

Proyecto Final Smart Home System

Nombre: Maite Quintana Arreguin
Dependencia: Universidad de Guanajuato
Yuriria, Gto.
Correo: maytequintana05@gmail.com

Nombre: Martin Garcia Hurtado
Dependencia: Universidad de Guanajuato
Yuriria, Gto.
Correo: martingarciahurtado@gmail.com

Resumen—

SMART HOME SYSTEM es un sistema inteligente para una casa donde se controlará con el teléfono móvil.

Material:

2 ProtoBoard.
Cable para protoboard.
Jumpers
1 Modulo Esp32
8 Leds
8 Resistencias de 1K
2 Ventiladores
1 Push button
2 Motores

I. INTRODUCCION

Las casas inteligentes no son algo del futuro. Hoy en día existen hogares o edificios inteligentes, capaces de recibir información e interactuar con el usuario para ofrecer condiciones de confort de manera eficiente.

El objetivo de estas casas inteligentes es que las instalaciones eléctricas, de gas o de agua estén monitorizadas y controladas a distancia desde un dispositivo electrónico, normalmente el móvil o un ordenador. La automatización de los hogares, gracias a la telemedida y la domótica, está sirviendo para disponer de casas a medida, ajustadas a las necesidades particulares de cada individuo. Las interacciones que se pueden llevar a cabo pueden ser diseñadas para garantizar la

seguridad, la comodidad e incluso el ahorro de energía.

Una casa inteligente te puede avisar en el momento en el que exista una fuga de agua antes de que aparezca la gotera, del mismo modo que puede detectar incendios antes de ver el fuego. El sistema no solo es capaz de detectarlo y mandar el aviso, sino que también puede actuar en consecuencia hasta la llegada del especialista.

Con las casas inteligentes, además, serás capaz de controlar las cámaras en tiempo real desde tu móvil.

Las casas inteligentes aumentan el confort del usuario. Gracias a la domótica podrás seleccionar antes de llegar a casa, desde la temperatura del salón, hasta la iluminación de toda la casa.

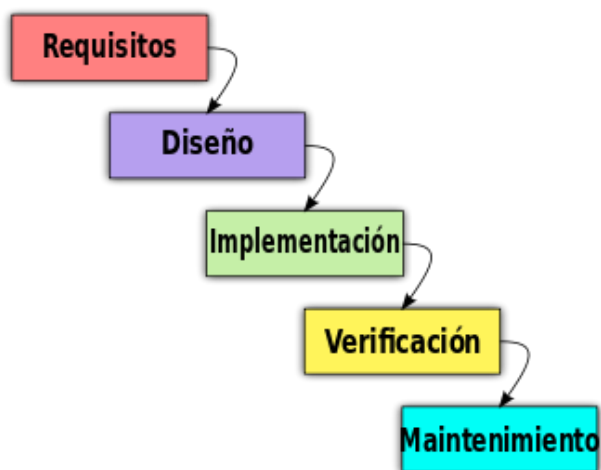
Las capacidades de las casas inteligentes son muy amplias. Todo lo que tenga que ver con el encendido y apagado de los equipos e instalaciones de iluminación y climatización, combinado con la incorporación de sensores de presencia, de temperatura o de cualquier otro tipo, hacen que las posibilidades de personalización de las casas inteligentes sean casi infinitas.

Además de la seguridad y el confort que aportan las casas inteligentes, es muy importante la capacidad que tienen estos sistemas para el ahorro y la eficiencia energética.

El diseño de las instalaciones de las casas inteligentes incluye equipos eficientes que, mediante su gestión integrada, consigue generar ahorros de agua, electricidad y combustible.

La metodología en cascada es un modelo lineal de diseño de software que emplea un proceso de diseño secuencial. El desarrollo fluye secuencialmente desde el punto inicial hasta el punto final, con varias etapas diferentes: planteamiento, iniciación, análisis, diseño, construcción, pruebas, implementación y mantenimiento.

El cuál es el método que se implementó en este proyecto.



II. MARCO TEORICO

Casa inteligente o domótica.

Las viviendas inteligentes utilizan una tecnología domótica, es decir, un conjunto de sistemas que están programados para controlar de manera automática el sistema energético, de seguridad, y comunicación de todos los espacios de la casa.

