

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE CIÊNCIA DA COMPUTAÇÃO

HENRIQUE DE SOUZA LA PORTA

**Planejando Viagens: Desenvolvimento do
sistema Web para planejamento de viagens**

Monografia apresentada como requisito parcial
para a obtenção do grau de Bacharel em Ciência
da Computação

Orientador: Prof. Dr. Marcelo Soares Pimenta

Porto Alegre
2018

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. Rui Vicente Oppermann

Vice-Reitora: Prof^a. Jane Fraga Tutikian

Pró-Reitor de Graduação: Prof. Wladimir Pinheiro do Nascimento

Diretora do Instituto de Informática: Prof^a. Carla Maria Dal Sasso Freitas

Coordenador do Curso de Ciência de Computação: Prof. Sérgio Luis Cechin

Bibliotecária-chefe do Instituto de Informática: Beatriz Regina Bastos Haro

RESUMO

Este trabalho tem como objetivo descrever a concepção e a implementação de uma aplicação Web de planejamento para uso dos funcionários e dos visitantes do blog Planejando Viagens. Esta aplicação visa facilitar o gerenciamento de informações e possibilitar a consulta dos dados de forma prática antes, durante e após a viagem, podendo também tornar-se uma ferramenta de controle e organização de gastos durante a viagem. São apresentadas no texto as características principais das tecnologias adotadas e de como foram empregadas no desenvolvimento, uma descrição da aplicação desenvolvida e suas funcionalidades, e de sua avaliação.

Palavras-chave: Planejamento de Viagens. Vue. Node.js.

ABSTRACT

This paper aims to describe the design and implementation of a planning Web application for the use of employees and visitors of the blog Planejando Viagens. This application aims to facilitate the information management and make it possible to consult the data in a practical way before, during and after the trip, and can also become a tool to control and organize expenses during the trip. In the text are presented the main characteristics of the adopted technologies and how they were used in the development, a description of the application developed and its functionalities, and their evaluation.

Keywords: Travel Planning. Vue. Node.js.

LISTA DE FIGURAS

Figura 3.1	Interface do sistema Inspirock	16
Figura 3.2	Interface do sistema Travefy	17
Figura 3.3	Interface do sistema Google Trip.....	18
Figura 3.4	Interface do sistema Travaa	19
Figura 3.5	Interface do sistema Trip30	20
Figura 3.6	Interface do sistema TripIt.....	21
Figura 4.1	Visualização de uma viagem na sua tela principal	24
Figura 4.2	Protótipo da tela de adicionar evento ao roteiro	24
Figura 4.3	Diagrama de Entidade-Relacionamento	27
Figura 4.4	Interface do Strapi.....	29
Figura 4.5	Trecho de código de uso da biblioteca Strapi	29
Figura 4.6	Console de APIs do Google.....	30
Figura 4.7	Trecho de código de uso do pacote vue2-google-maps	31
Figura 4.8	Exemplo de comportamento do componente gmap-autocomplete.....	31
Figura 4.9	Trecho de código de uso do pacote vue2-google-maps.....	32
Figura 5.1	Tela inicial da aplicação.....	33
Figura 5.2	Tela de cadastro da aplicação	34
Figura 5.3	Tela de autenticação da aplicação.....	34
Figura 5.4	Tela do painel de gerenciamento de viagens	35
Figura 5.5	Tela para criar uma viagem.....	35
Figura 5.6	Tela de visualização do roteiro como timeline	36
Figura 5.7	Tela para adicionar itens ao roteiro.....	36
Figura 5.8	Tela de visualização dos gastos	37

LISTA DE TABELAS

Tabela 3.1	Tabela de análise comparativa	22
Tabela 6.1	Tabela de análise das heurísticas de nielsen	38
Tabela 6.2	Tabela do plano de coleta de dados	40
Tabela 7.1	Tabela de análise comparativa completa	46

LISTA DE ABREVIATURAS E SIGLAS

API Application Programming Interface

JWT Javascript Web Token.

HTML HyperText Markup Language

CSS Cascading Style Sheets

JS JavaScript

JSON JavaScript Object Notation

NPM Node Package Manager

SUMÁRIO

1 INTRODUÇÃO	9
1.1 Objetivos	9
1.2 Estrutura.....	10
2 FUNDAMENTOS	11
2.1 Metodologia Ágil	11
2.2 Protótipos Adobe XD	11
2.3 JavaScript	12
2.4 Node.js.....	12
2.5 Vue	12
2.6 Nuxt	13
2.7 Vuetify	13
2.8 MongoDB.....	13
2.9 Heurísticas de Nielsen.....	14
3 TRABALHOS RELACIONADOS	16
3.1 Inspirock	16
3.2 Travefy	17
3.3 Google Trips	18
3.4 Travaa	19
3.5 Trip30	19
3.6 Tripit.....	20
3.7 Análise comparativa	21
4 DESENVOLVIMENTO	23
4.1 Requisitos e objetivos.....	23
4.2 Protótipos de baixa fidelidade.....	24
4.3 Entidades	25
4.4 Aplicação Web	28
4.5 Banco de Dados	28
4.6 Mapas e endereços	29
5 A APLICAÇÃO	33
5.1 Autenticação e cadastro.....	33
5.2 Como criar uma viagem	34
5.3 Como adicionar informações ao roteiro.....	35
5.4 Visualizar os gastos	36
6 ANÁLISE DA APLICAÇÃO	38
6.1 Avaliação Heurística	38
6.2 Avaliação de aplicação através de entrevista	39
6.3 Apresentação e análise dos dados coletados	40
7 CONCLUSÕES	43
7.1 Resumo de resultados	43
7.2 Resumo de resultados referentes as tecnologias utilizadas	44
7.3 Limitações do trabalho	45
7.4 Trabalhos futuros	45
REFERÊNCIAS	47
APÊNDICE A - ROTEIROS DE ENTREVISTA	48

1 INTRODUÇÃO

Atualmente o planejamento de viagens por conta própria, sem necessidade da intervenção de agências de viagens, tem se tornado cada vez mais popular e acessível devido a grande quantidade de informações disponíveis na internet. Porém, para alguns viajantes surge a dificuldade de organizar esse grande volume de dados e os possíveis custos envolvidos, principalmente quando as viagens são longas, incluem mais de um destino e envolvem mais de uma pessoa. Além disso, outra dificuldade é armazenar as diversas informações que serão necessárias durante a viagem de forma prática e acessível.

Dentro desse contexto existe o blog Planejando Viagens, que tem como objetivo ajudar outros viajantes a planejar viagens, através da disponibilização de informações e dicas sobre diversos lugares do mundo. Atualmente os autores do blog utilizam tabelas de Excel como a principal forma de organização de informações durante o planejamento de suas viagens. Entre os dados utilizados estão informações sobre cidades, datas, hospedagem, transportes, atrações turísticas, previsão de custos, entre outros, que são impressos posteriormente para consulta durante a viagem. O controle de custos ao longo da viagem é realizado através de anotações diárias no papel. Nesse processo informações podem ser perdidas ou esquecidas, além de não existir uma facilidade de acesso remoto das informações a qualquer momento.

O desenvolvimento do sistema Web de planejamento de viagens tem como foco facilitar a organização de informações referentes a roteiros de viagem, hospedagens, transportes, além da criação de estimativas de gastos. Todas essas informações serão editáveis pelo usuário, existindo a possibilidade de adicionar novos dados ao longo do tempo, como os gastos reais que estiverem ocorrendo durante a viagem. Portanto, se tornando uma plataforma que possibilita o planejamento e a organização de todos os aspectos de uma viagem antes, durante e depois da realização da mesma. Além disso, podendo também ser usado como uma ferramenta de controle de gastos durante viagens.

1.1 Objetivos

O objetivo deste trabalho é o desenvolvimento de uma aplicação Web de planejamento de viagens para uso dos funcionários e dos visitantes do blog Planejando Viagens. Os objetivos secundários são facilitar a organização de informações durante o planejamento de viagens e possibilitar a consulta dos dados de forma prática durante e após a

viagem. Além disso, também tem como objetivo se tornar uma ferramenta eficiente de controle e organização de gastos durante a viagem.

1.2 Estrutura

Este trabalho é dividido em 6 capítulos. O capítulo 2 apresenta os fundamentos e tecnologias utilizados para o desenvolvimento deste trabalho. O capítulo 3 apresenta aplicações com objetivos semelhantes aos deste trabalho, analisando seus pontos negativos, positivos e suas limitações. O capítulo 4 detalha o desenvolvimento da aplicação Web Planejando Viagens e os métodos utilizados relacionados com as tecnologias empregadas. O capítulo 5 apresenta a aplicação desenvolvida e as suas funcionalidades. O capítulo 6 apresenta as formas de avaliação da aplicação e a análise dos dados obtidos. O capítulo 7 apresenta as conclusões do trabalho e os resultados alcançados, além das limitações do trabalho e sugestões para trabalhos futuros.

2 FUNDAMENTOS

Neste capítulo será apresentada uma descrição dos conceitos, dos elementos e das tecnologias que servem de base para o desenvolvimento deste projeto, que foca fortemente nas escolhas das tecnologias indicadas para o desenvolvimento Web. Serão apresentados conceitos referentes à Metodologia Ágil, Protótipos Adobe XD, JavaScript, Node.js, Vue, Nuxt, Vuetify, MongoDB e Heurísticas de Nielsen.

2.1 Metodologia Ágil

Para o desenvolvimento do software deste projeto é utilizado a aplicação das metodologias ágeis, que tem como objetivo acelerar os processos, focando em uma melhoria contínua e incremental. Ao entregar parcelas funcionais do projeto aos usuários, permite que seja possível uma resposta rápida quando alterações são necessárias.

A metodologia realiza a subdivisão dos objetivos, em metas menores divididas em diferentes sprints. Uma sprint é uma janela de tempo para a conclusão das metas selecionadas, proporcionando entregas parciais ao final de cada sprint. Antes do início de uma sprint é realizada uma etapa de planejamento das ações a serem executadas e ao final é realizada uma retrospectiva de avaliação.

Assim, cada funcionalidade passa pelas etapas de planejamento, análise de requisitos, projeto, codificação e testes dentro das sprints. A adoção desta metodologia proporciona o aumento de comunicação da equipe, da organização para alcance das metas, da produtividade e da velocidade de reação a mudanças (HELABS, 2018).

2.2 Protótipos Adobe XD

Adobe XD (ADOBE, 2018) é uma solução de design para sites e aplicativos, possibilitando desenvolver protótipos de design e comportamentos da aplicação. Assim designers que utilizam esta ferramenta aumentam a qualidade, a precisão e a velocidade de criação dos seus trabalhos.

Neste trabalho é utilizado o Adobe XD para desenvolvimento dos protótipos de comportamento da aplicação, usando elementos fornecidos pela própria Adobe, para validar estes protótipos com o cliente antes que seja investido tempo de desenvolvimento.

