

**FECAP**  
**ANÁLISE E DESENVOLVIMENTO DE SISTEMAS**

**SMART HOUSE – ALGORITMO**  
**PROJETO INTEGRADOR**

**Algoritmos e Lógica de Programação**

**Professora: Lucy Mari Tabuti**

**GRUPO 42**

**Caio Araujo - 19010077**

**Gustavo Bitencourt Silva - 25027720**

**Jorge Alves Marinho - 25027029**

**Matheus Breda Andreo - 25027317**

**São Paulo**

**1 / 2025**

# Algoritmo CasaInteligente

**// --- Seção de Declaração de Constantes ---**

constante inteiro MAX\_DISPOSITIVOS = 10 **// Define o número máximo de dispositivos que podem ser cadastrados.**

constante inteiro MAX\_USUARIOS = 2 **// Define o número máximo de usuários que podem ser cadastrados.**

constante inteiro NUM\_SENSORES = 5 **// Define o número de sensores na casa (Quarto A, Quarto B, Sala, Cozinha, Piscina).**

**// --- Seção de Definição de Tipos de Dados Estruturados (Registros) ---**

**// Tipo de dado para representar um dispositivo inteligente.**

tipo Dispositivo = registro

nome: cadeia **// Nome amigável do dispositivo (ex: "Luz da Sala").**

idSensor: inteiro **// ID do sensor associado a este dispositivo (1 a NUM\_SENSORES).**

estado: booleano **// Estado atual do dispositivo: verdadeiro (ligado) ou falso (desligado).**

consumoHora: real **// Consumo de energia do dispositivo em kWh por hora.**

fimregistro

**// Tipo de dado para representar um usuário do sistema.**

tipo Usuario = registro

nome: cadeia **// Nome do usuário.**

```
senha: cadeia    // Senha do usuário para acesso.

nivelAcesso: inteiro // Nível de acesso do usuário: 1 (Básico), 2
(Avançado).

fimregistro


// --- Seção de Declaração de Variáveis ---


// Vetores (arrays) para armazenar múltiplos dispositivos e usuários.

vetor dispositivos[MAX_DISPOSITIVOS] de Dispositivo

vetor usuarios[MAX_USUARIOS] de Usuario


// Matriz para armazenar o consumo diário de energia de cada sensor.

// A primeira dimensão representa o ID do sensor (0 a NUM_SENSORES -
1).

// A segunda dimensão representa o dia do mês (0 a 30 para até 31 dias).

matriz consumoDiario[NUM_SENSORES][31] de real


// Vetor para armazenar mensagens de alerta de consumo elevado.

vetor alertas[10] de cadeia


// Variáveis para controlar o número de dispositivos e usuários
cadastrados.

inteiro numDispositivos = 0

inteiro numUsuarios = 0
```

**// Variável para simular o dia atual (usado para registrar o consumo diário).**

inteiro diaAtual = 1

**// --- Seção de Definição de Funções e Procedimentos ---**

**// Função para cadastrar um novo usuário no sistema.**

funcao cadastrarUsuario()

**// Verifica se o número máximo de usuários não foi atingido.**

se numUsuarios < MAX\_USUARIOS entao

    escreva "Digite o nome do usuário: "

    leia usuarios[numUsuarios].nome

    escreva "Digite a senha do usuário: "

    leia usuarios[numUsuarios].senha

    escreva "Digite o nível de acesso (1 - Básico, 2 - Avançado): "

    leia usuarios[numUsuarios].nivelAcesso

**// Incrementa o contador de usuários cadastrados.**

numUsuarios <- numUsuarios + 1

    escreva "Usuário cadastrado com sucesso!"

senao

    escreva "Número máximo de usuários atingido."

