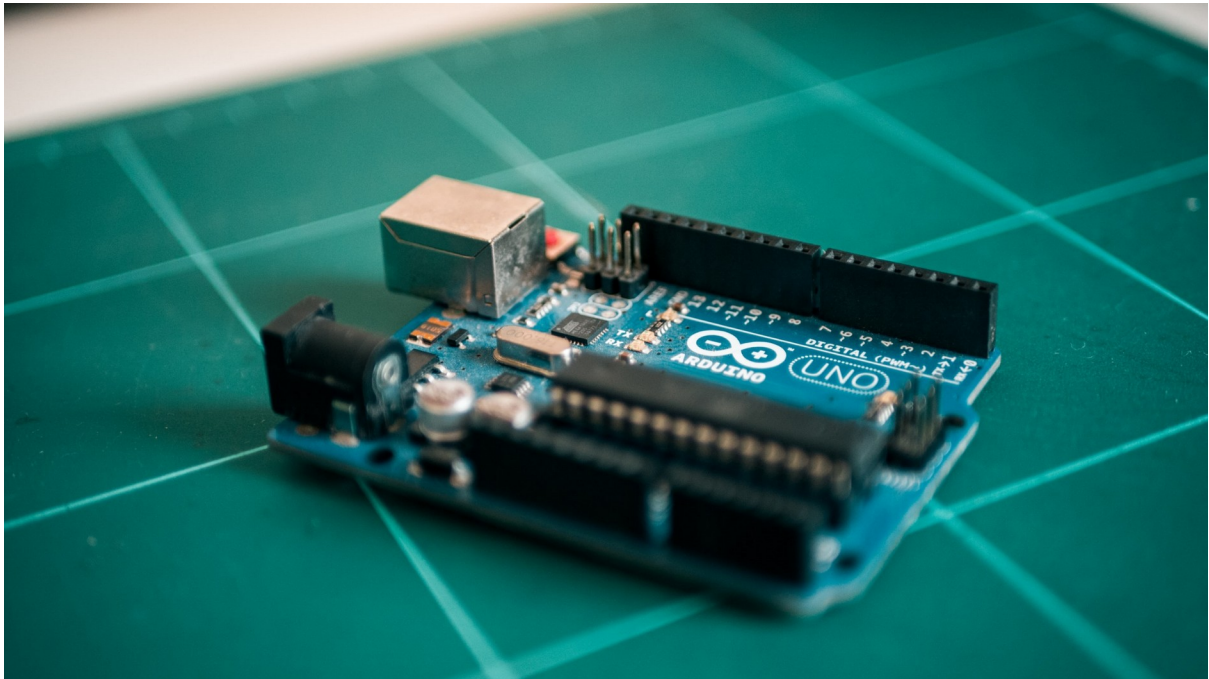


Domotech

PROYECTO CON ARDUINO



Juan Antonio Rojas Jimenez

Alex Merchán Sánchez

SMX2A

ÍNDICE

Introducción	3
1.1 ¿En qué consiste nuestro proyecto?	3
1.2 Elaboración del proyecto	4
1.3 ¿Cómo realizaremos el proyecto?	6
1.4 Objetivos del trabajo	7
2. Cuerpo del proyecto	8
2.1 Materiales	8
2.1.1 Primeros materiales pensados:	8
2.1.2 Materiales definitivos:	9
2.2 Arduino y sus módulos (Add-ons)	10
2.3 Conexiones del Arduino	14
2.4 Montaje	14
2.5 Fases de testeo	16
2.6 Montaje Final	17
2.7 Código de Arduino	18
3. Página Web	21
4. Aplicación	24
4.1 Estética de la Aplicación	24
4.2 Código de la Aplicación	26
4.2.1 Control Manual	26
4.2.2 Control Por Voz	29
Conclusion	32
Annexos	32
Webgrafia	33

1. Introducción

1.1 ¿En qué consiste nuestro proyecto?

El proyecto lo estuvimos pensando durante un tiempo sobre qué realizarlo, y surgieron estas ideas:

- Robot Inteligente
- Coche Sigue Líneas
- Alarma Domestica
- Casa Domòtica

Debatimos que el robot inteligente no lo veíamos muy versátil y mejorable a largo plazo, en el coche sigue líneas lo hacían demasiadas personas y queríamos hacer algo distinto a los demás, en el caso de la alarma doméstica nos gustaba, pero la veíamos muy poco trabajo para lo que era un proyecto y por último pensamos en que la casa domótica era perfecta, ya que se podía ampliar a largo plazo añadiendo módulos u órdenes.

El proyecto escogido es la construcción de una casa domotizada, la cual hemos desarrollado en tres partes:

Primero, hemos pensado en crear una maqueta de una casa simple con diversas habitaciones, y el techo descubierto para tener una vista amplia de lo que está sucediendo en el interior.

Segundo se compone de la domótica de la casa con el Arduino y sus respectivos módulos: DHT11 para visualizar la temperatura y humedad del aire, de tal forma que podamos climatizar nuestra casa, HC-05 es un módulo que nos permite conectarnos vía bluetooth con dispositivos remotos, en este caso nuestro teléfono móvil.

Y por último el sensor PIR para poder hacer una alarma con él. Con estas funciones que nos ofrecen los módulos hemos pensado en generar una app personalizada que nos dé la visualización de los datos y el control total de la casa de manera manual o por voz.

1.2 Elaboración del proyecto

Respecto a la construcción de la maqueta, en un principio pensamos en hacerla con cartón pluma, un material moldeable que se usa para maquetas, silicona que nos será útil para pegar las piezas necesarias y alambres enrollados dentro de las paredes para reforzar la estructura de la casa. Tras un tiempo hablando con nuestro profesor y planteando como poder hacerla más técnica y tener un control total de ella, empezamos hacer el prototipo de la maqueta con la página web “Tinkercad” para una vez realizada poder imprimirla con las impresoras 3D que dispone el centro. Esta forma tardará más en realizarse, ya que será momento de medir los módulos, LEDs, y como realizar las puertas, paredes, ventanas y encajes, con esto nos aseguramos que el resultado será más fiable, técnico y limpio que no realizada de forma manual.

Ya que el volumen de la maqueta (*Figura 1*) era muy grande para imprimirlo en la plataforma de la impresora 3D es 18x18cm y nuestra maqueta tiene unas dimensiones 32x24cm, en estas condiciones nos vimos obligados a dividir las en 4 módulos diferentes para poder imprimirla.

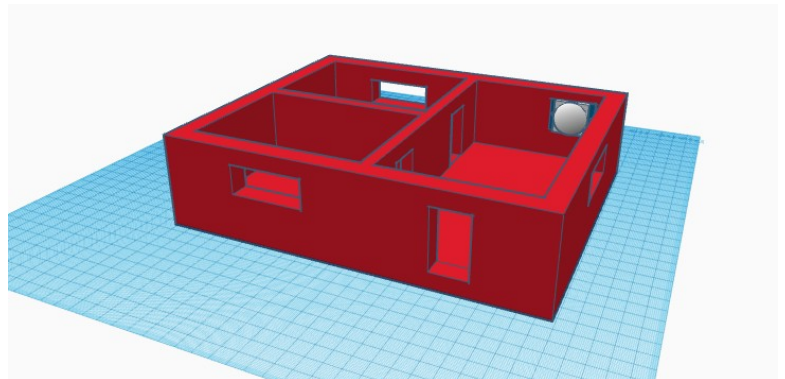


Figura 1 imagen maqueta

Gracias a este vimos que era una buena idea realizar las paredes huecas de tal forma que facilita el paso del cableado hacia los diferentes módulos y leds distribuidos por toda la casa, un doble suelo en el cual se va a esconder la placa de arduino y los módulos para hacerla de forma limpia y ordenada.

Antes de realizar el montaje del arduino, en la maqueta lo hicimos fuera de ella en una protoboard para poder ver las conexiones y cables, de esta forma nos permite modificar cambios o imprevistos que puedan surgir en el proceso de la construcción. Surgieron varios problemas con los sensores, en el caso del módulo bluetooth HC-05 no se conectaba al móvil y no enviaba datos correctamente, y en el sensor DHT11 no nos funcionaba el primero, ya que estaba fundido y no daba los valores correspondientes con la estación meteorológica que dispone el centro en sus instalaciones luego de cambiarlo y poder calibrar nos daba valores más aproximados. También se hicieron varias modificaciones en el código, puesto que nos daba errores y era muy extenso arreglando esas dos cosas y comentando que hace cada línea pudimos hacerlo más breve y sencillo de entender.

1.3¿Cómo realizaremos el proyecto?

