

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA PARAÍBA  
CAMPUS CAJAZEIRAS  
CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE  
SISTEMAS**

***ANÁLISE DE DADOS TURÍSTICOS COM BASE EM MÍDIAS SOCIAIS***

**DIONES ROCHA GOMES**

**Cajazeiras  
2024**

**DIONES ROCHA GOMES**

***ANÁLISE DE DADOS TURÍSTICOS COM BASE EM MÍDIAS SOCIAIS***

Trabalho de Conclusão de Curso apresentado junto ao Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas do Instituto Federal de Educação, Ciência e Tecnologia da Paraíba - Campus Cajazeiras, como requisito à obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas.

Orientador

Prof. Dr. Fabio Gomes de Andrade.

**Cajazeiras**

**2024**

IFPB / Campus Cajazeiras  
Coordenação de Biblioteca  
Biblioteca Prof. Ribamar da Silva  
Catalogação na fonte: Cícero Luciano Félix CRB-15/750

G633i Gomes, Diones Rocha.  
Análise de dados turísticos com base em mídias sociais / Diones Rocha Gomes.– 2024.  
  
60f. : il.  
  
Trabalho de Conclusão de Curso (Tecnólogo em Análise e Desenvolvimento de Sistemas) - Instituto Federal de Educação, Ciência e Tecnologia da Paraíba, Cajazeiras, 2024.  
  
Orientador(a): Prof. Dr. Fabio Gomes de Andrade.  
  
1. Desenvolvimento de sistemas. 2. Mineração de dados. 3. Mídias sociais. 4. Turismo. I. Instituto Federal de Educação, Ciência e Tecnologia da Paraíba. II. Título.

IFPB/CZ

CDU: 004.4(043.2)



SERVIÇO PÚBLICO FEDERAL  
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA PARAÍBA  
COORDENAÇÃO DO CURSO SUPERIOR EM ANÁLISE E DESENVOLVIMENTO DE  
SISTEMAS - CAMPUS CAJAZEIRAS



ATA 20/2024 - CADS/UNINFO/DDE/DG/CZ/REITORIA/IFPB

**ATA DE DEFESA DO TRABALHO DE CONCLUSÃO DE CURSO (TCC)  
CURSO: ANÁLISE E DESENVOLVIMENTO DE SISTEMAS (ADS)**

Às 15h00 do dia 30 do mês de JULHO do ano de 2024, o(a) **aluno(a) DIONES ROCHA GOMES**, matrícula **201612010180**, apresentou, como parte dos requisitos para obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas, seu trabalho de conclusão de curso, tendo como título "**Análise de dados turísticos com base em mídias sociais**". Constituíram a banca examinadora os professores **Fabio Gomes de Andrade** (orientador), **Diogo Dantas Moreira** (examinador) e **Antônio Ricart Jacinto de Oliveira Medeiros** (examinador).

Após a apresentação e as observações dos membros da Banca Examinadora, ficou definido que o trabalho foi considerado **APROVADO** com nota **95**, com a condição de que o (a) aluno (a) entregue, no prazo máximo de 30 dias, a versão final do trabalho com as correções sugeridas pelos membros da banca examinadora. Eu, Francisco Paulo de Freitas Neto, Coordenador do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, lavrei a presente ata, que segue assinada digitalmente por mim e pelos membros da banca examinadora.

Cajazeiras, 1 de agosto de 2024.

Documento assinado eletronicamente por:

- Francisco Paulo de Freitas Neto, COORDENADOR(A) DE CURSOS - FUC1 - CADS-CZ, em 01/08/2024 07:40:09.
- Diones Rocha Gomes, ALUNO (201612010180) DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS - CAJAZEIRAS em 01/08/2024 11:03:09.
- Fabio Gomes de Andrade, PROFESSOR ENS BASICO TECN TECNOLOGICO, em 01/08/2024 14:09:26.
- Antonio Ricart Jacinto de Oliveira Medeiros, PROF ENS BAS TEC TECNOLOGICO-SUBSTITUTO, em 02/08/2024 13:33:59.
- Diogo Dantas Moreira, PROFESSOR ENS BASICO TECN TECNOLOGICO, em 12/08/2024 14:45:01.

Este documento foi emitido pelo SUAP em 01/08/2024. Para comprovar sua autenticidade, faça a leitura do QRCode ao lado ou acesse <https://suap.ifpb.edu.br/autenticar-documento/> e forneça os dados abaixo:

Código 586022  
Verificador: 542ce08fc4  
Código de Autenticação:



*Dedico este trabalho à minha querida esposa, Cícera Henrique Gomes e ao meu amado filho, Asaph Henrique Gomes. Tê-los ao meu lado foi essencial para a conclusão deste trabalho. Grato a Deus por suas vidas.*

## AGRADECIMENTOS

A Deus, pela minha vida, pela saúde e pelas capacidades concedidas, que me permitiram dedicar-me ao curso e concluí-lo da melhor forma, superando todas as adversidades.

À minha esposa, Cícera Henrique, pelo amor, companheirismo e por sempre acreditar em minha capacidade, mesmo nos momentos mais desafiadores, nunca permitindo que eu desanimasse.

Ao meu pai, Vicente de Paulo Gomes Pereira, e à minha mãe, Gorete de Sousa Rocha Gomes, que sempre me deram apoio e confiaram em mim. Eles são os meus maiores exemplos de pessoas batalhadoras e de bom caráter, nunca permitindo que me faltasse algo desde o começo. Sem eles, eu não teria chegado a lugar algum.

À toda a instituição, bem como a todos os professores e funcionários, por terem proporcionado desde o início um ambiente propício e toda a estrutura necessária para a conclusão do curso.

Ao meu orientador, Professor Dr. Fábio Gomes de Andrade, expresso minha gratidão por toda a orientação fornecida ao longo do desenvolvimento do trabalho, desde a escolha do tema até as valiosas dicas, sugestões e correções, as quais foram fundamentais para sua conclusão.

Aos meus irmãos em Cristo, Ageu Ribeiro Cruz, Amanda Oliveira de Sousa Ribeiro, Maria Lúcia de Fátima Oliveira Sousa e Antônio Vicente de Sousa Neto, pelo constante apoio que me ofereceram aqui na cidade de Cajazeiras ao longo de todo o curso.

Aos meus amigos, Antoniel Damião e James Amarante que sempre me apoiaram e incentivaram a não desistir, saibam que sou muito grato a vocês.

Aos meus colegas de curso, cuja presença foi fundamental para uma experiência enriquecedora ao longo deste período. Sempre estiveram prontos para compartilhar conhecimentos e oferecer apoio nos momentos de desafio.

Meu mais sincero “**Muito Obrigado!**”

*"Não há um único centímetro quadrado, em todos os domínios de nossa existência, sobre o qual Cristo, que é soberano sobre tudo, não clame: É meu!"*

Abraham Kuyper

## RESUMO

O hábito de viajar está se tornando cada vez mais comum entre as pessoas. No entanto, para muitos turistas, decidir o destino de sua próxima viagem não é uma tarefa simples. Mesmo após a escolha do destino, a falta de informações sobre os locais a visitar, a localização dos principais pontos de interesse e o tempo necessário para explorar cada lugar podem resultar em experiências frustrantes. Para melhorar o planejamento de suas viagens, muitos turistas recorrem a fotos e vídeos compartilhados por outros viajantes que visitaram o destino desejado. No entanto, a análise dessas informações consome tempo e é difícil para uma pessoa processar uma grande quantidade de dados. Com o objetivo de superar essas limitações, este trabalho de conclusão de curso propõe uma ferramenta que extrai e analisa itinerários de turistas com base em fotos publicadas em mídias sociais. Além disso, a ferramenta auxilia os viajantes na escolha de seu próximo destino turístico, proporcionando uma visualização gráfica dos principais pontos turísticos de uma cidade ou região específica, juntamente com os itinerários mais comuns realizados por outros viajantes nesse local.

**Palavras-chave:** Análise de Dados. Mineração de Mídias Sociais. Análise de Trajetórias.

## **ABSTRACT**

The habit of traveling is becoming increasingly common among people. However, for many tourists, deciding on the destination for their next trip is not a simple task. Even after choosing the destination, the lack of information about places to visit, the location of main points of interest, and the time needed to explore each location can lead to frustrating experiences. To improve their travel planning, many tourists rely on photos and videos shared by other travelers who have visited the desired destination. However, analyzing this information takes time and it's challenging for an individual to process a large amount of data. With the aim of overcoming these limitations, this thesis proposes a tool that extracts and analyzes tourists' itineraries based on photos posted on social media. Furthermore, the tool assists travelers in choosing their next tourist destination by providing a graphical visualization of the main tourist attractions in a specific city or region, along with the most common itineraries followed by other travelers in that location.

**Keywords:** Data Analysis. Social Media Mining. Trajectory Analysis.

## LISTA DE FIGURAS

Figura 1 – Página de pesquisa do OSM . . . . .	18
Figura 2 – Página de download do Geofabrik . . . . .	19
Figura 3 – Resposta da requisição para recuperar fotos dentro de um bounding box . . . . .	22
Figura 4 – Resposta da requisição para recuperar os metadados de uma foto .	23
Figura 5 – Exemplo de código . . . . .	24
Figura 6 – Exemplificação da criação de um componente . . . . .	25
Figura 7 – Exemplificação da importação do componente . . . . .	26
Figura 8 – Resultado da utilização dos componentes . . . . .	27
Figura 9 – Exemplificação da criação de um repositório . . . . .	30
Figura 10 – Representação da arquitetura do sistema . . . . .	32
Figura 11 – Representação do esquema do banco de dados . . . . .	34
Figura 12 – Diagrama de atividade do fluxo de importação dos <i>shapefiles</i> . . . .	36
Figura 13 – Visualização parcial da tabela de lugares . . . . .	37
Figura 14 – Diagrama de sequência da comunicação entre os componentes da ferramenta de coleta e processamento de dados . . . . .	38
Figura 15 – Procedimento armazenado responsável pelo geotagging das fotos .	39
Figura 16 – Exemplo de itinerário gerado . . . . .	41
Figura 17 – Diagrama de sequência da comunicação entre front-end e o gerenci- amento da aplicação . . . . .	43
Figura 18 – Busca por pontos de interesse filtrando por alguma das categorias de lugares mais visitados . . . . .	44
Figura 19 – Busca pelos itinerários mais frequentes de uma região . . . . .	45
Figura 20 – Página inicial do sistema . . . . .	56
Figura 21 – Página de pesquisa do sistema . . . . .	56
Figura 22 – Página de galeria . . . . .	57
Figura 23 – Página de itinerários . . . . .	57

## LISTA DE ABREVIATURAS E SIGLAS

<i>API</i>	<i>Application Programming Interface</i>
<i>GPS</i>	<i>Sistema de Posicionamento Global</i>
<i>HTTP</i>	<i>Hypertext Transfer Protocol</i>
<i>JDK</i>	<i>Java Development Kit</i>
<i>JPA</i>	<i>Java Persistence API</i>
<i>JPQL</i>	<i>Java Persistence Query Language</i>
<i>JSON</i>	<i>Javascript Object Notation</i>
<i>Odbl</i>	<i>Open Data Commons Open Database License</i>
<i>OMT</i>	<i>Organização Mundial do Turismo</i>
<i>OSM</i>	<i>Open Street Map</i>
<i>OSMF</i>	<i>Open Street Map Foundation</i>
<i>PBF</i>	<i>Protocol Buffer Binary Format</i>
<i>REST</i>	<i>Representational State Transfer</i>
<i>SOAP</i>	<i>Simple Object Access Protocol</i>
<i>SQL</i>	<i>Structured Query Language</i>
<i>TCC</i>	<i>Trabalho de Conclusão de Curso</i>
<i>URL</i>	<i>Uniform Resource Locator</i>
<i>XML</i>	<i>Extensible Markup Language</i>

# SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>11</b>
1.1	Motivação	12
1.2	OBJETIVOS	14
<b>1.2.1</b>	<b>Objetivo Geral</b>	<b>14</b>
<b>1.2.2</b>	<b>Objetivo Específico</b>	<b>14</b>
1.3	TRABALHOS RELACIONADOS	14
1.4	Metodologia	15
1.5	ESTRUTURA E ORGANIZAÇÃO DO DOCUMENTO	16
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b>	<b>17</b>
2.1	OPENSTREETMAP	17
<b>2.1.1</b>	<b>Uso do OpenStreetMap</b>	<b>18</b>
2.2	<i>FLICKR</i>	19
<b>2.2.1</b>	<b>API do Flickr</b>	<b>20</b>
2.3	<i>NODE.JS</i>	24
2.4	<i>REACT.JS</i>	25
2.5	<i>JAVA</i>	27
2.6	<i>SPRING BOOT</i>	28
<b>2.6.1</b>	<b><i>SPRING DATA JPA</i></b>	<b>29</b>
2.7	<i>CONSIDERAÇÕES FINAIS DO CAPÍTULO</i>	30
<b>3</b>	<b><i>SOLUÇÃO PROPOSTA</i></b>	<b>31</b>
3.1	Análise	31
<b>3.1.1</b>	<b>Stakeholders</b>	<b>31</b>
<b>3.1.2</b>	<b>Requisitos funcionais</b>	<b>31</b>
3.2	ARQUITETURA DO SISTEMA	32
3.3	ESQUEMA LÓGICO DO BANCO DE DADOS	33

3.4	IMPLEMENTAÇÃO . . . . .	35
3.4.1	<b>Importação dos dados espaciais . . . . .</b>	<b>35</b>
3.4.2	<b><i>Ferramenta de coleta e processamento de dados . . . . .</i></b>	<b>37</b>
3.4.2.1	<i>O processo de coleta das fotos . . . . .</i>	37
3.4.2.2	<i>O processo de geotagging das fotos . . . . .</i>	39
3.4.2.3	<i>O processo de identificação dos itinerários . . . . .</i>	40
3.4.3	<b><i>Gerenciamento da aplicação . . . . .</i></b>	<b>41</b>
3.4.4	<b><i>Aplicação Web . . . . .</i></b>	<b>42</b>
4	<b>CONSIDERAÇÕES . . . . .</b>	<b>46</b>
	<b>REFERÊNCIAS . . . . .</b>	<b>48</b>
	<b>APÊNDICE A – DICIONÁRIO DE DADOS LÓGICO . . . . .</b>	<b>50</b>
	<b>APÊNDICE B – TELAS DA APLICAÇÃO WEB . . . . .</b>	<b>56</b>

# 1 INTRODUÇÃO

Ao longo da história, sempre existiu uma necessidade inata no ser humano de se relacionar com seus semelhantes. Essa forma de relacionamento se dá de maneira diversificada, principalmente no mundo contemporâneo. A necessidade de se conviver em grupo, com os mesmos ou diferentes interesses, pode render inúmeras vantagens, quando bem exploradas. Tal convívio pode se manifestar por meio de comunidades, ambientes escolares, esferas políticas, entre outras. Com o notório avanço da Internet e suas tecnologias ao longo dos anos, as mídias sociais têm ganhado um espaço amplo, resultando em uma grande aceitação por parte de seus usuários, uma vez que as mesmas dão suporte às redes sociais que conhecemos nos dias de hoje.

