

*Tutoriales*  
*Diseño en*  
***Medios***  
***Interactivos***

# 03

## *Arduino Uno Wifi: configuración*

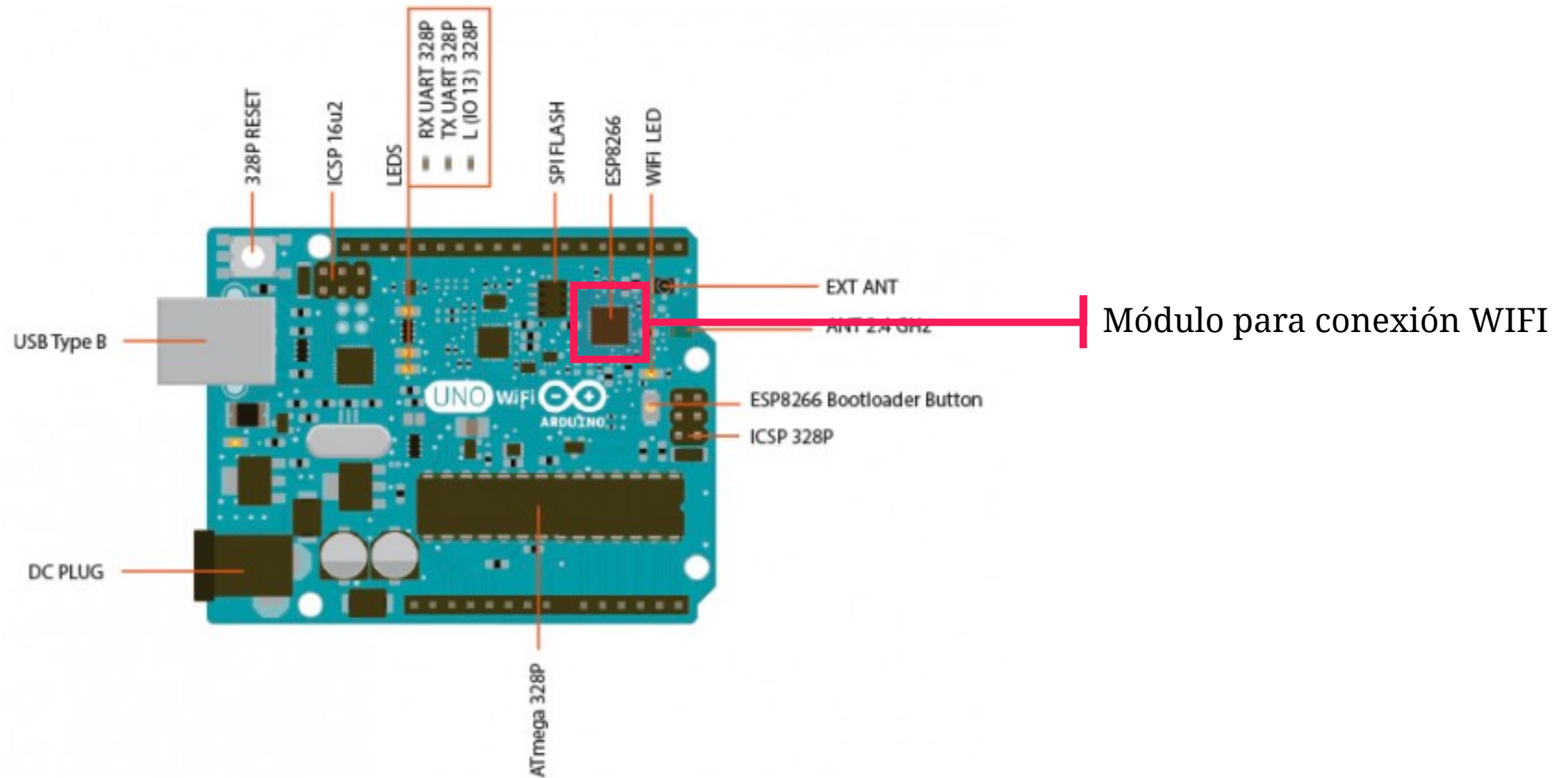
- a.*** Configurar Arduino
- b.*** Configurar Arduino IDE
- c.*** Controlar la placa con Processing

La tarjeta **Arduino UNO Wifi** es una placa Arduino UNO con un modulo **ESP8266** integrado el cual permite conectividad wifi y capacidad de programar la placa **OTA** (over the air)

