

**UNIVERSIDADE MOGI DAS CRUZES
ENGENHARIA DE SOFTWARE**

**ARTHUR SILVA BARROS
MILENA KAORI IANATA
NICOLAS PINELLO PONTES
RAFAEL BUENO SILVA
VICTOR SIQUETTO VIEIRA DA SILVA**

**POWERHOME – SISTEMA INTELIGENTE DE MONITORAMENTO E
CONTROLE DE ENERGIA RESIDENCIAL**

**MOGI DAS CRUZES
2025**

UNIVERSIDADE MOGI DAS CRUZES
ENGENHARIA DE SOFTWARE

ARTHUR SILVA BARROS
MILENA KAORI IANATA
NICOLAS PINELLO PONTES
RAFAEL BUENO SILVA
VICTOR SIQUETTO VIEIRA DA SILVA

POWERHOME – SISTEMA INTELIGENTE DE MONITORAMENTO E
CONTROLE DE ENERGIA RESIDENCIAL

Trabalho de conclusão de curso para obtenção do
título de graduação em Engenharia de Software
apresentado à Universidade de Mogi das Cruzes –
UMC.

Orientador: Prof. Terigi Augusto

MOGI DAS CRUZES
2025

Tema do Projeto:

PowerHome – Sistema Inteligente de Monitoramento e Controle de Energia Residencial.

Equipe:

A equipe é composta por 5 membros, com as seguintes funções:

Rafael Bueno Silva	PO (Product Owner) e Gestor de Projeto
Arthur Silva Barros	Scrum Master e Analista de Sistemas
Victor Siquetto Vieira Da Silva	Desenvolvedor Full-Stack (Backend e Frontend)
Nicolas Pinello Pontes	Desenvolvedor IoT (Hardware e Firmware)
Milena Kaori Ianata	Designer UI/UX e QA (Analista de Qualidade)

Objetivos:**Objetivos Gerais:**

- Desenvolver uma solução tecnológica integrada (hardware e software) que permita aos usuários residenciais monitorar e gerenciar ativamente seu consumo de energia elétrica.
- Capacitar os usuários com dados claros e acionáveis para promover a redução de desperdícios e a adoção de hábitos de consumo mais conscientes e econômicos.

Objetivos Específicos:

- Construir um protótipo funcional do dispositivo IoT utilizando um microcontrolador ESP32, um sensor de corrente SCT-013 e um módulo relé.
- Implementar um backend robusto para receber, processar e armazenar os dados de consumo de forma segura e eficiente.
- Desenvolver um aplicativo mobile multiplataforma (React Native ou Flutter) e um painel web (React.js) como interfaces para o usuário.
- Implementar funcionalidades de controle remoto (ligar/desligar) e automação de aparelhos.
- Criar um sistema de relatórios visuais (diário, semanal, mensal) e alertas personalizáveis sobre o consumo.

Descrição do Minimundo:

O minimundo a ser modelado é o de uma residência inteligente focada na gestão de energia. Nele, um usuário se cadastra em uma plataforma para gerenciar sua casa. Este usuário pode registrar múltiplos dispositivos de hardware, que são acoplados a eletrodomésticos para medir o consumo de energia. Para organização, o usuário pode agrupar estes dispositivos em diferentes cômodos. Cada dispositivo envia, periodicamente, leituras de consumo (em Watts) para o sistema, que as armazena com data e hora. O usuário pode interagir com o sistema para visualizar esses dados, controlar os dispositivos remotamente (ligar/desligar), criar automações baseadas em horários e dias da semana, e configurar alertas para ser notificado caso o consumo de um aparelho ultrapasse um determinado limite. O banco de dados deve, portanto, suportar o cadastro de usuários, o gerenciamento de seus dispositivos e cômodos, e o armazenamento de um grande volume de dados de consumo, além das regras de automação e alerta.

Requisitos Funcionais de Usuário

Tela	Código	Descrição
Login / Cadastro	RF01	Usuário pode se cadastrar no sistema com e-mail e senha.
	RF02	Usuário pode efetuar login no sistema.
	RF03	Usuário pode recuperar sua senha caso a esqueça.
Gerenciamento	RF04	Usuário pode adicionar um novo dispositivo PowerHome à sua conta.
	RF05	Usuário pode renomear e remover um dispositivo existente.
	RF06	Usuário pode agrupar dispositivos por cômodos (ex: Sala, Cozinha).
Dashboard	RF07	Usuário pode visualizar o consumo em tempo real de um aparelho específico.
	RF08	Usuário pode visualizar o consumo consolidado por cômodo e o total da residência.
	RF09	Usuário pode ligar ou desligar um aparelho remotamente.
Relatórios	RF10	Usuário pode visualizar o histórico de consumo em gráficos (diário, semanal, mensal).

	RF11	Usuário pode comparar o consumo entre diferentes períodos (ex: esta semana vs. semana passada).
Automação	RF12	Usuário pode agendar horários para ligar/desligar um aparelho.
Alertas	RF13	Usuário pode definir um limite de consumo (em R\$ ou Watts) para receber um alerta.
	RF14	Usuário pode visualizar uma notificação no aplicativo quando um alerta for disparado.
Tarifação	RF15	Usuário pode inserir/atualizar sua tarifa de energia (R\$/kWh).

Requisitos Funcionais de Sistema

Código	Descrição
RF16	O sistema deve permitir o armazenamento seguro e gerenciamento dos dados de cadastro.
RF17	O sistema deve validar as credenciais do usuário e gerenciar a sessão de acesso via token.
RF18	O sistema deve fornecer um mecanismo seguro para redefinição de senha.
RF19	O sistema deve prover um processo seguro de pareamento para novos dispositivos IoT.
RF20	O sistema deve suportar a criação, edição, exclusão e agrupamento de dispositivos.
RF21	O sistema deve receber, processar e disponibilizar os dados do dispositivo IoT em tempo real.
RF22	O sistema deve agregar os dados de consumo para exibir totais por cômodo e residência.
RF23	O sistema deve transmitir um comando de ligar/desligar para o dispositivo IoT correspondente.
RF24	O sistema deve processar e armazenar os dados históricos para a geração dos relatórios.

