



UNIVERSIDADE DO OESTE DE SANTA CATARINA
Campus de São Miguel do Oeste
Curso de Ciência da Computação



FERNANDO CAMILO SCHNEIDER
JEAN CARLO TORAL
JOÃO ULISSES PORTO ALEGRE CIRIACO TEIXEIRA

PARK'ALOT: Sistema para gerenciamento de estacionamentos pagos.

São Miguel do Oeste
2024

FERNANDO CAMILO SCHNEIDER
JEAN CARLO TORAL
JOÃO ULISSES PORTO ALEGRE CIRIACO TEIXEIRA

PARK'ALOT: Sistema para gerenciamento de estacionamento pagos.

Trabalho integrado dos componentes curriculares
Banco de Dados II, Engenharia de Software I e
Programação II.

Orientadores: Prof. Roberson Junior Fernandes Alves, Profa. Otilia Donato Barbosa.

São Miguel do Oeste
2024

LISTA DE ILUSTRAÇÕES

Figura 1: Diagrama Relacional.....	6
Figura 2: Diagrama de Casos de Uso.....	7
Figura 3: Diagrama de Classes.....	8
Figura 4: Diagrama de Atividades.....	9
Figura 5: Diagrama de Estados.....	10
Figura 6: Diagrama de Sequência.....	11
Figura 7: Organização de arquivos.....	12
Figura 8: Arquivo de propriedades.....	13
Figura 9: Página de cadastro.....	14
Figura 10: Página de login.....	15
Figura 11: Página para reservar vagas.....	15
Figura 12: Modal para adicionar novo carro.....	16

SUMÁRIO

1 INTRODUÇÃO.....	5
2 DESENVOLVIMENTO.....	5
2.1 MODELAGEM.....	6
2.2 BACKEND.....	11
2.3 FRONTEND.....	13
3 CONCLUSÃO.....	14
4 REFERÊNCIAS.....	15

1 INTRODUÇÃO

O tráfego de veículos nas cidades brasileiras é um grande desafio para o cenário urbano devido ao aumento constante da frota veicular, que já ultrapassa 119 milhões de veículos em dezembro de 2023, segundo estudos do Instituto Brasileiro de Planejamento e Tributação (IBPT).

Com base nos dados oficiais da ABRAPARK, o setor de estacionamentos representa um segmento significativo da economia brasileira, contando com aproximadamente 11 mil estabelecimentos comerciais que totalizam mais de 3 milhões de vagas disponíveis em todo país. O setor é responsável pela geração de mais de 300 mil empregos diretos, demonstrando sua relevância não apenas para a mobilidade urbana, mas também para o mercado de trabalho nacional. Este cenário evidencia a necessidade de sistemas eficientes de gerenciamento que possam otimizar a operação destes estabelecimentos, melhorando tanto a experiência dos usuários quanto a gestão dos recursos disponíveis.

Diante deste contexto, este trabalho apresenta o desenvolvimento de um sistema de gerenciamento de estacionamentos, visando otimizar a utilização do espaço e melhorar a experiência dos usuários através de tecnologias modernas que atendam às demandas atuais do setor.

2 DESENVOLVIMENTO

Para facilitar o desenvolvimento do projeto, foi utilizado o banco de dados relacional PostgreSQL, já desenvolvido anteriormente, juntamente com a ferramenta Dbeaver. Na parte de programação foi utilizado a linguagem Java juntamente com a ferramenta Visual Studio Code como ambiente de desenvolvimento para fazer a parte do back-end com Spring Boot, com o intuito de fazer a conexão com o banco de dados. No desenvolvimento dos modelos foram utilizados os requisitos já levantados anteriormente junto com a ferramenta Visual paradigm. Para fazer o versionamento do projeto foi utilizado o GitHub. No planejamento das atividades e das divisões de tarefas foi utilizado o Google Drive.

2.1 MODELAGEM

Depois de levantar todos os requisitos necessários para o desenvolvimento do sistema foi iniciada a construção do novo diagrama relacional, seguindo as atualizações feitas no banco de dados como demonstra a figura abaixo.