*a.*

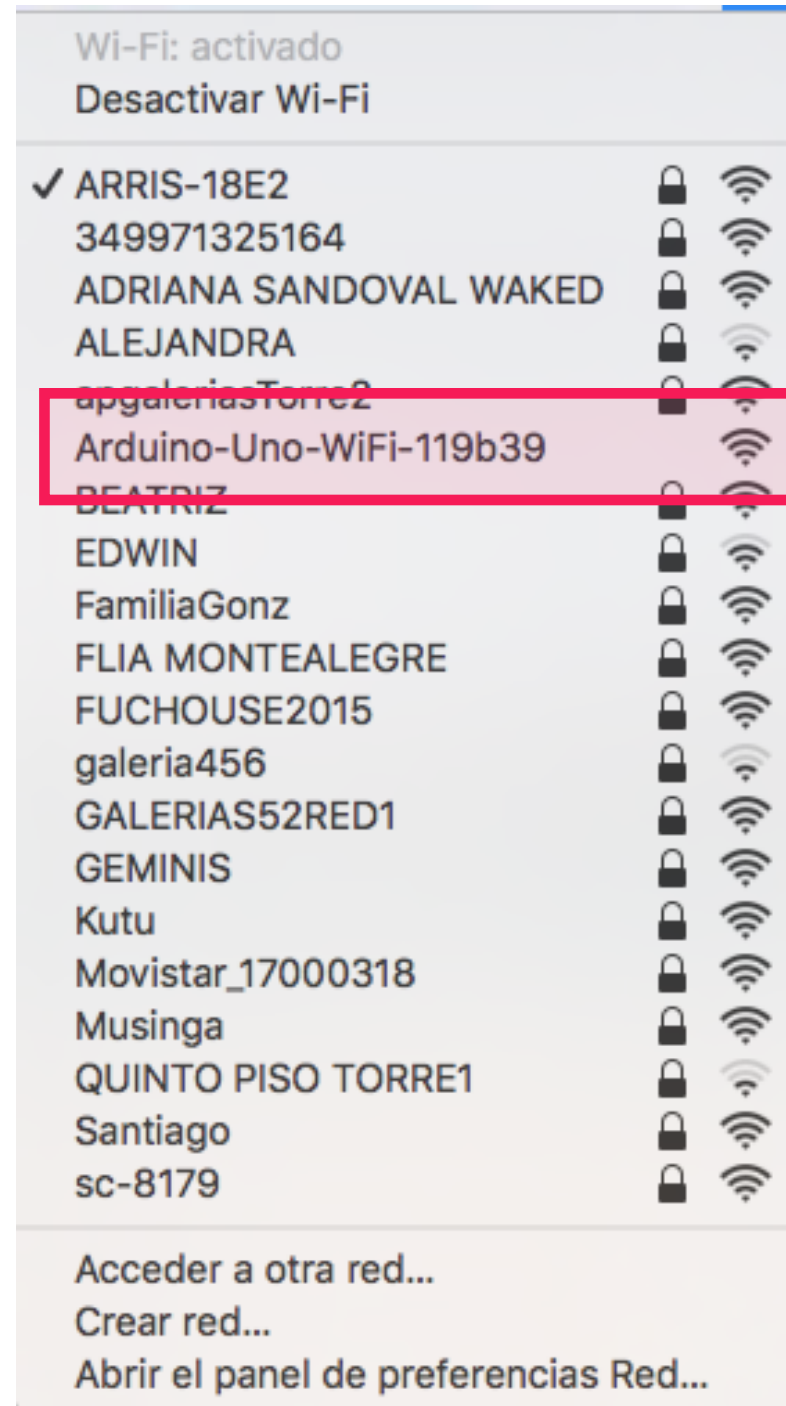
*Configurar Arduino*

1. Conectar la placa con un cable **USB B** (de conectar impresora) a un computador o a una fuente de poder

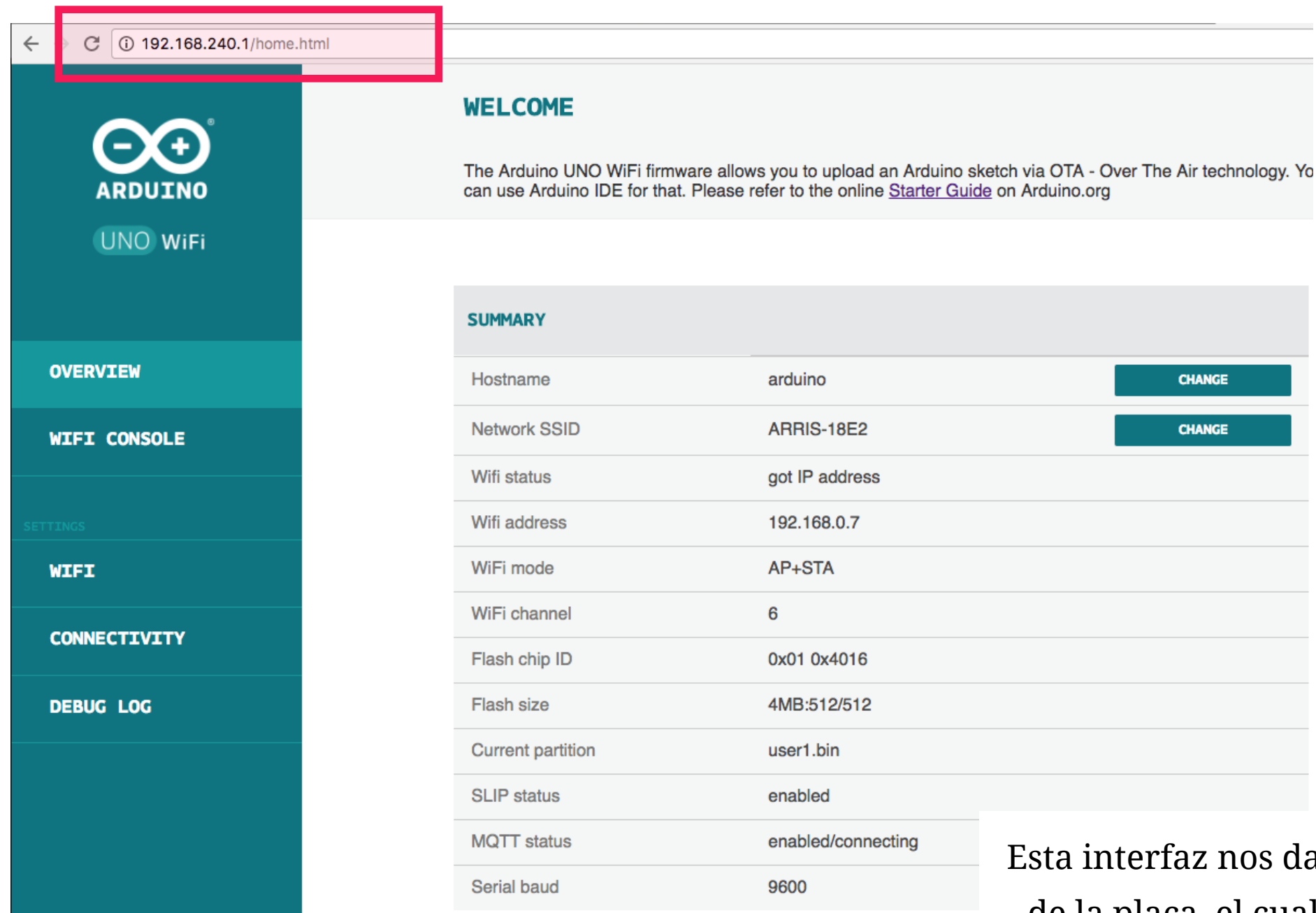


2. Esperar **un minuto** despues de que el LED de la placa muestre que está encendida

3. Ir a la ventana de redes **WIFI** del computador. Debe aparecer una red con nombre **Arduino-Uno-WiFi-xxxxxx**. Conectarse a esta red



