

Experiencia educativa:

Internet de las cosas

Catedrático:

M. I. Sergio Francisco. Hernández Machuca

Alumnos:

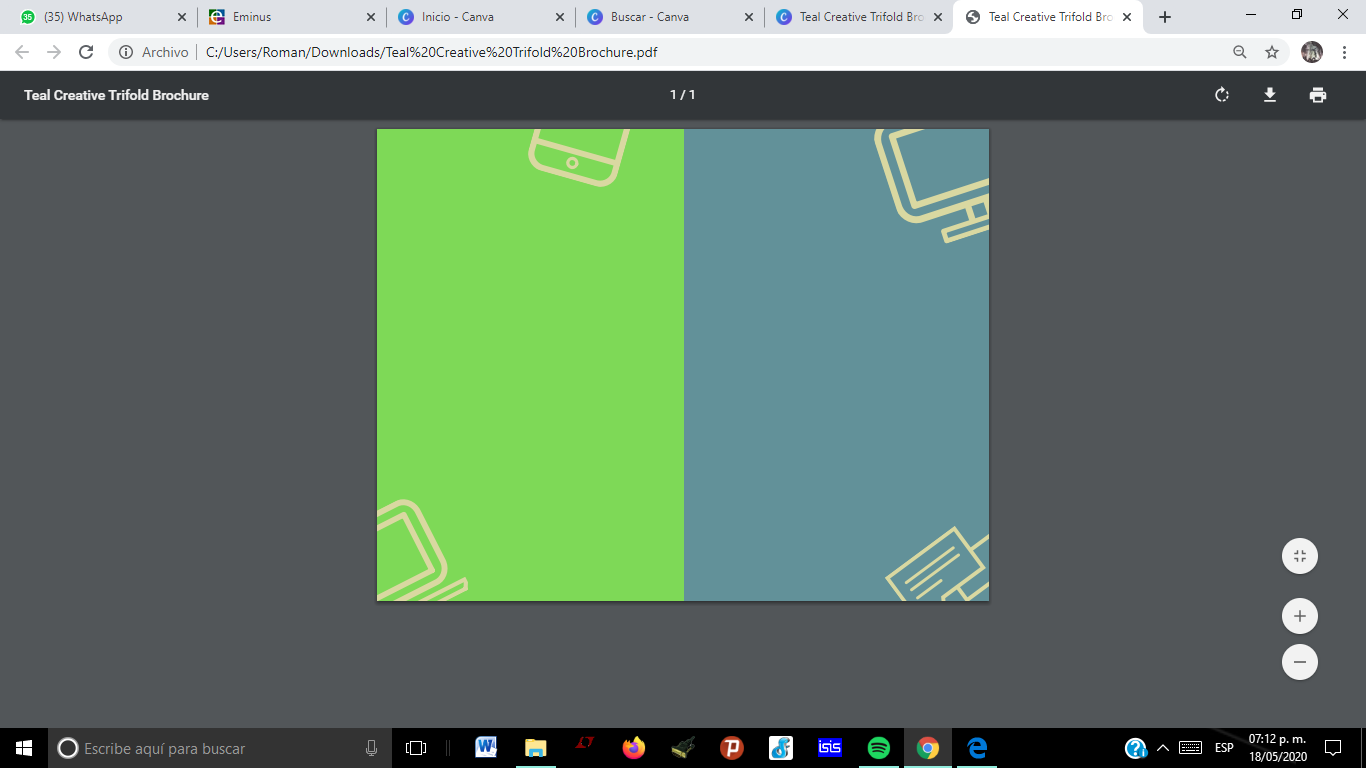
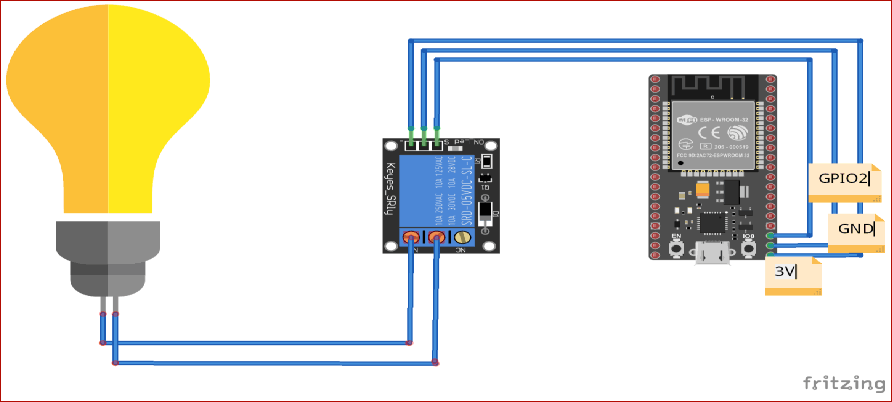