2.3 JavaScript

A linguagem de programação JavaScript foi criada e é mantida pela European Computer Manufacturer's Association (ECMA). O JavaScript (MOZILLA, 2018) é uma linguagem de programação que permite implementar itens complexos em páginas Web, comum em páginas que envolvam mais do que mostrar informações estáticas. Sendo uma linguagem orientada a objetos, permite manipular todos os elementos de uma página como objetos distintos. O uso do Javascript com o HTML é o que possibilita a atualização dinâmica dos dados na tela neste trabalho por exemplo.

Existem três camadas básicas no desenvolvimento Web: O HTML responsável pela camada de informação, o CSS responsável pela folha de estilos e formatação e o JavaScript que é a terceira camada responsável pelo comportamento, onde se manipula as duas primeiras camadas de HTML e CSS (TABLELESS, 2018).

2.4 Node.js

Como visto na seção anterior o Javascript é uma poderosa ferramenta de programação para desenvolvimento de páginas Web dinâmicas e iterativas, normalmente sendo executado diretamente no navegador. Dentro desse contexto o Node.js (NODE.JS, 2018) é uma plataforma de desenvolvimento de aplicações Web em Javascript, baseado na *engine* do V8 do Chrome que permite rodar código em Javascript no servidor.

O Node.js é orientado a execução de eventos assíncronos projetado para criar aplicativos de rede escalonáveis. Possui gerenciador de pacotes o NPM (Node Package Manager) que facilita o processo de instalação de novas bibliotecas. O NPM é utilizado neste projeto tanto no desenvolvimento da aplicação no servidor como também no cliente, para instalação de bibliotecas auxiliares.

2.5 Vue

A sua documentação oficial (VUEJS, 2018a) define o Vue como um framework para construção de interfaces de usuário. Projetado para ser adaptável, sua biblioteca principal é focada para desenvolvimento da camada visual, sendo facilmente integrável com outras bibliotecas ou projetos existentes. Quando usado em conjunto com ferramentas e

bibliotecas de apoio também é perfeitamente capaz de dar poder a sofisticadas aplicações de página única.

2.6 Nuxt

Criado sobre o ecossistema Vue, o Nuxt.js (VUEJS, 2018b) é um framework que fornece uma experiência extremamente simplificada para escrever aplicações, podendo ser utilizado como um gerador de páginas estáticas e também como framework de apoio para o desenvolvimento de aplicações em Vue.

2.7 Vuetify

Vuetify (VUETIFY, 2018) é um framework HTML, CSS, e JS utilizado para desenvolvimento de interfaces. Esse framework oferece componentes e comportamentos desenvolvidos para compatibilidade com o Vue. Além disso, os componentes se apresentam preparados para seus conteúdos serem manipulados dinamicamente e prontos para que se adaptem a tela de qualquer dispositivo de navegação. Também é fornecida em sua biblioteca a padronização dos elementos, mantendo a aparência da aplicação uniforme em diferentes telas. Este framework é utilizado no trabalho principalmente por oferecer um conjunto de elementos suportados pelos principais navegadores de internet e por ser totalmente compatível com a linguagem desenvolvida o Vue.

2.8 MongoDB

O MongoDB (MONGODB, 2018) é um banco de dados orientado a documentos. Os dados são armazenados em documentos JSON com esquema dinâmico, possibilitando o armazenamento de registros sem se preocupar com número de campos, de tipos e de valores. A utilização do MongoDB neste trabalho é motivada principalmente pelo fato da mesma ser facilmente integrada com o Strapi. Além disso, sua utilização também é recomendada pelos próprios desenvolvedores do Strapi.

2.9 Heurísticas de Nielsen

As heurísticas foram originalmente projetadas por Nielsen and Molich (1990) e o processo de Avaliação Heurística pode ser descrito como: Uma lista de diretrizes de projeto que é usada como base para avaliação de um sistema ou aplicação. A lista original de heurísticas foi revisada, a fim de torná-las mais compreensíveis. A nova, e lista de heurísticas projetada por Nielsen (1993), é a seguinte:

1. **Visibilidade do estado do sistema:** O sistema deve sempre manter o usuário informado sobre o que está acontecendo no momento da interação.
2. **Equivalência entre o sistema e o mundo real:** O sistema deve falar a linguagem do usuário e cada palavra, frase e conceito deve ser familiar ao usuário. Não sendo orientado a termos técnicos, seguindo as convenções do mundo real fazendo a informação aparecer em uma ordem natural e lógica.
3. **Liberdade e controle do usuário:** Os usuários podem escolher funções do sistema por engano e precisarão de uma "saída de emergência" visível de maneira clara, que viabilize deixar um estado indesejado sem ter que passar por um diálogo extenso. Suporte desfazer e refazer, permitindo retornar a estados anteriores.
4. **Consistência e padrões:** Manter a consistência visual e de linguagem. Manter padrões de interação em diferentes contextos. O usuário não deve se perguntar se palavras, situações ou ações diferentes significam a mesma coisa.
5. **Prevenção de erro:** Ainda melhor do que boas mensagens de erro é um projeto cuidadoso, que evita um problema de ocorrer em primeiro lugar.
6. **Reconhecer ao invés de relembrar:** Torne objetos, ações e opções visíveis. O usuário não deve ter que lembrar informações de uma parte da interação para outra. As instruções de uso do sistema devem ser visíveis ou facilmente recuperáveis sempre que apropriado.
7. **Flexibilidade e eficiência de uso:** O sistema deve atender a usuários inexperientes e experientes. Possuindo atalhos que facilitem o uso de usuários experientes, mesmo que nunca vistos por usuários inexperientes. Além de permitir que os usuários personalizem as ações necessárias.
8. **Estética e design minimalista:** Os diálogos não devem conter informações irrelevantes ou raramente necessárias. Cada informação desnecessária em um diálogo compete com as informações relevantes de informação e diminui sua visibilidade.

9. **Auxiliar usuários a reconhecer, diagnosticar e recuperar ações erradas:** Mensagens de erro claras, com textos simples e diretos, sem códigos e sim o conduzindo à possíveis soluções.
10. **Ajuda e documentação:** Mesmo que o sistema possa ser utilizado sem documentação, ela pode ser necessária para fornecer ajuda ao usuário. Qualquer informação desse tipo deve ser fácil acesso e pesquisa, focada na tarefa do usuário, listar as etapas concretas a serem executadas e não ser muito extensa.

3 TRABALHOS RELACIONADOS

Neste capítulo apresentaremos diferentes ferramentas encontradas no mercado que possuem objetivos relacionados ao desenvolvido neste trabalho. Sendo elas o Inspirock, o Travefy, o Google Trips, o Travaa, o Trip30 e o Tripit. No final desse capítulo também será realizada uma análise comparativa dos sistemas apresentados.

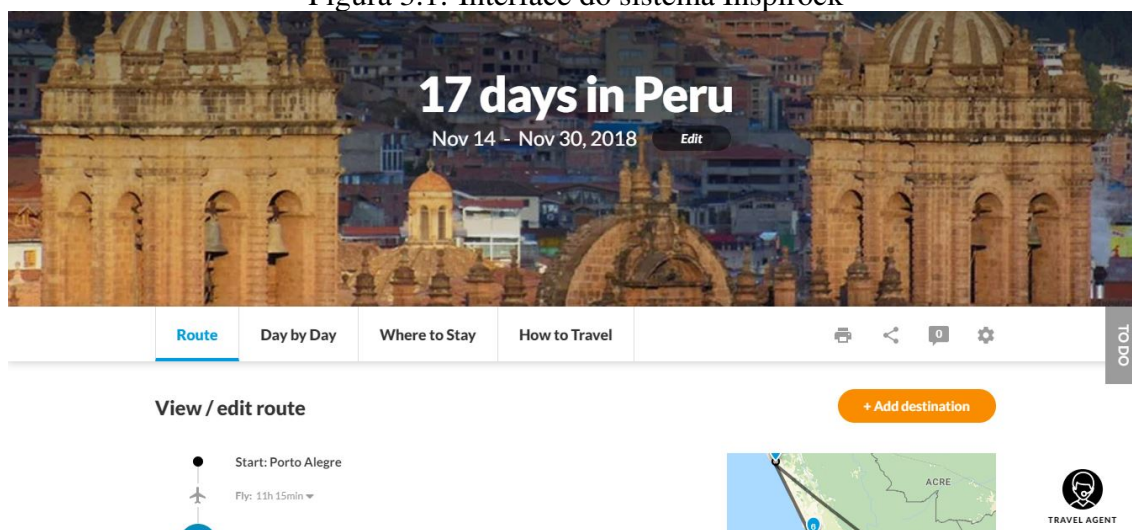
3.1 Inspirock

Projeto iniciado em 1991, visando simplificar o planejamento de viagens, o Inspirock atualmente é um dos serviços mais completos desta pesquisa.

No Inspirock logo após a criação de um novo plano de viagem um roteiro preenchido com informações sobre a cidade, dicas de hospedagem, transportes e passeios já é apresentado. O roteiro pode ser visualizado em forma de timeline, de calendário ou como mapa.

O sistema também permite que sejam cadastradas livremente pelo usuário novas atrações, hospedagens e transportes. Além disso, também é possível selecionar destinos e atrações recomendados automaticamente pelo sistema. Outro ponto positivo é que a ferramenta não apresenta nenhum custo ao usuário para criar o roteiro.

Figura 3.1: Interface do sistema Inspirock



Fonte: captura de tela do aplicativo pelo autor.

O único ponto negativo do Inspirock identificado nesta pesquisa é que não existe a opção de gerenciamento de gastos. Mesmo sendo possível informar os valores de hos-

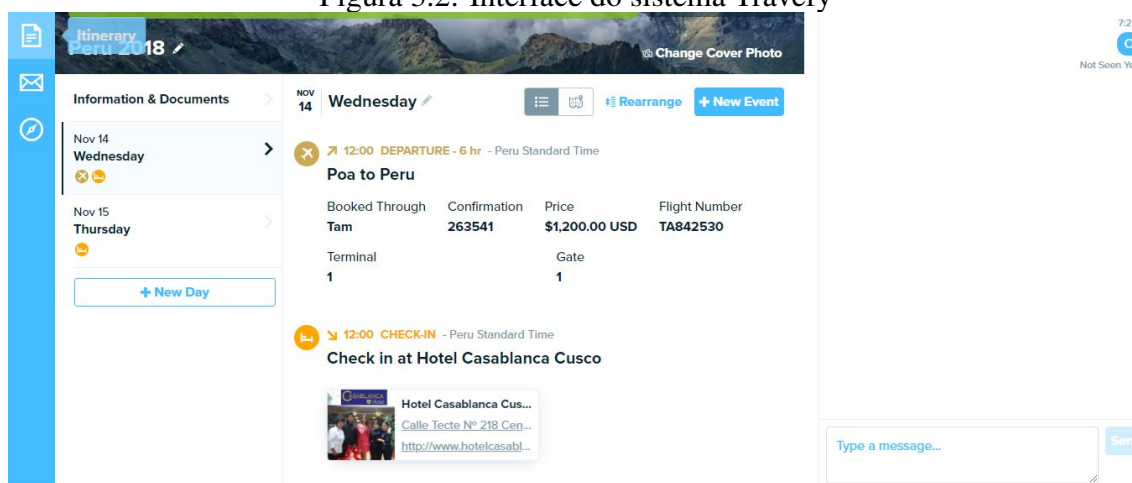
pedagens e passeios, esta informação não é apresentada de uma forma geral ao usuário.

