
GoExcursões
Desenvolvimento de Aplicação web para
Gerenciamento e Divulgação de excursões
em React

Douglas Rodrigues Oliveira

PROJETO DE CONCLUSÃO DE CURSO

Data de Depósito: 20/11/2023

Assinatura: _____

GoExcursões

Desenvolvimento de Aplicação web para Gerenciamento e Divulgação de excursões em React

Douglas Rodrigues Oliveira

Prof^ª. Cristina Maria Valadares de Lima

Monografia apresentada ao Instituto Superior de Educação do UNIFOR/MG, como requisito parcial para obtenção do título de bacharel em Ciência da Computação, sob a orientação da Prof^ª. Cristina Maria Valadares de Lima

Unifor-MG - Formiga

11/2023

Eu dedico este trabalho a todas as pessoas que sempre me apoiaram e confiaram na minha capacidade

Agradecimentos

Agradeço primeiramente a Deus pela sabedoria e por direcionar os meus passos até o fim desta jornada. Agradeço a Michelly Amaral minha noiva, por ser grande incentivadora e motivadora durante o processo. Agradeço a minha avó Maria Aparecida por ser o meu suporte e estar sempre presente. Agradeço a minha mãe Ana Paula, por ser pilar de sustentação essencial para mim em minha formação pessoal e acadêmica. Agradeço também ao meu patrão Edil Lopes, pelo grande apoio e por viabilizar o meu crescimento profissional. Agradeço aos meus familiares e amigos pelo carinho e por se fazerem presentes ao longo de toda a trajetória.

"O progresso é impossível sem mudança, e aqueles que não conseguem mudar as suas mentes não conseguem mudar nada."

George Bernard Shaw

Sumário

Lista de Figuras	ii
1 Introdução	1
1.1 Considerações Iniciais	1
1.2 Objetivos do Trabalho	3
1.2.1 Objetivo Geral	3
1.2.2 Objetivos Específicos	3
1.3 Estrutura da Monografia	3
2 Referencial Teórico	5
2.1 Considerações Iniciais	5
2.2 Excursões e Viagens	5
2.3 Tecnologias Utilizadas no Desenvolvimento da Aplicação Web	6
2.4 Metodologia Ágil - Scrum	6
2.5 Banco de dados	6
2.5.1 SQL	7
2.5.2 MySQL	7
2.6 Backend	7
2.7 Java	8
2.8 JPA	9
2.9 Hibernate	9
2.10 Framework Spring	10
2.10.1 Spring Boot	10
2.11 Web Service	11
2.12 Web Service Restfull	12
2.13 Frontend	12
2.14 HTML	13
2.15 CSS	13

2.16	React	13
2.17	JavaScript	14
2.18	TypeScript	15
2.19	Considerações Finais	16
3	Trabalhos relacionados	17
3.1	Considerações Iniciais	17
3.2	TripAdvisor	17
3.2.1	Importância para o Estudo	19
3.3	Hurb	19
3.3.1	Importância para o Estudo	20
3.3.2	Considerações Finais	20
4	Metodologia e Desenvolvimento	23
4.1	Considerações Iniciais	23
4.2	Metodologia escolhida	23
4.3	Levantamento de Requisitos	24
4.4	Planejamento	26
4.5	Modelagem	27
4.5.1	Caso de Uso	27
4.5.2	Diagrama Conceitual	28
4.5.3	Diagrama Entidade Relacionamento	28
4.5.4	Diagrama de Classe	30
4.6	Backend Spring Boot	32
4.6.1	Trabalhando com Imagens	34
4.7	Conclusão do Backend	37
4.8	Frontend React	38
4.8.1	Dashboard Usuário	44
4.8.2	Dashboard Administrativo	44
4.9	Considerações finais	46
5	Testes e Resultados	49
6	Conclusões	51
6.1	Considerações Finais	51
6.2	Contribuições	52
6.3	Dificuldades Encontradas	53
6.4	Trabalhos Futuros	53

Lista de Figuras

1.1	Poluição visual com poste cheio de propagandas afixadas.(RIGON, 2013)	1
1.2	O crescimento do uso da internet nos domicílios e em cada região do Brasil. (NERY; BRITTO, 2022)	2
2.1	Exemplo de sincronização de aplicações web com banco de dados através do backend.(SOUTO, 2023)	9
2.2	Arquitetura simplificada do Hibernate.(LECHETA, 2015)	10
2.3	Componente em React parte do código da aplicação GoExcursões.	14
3.1	Exemplo de avaliações de usuário no TripAdvisor.	18
3.2	Exemplo de ranking e classificação de um estabelecimento no TripAdvisor.	18
3.3	Variedade de serviços oferecidos.	20
4.1	Esboço de Metodologia ágil Scrum.(RIBAS,)	24
4.2	Exemplo de divulgação de excursão com cartaz. Foto tirada em outubro 2023 pelo próprio autor.	24
4.3	Exemplo de divulgação de excursão em rede social.	25
4.4	Caso de uso cliente. Fonte: Próprio Autor.	28
4.5	Caso de uso organizador/agenciador. Fonte: Próprio Autor.	29
4.6	Diagrama Conceitual. Fonte: Próprio Autor.	29
4.7	Diagrama Entidade Relacionamento. Fonte: Próprio Autor.	30
4.8	Diagrama de Classe. Fonte: Próprio Autor.	32
4.9	Exemplo de Entidade no backend. Fonte: Próprio Autor.	33
4.10	Exemplo de Records no backend. Fonte: Próprio Autor.	33
4.11	Exemplo de Repository estendendo JpaRepository no backend. Fonte: Próprio Autor.	34
4.12	Exemplo de Repository com @Query no backend. Fonte: Próprio Autor.	35
4.13	Exemplo de Service no backend. Fonte: Próprio Autor.	35
4.14	Exemplo de Controller no backend. Fonte: Próprio Autor.	36

4.15	Métodos que controlarão as requisições de upload e listagem de imagens. Fonte: Próprio Autor.	36
4.16	Métodos que cuidarão da lógica de armazenamento e listagem de imagens. Fonte: Próprio Autor.	37
4.17	Página inicial da aplicação. Fonte: Próprio Autor.	39
4.18	Página com detalhes da excursão. Fonte: Próprio Autor.	40
4.19	Página de login. Fonte: Próprio Autor.	41
4.20	Página de cadastro de login. Fonte: Próprio Autor.	42
4.21	Página de cadastro dados pessoais. Fonte: Próprio Autor.	42
4.22	Etapa 1 do processo de reserva. Fonte: Próprio Autor.	43
4.23	Etapa 2 do processo de reserva. Fonte: Próprio Autor.	43
4.24	Etapa 3 do processo de reserva. Fonte: Próprio Autor.	44
4.25	Dados cadastrais do usuario. Fonte: Próprio Autor.	45
4.26	Listagem de reservas do usuario. Fonte: Próprio Autor.	45
4.27	Detalhes reserva do usuario. Fonte: Próprio Autor.	46
4.28	Página de cadastro e edição de excursões. Fonte: Próprio Autor.	46
4.29	Listagem de excursões cadastradas. Fonte: Próprio Autor.	47
4.30	Listagem de reservas da excursão selecionada. Fonte: Próprio Autor.	47
4.31	Detalhes reserva. Fonte: Próprio Autor.	48

Lista de Siglas

API - *Interface de Programação de Aplicações*

CRUD - *Create, Read, Update, Delete (criar, ler, atualizar, excluir)*

CSS - *Cascading Style Sheets*

HTML - *HyperText Markup Language*

HTTP - *Hypertext Transfer Protocol*

HTTPS - *Hypertext Transfer Protocol Secure*

JPA - *Java Persistence API*

JWT- *JSON Web Token*

UAI - *User Interface (Interface do Usuário)*

URL - *Uniform Resource Locator*

XML - *eXtensible Markup Language*

Resumo

Oliveira, D. R. *Aplicação web para Gerenciamento e Divulgação de excursões em React GoExcursões*. Monografia (Graduacao) — Centro universitario de Formiga – Unifor-mg – Formiga, 2023.

Com os efeitos causados pela poluição visual das propagandas fixadas em postes e muros e a revolução positiva proporcionada pelas aplicações web na busca de viagens. Reconhecendo esses impactos adversos das tradicionais estratégias de divulgação no ambiente urbano, este estudo propõe uma solução inovadora: o desenvolvimento de uma aplicação web dedicada ao gerenciamento e divulgação de excursões para agenciadores de viagens. A pesquisa examina a poluição visual como um problema social e estético, destacando os desafios enfrentados pela sociedade contemporânea na gestão desses impactos. Paralelamente, explora os benefícios crescentes das aplicações web no setor de viagens, destacando sua influência na simplificação do processo de planejamento e reserva para os viajantes. Este projeto concentra-se no desenvolvimento e implementação de uma aplicação web específica para agenciadores de viagens. A plataforma visa oferecer uma solução prática e eficaz para a divulgação e gerenciamento de excursões, proporcionando aos viajantes um ambiente centralizado para pesquisa, comparação e reserva de viagens. Ao analisar casos de sucesso e desafios enfrentados por outras plataformas similares, a pesquisa busca identificar as melhores práticas e estratégias para otimizar a eficácia da aplicação proposta. Além disso, explora os potenciais impactos positivos na redução da poluição visual nas áreas urbanas, à medida que os agenciadores adotam plataformas online para promover suas excursões. Ao concluir, este projeto visa contribuir para a compreensão das transformações em curso no setor de viagens, destacando a importância de soluções tecnológicas inovadoras na promoção de práticas mais sustentáveis, eficientes e esteticamente agradáveis.

Palavras-chave: Web Service, React, Spring Boot, Excursões, MySQL

Abstract

Oliveira, D. R. *Web application for managing and publicizing excursions in React GoExcursions*. Monografia (Graduacao) — Centro universitario de Formiga – Unifor-mg – Formiga-MG, 2023.

With the effects caused by the visual pollution of advertisements nailed to poles and walls and the positive revolution provided by web applications in the search for travel. Recognizing these adverse impacts of traditional advertising strategies in the urban environment, this study proposes an innovative solution: the development of a web application dedicated to managing and advertising excursions for travel agents. The research examines visual pollution as a social and aesthetic problem, highlighting the challenges faced by contemporary society in managing these impacts. At the same time, it explores the growing benefits of web applications in the travel sector, highlighting their influence in simplifying the planning and booking process for travelers. This project focuses on the development and implementation of a specific web application for travel agents. The platform aims to offer a practical and effective solution for advertising and managing tours, providing travelers with a centralized environment for searching, comparing and booking trips. By analyzing success stories and challenges faced by other similar platforms, the research seeks to identify best practices and strategies to optimize the effectiveness of the proposed application. In addition, it explores the potential positive impacts on reducing visual pollution in urban areas as travel agents adopt online platforms to promote their tours. In conclusion, this project aims to contribute to understanding the ongoing transformations in the travel sector, highlighting the importance of innovative technological solutions in promoting more sustainable, efficient and aesthetically pleasing practices.

Keywords: Web Service, React, Spring Boot, Excursions, MySql

Introdução

1.1 Considerações Iniciais

A sociedade contemporânea testemunha uma metamorfose profunda, onde as tradicionais paisagens urbanas, marcadas pela poluição visual de propagandas em postes e muros, coexistem com a crescente influência positiva das tecnologias digitais. Este estudo propõe uma análise crítica dessa dualidade, reconhecendo os desafios da poluição visual e, simultaneamente, explorando o potencial transformador do crescimento do uso da internet no Brasil.

A Figura 1.1 ilustra vividamente a poluição visual em ambientes urbanos, com postes repletos de propagandas, evidenciando os desafios estéticos enfrentados pela sociedade.



Figura 1.1: Poluição visual com poste cheio de propagandas afixadas.(RIGON, 2013)

Conforme apontado por pesquisas recentes, o Brasil experimenta um notável aumento no acesso à internet. De acordo com (NERY; BRITTO, 2022), a “Internet chega a 90,0% dos domicílios do país em 2021, com alta de 6 pontos percentuais frente a 2019, quando 84,0% dos domicílios tinham acesso à grande rede”, evidenciando uma mudança significativa nos padrões

de comportamento da população. Esse fenômeno não apenas destaca a penetrante presença da tecnologia na vida cotidiana, mas também reflete a transição para uma era digital em constante expansão.

A Figura 1.2 revela não apenas um aumento substancial no acesso à internet em domicílios brasileiros, mas também oferece insights valiosos sobre como diferentes regiões respondem a essa tendência. Tal fenômeno não apenas demonstra a abrangência da conectividade, mas também sugere variações significativas nos padrões de comportamento online.

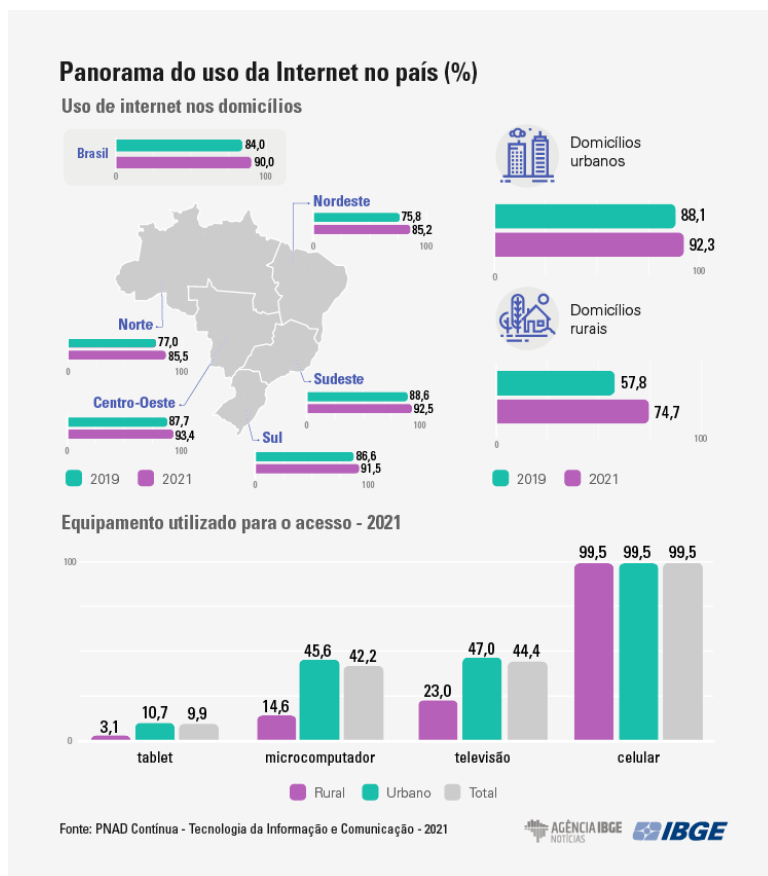


Figura 1.2: O crescimento do uso da internet nos domicílios e em cada região do Brasil. (NERY; BRITTO, 2022)

Paralelamente a esse crescimento, a internet se tornou uma ferramenta essencial na busca de informações, produtos e serviços, incluindo a organização de viagens. A ascensão das "travel tech" que pode ser traduzido como "tecnologias de viagem" empresas inovadoras que utilizam tecnologia para transformar a indústria de viagens - é um testemunho claro dessa mudança de paradigma.

De acordo com (SEBRAE, 2023), estas empresas têm desempenhado um papel crucial na remodelação da experiência do viajante, introduzindo soluções inovadoras que vão desde a personalização de itinerários até a simplificação de processos de reserva. A influência dessas "travel techs" não só responde às demandas contemporâneas por praticidade, mas também molda ativamente a forma como percebemos e consumimos viagens.

Diante deste cenário de transformação, surge a necessidade de soluções inovadoras que não apenas aproveitem a onda crescente de conectividade, mas também ofereçam respostas práticas

aos desafios enfrentados pela sociedade contemporânea. É nesse contexto que este projeto se insere, propondo uma aplicação web dedicada ao gerenciamento e divulgação de excursões por agenciadores de viagens. Esta plataforma não só abraça as possibilidades proporcionadas pela internet, mas também representa uma resposta à demanda por experiências de viagem mais personalizadas e eficientes, alinhando-se assim com as tendências introduzidas pelas "travel techs". Ao explorar essa abordagem inovadora, buscamos contribuir para uma nova era na qual a tecnologia não apenas facilita, mas também aprimora a jornada do viajante moderno.

1.2 Objetivos do Trabalho

Neste tópico serão apresentados os objetivos geral e específicos que serão utilizados para gerenciamento e conclusão deste projeto.

1.2.1 Objetivo Geral

Este trabalho tem como objetivo principal desenvolver uma solução inovadora para os desafios enfrentados pela sociedade contemporânea, abordando a poluição visual causada por propagandas em postes e muros, ao mesmo tempo em que capitaliza as oportunidades proporcionadas pelo crescente uso da internet no Brasil. O foco central é a criação de uma aplicação web dedicada ao gerenciamento e divulgação de excursões por agenciadores de viagens.

1.2.2 Objetivos Específicos

- Analisar a Poluição Visual Urbana: Investigar e analisar criticamente os impactos da poluição visual urbana causada por propagandas em postes e muros, identificando desafios estéticos e sociais.
- Avaliar o Papel das "Travel Techs": Investigar o papel das "travel techs" na transformação da indústria de viagens, destacando as inovações introduzidas por essas empresas e seu impacto na experiência do viajante.
- Contribuir para a Compreensão das Transformações Sociais: Contribuir para a compreensão mais ampla das transformações sociais em curso, destacando como inovações tecnológicas, como a aplicação web proposta, podem influenciar positivamente práticas tradicionais e melhorar a qualidade de vida na sociedade moderna.

