CURSO DE ARDUINO

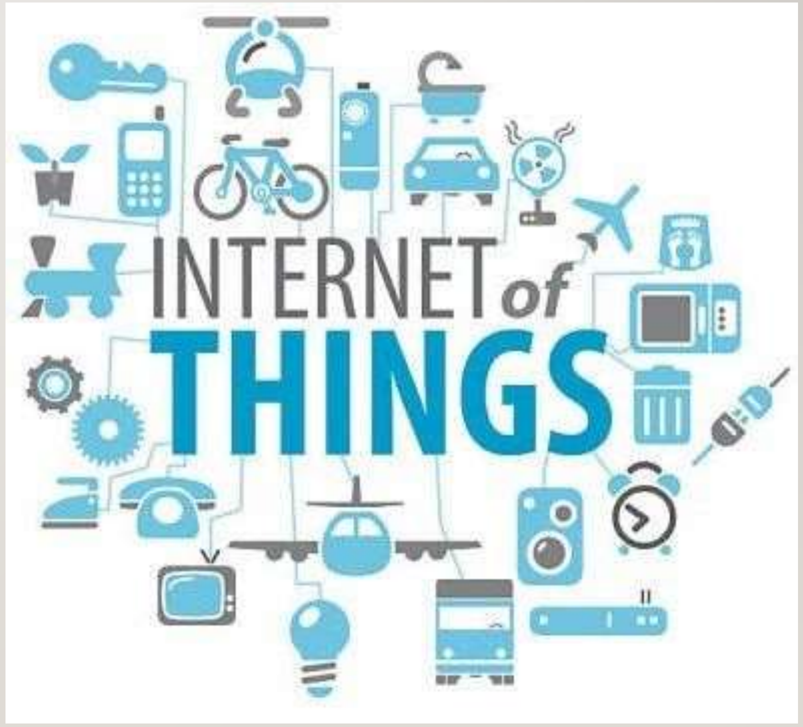
DIRIGIDO POR: MIGUEL ANGEL CALIFA URQUIZA

AGENDA

- Que es IoT en el mundo actual.
- Aplicaciones IoT.
- Tarjetas de expansión WiFi.
- Practica con módulos WiFi.

QUE ES IOT?

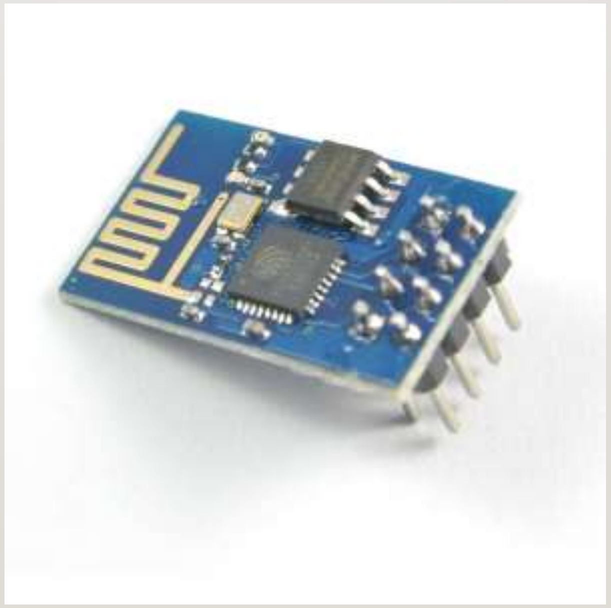
- IoT es un concepto que hace referencia a la interconexión de todos los objetos con internet.