RF25	O sistema deve realizar cálculos comparativos sobre os dados históricos de consumo.
RF26	O sistema deve armazenar e executar as regras de agendamento de automação.
RF27	O sistema deve monitorar o consumo e disparar uma notificação push quando um limite for atingido.
RF28	O sistema deve armazenar a tarifa do usuário e usá-la para todos os cálculos de custo.

Requisitos Não Funcionais de Usuário

Categoria	Código	Descrição
Desempenho	RNF01	A visualização do consumo em tempo real deve ser atualizada em até 5 segundos após a medição.
	RNF02	O comando de ligar/desligar um aparelho deve ser executado em no máximo 3 segundos.
	RNF03	As consultas aos relatórios com dados de até 3 meses devem carregar em menos de 4 segundos.
Usabilidade	RNF04	A interface do aplicativo deve ser intuitiva, seguindo as diretrizes de design do Android e iOS.
	RNF05	O processo de adicionar um novo dispositivo não deve exceder 5 passos guiados no aplicativo.
	RNF06	A terminologia e os textos do sistema devem ser claros e de fácil compreensão para um usuário leigo.
Compatibilidade	RNF07	O aplicativo mobile deve ser compatível com as 3 últimas versões dos sistemas Android e iOS.

Requisitos Não Funcionais de Sistema

Categoria	Código	Descrição
Confiabilidade	RNF08	O backend do sistema deve ter uma disponibilidade (uptime) de 99.5% (considerando o escopo de protótipo).

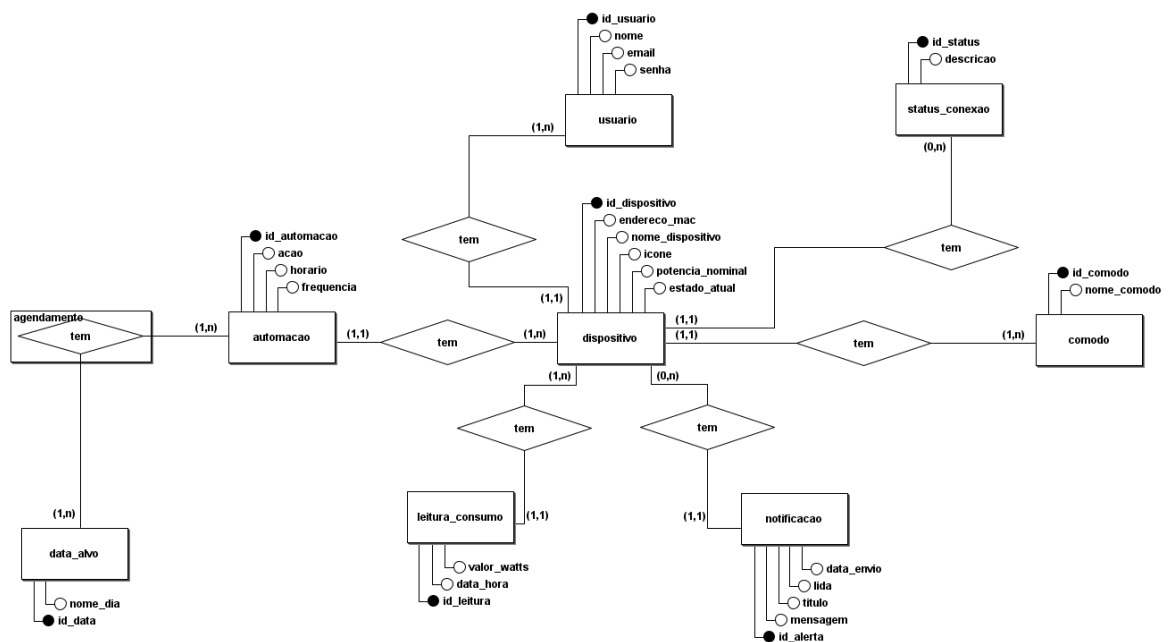
	RNF09	O dispositivo IoT deve se reconectar automaticamente à rede Wi-Fi em caso de perda de conexão.
	RNF10	O sistema deve realizar backups diários do banco de dados para prevenir perda de dados.
Segurança	RNF11	Toda a comunicação entre cliente, servidor e dispositivo IoT deve ser criptografada (HTTPS e MQTT sobre TLS).
	RNF12	As senhas dos usuários devem ser armazenadas no banco de dados usando hashing e salt.
	RNF13	O sistema deve ser protegido contra vulnerabilidades comuns da web, como SQL Injection e XSS.
Escalabilidade	RNF14	A arquitetura do backend deve suportar, inicialmente (no protótipo), até 1.000 dispositivos conectados simultaneamente.

Regras de Negócio

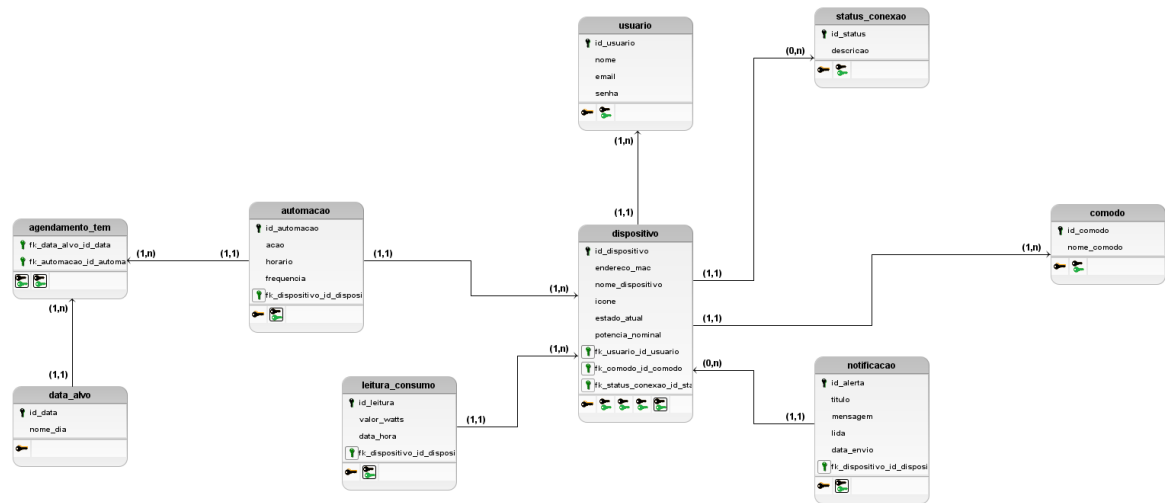
Categoria	Descrição
Gestão de Cadastro e Unicidade	O e-mail de um usuário deve ser único em todo o sistema. Não podem existir dois usuários com o mesmo e-mail.
	Um dispositivo de hardware (identificado por seu MAC Address) deve ser único em todo o sistema.
Relacionamentos e Posse (Hierarquia)	Um dispositivo deve pertencer a um, e somente um, usuário.
	Um usuário pode ter vários dispositivos cadastrados.
	Um cômodo pertence a um, e somente um, usuário.
	Um dispositivo pode ser alocado em no máximo um cômodo por vez (ou em nenhum).
Lógica de Automação e Alertas	Uma regra de automação é específica para um único dispositivo.
	Um dispositivo pode ter no máximo um alerta de consumo ativo por vez.