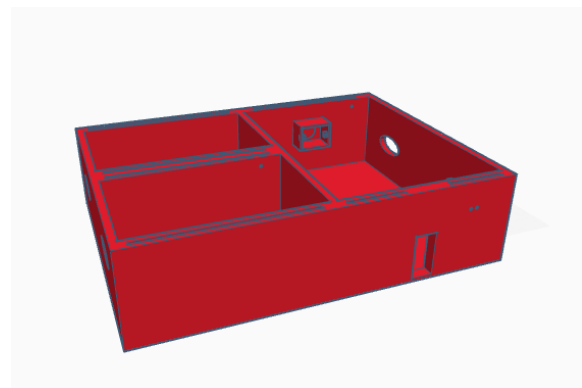
Para la realización del proyecto empezamos a ver los materiales necesarios que deberíamos usar para hacer una lista y pedirlos por alguna tienda que los proporcione en buen estado y envío seguro, porque los módulos son susceptibles a romperse. Vimos que disponíamos de varios de los materiales necesarios para la casa domótica, solo tuvimos que pedir el módulo HC-05, cableado extra y kit de leds y resistencias. Una vez que sabíamos que el material ya lo teníamos, nos centramos en el diseño de la maqueta en Tinkercad y la memoria para la explicación básica del proyecto al profesorado. El diseño nos llevó más tiempo de lo esperado, ya que hubo errores de enganches, medidas incorrectas. Una vez acabado el diseño lo enviamos a imprimir lo más rápido posible, por el hecho de que es un proceso en el que cada módulo tarda un aproximado de 24 horas en imprimirse y de esta manera ocupamos durante un tiempo prolongado la impresora 3D.

Antes de imprimir el prototipo de la maqueta a gran escala, se decidió hacer una impresión a pequeña escala (*Figura 2*) para comprobar si había un fallo en la impresión o diseño, la calidad de los enganches entre cada módulo y así poder visualizar un resultado aproximado de como quedaría en una versión a gran escala.



Figura 2 imagen Impresión del prototipo a pequeña escala

Al ver la impresión(*Figura 3*) nos dimos cuenta de que los enganches no eran óptimos y no funcionarán correctamente para unirse entre ellos, por lo que se descartaron, las ventanas se hicieron en forma ovalada para que a la hora de imprimirse no necesitará de soportes extras y se hizo un soporte para el sensor PIR de esta forma se mantiene estable en la maqueta.



1.4 Objetivos del trabajo

Los objetivos que queremos lograr con este proyecto no es solo sacar buena nota, sino poder aprender conceptos del proyecto como serían:

- Aprendizaje del funcionamiento de Arduino.
- La variedad de módulos que tiene Arduino y como conectarlos correctamente.
- Manejar y utilizar la programación de Arduino correctamente.
- El funcionamiento de programas externos para crear la maqueta en 3D.
- Cómo usar programas para la creación de aplicaciones.

También queremos que sea funcional y sencilla la app de tal forma que cualquier usuaria pueda usarla fácilmente, a la hora de querer ampliar el proyecto sea sencillo de modificar el código, funcionamiento de la casa domótica en la maqueta hecha a medida y sobre todo que sea algo que podamos usar en un futuro.

Todo esto fue realizado en días, pensando la idea, creando la maqueta, el código del proyecto, errores y pruebas de los módulos, etc.

Esto queda reflejado en el diagrama de Gantt que tenemos a continuación.

Actividad realizada	12/2/22	19/2/22	26/2/22	5/3/22	12/3/22	19/3/22	26/3/22	2/4/22	9/4/22	16/4/22	23/4/22	30/4/22	7/5/22	14/5/22	21/5/22	28/5/22
Busqueda de una Idea	■	■														
Diseño de la maqueta			■	■	■	■										
Código Arduino						■	■	■	■							
Pruebas de Arduino									■	■	■	■				
Creación de la página									■	■	■	■	■			
Documentación										■	■	■	■			
Montaje de la maqueta												■	■			
Manual de usuario													■	■		
Power Point															■	■
Ensayo de la Presentación																■

2. Cuerpo del proyecto

2.1 Materiales

2.1.1 Primeros materiales pensados:



En el primer contacto que tuvimos con la idea de proyecto definitivo, se nos ocurrió que la maqueta podría tener un buen resultado construyendo a mano con cartón pluma (Figura 4) (un material generalmente resistente para proyectos similares o en los que es necesario construir una estructura).

Figura 4 imagen Material Cartón pluma

Este cartón contaría con alambres (Figura 5) enredados en el interior del propio cartón pluma para reforzar aún más la estructura, ya que en el viaje del instituto a casa y a la inversa, la estructura sufre mucho.



Figura 5 imagen Material Alambres

Para unir todas las paredes y suelos tuvimos muchas ideas, como por ejemplo un superglue, silicona caliente, etc.(Figura 6) Finalmente vimos esta opción de maqueta no demasiado viable, ya que al ser una sola pieza en los viajes se acabaría cayendo alguna pared o rompiendo alguna parte fundamental.



Figura 6 imagen Material Silicona

2.1.2 Materiales definitivos:

Al final, la idea de la maqueta con el material de cartón pluma fue descartada por la dificultad y delicadeza del material para moldearlo a nuestro gusto, pero lo que realmente nos hizo cambiar la idea fue lo frágil que sería para transportarlo del instituto a casa y viceversa. Finalmente (por la recomendación de nuestro profesor Rubén Arroyo) decidimos crear la maqueta en 3D, e imprimirla con las impresoras en 3D que dispone el centro, con lo que la maqueta está hecha de un plástico duradero con el PLA.

Y ya que pensamos en poner el Arduino debajo de la maqueta, también necesitaremos hacer patas para elevar la maqueta, las cuales estarán hechas de madera y tendrán unos 3-4 cm de largo.

Lista de materiales:

Cableado para las conexiones	
Placa Arduino	
Sensor PIR	
Modulo bluetooth(HC-05)	
Módulo de temperatura y humedad (DHT11)	

Buzzer	
Pack de electronica básica para Arduino	

2.2 Arduino y sus módulos (Add-ons)

Placa Arduino:

Escogimos la opción de Arduino por el bajo coste que tiene y sus funciones a la hora de poder imaginar una variación de ideas y proyectos. Es una placa (Figura 7) sencilla el Arduino UNO R3 pero nos permite hacer nuestro proyecto sin ningún tipo de problema.

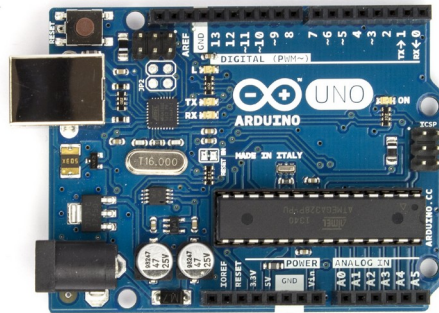


Figura 7 imagen placa

Las especificaciones técnicas del Arduino son las siguientes:

- Microcontrolador: ATmega328P.
- Velocidad de reloj: 16 MHz.
- Voltaje de trabajo: 5V.
- Voltaje de entrada: 7,5 a 12 voltios.
- Pinout: 14 pines digitales (6 PWM) y 6 pines analógicos.
- 1 puerto serie por hardware.
- Memoria: 32 KB Flash (0,5 para bootloader), 2KB RAM y 1KB Eeprom.

Módulo bluetooth:

Hemos decidido implementarlo al proyecto, de esta forma podemos hacer paralelamente una aplicación que nos proporcionará un uso más sencillo de la casa domótica por lo que sino tenemos el módulo bluetooth en este caso es el modelo HC-05 (Figura 8) no podríamos hacer la conexión con nuestro dispositivo móvil.



Figura 8 módulo de bluetooth HC-05

Las especificaciones técnicas de este módulo son las siguientes:

- Voltaje de operación: 3.6V - 6V DC.
- Consumo corriente: 50mA.
- Bluetooth: V2.0+EDR.
- Frecuencia: Banda ISM 2.4GHz.
- Modulación: GFSK(Gaussian Frequency Shift Keying).
- Potencia de transmisión: 4dBm, Clase 2.
- Sensibilidad: -84dBm a 0.1% BER.
- Alcance 10 metros.
- Interface comunicación: Serial TTL.
- Velocidad de transmisión: 1200 bps hasta 1.3Mbps.
- Baudrate por defecto: 38400,8,1,n.
- Seguridad: Autenticación y encriptación.
- Temperatura de trabajo: -20C a +75 C.
- Compatible con Android.
- Dimensiones: 37*16 mm.
- Peso: 3.6 gramos.

