

# Planejamento e Arquitetura: Sistema de Automação Escolar Inteligente

## 1. Análise de Contexto do Ambiente Escolar com Identificação de Oportunidades

O ambiente escolar, apesar de incorporar tecnologias educacionais, ainda opera com uma infraestrutura predial reativa e manual. A gestão de acesso, climatização, iluminação e equipamentos depende da intervenção humana, gerando ineficiências e riscos.

### Oportunidades Identificadas:

- **Automação por Presença (Controle de Acesso):** Utilizar cartões RFID para liberar o acesso à sala e, simultaneamente, "ativar" o ambiente, ligando um conjunto pré-definido de equipamentos (luzes, projetor, computadores). Isso garante que os recursos só sejam consumidos quando há um usuário autorizado presente.
- **Automação por Condição (Monitoramento Ambiental):** Uma vez que a sala está "ativa", usar sensores de temperatura, luminosidade e umidade para modular os sistemas de forma inteligente, mantendo o ambiente confortável e economizando energia (ex: diminuir a intensidade das luzes artificiais se a luz natural for suficiente).
- **Segurança Elétrica Proativa:** Monitorar a qualidade da rede elétrica em tempo real para emitir alertas sobre anomalias, protegendo equipamentos caros contra danos e aumentando a segurança geral.
- **Gestão Centralizada e Baseada em Dados:** Desenvolver um dashboard que permita à administração cadastrar usuários, ambientes e dispositivos, além de visualizar o status de todas as salas, históricos de acesso e leituras de sensores.

## 2. Definição Clara do Problema e Escopo do Projeto

**Problema:** A gestão manual de salas de aula leva a um duplo problema: **desperdício de recursos** (equipamentos ligados em salas vazias) e **ambiente subótimo** (condições de temperatura e iluminação não ideais para o aprendizado). Adicionalmente, a falta de monitoramento elétrico expõe equipamentos a riscos de danos. Falta uma solução integrada que combine a gestão de presença com a otimização ambiental de forma autônoma.

### Escopo do Projeto:

- **Inclusões:**
  - Desenvolvimento de um sistema embarcado por sala, integrando leitor RFID, sensores (temperatura, umidade, luminosidade, qualidade da rede elétrica) e atuadores (relés).
  - Criação de uma aplicação back-end para processar dados dos nós (acessos via RFID e leituras de sensores em JSON).

- Implementação de um motor de regras de dupla condição (presença + condição ambiental).
- Desenvolvimento de uma aplicação web (front-end) para gerenciamento (ambientes, dispositivos, usuários) e visualização de status em tempo real.
- Uso de banco de dados para persistir todos os registros de acesso, leituras e ações.
- **Exclusões (Fora do Escopo Inicial):**
  - Integração com sistemas de gestão acadêmica (ERP).
  - Geração de relatórios analíticos complexos de consumo energético.
  - Desenvolvimento de um aplicativo móvel nativo.

**Justificativa Técnica:** A arquitetura híbrida é poderosa. O ESP32 é a escolha ideal por possuir Wi-Fi nativo e GPIOs suficientes para conectar um leitor RFID (SPI), múltiplos sensores (I2C/One-Wire) e relés. A comunicação via MQTT é superior ao UDP para esta aplicação, pois garante a entrega de comandos críticos (ligar/desligar). A stack de software (Node.js/React/MongoDB) é moderna, escalável e perfeitamente adequada para lidar com os dados em JSON e a interface de gerenciamento web.

### 3. Arquitetura Inicial da Solução

#### 3.1. Diagrama de Blocos e Fluxo de Dados

- **Nó Embarcado (por sala):** ESP32, Leitor RFID (MFRC522), Sensores (DHT22, LDR, etc.), Módulo de Relés.
- **Rede Local (Wi-Fi):** Conecta os nós ao servidor.
- **Servidor Back-end:** API, Motor de Regras, Broker MQTT.
- **Banco de Dados:** Armazena configurações, usuários, logs de acesso e séries temporais de sensores.
- **Cliente Front-end:** Dashboard de gerenciamento e visualização.

#### Fluxo de Dados Principal:

1. **Ativação:** Um professor passa o cartão RFID no leitor da sala.
2. O ESP32 envia o ID do cartão para o Back-end via MQTT.
3. O Back-end valida o ID no Banco de Dados.
4. Se autorizado, o Back-end retorna um comando de "Ativar Sala" e registra o acesso.
5. O ESP32 recebe o comando e aciona os relés para o "cenário de entrada" (liga luzes, projetor, etc.). A sala agora está em modo **"Ocupado"**.
6. **Monitoramento Contínuo:** Enquanto em modo "Ocupado", o ESP32 periodicamente lê os sensores (temperatura, luminosidade) e envia os dados em JSON para o Back-end.
7. O Motor de Regras no Back-end analisa esses dados e, se necessário, envia comandos de ajuste (ex: "Desligar fileira de luzes B").
8. **Desativação:** O professor passa o cartão na saída (ou o sistema entra em timeout), o Back-end envia um comando de "Desativar Sala", e o ESP32 desliga todos os equipamentos.