De acordo com dados divulgados pelo *Instituto de Ensino e Pesquisa* (INSPER, 2022) quase 5 bilhões de pessoas em todo o mundo foram registradas como usuárias regulares da Internet, representando cerca de 63% da população global. Este aumento tem sido acompanhado pelo crescimento significativo do uso das mídias sociais. Segundo o relatório "*Digital 2022: Global Overview Report*", divulgado pelo site *Datareportal* (DATAREPORTAL, 2022) em colaboração com a *We Are Social*<sup>1</sup> e *Hootsuite*<sup>2</sup>, aproximadamente 58,4% da população mundial já está ativa em plataformas de mídia social, totalizando 4,62 bilhões de pessoas conectadas em redes como *Facebook*<sup>3</sup>, *X*<sup>4</sup> (antigo Twitter) e *Instagram*<sup>5</sup>.

Com o expressivo crescimento das mídias sociais, a prática de fotografar e compartilhar imagens capturadas tornou-se uma rotina comum para muitos viajantes ao redor do mundo. Agora, em vez de manter álbuns físicos tradicionais, eles optam por compartilhar suas fotos em plataformas virtuais, como mídias sociais e sites de fotografia. A inserção das pessoas no mundo digital passou, também, a criar novos imaginários e, conseqüentemente, a exercer influência sobre as escolhas de destinos turísticos dos viajantes, conforme destacado por Ferrari e Gândara (2015).

Dessa forma, nota-se que muitas pessoas se inspiram em viagens realizadas por outras, as quais, por sua vez, tendem a registrar e compartilhar imagens dos locais visitados durante suas férias. Nesse contexto, as fotografias têm assumido um papel fundamental na criação e promoção da imagem de destinos turísticos, destacando suas

---

<sup>1</sup> <https://wearesocial.com/us/>

<sup>2</sup> <https://www.hootsuite.com/about>

<sup>3</sup> <https://about.fb.com/>

<sup>4</sup> <https://about.twitter.com/pt.html>

<sup>5</sup> <https://about.instagram.com/about-us>

características, qualidades e atrativos(MORETTI et al., 2016).

## 1.1 MOTIVAÇÃO

Em 2024, estima-se que o turismo global recupere completamente os níveis pré-pandemia, com um crescimento projetado de 2% acima dos níveis de 2019, conforme indicado pelo Barômetro Mundial do Turismo<sup>6</sup>.

O boletim da Organização Mundial do Turismo (OMT, 2024) destaca que a contribuição econômica total do turismo atingiu US\$ 3,3 trilhões em 2023, representando cerca de 3% do PIB mundial. A recuperação sustentada do setor é evidente nos indicadores, com a capacidade aérea internacional e a demanda de passageiros recuperando cerca de 90% dos níveis pré-pandêmicos até outubro de 2023, de acordo com os dados publicados pela Associação Internacional de Transporte Aéreo (IATA, 2023).

Além disso, as taxas globais de ocupação em estabelecimentos de hospedagem atingiram 65% em novembro de 2023, um aumento em relação aos 62% de novembro de 2022. Com uma estimativa de 1,3 bilhões de chegadas internacionais em 2023, o turismo internacional alcançou 88% do total pré-pandemia. A OMT projeta uma recuperação total do turismo internacional em 2024, com uma perspectiva positiva refletida no Índice de Confiança no Turismo da OMT, indicando que 67% dos profissionais de turismo esperam um desempenho melhor ou muito melhor em 2024 em comparação com 2023 (OMT, 2024). Esses dados apontam para uma retomada significativa do turismo, impactando positivamente as economias, o emprego e o crescimento em todo o mundo.

Atualmente, com os avanços tecnológicos, qualquer pessoa com um *smartphone* é capaz de capturar e registrar suas experiências fotográficas. Os dispositivos atuais têm a capacidade de capturar imagens com qualidade *FULL HD*, algo que anteriormente só era possível com câmeras profissionais. Além disso, os *smartphones* mais recentes são capazes até mesmo de gravar vídeos em 4K, oferecendo uma experiência cinematográfica diretamente na palma da mão do usuário. Essa facilidade de captura e compartilhamento tem contribuído significativamente para o aumento da quantidade de fotos e vídeos de viagens disponibilizados nas redes sociais.

Diante disso, com todos esses recursos disponíveis, pessoas podem observar as experiências de viagens compartilhadas por outros usuários por meio de mídias

<sup>6</sup> <https://www.unwto.org/un-tourism-world-tourism-barometer-data>

sociais, como *Flickr*<sup>7</sup>, *Instagram*<sup>8</sup>, *Facebook*<sup>9</sup> e *Google Fotos*<sup>10</sup>. Essa prática se tornou uma ferramenta útil na seleção de destinos e na elaboração de itinerários para futuras viagens. Por exemplo, hoje em dia, é comum que as pessoas examinem álbuns de viagem para identificar destinos populares e pontos turísticos mais procurados em determinadas regiões.

Apesar da possibilidade de escolher um destino turístico com base nas viagens de outros, a definição do local a ser visitado e do roteiro a ser seguido ainda é uma tarefa desafiadora para muitos. Um importante fator que dificulta a tomada desse tipo de decisão é que muitas fotos não incluem descrições detalhadas que identifiquem o lugar retratado. Outra limitação pode surgir devido à quantidade de álbuns analisada por cada pessoa. Como normalmente cada pessoa consegue analisar apenas uma pequena quantidade de álbuns, é possível que muitos lugares interessantes, mas pouco visitados, não sejam percebidos pelo turista.

Essas restrições frequentemente resultam em experiências frustrantes ou abaixo das suas expectativas para muitos turistas. Isso se deve ao fato de que, mesmo com um plano bem definido sobre o destino da viagem e os locais a serem visitados, eles muitas vezes não conseguem explorar todos os pontos de interesse ao redor dos lugares desejados, simplesmente porque desconhecem as preferências e as rotas percorridas pelos turistas que divulgaram as fotografias de suas viagens.

Para ilustrar a questão mencionada, considere-se o seguinte cenário: muitas pessoas aspiram conhecer a cidade de Paris, na França, renomada por seus emblemáticos pontos turísticos, como a *Torre Eiffel* e o *Palácio de Versailles*. Dado que o *Palácio de Versailles* está situado nos arredores de Paris, é plausível que durante o deslocamento até lá, haja diversos locais de interesse que os turistas poderiam aproveitar para explorar. No entanto, muitos podem não estar cientes de que o tempo necessário para o deslocamento e a visita ao Palácio pode reduzir bastante a possibilidade de visita de outros pontos de seu interesse.

Com o objetivo de resolver as limitações mencionadas, esse Trabalho de Conclusão de Curso (TCC) propõe uma ferramenta que analisa os dados turísticos, em especial as fotos publicadas em mídias sociais por turistas e viajantes, para auxiliar os viajantes no planejamento de suas viagens.

---

<sup>7</sup> <https://www.flickr.com/about>

<sup>8</sup> <https://about.instagram.com/about-us>

<sup>9</sup> <https://about.fb.com/>

<sup>10</sup> <https://photos.google.com/>

## 1.2 OBJETIVOS

Esta seção delinea os objetivos deste Trabalho de Conclusão de Curso (TCC).

### 1.2.1 Objetivo Geral

O presente trabalho de conclusão de curso tem como objetivo geral desenvolver uma ferramenta destinada a auxiliar os turistas na tarefa de planejar e definir o itinerário de suas viagens.

### 1.2.2 Objetivo Específico

Os objetivos específicos deste trabalho incluem:

- Desenvolver uma solução automatizada para associar as fotos de viagens aos locais onde foram capturadas;
- Criar uma solução para identificar, a partir das fotos identificadas, o itinerário realizado por cada viajante em um determinado destino;
- Gerar uma base de dados centralizada que sirva como base para o desenvolvimento de trabalhos futuros que abordem a recomendação de itinerários turísticos.

## 1.3 TRABALHOS RELACIONADOS

O presente TCC está relacionado a algumas aplicações disponíveis na internet. Um exemplo é o *TripAdvisor*<sup>11</sup>, um site de viagens que oferece informações e avaliações sobre diversos destinos turísticos, além de contar com fóruns interativos. Este site foi pioneiro na adoção de conteúdo gerado pelos usuários, que constituem a maior parte do seu conteúdo, e seu modelo de negócio é baseado principalmente em publicidade.

O *Local Guides*<sup>12</sup> é uma comunidade global de exploradores que contribuem com comentários, compartilham fotos, respondem a perguntas, adicionam ou editam lugares e verificam informações no *Google Maps*<sup>13</sup>. Por outro lado, o *Yelp*<sup>14</sup> é um guia urbano que opera como uma rede social. Os usuários compartilham suas experiências nos locais que visitaram, fornecendo breves críticas e relatos sobre restaurantes, hotéis, bares, praças e eventos.

---

<sup>11</sup> <https://www.tripadvisor.com.br/>

<sup>12</sup> <https://maps.google.com/localguides/>

<sup>13</sup> <https://www.google.com.br/maps/>

<sup>14</sup> <https://www.yelp.com.br/faq>

O *Google Travel*<sup>15</sup> é uma plataforma online desenvolvida e gerenciada pelo Google que permite aos usuários planejar viagens, oferecendo informações detalhadas a respeito de destinos turísticos, reserva de voos, hospedagem em hotéis e pacotes de viagem.

Tendo em vista todas essas aplicações que abordam o tema relacionado a viagens turísticas, conclui-se que nenhuma das aplicações mencionadas abordam a questão de itinerários turísticos. Por isso, justifica-se a proposição deste trabalho e desenvolvimento de uma ferramenta que possibilite a visualização dos principais pontos de interesse de uma determinada cidade ou região. Além de possibilitar ao usuário a visualização dos principais itinerários realizados por turistas em uma determinada localidade.

#### 1.4 METODOLOGIA

A metodologia usada para a implementação deste trabalho envolveu as seguintes etapas:

- **Estudo sobre o estado da arte:** Consistiu no levantamento dos principais trabalhos que se relacionam com a ferramenta proposta neste trabalho. É importante ressaltar que essa atividade foi realizada durante todo o desenvolvimento do trabalho, a fim de se manter este conhecimento sempre atualizado;
- **Análise e projeto do sistema:** Nesta fase, ocorreu a especificação do projeto, incluindo o levantamento dos requisitos funcionais, a proposição da arquitetura da ferramenta e a especificação do esquema de dados usado para a implementação do banco de dados;
- **Geração do banco de locais de interesse:** Esta etapa envolveu a implementação do banco de dados contendo os locais de interesse de algumas localidades. Para cada local de interesse, foram armazenados o seu nome, a sua categoria (museu, castelo, parque, cemitério, entre outros) e a geometria referente à sua localização;
- **Geração do banco de fotos:** Esta etapa consistiu na implementação do banco de dados contendo os metadados que descrevem fotos compartilhadas por turistas em uma mídia social de compartilhamento de fotos;
- **Geotagging das fotos:** Este processo consistiu em associar cada foto do banco de dados a uma localidade conhecida. Esse processo foi realizado de forma automatizada, por meio da análise das informações de geolocalização contidas nos metadados de cada foto;

---

<sup>15</sup> <https://www.google.com/travel/>

- **Identificação de itinerários:** Nesta etapa, foi implementado um módulo que utilizou os resultados obtidos através do processo de *geotagging* para identificar os itinerários percorridos pelos turistas responsáveis pela publicação das fotos coletadas;
- **Elaboração do documento final:** Por fim, ocorreu a elaboração do documento final do trabalho de conclusão de curso.

## 1.5 ESTRUTURA E ORGANIZAÇÃO DO DOCUMENTO

O trabalho está organizado em quatro capítulos. O capítulo dois aborda a fundamentação teórica utilizada ao longo do projeto. O capítulo três oferece uma descrição detalhada da solução proposta, descrevendo o processo de análise, levantamento de requisitos, arquitetura do sistema, modelo de dados e implementação dos módulos da ferramenta. Por último, o capítulo quatro apresenta as conclusões sobre as atividades desenvolvidas ao longo do trabalho, juntamente com as considerações finais pertinentes.

## 2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo, serão discutidos os conceitos, definições e tecnologias utilizadas durante todo o desenvolvimento deste trabalho. Inicialmente, serão apresentados o *OpenStreetMap* e a rede social *Flickr*. Em seguida, serão abordadas as tecnologias *NODE.js* e *REACT.js*, utilizadas para o desenvolvimento do back-end e do front-end da ferramenta. E por fim, serão apresentadas as tecnologias *Java* e *Spring Boot*.

### 2.1 OPENSTREETMAP

Segundo Haklay e Weber (2008), o *OpenStreetMap* (OSM) é um projeto colaborativo que tem como objetivo criar um mapa global gratuito e editável. Os dados geográficos subjacentes ao mapa são considerados a saída principal do projeto. Sua criação foi motivada por vários fatores, incluindo restrições no acesso ou na disponibilidade de dados de mapas em grande parte do mundo e pelo advento de dispositivos portáteis de navegação por satélite.