Sensor Pir:

Como en esta maqueta representa una casa domótica, se ha pensado que no podría faltar un sistema de seguridad en el cual incluye un sensor PIR, buzzer y LEDs nos centraremos en el módulo PIR (*Figura 9*). Para mayor rango del sensor estará ubicado en la zona del salón y la función principal que tiene es hacer sonar el buzzer y encender una luz específica cuando detecte movimiento así nos daremos cuenta de que hay un intruso dentro de la casa.



Figura 9 imagen módulo PIR

Las especificaciones técnicas del sensor PIR son las siguientes:

- Voltaje de alimentación: 5 a 12 VDC.
- Consumo promedio: 1 mA.
- Rango de distancia: 3 a 7 mts ajustable.
- Ángulo de detección: 110°.
- Tensión de salida en alto: 3.3 VDC.
- Dimensiones: 3.2 x 2.4 x 1.8 cm.

Sensor de temperatura y humedad:

De diferentes módulos que hay nos decidimos por el sensor DHT11 (*Figura 10*) por su funciones comparado con el precio que tiene, nos permite saber la humedad y temperatura en celsius y fahrenheit, con el podemos controlar ventiladores, humidificadores o calefacciones así nos permite hacer una casa más automática y acogedora, en nuestro proyecto hemos dejado esto aparte y nos dedicamos a crear una app donde te muestra los datos vía bluetooth en tiempo real.

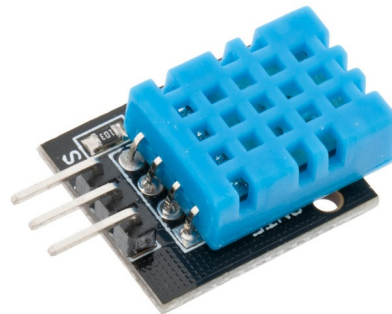


Figura 10 imagen DHT11

Las especificaciones técnicas del sensor DHT11 son las siguientes:

- Voltaje de Operación: 3V - 5V DC.
- Rango de medición de temperatura: 0 a 50 °C.
- Precisión de medición de temperatura: ± 2.0 °C.
- Resolución Temperatura: 0.1°C.
- Rango de medición de humedad: 20% a 90% RH.
- Precisión de medición de humedad: 5% RH.
- Resolución Humedad: 1% RH.

- Tiempo de sensación: 1 seg.
- Interface digital: Single-bus (bidireccional).
- Modelo: DHT11.
- Dimensiones: 16*12*5 mm.
- Peso: 1 gr.
- Carcasa de plástico celeste.

Buzzer:

El mini buzzer o buzzer (*Figura 11*) es un zumbador que nos ofrece sonido, hay dos modelos el pasivo y activo las diferencias son en que el activo solo se debe suministrar corriente al módulo y en el pasivo debe recibir una onda de frecuencia por eso nos decidimos por el pasivo para que sea más práctico.



Figura 11 imagen buzzer

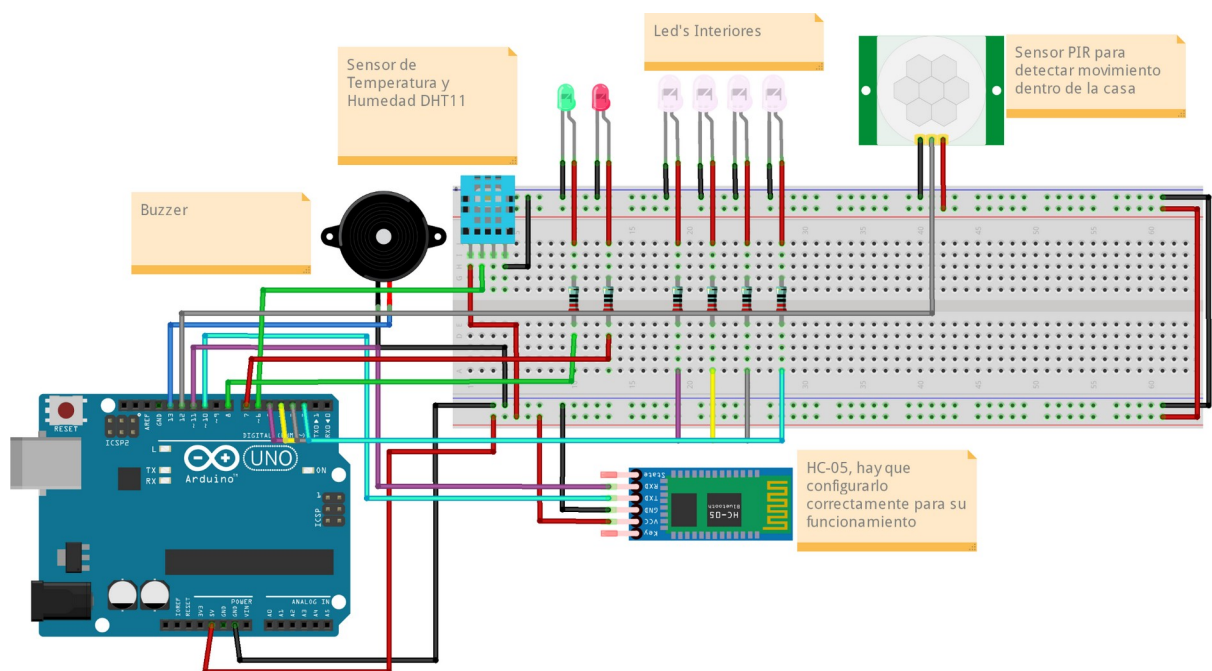
Para el proyecto era necesario implementarlo con la alarma de seguridad y que nos ofrezca un ruido que nos avise de que está sonando la alarma.

Las especificaciones técnicas del sensor Buzzer son las siguientes:

- Voltaje recomendado de funcionamiento: 3.3 a 5 V.
- Voltaje rango de operación: 3 a 7 V.
- Consumo de corriente: <30mA.
- Salida de sonido: >85 dB.
- Frecuencia generada: 2.3 KHz +- 400 Hz.
- Temperatura de Operación: -40° a 85° C.
- Diámetro de 1.2 cm.

- Peso: 2g.

2.3 Conexiones del Arduino



fritzing

Como podemos observar en la imagen son todas las conexiones necesarias para que la casa domótica funcione correctamente, está hecho con un programa llamado(Fritzing). Sirve de gran ayuda para realizar correctamente las conexiones a la hora de querer ampliar o variar algún módulo de sitio, nos fue útil para recordar el cableado.

2.4 Montaje

Primero de todo vamos a seleccionar los leds(Figura 12) que vamos a necesitar y las resistencias correspondientes (220Ω), luego colocamos estas resistencias entre el positivo del led y el cable que le dará la señal, y el negativo en el negativo de la placa base, luego elegimos los correspondientes pines del 2 al 5, para a la hora de enviar la señal y que el led se encienda.

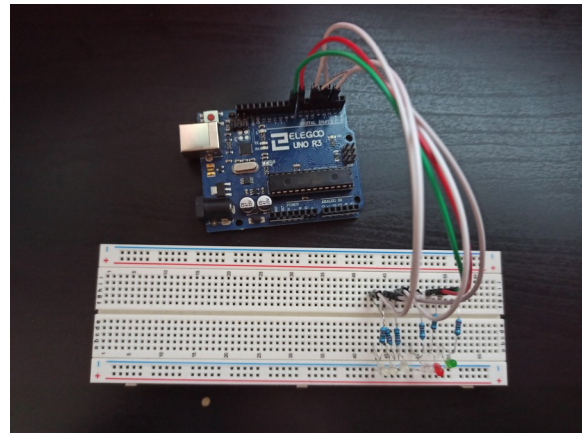


Figura 12 imagen LEDs

Luego conectamos el sensor de temperatura y humedad(DHT11)(Figura 13), en este no es necesario incorporar resistencias, y el cable de datos se conecta al correspondiente pin 6, una vez realizada las conexiones ya nos ofrece la información.

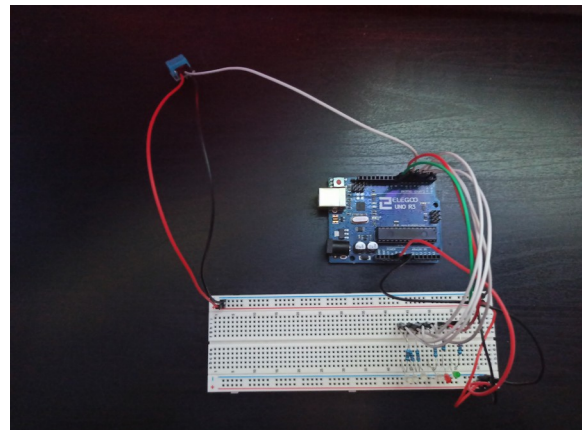


Figura 13 imagen DHT11

Para que el Arduino(Figura 14) se conecte a la aplicación, se utiliza el módulo bluetooth (HC-05) para conectarlo de manera adecuada, debemos cruzar el TX del módulo con el RX de la placa Arduino, y se hace el mismo proceso con el RX del módulo y el TX de la placa, así poder recibir y enviar datos. Los pines de datos son el RX del módulo al pin 11 de la placa y el TX al pin 10.