**4.** En un navegador ir a la dirección IP: **192.168.240.1** donde aparece la siguiente interfaz:



**WELCOME**

The Arduino UNO WiFi firmware allows you to upload an Arduino sketch via OTA - Over The Air technology. You can use Arduino IDE for that. Please refer to the online [Starter Guide](#) on Arduino.org

SUMMARY		
Hostname	arduino	<a href="#">CHANGE</a>
Network SSID	ARRIS-18E2	<a href="#">CHANGE</a>
Wifi status	got IP address	
Wifi address	192.168.0.7	
WiFi mode	AP+STA	
WiFi channel	6	
Flash chip ID	0x01 0x4016	
Flash size	4MB:512/512	
Current partition	user1.bin	
SLIP status	enabled	
MQTT status	enabled/connecting	
Serial baud	9600	

Esta interfaz nos da acceso al **Firmware** de la placa, el cual permite controlar el arduino sin necesidad de una conexión física con un computador

## 4. Ir a la ventana **WIFI** de la página



SUMMARY
Hostname
Network SSID
Wifi status
Wifi address
WiFi mode
WiFi channel
Flash chip ID
Flash size
Current partition
SLIP status
MQTT status

**5.** En esta página se puede cambiar el **Hostname** del arduino si se desea (no es necesario)

## WIFI CONFIGURATION

### HOSTNAME

arduino

CHANGE

### WIFI CONNECTION

To connect to a WiFi network, please select one of the detected networks, enter the password, and hit the connect button...

Network SSID

- ☐ -79dB Movistar\_17000318
- ☐ -91dB Movistar\_15251651
- ☐ -92dB sc-8179
- ☐ -89dB Movistar\_15163121
- ☒ -31dB ARRIS-18E2
- ☐ -82dB 349971325164
- ☐ -93dB Kutu

### WIFI STATUS

WiFi channel	6
Configured network	ARRIS-18E2
WiFi status	got IP address
WiFi address	192.168.0.7
WiFi rssi	-34dB
WiFi phy	11n
WiFi MAC	5c:cf:7f:11:9b:39
WiFi mode	AP+STA

SWITCH TO STA MODE

### ADVANCED

- ☒ DHCP
- ☐ Static IP

CHANGE



6. En la misma ventana conectar a la **red Wifi** a la que se tenga acceso en el momento.

**WIFI CONNECTION**

To connect to a WiFi network, please select one of the detected networks, enter the password, and hit the connect button...

Network SSID

- ☐ -78dB Movistar\_17000318
- ☐ -94dB Movistar\_15251651
- ☐ -87dB Movistar\_15163121
- ☐ -86dB galeria456
- ☐ -86dB 349971325164
- ☐ -82dB FLIA MONTEALEGRE
- ☒ -32dB ARRIS-18E2
- ☐ -93dB SOTRANSVEGA SAS
- ☐ -79dB FAMILIAESCE
- ☐ -85dB FamiliaGonz
- ☐ -88dB Orion
- ☐ -93dB FAJARDO RODRIGUEZ
- ☐ -91dB FUCHOUSE2015
- ☐ -88dB Musinga
- ☐ -70dB GEMINIS
- ☐ Hidden network

WiFi password, if applicable:

