Estaciona360: Documentação

Introdução

Descrição Geral do Projeto

- O Estaciona360 é uma aplicação de gerenciamento de estacionamentos, projetada para facilitar a administração e reserva de vagas em múltiplos estacionamentos.
- O sistema oferece funcionalidades para cadastro e gerenciamento carros e estacionamentos, além de permitir a reserva e pagamento de vagas. A motivação por trás da criação do Estaciona360 é melhorar a eficiência na gestão de estacionamento e oferecer uma solução prática para os usuários encontrarem e pagarem por vagas disponíveis.

Objetivos

- Fornecer uma interface amigável para usuários e administradores gerenciarem estacionamentos e carros.
- Permitir o cadastro e gerenciamento de veículos e estacionamentos.
- Facilitar a reserva de vagas com cálculos automáticos de preços e processamento de pagamentos.
- Oferecer funcionalidades de perfil para que os usuários possam atualizar suas informações pessoais.

Escopo

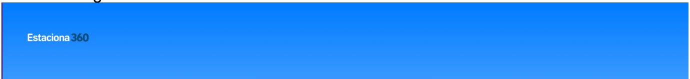
O Estaciona360 abrange:

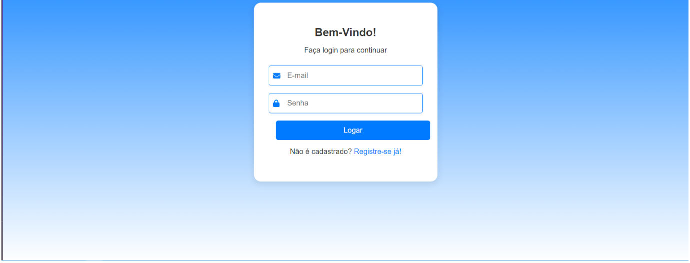
1. Página de login e cadastro de novos usuários.
2. Gerenciamento de usuários, carros e estacionamentos.
3. Reserva de vagas e cálculo de preços.
4. Alteração de dados de perfil do usuário.
5. Listagem e administração de vagas ocupadas e concluídas.
6. Visão Geral do Sistema
7. Arquitetura do Sistema
8. O Estaciona360 é uma aplicação web construída com React para a interface do usuário e ASP.NET Core para a API. O backend utiliza Entity Framework Core para o gerenciamento de banco de dados, enquanto o frontend interage com o backend através de chamadas HTTP.

Funcionalidades

Login e Cadastro:

Permite o login de usuários e o cadastro de novos usuários.


Estaciona360

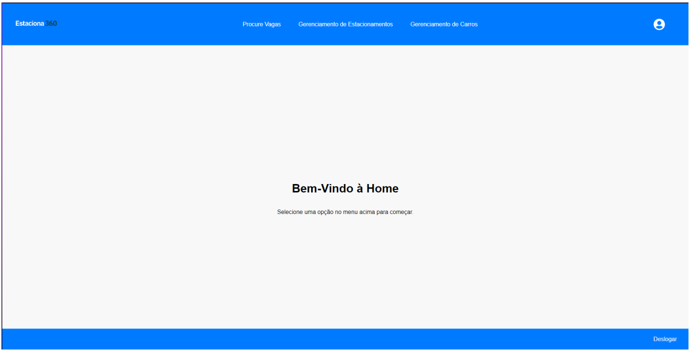

Bem-Vindo!

Faça login para continuar

E-mail
Senha
Logar
Não é cadastrado? Registre-se já!

Home:

Oferece navegação para diferentes funcionalidades como "Procure Vagas", "Gerenciamento de Estacionamentos", "Gerenciamento de Carros" e "Perfil".