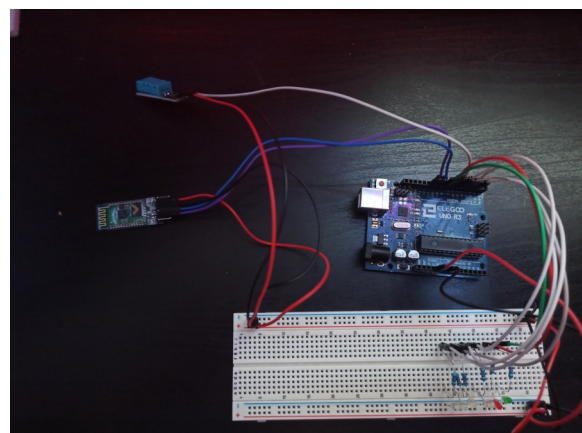


Figura 14 imagen HC-05

En el caso del sensor PIR las conexiones son sencillas(Figura 15), se conecta el negativo y positivo, en el negativo y positivo de la protoboard, para el cable de datos se conecta en el pin 12 de la placa base.

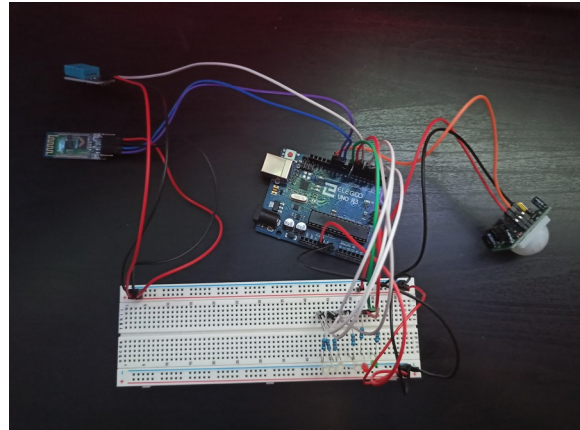


Figura 15 imagen PIR

Para que la alarma suene cuando detecte un movimiento(Figura 16), instalamos el buzzer, conectas el negativo en el negativo de la placa y el cable de datos en el pin 13 de la placa.

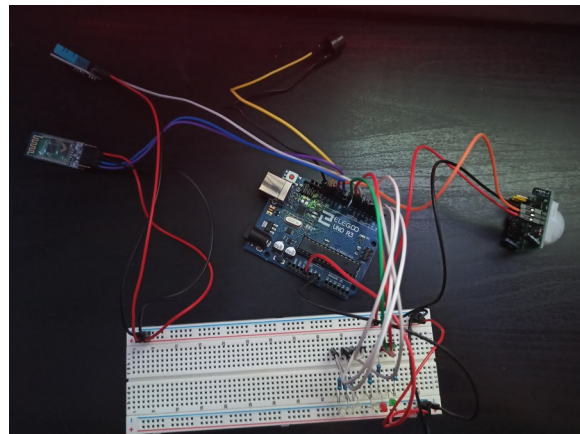


Figura 16 imagen Buzzer

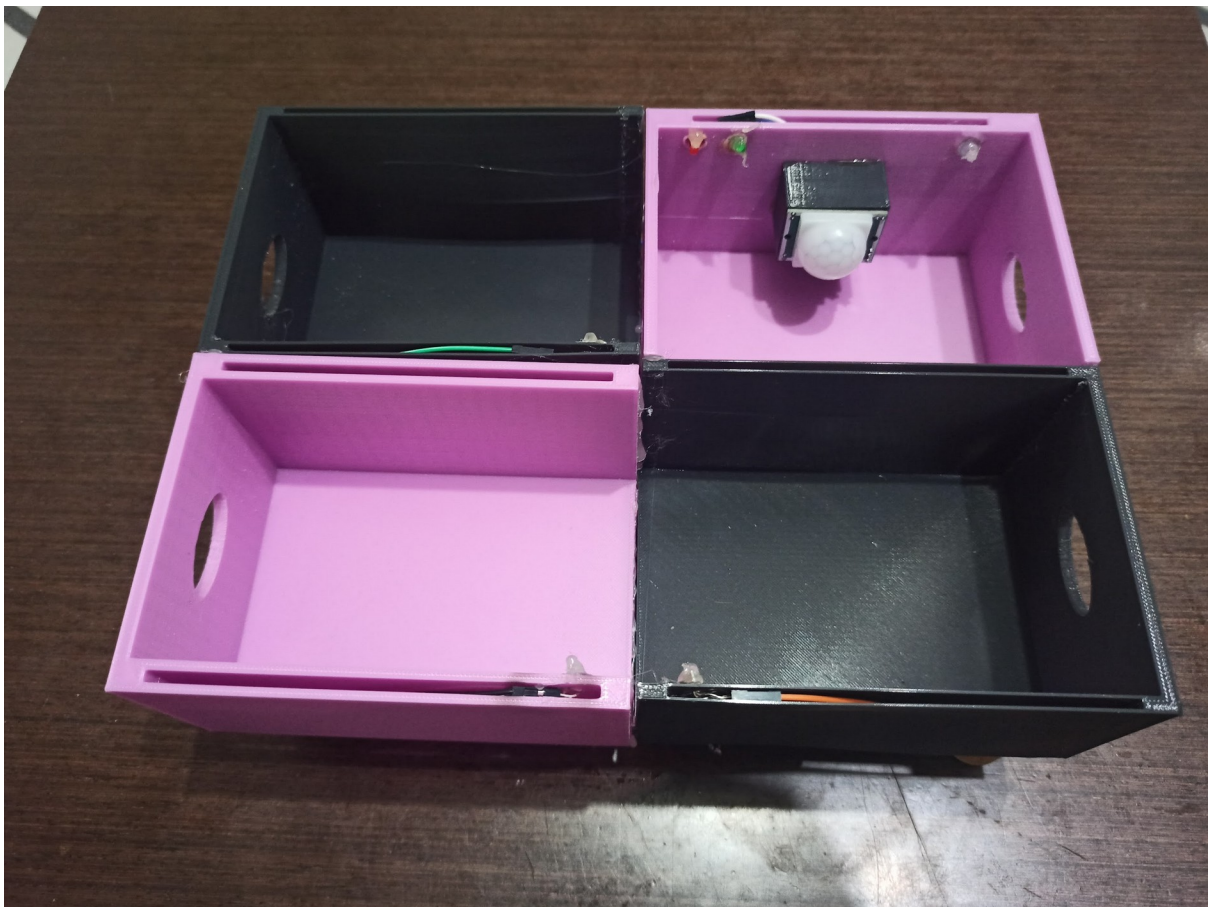
2.5 Fases de testeo

En las fases de testeo fuimos probando si los leds se encienden mediante la app, en los dos modos que tiene, y funcionaba perfectamente.

Luego comprobamos si enviaba los datos del sensor de temperatura y humedad(DHT11) a la app, y lo hacía sin ningún problema, pero debimos regular el sensor sumando o restando algún valor para que la medición fuera exacta.

Para la alarma tuvimos varios fallos, ya que nos daba errores de lectura y se activaba o mucho antes o no se activaba en ningún momento, por lo que en el módulo tiene dos potenciómetros que eran la regulación de la precisión del sensor, y modificando ese valor pudimos hacer que fuera más exacto.

2.6 Montaje Final



2.7 Código de Arduino

El código que hemos confeccionado se divide en varios apartados y comentado cada acción que hace, de esta manera se hace más sencillo la lectura y las modificaciones a largo plazo.

Se divide en los siguientes apartados:

Temperatura y humedad:

```
//SISTEMA DE TEMPERATURA Y HUMEDAD
celsius = dht.readTemperature(); //Lee y almacena la temperatura del sensor DHT11
fahrenheit = dht.readTemperature(true); //Lee y almacena la temperatura del sensor DHT11
humedad = dht.readHumidity(); //Lee y almacena la humedad del sensor DHT11

//Envía los datos del DHT11 como la temperatura, humedad e intensidad a la aplicación móvil por medio del Bluetooth
BT.print(celsius);
BT.print(" °C");
BT.print("|");
BT.print(fahrenheit);
BT.print(" °F");
BT.print("|");
BT.print(humedad);
BT.print(" %");
BT.print("|");
delay(1000);

//Se envían los datos del DHT11 al Monitor Serie
Serial.print(celsius);
Serial.print(" °C");
Serial.print("|");
Serial.print(fahrenheit);
Serial.print(" °F");
Serial.print("|");
Serial.print(humedad);
Serial.print(" %");
Serial.print("|");
delay(1000);
```

Para la temperatura y humedad hemos indicado las variables que necesitaremos para que en la app nos indique los datos de humedad, Celsius y Fahrenheit.