	Não é permitido criar duas regras de automação conflitantes para o mesmo dispositivo (ex: ligar às 22:00 e desligar às 22:00 no mesmo dia).
Integridade dos Dados e Operação	Cada leitura de consumo deve ser obrigatoriamente associada a um dispositivo existente.
	O valor de uma leitura de consumo (em Watts) não pode ser negativo.
	Um usuário não pode excluir um cômodo se houver dispositivos associados a ele. É necessário primeiro mover ou excluir os dispositivos.
	Um comando de controle (ligar/desligar) só pode ser enviado a um dispositivo cujo status de conexão esteja "online".

DER



Modelo Lógico



Dicionário do Banco de Dados

Tabela	usuário			
Descrição	Armazena os dados cadastrais de quem utiliza o sistema.			
Observação				
Campo				
Atributo	Descrição	Tipo de Dado	Tamanho	Restrição
id_usuario	Identificador único do usuário	INT	-	PK, Auto Increment
nome	Nome completo do usuário	VARCHAR	100	Not Null
email	Endereço de e-mail para login	VARCHAR	100	Not Null, Unique
senha	Hash da senha de acesso	VARCHAR	255	Not Null

Tabela	comodo			
Descrição	Representa os ambientes físicos da residência (ex: Sala, Quarto).			
Observação	fk_usuario_id - chave estrangeira de usuario			
Campo				
Atributo	Descrição	Tipo de Dado	Tamanho	Restrição
id_comodo	Identificador único do cômodo	INT	-	PK, Auto Increment
nome_comodo	Nome do ambiente	VARCHAR	50	Not Null
icone	Identificador visual do ambiente	VARCHAR	50	Null
fk_usuario_id	Chave estrangeira do proprietário	INT	-	FK, Not Null

Tabela	status_conexao			
Descrição	Tabela de domínio com os estados possíveis de um dispositivo.			
Observação				
Campo				
Atributo	Descrição	Tipo de Dado	Tamanho	Restrição
id_status	Identificador único do status	INT	-	PK, Auto Increment
descricao	Texto descritivo do estado	VARCHAR	20	Not Null, Unique

Tabela	dispositivo			
Descrição	Entidade central que representa o hardware IoT instalado.			
Observação	fk_usuario_id - chave estrangeira de usuario / fk_comodo_id - chave estrangeira de comodo / fk_status_id - chave estrangeira de status_conexao			
Campo				
Atributo	Descrição	Tipo de Dado	Tamanho	Restrição
id_dispositivo	Identificador único do dispositivo	INT	-	PK, Auto Increment
nome_dispositivo	Nome personalizado (ex: TV da Sala)	VARCHAR	50	Not Null
endereco_mac	Endereço físico da placa de rede	VARCHAR	17	Not Null, Unique
icone	Classe do ícone visual para a UI	VARCHAR	50	Not Null
potencia_nominal	Potência padrão em Watts	INT	-	Not Null
estado_atual	Estado do relé (0=Desligado, 1=Ligado)	BOOLEAN	-	Not Null, Default 0
fk_usuario_id	Usuário proprietário	INT	-	FK, Not Null
fk_comodo_id	Cômodo onde está instalado	INT	-	FK, Not Null
fk_status_id	Status atual de conexão	INT	-	FK, Not Null

Tabela	leitura_consumo			
Descrição	Histórico (log) de consumo de energia enviado pelos dispositivos.			
Observação	fk_dispositivo_id - chave estrangeira de dispositivo			
Campo				
Atributo	Descrição	Tipo de Dado	Tamanho	Restrição
id_leitura	Identificador único da leitura	BIGINT	-	PK, Auto Increment
valor_watts	Consumo medido em Watts	DECIMAL	10,2	Not Null
data_hora	Momento exato da medição	DATETIME	-	Not Null
fk_dispositivo_id	Dispositivo que gerou a leitura	INT	-	FK, Not Null

Tabela	notificacao			
Descrição	Histórico de alertas e avisos gerados pelo sistema.			
Observação	fk_dispositivo_id - chave estrangeira de dispositivo			
Campo				
Atributo	Descrição	Tipo de Dado	Tamanho	Restrição
id_alerta	Identificador único da notificação	INT	-	PK, Auto Increment
titulo	Título curto do aviso	VARCHAR	50	Not Null
mensagem	Texto detalhado do aviso	VARCHAR	255	Not Null
lida	Status de leitura (0=Não, 1=Sim)	BOOLEAN	-	Not Null, Default 0
data_envio	Data e hora do evento	DATETIME	-	Not Null
fk_dispositivo_id	Dispositivo relacionado ao aviso	INT	-	FK, Not Null

Tabela	automacao			
Descrição	Define as regras de agendamento criadas pelo usuário.			
Observação	fk_dispositivo_id - chave estrangeira de dispositivo			
Campo				
Atributo	Descrição	Tipo de Dado	Tamanho	Restrição
id_automacao	Identificador único da regra	INT	-	PK, Auto Increment
acao	Ação ('LIGAR' ou 'DESLIGAR')	VARCHAR	10	Not Null
horario	Horário de execução (HH:MM:SS)	TIME	-	Not Null
frequencia	Tipo ('UMA_VEZ', 'DIARIO')	VARCHAR	20	Not Null
fk_dispositivo_id	Dispositivo a ser controlado	INT	-	FK, Not Null

