



UNIVERSIDADE
LUSÓFONA

Projeto Smart Home

Grupo18

André Jesus, a22207061

Tomás Nave, a22208623

Arquitetura Avançada de Computadores | LEI | 03/01/2024

www.ulusofona.pt

Introdução

O presente relatório descreve o projeto de criação de uma simulação de uma **smart home** utilizando placas Arduino e o programa Tinkercad. O objetivo é criar uma experiência interativa onde o utilizador pode interagir com diferentes divisões da casa, simulando diversas funcionalidades através de programação em linguagem C.

Link Do Projeto no tinkercad: https://www.tinkercad.com/things/4xIDdezuzLP-copy-ofprojeto-final/editel?returnTo=%2Fdashboard%3Ftype%3Dcircuits%26collection%3Ddesigns&sharecode=WYkaswyWTv2jRkZzXbOM3THlfjDAMvIxZp1_QVAnIDw

Especificação Técnica

O projeto será composto por cinco placas Arduino, cada uma representando uma divisão específica da casa: Porta, Master (Corredor), Quarto, Casa de Banho e Sala. Cada Arduino terá componentes e sensores específicos para simular as características de cada divisão.

Falando agora mais sobre cada componente presente em cada Arduino:

- **Arduino (Porta):**

- Servo controlado por Keypad para simular a abertura da porta com um código.
- Alarme sonoro ativado após três tentativas de código da porta incorreto (Piezo).

Resumo:

Este Arduino vai ter um servo que servirá para simular uma porta que vai ser aberta através da colocação de um código num Keypad. Se o código for colocado 3 vezes e estiver incorreto, ou seja, não coincidir com o código original irá ser acionado um alarme de som na porta.

Código:

```
#include <Servo.h>
#include <Keypad.h>
```

```
Servo myServo;
```

```
const int servoPin = 13; // Pino de controle do servo
const int piezoPin = 10; // Pino do piezo
```

```
const char* codigoCorreto = "1234"; // Código de 4 dígitos
char codeEntered[5];
int tentativasIncorretas = 0;
```

```
const byte LINHAS    = 4; // quatro linhas
const byte COLUNAS = 4; // quatro colunas
```

```
char keys[LINHAS][COLUNAS] = {
  {'1','2','3','A'},
  {'4','5','6','B'},
  {'7','8','9','C'},
  {'*','0','#','D'}
};
```

```
byte linhasPins[LINHAS] = {9, 8, 7, 6}; // Conectar os pinos de linha
ao pino digital
byte colunasPins[COLUNAS] = {5, 4, 3, 2}; // Conectar os pinos de coluna
ao pino digital
```

```
Keypad keypad = Keypad(makeKeymap(keys), linhasPins, colunasPins, LINHAS,
COLUNAS);
```

```
void setup() {
  myServo.attach(servoPin);
  myServo.write(0);
  pinMode(piezoPin, OUTPUT);
  Serial.begin(9600);
  Serial.println("Inicializacao completa");
}
```

```
void loop() {
  char key = keypad.getKey();
  if (key && strlen(codeEntered) < 4) {
    Serial.print("Tecla pressionada: ");
    Serial.println(key);
    codeEntered[strlen(codeEntered)] = key;
    // Imprime o código atual
    Serial.print("Codigo atual: ");
    Serial.println(codeEntered);
    delay(50); // Aguarda para evitar leituras repetidas
  }
  if (strlen(codeEntered) == 4) {
    codeEntered[4] = '\0'; // Adiciona um caractere nulo ao final da
string
    Serial.println("Codigo inserido completo: ");
    Serial.println(codeEntered);
    if (strcmp(codeEntered, codigoCorreto) == 0) {
      Serial.println("Codigo correto. Abrindo a porta.");
      abrirPorta(); // Função para abrir o servo
      // Zera a contagem de tentativas após inserir o código correto
      tentativasIncorretas = 0;
    } else {
      Serial.println("Codigo incorreto.");
      // Incrementa a contagem de tentativas
      tentativasIncorretas++;
      if (tentativasIncorretas == 3) {
        Serial.println("Tres tentativas incorretas. Acionando o
alarme.");
        acionarAlarme(); // Função para acionar o alarme
      }
    }
  }
}
```

```

        tentativasIncorretas = 0;
    }
}
for (int i = 0; i < 5; i++) {
    codeEntered[i] = '\0';
}
}
}

void abrirPorta() {
    Serial.println("Abrindo a porta...");
    myServo.write(90); // Gira o servo para 90 graus
    delay(500);
    myServo.write(0); // Gira o servo de volta para 0 graus
    Serial.println("Porta fechada.");
}

void acionarAlarme() {
    Serial.println("Acionando o alarme!");
    tone(piezoPin, 1500);
    delay(500); // Aguarda 500ms antes de retornar
    noTone(piezoPin); // Desliga o som
}
}