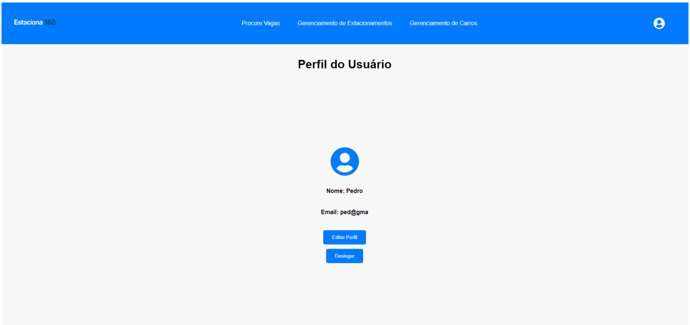

Estaciona360
Procure Vagas
Gerenciamento de Estacionamentos
Gerenciamento de Carros
Deslogar

Bem-Vindo à Home

Selecione uma opção no menu acima para começar.

Perfil:

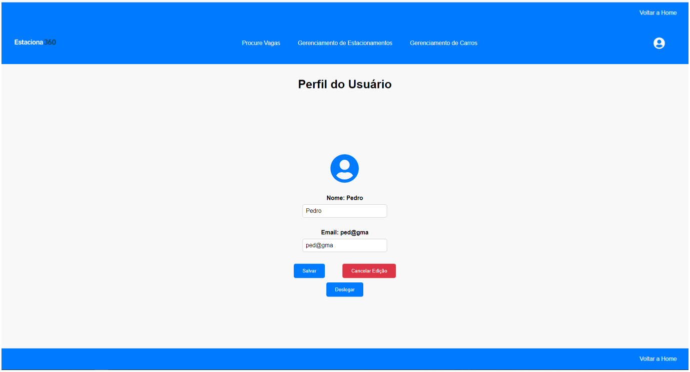
Permite a alteração de informações pessoais do usuário.


Estaciona360
Procure Vagas
Gerenciamento de Estacionamentos
Gerenciamento de Carros
Editar Perfil
Deslogar

Perfil do Usuário

Nome: Pedro

Email: ped@gma


Voltar a Home
Estaciona360
Procure Vagas
Gerenciamento de Estacionamentos
Gerenciamento de Carros
Salvar
Cancelar Edição
Deslogar
Voltar a Home

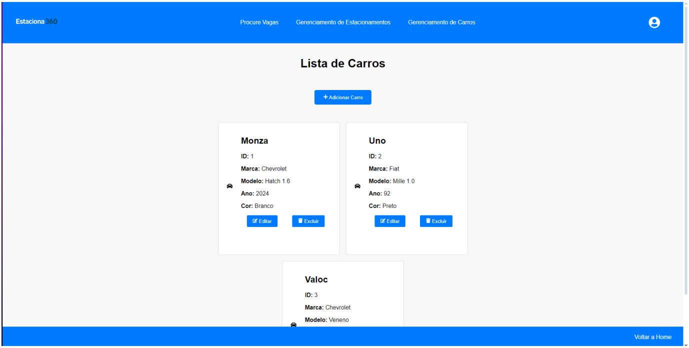
Perfil do Usuário

Nome: Pedro
Pedro

Email: ped@gma
ped@gma

Carros:

Funcionalidades para adicionar, editar e excluir carros.


Estaciona360
Procure Vagas
Gerenciamento de Estacionamentos
Gerenciamento de Carros
+ Adicionar Carro
Editar
Excluir
Editar
Excluir
Voltar a Home

Lista de Carros

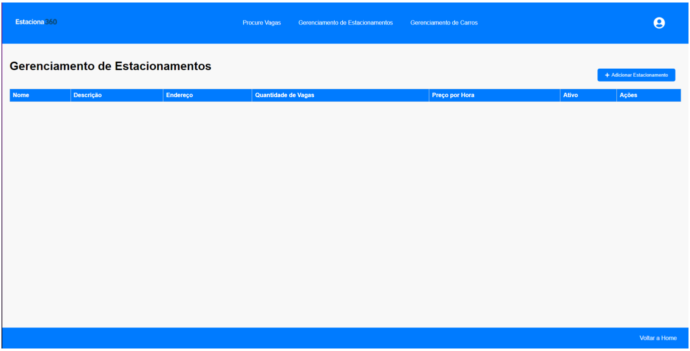
Monza
ID: 1
Marca: Chevrolet
Modelo: Hatch 1.6
Ano: 2024
Cor: Branco

Uno
ID: 2
Marca: Fiat
Modelo: Mille 1.0
Ano: 92
Cor: Preto

Valoc
ID: 3
Marca: Chevrolet
Modelo: Veneno

Estacionamentos:

Funcionalidades para adicionar, editar e excluir estacionamentos, com associação a endereços.

