|  |
| --- |
| Software para Robots Jordán Pascual : pascualjordan@uniovi.es |

|  |  |
| --- | --- |
| 7 | Objetos conectados - Parte 1 |

# Objetos conectados

En esta práctica se presenta como utilizar varios componentes electrónicos y como conectar Arduino a internet para crear un servidor Web.

## Servidor Web (Con Arduino LAN)

En este ejemplo veremos cómo crear un servidor web en un controlador Arduino. Permitiremos que el servidor apague o encienda una luz dependiendo de las peticiones que reciba.

|  |  |
| --- | --- |
| https://www.arduino.cc/en/uploads/Main/ArduinoEthernetShield_R3_Front_450px.jpg | Arduino Ethernet Shield: esta placa de extensión permite conectar Arduino a internet mediante un cable de red.  Soporta varios sockets de conexión simultáneos. Para su uso se recomienda utilizar la Ethernet Library. [https://www.arduino.cc/en/Reference/Ethernet](https://www.arduino.cc/en/Reference/Ethernet%20)  Incluye lector de tarjetas micro SD. |

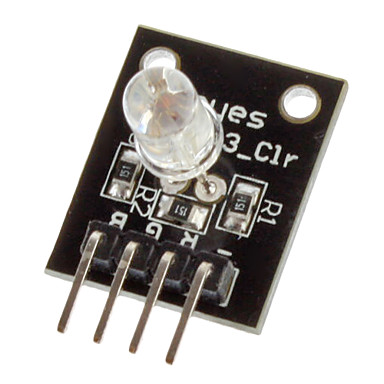
**Construcción del circuito:**

Encajar la placa de extensión cuidadosamente en la placa Arduino. Si la placa está dañada o mal encajada puede probar un mal funcionamiento parcial (de algunos pines) o total.

|  |
| --- |
| Al encajar algunos modelos de placa **el pin digital 13** queda debilitado. |

Conectamos el cable de red a una toma de un ordenador. Estos ordenadores no tienen IP publica por lo que se generará un servidor local accesible por ordenadores que estén en la misma red.

**Conectamos la patilla R del LED RGB a la patilla 8, y la patilla - del led a una GND**



**Programación de Arduino:**

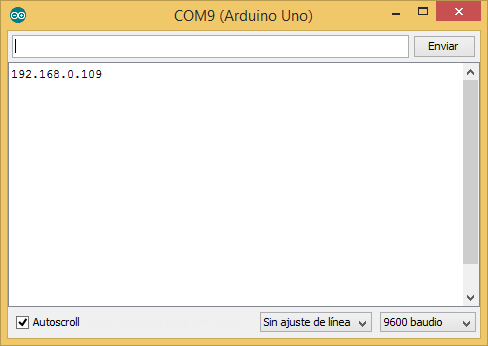
Requiere incluir dos librerías oficiales de Arduino Program -> Include Library -> (SPI & Ethernet).

|  |
| --- |
| Importante: hay que cambiar la IP, a cada pareja se le asignará una IP. |