1.3 Estrutura da Monografia

Este trabalho está estruturado em seis capítulos. O primeiro capítulo introduz o assunto, apresentando considerações iniciais que motivaram a concepção do projeto, além de expor os objetivos geral e específicos deste trabalho. No segundo capítulo, fornecemos um embasamento teórico essencial para compreender os elementos principais da pesquisa, incluindo uma breve explicação das tecnologias utilizadas no desenvolvimento da aplicação web. No terceiro capítulo, realizamos um estudo abrangente da arte, investigando soluções existentes no gerenciamento de excursões e avaliando plataformas online de viagens. Analisamos tanto casos de sucesso

quanto desafios enfrentados por aplicações web semelhantes. O quarto capítulo descreve detalhadamente a metodologia adotada para o desenvolvimento da aplicação web. No quinto capítulo, apresentamos os testes realizados e os resultados obtidos. Por fim, no sexto capítulo, concluímos o trabalho, recapitulando os objetivos e destacando as contribuições para a área, ao mesmo tempo em que definimos possíveis direções para trabalhos futuros.

Referencial Teórico

2.1 Considerações Iniciais

No âmbito das excursões e viagens, é crucial entender a dinâmica desses eventos e como as tecnologias modernas podem aprimorar essa experiência. Inicialmente, é importante considerar a complexidade do gerenciamento de excursões, que envolve desde a organização logística até a divulgação eficiente. Nesse contexto, a aplicação de tecnologias web pode desempenhar um papel significativo na simplificação e otimização desses processos.

2.2 Excursões e Viagens

Viagens e excursões desempenham um papel crucial na vida das pessoas, proporcionando experiências enriquecedoras, lazer e oportunidades para explorar novos destinos. No contexto do projeto, a análise desse universo visa compreender as necessidades dos viajantes e dos agenciadores de excursões, identificando como a tecnologia pode aprimorar essa interação.

O turismo é uma importante fonte de renda para muitas pessoas e comunidades. No Brasil, por exemplo, o turismo representou 7,8% do PIB em 2023, gerando quase 8 milhões de empregos. O turismo também pode contribuir para a promoção da cultura e do intercâmbio entre diferentes culturas. Ao viajar para outros países ou regiões, as pessoas têm a oportunidade de conhecer novas culturas, aprender sobre diferentes formas de vida e apreciar a diversidade do mundo. O turismo também pode proporcionar experiências de lazer e enriquecimento pessoal. Ao viajar, as pessoas podem relaxar, desfrutar da natureza, conhecer novas pessoas e ter novas experiências. (MACIEL, 2023)

Agenciadores de excursões enfrentam desafios ao divulgar seus pacotes, especialmente no ambiente urbano, onde a poluição visual pode obscurecer suas mensagens. A transição para plataformas digitais oferece uma alternativa inovadora, potencialmente reduzindo a dependência

de métodos tradicionais e ampliando o alcance para um público mais amplo.(ICASSATTI, 2023)

A tecnologia desempenha um papel transformador na indústria de viagens. Plataformas online, conhecidas como "travel techs,"simplificam o processo de busca, comparação e reserva de viagens. Essas soluções não apenas atendem às expectativas contemporâneas por praticidade, mas também moldam a maneira como as pessoas percebem e planejam suas viagens.(ICASSATTI, 2023)

2.3 Tecnologias Utilizadas no Desenvolvimento da Aplicação Web

Ao abordar o desenvolvimento da aplicação web dedicada ao gerenciamento e divulgação de excursões, várias tecnologias desempenham papéis essenciais. O banco de dados MySQL fornece uma base sólida para o armazenamento e recuperação eficiente de dados, enquanto o backend, desenvolvido em Java com Hibernate e JPA, oferece robustez e flexibilidade na manipulação das informações.

A utilização do Spring Boot no backend proporciona uma estrutura ágil para o desenvolvimento, facilitando a criação de APIs robustas. O frontend da aplicação, construído com HTML, CSS, React, JavaScript e TypeScript, oferece uma interface amigável e responsiva para os usuários.

O uso de JSON como formato de intercâmbio de dados permite uma comunicação eficiente entre o frontend e o backend.

A implementação de web services e web services RESTful é fundamental para a integração eficaz de diferentes componentes da aplicação, garantindo uma experiência contínua para os usuários finais.

Essas escolhas tecnológicas foram cuidadosamente consideradas para garantir não apenas a funcionalidade da aplicação, mas também a eficiência, a escalabilidade e a usabilidade. A combinação dessas tecnologias forma uma estrutura sólida que sustenta a proposta inovadora deste trabalho, oferecendo uma solução moderna e eficaz para o gerenciamento e divulgação de excursões.

2.4 Metodologia Ágil - Scrum

O Scrum é uma metodologia ágil amplamente utilizada no desenvolvimento de software. Ela se destaca por sua abordagem iterativa e incremental, dividindo o projeto em ciclos chamados sprints. Cada sprint representa um período fixo, geralmente de duas a quatro semanas, durante o qual as atividades são planejadas, executadas e revisadas.(SUTHERLAND, 2014)

2.5 Banco de dados

Um banco de dados é um sistema organizado para coletar, armazenar, gerenciar e fornecer acesso a dados. Ele serve como um repositório centralizado para armazenar informações que podem ser facilmente recuperadas, atualizadas e gerenciadas. Os bancos de dados desempenham um papel crucial em uma variedade de aplicações e contextos, desde sistemas de gerenciamento

de conteúdo online até aplicativos empresariais e científicos.(SUEHRING, 2002)

Principais Tipos de Bancos de Dados:

- Relacional: Organiza os dados em tabelas relacionadas umas às outras. Exemplos incluem MySQL, PostgreSQL, SQL Server.
- NoSQL: Não utiliza o modelo relacional. É adequado para dados não estruturados ou semiestruturados. Exemplos incluem MongoDB, Cassandra, Redis.
- Objeto-Relacional: Combina características de bancos de dados relacionais e orientados a objetos.
- In-Memory: Armazena dados na memória principal (RAM) para acesso mais rápido. Exemplos incluem Redis, Memcached.

2.5.1 SQL

SQL é uma linguagem de programação usada por quase todos os bancos de dados relacionais para consultar, manipular e definir dados e fornecer controle de acesso. O SQL foi desenvolvido pela primeira vez na IBM nos anos 1970, com a Oracle como principal contribuinte, o que levou à implementação do padrão SQL ANSI; o SQL estimulou muitas extensões de empresas como IBM, Oracle e Microsoft. Embora o SQL ainda seja amplamente usado hoje em dia, novas linguagens de programação estão começando a aparecer.(ORACLE,)

2.5.2 MySQL

MySQL é um sistema de gerenciamento de banco de dados relacional de código aberto baseado em SQL. Ele foi projetado e otimizado para aplicativos da web e pode ser executado em qualquer plataforma. Como surgiram requisitos novos e diferentes com a internet, o MySQL tornou-se a plataforma preferida para desenvolvedores da web e aplicativos baseados na web. Como foi projetado para processar milhões de consultas e milhares de transações, o MySQL é uma escolha popular para empresas de comércio eletrônico que precisam gerenciar várias transferências de dinheiro. A flexibilidade sob demanda é o principal recurso do MySQL.(DUBOIS, 2013)

O MySQL é um sistema de gerenciamento de banco de dados(SGBD) por trás de alguns dos principais sites e aplicativos baseados na web do mundo, incluindo Airbnb, Uber, LinkedIn, Facebook, Twitter e YouTube.(ORACLE,)

2.6 Backend

O termo "backend" refere-se à parte de um sistema de software responsável pelo processamento e gerenciamento dos dados, regras de negócios e lógica da aplicação. É a camada que opera nos bastidores, manipulando informações e garantindo que o frontend (parte visível e interativa para o usuário) funcione conforme o esperado.(TOTVS, 2020)

Principais Funções do Backend:

- **Lógica de Negócios:** O backend implementa as regras e a lógica que definem o comportamento da aplicação, processando as operações e decisões fundamentais para a funcionalidade do sistema.
- **Gerenciamento de dados:** Responsável pelo armazenamento, recuperação e manipulação de dados. Isso envolve interações com bancos de dados e outros sistemas de armazenamento.
- **Segurança:** Implementa medidas de segurança para proteger dados sensíveis e garantir que apenas usuários autorizados tenham acesso a determinadas funcionalidades.
- **Integração de Sistemas:** Facilita a comunicação entre diferentes partes do sistema e integração com serviços externos, quando necessário.
- **Processamento de Requisições do Cliente:** Recebe as requisições provenientes do frontend, processa-as e retorna as respostas apropriadas.
- **Lida com a Lógica do Servidor:** Executa operações que não são viáveis ou seguras de serem realizadas no frontend.

2.7 Java

Java é uma linguagem de programação criada por James Gosling e sua equipe na Sun Microsystems, lançada oficialmente em 23 de maio de 1995. Desde então, Java tem se destacado como uma das linguagens mais populares e amplamente utilizadas no mundo do desenvolvimento de software. De acordo com o índice TIOBE em novembro de 2023 java ocupa a 4^a colocação entre as linguagens mais utilizadas no mundo.(INDEX, 2023)

Uma das características mais marcantes do Java é sua portabilidade. O código Java é compilado para um formato intermediário chamado bytecode, que pode ser executado em qualquer dispositivo que tenha uma Máquina Virtual Java (JVM) adequada. Isso torna Java uma escolha ideal para desenvolvimento multiplataforma.(LUCKOW; MELO, 2010)

Java é uma linguagem orientada a objetos, seguindo os princípios da programação orientada a objetos, como encapsulamento, herança e polimorfismo. Além disso, a linguagem foi projetada com um foco significativo em segurança, executando código em um ambiente seguro da JVM para prevenir problemas comuns de segurança.(ARAGÃO, 2013)

Com um vasto ecossistema de bibliotecas, frameworks e ferramentas, Java é versátil, facilitando o desenvolvimento de uma ampla variedade de aplicativos, desde aplicações desktop até sistemas distribuídos em larga escala. A linguagem é comumente utilizada no desenvolvimento de aplicações web e empresariais, assim como em dispositivos móveis, embora Kotlin tenha se tornado a linguagem preferida para o desenvolvimento Android.

A figura 2.1 exemplifica uma comunicação das aplicações web que rodam nos navegadores, fazendo comunicação com o banco de dados, utilizando o backend como uma ponte. No backend contém a linguagem Java por exemplo, que com auxílio do Hibernate e Spring boot faz a inserção e consulta dos dados.

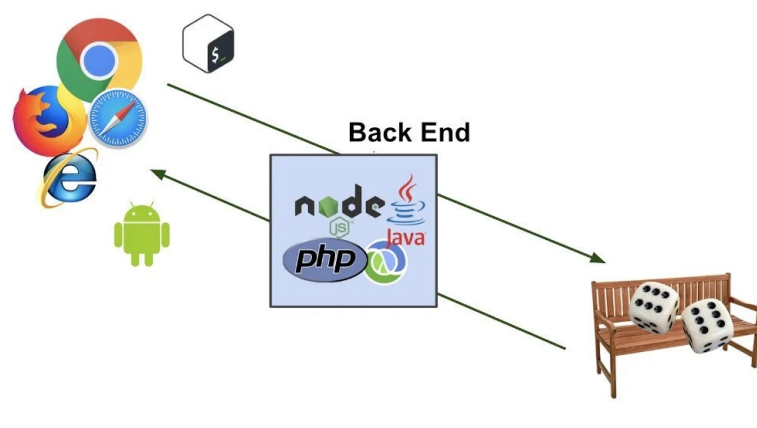


Figura 2.1: Exemplo de sincronização de aplicações web com banco de dados através do back-end.(SOUTO, 2023)

2.8 JPA

A Java Persistence API (JPA) é uma especificação do Java que descreve a gestão de dados entre objetos Java e o banco de dados relacional. Ela fornece um meio de mapear objetos Java para entidades do banco de dados e vice-versa.(LECHETA, 2015)

Principais Conceitos e Funcionalidades do JPA:

- Entidades: As entidades JPA são objetos Java que representam dados armazenados no banco de dados. Essas entidades são anotadas com `@Entity` para indicar que são mapeadas para tabelas no banco de dados.
- Mapeamento Objeto-Relacional (ORM): JPA simplifica o mapeamento entre objetos Java e tabelas de banco de dados. A anotação `@Table` pode ser usada para especificar o nome da tabela associada à entidade.
- Identificadores: A anotação `@Id` é usada para especificar a chave primária da entidade. O JPA oferece suporte a várias estratégias de geração de identificadores, como `GenerationType.IDENTITY` para colunas autoincrementadas.
- Relacionamentos entre Entidades: O JPA permite definir relacionamentos entre entidades, como `@OneToOne`, `@OneToMany`, e `@ManyToOne`. Esses relacionamentos são mapeados para chaves estrangeiras no banco de dados.
- Consultas JPQL (Java Persistence Query Language): O JPQL é uma linguagem de consulta semelhante ao SQL, mas opera em termos de entidades JPA. Ele permite realizar consultas orientadas a objetos no lugar de consultas SQL tradicionais.

2.9 Hibernate

O Hibernate é uma estrutura de mapeamento objeto-relacional (ORM) para a linguagem Java. Ele simplifica a interação entre objetos Java e bancos de dados relacionais, permitindo que os desenvolvedores trabalhem com objetos Java no código, enquanto o Hibernate se encarrega de

traduzir esses objetos para instruções SQL compreensíveis pelos bancos de dados.(LUCKOW; MELO, 2010)

Segundo (LECHETA, 2015) o Hibernate e Spring são dois dos mais famosos frameworks Java do mercado, sendo o Hibernate uma implementação da JPA só que apresentando uma gama maior de funcionalidades.

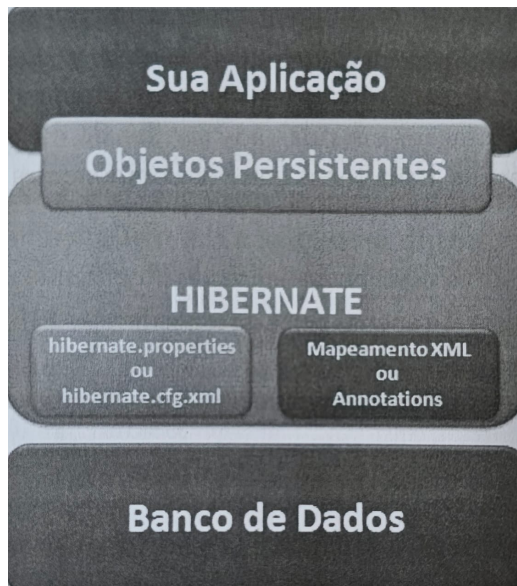


Figura 2.2: Arquitetura simplificada do Hibernate.(LECHETA, 2015)

A figura 2.2 ilustra a arquitetura do Hibernate e sua integração com a aplicação desenvolvida. Esta representação visual oferece insights sobre como o Hibernate atua como uma camada de persistência entre objetos Java e o banco de dados relacional.

2.10 Framework Spring

Segundo (SANTANA, 2021) O Spring é um framework Java com uma vasta quantidade de projetos, como o Spring Boot por exemplo. Além do Spring Boot é possível também utilizar outros frameworks com o Spring e até mesmo as bibliotecas do Enterprise Java Beans (EJB). O Spring é bastante antigo — sua primeira versão foi publicada em 2002 — e é um projeto robusto e estável.

2.10.1 Spring Boot

O Spring Boot é uma extensão do framework Spring que oferece uma abordagem simplificada para o desenvolvimento de aplicações Java. Sua proposta é proporcionar maior produtividade ao eliminar a complexidade associada à configuração manual, seguindo a filosofia de "convenção sobre configuração".(SANTANA, 2021)

O uso do Spring Boot transforma o desenvolvimento de aplicações Java, proporcionando maior agilidade, redução de código boilerplate ¹ e facilitando a criação de aplicações robustas

¹Boilerplate refere-se a seções de código repetitivas, padronizadas e necessárias em muitos lugares em um programa. Essas seções não contribuem significativamente para a lógica da aplicação, mas são necessárias para garantir o funcionamento adequado do código.

e escaláveis.

Principais Conceitos e Características:

1. **Convenção sobre Configuração:** Spring Boot adota uma abordagem baseada em convenções, reduzindo a necessidade de configurações extensas. Isso permite que os desenvolvedores se concentrem mais na lógica de negócios do que em detalhes de configuração.
2. **Configuração Automática:** A configuração automática do Spring Boot é baseada nas dependências presentes no projeto. Por exemplo, a detecção de um banco de dados configura automaticamente o DataSource, simplificando o processo de configuração.
3. **Starters:** Os starters do Spring Boot são conjuntos de dependências pré-configuradas para cenários comuns, como desenvolvimento web, acesso a banco de dados e segurança. Esses starters simplificam o gerenciamento de dependências e aceleram o desenvolvimento.
4. **Embedded Containers:** O suporte a containers embarcados, como Tomcat e Jetty, elimina a necessidade de configurações externas de servidores. Isso facilita a execução e o teste da aplicação sem a necessidade de um servidor externo.
5. **Spring Boot Actuator:** O Spring Boot Actuator fornece recursos para monitoramento e gerenciamento de aplicações em produção. Ele expõe endpoints para informações detalhadas sobre a aplicação, como métricas, estado da aplicação e informações do ambiente.
6. **DevTools:** Os Spring Boot DevTools oferecem ferramentas para facilitar o desenvolvimento, incluindo a reinicialização automática da aplicação em resposta a alterações no código-fonte.
7. **Integração com Spring Framework:** Baseado no poderoso framework Spring, o Spring Boot herda os conceitos fundamentais, como inversão de controle (IoC), injeção de dependência e programação orientada a aspectos (AOP).

