

RELATÓRIO TÉCNICO DO PROJETO

Sistema de Manutenção de Ativos

Projeto Integrador – Práticas de Programação

Prof. Me. Marcel Santos Silva

ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

Will Robson Alves da S. L. Jr. Cristo

17/06/2025

Resumo

A aplicação tem como objetivo aumentar a eficiência na manutenção dos ativos de uma empresa, evitando ociosidade, impactos negativos na produção e desperdícios de recursos. O sistema consiste em três cadastros principais: Ativos (Equipamentos), Manutenções Programadas e Ordens de Serviços. O projeto visa apresentar KPIs como Volumetria e Comparativo das diferenças entre os contadores.

1. Introdução

Em um ambiente empresarial com diversos equipamentos, faz-se necessário o controle sistemático das manutenções para manter a operacionalidade e eficiência da produção. A realização de manutenções preventivas e um bom planejamento baseado em dados históricos são essenciais para evitar desgastes prematuros e incidentes. Este documento discorre sobre a aplicação das "Práticas de Programação" aprendidas no curso de Análise e Desenvolvimento de Sistemas - UNIMAR (EAD).

1.1 Tecnologias e Ferramentas Utilizadas

1.1.1 Linguagem de Programação

A linguagem de programação selecionada para o projeto é o **Python**, devido à sua clareza, simplicidade e vasta biblioteca. Sua sintaxe limpa e ecossistema

versátil permitem a integração com outras tecnologias e futuras expansões em áreas como análise de dados e inteligência artificial.

1.1.2 Framework Web

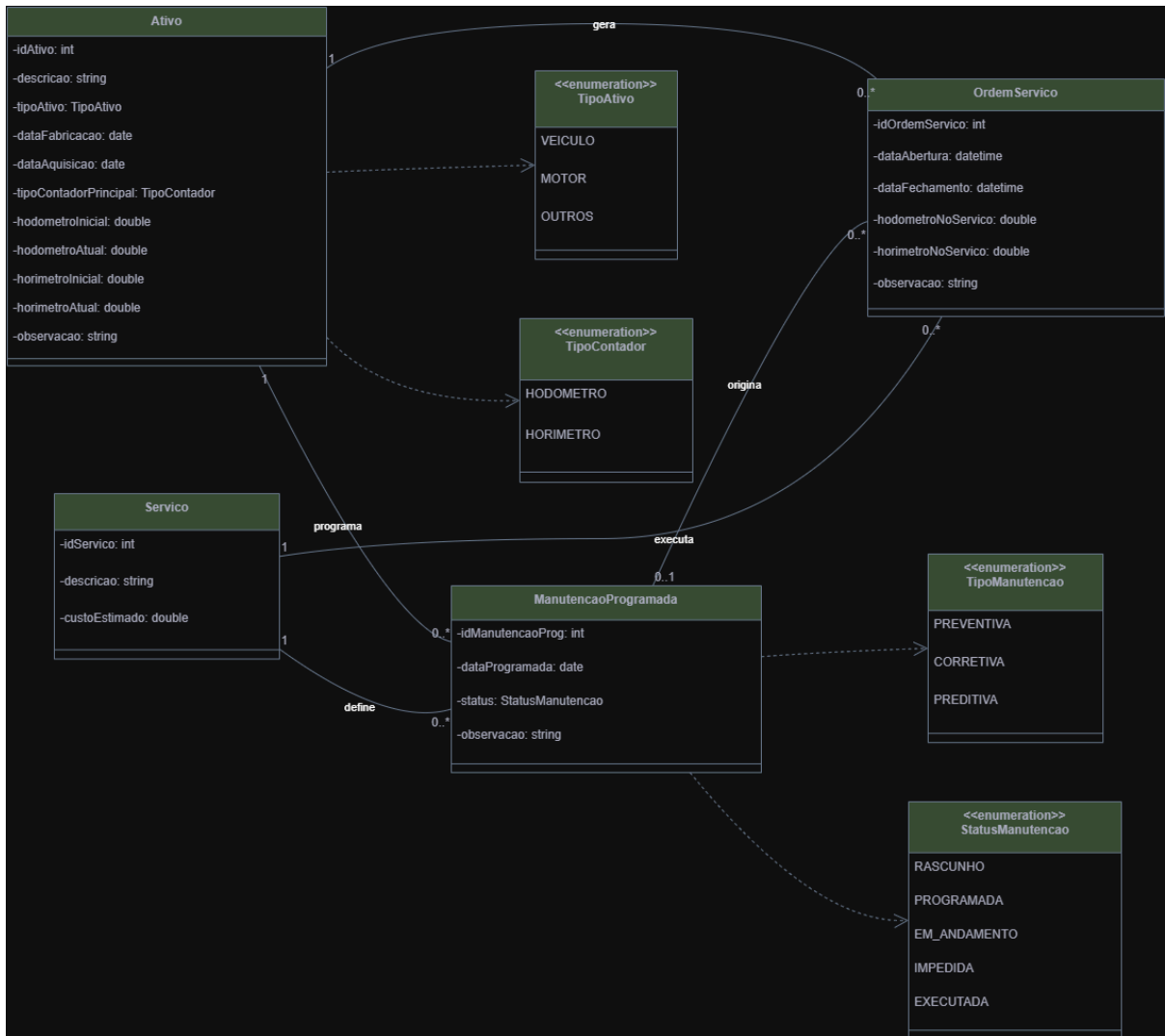
O framework escolhido foi o **Django**, por sua filosofia "Batteries-Included", oferecendo um ecossistema completo e robusto. O Django fornece um poderoso **ORM (Object-Relational Mapper)**, interface de administração (Admin Panel) e promove a "Convenção sobre Configuração", resultando em um código consistente e seguro.

