

FACULDADE DE INFORMÁTICA E ADMINISTRAÇÃO PAULISTA

Projeto: Sistema Automotivo - Diagnóstico, Orçamento e Agendamento de Serviços

Análise e Desenvolvimento de Sistemas

RM - 555368 - Lucas Kenji Miyahira

RM - 557837 - Rafael Rodrigues de Almeida

RM - 557753 - Wesley de Souza Oliveira

Sumário

1. Introdução
2. Visão Geral da Solução
3. Funcionalidades
4. Endpoints da API Restful
5. Protótipos de Telas
6. Modelo do Banco de Dados
7. Diagrama de Classes
8. Procedimentos para Rodar a Aplicação

1. Introdução

Este projeto oferece uma solução integrada para o setor automotivo, digitalizando e automatizando o diagnóstico, orçamento e agendamento de serviços. A proposta é fornecer aos clientes uma experiência prática e transparente, otimizada para as oficinas parceiras.

2. Visão Geral da Solução

A plataforma utiliza inteligência artificial para processar sintomas descritos pelo cliente e gerar diagnósticos automatizados, além de gerar orçamentos e agendar serviços diretamente na plataforma.

3. Funcionalidades

1. **Diagnóstico Online:** Analisa sintomas descritos pelo usuário e sugere causas prováveis usando IA.
 2. **Geração de Orçamentos:** Estima os custos com base nos serviços e peças necessários.
 3. **Agendamento de Serviços:** Permite ao cliente agendar o serviço conforme disponibilidade.
-

4. Tabela de Endpoints da API Restful

Método	Endpoint	Descrição
DELETE	http://localhost:8080/ideiasinaction_war/api/rest/agendamento/{id}	Exclui um agendamento específico pelo ID
GET	http://localhost:8080/ideiasinaction_war/api/rest/agendamento/all	Retorna uma lista de todos os agendamentos
POST	http://localhost:8080/ideiasinaction_war/api/rest/agendamento/cadastre	Criação de um novo agendamento
PUT	http://localhost:8080/ideiasinaction_war/api/rest/agendamento/{id}	Atualiza um agendamento específico pelo ID
DELETE	http://localhost:8080/ideiasinaction_war/api/rest/cliente/{id}	Exclui um cliente específico pelo ID
GET	http://localhost:8080/ideiasinaction_war/api/rest/cliente/all	Retorna uma lista de todos os clientes
POST	http://localhost:8080/ideiasinaction_war/api/rest/cliente	Criação de um novo cliente
PUT	http://localhost:8080/ideiasinaction_war/api/rest/cliente/{id}	Atualiza um cliente específico pelo ID
DELETE	http://localhost:8080/ideiasinaction_war/api/rest/endereco-cliente/{id}	Exclui um endereço de cliente específico
GET	http://localhost:8080/ideiasinaction_war/api/rest/endereco-cliente/all	Retorna uma lista de todos os endereços dos clientes
POST	http://localhost:8080/ideiasinaction_war/api/rest/endereco-cliente	Criação de um novo endereço de cliente
PUT	http://localhost:8080/ideiasinaction_war/api/rest/endereco-cliente/{id}	Atualiza um endereço de cliente específico
DELETE	http://localhost:8080/ideiasinaction_war/api/rest/endereco-oficina/{id}	Exclui um endereço de oficina específico
GET	http://localhost:8080/ideiasinaction_war/api/rest/endereco-oficina/all	Retorna uma lista de todos os endereços das oficinas