Criado por Steve Coast no Reino Unido em 2004, o projeto foi motivado pelo notável crescimento da *Wikipédia*, que oferecia um serviço semelhante, bem como pela predominância de dados de mapas proprietários tanto no Reino Unido quanto em outras regiões (LARDINOIS, 2014). Desde então, a base de dados do projeto tem crescido anualmente, com milhões de usuários registrados. Esses usuários têm a oportunidade de contribuir coletando dados através de pesquisas manuais, dispositivos GPS, fotografias aéreas e outras fontes disponíveis gratuitamente.

Por fim, é importante destacar que o projeto OSM é uma iniciativa de acesso gratuito, o que significa que disponibiliza seus dados geográficos sem custo algum. Qualquer pessoa tem a liberdade de utilizar esses dados para qualquer finalidade, contanto que atribua a autoria ao *OpenStreetMap* e seus contribuidores.

Atualmente, o OSM é regido pela licença da *Open Data Commons Open Database License* (ODbL)<sup>1</sup>. A Figura 1 ilustra a página de pesquisa do site do OSM.

<sup>1</sup> <https://opendatacommons.org/licenses/odbl/>



Figura 2 – Página de download do Geofabrik

## Extratos de dados do OpenStreetMap

Os arquivos de dados do OpenStreetMap fornecidos neste servidor não contêm os nomes de usuário, IDs de usuário e IDs de changeset dos objetos OSM porque esses campos são assumidos como contendo informações pessoais sobre os contribuidores do OpenStreetMap e, portanto, estão sujeitos a regulamentações de proteção de dados na União Europeia. Extratos com metadados completos estão disponíveis apenas para contribuidores do OpenStreetMap.

Bem-vindo ao servidor de download gratuito da Geofabrik. Este servidor tem extratos de dados do projeto OpenStreetMap que são normalmente atualizados todos os dias. Selecione seu continente e então seu país de interesse na lista abaixo. (Se você foi direcionado para esta página de outro lugar e não está familiarizado com o OpenStreetMap, recomendamos fortemente que você leia sobre o OSM antes de usar os dados.) Este serviço de download de dados abertos é oferecido gratuitamente pela Geofabrik GmbH.

Willkommen im Geofabrik-Downloadserver. Aquí estão os dados atualizados do projeto OpenStreetMap, que normalmente são atualizados. Wählen Sie aus dem Verzeichnis unten den Kontinent und ggf. das Land, für die Sie Daten benötigen. (Se você estiver nesta página e no OpenStreetMap não souber, isso é uma vantagem, se o projeto for implementado, antes de usar os dados.) Esses downloads são gratuitos para a Geofabrik GmbH.

Clique no nome da região para ver a página de visão geral dessa região ou selecione um dos links de extensão de arquivo para acesso rápido.

Sub-região	Links rápidos		
	.osm.pbf	.shp.zip	.osm.bz2
<a href="#">África</a>	<a href="#">[osm.pbf]</a> (6,5 GB)	X	<a href="#">[osm.bz2]</a>
<a href="#">Antártica</a>	<a href="#">[osm.pbf]</a> (31,4 MB)	<a href="#">[shp.zip]</a>	<a href="#">[osm.bz2]</a>
<a href="#">Ásia</a>	<a href="#">[osm.pbf]</a> (13,1 GB)	X	<a href="#">[osm.bz2]</a>
<a href="#">Austrália e Oceania</a>	<a href="#">[osm.pbf]</a> (1,2 GB)	X	<a href="#">[osm.bz2]</a>
<a href="#">América Central</a>	<a href="#">[osm.pbf]</a> (674 MB)	X	<a href="#">[osm.bz2]</a>
<a href="#">Europa</a>	<a href="#">[osm.pbf]</a> (28,9 GB)	X	<a href="#">[osm.bz2]</a>
<a href="#">América do Norte</a>	<a href="#">[osm.pbf]</a> (14,8 GB)	X	<a href="#">[osm.bz2]</a>
<a href="#">América do Sul</a>	<a href="#">[osm.pbf]</a> (3,3 GB)	X	<a href="#">[osm.bz2]</a>

[Detalhes técnicos](#) sobre este serviço de download.

Fonte: Elaborada pelo autor, (2024).

A *Geofabrik* é uma companhia alemã que mantém uma parceria com a OSMF. Sua plataforma funciona como um servidor gratuito que facilita a extração de dados do OSM. Os dados disponíveis nesse servidor são atualizados manualmente diariamente. É importante destacar que a *Geofabrik* oferece a opção de extração de dados em três diferentes formatos: *XML*, *PBF* e *shapefiles*. Essa variedade é importante, pois amplia o acesso aos dados do OSM para um público mais diversificado. Adicionalmente, é importante mencionar que o próprio OSM possibilita a exportação de regiões no formato *XML*.

Os dados utilizados neste projeto foram extraídos no formato *shapefile*, visando uma maior praticidade na hora de importar os dados na base de dados usada para a sua implementação. Para a realização da análise proposta neste trabalho de conclusão de curso, optou-se por selecionar os dados referentes às cidades de Paris, na França, e Rio de Janeiro, no Brasil.

## 2.2 FLICKR

O *Flickr*<sup>4</sup> é uma plataforma de hospedagem e compartilhamento de imagens, incluindo fotografias, desenhos e ilustrações, que oferece recursos avançados para organização de fotos e vídeos. Caracterizado como uma mídia social, permite aos usuários criar álbuns para armazenar suas fotografias e interagir com outros usuários da plataforma. Em 2018, o *Flickr* foi adquirido pela *SmugMug*<sup>5</sup>, uma empresa independente

<sup>4</sup> <https://www.flickr.com/about>

<sup>5</sup> <https://www.smugmug.com/>

especializada em hospedagem e compartilhamento de imagens, que se tornou sua proprietária atual. Anteriormente, o *Flickr* fazia parte do *Yahoo!* (GIBBS, 2019).

A plataforma *Flickr* organiza e classifica as fotos por meio de categorias, conhecidas como *tags* (ou etiquetas), que são atribuídas pelos próprios usuários ao carregarem as fotografias na plataforma. Isso facilita significativamente o processo de recuperação de fotos, tornando-o rápido e eficiente. Além disso, os usuários podem organizar suas próprias fotos em álbuns, permitindo a criação de coleções personalizadas.

O *Flickr* também disponibiliza uma *API* que permite o acesso de forma programática às informações sobre as fotos e vídeos que foram carregados em sua plataforma pelos seus usuários. No entanto, o acesso está sujeito ao cumprimento de requisitos e restrições estabelecidos pelos proprietários das fotos.

### 2.2.1 API do Flickr

Os dados utilizados na implementação deste trabalho são provenientes da rede social *Flickr* e foram obtidos mediante a utilização da sua *API*, que disponibiliza os dados sob a licença da empresa *SmugMug*.

A *API* do *Flickr* oferece uma ampla gama de métodos para que os usuários que compartilham suas fotos na plataforma possam gerenciar suas informações e integrá-las a outras aplicações. Além disso, ela disponibiliza operações que permitem que programas realizem consultas e recuperem os metadados das fotos compartilhadas pelos usuários da rede.

Nesta seção, abordaremos exclusivamente as operações da *API* que foram utilizadas na implementação deste trabalho. No entanto, informações detalhadas sobre todas as operações disponíveis podem ser acessadas na página da *API*, denominada *App Garden*<sup>6</sup>.

Para utilizar a *API* do *Flickr*, é necessário primeiro obter uma chave de acesso que permite o seu acesso. Em seguida, as requisições podem ser feitas por meio de uma requisição *HTTP* aos endpoints das operações que se deseja executar. Em cada solicitação, deve ser informado o método a ser invocado, os valores dos parâmetros que devem ser utilizados e o formato no qual a resposta deve ser enviada.

A *API* do *Flickr* disponibiliza uma operação chamada "search", que possibilita a recuperação de fotos georreferenciadas tiradas em uma determinada localidade. Essa operação está contida no pacote "photos". Para executá-la, é necessário fornecer

---

<sup>6</sup> <https://www.flickr.com/services/api/>

um *Bounding Box*, que representa o menor retângulo que cobre a região desejada. Esse *Bounding box* é definido por quatro coordenadas: a menor longitude (minX), a menor latitude (minY), a maior longitude (maxX) e a maior latitude (maxY). Além das informações de geolocalização, outros parâmetros opcionais podem ser utilizados com a finalidade de tornar a busca pelas fotos mais precisa.

Para invocar a operação de busca (*search*), observa-se que os seguintes parâmetros foram fornecidos para executar essa operação:

- **method**: representa o método que deve ser invocado;
- **api\_key**: representa a chave de acesso da aplicação que está fazendo a invocação;
- **min\_taken\_date**: representa a data mínima de postagem da foto. Quando este parâmetro é informado, o método retorna as fotos registradas cuja a data em que foram capturadas é igual ou superior à data passada como parâmetro. O valor correspondente ao parâmetro pode ser informado usando o padrão ISO 8601/padrão SQL<sup>7</sup>;
- **max\_taken\_date**: representa a data máxima de captura da foto. Quando ele é informado, o método retorna as fotos registradas cuja a data seja igual ou inferior a data referente ao parâmetro. O valor correspondente ao parâmetro também pode informado usando o padrão SQL;
- **bbox**: representa os quatro valores separados por vírgula que compõem o menor retângulo que cobre a região geográfica desejada que é composta pelos seguintes valores: Longitude\_minima, Latitude\_minima, Longitude\_maxima, Latitude\_maxima;
- **accuracy**: representa o nível de precisão registrado das informações de localização. O intervalo de números correspondente ao parâmetro é de 1 até 16. O valor máximo é o padrão deste parâmetro;
- **has\_geo**: representa o parâmetro que indica que deve ser retornada apenas as fotos que possuem informações de geolocalização. O número 1 indica as fotos que possuem gerenciamento. O número 0, as fotos que não possuem georreferenciamento;
- **format**: representa o parâmetro correspondente ao formato de resposta a ser recebido. A resposta da requisição pode ser retornada nos formatos *JSON*, *JSOP*, *XML (REST)* e *PHP serial*.

<sup>7</sup> <https://tecspace.com.br/paginas/aula/postgresql/guia/datatype-datetime.html>

A *API* retorna, como resposta à invocação, uma lista paginada que contém informações sobre as fotos que satisfazem todas as restrições definidas na requisição. Na Figura 3, é apresentado um trecho do arquivo *JSON* recebido como resposta da requisição.

**Figura 3 – Resposta da requisição para recuperar fotos dentro de um bounding box**

```
1  {
2    "photos": {
3      "page": 1,
4      "pages": 186,
5      "perpage": 250,
6      "total": 46307,
7      "photo": [
8        {
9          "id": "53920365050",
10         "owner": "63281384@N00",
11         "secret": "f2300fcc82",
12         "server": "65535",
13         "farm": 66,
14         "title": "Detalhe do Interior da Capela Inferior de Sainte-Chapelle",
15         "ispublic": 1,
16         "isfriend": 0,
17         "isfamily": 0
18       },
19       {↔},
30      {↔},
```

Fonte: Elaborada pelo autor, (2024).

A resposta à requisição, conforme ilustrado na Figura 3, é uma lista paginada em formato *JSON*, que contém as principais informações necessárias para consultar os metadados de cada foto presente na lista. Por meio dessa lista, é possível acessar dados como a identificação do proprietário da foto, o título atribuído a ela pelo proprietário e, especialmente, a identificação da própria foto, que é o parâmetro primordial para acessar os metadados da foto.

A *API* oferece a capacidade de recuperar os metadados detalhados de cada foto compartilhada na rede. Esse processo é realizado através da operação *getInfo*, que está disponível no pacote "photos". Para invocar essa operação, é necessário informar os seguintes parâmetros:

- **method:** representa o método que deve ser invocado;
- **api\_key:** representa a chave de acesso da aplicação que está fazendo a invocação;
- **photo\_id:** representa o número de identificação da foto da qual se deseja recuperar os metadados;
- **format:** representa o parâmetro correspondente ao formato de resposta a ser recebido. A resposta da requisição pode ser retornada nos formatos *JSON*, *JSOP*, *XML (REST)* e *PHP serial*.

A API retorna, como resposta à invocação, um arquivo *JSON* contendo as principais informações relacionadas aos metadados das fotos. A Figura 4 mostra o arquivo *JSON* retornado como resposta da requisição.

**Figura 4 – Resposta da requisição para recuperar os metadados de uma foto**

```

1  {
2  "photo": {
3    "id": "49684770906",
4    "secret": "11cde10fa4",
5    "server": "65535",
6    "farm": 66,
7    "dateuploaded": "1584836174",
8    "isfavorite": 0,
9    "license": "0",
10   "safety_level": "0",
11   "rotation": 0,
12   "originalsecret": "6107cb1d84",
13   "originalformat": "jpg",
14   "owner": {
15     "nsid": "110676687@N06",
16     "username": "W & V",
17     "realname": "Washington & Vinicius",
18     "location": "Madri, Espanha",
19     "iconserver": "4184",
20     "iconfarm": 5,
21     "path_alias": null,
22     "gift": {}
23   },
24   "title": {
25     "_content": "versales"
26   },
27   "description": {}
28 }

```

Conforme ilustrado na Figura 4, a resposta à requisição é um documento *JSON* que contém os metadados da foto cujo identificador foi fornecido como parâmetro na requisição. Este documento fornece informações como a identificação da foto e detalhes relacionados ao proprietário, como nome e localização. Além disso, inclui informações sobre a localização da foto, como o lugar onde foi tirada, entre outros.

### 2.3 *NODE.JS*

Segundo Tilkov e Vinoski (2010), o *Node.js* é uma tecnologia desenvolvida por Ryan Dahl em 2009, que utiliza como base a linguagem *Javascript*. Esta tecnologia capacita a criação de aplicações capazes de lidar com um grande volume de acessos de usuários. O *Node.js* é orientado a eventos e o processamento de requisições é não bloqueante, o que permite que operações que bloqueiam a execução do programa, como operações de *entrada e saída de dados*, tornem-se não bloqueantes.