```

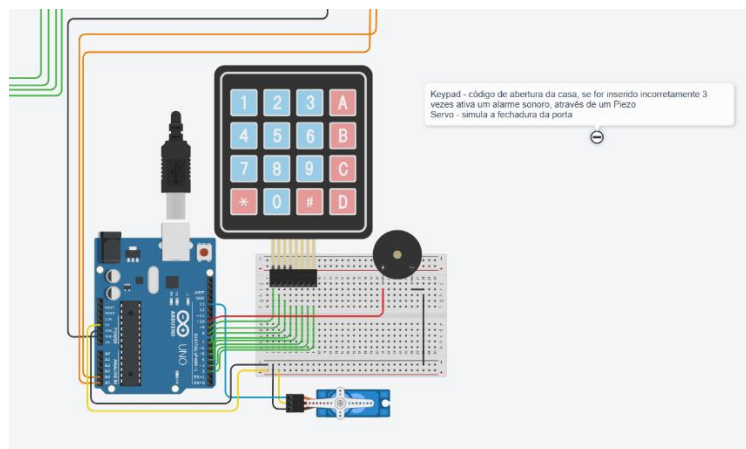


Figura 1- Circuito da Porta

- **Arduíno (Master):**
 - Conexão com todas as outras divisões através de pipes.
 - Capacidade de receber e transmitir comandos para outras divisões.

Resumo:

Este Arduino vai funcionar como se fosse o corredor, ou seja, vai ser a conexão com todas as outras divisões da casa, isto é, todos os outros Arduínos. Estas conexões vão ser feitas através de pipes, estes pipes vão ter a função de fazer a comunicação de informação entre o corredor e as outras divisões todas, como se fosse a simulação de uma pessoa entrar ou sair de cada divisão.

O utilizador ao estar no master pode escrever no terminal "desligar luzes casa". Se isso acontecer todas as luzes de cada divisão serão desligadas.

Pode também escrever no terminal "temperatura sala" que irá dizer no terminal, que simula uma resposta por voz, a temperatura atual da sala, o mesmo acontece se escrever "temperatura quarto". Por fim se escrever no terminal "ligar ar condicionado sala" e "ligar ar condicionado quarto" irá ligar o motor do ar condicionado na respetiva divisão.

Código:

```
#include <Wire.h>
#define SalaAddress 0x08
#define QuartoAddress 0x09
#define CasaDeBanhoAddress 0x10
#define CozinhaAddress 0x11

float temperaturaSala;
float temperaturaQuarto;

void setup() {
  Serial.begin(9600);
  Wire.begin();
}

void loop() {
  if (Serial.available() > 0) {
    String input = Serial.readStringUntil('\n');
    if (input.equals("temperatura sala")) {
      Wire.requestFrom(SalaAddress, sizeof(temperaturaSala));
      while (Wire.available() < sizeof(temperaturaSala));
      Wire.readBytes((byte*)&temperaturaSala, sizeof(temperaturaSala));
      Serial.print("Temperatura da Sala: ");
      Serial.print(temperaturaSala);
      Serial.println(" C");
    } else if (input.equals("temperatura quarto")) {
      Wire.requestFrom(QuartoAddress, sizeof(temperaturaQuarto));
      while (Wire.available() < sizeof(temperaturaQuarto));
      Wire.readBytes((byte*)&temperaturaQuarto,
sizeof(temperaturaQuarto));
      Serial.print("Temperatura do Quarto: ");
      Serial.print(temperaturaQuarto);
      Serial.println(" C");
    } else if (input.equals("desligar luzes casa")) {
      Serial.println("A desligar todas as luzes");
    }
  }
}
```

```

        desligarLuzesCasa();
    }else if (input.equals("ligar ar condicionado sala")) {
        Serial.println(" A ligar ar condicionado sala");
        ligarACSala();
    }else if (input.equals("ligar ar condicionado quarto")) {
        Serial.println("A ligar ar condicionado quarto");
        ligarACquarto();
    }
}

void desligarLuzesCasa() {
    // Enviar comando para desligar luzes para a Sala
    Wire.beginTransaction(SalaAddress);
    Wire.write('s');
    Wire.endTransmission();
    // Enviar comando para desligar luzes para o Quarto
    Wire.beginTransaction(QuartoAddress);
    Wire.write('q');
    Wire.endTransmission();
    // Enviar comando para desligar luzes para a Cozinha
    Wire.beginTransaction(CozinhaAddress);
    Wire.write('c');
    Wire.endTransmission();
    // Enviar comando para desligar luzes para a Casa de Banho
    Wire.beginTransaction(CasaDeBanhoAddress);
    Wire.write('d');
    Wire.endTransmission();
}

void ligarACSala() {
    Wire.beginTransaction(SalaAddress);
    Wire.write('a');
    Wire.endTransmission();
}

void ligarACquarto() {
    Wire.beginTransaction(QuartoAddress);
    Wire.write('u');
    Wire.endTransmission();
}