APLICACIONES DEL CONCEPTO IOT

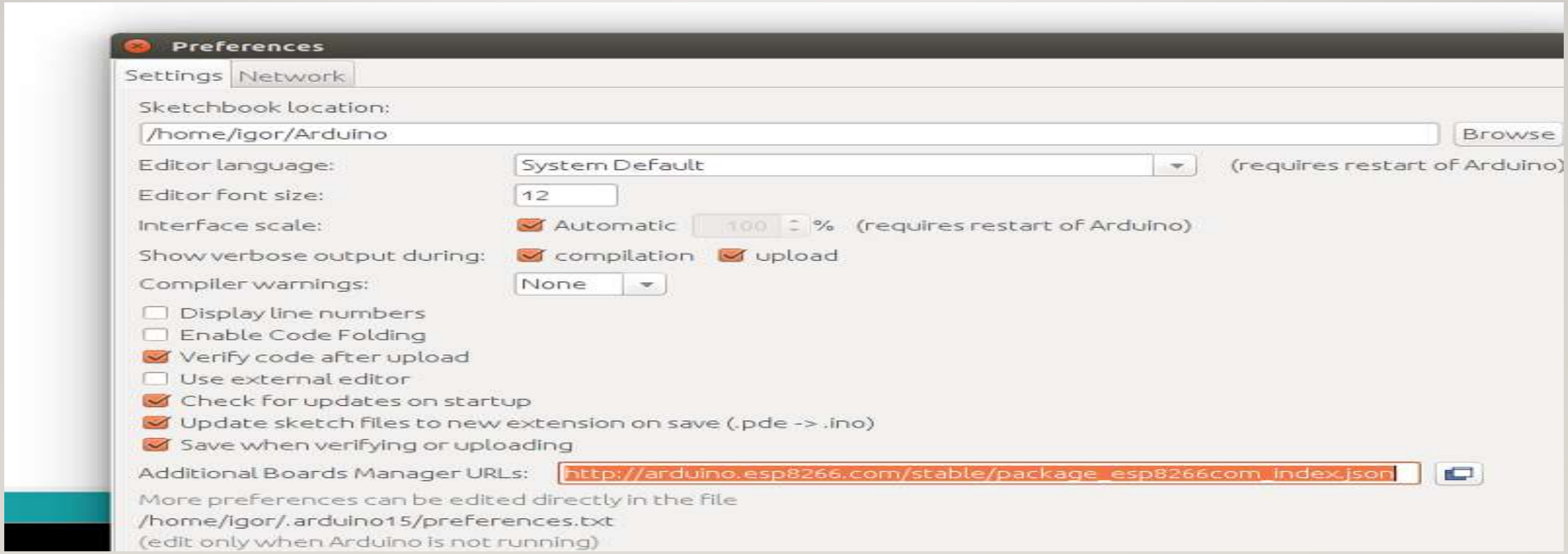
TARJETAS DE EXPANSION WIFI

Arduino WiFi – ESP8266 (Las mas utilizadas en la industria)



PRACTICA CON LOS MODULOS WIFI (INSTALACION BOARD MANAGER)

```
https://dl.espressif.com/dl/package_esp32_index.json, http://arduino.esp8266.com/stable/package_esp8266com_index.json
```



Preferences

Settings | Network

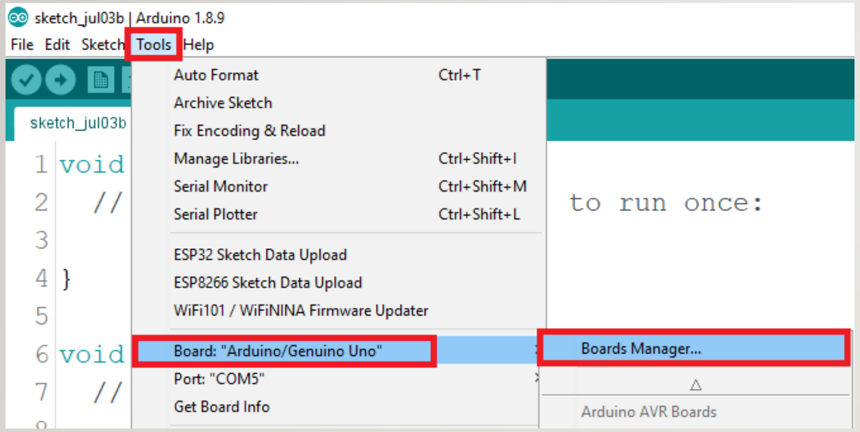
Sketchbook location:
/home/igor/Arduino
Editor language: System Default (requires restart of Arduino)
Editor font size: 12
Interface scale: [✓] Automatic 100 % (requires restart of Arduino)
Show verbose output during: [✓] compilation [✓] upload
Compiler warnings: None

- [ ] Display line numbers
- [ ] Enable Code Folding
- [✓] Verify code after upload
- [ ] Use external editor
- [✓] Check for updates on startup
- [✓] Update sketch files to new extension on save (.pde -> .ino)
- [✓] Save when verifying or uploading

Additional Boards Manager URLs: http://arduino.esp8266.com/stable/package_esp8266com_index.json