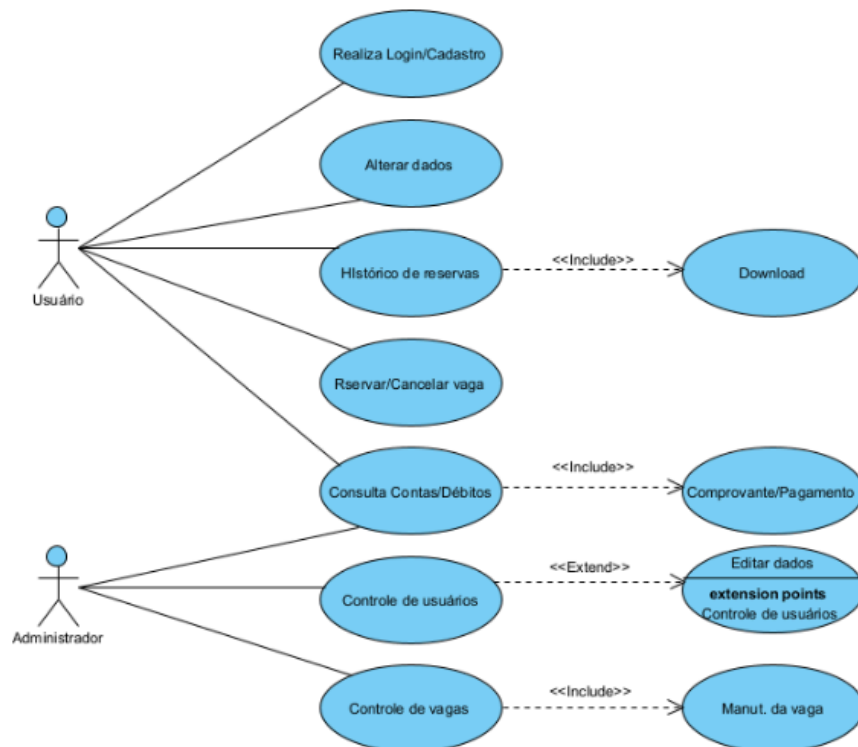
Figura 1: Diagrama Relacional



Fonte: Os autores(2024)

Em seguida foi feito o diagrama de casos de uso, cuja função é descrever como os usuários interagem com o sistema, além disso facilitam a organização dos requisitos de um sistema.