```

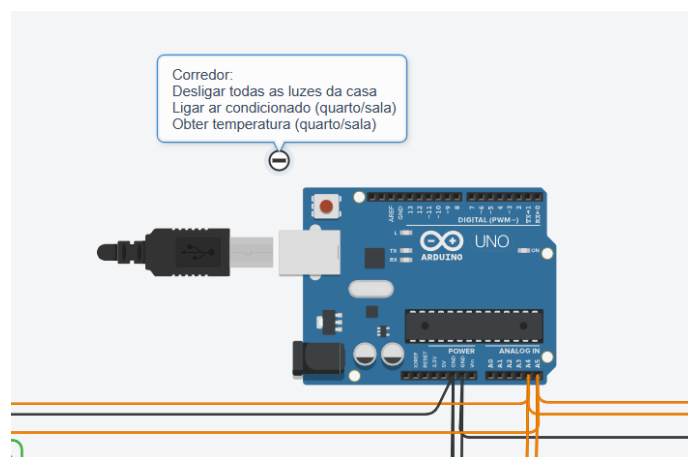


Figura 2- Circuito do Corredor

- **Arduíno (Quarto):**

- LED para simular a luz do quarto, controlado por botão.
- Motor para simular ar-condicionado, ativado por “comando de voz” (escrito no terminal).
- Sensor de temperatura com exibição no LCD.

Resumo:

Este Arduíno vai ter um led que vai simular a luz do quarto, esse led vai ser ligado ou desligado por um botão. Este Arduíno vai possuir também um sensor de temperatura que vai passar a informação da temperatura para um LCD. Por fim o Arduíno vai ter um motor a simular um ar-condicionado, que vai ser desligado ou ligado através do sensor de voz (terminal) ao escrever a mensagem "ar condicionado ".

Código:

```
#include <LiquidCrystal.h>
#include <Wire.h>
#define MyAddressQuarto 0x09

int LED = 13;
int BOTA0 = 12;
int ESTADO_BOTA0 = 0;
int VAR = 0;
int VAR2 = 0;

//variaveis AC
int MOTOR_AC = 11;
bool ACLigado = false;

//variaveis Sensor Temperatura
const int sensorTemp1 = A0;
float tmp_deg;

//variaveis LCD
LiquidCrystal lcd (10,9,7,6,5,4);

void setup()
{
  lcd.begin(16,2);
  lcd.clear();
  lcd.setCursor(0, 0);
  pinMode(MOTOR_AC, OUTPUT);
  pinMode(LED, OUTPUT);
  pinMode(BOTA0, INPUT);
  Wire.begin(MyAddressQuarto);
  Wire.onRequest(requestEvent);
  Wire.onReceive(comunicacao);
  Serial.begin(9600);
```

```

}
void loop()
{
    ////////////////////////////////////LUZ////////////////////////////////////
    VAR = digitalRead(BOTAO);
    if (VAR == 1 && VAR2 == 0) {
        ESTADO_BOTAO = 1 - ESTADO_BOTAO;
        delay(20);
    }

    VAR2=VAR;
    if(ESTADO_BOTAO == 1){
        digitalWrite(LED, 1);
    }else{
        digitalWrite(LED, 0);
    }

    ////////////////////////////////////AC////////////////////////////////////
    if (Serial.available()) {
        String comando = Serial.readStringUntil('\n');
        if (comando == "ar condicionado") {
            if (ACLigado) {
                Serial.print("A desligar AC\n");
                desligarAC();
            } else {
                Serial.print("A Ligar AC\n");
                ligarAC();
            }
        }
    }
}

```

```

//------(sensor temperatura)-----

```

```

float tmp_raw = analogRead(sensorTemp1);
float tmp_volt = (tmp_raw / 1024.0)*5000;
tmp_deg = (tmp_volt - 500)/10;
delay(50);

```

```

//------(LCD)-----

```

```

lcd.setCursor(0, 0);
lcd.print("Temp: ");
int parteInteira = (int)tmp_deg;
int parteDecimal = (tmp_deg - parteInteira) * 10;
if (parteDecimal < 0) {
    parteDecimal= parteDecimal*-1;
}
lcd.print(parteInteira);
lcd.print(".");
lcd.print(parteDecimal);
lcd.print(" C ");
delay(50);
}

```

```

//------(Funcoes Auxiliares)-----

```

```

void ligarAC() {
    if (!ACLigado) {
        digitalWrite(MOTOR_AC, HIGH); // Liga o motor
    }
}

```



```

    ACLigado = true;
}
}
void desligarAC() {
    if (ACLigado) {
        digitalWrite(MOTOR_AC, LOW); // Desliga o motor
        ACLigado = false;
    }
}
void requestEvent() {
    Wire.write((byte*)&tmp_deg, sizeof(tmp_deg));
}
void comunicacao(int i){
    char received = Wire.read();
    if(received == 'u'){
        ligarAC();
        Serial.println("Ar condicionado ligado");
    }else if(received == 'q'){
        ESTADO_BOTAO = 0;
        digitalWrite(LED, LOW); // Desliga a luz
        Serial.println("Luz desligada");
    }
}
}