3.2 Travefy

O sistema Travefy tem como foco a criação de roteiros, principalmente para viagens em grupo. Sendo assim, permite adicionar outros usuários que também podem editar as informações, além de disponibilizar um chat para troca de mensagens e vincular usuários a eventos no roteiro. No Travefy também é possível adicionar hospedagens, transportes, atrações e informações.

O sistema apresenta uma interface amigável, porém ao tentar adicionar um evento são solicitadas muitas informações, mesmo que com campos não obrigatórios tornam a experiência de cansativa. Outro ponto negativo é que o sistema não apresenta informações sobre os gastos da viagem. Mesmo que permita ser adicionado um valor vinculado a cada evento, este valor não é apresentado de uma forma geral, não existindo uma soma dos gastos totais, apenas eventos específicos.

Figura 3.2: Interface do sistema Travefy



Fonte: captura de tela do aplicativo pelo autor.

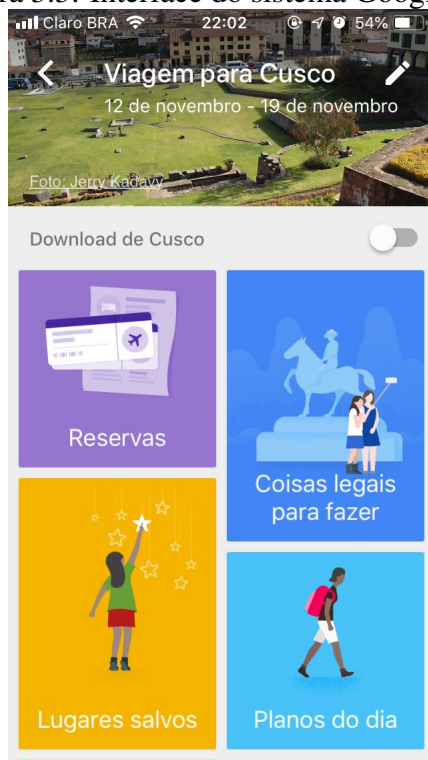
Apesar dos pontos negativos, o Travefy aborda a atividade de uma maneira bem interessante, principalmente para grupos, o que não é o foco deste trabalho. Após 10 dias de teste gratuitos existe uma taxa que deve ser paga tanto para viagens pessoais como para grupos. No plano mais básico a taxa mensal é de \$16 por usuário.

3.3 Google Trips

De todas as ferramentas analisadas o Google Trips é a única que não possui uma versão Web, funcionando somente para dispositivos móveis através de um aplicativo, lançado em setembro de 2016.

O aplicativo é desenvolvido para realizar gerenciamento de reservas, planejamentos diários, planejamento de viagens e para permitir a visualização de atividades perto da sua localização. Possuindo uma interface gráfica simples de se utilizar ao tentar criar novos planos, oferece automaticamente roteiros baseados nos dados de outros viajantes. Assim é possível adicionar atrações, hospedagens e transportes ao roteiro.

Figura 3.3: Interface do sistema Google Trip



Fonte: captura de tela do aplicativo pelo autor.

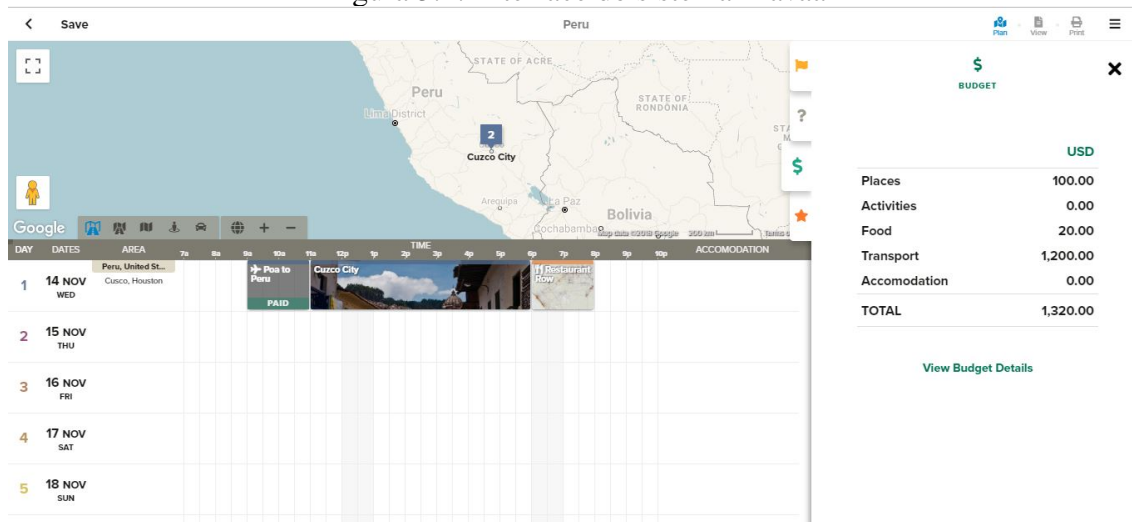
Apesar dos seus pontos positivos, ele apresenta um comportamento estranho ao tentar programar atividades diárias. O sistema automaticamente oferece locais para o usuário, não sendo possível realizar edições e nem alterar a ordem em que aparecem. Sendo assim somente uma lista de lugares a serem visitados que não seguem a ordem cronológica que o usuário deseja programar. Também não existe a opção de adicionar gastos no Google Trips, logo não sendo possível gerenciá-los.

3.4 Travaa

O sistema com maior pontuação analisada neste trabalho é o Travaa, uma aplicação criada na Nova Zelândia e que segundo seus criadores tem como objetivo ajudar no planejamento de uma viagem por completo. Após avaliação, essa é a aplicação que mais se aproxima dos objetivos deste trabalho.

O sistema do Travaa permite que sejam construídos roteiros informando atrações, cidades, hospedagens, transportes e alimentação. Sendo possível adicionar os gastos relacionados a cada um destes campos. Até aqui não se diferencia de outras aplicações analisadas, sendo então o seu diferencial a possibilidade de se visualizar os gastos separados por categoria.

Figura 3.4: Interface do sistema Travaa



Fonte: captura de tela do aplicativo pelo autor.

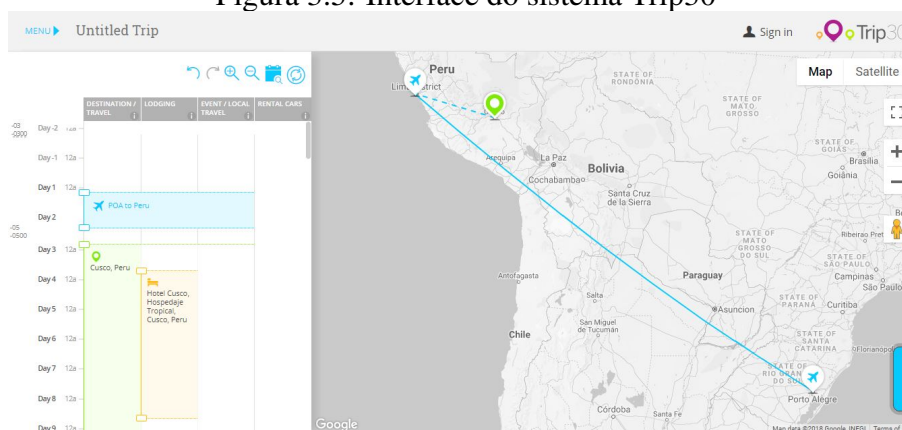
Apesar de possuir boas funcionalidades e de oferecer um serviço gratuito, a aplicação apresenta uma interface complicada para visualização dos eventos inseridos no calendário, sendo esse seu maior ponto negativo. Além disso, não existe versão em português dessa aplicação.

3.5 Trip30

Aplicação que teve seu desenvolvimento iniciado em 2012, inicialmente como uma start-up, foi criada com o objetivo de ajudar os usuários a planejar suas viagens de forma que pudessem aproveitar o processo de planejamento.

O Trip30 é focado totalmente na visualização de um mapa e de um calendário e permite que sejam adicionadas pelo usuário as informações como cidade, atrações, hospedagens e roteiros. A aplicação é focada no planejamento do roteiro, não sendo possível adicionar os valores referentes às atividades criadas.

Figura 3.5: Interface do sistema Trip30



Fonte: captura de tela do aplicativo pelo autor.

Também apresenta uma interface não muito agradável de interagir, em alguns momentos tornasse confuso navegar pelo calendário apresentado na tela, dificultando o entendimento do roteiro.

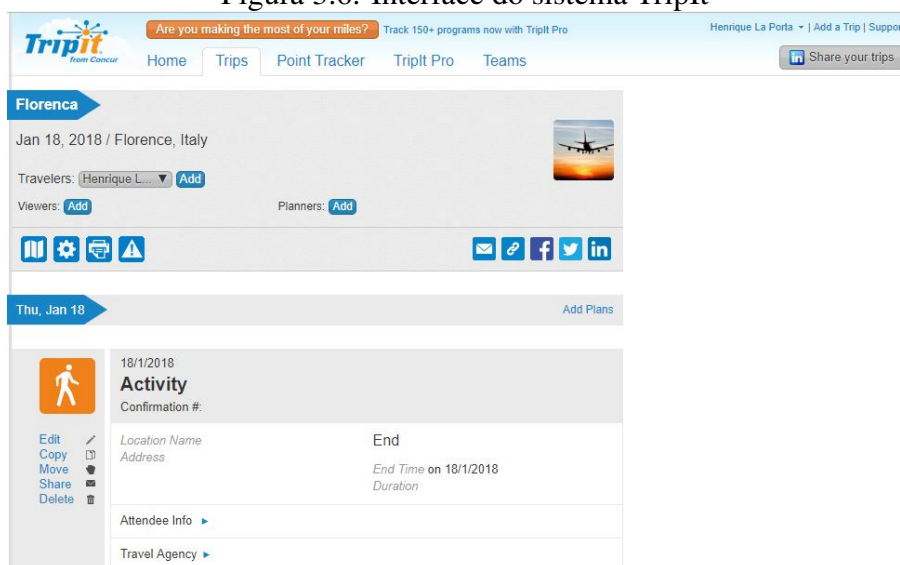
3.6 Tripit

O Tripit foi criado em 2006 pela empresa Concur e dentre todas aplicações analisadas é a única que apresenta uma interface desatualizada na sua versão Web. Muitas iterações não são amigáveis ao usuário, como a tela para inserir itens ao roteiro que se torna complicada.

A aplicação permite a visualização do roteiro como timeline, adicionar itens ao roteiro de acordo com o tipo do mesmo, compartilhar o roteiro, exportar o roteiro. Também é possível informar valores, porém não são utilizados para informações de gastos gerais.