|  |
| --- |
| #include <SPI.h> //Importamos librería comunicación SPI  #include <Ethernet.h> //Importamos librería Ethernet  byte mac**[]** **=** **{**0x54**,** 0x55**,** 0x58**,** 0x10**,** 0x00**,** 0x24**};**  EthernetServer servidor**(**80**);** //Puerto en el 80  IPAddress dnServer**(**156**,** 35**,** 14 **,** 2**);**  IPAddress gateway**(**192**,** 168**,** 61**,** 13**);**  IPAddress subnet**(**255**,** 255**,** 255**,** 0**);**  // Que cada uno ponga la IP de su grupo 201, 202, 203  IPAddress ip**(**192**,** 168**,** 61**,** 250**);**  int led **=** 8**;**  void setup**()**  **{**  Serial**.**begin**(**9600**);**  Ethernet**.**begin**(**mac**,** ip**,** dnServer**,** gateway**,** subnet**);**  servidor**.**begin**();**  Serial**.**println**(**"Setup"**);**  // Imprimir la IP  Serial**.**println**(**Ethernet**.**localIP**());**    // Inicializar el led  pinMode**(**led**,**OUTPUT**);**  digitalWrite**(**led**,**LOW**);**  **}**  void loop**()**  **{**  EthernetClient cliente **=** servidor**.**available**();**  **if** **(**cliente**)** **{**  Serial**.**println**(**"nueva peticion"**);**  String peticion**=**""**;**  **while** **(**cliente**.**connected**())** **{**  **if** **(**cliente**.**available**())** **{**    char c **=** cliente**.**read**();** //Leer petición carácter a carácter  peticion**.**concat**(**c**);** // concatenar en un string    // Ha acabado la peticion http  // Si contiene index responder con una web    // la petición ha acabado '\n' y contiene la cadena "index"  **if** **(**c **==** '\n' **&&** peticion**.**indexOf**(**"index"**)** **!=** **-**1**)** **{**  Serial**.**println**(**"Responder"**);**  // Serial.println(peticion);    // contiene la cadena "encender+"  **if(**peticion**.**indexOf**(**"encender"**)** **!=** **-**1 **)**  **{**  digitalWrite**(**led**,**HIGH**);**  Serial**.**println**(**"Encender Led"**);**  **}**  // contiene la cadena "apagar"  **if(**peticion**.**indexOf**(**"apagar"**)** **!=** **-**1**)**  **{**  Serial**.**println**(**"Encender led"**);**  digitalWrite**(**led**,**LOW**);**  **}**    // Enviamos al cliente una respuesta HTTP  cliente**.**println**(**"HTTP/1.1 200 OK"**);**  cliente**.**println**(**"Content-Type: text/html"**);**  cliente**.**println**();**  **break;**  **}**  **}**  **}**    // Pequeña pausa para asegurar el envio de datos  delay**(**1000**);**  cliente**.**stop**();**// Cierra la conexión  **}**  **}** |

### Prueba:

Ejecutamos la aplicación, abrimos el monitor, vemos la IP que se ha asignado al servidor.



Desde un navegador (conectado a la misma red) ejecutamos las siguientes peticiones:

http://**<mi\_ip>**/index?encender

http:// **<mi\_ip>**/index?apagar

### Retornar HTML

Aunque no es muy recomendable que un Arduino Uno funcione como servidor Web es técnicamente posible, aunque solo será apropiado si sirve páginas de muy poco peso que vayan a tener muy pocos usuarios (o uno).

Si la página Web es grande debemos utilizar la tarjeta SD para almacenarla, si es una página muy sencilla se podría crear desde código.

Incluir en la respuesta el siguiente código:

|  |
| --- |
| // Enviamos al cliente una respuesta HTTP  cliente**.**println**(**"HTTP/1.1 200 OK"**);**  cliente**.**println**(**"Content-Type: text/html"**);**  cliente**.**println**();**  cliente**.**println**(**"<html>"**);**  cliente**.**println**(**"<body>"**);**  cliente**.**println**(**"<h1>Control de luz por internet</h1>"**);**  cliente**.**println**(**"<h2><a href='index.html?p=encender'>Encender</a></h2>"**);**  cliente**.**println**(**"<h2><a href='index.html?p=apagar'>Apagar</a></h2>"**);**  cliente**.**println**(**"</b></body>"**);**  cliente**.**println**(**"</html>"**);**  **break;** |

## Servidor Web (Con Arduino WeMos D1)

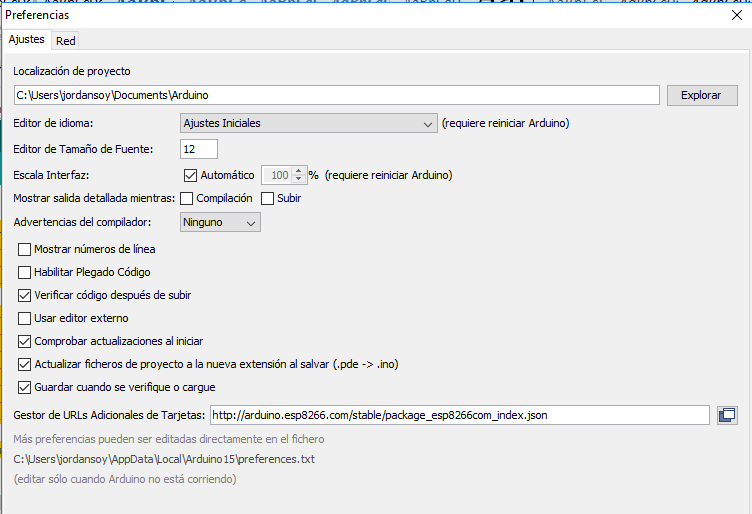
En este ejemplo veremos cómo crear un servidor web con la placa Arduino Wemos D0.