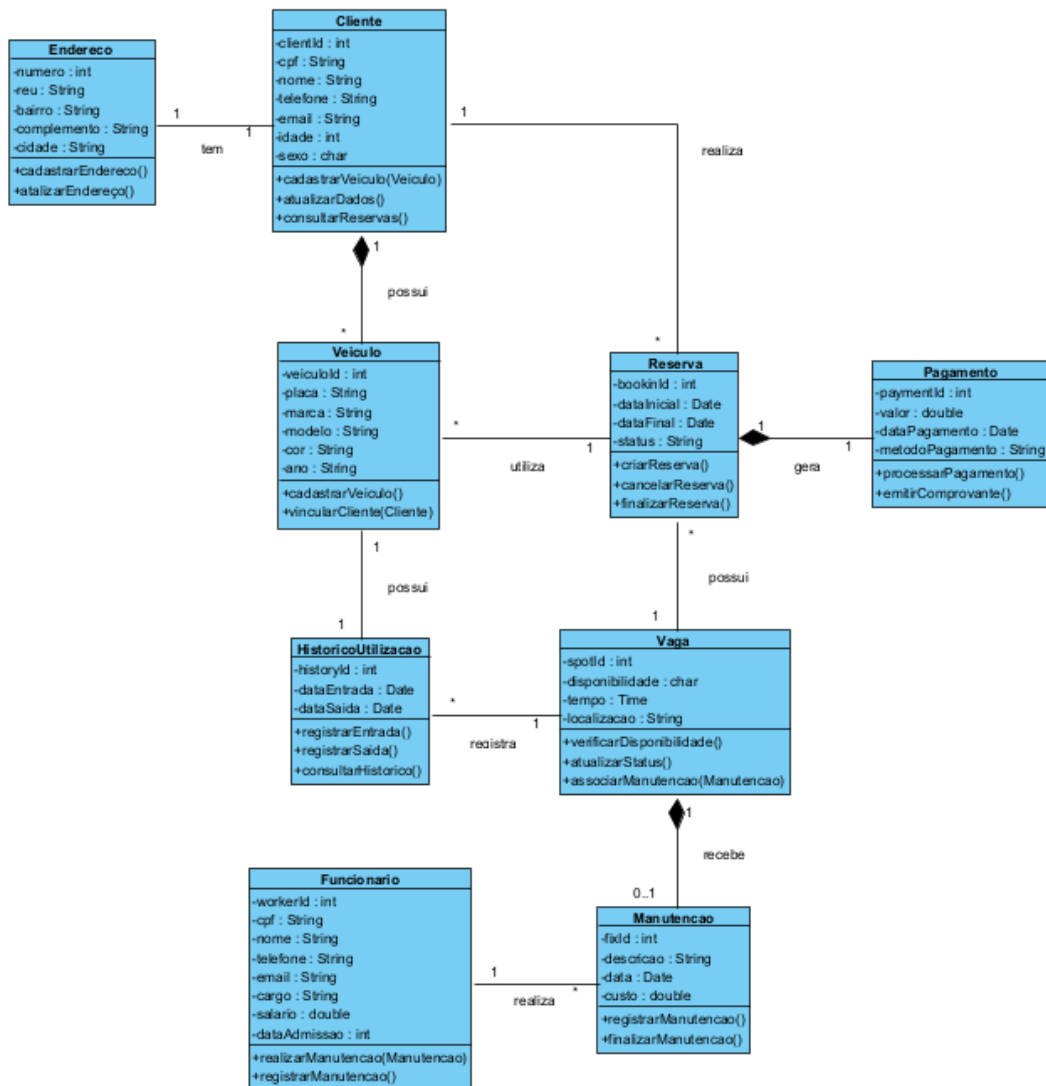
Figura 2: Diagrama de Casos de Uso



Fonte: Os autores(2024)

Para ter uma visão da estrutura do sistema de forma estática, e visualizar as interfaces e seus relacionamentos foi utilizado um diagrama de classes, exposto abaixo, modelado utilizando o Visual Paradigm, onde tem a possibilidade de verificar as ligações das tabelas e demais informações importantes.

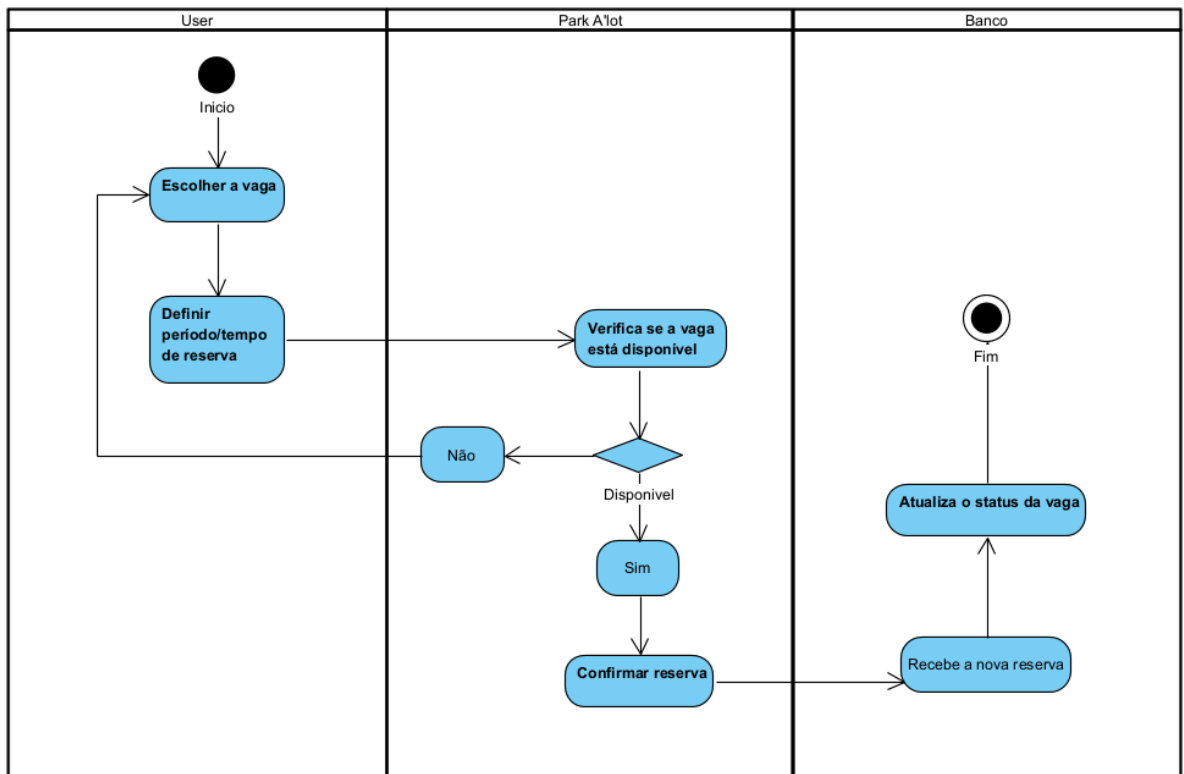
Figura 3: Diagrama de Classes



Fonte: Os autores(2024)