Tabela	data_alvo			
Descrição	Tabela auxiliar para definir dias da semana ou datas específicas.			
Observação				
Campo				
Atributo	Descrição	Tipo de Dado	Tamanho	Restrição
id_data	Identificador único	INT	-	PK, Auto Increment
nome_dia	Nome do dia da semana	VARCHAR	15	Null
data_especifica	Data específica para eventos únicos	DATE	-	Null

Tabela	agendamento			
Descrição	Tabela de ligação N:N entre Automação e Data Alvo.			
Observação	fk_automacao_id - chave estrangeira de automacao / fk_data_alvo_id - chave estrangeira de data_alvo			
Campo				
Atributo	Descrição	Tipo de Dado	Tamanho	Restrição
fk_automacao_id	Referência à regra de automação	INT	-	FK
fk_data_alvo_id	Referência ao dia ou data	INT	-	FK

Script de criação do banco de dados e das tabelas

- `CREATE DATABASE powerhome;`
- `USE powerhome;`

```
CREATE TABLE usuario (
    id_usuario INT AUTO_INCREMENT,
    nome VARCHAR(100) NOT NULL,
    email VARCHAR(100) NOT NULL UNIQUE,
    senha VARCHAR(255) NOT NULL,
    ativo INT DEFAULT 1,
    CONSTRAINT pk_usuario PRIMARY KEY (id_usuario),
    CONSTRAINT ck_usuario_ativo CHECK (ativo IN (0, 1))
);
```

```
CREATE TABLE comodo (
    id_comodo INT AUTO_INCREMENT,
    nome_comodo VARCHAR(50) NOT NULL,
    icone VARCHAR(50),
    fk_usuario_id INT NOT NULL,
    ativo INT DEFAULT 1,
    CONSTRAINT pk_comodo PRIMARY KEY (id_comodo),
    CONSTRAINT fk_comodo_usuario FOREIGN KEY (fk_usuario_id)
        REFERENCES usuario(id_usuario) ON DELETE CASCADE ON UPDATE CASCADE,
    CONSTRAINT ck_comodo_ativo CHECK (ativo IN (0, 1))
);
```

```
CREATE TABLE status_conexao (
    id_status INT AUTO_INCREMENT,
    descricao VARCHAR(20) NOT NULL UNIQUE,
    ativo INT DEFAULT 1,
    CONSTRAINT pk_status PRIMARY KEY (id_status),
    CONSTRAINT ck_status_ativo CHECK (ativo IN (0, 1))
);
```

```
CREATE TABLE dispositivo (
    id_dispositivo INT AUTO_INCREMENT,
    nome_dispositivo VARCHAR(50) NOT NULL,
    endereco_mac VARCHAR(17) NOT NULL UNIQUE,
    icone VARCHAR(50) NOT NULL,
    potencia_nominal INT NOT NULL,
    estado_atual TINYINT(1) NOT NULL DEFAULT 0,
    fk_usuario_id INT NOT NULL,
    fk_comodo_id INT NOT NULL,
    fk_status_id INT NOT NULL,
    ativo INT DEFAULT 1,
    CONSTRAINT pk_dispositivo PRIMARY KEY (id_dispositivo),
    CONSTRAINT fk_dispositivo_usuario FOREIGN KEY (fk_usuario_id) REFERENCES usuario(id_usuario) ON DELETE CASCADE ON UPDATE CASCADE,
    CONSTRAINT fk_dispositivo_comodo FOREIGN KEY (fk_comodo_id) REFERENCES comodo(id_comodo) ON UPDATE CASCADE,
    CONSTRAINT fk_dispositivo_status FOREIGN KEY (fk_status_id) REFERENCES status_conexao(id_status) ON UPDATE CASCADE,
    CONSTRAINT ck_dispositivo_ativo CHECK (ativo IN (0, 1)),
    CONSTRAINT ck_dispositivo_potencia CHECK (potencia_nominal > 0)
);
```

```

● CREATE TABLE leitura_consumo (
    id_leitura BIGINT AUTO_INCREMENT,
    valor_watts DECIMAL(10,2) NOT NULL,
    data_hora TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
    fk_dispositivo_id INT NOT NULL,
    ativo INT DEFAULT 1,
    CONSTRAINT pk_leitura PRIMARY KEY (id_leitura),
    CONSTRAINT fk_leitura_dispositivo FOREIGN KEY (fk_dispositivo_id) REFERENCES dispositivo(id_dispositivo) ON DELETE CASCADE ON UPDATE CASCADE,
    CONSTRAINT ck_leitura_ativo CHECK (ativo IN (0, 1)),
    CONSTRAINT ck_leitura_positiva CHECK (valor_watts >= 0)
);

● CREATE TABLE notificacao (
    id_alerta INT AUTO_INCREMENT,
    titulo VARCHAR(50) NOT NULL,
    mensagem VARCHAR(255) NOT NULL,
    lida TINYINT(1) NOT NULL DEFAULT 0,
    data_envio TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
    fk_dispositivo_id INT NOT NULL,
    ativo INT DEFAULT 1,
    CONSTRAINT pk_notificacao PRIMARY KEY (id_alerta),
    CONSTRAINT fk_notificacao_dispositivo FOREIGN KEY (fk_dispositivo_id) REFERENCES dispositivo(id_dispositivo) ON DELETE CASCADE ON UPDATE CASCADE,
    CONSTRAINT ck_notificacao_ativo CHECK (ativo IN (0, 1))
);

● CREATE TABLE automacao (
    id_automacao INT AUTO_INCREMENT,
    acao VARCHAR(10) NOT NULL,
    horario TIME NOT NULL,
    frequencia VARCHAR(20) NOT NULL,
    fk_dispositivo_id INT NOT NULL,
    ativo INT DEFAULT 1,
    CONSTRAINT pk_automacao PRIMARY KEY (id_automacao),
    CONSTRAINT fk_automacao_dispositivo FOREIGN KEY (fk_dispositivo_id) REFERENCES dispositivo(id_dispositivo) ON DELETE CASCADE ON UPDATE CASCADE,
    CONSTRAINT ck_automacao_ativo CHECK (ativo IN (0, 1))
);

● CREATE TABLE data_alvo (
    id_data INT AUTO_INCREMENT,
    nome_dia VARCHAR(15),
    data_especifica DATE,
    ativo INT DEFAULT 1,
    CONSTRAINT pk_data_alvo PRIMARY KEY (id_data),
    CONSTRAINT ck_data_ativo CHECK (ativo IN (0, 1))
);

● CREATE TABLE agendamento (
    fk_automacao_id INT NOT NULL,
    fk_data_alvo_id INT NOT NULL,
    ativo INT DEFAULT 1,
    CONSTRAINT pk_agendamento PRIMARY KEY (fk_automacao_id, fk_data_alvo_id),
    CONSTRAINT fk_agendamento_automacao FOREIGN KEY (fk_automacao_id) REFERENCES automacao(id_automacao) ON DELETE CASCADE ON UPDATE CASCADE,
    CONSTRAINT fk_agendamento_data FOREIGN KEY (fk_data_alvo_id) REFERENCES data_alvo(id_data) ON UPDATE CASCADE,
    CONSTRAINT ck_agendamento_ativo CHECK (ativo IN (0, 1))
);