A versão Web gratuita deixa a desejar em suas funcionalidades, mas é possível que as versões pagas, não analisadas neste trabalho, apresentem grande diferenças de interface e qualidade. A maior dificuldade identificada é a interação com a interface da aplicação, gerando uma grande dificuldade para poder inserir um item ao roteiro por exemplo.

Figura 3.6: Interface do sistema TripIt



Fonte: captura de tela do aplicativo pelo autor.

3.7 Análise comparativa

Após realizar esta pesquisa é possível extrair dos resultados um quadro comparativo de funcionalidades presentes em cada aplicação. Para facilitar a compreensão, as funcionalidades estão separadas por categoria e mostradas na tabela 3.1.

- **Distribuição:** Refere-se a forma que um usuário pode ter acesso a aplicação.
- **Construção:** Funcionalidades da aplicação que refere-se a construção de uma viagem, roteiro e inserção de dados.
- **Visualização:** Formas que o usuário pode visualizar o seu roteiro com as informações inseridas.
- **Armazenamento:** Agrupam-se as funcionalidades de armazenamento de roteiros de cada aplicação e também possibilidade de compartilhamento.

Através desta análise, é possível observar que apenas uma das aplicações analisadas possui a opção de visualização dos gastos informados no roteiro. Outras duas aplicações permitem que sejam informados valores referentes a itens do roteiro, mas não permitem a visualização dos gastos. As demais aplicações não possuem a funcionalidade de informar gastos.

Esta pesquisa permite visualizar as lacunas presentes nas funcionalidades das aplicações estudadas e também identificar funcionalidades que devem estar presentes neste trabalho.

Tabela 3.1: Tabela de análise comparativa

Funcionalidades / Aplicação		<i>Inspirock</i>	<i>Travefy</i>	<i>Google Trip</i>	<i>Trip30</i>	<i>Tripit</i>	<i>Travaa</i>
<i>Distribuição</i>	Versão gratuita	X	X	X	X	X	X
	Versão paga		X			X	
	Versão Web	X	X		X	X	X
	Versão mobile	X	X	X	X	X	X
<i>Construção</i>	Adicionar atrações	X	X	X	X	X	X
	Adicionar hospedagens	X	X	X	X	X	X
	Adicionar transporte	X	X	X	X	X	X
	Recomendações para o roteiro	X		X			
	Possível informar valores	X	X				X
	Facilidade de inserção de dados	F	D	F	M	D	F
<i>Visualização</i>	Roteiro como timeline	X	X		X	X	X
	Roteiro como mapa	X	X	X	X		X
	Roteiro como calendário	X			X		
	Visualizar informações de gastos						X
	Exportar roteiro (Pdf, Doc)	X	X	X	X	X	X
<i>Armazenamento</i>	Roteiro	X	X	X	X	X	X
	Compartilhamento	X	X	X	X	X	X

Legenda: F = Fácil, M = Médio e D = Difícil

Fonte: elaborado pelo autor.

4 DESENVOLVIMENTO

Neste capítulo será apresentado o desenvolvimento da aplicação Web Planejando Viagens, que tem como objetivo facilitar a organização de informações referentes a roteiros de viagem, hospedagem, transportes, além da criação de estimativas de gastos.

Nas próximas seções serão apresentados os requisitos e os objetivos utilizados, os protótipos de baixa fidelidade e os elementos de domínio do problema. Além disso, também serão abordados os temas de desenvolvimento da aplicação Web, do banco de dados e do sistema de mapas e endereços.

4.1 Requisitos e objetivos

Para realizar a identificação dos requisitos, foi feita uma análise das funcionalidades existentes em outras aplicações relacionadas e a mesma foi apresentada aos funcionários do blog Planejando Viagens. A partir disso, foram identificadas quais funcionalidades seriam relevantes considerando as necessidades existentes e as tarefas executadas através de tabelas atualmente.

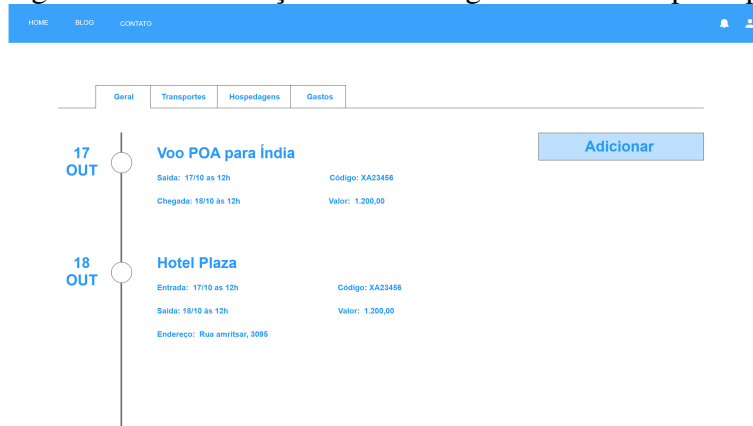
Portanto, definiu-se que a aplicação deve:

- Permitir a construção e a edição de roteiros, através da definição de destinos e locais pelo usuário.
- Permitir a inserção e a edição de hospedagens no roteiro.
- Permitir a inserção e a edição de passagens no roteiro, sejam aéreas, terrestres, marítimas e etc.
- Permitir a inserção, o controle e a edição de gastos que ocorrem ou podem vir a ocorrer, na execução de um roteiro.
- Possibilitar ao usuário visualizar os gastos separados por categoria.
- Permitir que os roteiros sejam armazenados no sistema.
- Ter interface responsiva, ou seja, adaptável a qualquer dispositivo de qualquer tamanho de tela.

4.2 Protótipos de baixa fidelidade

Com objetivo de obter uma visão geral de como se desenvolveria a estrutura do front-end e seu comportamento, foram construídos protótipos de baixa fidelidade. Estes protótipos apresentam a estrutura visual do objetivo que se deseja focando no conceito de funcionamento do sistema e não apresenta detalhes de interface, que precisam refletir no sistema final.

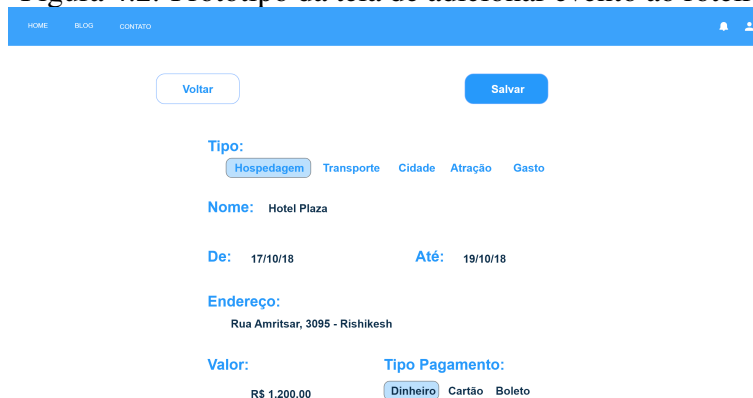
Figura 4.1: Visualização de uma viagem na sua tela principal



Fonte: elaborado pelo autor.

Através destes protótipos de telas, foi possível validar com os funcionários do blog Planejando Viagens se tal abordagem para a execução das atividades satisfazia suas necessidades, já que protótipos são iterativos, ou seja, é possível clicar nos botões e visualizar a transição das telas simulando realmente o uso do sistema.

Figura 4.2: Protótipo da tela de adicionar evento ao roteiro



Fonte: elaborado pelo autor.

O uso destes protótipos ajudou no entendimento dos requisitos de cada funcionalidade, fornecendo informações sobre a real necessidade do usuário e do sistema,

gerando uma facilidade para a definição das metas que poderiam ser executadas dentro de uma sprint do projeto.

4.3 Entidades

Este projeto trabalha com um grupo pequeno de entidades que se resumem às instâncias mais básicas planejadas para a aplicação. Importante lembrar que as entidades podem ser criadas pelo usuário ou ser utilizado um método de pesquisa por API de terceiros, assim é armazenado somente o identificador para pesquisa nesta API. A primeira delas representa uma atração, ou seja, um local ou um endereço a ser visitado no mapa. Uma entidade atração é composta dos seguintes atributos:

- **ID:** Identificador único, gerado no momento de sua criação.
- **Nome:** Nome dado a atração.
- **Descrição:** Breve descrição da atração.
- **Site:** Url para o site ou outro endereço com informações.
- **Endereço:** Endereço real do local para ser encontrado no mapa.
- **Data:** Data que o usuário definiu que irá realizar a visita.
- **Valor:** Numeral referente ao valor da atração.
- **Id Roteiro:** Identificador para a qual roteiro ela se relaciona.

Outra entidade que se relaciona a um Roteiro é Hospedagem. Representando locais onde o usuário pretende se hospedar no dia a dia de seu roteiro. Uma entidade **Hospedagem** é composta por:

- **ID:** Identificador único, gerado no momento de sua criação.
- **Nome:** Nome dado a hospedagem.
- **Site:** Url para o site ou outro endereço com informações.
- **Endereço:** Endereço real do local para ser encontrado no mapa.
- **Data de chegada:** Data definida para início da hospedagem, ou seja, data de check-in.
- **Data de saída:** Data de saída da hospedagem, ou seja, data de check-out.
- **Valor:** Numeral referente ao valor da hospedagem.
- **Id Roteiro:** Identificador do roteiro ao qual as informações são relacionadas.

A Passagem é outra entidade que se relaciona diretamente com o Roteiro. Essa entidade representa qualquer tipo de transporte necessário para a execução do roteiro.

Uma entidade **Passagem** é composta por:

- **ID:** Identificador único, gerado no momento de sua criação.
- **Origem:** Cidade de origem.
- **Destino:** Cidade de destino.
- **Data de partida:** Data de início da viagem.
- **Data de chegada:** Data de chegada no destino.
- **Valor:** Numeral referente ao valor da passagem.
- **Id Roteiro:** Identificador para a qual roteiro ela se relaciona

A entidade Roteiro relaciona todas as entidades apresentadas acima. A relação do objeto roteiro representa a união de todas as informações do usuário vindas da entidade Atração, Hospedagem e Passagem. A entidade **Roteiro** é composta por:

- **ID:** Identificador único gerado no momento de sua criação.
- **Nome:** Nome dado pelo usuário para o roteiro.
- **Data início:** Data de início do roteiro.
- **Data fim:** Data fim do roteiro.
- **ID viagem:** Referência ao identificador único da entidade Viagem ao qual este Roteiro pertence.

A entidade Viagem representa todos os atributos necessários para identificação de uma viagem. Uma entidade **Viagem** é composta por:

- **ID:** Identificador único gerado no momento de sua criação.
- **Nome:** Nome ou título dado à viagem, atribuído pelo usuário no momento de sua criação.
- **ID usuário:** Referência ao identificador único da entidade usuário ao qual esta viagem pertence.

