



# HouseTIC

---

Manual Técnico

**Materia:**

Programación de Sistemas  
Telemáticos

**Título:**

Proyecto de ingeniería usando  
plataformas de prototipado con  
servicios basados en red

**Profesora:**

Msig. Adriana E. Collaguazo  
Jaramillo

**Grupo: 6**

**Integrantes:**

Juan Guillermo Vera García  
Alfredo Andrés Ante Vargas  
Hugo Martin Andérica Urquizo  
José Daniel Moreno González

## CONTENIDO

1. RESUMEN EJECTUTIVO .....	3
2. INTRODUCCION .....	3
3. DESCRIPCION DEL PROBLEMA.....	3
4. OBJETIVOS ESPECIFICOS .....	3
5. PLANTEAMIENTO DE LA SOLUCION .....	4
6. RECURSOS .....	4
8. DIAGRAMAS.....	5
6	
9. CODIGO FUENTE .....	6
10. ANALISIS DE PRESUPUESTO .....	11
11. CONCLUSIONES.....	11
12. REFERENCIAS.....	11

## **1. RESUMEN EJECTUTIVO**

Este proyecto llamado HouseTIC busca mejorar las comodidades del hogar mediante la posibilidad de controlar los distintos dispositivos dentro del mismo de manera remota, tanto como par cuando el usuario se encuentre dentro del sitio o fuera de este. El sistema se encuentra en fase de prototipado a menor escala, pero esta basado en los mismos principios que corresponderían a una implementación a escala real.

Este dispositivo de control domótico esta enfocado a todas aquellas personas de clase media alta que busquen una mayor comodidad tanto en su hogar como en su zona de trabajo.

## **2. INTRODUCCION**

El presente manual muestra de forma detallada toda la información técnica acerca del funcionamiento y desarrollo del sistema empleado para el control domótico de una casa promedio.

Se detalla el procedimiento de implementación del aplicativo móvil para el sistema operativo Android en el software Android Studio. Incluye una descripción de la funcionalidad del sistema, tanto en hardware como en software, así como un detalle del presupuesto general del proyecto.

## **3. DESCRIPCION DEL PROBLEMA**

El avance tecnológico ha hecho que el hombre se vuelva dependiente en gran medida de los diversos dispositivos electrónicos de uso cotidiano. En la actualidad, la gran mayoría de los hogares tienen el inconveniente de que, para activar un dispositivo electrónico, se debe dirigirse al mismo y accionarlo manualmente. Esto puede resultar un poco incomodo en casas de grandes dimensiones o en casos en los que el usuario no se encuentre en su domicilio y sin embargo requiera encender o apagar un dispositivo.

Este proyecto proporciona un sistema capaz de poder controlar todos estos dispositivos desde cualquier lugar, sea dentro de la casa o en cualquier otro sitio.

## **4. OBJETIVOS ESPECIFICOS**

- Ofrecer información técnica acerca de la configuración de la aplicación móvil.
- Especificar detalladamente la estructura del aplicativo y su diseño.
- Definir especificaciones acerca de requerimientos de hardware y software necesarios para el correcto funcionamiento de la aplicación.
- Describir los métodos y herramientas empleados para el desarrollo del aplicativo

## **5. PLANTEAMIENTO DE LA SOLUCION**

La solución consiste en facilitar el día a día en el hogar o área de trabajo, brindando la oportunidad de controlar dispositivos de uso cotidiano mediante solo un clic desde un smartphone con conexión a internet.

Para la realización de este proyecto se implementó un sistema constituido por sensores, motores y leds controlados mediante un Arduino Uno. Se utilizó una base de datos remota RemoteMySQL, en la cual se registraron los estados de los dispositivos a controlar (encendido o apagado). Paralelamente se diseñó una aplicación móvil en Android Studio, la cual permite al usuario interactuar con el sistema y definir manualmente el estado de determinado equipo de determinada área de la casa. La base de datos lee los cambios realizados por el usuario a través de la aplicación móvil, y comunica esta información con el Arduino para controlar un determinado dispositivo.