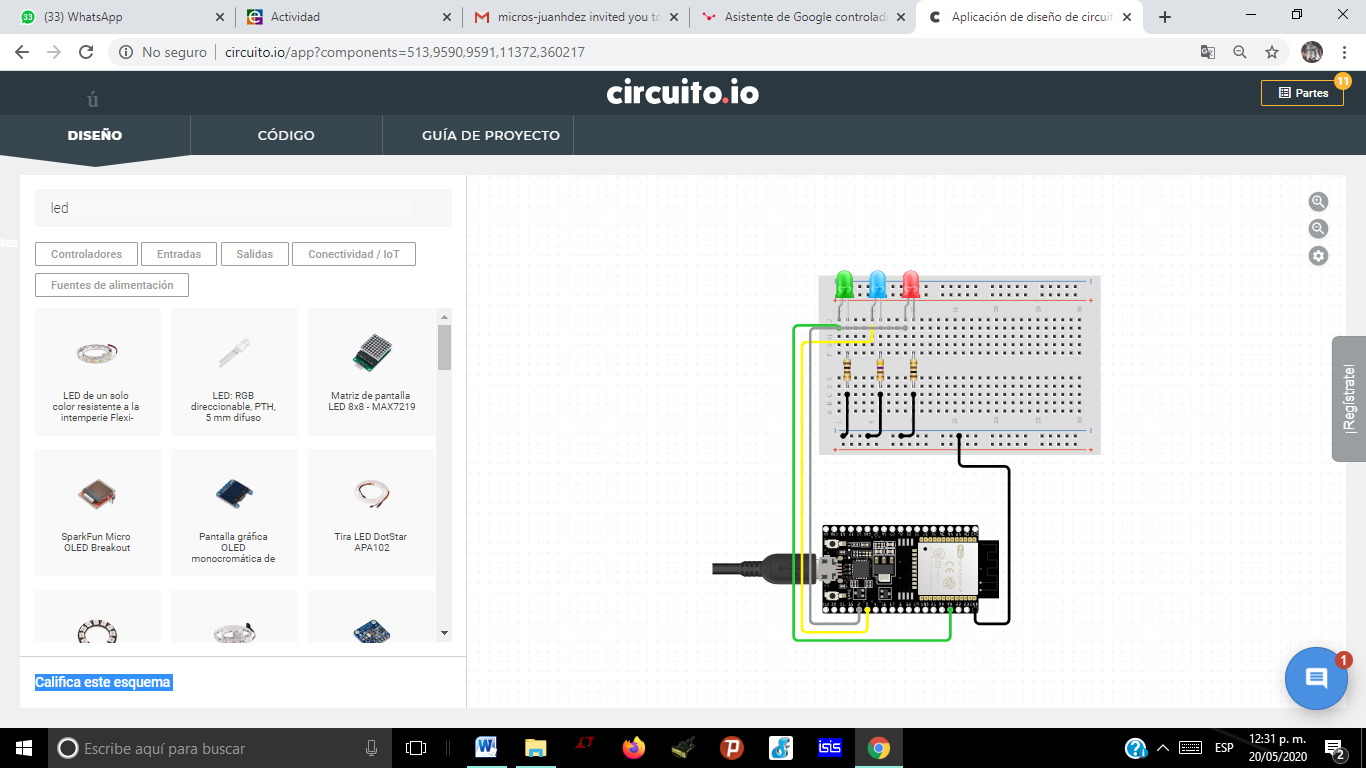
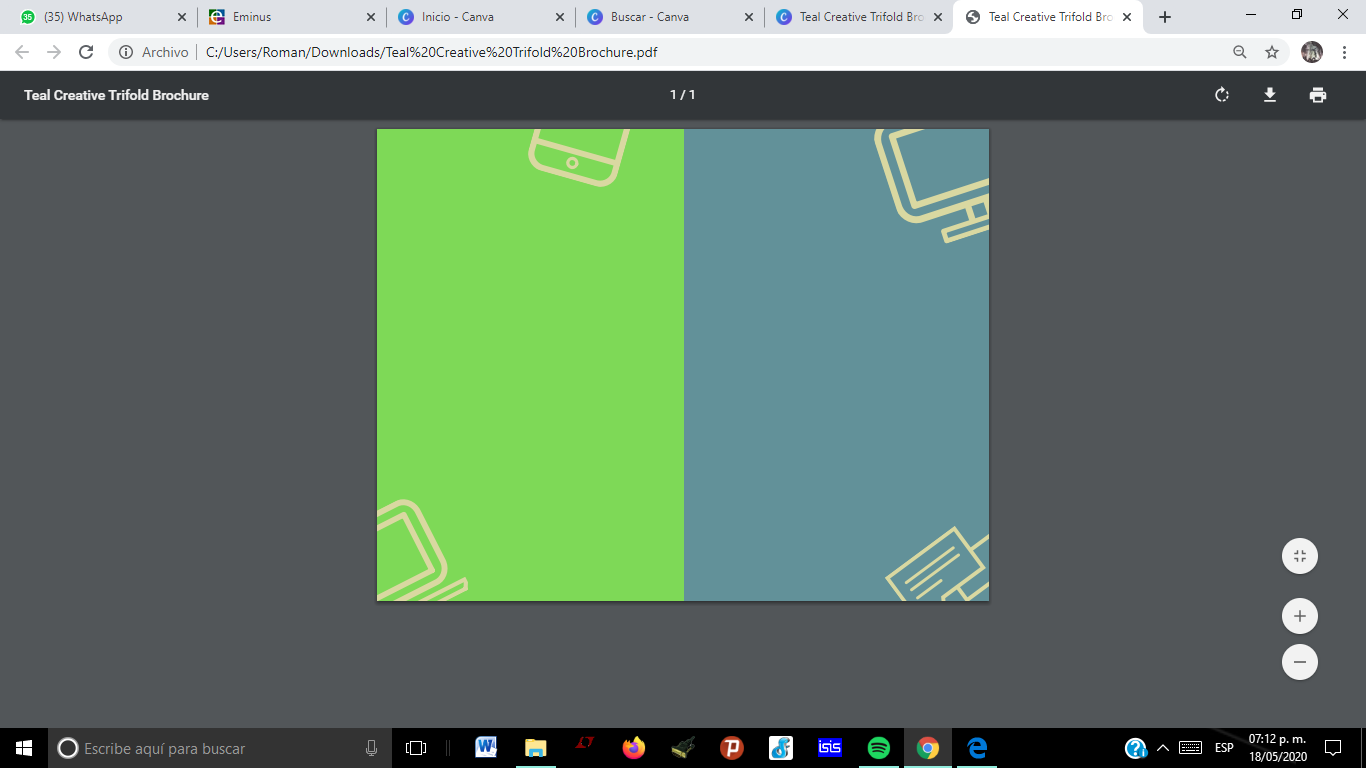
Hernández Reyes Roberto Saúl S17012854