```

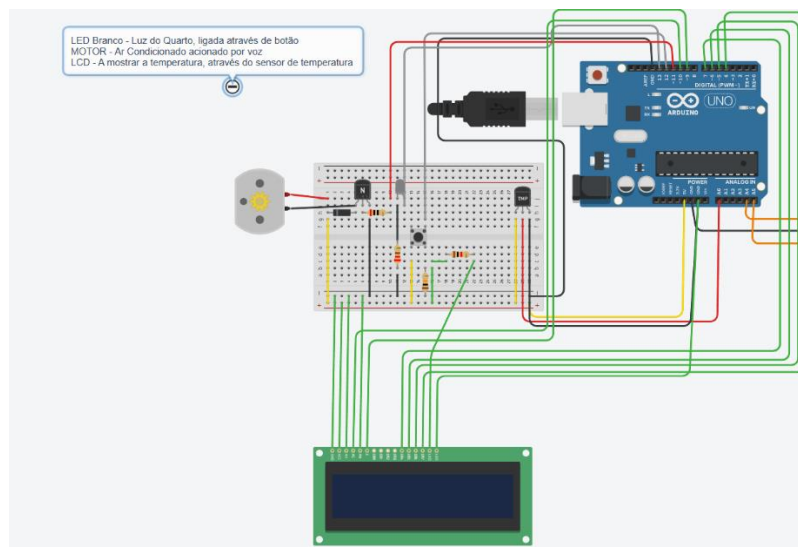


Figura 3- Circuito do Quarto

- **Arduíno (Casa de Banho):**

- LED para luz controlado por botão.
- Chuveiro ativado por “comando de voz” (escrito no terminal).
- Autoclismo ativado por botão, com LED indicador.
- Torneira automática acionada por sensor de proximidade.

Resumo:

Este Arduíno vai ter um led que será ligado através de um botão onde quando carregado ficará ligado ou desligado consoante o seu estado atual, vai ter também um chuveiro que será ativado através de "voz", como não existe nada que simule a voz será ativado quando o utilizador escrever no terminal "chuveiro", ligando ou desligando o motor consoante o seu estado atual, vamos ter também um botão de autoclismo que quando carregado ativará um led azul durante alguns segundos que simboliza que o autoclismo foi puxado, por fim vamos ter uma torneira automática, ou seja um sensor de proximidade da torneira que irá fazer com a água saia da torneira, que servirá para lavar as mãos por exemplo, onde irá acender um led verde sempre que o utilizador estiver com as mãos a 10cm da torneira, e apagará o led quando isso não estiver a ser cumprido.

Código:

```
#include <Wire.h>
#define MyAddressCasaDeBanho 0x10

//variaveis luz
int LED = 9;
int BOTAO = 10;
int ESTADO_BOTAO = 0;
int VAR = 0;
int VAR2 = 0;

//variaveis autoclismo
int LEDAutoclismo = 8;
int BOTAO_Autoclismo = 11;
int ESTADO_BOTAO_Autoclismo = 0;
int VARAutoclismo = 0;
int VAR2Autoclismo = 0;

//variaveis chuveiro
int MOTOR_chuveiro = 12;
bool chuveiroLigado = false;

//variaveis torneira
const int sensorPin = 7;
const int LED_PIN = 6;
```

```

void setup(){
  pinMode(MOTOR_chuveiro, OUTPUT);
  pinMode(LEDAutoclismo, OUTPUT);
  pinMode(BOTAO_Autoclismo, INPUT);
  pinMode(LED, OUTPUT);
  Wire.begin(MyAddressCasaDeBanho);
  Wire.onReceive(desligar);
  pinMode(BOTAO, INPUT);
  pinMode(LED_PIN, OUTPUT);
  Serial.begin(9600);
}

void loop(){
  //-----(luz)-----
  VAR = digitalRead(BOTAO);
  if (VAR == 1 && VAR2 == 0) {
    ESTADO_BOTAO = 1 - ESTADO_BOTAO;
    delay(20);
  }
  VAR2=VAR;
  if(ESTADO_BOTAO == 1){
    digitalWrite(LED, 1);
  }else{
    digitalWrite(LED, 0);
  }
  //-----(autoclismo)-----
  VARAutoclismo = digitalRead(BOTAO_Autoclismo);
  if (VARAutoclismo == 1 && VAR2Autoclismo == 0) {
    ESTADO_BOTAO_Autoclismo = 1 - ESTADO_BOTAO_Autoclismo;
    delay(20);
    if (ESTADO_BOTAO_Autoclismo == 1) {
      digitalWrite(LEDAutoclismo, HIGH);
      delay(700);
      digitalWrite(LEDAutoclismo, LOW); // Desliga o LED
    }
  }
  VAR2Autoclismo = VARAutoclismo;
  //-----(chuveiro)-----
  if (Serial.available()) {
    String comando = Serial.readStringUntil('\n');
    if (comando == "chuveiro") {
      if (chuveiroligado) {
        desligarChuveiro();
      }
    }
  }
  //-----(torneira)-----
  float readDist(int pingPin);
  float distance = readDist(sensorPin);
  if (distance <= 10 ) {
    digitalWrite(LED_PIN, HIGH); // Acende o LED azul
  } else {
    digitalWrite(LED_PIN, LOW); // Apaga o LED azul
  }
}

