

**Pontifícia Universidade Católica do Paraná**

**Escola Politécnica**

**Curitiba**

**2022**

**Grupo 2**

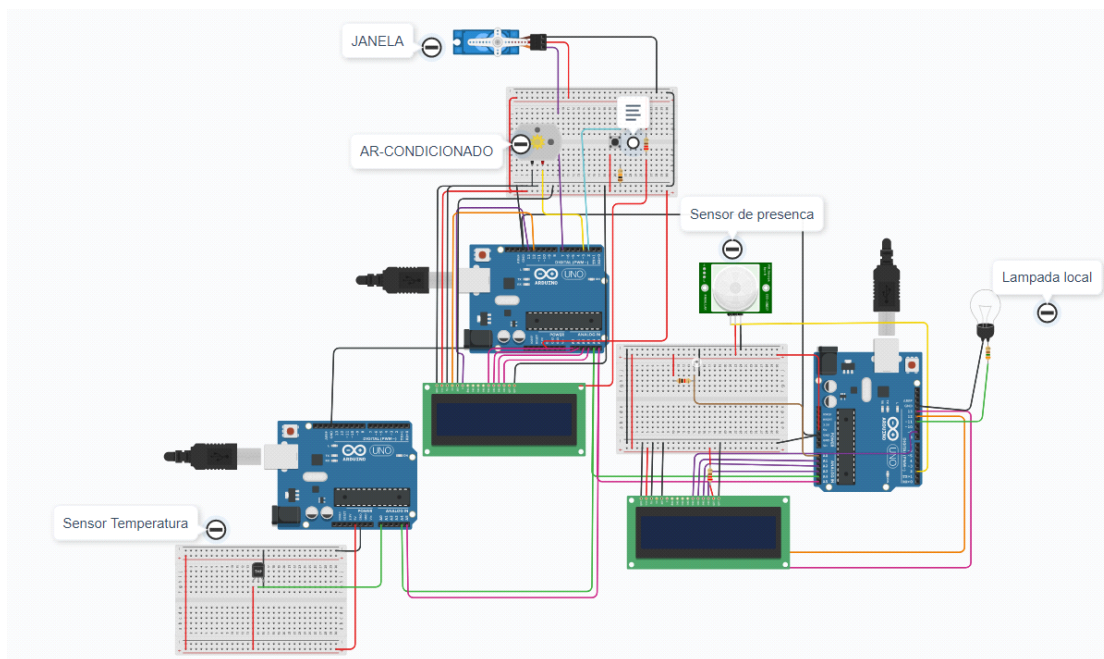
**Enzo Bottan Coutinho, Gustavo Furini, Henrique Conceição, Leonardo Nervino, Pedro Henrique Rolkouski.**

Descrição resumida (uma página) do projeto a ser desenvolvido pela equipe:

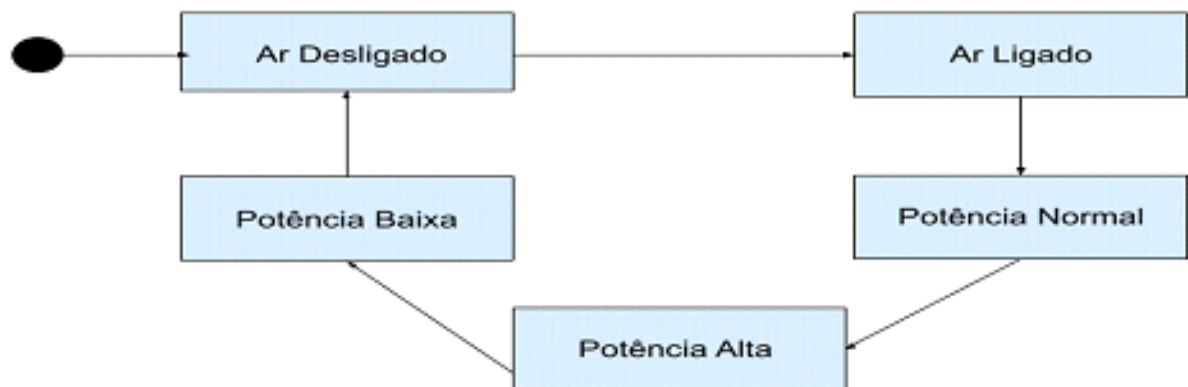
O projeto a ser desenvolvido é um sistema de automação de um ar condicionado e uma janela que se fecha conforme certas temperaturas e se houver movimento, junto com uma lâmpada que acende e apaga conforme horário e um sensor de presença.

