



Evidencia de Reto IoT

Red domótica

Profesores

Dr. Enrique González Guerrero

Rodolfo Mayo Camposeco

Alejandro Paredes Balgañón - A01351746

2 de Diciembre del 2021

Identificación del problema

Se busca diseñar e implementar un prototipo de un sistema digital capaz de obtener datos mediante el uso de sensores, procesarlos y depositarlos como información en una plataforma en internet, llamada mysql, la cual consiste en crear una base de datos con las entidades y atributos de cada una, para su posterior análisis y visualización. Además se planea crear un prototipo con leds, sensores y el nodeMCU para ver de forma visual el comportamiento que se tendría.

Marco teórico

El incremento en la capacidad de procesamiento, los avances tecnológicos y los avances en la optimización de procesos, han permitido que dispositivos de procesamiento digital se integren con sensores en diferentes procesos, habilitando la transformación digital de nuestra sociedad.

Adicionalmente la conectividad de datos a muy bajo costo ha permitido la interconexión de estos dispositivos a internet, con lo cual se ha logrado obtener información en tiempo real de los sensores con que cuentan dichos dispositivos, además de permitir controlar los dispositivos por medio de esta conexión a internet. Gracias a esto, se ha podido resolver y mejorar distintas situaciones, pero en este caso nos vamos a centrar en cómo puede ayudar a reducir el consumo de luz en un hogar.

Justificación del problema

La red domótica que se planea implementar, es un aparato que nos permita controlar la luz del hogar, al igual que otros dispositivos electrónicos que tengan que ver con la electricidad, un ejemplo es la televisión. El objetivo es que se puedan controlar por medio de nuestro celular y tener acceso a prender o apagar el dispositivo seleccionado, o poner diferentes modos para que dependiendo del valor se tome una decisión, es decir que por medio de sensores se pueda detectar cuando una persona esté cerca y de esta forma tomar la decisión de prenderse o apagarse, a sí mismo poder hacer otro caso en donde dependiendo la hora, los dispositivos se prenden o se apagan.

La ubicación de la red domótica que se planea poner, estaría en la red eléctrica de la casa, en la cual podemos saber que los principales componentes serían los focos de luz junto a los controladores que servirán para controlar el encendido o el apagado de cada uno de los focos, además de otros dispositivos como es el caso de la televisión o hasta podría tomarse en cuenta el aire acondicionado.

Metodología para la solución del reto

El proyecto tendrá 2 nodos de operación:

Nodo 1: Iluminación. Es capaz de controlar las luces, tendrá como función saber si se dejó alguna luz prendida de la casa, ya que podrás prenderlas y apagarlas cuando quieras. Otra cosa con la que cuenta es con un sensor de distancia, este modo consiste en ver si alguien se encuentra en la habitación para mantener la luz prendida o apagarla si nadie se encuentra. Y el último modo es que se apaguen o se prendan por medio de un horario, esto quiere decir que dependiendo la hora del día se va a ejecutar una acción.

Tres tipos de encendido/apagado:

- Horario:

A través de una variable de red se entra en este modo de operación y se realizará una consulta del horario. Dependiendo el horario recibido, se realizará la operación indicada por horario, se enciende o se apaga la lámpara.

- Sensor de distancia:

A través de una variable de red se entra en este modo de operación y la consulta del valor se realiza a través del sensor. Dependiendo del valor obtenido, se enciende o se apaga el equipo.

- Comando:

A través de una variable de red se entra en este modo de operación. Se controlará el encendido o apagado del equipo y deberá permanecer en ese estado hasta recibir un nuevo mensaje con nueva orden.

Nodo 2: Televisión. Es capaz de controlar un equipo de entretenimiento. Este nodo podrá ser controlado desde tu dispositivo, es decir podrás prenderla y apagarla cuando quieras, y también cuenta con el modo del sensor a distancia para poder detectar si alguien se encuentra para mantener la televisión encendida, y si no hay nadie, apagarla.

Dos tipos de encendido/apagado:

- Sensor de distancia:

A través de una variable de red se entra en este modo de operación y la consulta del valor se realiza a través del sensor. Dependiendo del valor obtenido, se enciende o se apaga el equipo. Cabe destacar que los datos del sensor se guardarán en la base de datos.

- Comando:

A través de una variable de red se entra en este modo de operación. Se controlará el encendido o apagado del equipo y deberá permanecer en ese estado hasta recibir un nuevo mensaje con nueva orden.

Lógica

H=modo horario

S=modo sensor de distancia

C=modo comando

O=hora

D=distancia

E=encender

A=apagar

Modo horario

H=O

| O | Salida |
|---|--------|
| 0 | 0 |
| 1 | 1 |

Modo Sensor de Distancia

S=D

| D | Salida |
|---|--------|
| 0 | 0 |
| 1 | 1 |

Modo Comando

C=EA'

| E | A | Salida |
|---|---|--------|
| 0 | 0 | 0 |

| | | |
|---|---|---|
| 0 | 1 | 0 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

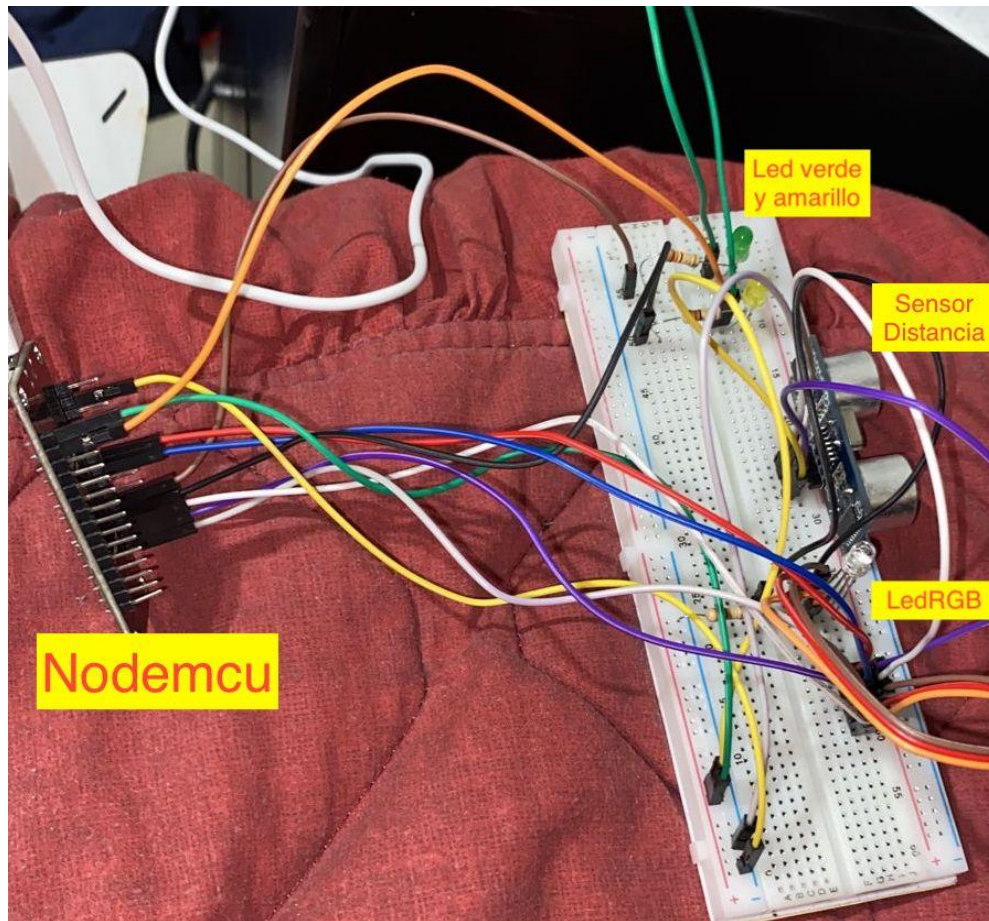
Es decir que si junto todos los comandos, mi función final quedaría como:

$$F=O+D+EA'$$

| O | D | E | A | Salida |
|---|---|---|---|--------|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 0 | 1 |

| | | | | |
|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 |
|---|---|---|---|---|

Arquitectura del prototipo



El prototipo creado cuenta con el nodemcu, el led rgb que es utilizado para representar el nodo de televisión, también los leds colores verdes y el otro amarillo para representar las dos lamparas en el nodo de iluminación, y también cuenta con el sensor de distancia que se aplica en ambos nodos.

Como se puede observar en la imagen, para conectar el Nodemcu en la placa se ocuparon cables macho-hembra, mientras que los demás dispositivos fueron conectados con machos-machos.

Código

// Este código funciona con la siguiente configuración:

// Board support API esp8266 (by ESP8266 Community) version 2.5.2

```
#include <ESP8266WiFi.h>
```

```
#include <PubSubClient.h>
```

```
#include <WiFiClient.h>
```

```
#include <MySQL_Connection.h>
```

```
#include <MySQL_Cursor.h>
```

```
//#define WIFI_SSID "Tec-IoT"
```

```
//#define WIFI_PASSWORD "PreguntarADeptoTI"
```

```
//#define WIFI_SSID "TC1004B"
```

```
//#define WIFI_PASSWORD "Tec-IoT2020"
```

```
#define WIFI_SSID "Totalplay-EDA5"
```

```
#define WIFI_PASSWORD "EDA54DAAvAZM253K"
```

```
#define SENSOR_A0 A0
```

```
#define Trigger D6
```

```
#define Echo D5
```

```
float tiempo;
```

```
float distancia;
```

```
long randomNumber;
```

```
#define LED1 D0 //R-Rojo ... IMPORTANTE: Validar si el LED es de ánodo o de cátodo común
```

```
#define LED2 D1 //G-Verde
```

```
#define LED3 D2 //B-Azul
```

```
#define LED4 D3 //Lampara 1
```

```
#define LED5 D4 //Lampara 2
```

```
WiFiClient espClient;
```

```
PubSubClient client(espClient);
```

```
MySQL_Connection conn((Client *)&espClient);
```

```
IPAddress server_addr(192, 168, 100, 18); // MySQL server IP
```

```
char user[] = "nodeMCU"; // MySQL user
```

```

char password[] = "nodeMCU";          // MySQL password

char insertSTMT[]="INSERT into TC1004B.Sensores (nodeID,sensorID,sensorValue)
VALUES(1,1,%d)";
char query[128];

const char *mqtt_server = "broker.hivemq.com";
const char *channelTopic = "PruebaAleIoT";


long tiempoAnterior = 0;
char msg[50];
int lecturaSensorA0 = 0;
int umbral = 50; //Este valor define el umbral abajo del que se enviará un mensaje al
"broker"

void setup_wifi() {
    delay(100);
    Serial.println();
    Serial.print("macAddress: ");
    Serial.println(WiFi.macAddress());
    //Iniciar por conectar con la red WiFi
    Serial.print("Conectando WiFi --> ");
    Serial.println(WIFI_SSID);
    WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
    }
    randomSeed(micros());
    Serial.println("");
    Serial.println("WiFi conectado!");
    Serial.print("IP address: ");
    Serial.println(WiFi.localIP());
} // End setup_wifi()

```



```

void callback(char* topic, byte* payload, unsigned int length) {
    char *cstring = (char *) payload;
    cstring[length] = '\0';    // Adds a terminate to end of string based on length of
current payload
    Serial.println(cstring);
    switch (cstring[1]) {
        case 'A': //Comando para apagar la tv led rojo
            Serial.println("\tRojo");
            digitalWrite(LED1, HIGH);
            digitalWrite(LED2, LOW);
            digitalWrite(LED3, HIGH);
            digitalWrite(LED4, LOW);
            digitalWrite(LED5, LOW);
            break;

        case 'E': //Comando para encenderla tv led verde
            Serial.println("\tVerde");
            digitalWrite(LED1, LOW);
            digitalWrite(LED2, HIGH);
            digitalWrite(LED3, HIGH);
            digitalWrite(LED4, LOW);
            digitalWrite(LED5, LOW);
            break;

        case 'S': //Comando para tomar dato del sensor y dependiendo del valor, prende o apaga
            Serial.println("\tAzul");
            digitalWrite(LED1, LOW);
            digitalWrite(LED2, LOW);
            digitalWrite(LED3, HIGH);
            digitalWrite(LED4, LOW);
            digitalWrite(LED5, LOW);
            delay(4000);
            long t; //timepo que demora en llegar el eco
            long d; //distancia en centimetros
            digitalWrite(Trigger, HIGH);
            delayMicroseconds(10);    //Enviamos un pulso de 10us

