

SIGMI

Claudio Adamiwicz, Evandro José da Silva, Lauriana Paludo

Relatório Técnico GTI 2024-1



Relatório Técnico

Título do trabalho: SIGMI – Sistema de Gestão de Equipamentos

ADAMOWICZ, Claudio - claudio@cwbbtech.com.br

SILVA, Evandro – vandoopl@gmail.com

PALUDO, Lauriana – lauriana.paludo@ifpr.edu.br

Resumo

Controle sobre as demandas de manutenção, Manutenção de equipamentos, Progresso de manutenção.

Introdução

Este relatório técnico apresenta o SIGMI, um sistema de gestão de manutenção de equipamentos desenvolvido especificamente para microempreendedores individuais (MEIs). O SIGMI foi projetado para permitir que esses profissionais gerenciem, de forma eficaz e organizada, as entradas e saídas de equipamentos em manutenção, superando os desafios associados ao gerenciamento manual e à falta de ferramentas acessíveis no mercado.

O objetivo do SIGMI é promover a eficiência operacional e o crescimento econômico na região de Curitiba e sua região metropolitana (RMC), contribuindo para a geração de empregos e o fortalecimento de micro e pequenas empresas. Através da inovação tecnológica, o sistema apoia o empreendedorismo ao fornecer ferramentas que facilitam a estruturação e gestão dos negócios, permitindo que os MEIs aloque de forma mais eficaz seus recursos e tempo.

A solução proposta pelo SIGMI visa resolver problemas comuns enfrentados pelos MEIs, como atrasos na entrega de serviços e falta de informações detalhadas para os clientes. Ao melhorar o controle das demandas de manutenção, o SIGMI busca garantir que os microempreendedores possam oferecer serviços de alta qualidade de maneira mais rápida e

eficiente, aumentando a transparência e a satisfação do cliente.

Comparado a soluções anteriores, como as propostas por Bule (2022) e Sato, Oliveira e Alves (2022), o SIGMI diferencia-se por ser uma ferramenta acessível, intuitiva e adaptada às necessidades específicas dos MEIs que trabalham com manutenção de equipamentos eletrônicos. Ele se propõe a oferecer uma solução prática e de baixo custo, alinhando-se às necessidades do público-alvo e contribuindo para a expansão dos negócios e o fortalecimento econômico da região.

2 – Materiais e Métodos

O desenvolvimento do SIGMA seguiu com uma metodologia ágil, com iterações para o desenvolvimento, teste e validação. Foram utilizadas no desenvolvimento de tecnologias web, como Python, HTML, JavaScript, e banco de dados Mysql.

3 – SIGMI

3.1 - PERSONA





SAMUEL OLIVEIRA

TÉCNICO DE MANUTENÇÃO DE EQUIPAMENTOS ELETRÔNICOS

ADULTO (26 – 40 ANOS)

Mini-bio

Samuel possui uma rotina corrida mas se esforça para oferecer o melhor serviço aos seus clientes. Desde a adolescência, Samuel é apaixonado por resolver problemas em computadores. Seu dia a dia é focado em encontrar soluções eficientes para garantir que equipamentos, aplicativos e periféricos de seus clientes estejam em bom funcionamento. Nos momentos de lazer, Samuel é descomplicado e dedica seu tempo a atividades simples, explorando parques locais, praticando caminhadas, se aventurando na cozinha e se reunindo com seus amigos e familiares. Busca equilibrar o profissionalismo com a vida pessoal, mantendo-se acessível e comprometido com o serviço de qualidade através de cursos, treinamentos e eventos na área de informática sempre que possível e também dedicando momentos para si.

