

Descrição da arquitetura final.

1. Ao término do projeto, serão apresentados três sistemas principais que servirão como base para a demonstração da proposta inicial, todos destinados à implementação na sala Maker da FECAP.

O primeiro sistema consiste em um motor acoplado a um conjunto de engrenagens, responsável pelo acionamento de subida e descida da tomada suspensa presente no ambiente.

O segundo sistema tem como objetivo o controle de presença das ferramentas disponíveis na sala, utilizando sensores para monitorar automaticamente a retirada e o retorno de cada item.

O terceiro sistema visa o gerenciamento do fornecimento de energia elétrica nas tomadas da sala, permitindo o controle do funcionamento de equipamentos como ferros de solda, garantindo maior segurança e eficiência energética.

Em todos os sistemas, é utilizado o módulo MIO, desenvolvido pela empresa Flex, com a função de acionar e desativar os relés, realizando a distribuição controlada de energia elétrica para os componentes necessários de cada subsistema.

.

1.2. Componentes detalhados:

Sistema de engrenagens – Tomada suspensa:

1 um carretel.

1 engrenagem impressa em 3D.

1 fio.

1 extensão de energia.

3 pedaços de folha de cobre.

3 Escovas de grafites.

1 pedaço de MDF.

2 Pedaços de madeira.

1 motor.

Sistema presença de ferramenta:

1 suporte de madeira.

2 pregos para sustentação.

2 resistores fotovoltaicos.

2 leds RGB.

1 Arduino (uno).

1 protoboard.

2 resistores de 300 ohms.

Sistema de ligar e desligar régua de energia:

1 régua de energia.

1.3. Descrição de sistemas detalhados.

Sistema de engrenagens – tomada suspensa:

O sistema é composto por um fio enrolado em um carretel de plástico, permitindo o acionamento do carretel para realizar o movimento de subida ou descida da régua de energia conectada à extremidade do fio.

Em uma das laterais externas do carretel, há uma placa de MDF fixada juntamente com uma engrenagem responsável pela transmissão mecânica do movimento. Na outra extremidade, está localizado o conjunto de alimentação elétrica do sistema.

O sistema de alimentação é formado por três segmentos de folha de cobre dispostos na superfície externa do carretel. Esses segmentos entram em contato com escovas de grafite, que têm a função de conduzir a corrente elétrica proveniente da fonte externa para o interior do sistema. A energia é transferida das escovas para as folhas de cobre e, em seguida, para o circuito interno, garantindo a continuidade elétrica mesmo durante a rotação do carretel.

O motor elétrico está posicionado no centro do carretel e é responsável pelo acionamento do mecanismo de enrolamento e desenrolamento do fio.

Quando a energia é fornecida pelas escovas, o motor é alimentado e realiza o movimento de subida ou descida da régua de energia conforme o sentido de rotação estabelecido.

Sistema de presença de ferramenta:

O sistema consiste em um suporte de madeira equipado com dois pregos, que servem como base para o encaixe das ferramentas. Logo abaixo desses pregos há dois orifícios, posicionados de forma que, na parte posterior, estão instalados os sensores fotovoltaicos. Acima dos pregos, encontram-se os LEDs RGB responsáveis pela sinalização visual.

O funcionamento é baseado na variação da incidência de luz detectada pelos sensores, que estão conectados ao módulo MIO. Em operação normal, quando uma ferramenta é posicionada no suporte, ela bloqueia a passagem de luz até os sensores, provocando a ausência de luminosidade. Nesse

estado, o sistema aciona a iluminação verde dos LEDs, indicando que a ferramenta está devidamente alocada.

Quando a ferramenta é retirada, a luz ambiente volta a incidir sobre os sensores fotovoltaicos, que registram essa variação e enviam o sinal ao módulo de controle. Como resposta, o sistema aciona a iluminação vermelha dos LEDs, sinalizando a ausência da ferramenta no suporte.

Sistema de alimentação de energia – Ferros de solda:

Como descrito no primeiro documento, um dos principais dispositivos que queríamos criar era uma fonte inteligente para controlar os ferros de solda. Conseguimos isso ligando a fonte diretamente no MIO. Primeiro pegamos a entrada da fonte e arrancamos ela ficando apenas os fios, deste modo conseguimos ligar eles nos relés do dispositivo da Flex, fazendo com conseguíssemos criar de uma maneira um dispositivo que fosse programável, assim ligando e desligando no momento que queríamos a energia.

Código comentado:

```
# Importações bibliotecas
from sqlalchemy import create_engine, Column, Integer, String
from sqlalchemy.orm import declarative_base, sessionmaker
import os
from dotenv import load_dotenv

# Carrega as variáveis de ambiente do arquivo .env
load_dotenv()

# Lê as variáveis de ambiente definidas no arquivo .env
usuario = os.getenv("usuario")
senha = os.getenv("senha")
host = os.getenv("host")
porta = os.getenv("porta")
banco = os.getenv("banco")

# Cria a engine de conexão com o banco de dados MySQL usando o driver
# 'pymysql'
# O SQLAlchemy usará essa engine para se comunicar com o banco
engine =
create_engine(f'mysql+pymysql://{usuario}:{senha}@{host}:{porta}/{banco}')

# Cria a classe base para os modelos (ORM)
Base = declarative_base()
```

```

# Define a tabela "mensagens" como uma classe
class mensagens(Base):
    __tablename__ = 'mensagens' # nome da tabela no banco

    # Definição das colunas e tipos de dados
    ID_Mensagem = Column(Integer, primary_key=True, autoincrement=True)
# chave primária
    Mensagem = Column(String(200)) # texto da mensagem (até 200
caracteres)
    IP_Cliente = Column(String(15)) # IP do cliente que enviou
    Porta_Cliente = Column(Integer) # porta do cliente

# Define a tabela "sensores" como uma classe Python
class sensores(Base):
    __tablename__ = 'sensores' # nome da tabela no banco

    # Definição das colunas e tipos de dados
    ID_Mensagem = Column(Integer, primary_key=True, autoincrement=True)

# chave primária

    Sensor = Column(String(200)) # nome ou tipo do sensor
    IP_Cliente = Column(String(15)) # IP do cliente que enviou os dados
    Dados = Column(String(200)) # dados enviados pelo sensor

# Cria fisicamente as tabelas no banco
Base.metadata.create_all(engine)

# Cria uma fábrica de sessões para interagir com o banco
Session = sessionmaker(bind=engine)
session = Session() # inicia uma nova sessão

session.commit() # confirma transações pendentes
session.close() # encerra a sessão com o banco

