

UNIVERSIDADE FEDERAL DE OURO PRETO
INSTITUTO DE CIÊNCIAS EXATAS E BIOLÓGICAS
DEPARTAMENTO DE COMPUTAÇÃO

VICTOR HUGO MARTINS

Orientador: Prof. Dr. Anderson Almeida Ferreira

Coorientador: Prof. Dr. Rodrigo Meira Martoni

**MEDIAÇÕES DAS PLATAFORMAS AIRBNB E BOOKING NO SETOR
DE ALOJAMENTO: ELABORAÇÃO DE UM INSTRUMENTAL DE
EXTRAÇÃO E ANÁLISE DE DADOS**

Ouro Preto, MG
2023

UNIVERSIDADE FEDERAL DE OURO PRETO
INSTITUTO DE CIÊNCIAS EXATAS E BIOLÓGICAS
DEPARTAMENTO DE COMPUTAÇÃO

VICTOR HUGO MARTINS

**MEDIAÇÕES DAS PLATAFORMAS AIRBNB E BOOKING NO SETOR DE
ALOJAMENTO: ELABORAÇÃO DE UM INSTRUMENTAL DE EXTRAÇÃO E
ANÁLISE DE DADOS**

Monografia apresentada ao Curso de Ciência da Computação da Universidade Federal de Ouro Preto como parte dos requisitos necessários para a obtenção do grau de Bacharel em Ciência da Computação.

Orientador: Prof. Dr. Anderson Almeida Ferreira

Coorientador: Prof. Dr. Rodrigo Meira Martoni

Ouro Preto, MG
2023



FOLHA DE APROVAÇÃO

Victor Hugo Martins

MEDIAÇÕES DA PLATAFORMAS AIRBNB E BOOKING NO SETOR DE ALOJAMENTO: ELABORAÇÃO DE UM INSTRUMENTAL DE EXTRAÇÃO E ANÁLISE DE DADOS

Monografia apresentada ao Curso de Ciência da Computação da Universidade Federal de Ouro Preto como requisito parcial para obtenção do título de Bacharel em Ciência da Computação

Aprovada em 24 de Agosto de 2023.

Membros da banca

Anderson Almeida Ferreira (Orientador) - Doutor - Universidade Federal de Ouro Preto
Rodrigo Meira Martoni (Coorientador) - Doutor - Universidade Federal de Ouro Preto
Angela Teberga de Paula (Examinadora) - Doutora - Universidade de Brasília
Valéria de Carvalho Santos (Examinadora) - Doutora - Universidade Federal de Ouro Preto

Anderson Almeida Ferreira, Orientador do trabalho, aprovou a versão final e autorizou seu depósito na Biblioteca Digital de Trabalhos de Conclusão de Curso da UFOP em 24/08/2023.



Documento assinado eletronicamente por **Anderson Almeida Ferreira, PROFESSOR DE MAGISTERIO SUPERIOR**, em 30/08/2023, às 09:02, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



A autenticidade deste documento pode ser conferida no site http://sei.ufop.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **0577019** e o código CRC **637E90D1**.

Agradecimentos

Agradeço a todos os professores que me ajudaram nesse processo e a todos os familiares que me incentivaram. Agradeço a todos os amigos que me fizeram chegar até aqui, em especial Cibele e Sabrina, por todas as experiências, lutas e alegrias compartilhadas.

Resumo

Nos últimos anos, surgiram muitos mercados online que utilizam a internet para facilitar o comércio de diversos serviços. Um desses mercados é o Airbnb, que se destacou no ramo do turismo ao oferecer hospedagens em casas de família e experiências turísticas. Contemplando para fins comparativos as plataformas Airbnb e Booking, cujo segmento de locação de imóveis para curtas temporadas se projetos na última década, este trabalho busca promover transparência e fornecer dados para estudo acerca da atuação destas plataformas. Além disso, visa também descrever o desenvolvimento de um sistema web capaz de extrair dados dessas plataformas, por meio de consultas automatizadas, permitindo uma análise facilitada, usando técnicas de ciência de dados e métodos de agrupamento, como, por exemplo, BIRCH, DBSCAN e K-Derivados (K-Modes e K-Prototypes).