password

**CONNECT**

Seleccionar red

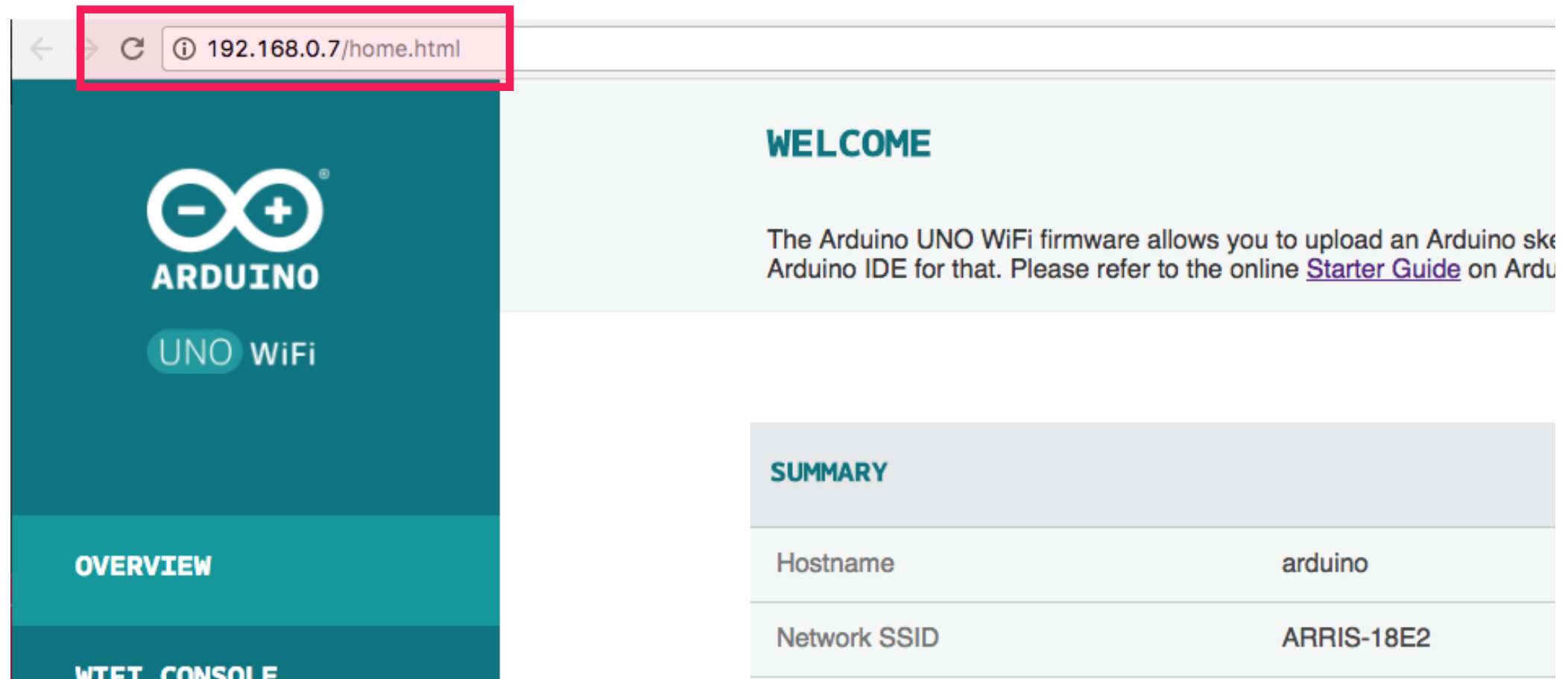
Escribir contraseña

Conectar

7. Después de la conexión aparecerá una pantalla con la **IP** que se le asignó al arduino. Ir a esa dirección.



8. Esta es la nueva IP del Arduino y desde la cual se trabajará. La dirección puede cambiar cada vez que se conecte a una red Wifi




Después de haber configurado la placa  
se correrá un programa en Arduino  
para controlar la placa via Wifi

***b.***

*Configurar Arduino IDE*

1. Tener instalada la última versión del IDE de Arduino ([arduino.cc/en/Main/Software](https://arduino.cc/en/Main/Software))

## Download the Arduino IDE




### ARDUINO 1.8.5

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software.

This software can be used with any Arduino board. Refer to the [Getting Started](#) page for Installation instructions.

**Windows** Installer  
**Windows** ZIP file for non admin install

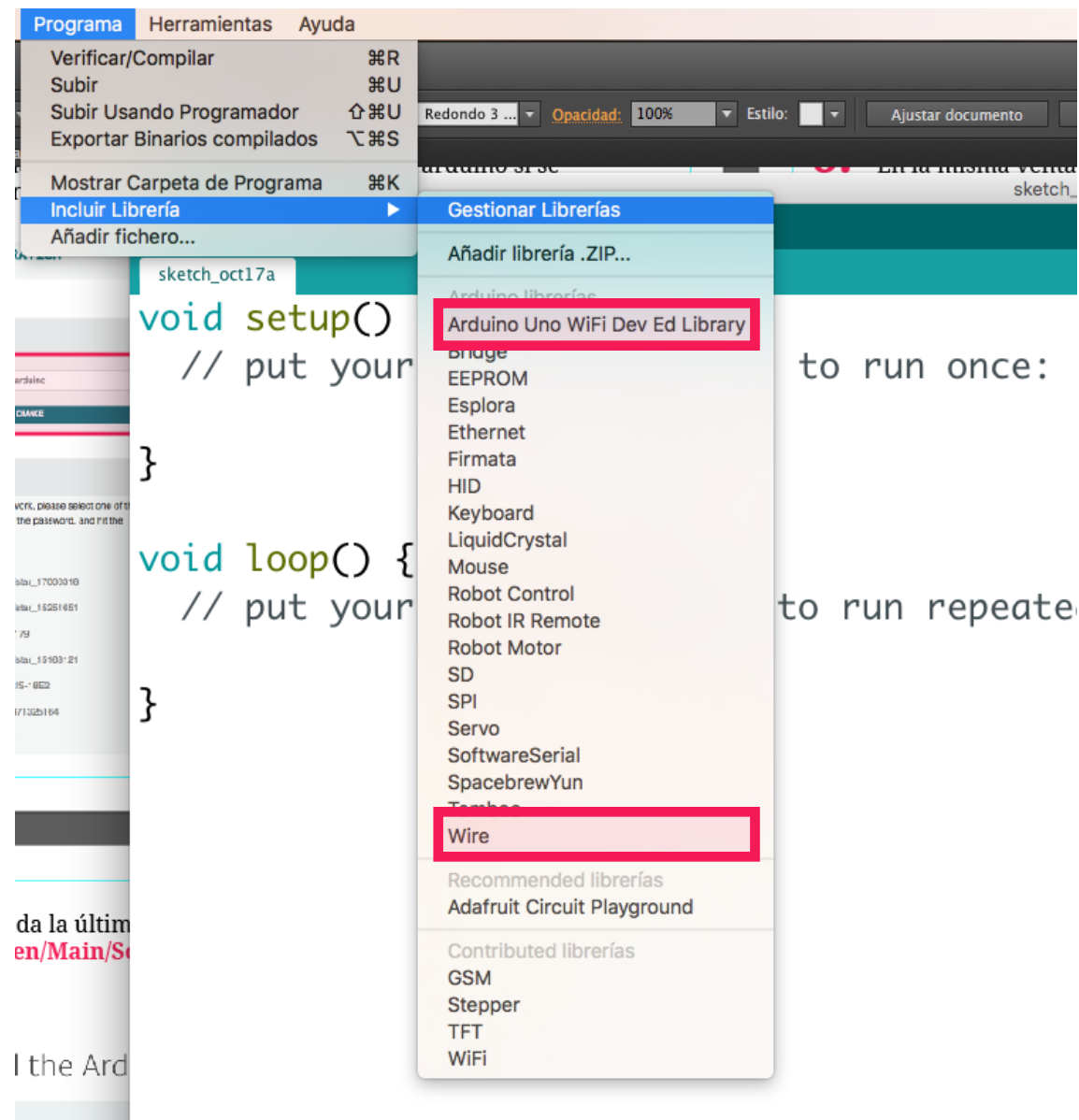
**Windows app** 

**Mac OS X** 10.7 Lion or newer

**Linux** 32 bits  
**Linux** 64 bits  
**Linux** ARM

[Release Notes](#)  
[Source Code](#)  
[Checksums \(sha512\)](#)