A entidade Usuário define o usuário do sistema e suas informações são obtidas durante o cadastro de um usuário na aplicação via a interface. Uma entidade **Usuário** é composta por:

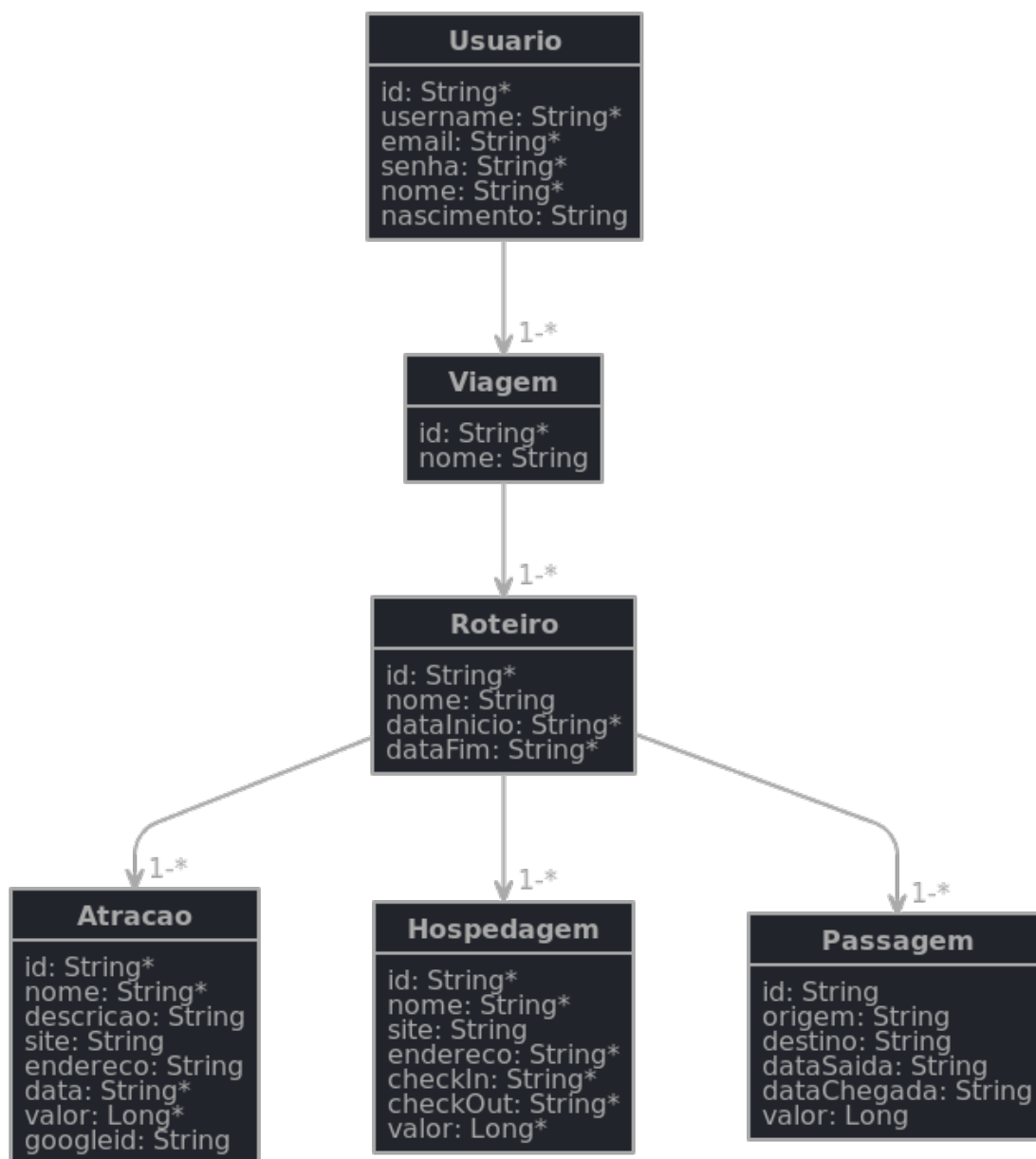
- **ID:** Identificador único, gerado no momento de sua criação.
- **Username:** Um nome de usuário informado pelo usuário, que deve ser único em

todo o sistema.

- **E-mail:** Endereço de e-mail informado pelo usuário durante o cadastro.
- **Senha:** Senha informada pelo usuário durante o cadastro.

Pelo fato do sistema possuir poucas entidades definidas o diagrama de relacionamento delas acaba sendo simples e permite maior entendimento do funcionamento do sistema. A figura 4.3 mostra o digrama de relacionamento das entidades.

Figura 4.3: Diagrama de Entidade-Relacionamento



Fonte: elaborado pelo autor.

4.4 Aplicação Web

A aplicação Web é baseada em uma única página HTML, que é carregada apenas uma única vez para o usuário e isto ocorre no momento de acesso.

O dinamismo da aplicação e a resposta a comando dos usuários é desenvolvido com o auxílio do *framework* Nuxt da biblioteca em Javascript Vue.js. Através de documentos Javascript a mesma se responsabiliza pela renderização dos componentes gráficos, seja para inclusão, exclusão, modificação ou para realizar o controle da página executando trechos de código.

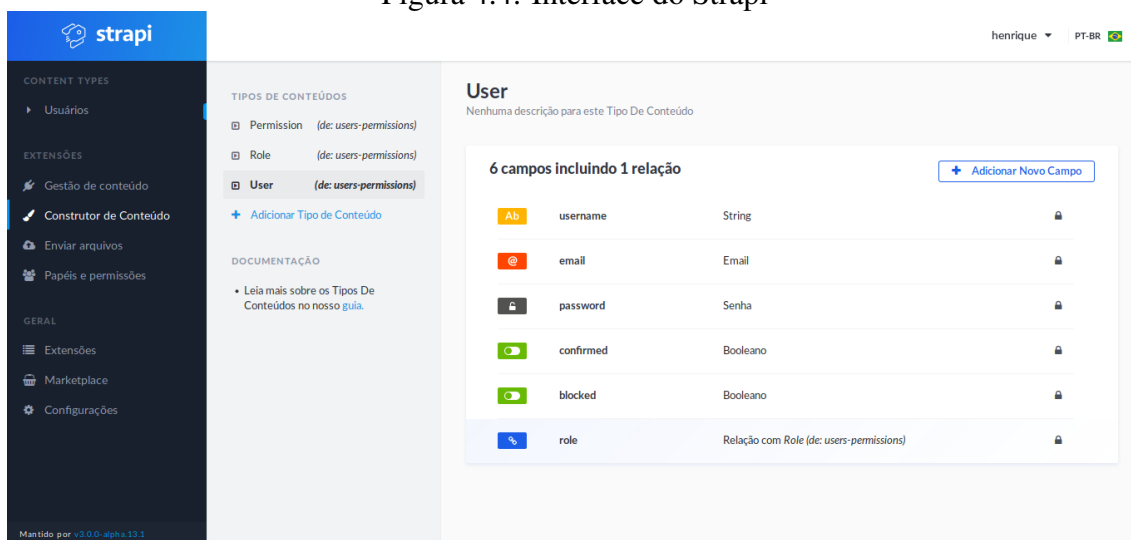
O Vuetify, um framework de componentes para Vue.js, é utilizado para realizar a padronização das interfaces, do comportamento de componentes e dos elementos da página. Além disso, também contribui para que a aplicação Web seja responsiva, permitindo que a mesma funcione em diferentes tamanhos de telas e de dispositivos presentes no mercado.

4.5 Banco de Dados

Para definir a ferramenta que seria utilizada para gerenciamento e criação do banco de dados, o principal ponto levado em consideração foi o uso de uma ferramenta que permitisse que pessoas com o conhecimento básico sobre a base de dados conseguissem realizar pequenas alterações na estrutura. De forma que no futuro manutenções básicas possam ser realizadas facilmente. Assim foi escolhido o Strapi, que é disponibilizado de forma open-source.

Desenvolvido para criar uma API REST utilizando a linguagem Node.js, o Strapi fornece uma interface Web para realização da administração da biblioteca, sendo possível fazer manutenção, além de visualizar toda a estrutura dos dados armazenados. O gerenciamento de usuários e a autenticação via API através do JWT são fornecidos pelo Strapi. Além disso, o Strapi realiza automaticamente a integração com o Banco de dados, sendo utilizado o MongoDB por ser o que possui melhor integração com a aplicação.

Figura 4.4: Interface do Strapi



Fonte: Captura de tela do aplicativo pelo autor.

Para tornar possível a comunicação entre a API e a aplicação é utilizada uma biblioteca javascript do Strapi, não sendo obrigatório o uso da mesma.

Realizando a importação da biblioteca para uso no seu código é possível fazer o uso de suas funções, como por exemplo na figura 4.5 que mostra a chamada de *login* que automaticamente resolve o armazenamento da chave de autenticação JWT via API facilitando o desenvolvimento do sistema.

Figura 4.5: Trecho de código de uso da biblioteca Strapi

```
async login( email, password ){
  return await this.strapi.login(email, password)
}
```

Fonte: elaborado pelo autor.

4.6 Mapas e endereços

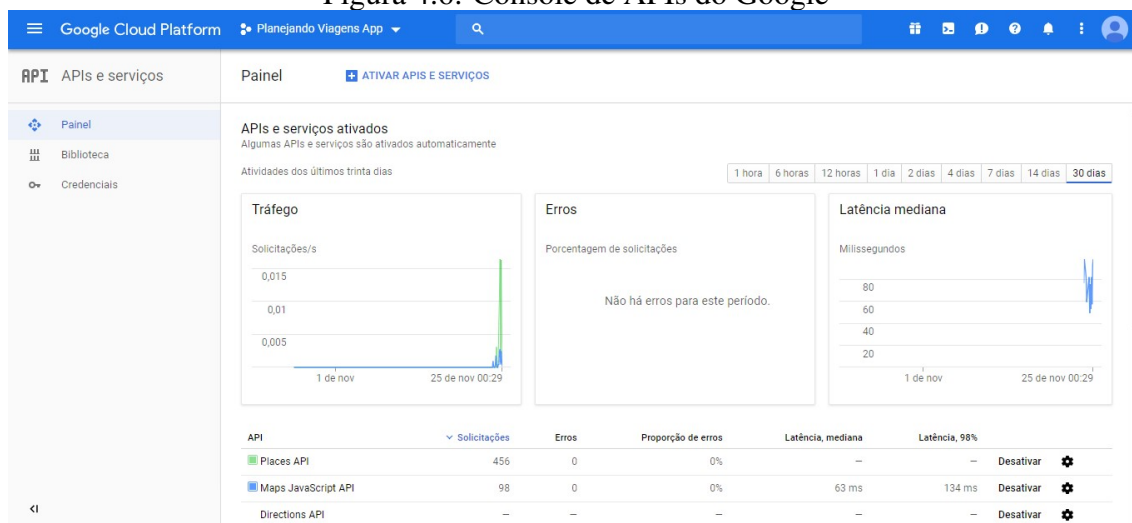
Um dos desenvolvimentos que não estava previsto inicialmente no trabalho é permitir que o usuário visualize em um mapa os pontos definidos no seu roteiro previamente. Para adicionar essa funcionalidade foi necessário a escolha de uma ferramenta que disponibilizasse suporte a mapas e dentre as ferramentas existentes se destacam o OpenStreetMap API e o Google Maps API. A primeira possui a vantagem de ser uma ferramenta de

código aberto, sem restrições legais e técnicas para a sua utilização. Porém, a segunda opção acabou sendo escolhida devido a sua base de documentação mais extensa e detalhada. Além disso, outro fator considerado foi a prévia experiência do autor deste trabalho com o uso da API, permitindo uma curva de aprendizagem menor.