Detalhes Pessoais

Localização Pinhais - PR	Renda Familiar De R\$3.501,00 a R\$6.500,00	Nível Educacional Ensino superior completo	Status de Relacionamento Noivo(a)
------------------------------------	---	--	---

Carreira

Empresa Autônomo	Tamanho da Empresa Microempresa	Responsabilidades Profissionais Samuel é responsável pelo atendimento receptivo dos clientes, avaliação dos equipamentos, elaboração de orçamentos, registro dos serviços em uma planilha eletrônica, execução das manutenções e comunicação com os clientes via WhatsApp.
----------------------------	---	--

Objetivos O trabalho é mensurado a partir da receita gerada. Samuel busca a expansão do seu negócio e para isso possui metas para melhorar o controle dos processos realizados e otimizar o tempo de execução das manutenções dos equipamentos.	Desafios Samuel realiza a gestão de suas atividades de forma manual e enfrenta riscos quanto à integridade e disponibilidade das informações e históricos de atendimentos. O tempo despendido nesses processos acarreta em atrasos na execução dos serviços e nos prazos informados aos clientes. Atualmente não possui informações e métricas centralizadas sobre o trabalho realizado para identificar os pontos que necessitam de melhoria.
---	--

Canais de Comunicação



3.2 StoryBoard

SAMUEL TRANSFORMOU SEU NEGÓCIO COM O SIGMI



SAMUEL POSSUI UM PEQUENO NEGÓCIO DE MANUTENÇÃO DE EQUIPAMENTOS DE INFORMÁTICA



GOSTARIA DE MELHOR ORGANIZAR E GERENCIAR AS DEMANDAS DE SERVIÇO POIS EXTRAPOLOU O PRAZO DE ALGUNS CLIENTES E PRECISOU TRABALHAR ATÉ TARDE POR DIAS PARA REORGANIZAR TUDO



PERCEBEU QUE SEM UM SISTEMA PARA GERENCIAR SUAS DEMANDAS NÃO POSSUI TEMPO NEM INFORMAÇÕES PARA REALIZAR MELHORIAS



ENTÃO ELE CONHECEU O SIGMI E A POSSIBILIDADE DE GERENCIAR DE FORMA AUTOMATIZADA OS SERVIÇOS DE MANUTENÇÃO E A COMUNICAÇÃO COM OS CLIENTES



CONQUISTOU TRABALHO ORGANIZADO SEM SOBRECARGAS E CLIENTES SATISFEITOS COM O SERVIÇO ÁGIL E COM A TRANSPARÊNCIA DAS INFORMAÇÕES

3.3 - Vision Board

THE PRODUCT VISION BOARD

Modelado para:

Modelado por:

Claudio Cesar Adamowicz

Evandro José da Silva

Em: 04/10/2023

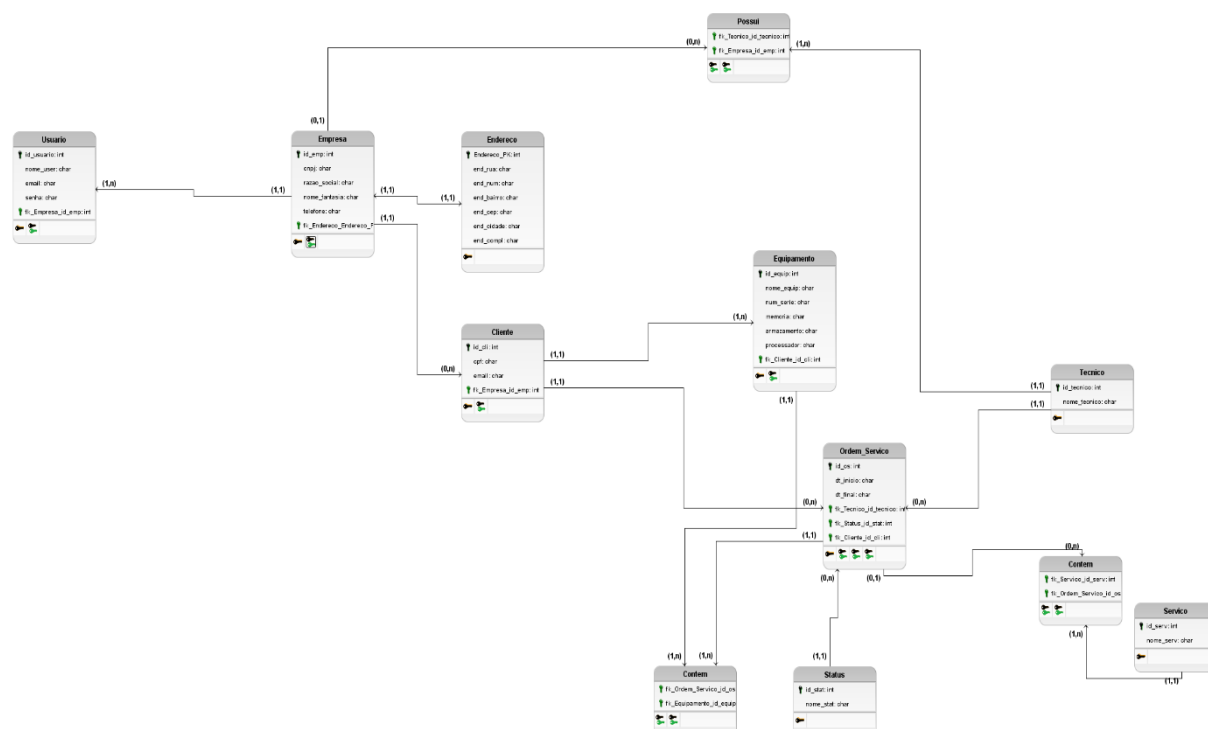
Iteração: #1

Declaração de Visão

Nosso software de gerenciamento de entradas e saídas de equipamentos para manutenção visa capacitar MEIs da área de tecnologia a oferecer serviços de alta qualidade, proporcionando controle eficaz, eficiência operacional e satisfação do cliente.

