

ALGORITMOS DE PROGRAMAÇÃO

A sequência lógica de instruções que resolve um problema ou executa uma tarefa.

É a base do desenvolvimento do projeto, guiando o comportamento do dispositivo passo a passo.

```
--- STATUS DO SISTEMA ---  
Temperatura: 23.30°C  
Umidade: 62.50%  
LDR valor: 133  
Luminosidade: 0%  
Pessoas: 0 (Tags Hist: 0)  
Temp. Preferida: 25.00°C  
Lum. Preferida: 50%  
Prefs Atualizadas: NAO  
Diff Temp: -1.70°C  
  
--- FLAGS ---  
modoManualIllum: NAO  
modoManualClima: NAO  
ilumAtiva: NAO  
monitorandoLDR: SIM  
atualizandoPref: NAO  
prefsAtualizadas: NAO  
erroSensor: NAO  
erroConexao: NAO  
  
--- CLIMA ---  
Ligado: NAO  
Velocidade: 0  
Umidificando: NAO  
Aleta V: NAO  
Aleta H: NAO  
Timer: 0  
  
--- SISTEMA ---  
WiFi: Conectado  
-----
```

FUNÇÕES EXEMPLO →

LEITURA DO CARTÃO:

```
// Verifica se uma nova tag RFID foi aproximada. Se sim, lê o ID da tag.
void processarNFC() {
    // `PICC_IsNewCardPresent()` : Há um cartão novo?
    // `PICC_ReadCardSerial()` : Conseguiu ler o número de série?
    if (!mfr522.PICC_IsNewCardPresent() || !mfr522.PICC_ReadCardSerial()) {
        return; // Se não, sai da função.
    }

    String tag = ""; // String para montar o ID da tag.
    for (byte i = 0; i < mfr522.uid.size; i++) { // Percorre os bytes do ID.
        if (mfr522.uid.uidByte[i] < 0x10) tag += "0"; // Adiciona um "0" à esquerda se for menor que 16 (ex: 0F em vez de F).
        tag += String(mfr522.uid.uidByte[i], HEX); // Converte o byte para hexadecimal e adiciona à string.
    }
    tag.toUpperCase(); // Converte para maiúsculas para padronizar.

    debugPrint("Tag NFC lida: " + tag);
    gerenciarPresenca(tag); // Chama a função que trata a entrada/saída da pessoa.

    mfr522.PICC_HaltA(); // Para a comunicação com a tag atual (importante para ler novas tags).
    mfr522.PCD_StopCrypto1(); // Para a criptografia (se estiver ativa).
}
```

FUNÇÕES EXEMPLO →

TESTE INICIAL DOS RELÉS:

```
// Testa todos os relés, ligando e desligando cada um individualmente.  
// Ajuda a verificar se a fiação e os relés estão funcionando.  
void testarReles() {  
    debugPrint("=== INICIANDO TESTE DOS RELÉS ===");  
    int pinosRele[] = {RELE_1, RELE_2, RELE_3, RELE_4};  
  
    for (int i = 0; i < 4; i++) {  
        int pino = pinosRele[i];  
        debugPrint("Testando Relé " + String(i + 1) + " (Pino " + String(pino) + ")");  
  
        digitalWrite(pino, LOW); // Liga o relé (LOW = ON).  
        debugPrint("  Relé " + String(i + 1) + " LIGADO (nível BAIXO no pino).");  
        delay(200); // Mantém ligado por um tempo.  
  
        digitalWrite(pino, HIGH); // Desliga o relé (HIGH = OFF).  
        debugPrint("  Relé " + String(i + 1) + " DESLIGADO (nível ALTO no pino).");  
        delay(200); // Mantém desligado por um tempo.  
    }  
    debugPrint("=== TESTE DOS RELÉS CONCLUÍDO ===");  
}
```

APRESENTAÇÃO DO SITE

[LINK DO SITE](#)