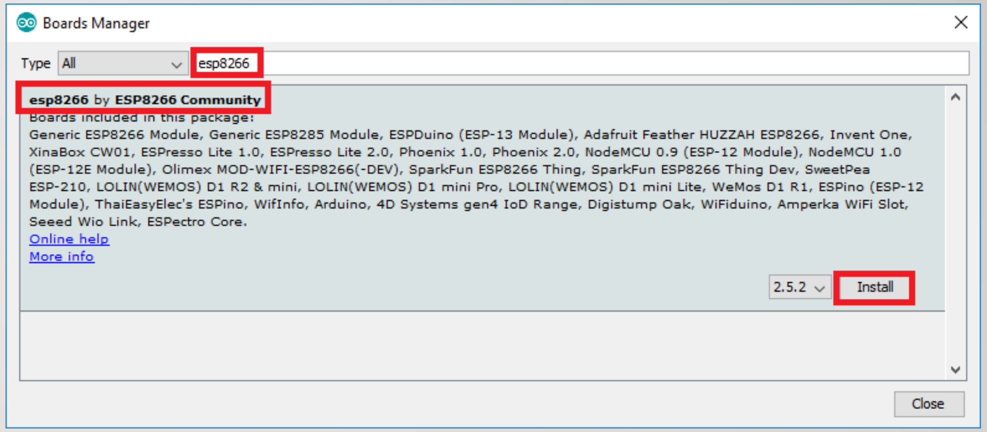
More preferences can be edited directly in the file
/home/igor/.arduino15/preferences.txt
(edit only when Arduino is not running)

INSTALAR LAS BOARDS ESP8266

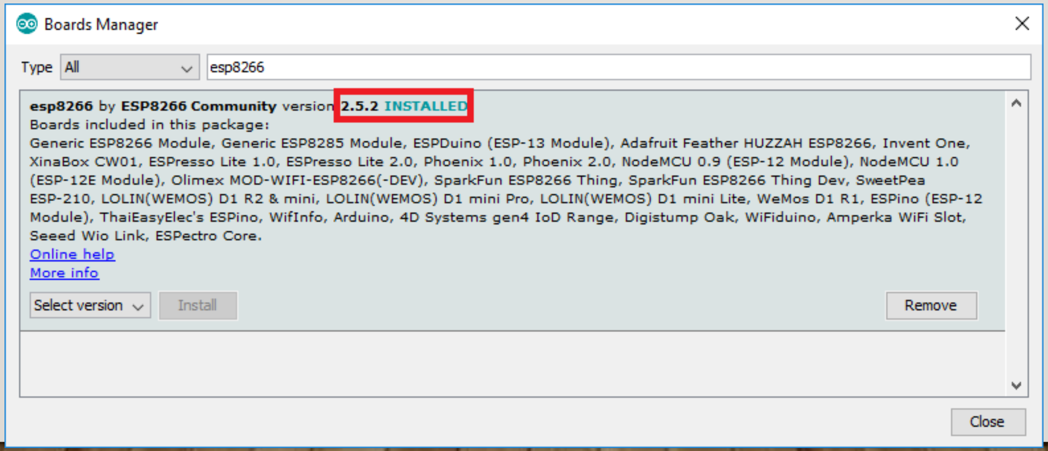

sketch_jul03b | Arduino 1.8.9
File Edit Sketch Tools Help
sketch_jul03b
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Manage Libraries... Ctrl+Shift+I
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+L
ESP32 Sketch Data Upload
ESP8266 Sketch Data Upload
WiFi101 / WiFiNINA Firmware Updater
Board: "Arduino/Genuino Uno"
Port: "COM5"
Get Board Info
Boards Manager...
Arduino AVR Boards

```
void
  //
}

void
  //
```

```
to run once:
```