```

```
digitalWrite(Triple, LOW);
t = pulseIn(Echo, HIGH); //obtenemos el ancho del pulso
d = t / 59;           //escalamos el tiempo a una distancia en cm
Serial.print("Distancia: ");
Serial.print(d);      //Enviamos serialmente el valor de la distancia
Serial.print("cm");
Serial.println();
delay(100);           //Hacemos una pausa de
if (d<50){
  Serial.println(d);
  digitalWrite(LED1, LOW);
  digitalWrite(LED2, HIGH);
  digitalWrite(LED3, HIGH);
  break;
}
else{
  digitalWrite(LED1, HIGH);
  digitalWrite(LED2, LOW);
  digitalWrite(LED3, HIGH);
  break;
}
break;
```

```
case '1': //Comando para prender lampara 1 led amarillo
  Serial.println("\tRojo");
  digitalWrite(LED1, LOW);
  digitalWrite(LED2, LOW);
  digitalWrite(LED3, LOW);
  digitalWrite(LED4, HIGH);
  digitalWrite(LED5, LOW);
  break;
```

```
case '2': //Comando para prender lampara 2 led verde
  Serial.println("\tRojo");
  digitalWrite(LED1, LOW);
  digitalWrite(LED2, LOW);
  digitalWrite(LED3, LOW);
```

```
digitalWrite(LED4, LOW);  
digitalWrite(LED5, HIGH);  
break;
```

```
case '3': //Comando para prender lampara 1 y 2 led amarillo y verde  
Serial.println("\tRojo");  
digitalWrite(LED1, LOW);  
digitalWrite(LED2, LOW);  
digitalWrite(LED3, LOW);  
digitalWrite(LED4, HIGH);  
digitalWrite(LED5, HIGH);  
break;
```

```
case '4': //Comando para apagar lampara 1 y 2  
Serial.println("\tRojo");  
digitalWrite(LED1, LOW);  
digitalWrite(LED2, LOW);  
digitalWrite(LED3, LOW);  
digitalWrite(LED4, LOW);  
digitalWrite(LED5, LOW);  
break;
```

```
case '5': //Comando para prender lampara dependiendo la hora  
Serial.println("\tRojo");  
digitalWrite(LED1, LOW);  
digitalWrite(LED2, LOW);  
digitalWrite(LED3, LOW);  
randomNumber=random(1,24);  
Serial.println(randomNumber);  
if (randomNumber<12 && randomNumber>=8){  
digitalWrite(LED4, HIGH);  
digitalWrite(LED5, LOW);  
}  
if (randomNumber>=12 && randomNumber<22){  
digitalWrite(LED4, HIGH);  
digitalWrite(LED5, HIGH);  
}  
}
```

```

if (randomNumber>=22 && randomNumber<3){
digitalWrite(LED4, LOW);
digitalWrite(LED5, HIGH);
}
if (randomNumber>=3 && randomNumber<8){
digitalWrite(LED4, LOW);
digitalWrite(LED5, LOW);
}
break;

```

```

case '6': //Comando para prender lampara dependiendo la distancia
Serial.println("\tAzul");
digitalWrite(LED1, LOW);
digitalWrite(LED2, LOW);
digitalWrite(LED3, LOW);
digitalWrite(LED4, LOW);
digitalWrite(LED5, LOW);
delay(4000);
digitalWrite(Trip, HIGH);
delayMicroseconds(10);      //Enviamos un pulso de 10us
digitalWrite(Trip, LOW);
t = pulseIn(Echo, HIGH); //obtenemos el ancho del pulso
d = t / 59;      //escalamos el tiempo a una distancia en cm
Serial.print("Distancia: ");
Serial.print(d);      //Enviamos serialmente el valor de la distancia
Serial.print("cm");
Serial.println();
delay(100);      //Hacemos una pausa de
if (d<50){
Serial.println(d);
digitalWrite(LED4, HIGH);
digitalWrite(LED5, LOW);
break;
}
else{
digitalWrite(LED4, LOW);
digitalWrite(LED5, HIGH);
}

```

```

break;
}
break;

case '7': //Comando para prender lampara dependiendo la distancia y el horario
Serial.println("\tAzul");
digitalWrite(LED1, LOW);
digitalWrite(LED2, LOW);
digitalWrite(LED3, LOW);
digitalWrite(LED4, LOW);
digitalWrite(LED5, LOW);
delay(4000);
digitalWrite(Trip, HIGH);
delayMicroseconds(10); //Enviamos un pulso de 10us
digitalWrite(Trip, LOW);
t = pulseIn(Echo, HIGH); //obtenemos el ancho del pulso
d = t / 59; //escalamos el tiempo a una distancia en cm
Serial.print("Distancia: ");
Serial.print(d); //Enviamos serialmente el valor de la distancia
Serial.print("cm");
Serial.println();
delay(100); //Hacemos una pausa de
randomNumber=random(1,24);
Serial.println(randomNumber);
if (randomNumber<12 && randomNumber>=1 && d<50){
digitalWrite(LED4, HIGH);
digitalWrite(LED5, HIGH);
}
if (randomNumber>=22 && randomNumber<=24 || d>=50){
digitalWrite(LED4, HIGH);
digitalWrite(LED5, HIGH);
delay(400);
digitalWrite(LED4, LOW);
digitalWrite(LED5, LOW);
}
break;

```

```

    }
} // End callback(...)

void reconnect() {
    // Ciclar hasta lograr reconexión con "broker"
    while (!client.connected()) {
        Serial.print("Intentando conexión MQTT ...");
        // Create a random client ID
        String clientId = "ESP8266Client-";
        clientId += String(random(0xffff), HEX);
        //Intentar reconexión ...
        // ... en caso de que el "broker" tenga clientId, username y password
        // ... cambiar la siguiente línea por --> if (client.connect(clientId,userName,password))
        if (client.connect(clientId.c_str())) {
            Serial.println("Conectado al 'broker' MQTT!!!");
            //Ya conectado al "borker" MQTT suscribirse al tópico
            client.subscribe(channelTopic);
        } else {
            Serial.print("Error de conexión, rc=");
            Serial.print(client.state());
            Serial.println(" ... reintentando en 5 seg.");
            // Esperar 6 segundos para el próximo intento de conexión
            delay(6000);
        }
    }
} //end reconnect()

```

```

void setup() {
    pinMode(LED1, OUTPUT);
    pinMode(LED2, OUTPUT);
    pinMode(LED3, OUTPUT);
    pinMode(LED4, OUTPUT);
    pinMode(LED5, OUTPUT);
    pinMode(Trigger, OUTPUT); //pin como salida
    pinMode(Echo, INPUT); //pin como entrada
    digitalWrite(Trigger, LOW); //Inicializamos el pin con 0
    Serial.begin(9600);
}