Sánchez López Luis Uriel S17012865

Velásquez Reyes Román Gabriel S17012869

Proyecto sistema de control y seguridad

MANUAL USUARIO



Definición del problema

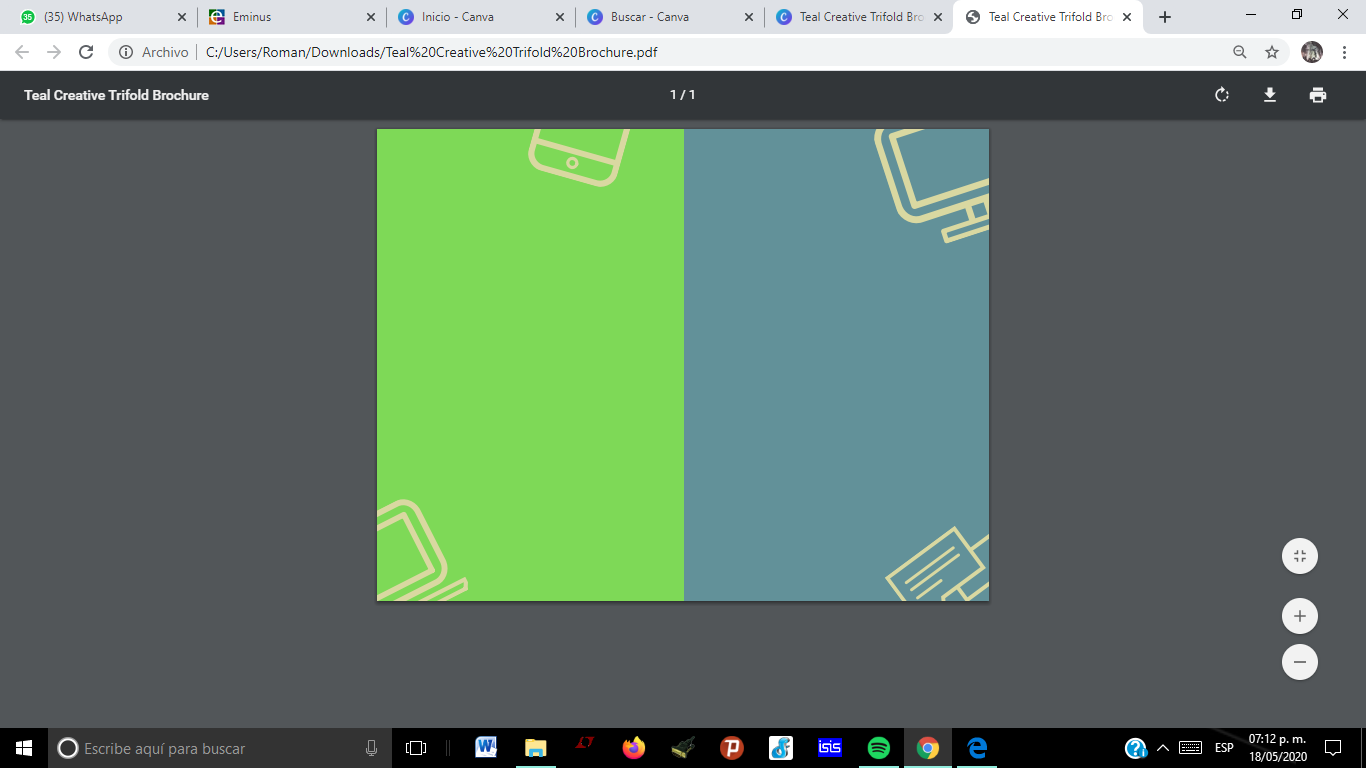
INTRODUCCIÓN

Ideamos un sistema de comunicación por medio de internet y enlaces de Adafruit e IFFFT, para optimizar, haciendo accesible y un acercamiento a este tipo de tecnologías a la vida de las personas, de tal forma de tener el control de su casa, habitación en su dispositivo móvil. El sistema consta de un microcontrolador el ESP32 y una secuencia de leds como hardware, la parte importante se desarrolla en el software, utilizamos el Arduino IDE para comunicarnos con nuestra placa. Puede tener control por medio de una laptop, ingresando con su usuario de Adafruit donde se encuentran los interruptores los cuales se encargan de poner ON/OFF a las salidas, otro punto destacable, es que estos interruptores con ayuda de IFFFT, se pueden controlar por medio de la aplicación de Google Asistente, así por medio de comandos de voz previamente programados, responderá y realizará nuestra petición con éxito

En el contexto que nos encontramos, se ha tornado complicado en cuanto a seguridad, se han realizado saqueos, de negocios locales, hogares, existen sistemas de alarma, los cuales sus precios son muy elevados para los comerciantes, con nuestra propuesta el componente de mayor valor es la placa y económico, el cual nos podemos comunicar desde la comodidad de nuestro hogar, podemos accionar algún foco que alumbre el local solitario, o active algún actuador que aumente la seguridad dentro del local.