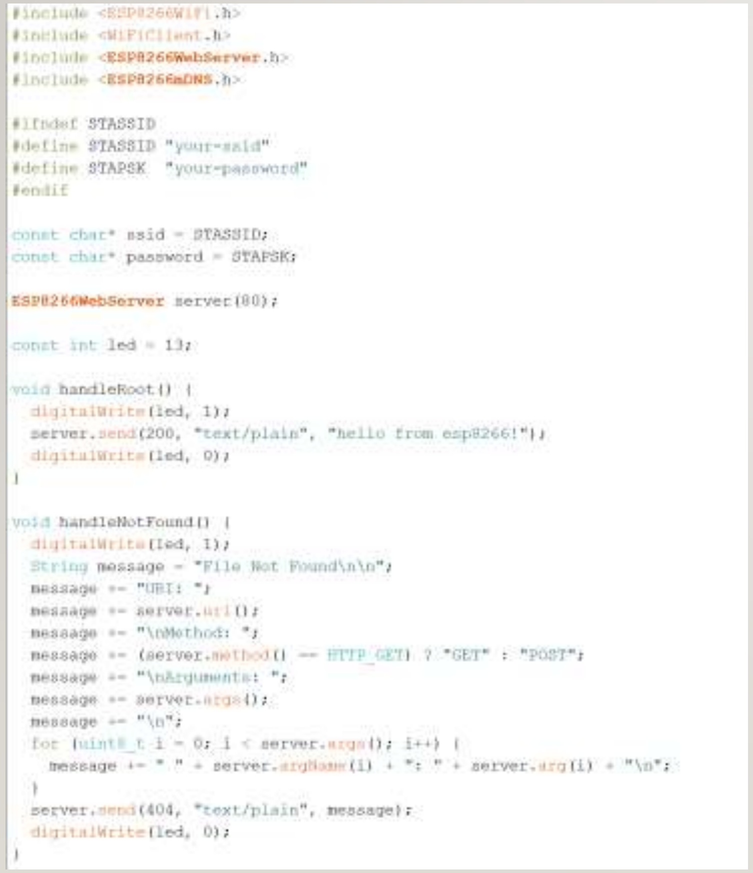


esp8266 by **ESP8266 Community**
Boards included in this package:
Generic ESP8266 Module, Generic ESP8285 Module, ESPDuino (ESP-13 Module), Adafruit Feather HUZZAH ESP8266, Invent One, XinaBox CW01, ESPresso Lite 1.0, ESPresso Lite 2.0, Phoenix 1.0, Phoenix 2.0, NodeMCU 0.9 (ESP-12 Module), NodeMCU 1.0 (ESP-12E Module), Olimex MOD-WIFI-ESP8266(-DEV), SparkFun ESP8266 Thing, SparkFun ESP8266 Thing Dev, SweetPea ESP-210, LOLIN(WEMOS) D1 R2 & mini, LOLIN(WEMOS) D1 mini Pro, LOLIN(WEMOS) D1 mini Lite, WeMos D1 R1, ESPino (ESP-12 Module), ThaiEasyElec's ESPino, WifInfo, Arduino, 4D Systems gen4 IoD Range, Digistump Oak, WiFiduino, Amperka WiFi Slot, Seeed Wio Link, ESPectro Core.
Online help
More info