```

Script de povoamento de todas as tabelas com no mínimo 10 tuplas em cada tabela

```

● INSERT INTO usuario (nome, email, senha) VALUES
('Rafael Bueno', 'rafael@powerhome.com', 'hash123'), ('Milena Kaori', 'mlena@powerhome.com', 'hash123'),
('Victor Siquetto', 'victor@powerhome.com', 'hash123'), ('Nicolas Pontes', 'nicolas@powerhome.com', 'hash123'),
('Arthur Barros', 'arthur@powerhome.com', 'hash123'), ('João Silva', 'joao@email.com', 'senha456'),
('Maria Oliveira', 'maria@email.com', 'senha789'), ('Carlos Souza', 'carlos@email.com', 'senha321'),
('Ana Pereira', 'ana@email.com', 'senha654'), ('Lucas Lima', 'lucas@email.com', 'senha987');

● INSERT INTO comodo (nome_comodo, icone, fk_usuario_id) VALUES
('Sala de Estar', 'fa-couch', 1), ('Quarto Principal', 'fa-bed', 1), ('Cozinha', 'fa-utensils', 1),
('Escritório', 'fa-laptop', 2), ('Sala de Jantar', 'fa-utensils', 2), ('Garagem', 'fa-car', 3),
('Jardim', 'fa-leaf', 3), ('Varanda', 'fa-sun', 4), ('Quarto Hóspedes', 'fa-bed', 5), ('Banheiro Social', 'fa-bath', 5);

● INSERT INTO status_conexao (descricao) VALUES
('Online'), ('Offline'), ('Erro'), ('Sincronizando'), ('Em Manutenção'),
('Atualizando'), ('Bateria Fraca'), ('Pareando'), ('Desabilitado'), ('Desconhecido');

● INSERT INTO dispositivo (nome_dispositivo, endereco_mac, icone, potencia_nominal, estado_atual, fk_usuario_id, fk_comodo_id, fk_status_id) VALUES
('TV da Sala', 'AA:BB:CC:00:01', 'fa-tv', 150, 1, 1, 1, 1), ('Ar Condicionado', 'AA:BB:CC:00:02', 'fa-wind', 2200, 0, 1, 2, 2),
('Geladeira', 'AA:BB:CC:00:03', 'fa-plug', 300, 1, 1, 3, 1), ('Lâmpada Escritório', 'AA:BB:CC:00:04', 'fa-lightbulb', 15, 1, 2, 4, 1),
('Cafeteira', 'AA:BB:CC:00:05', 'fa-coffee', 800, 0, 2, 5, 1), ('Portão Eletrônico', 'AA:BB:CC:00:06', 'fa-dungeon', 50, 1, 3, 6, 1),
('Cortina Smart', 'AA:BB:CC:00:07', 'fa-blinds', 30, 0, 3, 7, 3), ('Câmera Varanda', 'AA:BB:CC:00:08', 'fa-video', 10, 1, 4, 8, 1),
('Ventilador', 'AA:BB:CC:00:09', 'fa-fan', 60, 0, 5, 9, 1), ('Aquecedor', 'AA:BB:CC:00:10', 'fa-fire', 1500, 1, 5, 10, 4);

```

- INSERT INTO leitura_consumo (valor_watts, data_hora, fk_dispositivo_id) VALUES**
 (145.50, '2025-11-28 10:00:00', 1), (148.20, '2025-11-28 10:05:00', 1), (290.00, '2025-11-28 10:00:00', 3),
 (305.10, '2025-11-28 11:00:00', 3), (15.00, '2025-11-28 18:00:00', 4), (15.00, '2025-11-28 19:00:00', 4),
 (800.00, '2025-11-28 08:00:00', 5), (50.00, '2025-11-28 12:30:00', 6), (10.00, '2025-11-28 00:00:00', 8),
 (1500.0, '2025-11-28 22:00:00', 10);
- INSERT INTO notificacao (titulo, mensagem, lida, fk_dispositivo_id) VALUES**
 ('Consumo Alto', 'A TV consumiu acima da média.', 0, 1), ('Dispositivo Offline', 'O Ar Condicionado perdeu conexão.', 0, 2),
 ('Porta Aberta', 'A Geladeira ficou aberta por muito tempo.', 1, 3), ('Lâmpada Ligada', 'Lâmpada acessa por mais de 10h.', 0, 4),
 ('Café Pronto', 'Sua cafeteira terminou o ciclo.', 1, 5), ('Portão Acionado', 'O portão foi aberto manualmente.', 0, 6),
 ('Erro Crítico', 'Falha no motor da cortina.', 0, 7), ('Movimento Detectado', 'Câmera detectou movimento na varanda.', 1, 8),
 ('Temperatura Alta', 'O Aquecedor está ligado há 5 horas.', 0, 10), ('Bem-vindo', 'Novo dispositivo Ventilador configurado.', 1, 9);
- INSERT INTO automacao (acao, horario, frequencia, fk_dispositivo_id) VALUES**
 ('LIGAR', '18:00:00', 'DIARIO', 1), ('DESLIGAR', '23:00:00', 'DIARIO', 1), ('LIGAR', '22:00:00', 'DIARIO', 2),
 ('DESLIGAR', '06:00:00', 'DIARIO', 2), ('LIGAR', '07:00:00', 'SEMANAL', 5), ('LIGAR', '18:30:00', 'DIARIO', 4),
 ('DESLIGAR', '22:30:00', 'DIARIO', 4), ('LIGAR', '20:00:00', 'UMA_VEZ', 10), ('DESLIGAR', '08:00:00', 'DIARIO', 8),
 ('LIGAR', '18:00:00', 'DIARIO', 8);
- INSERT INTO data_alvo (nome_dia, data_especifica) VALUES**
 ('Segunda', NULL), ('Terça', NULL), ('Quarta', NULL), ('Quinta', NULL), ('Sexta', NULL), ('Sabado', NULL), ('Domingo', NULL),
 (NULL, '2025-12-25'), (NULL, '2025-12-31'), (NULL, '2025-01-01');
- INSERT INTO agendamento (fk_automacao_id, fk_data_alvo_id) VALUES**
 (1, 1), (1, 2), (1, 3), (1, 4), (1, 5), (3, 6), (3, 7), (5, 1), (8, 8), (8, 9);