Palavras-chave: Airbnb. Booking. Mineração de Dados. Web Scraping. Aprendizado de máquina não supervisionado.

Abstract

In recent years, many online marketplaces have emerged that use the internet to facilitate trade in various services. One of these markets is Airbnb, which stood out in the tourism business by offering accommodation in family homes and tourist experiences. Contemplating for comparative purposes the Airbnb and Booking platforms, whose short-term real estate rental segment has been projects in the last decade. This work also seeks for promoting transparency and provide data for study about performance of such platforms and also describe the development of a web system capable of extract data from these platforms through automated queries, allowing an easier analysis using data science techniques and clustering techniques, such as BIRCH, DBSCAN and K-Derivatives (K-Modes and K-Prototypes).

Keywords: Airbnb. Booking. Data Mining. Web Scraping. Unsupervised machine learning.

Lista de Ilustrações

Figura 2.1 – Arquivo de configuração de variáveis de Pesquisa	9
Figura 2.2 – Modelo de algoritmo simples para scrap via selenium	9
Figura 3.1 – Hierarquia entre Atores do DashAcomodações	20
Figura 3.2 – Diagrama de Caso de Uso do DashAcomodações	21
Figura 3.3 – Diagrama ER para o DashAcomodações	29
Figura 3.4 – Diagrama Relacional para o DashAcomodações	30
Figura 3.5 – Função implementada para encontrar o texto correspondente ao Tipo de Propriedade de uma Acomodação do Booking	32
Figura 3.6 – Exemplo de Identificação de Acomodações do Booking	33
Figura 3.7 – Arquivo de configuração de credenciais do banco de dados	35
Figura 3.8 – Método implementado para alterar permissão do usuário	36
Figura 3.9 – Acesso da aplicação a método implementado na controller de Autenticação	36
Figura 3.10–Tipagem para dados da requisição do método para alterar nível de permissão do usuário	36
Figura 3.11–Tipagem para dados da requisição do método para alterar nível de permissão do usuário	37
Figura 3.12–Tela inicial do DashAcomodações	38
Figura 3.13–Implementação do componente de input por Área de Texto	39
Figura 4.1 – Link para acesso da página de cadastro	41
Figura 4.2 – Tela para cadastro	41
Figura 4.3 – Tela exibindo listagem com usuários do DashAcomodações	42
Figura 4.4 – Tela de Login	42
Figura 4.5 – Menu exibido após autenticação do usuário logado	43
Figura 4.6 – Tela exibindo formulário para inserção dos critérios de configuração de uma nova pesquisa	43
Figura 4.7 – Tela exibindo listagem com todas as pesquisas solicitadas pelo usuário	44
Figura 4.8 – Tela exibindo dados sobre acomodações coletadas por uma pesquisa específica	45

Lista de Tabelas

Tabela 3.1 – Descrição dos Atores do DashAcomodações	19
Tabela 3.2 – Descrição dos Serviços oferecidos pelo DashAcomodações	20
Tabela 3.3 – Exemplos de status possíveis para a Super Pesquisa	31
Tabela 3.4 – Parâmetros para agrupamento	34
Tabela A.1 – Descrição dos atributos da entidade Acomodação do Airbnb	52
Tabela A.2 – Descrição dos atributos da entidade Acomodação do Booking	53
Tabela A.3 – Descrição dos atributos da entidade Área de Pesquisa	53
Tabela A.4 – Descrição dos atributos da entidade Configuração de Pesquisa	54
Tabela A.5 – Descrição dos atributos da entidade Localização	54
Tabela A.6 – Descrição dos atributos da entidade Pesquisa	54
Tabela A.7 – Descrição dos atributos da entidade Super Pesquisa	55
Tabela A.8 – Descrição dos atributos da entidade Usuário	55