Diseño de la solución

Para el diseño de nuestro circuito, realizamos un prototipo, anterior mente, utilizamos códigos, para asegurarnos que nuestra placa ESP32, se encontrara funcionando con sus controladores adecuados y librerías, cabe mencionar que se utilizó el software de Arduino IDE, dentro se instaló el controlador para que pudiera comunicarse con el software y para facilitar el uso de las librerías de Adafruit y sus respectivos controladores.



3. Declaramos nuestros pines de salida

int output=2;

int output2=4;

int output3=16;

4. Creamos objetos WiFiClient y Adafruit\_MQTT\_Client como variables globales, y creeamos una instancia del feed para su LED\_Control.

Cliente de WiFiClient;

Adafruit\_MQTT\_Client mqtt (& client, AIO\_SERVER, AIO\_SERVERPORT, AIO\_USERNAME, AIO\_KEY);

Adafruit\_MQTT\_Subscribe LED\_Control = Adafruit\_MQTT\_Subscribe (& mqtt, AIO\_USERNAME "/ feeds / LED\_Control");

1. Al comienzo del programa, debemos agregar las siguientes bibliotecas para conectarnos a las librerías de Adafruit

#include <WiFi.h>

#include "Adafruit\_MQTT.h"

#include "Adafruit\_MQTT\_Client.h"

2. Definimos la SSID de red, contraseña, su nombre de usuario Adafruit y su clave AIO que obtuvo al configurar Adafruit IO.

#define WLAN\_SSID "INFINITUM5D9E\_2.4"

#define WLAN\_PASS "Bacon10deMayo"

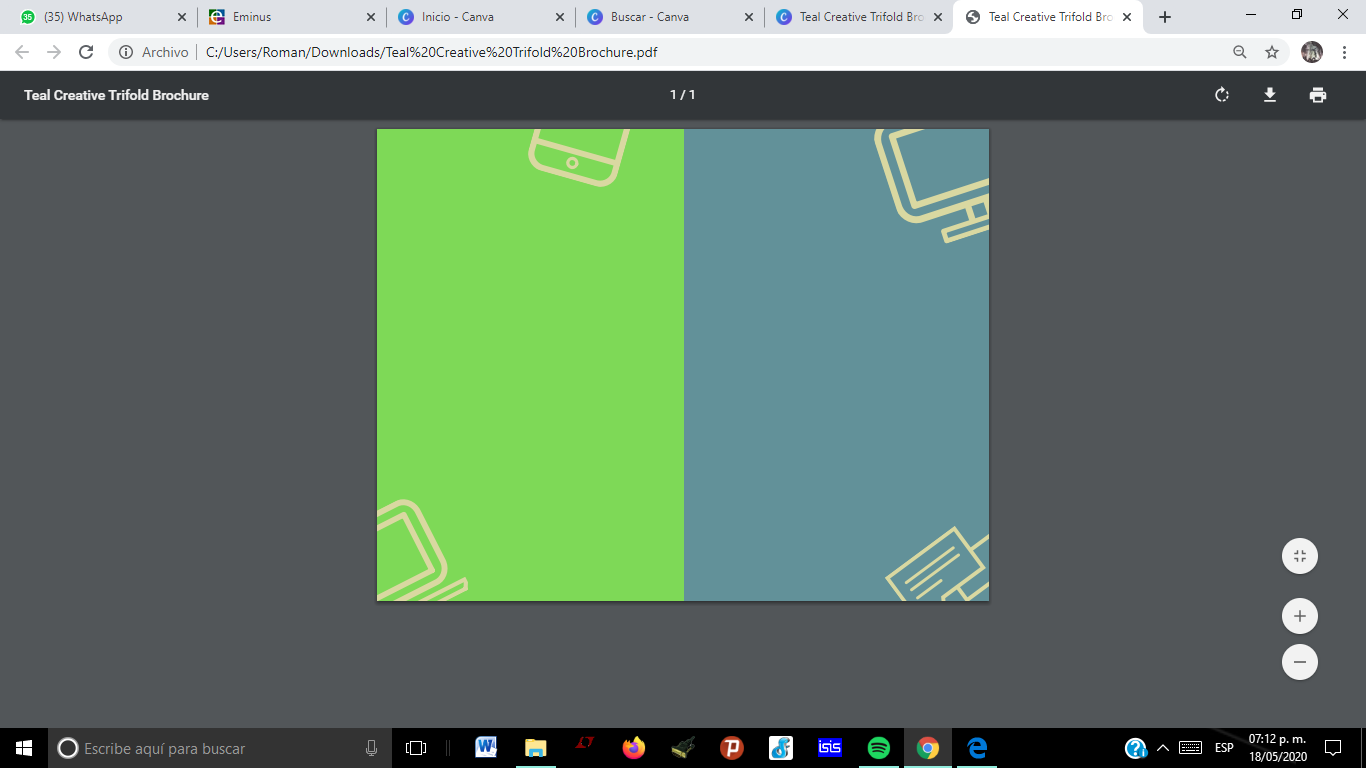
#define AIO\_SERVER "io.adafruit.com"

