

# Sistema de Domótica

## Instructivos

### Breve Descripción

Este proyecto inició dentro del ámbito académico en la Universidad Nacional de Córdoba - Facultad de Ciencias Exactas Físicas y Naturales. En él se desarrolló un sistema de domótica activado por voz y empleando Google Voice Acces, una aplicación creada con MIT App Inventor, una placa ESP8266 NodeMCU programada utilizando la IDE de Arduino y un juego de 2 relés y un interruptor.

### Para quién se hizo y con qué objeto

Se realizó para un centro de rehabilitación de acuerdo con los deseos de la ingeniera que allí trabaja de acercar la experiencia de domótica a muchos pacientes que podrían ser potenciales usuarios de un sistema así para ser más independientes.

Sin embargo, la falta de contacto de los pacientes con estas tecnologías, hace que esto resulte una opción muy distante, desconocida y hasta de ciencia ficción; mientras que aquellos que, si han oído de domótica, lo ven como una opción demasiado costosa sin comprobar previamente su utilidad.

Este proyecto, si bien puede ser utilizado por todas las personas que acuden a dicha institución, fue pensado especialmente para pacientes con lesiones medulares altas o con algún otro tipo de lesión que les impida desplazarse por sí mismos y que además presenten pérdida del control motor fino. De esta manera, el control de los artefactos electrónicos de la casa solo a través del uso de la voz se convierte en una de sus mayores ventajas.

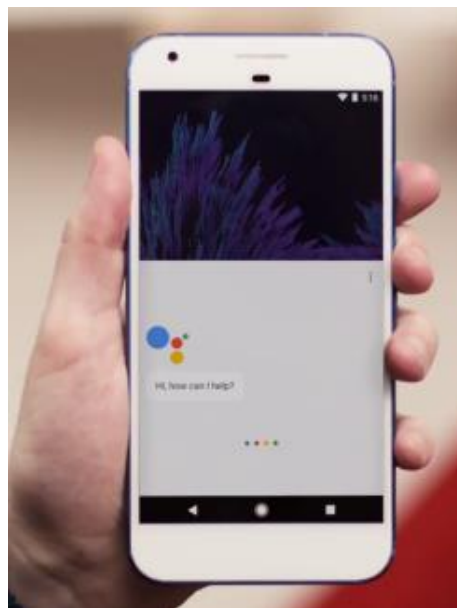
### Materiales

- Taladro
- Destornillador Phillips
- Llaves alem
- Solador de estaño
- Cables Dupont o Jumper
- Cables de cobre de 6mm de diámetro
- Tacos Fisher
- Tornillos varios
- Tester
- Cinta aisladora

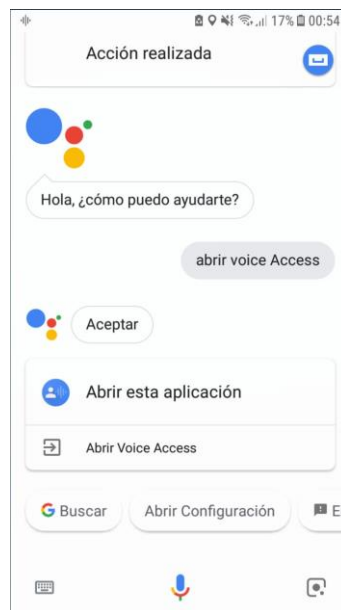
- Pistola de silicona
- NodeMCU ESP8266 con su cable USB-microUSB para programarlo
- Módulo de 2 Relés de 5V
- Transformador de celular o 220V AC a 5V DC – 1A
- Porta lámparas
- Tecla y tomacorriente acoplados
- Celular con Android, Google Assistant y Google Voice Access correctamente instalados y configurados.

## Modo de Funcionamiento

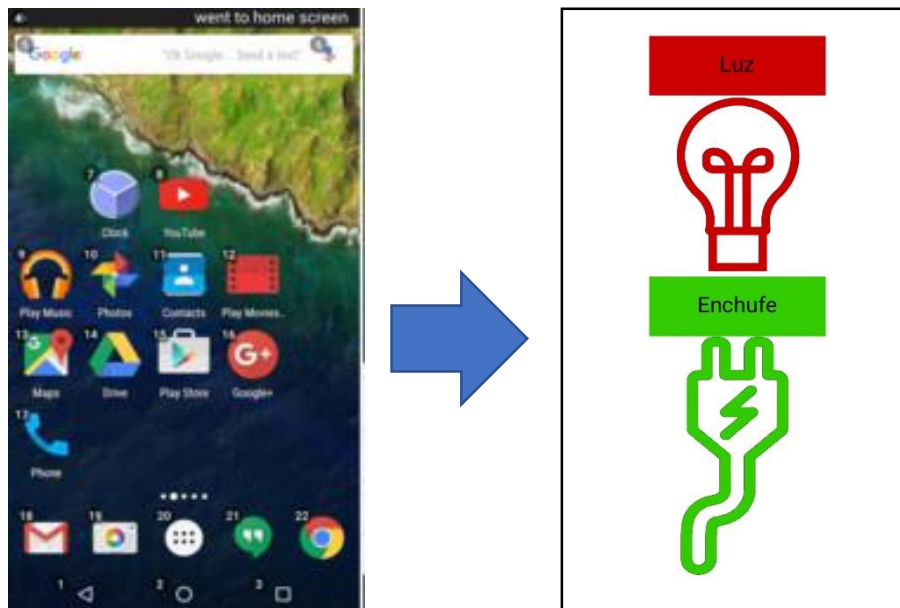
1. Encender el sistema a través de un interruptor a un lado de la caja estanca, si no estuviere encendido.
2. Decir “Ok Google”, para iniciar el Asistente por voz de Google en el celular con Android que se requiere.



3. Decir el comando “Abrir Voice Access”



4. Con Voice Access ejecutándose, decir “Abir Domótica”, o el nombre de la aplicación desarrollada en MIT App Inventor (explicado en la siguiente sección).



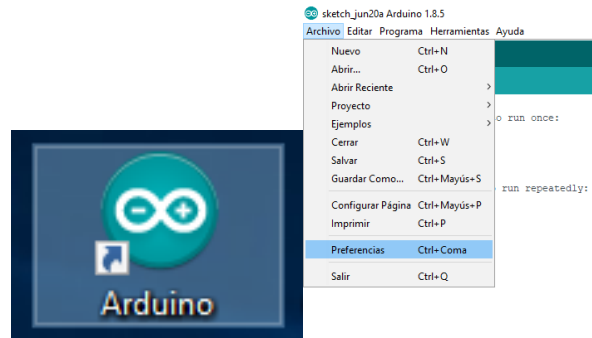
5. Con la aplicación abierta, y Voice Access activado, cualquier frase que contenga las palabras en los botones (“Luz” y “Enchufe”) accionarán el prendido o apagado de los elementos asociados a ellos.
6. Para desactivar Voice Access y dejar de utilizar el sistema, se debe decir “Notificaciones” y luego “Voice Access”.