Sistema de iluminación:

```

//Sistema de IR
const int sensorPin = 12; //Sensor de IR conectado al Pin 12
const int LEDIV = 8; //Led Interior Verde
const int LEDIR = 7; //Led Interior Rojo

//Sistema de luces interiores
const int LEDG1 = 2; //Led H.G 1
const int LEDG2 = 3; //Led H.G 2
const int LEDP1 = 4; //Led H.P 1
const int LEDP2 = 5; //Led H.P 2

```

Iniciamos los pines de los LEDs de tal forma que cuando demos la señal se enciendan.

Sistema de alarma:

```

//SISTEMA DE ALARMA AL ENTRAR EN LA PUERTA
int value = 0; //Declaramos una variable llamada value donde guardara el valor de la luz leida por el sensor
value = digitalRead(sensorPin); //el valor va a ser leído por el sensor

digitalWrite(LEDIV,HIGH); //Led verde indica que no hay movimiento y todo esta seguro
digitalWrite(LEDIR,LOW); //Led Rojo indicara que hay movimiento inesperado

if (value == HIGH) //Aquí indicamos que si el valor esta encendido, en el Serial mostrara que no hay obstaculo y p
{
  Serial.println("Seguro: No hay movimiento");
  noTone(buzzer);
  delay(1000);
}
//En caso contrario
else
{
  digitalWrite(LEDIR,HIGH); //El Led Rojo se enciende, indicando que hay movimiento
  digitalWrite(LEDIV,LOW); //El Led Verde se apaga
  delay(1000);
  tone(buzzer,2500); //Suena un tono
  delay(1000); //Durante 150 ms
  tone(buzzer,1500); //Suena otro tono
  delay(1000); //Durante 150 ms
  noTone(buzzer);
  Serial.println("Advertencia: Detectó movimiento"); //Manda un mensaje al Serial donde indica que hay movimiento
}

```

En el apartado de alarma hemos indicado que el valor inicial es 0 y que al detectar movimiento con el sensor PIR cambie a 1, lo que esto provoca que se active la alarma, si no es el caso está desactivada.

Sistema de iluminación y apertura de puertas por voz:

```

//SISTEMA DE LUCES INTERIORES Y SISTEMA DE ABRIR-CERRAR PUERTAS POR SWITCH Y VOZ
if(BT.available())
{ //Si llega un dato por el puerto BT se envía al monitor Serial
  char Mensaje = BT.read(); //Guarda en la variable "Mensaje" el dato recibido por Bluetooth

  if(Mensaje == 'A')
  {
    Serial.println("Se ha encendido la habitacion Grande 1");
    digitalWrite(LEDG1,HIGH);
  }
  else if(Mensaje == 'a')
  {
    Serial.println("Se ha apagado la habitacion Grande 1");
    digitalWrite(LEDG1,LOW);
  }
  else if(Mensaje == 'B')
  {
    Serial.println("Se ha encendido la habitacion Grande 2");
    digitalWrite(LEDG2,HIGH);
  }
  else if(Mensaje == 'b')

```

Los datos se guardan en la variable mensaje y cuando en la app se manda un texto o un dato que sea igual a la letra correspondiente hace una acción, por ejemplo: encender habitación grande 1 y encender todo.

Librerías y pines especiales:

```

// LIBRERIAS
#include <SoftwareSerial.h> //Libreria Software Serial para Bluetooth
#include <DHT.h> //Librería paa usar sensores DHT
#include <DHT_U.h> //Complemento de la librería DHT

//PINES ESPECIALES
#define DHTPIN 6 //Se define el pin del DHT
#define DHTTYPE DHT11 //Se define el tipo de sensor DHT-11

//OBJETOS DE LAS LIBRERÍAS
DHT dht(DHTPIN, DHTTYPE); // Crea un objeto de la librería DHT, indicando así el sensor
SoftwareSerial BT(10,11); //Definir los pines RX y TX del Arduino conectados al Bluetooth

//OBJETOS
char Mensaje; //Variable que mandara los mensajes a los Switches y por voz

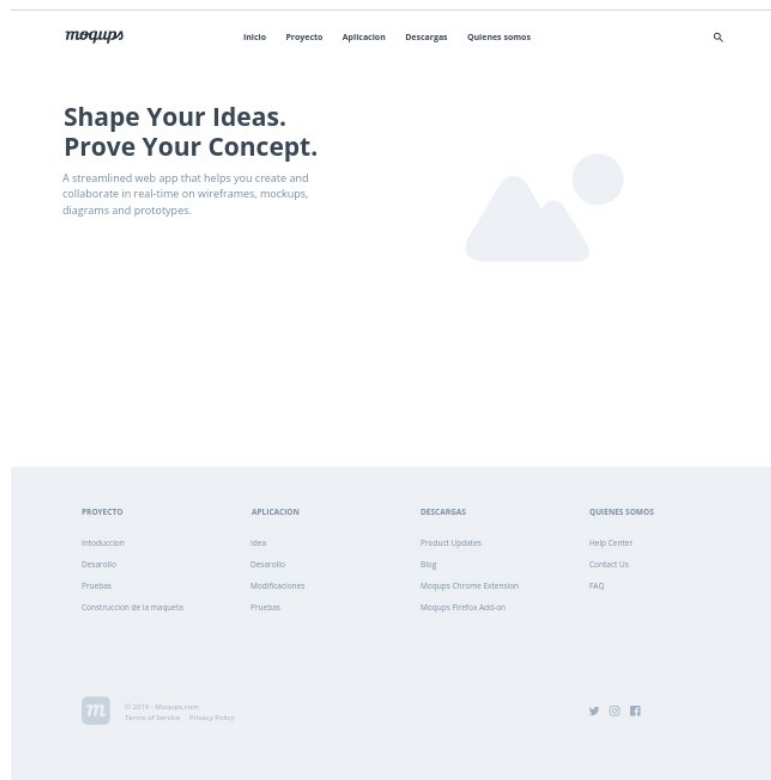
```

Como hemos modificado pines e insertado los módulos DHT11 y HC-05 se ha modificado los pines e indicado que sensor es para que el arduino lo reconozca correctamente.

3. Página Web

Para este proyecto de síntesis hemos decidido desarrollar una página en función a nuestro proyecto. En vez de empezar a desarrollar una página a lo loco sin una idea de cómo la queríamos, utilizamos la página web de Moqups recomendada por Rubén Arroyo (Figura 17).

Esta consta de un Inicio simple con imágenes de nuestro proyecto o relacionadas con Arduino las cuales al clicar en ellas, te redireccionará a su propio apartado dentro de la página, es decir, si clicas en la imagen del código accederás al apartado de código. En esta página hemos decidido que para la cantidad de apartados que teníamos sería una buena idea crear un menú con desplegables para cada grupo de apartados. Con esto, tendríamos los apartados de Inicio, Proyecto, Aplicación, Manual de usuario, Quienes somos y Descargas.



a 17 imagen Diseño

Fi
ur

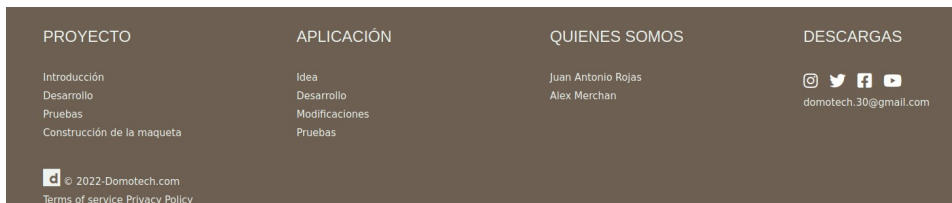


Figura 18 imagen Footer

Esta página también tendrá un footer(Figura 18) extenso con accesos directos a todos los apartados de la página de la misma forma que el menú del header.

El footer también está compuesto por una marca de copyright (simplemente para dar un toque estético y profesional) (Figura 19), además de un correo electrónico de contacto para el usuario y enlaces directos a las redes sociales con el respectivo logotipo de la red social.



Figura 19 imagen Redes

En cualquier apartado de nuestra página, tanto el header como el footer será visible, ya que se mantiene en todo momento. En los propios apartados, se mantendrán los mismos diseños, es decir, una columna para texto y otra para imagen, o alternando entre estas.