#define AIO\_SERVERPORT 1883

#define AIO\_USERNAME "romanvelrey"

#define AIO\_KEY "aio\_PZGZ44vLXiCIcMORzCMCh7UJlZW7"

Código



5. En Set Up definimos las funciones de nuestro pin LED, velocidad de transmisión y nos conectaremos a WiFi y al servidor MQTT.

void setup() {

Serial.begin(115200);

delay(10);

pinMode(2,OUTPUT);

Serial.println(); Serial.println();

Serial.print("Connecting to ");

Serial.println(WLAN\_SSID);

WiFi.begin(WLAN\_SSID, WLAN\_PASS);

while (WiFi.status() != WL\_CONNECTED) {

delay(500);

Serial.print(".");

}

Serial.println();

Serial.println("WiFi connected");

Serial.println("IP address: "); Serial.println(WiFi.localIP());

mqtt.subscribe(&LED\_Control);

}

6. En la función Loop verificar si nuestra suscripción se ha actualizado y actuar en consecuencia.

Void loop () {

MQTT\_connect ();

Adafruit\_MQTT\_Subscribe \* suscripción;

while ((suscripción = mqtt.readSubscription (5000))) {

if (suscripción == & LED\_Control) {

Serial.print (F ("Got:"));

Serial.println ((char \*) LED\_Control.lastread);

if (! strcmp ((char \*) LED\_Control.lastread, "ON"))

{

digitalWrite (2, HIGH);

