



Outubro 2024

ViaBilidade

Solução analítica de manutenções

SUMÁRIO

03 Integrantes

04 Desafio

05 Solução

06 Diagrama de Classes

07 Descrição das Classes

08 Diagrama de Entidade e Relacionamento

09 Funcionalidades do Código

10 End-points Configurados

Integrantes



Julia Rodrigues

Project Analyst



Henrique Batista

Back-End Developer



Felipe Gonçalves

Front-End Developer &
Designer

DESAFIO

Atualmente, a CCR enfrenta um grande desafio relacionado à manutenção de trens e trilhos. O tempo disponível para iniciar e concluir as manutenções é restrito a apenas 2h30 por dia, o que dificulta a identificação rápida do tipo de manutenção necessário e, muitas vezes, impede a conclusão da tarefa. Quando uma manutenção não é finalizada a tempo, o problema pode se agravar, levando a mais complicações.

Além disso, o controle das manutenções preventivas e corretivas ainda é feito manualmente, sem uma plataforma centralizada que permita aos técnicos visualizar ou entender claramente os problemas a serem resolvidos. A falta de um registro digital dificulta a identificação de manutenções recorrentes e o planejamento de processos mais eficientes para futuras intervenções.

SOLUÇÃO

Descrição geral do projeto

Nossa solução consiste em desenvolver uma plataforma que ofereça aos técnicos de manutenção uma visão clara e estruturada de todo o processo de manutenções, permitindo que compreendam o que precisa ser feito, como deve ser realizado e qual o fluxo necessário para cada tarefa, utilizando Procedimentos Operacionais Padrão (POP's).

A plataforma também registrará automaticamente o início e a conclusão de cada manutenção, permitindo que os gerentes de operação gerem relatórios detalhados. Esses relatórios ajudarão a monitorar o tempo gasto nas manutenções, identificar os POP's mais utilizados e detectar problemas recorrentes. Com essas informações, será possível tomar decisões mais embasadas, visando a implementação de melhorias que reduzam a repetição de manutenções.

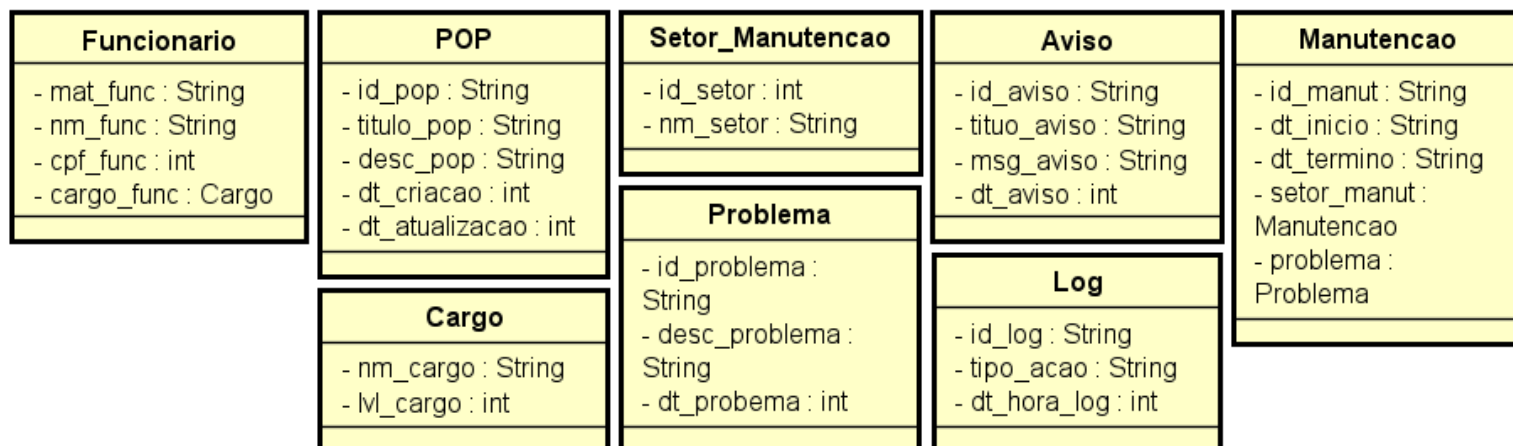
Essa solução otimizará o tempo entre manutenções, aumentará a eficiência operacional e contribuirá para uma gestão mais estratégica dos processos de manutenção.