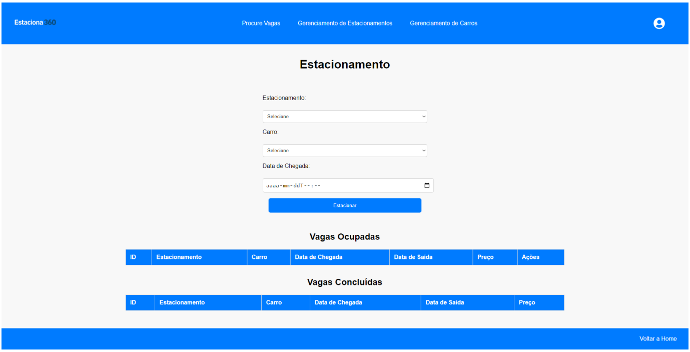

Estaciona360
Procure Vagas
Gerenciamento de Estacionamentos
Gerenciamento de Carros
+ Adicionar Estacionamento
Voltar a Home

Gerenciamento de Estacionamentos

Nome	Descrição	Endereço	Quantidade de Vagas	Preço por Hora	Ativo	Ações

Procure Vagas:

Permite a reserva de vagas com cálculo de preço baseado na data de chegada e saída. Configuração do Ambiente.


Estaciona360
Procure Vagas
Gerenciamento de Estacionamentos
Gerenciamento de Carros
Estacionar
Voltar a Home

Estacionamento

Estacionamento: Selecione
Carro: Selecione
Data de Chegada: aaaa-mm-ddT--:--

Vagas Ocupadas

ID	Estacionamento	Carro	Data de Chegada	Data de Saída	Preço	Ações

Vagas Concluídas

ID	Estacionamento	Carro	Data de Chegada	Data de Saída	Preço

Interface do Usuário

Descrição das Funcionalidades da Interface

Login: Tela para autenticação de usuários e criação de novos usuários.

Home: Navegação para diferentes funcionalidades da aplicação.

Perfil: Tela para visualização e edição das informações pessoais do usuário.

Gerenciamento de Carros: Tela para adicionar, editar e excluir carros.

Gerenciamento de Estacionamentos: Tela para adicionar, editar e excluir estacionamentos.

Procure Vagas: Tela para reserva de vagas com cálculo de preço.
Banco de Dados

Requisitos de Software e Hardware

Backend: .NET 6.0, SQL Server
Frontend: React, Node.js
Hardware: Requisitos mínimos de hardware dependem do ambiente de execução.

Instruções de Instalação

Backend:

Instale o .NET 6.0 SDK.
Clone o repositório do projeto.
Restaure os pacotes e execute a aplicação com dotnet run.

Frontend:

Instale o Node.js.
Navegue até o diretório do frontend.
Instale as dependências com npm install.
Inicie o servidor de desenvolvimento com npm start.
Configuração do Ambiente de Desenvolvimento
Configure as variáveis de ambiente para o banco de dados e outras configurações necessárias.
Certifique-se de que o backend e o frontend estejam apontando para as URLs corretas.

Dependências

Backend:

Microsoft.AspNetCore.Mvc.NewtonsoftJson
Microsoft.EntityFrameworkCore.SqlServer
Swashbuckle.AspNetCore
Microsoft.EntityFrameworkCore.Tools

Frontend:

react
react-dom
axios
scss

Desenvolvimento

Estrutura do Projeto

Frontend:

src/components: Componentes reutilizáveis como TopBar e BottomBar.
src/pages: Páginas principais como Login, Home e Gerenciamento.
src/services: Serviços para interagir com a API.
src/styles: Arquivos SCSS para estilização.

Backend:

Controllers: Controladores para gerenciar as requisições da API.
Models: Modelos das entidades do banco de dados.