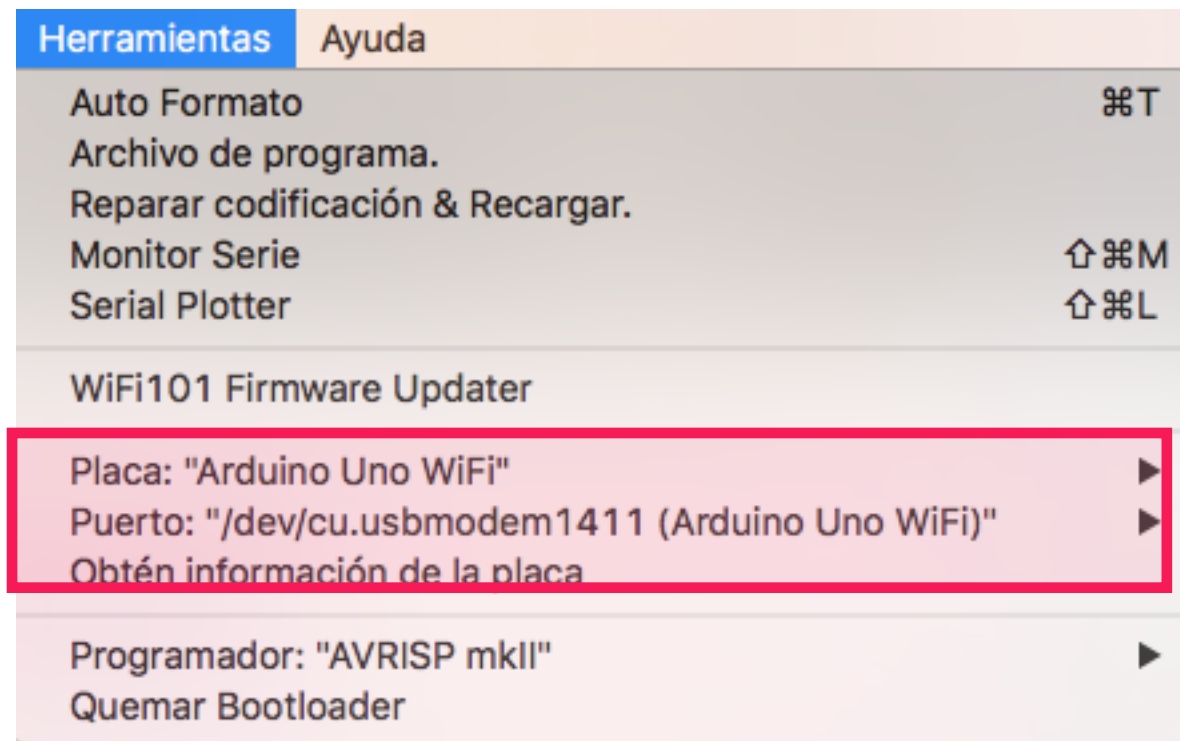
## 2. Crear un nuevo programa de Arduino e importar las librerías **Arduino Uno Wifi Dev Ed Library** y **Wire**



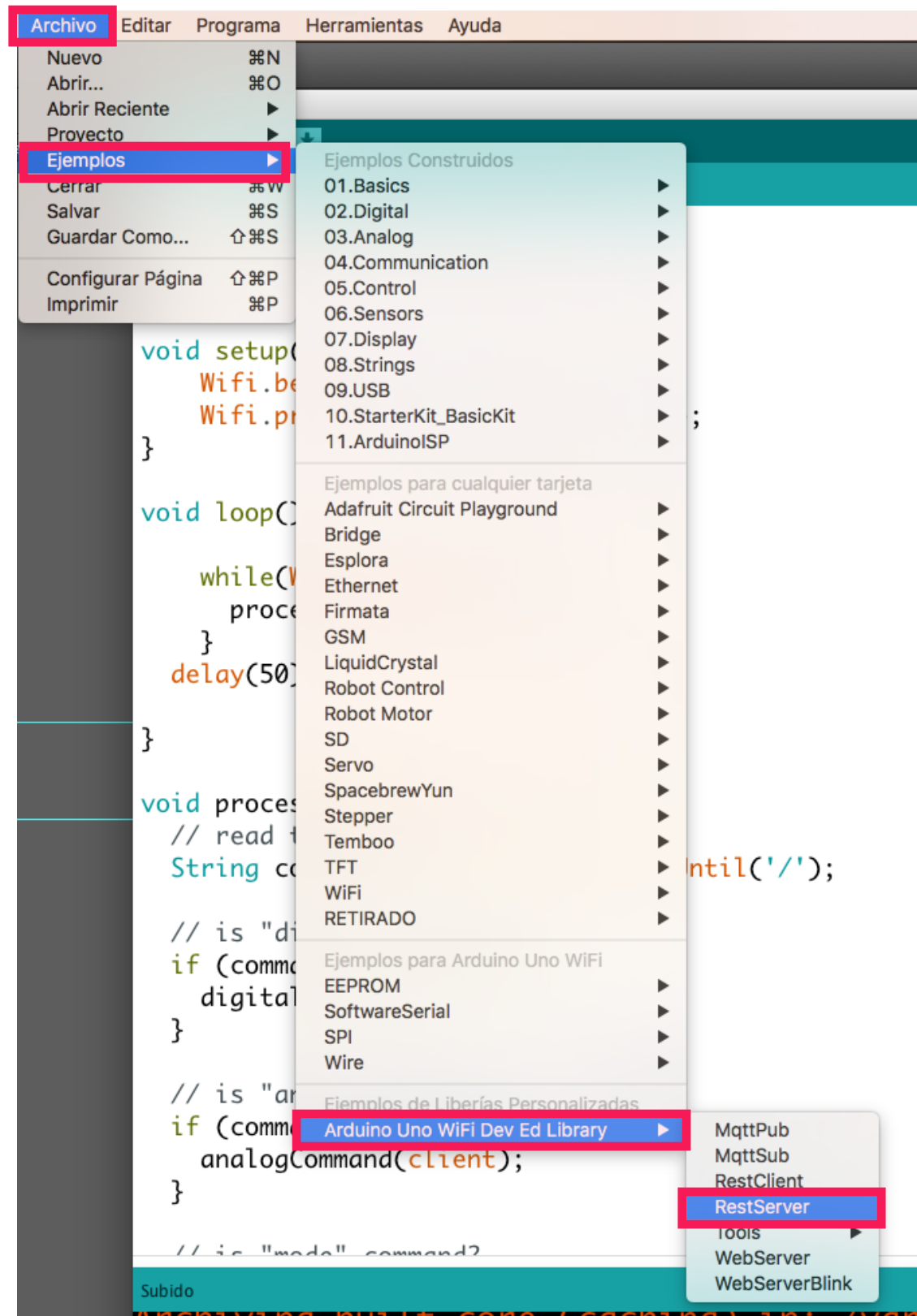
Si la librería de Arduino Wifi no aparece en las opciones se puede descargar acá:

[github.com/arduino-libraries/UnoWiFi-Developer-Edition-Lib/releases/tag/0.0.3](https://github.com/arduino-libraries/UnoWiFi-Developer-Edition-Lib/releases/tag/0.0.3)

### 3. Asegurarse de que están seleccionados la placa y el puerto correcto



## 4. Abrir el ejemplo RestServer y subirlo a la placa



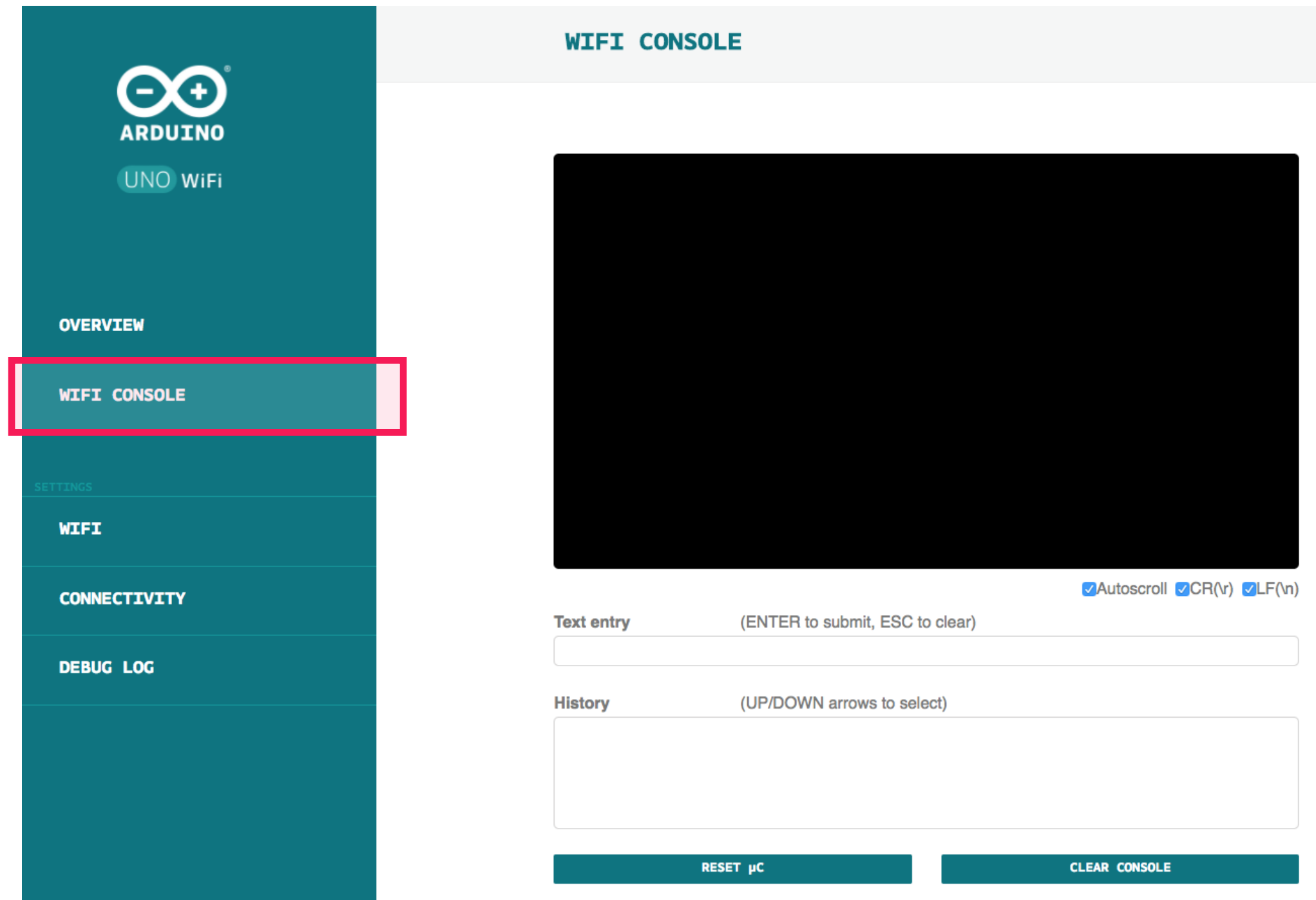
Este es un ejemplo de Arduino que permite acceder a los pines de la placa utilizando **REST**

**REST** es una forma de comunicación en la cual el servidor esta esperando comandos para **cambiar de estado**



## 5. Controlar un LED de la placa a través de Wifi

*a.* Volver a la interfaz del Firmware en el navegador e ir a **Wifi console**



***b.*** Definir en la consola el LED 13 como un pin de salida con **/arduino/mode/13/output**

The screenshot shows a web-based interface for interacting with an Arduino. At the top, a black console area displays the response "HTTP/1.1 200 OK" and "Pin D13 configured as OUTPUT!". Below this is a "Text entry" field with a blue border and a red highlight, containing a cursor. To the right of the text entry field are three checked checkboxes: "Autoscroll", "CR(\r)", and "LF(\n)". Below the text entry field is a "History" list with a red highlight, showing the command "/arduino/mode/13/output".