Los beneficios de una casa inteligente: Hacerse más cómoda, atractiva y sobre todo con una funcionalidad de última tecnología que le da sofisticación, prestancia y distinción al diseño y arquitectura de los espacios.

Las casas inteligentes brindan oportunidades para mejorar la calidad de vida y cambiar de manera positiva la realidad cotidiana, sobre todo para personas con capacidades diferentes y limitadas.

Apk Inventor: Es un entorno de desarrollo de software creado por Google Labs para la elaboración de aplicaciones destinadas al sistema operativo Android. El usuario puede, de forma visual y a partir de un conjunto de herramientas básicas, ir enlazando una serie de bloques para crear la aplicación. El sistema es gratuito y se puede descargar fácilmente de la web.

Las aplicaciones creadas con App Inventor están limitadas por su simplicidad, aunque permiten cubrir un gran número de necesidades básicas en un dispositivo móvil.

Con Google App Inventor, se espera un incremento importante en el número de aplicaciones para Android debido a dos grandes factores: la simplicidad de uso, que facilitará la aparición de un gran número de nuevas aplicaciones; y Google Play, el centro de distribución de aplicaciones para Android donde cualquier usuario puede distribuir sus creaciones libremente.

Modulo ESP32

ESP32 es la denominación de una familia de chips SoC de bajo costo y consumo de energía, con tecnología Wi-Fi y Bluetooth de modo dual integrada. El ESP32 emplea un microprocesador Tensilica Xtensa LX6 en sus variantes de simple y doble núcleo e incluye interruptores de antena, balun de radiofrecuencia, amplificador de potencia, amplificador receptor de bajo ruido, filtros, y módulos de administración de energía. El ESP32 fue creado y desarrollado por Espressif Systems y es fabricado por TSMC utilizando su proceso de 40 nm.¹ Es un sucesor de otro SoC, el ESP8266.

III. METODOLOGIA

La casa domótica no requiere un diagrama como tal, cada conexión, se verificaba su funcionalidad mediante pruebas, cada led para simular la luz de la casa fue conectado en cada uno de los cuartos, con madera, al igual que los ventiladores y los motores. Mediante el Módulo Esp32 será programado para el control de cada uno mediante una aplicación para dispositivos móviles con el sistema operativo Android.

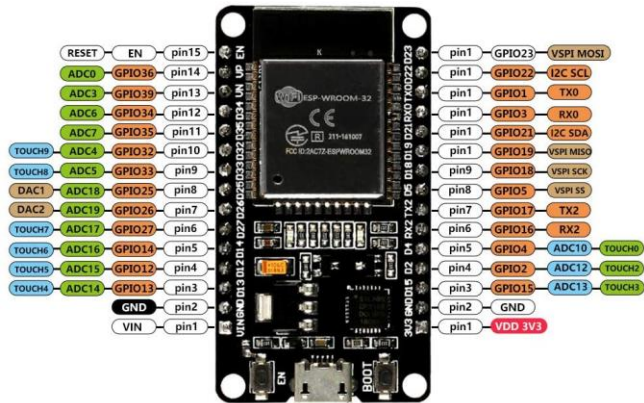
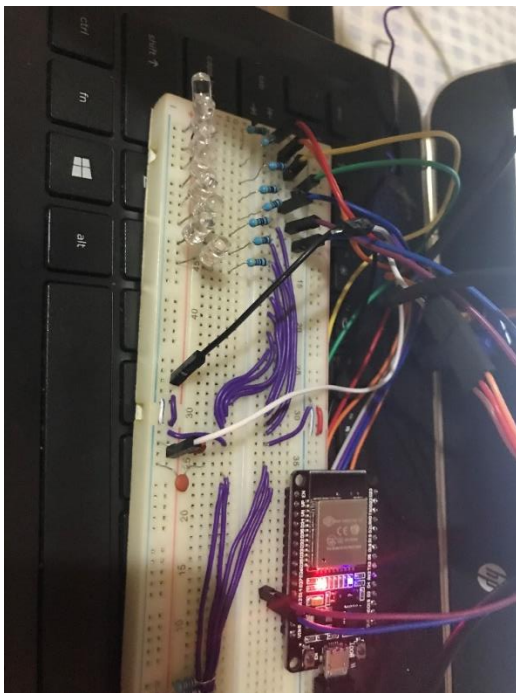
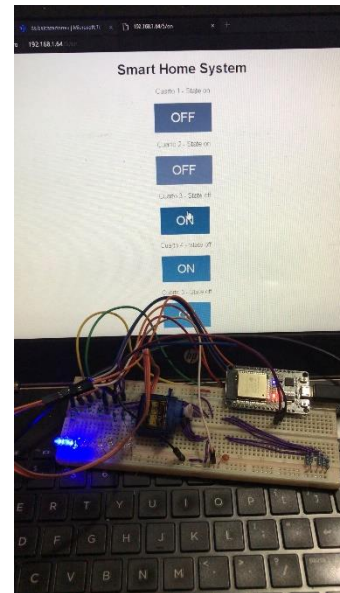