Segundo uma pesquisa realizada pela *Node.js Foundation*<sup>8</sup> em 2016, 96% dos projetos de IoT foram desenvolvidos utilizando *JavaScript/Node*. Desde sua criação, o *Node.js* tem sido cada vez mais adotado por diversas empresas para o desenvolvimento de uma ampla gama de projetos, desde aplicações simples até sistemas complexos de comércio eletrônico. Empresas como *Walmart*, *Paypal* e *Netflix* estão entre as que utilizam essa tecnologia. A Figura 5 mostra um exemplo simples de código para criação de um servidor *HTTP*.

Figura 5 – Exemplo de código

```
1 const http = require('http');
2
3 const hostname = '127.0.0.1';
4 const port = 3000;
5
6 const server = http.createServer((req, res) => {
7   res.statusCode = 200;
8   res.setHeader('Content-Type', 'text/plain');
9   res.end('Hello World');
10 });
11
12 server.listen(port, hostname, () => {
13   console.log(`Server running at http://${hostname}:${port}/`);
14 });
```

Fonte: Elaborada pelo autor, (2024).

<sup>8</sup> <https://www.treinaweb.com.br/blog/node-js-por-que-voce-deve-conhecer-essa-tecnologia>

A Figura 5 apresenta a definição das configurações para a criação do servidor *HTTP*. Neste arquivo são definidas algumas configurações básicas, como o *host*, que é um endereço privado que os computadores utilizam para se referirem a si mesmos, e a porta, que é um número utilizado pelos servidores como ponto de extremidade para o endereço *IP*. Depois de realizar essas configurações, é possível acessar o servidor através da *url*: `http://localhost:3000` em um navegador local.

## 2.4 REACT.JS

Conforme mencionado por Fedosejev (2015), o *React.js* é uma biblioteca desenvolvida pelo *Facebook* em 2011 e disponibilizada como código aberto em 2013. Desde então, tornou-se uma das bibliotecas *JavaScript* mais populares entre os desenvolvedores. O *React* opera como um conjunto de funções padronizadas que os desenvolvedores podem utilizar para construir aplicações, principalmente focadas no desenvolvimento de interfaces com usuários. Atualmente, diversas aplicações renomadas são desenvolvidas com ele, incluindo o próprio *Facebook*, *Instagram*, e o serviço de *streaming Netflix*, entre outros (SILVA, 2021).

O *React* apresenta um conceito de componentes que possibilita a modularização da interface com o usuário, sendo possível criar comportamentos e atributos específicos para cada componente de forma individual e independente. Além de ser de fácil compreensão, esse conceito reduz significativamente a quantidade de código necessário, o que acelera o desenvolvimento de aplicações. Um exemplo de modelo de componente pode ser visto na Figura 6.

Figura 6 – Exemplificação da criação de um componente

```
1 import React from 'react';
2
3 export default function myButton(props) {
4   return (
5     <button style={props.estilo}>
6       <text>{props.titulo}</text>
7     </button>
8   );
9 }
10
```

Na Figura 6, o arquivo apresenta um exemplo simples de criação de um componente denominado *MyButton*. Esse componente aceita um único argumento de objeto chamado *props*, que representa suas propriedades. Em seguida, ele retorna um elemento *React* que descreve o conteúdo a ser exibido na tela.

Na Figura 7, é apresentado o processo de importação e implementação de um arquivo contendo um modelo de componente. A partir desse componente importado, é possível criar várias instâncias e passar atributos específicos para cada uma delas, tanto funções quanto estilização.

**Figura 7 – Exemplificação da importação do componente**

```
1 import React from 'react';
2 import MyButton from './component/myButton';
3
4 export default function App(props) {
5   return (
6     <div>
7       <MyButton estilo={{ backgroundColor: 'red' }} titulo= 'Botão Vermelho.'/>
8       <MyButton estilo={{ backgroundColor: 'green' }} titulo= 'Botão Verde.'/>
9       <MyButton estilo={{ backgroundColor: 'blue' }} titulo= 'Botão Azul.'/>
10    </div>
11  );
12 }
13
```

Fonte: Elaborada pelo autor, (2024).

Na Figura 7, é possível visualizar como é realizada a importação de um componente no arquivo principal da aplicação. O código exibido cria três instâncias do componente, passando atributos diferentes através do *props* para cada uma delas. No exemplo, são passadas diretrizes de estilização para cada instância.

Por fim, na Figura 8, é demonstrado o resultado da aplicação do conceito de componente, evidenciando a simplicidade e facilidade de implementação desse modelo.

Figura 8 – Resultado da utilização dos componentes



Fonte: Elaborada pelo autor, (2024).

Conforme ilustrado na Figura 8, o componente é renderizado na tela de acordo com as diretrizes especificadas através do *props*. Este exemplo simples demonstra como os componentes são definidos e funcionam no *React*, evidenciando como essa abordagem agiliza e simplifica o desenvolvimento de aplicações web.

## 2.5 JAVA

Java é linguagem de programação e ambiente computacional criado pela *Sun Microsystems* na década de 90. Se tornou popular graças à possibilidade de escrever o código apenas uma vez e executá-lo em diferentes dispositivos, passando a ser implementada em praticamente qualquer aplicação, desde sites, celulares e videogames (MELO, 2020).

O código é baseado em classes e orientado a objetos, com seu foco voltado para segurança, portabilidade e alto desempenho. Tendo como característica uma sintaxe similar a C/C++, possuindo uma extensa biblioteca de rotinas e *APIs* para trabalhar com recursos de rede, e também um poderoso gerenciamento de memória de forma automática.

O código Java é compilado em *bytecode*, que então é interpretado e executado pela Máquina Virtual Java (JVM). Dessa forma o sistema, ou aplicação em Java se torna portátil, podendo ser executado em praticamente qualquer ambiente ou dispositivo no qual a JVM esteja instalada. Essa versatilidade é útil pois permite reutilizar códigos de um ambiente para outro sem se preocupar com incompatibilidades.

Composta por um grande número de tecnologias, a plataforma se divide em ambiente de desenvolvimento e de execução do software. Existe três plataformas que englobam as ferramentas necessárias para criação e execução de software e sistemas,

a máquina virtual, o kit de desenvolvimento *JDK*, o compilador e outras ferramentas utilitárias (MELO, 2020).

Atualmente, a linguagem Java pertence a Oracle e tem seu destaque no desenvolvimento de aplicações corporativas, sendo atualmente uma das linguagens mais utilizadas no mundo, como demonstra a pesquisa State of Developer 2021 (JETBRAINS, 2021). Ademais, em uma pesquisa realizada, em agosto de 2024 pela TIOBE Index, aponta a linguagem Java como a quarta linguagem de programação no ranking (TIOBE, 2024).

## 2.6 *SPRING BOOT*

O *Spring Boot* é um *framework* Java que tem como objetivo facilitar a etapa de criação e configuração de uma aplicação, o que anteriormente era um processo demorado, pois exigia a manipulação de vários arquivos e dependências. Este traz em seu núcleo o conceito de injeção de dependência.

Os *frameworks* que fazem injeção de dependência, assim como o *Spring*, possibilitam o fácil gerenciamento de projetos grandes expondo o relacionamento entre objetos dentro da aplicação através de convenções e anotação, ao invés dos objetos possuírem um alto acoplamento entre eles.

O *Spring* atua como mediador entre as diferentes classes da aplicação e gerencia as dependências do projeto. Utilizando o *framework Spring Boot*, os desenvolvedores de aplicações podem criar aplicações com maior agilidade e desenvolver as funcionalidades necessárias com a menor preocupação com as configurações do *Spring*. Dentre as principais características do *Spring Boot* temos (GUTIERREZ, 2016):

- Facilidade de rapidez para criação de aplicações;
- Autoconfiguração;
- Integração de dependências automaticamente;
- Gerenciamento de configurações eficaz;
- Suporte em contêiner para Servlet embutido;
- Inicialização da aplicação através de uma classe *SpringApplication*;

Além disso, o *Spring Boot* possui um servidor de aplicação *Web* embutido (*Web Server*), podendo ser configurado e, ao executar a aplicação, ele é iniciado automaticamente, implantando e realizando a inicialização da aplicação com suas respectivas configurações. O servidor de aplicação nativo ao *Spring Boot* é o *Apache Tomcat*. Mas pode ser substituído por outros servidores de aplicação diferentes, tais como *Jetty* e *Undertow* (WEBB, 2017).

### 2.6.1 **SPRING DATA JPA**

O *Spring Data JPA* é um *framework* que o *Spring* utiliza para facilitar o desenvolvimento de aplicações que utilizam persistência de dados. Este módulo do *Spring* permite que os desenvolvedores criem métodos de consulta personalizados com base no nome do método, e também disponibiliza métodos para realizar operações de *CRUD* (Create, Read, Update e Delete) no banco de dados.

Uma das características mais brilhantes do *Spring Date JPA* é a capacidade de criar repositórios de forma automática. Repositórios no *Spring* são interfaces que podem ser definidas para acessar dados. Eles disponibilizam métodos comuns para persistência e podem criar *queries* a partir do nome de métodos personalizados em

tempo de execução. Ademais, para *queries* mais complexas, pode-se também defini-las utilizando *Java Persistence Query Language* (JPQL) ou SQL nativa com a anotação `@Query` acima do método. A Figura 9 mostra uma interface que apresenta um repositório para a classe *Exemplo*.

Figura 9 – Exemplificação da criação de um repositório

```
1 package com.example.repository;
2
3 > import ...
4
5
6
7 @Repository
8 public interface RepositorioExemplo extends CrudRepository<Exemplo, Long> {
9
10 }
```

Fonte: Elaborada pelo autor, (2024).

Após a criação do repositório conforme ilustrado na Figura 9, já é possível realizar operações no banco. A anotação `@Repository`, indica que uma classe possui papel *Data Access Object* (DAO). *CrudRepository*, como o nome sugere, fornece funcionalidades de *CRUD* para a entidade especificada no tipo genérico esperado, seguida pelo tipo da sua chave primária.

## 2.7 CONSIDERAÇÕES FINAIS DO CAPÍTULO

Este capítulo proporcionou uma compreensão abrangente das principais tecnologias utilizadas no desenvolvimento deste projeto. Inicialmente, foi discutido o papel do *OpenStreetMap* e sua relevância para este trabalho. Em seguida, explorou-se a plataforma de mídia social *Flickr*, a partir da qual foram coletados os dados utilizados neste trabalho. Também foram apresentadas as tecnologias *Node.js* e *React.js*, responsáveis pela implementação do *back-end* e *front-end* da ferramenta, respectivamente. Por fim, foram apresentadas as tecnologias *Java* e *Spring Boot*, utilizadas no processo de implementação do módulo de coleta e processamento de dados. O próximo capítulo descreverá a implementação do sistema proposto neste trabalho de conclusão de curso.

## 3 SOLUÇÃO PROPOSTA

Este capítulo apresenta de forma detalhada o desenvolvimento da solução proposta neste TCC, que visa a implementação de uma ferramenta para a análise de itinerários turísticos a partir de dados extraídos de redes sociais de compartilhamento de fotos. Inicialmente, são abordados os procedimentos de análise, onde são delineados os requisitos funcionais que devem ser oferecidos pela ferramenta. Em seguida, é explorado o processo de implementação da ferramenta, detalhando os componentes da arquitetura do sistema e o esquema de dados que foi utilizado para a implementação da sua base de dados, além dos detalhes de implementação de cada componente.

### 3.1 ANÁLISE

Este tópico apresenta os resultados obtidos na etapa de análise de requisitos proposta neste TCC, sendo dividido em sub tópicos: Stakeholders e Requisitos funcionais.

#### 3.1.1 Stakeholders

Os *stakeholders* (as partes interessadas) identificados para a ferramenta em questão são dois atores. Um deles é o usuário comum, que pode ser qualquer pessoa interessada em informações sobre pontos turísticos e roteiros turísticos. O outro ator são estabelecimentos comerciais como: *shoppings*, restaurantes, padarias, bares, entre outros. Como a ferramenta aborda os principais pontos de interesse de uma determinada localidade, isso pode influenciar os proprietários a criarem diferenciais, a fim de transformarem os seus estabelecimentos em um ponto de interesse em sua respectiva localidade.

#### 3.1.2 Requisitos funcionais

Definidos os atores, foram estabelecidos três requisitos funcionais para a solução proposta. São eles:

- **Exibição de locais (RF 01):** a ferramenta deve possibilitar ao usuário visualizar os pontos de interesse em uma determinada cidade ou região específica. Ao utilizar essa funcionalidade, o usuário deve ser capaz de filtrar os locais apresentados de acordo com uma categoria de seu interesse, como igrejas, museus, parques, castelos, entre outros;
- **Exibição dos locais mais visitados (RF 02):** a ferramenta deve permitir ao usuário visualizar os pontos turísticos mais frequentados por visitantes em uma

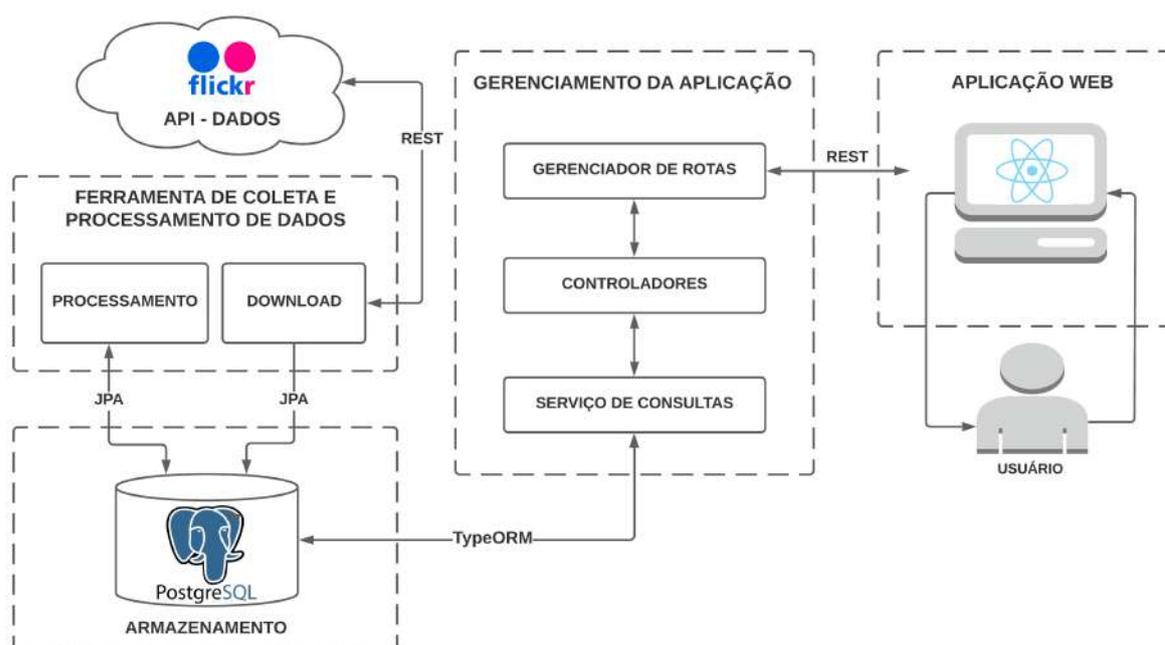
cidade ou região determinada. Semelhantemente ao requisito RF 01, a ferramenta deve oferecer a opção de filtrar os locais por categorias específicas;

- **Exibição dos itinerários mais frequentes (RF 03):** a ferramenta possibilitará ao usuário visualizar os itinerários mais frequentes realizados por turistas em uma cidade ou região específica.