|  |  |
| --- | --- |
|  | Arduino WeMos D1 es una placa Arduino compatible. Que cuenta con un módulo WiFi ESP8266 Integrado.  Esta placa no está incluida en el IDE por lo que debemos instalar sus drivers.  Todos los pines digitales de esta placa admiten PWM  El Led integrado está en el pin D5 en lugar del 13  Tiene un único pin de entrada analógica  Especificación completa:  <https://www.wemos.cc/product/d1.html> ,  4M de memoria Flash,  \*Versión mini : <https://www.wemos.cc/product/d1-mini.html> |

**Instalación de la placa:**

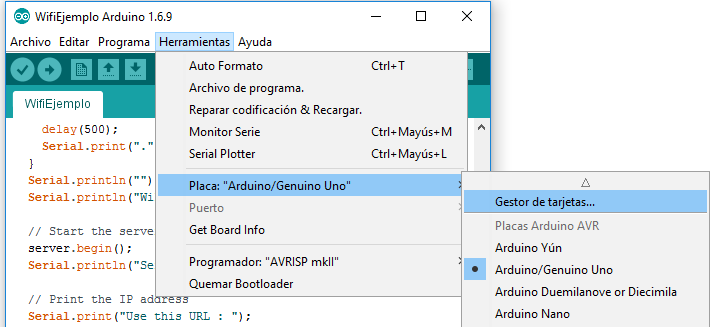
Debemos instalar el plugin ESP8266 (es el módulo WiFi) Para ello abrir el IDE de arduino y entrar en **Archivo -> Preferencias.**

“*Localizar el campo Gestor de URLs Adicionales de Tarjetas*” e incluir la URL:<http://arduino.esp8266.com/package_esp8266com_index.json>

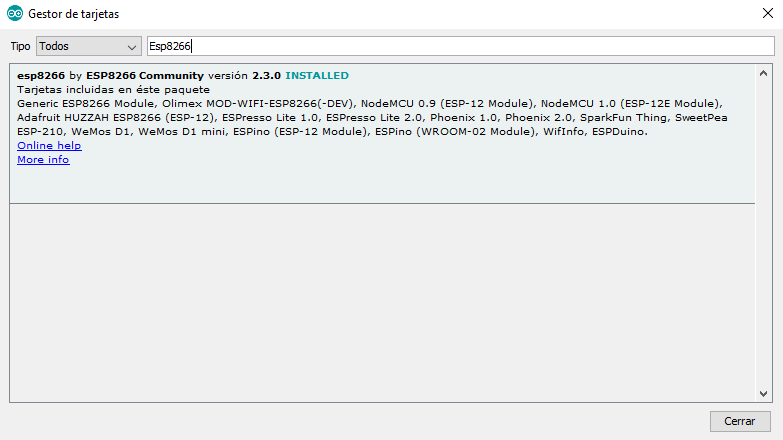


Pulsar **Ok** y volver a la pantalla principal del IDE de Arduino.

