

Sistema de Gestão de Transporte Público

Paulo Ricardo dos Reis, Vinícios Bidin

Departamento de Ciência da Computação

Universidade do Estado de Santa Catarina (UDESC)

1. Descrição do problema

Nosso software é um sistema de gestão de transporte público e visa solucionar diversos problemas relacionados à operação e administração desse tipo de serviço:

- Otimizar o planejamento das rotas, horários e alocação de recursos, levando em consideração a demanda de passageiros, a capacidade dos veículos e outras variáveis relevantes.
- Auxiliar no controle e manutenção da frota de veículos, fornecendo informações sobre o estado dos veículos, programação da manutenção preventiva, gerenciamento de combustível e monitoramento do desempenho dos veículos.
- Ajuda a integrar outros sistemas públicos de transportes como metrô e trem, permitindo uma visão unificada da operação e facilidade na transferência entre modos e uma melhor experiência para os passageiros.
- Oferece recursos como aplicativo móvel para consulta de horários, planejamento de rotas, informações em tempo real sobre a chegada dos veículos, sistemas de pagamento eletrônico e outras funcionalidades que melhoram a experiência dos passageiros.

1.1 Stakeholders

- **Passageiros:** pessoas que utilizam do transporte público com frequência, pois são diretamente afetados pela qualidade, disponibilidade, confiabilidade e custo do serviço
- **Funcionários:** motoristas, operadores, técnicos, administradores, e outros funcionários envolvidos na operação e gestão do transporte público.
- **Empresas de transporte público:** empresas que fazem a gestão do transporte público, já que o software tem várias funcionalidades úteis para a empresa como otimização de planejamento das rotas, horários e alocação de recursos, auxilia na manutenção da frota, entre outros

2. Requisitos funcionais e não-funcionais

O software deve seguir tais requisitos funcionais e não-funcionais listados abaixo:

2.1 Requisitos funcionais

- **Planejamento de rotas:** deve permitir o planejamento eficiente das rotas, considerando demanda de passageiros, horários, capacidade dos veículos e restrições operacionais.
- **Alocação de recursos:** deve ser possível realizar a alocação adequada de veículos, motoristas e outros recursos necessários para garantir a operação eficiente do sistema.
- **Gestão de frota:** deve ser possível monitorar o estado dos veículos, programar manutenção preventiva, registrar dados de desempenho e gerenciar abastecimento de combustível.

- **Cadastro de usuários:** deve ser possível cadastrar usuários sejam eles funcionários da empresa ou usuários do transporte.
- **Consulta de horários:** deve ser possível realizar consultas de horários das linhas de transporte público assim como informações em tempo real sobre a chegada dos veículos.
- **Pagamento eletrônico:** deve ser possível a realização do pagamento da passagem antecipada direto no software.
- **Integração de sistemas:** deve ser capaz de integrar diferentes meios de transportes, como ônibus, metrô, trens, permitindo uma visão unificada e facilitando a transferência entre modos.

2.2 Requisitos não-funcionais

- **Confiabilidade:** deve ser confiável, minimizando a ocorrência de falhas e interrupções no serviço..
- **Desempenho:** deve ter um desempenho eficiente, processando e apresentando informações em tempo real e sem atrasos significativos.
- **Usabilidade:** a interface do usuário tem que ser intuitiva e de fácil utilização, considerando diferentes perfis de usuário.
- **Segurança:** o sistema deve garantir a segurança dos dados pessoais dos usuário cadastrados com criptografia adequada.
- **Integração com sistemas externos:** deve ser capaz de integrar-se a outros sistemas externos, como sistemas de pagamento eletrônico, sistemas de informação aos passageiros e sistemas de controle de tráfego.
- **Acessibilidade:** deve ser acessível a todos os usuários, incluindo pessoas com necessidades especiais, fornecendo recursos e interfaces adequadas aos padrões.

3. Estimativa da duração do projeto

Para calcular uma estimativa para o tempo de duração do projeto utilizaremos o modelo COCOMO, onde primeiros precisamos contar os elementos do software, seja eles, números de Entradas Externas (EE), Saídas Externas(SE), Consultas externas(CE), Arquivos Lógicos Internos(ALI) e Arquivos de Interface Externos(AIE). Segue a contagem abaixo e a definição do nível de complexidade definido de acordo com a quantidade de campos e entidades envolvidas:

- Número de Entradas Externas (EE):
 1. Cadastro novo usuário (baixo)
 2. Cadastro novo funcionário (baixo)
 3. Cadastro nova rota (baixo)
 4. Cadastro novo veículo (baixo)
 5. Cadastro novo cargo (baixo)
 6. Cadastro novo feedback (baixo)
 7. Alteração dados usuário (média)
 8. Alteração dados funcionário (média)
 9. Alteração dados rota (média)
 10. Alteração dados veículo (média)
 11. Alteração dados cargo (média)