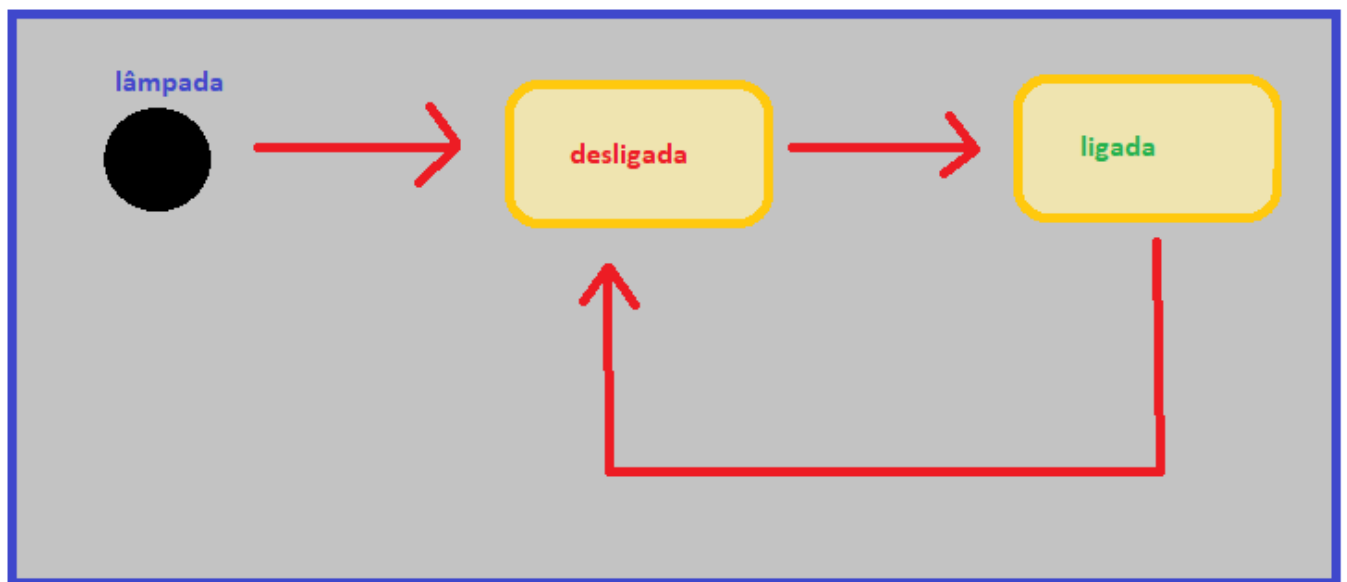
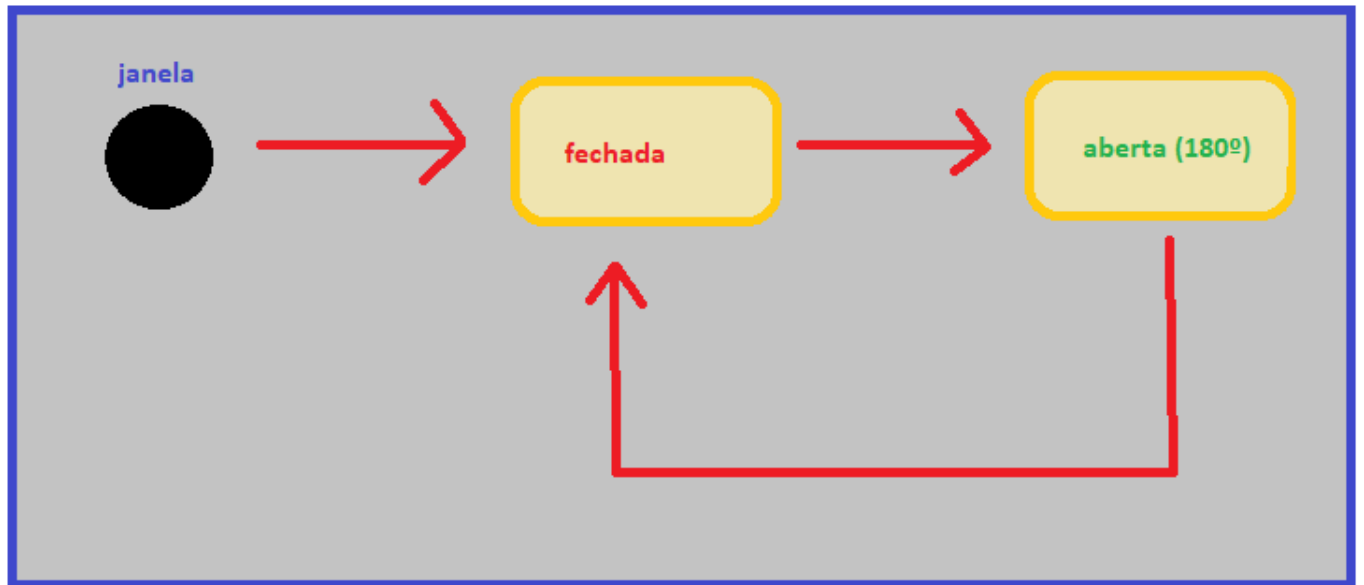
Conta com um micro servo como uma janela, botão para ligar e desligar o ar condicionado, um motor CC como sendo o ar condicionado, um sensor de temperatura TMP36 para medir a temperatura.