POST	http://localhost:8080/ideiasinaction_war/api/rest/endereco-oficina	Criação de um novo endereço de oficina
PUT	http://localhost:8080/ideiasinaction_war/api/rest/endereco-oficina/{id}	Atualiza um endereço de oficina específico
DELETE	http://localhost:8080/ideiasinaction_war/api/rest/falhas/{id}	Exclui um registro de falha específico pelo ID
GET	http://localhost:8080/ideiasinaction_war/api/rest/falhas/all	Retorna uma lista de todas as falhas
POST	http://localhost:8080/ideiasinaction_war/api/rest/falhas	Criação de um novo registro de falha
PUT	http://localhost:8080/ideiasinaction_war/api/rest/falhas/{id}	Atualiza um registro de falha específico
DELETE	http://localhost:8080/ideiasinaction_war/api/rest/funcionario/{id}	Exclui um funcionário específico pelo ID
GET	http://localhost:8080/ideiasinaction_war/api/rest/funcionario/all	Retorna uma lista de todos os funcionários
POST	http://localhost:8080/ideiasinaction_war/api/rest/funcionario	Criação de um novo funcionário
PUT	http://localhost:8080/ideiasinaction_war/api/rest/funcionario/{id}	Atualiza um funcionário específico pelo ID
DELETE	http://localhost:8080/ideiasinaction_war/api/rest/oficina/{id}	Exclui uma oficina específica pelo ID
GET	http://localhost:8080/ideiasinaction_war/api/rest/oficina/all	Retorna uma lista de todas as oficinas
POST	http://localhost:8080/ideiasinaction_war/api/rest/oficina	Criação de uma nova oficina
PUT	http://localhost:8080/ideiasinaction_war/api/rest/oficina/{id}	Atualiza uma oficina específica pelo ID
DELETE	http://localhost:8080/ideiasinaction_war/api/rest/orcamento/{id}	Exclui um orçamento específico pelo ID
GET	http://localhost:8080/ideiasinaction_war/api/rest/orcamento/all	Retorna uma lista de todos os orçamentos
POST	http://localhost:8080/ideiasinaction_war/api/rest/orcamento	Criação de um novo orçamento
PUT	http://localhost:8080/ideiasinaction_war/api/rest/orcamento/{id}	Atualiza um orçamento específico pelo ID
DELETE	http://localhost:8080/ideiasinaction_war/api/rest/ordemdeservico/{id}	Exclui uma ordem de serviço específica pelo ID
GET	http://localhost:8080/ideiasinaction_war/api/rest/ordemdeservico/all	Retorna uma lista de todas as ordens de serviço
POST	http://localhost:8080/ideiasinaction_war/api/rest/ordemdeservico	Criação de uma nova ordem de serviço
PUT	http://localhost:8080/ideiasinaction_war/api/rest/ordemdeservico/{id}	Atualiza uma ordem de serviço específica
DELETE	http://localhost:8080/ideiasinaction_war/api/rest/pagamento/{id}	Exclui um pagamento específico pelo ID
GET	http://localhost:8080/ideiasinaction_war/api/rest/pagamento/all	Retorna uma lista de todos os pagamentos
POST	http://localhost:8080/ideiasinaction_war/api/rest/pagamento	Criação de um novo pagamento
PUT	http://localhost:8080/ideiasinaction_war/api/rest/pagamento/{id}	Atualiza um pagamento específico pelo ID
DELETE	http://localhost:8080/ideiasinaction_war/api/rest/pecas/{id}	Exclui uma peça específica pelo ID
GET	http://localhost:8080/ideiasinaction_war/api/rest/pecas/all	Retorna uma lista de todas as peças
POST	http://localhost:8080/ideiasinaction_war/api/rest/pecas	Criação de uma nova peça
PUT	http://localhost:8080/ideiasinaction_war/api/rest/pecas/{id}	Atualiza uma peça específica pelo ID
DELETE	http://localhost:8080/ideiasinaction_war/api/rest/servicos/{id}	Exclui um serviço específico pelo ID
GET	http://localhost:8080/ideiasinaction_war/api/rest/servicos/all	Retorna uma lista de todos os serviços
POST	http://localhost:8080/ideiasinaction_war/api/rest/servicos	Criação de um novo serviço
PUT	http://localhost:8080/ideiasinaction_war/api/rest/servicos/{id}	Atualiza um serviço específico pelo ID
DELETE	http://localhost:8080/ideiasinaction_war/api/rest/telefone-cliente/{id}	Exclui um telefone de cliente específico
GET	http://localhost:8080/ideiasinaction_war/api/rest/telefone-cliente/all	Retorna uma lista de todos os telefones dos clientes
POST	http://localhost:8080/ideiasinaction_war/api/rest/telefone-cliente	Criação de um novo telefone de cliente

PUT	http://localhost:8080/ideiasinaction_war/api/rest/telefone-cliente/{id}	Atualiza um telefone de cliente específico
DELETE	http://localhost:8080/ideiasinaction_war/api/rest/telefone-oficina/{id}	Exclui um telefone de oficina específico
GET	http://localhost:8080/ideiasinaction_war/api/rest/telefone-oficina/all	Retorna uma lista de todos os telefones das oficinas
POST	http://localhost:8080/ideiasinaction_war/api/rest/telefone-oficina	Criação de um novo telefone de oficina
PUT	http://localhost:8080/ideiasinaction_war/api/rest/telefone-oficina/{id}	Atualiza um telefone de oficina específico
DELETE	http://localhost:8080/ideiasinaction_war/api/rest/veiculo/{id}	Exclui um veículo específico pelo ID
GET	http://localhost:8080/ideiasinaction_war/api/rest/veiculo/all	Retorna uma lista de todos os veículos
POST	http://localhost:8080/ideiasinaction_war/api/rest/veiculo	Criação de um novo veículo
PUT	http://localhost:8080/ideiasinaction_war/api/rest/veiculo/{id}	Atualiza um veículo específico pelo ID

5. Protótipos de Telas

- **Tela de Login e Cadastro:** Interface para autenticação do usuário.
- **Tela de Diagnóstico:** Interface para inserção de sintomas do veículo.
- **Tela de Orçamento:** Exibe orçamento com detalhes de serviços e peças.
- **Tela de Agendamento:** Interface para escolha de data e horário do serviço.
- **Tela de Confirmação:** Exibe resumo do agendamento e informações de pagamento.

