**Logotipo, nome da empresa

Descrição gerada automaticamente Sistema de Recomendação de Carros**

   
 **Nome do Estudante:** Wander Jean Hanemann  
**Curso:** Engenharia de Software    
**Data de Entrega:** 12/04/2024

**Resumo**

O projeto consiste no desenvolvimento de um Sistema de Recomendação de Carros que utilizará algoritmos de inteligência artificial para recomendar veículos com base nas preferências e necessidades dos usuários. O sistema visa facilitar a escolha de um carro, fornecendo informações detalhadas sobre os veículos recomendados, como especificações técnicas, avaliações de especialistas e opiniões de consumidores.

**1. Introdução:**

**Contexto:** No cenário atual, a escolha de um carro pode ser complexa devido à variedade de opções disponíveis no mercado automotivo. Um sistema de recomendação de carros pode ajudar os consumidores a tomarem decisões mais informadas.

**Justificativa:** O sistema é relevante para o campo da engenharia de software por sua capacidade de aplicar algoritmos de inteligência artificial para analisar as preferências dos usuários e recomendar carros que atendam às suas necessidades específicas.

**Objetivos:** O objetivo principal do projeto é desenvolver um sistema que recomende carros com base nas preferências e necessidades dos usuários. Além disso, o sistema deve ser capaz de fornecer informações detalhadas sobre os carros recomendados, como especificações técnicas, avaliações de especialistas e opiniões de consumidores.

**2. Descrição do Projeto:**

**Tema do Projeto:** O sistema de recomendação de carros é uma aplicação de inteligência artificial que analisa as preferências dos usuários e recomenda veículos com base em critérios como marca, modelo, preço, desempenho, eficiência energética, segurança, entre outros.

**Problemas a Resolver:** O projeto visa resolver os problemas enfrentados pelos consumidores ao escolher um carro, como a falta de informação sobre as opções disponíveis e a dificuldade de comparar diferentes modelos.

**Limitações:** O sistema tem como limitação a dependência de dados precisos e atualizados sobre os veículos disponíveis no mercado, bem como a necessidade de garantir a privacidade e segurança dos dados dos usuários.

**3. Especificação Técnica**

**3.1. Requisitos de Software**

**Requisitos Funcionais (RF):**

O sistema deve permitir aos usuários especificar suas preferências em relação a características como marca, tipo de carro, consumo de combustível, potência do motor, entre outros.

O sistema deve recomendar carros com base nas preferências dos usuários e em dados disponíveis sobre os veículos.

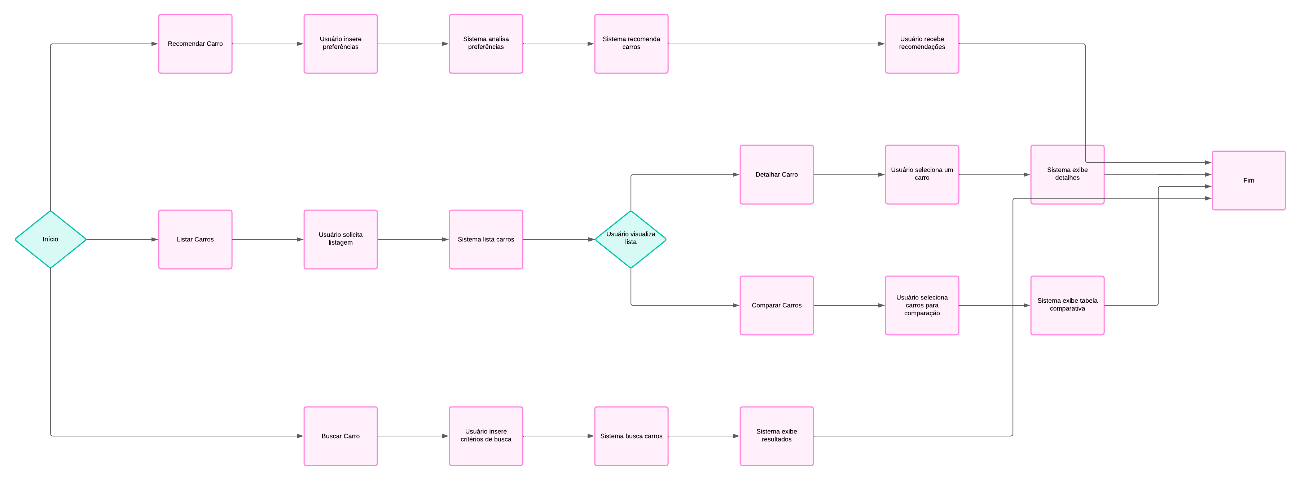
O sistema deve fornecer informações detalhadas sobre os carros recomendados, incluindo especificações técnicas, avaliações e opiniões de consumidores.