Para o uso do Google Maps API é necessário primeiramente a criação de uma chave de identificação, que é posteriormente vinculada a este projeto. Tal chave é responsável pela identificação e controle de acesso das funcionalidades nos servidores do Google. Através dela é realizado o controle de quantidade de requisições dos serviços que são limitados de acordo com o plano, no caso deste trabalho o plano utilizado é o gratuito.

A API fornece um painel (Figura 4.6) de acompanhamento de estatísticas de uso dos serviços, onde é possível gerenciar acessos de segurança, ocorrência de erros e restringir o uso da chave da aplicação.

Figura 4.6: Console de APIs do Google



Fonte: Captura de tela do aplicativo pelo autor.

Para viabilizar a utilização do mapa na aplicação é feito o uso do pacote `vue2-google-maps` disponível via NPM. Ao importar este pacote para a aplicação ele acompanha componentes Vue já configurados para se comunicar com a API do Google. Não é necessário realizar o controle dos eventos de `listener` gerados pelos elementos do mapa ou do input do sistema de `autocomplete`. Como vemos na imagem abaixo (Figura 4.7) o uso de dois componentes: o `GmapMap` utilizado para a exibição do mapa e o `GmapMarker`, que realiza a inserção das marcações dentro do componente do mapa.

Figura 4.7: Trecho de código de uso do pacote `vue2-google-maps`

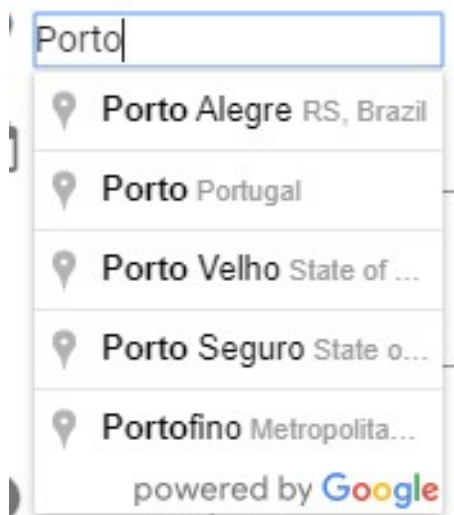
```

<GmapMap
  :center="center"
  :zoom="7"
  map-type-id="roadmap"
>
  <GmapMarker
    :key="index"
    v-for="(m, index) in markers"
    :position="m.position"
    :clickable="true"
    :draggable="true"
    @click="center=m.position"
  />
</GmapMap>

```

Fonte: elaborado pelo autor.

Ao utilizar os parâmetros do componente é possível alterar dinamicamente a posição geográfica que está sendo exibida pelo mapa do componente, fornecendo através parâmetro `center` um valor de latitude e longitude. Outras funcionalidades são a possibilidade de alterar o zoom do mapa e o tipo de exibição, como por exemplo satélite ou relevo.

Figura 4.8: Exemplo de comportamento do componente `gmap-autocomplete`

Fonte: elaborado pelo autor.

Para receber as informações de endereço é utilizado o componente `gmap-autocomplete`, que fornece a função de auto completar a informação que está sendo digitada pelo usuário, como mostra a figura 4.6. Sendo possível pesquisar um país, uma cidade, uma rua, o nome de um estabelecimento, de uma atração turística, entre outros. Após o usuário selecionar uma das opções disponíveis, são recebidas informações sobre o local seleci-

onado através da Google Places API. Entre estas informações estão dados como: nome, endereço, imagens, avaliações, posição geográfica e outros.

Figura 4.9: Trecho de código de uso do pacote vue2-google-maps

```
<gmap-autocomplete  
  @place_changed="setPlace">  
</gmap-autocomplete>
```

Fonte: elaborado pelo autor.

Para uso desta funcionalidade basta adicionar o componente, como mostra a imagem acima (Figura 4.9) e informar uma função que irá receber por parâmetro o objeto retornado da Google Places API, no caso a função `setPlace`.

5 A APLICAÇÃO

Este capítulo apresenta a aplicação desenvolvida e as suas funcionalidades, mostrando o comportamento visual da aplicação de acordo com a iteração do usuário. Serão abordados os processos de autenticação e cadastro, de como criar uma viagem, de como adicionar informações ao roteiro e de como visualizar gastos.

5.1 Autenticação e cadastro

Um dos requisitos para o usuário ter acesso a aplicação é estar logado, então assim que entrar na tela inicial da aplicação o mesmo possui duas opções: cadastrar uma nova conta no sistema ou fazer login com uma conta já existente.

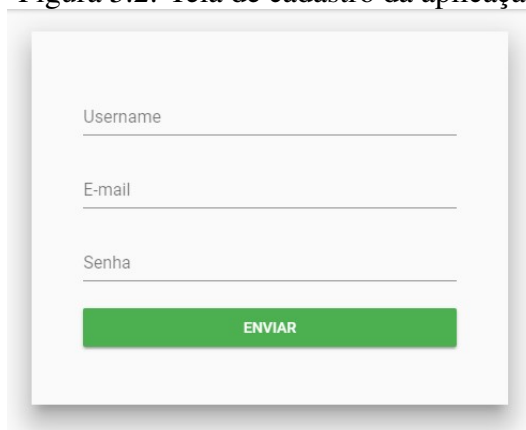
Figura 5.1: Tela inicial da aplicação



Fonte: elaborado pelo autor.

Para cadastrar uma nova conta o usuário deve clicar no botão "Registrar", como apresentado na tela inicial (Figura 5.1) e escolher um nome de usuário, uma senha e o e-mail que será vinculado a conta. Ao preencher todas as informações corretamente e clicar no botão "Enviar" da tela de registro (Figura 5.2) o usuário será redirecionado para a sua página de usuário.

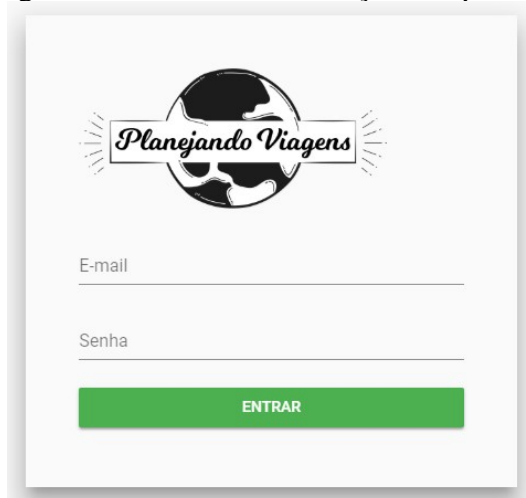
Figura 5.2: Tela de cadastro da aplicação

A screenshot of a registration form. It features three input fields labeled 'Username', 'E-mail', and 'Senha' (Password). Below the fields is a green button with the text 'ENVIAR' (SEND).

Fonte: elaborado pelo autor.

Caso já possua cadastro, o usuário deve clicar no botão "Login" apresentado na tela inicial (Figura 5.1) e preencher os dados solicitados. Apenas duas informações são necessárias para realizar o login: usuário ou e-mail e a senha.

Figura 5.3: Tela de autenticação da aplicação

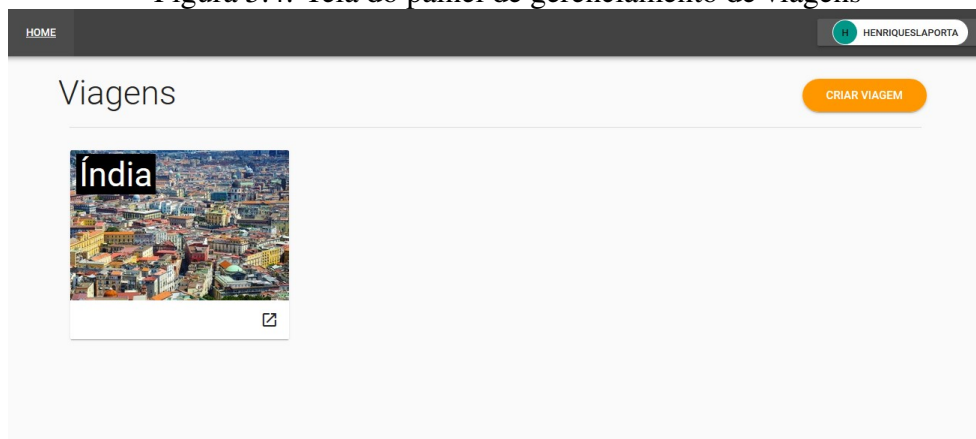
A screenshot of a login form. At the top center is a logo for 'Planejando Viagens' featuring a globe with a banner across it. Below the logo are two input fields labeled 'E-mail' and 'Senha'. At the bottom is a green button with the text 'ENTRAR' (ENTER).

Fonte: elaborado pelo autor.

5.2 Como criar uma viagem

Um dos objetivos deste trabalho é possibilitar a organização de informações de roteiros e de gastos de uma viagem. Para que o usuário possa ter acesso a funcionalidade de criar uma viagem no sistema ele precisa estar logado, como foi mostrado na seção 5.1. Após estar devidamente logado, a sua página principal de usuário é um painel (Figura 5.4) onde é possível gerenciar as viagens vinculadas a sua conta.

Figura 5.4: Tela do painel de gerenciamento de viagens



Fonte: elaborado pelo autor.

Nesta tela do painel é possível visualizar um botão “Criar Viagem”. Ao clicar neste botão uma janela é aberta (Figura 5.5) solicitando ao usuário as informações de nome, data de início e data de fim. Após finalizar esta etapa, o sistema cria uma viagem e automaticamente também inicia um roteiro vinculado a mesma. A viagem criada passa a constar no painel e pode ser acessada ao clicar no botão “Abrir”.