6. Modelo do Banco de Dados

O modelo de banco de dados foi projetado para atender as necessidades de um sistema automotivo que realiza diagnósticos, gera orçamentos e permite o agendamento de serviços. Abaixo, descrevemos as principais tabelas e suas funções:

- **T_CLIENTE:** Esta tabela armazena as informações pessoais dos clientes, incluindo CPF, nome, e-mail e sexo. Cada cliente possui um identificador único gerado automaticamente.
- **T_TELEFONE_CLIENTE:** Contém os números de telefone dos clientes, que podem ser classificados como residencial, comercial ou celular. Esta tabela possui uma relação com a tabela de clientes para vincular cada número de telefone ao seu respectivo cliente.
- **T_ENDERECO_CLIENTE:** Armazena o endereço completo dos clientes, incluindo logradouro, número, CEP, bairro, cidade e estado. Cada endereço está vinculado a um cliente específico por meio de uma chave estrangeira.
- **T_VEICULO:** Registra as informações dos veículos dos clientes, como placa, marca, modelo, ano de fabricação, cor e tipo de combustível. Cada veículo está associado a um cliente e possui um identificador exclusivo.

- **T_OFICINA:** Tabela que guarda os dados das oficinas parceiras, incluindo CNPJ, nome e e-mail. Cada oficina tem um identificador único.
- **T_TELEFONE_OFICINA:** Armazena os números de telefone das oficinas, categorizados como residencial, comercial ou celular. Esta tabela está associada à tabela de oficinas para vincular cada número ao respectivo estabelecimento.
- **T_ENDERECO_OFICINA:** Contém os endereços das oficinas, com campos para logradouro, número, CEP, bairro, cidade e estado. Cada endereço é vinculado a uma oficina específica.
- **T_FUNCIONARIO:** Esta tabela armazena dados dos funcionários que trabalham nas oficinas, como CPF, nome, sexo e função. Cada funcionário está associado a uma oficina através de uma chave estrangeira.
- **T_AGENDAMENTO:** Registra os agendamentos de serviços, com informações de data e hora do agendamento, além de uma relação com o cliente e a oficina envolvidos no agendamento.
- **T_ORCAMENTO:** Armazena os orçamentos gerados para os serviços solicitados, incluindo o valor total, situação do orçamento, e vínculos com o veículo e a oficina responsáveis pelo serviço.
- **T_SERVICOS:** Tabela que registra os tipos de serviços oferecidos, com descrição do serviço, custo e tempo estimado. Cada serviço está associado a um orçamento.
- **T_FALHAS:** Contém as falhas detectadas nos veículos, com uma descrição da falha, solução proposta e uma categorização de gravidade (baixa, média ou alta). Cada falha está vinculada a um orçamento e a um veículo específico.
- **T_PECAS:** Registra as peças necessárias para execução dos serviços, incluindo marca, quantidade, valor e descrição. Cada peça está associada a um serviço e a um orçamento.
- **T_ORDEM_DE_SERVICO:** Esta tabela representa as ordens de serviço, com dados de início e fim, status, e associações com o orçamento, funcionário e veículo.
- **T_METODO_PAGAMENTO:** Tabela que armazena os métodos de pagamento utilizados, incluindo a forma de pagamento, tipo (crédito, débito, etc.) e possíveis descontos aplicados. Cada método de pagamento está vinculado a uma ordem de serviço.

Esse modelo de banco de dados é estruturado para garantir a integridade dos dados e a rastreabilidade de informações essenciais para o funcionamento do sistema, como clientes, veículos, serviços, agendamentos e ordens de serviço.

7. Diagrama de Classes

O diagrama de classes ilustra as relações entre as classes principais do sistema:

- **Agendamento**

- **Cliente**
 - **Cor**
- **Endereco**
 - **Falhas**
- **Funcionario**
 - **Oficina**
- **Orcamento**
- **OrdemDeServico**
 - **Pagamento**
 - **Pecas**
 - **Pessoa**
 - **Servicos**
 - **Telefone**
 - **Veiculo**

Cada uma dessas classes foi projetada para gerenciar diferentes aspectos do sistema, desde o armazenamento de dados dos clientes até o gerenciamento dos serviços prestados e peças utilizadas.

8. Procedimentos para Rodar a Aplicação

1. **Pré-requisitos:** Instale o Java Development Kit (JDK) 8 ou superior e uma IDE como IntelliJ ou Eclipse.
2. **Configuração do Banco de Dados:** Execute o script SQL fornecido para criar as tabelas e ajuste as configurações de conexão no código.