## **6. RECURSOS**

Para la implementación del prototipo se utilizó el siguiente hardware:

- Arduino UNO
- Leds
- Motor DC
- Servo Motor
- Relés
- Sensor de proximidad
- Pulsador
- Bocina
- Fuente 5V
- Modulo Wifi

Las herramientas de software empleadas fueron:

- Android Studio
- RemoteMySQL
- Arduino
- MicroPython

## **7. IMPLEMENTACIÓN DEL PROYECTO**

### **Aplicativo móvil**

Se inicio creando el aplicativo en Android Studio. Se crearon 8 diferentes activities de las cuales el primero se empleó para una interfaz de login que funciona con un ingreso a través de una cuenta de Google.

El segundo activity consiste en un menú principal el cual se agregó un TextView “estaciones”, seguido de 6 botones cada uno con imágenes que indican su funcionalidad; 5 de ellos abren un activity diferente, cada uno para el control de un área

específica del hogar. Dichos activities constan de un TextView correspondiente a cada dispositivo que se quiera controlar acompañado de un switch para cambiar su estado a activado o desactivado.

Por último, el botón restante del menú principal tiene como función salir del menú principal hacia la pagina de login.

## **Base de Datos**

Se creo una base de datos en la nube RemoteMySQL, en esta se crearon 4 tablas: cuarto, cuenta, dispositivo y hogar. La tabla Cuenta almacena información de usuario, contraseña correo y número de teléfono; se fijó como clave primaria el nombre de usuario.

La tabla Hogar contiene la información de un hogar especifico en el cual se instalará el sistema. Sus datos son: un id que se usara como clave primaria, dirección, terreno, numero de cuartos que tiene el domicilio y tiene como clave foránea el nombre de usuario de la tabla Cuenta.

La tabla Cuarto fue creada para almacenar cuartos o estaciones del domicilio. Los datos que contiene son: id, nombre, área, numero de dispositivos y como clave foránea el id del hogar correspondiente.

Por último, está la tabla Dispositivos, que almacena los dispositivos que se van a controlar en un cuarto determinado. Tiene como datos un id, el nombre, descripción, estado (encendido o apagado), y como clave foránea el id del cuarto.

## **Arduino y dispositivos electrónicos**

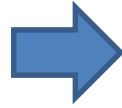
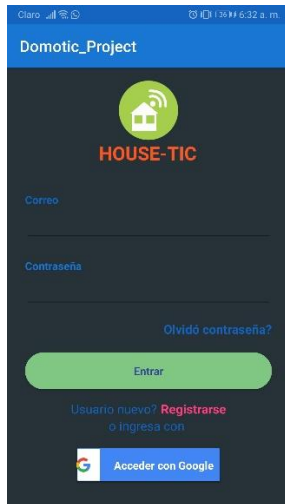
Para el circuito se utilizó un Arduino UNO el cual se conectó por medio de 8 relés a los diferentes dispositivos. Adicionalmente se conectó al mismo un módulo Wifi ESP32, para darle conectividad a internet y permitirle acceder a la base de datos.

## **Maqueta**

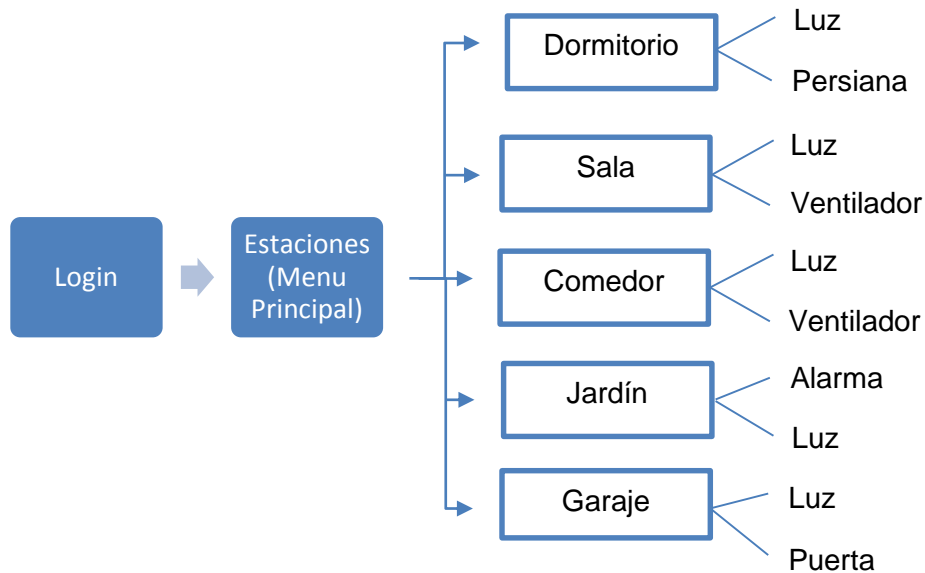
Para mostrar la funcionalidad del sistema implementado, se implementó una maqueta que consiste en una casa de juguete, separada por salas que representarían las estaciones y adaptada con dispositivos electrónicos demostrativos de funciones específicas del sistema, es decir, los leds para simular los focos eléctricos, motores dc para simular ventiladores, la alarma en conjunto con el sensor de proximidad, motores de paso y servo para controlar ventanas y puertas, etc.