fimse

fimfuncao

**// Função para cadastrar um novo dispositivo inteligente.**

```
funcao cadastrarDispositivo()
```

```
    // Verifica se o número máximo de dispositivos não foi atingido.
```

```
    se numDispositivos < MAX_DISPOSITIVOS entao
```

```
        escreva "Digite o nome do dispositivo: "
```

```
        leia dispositivos[numDispositivos].nome
```

```
        escreva "Digite o ID do sensor associado (1 a 5): "
```

```
        leia dispositivos[numDispositivos].idSensor
```

```
    // Valida se o ID do sensor está dentro do intervalo permitido.
```

```
        se dispositivos[numDispositivos].idSensor < 1 ou
```

```
dispositivos[numDispositivos].idSensor > NUM_SENSORES entao
```

```
        escreva "ID de sensor inválido."
```

```
        retorne // Sai da função se o ID do sensor for inválido.
```

```
    fimse
```

```
    // Inicializa o estado do dispositivo como desligado.
```

```
    dispositivos[numDispositivos].estado <- falso
```

```
    escreva "Digite o consumo por hora (em kWh): "
```

```
    leia dispositivos[numDispositivos].consumoHora
```

```
    // Incrementa o contador de dispositivos cadastrados.
```

```
    numDispositivos <- numDispositivos + 1
```

```
    escreva "Dispositivo cadastrado com sucesso!"
```

```
senao
```

```
    escreva "Número máximo de dispositivos atingido."
```

```
    fimse
```

```
fimfuncao
```

**// Função para exibir o consumo de energia de um sensor específico.**

funcao exibirConsumoEnergia()

inteiro idSensor

escreva "Digite o ID do sensor (1 a 5) para visualizar o consumo: "

leia idSensor

**// Valida se o ID do sensor está dentro do intervalo permitido.**

se idSensor >= 1 e idSensor <= NUM\_SENSORES entao

escreva "Consumo diário do Sensor ", idSensor, ":"

**// Loop para exibir o consumo de cada dia registrado para o sensor.**

para inteiro i de 1 ate diaAtual faca

escreva "Dia ", i, ": ", consumoDiario[idSensor - 1][i - 1], " kWh"

fimpara

senao

escreva "ID de sensor inválido."

fimse

fimfuncao

**// Função para controlar (ligar/desligar) um dispositivo remotamente.**

funcao controlarDispositivo()

cadeia nomeDispositivo

inteiro encontrado = -1 **// Variável para indicar se o dispositivo foi encontrado.**

escreva "Digite o nome do dispositivo que deseja controlar: "

leia nomeDispositivo

**// Loop para procurar o dispositivo pelo nome no vetor de dispositivos.**

```

para inteiro i de 0 ate numDispositivos - 1 faca

    se dispositivos[i].nome = nomeDispositivo entao

        encontrado <- i // Armazena o índice do dispositivo encontrado.

        pare // Sai do loop assim que o dispositivo é encontrado.

    fimse

fimpara

// Se o dispositivo foi encontrado.

se encontrado != -1 entao

    escreva "Estado atual do dispositivo ", dispositivos[encontrado].nome, ": "

    // Exibe o estado atual do dispositivo.

    se dispositivos[encontrado].estado entao

        escreva "Ligado"

    senao

        escreva "Desligado"

    fimse

    escreva "\nDigite o novo estado (ligar/desligar): "

    cadeia novoEstado

    leia novoEstado

    // Altera o estado do dispositivo com base na entrada do usuário.

    se novoEstado = "ligar" entao

        dispositivos[encontrado].estado <- verdadeiro

        escreva "Dispositivo ", dispositivos[encontrado].nome, " ligado."

    senao se novoEstado = "desligar" entao

```

```

    dispositivos[encontrado].estado <- falso

    escreva "Dispositivo ", dispositivos[encontrado].nome, " desligado."

senao

    escreva "Comando inválido."

fimse

senao

    escreva "Dispositivo não encontrado."

fimse

fimfuncao

// Função para verificar se houve consumo de energia elevado em algum sensor.

funcao verificarConsumoElevado()

    real limiteElevado = 5.0 // Define um limite de consumo diário (em kWh) considerado elevado.

    inteiro numAlertas = 0 // Contador para o número de alertas gerados.

// Loop para verificar o consumo de cada sensor.

    para inteiro i de 0 ate NUM_SENSORES - 1 faca

        real consumoTotalDia = 0.0

// Verifica se já houve registro de consumo para o dia atual.

        se diaAtual > 0 entao

            consumoTotalDia <- consumoDiario[i][diaAtual - 1] // Obtém o consumo do sensor no dia atual.

// Se o consumo do dia atual exceder o limite elevado.

            se consumoTotalDia > limiteElevado entao

```



**// Verifica se ainda há espaço para mais alertas.**

se numAlertas < 10 entao

**// Cria uma mensagem de alerta e a armazena no vetor de alertas.**

alertas[numAlertas] <- "Alerta: Consumo elevado no sensor " + (i + 1) + " no dia " + diaAtual + " (" + consumoTotalDia + " kWh)."