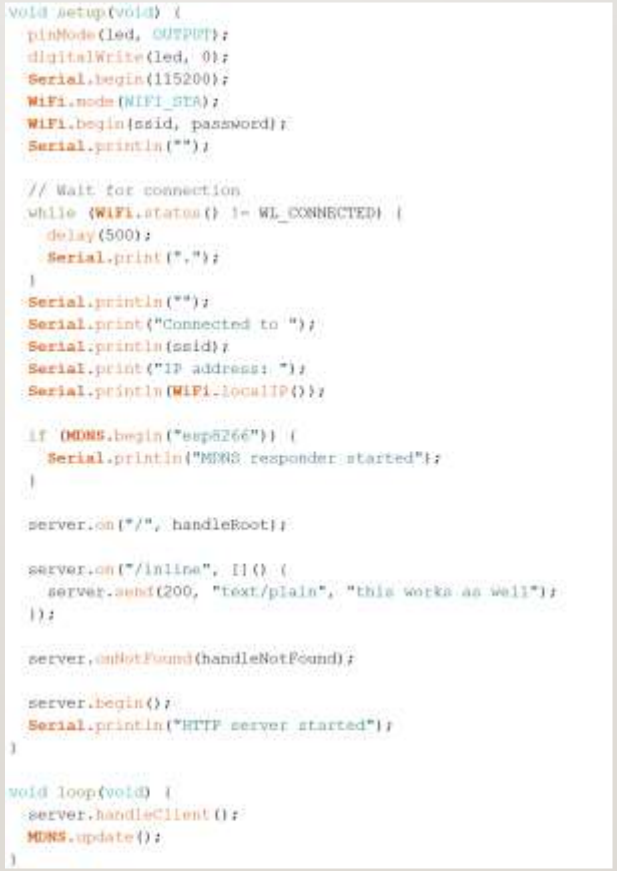


esp8266 by **ESP8266 Community** version **2.5.2 INSTALLED**
Boards included in this package:
Generic ESP8266 Module, Generic ESP8285 Module, ESPDuino (ESP-13 Module), Adafruit Feather HUZZAH ESP8266, Invent One, XinaBox CW01, ESPresso Lite 1.0, ESPresso Lite 2.0, Phoenix 1.0, Phoenix 2.0, NodeMCU 0.9 (ESP-12 Module), NodeMCU 1.0 (ESP-12E Module), Olimex MOD-WIFI-ESP8266(-DEV), SparkFun ESP8266 Thing, SparkFun ESP8266 Thing Dev, SweetPea ESP-210, LOLIN(WEMOS) D1 R2 & mini, LOLIN(WEMOS) D1 mini Pro, LOLIN(WEMOS) D1 mini Lite, WeMos D1 R1, ESPino (ESP-12 Module), ThaiEasyElec's ESPino, WifInfo, Arduino, 4D Systems gen4 IoD Range, Digistump Oak, WiFiduino, Amperka WiFi Slot, Seeed Wio Link, ESPectro Core.
Online help
More info

1ª PRACTICA: CREAR UN SERVIDOR

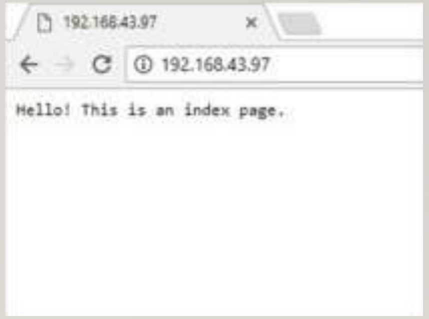


```
#include <ESP8266WiFi.h>
#include <WiFiClient.h>
#include <ESP8266WebServer.h>
#include <ESP8266mDNS.h>

#ifndef STASSID
#define STASSID "your-ssid"
#define STAPSK  "your-password"
#endif

const char* ssid = STASSID;
const char* password = STAPSK;

ESP8266WebServer server(80);

const int led = 13;

void handleRoot() {
  digitalWrite(led, 1);
  server.send(200, "text/plain", "hello from esp8266!");
  digitalWrite(led, 0);