Fig.2 Datasheet Modulo Esp32.

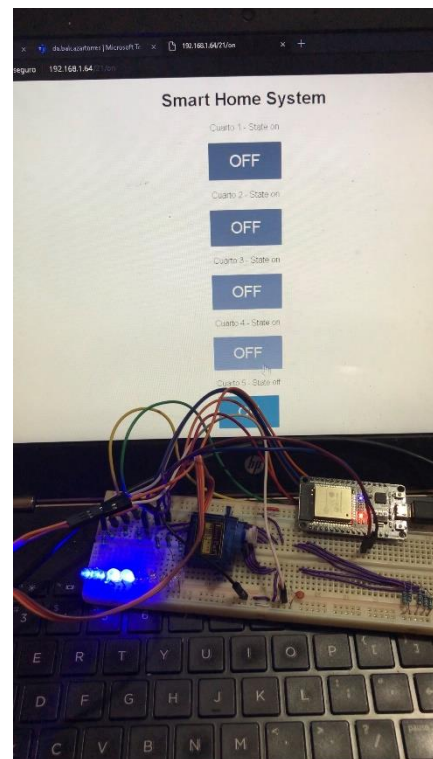
Al terminar de armar el circuito dio como resultado al que se mutra en la siguiente figura.



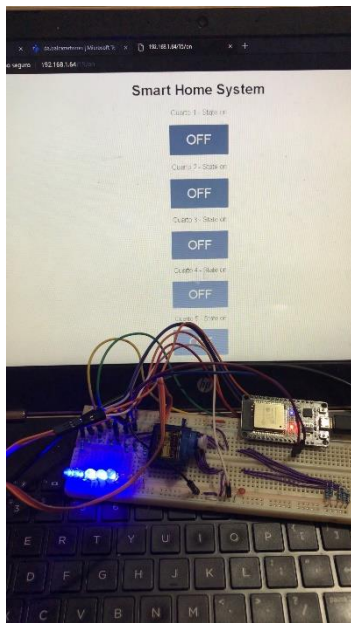
Ya una vez funcionando nuestro servidor web pusimos a prueba si funciona el encendido y apagado de cada función como se muestran a continuación.