```

Lógica embarcada:

A integração entre sensores, atuadores e a lógica embarcada foi desenvolvida utilizando o dispositivo MIO, da empresa Flex, como unidade central de controle e processamento. Esse módulo é responsável por receber os sinais dos sensores instalados nos sistemas, interpretar a lógica de operação embarcada e acionar os atuadores correspondentes de acordo com as condições detectadas.

Na camada física, os sensores digitais, como os sensores fotovoltaicos e motores, estão conectados diretamente às entradas digitais do MIO. Esses sensores realizam a leitura de variáveis físicas, como a presença ou ausência

de luz ou o posicionamento de ferramentas. Os sinais captados são enviados ao MIO, que processa as informações conforme a lógica configurada. Nas saídas do MIO, estão conectados os atuadores, responsáveis pela execução das ações do sistema. Entre eles estão os relés, que comandam o acionamento de motores e circuitos elétricos, e os LEDs RGB, utilizados para sinalização visual de estados operacionais.

Relato dos testes e validações realizados:

Inicialmente, foram feitos testes individuais de hardware, verificando o funcionamento dos sensores fotovoltaicos, LEDs RGB, relés e do motor responsável pelo acionamento da régua de tomadas suspensa. Cada componente foi energizado separadamente para avaliar sua resposta elétrica e assegurar que não houvesse falhas de ligação ou mal contato nos circuitos.

Para o sistema de controle de ferramentas, foram feitos ensaios práticos simulando a colocação e a retirada dos objetos no suporte. Observou-se que, ao cobrir o sensor fotovoltaico, a ausência de luz era corretamente identificada, acionando o LED verde e indicando a presença da ferramenta. Quando o sensor voltava a receber luz, o sistema alterava automaticamente para o LED vermelho, confirmando a detecção da ausência.

No sistema de tomada suspensa, o motor foi testado em ambos os sentidos de rotação, controlado pelos relés do MIO. Foram verificadas a estabilidade do movimento, o tempo de resposta entre o comando e o acionamento, além da integridade das conexões elétricas durante o processo de subida e descida.

Por fim, foi realizada uma validação integrada, conectando todos os sistemas ao módulo MIO e simulando as condições de uso real. Essa etapa confirmou a sincronização entre sensores e atuadores, o correto acionamento dos relés e a resposta imediata do sistema às mudanças detectadas.

Com base nesses testes, concluiu-se que todos os subsistemas apresentaram desempenho satisfatório, resposta estável e operação segura, atendendo aos objetivos propostos para a automação da sala Maker.

Demonstração funcional do sistema em operação simulada:

A demonstração foi realizada em ambiente controlado, simulando as condições reais da sala Maker. Durante o teste, o módulo MIO controlou o motor responsável pela régua de tomadas suspensa, permitindo o movimento de subida e descida de forma estável.

No sistema de ferramentas, os sensores fotovoltaicos responderam corretamente: ao posicionar a ferramenta, o LED verde acendeu indicando presença; ao retirá-la, o LED vermelho sinalizou ausência.

Os testes mostraram que os sistemas funcionam de maneira integrada, com respostas rápidas e sem falhas, comprovando a eficiência e viabilidade da automação proposta.

Backlog final com histórico dos 8 sprints

Semana 1 — Infraestrutura inicial e primeiros testes

- **P0 — Estudo do ambiente Maker FECAP (Concluído)**

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 (Concluído)**

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 (Concluído)**

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

Entregável: diagrama validado pela equipe.

- **P0 — Desenvolvimento do servidor UDP em Python (Concluído)**

Criar servidor UDP que consiga se comunicar com o MIO.

Implementar logs de conexão e mensagens recebidas.

Entregável: servidor rodando localmente, pronto para testes. (Concluído)

Semana 2 — Comunicação bidirecional e primeiros atuadores

P0 — Teste de comunicação entre MIO e Servidor UDP (concluído)

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) (Concluído)**

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) (mudança – invés do APP, construímos um site)

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 – (Mudança – Site)**

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 (Concluído)

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) (Concluído)**

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 (Mudança – Site do Maker)

Estrutura básica de API em Python (FastAPI ou Flask).

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 (Concluído)**

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) (Não realizado)**

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: (Não Realizado)**

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: (Não Realizado)**

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 (Não Realizado)**

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

Entregável: desligamento geral testado.

- **P1 — Geração de relatórios mensais (Não Realizado)**

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 (Concluído)**

Trocar senhas padrão do MIO.

Configurar VLAN IoT e firewall.

Entregável: rede isolada e documentada.

Semana 8 — Finalização e entrega piloto

P0 — Testes integrados (Concluído)

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

Entregável: sistema validado com professor em teste real.

- **P1 — Treinamento de uso para professores/coordenadores (Concluído)**

Manual rápido + apresentação.

Entregável: guia de usuário.