### 3.2 ARQUITETURA DO SISTEMA

Esta seção tem como objetivo discutir a arquitetura da ferramenta, detalhando seus módulos, componentes e seus comportamentos correspondentes. Os módulos são definidos como elementos do sistema responsáveis por executar tarefas específicas. A arquitetura utilizada para a implementação da ferramenta está descrita na Figura 10.

Figura 10 – Representação da arquitetura do sistema



Fonte: Elaborado pelo autor, (2024).

A arquitetura compreende três aplicações distintas. A primeira delas é a ferramenta de coleta e processamento de dados, encarregada de automatizar o *download* dos metadados das fotos disponíveis no *Flickr*, os quais são utilizados para alimentar o banco de dados. Essa aplicação é constituída por um módulo com dois submódulos.

O primeiro submódulo, chamado de *download*, automatiza o processo de extração e persistência dos metadados das fotos. Já o segundo submódulo, denominado *proces-*

*samento*, tem a função de processar os metadados das fotos previamente armazenados no banco de dados da ferramenta, com o objetivo de gerar o itinerário individual de cada usuário.

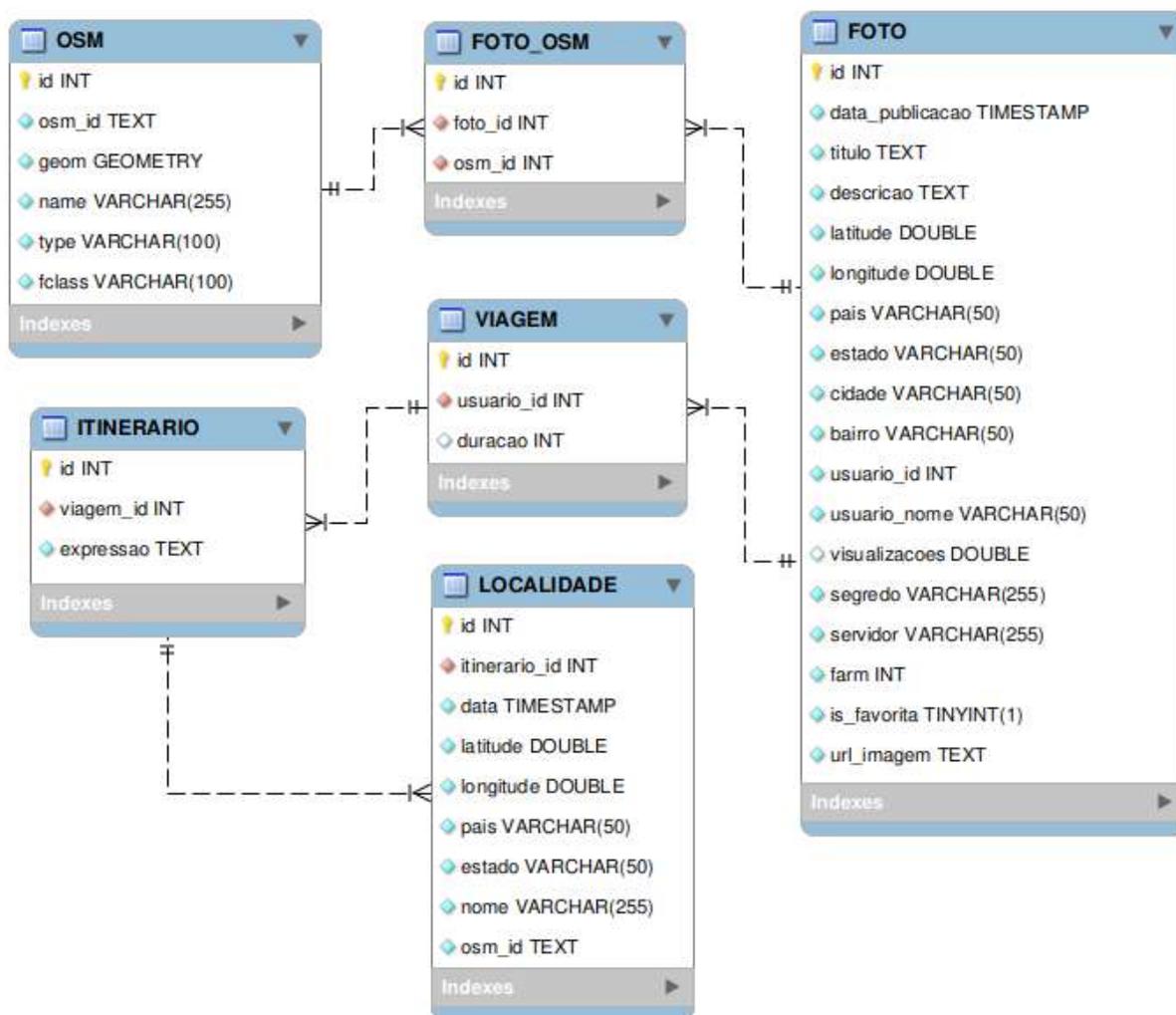
A segunda aplicação consiste no gerenciamento da aplicação, encarregada de receber e enviar as requisições realizadas pela aplicação de visão. Esta aplicação possui um módulo com três submódulos. O primeiro submódulo, chamado de *gerenciamento de rotas*, tem a responsabilidade de definir os métodos *HTTP* por meio da criação dos *endpoints*. O segundo submódulo, denominado *controladores*, atua como intermediário na comunicação entre o gerenciamento de rotas e o serviço de consultas, realizando o tratamento das informações antes de encaminhá-las para o submódulo específico. Por fim, o terceiro submódulo, o *serviço de consultas*, é responsável por gerenciar a conexão com o banco de dados, realizando as consultas e retornando os dados conforme os parâmetros predefinidos.

A terceira é a aplicação *web*, encarregada de disponibilizar os dados de forma sistematizada através de páginas *web*. Cada página é estruturada conforme o tipo de informação a ser apresentada, com o intuito de garantir a melhor forma de se apresentar os dados para o usuário final.

### 3.3 ESQUEMA LÓGICO DO BANCO DE DADOS

O esquema utilizado na implementação do banco de dados está descrito na Figura 11. Este esquema concentra-se nos metadados das fotos compartilhadas pelos usuários da plataforma de mídia social *Flickr*, juntamente com os dados de geolocalização das cidades utilizadas como estudo de caso, sendo Paris, na França, e Rio de Janeiro, no Brasil. Os detalhes sobre os pontos de interesse nessas cidades foram obtidos através do *OpenStreetMap*.

Figura 11 – Representação do esquema do banco de dados



Fonte: Elaborado pelo autor, (2024).

Na Figura 11, é possível visualizar o esquema lógico do banco de dados, composto pelas seguintes tabelas:

- **Foto**: Esta tabela é responsável por armazenar os metadados das fotos. Para cada foto coletada, registram-se informações como data de publicação, título, país, estado, cidade, bairro, latitude, longitude, identificação e nome do usuário. Importante ressaltar que essa tabela é desnormalizada, a fim de se alcançar um melhor desempenho nas consultas;
- **Osm**: Esta tabela armazena as informações sobre os pontos de interesse existentes em cada cidade. Para cada local, são armazenados o código, as informações de geometria (que representam a localização), o seu identificador no *OpenStreet-Map*, a categoria, o nome e o tipo de construção;

- **Foto\_Osm:** Representa a tabela de junção dos resultados do *geotagging* das fotos, onde cada imagem é vinculada aos locais em que foi capturada, considerando que uma foto pode estar relacionada a múltiplos locais. Por exemplo, em Paris, França, o Museu do Louvre é um importante ponto turístico. Uma foto tirada na pirâmide de vidro do Louvre, mesmo estando fora do museu, pode ser associada a ele por fazer parte da mesma localidade. Para isso, a tabela armazena as chaves estrangeiras das respectivas tabelas (Foto e Osm);
- **Viagem:** Armazena informações sobre as viagens de cada usuário. Para cada usuário, a tabela contém uma chave estrangeira que representa sua identificação na tabela de fotos, além de armazenar informações sobre a duração da viagem;
- **Itinerário:** Armazena informações sobre o trajeto realizado por cada usuário durante suas viagens. Para cada itinerário, a tabela registra uma expressão literal composta pelo ID de cada local visitado pelo usuário, juntamente com uma chave estrangeira que representa a identificação da viagem na tabela Viagem;
- **Localidade:** Registra as informações relacionadas às localidades que compõem cada itinerário. Para cada localidade, a tabela armazena informações como data de visita, latitude, longitude, país, estado, nome e identificação do local no *OpenStreetMap*. Além disso, armazena a chave estrangeira que faz referência à identificação do itinerário na tabela principal.

O banco de dados utilizado para armazenar os dados foi o *PostgreSQL*, na versão 12.15, com a utilização da extensão *PostGIS* para o tratamento de dados geoespaciais. Detalhes adicionais sobre o esquema do banco de dados estão disponíveis no Apêndice A deste documento.

### 3.4 IMPLEMENTAÇÃO

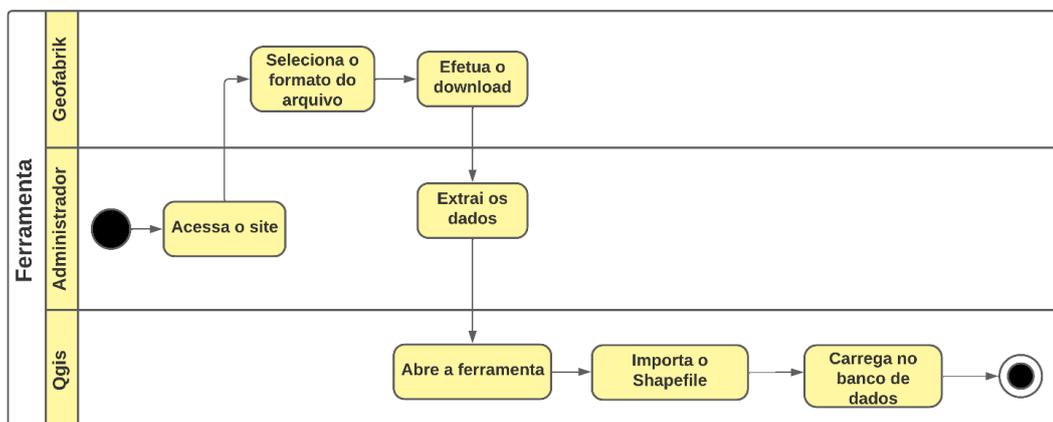
Esta seção descreve o processo de implementação da ferramenta. Inicialmente, é abordado o processo de extração e importação dos dados espaciais das cidades escolhidas para análise. Depois, é feita a apresentação do processo de implementação de cada uma das três aplicações que compõem a sua arquitetura.

#### 3.4.1 Importação dos dados espaciais

Para este projeto, selecionamos Paris, na França, e Rio de Janeiro, no Brasil, como estudos de caso. Essas cidades foram escolhidas por serem destinos turísticos significativos em seus continentes. Os dados geográficos dessas localidades são obtidos do *OpenStreetMap*, através da página de *download* do *Geofabrik*, com preferência

pelos arquivos no formato *shapefile*. Na Figura 12, apresentamos um diagrama de atividade que ilustra o fluxo desse processo.

**Figura 12 – Diagrama de atividade do fluxo de importação dos *shapefiles***



Fonte: Elaborado pelo autor, (2024).

A Figura 12 ilustra o processo de importação dos dados geoespaciais das cidades selecionadas para a análise deste estudo. Inicialmente, foi acessada a página de *download* gratuito do *Geofabrik*, onde estão disponíveis os dados geoespaciais de diversas localidades, regiões e sub-regiões. Após selecionar a localidade desejada, é possível selecionar o formato do arquivo e realizar o *download*.

Para importar os dados no banco de dados, utilizamos o *Qgis*, uma ferramenta de código aberto e multiplataforma, que permite a visualização, edição e análise de dados georreferenciados (FLENNIKEN et al., 2020). Por meio dessa ferramenta, os dados extraídos no formato *shapefile* foram importados e carregados no banco de dados da ferramenta na tabela *Osm*, que é a responsável por armazenar os dados dos lugares. A Figura 13 mostra uma visualização parcial da tabela de lugares após o seu povoamento. Nela, é possível observar os dados de vários lugares localizados na cidade do Rio de Janeiro.