Lista de Abreviaturas e Siglas

API	Application Programming Interface
BIRCH	Balance Incremental Clustering using Hierarchies
CF	Clustering Feature
CORS	Cross-Origin Resource Sharing
CSR	Client-Side Rendering
CSS	Cascading Style Sheets
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
HTML	HyperText Markup Language
HTTP	Hypertext Transfer Protocol
JSON	JavaScript Object Notation
OTA	Online Travel Agency
SGBD	Sistema Gerenciador de Banco de Dados
SSR	Server-Side Rendering
UFOP	Universidade Federal de Ouro Preto
UI	User Interface
XML	Extensible Markup Language
XPath	XML Path

Sumário

1	Introdução	1
1.1	Justificativa	3
1.2	Objetivo	4
1.3	Organização do Trabalho	5
2	Revisão Bibliográfica	6
2.1	Fundamentação Teórica	6
2.1.1	Uberização	6
2.1.2	Web Scraping	7
2.1.2.1	Airbnb Data Collection	7
2.1.2.2	Selenium	8
2.1.3	Agrupamento	10
2.1.4	Back-end e Front-end	13
2.1.4.1	Back-end	13
2.1.4.2	Front-end	14
2.2	Trabalhos Relacionados	16
3	Desenvolvimento	18
3.1	Concepção de Software	18
3.1.1	Objetivo	18
3.1.2	Missão e Descrição do Produto	19
3.1.3	Escopo Positivo e Negativo	19
3.1.3.1	Positivo	19
3.1.3.2	Negativo	19
3.1.4	Descrição dos Atores	19
3.1.5	Diagrama de Caso de Uso	20
3.1.6	Backlog	21
3.1.6.1	Serviço de Autenticação	21
3.1.6.2	Serviço de Navegação	21
3.1.6.3	Serviço de Solicitação de Pesquisa	21
3.1.6.4	Serviço de Gestão de Usuários	22
3.1.6.5	Serviço de Gestão do Sistema	22
3.1.7	Estórias de Usuário	22
3.1.7.1	Serviço de Autenticação	22
3.1.7.2	Serviço de Navegação	23
3.1.7.3	Serviço de Solicitação de Pesquisa	23
3.1.7.4	Serviço de Gestão de Usuário	24
3.1.7.5	Serviço de Gestão de Sistema	24

3.1.8	Cenários de Teste	25
3.1.8.1	Serviço de Autenticação	25
3.1.8.2	Serviço de Navegação	26
3.1.8.3	Serviço de Solicitação de Pesquisa	26
3.1.8.4	Serviço de Gestão de Usuários	27
3.1.8.5	Serviço de Gestão do Sistema	27
3.2	Modelagem do Banco de Dados	28
3.3	Extração dos dados	29
3.3.1	Extração dos dados dos anúncios do Airbnb	31
3.3.2	Extração dos dados dos anúncios do Booking	31
3.4	Agrupamento	33
3.5	Implementação do back-end e front-end	34
3.5.1	Back-end	35
3.5.2	Front-end	37
4	Estudo de Caso	41
5	Conclusão	46
	Referências	48
	Apêndices	51
	APÊNDICE A Descrição dos atributos da Entidades do Sistema	52

1 Introdução

A empresa californiana Airbnb, que foi fundada em 2008 por Joe Gebbia, Nathan Blecharczyk e Brian Chesky enquanto ainda eram estudantes, alcançou um valor de mercado superior a 80 bilhões de dólares em meados de agosto de 2023, após picos ainda mais elevados entre 100 e 131 bilhões durante o ano de 2022 ([COMPANIESMARKETCAP.COM, 2023](#)) A plataforma, que inicialmente propôs a oferta de colchões de ar pela internet, rapidamente se transformou em um catalisador global de ofertas similares, presente em mais de 81 mil cidades em 191 países, incluindo o Brasil, com mais de 220 mil anúncios que incluem quartos simples, lofts históricos e apartamentos em frente ao mar ([AIRBNB, 2018](#)). No entanto, apesar de seu impacto cultural, o Airbnb é muito mais do que uma mera ferramenta de hospedagem temporária.