Para descrever os passos a serem percorridos para a conclusão de um método ou algoritmo específico, neste caso a reserva de uma vaga, foi feito um diagrama de atividades conforme demonstra a figura abaixo.

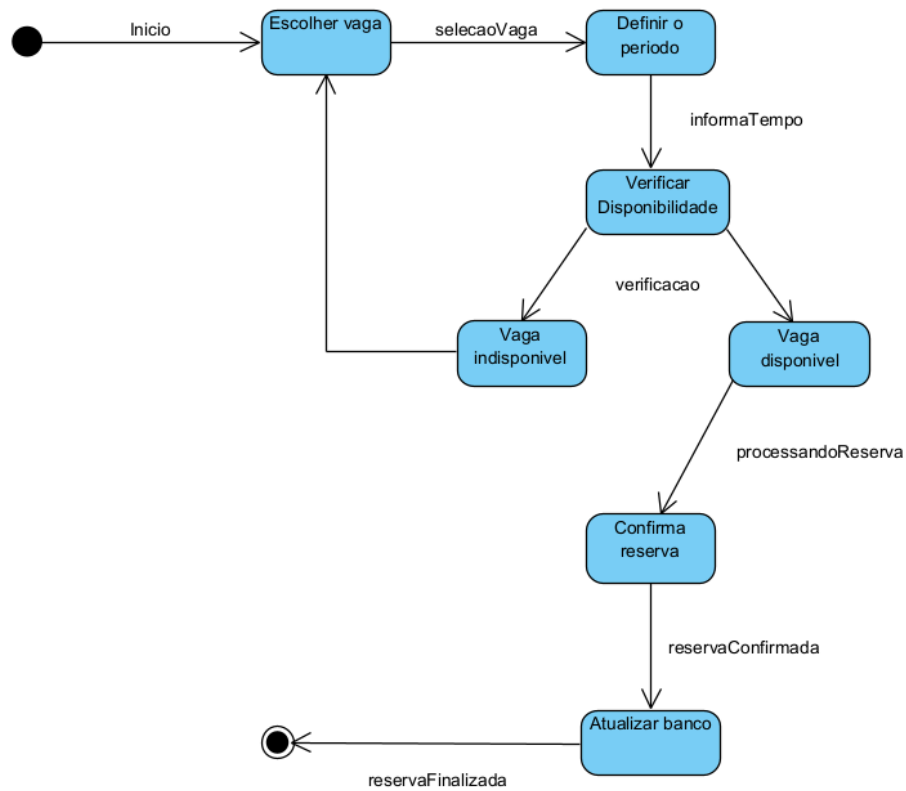
Figura 4: Diagrama de Atividades



Fonte: Os autores(2024)

Em seguida foi feito o Diagrama de Estados, que é utilizado para acompanhar os estados por que passa uma instância de uma classe, ou representar os estados de um Caso de Uso ou mesmo de um subsistema ou sistema completo.