Figura 13 – Visualização parcial da tabela de lugares

id [PK] integer	geometria geometry	pais text	estado text	categoria text	nome text
3404747	0106000020E61000000100000001030...	Brasil	Rio de Janeiro	Atração	Paróquia Nossa Senhora Rainha da Paz
5473745	0106000020E61000000100000001030...	Brasil	Rio de Janeiro	Atração	Porcino Park Center
5476618	0104000020E61000000100000001010...	Brasil	Rio de Janeiro	Atração	Pórtico
5476835	0104000020E61000000100000001010...	Brasil	Rio de Janeiro	Atração	Dunas do Rosado
5477131	0104000020E61000000100000001010...	Brasil	Rio de Janeiro	Atração	Catavento
5479085	0104000020E61000000100000001010...	Brasil	Rio de Janeiro	Atração	Pedra do Sapo
5479156	0104000020E61000000100000001010...	Brasil	Rio de Janeiro	Atração	Pórtico
5479224	0104000020E61000000100000001010...	Brasil	Rio de Janeiro	Atração	Casa do Barão de Serra Branca
5479321	0104000020E61000000100000001010...	Brasil	Rio de Janeiro	Atração	Pórtico
3424525	0106000020E61000000100000001030...	Brasil	Rio de Janeiro	Atração	Citibank Hall
3431773	0104000020E61000000100000001010...	Brasil	Rio de Janeiro	Atração	Ayrton Senna Túmulo
5479800	0104000020E61000000100000001010...	Brasil	Rio de Janeiro	Atração	Estátua de Santa Rita
3432536	0106000020E61000000100000001030...	Brasil	Rio de Janeiro	Atração	Paróquia Sagrada Família

Fonte: Elaborado pelo autor, (2024).

### 3.4.2 Ferramenta de coleta e processamento de dados

As informações referentes aos metadados fotos do *Flickr* foram obtidos por meio de sua *API* de dados, utilizando o modelo de arquitetura *REST*. A ferramenta de coleta e processamento de dados possui um módulo com dois submódulos: *Download* e *Processamento*. Para o seu desenvolvimento, foi utilizada a linguagem de programação Java e o *framework Spring Boot*<sup>1</sup>.

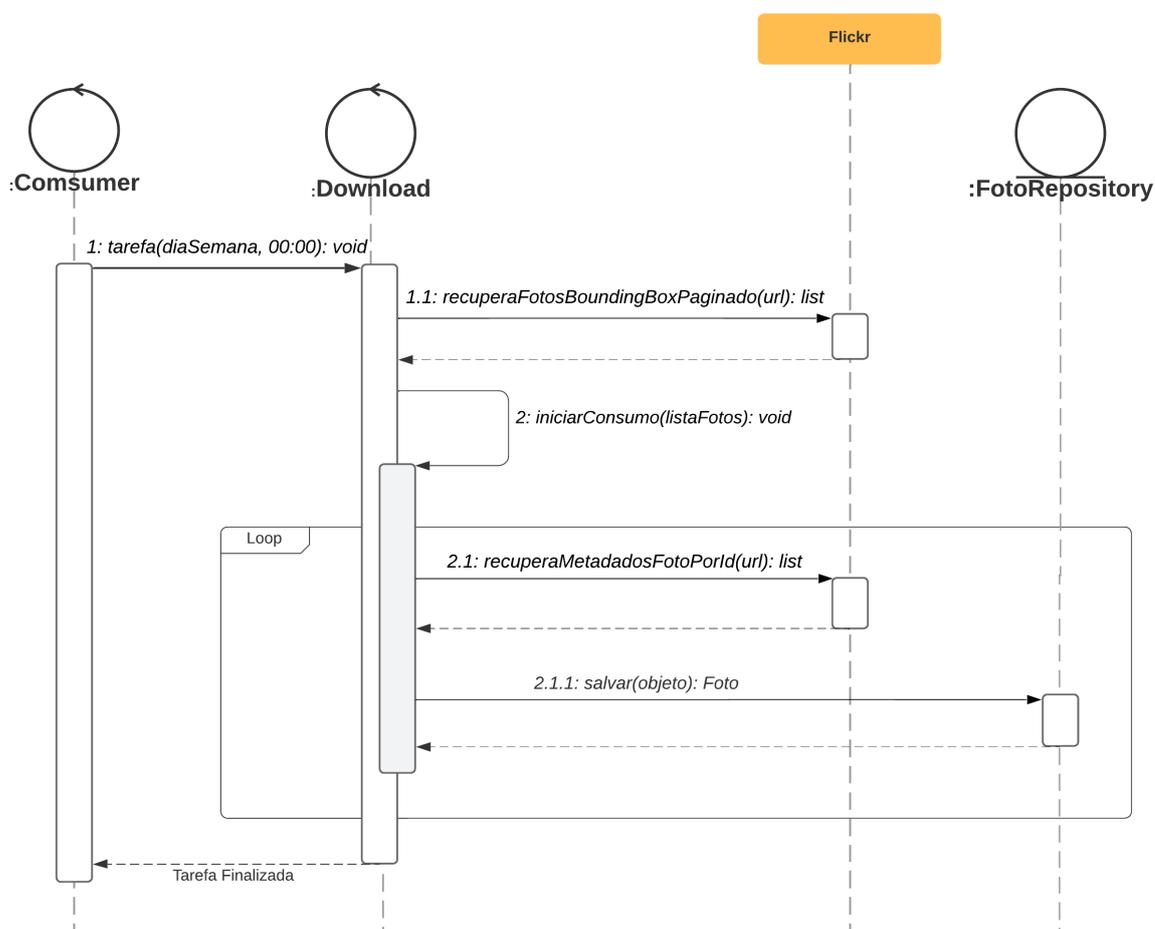
#### 3.4.2.1 O processo de coleta das fotos

O submódulo de *download* é responsável por automatizar todo o processo de extração dos metadados das fotos. Para consumir os dados ofertados, foi implementado o serviço *FotoService*. Esse serviço foi construído utilizando *WebClient*<sup>2</sup>, que funciona de forma assíncrona e não bloqueante e é fornecida através da estrutura do *Spring Reactive* para abstrair o processo de chamadas *HTTP* para os serviços ofertados pela *API* de dados do *Flickr*. O serviço possui três métodos distintos. Na Figura 14 é apresentado um diagrama de sequência que ilustra a interação entre os submódulos da ferramenta de coleta e processamento de dados.

<sup>1</sup> <https://spring.io/projects/spring-boot>

<sup>2</sup> <https://docs.spring.io/spring-framework/reference/web/webflux-webclient.html>

Figura 14 – Diagrama de sequência da comunicação entre os componentes da ferramenta de coleta e processamento de dados



Fonte: Elaborado pelo autor, (2024).

O primeiro método, *recuperaFotosBoundingBoxPaginado*, conecta-se à API do *Flickr* e consome o serviço chamado *search*. Ele retorna um arquivo no formato *JSON* como resposta, representando uma lista paginada de fotos. Cada foto contém as coordenadas de latitude e longitude dentro da caixa delimitadora passado como parâmetro para o método.

O segundo método, *recuperaMetadadosFotoPorId*, também se conecta à API do *Flickr* e utiliza o serviço chamado *getInfo*. Ele retorna um arquivo no formato *JSON* contendo os metadados da foto, cuja identificação foi passada como parâmetro para o método.

O terceiro método, *IniciarConsumo*, itera sobre a resposta do método *recuperaFotosBoundingBoxPaginado*. Para cada foto recuperada, ele passa a identificação para o

método *recuperaMetadadosFotoPorId*, que por sua vez, recupera os metadados da foto, cuja identificação foi passada como parâmetro. Em seguida, é realizado o *binding* da resposta da requisição no formato *JSON* para a classe da lógica de negócio chamada *Foto*.

Por fim, é utilizada a interface *FotoRepository* para abstrair as interações com o banco de dados. Essa interface estende a interface *JpaRepository*<sup>3</sup>, onde já estão definidos os principais métodos de *CRUD*, como por exemplo, o método *save*, utilizado para persistir o objeto *Foto* no banco de dados. Ao finalizar a persistência dos dados, é realizado o *geotagging* das fotos armazenadas no banco de dados.

### 3.4.2.2 O processo de geotagging das fotos

O processo de *geotagging* consiste em identificar, por meio da latitude e longitude presente nos metadados de cada foto, os locais onde elas foram retratadas. Para isso, é chamado um procedimento armazenado responsável por realizar esse processo. O código desse procedimento armazenado está ilustrado na Figura 15.

**Figura 15 – Procedimento armazenado responsável pelo geotagging das fotos**

```

1 CREATE OR REPLACE FUNCTION geotagging() RETURNS void LANGUAGE plpgsql
2 AS $FUNCTION$
3     DECLARE
4         rec RECORD;
5     BEGIN
6         FOR rec IN
7             SELECT
8                 f.id AS id_foto,
9                 o.id AS id_osm
10            FROM
11                fotos f, osm o
12            WHERE
13                st_contains(o.geom, st_geomfromtext(concat('POINT(', f.longitude, ' ', f.latitude, ')'), 4326))
14        LOOP
15            INSERT INTO foto_osm(id_foto, id_osm) VALUES (rec.id_foto, rec.id_osm);
16        END LOOP;
17    END $FUNCTION$;
18

```

Fonte: Elaborado pelo autor, (2024).

Como demonstrado na Figura 15, o procedimento armazenado consiste em transformar a latitude e longitude de cada foto presente na tabela de *Foto* em um ponto por meio da função *ST\_GEOMFROMTEXT*<sup>4</sup>. Em seguida, usando a função *ST\_CONTAINS*<sup>5</sup>, verifica se o ponto representado pela latitude e longitude da foto está completamente dentro da geometria que representa cada lugar armazenado na

<sup>3</sup> <https://docs.spring.io/spring-data/jpa/reference/>

<sup>4</sup> [https://postgis.net/docs/ST\\_GeomFromText.html](https://postgis.net/docs/ST_GeomFromText.html)

<sup>5</sup> [https://postgis.net/docs/ST\\_Contains.html](https://postgis.net/docs/ST_Contains.html)

tabela *Osm*. Por fim, após a conclusão do processo de *geotagging*, são persistidos a identificação da foto e a identificação do lugar na tabela de junção *Foto\_Osm*.

### 3.4.2.3 O processo de identificação dos itinerários

O submódulo de *processamento* tem a responsabilidade de automatizar o processo de geração do itinerário realizado por cada usuário registrado na base de dados. É importante salientar que, nesse contexto, foi considerado que o período de tempo que caracteriza uma viagem é de trinta dias. Dessa forma, as fotos capturadas dentro desse intervalo são agrupadas como pertencentes à mesma viagem, enquanto aquelas tiradas após esse período são consideradas parte de uma nova viagem. Durante o desenvolvimento dessa tarefa foi implementado o serviço chamado *ItinerarioService*.

Este serviço possui um método chamado *gerarItinerario*, onde inicialmente é recuperada uma lista de usuários cujas fotos estão armazenadas na base de dados. Em seguida é realizada uma iteração sobre a lista de usuários para recuperar todos os registros de fotos com base na identificação do usuário proprietário. Posteriormente, outra iteração é realizada sobre uma lista de fotos, com a finalidade de descobrir o itinerário realizado pelo usuário durante as suas viagens.

Para essa tarefa, é realizada uma verificação inicial utilizando o método *verificaDuracaoViagem*, responsável por controlar a duração da viagem do usuário com base na data em que a foto foi registrada. Após verificar o prazo de duração para uma viagem, é invocado o método *criaViagem*, responsável por criar um objeto do tipo *Viagem*. O retorno deste método é uma instância da classe *Viagem*.

Em seguida, é invocado o método *criaItinerario*, que é responsável por criar um objeto do tipo *Itinerario*. O retorno deste método é uma instância da classe *Itinerario*. E, por fim, é invocado o método *criaLocalidade*, que é responsável por criar e retornar uma instância do objeto do tipo *Localidade*. Ao final deste processo, após a instanciação de todas as classes da lógica de negócio, que determinam as informações de viagens, itinerários e localidades usadas para povoar as tabelas de mesmo nome e que foram abordadas na seção 3.3, finalmente, por meio do método *save* da interface *ViagemRepository* o objeto *Viagem* é persistido na base de dados.

Este processo é aplicado igualmente ao objeto *Itinerario* e *Localidade*, uma vez que a classe *Viagem* foi anotada com a propriedade *CascadeType.PERSIST* do JPA<sup>6</sup>, que propaga a operação de persistência de um objeto pai para um objeto filho. Esse processo é repetido até que todos os usuários da base tenham os seus itinerários

<sup>6</sup> <https://spring.io/projects/spring-data-jpa>

processados. A Figura 16 mostra um exemplo de um itinerário gerado após esse processo.

**Figura 16 – Exemplo de itinerário gerado**

usuario	pais	cidade	latitude	longitude	localidade	🔒	data
character var	character varying (255)		timestamp without tim				
Craig Vitter	France	Paris	48.855388	2.344947	Arc de Triomphe		2015-10-11 14:13:20
Craig Vitter	France	Paris	48.845127	2.353741	Basilique du Sacré-Cœur		2015-10-13 10:16:03
Craig Vitter	France	Paris	48.847936	2.350730	Cathédrale Notre-Dame		2015-10-10 10:29:48
Craig Vitter	France	Paris	48.855450	2.345386	Cavae des Arènes de Lutèce		2015-10-10 11:52:43
Craig Vitter	France	Paris	48.873652	2.295058	Champ de Mars		2015-10-09 19:15:48
Craig Vitter	France	Paris	48.855450	2.345386	Gradins		2015-10-10 11:58:53
Craig Vitter	France	Paris	48.855883	2.344891	Île de la Cité		2015-10-08 11:20:31
Craig Vitter	France	Paris	48.860500	2.337933	Jardin des Tuileries		2015-10-14 09:10:47
Craig Vitter	France	Paris	48.866055	2.320313	Jardin du Luxembourg		2015-10-08 17:39:16
Craig Vitter	France	Paris	48.855388	2.344947	Jardin du Palais Royal		2015-10-10 17:31:02

Fonte: Elaborado pelo autor, (2024).

### 3.4.3 Gerenciamento da aplicação

O gerenciamento da aplicação é encarregado de receber e processar todas as requisições realizadas pela aplicação web. Para desenvolvê-lo, utilize o ambiente de execução *Node.js*. Esta aplicação foi dividida em três módulos: *gerenciador de rotas*, *controladores* e *serviço de consultas*.