```

```

void ligarChuveiro() {
    if (!chuveiroLigado) {
        digitalWrite(MOTOR_chuveiro, HIGH); // Liga o motor
        chuveiroLigado = true;
    }
}

void desligarChuveiro() {
    if (chuveiroLigado) {
        digitalWrite(MOTOR_chuveiro, LOW); // Desliga o motor
        chuveiroLigado = false;
    }
}

float readDist(int pingPin) {
    float duracao, cm;
    pinMode(pingPin, OUTPUT);
    digitalWrite(pingPin, LOW);
    delayMicroseconds(2);
    digitalWrite(pingPin, HIGH);
    delayMicroseconds(5);
    digitalWrite(pingPin, LOW);
    pinMode(pingPin, INPUT);
    duracao = pulseIn(pingPin, HIGH);
    cm = (duracao / 29.4) / 2;
    return cm;
}

void desligar(int i){
    char received = Wire.read();
    if(received == 'd'){
        ESTADO_BOTAO = 0;
        digitalWrite(LED, LOW); // Desliga a luz
        Serial.println("Luz desligada");
    }
}
}

```

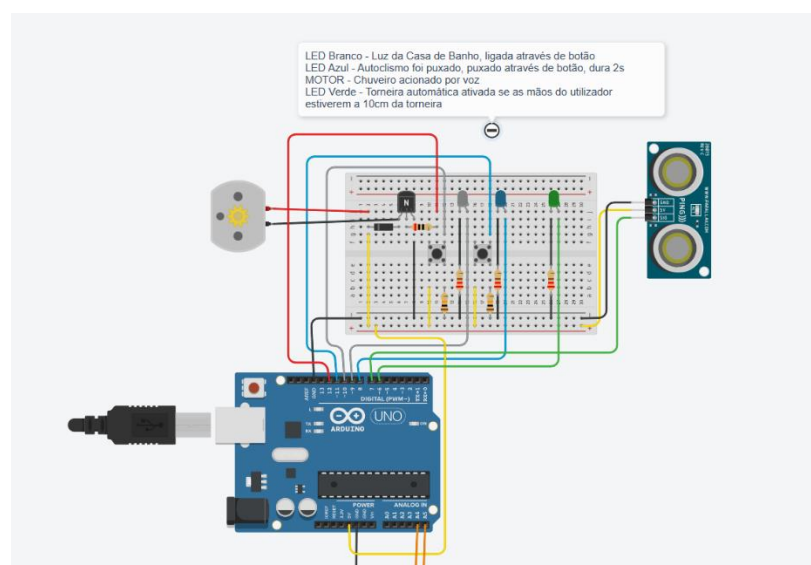


Figura 4 - Circuito da Casa de Banho

- **Arduíno (Sala):**

- LED para simular a luz da sala, controlado por “comando de voz” (escrito no terminal).
- Motor para simular ar-condicionado, ativado por “comando de voz” (escrito no terminal).
- Sensor de temperatura com exibição no LCD.
- LED para simular uma televisão, ativado por “comando de voz” (escrito no terminal).

Resumo:

Este Arduíno vai ter um led que vai simular a luz da sala, esse led vai ser ligado ou desligado por sensor de voz, ou seja, escrevendo "luz" no terminal. Depois vai possuir um motor que vai simular como se fosse um ar-condicionado. O utilizador ao escrever "ar condicionado". Depois este Arduíno vai ter um sensor de temperatura que vai passar a informação da temperatura da sala para um LCD. Vai ter também um led que vai simular uma televisão este led vai ser ligado por voz, ou seja, no terminal ao ser escrito "televisao".