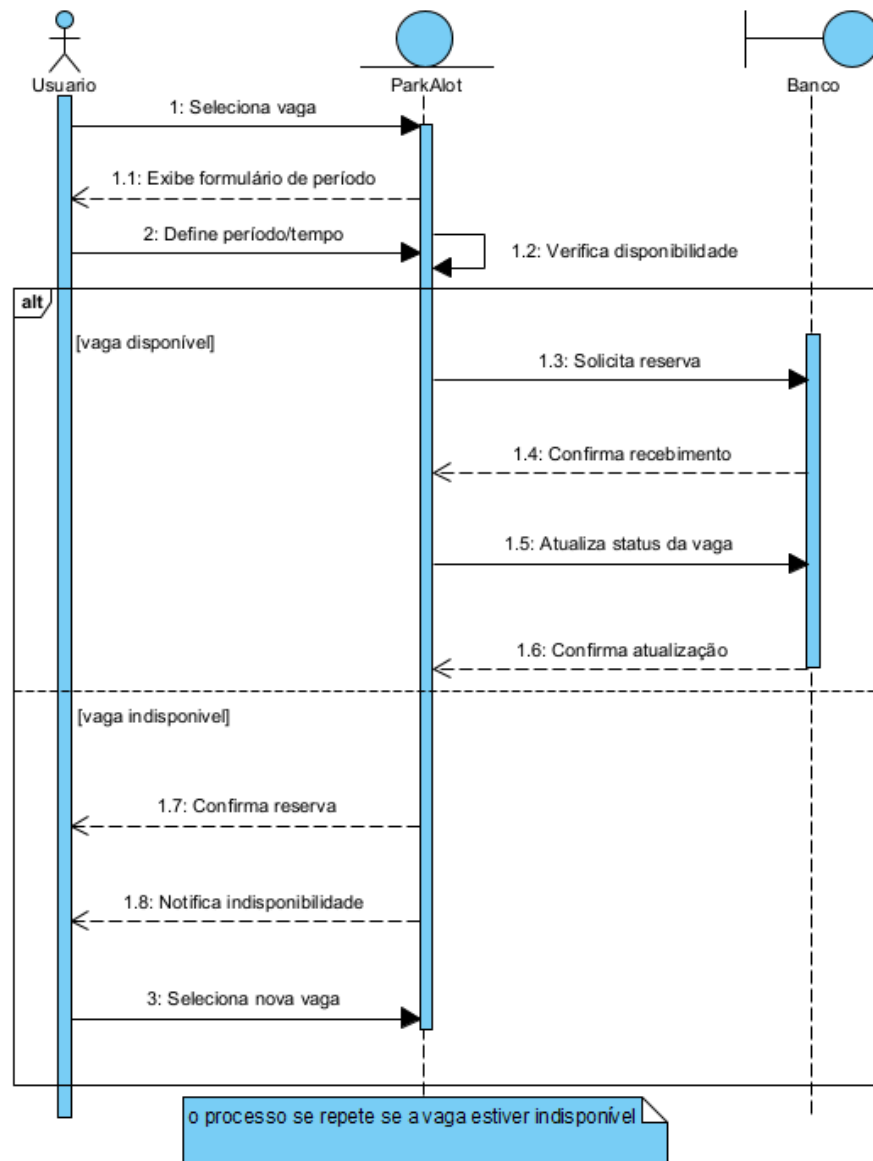
Figura 5: Diagrama de Estados



Fonte: Os autores(2024)

Para concluir a parte da modelagem foi feito o Diagrama de Sequência que enfatiza a ordenação das mensagens trocadas entre os objetos.

Figura 6: Diagrama de Sequência



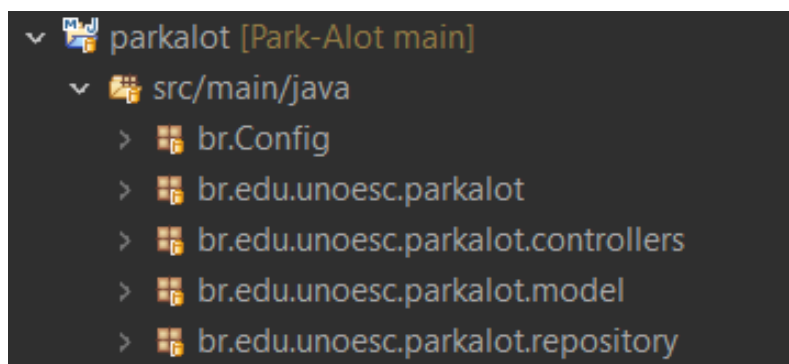
Fonte: Os autores(2024)

2.2 BACKEND

Para o início do desenvolvimento do backend, foram utilizadas as ferramentas Java e Spring Boot. A estrutura da aplicação foi projetada de forma simplificada, priorizando maior legibilidade e facilidade para possíveis correções futuras. A

organização do projeto segue uma separação clara de arquivos em pastas específicas, como ilustrado na figura 7.