2.11 Web Service

Um Web Service é uma tecnologia que permite a comunicação e a troca de dados entre sistemas distintos pela internet. Ele fornece uma interface padrão para que aplicativos diferentes possam interagir entre si, independentemente da linguagem de programação ou da plataforma em que foram desenvolvidos. Os Web Services são independentes de protocolos, o que significa que podem ser implementados usando diferentes protocolos de comunicação, como HTTP, HTTPS, SOAP (Simple Object Access Protocol), REST (Representational State Transfer), entre outros.(LECHETA, 2015)

Os Web Services aderem a padrões abertos, garantindo que a comunicação entre os sistemas seja transparente e baseada em normas reconhecidas globalmente. Normalmente, os dados trocados entre os sistemas são formatados em XML (*Extensible Markup Language*) ou JSON (*JavaScript Object Notation*), tornando-os legíveis tanto para máquinas quanto para seres humanos.(LECHETA, 2015)

Os Web Services desempenham um papel fundamental na construção de aplicações modernas, permitindo a construção de sistemas distribuídos flexíveis e interoperáveis. Sua implementação adequada pode melhorar significativamente a eficiência e a escalabilidade de sistemas de software.(LECHETA, 2015)

Tipos de Web Services:

- SOAP (*Simple Object Access Protocol*): Utiliza XML para formatação de mensagens e é geralmente encapsulado em protocolos como HTTP ou SMTP. Possui um conjunto mais estruturado de padrões.
- REST (*Representational State Transfer*): Baseado nos princípios da arquitetura REST, utiliza protocolos web padrão (HTTP, URI, XML, JSON) e é mais simples e flexível em comparação com o SOAP.

2.12 Web Service Restfull

Um Web Service RESTful (*Representational State Transfer*) é um tipo de serviço web que segue os princípios da arquitetura REST. Ao contrário dos serviços SOAP, os Web Services RESTful são conhecidos por sua simplicidade, flexibilidade e eficiência. Eles são amplamente utilizados para criar APIs (Interfaces de Programação de Aplicações) que possibilitam a comunicação e a integração entre sistemas distribuídos na web.(RICHARDSON; RUBY, 2008)

Esse tipo de Web Service utiliza uma arquitetura baseada em recursos(dados ou serviços) que são identificados por URIs (Uniform Resource Identifiers), e as interações que são baseadas na manipulação desses recursos. Cada requisição do cliente para o servidor contém todas as informações necessárias para entender e processar a solicitação.

As requisições são feitas utilizando métodos HTTP padrão, como GET, POST, PUT e DELETE, para realizar operações nas representações dos recursos. Cada método tem um significado específico, simplificando a interface e proporcionando uma correspondência direta entre as operações e os métodos HTTP. Os recursos podem ser representados em diferentes formatos, como XML ou JSON, facilitando a leitura tanto por máquinas quanto por humanos.(RICHARDSON; RUBY, 2008)

Exemplo de Uso de Métodos HTTP em Web Services RESTful:

- GET: Obter dados de um recurso.
- POST: Criar um novo recurso.
- PUT: Atualizar um recurso existente.
- DELETE: Remover um recurso.

2.13 Frontend

O termo frontend refere-se à parte de um aplicativo ou sistema que interage diretamente com os usuários. É a interface visível, responsável por apresentar dados, receber entradas e fornecer uma experiência interativa. Essa camada é construída usando tecnologias como HTML, CSS e JavaScript.(FLANAGAN, 2012)

2.14 HTML

O HTML (*Hypertext Markup Language*) é a linguagem padrão para a criação e a estruturação de páginas web. Ele fornece uma maneira de organizar o conteúdo, como texto, imagens, links e outros elementos, para serem exibidos em navegadores web. A estrutura básica do HTML é composta por tags que envolvem o conteúdo e fornecem informações sobre como esse conteúdo deve ser apresentado.(SILVA, 2018)

A versão mais recente do HTML é o HTML5, introduzindo novos elementos e atributos para melhorar a semântica e fornecer suporte integrado para áudio, vídeo, gráficos e aplicações interativas.(SILVA, 2018)

O HTML é a espinha dorsal da web, fornecendo uma estrutura para criar conteúdo acessível e interativo. Combinado com CSS e JavaScript, forma a base para o desenvolvimento web moderno.(SILVA, 2018)

2.15 CSS

O CSS (*Cascading Style Sheets*) é uma linguagem de estilo utilizada para descrever a apresentação de um documento HTML. Ele permite que os desenvolvedores controlem a aparência e o layout dos elementos na página, proporcionando uma experiência visual agradável e consistente para os usuários.(SILVA, 2018)

O CSS funciona por meio de seletores que identificam os elementos HTML e atribuem a eles propriedades de estilo. As propriedades controlam aspectos como cor, fonte, espaçamento, margens, bordas, entre outros. (SILVA, 2018)

O CSS desempenha um papel fundamental na criação de interfaces atraentes e funcionais para os usuários. Com as práticas modernas, como Flexbox e Grid, e ferramentas como pré-processadores, os desenvolvedores têm à disposição recursos poderosos para estilizar eficientemente páginas web.(SILVA, 2018)

2.16 React

React é uma biblioteca JavaScript mantida pelo Facebook, projetada para facilitar a criação de interfaces de usuário interativas e reativas. Uma das principais características do React é a sua abordagem declarativa para a construção de componentes, permitindo que os desenvolvedores descrevam como a interface deve ser exibida em diferentes estados.(CHOI, 2020)

O React organiza a interface em componentes reutilizáveis e autocontidos. Cada componente pode ter seu próprio estado interno e propriedades que controlam seu comportamento e aparência. Os componentes utilizam o JSX (*JavaScript XML*) que é uma extensão da sintaxe JavaScript que permite escrever marcação HTML dentro do código JavaScript. Isso facilita a criação de elementos React de forma concisa.(STEFANOV, 2021)

React revolucionou o desenvolvimento frontend ao introduzir um modelo de programação declarativo e reativo. Sua popularidade é evidenciada pelo número significativo de empresas e desenvolvedores que o adotaram para criar interfaces modernas e dinâmicas.(STEFANOV, 2021)

Benefícios do React:

- Reatividade Eficiente: React atualiza apenas as partes da interface que foram alteradas, proporcionando uma experiência reativa e eficiente.
- Reutilização de Componentes: A abordagem baseada em componentes facilita a reutilização de código, contribuindo para uma manutenção mais fácil e escalável.
- Comunidade Ativa: React tem uma comunidade robusta e ativa, com uma ampla gama de bibliotecas e ferramentas adicionais disponíveis.

A Figura 2.3 mostra um componente em React. Que no caso se trata de um "SnackBar" junto com um "Alert" utilizado para exibir notificações para os usuários.

```
import { Alert, Snackbar } from "@mui/material";
import * as React from "react";
...
interface Props {
  mensagem: string,
  open: boolean,
  tipoSnack: "error" | "info" | "success" | "warning";
  setOpen: React.Dispatch<React.SetStateAction<boolean>>,
}

function SnackAlert({ open, setOpen, mensagem, tipoSnack }: Props) {
  return (
    <Snackbar
      open={open}
      autoHideDuration={tipoSnack === "success" ? 1000 : 2000}
      onClose={() => setOpen(false)} anchorOrigin={{ vertical: "top", horizontal: "right" }}
      sx={{ marginTop: 8 }}
    >
      <Alert severity={tipoSnack} variant="filled" sx={{ width: "100%" }}>
        {mensagem}
      </Alert>
    </Snackbar>
  );
}

export default SnackAlert;
```

Figura 2.3: Componente em React parte do código da aplicação GoExcursões.

2.17 JavaScript

JavaScript, uma linguagem de programação de alto nível e interpretada, inicialmente concebida para operar nos navegadores web, transcendeu seu papel original no frontend e consolidou-se como uma ferramenta versátil para o desenvolvimento web. Uma das características fundamentais do JavaScript é sua tipagem dinâmica, permitindo que as variáveis acomodem diferentes tipos de dados ao longo do tempo, conferindo flexibilidade ao desenvolvedor. (FLANAGAN, 2012)

Com estruturas de controle como instruções condicionais e loops, o JavaScript oferece recursos robustos para o controle de fluxo no código. Funções, elementos essenciais, possibilitam a organização e reutilização eficiente do código, aceitando parâmetros e retornando valores. O sistema de objetos e arrays contribui para a representação e manipulação de dados de maneira eficaz. (PRESCOTT; SOUZA, 2016)

Além de seu papel primordial na manipulação do Document Object Model (DOM) para interatividade nas páginas web, o JavaScript abraçou a assincronicidade. Essa característica é crucial para lidar com operações assíncronas, como eventos e chamadas de API, sem bloquear a execução do código, proporcionando uma experiência de usuário mais eficiente.(PEREIRA, 2014)

O ecossistema JavaScript foi ampliado significativamente com a introdução do Node.js, permitindo a execução de JavaScript no lado do servidor. O npm(*Node Package Manager*), um gerenciador de pacotes, simplificou a instalação e gestão de bibliotecas e ferramentas. Além disso, frameworks e bibliotecas, como React, Angular e Vue.js para o frontend, e Express.js para o backend, destacam-se como pilares do desenvolvimento moderno em JavaScript.(STEFANOV, 2021)

O impacto do JavaScript no cenário de desenvolvimento web é indiscutível. Sua comunidade ativa, vasto ecossistema e constante evolução evidenciam a importância contínua dessa linguagem na construção de interfaces dinâmicas e na viabilização de aplicações web eficazes.(STEFANOV, 2021)

2.18 TypeScript

O TypeScript é uma extensão do JavaScript que introduz o conceito de tipagem estática, fornecendo aos desenvolvedores uma camada adicional de verificação de tipos durante a fase de desenvolvimento. Desenvolvido pela Microsoft, o TypeScript visa melhorar a qualidade, legibilidade e escalabilidade do código JavaScript.(GRYNHAUS et al., 2021)

A principal característica do TypeScript é seu sistema de tipos estáticos, permitindo aos desenvolvedores declarar tipos para variáveis, parâmetros de função e retornos de função. Essa abordagem proporciona benefícios significativos, como a detecção precoce de erros e a criação de código mais robusto, facilitando a manutenção em projetos complexos.(ALVES; BACH, 2021)

O TypeScript mantém total compatibilidade com o JavaScript, permitindo a migração gradual de projetos existentes para a nova linguagem. Além disso, sua integração eficiente com ferramentas populares de desenvolvimento, como o Visual Studio Code, contribui para uma experiência de desenvolvimento aprimorada.(GOLDBERG, 2022)

Ao transpilar para JavaScript padrão durante a fase de compilação, o TypeScript é executado em qualquer ambiente que suporte JavaScript, garantindo interoperabilidade com bibliotecas e frameworks existentes. Essa flexibilidade permite que os desenvolvedores desfrutem dos benefícios do TypeScript enquanto aproveitam o vasto ecossistema JavaScript.(ALVES; BACH, 2021)

Em suma, o TypeScript representa uma evolução significativa no desenvolvimento web, oferecendo vantagens substanciais em termos de qualidade de código, escalabilidade e colaboração. Sua crescente adoção na comunidade de desenvolvedores destaca sua relevância e impacto positivo no desenvolvimento de aplicações modernas.(GOLDBERG, 2022)

2.19 Considerações Finais

Ao encerrar este percurso pelo referencial teórico, emergem reflexões essenciais sobre as tecnologias que moldam o cenário contemporâneo de viagens e excursões. A convergência de plataformas online, inovações em desenvolvimento web e o crescimento exponencial da conectividade delineiam um horizonte de possibilidades que ressoa fortemente com os objetivos propostos neste projeto.

O fenômeno da ascensão das "travel techs" empresas inovadoras que utilizam a tecnologia para revolucionar a indústria de viagens, é incontestavelmente um catalisador de mudança. A capacidade dessas empresas em personalizar itinerários, simplificar processos de reserva e redefinir a experiência do viajante destaca o potencial transformador das tecnologias digitais. (ICASSATTI, 2023)

A escolha de tecnologias específicas, como HTML, CSS, React, Java, e outras, para o desenvolvimento da aplicação web proposta, não é apenas uma decisão técnica, mas uma resposta consciente aos padrões e práticas emergentes. Essas tecnologias não apenas habilitam a funcionalidade desejada, mas também incorporam eficiência, escalabilidade e uma experiência do usuário moderna.

Neste contexto, a tecnologia não é meramente uma facilitadora, mas uma diferenciadora. Proporciona a oportunidade não apenas de otimizar processos, mas de redefinir a interação entre agenciadores de viagens e potenciais exploradores. A aplicação web, ao adotar essas tecnologias, não apenas se insere no panorama digital, mas busca moldar ativamente as expectativas e experiências dos usuários

Trabalhos relacionados

3.1 Considerações Iniciais

Nesta seção serão apresentados alguns trabalhos que de certa forma estão relacionados com os assuntos ou tecnologias deste projeto bem como as ferramentas de apoio e métodos usados pelos autores, também serão apresentados os resultados obtidos em tais estudos.

3.2 TripAdvisor

O TripAdvisor é uma plataforma online global que desempenha um papel fundamental na indústria de viagens, proporcionando um espaço para avaliações, recomendações e informações relevantes para viajantes. Fundado em 2000, o TripAdvisor se tornou uma referência essencial para aqueles que buscam insights sobre destinos, acomodações, restaurantes e atividades. (LLC, 2023)

Principais Características do TripAdvisor:

- **Avaliações e Comentários:** O TripAdvisor é conhecido por seu vasto banco de dados de avaliações e comentários deixados por viajantes reais. Essas avaliações abrangem uma variedade de estabelecimentos, desde hotéis e restaurantes até atrações turísticas.

A Figura 3.1 mostra o exemplo de avaliações feitas por usuários na plataforma, que podem auxiliar outros viajantes na escolha do seu destino.

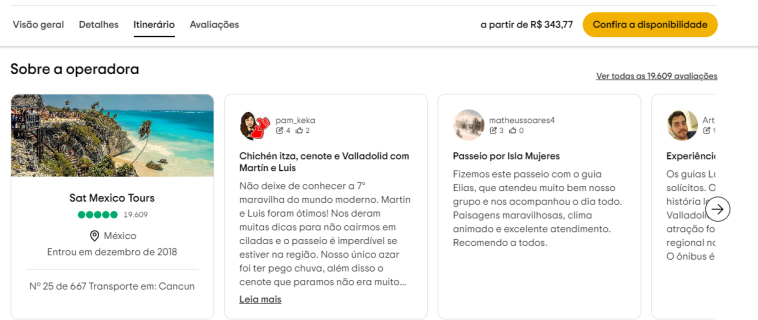


Figura 3.1: Exemplo de avaliações de usuário no TripAdvisor.

- **Rankings e Classificações:** A plataforma classifica os estabelecimentos com base nas avaliações e classificações recebidas. Isso permite que os usuários identifiquem rapidamente os locais mais bem avaliados em uma determinada área.

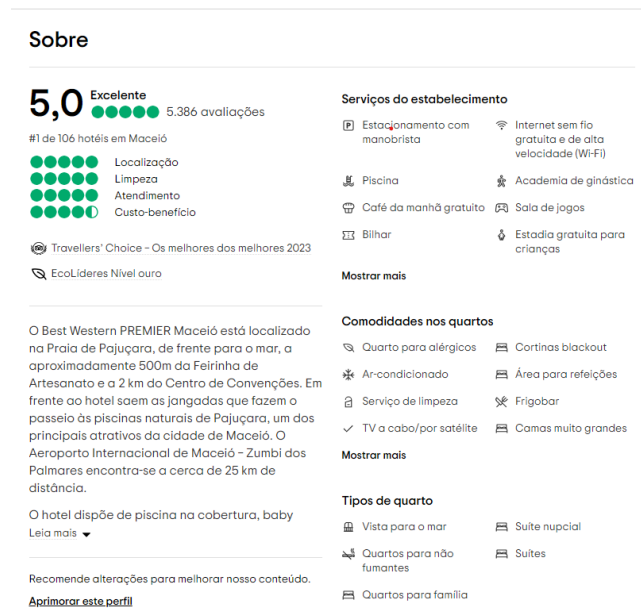


Figura 3.2: Exemplo de ranking e classificação de um estabelecimento no TripAdvisor.

A Figura 3.2 mostra o exemplo de ranking e classificações feitas pela plataforma para orientar os viajantes ao buscar um hotel ou destino.

- **Fotos e Vídeos:** Os usuários podem fazer upload de fotos e vídeos de suas experiências, proporcionando uma visão visual detalhada dos destinos e estabelecimentos.
- **Fóruns e Comunidades:** Além das avaliações, o TripAdvisor oferece fóruns e comunidades onde os viajantes podem trocar informações, fazer perguntas e obter conselhos personalizados.
- **Reservas e Planejamento:** A plataforma permite que os usuários façam reservas diretamente para hotéis, restaurantes e atividades, simplificando o processo de planejamento de viagens.

3.2.1 Importância para o Estudo

O TripAdvisor exerce uma influência significativa nas decisões dos viajantes, fornecendo informações confiáveis e experiências reais de outros usuários.

Para os estabelecimentos, ter uma presença positiva no TripAdvisor é crucial, pois influencia diretamente a escolha dos consumidores. O modelo de negócios da plataforma também inclui oportunidades de publicidade para empresas.

Ao analisar as avaliações e tendências no TripAdvisor, é possível identificar padrões de preferências dos viajantes, insights valiosos para a personalização de serviços.

O estudo do TripAdvisor também revela desafios enfrentados pelos estabelecimentos, como críticas negativas, e abre portas para oportunidades de melhorias baseadas no feedback dos usuários.

3.3 Hurb

A Hurb, é uma plataforma brasileira de viagens que tem desempenhado um papel proeminente no cenário do turismo online. A plataforma se destaca por oferecer pacotes de viagens, hospedagens, passeios e experiências com descontos e promoções atrativas.

A Hurb foi fundada em 2011 pelos irmãos João Ricardo Mendes (JR) e José Eduardo Mendes (Duda), que tiveram o apoio do amigo Antônio Gomes (Tuninho). Chamada de Hotel Urbano naquela época, a plataforma realizou os sonhos de mais de 6 milhões de pessoas ao democratizar o acesso às viagens ao redor do mundo. De lá pra cá os três amigos, conseguiram simplificar e melhorar a experiência de viagem, deixando-a mais acessível, humana, e o melhor, mais barata para todas as pessoas.(S.A., 2023)

Principais Características da Hurb:

- **Pacotes de Viagens Acessíveis:** A Hurb é conhecida por sua proposta de oferecer pacotes de viagens com preços acessíveis, permitindo que os usuários encontrem ofertas vantajosas para destinos variados.
- **Variedade de Experiências:** Além de hospedagens e pacotes, a Hurb oferece uma ampla gama de experiências, incluindo passeios, ingressos para atrações turísticas e opções de lazer. A Figura 3.3 mostra a variedade de serviços oferecidos pela plataforma.
- **Descontos e Promoções:** A plataforma se destaca por suas promoções e descontos expressivos, incentivando os usuários a explorarem novos destinos ou desfrutarem de experiências únicas a preços competitivos.
- **Interface Amigável:** A Hurb investe em uma interface amigável, facilitando a navegação e pesquisa por parte dos usuários. Isso contribui para uma experiência positiva do usuário.
- **Avaliações e Feedback:** Assim como o TripAdvisor, a Hurb permite que os usuários deixem avaliações e feedback sobre as experiências adquiridas, fornecendo informações úteis para outros viajantes.