```

```

setup_wifi();
client.setServer(mqtt_server, 1883); //Usando el puerto 1883, estándar en MQTT
client.setCallback(callback);
delay(2000);
Serial.println("\n");

Serial.println("Conectando con la base de datos ");

while (conn.connect(server_addr, 3306, user, password) != true) {
    delay(200);
    Serial.print ( "." );
}

Serial.println("");
Serial.println("Conectado al servidor SQL!!!");
delay(10);
}

void loop() {
    /*long t; //timepo que demora en llegar el eco
    long d; //distancia en centímetros

    digitalWrite(Triquer, HIGH);
    delayMicroseconds(10);      //Enviamos un pulso de 10us
    digitalWrite(Triquer, LOW);

    t = pulseIn(Echo, HIGH); //obtenemos el ancho del pulso
    d = t / 59;      //escalamos el tiempo a una distancia en cm

    Serial.print("Distancia: ");
    Serial.print(d);    //Enviamos serialmente el valor de la distancia
    Serial.print("cm");
    Serial.println();
    delay(100);      //Hacemos una pausa de
sprintf(query,insertSTMT,d);
    Serial.println("Registrando datos.");
    Serial.println(query);

```

```

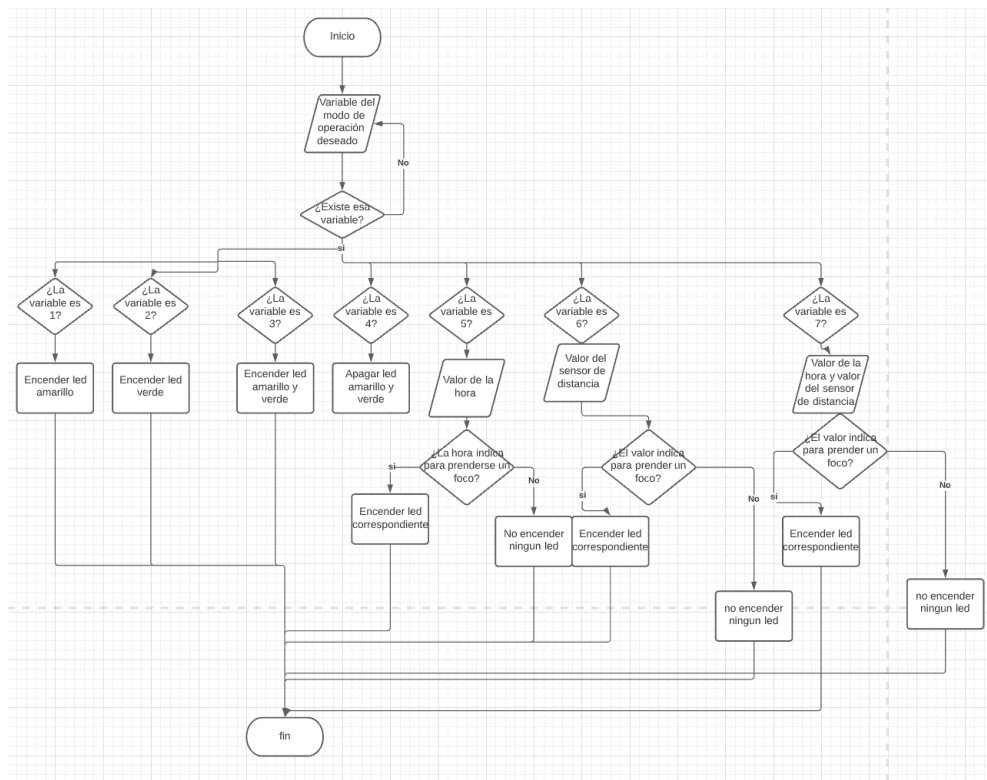
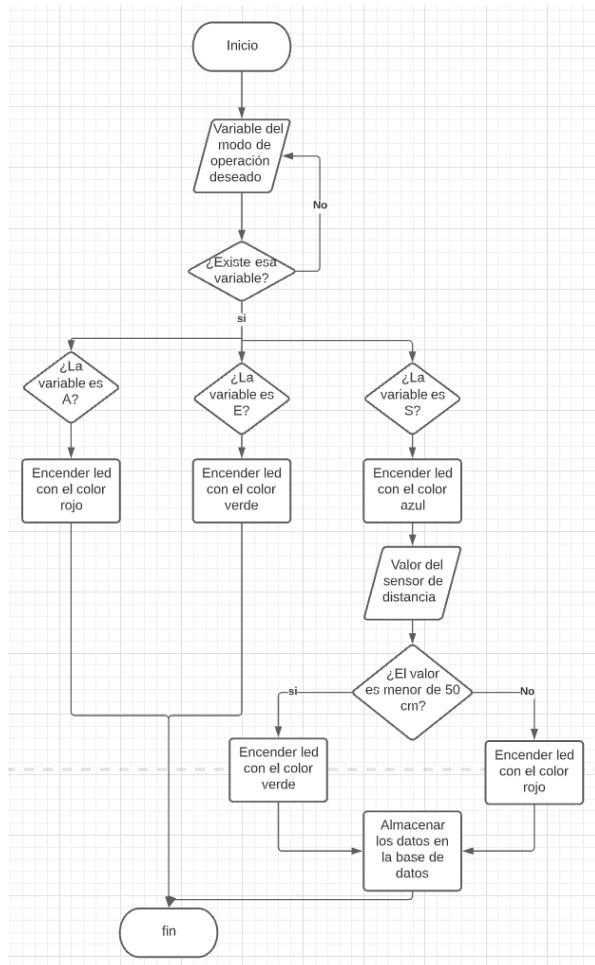
MySQL_Cursor *cur_mem = new MySQL_Cursor(&conn);
cur_mem->execute(query);
delete cur_mem;*/

if (!client.connected()) {
    reconnect();
}
client.loop();
long ahora = millis(); //Obtener el reloj del sistema en milisegundos
//Validar (por control de tiempo) si ya pasaron 0.5 segundos desde el reporte anterior
if (ahora - tiempoAnterior > 500) {
    tiempoAnterior = ahora;
    int lecturaSensorA0 = analogRead(SENSOR_A0);
    String msg = "The resistance value is: ";
    msg = msg + lecturaSensorA0;
    if (lecturaSensorA0 > umbral) //Este 'if' funciona, pero su construcción no cumple con
Codigo-Seguro, ¿por qué?
    msg = "0: " + msg; //Cuando el valor del sensor>umbral
    else
    msg = "1: " + msg; //Cuando el valor del sensor<=umbral
    char message[58];
    msg.toCharArray(message, 58);
    Serial.println(message);

    if (lecturaSensorA0 < umbral) { //publish sensor data to MQTT broker
        client.publish(channelTopic, message);
    }
}
} //End loop

```


Diagramas de flujo



Entrevista

- Perfil del cliente

Es un doctor que no tiene tanto tiempo, es joven y le importa mucho la familia, con dos hijos.