Figura 7: Organização de arquivos



Fonte: Os autores(2024)

No pacote **br.edu.unoesc.parkalot**, encontra-se a classe principal do projeto. Já na pasta Controllers, estão todas as classes responsáveis pelo controle do sistema, incluindo as de Cliente, Veículo, Vaga, Reserva e uma específica para o relacionamento entre Veículo e Vaga. Na pasta Models, encontram-se os scripts para a criação da base de dados. Apesar de o projeto incluir todos os campos necessários para a execução completa, a versão final utilizou apenas as tabelas de Cliente, Veículo, Vaga e Reserva. A configuração da conexão com o banco de dados foi realizada no arquivo de propriedades **application.properties**, conforme apresentado na figura 8.

Figura 8: Arquivo de propriedades

```
1 spring.application.name=parkalot
2
3 server.port=8080
4 server.address=0.0.0.0
5 spring.h2.console.enabled=true
6 spring.h2.console.path=/h2-console
7 spring.datasource.url=jdbc:postgresql://localhost:5433/pparkalot
8 spring.datasource.username=postgres
9 spring.datasource.password=postgres
10 spring.jpa.properties.hibernate.jdbc.lob.non_contextual_creation=true
11
12 spring.jpa.properties.hibernate.id.new_generator_mappings=true
13 spring.jpa.hibernate.ddl-auto=update
14
```

Fonte: Os autores(2024)

Foram implementadas diversas requisições para garantir o funcionamento da aplicação:

- **Tabela Cliente:** além das operações básicas de CRUD (Create, Read, Update, Delete), foram desenvolvidas requisições para verificação de CPF e verificação de e-mail — utilizadas na área de cadastro —, além da requisição de login, que valida os dados inseridos pelo usuário e verifica sua existência na base.
- **Tabela Veículo:** foram realizadas apenas as requisições básicas de CRUD.
- **Tabela Vaga:** foram criadas requisições de listagem, incluindo uma para exibir todas as vagas e outra específica para vagas disponíveis.
- **Tabela Reserva:** além das operações CRUD, implementou-se uma requisição para relacionar veículo e vaga, que altera automaticamente o status da vaga para "Reservada". Também foi criada uma requisição para listar todas as reservas associadas ao cliente logado no sistema.

2.3 FRONTEND

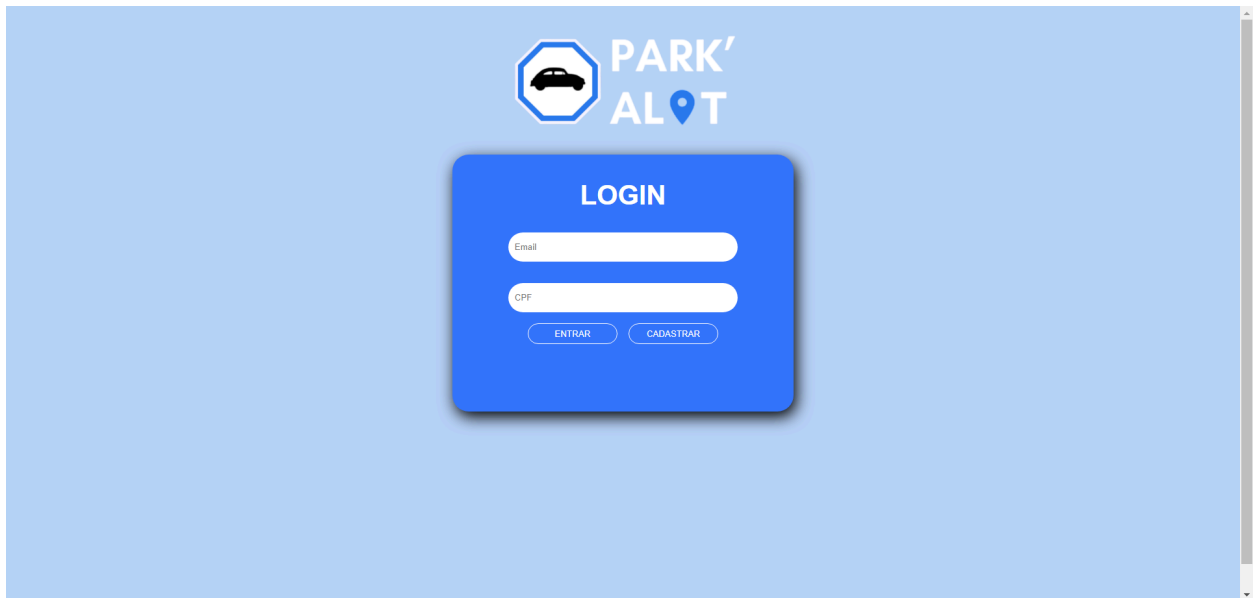
Para testar todas essas requisições, utilizou-se a ferramenta Insomnia. A parte do frontend foi desenvolvida utilizando HTML, CSS e JavaScript, como demonstrado nas próximas imagens.