}

void handleNotFound() {
  digitalWrite(led, 1);
  String message = "File Not Found\n\n";
  message += "URI: ";
  message += server.uri();
  message += "\nMethod: ";
  message += (server.method() == HTTP_GET) ? "GET" : "POST";
  message += "\nArguments: ";
  message += server.args();
  message += "\n";
  for (uint8_t i = 0; i < server.args(); i++) {
    message += " " + server.argName(i) + ": " + server.arg(i) + "\n";
  }
  server.send(404, "text/plain", message);
  digitalWrite(led, 0);
}
```



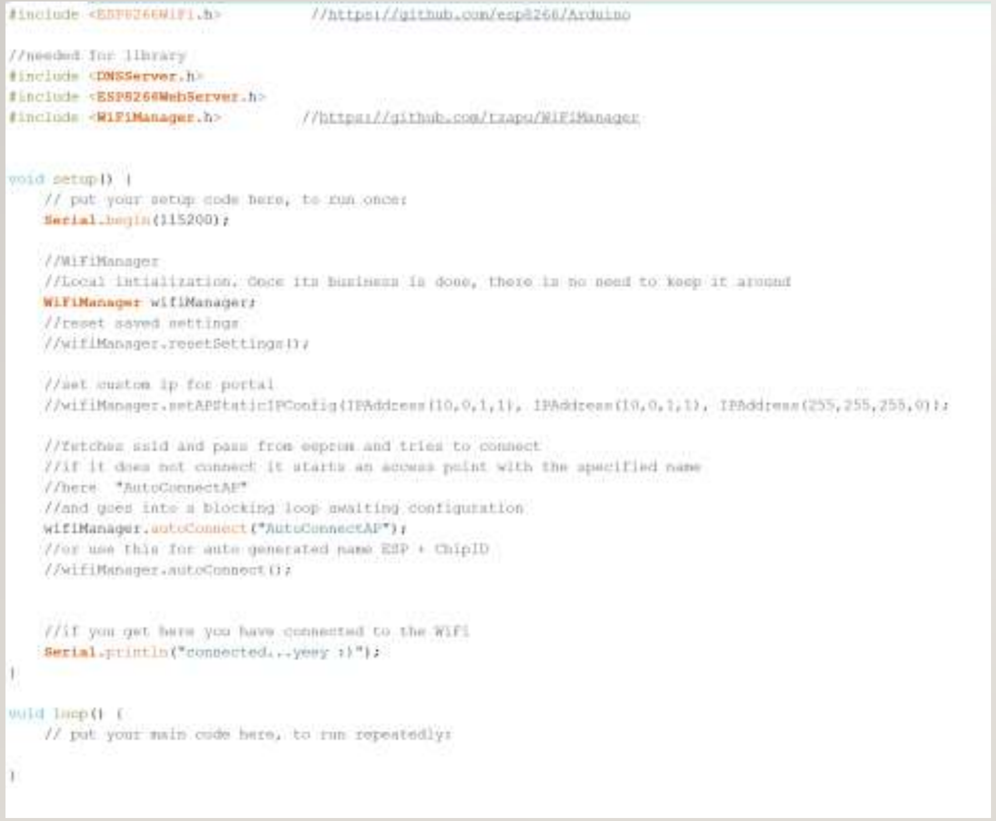
```
void setup(void) {
  pinMode(led, OUTPUT);
  digitalWrite(led, 0);
  Serial.begin(115200);
  WiFi.mode(WIFI_STA);
  WiFi.begin(ssid, password);
  Serial.println("");

  // Wait for connection
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.print("Connected to ");
  Serial.println(ssid);
  Serial.print("IP address: ");
  Serial.println(WiFi.localIP());

  if (MDNS.begin("esp8266")) {
    Serial.println("MDNS responder started");
  }

  server.on("/", handleRoot);

  server.on("/inline", []() {
    server.send(200, "text/plain", "this works as well");
  });

  server.onNotFound(handleNotFound);

  server.begin();
  Serial.println("HTTP server started");
}

void loop(void) {
  server.handleClient();
  MDNS.update();
}
```