**Requisitos Não Funcionais (RNF):**

O sistema deve ter uma interface intuitiva e fácil de usar.

O sistema deve ser capaz de lidar com uma grande quantidade de dados de forma eficiente.

O sistema deve garantir a privacidade e segurança dos dados dos usuários.



**3.2. Considerações de Design**

**• Considerações de Design:**

O sistema utilizará algoritmos de filtragem colaborativa e baseada em conteúdo para recomendar carros com base nas preferências dos usuários e nas características dos veículos.

A interface do usuário será projetada para permitir aos usuários especificar suas preferências de forma clara e intuitiva, e para apresentar as recomendações de forma organizada e informativa.

**• Visão Inicial da Arquitetura:**

O sistema será composto por três principais componentes: a interface de usuário, o motor de recomendação e o banco de dados de veículos.

A interface de usuário permitirá aos usuários especificar suas preferências e visualizar as recomendações de carros.

O motor de recomendação será responsável por analisar as preferências dos usuários e recomendar carros com base nessas preferências e nos dados disponíveis sobre os veículos.

O banco de dados de veículos conterá informações detalhadas sobre os carros disponíveis no mercado, incluindo especificações técnicas, avaliações e opiniões de consumidores.

**• Padrões de Arquitetura:**

O sistema seguirá o padrão arquitetural MVC (Model-View-Controller) para separar as camadas de apresentação, lógica de negócio e armazenamento de dados.

**3.3. Stack Tecnológica**

**• Linguagens de Programação:**

**Python:** Escolhido por ser uma linguagem versátil, com uma vasta gama de bibliotecas para IA, como TensorFlow e Scikit-learn. Sua sintaxe simples e legível facilita o desenvolvimento rápido e eficiente.

**JavaScript (Node.js):** Utilizado para o desenvolvimento do frontend, devido à sua ampla adoção e suporte para aplicações web.

**SQL:** Utilizado para interação com o banco de dados, devido à sua eficiência e capacidade de lidar com grandes conjuntos de dados.

**• Frameworks e Bibliotecas:**

**Flask (Python):** Escolhido para o desenvolvimento do backend devido à sua simplicidade e extensibilidade.

**React (JavaScript):** Utilizado para o desenvolvimento do frontend, devido à sua eficiência na criação de interfaces de usuário interativas. O React facilita a construção de interfaces de usuário dinâmicas e responsivas.

**TensorFlow (Python):** Utilizado para implementação de algoritmos de inteligência artificial, devido à sua eficácia e ampla adoção na comunidade. O TensorFlow oferece uma ampla gama de ferramentas e recursos para o desenvolvimento de modelos de IA.

**• Ferramentas de Desenvolvimento e Gestão de Projeto:**

**Git:** Utilizado para controle de versão do código fonte. O Git permite o trabalho colaborativo e o rastreamento de alterações no código.

**GitHub:** Utilizado para hospedagem do repositório Git, oferecendo recursos de gerenciamento de problemas e integração contínua.