**// Incrementa o contador de alertas.**

numAlertas <- numAlertas + 1

fimse

fimse

fimse

fimpara

**// Se houver algum alerta, exibe-os.**

se numAlertas > 0 entao

escreva "\n--- Alertas de Consumo Elevado ---"

**// Loop para exibir todos os alertas gerados.**

para inteiro i de 0 ate numAlertas - 1 faca

escreva alertas[i]

fimpara

fimse

fimfuncao

**// Função para gerar sugestões de economia de energia.**

funcao gerarSugestoesEconomia()

escreva "\n--- Sugestões para Economia de Energia ---"

escreva "1. Verifique se luzes e aparelhos estão desligados quando não estiverem em uso."

escreva "2. Utilize lâmpadas LED, que são mais eficientes."

escreva "3. Evite deixar aparelhos em modo stand-by."

escreva "4. Utilize a máquina de lavar roupa e a lava-louças com carga total."

escreva "5. Considere instalar painéis solares."

fimfuncao

**// Função para simular a coleta de dados de consumo dos sensores.**

**// Em um sistema real, esta função seria substituída pela leitura dos sensores.**

funcao simularConsumo()

**// Loop para simular o consumo de cada sensor.**

para inteiro i de 0 ate NUM\_SENSORES - 1 faca

**// Gera um valor aleatório para simular o consumo do sensor no dia atual.**

consumoDiario[i][diaAtual - 1] <- aleatorio(0, 3) + aleatorioReal()

fimpara

**// Após simular o consumo, verifica se houve algum consumo elevado.**

verificarConsumoElevado()

**// Avança para o próximo dia.**

diaAtual <- diaAtual + 1

**// Se o dia atual exceder 31, reinicia para o dia 1 do próximo mês (simulação).**

se diaAtual > 31 entao

```
diaAtual <- 1
```

```
fimse
```

```
fimfuncao
```

```
// --- Seção do Bloco Principal do Algoritmo ---
```

```
inicio
```

```
// Pré-cadastro de um usuário administrador (exemplo).
```

```
usuarios[0].nome <- "admin"
```

```
usuarios[0].senha <- "123456"
```

```
usuarios[0].nivelAcesso <- 2
```

```
numUsuarios <- 1
```

```
// Pré-cadastro de alguns dispositivos (relacionados aos sensores).
```

```
dispositivos[0].nome <- "Luz Quarto A"
```

```
dispositivos[0].idSensor <- 1
```

```
dispositivos[0].estado <- falso
```

```
dispositivos[0].consumoHora <- 0.05
```

```
numDispositivos <- 1
```

```
dispositivos[1].nome <- "Ar Condicionado Quarto A"
```

```
dispositivos[1].idSensor <- 1
```

```
dispositivos[1].estado <- falso
```

```
dispositivos[1].consumoHora <- 1.5
```

```
numDispositivos <- 2
```

```
dispositivos[2].nome <- "Luz Quarto B"
```

```
dispositivos[2].idSensor <- 2
```

```
dispositivos[2].estado <- falso
```

```
dispositivos[2].consumoHora <- 0.05
```

```
numDispositivos <- 3
```

```
dispositivos[3].nome <- "TV Sala"
```

```
dispositivos[3].idSensor <- 3
```

```
dispositivos[3].estado <- falso
```

```
dispositivos[3].consumoHora <- 0.2
```

```
numDispositivos <- 4
```

```
dispositivos[4].nome <- "Geladeira Cozinha"
```

```
dispositivos[4].idSensor <- 4
```

```
dispositivos[4].estado <- verdadeiro // Supondo que a geladeira está  
sempre ligada.
```

```
dispositivos[4].consumoHora <- 0.15
```

```
numDispositivos <- 5
```

```
dispositivos[5].nome <- "Bomba Piscina"
```

```
dispositivos[5].idSensor <- 5
```

```
dispositivos[5].estado <- falso
```

```
dispositivos[5].consumoHora <- 0.75
```

```
numDispositivos <- 6
```