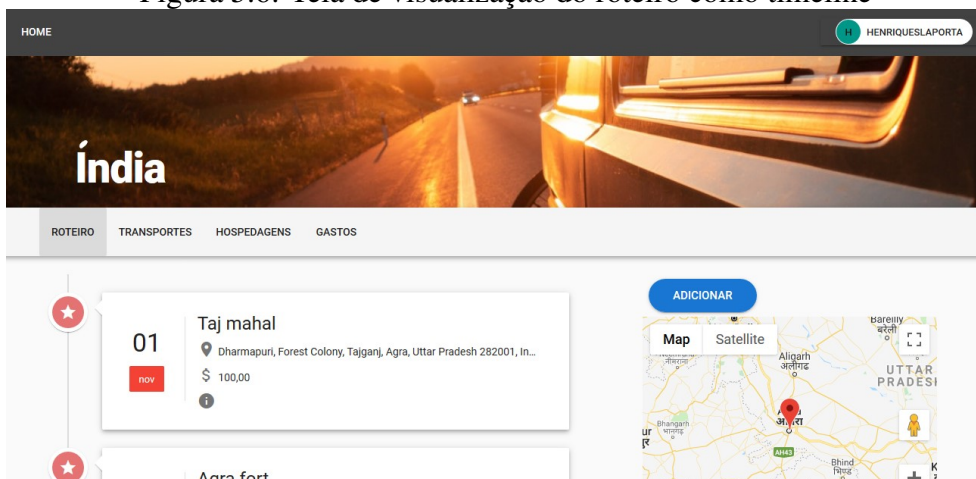
Figura 5.5: Tela para criar uma viagem

Fonte: elaborado pelo autor.

5.3 Como adicionar informações ao roteiro

Uma funcionalidade básica disponível para que o usuário possa criar e gerenciar seus próprios roteiros é a possibilidade de inserir eventos. Um evento pode representar diferentes tipos de informação, podendo ser uma Atividade, uma Hospedagem, um Transporte ou um Gasto. O único tipo de evento que não pode ser adicionado a timeline do roteiro é o tipo de gasto, que apenas é apresentado na página de visualização de gastos.

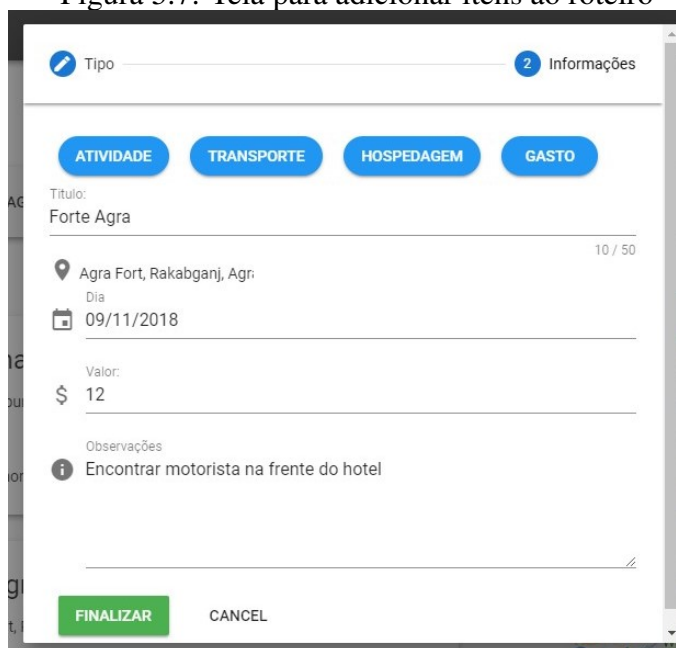
Figura 5.6: Tela de visualização do roteiro como timeline



Fonte: elaborado pelo autor.

Ao clicar no botão “Adicionar” posicionado a direita da timeline uma nova janela é aberta (Figura 5.7), na qual o usuário deve informar a categoria do item a ser adicionado e as informações referentes ao mesmo.

Figura 5.7: Tela para adicionar itens ao roteiro



Fonte: elaborado pelo autor.

5.4 Visualizar os gastos

Na tela principal de gerenciamento de uma viagem é possível observar um menu localizado abaixo do título da viagem. Ao clicar no botão “Gastos” o usuário é redireci-

onado para a tela de gerenciamento dos custos da viagem, onde existem duas formas de visualização: os gastos gerais por categoria (Figura 5.8) ou detalhados por dia e categoria no formato de tabela.

Figura 5.8: Tela de visualização dos gastos



Fonte: elaborado pelo autor.

Para alternar entre a forma de visualização dos gastos o usuário deve selecionar no menu exibido a opção "dia a dia" que irá alternar para o formato de tabela. O formato tabela apresenta uma linha de entrada na tabela para cada dia da viagem e as colunas representam o tipo de gasto com o valor referente somente ao dia da linha.

6 ANÁLISE DA APLICAÇÃO

Neste capítulo serão apresentadas as duas formas utilizadas para avaliar a aplicação e a análise dos resultados obtidos. Para validar o estado atual em que o sistema se encontra foram utilizadas como formas de avaliação as heurísticas de Nielsen e uma pesquisa qualitativa realizada através de entrevista.

6.1 Avaliação Heurística

A primeira forma de avaliação da aplicação foi uma avaliação heurística, usando como base as heurísticas apresentadas por Nielsen (1993) e descritas na seção 2.9. As heurísticas são utilizadas para analisar os pontos positivos e negativos da interface de interação com o usuário. A análise heurística apresentada nessa seção foi realizada pelo próprio autor deste trabalho.

A partir dos dados obtidos (Tabela 6.1) é possível identificar que um dos pontos que obteve o melhor resultado foi o da heurística de equivalência do sistema com o mundo real. Além disso, outra heurística que obteve a maior pontuação foi a de reconhecer ao invés de lembrar, o que permite que a aplicação seja mais intuitiva ao usuário.

Tabela 6.1: Tabela de análise das heurísticas de Nielsen

Heurística	1	2	3	4	5	6	7
Visibilidade do estado do sistema	Ruim					X	Bom
Equivalência entre o sistema e o mundo real	Ruim						X Bom
Liberdade e controle do usuário	Ruim			X			Bom
Consistência e padrões	Ruim					X	Bom
Prevenção de erro	Ruim		X				Bom
Reconhecer ao invés de lembrar	Ruim						X Bom
Flexibilidade e eficiência de uso	Ruim					X	Bom
Estética e design minimalista	Ruim					X	Bom
Auxiliar usuários a reconhecer, diagnosticar e recuperar ações erradas	Ruim				X		Bom
Ajuda e documentação	Ruim	X					Bom

Fonte: elaborado pelo autor.

As demais heurística como visibilidade do estado do sistema, consistência e padrões, flexibilidade e eficiência de uso e estética e design minimalista apresentam uma boa avaliação com nota 6, mas são aspectos do sistema que ainda poderiam ser revisados e aprimorados.

Também é possível identificar que o sistema apresenta um baixo desempenho nas heurísticas de documentação, de prevenção de erros e de liberdade e controle do usuário. Na prevenção de erros, ao adicionar itens ao roteiro, o sistema permite que algumas informações fiquem em branco no formulário e ao inserir determinado item ocorrem alguns erros de formatação na exibição. A falta de uma documentação de auxílio para o usuário é o ponto mais crítico do sistema. Mesmo que o sistema desenvolvido seja intuitivo, caso o usuário necessite de ajuda para alguma ação o sistema não fornece auxílio.

6.2 Avaliação de aplicação através de entrevista

A segunda forma de avaliação da aplicação foi uma pesquisa qualitativa realizada através de entrevista com a autora do Blog Planejando Viagens após a apresentação da aplicação. A entrevista realizada em 1 de dezembro de 2018 foi semiestruturada e com questões abertas para possibilitar o melhor entendimento da perspectiva da entrevistada (ROESCH, 2005). Como instrumento de coleta de dados foi utilizado um questionário construído com base nos objetivos deste estudo. O roteiro de entrevista está no Apêndice A.

A entrevistada é do sexo feminino, tem 28 anos, é formada em Administração com ênfase em Comércio Internacional e trabalha como analista de logística em uma empresa do ramo industrial. Criou o blog Planejando Viagens há pouco mais de um ano como projeto paralelo que pretende tornar fonte de renda nos próximos anos.

Os dados coletados para avaliação da aplicação tem como objetivos identificar os pontos apresentados no quadro abaixo (Tabela 6.2).

Tabela 6.2: Tabela do plano de coleta de dados

Objetivos da entrevista
Identificar as principais dificuldades existentes antes da aplicação
Identificar os principais pontos positivos da aplicação
Identificar qual é o impacto da aplicação na rotina de planejamento de viagens
Identificar quais os pontos da aplicação não atenderem as expectativas
Identificar quais pontos poderiam ser desenvolvidos para agregar valor à aplicação

Fonte: elaborado pelo autor.

6.3 Apresentação e análise dos dados coletados

Nessa seção serão apresentados e analisados os dados obtidos através da entrevista.

- **Identificação das principais dificuldades existentes antes da aplicação**

A partir da entrevista foi identificado que antes da aplicação existia a dificuldade para fazer a gestão de tantas informações envolvidas no planejamento de uma viagem. Isso ocorria principalmente em roteiros que envolviam várias cidades e o volume de dados acabava ficando muito grande. O Excel era a principal ferramenta utilizada para o armazenamento e organização da informações, porém existia a dificuldades de centralizar os dados em um mesmo lugar ou na mesma planilha devido a diversidade de informações existentes.

Outra dificuldade era a gestão dos gastos de cada viagem. Os gastos eram anotados no papel ou no celular durante a viagem e depois transferidos para o Excel após a finalização da mesma. Além disso, seriam somados com os gastos já identificados anteriormente, como passagem aérea, hospedagens que tiveram o pagamento realizado antes da viagem e outros.

- **Identificação dos principais pontos positivos da aplicação**

Como principal ponto positivo da aplicação foi identificado o gerenciamento de gastos interligado com o roteiro. Ao planejar o roteiro e ao fazer reservas os custos já serão adicionais ao controle de gastos automaticamente. Assim evitando a necessidade de duas formas de controle diferentes no planejamento da viagem, uma para roteiro e

uma para gastos. Além disso, a divisão dos custos por categoria também foi um ponto importante, pois ao final da viagem já é possível visualizar os gastos resumidos.

Outro ponto identificado como muito bom, e que não estava nas necessidades iniciais da entrevistada, é o da aplicação já marcar no mapa os lugares que são adicionados ao roteiro. Após adicionar os lugares que deseja visitar já é possível visualizar a localização de cada um no mapa e identificar se é necessário fazer alterações para tornar o roteiro mais eficiente.

- **Identificação do impacto da aplicação na rotina de planejamento de viagens**

Como principal impacto da aplicação na rotina de planejamento de viagens foi identificado que a mesma facilita muito a organização de informações. Além disso, reduz quase totalmente a necessidade do uso de diversas planilhas de Excel para organizar os dados de cada viagem e torna a gestão dos gastos mais eficiente.

- **Identificação dos pontos que não atenderem as expectativas**

Um dos principais pontos que não atenderam a expectativa na aplicação foi não existir a possibilidade de exportar o conteúdo. Isso é um ponto importante para a entrevistada, pois os dados das viagens também serão usados nos posts do blog, que falam sobre roteiro e gastos. Se houvesse a opção de exportar o conteúdo isto facilitaria muito a construção desses posts. Além disso, isso seria importante também para acessar os dados quando não houver internet, pois a aplicação no momento só pode ser acessada online.

Não ser possível acessar a aplicação sem internet foi outro ponto que não atendeu as expectativas, pois durante viagens nem sempre é possível acessar a internet, prejudicando os dados que seriam adicionados durante a viagem.