Com a melhoria de manutenções, o usuário final sofrerá menos com problemas recorrentes na linha.



DIAGRAMA DE CLASSES

Aplicação visual das classes



DESCRIÇÃO DAS CLASSES

Funcionário

- **mat_func:** Matrícula do funcionário.
- **nm_func:** Nome do funcionário
- **cargo_func:** Cargo que o funcionário ocupa
- **cpf_func:** CPF do funcionário

Pop

- **id_pop:** Atributo de identificação do POP
- **titulo_pop:** Título do POP
- **desc_pop:** A descrição do POP
- **dt_criacao:** A data que foi criado o POP
- **dt_atualizacao:** A data que foi modificado o POP

Setor_Mantencao

- **id_setor:** Atributo de identificação do Setor
- **nm_setor:** Nome do Setor

Aviso

- **id_aviso:** Atributo de identificação do Aviso
- **titulo_aviso:** Título do Aviso
- **msg_aviso:** A mensagem de Aviso
- **dt_aviso:** Data do Aviso

Problema

- **id_problema:** Atributo de identificação do Problema
- **desc_problema:** A descrição do Problema
- **dt_problema:** Data do Problema

Cargo

- **nm_cargo:** Nome do Cargo
- **lvl_cargo:** O nível do Cargo (atribuido a privilégios)