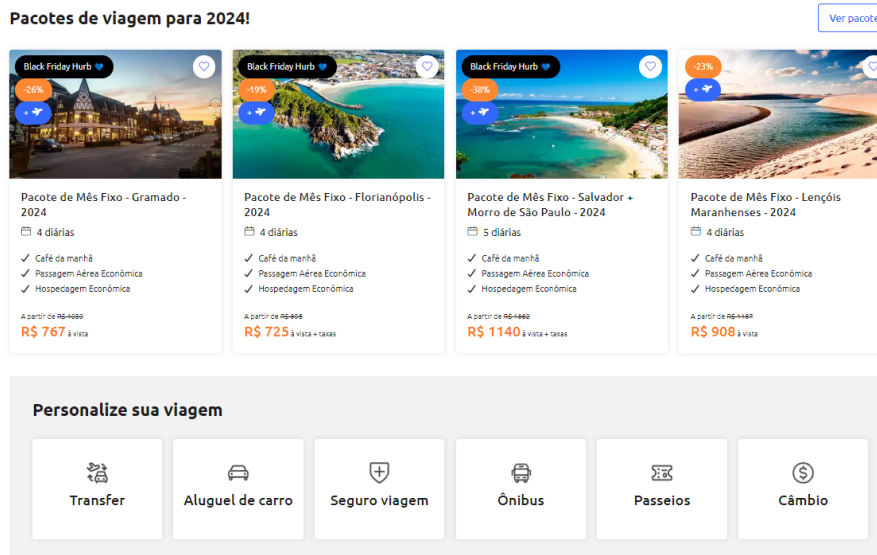


Figura 3.3: Variedade de serviços oferecidos.

3.3.1 Importância para o Estudo

A Hurb desafia a ideia de que viagens são luxos inacessíveis, buscando democratizar o turismo ao tornar opções atrativas mais acessíveis para um público diversificado.

Ao estudar a Hurb, é possível compreender estratégias eficazes de marketing online, especialmente no contexto de atrair consumidores por meio de ofertas irresistíveis.

A diversidade de pacotes e experiências oferecidas pelo Hurb destaca a importância de personalizar a experiência do usuário para atender às preferências individuais.

A análise da Hurb também proporciona insights sobre os desafios enfrentados e oportunidades presentes no mercado de viagens online no Brasil.

3.3.2 Considerações Finais

O TripAdvisor, como parte do estudo da arte, emerge como uma peça chave no ecossistema digital de viagens. Com seu impacto na tomada de decisões dos viajantes e influência nas práticas de negócios, compreender a dinâmica dessa plataforma é fundamental para o desenvolvimento de soluções inovadoras na indústria de viagens.

A Hurb, como parte integrante do estudo da arte, representa um modelo de negócios inovador que influenciou significativamente a forma como os brasileiros planejam e desfrutam de suas viagens. Sua abordagem única e sucesso no mercado são indicativos do dinamismo e das oportunidades presentes na indústria de viagens online.

A aplicação GoExcursões se diferencia do TripAdvisor e da Hurb ao proporcionar uma plataforma mais acessível para pequenos agenciadores. A essência da aplicação é criar um espaço onde diversos agenciadores possam divulgar suas excursões, oferecendo aos clientes uma variedade de opções em um único lugar. Isso amplia as oportunidades para os pequenos negócios e simplifica o acesso dos usuários a uma gama diversificada de experiências de viagem. Em contraste, o TripAdvisor e a Hurb, embora desempenhem papéis importantes, podem ser mais

centralizados em grandes operadoras ou avaliações individuais, enquanto o GoExcursões busca promover a diversidade e a descentralização no mercado de viagens e excursões.

Metodologia e Desenvolvimento

4.1 Considerações Iniciais

A metodologia adotada para o desenvolvimento da aplicação web dedicada ao gerenciamento e divulgação de excursões por agenciadores de viagens é um elemento crucial deste trabalho. Essa seção visa proporcionar uma compreensão clara das abordagens e estratégias empregadas para atingir os objetivos propostos.

4.2 Metodologia escolhida

A seleção da metodologia baseou-se na natureza inovadora do projeto, que busca integrar tecnologias web para aprimorar a divulgação e gestão de excursões. Considerando a abordagem prática e voltada para o desenvolvimento de software, optou-se por uma metodologia ágil, em particular, a metodologia Scrum. Essa escolha fundamenta-se na flexibilidade proporcionada por metodologias ágeis, permitindo ajustes contínuos com base nas demandas do projeto.

A Figura 4.1 o passo a passo utilizado pela metodologia Scrum, onde a partir da ideia do produto e das necessidades apresentadas, é montado um planejamento e definido as etapas do projeto que são chamadas de *Sprints*¹. Essas etapas são passadas para a equipe de desenvolvimento e no decorrer dos dias, vão sendo realizadas reuniões chamadas de *Daily*². Após a conclusão de todas as *Sprints* é feita uma nova reunião onde é feita a revisão e caso seja necessário alguma correção ou implementação adicional são apresentadas novas *Sprints*. Caso ocorra tudo bem o produto é implementado e disponibilizado para o cliente.(RIBAS,)

¹Sprints são ciclos de desenvolvimento fixos e curtos, geralmente com duração de duas a quatro semanas. Durante cada sprint, uma equipe Scrum trabalha para entregar um incremento de produto potencialmente utilizável.

²Daily é uma reunião realizada diariamente, geralmente no mesmo horário e local. Ela é uma reunião curta, limitada a um tempo fixo, comumente entre 15 a 30 minutos.

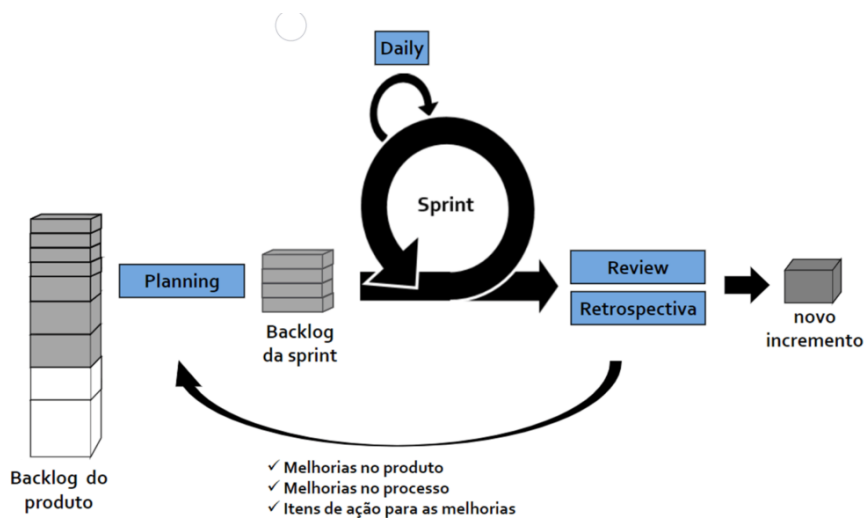


Figura 4.1: Esboço de Metodologia ágil Scrum.(RIBAS,)

4.3 Levantamento de Requisitos

O processo de levantamento de requisitos para o desenvolvimento da aplicação web GoExcursões envolveu reuniões fundamentais com os agenciadores. Foram consideradas as práticas vigentes antes da introdução da aplicação, destacando as seguintes informações coletadas:

1. Divulgação Atual: Agenciadores de viagens dependem fortemente de métodos tradicionais, como cartazes afixados em locais específicos e divulgações nas redes sociais, principalmente no Facebook e Instagram, para promover suas excursões. Como mostrado nas figuras 4.2 e 4.3.

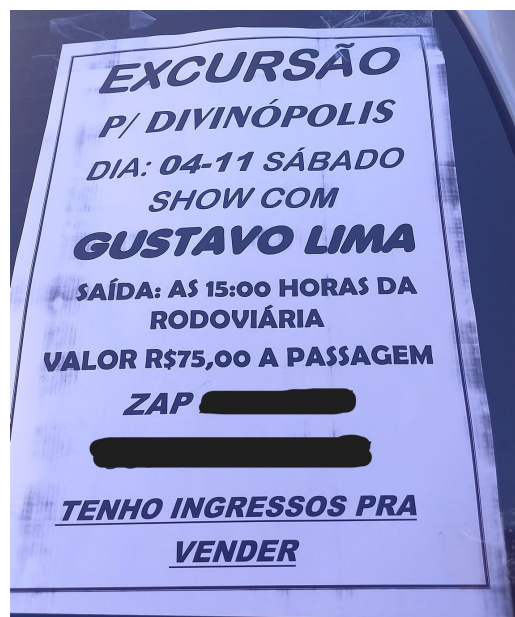


Figura 4.2: Exemplo de divulgação de excursão com cartaz. Foto tirada em outubro 2023 pelo próprio autor.

2. Controle das Excursões: O controle operacional das excursões é realizado por meio de



Figura 4.3: Exemplo de divulgação de excursão em rede social.

métodos manuais, incluindo o uso de planilhas e documentos em papel. Isso envolve o registro de participantes, gerenciamento de pagamentos e outras informações relevantes.

Já por parte dos viajantes, os desafios encontrados foram:

1. Busca Manual em Redes Sociais: Viajantes enfrentam desafios significativos ao buscar viagens, precisando recorrer a páginas no Facebook e Instagram. Essa abordagem manual pode ser demorada e inconveniente.
2. Propagandas na Rua: A dificuldade em encontrar viagens é agravada pela falta de centralização das informações, levando os viajantes a dependerem de propagandas físicas encontradas nas ruas.

Necessidades Identificadas

Centralização da Divulgação: Os agenciadores expressaram a necessidade de uma plataforma centralizada para divulgação de excursões, simplificando o processo e aumentando a visibilidade.

Automação do Controle: A transição de métodos manuais para um sistema automatizado foi destacada como uma necessidade para melhorar a eficiência e a precisão no gerenciamento de excursões.

Facilidade para Viajantes: A busca por viagens precisa ser simplificada para os viajantes, proporcionando uma experiência mais eficaz e eliminando a necessidade de buscas manuais em redes sociais.

Impactos Esperados da Aplicação

Redução de Custo e Tempo: A automação do controle de excursões visa reduzir a dependência de documentos físicos, minimizando custos e economizando tempo.

Centralização da Informação: A plataforma centralizada de divulgação busca consolidar todas as informações de excursões, proporcionando uma fonte única para agenciadores e viajantes.

Facilidade de Busca para Viajantes: A aplicação pretende oferecer aos viajantes uma maneira conveniente de encontrar e comparar excursões, eliminando a necessidade de buscas manuais e aumentando a eficiência do processo.

4.4 Planejamento

A partir das informações obtidas no levantamento de requisitos, delineou-se um planejamento preciso para a aplicação, considerando as necessidades tanto dos agenciadores quanto dos clientes. Para os agenciadores, a aplicação oferecerá uma interface dedicada ao cadastro de excursões, simplificando o processo e permitindo uma gestão eficiente de suas ofertas. Esta funcionalidade será incorporada em uma tela intuitiva, onde os agenciadores poderão fornecer detalhes abrangentes sobre cada excursão, incluindo datas, destinos, itinerários e outras informações relevantes.

Por outro lado, os clientes terão à disposição uma área na aplicação que apresentará uma lista completa de excursões disponíveis. Essa listagem será projetada para oferecer uma visão abrangente e fácil de entender das opções de viagem. Ao explorar esta seção, os clientes

poderão selecionar excursões específicas que despertem seu interesse. Após a seleção, a aplicação possibilitará a realização de reservas de forma direta e intuitiva.

A experiência do cliente será aprimorada pela facilidade de navegação e pela clareza das informações apresentadas. Além disso, a aplicação integrará um sistema de reservas eficiente, assegurando que os clientes possam confirmar sua participação nas excursões desejadas de maneira rápida e direta. As solicitações de reserva serão automaticamente encaminhadas aos agenciadores responsáveis, facilitando a comunicação entre ambas as partes.

Essa abordagem estratégica busca otimizar tanto a perspectiva dos agenciadores, ao simplificar a gestão de excursões, quanto a experiência dos clientes, ao oferecer uma plataforma amigável para descoberta e reserva de viagens.

A partir disso será criada a modelagem para facilitar a criação do banco de dados. Em seguida será desenvolvido o servidor backend utilizando Spring Boot, onde serão recebidas as requisições da aplicação o web desenvolvida logo em seguida utilizando a biblioteca React com TypeScript.

4.5 Modelagem

Na seção de modelagem, abordaremos a parte teórica e prática que serve de base para a criação do projeto. Este capítulo é crucial para entender as decisões feitas ao longo do processo, desde a definição das entidades até a arquitetura geral do sistema. Ao explorar a modelagem, será possível ver como as diferentes partes se encaixam para formar uma aplicação completa e funcional. Vamos agora analisar as escolhas de design, organização e estrutura que sustentam o projeto.

4.5.1 Caso de Uso

A seção de Modelagem destaca os casos de uso essenciais para a aplicação, delineando as interações entre os atores Cliente e Organizador/Agenciador. As figuras 4.4 e 4.5 oferecem uma visão abrangente das funcionalidades disponíveis para cada ator.

O ator Organizador/Agenciador, após realizar o login e autenticação na aplicação, terá acesso a um conjunto de operações cruciais. Estas incluem o cadastro de novas excursões, o controle detalhado das excursões existentes e a análise das reservas feitas pelos clientes. Importante ressaltar que o Organizador/Agenciador desempenha um papel fundamental na confirmação ou recusa das reservas, garantindo o gerenciamento eficiente das excursões.

Por outro lado, o ator Cliente, também autenticado na aplicação, terá a capacidade de explorar a lista de excursões disponíveis. A busca pode ser refinada por critérios como cidade de destino ou data de saída. Ao selecionar uma excursão, o Cliente pode efetuar a reserva, contribuindo para uma experiência personalizada e eficiente. Além disso, ambos os atores podem acessar e controlar seus dados cadastrais, bem como visualizar o histórico de reservas.

Esses casos de uso refletem a estrutura funcional da aplicação, garantindo uma experiência completa e integrada para ambas as partes envolvidas, clientes e organizadores/agenciadores.

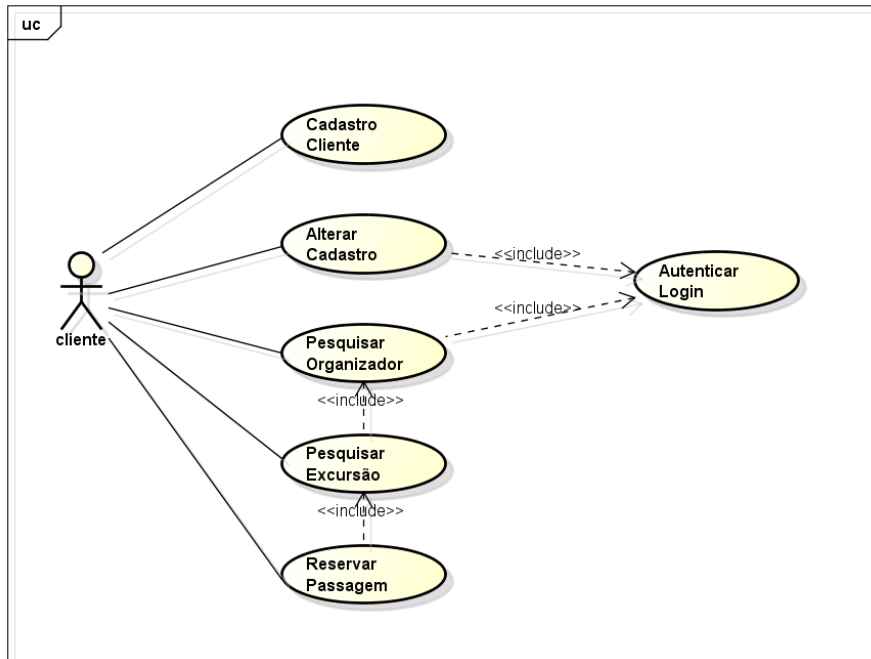


Figura 4.4: Caso de uso cliente. Fonte: Próprio Autor.

4.5.2 Diagrama Conceitual

A Figura 4.6 apresenta o diagrama conceitual, uma ferramenta crucial para definir a estrutura de dados essencial à aplicação. Este diagrama tem como objetivo identificar as entidades principais, seus atributos e relacionamentos, proporcionando uma base sólida para a elaboração subsequente do modelo conceitual e lógico do banco de dados.

Esse processo de visualização e validação é crucial para garantir que a estrutura de dados proposta seja robusta, eficiente e alinhada com os objetivos da aplicação. A clareza proporcionada pelo diagrama conceitual facilita a transição para as fases subsequentes do desenvolvimento, garantindo uma implementação coesa e orientada pelos requisitos identificados.

4.5.3 Diagrama Entidade Relacionamento

Na figura 4.7 é exibido o Diagrama Entidade Relacionamento (DER) que apresenta como foi montada a estrutura do banco de dados que serve como fonte de dados para o servidor backend que faz toda a troca de informações entre o clientes e organizadores.

A descrição do DER (Diagrama de Entidade-Relacionamento) proporciona uma visão detalhada das tabelas essenciais para a aplicação, gerenciadas pelo banco de dados MySQL. Destacando a estrutura e as relações entre as entidades, essa descrição esclarece o papel de cada tabela na funcionalidade global do sistema.