1.1.3 Banco de Dados

O sistema de gerenciamento de banco de dados escolhido foi o **PostgreSQL**, devido à sua capacidade superior de lidar com dados complexos e semi estruturados, suporte ao tipo de dado JSONB e arquitetura extensível. Sua confiabilidade e escalabilidade garantem alta performance e integridade de dados.

1.1.4 Diagrama de Classe

[Diagrama de Classe do Sistema de Manutenção de Ativos]



2. Arquitetura e Implementação do Projeto

2.1 Modelagem de Dados

Detalhes da implementação dos modelos de dados:

- **Modelo Ativo:** Descrição, Tipo de Ativo, Data de Fabricação, Data de Aquisição, Tipo de contador principal, Hodômetro inicial/atual, Horímetro inicial/atual, Observação.
- **Modelo ManutencaoProgramada:** Ativo, Data Programada, Tipo de Manutenção, Serviço, Status, Observação. Relacionamento com o modelo Ativo.
- **Modelo OrdemServico:** Ativo, Data, Serviço, Contadores Atuais, Manutenção Origem, Observação. Relacionamento com o modelo Ativo e ManutencaoProgramada.

Os scripts de criação da estrutura do banco de dados serão entregues separadamente. A arquitetura geral da aplicação web é baseada no framework Django, seguindo o padrão MVT - Model-View-Template.

2.2 Implementação das Telas (Visões e Templates)

2.2.1 Acesso à Demonstração

<https://sites.google.com/view/demonstracao-manutencao-ativos/home>

2.2.2 Telas de Cadastro (CRUD)

O sistema dispõe dos cadastros de Ativos, Programações e Ordens de Serviços. Cada tela permite as operações de Inserção, Atualização e Exclusão de registros.

2.2.3 Tela de Consulta/Pesquisa

É possível consultar os cadastros digitando no campo de pesquisa pela descrição do ativo.

2.2.4 Funcionalidades Adicionais

- A tela de Dashboard apresenta indicadores (KPI) como quantidade total de ativos, quantidade de programações cadastradas e quantidade de ordens de serviço.
- A tela possui responsividade, ajustando automaticamente a escala, tamanho e posição dos elementos.

