

Análise de Requisitos

Requisitos Funcionais:

1. Cadastro de Veículos:

- O sistema deve permitir que os usuários cadastrem novos veículos, incluindo informações como tipo, marca, modelo e ano.

2. Registro Único dos Veículos:

- Cada veículo cadastrado deve ter um identificador único no sistema.

3. Associação com Motorista:

- Durante o cadastro de um veículo, o usuário deve ser capaz de associá-lo a um motorista responsável.

4. Validação de Motorista:

- Ao associar um veículo a um motorista, o sistema deve verificar se o motorista é maior de idade e possui a habilitação adequada para o tipo de veículo.

5. Operações de Veículos:

- Os usuários devem ser capazes de consultar, atualizar e excluir informações sobre veículos cadastrados no sistema.

6. Cadastro de Motoristas:

- O sistema deve permitir que os usuários cadastrem novos motoristas, incluindo informações como nome, idade e tipo de habilitação.

7. Atualização de Motoristas:

- Os usuários devem poder atualizar informações dos motoristas cadastrados, como nome, idade e tipo de habilitação.

8. Relatórios:

- O sistema deve possibilitar a geração de relatórios, incluindo:

- Lista de veículos associados a um determinado motorista.
- Lista de todos os veículos cadastrados.
- Lista de todos os motoristas cadastrados.

Requisitos Não Funcionais:

1. Integridade dos Dados:

- Os dados armazenados no sistema devem ser consistentes e precisos, garantindo a integridade dos registros.

2. Usabilidade:

- A interface do sistema deve ser intuitiva e fácil de usar, permitindo que os usuários realizem as operações de forma eficiente.

3. Desempenho:

- O sistema deve ser capaz de lidar com um grande volume de dados e realizar operações de forma rápida e responsiva.

4. Segurança:

- O acesso às informações do sistema deve ser controlado e protegido, garantindo que apenas usuários autorizados possam visualizar e modificar os dados.

5. Portabilidade:

- O sistema deve ser desenvolvido de forma a ser facilmente portátil para diferentes plataformas e ambientes de execução.

6. Manutenibilidade:

- O código-fonte do sistema deve ser bem organizado e documentado, facilitando futuras atualizações e manutenções.