+

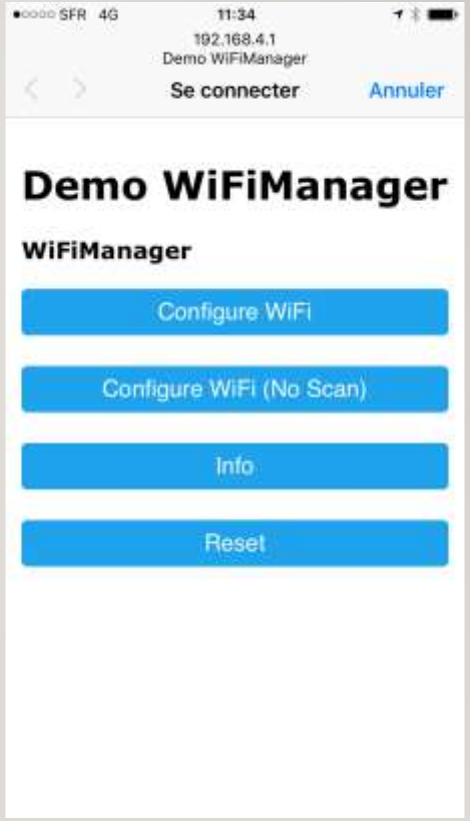


```
Hello! This is an index page.
```

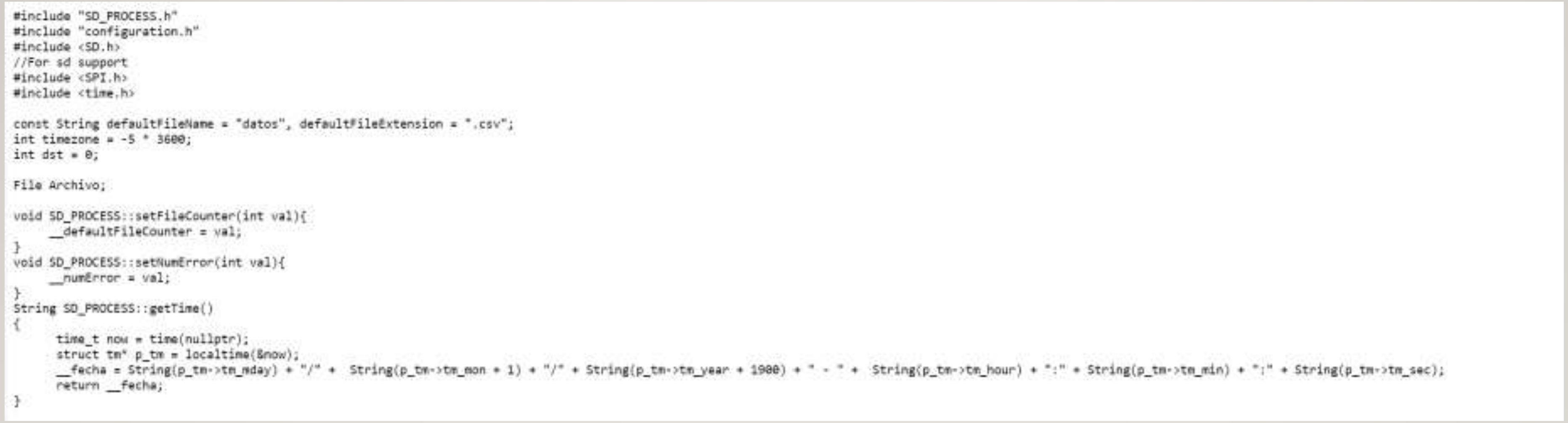
2ª PRACTICA: CONECTARSE A UN WIFI



```
#include <ESP8266WiFi.h>          //https://github.com/esp8266/Arduino

//needed for library
#include <DNSServer.h>
#include <ESP8266WebServer.h>
#include <WiFiManager.h>         //https://github.com/tzapu/WiFiManager


void setup() {
    // put your setup code here, to run once:
    Serial.begin(115200);

    //WiFiManager
    //Local intialization. Once its business is done, there is no need to keep it around
    WiFiManager wifiManager;
    //reset saved settings
    //wifiManager.resetSettings();

    //set custom ip for portal
    //wifiManager.setAPStaticIPConfig(IPAddress(10,0,1,1), IPAddress(10,0,1,1), IPAddress(255,255,255,0));

    //fetches ssid and pass from eeprom and tries to connect
    //if it does not connect it starts an access point with the specified name
    //here  "AutoConnectAP"
    //and goes into a blocking loop awaiting configuration
    wifiManager.autoConnect("AutoConnectAP");
    //or use this for auto generated name ESP + ChipID
    //wifiManager.autoConnect();


    //if you get here you have connected to the WiFi
    Serial.println("connected...yeey :)");
}

void loop() {
    // put your main code here, to run repeatedly:

}
```



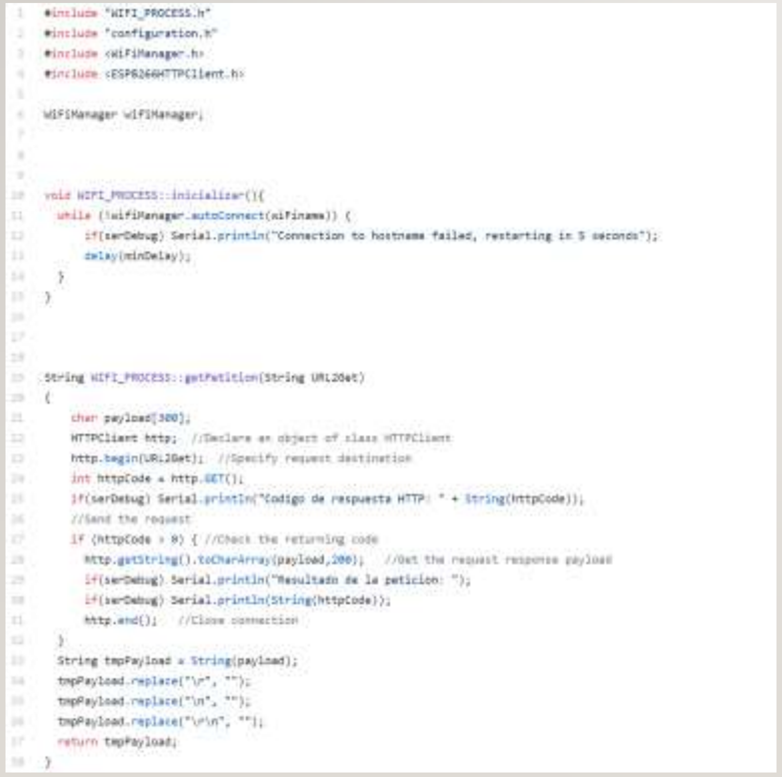
3ª PRACTICA: OBTENER LA HORA


```
#include "SD_PROCESS.h"
#include "configuration.h"
#include <SD.h>
//For sd support
#include <SPI.h>
#include <time.h>

const String defaultFileName = "datos", defaultFileExtension = ".csv";
int timezone = -5 * 3600;
int dst = 0;

File Archivo;

void SD_PROCESS::setFileCounter(int val){
	__defaultFileCounter = val;
}
void SD_PROCESS::setNumError(int val){
	__numError = val;
}
String SD_PROCESS::getTime()
{
	time_t now = time(nullptr);
	struct tm* p_tm = localtime(&now);
	__fecha = String(p_tm->tm_mday) + "/" +  String(p_tm->tm_mon + 1) + "/" + String(p_tm->tm_year + 1900) + " - " +  String(p_tm->tm_hour) + ":" + String(p_tm->tm_min) + ":" + String(p_tm->tm_sec);
	return __fecha;
}
```