#### 3.2. Topologia da Rede

A topologia será **estrela**, com cada nó embarcado se comunicando de forma independente com o servidor central via Wi-Fi.

### 3.3. Estratégia de Conectividade

- **Conectividade: Wi-Fi (IEEE 802.11).**
- **Protocolo: MQTT** será o protocolo de comunicação principal devido à sua confiabilidade (QoS), baixo overhead e modelo publish/subscribe, ideal para enviar comandos e receber dados de múltiplos dispositivos IoT de forma eficiente.

## 4. Product Backlog (Framework Scrum)

ID	User Story / Tarefa	Prioridade
<b>Épico: Controle de Acesso e Ocupação</b>		
US01	Como professor, quero passar meu cartão RFID para ativar os equipamentos da sala.	<b>Must</b>
US02	Como sistema, quero validar um ID de cartão em um banco de dados de usuários autorizados.	<b>Must</b>
US03	Como sistema, quero desligar tudo quando a sala for desativada via RFID ou timeout.	<b>Should</b>
<b>Épico: Automação Ambiental</b>		
US04	Como sistema, quero receber leituras de sensores (temp, luz) em JSON.	<b>Must</b>
US05	Como sistema, quero ligar o ar-condicionado se a sala estiver ocupada e a temperatura alta.	<b>Must</b>

US06	Como sistema, quero ajustar as luzes com base na luminosidade se a sala estiver ocupada.	<b>Should</b>
US07	Como sistema, quero registrar todas as leituras e ações dos atuadores.	<b>Must</b>
<b>Épico:</b> Gestão e Dashboard		
US08	Como admin, quero cadastrar ambientes, dispositivos e usuários (com seus cartões RFID).	<b>Must</b>
US09	Como admin, quero visualizar o status (ocupado/livre, temperatura) de todas as salas.	<b>Should</b>

## 5. Planejamento dos 8 Sprints (Sprints de 2 semanas)

- **Sprint 1: Prova de Conceito - Acesso Remoto**
  - **Objetivo:** Validar um cartão RFID contra um back-end.
  - **Tarefas:** Montar circuito (ESP32 + RFID). Criar API no back-end para validar um ID. Fazer o ESP32 consultar a API e acender um LED verde/vermelho.
- **Sprint 2: Atuação e Estado**
  - **Objetivo:** Controlar um relé e manter o estado da sala.
  - **Tarefas:** Integrar módulo de relés. Implementar a lógica de "Ocupado"/"Livre" no back-end. Fazer o relé ser acionado no acesso e desligado na saída.
- **Sprint 3: Integração de Sensores**
  - **Objetivo:** Enviar dados de sensores quando a sala está ocupada.
  - **Tarefas:** Adicionar sensor de temperatura (DHT22). Programar o ESP32 para enviar leituras em JSON via MQTT somente quando a sala estiver no estado "Ocupado". Persistir dados no DB.
- **Sprint 4: Primeira Regra de Automação e CRUD de Ambientes**
  - **Objetivo:** Criar a primeira regra inteligente e a base do gerenciamento.
  - **Tarefas:** Implementar a regra de temperatura (US05) no back-end. Desenvolver a API e a tela de front-end para cadastrar/listar ambientes (US08).
- **Sprint 5: CRUD completo (Dispositivos, Usuários) e associação.**

- **Sprint 6:** Implementação das demais regras de automação (luminosidade, umidade).
- **Sprint 7:** Desenvolvimento do Dashboard de visualização em tempo real.
- **Sprint 8:** Implementação do monitoramento de rede elétrica e sistema de alertas.

## 6. Definição dos Papéis da Equipe e Rotina de Acompanhamento

- **Product Owner (PO):** Define e prioriza as funcionalidades no Product Backlog.
- **Scrum Master (SM):** Facilita o processo Scrum e remove impedimentos.
- **Development Team:** Equipe multidisciplinar (hardware, embarcado, back-end, front-end) que desenvolve o produto.

### Rotina de Acompanhamento:

- **Daily Meeting (Adaptado):** Reunião diária de 15 minutos para sincronização (O que fiz ontem? O que farei hoje? Há impedimentos?).

## 7. Referencial Técnico e Benchmarking

- **Padrões e Protocolos:**
  - RFID: ISO/IEC 14443.
  - Comunicação: Wi-Fi (IEEE 802.11), MQTT.
  - Formato de Dados: JSON; API: RESTful.
- **Bibliotecas e Frameworks:**
  - **Embarcado:** Arduino Framework (C++), MFRC522.h, Adafruit\_Sensor.h, PubSubClient.h.
  - **Back-end:** Node.js com