Código:

```
#include <LiquidCrystal.h>
#include <Wire.h>
#define MyAddressSala 0x08
const int sensorTemp1 = A0;

//variaveis AC
int MOTOR_AC = 13;
bool ACLigado = false;

//variaveis luz
int LEDLuz = 12;
bool luzLigada = false;

//variaveis Televisao
int LEDTelevisao = 11;
bool TelevisaoLigada = false;

//variaveis LCD
LiquidCrystal lcd (10,9,8,7,6,5);

const float temperaturLimite = 25.0;
float tmp_deg;

void setup() {
  pinMode(LEDTelevisao,OUTPUT);
  pinMode(LEDLuz, OUTPUT);
```

```
pinMode(MOTOR_AC, OUTPUT);
lcd.begin(16,2);
lcd.clear();
lcd.setCursor(0, 0);
Wire.begin(MyAddressSala);
Wire.onRequest(requestEvent);
Wire.onReceive(comunicacao);
```

```
Serial.begin(9600);
```

```
}
```

```
void loop() {
  float tmp_raw = analogRead(sensorTemp1);
  float tmp_volt = (tmp_raw / 1024.0) * 5000;
  tmp_deg = (tmp_volt - 500) / 10;
  delay(50);
```

```
/////////////////////////////////LUZ/////////////////////////////////
if (Serial.available()) {
  String comando = Serial.readStringUntil('\n');
  if (comando.equals("luz")) {
    if (luzLigada) {
      Serial.print("A desligar Luz\n");
      desligarLuz();
    } else {
      Serial.print("A ligar Luz\n");
      ligarLuz();
    }
  } else if (comando.equals("ar condicionado")) {
    if (ACLigado) {
      Serial.print("A desligar AC\n");
      desligarAC();
    } else {
      Serial.print("A ligar AC\n");
      ligarAC();
    }
  } else if (comando.equals("televisao")) {
    if (TelevisaoLigada) {
      Serial.print("A desligar TV\n");
      desligarTelevisao();
    } else {
      Serial.print("A ligar TV\n");
      ligarTelevisao();
    }
  }
}
}

//------(LCD)-----
  lcd.setCursor(0, 0);
  lcd.print("Temp: ");
  int parteInteira = (int)tmp_deg;
  int parteDecimal = (tmp_deg - parteInteira) * 10;
  if (parteDecimal < 0) {
    parteDecimal= parteDecimal*-1;
  }
}
```

```

        lcd.print(parteInteira);
        lcd.print(".");
        lcd.print(parteDecimal);
        lcd.print(" C ");
        delay(50);
    }
    ////////////////////////////////////Funcoes Auxiliares////////////////////////////////////
    void requestEvent() {
        Wire.write((byte*)&tmp_deg, sizeof(tmp_deg));
    }
    void ligarLuz() {
        if (!luzLigada) {
            digitalWrite(LEDLuz, HIGH); // Liga a luz
            luzLigada = true;
        }
    }
    void desligarLuz() {
        if (luzLigada) {
            digitalWrite(LEDLuz, LOW); // Desliga a luz
            luzLigada = false;
        }
    }
    void ligarTelevisao() {
        if (!TelevisaoLigada){
            digitalWrite(LEDTelevisao, HIGH); // Liga televisao
            TelevisaoLigada = true;
        }
    }
    void desligarTelevisao() {
        if (TelevisaoLigada){
            digitalWrite(LEDTelevisao, LOW); // Desliga televisao
            TelevisaoLigada = false;
        }
    }
    void ligarAC() {
        if (!ACLigado) {
            digitalWrite(MOTOR_AC, HIGH); // Liga o motor
            ACLigado = true;
        }
    }
    void desligarAC() {
        if (ACLigado) {
            digitalWrite(MOTOR_AC, LOW); // Desliga o motor
            ACLigado = false;
        }
    }
    void comunicacao(int i){
        char received = Wire.read();
        if(received == 'a'){
            ligarAC();
            Serial.println("Ar condiciondo ligado");
        }else if(received == 's'){
            desligarLuz();
            Serial.println("Luz desligada");
        }
    }
}

```

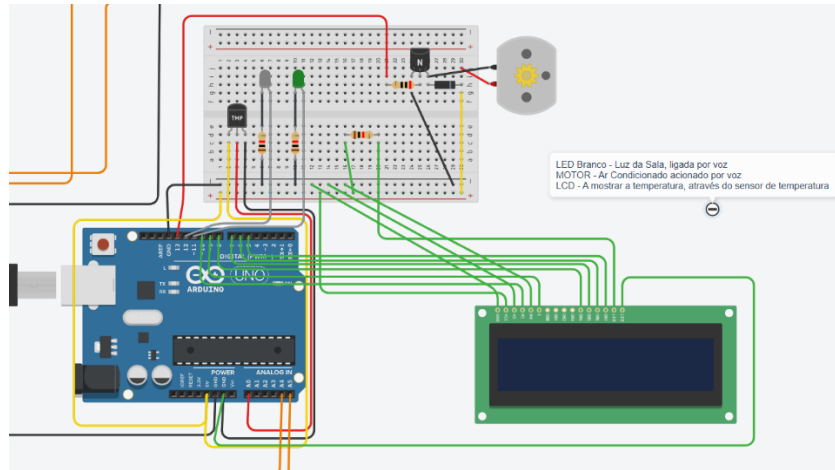


Figura 5 - Circuito da Sala

- **Arduíno (Cozinha):**

- Luz controlada por “comando de voz” (escrito no terminal).
- Sensor de fumo simulando deteção de incêndio.
- Alarme de incêndio (Piezo).

Resumo:

Este Arduíno vai ter uma luz que será ativada por "voz", ou seja, quando o utilizador escrever no terminal "luz", irá ligar ou desligar um led consoante o seu estado atual, iremos ter também um sensor de fumo (sensor de gás no Tinkercad) que irá soar um alarme sempre que existir fumo a mais, isto é uma espécie simulação de incêndio.