# Proceso de Elaboración

## 1º Etapa – Programación del NodeMCU

### Paso 1 - Instalación de la librería

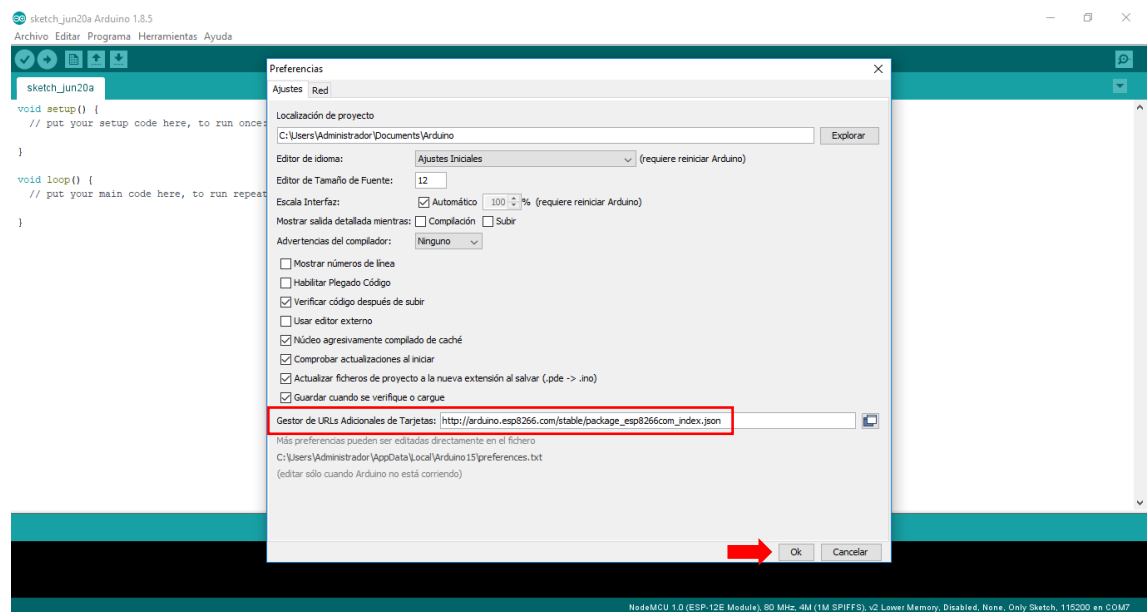
- a) Abrir la IDE de Arduino y dirigirse al menú Archivo → Preferencias



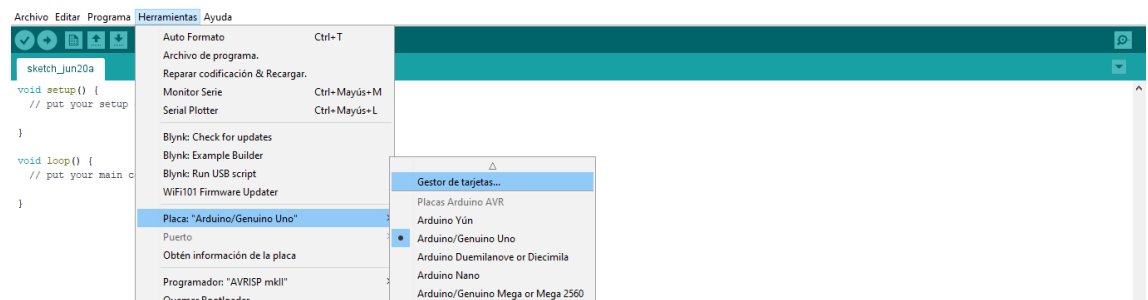
- b) En la ventana preferencias, copiar y pegar el siguiente link:

[http://arduino.esp8266.com/stable/package\\_esp8266com\\_index.json](http://arduino.esp8266.com/stable/package_esp8266com_index.json)

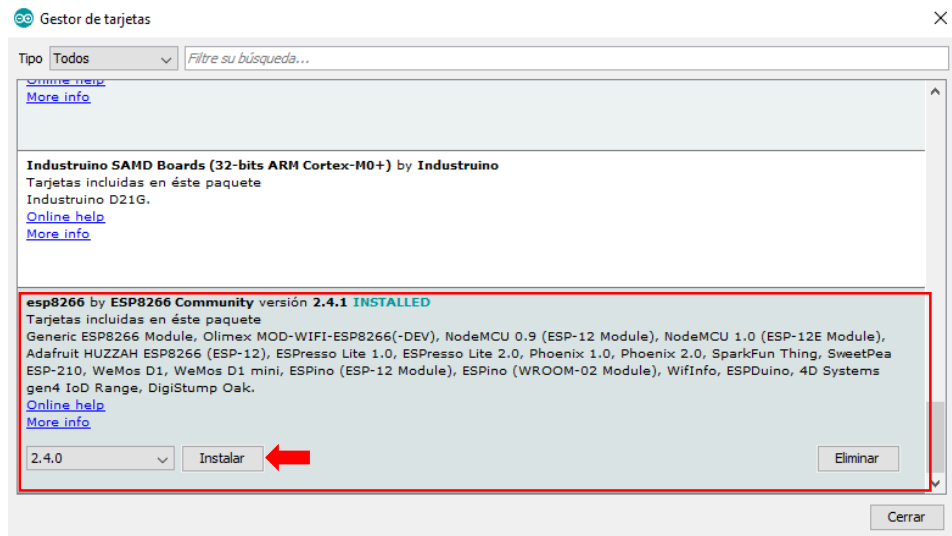
en el cuadro de texto identificado como “Gestor de URLs Adicionales de Tarjetas”, de la siguiente forma y clicar “OK”:



- c) Abrir el menú Herramientas → Placa: → Gestor de Tarjetas



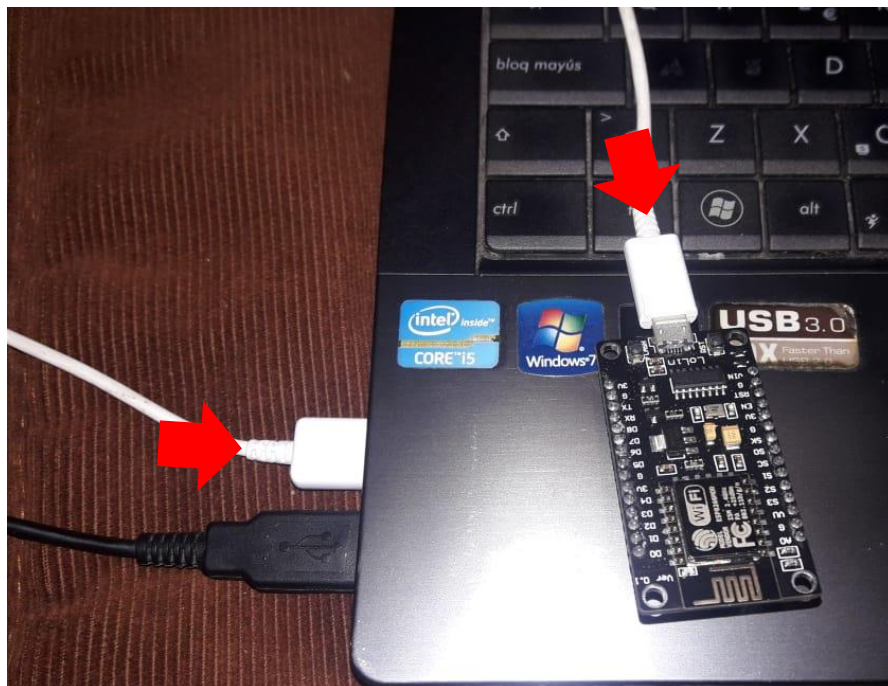
d) En la ventana del Gestor de Tarjetas, deslizar la barra de navegación hacia abajo hasta encontrar el cuadro nombrado como “esp8266”, seleccionar la última versión en el menú desplegable y presionar “Instalar”:



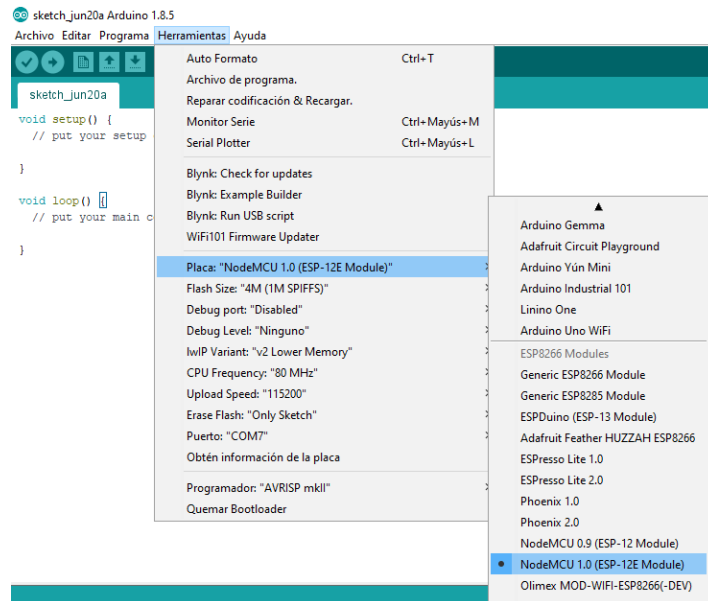
Esperar a que termine la instalación y luego presionar “Cerrar”:

## Paso 2 - Configuraciones Previas

a) Conectar el NodeMCU en algún puerto USB:



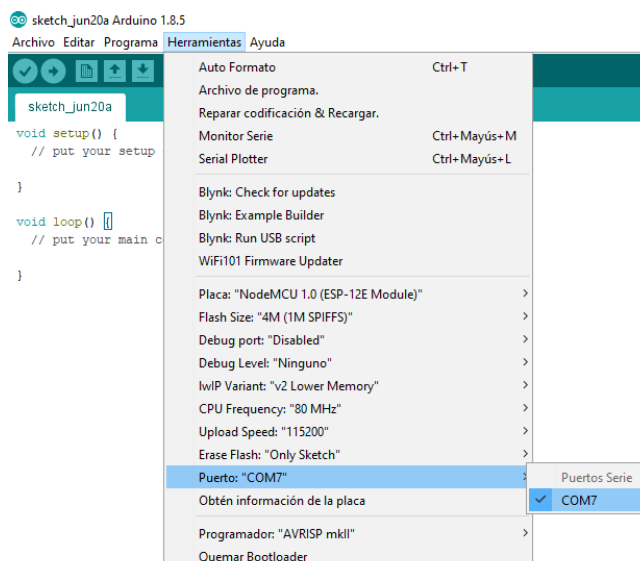
- b) En el menú Herramientas → Placa, buscar y seleccionar “NodeMCU (1.0 ESP 12-E Module)”



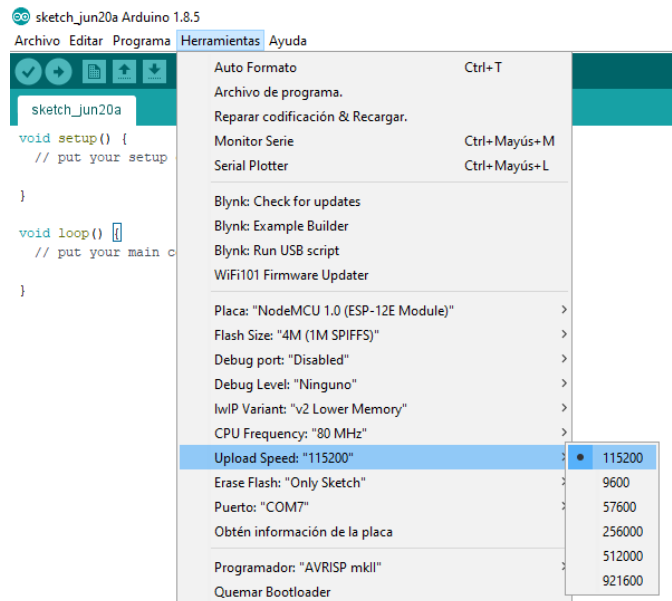
NOTA: Se selecciona NodeMCU 1.0 porque estamos trabajando con un NodeMCU versión 3, para las versiones 1 y 2 seleccionar NodeMCU 0.9



- c) En el menú Herramientas → Puerto, buscar y seleccionar el puerto adecuado.



- d) En el menú Herramientas → Upload Speed, buscar y seleccionar la opción “115200”.



### Paso 3 – El código

- a) En el área de trabajo de la IDE de Arduino, copiar y pegar lo siguiente:

```
#include <ESP8266WiFi.h>

const char* ssid = "SSID";
const char* password = "password";

// Instancio el Server seteado para que escuche por el puerto 301

IPAddress ip(192, 168, 0, 107); //set static ip
IPAddress gateway(192, 168, 0, 1); //set getteway
IPAddress subnet(255, 255, 255, 0); //set subnet
WiFiServer server(301); //Port 301 to access server

void setup() {
  Serial.begin(115200);
  delay(10);

  // prepare GPIO2
  pinMode(5, OUTPUT); //Luz=D3=GP 0
  digitalWrite(5, 1); //Luz apagada (relé normalmente abierto)
  pinMode(4, OUTPUT); //Enchufe=D4=GP 2
  digitalWrite(4, 1); //Enchufe encendido (relé normalmente cerrado)

  // Connect to WiFi network
  Serial.println();
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);

  WiFi.config(ip, gateway, subnet);
  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected");

  // Start the server
  server.begin();
  Serial.println("Server started");
```

```

// Print the IP address
Serial.println(WiFi.localIP());
}

void loop() {
// Se conectó alguien?
WiFiClient client = server.available();
if (!client) {
return;
}

// Hay alguien conectado!
Serial.println("new client");
while(!client.available()){
delay(1);
}

// Leo el pedido
String req = client.readStringUntil('\r');
Serial.println(req);
client.flush();

// Evalúo el pedido
int luz;
int enchufe;
if (req.indexOf("/luz/0") != -1){ //pedí apagar luz
luz = 0;
digitalWrite(5, 1);}
else if (req.indexOf("/luz/1") != -1){ //pedí prender luz
luz = 1;
digitalWrite(5, 0);}
else if (req.indexOf("/enchufe/0") != -1){ //pedí apagar enchufe
enchufe = 0;
digitalWrite(4, 0);}
else if (req.indexOf("/enchufe/1") != -1){ //pedí prender enchufe
enchufe = 1;
digitalWrite(4, 1);}
else {
Serial.println("invalid request");
//client.stop();
//return;
}

client.flush();

// Para la pagina web
String s = "HTTP/1.1 200 OK\r\nContent-Type: text/html\r\n\r\n<!DOCTYPE
HTML>\r\n<html>\r\nLuz is now "; //<html> me marca que se abre código en html
s += (luz)?"ON":"OFF";
s += "\r\nEnchufe is now ";
s += (enchufe)?"ON":"OFF";
s += "</html>\n"; //se cierra el código en html

// Send the response to the client
client.print(s);
delay(1);
Serial.println("Client disonnected");

// Cuando termino el loop el cliente se desconecta
}