- Número de Saídas Externas (SE):
 1. Relatório de novos usuários (média)
 2. Relatório de novos feedbacks (média)
- Número de Consultas Externas (CE):
 1. Buscas de rotas (baixo)
 2. Buscas de veículos (baixo)
- Número de Arquivos Lógicos Internos (ALI):
 1. Arquivo para tabela de usuário (média)
 2. Arquivo para tabela de funcionário (média)
 3. Arquivo para tabela de rota (média)
 4. Arquivo para tabela de veículo (média)
 5. Arquivo para tabela de cargo (média)
 6. Arquivo para tabela de feedback (média)
 7. Mensagem de login inválido (baixo)
 8. Mensagem de feedback enviado (baixo)
 9. Mensagem de cadastro (baixo)
 10. Mensagem de erro de alterações (baixo)
- Número de Arquivos de Interface Externos (AIE):
 1. Interface externa para sistema de transporte público (média)

Feita a contagem dos elementos de software, precisamos achar os pontos de função não ajustados (PFNA). Usaremos a tabela de pesos para cada elemento conforme a tabela abaixo:

Elemento\Complexidade	Baixa	Média	Alta
Entradas Externas (EE)	3	4	6
Saídas Externas (SE)	4	5	7
Consultas Externas (CE)	3	4	6
Arquivos Lógicos Internos (ALI)	7	10	15
Arquivos de Interface Externos (AIE)	5	7	10

Figura 1. Tabela pesos para cada elemento conforme complexidade

Para o cálculo usaremos a fórmula:

$$PFNA = \sum \text{elemento} \times \text{peso}$$

No nosso caso, teremos:

$$\begin{aligned}
 EE &= 6 \times 3 + 5 \times 4 = 38 \\
 SE &= 2 \times 5 = 10 \\
 CE &= 2 \times 4 = 8 \\
 ALI &= 4 \times 7 + 6 \times 10 = 88
 \end{aligned}$$

$$AIE = 1 \times 7 = 7$$

Fazendo a soma de todos:

$$PFNA = 38 + 10 + 8 + 88 + 7 = 151$$

Com os PFNA calculados, o próximo passo é calcular o valor das linhas de código (LOC). Como o programa é escrito em JAVA, cada PFNA equivale a 53 em LOC, conforme a tabela abaixo:

Linguagem	LOC
FORTTRAN 77	107
Java	53
PASCAL	91
PERL	27

Figura 2. Conversão PFNA em LOC

Então, para nosso projeto teremos:

$$LOC = 151 \times 53 = 8003$$

Transformando esse valor para KLOCS, temos 8,003 KLOCS. Com esse valor podemos calcular o ESFORÇO (calculado em pessoa-mês) e a DURAÇÃO (em meses) do projeto. Para o cálculo, definimos a complexidade do projeto como média, então teremos:

$$Esforço = 3 \times KLOC^{1,12}$$

$$Esforço = 3 \times 8,003^{1,12}$$

$$Esforço = 31pm$$

$$Duração = 2,5 \times Esforço^{0,35}$$

$$Duração = 2,5 \times 31^{0,35}$$

$$Duração = 9 meses$$

Com isso, utilizando a estimativa COCOMO, temos que o projeto teria DURAÇÃO de 9 meses e ESFORÇO de 31 pessoas-mês.

4. Diagrama de Classes UML

O diagrama UML está na pasta junto deste arquivo para melhor visualização.

5. Testes Unitários

Para validação do código e das funções implementadas, foram realizados Testes Unitários, que se encontram na pasta "Test" junto deste arquivo.

Como o código desenvolvido é apenas um começo do projeto, os testes aplicados cobrem apenas as partes mais críticas de cada classe do código. Abaixo segue a listagem do porquê da realização destes testes em cada classe:

- **Classe Cargo:** para garantir que o funcionário seja registrado em seu cargo corretamente.
- **Classe Categoria Transporte:** para garantir que a rota seja atribuída a uma categoria de transporte corretamente.
- **Classe Passageiro:** para garantir que o feedback seja registrado corretamente e a compra de passagem seja registrada corretamente.
- **Classe Rota:** para garantir que o veículo seja atribuído a uma rota corretamente e ainda, que seja uma rota válida, considerando que uma rota não pode ter o mesmo destino e origem, ou então uma distância menor ou igual à zero.
- **Classe Veículo:** para garantir que o funcionário seja atribuído a um veículo corretamente e que passageiros também sejam atribuídos corretamente.

As categorias Feedback, Funcionário , Passagem não possuem métodos considerados críticos, por isso a ausência de testes nas mesmas.