- Preguntas y respuesta de la entrevista

¿Usted o alguien de su familia se traspasa varias veces de un lugar a otro dentro de su casa (al jardín, cocina, cuarto, etc..)?

Si, su familia se traspasa varias veces, pero él no, ya que no está mucho tiempo en su casa.

¿Alguien de su familia suele dejar las luces prendidas? ¿Qué tan frecuente es?

Según su esposa normalmente si, sobretodo sus hijas, ya que se les olvida al moverse por las tareas del hogar

¿Tiene una gran cantidad de focos y botones para los focos en su casa?

Considera que tienen una cantidad normal de focos, y en cada foco tiene un switch

¿Cuál es el promedio de su recibo de luz?

El recibo sale caro, 4 mil pesos de recibo

¿Usted cree que gasta más en luz de lo que debería? ¿A qué podría deberse?

Gasta mucha luz, debido al usar el clima cuando no se ocupa o prender las luces cuando podrían evitarlo

¿Utiliza algún tipo de energía renovable en su hogar? De ser así, ¿cuáles utiliza?

No utilizan energía renovables, pero le interesaría en un futuro

¿En qué le beneficiaría poder controlar las luces de su casa desde cualquier parte del mundo?

Para poder disminuir el gasto que se genera, ya que al poder saber que luces están prendidas y saber si hay una persona en el cuarto, le podría permitir apagarla y ahorrar en luz.

¿Qué le mejoraría al servicio? (el servicio consiste en poder controlar las luces del hogar, ya sea apagarlas desde el móvil, dejarlas prendidas una cierta cantidad de tiempo o que se apaguen/prendan cuando alguien pase, depende de como se configure para el usuario)

Por ahora nada, me gustaría probarlo para ver que tal funciona y si en verdad podría ayudarme a disminuir el uso de luz.

- Comentario del Rol de observador:

cliente: es un señor de 40 años, que tiene un negocio chico de ropa

Conversación resumida: Empieza con preguntas personales (nombre, años, a que se dedica) para generar una entrevista más amena, como primera pregunta clave la cual fue “¿cómo afectó la pandemia en su vida?” ya que después de esto el cliente platico como lo perjudicó personalmente y en su negocio.

El entrevistador le preguntó cómo son las medidas de sanitización se manejan en su negocio, todo esto con el objetivo de que el entrevistador pueda ofrecerle el producto relacionado a la pandemia, el cual consiste en tomar la temperatura del usuario y negar la temperatura a las personas que no cuenten con el indicador, además lleva un registro de las personas dentro del establecimiento y el registro se actualiza cuando alguien sale o entra.

Opinión de la entrevista: Durante la entrevista el cliente mantuvo su actuación, y el entrevistador hizo un buen trabajo al preguntar sus datos y dar una breve introducción para después presentar sobre su producto para poder saber su opinión y en qué podría mejorarse, al finalizar hubo un cierre cordial por las dos partes.

- Comentario del Rol del cliente:

Como cliente es difícil actuar en algo que no estás acostumbrado, pero al tener referencia y conocidos, se me hizo bastante fácil interpretar lo que me pedían, además me pude dar cuenta la importancia que tiene un producto cuando se define a qué clase de clientes va dirigido su servicio/producto.

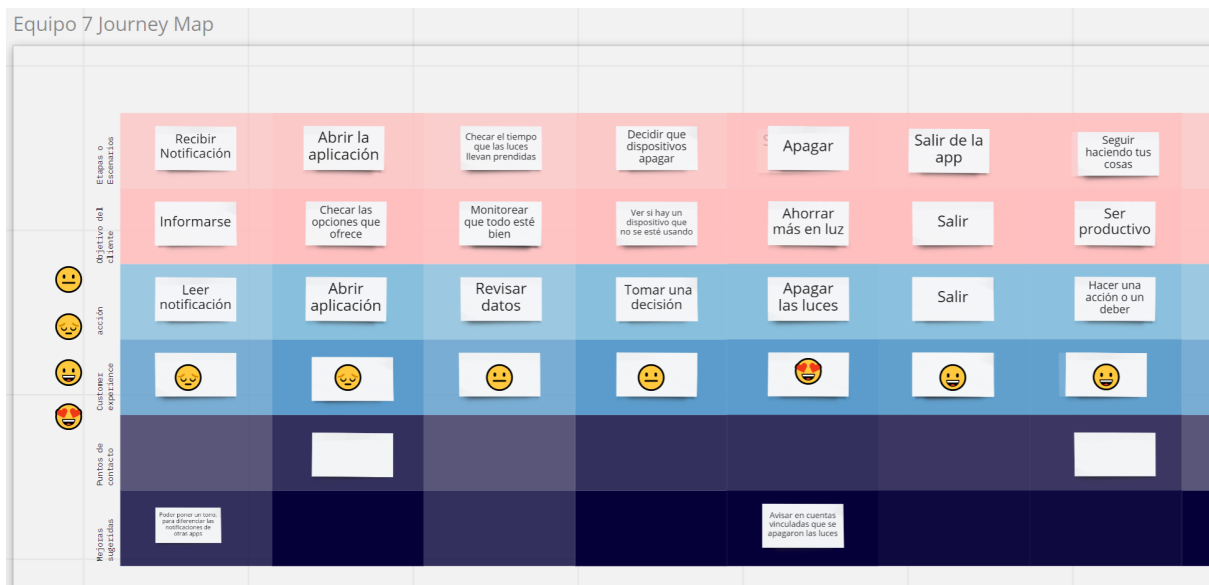
- Comentario del Rol del entrevistador:

Es difícil tomar en cuenta las respuestas del cliente mientras se hacen las preguntas. Por eso, es vital el rol del observador, ya que él puede analizar sus respuestas, tomar nota y poder hacer preguntas de seguimiento. Si pudiera hacerlo otra vez, intentaré tomar más en cuenta sus respuestas para poder improvisar preguntas nuevas más específicas a la información que nos acaba de dar él.