Grupo Alvo	Necessidades	Produto	Valor	Concorrência
<p>Quem são os usuários-alvo para seu produto?</p> <p>Microempreendedores Individuais (MEIs) da área de tecnologia que realizam manutenção de equipamentos, como computadores, dispositivos eletrônicos, impressoras, etc. Técnicos de manutenção de equipamentos que trabalham para esses MEIs ou são MEIs eles próprios.</p> <p>Quem são os clientes potenciais?</p> <p>Microempreendedores Individuais (MEIs) que atuam na área de tecnologia e desejam melhorar o controle e a eficiência de suas operações de manutenção. Técnicos de manutenção de equipamentos que buscam uma solução de software para gerenciar seus próprios negócios de MEI de forma mais eficiente e atender melhor aos clientes.</p>	<p>Que problema você quer resolver?</p> <p>O produto resolve a necessidade de controle eficaz sobre as demandas de manutenção de equipamentos para MEIs da área de tecnologia. Isso inclui a atualização de status de manutenção, comunicação com clientes e organização eficiente das operações de manutenção.</p> <p>Que objetivo deseja atingir?</p> <p>O objetivo principal do produto é melhorar a eficiência operacional e a satisfação do cliente na área de manutenção de equipamentos. Ele visa ajudar os MEIs a entregar serviços de alta qualidade de forma mais rápida e eficiente, evitando atrasos na entrega de equipamentos e garantindo que os clientes estejam bem informados sobre o progresso da manutenção.</p> <p>Como seu produto cria valor para os usuários?</p> <p>O produto cria valor para o usuário ao automatizar tarefas de gerenciamento que normalmente seriam demoradas e propensas a erros. Ele fornece uma plataforma centralizada para monitorar e controlar todas as atividades relacionadas à manutenção, economizando tempo e reduzindo o estresse relacionado à gestão de tarefas manuais. Também gera valor ao cliente final ao gerar visibilidade sobre as etapas do serviço que está consumindo e com isso fortalece a confiança e propicia a fidelização do cliente.</p> <p>Que emoções ele evoca?</p> <p>O produto pode propiciar alívio e satisfação para os MEIs, pois eles podem confiar em uma ferramenta que os ajuda a manter o controle de suas operações e oferecer um serviço mais eficiente aos clientes. Além disso, a sensação de profissionalismo e confiabilidade associada a um sistema de gerenciamento bem organizado pode gerar confiança tanto nos MEIs quanto nos clientes.</p>	<p>Quais as principais funcionalidades que você deseja oferecer?</p> <p>Cadastro do cliente Cadastro da demanda de manutenção Envio de notificações ao cliente Cadastro de Produto Gerar Código de barras / QRCode</p> <p>O que torna o seu produto único?</p> <p>O que torna nosso produto único é a combinação de um software intuitivo e especializado em gerenciamento de entradas e saídas de equipamentos para manutenção, projetado especificamente para atender às necessidades dos Microempreendedores Individuais (MEIs) da área de tecnologia. Além disso, nosso software se destaca pela ênfase na facilidade de uso, automação de tarefas manuais, rastreamento preciso de equipamentos e comunicação eficaz com os clientes, oferecendo um conjunto completo de ferramentas para melhorar a eficiência operacional e a satisfação do cliente. Isso o diferencia de soluções mais genéricas ou complexas, atendendo a um público específico com uma solução adaptada às suas necessidades exclusivas.</p>	<p>Quais são seus objetivos de negócio?</p> <p>Aquisição de Clientes, Retenção de Clientes, Eficiência operacional, Satisfação do Cliente, Inovação do Produto, Valor para o Cliente, Receita.</p> <p>Qual o valor de seu produto para a sua empresa?</p> <p>Receita, Fidelização de Clientes, Diferenciação Competitiva, Parcerias Estratégicas.</p>	<p>Quem são seus concorrentes?</p> <p>Nossos concorrentes são: Outros Software de Gerenciamento de Serviços, Sistemas Personalizados, Concorrentes Locais, Soluções de Planilhas, Startups e Empresas de Software Específicas para MEIs.</p> <p>Quais as alternativas para o seu produto?</p> <p>Tiny, Conta Azul.</p>

3.4 – Diagrama de Classes



3.5 Guia de Estilo e Padrões

3.5.1. Design Visual

- **Paleta de Cores:** Utilizar a paleta de cores já definida para manter a consistência visual em todas as páginas. A paleta deve proporcionar uma boa legibilidade e ser agradável aos usuários.
- **Tipografia:** Escolher fontes simples e legíveis. Manter um tamanho de fonte que seja acessível em diferentes dispositivos.
- **Espaçamento:** O design deve ser espaçado e limpo, evitando o acúmulo de elementos visuais que possam confundir o usuário.
- **Layout:** Utilizar layouts responsivos para que a interface funcione bem em dispositivos móveis e desktops.

3.5.2. Padrões de Código

HTML/CSS:

- Manter o código bem indentado e estruturado.
- Utilizar classes e IDs significativos para os elementos.
- Centralizar o uso de CSS em arquivos separados no diretório ../static/css.

JavaScript:

- Utilizar funções claras e bem definidas, com nomes descritivos.
- Manter o código modular para facilitar a manutenção.
- Evitar manipulações diretas no DOM, sempre que possível, utilizando bibliotecas ou frameworks adequados.

Python (Flask):

- Seguir a estrutura padrão do Flask com organização de rotas no main.py e lógica de banco de dados em conexasdb.py.
- Utilizar docstrings e comentários claros para explicar a lógica do código.
- Manter as rotas e funções concisas e reutilizáveis.

3.5.3. Organização de Arquivos

- **Estrutura de Diretórios:**
 - templates/: Para armazenar todos os arquivos HTML.
 - static/css/: Para todos os arquivos de estilo CSS.
 - static/js/: Para os scripts JavaScript.
 - conexasdb.py: Para interações com o banco de dados.
 - main.py: Para a lógica principal da aplicação e rotas.