O módulo de *gerenciamento de rotas* recebe as requisições provenientes da aplicação web. Para essa tarefa, foi criado um arquivo chamado *Routes*, onde encontram-se as funções do objeto *Router*, que são responsáveis por abstrair os métodos *HTTP* incorporados aos controladores previamente definidos da aplicação, com o objetivo de executar determinadas tarefas. Para o seu desenvolvimento foi utilizado o *framework Express*<sup>7</sup>.

O módulo *controladores* é responsável por gerenciar as requisições advindas dos métodos *HTTP*, processando as chamadas feitas para os *endpoints* e encaminhando os parâmetros necessários ao serviço de consulta específico.

O módulo de *serviço de consultas* é encarregado de interagir com o banco de dados através de parâmetros previamente definidos advindos dos controladores, a fim de executar consultas na base de dados. Para o seu desenvolvimento, o *framework*

<sup>7</sup> <https://expressjs.com/pt-br/>

*TypeORM*<sup>8</sup> foi utilizado para abstrair o processo de interação com o banco de dados. As respostas das consultas são retornadas em um documento *JSON* ao controlador específico, que, por sua vez, retorna o resultado para a *aplicação web*.

#### 3.4.4 *Aplicação Web*

A *aplicação web* desempenha o papel de interação e exibição dos dados para o usuário final. Para sua implementação, o *framework React.js* foi adotado para construir as telas e componentes visuais da aplicação. Esta escolha foi motivada pela facilidade de criação e reutilização de componentes em várias partes do sistema, agilizando a construção da interface de forma prática (FEDOSEJEV, 2015). Além disso, também utilizou-se a biblioteca *Axios*<sup>9</sup>, que funciona como um cliente *HTTP*, permitindo que a aplicação utilize a arquitetura *Restful* como meio de comunicação entre a *aplicação web* e o *back-end* da ferramenta.

A aplicação oferece diferentes tipos de consultas, como a pesquisa dos principais pontos turísticos de uma cidade ou região, bem como a filtragem dos pontos turísticos de uma localidade com base nos principais tipos de pontos de interesse, como igrejas, parques, praias e outros. Além disso, os itinerários turísticos podem ser visualizados de maneira gráfica em um mapa, proporcionando uma experiência mais interativa ao usuário.

A Figura 17 apresenta um diagrama de sequência que ilustra a comunicação entre a aplicação web e o gerenciamento da aplicação, evidenciando uma requisição feita para obter os dados das principais atrações turísticas de uma região específica.

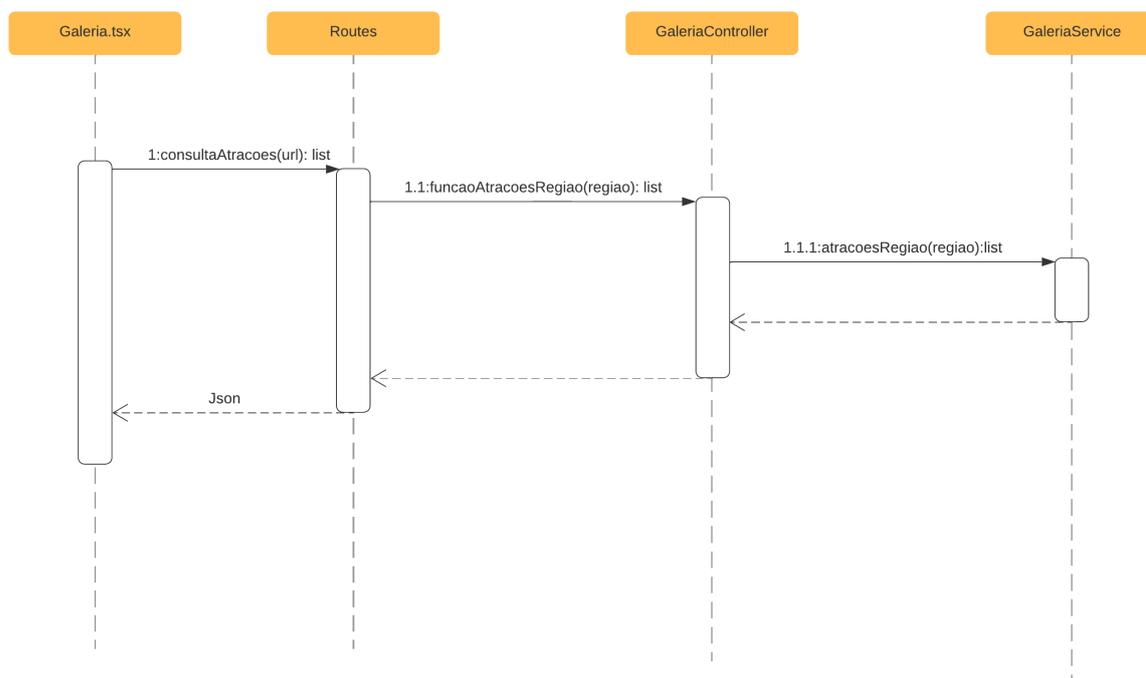
Nela, observa-se que o *front-end* envia solicitações ao módulo de gerenciamento da aplicação por meio de uma *URL*, as quais são recebidas pelo gerenciador de rotas. Este, por sua vez, mapeia a requisição para o *endpoint* correspondente e seu controlador vinculado, que então consulta o serviço específico da requisição obter os dados relacionados aos parâmetros passados na requisição. Após a obtenção dos dados, o gerenciador de rotas recebe essas informações e as retorna como resposta da consulta em formato *JSON*, para ser usado no *front-end*.

---

<sup>8</sup> <https://typeorm.io/>

<sup>9</sup> <https://axios-http.com/ptbr/docs/intro>

**Figura 17 – Diagrama de sequência da comunicação entre front-end e o gerenciamento da aplicação**



Fonte: Elaborado pelo autor, (2024).

O resultado da busca pelas principais atrações de uma determinada região é apresentado de forma gráfica por meio de *cards* contendo o nome e a foto representativa do local, contribuindo para a implementação do RF01. O RF02 pode ter sua implementação visualizada na Figura 18, que mostra o resultado de uma busca por pontos de interesse na cidade do Rio de Janeiro, filtrando pela categoria de lugares mais visitados, como praias, parques e museus, proporcionando ao usuário a possibilidade de selecionar a categoria do lugar que ele deseja. Como ilustrado na imagem, na parte horizontal, há botões que descrevem o nome das categorias de lugares mais visitados. Neste exemplo, o usuário pode selecionar uma categoria de seu interesse.

**Figura 18 – Busca por pontos de interesse filtrando por alguma das categorias de lugares mais visitados**



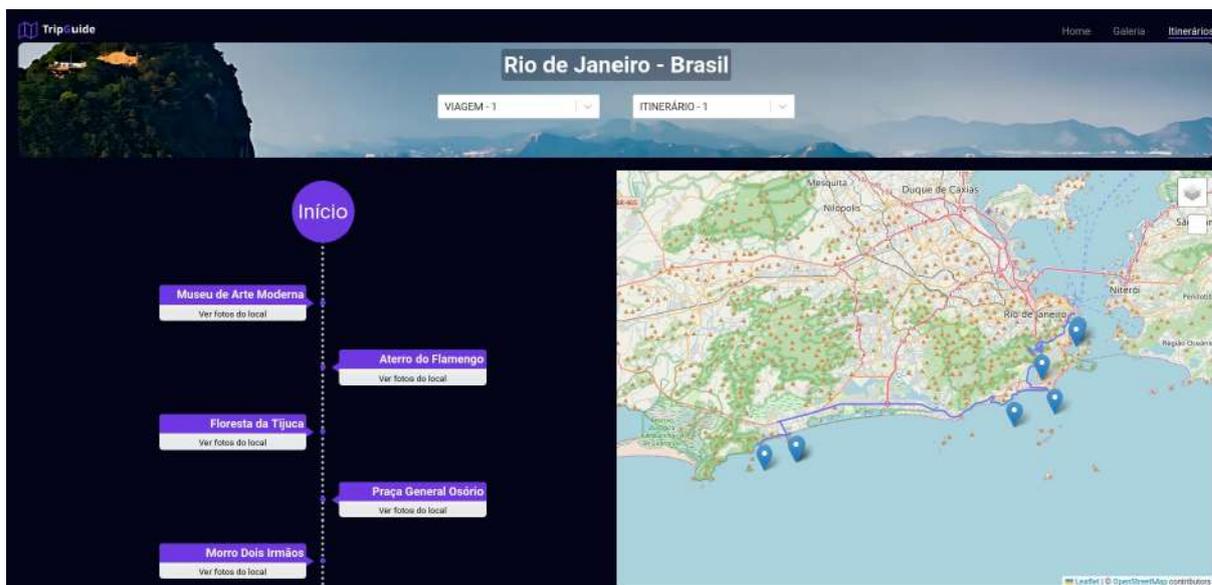
Fonte: Elaborado pelo autor, (2024).

Para a visualização dos itinerários mais comuns realizados pelos usuários de uma determinada região, conforme definido no RF03, optou-se por exibir os dados por meio de um mapa. Essa decisão foi tomada visando uma melhor compreensão das informações por parte dos usuários. Ainda é possível visualizar a trajetória realizada de um ponto para outro no mapa, o que torna a experiência visual do usuário mais agradável. Para implementar o mapa, utilizou-se a biblioteca *Leaflet*<sup>10</sup>, selecionada devido à sua natureza de código aberto, facilidade de uso e ampla gama de funcionalidades que facilitam a criação de sistemas de mapas gráficos.

A Figura 19 mostra o resultado de uma consulta pelos itinerários mais frequentes na cidade do Rio de Janeiro.

<sup>10</sup> <https://leafletjs.com/>

Figura 19 – Busca pelos itinerários mais frequentes de uma região



Fonte: Elaborado pelo autor, (2024).

## 4 CONSIDERAÇÕES

Viajar é uma atividade cada vez mais comum para muitas pessoas. O grande crescimento de serviços de hospedagem de curta duração e das companhias aéreas de baixo custo encolheram o mundo e o tornaram mais acessível para aqueles que têm o desejo de sair e conhecer novos lugares. Neste contexto, muitas das experiências de viagens atuais são influenciadas pelas mídias sociais que, por sua vez, nos enviam a lugares distantes em busca de destinos autênticos e não descobertos, ou somente para recriar as boas experiências dos influenciadores, que nos permitiram desfrutar visualmente dos lugares por onde passaram em uma de suas viagens de férias.

Entretanto, é um fato que nem sempre é possível contemplar todos os lugares que se deseja visitar. Muitas vezes isto ocorre pela falta de conhecimento geográfico, que limita o conhecimento relacionado aos pontos de interesses que circundam um determinado lugar, pois, muitas vezes, ao observar somente um destino específico, muitos outros deixam de ser contemplados, pelo simples fato de não ter um itinerário bem definido. Assim, a falta da organização do itinerário pode fazer com que o turista tenha uma experiência insatisfatória ou até mesmo frustrante durante a sua viagem. Objetivando contornar esses problemas, este trabalho propôs o desenvolvimento de uma ferramenta que tem como objetivo auxiliar pessoas a organizarem as suas viagens, descobrindo os principais pontos de interesse e os itinerários mais comuns em uma determinada localidade.

A ferramenta desenvolvida compriu todos os requisitos funcionais definidos em sua especificação, possibilitando ao usuário uma melhor análise e conhecimento a respeito dos itinerários mais frequentes de uma determinada cidade ou região, além de simplificar e tornar mais eficiente a análise de grandes quantidades de fotos por parte do usuário.

Após o desenvolvimento do trabalho, pode-se dizer que todos os objetivos específicos foram alcançados. O primeiro objetivo específico foi atingido, pois a ferramenta, por meio do processo de *geotagging*, conseguiu associar cada imagem ao local onde ela foi retratada. O segundo objetivo específico foi concluído com a implementação de um serviço responsável por processar os itinerários de cada usuário presente no banco de dados, permitindo que o usuário visualize os itinerários mais frequentes em um determinado destino. O terceiro objetivo específico foi atingido com a implementação de um serviço responsável por consumir e processar os metadados das fotos e carregá-los na base de dados de forma automática.

Alguns trabalhos futuros podem ser desenvolvidos para melhorar a ferramenta desenvolvida neste TCC. Uma delas é a migração dos dados dos itinerários para uma nova base de dados baseada em grafo, o que simplificaria a manipulação e ampliaria as possibilidades de exposição dos dados, incluindo a recomendação de itinerários turísticos personalizados. Outra melhoria que pode ser abordada é a colaboração dos usuários, destacando a possibilidade de contribuir com a evolução da ferramenta, por meio de um cadastro, onde os colaboradores podem contribuir disponibilizando os metadados de suas fotos para enriquecer ainda mais a base de dados da ferramenta. Além disso, é possível aprimorar ainda mais o processamento dos itinerários turísticos realizados pela ferramenta, integrando com alguma biblioteca de aprendizado de máquina para auxiliar na identificação das fotos, além de possibilitar que os usuários avaliem os itinerários e os classifiquem de acordo com experiências pessoais. Por último, a experiência visual do usuário pode ser aprimorada por meio de melhorias no *front-end* da ferramenta.

## REFERÊNCIAS

DATAREPORTAL. **Digital 2022: Global Overview Report**. 2022. Acesso em 20 de jan. de 2024. Disponível em: <<https://datareportal.com/reports/digital-2022-global-overview-report>>.

FEDOSEJEV, A. **React.js Essentials**. [S.l.]: Packt Publishing Ltd, 2015. (A fast-paced guide to designing and building scalable and maintainable web apps with React.js.). ISBN 978-1-78355-162-0.

FERRARI, C. M. M.; GÂNDARA, J. M. Fotografias de viagens: replicando cenas da viagem perfeita em Curitiba/pr. **Caderno Virtual de Turismo**, v. 15, n. 2, dez. 2015. Disponível em: <<https://www.ivt.coppe.ufrj.br/caderno/article/view/975>>.

FLENNIKEN, J. M.; STUGLIK, S.; IANNONE, B. V. Quantum gis (qgis): An introduction to a free alternative to more costly gis platforms: For359/fr428, 2/2020. **EDIS**, v. 2020, n. 2, p. 7, Mar. 2020. Disponível em: <<https://journals.flvc.org/edis/article/view/108810>>.