Código:

```
#include <Wire.h>
#define MyAddressCozinha 0x11
const int buzzerPin = 11;
const int sensorFumo = A0;
```



```

// Variáveis luz
int LEDLuz = 13;
bool luzLigada = false;

void setup() {
  pinMode(LEDLuz, OUTPUT);
  pinMode(buzzerPin, OUTPUT);
  Wire.begin(MyAddressCozinha);
  Wire.onReceive(desligar);
  Serial.begin(9600);
}

void loop() {
  float sensorValue;
  sensorValue = analogRead(sensorFumo);
  if (sensorValue >= 470) {
    tone(buzzerPin, 1500);
    delay(10);
    noTone(buzzerPin);
  }

  if (Serial.available()) {
    String comando = Serial.readStringUntil('\n');
    if (comando.equals("luz")) {
      if (luzLigada) {
        Serial.print("A desligar Luz\n");
        desligarLuz();
      } else {
        Serial.print("A ligar Luz\n");
        ligarLuz();
      }
    }
    delay(50);
  }
}

void ligarLuz() {
  if (!luzLigada) {
    digitalWrite(LEDLuz, HIGH); // Liga a luz
    luzLigada = true;
  }
}

void desligarLuz() {
  if (luzLigada) {
    digitalWrite(LEDLuz, LOW); // Desliga a luz
    luzLigada = false;
  }
}

void desligar(int i){
  char received = Wire.read();
  if(received == 'c'){
    desligarLuz();
    Serial.println("Luz desligada");
  }
}

```

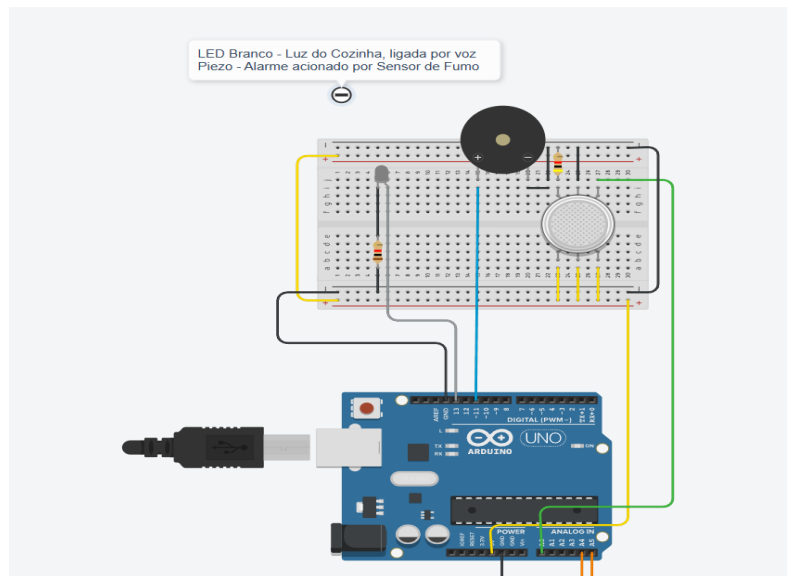


Figura 6 - Circuito da Cozinha

Funcionalidades Previstas

- **Comunicação entre Divisões:**
 - Utilização de pipes para transmitir dados entre as divisões.
- **Controlo Central (Master):**
 - Acionamento de luzes por comando central.
 - Controlo do ar-condicionado em diferentes divisões.
- **Interatividade por Divisão:**
 - Comandos específicos para cada divisão.

Resultados Esperados

- **Simulação Realista:**
 - Cada divisão da casa deve responder de forma realista aos comandos do utilizador.
 - Interação fluída e lógica entre as divisões.
- **Integração de Sensores:**

- Correta leitura e resposta dos sensores de temperatura, proximidade, “voz”, entre outros.
- **Controlo Central Eficaz:**
 - O controlo central (Master) deve coordenar eficientemente as ações nas diferentes divisões.

Abordagem a Realizar

O desenvolvimento do projeto seguirá as seguintes etapas:

Implementação Individual das Divisões:

- Programação específica para cada Arduino e seus componentes.

Integração e Testes Locais:

- Teste de interação entre as divisões individualmente.

Implementação do Master (Corredor):