}

else

{

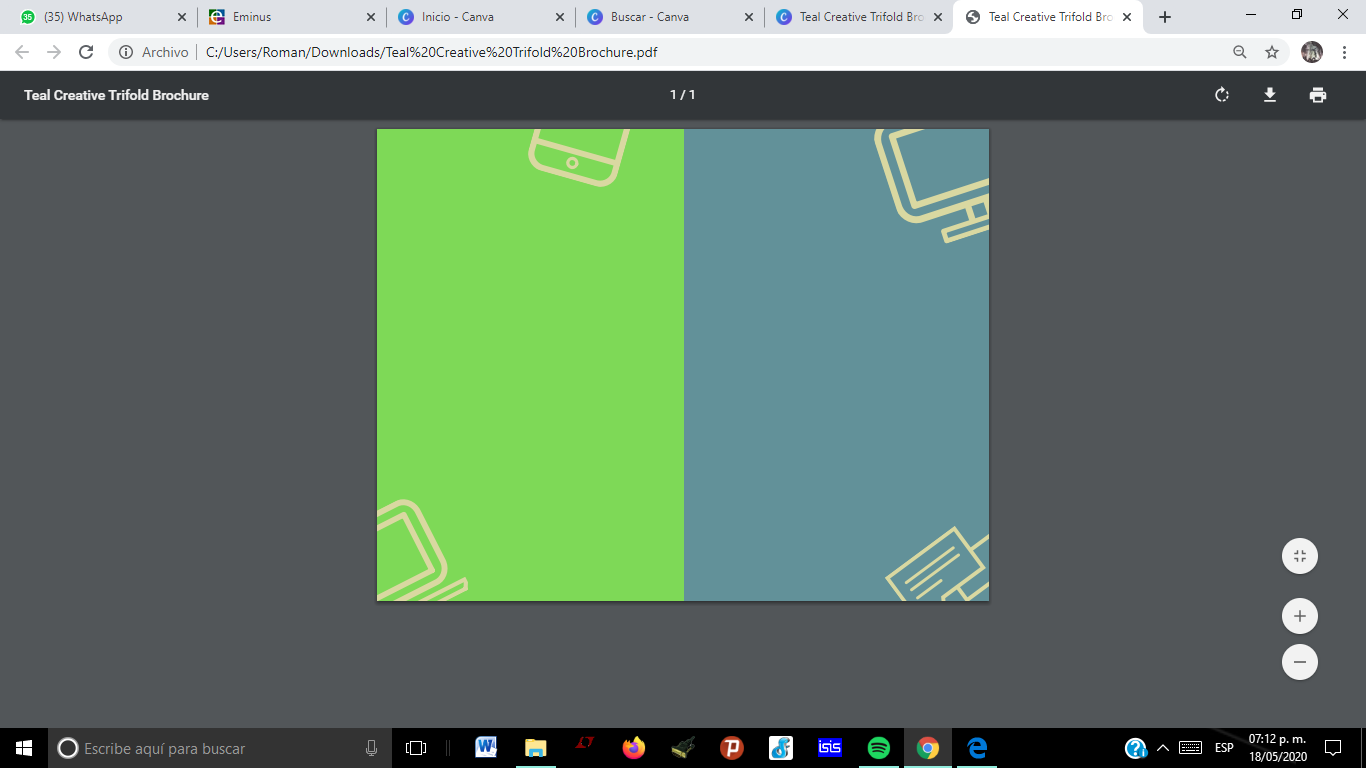
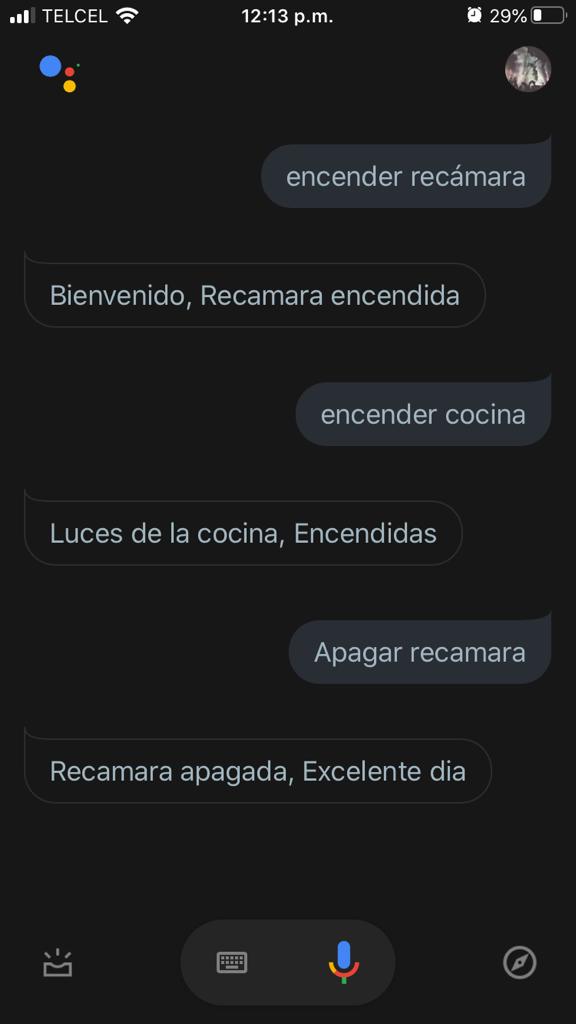
digitalWrite (2, BAJO);

}

}

}

}



Una vez que su código esté listo con modificaciones, estará listo para cargarlo en su ESP32 desde Arduino IDE.