HTTP/1.1 200 OK  
Pin D13 configured as OUTPUT!

✓Autoscroll ✓CR(\r) ✓LF(\n)

Text entry (ENTER to submit, ESC to clear)

History (UP/DOWN arrows to select)

/arduino/mode/13/output

Mensaje de respuesta

Escribir comando

Comandos usados

En este programa se pueden utilizar los siguientes comandos:

```
"/arduino/digital/13" -> digitalWrite(13)
"/arduino/digital/13/1" -> digitalWrite(13, HIGH)
"/arduino/analog/2/123" -> analogWrite(2, 123)
"/arduino/analog/2" -> analogRead(2)
"/arduino/mode/13/input" -> pinMode(13, INPUT)
"/arduino/mode/13/output" -> pinMode(13, OUTPUT)
```

**C.** Escribir en la consola **arduino/digital/13/1** para encender el LED.  
Después **arduino/digital/13/0** para apagarlo

```
HTTP/1.1 200 OK
```

```
Pin D13 configured as OUTPUT!
```

```
HTTP/1.1 200 OK
```

```
Pin D13 set to 0
```

```
HTTP/1.1 200 OK
```

```
Pin D13 set to 1
```

```
HTTP/1.1 200 OK
```

```
Pin D13 set to 0
```

Para utilizar los comandos de la placa no es necesario hacerlo desde la consola de la página. También se puede acceder por la **url** del comando

C.

*Controlar la placa con Processing*

# 1. Instalar y abrir Processing ([processing.org/download/](https://processing.org/download/))

Download Processing. Processing is available for Linux, Mac OS X, and Windows. Select your choice to download the software below.



3.3.6 (4 September 2017)

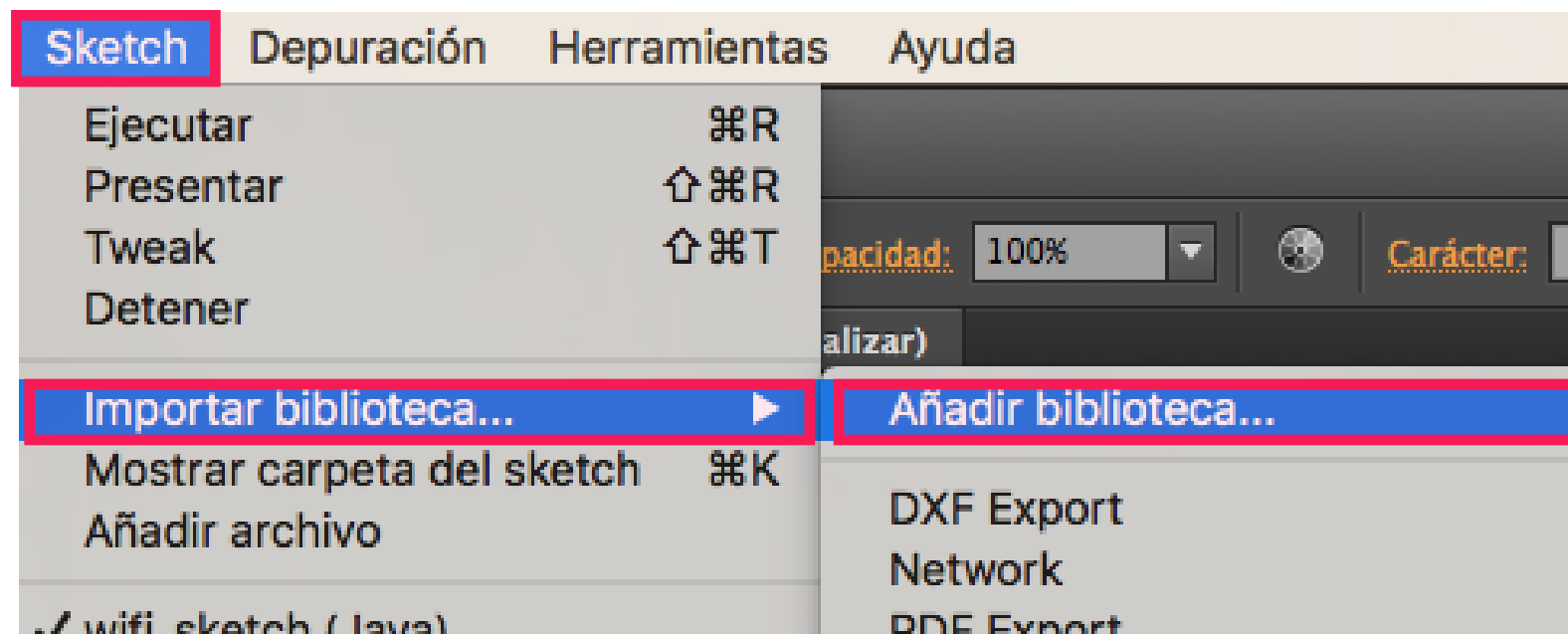
[Windows](#) 64-bit  
[Windows](#) 32-bit