No sensor de presença usa um sensor PIR, sensor de luz ambiente para identificar dia ou noite, uma lâmpada para acender quando for noite. Possui duas telas LCD, uma ligada ao ar condicionado, informando se está ligado ou desligado, após ligado mostra temperatura e a potência do ar, já no outro LCD informa se tem ou não presença e se está dia ou noite.

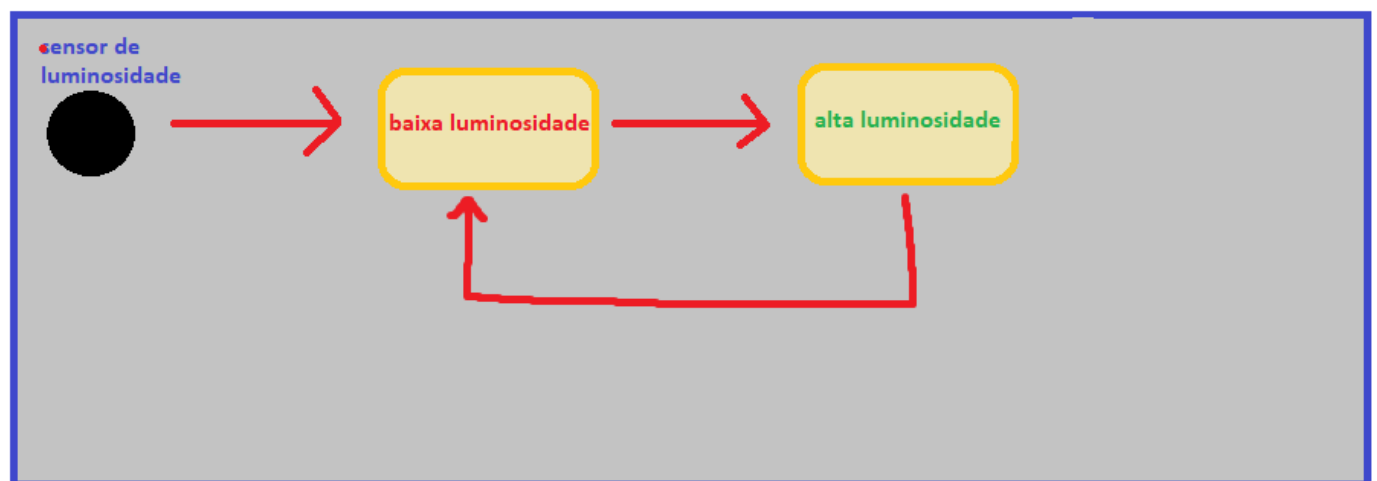
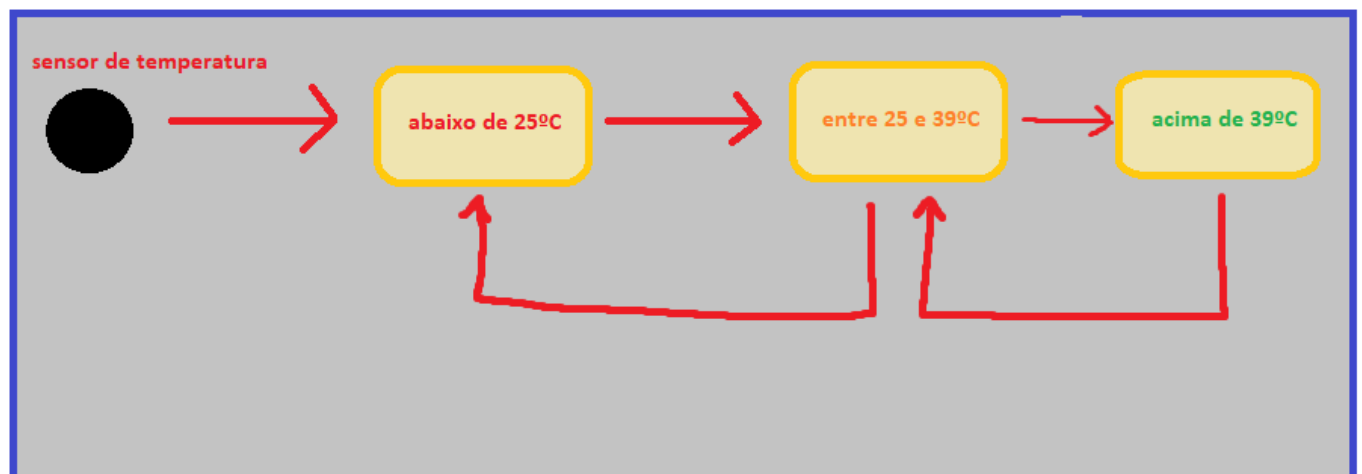