Cómo nombrado anteriormente, Domotech tiene sus propias redes sociales en las cuales se subirá contenido relacionado a fotos del proyecto.

Instagram: https://www.instagram.com/domotech_oficial/

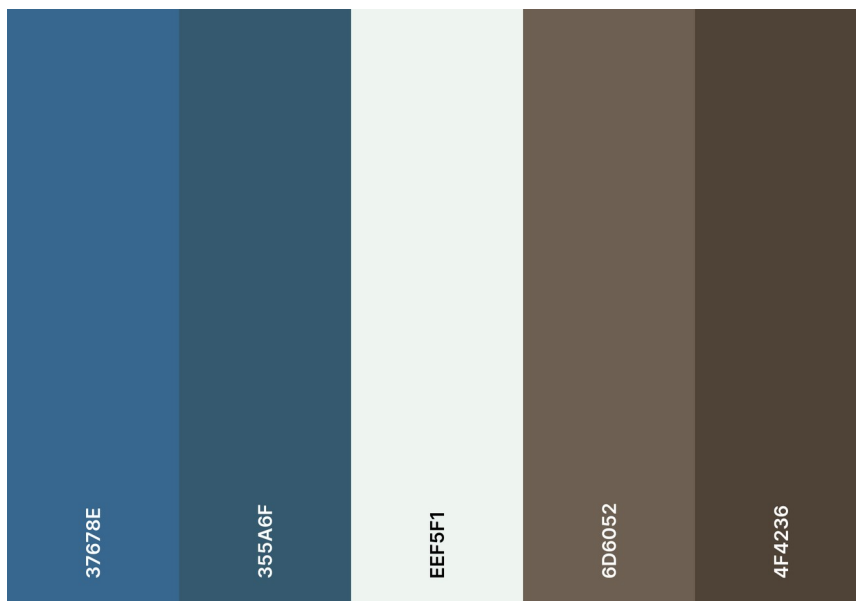
Twitter: <https://twitter.com/domotech30> ,

Facebook: <https://www.facebook.com/Domotech-101981325843660>

Youtube: <https://www.youtube.com/channel/UC5atQyytu92YsjU24LoLfgA>



Durante el diseño de nuestra página en Moqups nos pusimos de acuerdo para darle un detalle y desarrollar la página con aspectos que estén lo más relacionado posible con Arduino, por lo que en el caso de los colores utilizados en la página, fueron mayormente los colores de la placa Arduino Uno extraídos gracias a la web Colors:



Paleta de Colores

4. Aplicación

4.1 Estética de la Aplicación

Ya que nuestro proyecto se basa en la automatización de una casa convirtiéndola en una casa domótica, hemos pensado que una aplicación básica que nos permita controlar las funciones básicas que tiene el Arduino de forma remota con el módulo bluetooth(HC-05) para que el rango de acción sea reducido y no puedan conectarse otras personas y provocar fallos en él, o encender y/o apagar la alarma y LEDs.

Para la creación de nuestra app pensamos en varios como Android Studio, MobinCube, App Inventor y Score Apps. Pero ya que no teníamos mucha idea de cómo funcionaba una app, elegimos un programa de drag and drop por bloques, nos quedamos con App Inventor, por la experiencia que tenemos con él.

La aplicación tiene un diseño simple con botones, texto e imágenes para que sea intuitiva.

Como podemos ver en la app(Figura 20) nos muestra como se llama nuestro proyecto, en lo que va a mostrar la app que es el control de la Domotech(Casa Domótica) dos botones con los que podemos acceder a los dos controles que ofrece la app que son el control manual y por voz.

Toda la temática de la app fue combinada con los colores que tiene el Arduino para hacerla estéticamente similar y sencilla.

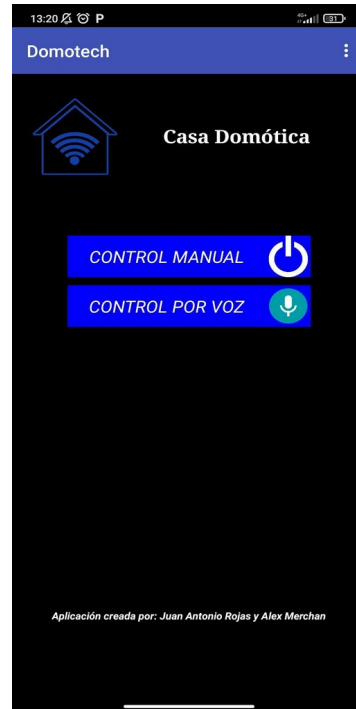


Figura 20 imagen App Front

Cuando presionamos el botón de control manual en la pantalla de inicio nos lleva a este apartado de la app(Figura 21) donde podemos visualizar que sigue la misma temática que la pantalla de inicio, pero incluyendo los colores verde y rojo para indicar si está conectado o no.

En el primer apartado vemos la conexión de la app con el bluetooth de la placa, nos saldrá un desplegable y lo conectamos. Seguimos con la posibilidad de activar y desactivar la alarma, control de las luces de la casa y unos displays de la temperatura y humedad que hay en la casa gracias al sensor instalado DHT11.



Figura 21 imagen App Front

En el apartado de voz(Figura 22) hicimos una variación sobre el control de voz, de tal forma que incluimos el botón de conectarse a bluetooth con el símbolo igual que con el micrófono, de esta forma es más intuitivo y vistoso. También tenemos el botón de comandos que te permite ver todos los disponibles para controlar la casa.

Luego pensamos en cómo hacer para que supiéramos que reconoce el micrófono para mandar las órdenes e insertamos una zona de texto en la que te muestra lo que ha reconocido.

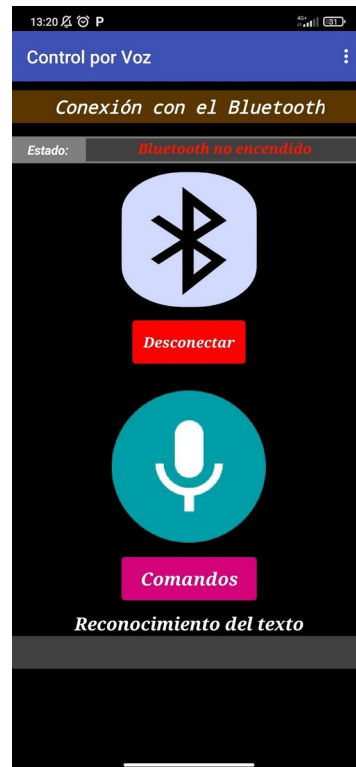


Figura 22 imagen App Front

4.2 Código de la Aplicación

4.2.1 Control Manual

Para el código de la aplicación estuvimos investigando como poder conectar el módulo bluetooth(HC-05) con el teléfono móvil que de tal forma pudiera mandar datos entre ellos, para que de esta manera se pudiera visualizar las temperaturas del sensor de temperatura y humedad(DHT11) y mandar las órdenes que insertamos a la placa Arduino.

Para hacer todo este proceso nos tuvimos que inspirar en varios proyectos similares al nuestro, a causa que desde un principio nuestro módulo bluetooth(HC-05) no se conectaba correctamente con la app y tuvimos que resetear los valores por defecto de fábrica, porque producía errores. Una vez hecho eso le indicamos en qué frecuencia trabajaría para que se conectara al Arduino y a la aplicación del teléfono móvil sin ningún tipo de problema.

A continuación vamos a hacer una breve explicación de los bloques que hemos introducido para hacer que funcione todo correctamente:

En los bloques(Figura 23) que vemos se usan en la pantalla de inicio y nos permite ir a las pantallas de la app correspondiente cuando presionamos los botones.



Figura 23 imagen App Back-End

Cuando entramos en la pantalla 2(Figura 24) si el Bluetooth no está encendido muestra el mensaje “Bluetooth no encendido” de color rojo, también sucede el caso con el estado de la alarma “Encendida” y en rojo. Pone todos los switches en falso para que a la hora establecerse la conexión no de falsos datos al módulo bluetooth(HC-05).

Si no es el caso ponemos el aviso de “No conectado” e igualmente ponemos todos los switches en falso, pero los que nos va a ofrecer los datos del sensor DHT11 lo dejamos que es texto para que más adelante se pueda escribir en él.