A tabela *usuario* é fundamental, contendo informações cruciais como o identificador sequencial, o email (utilizado como login), senha e tipo de usuário. A restrição de chave primária garante a unicidade do email, enquanto o tipo de usuário diferencia entre clientes e organizadores, determinando seus privilégios na aplicação.

A tabela *dadoscadastrais* complementa a informação pessoal do usuário, incluindo nome,

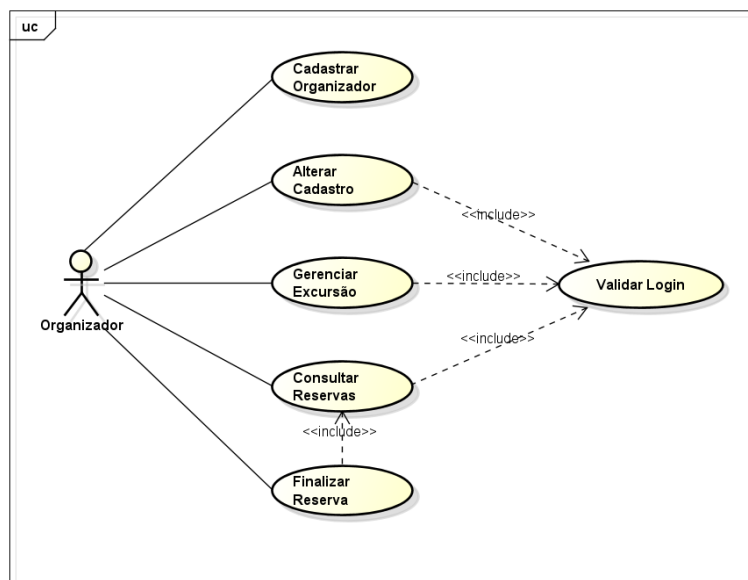


Figura 4.5: Caso de uso organizador/agenciador. Fonte: Próprio Autor.

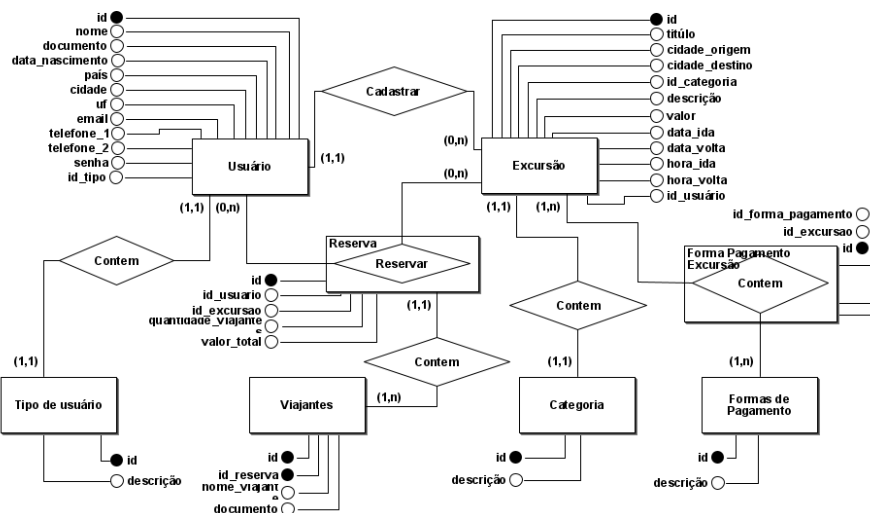


Figura 4.6: Diagrama Conceitual. Fonte: Próprio Autor.

telefone, cidade, UF, etc. A chave estrangeira *id_usuario_dadoscadastrais* estabelece a ligação com a tabela *usuario*, conectando os dados cadastrais ao usuário correspondente.

Para a tabela *excursão*, a complexidade aumenta, com a inclusão de chaves estrangeiras que referenciam o organizador, a categoria e as formas de pagamento disponíveis. A tabela *categoria* já é predefinida, simplificando a atribuição de categorias às excursões.

A tabela *reserva* reflete o processo de reserva, vinculando uma excursão e um usuário através de chaves estrangeiras, enquanto a tabela *viajantes* armazena os detalhes individuais dos viajantes associados a uma reserva.

Essa modelagem robusta oferece uma base sólida para a implementação do banco de dados, alinhando-se aos requisitos funcionais da aplicação e garantindo uma gestão eficiente de dados.

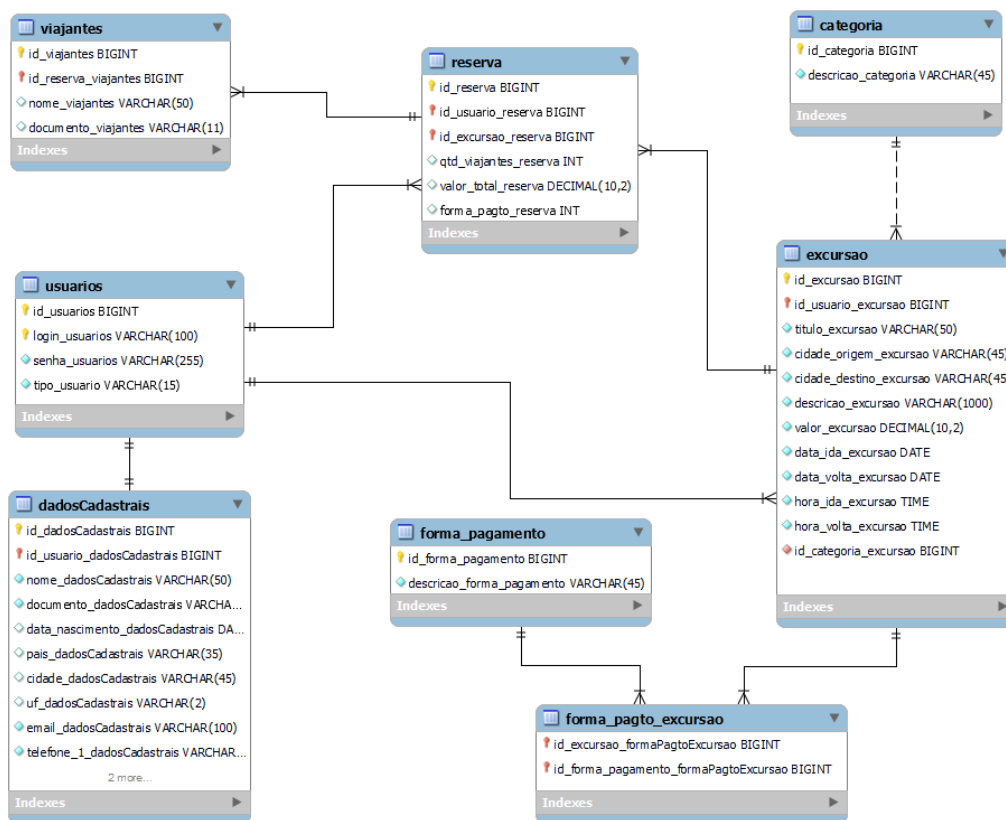


Figura 4.7: Diagrama Entidade Relacionamento. Fonte: Próprio Autor.

4.5.4 Diagrama de Classe

O diagrama de classes representado na figura 4.8 apresenta a estrutura fundamental do sistema, ilustrando as entidades essenciais e suas relações. Abaixo, segue uma descrição detalhada de cada classe no contexto da aplicação:

- **Categoria**: Representa as categorias atribuídas às excursões, proporcionando uma classificação específica para cada tipo de evento, como Lazer, Show, Concurso e Evento.
- **DadosCadastrais**: Contém informações detalhadas sobre os usuários, como nome, documento, data de nascimento, endereço e outras informações cadastrais. Esses dados estão vinculados à entidade *Usuario*, estabelecendo uma relação direta com a identidade do usuário.
- **Usuario**: Representa os usuários da aplicação, incluindo agenciadores/organizadores e clientes. Armazena informações básicas, como login, senha e tipo de usuário (diferenciando entre organizadores e clientes).
- **Excursao**: Reflete as excursões organizadas pelos agenciadores. Inclui detalhes como título, cidades de origem e destino, descrição, valor, datas e horários de ida e volta, categoria, status e outras informações relevantes.
- **FormaPagamento**: Contém as opções de formas de pagamento disponíveis para as reservas, proporcionando flexibilidade ao cliente na escolha do método de pagamento.

- **Município:** Apresenta informações sobre os municípios, incluindo nome e UF. Pode ser usado para informações geográficas associadas a excursões.
- **Reserva:** Registra as reservas feitas pelos clientes para excursões específicas. Mantém informações como quantidade de viajantes, valor total, forma de pagamento e status da reserva.
- **Viajantes:** Armazena detalhes dos viajantes associados a uma reserva, incluindo nome e documento.

O diagrama de classes fornece uma visão clara das entidades e de como estão interconectadas, sendo essencial para o entendimento da estrutura de dados subjacente à aplicação web React.

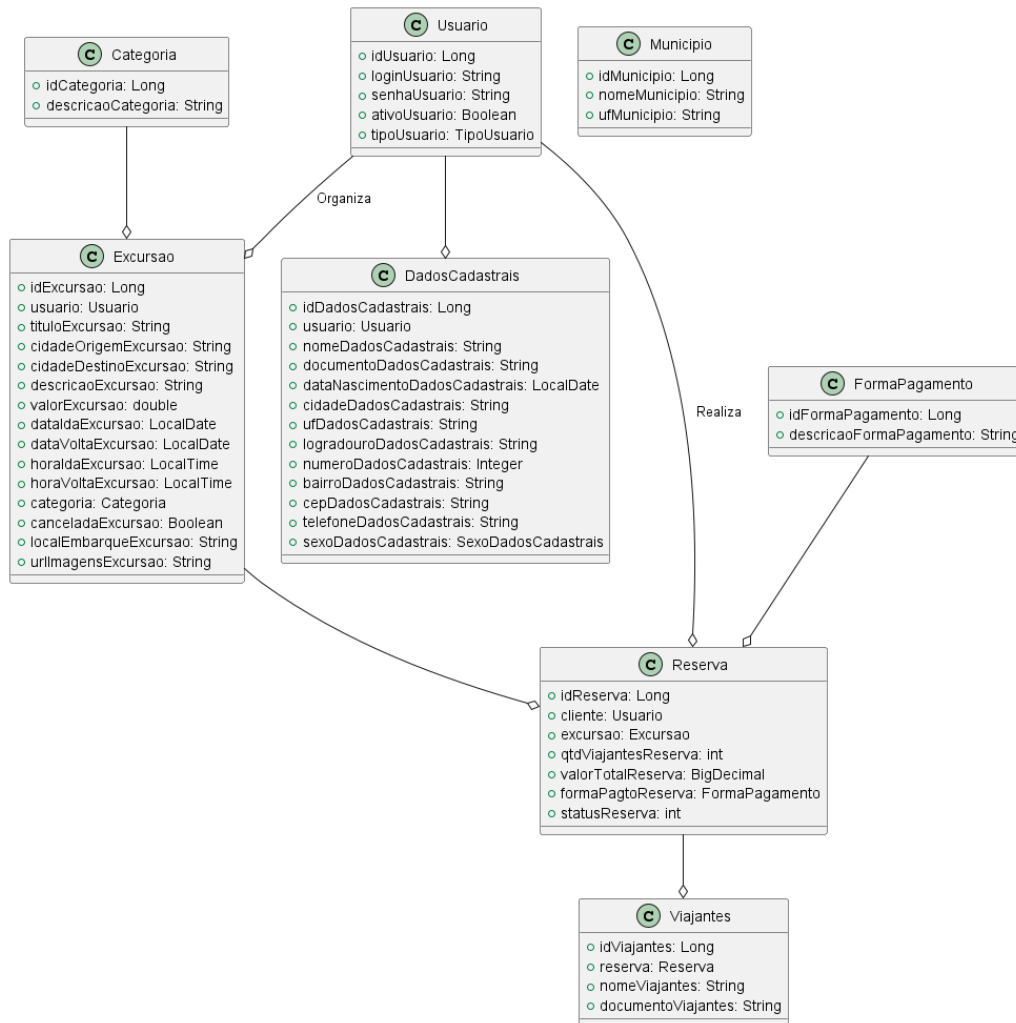


Figura 4.8: Diagrama de Classe. Fonte: Próprio Autor.

4.6 Backend Spring Boot

Na etapa seguinte à criação do banco de dados, concentramos nossos esforços na construção do backend da aplicação GoExcursão.

As entidades foram meticulosamente modeladas com base no diagrama de classes fornecido, sendo posteriormente traduzidas diretamente para classes Java, utilizando anotações JPA. Cada atributo das entidades foi cuidadosamente mapeado para a correspondente coluna no banco de dados. Para ilustrar, apresentamos um exemplo da entidade Usuario na Figura 4.9:

A criação de records se mostrou uma escolha assertiva para representar objetos simples destinados a requisições. Esses records, como o DadosUsuarioListagem ilustrado na Figura 4.10, refletem a estrutura das entidades sem incorporar a complexidade associada às regras de negócios.

Os repositórios JPA, estendendo JpaRepository, foram desenhados para proporcionar operações de consulta e manipulação no banco de dados relacionadas às entidades. O exemplo do

```

1 package tcc.goexcursao.api60Excursao.domain.usuario;
2
3 > import ...
13
14 @Table(name= "usuarios")
15 @Entity(name= "Usuario")
16 @Getter
17 @Setter
18 @NoArgsConstructor
19 @AllArgsConstructor
20 @EqualsAndHashCode(of = "idUsuario")
21 public class Usuario implements UserDetails {
22
23     @Id
24     @GeneratedValue(strategy = GenerationType.IDENTITY)
25     @Column(name = "id_usuarios")
26     private Long idUsuario;
27     @Column(name = "login_usuarios")
28     private String loginUsuario;
29     @Column(name = "senha_usuarios")
30     private String senhaUsuario;
31     @Column(name = "ativo_usuarios")
32     private Boolean ativoUsuario;
33     @Enumerated(EnumType.STRING)
34     @Column(name = "tipo_usuario")
35     private TipoUsuario tipoUsuario;
36
37     @OneToOne
38     @JoinColumn(name = "id_dadoscadastrais_usuarios")
39     private DadosCadastrais dadosCadastraisUsuario;
40

```

Figura 4.9: Exemplo de Entidade no backend. Fonte: Próprio Autor.

```

1 package tcc.goexcursao.api60Excursao.domain.usuario;
2
3 public record DadosUsuarioListagem(
4     Long idUsuario,
5     String loginUsuario,
6     String senhaUsuario,
7     Boolean ativoUsuario,
8     TipoUsuario tipoUsuario
9 ) {
10
11     @
12     public DadosUsuarioListagem(Usuario usuario){
13         this(usuario.getIdUsuario(),
14             usuario.getLoginUsuario(),
15             usuario.getSenhaUsuario(),
16             usuario.getAtivoUsuario(),
17             usuario.getTipoUsuario());
18
19     }
20
21     @
22     public DadosUsuarioListagem(DadosUsuarioListagem dadosUsuarioListagem) {
23         this(dadosUsuarioListagem.idUsuario,
24             dadosUsuarioListagem.loginUsuario,
25             dadosUsuarioListagem.senhaUsuario(),
26             dadosUsuarioListagem.ativoUsuario,
27             dadosUsuarioListagem.tipoUsuario);
28
29     }
30

```

Figura 4.10: Exemplo de Records no backend. Fonte: Próprio Autor.

```
UsuarioRepository.java x
1 package tcc.goexcursao.api60Excursao.domain.usuario;
2
3 > import ...
8
9 @Repository
10 public interface UsuarioRepository extends JpaRepository<Usuario, Long> {
11     UserDetails findByLoginUsuario(String login);
12
13     Page<DadosUsuarioListagem> findAllByAtivoUsuarioTrue(Pageable paginacao);
14
15     Page<DadosUsuarioListagem> findAllByAtivoUsuarioFalse(Pageable paginacao);
16
17     Boolean existsByLoginUsuario(String loginUsuario);
18 }
```

Figura 4.11: Exemplo de Repository estendendo JpaRepository no backend. Fonte: Próprio Autor.

UsuarioRepository, apresentado na Figura 4.11, ilustra essa abordagem.

Certos repositórios exigiram consultas personalizadas, empregando a anotação @Query. O ExcursãoRepository, evidenciado na Figura 4.12, é um exemplo dessa prática.

As classes de serviço (Service) desempenham o papel de guardiãs da lógica de negócios, interagindo de maneira coordenada com os repositórios. Estas são responsáveis por aplicar regras específicas e coordenar operações complexas. Em contrapartida, as classes de controle (Controller) recebem as requisições HTTP, invocam os métodos apropriados na classe de serviço e retornam respostas adequadas. Aqui, apresentamos exemplos de uma classe de serviço na Figura 4.13 e de controle na Figura 4.14.

4.6.1 Trabalhando com Imagens

A gestão de imagens na aplicação GoExcursão é uma funcionalidade crucial para proporcionar uma experiência envolvente aos usuários. A abordagem adotada enfatiza uma clara separação de responsabilidades entre a classe de serviço (ExcursaoService) e a classe de controle (ExcursaoController).

Upload de Imagens: Quando um usuário realiza o upload de imagens por meio da aplicação frontend, a classe de controle (ExcursaoController) recebe essas requisições HTTP. Atuando como intermediário entre a interface do usuário e a lógica de negócios, os métodos, como evidenciado na Figura 4.15, gerenciam essas requisições, acionando posteriormente os métodos correspondentes na classe de serviço (ExcursaoService), conforme ilustrado na Figura 4.16.

Lógica de Armazenamento: A classe de serviço (ExcursaoService) assume a responsabilidade pela implementação da lógica de negócios associada ao armazenamento de imagens. MultiPartFile¹ permite a recepção dos arquivos enviados no formato form_data². Posterior-

¹MultiPartFile: Interface do Spring que representa um arquivo recebido em uma solicitação multipartform-data.