```

- b) Modificar la tercera y la cuarta línea del código anterior colocando en lugar de SSID el nombre de la red WiFi a la cual se conectará el NodeMCU y en lugar de password la contraseña de la misma. Por ejemplo, si deseo conectarme a la red llamada “Fibertel Wifi637 2.4GHz” cuya contraseña es “Estrada111” esas líneas deberían quedar como:

```

const char* ssid = "Fibertel Wifi637 2.4GHz";
const char* password = "Estrada111";

```

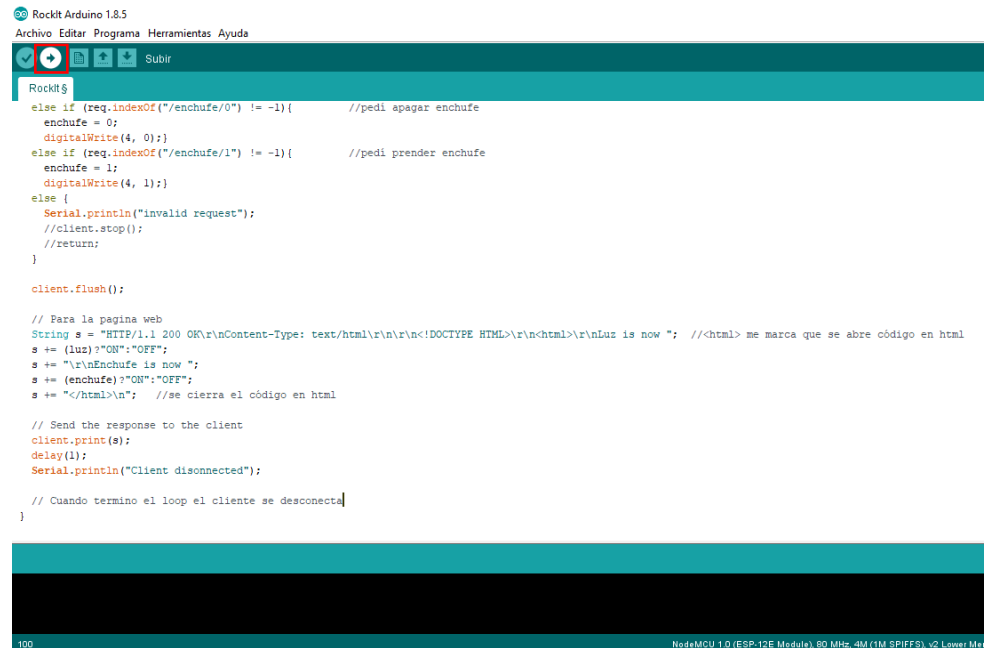
- c) Prestar especial atención a la línea:

```
IPAddress ip(192, 168, 0, 107);
```



que nos indica el número de IP que poseerá el NodeMCU y con el cual podemos referirnos al servidor local que se crea.

- d) Presionar “Subir” y esperar a que termine el proceso:



```
Rockit Arduino 1.8.5
Archivo Editar Programa Herramientas Ayuda

Rockit$

else if (req.indexOf("/enchufe/0") != -1){ //pedi apagar enchufe
  enchufe = 0;
  digitalWrite(4, 0);}
else if (req.indexOf("/enchufe/1") != -1){ //pedi prender enchufe
  enchufe = 1;
  digitalWrite(4, 1);}
else {
  Serial.println("invalid request");
  //client.stop();
  //return;
}

client.flush();

// Para la pagina web
String s = "HTTP/1.1 200 OK\r\nContent-Type: text/html\r\n\r\n<!DOCTYPE HTML>\r\n\r\n<html>\r\n\r\nLuz is now "; //<html> me marca que se abre código en html
s += (luz)? "ON": "OFF";
s += "\r\n\r\nLuz is now ";
s += (enchufe)? "ON": "OFF";
s += "</html>\r\n"; //se cierra el código en html

// Send the response to the client
client.print(s);
delay(1);
Serial.println("Client disconnected");

// Cuando termino el loop el cliente se desconecta
}
```

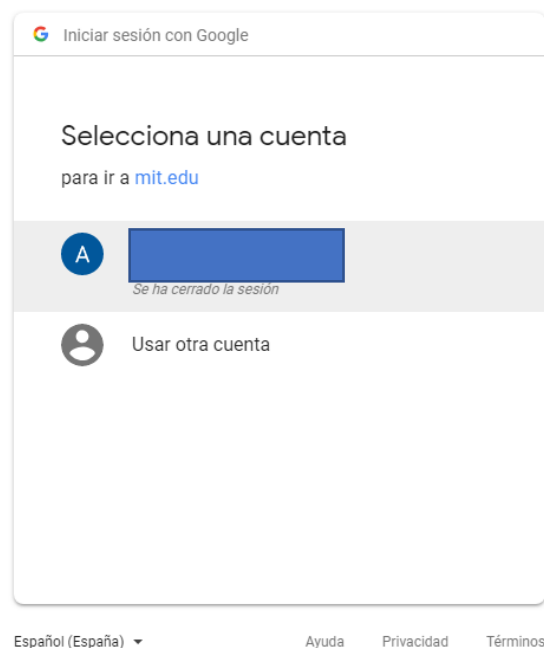
## 2º Etapa – Creación de la App

### Paso 1 – Inicialización del Proyecto

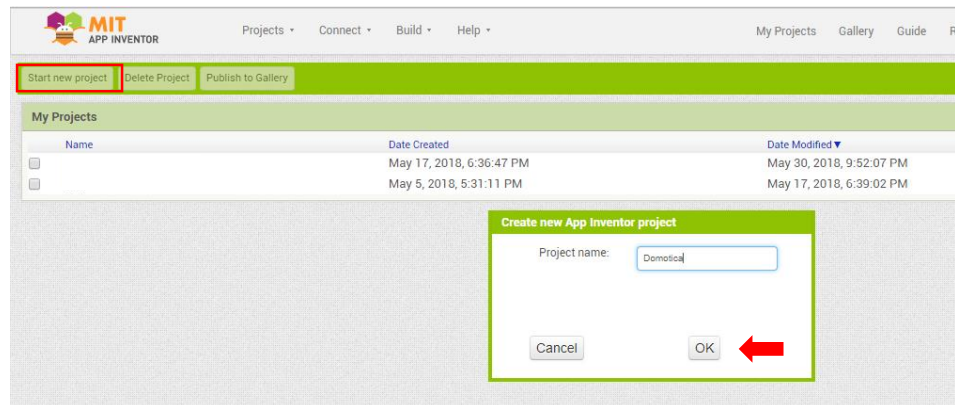
- a) En el navegador de internet, abrir el URL:

<http://ai2.appinventor.mit.edu/>

e iniciar sesión con una cuenta de Google:

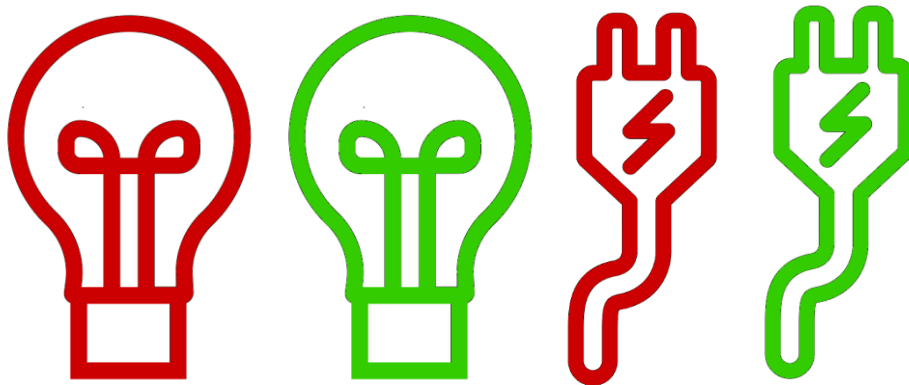


- b) Clickear “Start New Project”, completar el nombre del proyecto, luego presionar “OK”:

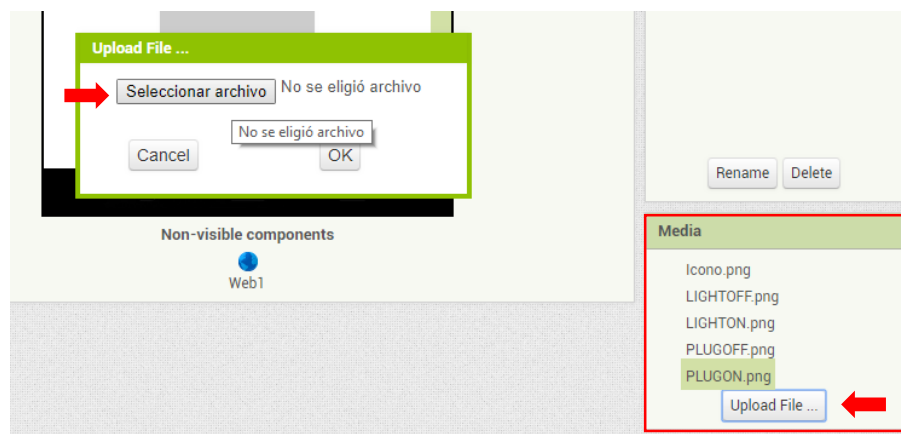


## Paso 2 – Interfaz Gráfica de la App

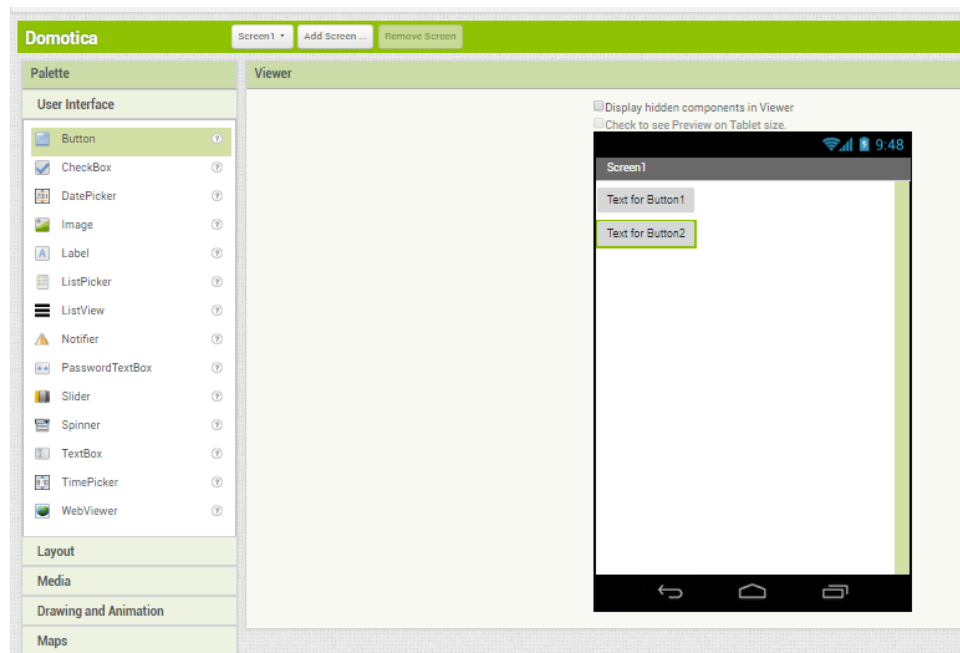
- a) Buscar cuatro imágenes que representarán el apagar una luz, prender una luz, apagar un enchufe y prender un enchufe tales como la siguientes:



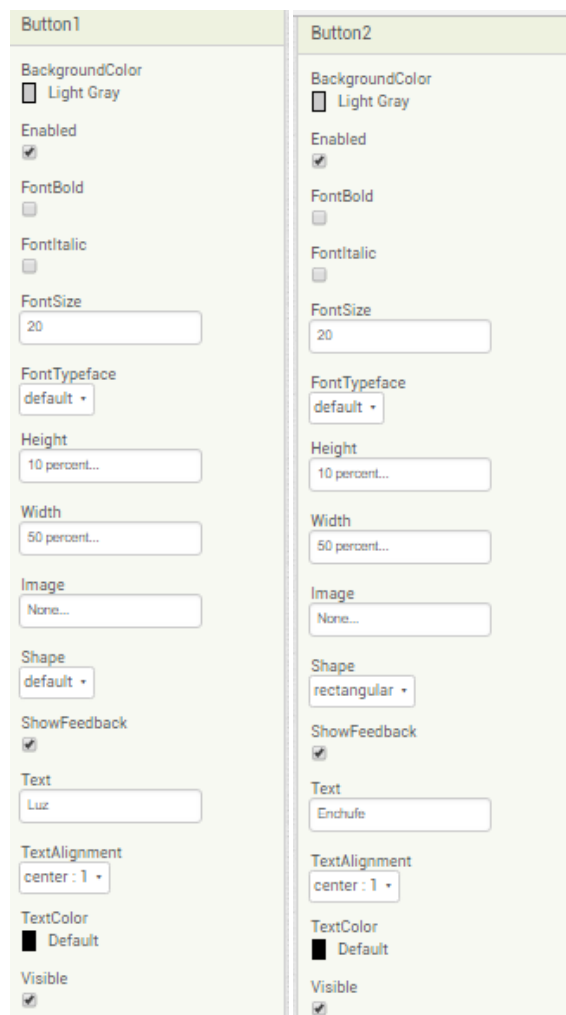
Guardarlas en la computadora y cargarlas a la app desde el apartado “Media”:



- b) Arrastrar 2 botones desde el menú de la izquierda hasta el centro de la página:



Configurando cada uno en la sección “Properties” como sigue:



- c) Colocar dos botones más (Botones 3 y 4) debajo de los botones 1 y 2, configurarlos de la misma forma, pero otorgándoles a ambos un alto y ancho del 40 %.
- d) Arrastrar otros 4 botones (Botones 5 al 8) y configurarlos de la siguiente forma:

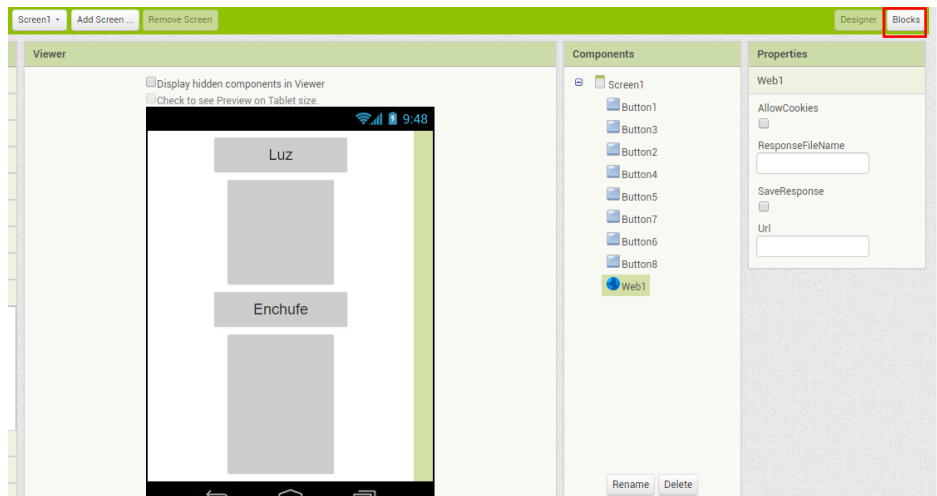
Button5	Button6	Button7	Button8
BackgroundColor ■ Default	BackgroundColor ■ Default	BackgroundColor ■ Default	BackgroundColor ■ Default
Enabled <input checked="" type="checkbox"/>	Enabled <input checked="" type="checkbox"/>	Enabled <input checked="" type="checkbox"/>	Enabled <input checked="" type="checkbox"/>
FontBold <input type="checkbox"/>	FontBold <input type="checkbox"/>	FontBold <input type="checkbox"/>	FontBold <input type="checkbox"/>
FontItalic <input type="checkbox"/>	FontItalic <input type="checkbox"/>	FontItalic <input type="checkbox"/>	FontItalic <input type="checkbox"/>
FontSize 14.0	FontSize 14.0	FontSize 14.0	FontSize 14.0
FontTypeface default ▾	FontTypeface default ▾	FontTypeface default ▾	FontTypeface default ▾
Height Automatic...	Height Automatic...	Height Automatic...	Height Automatic...
Width Automatic...	Width Automatic...	Width Automatic...	Width Automatic...
Image LIGHTOFF.png...	Image PLUGOFF.png...	Image LIGHTON.png...	Image PLUGON.png...
Shape default ▾	Shape default ▾	Shape default ▾	Shape default ▾
ShowFeedback <input checked="" type="checkbox"/>	ShowFeedback <input checked="" type="checkbox"/>	ShowFeedback <input checked="" type="checkbox"/>	ShowFeedback <input checked="" type="checkbox"/>
Text Text for Button5	Text Text for Button6	Text Text for Button7	Text Text for Button8
TextAlignment center : 1 ▾	TextAlignment center : 1 ▾	TextAlignment center : 1 ▾	TextAlignment center : 1 ▾
TextColor ■ Default	TextColor ■ Default	TextColor ■ Default	TextColor ■ Default
Visible <input type="checkbox"/>	Visible <input type="checkbox"/>	Visible <input type="checkbox"/>	Visible <input type="checkbox"/>

- e) Arrastrar un objeto de tipo Web, desde la librería de conectividad:



### Paso 3 – Programación en Bloques

a) Presionar “Blocks”:



b) Arrastrar desde el menú de la derecha los bloques necesarios y completarlos para que queden de la siguiente forma:

```
when Screen1.Initialize
do
  set Button1.BackgroundColor to red
  set Button2.BackgroundColor to green
  set Button3.Image to Button5.Image
  set Button4.Image to Button8.Image
```

```
initialize global flagluz to false
initialize global flagenchufe to true
```

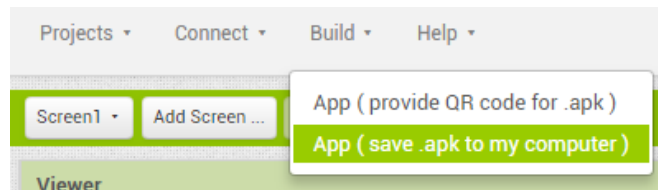
```
when Button1.Click
do
  if not get global flagluz
  then
    set Web1.Url to http://192.168.0.107/luz/1
    call Web1.Get
    set Button1.BackgroundColor to green
    set Button3.Image to Button7.Image
    set global flagluz to true
  else
    set Web1.Url to http://192.168.0.107/luz/0
    call Web1.Get
    set Button1.BackgroundColor to red
    set Button3.Image to Button5.Image
    set global flagluz to false
```

```
when Button3.Click
do
  if not get global flagluz
  then
    set Web1.Url to http://192.168.0.107/luz/1
    call Web1.Get
    set Button1.BackgroundColor to green
    set Button3.Image to Button7.Image
    set global flagluz to true
  else
    set Web1.Url to http://192.168.0.107/luz/0
    call Web1.Get
    set Button1.BackgroundColor to red
    set Button3.Image to Button5.Image
    set global flagluz to false
```

```
when Button2.Click
do
  if not get global flagenchufe
  then
    set Web1.Url to http://192.168.0.107/enchufe/1
    call Web1.Get
    set Button2.BackgroundColor to green
    set Button4.Image to Button8.Image
    set global flagenchufe to true
  else
    set Web1.Url to http://192.168.0.107/enchufe/0
    call Web1.Get
    set Button2.BackgroundColor to red
    set Button4.Image to Button6.Image
    set global flagenchufe to false
```

```
when Button4.Click
do
  if not get global flagenchufe
  then
    set Web1.Url to http://192.168.0.107/enchufe/1
    call Web1.Get
    set Button2.BackgroundColor to green
    set Button4.Image to Button8.Image
    set global flagenchufe to true
  else
    set Web1.Url to http://192.168.0.107/enchufe/0
    call Web1.Get
    set Button2.BackgroundColor to red
    set Button4.Image to Button6.Image
    set global flagenchufe to false
```

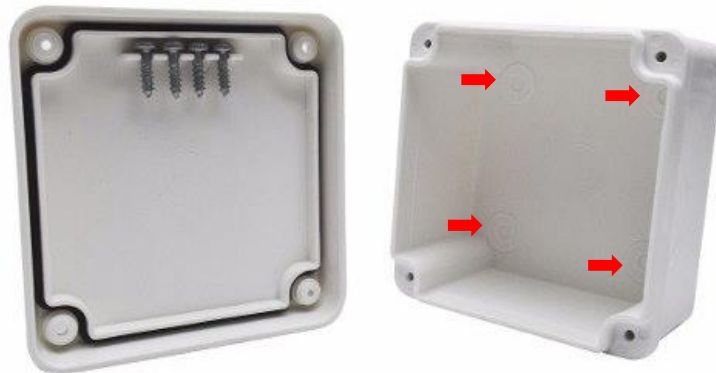
- c) En el menú Build, seleccionar “Apk (save .apk to my computer)”. Esta opción descargará un archivo de tipo .apk el cual deberá ser ejecutado en un celular con Sistema Operativo Android para instalar la app.