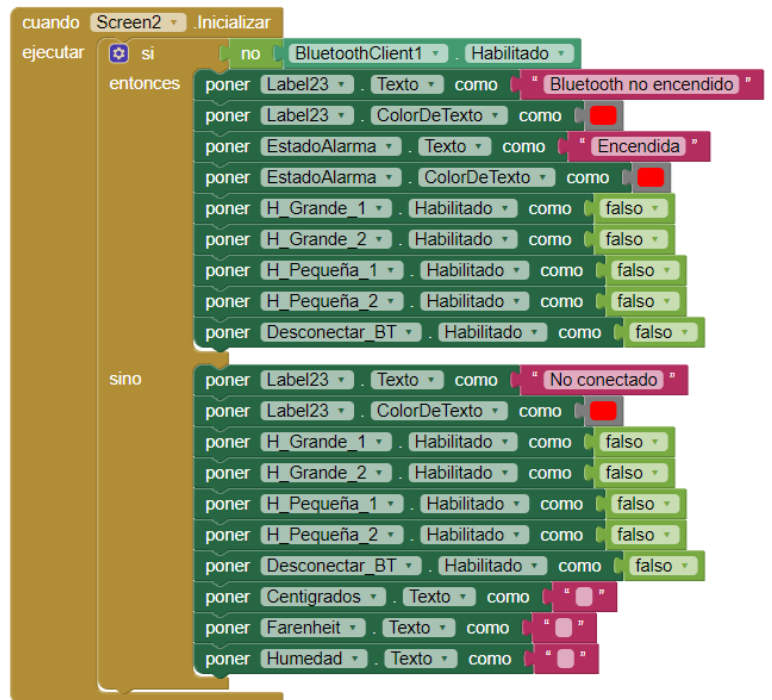


Figura 24 imagen App Back-End

Cuando le damos al botón de conectar_BT cambiamos la variable de BT cliente a “DirreccionesYNombres” para que se pueda hacer las comunicaciones correctamente entre app y módulo bluetooth(Figura 25).

Una vez hecho esto, si el cliente se conecta correctamente te muestra el texto de “Conectado” y el de la alarma “Encendida” en verde, habilita todos los switches para que pueda mandar los datos y notifica de que la conexión se ha establecido con una ventana emergente.

Si no es el caso, pone “Error de Conexión” en rojo.

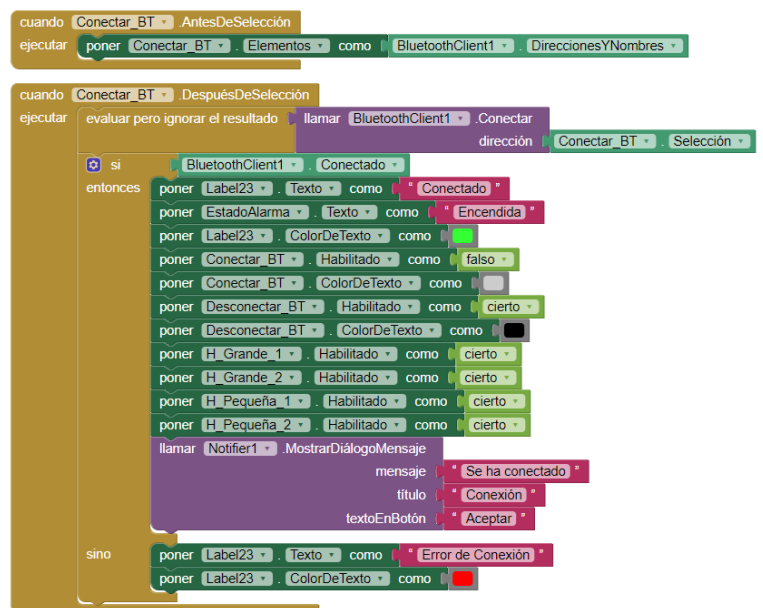


Figura 25 imagen App Back-End

Inicia(Figura 26) una global de datos “datos_entradaBT” luego crea una lista de datos vacía con el nombre de “lista_datos”.

Una vez hecho eso se activa el módulo que integramos de reloj que ve cuando está activado y recibe los datos del sensor por orden, de forma que no habrá problemas de conexión o datos.

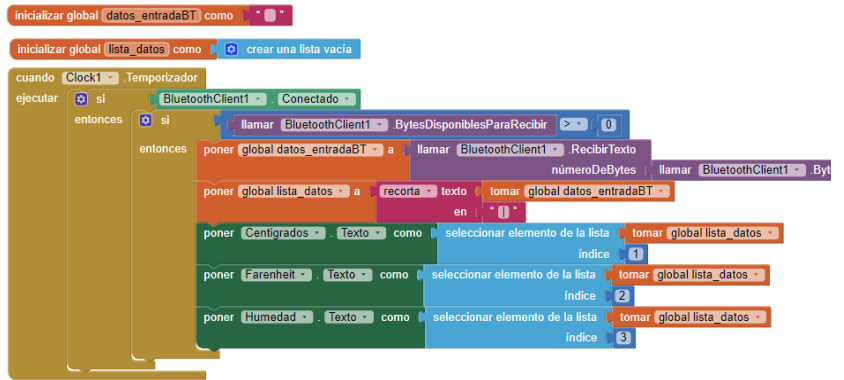


Figura 26 imagen App Back-End

Al presionar el botón de desconectar del bluetooth, llama al módulo BluetoothClient y lo desconecta(Figura 27), envía una notificación en forma de ventana emergente en el que nos muestra “Se ha desconectado”. Vuelve a poner en la zona “Desconectado”, habilita de nuevo el botón de conectarse y deshabilita el de desconexión.

Pone todos los datos que ha enviado del módulo DHT11 a cero y deshabilita todos los switches para que no envíen más datos y no se envían datos erróneos al conectarse de nuevo.

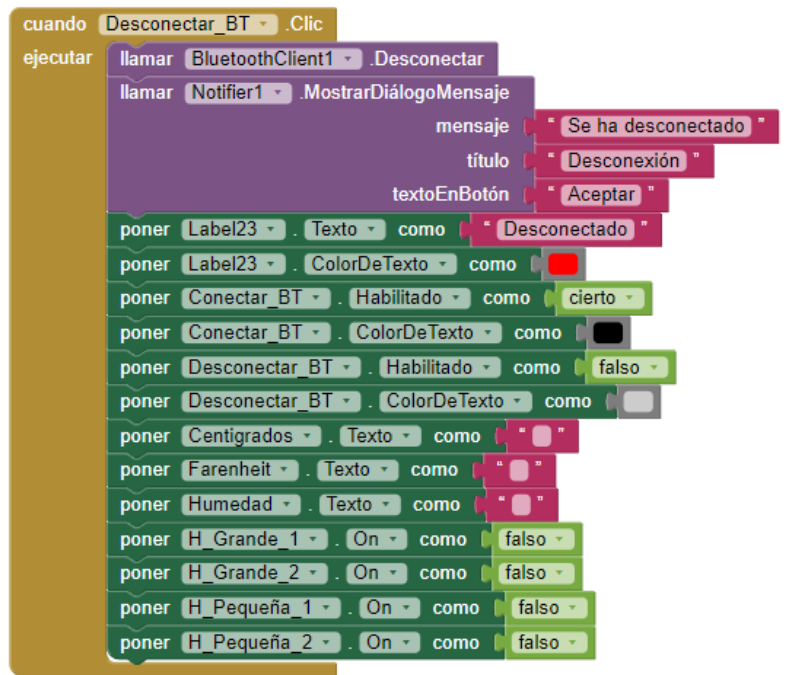


Figura 27 imagen App Back-End

Aquí indicamos(Figura 28) que cuando le demos clic al botón de desactivar alarma, habilite la opción y envíe un dato por el bluetoothClient para que la Arduino pueda descifrarlo y comparar con lo que había puesto en su código. Y muestra que está apagada.

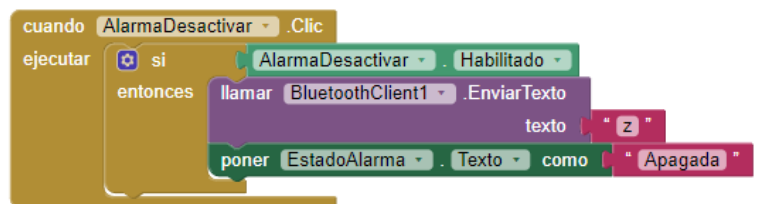


Figura 28 imagen App Back-End

Si el switch(Figura 29) está encendido y cambia, entonces envía un dato que es en este caso la letra A sino es la letra a que es minúscula. Todo esto ya que al enviarse el código lo compara con las letras que tiene y si coinciden hacen la acción indicada.