Para cada tabela do banco deverá ser criada:

Uma procedure para a inclusão de novos registros;

Uma procedure para atualização dos registros;

Uma procedure para exclusão lógica dos registros;

```

DELIMITER //
• CREATE PROCEDURE sp_incluir_usuario(IN p_nome VARCHAR(100), IN p_email VARCHAR(100), IN p_senha VARCHAR(255))
• BEGIN
    INSERT INTO usuario (nome, email, senha) VALUES (p_nome, p_email, p_senha);
  END //
DELIMITER ;

• CALL sp_incluir_usuario('Teste User', 'teste@demo.com', '123456');
• SELECT * FROM usuario WHERE email = 'teste@demo.com';

DELIMITER //
• CREATE PROCEDURE sp_atualizar_usuario(IN p_id INT, IN p_nome VARCHAR(100), IN p_senha VARCHAR(255))
• BEGIN
    UPDATE usuario SET nome = p_nome, senha = p_senha WHERE id_usuario = p_id;
  END //
DELIMITER ;

• CALL sp_atualizar_usuario((SELECT MAX(id_usuario) FROM usuario), 'Teste User Alterado', '654321');
• SELECT * FROM usuario WHERE email = 'teste@demo.com';

DELIMITER //
• CREATE PROCEDURE sp_excluir_usuario(IN p_id INT)
• BEGIN
    UPDATE usuario SET ativo = 0 WHERE id_usuario = p_id;
  END //
DELIMITER ;

• CALL sp_excluir_usuario((SELECT MAX(id_usuario) FROM usuario));
• SELECT * FROM usuario WHERE email = 'teste@demo.com';

DELIMITER //
• CREATE PROCEDURE sp_incluir_comodo(IN p_nome VARCHAR(50), IN p_icone VARCHAR(50), IN p_user_id INT)
• BEGIN
    INSERT INTO comodo (nome_comodo, icone, fk_usuario_id) VALUES (p_nome, p_icone, p_user_id);
  END //
DELIMITER ;

• CALL sp_incluir_comodo('Teste Comodo', 'fa-box', 1);
• SELECT * FROM comodo WHERE nome_comodo = 'Teste Comodo';

DELIMITER //
• CREATE PROCEDURE sp_atualizar_comodo(IN p_id INT, IN p_nome VARCHAR(50), IN p_icone VARCHAR(50))
• BEGIN
    UPDATE comodo SET nome_comodo = p_nome, icone = p_icone WHERE id_comodo = p_id;
  END //
DELIMITER ;