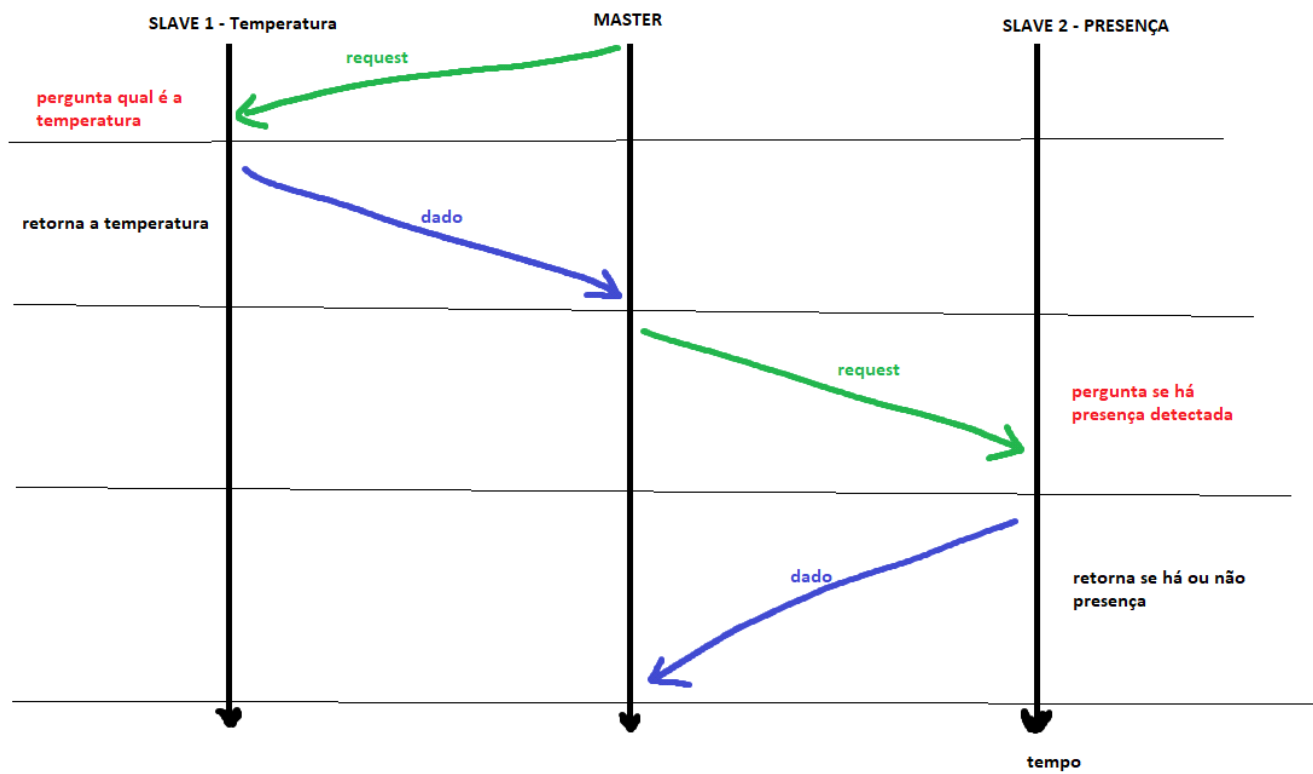
- **Ar-Condicionado**

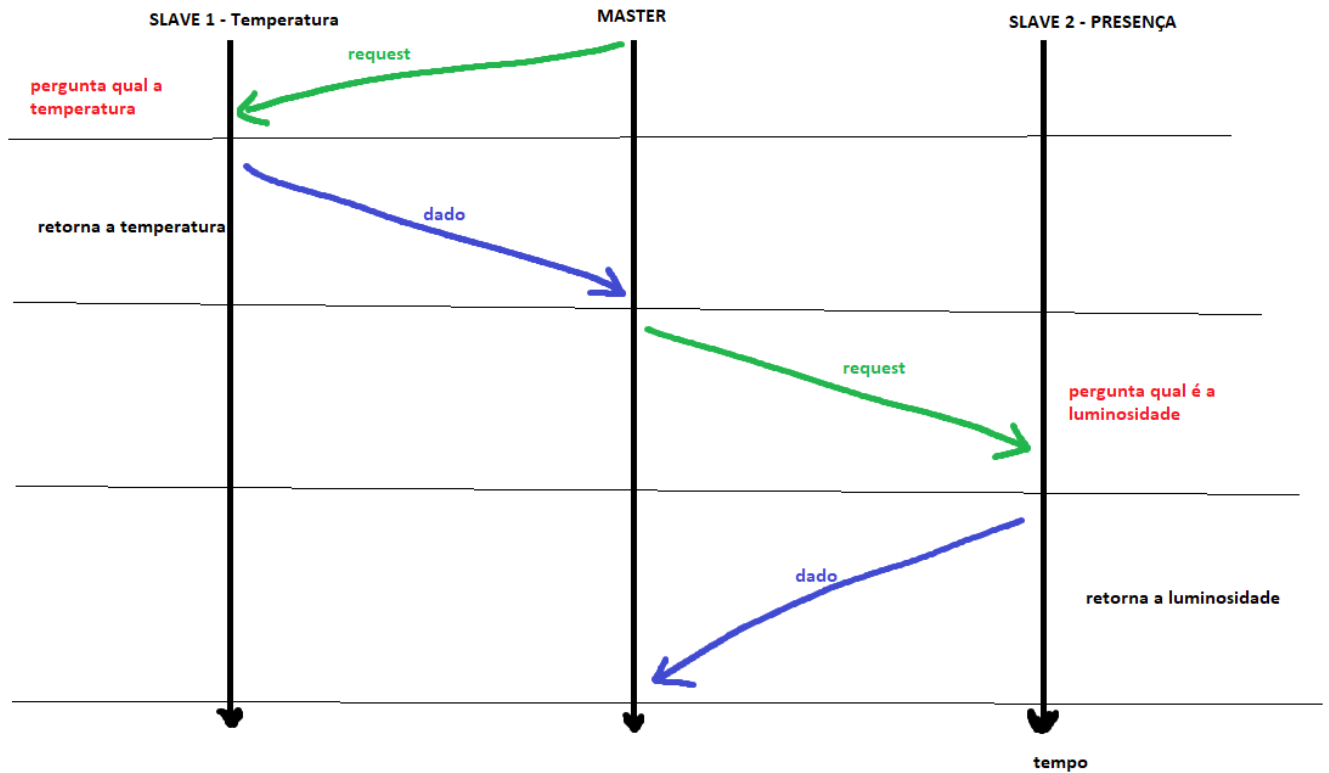




Visor LCD:







## Codigo Master

```
#include <LiquidCrystal.h>
```

```
#include <Wire.h>
```

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

```
const int ID_MASTER = 1;
```

```
const int ID_SLAVE_TEMP = 2;
```

```
const int ID_SLAVE_PIR = 3;
```

```
const int PORTA_MOTOR = 3;
```

```
float potencia = 0;
```

```
const int POTENCIA_MAXIMA = 1023;
```

```
const int POTENCIA_NORMAL = 400;
```

```
const int POTENCIA_BAIXA = 100;
```

```
Servo janela;
```

```
int posicao = 0;
```

```
byte temperatura;
```

```
byte presenca;
```

```
const int LCD_RS = 12;
```

```
const int LCD_E = 13;
```

```
const int LCD_DB4 = A0;
```

```
const int LCD_DB5 = A1;
```

```
const int LCD_DB6 = A2;