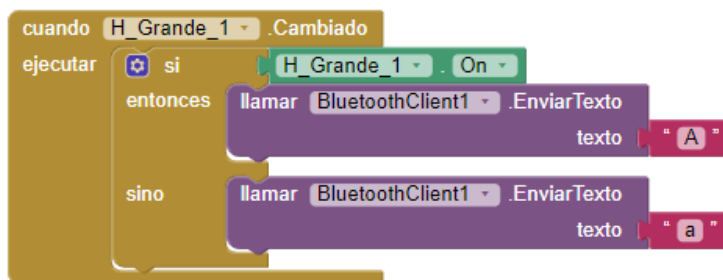


Figura 29 imagen App Back-End

4.2.2 Control Por Voz

Para el control por voz vimos en app inventor que había un módulo propio que te permite usar el micrófono que tiene integrado el teléfono móvil para reconocer la voz y poderla enviar o mostrar. Para nosotros esto fue una gran ventaja, porque se nos facilitaba la opción de hacer el sistema de comunicación entre la app y el Arduino.

Cuando entramos(Figura 30) en la pantalla de control de voz se ejecuta una acción que si el bluetooth cliente no está habilitado, pone el BT_Micrófono como falso, muestra en rojo el mensaje de "Bluetooth no encendido" sino es el caso, muestra el mensaje "No conectado".



Figura 30 imagen App Back-End

Antes de darle(Figura 31) al botón de Conectar_Bluetooth pone los element como "DirreccionesYNombres"

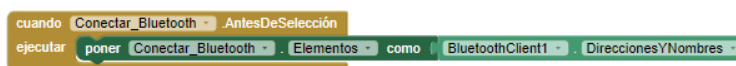


Figura 31 imagen App Back-End

Una vez presionamos el botón(Figura 32) para conectarse al bluetooth nos saldrá todos los dispositivos bluetooth que se ha emparejado alguna vez, cuando se ha conectado con el módulo nos muestra el mensaje de esta “Conectado” en verde y nos muestra una ventana emergente “Se ha conectado” sino es el caso nos indica que ha habido un “Error de Conexión”

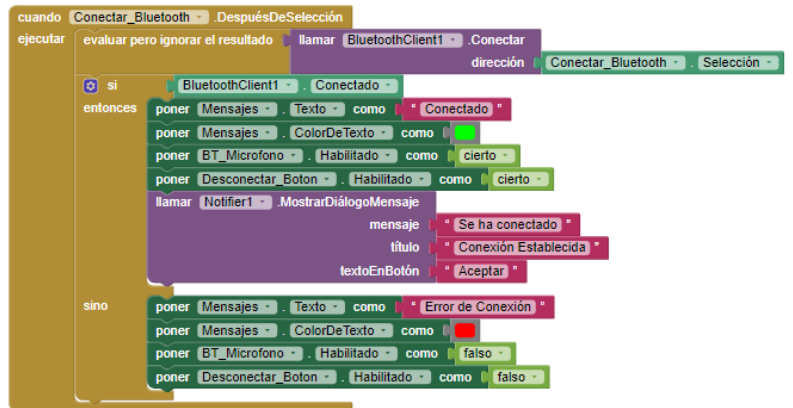


Figura 32 imagen App Back-End

Al hacer clic al botón(Figura 33) de Desconectar_Boton el bluetooth se desconecta, nos muestra la ventana emergente con el mensaje “Se ha desconectado”. Nos muestra en la ventana que está “Desconectado” el micrófono se deshabilita y vacía lo que nos muestra el reconocimiento de voz.



Figura 33 imagen App Back-End

Al presionar(Figura 34) vemos una ventana emergente con los comandos que podemos decir para que la casa domótica haga acciones.

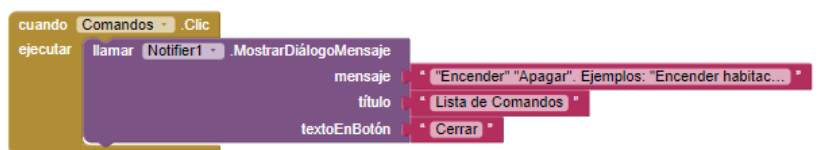


Figura 34 imagen App Back-End

Al clicar el botón(Figura 35) del Micrófono se llama al módulo de reconocimiento de voz para obtener el texto que vamos a mandar y mostrar.

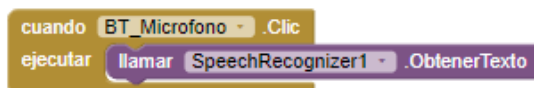


Figura 35 imagen App Back-End

Cuando(Figura 36) el reconocimiento de voz el resultado es igual a los que hemos declarado, envía un dato.

Ejemplo práctico: Si decimos encender habitación grande 1 envía un dato que es una A.

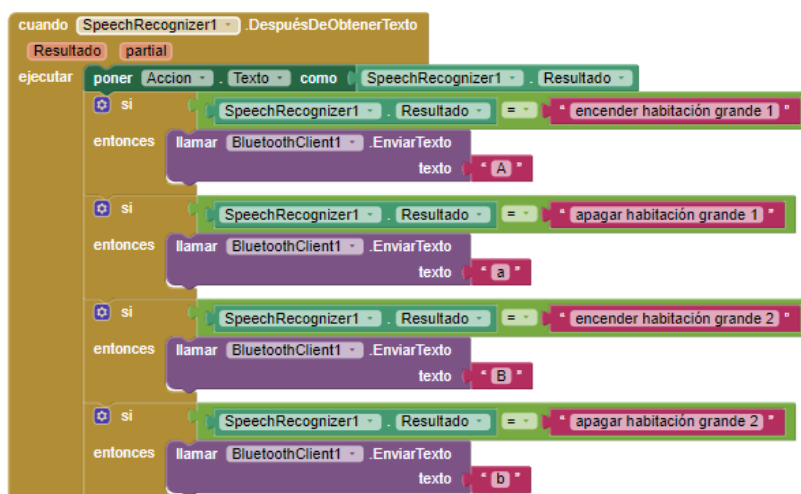


Figura 36 imagen App Back-End

Conclusion

La conclusión que hemos llegado al realizar el proyecto es que hacer algo con Arduino tiene infinitas posibilidades, a causa de ello hemos aprendido código, como funcionan los módulos del Arduino y sus diferencias. Al efectuar las conexiones de

los módulos y leds aprendimos sobre electrónica. También realizamos una app, y aprendimos a cómo diseñar la estética y la estructura del código para que sea ordenado.

Annexos

Página Web: [Domotech \(juananj.github.io\)](https://juananj.github.io)

Video de demostración: <https://youtu.be/xmqcKMASRzA>

Link de Fritzing: [Fritzing](#)

Link de App Inventor: [MIT App Inventor | Explore MIT App Inventor](#)

Link de Arduino IDE: [Software | Arduino](#)

Link de Moqups: [Herramienta de Maquetas, Esquemas & Prototipos UI En Línea · Moqups](#)

Link de Tinkercad: [Tinkercad | Crea diseños digitales 3D con CAD en línea | Tinkercad](#)

Webgrafia

Dynamo Electronics. (2021).“Sensor infrarrojo de movimiento PIR HC-SR501”. Recuperado de, [Sensor infrarrojo de movimiento PIR HC-SR501 - DynamoElectronics](#)

Naylamp Mechatronics. (2021).“MÓDULO BLUETOOTH HC-05”. Recuperado de [Módulo Bluetooth HC05 \(naylampmechatronics.com\)](#)

TodoMicro.(2017).“*Sensor de temperatura y humedad Dht11 Arduino*”. Recuperado de [Sensor De Temperatura Y Humedad Dht11 Arduino \(todomicro.com.ar\)](http://todomicro.com.ar)

Llamas, L. (2022). “*Detector de movimiento con Arduino y sensor PIR*”. Recuperado de [Detector de movimiento con Arduino y sensor PIR \(luisllamas.es\)](http://luisllamas.es)

TUTORIALES HETPRO.(2021).“*Bluetooth HC-06 y HC-05 Android Arduino*”. Recuperado de [Bluetooth HC-06 y HC-05 Android - Arduino TUTORIALES HETPRO \(hetpro-store.com\)](http://hetpro-store.com)

Naylamp Mechatronics. (2021). “*Configuración del módulo bluetooth HC-05 usando comandos AT*”. Recuperado de [Configuración del módulo bluetooth HC-05 usando comandos AT \(naylampmechatronics.com\)](http://naylampmechatronics.com)

Cómo conectar la app con el Arduino por el módulo HC-05. Red Arduino. (2021, noviembre 24).*Casa inteligente con Arduino UNO y visual studio*. [Vídeo]. Recuperado de [\(37\) CASA INTELIGENTE CON ARDUINO UNO Y VISUAL STUDIO - YouTube](https://www.youtube.com/watch?v=37CASAINTELIGENTECONARDUINOUNOYVISUALSTUDIO)

Esquivel, B.(2021). “*Proyecto Final: Casa Domótica*”. Recuperado de [Proyecto Final: Casa Domótica \(brandon22esquivel.wixsite.com\)](http://brandon22esquivel.wixsite.com)