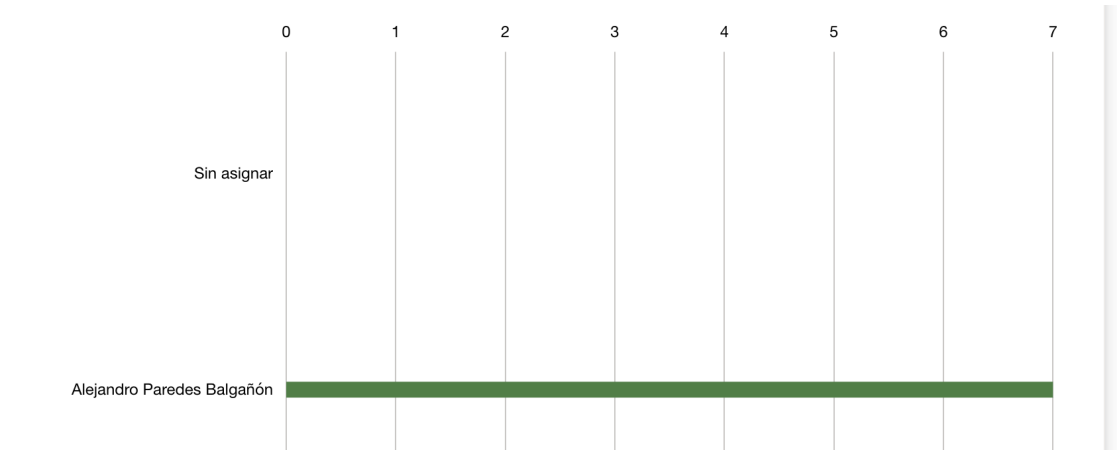
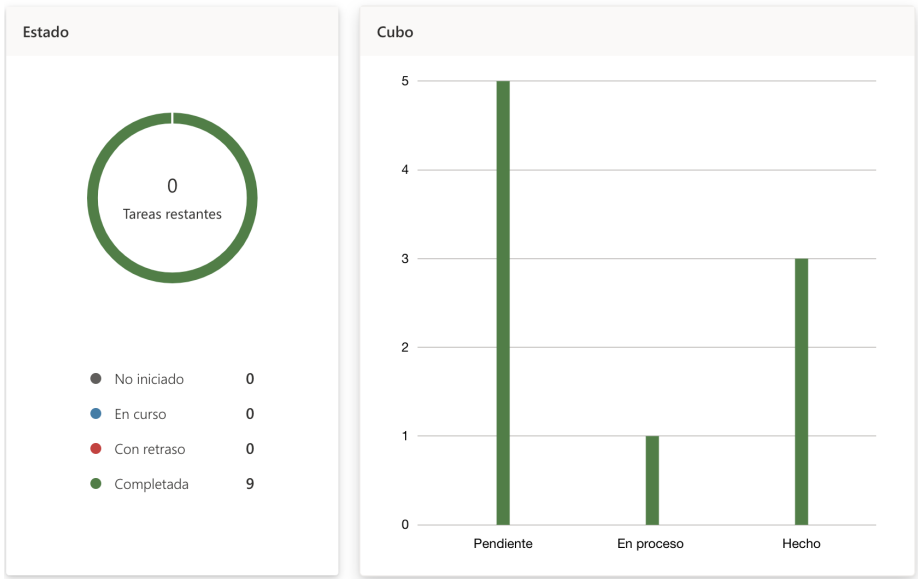
Mapa de Empatía