3.5.4. Boas Práticas

- **Validação:** Validar os dados do usuário tanto no front-end (JavaScript) quanto no back-end (Flask).
- **Feedback ao Usuário:** Fornecer mensagens claras de sucesso ou erro em operações como cadastros ou atualizações.

- **Segurança:** Implementar proteção contra ataques comuns, como SQL Injection e Cross-Site Scripting (XSS).
- **Testes:** Testar rotinas diretamente no banco de dados para garantir a integridade das operações e corrigir erros rapidamente.

3.6 Trechos de código fonte

Trecho de busca do técnico para seleção no campo.

```
# Função para buscar técnicos no banco de dados
def busca_tecnicos(query):
    conexao()
    cursor = conn.cursor()
    try:
        cursor.execute("SELECT id_tec, nome_tecnico FROM tbTecnico WHERE nome_tecnico LIKE %s", (f"%{query}%"))
        results = cursor.fetchall()
        return [{"id_tec": row[0], "nome_tecnico": row[1]} for row in results]
    finally:
        endconexao()
```

Trecho de busca de ordem de serviço no banco de dados e retorno na tela de gestão das O.S.

```
def buscar_ordens_servico():
    conexao()
    query = """
        SELECT os.id_os, os.data_inicio, os.data_final, s.nome_status, t.nome_tecnico, c.nome, os.evolutao
        FROM tbOrdemServico os
        LEFT JOIN tbStatus s ON os.id_status = s.id_stat
        LEFT JOIN tbTecnico t ON os.id_tecnico = t.id_tec
        LEFT JOIN tbCliente c ON os.id_cliente = c.id_cli
        ORDER BY os.data_inicio ASC;
    """
    cursor.execute(query)
    ordens = cursor.fetchall()
    endconexao()
```

Trecho para atualizar a ordem de serviço de acordo com seu novo status

```
def atualizar_status_ordem_servico(id_os, novo_status):
    conexao()
    query = """
        UPDATE tbOrdemServico
        SET id_status = %s
        WHERE id_os = %s
    """
    cursor.execute(query, (novo_status, id_os))
    conn.commit()
    endconexao()
```

Trecho para montar a pagina de gestão de ordem de serviços.

```
app.route('/gestaodeos')
def gestao_os():
    ordens = buscar_ordens_servico()

    ordens_por_status = {
        "Entrada": [],
        "Analise": [],
        "Manutencao": [],
        "Concluido": []
    }

    for ordem in ordens:
        status_nome = ordem[3] if ordem[3] else "Desconhecido"
        if status_nome in ordens_por_status:
            ordens_por_status[status_nome].append(ordem)

    return render_template('gestaodeos.html', ordens_servico=ordens_por_status)
```

Trecho para verificação de datas na hora do cadastro.

```
# Verificação das datas
if data_inicio > data_final:
    alert_message = "Erro: A data de saída não pode ser menor que a data de entrada."
    alert_type = "error"
    return render_template('cadastro_os.html', alert_message=alert_message, alert_type=alert_type)
```

Trecho para busca de clientes cadastrados no banco.

```
@app.route('/busca_cliente') #Rota para buscar cliente no banco
def busca_cliente():
    query = request.args.get('q') #Captura o valor do parâmetro de consulta
    if query:
        results = busca_clientes(query) #Verifica se o parâmetro q foi passado e não está vazio
        return jsonify(results) #Chama a função busca_clientes passando a consulta (query) como argument
    return jsonify([]) # Retorna uma lista vazia se nenhum parâmetro de consulta foi passado!
```