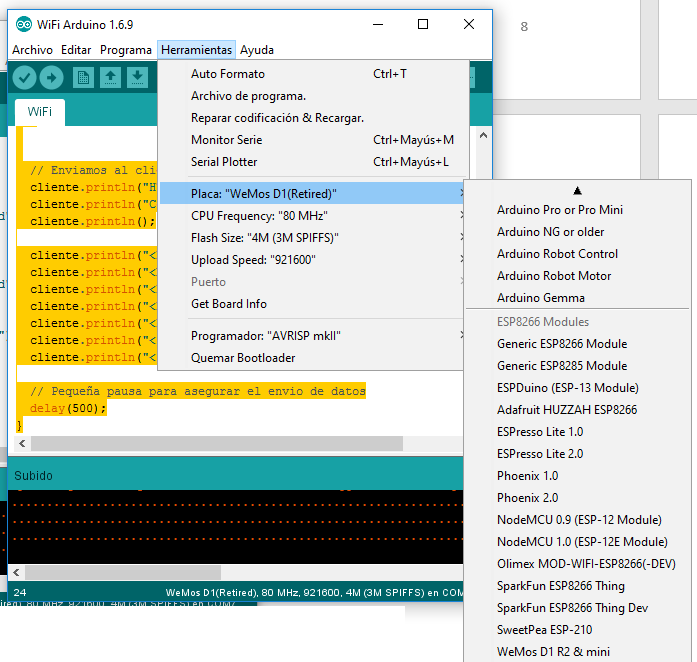
Seleccionar **Herramientas -> Placa -> Gestor de tarjetas…**



En el cuadro de búsqueda introducir **Esp8266** y después pulsar en **Instalar** la coincidencia “esp8266 by ESP8266 Community”.



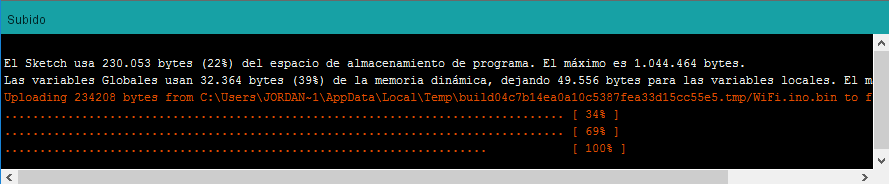
Luego nos aparecerá una lista de placas más amplia, bajábamos el scroll hasta encontrar **Wemos D1 (Retired)**



No debemos olvidarnos de seleccionar el **Puerto**, al igual que hacíamos con el resto de placas.

También se le puede cambiar la **CPU Frequency** a 160Mhz pero no es demasiado crítico.

**Probamos a ejecutar el programa vacío y debería aparecer el mensaje de subido.**

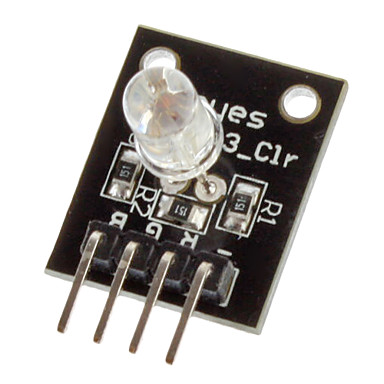
****

**Particularidades de la placa:**

* En el código los pines digitales se nombran con D2, D3, D4, etc.
* En el pin D5 tiene un led integrado (Está en el medio de la placa).

**Conectamos la patilla R del LED RGB a la patilla D8, y la patilla - del led a una GND**

Los pins digitales de está placa se mueven en un rango de valores 0 – 3.3V en lugar de 0 – 5V como ocurre en otras placas arduino.