2_leds_Encendidos



4_leds_Encendidos



5_leds_Encendidos

IV. CODIGO

```
#include <WiFi.h>

#include <ESP32Servo.h>

/***** Retardo*****/

int periodo = 10;

int periodo2=2000;

int periodo3=1000;

int periodo4=500;

unsigned long tiempo = 0;

// Ponemos nuestro wifi para probar

const char* ssid = /*"INFINITUM502D04" */"Martin_Garcia_H*";

const char* password = /*"23D581E62F" */"M4451072841g4582873H";

// Cambiamos el servidor web al puerto 80

WiFiServer server(80);

//Variable que guaradara la direccion HTTP

String header;

// Auxiliares para guardar las entradas que tiene

String output16State = "off";

String output5State = "off";

String output4State = "off";

String output3State = "off";
```

```
String output2State = "off";

String output21State = "off";

String output15State = "off";

String output14State = "off";

String output12State = "off";

// Assigna variables de entrada a los GPIO pins

const int output16 = 16;

const int output5 = 5;

const int output4 = 4;

const int output3 = 3;

const int output2 = 2;

const int output21 = 21;

const int output15 = 15;

const int output14 = 14;

const int output12 = 12;

Servo servo_4;

Servo servo_2;

IPAddress ip(192, 168, 1, 64);

IPAddress gateway(192, 168, 1, 254);

IPAddress subnet(255, 255, 255, 0);

void setup() {

    Serial.begin(115200);

    // Inicializar variables de salida

    pinMode(output16, OUTPUT);

    pinMode(output5, OUTPUT);

    pinMode(output4, OUTPUT);

    pinMode(output3, OUTPUT);

    pinMode(output21, OUTPUT);

    pinMode(output15, OUTPUT);

    pinMode(output14, OUTPUT);

    pinMode(output12, OUTPUT);

    // Apagar salidas

    digitalWrite(output16, LOW);

    digitalWrite(output5, LOW);

    digitalWrite(output4, LOW);

    digitalWrite(output3, LOW);
```

```
digitalWrite(output21, LOW);

digitalWrite(output15, LOW);

digitalWrite(output14, LOW);

digitalWrite(output12, LOW);

//motor

servo_4.attach(4);

servo_2.attach(2);

servo_4.write(0);

servo_2.write(0);

// delay(2000);

/**/

while (millis() < tiempo + periodo2){

/* ESPERA */

}

// Conectta a Wi-Fi con el SSID y contrasena que pusimos de ejemplo

while (millis() < tiempo + periodo){

/* ESPERA */

}

Serial.println();

WiFi.config(ip, gateway, subnet);

Serial.print("Connecting to ");

Serial.println(ssid);

WiFi.begin(ssid, password);

while (WiFi.status() != WL_CONNECTED) {

while (millis() < tiempo + periodo4){

/* ESPERA */

}

Serial.print(".");

}

// Imprime el IP local y empieza el servidor web

Serial.println("");

Serial.println("WiFi connected.");

Serial.println("IP address: ");

Serial.println(WiFi.localIP());

server.begin();

}
```

```
void loop(){
    tiempo=millis();

    WiFiClient client = server.available(); // Escucha a clientes entrantes

    if (client) {                                // Si queremos que un nuevo cliente se
conecte,

        Serial.println("New Client.");           // imprimir un mensaje en el puerto
serial

        String currentLine = "";                 // hacer una cadena para contener los
datos entrantes del cliente

        while (client.connected()) {            // Bucle mientras la cliente está
conectada

            if (client.available()) {            // Si hay bytes para leer de la cliente,

                char c = client.read();           // leer un byte, despues

                Serial.write(c);                  //imprímalo en el monitor de serie

                header += c;

                if (c == '\n') {                  // si el byte es un carácter de nueva línea

                    // si la línea actual está en blanco, tiene dos caracteres de nueva
línea seguidos.

                    // ese es el final de la solicitud HTTP del cliente, así que envíe una
respuesta:

                    if (currentLine.length() == 0) {

                        // Los encabezados HTTP siempre comienzan con un código de respuesta
(por ejemplo, HTTP / 1.1 200 OK)

                        // y un tipo de contenido para que el cliente sepa lo que viene,
luego una línea en blanco:

                        client.println("HTTP/1.1 200 OK");

                        client.println("Content-type:text/html");

                        client.println("Connection: close");

                        client.println();

                        // prende el GPIOs on and off luz de los cuartos

                        if (header.indexOf("GET /16/on") >= 0) {

                            Serial.println("GPIO 16 on");

                            output16State = "on";

                            digitalWrite(output16, HIGH);

                        } else if (header.indexOf("GET /16/off") >= 0) {

                            Serial.println("GPIO 16 off");

                            output16State = "off";

                            digitalWrite(output16, LOW);

                        } else if (header.indexOf("GET /5/on") >= 0) {

                            Serial.println("GPIO 5 on");

                        }

                    }

                }

            }

        }

    }

}
```

```

output5State = "on";
digitalWrite(output5, HIGH);
} else if (header.indexOf("GET /5/off") >= 0) {
    Serial.println("GPIO 5 off");
    output5State = "off";
    digitalWrite(output5, LOW);
} else if (header.indexOf("GET /4/on") >= 0) {
    Serial.println("GPIO 4 on");
    output4State = "on";
    digitalWrite(output4, HIGH);
    servo_4.write(90);
    while (millis() < tiempo + periodo3){
} else if (header.indexOf("GET /4/off") >= 0) {
    Serial.println("GPIO 4 off");
    output4State = "off";
    servo_4.write(0);
    while (millis() < tiempo + periodo3){
}
    digitalWrite(output4, LOW);
} else if (header.indexOf("GET /3/on") >= 0) {
    Serial.println("GPIO 3 on");
    output3State = "on";
    digitalWrite(output3, HIGH);
} else if (header.indexOf("GET /3/off") >= 0) {
    Serial.println("GPIO 3 off");
    output3State = "off";
    digitalWrite(output3, LOW);
} else if (header.indexOf("GET /21/on") >= 0) {
    Serial.println("GPIO 21 on");
    output21State = "on";
    digitalWrite(output21, HIGH);
} else if (header.indexOf("GET /21/off") >= 0) {
    Serial.println("GPIO 21 off");
    output21State = "off";
    digitalWrite(output21, LOW);
} else if (header.indexOf("GET /15/on") >= 0) {