```

```
const int LCD_DB7 = A3;
```

```
LiquidCrystal lcd(LCD_RS, LCD_E, LCD_DB4, LCD_DB5, LCD_DB6, LCD_DB7);
```

```
byte infotemp, infopir;
```

```
boolean status = false;
```

```
boolean recebeu_msg = false;
```

```
enum{
```

```
    OCIOSO,
```

```
    ATUALIZANDO_AR, PEDINDO_SLAVE1,PEDINDO_SLAVE2,  
    ESPERANDO_SLAVE1, ESPERANDO_SLAVE2,  
    ATUALIZANDO_JANELA} estado = OCIOSO;
```

```
void setup(){  
    janela.attach(7);  
    pinMode(2,INPUT);  
    attachInterrupt(0, statusAr, RISING);  
    Wire.begin(ID_MASTER);  
    Wire.onReceive(receberMensagem);  
    lcd.begin(16, 2);  
    lcd.setCursor(0,1);  
    Serial.begin(9600);  
}
```

```
void statusAr(){  
    status = !status;  
    if(status){  
        estado = PEDINDO_SLAVE1;  
    }else{  
        estado = OCIOSO;  
    }  
}
```

```
    Serial.println(status);  
}
```



```
void loop()
{
  switch(estado){
    case OCIOSO:
      lcd.clear();
      lcd.setCursor(0,0);
      lcd.print("AR - Desligado");
      while(status == false){
        delay(1000);
      }
      lcd.clear();
      lcd.setCursor(0,0);
      lcd.print("Temperatura");
      break;
    case PEDINDO_SLAVE1:
      if(status){
        slaveTemp();
        estado = ESPERANDO_SLAVE1;
      }else{
        estado = OCIOSO;
      }
      break;
    case PEDINDO_SLAVE2:
      slavePIR();
      estado = ESPERANDO_SLAVE2;
      break;
    case ESPERANDO_SLAVE1:
```

```
    while(!recebeu_msg);  
    Serial.print("Temp recebida: ");  
        Serial.println(infotemp);  
    temperatura = infotemp;  
    recebeu_msg = false;  
    estado = PEDINDO_SLAVE2;  
    break;
```