Figura 9: Página de cadastro

A imagem mostra a interface de usuário para o cadastro no sistema PARK'ALOT. No topo, há o logotipo do sistema, que consiste em um ícone de um carro dentro de um octógono azul, seguido pelo texto "PARK'ALOT" em uma fonte moderna, onde o "A" é substituído por um ícone de localização. Abaixo do logotipo, o formulário "Cadastre-se" é apresentado em um cartão azul com cantos arredondados. O formulário contém os seguintes campos: "Informe o nome", "Informe o CPF", "Informe o e-mail", "Informe o telefone", "Selecione o sexo" (menu suspenso) e "dd/mm/aaaa" (campo de data com ícone de calendário). Abaixo dos campos, há um botão "Cadastrar" e um link "Já é cadastrado? Entre".

Fonte: Os autores(2024)

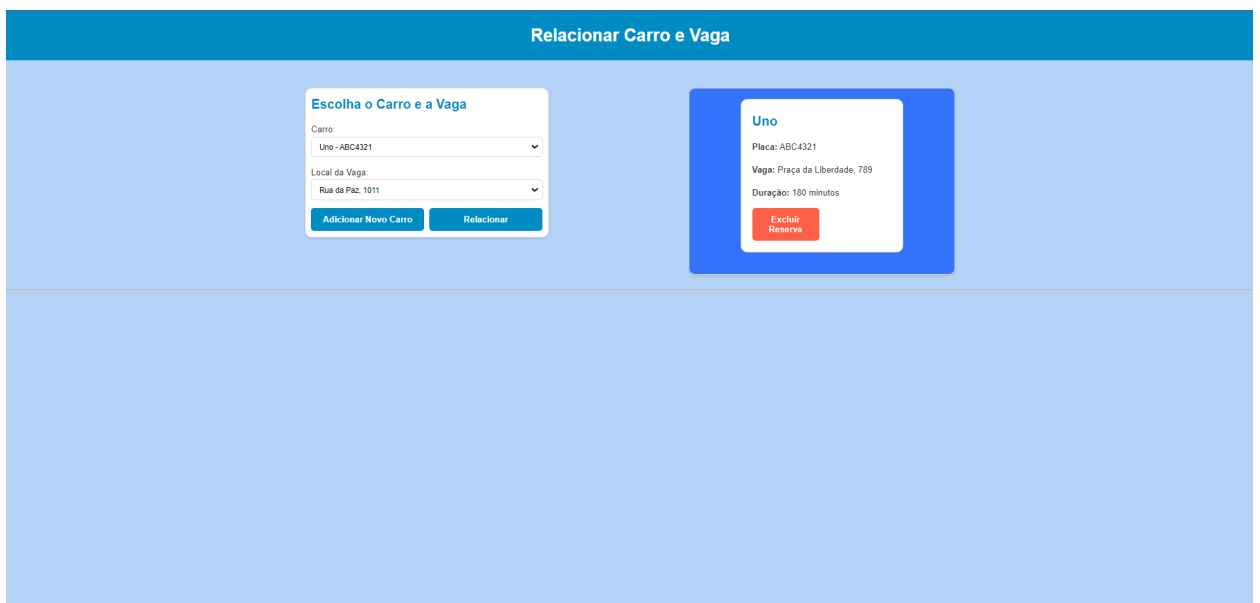
Figura 10: Página de login



The login page features a light blue background. At the top center is the PARK'ALOT logo, which includes a car icon inside a blue octagon and the text 'PARK'ALOT' in blue. Below the logo is a blue rounded rectangle with the word 'LOGIN' in white. Inside this rectangle are two white input fields labeled 'Email' and 'CPF'. At the bottom of the rectangle are two white buttons labeled 'ENTRAR' and 'CADASTRAR'.