Journey Map



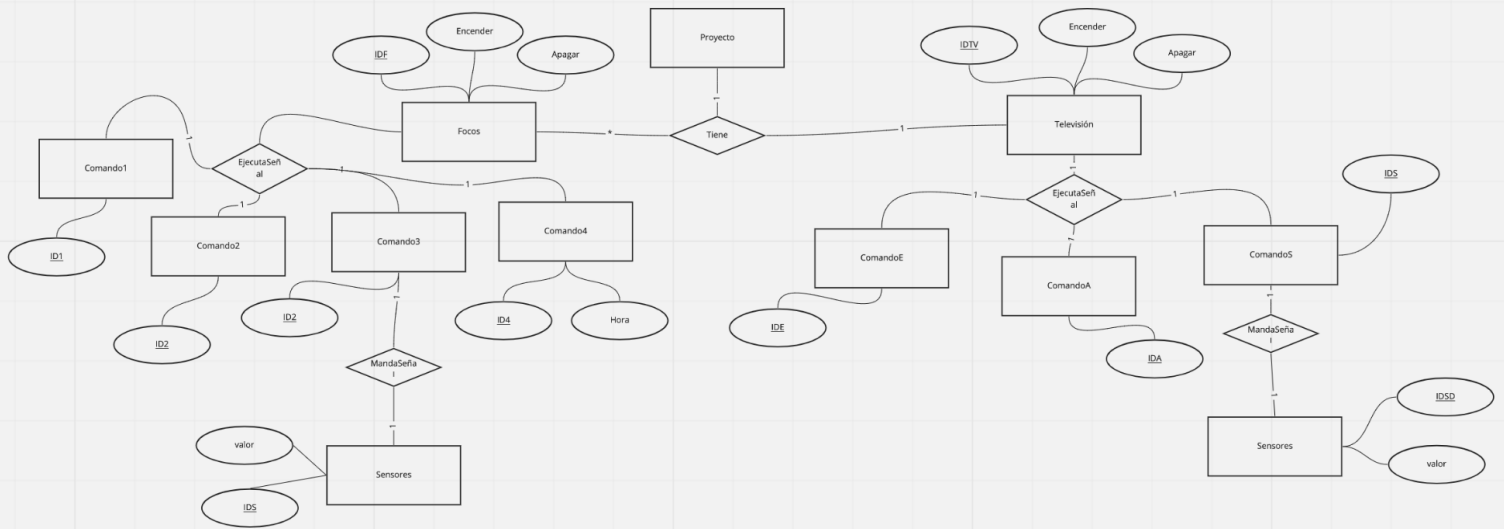
Planner



Dashboard



Modelo Entidad Relación

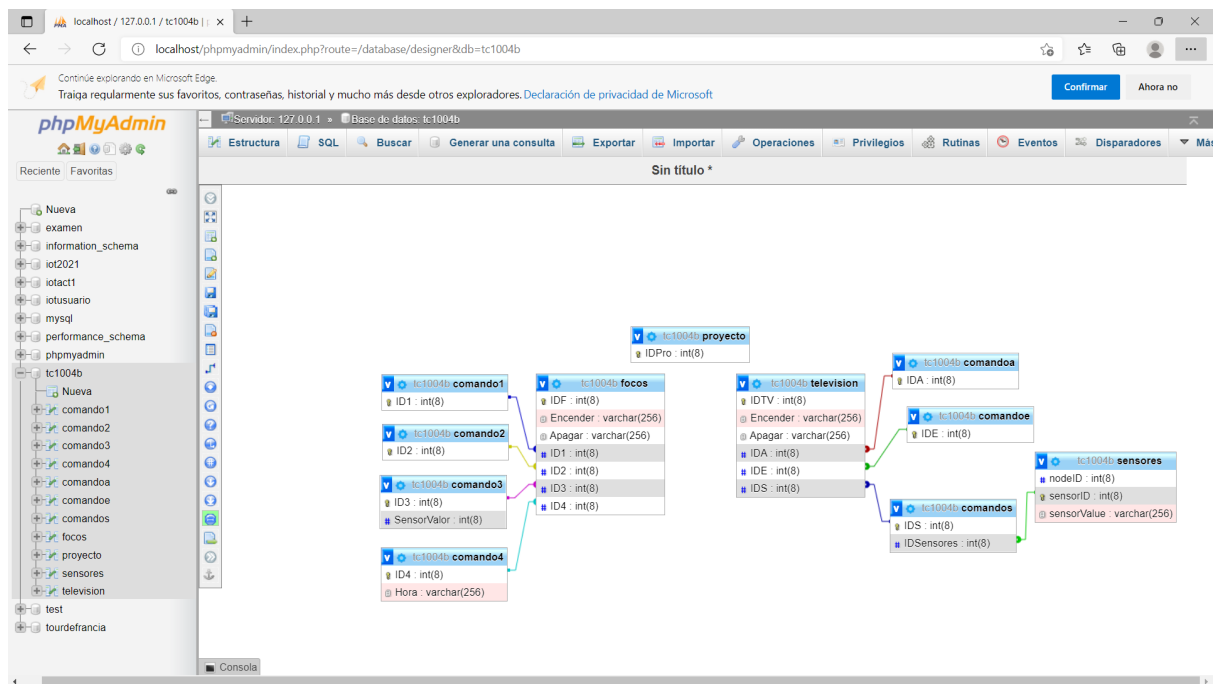


Mapeo Entidad Relación

| | |
|---|--|
| Proyecto (IDPRO) | |
| PK(IDPRO) | |
| Focos (IDF, Encender, Apagar, IDPRO) | |
| PK(IDF) | |
| FK(IDPRO) References Proyecto(IDPRO) | |
| Comando1 (ID1) | |
| PK(ID1) | |
| Comando2 (ID2) | |
| PK(ID2) | |
| Comando3 (ID3) | |
| PK(ID3) | |
| Comando4 (ID4, hora) | |
| PK(ID4) | |
| Television (IDTV, Encender, Apagar) | |
| PK(IDTV) | |
| ComandoA (IDA) | |
| PK(IDA) | |
| ComandoE (IDE) | |
| PK(IDE) | |
| ComandoS (IDS) | |
| PK(IDS) | |
| Sensores (IDSD, valor) | |
| PK(IDSD) | |

Diagrama de base de datos

The screenshot shows the phpMyAdmin interface for the database 'tc1004b'. The left sidebar displays the database structure, including tables like 'comando1', 'comando2', 'comando3', 'comando4', 'comandoa', 'comandoe', 'comandos', 'focos', 'proyecto', 'sensores', and 'television'. The main panel shows the 'Estructura' (Structure) tab for the 'comando1' table. The table has 4 columns: 'ID1' (int(8)), 'ID2' (int(8)), 'ID3' (int(8)), and 'SensorValor' (int(8)). The table is owned by 'InnoDB' and has a size of 16.0 KB.



Script SQL

```
-- phpMyAdmin SQL Dump
-- version 5.1.1
-- https://www.phpmyadmin.net/
--
-- Servidor: 127.0.0.1
-- Tiempo de generaci3n: 03-12-2021 a las 20:12:57
```

-- Versión del servidor: 10.4.21-MariaDB

-- Versión de PHP: 8.0.10

SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";

START TRANSACTION;

SET time_zone = "+00:00";

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;

/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS
*/;

/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;

/*!40101 SET NAMES utf8mb4 */;

--

-- Base de datos: `tc1004b`

--

-- -----

--

-- Estructura de tabla para la tabla `comando1`

--

CREATE TABLE `comando1` (
 `ID1` int(8) NOT NULL COMMENT 'ID COMANDO 1'
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

-- -----

--

-- Estructura de tabla para la tabla `comando2`

--

CREATE TABLE `comando2` (
 `ID2` int(8) NOT NULL COMMENT 'ID COMANDO 2'
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

-- -----

--

-- Estructura de tabla para la tabla `comando3`

--

```
CREATE TABLE `comando3` (  
  `ID3` int(8) NOT NULL COMMENT 'ID COMANDO 3',  
  `SensorValor` int(8) NOT NULL COMMENT 'VALOR DEL SENSOR EN EL FOCO'  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

-- -----

--

-- Estructura de tabla para la tabla `comando4`

--

```
CREATE TABLE `comando4` (  
  `ID4` int(8) NOT NULL COMMENT 'ID COMANDO 4',  
  `Hora` varchar(256) NOT NULL COMMENT 'HORA ACTUAL'  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

-- -----

--

-- Estructura de tabla para la tabla `comandoa`

--

```
CREATE TABLE `comandoa` (  
  `IDA` int(8) NOT NULL COMMENT 'ID DE COMANDO A'  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

-- -----

--

-- Estructura de tabla para la tabla `comandoe`

--

```
CREATE TABLE `comandoe` (  
  `IDE` int(8) NOT NULL COMMENT 'ID DE COMANDO E'  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

-- -----

--

-- Estructura de tabla para la tabla `comandos`

--

```
CREATE TABLE `comandos` (  
  `IDS` int(8) NOT NULL COMMENT 'ID DE COMANDO S',  
  `IDSensores` int(8) NOT NULL COMMENT 'SENSOR ID'  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

-- -----

--

-- Estructura de tabla para la tabla `focos`

--

```
CREATE TABLE `focos` (  
  `IDF` int(8) NOT NULL COMMENT 'ID DE NODO DE FOCOS',  
  `Encender` varchar(256) NOT NULL COMMENT 'ENCENDER FOCOS',  
  `Apagar` varchar(256) NOT NULL COMMENT 'APAGAR FOCOS',  
  `ID1` int(8) NOT NULL COMMENT 'ID DE Comando 1',  
  `ID2` int(8) NOT NULL COMMENT 'ID DE Comando 2',  
  `ID3` int(8) NOT NULL COMMENT 'ID DE Comando 3',  
  `ID4` int(8) NOT NULL COMMENT 'ID DE Comando 4'  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