case ESPERANDO\_SLAVE2:

```
    while(!recebeu_msg);  
    Serial.print("Presenca recebida: ");  
        Serial.println(infopir);  
    presenca = infopir;  
    recebeu_msg = false;  
    estado = ATUALIZANDO_AR;  
    break;
```

case ATUALIZANDO\_AR:

```
    lcd.setCursor(0,1);  
    lcd.print(temperatura);  
    lcd.print(" C");  
    if(infopir == 1){  
        lcd.setCursor(6,1);  
        lcd.print("P:SIM");  
        if(temperatura > 0 && temperatura < 25){  
            potencia = POTENCIA_BAIXA;  
            lcd.setCursor(13,1);  
            lcd.print("MIN");  
        }  
    }
```

```

        Serial.println("TEMPERATURA baixa utilizando pouca potencia");
    }else if(temperatura >= 25 && temperatura <= 39){
        potencia = POTENCIA_NORMAL;

        lcd.setCursor(13,1);

        lcd.print("NOR");

        Serial.println("TEMPERATURA RAZOAVEL utilizando potencia normal");
    }else if(temperatura >= 40){
        estado_janela(false);

        potencia = POTENCIA_MAXIMA;

        lcd.setCursor(13,1);

        lcd.print("MAX");

        Serial.println("TEMPERATURA ALTA utilizando potencia maxima");
    }
}
}else{
    potencia = POTENCIA_BAIXA;

    lcd.setCursor(6,1);

    lcd.print("P:NAO");

    lcd.setCursor(13,1);

    lcd.print("MIN");

    Serial.println("SEM PRESENCA utilizando pouca potencia");
}

analogWrite(PORTA_MOTOR, potencia);

estado = PEDINDO_SLAVE1;

break;

}

}

```

```
void slaveTemp(){
    Serial.println("Chamando slave Temperatura");
    Wire.beginTransmission(ID_SLAVE_TEMP);
    Wire.write('Master Requisitando Temperatura');
    Wire.endTransmission();
}
```

```
void slavePIR(){
    Serial.println("Chamando slave PIR");
    Wire.beginTransmission(ID_SLAVE_PIR);
    Wire.write('Master Requisitando PRESENCA');
    Wire.endTransmission();
}
```

```
void receberMensagem(int numBytesRecebidos)
{
    byte info = Wire.read();
    if (estado == ESPERANDO_SLAVE2){
        infopir = info;
    }else{
        infotemp = info;
    }
    recebeu_msg = true;
}
```

```
void estado_janela(bool abrir){
```

```

if(abrir){
  while(posicao <=180){
    posicao++;
    janela.write(posicao);
    delay(10);
  }

}else{
  while(posicao >=0){
    janela.write(posicao);
    posicao--;
    delay(10);
  }
}
}