²form_data: Tipo de codificação usado para upload de arquivos em formulários da web.

```

1 package tcc.goexcursao.api60Excursao.domain.excursao;
2
3 > import ...
13
4 usages  ⚡ Douglas R. Oliveira
14 public interface ExcursaoRepository extends JpaRepository<Excursao, Long> {
15     no usages  ⚡ Douglas R. Oliveira
16     Page<Excursao> findAllByCanceladaExcursaoTrue(Pageable paginacao);
17     1 usage  ⚡ Douglas R. Oliveira
18     @Query("SELECT e FROM Excursao e WHERE e.usuario = :usuario AND e.canceladaExcursao = false")
19     Page<Excursao> findAllByUsuario(Usuario usuario, Pageable paginacao);
20
21     1 usage  ⚡ Douglas R. Oliveira
22     Page<Excursao> findAllByCanceladaExcursaoFalse(Pageable paginacao);
23
24     1 usage  ⚡ Douglas R. Oliveira
25     @Transactional(readOnly = true)
26     @Query("""
27     SELECT e FROM Excursao e
28     WHERE
29     (:cidadeDestino IS NULL OR e.cidadeDestinoExcursao = :cidadeDestino)
30     AND (:dataInicio IS NULL OR e.dataIdaExcursao >= :dataInicio)
31     AND (:dataFinal IS NULL OR e.dataIdaExcursao <= :dataFinal)
32     AND e.canceladaExcursao = false
33     """)
34     Page<Excursao> buscarPorDestinoDataIda(@Param("cidadeDestino") String cidadeDestino,
35     @Param("dataInicio") LocalDate dataInicio,
36     @Param("dataFinal") LocalDate dataFinal,
37     Pageable paginacao);
38 }

```

Figura 4.12: Exemplo de Repository com @Query no backend. Fonte: Próprio Autor.

```

32 public Excursao cadastrar(DadosExcursao dadosExcursao){
33
34     var usuario = usuarioRepository.findById(dadosExcursao.getIdUsuarioExcursao()).orElse( other: null);
35     var categoria = categoriaRepository.findById(dadosExcursao.getCategoriaExcursao().getIdCategoria()).orElse( other: null);
36
37     if (usuario == null && !usuario.getAtivoUsuario()) {...}
38
39     if (categoria == null){...}
40
41     var excursao = new Excursao(dadosExcursao);
42     excursao.setUsuario(usuario);
43     excursao.setCategoria(categoria);
44     excursaoRepository.save(excursao);
45
46     excursao = excursaoRepository.getReferenceById(excursao.getIdExcursao());
47     excursao.setUrlImagensExcursao("http://localhost:60Excursosoes/Imagens/" + excursao.getIdExcursao() + "/" );
48
49     return excursao;
50 }
51
52 1 usage  ⚡ Douglas R. Oliveira
53 public void UploadImagem(MultipartFile imagem, Long id ) throws IOException {
54     String diretorioImagens = diretorioBase + id;
55     File diretorio = new File(diretorioImagens);
56     if (!diretorio.exists()) {
57         diretorio.mkdirs();
58     }
59     String filePath = diretorioImagens + "/" + imagem.getOriginalFilename();
60     try {
61         Files.write(Paths.get(filePath), imagem.getBytes());
62     } catch (IOException e) {
63         throw new RuntimeException(e);
64     }
65 }
66
67 }

```

Figura 4.13: Exemplo de Service no backend. Fonte: Próprio Autor.

```

ExcursaoRepository.java  ExcursaoController.java
no usages  Douglas R. Oliveira
51  @PostMapping
52  @Transactional
53  @ public ResponseEntity<DadosExcursaoListagem> cadastrar(@RequestBody @Valid DadosExcursao dadosExcursao, UriComponentsBuilder uriBuilder) {
54      var excursao = excursaoService.cadastrar(dadosExcursao);
55      var uri = uriBuilder.path("/{excursao/{id}").buildAndExpand(excursao.getIdExcursao()).toUri();
56      return ResponseEntity.created(uri).body(new DadosExcursaoListagem(excursao));
57  }
58
no usages  Douglas R. Oliveira
59  @PostMapping(value = "/upload/imagens/{id}", consumes = MediaType.MULTIPART_FORM_DATA_VALUE)
60  public ResponseEntity<String> uploadImagem(@RequestPart MultipartFile imagem, @PathVariable Long id) {
61      var excursao = excursaoRepository.getReferenceById(id);
62      if (excursao != null) {
63          try {
64              excursaoService.UploadImagem(imagem, excursao.getIdExcursao());
65              return ResponseEntity.ok().body("Imagem enviada com sucesso.");
66          } catch (Exception e) {
67              return ResponseEntity.status(HttpStatus.INTERNAL_SERVER_ERROR).body("Erro ao fazer o upload da imagem.");
68          }
69      }
70      return ResponseEntity.status(HttpStatus.INTERNAL_SERVER_ERROR).body("Excursao não encontrada.");
71  }
72
no usages  Douglas R. Oliveira
73  @GetMapping("/{imagens/{id}*}")
74  public ResponseEntity<Object> listaImagens(@PathVariable Long id) {
75      Excursao excursao = excursaoRepository.findById(id).orElse(null);
76      if (excursao != null) {
77          try {
78              var listaImagens = excursaoService.listaImagens(excursao);
79              System.out.println(listaImagens);
80              return ResponseEntity.ok(listaImagens);
81          } catch (Exception e) {

```

Figura 4.14: Exemplo de Controller no backend. Fonte: Próprio Autor.

```

no usages  Douglas R. Oliveira
@PostMapping(value = "/upload/imagens/{id}", consumes = MediaType.MULTIPART_FORM_DATA_VALUE)
public ResponseEntity<String> uploadImagem(@RequestPart MultipartFile imagem, @PathVariable Long id) {
    var excursao = excursaoRepository.getReferenceById(id);
    if (excursao != null) {
        try {
            excursaoService.UploadImagem(imagem, excursao.getIdExcursao());
            return ResponseEntity.ok().body("Imagem enviada com sucesso.");
        } catch (Exception e) {
            return ResponseEntity.status(HttpStatus.INTERNAL_SERVER_ERROR).body("Erro ao fazer o upload da imagem.");
        }
    }
    return ResponseEntity.status(HttpStatus.INTERNAL_SERVER_ERROR).body("Excursao não encontrada.");
}

no usages  Douglas R. Oliveira
@GetMapping("/{imagens/{id}*}")
public ResponseEntity<Object> listaImagens(@PathVariable Long id) {
    Excursao excursao = excursaoRepository.findById(id).orElse(null);
    if (excursao != null) {
        try {
            var listaImagens = excursaoService.listaImagens(excursao);
            System.out.println(listaImagens);
            return ResponseEntity.ok(listaImagens);
        } catch (Exception e) {
            return ResponseEntity.status(HttpStatus.INTERNAL_SERVER_ERROR).body("Erro ao fazer o upload da imagem.");
        }
    }
    return ResponseEntity.status(HttpStatus.NOT_FOUND).body("Excursao não encontrada.");
}

```

Figura 4.15: Métodos que controlarão as requisições de upload e listagem de imagens. Fonte: Próprio Autor.

```

1 usage  ↑ Douglas R. Oliveira
public void UploadImagem(MultipartFile imagem, Long id ) throws IOException {
    String diretorioImagens = diretorioBase + id;
    File diretorio = new File(diretorioImagens);
    if (!diretorio.exists()) {
        String filePath = diretorioImagens + "/" + imagem.getOriginalFilename();
        try {
            Files.write(Paths.get(filePath), imagem.getBytes());
        } catch (IOException e) {
            throw new RuntimeException(e);
        }
    }
}

1 usage  ↑ Douglas R. Oliveira *
public List<String> listaImagens(Excursao excursao){
    File diretorioImagens = new File( pathname: diretorioBase + excursao.getIdExcursao());
    if (!diretorioImagens.exists() || !diretorioImagens.isDirectory()) {
        return null;
    }
    File[] arquivos = diretorioImagens.listFiles();
    if (arquivos == null || arquivos.length == 0) {
        return null;
    }
    List<String> nomesImagens = Arrays.stream(arquivos) Stream<File>
        .filter(File::isFile)
        .map(file -> excursao.getUrlImagensExcursao() + file.getName()) Stream<String>
        .collect(Collectors.toList());

    return nomesImagens;
}

```

Figura 4.16: Métodos que cuidarão da lógica de armazenamento e listagem de imagens. Fonte: Próprio Autor.

mente, essas imagens são armazenadas no servidor local, seguindo um padrão de nomenclatura que incorpora o ID da excursão associada. Simultaneamente, os caminhos para essas imagens são persistidos no banco de dados, estabelecendo uma conexão eficiente entre a camada de lógica de negócios e a camada de persistência.

Listagem de Imagens: Para a exibição das imagens, a classe de controle (ExcursaoController) coordena as requisições HTTP relacionadas à listagem de imagens. Essas requisições acionam os métodos correspondentes na classe de serviço (ExcursaoService), que realiza a lógica necessária para retornar as informações relevantes. A comunicação eficiente entre as camadas permite que o frontend exiba as imagens associadas a uma excursão.

4.7 Conclusão do Backend

A implementação do backend na aplicação GoExcursão segue uma abordagem organizada e eficiente. Utilizando registros (records) para representar objetos imutáveis, a aplicação adota uma arquitetura modular com clareza na separação de responsabilidades. As entidades JPA são mapeadas diretamente para as tabelas do banco de dados, simplificando operações CRUD com os repositórios JPA.

O uso do Flyway para migração de banco de dados oferece uma solução automatizada e controlada, garantindo a consistência entre a estrutura do banco de dados e o código-fonte da aplicação. A gestão de imagens é tratada de maneira eficiente, com a separação clara de responsabilidades entre as classes de serviço e controle.

A estrutura do código favorece a manutenção e evolução da aplicação, seguindo boas práticas de desenvolvimento. A integração de ferramentas como Flyway e a clareza na definição de

interfaces entre as camadas contribuem para um sistema coeso, legível e preparado para futuras expansões. Em resumo, a implementação do backend na aplicação GoExcursão adota práticas que promovem consistência, escalabilidade e facilidade de manutenção.

Além disso, a aplicação incorpora um sólido processo de autenticação, utilizando o Spring Security e tokens JWT (JSON Web Token). O Spring Security gerencia de forma robusta a autenticação e autorização dos usuários, enquanto os tokens JWT proporcionam uma camada adicional de segurança, garantindo a transmissão segura de informações de autenticação entre o frontend e o backend. Essa combinação oferece uma solução confiável para a autenticação, proporcionando uma experiência segura e protegida para os usuários da aplicação GoExcursão.

4.8 Frontend React

A construção do frontend da aplicação GoExcursões foi realizada utilizando o editor de texto Visual Studio Code (VS Code), uma ferramenta robusta amplamente adotada pelos desenvolvedores devido à sua flexibilidade e suporte a diversas extensões que aprimoram a eficiência no desenvolvimento.

Tecnologias Utilizadas:

- **HTML:** A linguagem de marcação padrão para a criação de páginas web. HTML foi empregado na estruturação e organização do conteúdo das páginas da aplicação.
- **CSS:** Responsável pela estilização e apresentação visual das páginas web. O CSS foi utilizado para criar layouts atraentes e garantir uma experiência do usuário consistente.
- **Material-UI:** Uma biblioteca de componentes React que implementa o Material Design, fornecendo uma interface de usuário coesa e agradável. Os componentes do Material-UI facilitam o desenvolvimento de interfaces modernas e responsivas.
- **React com TypeScript:** O React é uma biblioteca JavaScript para a construção de interfaces de usuário, e o TypeScript é uma extensão desse, adicionando tipagem estática ao JavaScript. A combinação de React com TypeScript traz benefícios como autocompletar, refatoração segura e detecção de erros durante o desenvolvimento.
- **Axios:** Uma biblioteca para fazer requisições HTTP. No contexto do frontend, o Axios foi utilizado para realizar requisições à API do backend, possibilitando a comunicação eficiente entre o frontend e o backend da aplicação.

Aspectos Relevantes:

- **Estruturação das Telas:** Cada tela da aplicação GoExcursões foi estruturada usando HTML para definir os elementos da página. A organização lógica e semântica proporciona uma base sólida para o desenvolvimento e a manutenção.
- **Estilização com CSS:** O CSS foi empregado para estilizar as telas, garantindo uma apresentação visualmente atraente e uma experiência do usuário consistente. A estilização

também contribui para a responsividade das páginas, permitindo uma adaptação eficiente a diferentes tamanhos de tela.

- **Material-UI para Componentes React:** A biblioteca Material-UI simplifica a criação de componentes React que seguem as diretrizes de design do Material Design. Isso resulta em uma interface intuitiva e moderna, com componentes facilmente personalizáveis.
- **React com TypeScript para Desenvolvimento Eficiente:** A combinação do React com TypeScript oferece vantagens como tipagem estática, facilitando a detecção de erros durante o desenvolvimento. Isso resulta em um código mais robusto e menos propenso a bugs.
- **Gestão de Estado com React Context API:** A Context API do React foi utilizada para gerenciar o estado global da aplicação, permitindo o compartilhamento eficiente de dados entre componentes sem a necessidade de "props drilling" excessivo.
- **Requisições Assíncronas com Axios:** Para interagir com o backend, o Axios foi utilizado para realizar requisições assíncronas. Isso possibilita a obtenção e envio de dados entre o frontend e o backend de maneira eficiente, contribuindo para uma experiência do usuário fluida.

A escolha cuidadosa dessas tecnologias e abordagens reflete o compromisso em criar uma aplicação web moderna, eficiente e escalável. A utilização do VS Code como ambiente de desenvolvimento oferece uma experiência integrada, e as tecnologias adotadas garantem uma construção sólida do frontend da aplicação GoExcursões.

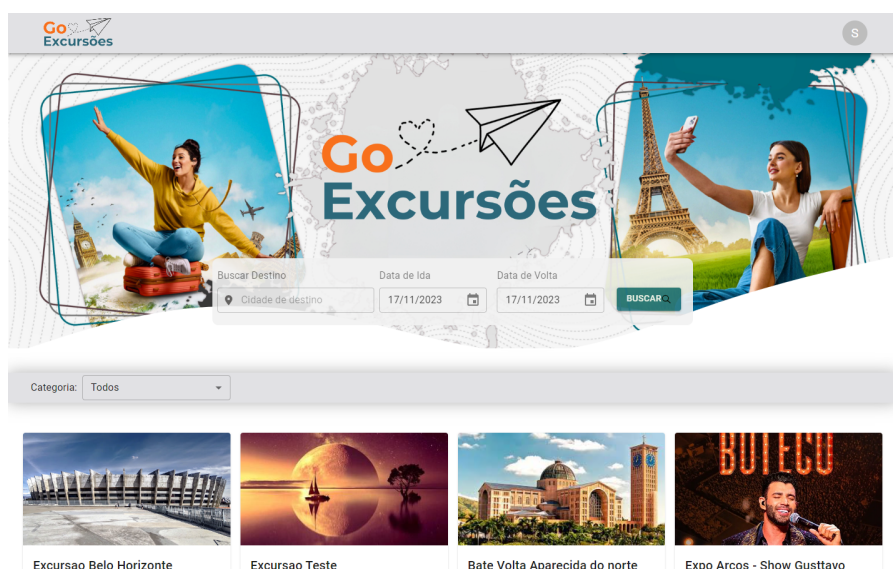


Figura 4.17: Página inicial da aplicação. Fonte: Próprio Autor.

A figura 4.17 apresenta a tela inicial da aplicação GoExcursões que foi projetada para oferecer uma experiência intuitiva aos usuários. Destaca-se pela barra superior com um logo clicável, facilitando a navegação. O botão de login/menu adapta-se à autenticação do usuário, proporcionando acesso conveniente às funcionalidades específicas. O formulário de busca avançada, com

filtros de cidade e período de data, permite uma personalização eficaz das buscas por excursões. Abaixo, uma lista atraente de excursões é apresentada, incentivando os usuários a explorar opções relevantes de forma visualmente apelativa. Esses elementos combinados proporcionam uma experiência acolhedora e eficiente na página inicial da aplicação.

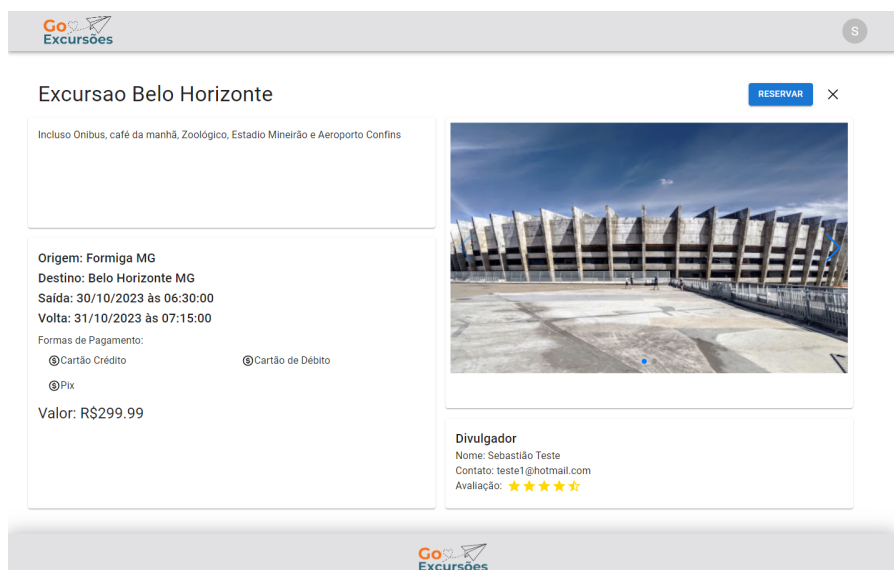


Figura 4.18: Página com detalhes da excursão. Fonte: Próprio Autor.

A figura 4.18 refere-se a página de detalhes da excursão que oferece uma visão abrangente das informações essenciais para os usuários. A descrição detalhada, acompanhada de um atraente card de imagens em formato de slide, proporciona uma imersão visual no destino da excursão. Dados cruciais, como data e horário de saída e retorno, cidades de origem e destino, formas de pagamento aceitas e o valor do pacote por viajante, são apresentados de forma organizada.