Fonte: Os autores(2024)

Figura 11: Página para reservar vagas



The page is titled 'Relacionar Carro e Vaga' in a blue header. It contains two main sections. The left section, titled 'Escolha o Carro e a Vaga', has two dropdown menus: 'Carro:' with 'Uno - ABC4321' and 'Local da Vaga:' with 'Rua da Paz, 1011'. Below these are two buttons: 'Adicionar Novo Carro' and 'Relacionar'. The right section, titled 'Uno', displays details for a specific car: 'Placa: ABC4321', 'Vaga: Praça da Liberdade, 789', and 'Duração: 180 minutos'. At the bottom of this section is a red button labeled 'Excluir Reserva'.

Fonte: Os autores(2024)

Figura 12: Modal para adicionar novo carro

The image shows a web application interface with a modal for adding a new car. The modal is titled "Adicionar Novo Carro" and contains the following fields and buttons:

- Modelo:
- Marca:
- Placa:
- Cor:
- Ano de Fabricação:
- Buttons: "Adicionar Carro" and "Fechar"

In the background, there is a form titled "Escolha o Carro e a Vaga" with the following fields and buttons:

- Carro:
- Local da Vaga:
- Buttons: "Adicionar Novo Carro" and "Relacionar"

There is also a card for a car named "Uno" with the following information:

- Placa: ABC4321
- Praça da Liberdade, 789
- Tempo: 180 minutos
- Buttons: "Excluir" and "Reserva"

Fonte: Os autores(2024)

3 CONCLUSÃO

O desenvolvimento deste sistema de gerenciamento de estacionamentos demonstrou o impacto positivo que as tecnologias podem trazer ao setor, além disso foi pensado para atender ambientes de alta demanda e fluxo intenso, como aeroportos, shoppings, estádios, e centros comerciais. Pelo fato destes locais apresentarem desafios específicos devido aos picos de utilização em determinados horários, causado pelo fluxo de veículos e pelas necessidades distintas que cada ambiente possui, é essencial implementar soluções ágeis que facilitem o acesso dos usuários e reduzam o tempo de espera, garantindo assim uma experiência gratificante.

A estrutura do projeto foi realizada em camadas bem definidas, separando claramente o backend e o frontend, permitindo um desenvolvimento organizado e com fácil manutenção. Utilizando Spring Boot com Java no backend, junto com o banco de dados PostgreSQL, proporcionou uma base sólida para o sistema, enquanto a interface desenvolvida em HTML, CSS e JavaScript oferece uma experiência intuitiva para os usuários.

Na modelagem foi realizada a criação de diversos diagramas , para garantir que o sistema fosse implementado de uma forma estruturada e alinhada com os requisitos coletados. As funcionalidades que foram implementadas no sistema atendem às necessidades básicas de um estacionamento eficiente e moderno.

Este sistema contribui para a melhoria da mobilidade urbana e do gerenciamento de estacionamentos, com potencial de ampliação para atender a demandas futuras. Por ter uma estrutura escalável ele pode ser adaptado para diversos cenários específicos, para que não contribua apenas para melhorar a mobilidade e a experiência dos usuários, mas também para aumentar a eficiência operacional dos estacionamentos.

Acredita-se que a expansão futura do projeto, com a integração de novas tecnologias e funcionalidades específicas, fortalecerá ainda mais sua aplicabilidade em diferentes setores e contextos.

4 REFERÊNCIAS

ROVER, Ardinete; MELLO, Regina Oneda. **Normas da ABNT: Orientações para a produção científica**. 2. ed. Joaçaba: Editora Unoesc, 2024.

O TEMPO. Frota brasileira fecha 2023 em 119.227.657, um veículo para cada 1,7 habitante. **O Tempo**, 2024. Disponível em: <https://www.otempo.com.br/economia/frota-brasileira-fecha-2023-em-119-227-657-um-veiculo-para-cada-1-7-habitante-1.3346324>. Acesso em: 20 de nov. de 2024.

ALVES, Roberson J. F. **Modelagem UML: Diagrama de Classes**. São Miguel do Oeste: Unoesc, 2024. 26 slides. Apresentação de slides.

ALVES, Roberson J. F. **Análise e Projeto OO com UML: Introdução e Casos de Uso**. São Miguel do Oeste: Unoesc, 2024. 68 slides. Apresentação de slides.

ALVES, Roberson J. F. **Modelagem UML: Diagrama de Atividades e de Estados**. São Miguel do Oeste: Unoesc, 2024. 36 slides. Apresentação de slides.

ALVES, Roberson J. F. **Modelagem UML: Diagrama de Sequência**. São Miguel do Oeste: Unoesc, 2024. 27 slides. Apresentação de slides.