

Entrega Projeto Interdisciplinar.

Dores: Nas primeiras aulas do semestre, foi percebido logo um problema que era corriqueiro/cotidiano – A falta de cuidado e preparo que a sala “Maker” de muitas faculdades e escolas possuem. Essas salas possuem equipamentos e ferramentas que não podem ser de fácil acesso para qualquer aluno que entrar no ambiente. Muitas vezes, serras, ferros de solda, lixadeiras e outros utensílios sempre estão ligados na rede elétrica fazendo com que qualquer aluno possa acioná-los, assim colocando a segurança de todos em risco. Olhando por este campo de visão vemos também um grande desperdício de energia que poderia ser poupada, fazendo com que a instituição economize bastante dinheiro e possa também estar alinhada com programas de ações ambientais. Como a ODS 12, que preza pelo consumo sustentável. Em terceiro lugar, também ouvimos diversas reclamações de lecionadores da FECAP, da quantidade de ferramentas que ficam jogadas e perdidas pela falta de alguma ferramenta de controle.

Ganhos: Com a nossa solução, buscamos proporcionar segurança para alunos e professores, garantindo maior estabilidade operacional, redução de custos e controle eficiente dos equipamentos da sala Maker. A proposta traz uma abordagem tecnológica avançada, unindo software, aplicativo mobile e sistemas embarcados na infraestrutura elétrica da sala, resultando em uma experiência moderna e confiável. Os principais beneficiários como, pais, alunos de graduação, pós-graduação e doutorado terão ainda mais confiança na instituição de ensino, fortalecendo sua imagem perante a sociedade como referência em inovação e responsabilidade. Além disso, a instituição passa a contar com ferramentas inteligentes de economia de energia e gestão de uso da sala, otimizando recursos e controlando melhor as aulas. Como diferencial, entregaremos também um aplicativo exclusivo para professores, que permitirá configurar e personalizar o ambiente conforme suas necessidades pedagógicas, tornando o processo mais ágil, seguro e eficaz.

Arquitetura Inicial

Objetivos: Controle de seguranças dos usuários da sala, automação de dispositivos da sala (ar-condicionado, Televisão e Luzes), Inventário de ferramenta e aplicativo para programação da sala.

Camada Física: Dispositivo MIO para controlar relés de tomadas inteligentes e dispositivos que irão funcionar como atuadores.

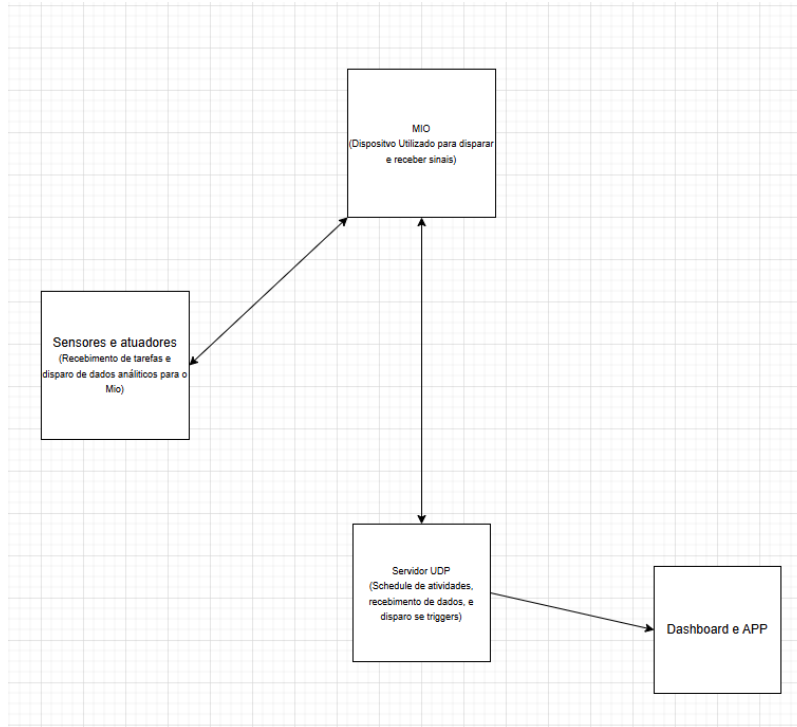
- Reed switches / sensores magnéticos em estações de ferramentas (contato seco).
- Sensores de corrente.
- PIR para zonas (opcional)
- Banco de dados UDP para receber e mandar requisições e armazenar dados sobre a sala de aula.