```

### **Codigo Slave Presenca**

```
#include <LiquidCrystal.h>
```

```
#include <Wire.h>
```

```
const int lampada = 11;
```

```
const int ID_MASTER = 1;
```

```
const int ID_SLAVE_PIR = 3;
```

```
const int pinoPIR = 2;
```

```
bool leitura = false;
const int LCD_RS = 12;
const int LCD_E = 13;
const int LCD_DB4 = 9;
const int LCD_DB5 = A1;
const int LCD_DB6 = A2;
const int LCD_DB7 = A3;
```

```
LiquidCrystal lcd(LCD_RS, LCD_E, LCD_DB4, LCD_DB5, LCD_DB6, LCD_DB7);
```

```
void setup()
```

```
{
  Wire.begin(ID_SLAVE_PIR);
  Wire.onReceive(pedidoLeitura);
  lcd.begin(16, 2);
  Serial.begin(9600);
  pinMode(pinoPIR, INPUT);
    pinMode(lampada, OUTPUT);
}
```

```
void loop()
```

```
{
  if(leitura){
    transmitirPresenca();
    verificaLuz();
  }
```

```
  aguardaPedido();
```

```

}

void verificaLuz(){
    int luz = analogRead(0);
    if(luz <= 788){
        lcd.setCursor(0,1);
        lcd.print("DIA");
        digitalWrite(lampada,0);
    }else{
        lcd.setCursor(0,1);
        lcd.print("NOITE");
        digitalWrite(lampada,1);
    }
    Serial.println(luz);
}

bool lerPresenca(){
    if(digitalRead(pinoPIR) == HIGH){
        lcd.clear();
        lcd.setCursor(0,0);
        lcd.print("HA PRESENCA");
        return true;
    }else{
        lcd.clear();
        lcd.setCursor(0,0);
        lcd.print("SEM PRESENCA");
        return false;
    }
}

```

```
void transmitirPresenca()
```

```
{  
    Wire.beginTransmission(ID_MASTER);  
    Wire.write(lerPresenca());  
    Wire.endTransmission();  
}
```

```
void pedidoLeitura(int numBytesRecebidos)
```

```
{  
    for (int count = 1; count < numBytesRecebidos; count++)  
    {  
        char c = Wire.read();  
        Serial.print(c);  
    }  
    char x = Wire.read();  
    leitura = true;  
}
```

```
void aguardaPedido()
```

```
{  
    leitura = false;  
    Serial.println("Sem pedidos!");  
    while (!leitura)delay(100);  
}
```



## Codigo Slave Ar

```
#include <Wire.h>
```

```
#include <LiquidCrystal.h>
```

```
const int ID_MASTER = 1;
```

```
const int ID_SLAVE_TEMP = 2;
```

```
const int PORTA_SENSOR_TEMP = A0;
```

```
bool leitura = false;
```

```
LiquidCrystal LCD(12,11,5,4,3,2);
```

```
void setup()
```

```
{
```

```
  Wire.begin(ID_SLAVE_TEMP);
```

```
  Wire.onReceive(pedidoLeitura);
```

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

```
}
```

```
void loop()
```

```
{
```

```
  if(leitura){
```

```
    byte temperatura = lerTemperatura();
```

```
    transmitirTemperatura(temperatura);
```

```

    }
    aguardaPedido();
}

float lerTemperatura(){
    int SensorTempTensao=analogRead(PORTA_SENSOR_TEMP);

    // Converte a tensao lida

    float Tensao = SensorTempTensao * 5;

    Tensao /= 1024;

    // Converte a tensao lida em Graus Celsius

    float temperatura = (Tensao - 0.5) * 100;

    return temperatura;
}

void transmitirTemperatura(byte temperatura)
{
    Serial.println("Enviando Temperatura");
    Wire.beginTransmission(ID_MASTER);
    Wire.write(temperatura);
    Wire.endTransmission();

    leitura = false;
}

void pedidoLeitura(int numBytesRecebidos)
{
    for (int count = 1; count < numBytesRecebidos; count++)

```

```
{
    char c = Wire.read();
    Serial.print(c);
}

    char x = Wire.read();
    leitura = true;
}

void aguardaPedido()
{
    leitura = false;
    Serial.println("Sem pedidos!");
    while (!leitura)delay(100);
}
```