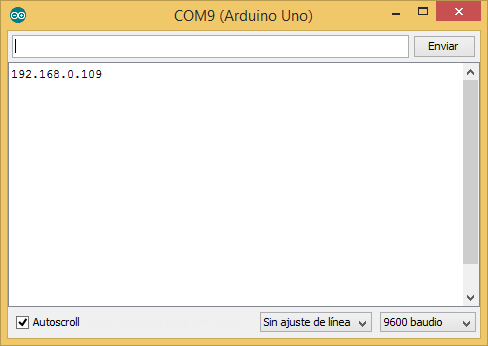


**Programación de Arduino:**

|  |
| --- |
| #include <ESP8266WiFi.h>  const char**\*** ssid **=** "ICTCampus"**;**  const char**\*** password **=** ""**;**  IPAddress dnServer**(**156**,** 35**,** 14 **,** 2**);**  IPAddress gateway**(**192**,** 168**,** 61**,** 13**);**  IPAddress subnet**(**255**,** 255**,** 255**,** 0**);**  // Que cada uno ponga la IP de su grupo 211, 212, 213  // Grupo 1 = 211 , Grupo 2 ) 212  IPAddress ip**(**192**,** 168**,** 61**,** 211**);**    int ledPin **=** D8**;**  WiFiServer server**(**80**);**    void setup**()** **{**  Serial**.**begin**(**9600**);**  Serial**.**print**(**"Start"**);**  delay**(**10**);**    pinMode**(**ledPin**,** OUTPUT**);**  digitalWrite**(**ledPin**,** LOW**);**    // Conectar a la red Wifi  Serial**.**print**(**"Conectando a "**+**String**(**ssid**));**    WiFi**.**config**(**ip**,** gateway**,** subnet**);**  WiFi**.**begin**(**ssid**,** password**);**  **while** **(**WiFi**.**status**()** **!=** WL\_CONNECTED**)** **{**  delay**(**500**);**  Serial**.**print**(**"."**);**  **}**    Serial**.**println**(**"Conectado al WiFi"**);**    // Iniciar un servidor  server**.**begin**();**  Serial**.**print**(**"Servidor Web en: http://"**);**  Serial**.**print**(**WiFi**.**localIP**());**  **}**    void loop**()** **{**  // Comprueba clientes conectados  WiFiClient cliente **=** server**.**available**();**  **if** **(!**cliente**)** **{**  // Mientras no haya cliente repito el bucle  **return;**  **}**    // Hay cliente!  Serial**.**println**(**"Nuevo cliente, esperar mientras no está listo"**);**  **while(!**cliente**.**available**()){**  delay**(**1**);**  **}**    // Leo la petición , hasta el salto de linea \n  String peticion **=** cliente**.**readStringUntil**(**'\r'**);**  Serial**.**println**(**peticion**);**  cliente**.**flush**();**    // Miro si contiene alguna de las dos cadenas  **if** **(**peticion**.**indexOf**(**"encender"**)** **!=** **-**1**)** **{**  digitalWrite**(**ledPin**,** HIGH**);**  **}**  **if** **(**peticion**.**indexOf**(**"apagar"**)** **!=** **-**1**){**  digitalWrite**(**ledPin**,** LOW**);**  **}**      // Enviamos al cliente una respuesta HTTP  cliente**.**println**(**"HTTP/1.1 200 OK"**);**  cliente**.**println**(**"Content-Type: text/html"**);**  cliente**.**println**();**  cliente**.**println**(**"<html>"**);**  cliente**.**println**(**"<body>"**);**  cliente**.**println**(**"<h1>Control de luz por internet</h1>"**);**  cliente**.**println**(**"<h2><a href='index.html?p=encender'>Encender</a></h2>"**);**  cliente**.**println**(**"<h2><a href='index.html?p=apagar'>Apagar</a></h2>"**);**  cliente**.**println**(**"</b></body>"**);**  cliente**.**println**(**"</html>"**);**  // Pequeña pausa para asegurar el envio de datos  delay**(**500**);**  **}** |

### Prueba:

Ejecutamos la aplicación, abrimos el monitor, vemos la IP que se ha asignado al servidor.



# Ejemplos

## Sensor de temperatura y humedad

En este ejemplo veremos cómo utilizar un sensor de temperatura y humedad.

|  |  |
| --- | --- |
| http://litbimg2.rightinthebox.com/images/384x384/201311/ocdsqq1385538101030.jpg | **Sensor de temperatura y humedad DHT11.** Recoge de forma analógica las mediciones de temperatura y humedad, la frecuencia de actualización es de unos dos segundos.  Alimentación: 3-5V DC  Rango de temperatura: 0 a 50ºC  Rango Humedad: 20-95% de humedad relativa |

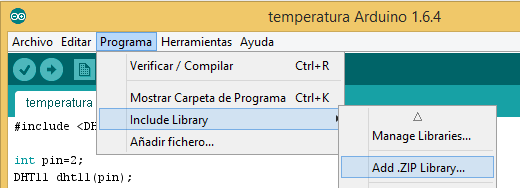
**Construcción del circuito**

* Conexiones
  + **–** a Gnd , **Pin central** a 5v, **S** a un pin digital

**Programa**  
  
Descargamos la librería dh11 para realizar un uso del sensor más "amigable" y ahorrarnos las conversiones de los valores analógicos.