[Linux](#) 64-bit  
[Linux](#) 32-bit  
[Linux](#) ARMv6hf

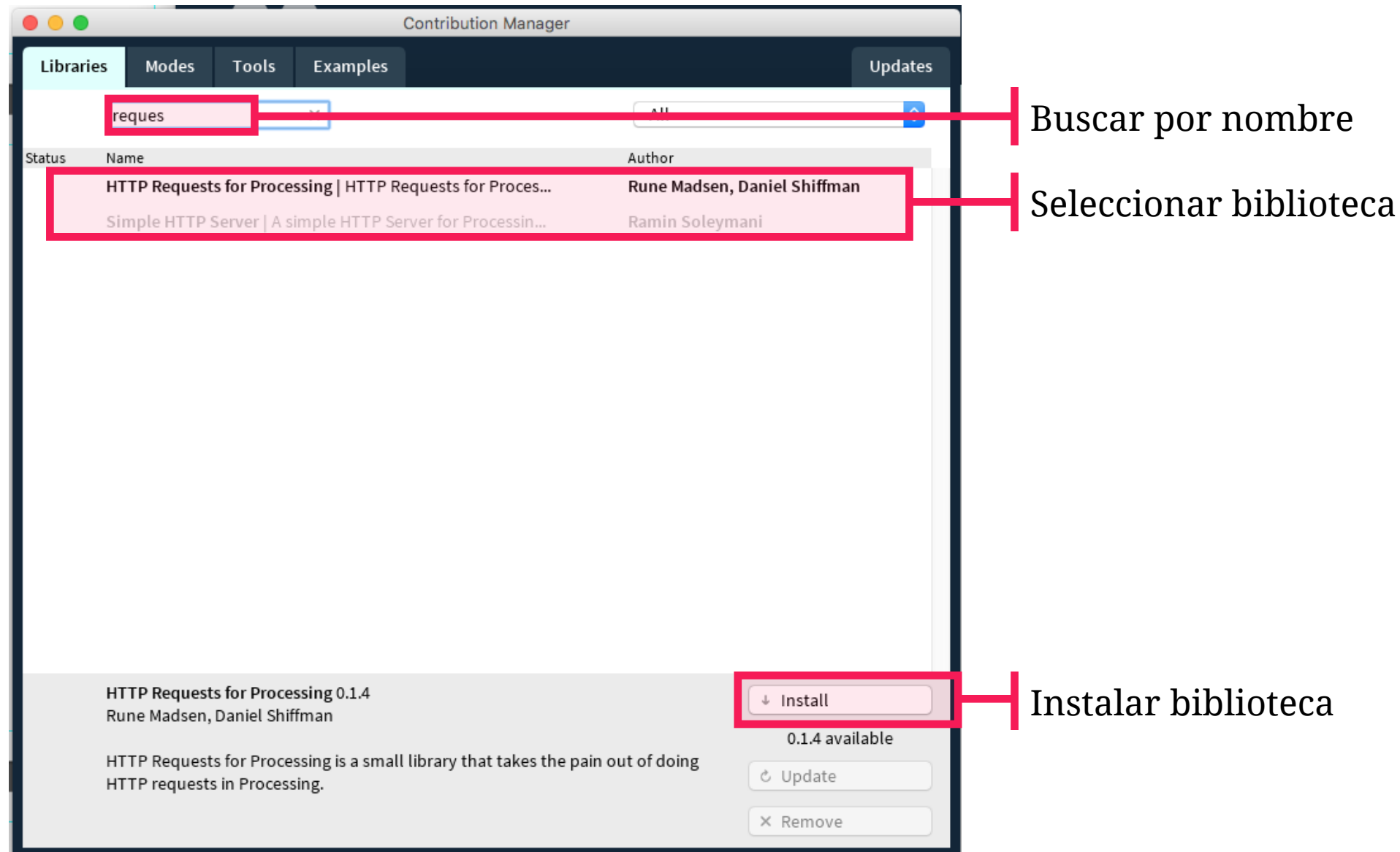
[Mac OS X](#)

## 2. Instalar Biblioteca **HTTP Requests for Processing**

*a.* Ir a sketch > Importar biblioteca > Añadir Biblioteca



**b.** En la ventana que aparece buscar la biblioteca **HTTP Requests for Processing**



Buscar por nombre

Seleccionar biblioteca

Instalar biblioteca

Esta biblioteca permite realizar **peticiones a una url**. En este caso se hará una petición llamada **GET** la cual busca recuperar datos de la **url**

### 3. Escribir un programa que encienda y apague el LED via Wifi al oprimir un botón en la pantalla

```
//importar biblioteca para hacer http requests
import http.requests.*;

//variable que guarda el estado del botón
boolean prendido = false;

void setup() {
  size(500, 500);
  noStroke();
  fill(255, 0, 0);
}

void draw() {
  background(255);

  //dibujar elipse del botón
  ellipse(width/2, height/2, 150, 150);
}
```

Este código se puede ver y descargar en: [laurajunco.github.io/medios/wifi\\_sketch/wifi\\_sketch.pde](https://laurajunco.github.io/medios/wifi_sketch/wifi_sketch.pde)

```

//función que se activa si el mouse es oprimido
void mousePressed() {

    //revisa que el mouse este tocando el botón
    if (dist(mouseX, mouseY, width/2, height/2) < 150) {
        //si el led esta apagado
        if (prendido == false) {
            //cambia el color a verde
            fill(0, 255, 0);

            //llamar url que prende el LED
            GetRequest get = new GetRequest("http://192.168.0.8/arduino/digital/13/1");
            get.send();

            //cambia el estado a prendido
            prendido = true;

            //si el led esta prendido
        } else {
            //cambia el color a rojo
            fill(255, 0, 0);

            //llamar url que apaga el LED
            GetRequest get = new GetRequest("http://192.168.0.8/arduino/digital/13/0");
            get.send();

            //cambia el estado a apagado
            prendido = false;
        }
    }
}

```

El formato de la url es **http://IPdelarduino/arduino/digital/Pin/**



### 3. Correr el programa. Hacer clic en la elipse



```
Pin D13 set to 0  
HTTP/1.1 200 OK
```



```
Pin D13 set to 1  
HTTP/1.1 200 OK
```