Trecho para carregamento do perfil do cliente.

```
80 @app.route('/perfil') #rota carregamento perfil html
81 def perfil():
82
83     dadosdocliente=buscadadosperfil()
84     field_names = ['RAZAO SOCIAL', 'CNPJ', 'NOME FANTASIA', 'ENDereco', 'NUMERO', 'CEP', 'BAIRRO', 'CIDADE', 'ES
85     fields_data = []
86     for row in dadosdocliente:
87         if isinstance(row, int) == True:
88             convertido=str(row)
89             fields_data.append(dict(zip(field_names, convertido)))
90
91         else:
92
93             fields_data.append(dict(zip(field_names, row)))
94     for teste in fields_data:
95         print(teste)
96     return render_template('perfil.html', fields_data=fields_data)
97
```

3.7 Interfaces Desenvolvidas

3.7.1 Tela de Login



A login screen mockup with a light blue background. At the top center, the text "Bem Vindo" is displayed in a bold, black, sans-serif font. Below this, a white rectangular form with rounded corners and a subtle drop shadow is centered. The form contains the following elements: a label "NOME" above a text input field with the placeholder "Digite seu Login"; a label "SENHA" above a text input field with the placeholder "Digite sua senha"; a button labeled "ENTRAR" with a light gray background and a thin black border; and a link "Cadastre-se aqui" in a smaller, gray font at the bottom right of the form.

Bem Vindo

NOME

Digite seu Login


SENHA

Digite sua senha

ENTRAR

[Cadastre-se aqui](#)

3.7.2 Menu Principal




Cadastro de Clientes

Cadastro de Equipamentos

Cadastro de Técnicos

Cadastro de Ordem de Serviço

Gestão de Ordem de Serviço



3.7.3 Perfil do Cliente

Dados da Empresa

Razao Social: EMPRESA TESTE

CNPJ: 11111111000101

Nome Fantasia: SIGMI

Endereco: Rua teste

Numero: 500

CEP: 80000000

Cidade: Pinhais

Estado: PR

Bairro: Jardim Claudia

Complemento: Casa 1000

Voltar

3.7.4 Tela de Cadastro de Cliente

Cadastro de Clientes

Tipo de Cliente:

Selecione

Selecione
Pessoa Física
Pessoa Jurídica

Cadastrar

Voltar

3.7.5 Tela Cadastro de Equipamento

Cadastro de Equipamento

Cliente

Nome do Equipamento

Número de Série

Memória (GB)

Armazenamento (GB)

Processador

3.7.6 Tela de Cadastro de Técnico

Cadastro de Técnico

Nome do Técnico

Cadastrar

Voltar

3.7.7 Tela de Cadastro de Ordem de Serviço

Cadastro de Ordem de Serviço

Cliente:

Equipamento:

Técnico:

Status:

Selecione

Data de Início:

dd/mm/aaaa

Data Final:

3.7.8 Tela de Gestão de Ordem de Serviço

Gestão de Ordens de Serviço

Entrada

ID	Data Início	Data Final	Status	Técnico	Cliente	Evolução	Ação
2	2024-07-19	2024-07-20	Entrada	Claudio	Alice Adamowicz	Teste de evolução	Editar
11	2024-08-06	2024-08-07	Entrada	Claudio	Joao	Joao	Editar

Analise

ID	Data Início	Data Final	Status	Técnico	Cliente	Evolução	Ação
4	2024-07-26	2024-07-19	Analise	Claudio		teste24-07	Editar
8	2024-07-30	2024-07-31	Analise	Claudio	cadamowicz		Editar
9	2024-07-30	2024-07-30	Analise	Claudio	cadamowicz		Editar
9	2024-07-30	2024-07-30	Analise	Claudio	cadamowicz		Editar