**// Variáveis para a tela de login e menu principal.**

inteiro opcao

cadeia nomeUsuario, senhaUsuario

inteiro usuarioLogado = -1 **// Variável para armazenar o índice do usuário logado no vetor de usuários.**

**// --- Tela de Login ---**

escreva "--- Sistema de Casa Inteligente ---"

escreva "Digite seu nome de usuário: "

leia nomeUsuario

escreva "Digite sua senha: "

leia senhaUsuario

**// Loop para verificar se as credenciais do usuário correspondem a algum usuário cadastrado.**

para inteiro i de 0 ate numUsuarios - 1 faca

se usuarios[i].nome = nomeUsuario e usuarios[i].senha = senhaUsuario  
entao

usuarioLogado <- i **// Se as credenciais coincidem, armazena o índice do usuário.**

pare **// Sai do loop após encontrar o usuário.**

fimse

fimpara

**// Se o usuário foi logado com sucesso.**

se usuarioLogado != -1 entao

escreva "\nBem-vindo(a), ", usuarios[usuarioLogado].nome, "!"

**// --- Menu Principal ---**

repita

escreva "\n--- Menu ---"

escreva "1. Visualizar Consumo de Energia"

**// Opções de controle e cadastro são exibidas apenas para usuários com nível de acesso avançado.**

se usuarios[usuarioLogado].nivelAcesso = 2 entao

escreva "2. Controlar Dispositivos"

escreva "3. Cadastrar Novo Usuário"

escreva "4. Cadastrar Novo Dispositivo"

fimse

escreva "5. Ver Alertas de Consumo"

escreva "6. Ver Sugestões de Economia"

escreva "0. Sair"

escreva "Escolha uma opção: "

leia opcao

**// Estrutura de escolha para executar a ação correspondente à opção do menu.**

escolha opcao

caso 1:

exibirConsumoEnergia()

caso 2:

**// Permite controlar dispositivos apenas para usuários com nível de acesso avançado.**

se usuarios[usuarioLogado].nivelAcesso = 2 entao

    controlarDispositivo()

senao

    escreva "Acesso negado."

fimse

caso 3:

**// Permite cadastrar novos usuários apenas para usuários com nível de acesso avançado.**

se usuarios[usuarioLogado].nivelAcesso = 2 entao

    cadastrarUsuario()

senao

    escreva "Acesso negado."

fimse

caso 4:

**// Permite cadastrar novos dispositivos apenas para usuários com nível de acesso avançado.**

se usuarios[usuarioLogado].nivelAcesso = 2 entao

    cadastrarDispositivo()

senao

    escreva "Acesso negado."

fimse

caso 5:

    verificarConsumoElevado()

caso 6:

gerarSugestoesEconomia()

caso 0:

escreva "Saindo do sistema."

caso contrario:

escreva "Opção inválida."

fimescolha

**// Simula a passagem do tempo e a coleta de dados de consumo a cada interação do menu.**

simularConsumo()

**ate opcao = 0 // O loop continua até que o usuário escolha a opção sair (0).**

senao

escreva "Usuário ou senha incorretos."

fimse

fimalgoritmo

## **Documentação Detalhada:**

### **1. Seção de Declaração de Constantes:**

- **MAX\_DISPOSITIVOS:** Define o limite máximo de dispositivos que podem ser gerenciados pelo sistema. Isso ajuda a dimensionar o vetor de dispositivos.



- **MAX\_USUARIOS:** Define o limite máximo de usuários que podem ter acesso ao sistema. Isso dimensiona o vetor de usuários.
- **NUM\_SENSORES:** Define o número fixo de sensores na casa, cada um correspondendo a uma área específica (Quarto A, Quarto B, Sala, Cozinha, Piscina).

## **2. Seção de Definição de Tipos de Dados Estruturados (Registros):**

- **Dispositivo:** Define um tipo de dado composto (registro) para agrupar informações relacionadas a um dispositivo inteligente, como seu nome, o sensor ao qual está conectado, seu estado atual (ligado/desligado) e seu consumo de energia por hora.
- **Usuario:** Define um tipo de dado composto para armazenar informações de um usuário do sistema, incluindo nome, senha para autenticação e nível de acesso (para controlar quais funcionalidades o usuário pode acessar).

## **3. Seção de Declaração de Variáveis:**

- **dispositivos:** Um vetor (array) que pode armazenar até `MAX_DISPOSITIVOS`