```

```

• CALL sp_atualizar_comodo((SELECT MAX(id_comodo) FROM comodo), 'Teste Comodo Alterado', 'fa-box-open');
• SELECT * FROM comodo WHERE nome_comodo = 'Teste Comodo Alterado';

DELIMITER //
• CREATE PROCEDURE sp_excluir_comodo(IN p_id INT)
BEGIN
    UPDATE comodo SET ativo = 0 WHERE id_comodo = p_id;
END //
DELIMITER ;

• CALL sp_excluir_comodo((SELECT MAX(id_comodo) FROM comodo));
• SELECT * FROM comodo WHERE nome_comodo = 'Teste Comodo Alterado';

DELIMITER //
• CREATE PROCEDURE sp_incluir_status(IN p_desc VARCHAR(20))
BEGIN
    INSERT INTO status_conexao (descricao) VALUES (p_desc);
END //
DELIMITER ;

• CALL sp_incluir_status('Teste Status');
• SELECT * FROM status_conexao WHERE descricao = 'Teste Status';

DELIMITER //
• CREATE PROCEDURE sp_atualizar_status(IN p_id INT, IN p_desc VARCHAR(20))
BEGIN
    UPDATE status_conexao SET descricao = p_desc WHERE id_status = p_id;
END //
DELIMITER ;

• CALL sp_atualizar_status((SELECT MAX(id_status) FROM status_conexao), 'Teste Status Alt');
• SELECT * FROM status_conexao WHERE descricao = 'Teste Status Alt';

DELIMITER //
• CREATE PROCEDURE sp_excluir_status(IN p_id INT)
BEGIN
    UPDATE status_conexao SET ativo = 0 WHERE id_status = p_id;
END //
DELIMITER ;

• CALL sp_excluir_status((SELECT MAX(id_status) FROM status_conexao));
• SELECT * FROM status_conexao WHERE descricao = 'Teste Status Alt';

DELIMITER //
• CREATE PROCEDURE sp_incluir_dispositivo(IN p_nome VARCHAR(50), IN p_mac VARCHAR(17), IN p_icone VARCHAR(50), IN p_potencia INT, IN p_user_id INT, IN p_comodo_id INT, IN p_status_id INT)
BEGIN
    INSERT INTO dispositivo (nome_dispositivo, endereco_mac, icone, potencia_nominal, fk_usuario_id, fk_comodo_id, fk_status_id)
    VALUES (p_nome, p_mac, p_icone, p_potencia, p_user_id, p_comodo_id, p_status_id);
END //
DELIMITER ;

• CALL sp_incluir_dispositivo('Teste Disp', 'TT:TT:TT:00:00', 'fa-test', 50, 1, 1, 1);
• SELECT * FROM dispositivo WHERE endereco_mac = 'TT:TT:TT:00:00';

DELIMITER //
• CREATE PROCEDURE sp_atualizar_dispositivo(IN p_id INT, IN p_nome VARCHAR(50), IN p_potencia INT)
BEGIN
    UPDATE dispositivo SET nome_dispositivo = p_nome, potencia_nominal = p_potencia WHERE id_dispositivo = p_id;
END //
DELIMITER ;

• CALL sp_atualizar_dispositivo((SELECT MAX(id_dispositivo) FROM dispositivo), 'Teste Disp Alterado', 100);
• SELECT * FROM dispositivo WHERE endereco_mac = 'TT:TT:TT:00:00';

DELIMITER //
• CREATE PROCEDURE sp_excluir_dispositivo(IN p_id INT)
BEGIN
    UPDATE dispositivo SET ativo = 0 WHERE id_dispositivo = p_id;
END //
DELIMITER ;

• CALL sp_excluir_dispositivo((SELECT MAX(id_dispositivo) FROM dispositivo));
• SELECT * FROM dispositivo WHERE endereco_mac = 'TT:TT:TT:00:00';

DELIMITER //
• CREATE PROCEDURE sp_incluir_leitura(IN p_valor DECIMAL(10,2), IN p_disp_id INT)
BEGIN
    INSERT INTO leitura_consumo (valor_watts, fk_dispositivo_id) VALUES (p_valor, p_disp_id);
END //
DELIMITER ;

• CALL sp_incluir_leitura(99.9, 1);
• SELECT * FROM leitura_consumo WHERE valor_watts = 99.9;

DELIMITER //
• CREATE PROCEDURE sp_atualizar_leitura(IN p_id BIGINT, IN p_valor DECIMAL(10,2))
BEGIN
    UPDATE leitura_consumo SET valor_watts = p_valor WHERE id_leitura = p_id;
END //
DELIMITER ;

• CALL sp_atualizar_leitura((SELECT MAX(id_leitura) FROM leitura_consumo), 88.8);
• SELECT * FROM leitura_consumo WHERE valor_watts = 88.8;

DELIMITER //
• CREATE PROCEDURE sp_excluir_leitura(IN p_id BIGINT)
BEGIN
    UPDATE leitura_consumo SET ativo = 0 WHERE id_leitura = p_id;
END //
DELIMITER ;

• CALL sp_excluir_leitura((SELECT MAX(id_leitura) FROM leitura_consumo));
• SELECT * FROM leitura_consumo WHERE valor_watts = 88.8;

```

```

DELIMITER //
• CREATE PROCEDURE sp_incluir_notificacao(IN p_titulo VARCHAR(50), IN p_msg VARCHAR(255), IN p_disp_id INT)
• BEGIN
    INSERT INTO notificacao (titulo, mensagem, fk_dispositivo_id) VALUES (p_titulo, p_msg, p_disp_id);
• END //
DELIMITER ;

• CALL sp_incluir_notificacao('Teste Notif', 'Mensagem teste', 1);
• SELECT * FROM notificacao WHERE titulo = 'Teste Notif';

DELIMITER //
• CREATE PROCEDURE sp_atualizar_notificacao(IN p_id INT, IN p_lida TINYINT(1))
• BEGIN
    UPDATE notificacao SET lida = p_lida WHERE id_alerta = p_id;
• END //
DELIMITER ;

• CALL sp_atualizar_notificacao((SELECT MAX(id_alerta) FROM notificacao), 1);
• SELECT * FROM notificacao WHERE titulo = 'Teste Notif';

DELIMITER //
• CREATE PROCEDURE sp_excluir_notificacao(IN p_id INT)
• BEGIN
    UPDATE notificacao SET ativo = 0 WHERE id_alerta = p_id;
• END //
DELIMITER ;

• CALL sp_excluir_notificacao((SELECT MAX(id_alerta) FROM notificacao));
• SELECT * FROM notificacao WHERE titulo = 'Teste Notif';

DELIMITER //
• CREATE PROCEDURE sp_incluir_automacao(IN p_acao VARCHAR(10), IN p_hora TIME, IN p_freq VARCHAR(20), IN p_disp_id INT)
• BEGIN
    INSERT INTO automacao (acao, horario, frequencia, fk_dispositivo_id) VALUES (p_acao, p_hora, p_freq, p_disp_id);
• END //
DELIMITER ;

• CALL sp_incluir_automacao('TESTE', '12:00:00', 'DIARIO', 1);
• SELECT * FROM automacao WHERE acao = 'TESTE';

DELIMITER //
• CREATE PROCEDURE sp_atualizar_automacao(IN p_id INT, IN p_hora TIME, IN p_freq VARCHAR(20))
• BEGIN
    UPDATE automacao SET horario = p_hora, frequencia = p_freq WHERE id_automacao = p_id;
• END //
DELIMITER ;

• CALL sp_atualizar_automacao((SELECT MAX(id_automacao) FROM automacao), '13:00:00', 'SEMANAL');
• SELECT * FROM automacao WHERE acao = 'TESTE';

DELIMITER //
• CREATE PROCEDURE sp_excluir_automacao(IN p_id INT)
• BEGIN
    UPDATE automacao SET ativo = 0 WHERE id_automacao = p_id;
• END //
DELIMITER ;

• CALL sp_excluir_automacao((SELECT MAX(id_automacao) FROM automacao));
• SELECT * FROM automacao WHERE acao = 'TESTE';

DELIMITER //
• CREATE PROCEDURE sp_incluir_data(IN p_dia VARCHAR(15), IN p_data DATE)
• BEGIN
    INSERT INTO data_alvo (nome_dia, data_especifica) VALUES (p_dia, p_data);
• END //
DELIMITER ;

• CALL sp_incluir_data('Feriado', '2025-01-01');
• SELECT * FROM data_alvo WHERE nome_dia = 'Feriado';

DELIMITER //
• CREATE PROCEDURE sp_atualizar_data(IN p_id INT, IN p_dia VARCHAR(15), IN p_data DATE)
• BEGIN
    UPDATE data_alvo SET nome_dia = p_dia, data_especifica = p_data WHERE id_data = p_id;
• END //
DELIMITER ;

• CALL sp_atualizar_data((SELECT MAX(id_data) FROM data_alvo), 'Feriado Alt', '2025-01-02');
• SELECT * FROM data_alvo WHERE nome_dia = 'Feriado Alt';

DELIMITER //
• CREATE PROCEDURE sp_excluir_data(IN p_id INT)
• BEGIN
    UPDATE data_alvo SET ativo = 0 WHERE id_data = p_id;
• END //
DELIMITER ;

• CALL sp_excluir_data((SELECT MAX(id_data) FROM data_alvo));
• SELECT * FROM data_alvo WHERE nome_dia = 'Feriado Alt';

DELIMITER //
• CREATE PROCEDURE sp_incluir_agendamento(IN p_auto_id INT, IN p_data_id INT)
• BEGIN
    INSERT INTO agendamento (fk_automacao_id, fk_data_alvo_id) VALUES (p_auto_id, p_data_id);
• END //
DELIMITER ;

• CALL sp_incluir_agendamento(1, 1);
• SELECT * FROM agendamento WHERE fk_automacao_id = 1 AND fk_data_alvo_id = 1;