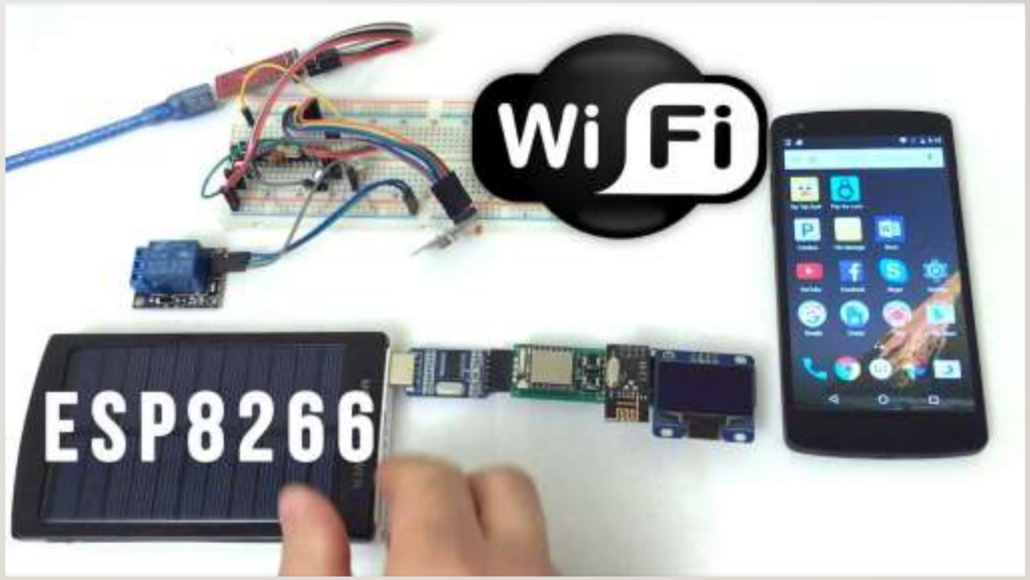
https://github.com/Yercar18/Dronefenix/blob/master/AirQ_Wemos/SD_PROCESS.cpp

4ª PRACTICA: PETICIÓN GET

https://github.com/Yercar18/Dronefenix/blob/master/AirQ_Wemos/WIFI_PROCESS.cpp



```cpp
#include "WIFI_PROCESS.h"
#include "configuration.h"
#include <WiFiManager.h>
#include <ESP8266HTTPClient.h>

WiFiManager wifiManager;



void WIFI_PROCESS::inicializar(){
  while (!wifiManager.autoConnect(wifiname)) {
      if(serDebug) Serial.println("Connection to hostname failed, restarting in 5 seconds");
      delay(minDelay);
  }
}



String WIFI_PROCESS::getPetition(String URL2Get)
{
    char payload[300];
    HTTPClient http;  //Declare an object of class HTTPClient
    http.begin(URL2Get);  //Specify request destination
    int httpCode = http.GET();
    if(serDebug) Serial.println("Codigo de respuesta HTTP: " + String(httpCode));
    //Send the request
    if (httpCode > 0) { //Check the returning code
      http.getString().toCharArray(payload,300);   //Get the request response payload
      if(serDebug) Serial.println("Resultado de la peticion: ");
      if(serDebug) Serial.println(String(httpCode));
      http.end();   //Close connection
  }
  String tmpPayload = String(payload);
  tmpPayload.replace("\r", "");
  tmpPayload.replace("\n", "");
  tmpPayload.replace("\r\n", "");
  return tmpPayload;
}
```



PREGUNTAS

TRABAJO INVESTIGATIVO

- Realice una petición post a un servidor local (Python – Node – Ruby ….)
- Usando el modulo SD comunique un archivo de texto en el servidor local.
- Con el led 13 instalado en la tarjeta Arduino realice el encendido y apagado remoto.