A seção dedicada ao divulgador exibe informações relevantes, como nome, contato e avaliação em estrelas, oferecendo uma referência sobre a confiabilidade do anunciante. Para realizar a reserva, o usuário pode acessar o botão designado, sendo redirecionado para a página de login se ainda não estiver autenticado. Essa abordagem intuitiva e informativa na página de detalhes visa facilitar a tomada de decisão dos usuários interessados em participar da excursão.

A funcionalidade de login na aplicação GoExcursões é acionada quando o usuário fornece os dados necessários e clica no botão "Entrar", conforme ilustrado na figura 4.19. A função correspondente denominada "efetuarLogin", utiliza o método `http.post` para enviar uma solicitação de login para o servidor backend. Os dados de login, como login e senha, são incluídos na requisição e enviados para o *endpoint*¹ "login" no backend.

Ao receber a resposta do servidor, a função de login processa o token de autenticação *JWT*² retornado em caso de autenticação bem-sucedida. Esse token é então armazenado no estado global gerenciado pelo *MobX*³, mais especificamente na classe *autenticacaoStore*. Essa aborda-

¹Endpoint: Em desenvolvimento de software, um "endpoint" refere-se a um ponto final ou URL específica em uma API (Interface de Programação de Aplicações) ou serviço web.

²JWT é um padrão aberto (RFC 7519) que define uma maneira compacta e autocontida de representar informações entre duas partes.

³MobX é uma biblioteca de gerenciamento de estado para React que oferece uma maneira simples e escalável

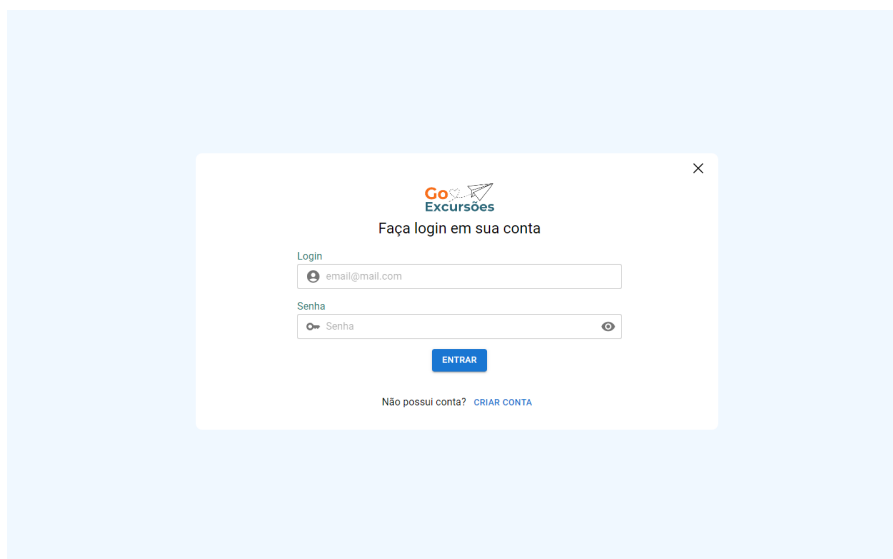


Figura 4.19: Página de login. Fonte: Próprio Autor.

gem permite que o estado de autenticação seja acessível e gerenciado de forma centralizada em toda a aplicação.

Em caso de erro de autenticação, a função exibe uma mensagem de erro para o usuário, proporcionando uma experiência informativa e orientada ao usuário. O objeto `http` mencionado na descrição é uma instância do *Axios*⁴, configurado com a URL base "`http://localhost:8080/`" para facilitar as requisições ao backend.

No backend existe um controlador Spring Boot com um método `login` que é mapeado para o endpoint `/login` e recebe os dados de autenticação no corpo da solicitação. Os dados de login são autenticados usando o `authenticationManager` e, se a autenticação for bem-sucedida, um token `JWT` é gerado usando a classe `tokenService`. As informações do usuário autenticado e o token são encapsulados em um objeto `RespostaAutenticacao` e retornados como uma resposta HTTP. O token contém informações como o ID do usuário, o login do usuário, a expiração do token, etc.

A classe `AutenticacaoStore` como mencionado anteriormente é uma implementação do MobX para gerenciar o estado relacionado à autenticação no frontend. Ela possui propriedades observáveis, como `estaAutenticado` e `usuario`, que mantêm o estado de autenticação do usuário. Os métodos `login` e `logout` são utilizados para atualizar o estado quando o usuário faz login ou logout.

A opção "Criar Conta" na tela de login permite que usuários sem cadastro realizem o processo de registro na aplicação. Ao selecionar essa opção, o usuário é redirecionado para a tela de cadastro, onde são solicitados os dados necessários para a criação da conta. A tela de cadastro é dividida em duas etapas.

Na primeira etapa (Figura 4.20), são fornecidos os dados de login, incluindo nome de usuário, senha e confirmação de senha. Essa etapa é crucial para estabelecer as credenciais de acesso

de gerenciar o estado da aplicação.

⁴Axios é uma biblioteca JavaScript que facilita o envio de solicitações HTTP para o servidor.

do usuário.

Na segunda etapa (Figura 4.21), são solicitados os dados pessoais, onde o usuário preenche informações adicionais, como nome, documento, telefone, e outros detalhes relevantes. Campos obrigatórios são destacados com "*", enfatizando a necessidade de preenchimento.

Ambas as etapas garantem a coleta completa e precisa das informações necessárias para criar a conta do usuário na aplicação. Esse processo visa simplificar a interação do usuário durante o cadastro, garantindo a segurança e a integridade dos dados fornecidos.

Figura 4.20: Página de cadastro de login. Fonte: Próprio Autor.

Figura 4.21: Página de cadastro dados pessoais. Fonte: Próprio Autor.

O processo de reserva no GoExcursão é projetado para ser simples e eficiente, composto por três etapas intuitivas:

Etapa 1: Seleção de Forma de Pagamento e Viajantes Na primeira etapa (Figura 4.22), o usuário logado seleciona a forma de pagamento desejada, além de especificar a quantidade de viajantes que participarão da excursão.

Etapa 2: Preenchimento dos Dados dos Viajantes Na segunda etapa (Figura 4.23), são fornecidos os dados detalhados de cada viajante. Para cada viajante, são solicitados nome e documento, garantindo que as informações estejam completas e precisas.

Etapa 3: Confirmação e Resumo da Reserva Na última etapa (Figura 4.24), é exibido um resumo da reserva, incluindo dados da excursão, o valor total a ser pago e uma lista com os viajantes informados. Se tudo estiver correto, o usuário pode clicar no botão de confirmação. Nesse ponto, a reserva é encaminhada para o organizador, que realizará uma análise e poderá aceitar ou recusar a reserva.

Esse processo é projetado para proporcionar uma experiência de reserva transparente e eficaz, garantindo que todas as informações necessárias sejam fornecidas de maneira organizada e facilitando a interação do usuário com a plataforma.

Go Excursões

Excursao Belo Horizonte

Incluso Ônibus, café da manhã, Z

Origem: Formiga MG
Destino: Belo Horizonte M
Saída: 30/10/2023 às 06:3
Volta: 31/10/2023 às 07:1
Formas de Pagamento:
● Cartão Crédito
○ Pix
Valor: R\$299.99

Informações da Reserva:

Excursao Belo Horizonte R\$ 1199,96

Viagem dia 30/10/2023 às 06:30:00h volta 31/10/2023 às 07:15:00h
Saindo de Formiga MG para Belo Horizonte MG

Formas de Pagamento:
 Cartão Crédito Cartão de Débito
 Pix

Divulgador
Nome: Sebastião Teste
Contato: teste1@hotmail.com
Avaliação: ★★★★★

Quantidade de Passageiros? - 4 +

RESERVAR

AVANÇAR

Figura 4.22: Etapa 1 do processo de reserva. Fonte: Próprio Autor.

Go Excursões

Excursao Belo Horizonte

Incluso Ônibus, café da manhã, Z

Origem: Formiga MG
Destino: Belo Horizonte M
Saída: 30/10/2023 às 06:3
Volta: 31/10/2023 às 07:1
Formas de Pagamento:
● Cartão Crédito
○ Pix
Valor: R\$299.99

Informações da Reserva:

Dados Viajante 1:
Nome Completo * Viagante 1 Documento * 123.456.789-10

Dados Viajante 2:
Nome Completo * Viagante 2 Documento * 879.546.123-20

Dados Viajante 3:
Nome Completo * Viagante 3 Documento * 897.654.213-44

Dados Viajante 4:
Nome Completo * Viagante 5 Documento * 654.325.415-62

RESERVAR

VOLTAR AVANÇAR

Figura 4.23: Etapa 2 do processo de reserva. Fonte: Próprio Autor.

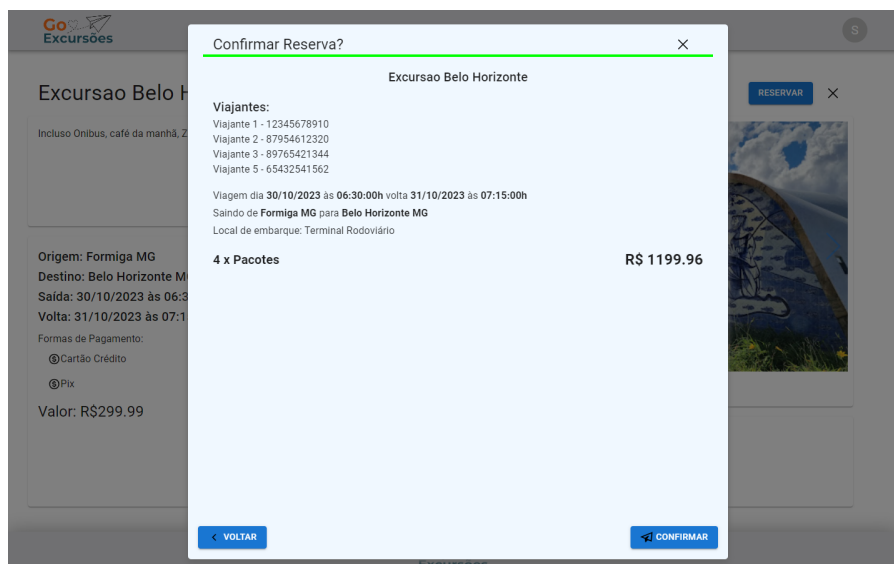


Figura 4.24: Etapa 3 do processo de reserva. Fonte: Próprio Autor.

4.8.1 Dashboard Usuário

O GoExcursão oferece aos usuários um dashboard abrangente com um menu lateral, proporcionando acesso fácil a diversas funcionalidades:

Dados Cadastrais do Usuário (Figura 4.25): Os usuários podem visualizar e, se necessário, alterar seus dados cadastrais. Essa opção fornece uma maneira conveniente de manter informações pessoais atualizadas.

Listagem de Reservas (Figura 4.26): Nesta seção, os usuários têm uma visão consolidada de todas as reservas efetuadas. Cada reserva é apresentada com alguns dados essenciais, incluindo o status atual. Um botão dedicado permite que o usuário acesse os detalhes completos de uma reserva específica.

Detalhes de Reserva (Figura 4.27): Ao selecionar uma reserva, os usuários têm acesso a informações detalhadas, como dados da excursão, valor total, lista de viajantes e status atual da reserva. Essa visão detalhada oferece transparência sobre o andamento da reserva.

Esse design de dashboard fornece uma experiência intuitiva e centralizada para que os usuários gerenciem seus dados e tenham controle total sobre suas reservas, contribuindo para uma interação eficiente com a plataforma GoExcursão.

4.8.2 Dashboard Administrativo

Os organizadores do GoExcursão contam com um dashboard administrativo que oferece funcionalidades específicas para a gestão eficiente das excursões e reservas:

Cadastro e Edição de Excursões (Figura 4.28): Os organizadores podem cadastrar novas excursões ou editar informações de excursões existentes. Essa página fornece um formulário abrangente para incluir todos os detalhes essenciais sobre uma excursão.

Listagem de Excursões (Figura 4.29): Uma visão geral de todas as excursões cadastradas, proporcionando uma visão consolidada para os organizadores. A partir desta lista, é possível acessar opções de edição e visualização detalhada.

The screenshot shows a web interface for 'Go Excursões' with a sidebar containing 'Dados Cadastrais', 'Reservas', and 'Home'. The main content area is titled 'Dados da Conta' and contains a form for 'Dados Cadastrais'. The form fields are as follows:

- Login: teste1@hotmail.com
- Senha: [input field] with an 'ALTERAR SENHA' link and a 'Tipo Usuário' dropdown.
- Nome Completo: Sebastião Teste
- Documento: 123.456.789-10
- Data Nascimento: 29/06/1995
- Cidade: Formiga
- UF: MG
- Logradouro: Rua Teste
- Numero: 21
- Bairro: Centro
- CEP: 12345-678
- Telefone: (37) 9 9988-5544
- Sexo: MASCULINO

A 'GRAVAR' button is located at the bottom of the form.

Figura 4.25: Dados cadastrais do usuario. Fonte: Próprio Autor.

The screenshot shows the same 'Dados da Conta' page, but the main content area displays a table titled 'Reservas'. The table has the following columns: 'Excursão', 'Destino', 'Quantidade Viajantes', 'Valor Total', 'Forma de Pagamento', 'Status', and 'Detalhes'. There is one reservation listed:

Excursão	Destino	Quantidade Viajantes	Valor Total	Forma de Pagamento	Status	Detalhes
Excursao Belo Horizonte	Belo Horizonte MG	4	R\$1199,96	Cartão Crédito	Solicitado	[icon]

Below the table is a pagination control showing '< 1 >'.

Figura 4.26: Listagem de reservas do usuario. Fonte: Próprio Autor.

Listagem de Reservas de uma Excursão (Figura 4.30): Ao selecionar uma excursão específica, os organizadores têm acesso à lista de reservas associadas a essa excursão. Isso inclui informações essenciais sobre cada reserva e fornece opções para edição e visualização detalhada.

Detalhes da Reserva (Figura 4.31): Os detalhes completos de uma reserva específica, incluindo informações da excursão, valor total, lista de viajantes e status atual da reserva. Os organizadores têm a capacidade de aceitar ou recusar uma reserva a partir desta visualização detalhada.

Esse dashboard administrativo proporciona aos organizadores todas as ferramentas necessárias para gerenciar efetivamente as excursões e as solicitações de reserva, contribuindo para uma administração intuitiva e eficiente na plataforma GoExcursão.

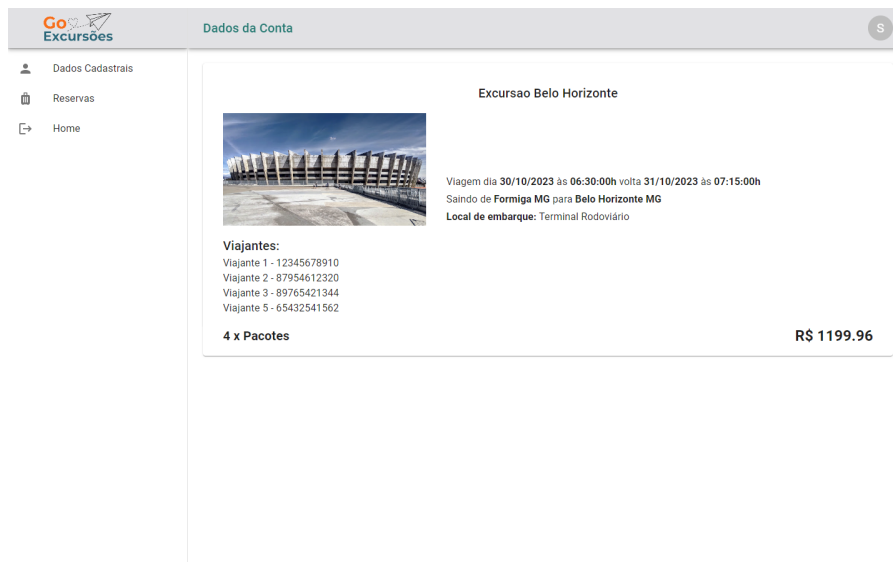


Figura 4.27: Detalhes reserva do usuario. Fonte: Próprio Autor.

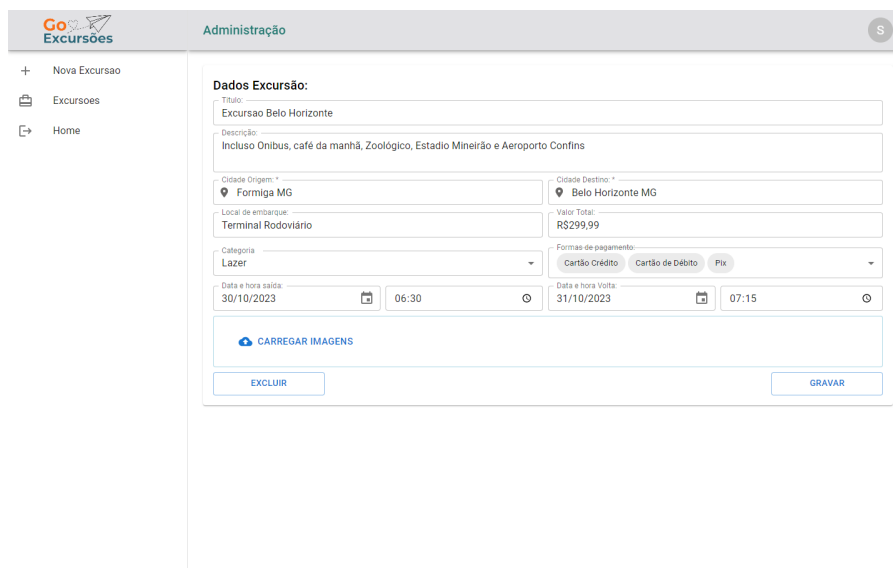


Figura 4.28: Página de cadastro e edição de excursões. Fonte: Próprio Autor.