Services: Lógica de negócios e interações com o banco de dados.
Repositories: Repositórios para o acesso aos dados.
Descrição das Camadas
Apresentação: Responsável pela interface do usuário, implementada com React.
Domínio: Contém a lógica de negócios e as regras de validação.
Repositório: Gerencia o acesso e a manipulação dos dados no banco de dados.
Serviços: Fornece métodos para interagir com o banco de dados e processar a lógica de negócios.

API

Documentação da API

Endpoints:

GET /api/Usuario: Lista todos os usuários.

POST /api/Usuario/login: Realiza login.

POST /api/Usuario: Adiciona um novo usuário.

PUT /api/Usuario/{id}: Atualiza um usuário.

DELETE /api/Usuario/{id}: Remove um usuário.

GET /api/Estacionamento: Lista todos os estacionamentos.

POST /api/Estacionamento: Adiciona um novo estacionamento.

PUT /api/Estacionamento/{id}: Atualiza um estacionamento.

DELETE /api/Estacionamento/{id}: Remove um estacionamento.

GET /api/Carro: Lista todos os carros.

POST /api/Carro: Adiciona um novo carro.

PUT /api/Carro/{id}: Atualiza um carro.

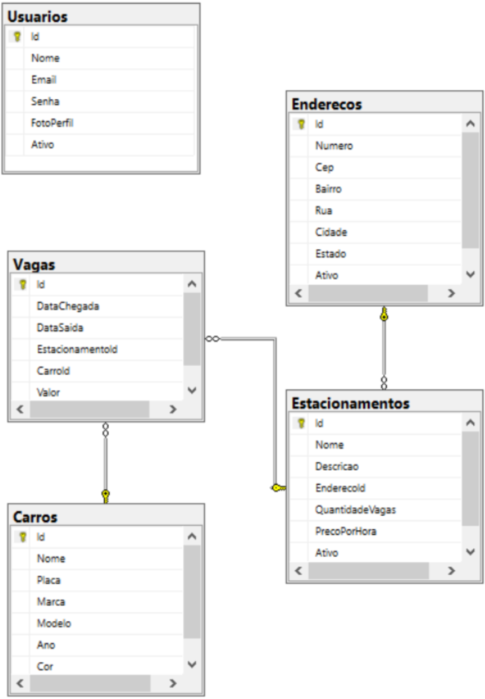
DELETE /api/Carro/{id}: Remove um carro.

POST /api/Vaga: Adiciona uma nova vaga.

PUT /api/Vaga/{id}: Atualiza uma vaga.

GET /api/Vaga: Lista todas as vagas.

Diagrama de Entidade-Relacionamento (ERD)


Usuarios
Id
Nome
Email
Senha
FotoPerfil
Ativo
Enderecos
Id
Numero
Cep
Bairro
Rua
Cidade
Estado
Ativo
Vagas
Id
DataChegada
DataSaida
EstacionamentoId
CarroId
Valor
Estacionamentos
Id
Nome
Descricao
EnderecoId
QuantidadeVagas
PrecoPorHora
Ativo
Carros
Id
Nome
Placa
Marca
Modelo
Ano
Cor

Usuário: id, nome, email, senha, FotoPerfil, ativo

Carro: id, nome, placa.marca.modelo,ano, cor

Estacionamento: id, nome, enderecoId, quantidadeVagas, precoPorHora, ativo

Endereço: id, CEP, bairro, número, rua, cidade, estado, ativo

Vaga: id, EstacionamentoId, CarroId, dataChegada, dataSaida, valor

Considerações Finais

Desenvolvimento ágil e integração contínua facilitaram a implementação e testes das funcionalidades.

A importância de uma documentação clara e atualizada para facilitar a manutenção e o onboarding de novos desenvolvedores.

Melhores Práticas

Utilizar práticas recomendadas para segurança no armazenamento de senhas e autenticação de usuários.

Manter o código modular e reutilizável para facilitar a manutenção e expansão do sistema.
Próximos Passos

Implementar novas funcionalidades baseadas no feedback dos usuários.

Melhorar a interface do usuário para uma experiência mais intuitiva.

Adicionar mais opções de personalização e relatórios.

Anexos

Referências e Recursos Adicionais

Documentação do React
Documentação do ASP.NET Core
Documentação do Entity Framework Core