```

```

Serial.println("GPIO 15 on");
output15State = "on";
digitalWrite(output15, HIGH);
} else if (header.indexOf("GET /15/off") >= 0) {
    Serial.println("GPIO 15 off");
    output15State = "off";
    digitalWrite(output15, LOW);
} else if (header.indexOf("GET /2/on") >= 0) {
    Serial.println("GPIO 2 on");
    output2State = "on";
    digitalWrite(output2, HIGH);
    servo_2.write(90);
    while (millis() < tiempo + periodo3){
}
} else if (header.indexOf("GET /2/off") >= 0) {
    Serial.println("GPIO 2 off");
    output2State = "off";
    servo_2.write(0);
    while (millis() < tiempo + periodo3){
}
    digitalWrite(output2, LOW);
} else if (header.indexOf("GET /14/on") >= 0) {
    Serial.println("GPIO 14 on");
    output14State = "on";
    digitalWrite(output14, HIGH);
} else if (header.indexOf("GET /14/off") >= 0) {
    Serial.println("GPIO 14 off");
    output14State = "off";
    digitalWrite(output14, LOW);
} else if (header.indexOf("GET /12/on") >= 0) {
    Serial.println("GPIO 12 on");
    output12State = "on";
    digitalWrite(output12, HIGH);
} else if (header.indexOf("GET /12/off") >= 0) {
    Serial.println("GPIO 12 off");
    output12State = "off";

```

```

digitalWrite(output12, LOW);

}

// Despliega pagina web

client.println("<!DOCTYPE html><html>");

client.println("<head><meta name=\"viewport\"
content=\"width=device-width, initial-scale=1\">");

client.println("<link rel=\"icon\" href=\"data:;\">");

// CSS  apagar/prende  botones

client.println("<style>html { font-family: Helvetica; display: inline-
block; margin: 0px auto; text-align: center;}");

client.println(".button { background-color: #195B6A; border: none;
color: white; padding: 16px 40px;");

client.println("text-decoration: none; font-size: 30px; margin: 2px;
cursor: pointer;}");

client.println(".button2 {background-color:
#77878A;}</style></head>");

//se modifiko y se agrego dos botones mas.

// client.println(".button2 {background-color: #77878A;}");

// client.println(".button2 {background-color:
#77878A;}</style></head>");

// El encabezado de la pagina

client.println("<body><h1>Smart Home System</h1>");

client.println("<p>Cuarto 1 - State " + output16State + "</p>");

if (output16State=="off") {

client.println("<p><a href=\"/16/on\"><button
class=\"button\">ON</button></a></p>");

} else {

client.println("<p><a href=\"/16/off\"><button class=\"button
button2\">OFF</button></a></p>");

}

// Mustrra el estado en el que se encuentra y Prende y Apaga
botons para GPIO 5

client.println("<p>Cuarto 2 - State " + output5State + "</p>");

// Si la estado t5State esta apagado, muestre opcion de Prender

if (output5State=="off") {

client.println("<p><a href=\"/5/on\"><button
class=\"button\">ON</button></a></p>");