## **8. DIAGRAMAS**

### **Diagrama de diseño del Proyecto**



## Diagrama de Despliegue



## 9. CODIGO FUENTE

```

Arduino:
#include <WiFi.h>                                     // esp32 library
#include <IOXhop_FirebaseESP32.h>                     // firebase library

#define FIREBASE_HOST "prueba123-46ebf.firebaseio.com" // the
project name address from firebase id
#define FIREBASE_AUTH "Z0kvKVCyVgcW98Wxs2YiJiFILCtOcCZaWg0UeAJH"
// the secret key generated from firebase
#define WIFI_SSID "APGUESTS"                          // input your home or public
wifi name
#define WIFI_PASSWORD "102030ap!"                      //password of wifi ssid
  
```

```

String fireStatus = "";
firebase
int led = 2;

void setup() {
  Serial.begin(9600);
  delay(1000);
  pinMode(2, OUTPUT);
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD); //try to connect
  with wifi
  Serial.print("Connecting to ");
  Serial.print(WIFI_SSID);
  while (WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
    delay(500);
  }
  Serial.println();
  Serial.print("Connected to ");
  Serial.println(WIFI_SSID);
  Serial.print("IP Address is : ");
  Serial.println(WiFi.localIP()); //print local IP address
  Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH); //
  connect to firebase
  Firebase.setString("LED_STATUS", "OFF"); //send initial
  string of led status
}

void loop() {

  fireStatus = Firebase.getString("LED_STATUS"); // get led status input
  from firebase
  if (fireStatus == "ON") { // compare the input of led status received from
  firebase
    Serial.println("Led Turned ON");
    digitalWrite(2, HIGH); // make output led ON
  }

  else if (fireStatus == "OFF") { // compare the input of led status received from
  firebase
    Serial.println("Led Turned OFF");
    digitalWrite(2, LOW); // make output led OFF
  }

  else {
    Serial.println("Wrong Credential! Please send ON/OFF");
  }
}

const int size_array = 5;
//const int rele=11;
int pin11 = 8;

```

```

int pin12 = 9;
int pin21 = 10;
int pin22 = 11;
int pin31 = 0;
int pin32 = 1;
int pin41 = 0;
int pin42 = 1;
int pin51 = 0;
int pin52 = 1;

int stt11 = 0;
int stt12 = 1;
int stt21 = 1;
int stt22 = 0;
int stt31 = 0;
int stt32 = 1;
int stt41 = 0;
int stt42 = 1;
int stt51 = 0;
int stt52 = 1;
int stt[10]; // Estados rele
int bd[10];
void setup() {
    // put your setup code here, to run once:
    Serial.begin(9600);
    //pinMode(rele,OUTPUT);
    pinMode(pin11,OUTPUT);
    pinMode(pin12,OUTPUT);
    pinMode(pin21,OUTPUT);
    pinMode(pin22,OUTPUT);
    pinMode(pin31,OUTPUT);
    pinMode(pin32,OUTPUT);
    pinMode(pin41,OUTPUT);
    pinMode(pin42,OUTPUT);
    pinMode(pin51,OUTPUT);
    pinMode(pin52,OUTPUT);
}

void loop() {
    // put your main code here, to run repeatedly:
    int A1[10] = {1,0,1,0,0,1,0,1,0,1}; //A1 estado rele
    stt[0]=stt11;
    stt[1]=stt12;
    stt[2]=stt21;
    stt[3]=stt22;
    stt[4]=stt31;
    stt[5]=stt32;
    stt[6]=stt41;
    stt[7]=stt42;
    stt[8]=stt51;
    stt[9]=stt52;