GIBBS, S. **Flickr bought by SmugMug as Yahoo breakup begins**. **The Guardian**. 2019. Acesso em 12 dez fev. de 2023. Disponível em: <<https://www.theguardian.com/technology/2018/apr/23/flickr-bought-by-smugmug-yahoo-breakup>>.

GUTIERREZ, F. **Pro Spring Boot**. [S.l.]: Apress, 2016. (A no-nonsense guide containing case studies and best practices for Spring Boot). ISBN 978-1-4842-1432-9.

HAKLAY, M.; WEBER, P. Openstreetmap: User-generated street maps. **IEEE Pervasive Computing**, v. 7, n. 4, p. 12–18, 2008.

IATA. **A forte recuperação de tráfego de passageiros continuou em agosto, mostram os dados de hoje da IATA**. 2023. Acesso em 10 de fev. de 2024. Disponível em: <<https://aeroin.net/a-forte-recuperacao-de-trafego-de-passageiros-continuou-em-agosto-mostram-os-dados-de-hoje-da-iata/>>.

INSPER. **Mundo se aproxima da marca de 5 bilhões de usuários de internet**. 2022. Acesso em 20 de jan. de 2024. Disponível em: <<https://www.insper.edu.br/noticias/mundo-se-aproxima-da-marca-de-5-bilhoes-de-usuarios-de-internet-63-da-populacao>>.

JETBRAINS. **The State of Developer Ecosystem 2021**. 2021. Acesso em 22 ago. de 2024. Disponível em: <<https://www.jetbrains.com/lp/devecosystem-2021/>>.

LARDINOIS, F. **Pelo amor aos dados de mapeamento abertos**. **TECHCRUNCH**. 2014. Acesso em 10 de fev. de 2024. Disponível em: <<https://techcrunch.com/2014/08/09/for-the-love-of-open-mapping-data/>>.

MELO, D. **O que é java? [guia para iniciantes]**. 2020. Acesso em 12 ago. de 2024. Disponível em: <<https://tecnoblog.net/responde/o-que-e-java-guia-para-iniciantes/>>.

MORETTI, S. L. d. A.; BERTOLI, B. J.; ZUCCO, F. D. A imagem de Blumenau no Instagram: um estudo sobre destino turístico em redes sociais usando equações estruturais. **Caderno Virtual de Turismo**, v. 16, n. 1, abr. 2016. Disponível em: <<https://www.ivt.coppe.ufrj.br/caderno/article/view/1045>>.

OMT. **Turismo global deve retomar nível pré-pandemia em 2024. Nações Unidas Brasil.** 2024. Acesso em 10 de fev. de 2024. Disponível em: <<https://brasil.un.org/pt-br/258446-turismo-global-deve-retomar-n%C3%ADvel-pr%C3%A9-pandemia-em-2024>>.

SILVA, M. S. **React - Aprenda Praticando.** [S.l.]: Novatec Editora Ltda, 2021. (Desenvolva Aplicações Web Reais com uso da Biblioteca React e de Seus Módulos Auxiliares). ISBN 978-65-86057-39-3.

TILKOV, S.; VINOSKI, S. Node.js: Using javascript to build high-performance network programs. **IEEE Internet Computing**, v. 14, n. 6, p. 80–83, 2010.

TIOBE. **TIOBE - The Software Quality Company.** 2024. Acesso em 15 ago. de 2024. Disponível em: <<https://www.tiobe.com/tiobe-index/>>.

WEBB, P. e. a. **How-to guides.** 2017. Acesso em 12 ago. de 2024. Disponível em: <<https://docs.spring.io/spring-boot/docs/current/reference/html/howto-embedded-web-servers.html>>.

## APÊNDICE A – DICIONÁRIO DE DADOS LÓGICO

FOTOS: Relação que armazena as informações sobre os metadados das fotos				
Atributo	Descrição	Tipo	Domínio	Restrição
id	Atributo que representa a identificação da foto	String(50)	String(50)	• Chave primária
data_publicacao	Atributo que representa a data de publicação da foto	String(50)	String(50)	• Não nulo
país	Atributo que representa o nome do país onde a foto foi registrada	String(50)	String(50)	• Não nulo
município	Atributo que representa o nome da cidade onde a foto foi registrada	String(50)	String(50)	• Não nulo
bairro	Atributo que representa o nome do bairro onde a foto foi registrada	String(50)	String(50)	• Não nulo
estado	Atributo que representa o nome do estado onde a foto foi registrada	String(50)	String(50)	• Não nulo
latitude	Atributo correspondente à latitude de onde a foto foi registrada	String(50)	String(50)	• Não nulo

longitude	Atributo correspondente à longitude de onde a foto foi registada	String(50)	String(50)	• Não nulo
usuario_id	Atributo que representa a identificação do usuário proprietário da foto	String(50)	String(50)	• Não nulo
usuario_nome	Atributo que representa o nome do usuário proprietário da foto	String(50)	String(50)	• Não nulo
Título	Atributo que representa o título da foto atribuído pelo proprietário da foto	String(50)	String(50)	• Não nulo
OSM_ID	Atributo que representa a identificação do lugar onde a foto foi tirada	String(50)	String(50)	• Chave estrangeira que referencia o atributo "OSM_ID" da relação "OSM"

**FOTO\_OSM: Relação que representa a tabela de junção entre a relação FOTO e OSM**

Atributo	Descrição	Tipo	Domínio	Restrição
ID	Atributo que representa a identificação do registro na tabela	Int	Números inteiros positivos	• Chave primária

foto_id	Atributo que representa a identificação da foto	Int	Números inteiros positivos	<ul style="list-style-type: none"> <li>• Chave estrangeira que referencia o atributo "id" da relação "FOTO"</li> </ul>
osm_id	Atributo que representa a identificação do lugar	Int	Números inteiros positivos	<ul style="list-style-type: none"> <li>• Chave estrangeira que referencia o atributo "id" da relação "OSM"</li> </ul>

**OSM: Relação que armazena as informações de geolocalização das localidades**

Atributo	Descrição	Tipo	Domínio	Restrição
id	Atributo que representa a identificação de lugar	Int	Números inteiros positivos	<ul style="list-style-type: none"> <li>• Chave primária</li> </ul>
code	Atributo que representa o código do lugar	Int	Números inteiros positivos	<ul style="list-style-type: none"> <li>• Não nulo</li> </ul>
geom	Atributo que representa a geometria do lugar	Geometry	Geometry	<ul style="list-style-type: none"> <li>• Não nulo</li> </ul>
osm_id	Atributo que identifica cada lugar em específico	Text	Text	<ul style="list-style-type: none"> <li>• Não nulo</li> </ul>
fclass	Atributo que representa a categoria de um lugar	String (100)	String (100)	<ul style="list-style-type: none"> <li>• Não nulo</li> </ul>
name	Atributo que representa o nome do lugar	String(255)	String(255)	<ul style="list-style-type: none"> <li>• Não nulo</li> </ul>
Type	Atributo que representa o tipo de construção	String(100)	String(100)	<ul style="list-style-type: none"> <li>• Não nulo</li> </ul>

**VIAGEM: Relação que armazena informações relacionadas as viagens do usuário**

<b>Atributo</b>	<b>Descrição</b>	<b>Tipo</b>	<b>Domínio</b>	<b>Restrição</b>
id	Atributo que representa a identificação da viagem	Int	Números inteiros positivos	• Chave primária
usuario_id	Atributo que representa a identificação do usuário	Int	Números inteiros positivos	• Chave estrangeira que referencia o atributo "usuario_id" da relação "FOTO"
duracao	Atributo que representa a duração em dias de cada viagem do usuário	Int	Números inteiros positivos	• Não nulo

**ITINERÁRIO: Relação que armazena as informações de cada itinerário realizado - pelo usuário**

<b>Atributo</b>	<b>Descrição</b>	<b>Tipo</b>	<b>Domínio</b>	<b>Restrição</b>
id	Atributo que representa a identificação do itinerário	Int	Números inteiros positivos	• Chave primária
viagem_id	Atributo que representa a identificação da viagem	Int	Números inteiros positivos	• Chave estrangeira que referencia o atributo "id" da relação "VIAGEM"
expressao	Atributo que representa a expressão literal que corresponde a junção de todos os "ID" dos lugares que o usuário visitou	Text	Text	• Não nulo

**LOCALIDADE: Relação que armazena as informações de cada lugar visitado em cada itinerário realizado pelo usuário**

<b>Atributo</b>	<b>Descrição</b>	<b>Tipo</b>	<b>Domínio</b>	<b>Restrição</b>
id	Atributo que representa a identificação da localidade	Int	Números inteiros positivos	• Chave primária
itinerario_id	Atributo que representa a identificação do itinerário realizado pelo usuário	Int	Números inteiros positivos	• Chave estrangeira que referencia o atributo "id" da relação "ITINERARIO"
data	Atributo que representa a data de visitação do lugar	Timestamp	Timestamp	• Não nulo
osm_id	Atributo que identifica cada lugar em específico	Text	Text	• Não nulo
latitude	Atributo que corresponde à latitude de onde a foto foi registrada	Double	Double	• Não nulo
longitude	Atributo que corresponde à longitude de onde a foto foi registrada	Double	Double)	• Não nulo
pais	Atributo que representa o nome do país a foto foi registrada	String(50)	String(50)	• Não nulo
estado	Atributo que representa o nome do estado onde a foto foi registrada	String(255)	String(255)	• Não nulo
nome	Atributo que representa o nome do lugar visitado pelo usuário	String(50)	String(50)	• Não nulo



## APÊNDICE B – TELAS DA APLICAÇÃO WEB

Figura 20 – Página inicial do sistema



Fonte: Elaborado pelo autor, (2024).

A Figura 20 mostra a página inicial do sistema, que informa ao usuário qual é o objetivo da ferramenta desenvolvida.

Figura 21 – Página de pesquisa do sistema



Fonte: Elaborado pelo autor, (2024).

A Figura 21 mostra a página de pesquisa do sistema, onde é possível realizar uma busca pelas principais localidades de uma determinada cidade ou região.

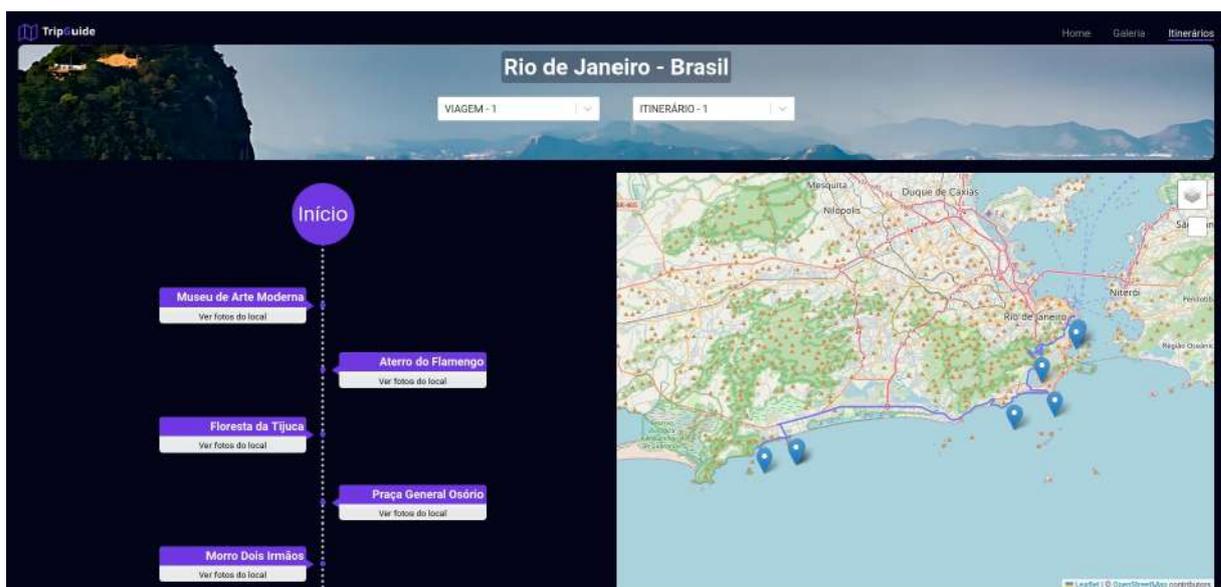
Figura 22 – Página de galeria



Fonte: Elaborado pelo autor, (2024).

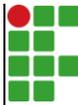
A Figura 22 mostra a página da galeria onde o usuário consegue visualizar as localidades com base no lugar selecionado. Ainda é possível a visualização dos locais mais visitados com base em uma categoria de interesse como parque, castelo, museu.

Figura 23 – Página de itinerários



Fonte: Elaborado pelo autor, (2024).

A Figura 22 mostra a página onde o usuário pode visualizar os itinerários mais frequentes de uma localidade.

	<b>INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA PARAÍBA</b>
	Campus Cajazeiras - Código INEP: 25008978
	Rua José Antônio da Silva, 300, Jardim Oásis, CEP 58.900-000, Cajazeiras (PB)
	CNPJ: 10.783.898/0005-07 - Telefone: (83) 3532-4100

## Documento Digitalizado Restrito

### Trabalho de Conclusão de Curso

<b>Assunto:</b>	Trabalho de Conclusão de Curso
<b>Assinado por:</b>	Diones Gomes
<b>Tipo do Documento:</b>	Anexo
<b>Situação:</b>	Finalizado
<b>Nível de Acesso:</b>	Restrito
<b>Hipótese Legal:</b>	Informação Pessoal (Art. 31 da Lei no 12.527/2011)
<b>Tipo da Conferência:</b>	Cópia Simples

Documento assinado eletronicamente por:

- **Diones Rocha Gomes, ALUNO (201612010180) DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS - CAJAZEIRAS**, em 27/08/2024 13:49:22.

Este documento foi armazenado no SUAP em 27/08/2024. Para comprovar sua integridade, faça a leitura do QRCode ao lado ou acesse <https://suap.ifpb.edu.br/verificar-documento-externo/> e forneça os dados abaixo:

Código Verificador: 1231081

Código de Autenticação: a695d315fb