Compilación

7. Agregamos la función MQTT Connect para que se establezca la conexión MQTT.

void MQTT\_connect () {

int8\_t ret;

if (mqtt.connected ()) {

return;

}

Serial.print ("Conectando a MQTT ...");

reintentos uint8\_t = 3; while ((ret = mqtt.connect ()) = 0) {

Serial.println (mqtt.connectErrorString (RET));

Serial.println ("Volviendo a intentar la conexión MQTT en 5 segundos ...");

mqtt.disconnect ();

retraso (5000);

reintentos--;

if (reintentos == 0) {

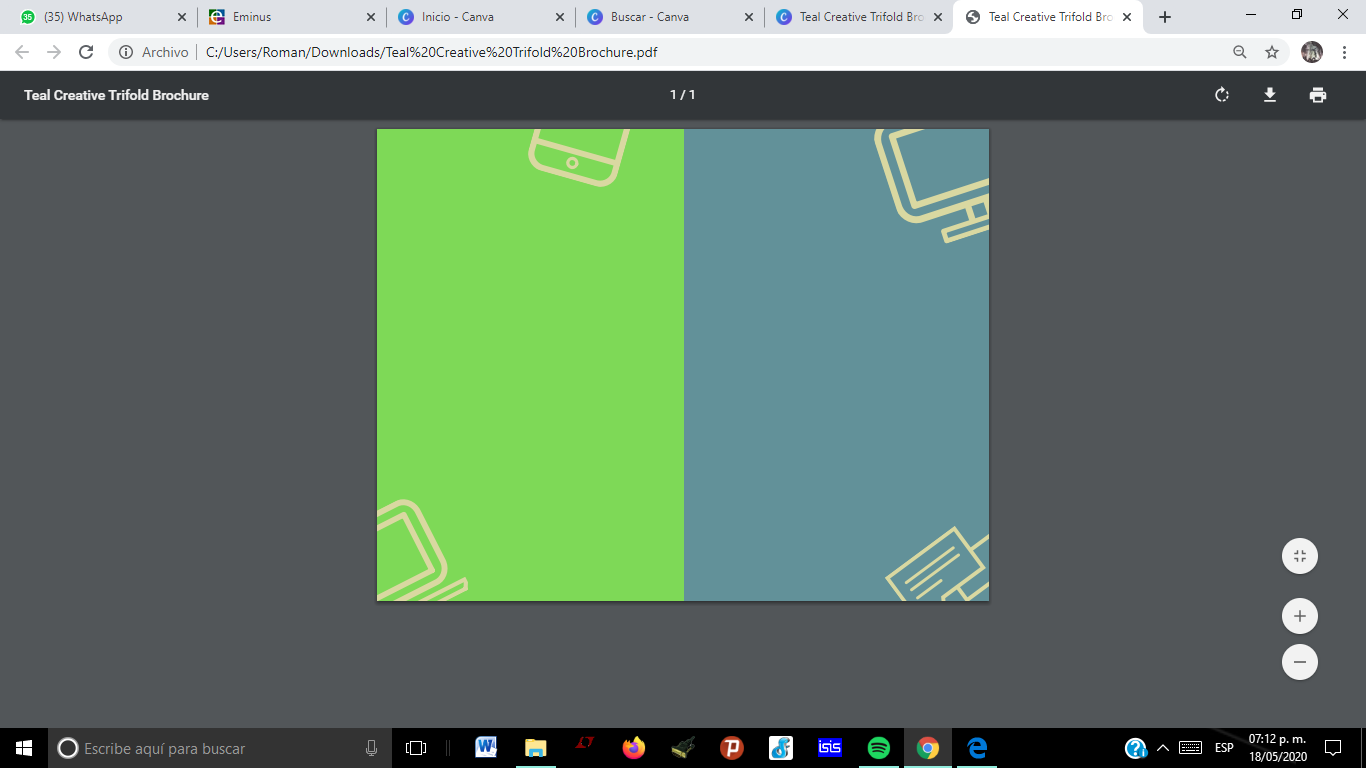
while (1);

}

}

Serial.println ("MQTT Connected!");

}



Pruebas Experimentales

Si realizamos correctamente una comunicación entre nuestra cuenta de Adafruit y el servicio de IFFFT, tendrá una excelente respuesta al solicitarle que encienda la luz de alguna recamará, como el nuestra habitación, cocina y etc.

El usuario también será capaz otro distinto comando de voz apagar las luces de los cuartos que encendió