3. Resultados e Considerações Finais

O desenvolvimento do projeto "Sistema de Manutenção de Ativos" culminou na entrega de uma aplicação web funcional que atende aos objetivos propostos na introdução deste relatório. A plataforma desenvolvida oferece uma solução robusta para o gerenciamento sistemático de equipamentos, visando aumentar a eficiência operacional e a vida útil dos ativos.

Os principais resultados alcançados incluem:

Implementação dos Módulos de Cadastro (CRUD): Foram desenvolvidos com sucesso os três cadastros centrais do sistema: Ativos, Manutenções Programadas e Ordens de Serviço. Cada módulo permite as operações

essenciais de criação, leitura, atualização e exclusão de registros, formando a base operacional da ferramenta.

Desenvolvimento de um Dashboard de Indicadores (KPIs): Uma tela de Dashboard foi criada para apresentar indicadores chave de desempenho, como a quantidade total de ativos, o número de programações de manutenção e o total de ordens de serviço cadastradas.

Funcionalidades Adicionais: O sistema conta com uma ferramenta de pesquisa que permite a consulta rápida de ativos por sua descrição. Além disso, a aplicação foi desenvolvida com um design responsivo, garantindo uma experiência de uso consistente em diferentes dispositivos ao ajustar automaticamente o layout dos elementos.

Um dos principais desafios do projeto foi a integração das tecnologias escolhidas — Python , Django e PostgreSQL — para criar uma arquitetura coesa e escalável, seguindo o padrão MVT (Model-View-Template). A modelagem de dados, com a definição de relacionamentos entre Ativos, Manutenções e Ordens de Serviço, foi uma etapa crucial que exigiu atenção para garantir a integridade e a lógica do negócio.

A aplicação representa um avanço significativo no controle de manutenções, saindo de um possível cenário de controles manuais para uma gestão centralizada e baseada em dados. Para evoluções futuras, sugere-se a implementação das seguintes melhorias:

Expansão dos KPIs: Desenvolver os indicadores de Volumetria e o comparativo entre contadores, conforme vislumbrado inicialmente no projeto.

Sistema de Notificações: Criar alertas automáticos para gestores sobre manutenções programadas que estão próximas do vencimento.

Autenticação de Usuários: Implementar um sistema de login com diferentes níveis de permissão (administrador, técnico, gestor).

Relatórios Avançados: Gerar relatórios detalhados sobre o histórico de manutenções por ativo, custos envolvidos e tempo de parada.

Integração com IA: Utilizar o ecossistema Python para desenvolver modelos de manutenção preditiva, analisando dados históricos para prever falhas antes que ocorram.

Conclui-se que o projeto foi bem-sucedido ao aplicar os conceitos de "Práticas de Programação" em uma solução prática e com grande potencial de otimização para o ambiente empresarial, promovendo a redução de custos e o aumento da produtividade através de um planejamento de manutenção mais eficaz.

4. Referências

1. **Tailwind CSS:** Utilizado para estilização, layout e responsividade. (<https://tailwindcss.com/docs>)
2. **Lucide Icons:** Biblioteca de ícones open-source. (<https://lucide.dev>)
3. **Google Fonts:** Uso da fonte "Inter". (<https://fonts.google.com/specimen/Inter>)
4. **Documentação do Django:** Uso de template tags e filtros. (<https://docs.djangoproject.com/en/stable/topics/templates>)
5. **PHOSPHOR ICONS:** Família de ícones flexíveis para interfaces. (<https://phosphoricons.com>)
6. **MDN WEB DOCS:** Documentação para desenvolvedores web, mantida pela Mozilla (criadora do Firefox) (<https://developer.mozilla.org/pt-BR/docs/Web/JavaScript>)

5. Apêndices

5.1 Link para o Repositório do Projeto

O código-fonte está disponível em:

<https://github.com/willjrcristo/manutencao-ativos-demo.git>