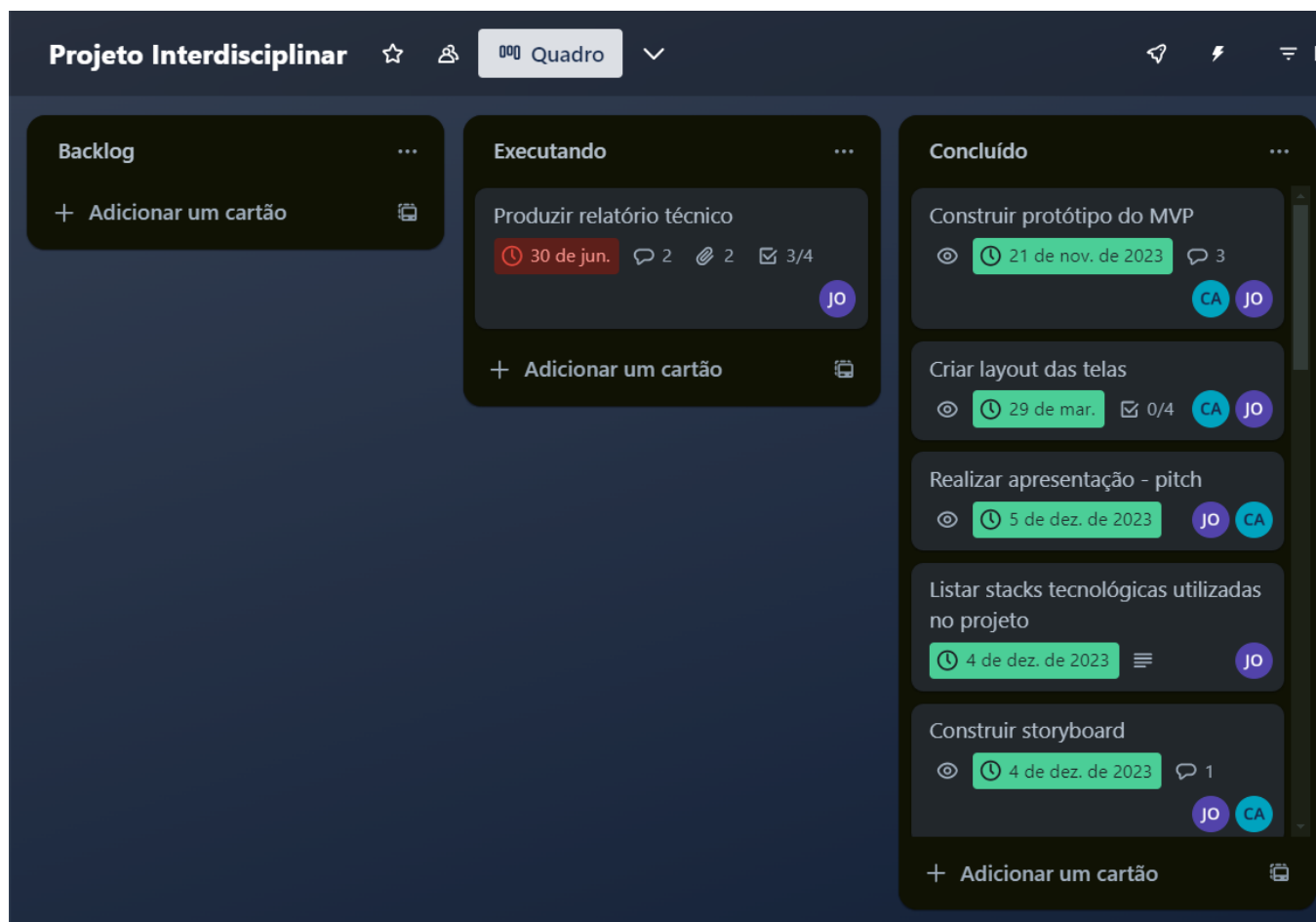
Manutencao

ID	Data Início	Data Final	Status	Técnico	Cliente	Evolução	Ação
3	2024-07-19	2024-07-26	Manutencao	Claudio	cadamowicz		Editar
7	2024-07-25	2024-07-31	Manutencao	Claudio	cadamowicz		Editar
5	2024-07-30	2024-07-31	Manutencao	Claudio	Alice Adamowicz		Editar
6	2024-07-31	2024-07-31	Manutencao	Claudio	Alice Adamowicz		Editar