```

```

} else {

client.println("<p><a href=\"/5/off\"><button class=\"button
button2\">OFF</button></a></p>");

}

// Muestra el estado actual y los botones ON / OFF para GPIO 4

client.println("<p>Cuarto 3 - State " + output3State + "</p>");

// Si output4State está desactivado, muestra el botón ON

if (output3State=="off") {

client.println("<p><a href=\"/3/on\"><button
class=\"button\">ON</button></a></p>");

} else {

client.println("<p><a href=\"/3/off\"><button class=\"button
button2\">OFF</button></a></p>");

}

client.println("<p>Cuarto 4 - State " + output21State + "</p>");

if (output21State=="off") {

client.println("<p><a href=\"/21/on\"><button
class=\"button\">ON</button></a></p>");

} else {

client.println("<p><a href=\"/21/off\"><button class=\"button
button2\">OFF</button></a></p>");

}

client.println("<p>Cuarto 5 - State " + output15State + "</p>");

if (output15State=="off") {

client.println("<p><a href=\"/15/on\"><button
class=\"button\">ON</button></a></p>");

} else {

client.println("<p><a href=\"/15/off\"><button class=\"button
button2\">OFF</button></a></p>");

}

//motor puerta

client.println("<p>Puerta Abrir/Cerrar - State " + output4State +
"</p>");

// If the output4State is off, it displays the ON button

if (output4State=="off") {

client.println("<p><a href=\"/4/on\"><button
class=\"button\">ON</button></a></p>");

} else {

```

```

        client.println("<p><a href=\"4/off\"><button class=\"button
button2\">OFF</button></a></p>");

    }

    //garage

    client.println("<p>Puerta Abrir/Cerrar - State " + output2State +
"</p>");

    // If the output4State is off, it displays the ON button

    if (output2State=="off") {

        client.println("<p><a href=\"2/on\"><button
class=\"button\">ON</button></a></p>");

    } else {

        client.println("<p><a href=\"2/off\"><button class=\"button
button2\">OFF</button></a></p>");

    }

    // Ventiladores.

    client.println("<p>Ventilacion Cuarto 1 - State " + output14State +
"</p>");

    if (output14State=="off") {

        client.println("<p><a href=\"14/on\"><button
class=\"button\">ON</button></a></p>");

    } else {

        client.println("<p><a href=\"14/off\"><button class=\"button
button2\">OFF</button></a></p>");

    }

    client.println("<p>Ventilacion Cuarto 2 - State " + output12State +
"</p>");

    if (output12State=="off") {

        client.println("<p><a href=\"12/on\"><button
class=\"button\">ON</button></a></p>");

    } else {

        client.println("<p><a href=\"12/off\"><button class=\"button
button2\">OFF</button></a></p>");

    }

    client.println("</body></html>");

    // La respuesta HTTP termina con otra línea en blanco

    client.println();

    // Salir del bucle while

    break;

} else { // si tienes una nueva línea, borra currentLine

    currentLine = "";

```

```

    }

    } else if (c != '\r') { // si tiene algo más que un carácter de retorno de
carro,

        currentLine += c;    // agregarlo al final de currentLine

    }

    }

    }

    // Limpia el encabezado

    header = "";

    // Para la conexion

    client.stop();

    Serial.println("Cliente disconnected.");

    Serial.println("");

    }

    }

```

V. CONCLUSIONES

Hay quienes creen que convivir con esta tecnología podría llevar a los usuarios a aislarse del mundo y a encerrarse entre las paredes de su casa. Quizá sean suposiciones demasiado oscuras y tempranas, dado que todavía no existe una muestra de usuarios lo suficientemente grande como para poder afirmar esto.

La realidad es que las casas domóticas tienen demasiados beneficios para los usuarios que la utilizan desde comodidad y seguridad además de entretenimiento con sus diferentes funcionalidades que se pueden ir implementando para satisfacer las necesidades de cada usuario.

VI. BIBLIOGRAFÍAS.

- [1] Electronics
<http://hyperphysics.phy-astr.gsu.edu/hbasees/Electronic/schmitt.html>.
- [2] Arduino
 xer, Jonathan; Blemings, Hugh (28 de diciembre de 2009). Practical Arduino: Cool Projects for Open Source Hardware (1ª edición). Apress. p. 500. ISBN 1430224770. Archivado desde el original el 5 de diciembre de 2010. Consultado el 17 de mayo de 2010