[http://www.prometec.net/wp-content/uploads/2014/10/DHT11.zip](http://www.prometec.net/wp-content/uploads/2014/10/DHT11.zip%20%20)

La agregamos a Arduino. **Programa -> Include Library**



|  |
| --- |
| #include <DHT11.h> // Libreria externa  int pin\_sensor **=** 8**;**  DHT11 dht11**(**pin\_sensor**);**  void setup**(){**  Serial**.**begin**(**9600**);**  **}**  void loop**(){**  float temperatura**,** humedad**;**  int codigoSalida **=** dht11**.**read**(**humedad**,** temperatura**);**    **if(** codigoSalida **==** 0**)** **{** // Tiene que retornar 0,  Serial**.**println**(**"Temperatura: "**+**String**(**temperatura**));**  Serial**.**println**(**"Humedad: "**+**String**(**humedad**));**  Serial**.**println**(**"--------------------------------"**);**  **}** **else** **{**  // No retorna 0 hay un error  Serial**.**println**(**"Error:"**+**String**(**codigoSalida**));**  **}**    delay**(**2000**);** // Actualizar cada dos segundos  **}** |

## Sensor de Gases y Humo

En este ejemplo veremos cómo utilizar un sensor detector de gas y humo.

|  |  |
| --- | --- |
|  | **Sensor MQ-2.** Sensores de gas equipados con un electroquímico capaces de detectar diferentes tipos de gases. Esta versión del sensor está pensada para interiores.  MQ-2: Metano, Butano, GLP (Gas licuado petróleo) y humo.  MQ-3: Alcohol, Etanol y humo  MQ-4: Metano, Gas GNC (Gas natural comprimido)  MQ-5: Gas natural  MQ-6: Gas butano  MQ-7: Monóxido de carbono  MQ-8: Hidrogeno  MQ-9: Monóxido de carbono en gases inflamables  MQ-131: Ozono  [http://playground.arduino.cc/Main/MQGasSensors](http://playground.arduino.cc/Main/MQGasSensors%20)  **La precisión del sensor se puede calibrar utilizando el tornillo de la parte inferior.** |

**Construcción del circuito**

* Conexiones
  + **VCC** a 5v , **GND** a tierra, **Dout** a un pin analógico

|  |
| --- |
| **Precaución**: después de usar el sensor quema, no tocarlo. |

**Programa**  
  
Mide los niveles de gas respecto a una calibración inicial.

|  |
| --- |
| float RO**;**  int pin\_sensor **=** A0**;**  void setup**()** **{**  Serial**.**begin**(**9600**);**  // Calibrar en base al ambiente  calibrarRO**();**  **}**    void loop**()** **{**    float voltaje**;**  float gas**;** // Get value of RS in a GAS  float ratio**;** // Get ratio RS\_GAS/RS\_air  int valorSensor **=** analogRead**(**pin\_sensor**);**    voltaje **=** **(**float**)**valorSensor**/**1024**\***5.0**;**  gas **=** **(**5.0**-**voltaje**)/**voltaje**;** // omit \*RL  ratio **=** gas**/**RO**;** // ratio = RS/R0    Serial**.**println**(**"Voltaje = "**+**String**(**voltaje**));**  Serial**.**println**(**"Gas = "**+**String**(**gas**));**  Serial**.**println**(**"Ratio de Gas en Aire = "**+**String**(**ratio**));**  delay**(**1000**);**  **}**  void calibrarRO**(){**  float voltaje**;**  float aireLimpio**;** // Get the value of RS via in a clear air  float valorSensor**;**    // Hacer la media de 100 mediciones  **for(**int x **=** 0 **;** x **<** 100 **;** x**++)**  **{**  valorSensor **=** valorSensor **+** analogRead**(**pin\_sensor**);**  **}**  valorSensor **=** valorSensor **/** 100.0**;**  // El valorSensor medio con el aire limpio  voltaje **=** **(**valorSensor**/**1024**)** **\*** 5.0**;**  aireLimpio **=** **(**5.0**-**voltaje**)** **/** voltaje**;**  RO **=** aireLimpio**/**9.8**;** // 9.8 es aire limpio    Serial**.**println**(**"R0 = "**+**String**(**RO**));**  **}** |