4.9 Considerações finais

O capítulo de metodologia traçou o desenvolvimento integral da aplicação GoExcursão, desde a concepção do banco de dados até a implementação do frontend e backend. A metodologia adotada priorizou a clareza e eficiência, destacando a modularidade e a separação de responsabilidades no código.

The screenshot shows the 'Go Excursões' administration interface. The header includes the logo and the word 'Administração'. A sidebar on the left contains navigation options: '+ Nova Excursao', 'Excursoes', and 'Home'. The main content area displays a table with the following data:

Título	Origem	Destino	Período	Valor Total	Editar	Reservas
Excursao Belo Horizonte	Formiga MG	Belo Horizonte MG	30/10/2023 à 31/10/2023	R\$299.99		
Excursao Teste	Divinópolis MG	Belo Horizonte MG	12/11/2023 à 13/11/2023	R\$299.99		
Expo Arcos - Show Gustavo Lima	Córrego Fundo MG	Arcos MG	17/11/2023 à 18/11/2023	R\$180.00		
Expo Arcos - Show Ana Castela	Córrego Fundo MG	Arcos MG	17/11/2023 à 18/11/2023	R\$180.00		

At the bottom of the table, there is a pagination control showing '< 1 >'.

Figura 4.29: Listagem de excursões cadastradas. Fonte: Próprio Autor.

The screenshot shows the 'Go Excursões' administration interface, similar to the previous one. The sidebar and header are identical. The main content area displays a table with the following data:

Título	Origem	Destino	Período	Valor Total	Editar	Reservas
Excursao Belo Horizonte	Formiga MG	Belo Horizonte MG	30/10/2023 à 31/10/2023	R\$299.99		
Excursao Teste	Divinópolis MG	Belo Horizonte MG	12/11/2023 à 13/11/2023	R\$299.99		
Expo Arcos - Show Gustavo Lima	Córrego Fundo MG	Arcos MG	17/11/2023 à 18/11/2023	R\$180.00		
Expo Arcos - Show Ana Castela	Córrego Fundo MG	Arcos MG	17/11/2023 à 18/11/2023	R\$180.00		

At the bottom of the table, there is a pagination control showing '< 1 >'.

Figura 4.30: Listagem de reservas da excursão selecionada. Fonte: Próprio Autor.

GO Excursões


Administração

+ Nova Excursao

Excursoes

Home

Excursao Belo Horizonte



Viagem dia 30/10/2023 às 06:30:00h volta 31/10/2023 às 07:15:00h
Saindo de Formiga MG para Belo Horizonte MG
Local de embarque: Terminal Rodoviário

Viajantes:
Viajante 1 - 12345678910
Viajante 2 - 87954612320
Viajante 3 - 89765421344
Viajante 5 - 65432541562

4 x Pacotes **R\$ 1199.96**

Figura 4.31: Detalhes reserva. Fonte: Próprio Autor.

Testes e Resultados

Os testes foram conduzidos em quatro dispositivos distintos: um notebook dell processador core i5, 16gb de ram tela de 15,6" , dois celulares, um galaxy s20 fe com tela de 6,3" com sistema operacional android 11 e outro galaxy s23 Ultra com tela de 6,8" com android versão 13 e uma televisão lg 50" para testar a responsividade em telas grandes, todos conectados a um servidor local por meio de uma rede sem fio. Durante os testes, foram realizadas requisições simultâneas para avaliar se todas eram atendidas corretamente. Além disso, foi verificada a responsividade dos componentes em diferentes tamanhos de tela.

O acompanhamento dos logs permitiu verificar se todas as ações estavam sendo executadas conforme o esperado. Os testes visaram garantir não apenas o correto funcionamento das funcionalidades, mas também a eficiência e a adaptabilidade da aplicação a diferentes dispositivos.

Os resultados obtidos demonstraram um desempenho satisfatório, com todas as requisições sendo atendidas adequadamente e a responsividade dos componentes sendo efetiva em todos os dispositivos testados. Os logs confirmaram o correto funcionamento das ações, validando a estabilidade e consistência da aplicação durante os testes.

Essa fase de testes e resultados foi crucial para assegurar a qualidade da aplicação GoExcursão, garantindo uma experiência positiva para os usuários em diversos dispositivos e cenários de uso.

Conclusões

6.1 Considerações Finais

As considerações finais deste projeto refletem o esforço conjunto na criação da aplicação Go-Excursão, buscando fornecer aos usuários uma experiência completa e satisfatória. Ao longo do desenvolvimento, foram adotadas práticas de engenharia de software que priorizam a eficiência, modularidade e escalabilidade.

A arquitetura modular do backend, utilizando tecnologias como Spring Boot, JPA, e ferramentas como Flyway, proporciona uma base sólida para a gestão de excursões e usuários. A implementação do frontend, utilizando React com TypeScript, Material-UI, Axios, e gerenciamento de estado com MobX, contribui para uma interface responsiva e de fácil navegação.

Destaca-se também a segurança implementada no sistema, com o uso do Spring Security e tokens JWT para autenticação, assegurando a proteção das informações transmitidas entre o frontend e o backend.

Os testes realizados, abrangendo diferentes dispositivos e cenários, corroboraram a eficácia da aplicação em atender às demandas dos usuários, validando a robustez e confiabilidade do sistema.

No contexto do projeto, a gestão de imagens, a criação de contas, a realização de reservas e a administração de excursões foram implementadas de forma coesa, promovendo uma experiência completa para os usuários finais.

Considerando o ciclo de desenvolvimento, desde o levantamento de requisitos até a fase de testes, foi possível observar a evolução e a maturidade do projeto. As metodologias ágeis adotadas, como o Scrum, contribuíram para uma gestão eficiente e flexível do desenvolvimento.

Em suma, as conclusões deste projeto destacam a realização de um trabalho sólido, alinhado com as expectativas iniciais. A aplicação GoExcursão está pronta para oferecer aos usuários uma

plataforma intuitiva, segura e funcional para a gestão de excursões, conectando organizadores e viajantes de maneira eficiente e agradável.

6.2 Contribuições

As contribuições deste projeto abrangem diversas áreas, refletindo a aplicação prática de conhecimentos em desenvolvimento de software e engenharia de sistemas. Algumas das principais contribuições incluem:

1. **Experiência do Usuário Aprimorada:** A criação da aplicação GoExcursão visa proporcionar uma experiência de usuário completa e satisfatória. Desde a navegação intuitiva até a gestão eficiente de reservas, o projeto concentra-se em atender às necessidades dos usuários finais, oferecendo uma plataforma amigável e funcional.
2. **Arquitetura Modular e Eficiente:** O design do backend, utilizando tecnologias como Spring Boot, JPA, e ferramentas como Flyway, demonstra uma abordagem modular que favorece a manutenção, escalabilidade e evolução do sistema. A implementação do frontend em React com TypeScript e MobX contribui para uma interface responsiva e dinâmica.
3. **Segurança na Autenticação:** A aplicação adota práticas robustas de segurança, incorporando o Spring Security e tokens JWT para autenticação. Essa camada de segurança garante a integridade e confidencialidade das informações transmitidas entre o frontend e o backend, protegendo os dados sensíveis dos usuários.
4. **Gestão Eficiente de Imagens:** A funcionalidade de gestão de imagens foi integrada de maneira eficiente, permitindo que os usuários associem imagens às excursões. Isso contribui para uma experiência visual rica e atrativa na plataforma.
5. **Processo de Reserva Simplificado:** O sistema oferece um processo de reserva simplificado em três etapas, considerando a seleção de pagamento, o preenchimento de dados dos viajantes e a revisão final. Essa abordagem facilita a interação dos usuários, tornando o processo mais acessível e compreensível.
6. **Testes Abrangentes:** Foram realizados testes em diferentes dispositivos e cenários, verificando a robustez da aplicação e sua responsividade em várias telas. Esse processo contribuiu para a identificação e correção de possíveis problemas, garantindo a qualidade do produto final.
7. **Administração de Excursões e Reservas:** Para usuários organizadores, a aplicação oferece um dashboard administrativo que permite o cadastro, edição e visualização de excursões, bem como a gestão eficiente de reservas associadas a cada excursão.
8. **Metodologias Ágeis:** A adoção de metodologias ágeis, como o Scrum, contribuiu para a gestão eficiente do desenvolvimento, permitindo adaptações rápidas a mudanças nos requisitos e promovendo uma abordagem colaborativa e flexível.

Em resumo, as contribuições deste projeto vão além da simples implementação de uma aplicação, abrangendo a entrega de uma solução completa que atende às demandas dos usuários finais, considerando aspectos de usabilidade, segurança e eficiência.

6.3 Dificuldades Encontradas

Durante o desenvolvimento do projeto GoExcursão, algumas dificuldades foram encontradas, mas foram superadas com esforço e resiliência. Algumas das principais dificuldades incluíram:

- **Integração Frontend e Backend:** Garantir uma integração suave entre o frontend e o backend pode ser desafiador, especialmente ao lidar com diferentes tecnologias e frameworks. Foram necessários esforços para garantir que as requisições HTTP, a autenticação e a comunicação entre as camadas fossem eficientes e sem falhas.
- **Gestão de Imagens:** Implementar uma solução eficiente para a gestão de imagens, incluindo o upload, armazenamento e exibição, pode ser complexo. A necessidade de equilibrar o desempenho da aplicação com o armazenamento eficiente de dados e imagens exigiu uma cuidadosa consideração da arquitetura.
- **Autenticação e Segurança:** Garantir a segurança adequada da aplicação, especialmente no que diz respeito à autenticação e autorização, é fundamental. A implementação do Spring Security e tokens JWT trouxe desafios específicos que demandaram uma compreensão profunda dessas tecnologias.
- **Responsividade:** Tornar a aplicação responsiva para diferentes tamanhos de tela e dispositivos pode ser um desafio, especialmente quando se lida com uma variedade de dispositivos, desde notebooks até smartphones e televisões. Foi necessário ajustar o layout e os componentes para garantir uma experiência consistente em todas as plataformas.
- **Testes Simultâneos:** Testar a aplicação em condições simuladas de uso intenso, com requisições simultâneas e em dispositivos diferentes, apresentou desafios específicos para garantir a robustez do sistema em ambientes dinâmicos.
- **Implementação do Backend:** Embora o Spring Boot e as ferramentas associadas ofereçam uma estrutura poderosa, sua configuração inicial e personalização específica para atender aos requisitos do projeto exigiram uma curva de aprendizado.

A superação dessas dificuldades contribuiu para o desenvolvimento de habilidades técnicas, tomada de decisões eficientes e uma compreensão mais aprofundada das tecnologias empregadas.

6.4 Trabalhos Futuros

- **Implementação de Função de Pagamento:** Adicionar uma função de pagamento integrada à aplicação, permitindo que os usuários efetuem transações financeiras de forma segura. Isso pode envolver a integração com serviços de pagamento online e a implementação de gateways de pagamento.

- Notificações de Reservas Aceitas: Implementar um sistema de notificações para usuários, informando sobre a aceitação de suas reservas. Isso pode ser realizado por meio de notificações push no aplicativo ou por e-mail, garantindo que os usuários estejam atualizados sobre o status de suas solicitações.
- Notificações para Organizadores: Adicionar funcionalidades de notificação para os organizadores, alertando sobre novas solicitações de reserva, atualizações no status das reservas ou outras informações relevantes. Isso pode melhorar a eficiência na gestão das excursões.
- Melhorias na Segurança: Continuar aprimorando as camadas de segurança da aplicação, garantindo que as melhores práticas sejam seguidas. Isso pode incluir atualizações regulares de dependências, análises de segurança e implementação de novas medidas de proteção.
- Análise de Desempenho: Realizar análises de desempenho mais detalhadas para identificar áreas de otimização na aplicação. Isso pode envolver a otimização de consultas ao banco de dados, o cache eficiente de dados e a melhoria geral da velocidade e responsividade da aplicação.
- Internacionalização: Se houver planos de expansão para mercados internacionais, considerar a implementação de recursos de internacionalização para suportar diferentes idiomas e regiões.
- Refatoração e Manutenção: Continuar a refatoração do código conforme necessário para manter a base de código limpa e aderente às melhores práticas de desenvolvimento. Isso facilitará futuras expansões e manutenções.
- Feedback dos Usuários: Coletar feedback dos usuários atuais e potenciais para identificar áreas de melhoria na usabilidade, design e funcionalidades da aplicação.

Ao abordar esses pontos, o projeto GoExcursão pode evoluir e oferecer uma experiência ainda mais robusta e satisfatória para seus usuários.

Referências Bibliográficas

ALVES, C.; BACH, J. **TypeScript para iniciantes: Aprenda TypeScript passo a passo**. [S.l.]: Independently Published, 2021. ISBN 9798529291153.

ARAGÃO, H. **JAVA E PROGRAMAÇÃO ORIENTADA A OBJETOS: uma abordagem didática: Java, Programação Orientada a Objetos, Programação, Orientação a Objetos**. [S.l.]: Helder Guimarães Aragão, 2013. ISBN 9788591355815.

CHOI, D. **Full-Stack React, TypeScript, and Node: Build cloud-ready web applications using React 17 with Hooks and GraphQL**. [S.l.]: Packt Publishing, 2020. ISBN 9781839214691.

DUBOIS, P. **MySQL**. [S.l.]: Addison-Wesley, 2013. (Developer's library). ISBN 9780321833877.

FLANAGAN, D. **JavaScript: O Guia Definitivo**. [S.l.]: Bookman Editora, 2012. ISBN 9788565837484.

GOLDBERG, J. **Aprendendo TypeScript: Melhore suas habilidades de desenvolvimento web usando JavaScript Type-Safe**. [S.l.]: Novatec Editora, 2022. ISBN 9788575228302.

GRYNHAUS, B. et al. **The TypeScript Workshop: A practical guide to confident, effective TypeScript programming**. [S.l.]: Packt Publishing, 2021. ISBN 9781838826765.

ICASSATTI, M. **Como a tecnologia vem tornando nossas viagens melhores — e menos burocráticas**. [S.l.], 2023. Data de acesso: 12/11/2023. Disponível em: <https://exame.com/casual/como-a-tecnologia-vem-tornando-nossas-viagens-melhores-e-menos-burocraticas/>.

INDEX, T. **TIOBE Index for November 2023**. [S.l.], 2023. Data de acesso: 15/11/2023. Disponível em: <https://www.tiobe.com/tiobe-index/>.

LECHETA, R. **Web services RESTful: Aprenda a criar web services RESTful em Java na nuvem do Google**. [S.l.]: Novatec Editora, 2015. ISBN 9788575224540.

LLC, T. **Sobre o TripAdvisor**. [S.l.], 2023. Data de acesso: 12/11/2023. Disponível em: <https://tripadvisor.mediaroom.com/US-about-us>.

LUCKOW, D.; MELO, A. de. **Programação Java para a Web**. [S.l.]: Novatec Editora, 2010. ISBN 9788575222386.

MACIEL, V. **Turismo será responsável por quase 8 milhões de empregos e 7,8% do PIB do Brasil em 2023, afirma WTTC**. [S.l.], 2023. Data de acesso: 12/11/2023. Disponível em: <https://encurtador.com.br/nwANV>.

NERY, C.; BRITTO, V. **Internet já é acessível em 90% dos domicílios do país em 2021**. [S.l.], 2022. Data de acesso: 12/11/2023. Disponível em: <https://encurtador.com.br/dhN37>.

ORACLE. **O que é um Banco de Dados?** [S.l.]. Data de acesso: 15/11/2023. Disponível em: <https://encurtador.com.br/aclxR>.

PEREIRA, C. **Aplicações web real-time com Node.js**. [S.l.]: Casa do Código, 2014. ISBN 9788566250930.

PRESCOTT, P.; SOUZA, F. **Programação em JavaScript**. [S.l.]: Babelcube Incorporated, 2016. ISBN 9781507129685.

RIBAS, T. **Scrum: O que É, Como Funciona e Exemplos Práticos [GUIA]**. [S.l.]. Data de acesso: 17/11/2023. Disponível em: <https://thomazribas.com/agile/scrum>.

RICHARDSON, L.; RUBY, S. **RESTful Web Services**. [S.l.]: O'Reilly Media, 2008. ISBN 9780596554606.

RIGON, R. **Registro foi feito na esquina das ruas Os Dezoito do Forte e Marquês do Herval**. [S.l.], 2013. Data de acesso: 15/11/2023. Disponível em: <https://encurtador.com.br/axAD6>.

S.A., H. T. **Sobre a Hurb**. [S.l.], 2023. Data de acesso: 12/11/2023. Disponível em: <https://institucional.hurb.com/nossa-historia>.

SANTANA, E. **Back-end Java: Microsserviços, Spring Boot e Kubernetes**. [S.l.]: Casa do Código, 2021. ISBN 9786586110623.

SEBRAE. **Travel techs, o que são e como podem ajudar os negócios**. [S.l.], 2023. Data de acesso: 14/11/2023. Disponível em: <https://encurtador.com.br/cnH07>.

SILVA, M. **Fundamentos de HTML5 e CSS3**. [S.l.]: Novatec Editora, 2018. ISBN 9788575227084.

SOUTO, M. **Front-end, Back-end e Full Stack**. [S.l.], 2023. Data de acesso: 15/11/2023. Disponível em: <https://www.alura.com.br/artigos/o-que-e-front-end-e-back-end>.

STEFANOV, S. **React: Up & Running**. [S.l.]: O'Reilly Media, 2021. ISBN 9781492051411.

SUEHRING, S. **MYSQL - A BÍBLIA**. [S.l.]: CAMPUS, 2002. ISBN 9788535210842.

SUTHERLAND, J. **SCRUM: A arte de fazer o dobro de trabalho na metade do tempo**. [S.l.]: LEYA BRASIL, 2014. ISBN 9788544100882.

TOTVS, E. **O que é back-end e qual seu papel na programação?** [S.l.], 2020. Data de acesso: 15/11/2023. Disponível em: <https://www.totvs.com/blog/developers/back-end/>.