

OBJETOS INTELIGENTES CONECTADOS

TEMA	Conexión de sistemas a través de IoT
FORMATO	Individual, parejas o grupos pequeños
TIEMPO DE PREPARACIÓN	1 hora
DURACIÓN DE LA ACTIVIDAD	1h30
NIVEL DE DIFICULTAD	Avanzado

OBJETIVOS PEDAGÓGICOS

- Desarrollar un proyecto de IoT
- Trabajar en la implementación de una aplicación con un sensor
- Trabajar en la implementación de una aplicación con un actuador
- Aprender a conectar sistemas electrónicos a través de Internet.

MATERIALES NECESARIOS Grupo 1

- Sala de informática con computadores conectados al Internet
- Placa NodeMCU o similar
- Protoboard
- Jumpers
- Potenciómetro

MATERIALES NECESARIOS Grupo 2

- Sala de informática con computadores conectados al Internet
- Placa NodeMCU o similar
- Protoboard
- Jumpers
- LEDs
- Resistencia 220R

OBJETOS INTELIGENTES CONECTADOS

Realización de la actividad:

- En esta actividad, los estudiantes crearán proyectos que se conectarán entre sí desde la plataforma Thingspeak. Para realizar la actividad, cada proyecto estará compuesto por 2 grupos diferentes: **Grupo 1** y **Grupo 2**.
- El **Grupo 1** se encargará de crear un circuito y código que hará lecturas en el canal Thingspeak y el **Grupo 2** se encargará de crear un proyecto que leerá los valores y creará una aplicación para el actuador.
- Si es posible, divida la clase en grupos pequeños. Cada grupo debe tener acceso a un computador con Arduino IDE instalado, una placa NodeMCU o similar y un cable para conectar la placa al computador. Si esto no es posible, realice la actividad por turnos para que todos los alumnos tengan acceso a los materiales, o trabaje en un formato colectivo, proyectando su pantalla. Los grupos pueden ser del tipo **Grupo 1** o **Grupo 2**.
- Organice los materiales y proporcione un kit con los componentes para cada grupo.
- Luego, pídale a cada grupo que conecte la placa a su computador y abra el IDE de Arduino.

Grupo 1

- Presente los pasos para transferir el siguiente código dentro de la placa.

Código

```
// Bibliotecas ThingSpeak e ESP8266
#include <ThingSpeak.h>
#include <ESP8266WiFi.h>

// Datos de su red: nombre y contraseña
char ssid[] = "xxxxxxxx"; // Nombre de la red SSID
char pass[] = "xxxxxxxx"; // Nombre de la contraseña

int status = WL_IDLE_STATUS;
WiFiClient client;

int sensorValue; // variable que almacena el valor leído por el sensor

// Informaciones disponibles en ThingSpeak, número del canal y clave de escritura
unsigned long canal = xxx;
const char * ClaveEscrituraAPI = "xxxxxxx";

void setup() {
  WiFi.begin(ssid, pass);
  ThingSpeak.begin(client);
}

void loop() {
  // Lee el valor de entrada en el pin analógico 0
  sensorValue = analogRead(A0);

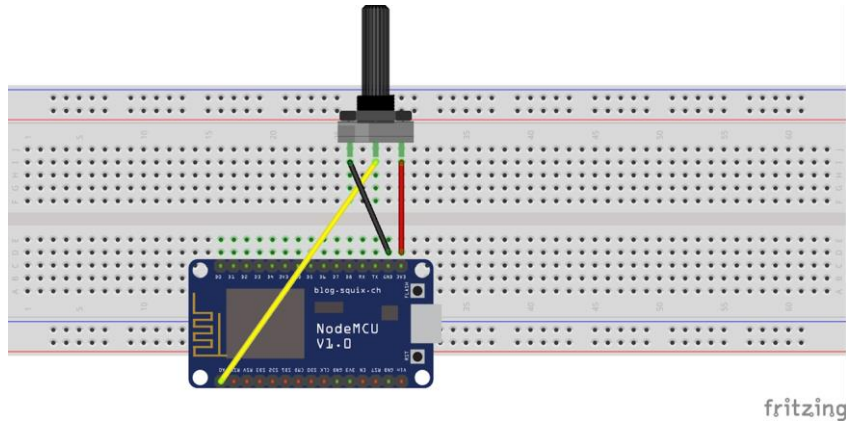
  // Escribe enThingSpeak:
  // (número de canal, número de campo, valor, clave de escritura)
  ThingSpeak.writeField(canal, 1, sensorValue, ClaveEscrituraAPI);

  delay(5000); // ThingSpeak acepta actualizaciones de los valores cada 5 segundos.
}
```

