo se fueron construyendo los dispositivos, permite reflexionar sobre la propia práctica, reforzar los aprendizajes construidos a lo largo de este proceso y poder volver a ese material disponible para próximos proyectos que se realicen.

Una vez terminado el proyecto, se sugiere reunir y organizar con el grupo el registro que se hizo del proceso realizado. Esta instancia de sistematización también permite movilizar capacidades vinculadas a la comunicación porque implica tomar decisiones respecto a cómo se quiere mostrar el proyecto a otros (otros grupos, otras escuelas, otros docentes, a la comunidad, etc.).

# Glosario

## Electrónica y Arduino

**Arduino:** Placa electrónica que contiene un microcontrolador programable y sistema de comunicación (USB y serial) que permite al usuario cargarle diversos programas así como también comunicarse con la misma. Del lado de la computadora se utiliza un IDE (entorno de desarrollo integrado) para generar el código, compilarlo y quemarlo en la placa. Existen múltiples IDE compatibles con las placas Arduino.

El microcontrolador posee entradas analógicas y digitales así como salidas digitales, PWM y servo. Las entradas y salidas digitales son las que permiten leer o escribir estados del tipo binarios. Pueden adoptar la forma de 0 ó 1, alto o bajo, verdadero o falso. Para prender y apagar los LED del semáforo utilizamos salidas digitales, las mismas están nombradas con números desde el 0 al 13.

Las entradas analógicas permiten leer información que puede adoptar diferentes niveles de tensión, tal como la lectura de un termómetro analógico, la posición de un potenciómetro, etc. Las mismas están identificadas en la placa como A0 a A5.

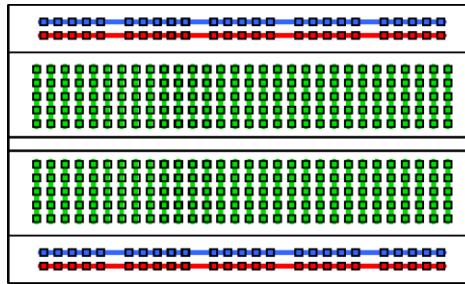
**Shield:** Placas de circuitos que se monta encima de la placa Arduino para expandir sus funcionalidades. Existen shields para otros tipos de microcontroladores y computadoras embebidas (Arduino Micro, RaspberryPi, etc). En general un shield sirve para ser utilizado con un único modelo de placa, en este caso para Arduino UNO.

El shield suele tener la misma forma que la placa Arduino y tienen pines de conexión que encastran perfectamente con los pines de esta. Los shields poseen diferentes usos como: comunicaciones, sensores, actuadores, interconexión con otros sistemas, sonido, protoboard y una larga lista de etcéteras.

**Puerto COM:** Es el puerto de comunicaciones a través del cual un sistema operativo informático se comunica con un dispositivo externo tal como una placa Arduino. La asignación de los mismos suele realizarse de forma automática al conectar la placa via USB. Dicha asignación suele ser dinámica, lo que significa que a veces cambia el número al conectar una misma placa en otro puerto USB o al conectar varias placas. En todos los IDE de programación es necesario especificar el puerto COM a través del cual nos comunicaremos con la placa Arduino.

**Protoboard:** Es una placa experimental que permite el prototipado rápido de circuitos electrónicos. Tiene orificios para insertar las patas de los componentes permitiendo que se conecten sin tener que recurrir a la soldadura.

El mismo posee una grilla de orificios que se encuentran conectados entre sí siguiendo el esquema de la imagen. Las líneas de conexión superior e inferior recorren la placa de punta a punta y suelen utilizarse para la alimentación del circuito, mientras que las líneas verdes se suelen utilizar para interconectar componentes. Tomar en cuenta que las líneas verdes se interrumpen en el centro de la placa. Generalmente se utilizan cables del tipo dupont para realizar conexiones en la protoboard.



**Sensor DHT11:** Se utiliza para medir humedad y temperatura. El sensor de temperatura consiste en un termistor, un dispositivo que cambia su resistencia en función de la temperatura que percibe. El sensor de humedad consta de un sustrato higroscópico (un dispositivo que atrae vapor de agua) conectado a dos electrodos que miden su resistencia. Cuanto mayor es la humedad del ambiente, mayor es también la conductividad del sustrato.

El DHT11 combina el sensado de ambas variables, integra también un circuito electrónico digital encargado de digitalizar la información y transmitirla al arduino mediante un pin digital a modo de paquete de información. Por este motivo es que para realizar un programa que utilice este sensor es necesario utilizar una librería que se encarga de gestionar la comunicación entre el Arduino y DHT11. El sensor posee 4 pines de conexión, dos de ellos son alimentación eléctrica (VCC y GND), mientras que un pin se utiliza para la comunicación. Hay un pin que no tiene uso. En algunos casos podremos encontrar el sensor montado en una pequeña placa de interconexión que solamente tiene 3 pines, descartando el pin que no tiene uso.