```



```

DELIMITER //
• CREATE PROCEDURE sp_atualizar_agendamento(IN p_auto_id INT, IN p_data_antiga INT, IN p_data_nova INT)
BEGIN
    UPDATE agendamento SET fk_data_alvo_id = p_data_nova
    WHERE fk_automacao_id = p_auto_id AND fk_data_alvo_id = p_data_antiga;
END //
DELIMITER ;

• CALL sp_atualizar_agendamento(1, 1, 2);
• SELECT * FROM agendamento WHERE fk_automacao_id = 1;

DELIMITER //
• CREATE PROCEDURE sp_excluir_agendamento(IN p_auto_id INT, IN p_data_id INT)
BEGIN
    UPDATE agendamento SET ativo = 0
    WHERE fk_automacao_id = p_auto_id AND fk_data_alvo_id = p_data_id;
END //
DELIMITER ;

• CALL sp_excluir_agendamento(1, 2);
• SELECT * FROM agendamento WHERE fk_automacao_id = 1;

```

Para cada integrante da equipe deve-se criar uma view que gere um relatório gerencial

```

• CREATE VIEW vw_detalhes_dispositivos AS
SELECT
    d.id_dispositivo,
    d.nome_dispositivo AS Aparelho,
    c.nome_comodo AS Localizacao,
    u.nome AS Proprietario,
    s.descricao AS Status_Rede,
    d.potencia_nominal AS Watts
FROM dispositivo d
JOIN comodo c ON d.fk_comodo_id = c.id_comodo
JOIN usuario u ON d.fk_usuario_id = u.id_usuario
JOIN status_conexao s ON d.fk_status_id = s.id_status
WHERE d.ativo = 1;

• SELECT * FROM vw_detalhes_dispositivos;

• CREATE VIEW vw_dispositivos_alerta AS
SELECT
    d.nome_dispositivo,
    c.nome_comodo,
    s.descricao AS Problema_Detectado
FROM dispositivo d
JOIN status_conexao s ON d.fk_status_id = s.id_status
JOIN comodo c ON d.fk_comodo_id = c.id_comodo
WHERE s.descricao IN ('Offline', 'Erro', 'Desconhecido', 'Bateria Fraca') AND d.ativo = 1;

• SELECT * FROM vw_dispositivos_alerta;

• CREATE VIEW vw_automacoes_dispositivo AS
SELECT
    d.nome_dispositivo,
    a.acao AS Acao_Programada,
    a.horario,
    a.frequencia
FROM dispositivo d
JOIN automacao a ON d.id_dispositivo = a.fk_dispositivo_id
WHERE d.ativo = 1 AND a.ativo = 1;

• SELECT * FROM vw_automacoes_dispositivo;

• CREATE VIEW vw_notificacoes_pendentes AS
SELECT
    n.titulo,
    n.mensagem,
    n.data_envio,
    d.nome_dispositivo
FROM notificacao n
JOIN dispositivo d ON n.fk_dispositivo_id = d.id_dispositivo
WHERE n.lida = 0 AND n.ativo = 1;

• SELECT * FROM vw_notificacoes_pendentes;

• CREATE VIEW vw_resumo_status AS
SELECT
    s.descricao AS Status,
    COUNT(d.id_dispositivo) AS Quantidade
FROM status_conexao s
LEFT JOIN dispositivo d ON s.id_status = d.fk_status_id AND d.ativo = 1
GROUP BY s.descricao;

• SELECT * FROM vw_resumo_status;


```

Devem ser criados 3 índices para agilizar consultas frequentes

- `CREATE INDEX idx_usuario_email ON usuario(email);`
- `EXPLAIN SELECT * FROM usuario WHERE email = 'rafael@powerhome.com';`
- `CREATE INDEX idx_dispositivo_mac ON dispositivo(endereco_mac);`
- `EXPLAIN SELECT * FROM dispositivo WHERE endereco_mac = 'AA:BB:CC:00:01';`
- `CREATE INDEX idx_leitura_data ON leitura_consumo(data_hora);`
- `EXPLAIN SELECT * FROM leitura_consumo WHERE data_hora = '2025-11-28 10:00:00';`

De acordo com o tema, o grupo deverá implementar uma transação

- `START TRANSACTION;`
- `UPDATE dispositivo SET estado_atual = 0 WHERE fk_comodo_id = 1;`
- `INSERT INTO notificacao (titulo, mensagem, lida, fk_dispositivo_id)`



`VALUES (
 'Manutenção',
 'Dispositivos da sala desligados preventivamente.',
 0,
 (SELECT id_dispositivo FROM dispositivo WHERE endereco_mac = 'AA:BB:CC:00:01')
);`
- `COMMIT;`
- `SELECT nome_dispositivo, estado_atual FROM dispositivo WHERE fk_comodo_id = 1;`
- `SELECT * FROM notificacao WHERE titulo = 'Manutenção';`