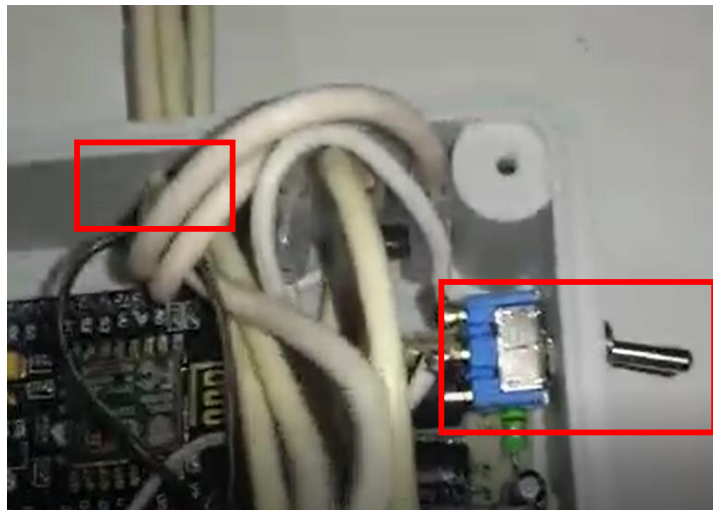
### 3º Etapa – Ensamble del Hardware

#### Paso 1 – Preparación de la caja estanca

- a) Realizar cuatro orificios en el fondo de la caja estanca. Cada uno en una esquina diferente de la caja.

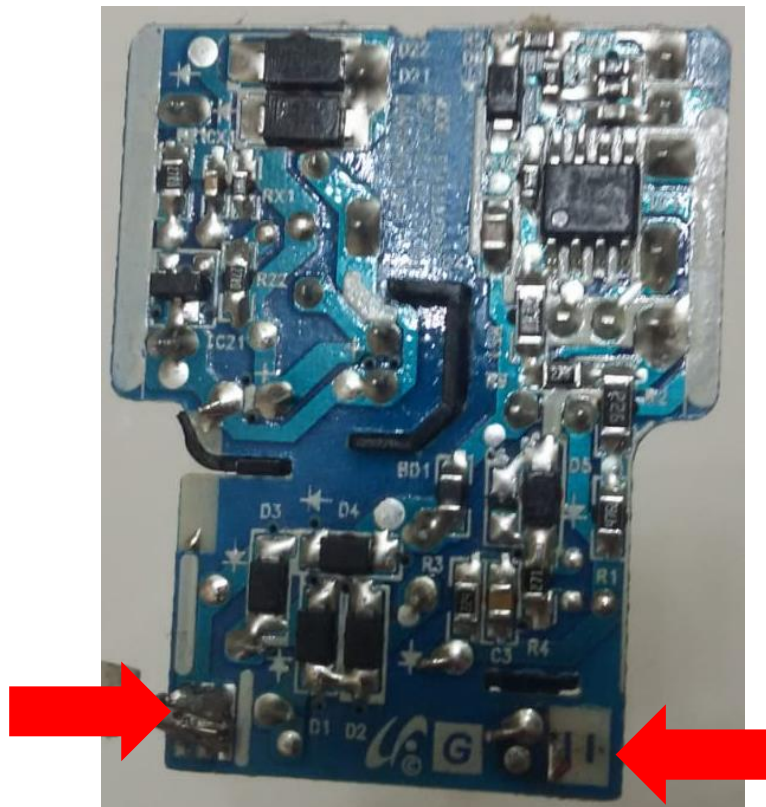


- b) Realizar dos orificios en las paredes laterales de la caja estanca; el primero para el switch principal del sistema y el segundo para los cables que irán al tomacorriente y switch (fase, neutro y tierra). Para realizar este proceso se recomienda posicionar previamente todos los elementos que irán dentro de la caja (módulo de dos relés, NodeMCU, placa del transformador 220V AC a 5V CD y todos los cables que se mencionen más abajo en esta etapa) y posicionar y medir el switch para hacer el orificio con las medidas y ubicación adecuadas.