```



```

for(int i=0; i<size_array; i++){
    if(A1[i]<stt[i]){
        //digitalWrite(rele, LOW);
        Serial.println("Encendido");
        Serial.println("i: "+String(i)+"--"+String(A1[i])+ "   j: "+String(i)+"--"+String(stt[i]));
        //delay(2500);
        if(i==0){
            digitalWrite(pin11, LOW);
            Serial.println("1");
        }else if(i==1){
            digitalWrite(pin12, LOW);
            Serial.println("2");
            //delay(2500);
        }else if(i==2){
            digitalWrite(pin21, LOW);
            Serial.println("3");
            //delay(2500);
        }else if(i==3){
            digitalWrite(pin22, LOW);
            Serial.println("4");
            //delay(2500);
        }else if(i==4){
            digitalWrite(pin31, LOW);
            Serial.println("5");
            //delay(2500);
        }else if(i==5){
            digitalWrite(pin32, LOW);
            Serial.println("6");
        }else if(i==6){
            digitalWrite(pin41, LOW);
            Serial.println("7");
        }else if(i==7){
            digitalWrite(pin42, LOW);
            Serial.println("8");
        }else if(i==8){
            digitalWrite(pin51, LOW);
            Serial.println("9");
        }else if(i==9){
            digitalWrite(pin52, LOW);
            Serial.println("10");
        }
    }else if(A1[i]>=stt[i] && stt[i]!=1){
        //digitalWrite(rele, HIGH);
        Serial.println("Apagado");
        Serial.println("i: "+String(i)+"--"+String(A1[i])+ "   j: "+String(i)+"--"+String(stt[i]));
        //delay(2500);
        if(i==0){
            digitalWrite(pin11, HIGH);
            Serial.println("1");
            //delay(2500);
        }
    }
}

```

```

    }else if(i==1){
        digitalWrite(pin12, HIGH);
        Serial.println("2");
        //delay(2500);
    }else if(i==2){
        digitalWrite(pin21, HIGH);
        Serial.println("3");
        //delay(2500);
    }else if(i==3){
        digitalWrite(pin22, HIGH);
        Serial.println("4");
        //delay(2500);
    }else if(i==4){
        digitalWrite(pin31, HIGH);
        Serial.println("5");
        //delay(2500);
    }else if(i==5){
        digitalWrite(pin32, HIGH);
        Serial.println("6");
    }else if(i==6){
        digitalWrite(pin41, HIGH);
        Serial.println("7");
    }else if(i==7){
        digitalWrite(pin42, HIGH);
        Serial.println("8");
    }else if(i==8){
        digitalWrite(pin51, HIGH);
        Serial.println("9");
    }else if(i==9){
        digitalWrite(pin52, HIGH);
        Serial.println("10");
    }
    }else if(A1[i]==stt[i]){
        Serial.println("Nada");
        Serial.println("i: "+String(i)+"--"+String(A1[i])+"   j: "+String(i)+"--"+String(stt[i]));
        //delay(4500);
    }
}
Serial.println("-----");
delay(6000);
}

```

## 10. ANALISIS DE PRESUPUESTO

	Cantidad	Precio unidad	Precio total
Relés	11	\$1,75	\$14
Sensor proximidad	1	\$2,5	\$2,5
Motor dc	3	\$1,5	\$4,5
Luces	10	\$0,15	\$1,5
Servo	2	\$3	\$6
Pulsador	1	\$1	\$1
Bocina	1	\$2	\$2
Arduino mega	1	\$20	\$20
Fuente 5V	1	\$9	\$9
Módulo Wifi	1	\$4,5	\$4,5
		<b>Total</b>	<b>\$65</b>

## 11. CONCLUSIONES

- El proyecto muestra de manera clara como es la implementación de un sistema de control domótico para un hogar promedio. Se pudo conocer la complejidad de su instalación tanto del hardware como de la creación del software requerido y su interacción con la base de datos remota.
- La conexión tanto del aplicativo móvil como del Arduino para la comunicación con la base de datos es un proceso complejo ya que es muy distinto a utilizar un servidor local.
- El proyecto es factible de realizar ya que resulta relativamente económico en comparación a dispositivos implementados hoy en día.

## 12. REFERENCIAS

- Base De Datos Con Android: <https://codigofacilito.com/articulos/base-de-datos-con-android>
- Exploring Home Automation and Domotics: <https://www.thoughtco.com/what-is-a-smart-house-domotics-177572>
- De MySQL a Arduino, leer datos de la base de datos: <https://www.rinconingenieril.es/de-mysql-a-arduino/>