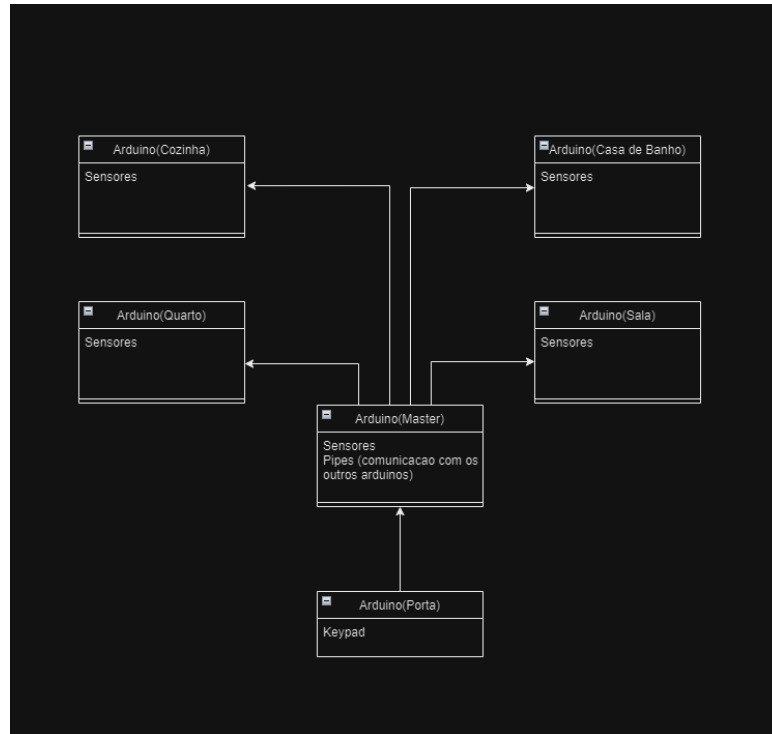
- Desenvolvimento do Arduino Master para coordenar todas as ações relativas ao mesmo.

Testes Globais e Ajustes:

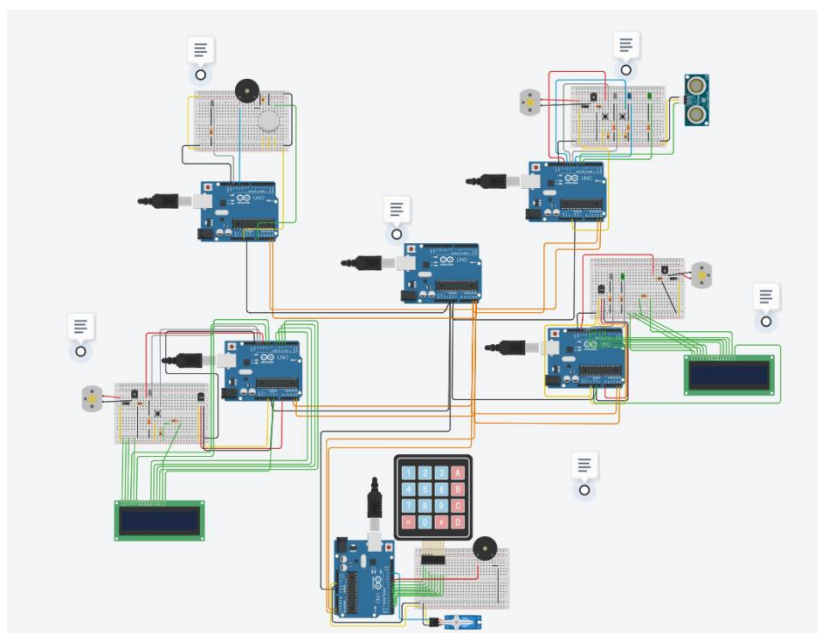
- Integração completa e teste global do sistema.
- Ajustes e correções conforme necessário.
-

Esquema do Projeto:

Antes da realização:



Depois da realização:



Resultados Obtidos

Com o andamento da realização do projeto, viemos a deparar nos com várias adversidades, das quais a eficiência e capacidade de respostas foram as principais razões para que o resultado esperado não fosse igual ao resultado obtido, visto que tivemos de optar por não realizar algumas funcionalidades previstas.

Tivemos de retirar algumas funcionalidades relativas à comunicação entre os diversos Arduínos, pois fazia com que a simulação do projeto deixasse de ser exequível, por fim fomos também forçados a retirar a parte do menu interativo com o utilizador, como forma de simulação da divisão onde o utilizador se queria dirigir, pois deixava, mais uma vez, a simulação bastante lenta, ao ponto de cada instrução demorar aproximadamente 15 segundos.

Com o retiro destas funcionalidades a simulação passou a ficar ligeiramente mais rápida, demorando em média 10 segundos para realizar cada instrução, variando muito com a complexidade da instrução solicitada.

Conclusão

Com este projeto pretendemos criar uma simulação de uma **smart home** interativa e realista, proporcionando ao utilizador uma experiência imersiva de controlo sobre diferentes aspetos de cada divisão. A utilização de sensores e comunicação entre placas Arduino proporcionará uma simulação eficaz e de certo modo complexa.

As principais dificuldades que nos deparámos foram, a baixa capacidade de resposta e a baixa performance da plataforma escolhida, o **tinkercad**, como o nosso projeto possui vários Arduínos a plataforma não tem a capacidade esperada para processar todas as funcionalidades que pretendíamos.

Por fim a comunicação entre os diversos Arduínos foi também algo bastante complexo e desafiador de implementar visto que o ponto chave do nosso projeto foi a comunicação entre as diversas divisões.

Para concluir, a conclusão deste projeto foi um sucesso, bastante enriquecedor a nível de conhecimento e interessante, apesar de todas as dificuldades obtidas, explicadas anteriormente, o resultado final foi bastante satisfatório e positivo.