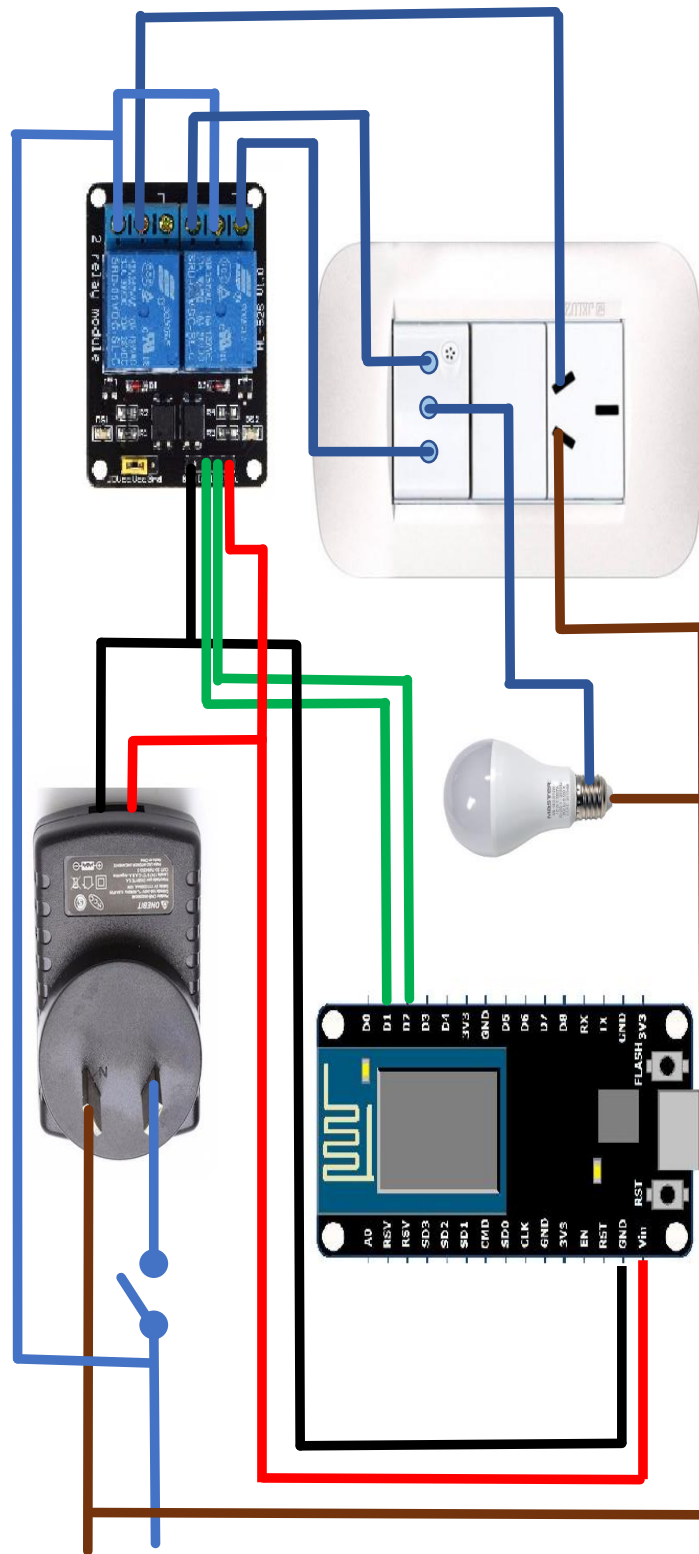
## Paso 2 – Preparación del contenido

- a) Desarmar un cargador de celular, deshacerse de la carcasa plástica y quedarse únicamente con la placa. Es imprescindible identificar los contactores de fase y neutro que normalmente se enchufan a la red domiciliaria dentro del circuito.





- b) Conectar todos los elementos de la caja (Relés, switch, transformador de celular y NodeMCU) y el conjunto de la tecla y el tomacorriente entre sí empleando los cables y siguiendo el siguiente esquema:



Es importante tener en cuenta que los cables representados con azul y marrón son extensiones del neutro y la fase de la red domiciliaria

- c) Colocar todos los otros elementos en la caja estanca de manera ordenada y prolija.



#### 4º Etapa – Instalación en la red domiciliaria

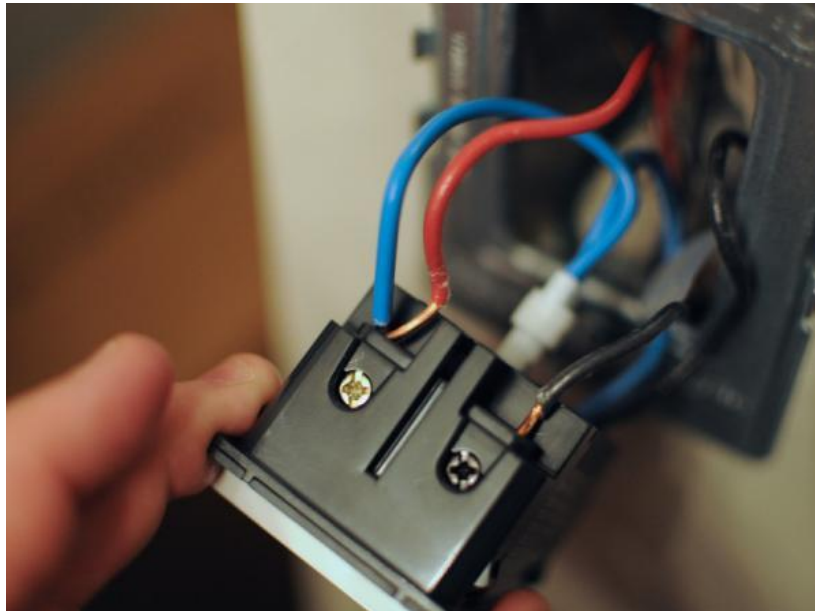
##### Paso 1 – Consideraciones Previas

- a) Bajar la llave termo magnética del lugar, cortando el suministro eléctrico a la zona del domicilio en la cual instalaremos el sistema de domótica.



- b) Buscar en el domicilio una tecla o tomacorriente que contengan detrás tanto fase como neutro y tierra. Los códigos de colores en Argentina para identificar estos cables son:

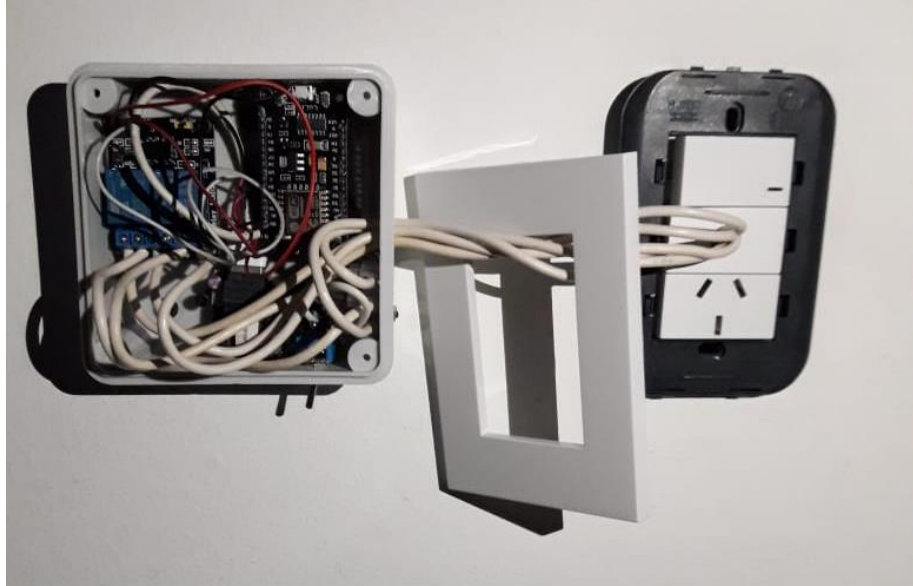
- Verde/Amarillo: Tierra
- Azul: Neutro
- Marrón, negro o gris: Fase



##### Paso 2 – Conexión a la instalación domiciliaria

- a) Desconectar el tomacorriente o tecla existente de la red domiciliaria.

- b) Conectar a la red domiciliaria, el conjunto tecla-tomacorriente armado en la etapa número 3. Atornillarlo a sus soportes en la pared.
- c) Con el taladro, hacer cuatro orificios en la pared y colocar en ellos los tacos Fisher.
- d) Atornillar en los tacos Fisher la caja estanca.



- e) Cerrar la caja estanca con sus tornillos y volver levantar la llave termo magnética de la red domiciliaria.

## Precauciones y otros requerimientos

- Verificar el funcionamiento del celular en cuestión y las aplicaciones requeridas.
- Ser prolijo y ordenado.
- Contar con conexión inalámbrica a internet con buen alcance en la zona de operación del sistema.
- Poseer conocimientos suficientes de electricidad cómo para realizar las instalaciones y no sufrir un choque eléctrico ni quemar otros artefactos.
- Cortar la luz antes de realizar la instalación.
- Tener experiencia con el uso de un taladro.
- Ser precavido con el uso de la batería del celular o Tablet que se emplee para el uso del sistema.
- No abrir la caja estanca mientras haya suministro a la red eléctrica domiciliaria.

## Autores y datos de contacto

- Nombre y Apellido:
  1. Arturo Mollinedo Garay
  2. Carolina Herrera
  3. Joaquín Seia
  4. Tamara Klincovitzky
- Correo electrónico:
  1. arturo\_mollinedo@hotmail.com
  2. herrera-carolina@outlook.es
  3. joacoseia18@hotmail.com
  4. tami\_klin@hotmail.com