-- -----

--

-- Estructura de tabla para la tabla `proyecto`

--

```
CREATE TABLE `proyecto` (  
  `IDPro` int(8) NOT NULL COMMENT 'Identificador de proyecto'  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

-- -----

--

-- Estructura de tabla para la tabla `sensores`

--

```
CREATE TABLE `sensores` (  
  `nodoID` int(8) NOT NULL COMMENT 'ID del nodo',  
  `sensorID` int(8) NOT NULL COMMENT 'ID SENSOR',  
  `sensorValue` varchar(256) NOT NULL COMMENT 'VALOR DEL SENSOR'  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

--

-- Volcado de datos para la tabla `sensores`

--

```
INSERT INTO `sensores` (`nodoID`, `sensorID`, `sensorValue`) VALUES  
(1, 1, '242');
```

-- -----

--

-- Estructura de tabla para la tabla `television`

--

```
CREATE TABLE `television` (  
  `IDTV` int(8) NOT NULL COMMENT 'ID DE TELEVISION',  
  `Encender` varchar(256) NOT NULL COMMENT 'Prender tv',  
  `Apagar` varchar(256) NOT NULL COMMENT 'Apagar tv',  
  `IDA` int(8) NOT NULL COMMENT 'ID DE COMANDO A',  
  `IDE` int(8) NOT NULL COMMENT 'ID DE COMANDO E',
```

```
`IDS` int(8) NOT NULL COMMENT 'ID DE COMANDO S'  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

```
--
```

```
-- Índices para tablas volcadas
```

```
--
```

```
--
```

```
-- Índices de la tabla `comando1`
```

```
--
```

```
ALTER TABLE `comando1`  
  ADD PRIMARY KEY (`ID1`);
```

```
--
```

```
-- Índices de la tabla `comando2`
```

```
--
```

```
ALTER TABLE `comando2`  
  ADD PRIMARY KEY (`ID2`);
```

```
--
```

```
-- Índices de la tabla `comando3`
```

```
--
```

```
ALTER TABLE `comando3`  
  ADD PRIMARY KEY (`ID3`);
```

```
--
```

```
-- Índices de la tabla `comando4`
```

```
--
```

```
ALTER TABLE `comando4`  
  ADD PRIMARY KEY (`ID4`);
```

```
--
```

```
-- Índices de la tabla `comandoa`
```

```
--
```

```
ALTER TABLE `comandoa`  
  ADD PRIMARY KEY (`IDA`);
```

```

--
-- Indices de la tabla `comandoe`
--
ALTER TABLE `comandoe`
  ADD PRIMARY KEY (`IDE`);

--
-- Indices de la tabla `comandos`
--
ALTER TABLE `comandos`
  ADD PRIMARY KEY (`IDS`),
  ADD KEY `IDSensores` (`IDSensores`);

--
-- Indices de la tabla `focos`
--
ALTER TABLE `focos`
  ADD PRIMARY KEY (`IDF`),
  ADD KEY `ID1` (`ID1`),
  ADD KEY `ID2` (`ID2`),
  ADD KEY `ID3` (`ID3`),
  ADD KEY `ID4` (`ID4`);

--
-- Indices de la tabla `proyecto`
--
ALTER TABLE `proyecto`
  ADD PRIMARY KEY (`IDPro`);

--
-- Indices de la tabla `sensores`
--
ALTER TABLE `sensores`
  ADD PRIMARY KEY (`sensorID`);

--
-- Indices de la tabla `television`

```

```

--
ALTER TABLE `television`
  ADD PRIMARY KEY (`IDTV`),
  ADD KEY `IDA` (`IDA`),
  ADD KEY `IDE` (`IDE`),
  ADD KEY `IDS` (`IDS`);

--
-- AUTO_INCREMENT de las tablas volcadas
--

--
-- AUTO_INCREMENT de la tabla `sensores`
--
ALTER TABLE `sensores`
  MODIFY `sensorID` int(8) NOT NULL AUTO_INCREMENT COMMENT 'ID SENSOR',
  AUTO_INCREMENT=2;

--
-- Restricciones para tablas volcadas
--

--
-- Filtros para la tabla `comandos`
--
ALTER TABLE `comandos`
  ADD CONSTRAINT `comandos_ibfk_1` FOREIGN KEY (`IDSensores`) REFERENCES
`sensores` (`sensorID`);

--
-- Filtros para la tabla `focos`
--
ALTER TABLE `focos`
  ADD CONSTRAINT `focos_ibfk_1` FOREIGN KEY (`ID1`) REFERENCES `comando1`
(`ID1`),
  ADD CONSTRAINT `focos_ibfk_2` FOREIGN KEY (`ID2`) REFERENCES `comando2`
(`ID2`),

```

```

ADD CONSTRAINT `focos_ibfk_3` FOREIGN KEY (`ID3`) REFERENCES `comando3`
(`ID3`),
ADD CONSTRAINT `focos_ibfk_4` FOREIGN KEY (`ID4`) REFERENCES `comando4`
(`ID4`);

--
-- Filtros para la tabla `television`
--
ALTER TABLE `television`
  ADD CONSTRAINT `television_ibfk_1` FOREIGN KEY (`IDA`) REFERENCES `comandoa`
(`IDA`),
  ADD CONSTRAINT `television_ibfk_2` FOREIGN KEY (`IDE`) REFERENCES `comandoe`
(`IDE`),
  ADD CONSTRAINT `television_ibfk_3` FOREIGN KEY (`IDS`) REFERENCES `comandos`
(`IDS`);
COMMIT;

/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;

```

Video de funcionamiento

<https://youtu.be/MDmPzVKYAu0>

Referencias:

BBVA. (SF). Conoce cuáles son las características de una casa inteligente y cómo tener una.. 3 diciembre 2021, de BBVA Sitio web: <https://www.bbva.mx/educacion-financiera/blog/que-es-una-casa-inteligente.html#:~:text=Se%20le%20llama%20casa%20inteligente,el%20uso%20de%20la%20dom%C3%B3tica.>

mysql. (sf). sql. 3 dic 2021, de mysql Sitio web: <https://www.mysql.com/>

TecBolivia(2013)Arduino Intermedio - Sensor Ultrasónico HC-SR04

“Youtube”: https://www.youtube.com/watch?v=b-5lcVebV_w&t=7s

turbo codigo(2020)Tutorial Fritzing 02. Circuito con NodeMCU ESP8266 y Ultrasonico HC-SR04 “Youtube”: <https://www.youtube.com/watch?v=WW9srEZkUI8&t=656s>