**Trello:** Utilizado para gestão de tarefas e acompanhamento do progresso do projeto. O Trello permite a organização e distribuição de tarefas de forma eficiente.

**VS Code**: Utilizado como ambiente de desenvolvimento integrado devido à sua extensibilidade e suporte a diversas linguagens de programação. O VS Code oferece recursos avançados de edição e depuração de código.

**3.4. Considerações de Segurança**

**• Proteção de Dados Sensíveis:**

**Questão de Segurança:** Os dados pessoais dos usuários, como informações de contato e preferências de carro, devem ser protegidos contra acesso não autorizado.

**Mitigação:** Utilizar técnicas de criptografia para proteger os dados sensíveis armazenados no banco de dados. Implementar políticas de acesso baseadas em funções para garantir que apenas usuários autorizados possam acessar esses dados.

**• Prevenção de Ataques de Injeção de SQL:**

**Questão de Segurança:** O sistema deve ser protegido contra ataques de injeção de SQL, nos quais um invasor tenta inserir comandos SQL maliciosos através de entradas de usuário.

**Mitigação:** Utilizar consultas parametrizadas ou ORM (Object-Relational Mapping) para evitar a construção de consultas SQL concatenando strings com entradas do usuário. Além disso, realizar validação e sanitização de entrada de dados para garantir que apenas dados válidos sejam aceitos.

**• Controle de Acesso:**

**Questão de Segurança:** Garantir que apenas usuários autorizados tenham acesso às funcionalidades e dados do sistema.

**Mitigação:** Implementar autenticação e autorização robustas. Utilizar tokens de sessão seguros e técnicas como OAuth para autenticação de usuários. Implementar políticas de controle de acesso baseadas em funções para limitar o acesso de usuários a dados e funcionalidades específicas.

**4. Próximos Passos**

Após a conclusão deste documento, os próximos passos para o desenvolvimento do Sistema de Recomendação de Carros incluem:

**Implementação dos Casos de Uso:** Iniciar a implementação dos casos de uso descritos neste documento, começando pelos mais críticos para a funcionalidade básica do sistema.

**Testes de Integração e Validação:** Realizar testes de integração para garantir que os diferentes componentes do sistema funcionem corretamente juntos. Em seguida, realizar testes de validação para garantir que o sistema atenda aos requisitos do usuário.

**Implementação de Melhorias e Ajustes:** Com base nos testes realizados e no feedback dos usuários, realizar melhorias no sistema e fazer ajustes onde necessário para garantir sua eficácia e usabilidade.

**Implementação de Recursos Adicionais:** Implementar recursos adicionais que possam melhorar a experiência do usuário e adicionar valor ao sistema, como personalização de recomendações e integração com redes sociais.

**Preparação para Portfólio II:** Preparar o sistema para a apresentação no Portfólio II, garantindo que todos os requisitos do projeto sejam atendidos e que o sistema esteja pronto para uso.

**Avaliação de Desempenho:** Realizar testes de desempenho para garantir que o sistema seja capaz de lidar com um grande volume de usuários e dados sem comprometer a velocidade e a eficiência.

**Documentação completa:** Finalizar a documentação do projeto, incluindo manuais de usuário e administrador, para facilitar a utilização e manutenção do sistema no futuro.

**Entrega Final do Projeto:** Após a conclusão de todos os passos anteriores, realizar a entrega final do projeto, garantindo que todos os requisitos tenham sido atendidos e que o sistema esteja pronto para uso.

## 5. Referências Flask: <https://flask.palletsprojects.com>

## React: <https://pt-br.legacy.reactjs.org/>

## TensowFlow: <https://www.tensorflow.org>

## JavaScript: <https://developer.mozilla.org/pt-BR/docs/Web/JavaScript>

## Python: <https://docs.python.org> SQL: <https://www.oracle.com/br/database/technologies/appdev/sql.html>