Concluido

ID	Data Início	Data Final	Status	Técnico	Cliente	Evolução	Ação
1	2024-07-20	2024-07-26	Concluido	Claudio	Alice Adamowicz	ccc	Editar

3.8 Quadro de gestão visual de tarefas do projeto (trello)



4 – Considerações Finais

Os dados coletados durante o desenvolvimento do SIGMI revelaram que uma significativa parcela dos MEIs da área de tecnologia em Curitiba e região metropolitana (RMC) enfrenta desafios consideráveis na gestão de manutenção de equipamentos. A maioria desses profissionais realiza tanto manutenção preventiva quanto corretiva, utilizando, em grande parte, ferramentas improvisadas, como WhatsApp e planilhas eletrônicas, para gerenciar suas demandas.

A ausência de um sistema especializado para gerenciar essas manutenções resulta em dificuldades na localização de informações, atrasos na execução dos serviços, falta de métricas para avaliação de desempenho e desafios na gestão de peças e planejamento de demandas. Além disso, a falta de um histórico estruturado dos atendimentos prejudica a eficiência e a qualidade do serviço prestado, limitando a capacidade dos profissionais em expandir seus negócios e melhorar a satisfação do cliente.

O SIGMI surge como uma solução para esses problemas, oferecendo uma ferramenta de gestão simplificada, mas robusta, adaptada às necessidades dos microempreendedores individuais. Os benefícios esperados incluem maior agilidade no processo de manutenção, melhor controle e organização das informações, segurança no armazenamento de dados, e a possibilidade de gerar métricas e indicadores para a tomada de decisões estratégicas.

A aceitação expressiva da proposta do SIGMI pelos respondentes, dos quais 100% manifestaram interesse em participar de testes de demonstração, destaca a relevância e a necessidade de uma solução como essa no mercado. O sistema não só otimiza o trabalho realizado pelos MEIs, mas também melhora o atendimento ao cliente, permitindo uma comunicação mais transparente e eficaz ao longo de todo o processo de manutenção.

Como pretensões futuras, o sistema contará com módulo financeiro, solitação de atendimentos pelo usuário via web e também irá gerar QR Code com o histórico de manutenção de cada equipamento.

Em conclusão, o SIGMI se alinha perfeitamente às necessidades dos microempreendedores, oferecendo uma ferramenta acessível que contribui para a melhoria da eficiência operacional, a expansão do negócio e o fortalecimento econômico da região.

Referências bibliográficas

BULE, A. A. *Sistema de gestão de assistência técnica. O caso da Itec Solutions. Monografia (Licenciatura em Informática) – Faculdade de Ciências, Universidade Eduardo Mondlane. Maputo, 2022.*

FÁCIL SISTEMAS. *Sistema para loja de informática. 2023. Disponível em: <https://facilsistemas.com.br/sistema-para-loja-de-informatica/>. Acesso em: 04 out. 2023.*

FALCORA TECNOLOGIA. *Sistema de Gestão para Lojas de Informática. 2022. Disponível em: <https://falcora.com.br/sistema-de-gestao-para-lojas-de-informatica/>. Acesso em: 04 out. 2023.*

SATO, B. H. M.; OLIVEIRA, J. T. C. C.; ALVES, R. C. *Desenvolvimento de software de gestão de tickets de suporte. Monografia (Análise e Desenvolvimento de Sistemas)*

– Universidade de Taubaté. Taubaté, 2022.

SIGECLOUD. Sistema de Gestão para Assistência Técnica e Loja de Informática. 2021. Disponível em: <https://www.sigecloud.com.br/erp-para-lojas-de-informatica>. Acesso em: 04 out. 2023.

SILVA, D. E. et al. Desenvolvimento de um sistema de gerenciamento para lojas de informática e periféricos. Monografia (Análise e Desenvolvimento de Sistemas) - Faculdade de Tecnologia de Americana “Ministro Ralph Biasi”. Americana, 2022.