OBJETOS INTELIGENTES CONECTADOS

Realización de la actividad:

- El circuito del proyecto sigue a continuación:



- Una vez que se hayan cambiado los parámetros de red, la contraseña, las claves de escritura y lectura y el canal Thingspeak, pídeles que actualicen los valores leídos en los gráfico del canal de la plataforma Thingspeak.

OBJETOS INTELIGENTES CONECTADOS

Realización de la actividad:

Grupo 2

- Presente los pasos para transferir el siguiente código dentro de la placa.

Código

```
#include <ThingSpeak.h>
#include <ESP8266WiFi.h>

int lectura;
int led = D7;

// Nombre de la red y contraseña
char ssid[] = "#####"; // Nombre de la red SSID
char pass[] = "#####"; // Nombre de la contraseña

int status = WL_IDLE_STATUS;
WiFiClient client;
int sensorValue; // variable que almacena el valor analógico leído

// Información disponible en ThingSpeak, número del canal y clave de escritura
unsigned long canal = #####;
const char * ClaveLecturaAPI = "#####";

void setup() {
  WiFi.begin(ssid, pass);
  ThingSpeak.begin(client);
  Serial.begin(115200);
  pinMode(led, OUTPUT);
}

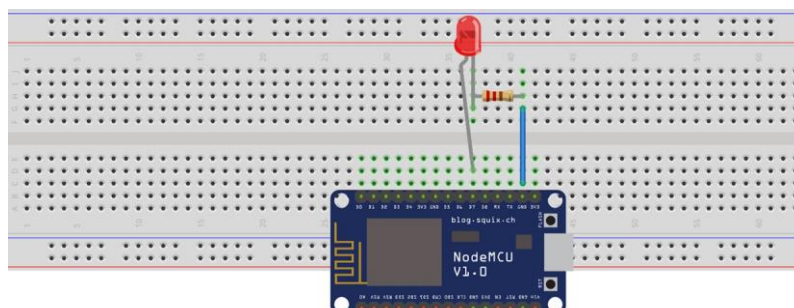
void loop() {
  //Lee el ultimo valor escrito en el canal a partir de los siguientes parámetros (CANAL, CAMPO, Clave de Lectura)
  lectura = ThingSpeak.readLongField(canal, 1, ClaveLecturaAPI);

  //Escribe el valor leído en el Serial Monitor
  Serial.print("Lectura: ");
  Serial.println(lectura);

  if(lectura == 1){
    digitalWrite(led, 1);
  }

  if(lectura == 0){
    digitalWrite(led, 0);
  }
}
```

- El circuito del proyecto sigue a continuación:



fritzin

Iniciativa

SAMSUNG

Desarrollo

LSI TEL

OBJETOS INTELIGENTES CONECTADOS

Realización de la actividad:

- Una vez que se hayan cambiado los parámetros de red, la contraseña, las claves de escritura y lectura y el canal Thingspeak, pídeles que enciendan y apaguen el LED utilizando el mismo sistema presentado en el plan de clase: Actuadores, del curso 6.
- Para conectar los proyectos desarrollados por los alumnos, los canales y claves de escritura y lectura deben ser los mismos. Así, **el Grupo 1** actualizará los valores de la lectura del potenciómetro y el **Grupo 2** leerá este valor y lo aplicará al LED.

Discusión y reflexión:

Después de completar la actividad, discuta los conceptos aprendidos con la clase, probando y proyectando el código, si es posible. Algunas preguntas que pueden orientar la discusión:

- ¿Qué es un programa de computador? ¿Cómo funciona este programa?
- ¿En qué parte del código, el programa lee información de Internet?
- ¿Por qué se actualiza el valor cada 5 segundos?
- ¿Qué pasaría si el fragmento de código que está dentro de la función *void loop* se pasa a *void setup*?
- ¿Qué otras aplicaciones podrían desarrollarse para crear alguna solución práctica utilizando la comunicación entre dos microcontroladores y el Internet de las cosas?

Créditos:

Nathan Rabinovitch(LSITec/USP)