Os defensores do Airbnb o veem como um novo modelo de negócios e um movimento social. Em um mundo onde a economia do compartilhamento é cada vez mais presente, o Airbnb é uma forma de cooperar e trocar serviços - com potencial para lucro pessoal - ajudando indivíduos a se tornarem microempresários e, assim, promover uma visão igualitária baseada em relações horizontais entre pessoas (SLEE, 2017, p. 22), em oposição a organizações hierárquicas, tudo graças à conectividade fornecida pela Internet. No entanto, a realidade operacional do conceito de "economia do compartilhamento" não é necessariamente tão simplista quanto parece, e complicações são encontradas em relação ao Airbnb.

Não apenas no nível do direito, algumas entidades ligadas ao lobby imobiliário, sindicatos de trabalhadores, associações de moradores e a indústria hoteleira têm se unido em oposição ao Airbnb e as mudanças ocasionadas por ele. Dentre as críticas levantadas pelos opositores, destacam-se a precarização das relações de trabalho e o encarecimento de serviços e aluguéis em bairros ou cidades com ampla gama de ofertas disponibilizadas na plataforma.

Um caso emblemático ocorreu em 2018, em Portugal, quando a Câmara Municipal de Lisboa tomou a decisão de restringir a localização dos apartamentos de aluguel temporário, como o Airbnb, e limitar o surgimento de novos empreendimentos em áreas já saturadas. A medida foi tomada devido à alta generalizada dos valores dos aluguéis, que tem levado moradores a deixarem os bairros tradicionais e valorizados da cidade. A proliferação do aluguel temporário é apontada como um dos motivos para o aumento dos preços dos aluguéis em Portugal. As mudanças na legislação permitiram que os municípios tenham maior controle sobre o assunto, incluindo a possibilidade de estabelecer “zonas de contenção” e fechar estabelecimentos que descumpram as normas estabelecidas. Tais regras incluem a obrigatoriedade de seguro para áreas comuns de edifícios com apartamentos Airbnb e a criação de um livro de regras traduzido para diferentes idiomas. Além de tais ações portuguesas, outras localidades europeias, como Berlim, Paris e Amsterdã, já adotaram medidas semelhantes ([MIRANDA, 2018](#)).

Ainda de acordo com Miranda, em 2016, a cidade de Berlim implementou um dos regulamentos mais rigorosos em relação ao aluguel de quartos, onde apenas seria permitido alugar um quarto por andar, caso contrário, o infrator estaria sujeito a uma multa de € 100 mil (R\$ 430 mil). Em Paris, várias localizações passaram a ter um limite máximo de aluguel para propriedades inteiras em plataformas online, sendo permitido apenas 120 dias por ano. Já na Holanda, as autoridades assinaram um acordo com o Airbnb em dezembro de 2016, estabelecendo um teto ainda mais restrito para os imóveis de Amsterdã, onde apenas é permitido o aluguel por um período máximo de 60 dias por ano.

Em uma situação de contradições, algumas associações de inquilinos e de combate à gentrificação comemoraram tais medidas, entidades ligadas ao turismo protestam, afirmando que prejudicam o setor.

Nessa situação, cabe destacar também a possível concorrência entre Airbnb e empresas de hospedagens mais relacionadas à hotelaria tradicional, como aquelas agenciadas pelo Booking, OTA (ou Online Travel Agency) que oferece uma plataforma para reunir reservas de hotéis, propriedades e outros prestadores de serviços em um único lugar.