Controle de Programação:

MIO executa activities/triggers:

- Detecção de remoção de ferramenta - início de timer notificação no aplicativo.
- Sobrecorrente detectada (via medidor) executar Master OFF.
- Scheduling (hora de aula) - ligar iluminação via relé/Ar-Condicionado e liberação de ferramentas e tomadas presas no teto.

Diagrama de Blocos



Topologia da rede de sensores e atuadores:

Estrutura física e lógica

- Sensores digitais (reed switches, botões, sensores de presença): conectados diretamente às 12 entradas digitais do MIO.
- Atuadores (tomadas e cargas de bancada): acionados pelos 10 relés do MIO - relés controlam contadores, que ligam/desligam as tomadas ou linhas elétricas.
- Medidor de energia: conectado ao RS-485 do MIO via protocolo Modbus RTU, fornecendo tensão, corrente e kWh.
- Dispositivos IR (ar-condicionado, TV, iluminação IR): controlados pelo IR blaster/learner do MIO, que emite comandos previamente aprendidos.

Estratégia de Conectividade:

Rede física

- **Ethernet (preferencial):** conexão do MIO à rede local via cabo assim tendo maior estabilidade.
- **Wi-Fi 2.4 GHz (alternativo):** usado se Ethernet não for viável.

Protocolos de comunicação

- **UDP (MIO - Servidor Python):**
- O servidor UDP escuta pacotes do MIO (comandos, estados de entradas/saídas).
- Comunicação leve, rápida e com baixa latência, ideal para automação.

HTTP/REST (Servidor - App/Backend):

- Professores usam o app - solicitações enviadas via HTTP/HTTPS ao backend.
- Backend comunica com o servidor UDP para acionar o MIO.

Backlog:

P0 = Crítico

P1 = Importante

P2 = Desejável

Semana 1 — Infraestrutura inicial e primeiros testes

- **P0 — Estudo do ambiente Maker FECAP**

Levantamento físico da sala (tomadas, rede elétrica, Wi-Fi, bancadas, ferramentas).

Documentar pontos possíveis para sensores e atuadores.

Entregável: mapa do ambiente e plano preliminar de instalação.

- **P0 — Conexão inicial do MIO à rede Wi-Fi 2.4 GHz**

Configuração de IP estático.

Validação de acesso ao Web UI do MIO.

Entregável: MIO acessível na rede IoT dedicada.

- **P0 — Estruturação do Diagrama de blocos e fluxos**

Representar sensores, atuadores, MIO, servidor UDP e app.

Entregável: diagrama validado pela equipe.

- **P0 — Desenvolvimento do servidor UDP em Python**

Criar servidor UDP que consiga se comunicar com o MIO.

Implementar logs de conexão e mensagens recebidas.

Entregável: servidor rodando localmente, pronto para testes.

Semana 2 — Comunicação bidirecional e primeiros atuadores

P0 — Teste de comunicação entre MIO e Servidor UDP

Enviar mensagem do MIO - Servidor e vice-versa.

Implementar parser básico de pacotes.

Entregável: mensagens de ida/volta registradas em log Python.

- **P1 — Conectar e acionar primeiro atuador (relé)**

Ligar/desligar um dispositivo real (lâmpada ou tomada de teste).

Validar registro do evento no servidor.

Entregável: primeira automação real executada pelo MIO.

P1 — Estruturação do UX do aplicativo mobile (professores)

Protótipo em Figma/Sketch.