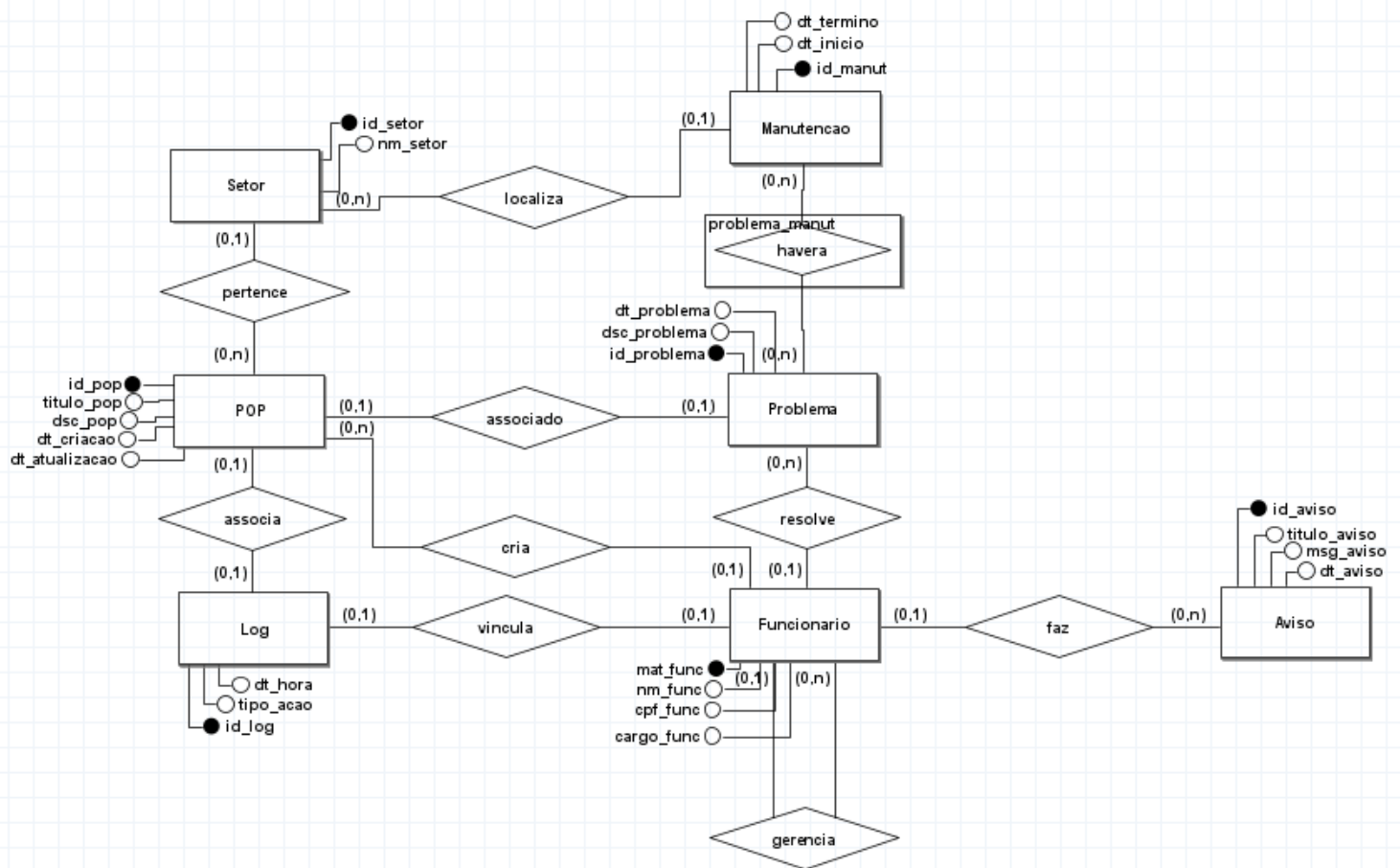
Log

- **id_log:** Atributo de identificação do Log de ação
- **tipo_acao:** Qual a ação da modificação/criação
- **dt_hora_log:** O dia e horário desta modificação/criação

Manutenção (*novo*)

- **id_manut:** Atributo de identificação da Manutenção
- **dt_inicio:** Em que momento foi iniciado a Manutenção
- **dt_termino:** Em que momento foi concluído a Manutenção

DIAGRAMA DE ENTIDADE E RELACIONAMENTO



FUNCIONALIDADES DO CÓDIGO

- **Verificação de Setor**

O sistema verifica qual setor a partir do número inserido.

```
static void mostrarNomeSetor(Setor_Mantencao setor){
    int num_setor = setor.getId();
    String setorText;
    if (num_setor == 1){
        JOptionPane.showMessageDialog( parentComponent: null, message: "Setor indicado: Elétrica");
        setorText = "eletrica";
    }
    else if (num_setor == 2){
        JOptionPane.showMessageDialog( parentComponent: null, message: "Setor indicado: Mecânica");
        setorText = "mecanica";
    }
    else if (num_setor == 3){
        JOptionPane.showMessageDialog( parentComponent: null, message: "Setor indicado: Hidráulica");
        setorText = "hidraulica";
    }
    else if (num_setor == 4){
        JOptionPane.showMessageDialog( parentComponent: null, message: "Setor indicado: Interno");
        setorText = "interno";
    }
    else{
        JOptionPane.showMessageDialog( parentComponent: null, message: "Não inserido corretamente");
    }
}
```

- **Fluxo de Cadastro**

O fluxo para o cadastro é: Funcionarios > POPs > Avisos > Problema > Cargo > Manutenções.

```
JOptionPane.showMessageDialog( parentComponent: null, funcionario);
JOptionPane.showMessageDialog( parentComponent: null, pop);
JOptionPane.showMessageDialog( parentComponent: null, aviso);
JOptionPane.showMessageDialog( parentComponent: null, problema);
JOptionPane.showMessageDialog( parentComponent: null, cargo);
JOptionPane.showMessageDialog( parentComponent: null, manutencao);
```

END-POINTS CONFIGURADOS

Recurso	URI	Verbo HTTP	Descrição	Códigos de Status
Cargo	/cargos	GET	Listar todos os cargos	200 OK
Cargo	/cargos	POST	Inserir um novo cargo	201 Created, 400 Bad
Cargo	/cargos/{id}	PUT	Atualizar cargo existente	200 OK, 404 Not Found
Cargo	/cargos/{id}	DELETE	Deletar cargo pelo ID	200 OK, 404 Not Found
Aviso	/avisos	GET	Listar todos os avisos	200 OK
Aviso	/avisos	POST	Inserir um novo aviso	201 Created, 400 Bad
Aviso	/avisos/{id}	PUT	Atualizar aviso existente	200 OK, 404 Not Found
Aviso	/avisos/{id}	DELETE	Deletar aviso pelo ID	200 OK, 404 Not Found
Greeting	/login	GET	Mensagem de saudação	200 OK