**Sensor de lluvia:** Los sensores de lluvia detectan la presencia de lluvia en el ambiente por la variación de conductividad que sufre el sensor. El mismo consta de dos cintas conductoras separadas que al mojarse entran en contacto. El sensor tiene 4 pines de conexión, dos de ellos son para alimentación eléctrica (VCC y GND). Los dos pines restantes son las salidas, una Analógica que devuelve un nivel de tensión en relación a la cantidad de agua presente sobre el sensor y otra salida Digital que se activa o desactiva indicando presencia de agua. Es posible calibrar el umbral de detección para la presencia utilizando el potenciómetro integrado en el sensor.

**El sensor de presión barométrica BMP180:** está diseñado para leer la presión atmosférica, que es la fuerza que ejerce el aire (atmósfera) sobre la superficie de la tierra. Este sensor integra un circuito electrónico que digitaliza la información registrada y se encarga de comunicarse con el Arduino utilizando el protocolo llamado I<sup>2</sup>C. Para leer la información del mismo es necesario utilizar una librería que resuelve la comunicación usando dicho protocolo. El dispositivo tiene 5 patas, dos de ellas se utilizan para la alimentación eléctrica (VCC y GND), existe un pin llamado 3.3 V que también se utiliza para alimentación en otro tipo de situaciones. Los pines SCL y SDA se utilizan para la comunicación serial I<sup>2</sup>C

## Internet de las cosas

**Panel de Control Adafruit:** Los sistemas IoT trabajan apoyándose en un servidor que se encarga de centralizar y gestionar la información que reportan los diversos sensores así como responder a las consultas de los dispositivos que buscan acceder a dicha información (mostrarla en pantalla, tomar decisiones, etc). Adafruit es una plataforma online con posibilidad de uso gratuito que ofrece el servicio de gestión de esta información. La misma ofrece un alto grado de compatibilidad con diversos estándares de trabajo IoT y se encuentra principalmente orientada al uso educativo.

**Feed:** fuente de datos en la que uno puede publicar y a la que puede suscribirse. Es decir, permite enviar datos, para que estos sean almacenados en el tiempo así como también leerlos, recibiendo las actualizaciones de quienes estén publicando allí. Es una forma de almacenar información en una gran base de datos de forma ordenada, utilizando el concepto de etiquetas tanto al momento de escribirla como el de leerla.

## Reconocimientos

Este trabajo es fruto del esfuerzo creativo de un enorme equipo de entusiastas y visionarios de la pedagogía de la innovación, la formación docente, la robótica, la programación, el diseño y la impresión 3D. Les agradecemos por el trabajo en equipo inspirador para traer a la realidad la obra que, en forma conjunta, realizamos INET y EDUCAR del Ministerio de Educación, Cultura, Ciencia y Tecnología de la Nación Argentina.

### **Contenidos**

#### **Equipo INET**

Alejandro Anchava  
Joreliz Andreyana Aguilera Barragán  
Omar Leandro Bobrow  
Alejandro Cesar Cáceres  
Ezequiel Luberto  
Gustavo Roberto Mesiti  
Alejandro Palestrini  
Judit Schneider  
Pablo Trangone

#### **Equipo Educar:**

Pablo Aristide  
Mayra Botta  
Anabela Cathcarth  
Eduardo Chiarella  
María Laura Costilla  
Diego Dorado  
Facundo Dyszel  
Federico Frydman  
Matías Rinaldi  
Uriel Rubilar  
Camila Stecher  
Carolina Sokolowicz  
Nicolás Uccello

Para la confección de esta obra se contó con el apoyo de la Universidad Pedagógica Nacional "UNIPE". En particular en el desarrollo de los capítulos 1 y 2, los cuales estuvieron a cargo de los profesores Fernando Raúl Alfredo Bordignon y Alejandro Adrián Iglesias.

### **Producción y comunicación**

Juliana Zugasti

### **Diseño y edición**

Leonardo Frino  
Mario Marrazzo

## **Corrección de estilo**

María Cecilia Alegre

## **Agradecimientos especiales**

Mariano Consalvo. Equipo ABP

Damián Olive. Equipo de ABP

María José Licio Rinaldi, Directora Nacional de Asuntos Federales INET, quien siempre acompañó a este equipo en todas las gestiones para su implementación

Estamos comprometidos en instalar la innovación en la escuela secundaria técnica: la robótica, la programación, el pensamiento computacional, los proyectos tecnológicos, el ABP, la impresión 3D, de manera más accesible para todos.

Agradecemos enormemente, docente, tu continua dedicación y compromiso con el futuro de tus estudiantes.

***¡Estamos ansiosos por saber qué es lo que vamos a crear juntos!***