Telas: login, agendamento de equipamentos, dashboard básico.

Entregável: protótipo navegável validado pelo time.

Semana 3 — Primeiras automações práticas

- **P0 — Programação inicial do Front-End Mobile**

Implementar telas de login e dashboard com status dos equipamentos.

Entregável: APK de teste com navegação inicial.

P0 — Testes de relés no bloqueio de corrente das tomadas

Ligar tomadas via contator acionado pelos relés do MIO.

Testar desligamento de segurança (Master OFF).

Entregável: bancada controlada pelo MIO.

Semana 4 — Infraestrutura avançada e agendamento

- **P0 — Teste do controle por sinais infravermelho (IR)**

Aprender e transmitir sinais de ar-condicionado, TV e iluminação IR.

Entregável: IR do MIO funcionando com pelo menos 2 dispositivos.

P1 — Desenvolvimento do Backend do App.

Integração do schedule com o servidor UDP e comandos ao MIO.

Entregável: agendamento de ligar/desligar funcionando via app.

Semana 5 — Monitoramento de ferramentas

- **P0 — Instalação de sensores de presença nas ferramentas**

Conectar sensores às entradas digitais do MIO.

Testar evento de retirada e devolução.

Entregável: log no servidor quando a ferramenta é removida.

- **P1 — Trigger local de não-devolução (MIO)**

Regra no MIO: se sensor não voltar ao estado inicial após X minutos - enviar alerta.

Entregável: notificação enviada ao servidor/app.

Semana 6 — Medição de energia e telemetria

- **P0 — Integração de medidor de energia:**

Configurar polling no MIO.

Enviar dados (corrente, tensão, kWh) ao servidor UDP.

Entregável: dashboard inicial com dados de consumo.

- **P1 — Exportação de dados para banco de dados:**

Estruturar banco de séries temporais.

Criar gráficos de consumo por tomada.

Entregável: painel básico de monitoramento.

Semana 7 — Relatórios e segurança

- **P0 — Implementar Master OFF via servidor UDP**

Comando de emergência direto do app - servidor UDP - MIO.

Entregável: desligamento geral testado.

- **P1 — Geração de relatórios mensais**

Consumo de energia, uso das ferramentas, incidentes.

Export em PDF/CSV.

Entregável: relatório gerado e enviado por e-mail.

- **P2 — Segurança de rede e credenciais**

Trocar senhas padrão do MIO.

Configurar VLAN IoT e firewall.

Entregável: rede isolada e documentada.

Semana 8 — Finalização e entrega piloto

P0 — Testes integrados

Fluxo completo: App - Servidor UDP - MIO - relé/sensor - notificação.

Entregável: sistema validado com professor em teste real.

- **P2 — Treinamento de uso para professores/coordenadores**

Manual rápido + apresentação.

Entregável: guia de usuário.

Papéis da Equipe:

PO: João Pedro Lima Paulo.

Scrum Master: Guilherme Alves de Oliveira e Oliveira.

Manager de programação e elétrica: Vinicius Binda.

Programação Front-End: Gustavo

Programação BackEnd: Giulia Nogueira.

Benchmarking

UDP/IP: comunicação leve e rápida entre o MIO e o servidor Python, garantindo baixa latência.

HTTP/HTTPS (REST): integração entre servidor e aplicativo mobile.

MQTT protocolo padrão para IoT, caso a solução evolua para cloud ou múltiplos dispositivos.

Ethernet ou Wi-Fi 2.4GHz: conectividade primária do MIO.

Servidor:

socket (nativo) - comunicação UDP.

asyncio -tratamento de conexões simultâneas e assíncronas.

sqlalchemy ou psycopg2 - integração com Banco de dados.

fastapi ou flask (em estudo) - criação de APIs REST para integração com app mobile.

Aplicativo:

Flutter - desenvolvimento multiplataforma (Android/iOS) com alta performance.

Firebase - autenticação, push notifications e analytics.