- **Identificação dos pontos que poderiam ser desenvolvidos para agregar valor à aplicação**

Entre os pontos identificados que poderiam ser desenvolvidos para agregar valor à aplicação um deles seria a possibilidade de adicionar um controle do orçamento disponível para a viagem. Isso seria feito adicionando o valor inicial disponível já dividido em dinheiro, cartão de crédito e outros. Ao adicionar os gastos da viagem os mesmos seriam descontados desse valor, restando apenas o valor ainda disponível no orçamento.

Além disso, um ponto interessante que poderia ser devolvido é a possibilidade de adicionar mais de um usuário na mesma viagem. Tornando possível dividir determinados custos em mais de uma pessoa, como hospedagem, por exemplo, pois muitas vezes o valor de um quarto é dividido em duas ou mais pessoas.

7 CONCLUSÕES

Neste capítulo serão apresentadas as conclusões do trabalho e os resultados alcançados considerando os objetivos propostos. Também serão analisadas as limitações encontradas, além de apresentar sugestões para trabalhos futuros.

7.1 Resumo de resultados

Este trabalho teve como principal objetivo o desenvolvimento de uma aplicação Web de planejamento de viagens para uso dos funcionários e dos visitantes do blog Planejando Viagens. Outros objetivos apresentados foram: facilitar a organização de informações durante o planejamento de viagens, possibilitar a consulta dos dados de forma prática durante e após a viagem e se tornar uma ferramenta eficiente de controle e organização de gastos durante a viagem.

Quanto ao objetivo principal de desenvolver uma aplicação Web de planejamentos de viagens, o mesmo foi atingido, visto que a aplicação foi desenvolvida conforme apresentado no capítulo 5. Ao realizar a análise do software com o uso das heurísticas de nielsen, visto na seção 6.1, o software demonstrou ser uma boa ferramenta para o planejamento de viagens.

A aplicação desenvolvida possibilita o agrupamento dos dados por categoria facilitando a busca de informações, além de organizar o roteiro e gastos tanto por dia quanto de forma macro. Essas funcionalidades foram apresentadas nas seções 5.3 e 5.4. Além disso, foi identificado na entrevista que a aplicação facilita a organização de informações, conforme apresentado na seção 6.3. Portanto, podemos concluir que o objetivo referente a organização de informações também foi alcançado.

A existência de uma timeline, apresentado na seção 5.3, para visualização do roteiro e dos custos depois de adicionados possibilita a consulta dos dados de forma prática durante e após a viagem, como era proposto em outro dos objetivos.

A aplicação também visava se tornar uma ferramenta eficiente de controle e de organização de gastos durante a viagem. Esse objetivo também foi atingido, visto que a aplicação possibilita adicionar todos os gastos da viagem por categorias, além de mostrar uma soma total do valor para facilitar o controle do usuário. Na tabela 7.1 é possível visualizar a comparação deste trabalho com os demais sistemas já existentes apresentados no capítulo 3. Sendo possível constatar que o controle de gastos é um dos principais pontos

fracos de grande parte dos sistemas similares analisados, a aplicação desenvolvida nesse trabalho acaba por se destacar nessa categoria. Além disso, o gerenciamento de gastos interligado ao roteiro e uma gestão de custos mais eficiente foram pontos identificados como pontos positivos na entrevista, conforme mostrado na seção 6.3.

Outro resultado do trabalho que não estava previsto nos objetivos iniciais foi o desenvolvimento de um sistema de mapas apresentado na seção 4.6. O sistema permite que o usuário visualize em um mapa os pontos definidos no seu roteiro previamente. Sendo possível pesquisar um país, uma cidade, uma rua, o nome de um estabelecimento, de uma atração turística, entre outros.

Durante a avaliação da aplicação realizada através de entrevista e apresentada na seção 6.3, foram identificados como principais pontos positivos: o gerenciamento de gastos interligado com o roteiro, a divisão de custos por categoria e a visualização no mapa dos lugares adicionados ao roteiro. Foi identificado que o impacto da aplicação na rotina de planejamento de viagens foi facilitar a organização de informações, a redução quase total da necessidade de diversas planilhas do Excel para organizar os dados de cada viagem e uma gestão de custos mais eficiente. Como pontos que não atenderam as expectativas foram citados o fato de não existir a possibilidade de exportar o conteúdo e nem de acessar a aplicação offline.

Além disso, também foi possível identificar, através da análise heurística aplicada na seção 6.1, que o sistema apresenta um baixo desempenho nas heurísticas de documentação, de prevenção de erros e de liberdade e controle do usuário. Sendo esses pontos que também devem ser aprimorados futuramente.

7.2 Resumo de resultados referentes as tecnologias utilizadas

No que tange as escolhas das tecnologias o uso do Vue foi uma ótima escolha, pois este possui uma boa comunidade e uma grande variedade de pacotes disponíveis via o NPM, tornando possível minimizar a curva de aprendizagem e de tempo de desenvolvimento.

As tecnologias de desenvolvimento do back-end demandaram bastante esforço de pesquisa para encontrar um framework que conseguisse suprir as necessidades das funcionalidades e que possuísse uma interface simples de gerenciamento. Assim permitindo que usuários sem conhecimento de programação conseguissem fazer uso apresentado na seção 4.5.

O Strapi é uma ferramenta que forneceu uma ótima experiência de desenvolvimento e de uso de sua aplicação, para criar a API utilizada neste trabalho. Junto com a biblioteca javascript para o front-end foi possível integrar ao sistema facilmente.

A utilização das API's do Google, facilitou o desenvolvimento do projeto abstraindo as funções de geolocalização e auto completar os endereços fornecidos pelo usuário como visto no seção 4.6. Isto juntamente com a linguagem Vue e com os pacotes existentes no NPM que fornecem componentes preparados para executar tais funcionalidades.

A avaliação realizada na seção 3.7 mostrou que a aplicação desenvolvida neste trabalho possui uma boa interface para uso com usuários e permitiu visualizar falhas no seu desenvolvimento que podem ser melhoradas futuramente.

7.3 Limitações do trabalho

Uma das limitações desse trabalho foi o desenvolvimento feito com base apenas nas necessidades do blog Planejando Viagens e não necessariamente nas necessidades do mercado. Outra limitação identificada foi que durante o desenvolvimento deste trabalho não foi possível realizar testes com mais de um usuário real para avaliar a qualidade da aplicação. Apenas a autora do blog Planejando Viagens e o próprio autor deste trabalho avaliaram a aplicação.

7.4 Trabalhos futuros

Tendo em vista que o projeto apresentado neste trabalho representa uma etapa inicial do sistema, existe ainda potencial para desenvolvimento de novas funcionalidades em trabalhos futuros, conforme abaixo:

- Documentação para auxílio dos usuários do sistema.
- Adicionar subcategorias de dados que podem ser inseridos nos roteiros.
- Possibilidade de exportação do roteiro para um arquivo de fácil acesso offline.
- Permitir que usuários compartilhem suas viagens.
- Aprimorar a prevenção de erros no sistema.
- Adicionar visualização dos gastos em forma de gráficos.

Tabela 7.1: Tabela de análise comparativa completa

Funcionalidades / Aplicação		Inspirock	Travefy	Google Trip	Trip30	Tripit	Travaa	Planejando Viagens
<i>Distribuição</i>	Versão gratuita	X	X	X	X	X	X	X
	Versão paga		X			X		
	Versão Web	X	X		X	X	X	X
	Versão mobile	X	X	X	X	X	X	X
<i>Construção</i>	Adicionar atrações	X	X	X	X	X	X	X
	Adicionar hospedagens	X	X	X	X	X	X	X
	Adicionar transporte	X	X	X	X	X	X	X
	Recomendações para o roteiro	X		X				
	Possível informar valores	X	X				X	X
	Facilidade para inserir dados ao roteiro	F	D	F	M	D	F	F
<i>Visualização</i>	Roteiro como timeline	X	X		X	X	X	X
	Roteiro como mapa	X	X	X	X		X	X
	Roteiro como calendário	X			X			
	Visualizar informações de gastos						X	X
	Exportar roteiro (Pdf, Doc)	X	X	X	X	X	X	
<i>Armazenamento</i>	Roteiro	X	X	X	X		X	X
	Compartilhamento	X	X	X	X		X	

Legenda: F = Fácil, M = Médio e D = Difícil

Fonte: elaborado pelo autor.

REFERÊNCIAS

- ADOBE. **What is Adobe XD?** 2018. Disponível em: <<https://helpx.adobe.com/xd/how-to/what-is-xd.html>>. Acesso em: 22 nov. 2018.
- HELABS. **O que é Scrum?** 2018. Disponível em: <<http://www.desenvolvimentoagil.com.br/scrum/>>. Acesso em: 22 nov. 2018.
- MONGODB. **Why Vuetify?** 2018. Disponível em: <<https://docs.mongodb.com/manual/introduction/>>. Acesso em: 20 nov. 2018.
- MOZILLA. **O que é JavaScript?** 2018. Disponível em: <https://developer.mozilla.org/pt-PT/docs/Web/JavaScript/O_que_é_o_JavaScript>. Acesso em: 30 out. 2018.
- NIELSEN, J. **Usability Engineering**. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993. ISBN 9780080520292.
- NIELSEN, J.; MOLICH, R. Heuristic evaluation of user interfaces. In: **Proceedings of the SIGCHI Conference on Human Factors in Computing Systems**. New York, NY, USA: ACM, 1990. (CHI '90), p. 249–256.
- NODE.JS. **About**. 2018. Disponível em: <<https://nodejs.org/en/about/>>. Acesso em: 31 out. 2018.
- ROESCH, S. M. A. **Projeto de estágio e de pesquisa em administração: guia para estágios, trabalhos de conclusão dissertações e estudos de caso**. São Paulo: Atlas, 2005.
- TABLELESS. **O que é JavaScript?** 2018. Disponível em: <<http://tableless.github.io/iniciantes/manual/js/>>. Acesso em: 14 nov. 2018.
- VUEJS. **Introdução Vue.js**. 2018. Disponível em: <<https://br.vuejs.org/v2/guide/>>. Acesso em: 8 nov. 2018.
- VUEJS. **Renderizando no Lado do Servidor**. 2018. Disponível em: <<https://br.vuejs.org/v2/guide/ssr.html>>. Acesso em: 8 nov. 2018.
- VUETIFY. **Why Vuetify?** 2018. Disponível em: <<https://vuetifyjs.com/pt-BR/getting-started/why-vuetify>>. Acesso em: 5 nov. 2018.

APÊNDICE A - ROTEIROS DE ENTREVISTA

1. Quais eram as principais dificuldades existentes antes da aplicação?
2. Considerando o que era esperado da aplicação, quais foram os principais pontos positivos identificados?
3. Quais os pontos não atenderam às expectativas e poderiam melhorar?
4. Qual a principal melhoria que o desenvolvimento dessa aplicação traz para a sua rotina de planejamento de viagens existente atualmente?
5. Quais pontos poderiam ser desenvolvidos para agregar valor à aplicação?