Ao fazer uma reserva por meio do Booking, o usuário estabelece um contrato com o prestador de serviço, a menos que seja informado de outra forma. As informações fornecidas na plataforma são baseadas nas informações fornecidas pelos prestadores de serviço. A empresa trabalha para manter todas as informações atualizadas, mas pode levar algum tempo para atualizar descrições de texto e listas de instalações das acomodações. É importante enfatizar que o Booking não é proprietário de nenhuma acomodação, mas sim uma empresa separada que trabalha em colaboração com os prestadores de serviço ([BOOKING.COM](https://www.booking.com), 2022).

Agenciando estabelecimentos tradicionais, o Booking lida com suas próprias contradições. Para fins de exemplificação, os hotéis espanhóis aproveitaram a pandemia para reduzir sua dependência das OTAs (*Online Travel Agencies*). Em 2019, as vendas diretas das diferentes redes representavam apenas um terço do total, mas em meados de 2022 aumentaram quase oito pontos percentuais, representando 41,3% das vendas totais. Por outro lado, as vendas através de agências online, como Booking, diminuíram seis pontos percentuais em dois anos. No ano anterior à pandemia, em 2019, esse tipo de empresas representavam 38,1% das vendas de quartos de hotel, em comparação com os 32% atuais.

Teoriza-se que a comercialização direta permite que as redes hoteleiras tenham maior independência na definição de seus preços. Com a venda direta, eles não precisam lidar com as comissões recebidas pelos intermediários, como essas agências de viagens, que, segundo as associações do setor turístico, são às vezes "abusivas" ([TOURINEWS](https://www.tourinews.com), 2022).

1.1 Justificativa

Acompanhado por seu rápido crescimento, o Airbnb enfrenta oposição em várias cidades por interferir no setor de alojamento tradicional e na legislação local. Quando casas inteiras são disponibilizadas para locação via Airbnb, elas são retiradas dos mercados imobiliários locais, o que pode aumentar os preços das locações para longas temporadas ou moradias convencionais. A presença de turistas em ambientes que até então eram frequentados apenas por moradores locais, sobretudo em bairros residenciais, pode causar perturbações diversas. Adicionalmente, há anfitriões que possuem várias ofertas de hospedagens na plataforma, o que sugere que esses anfitriões podem estar operando negócios no Airbnb, papel similar ao de uma imobiliária, sem, no entanto, cumprir com suas obrigações fiscais, ainda que os impostos sobre acomodações realizadas em hotéis, pousadas ou hostels sejam frequentemente usados na promoção de turismo local (STREITFELD, 2014), de forma que o Airbnb pode causar desdobramentos diversos no setor de turismo. Logo é necessário buscar dados sobre a expansão de tal plataforma no setor de hospedagens.

Considerando a complexidade das contradições envolvendo a presença do Airbnb em determinadas comunidades (em muitas delas, onde já havia a presença e atuação de empresas hoteleiras tradicionais), é imprescindível que os órgãos responsáveis pela regulação do mercado imobiliário realizem uma análise quantitativa sobre a dimensão e influência desse serviço nessas localidades específicas. Entretanto, a aquisição e manipulação de dados referentes a esse fenômeno podem ser tarefas extremamente desafiadoras e requerem a aplicação de esforços adicionais e efetivos por parte das entidades gerenciadoras. Desta forma, apesar da relevância da observação para estudos preliminares, ocorre a inviabilidade de buscar por todas as informações de forma manual para o que tange o Airbnb. O cenário se agrava ao expandir o objeto de pesquisa para o Booking.

Cabe dizer que a competição entre Booking e Airbnb pode ser considerada de menor importância, uma vez que são plataformas que atuam em segmentos diferentes no setor de acomodações. O que é mais relevante é a concorrência entre as empresas de hospedagem tradicionais agenciadas pelo Booking - aquelas que possuem CNPJ e pagam impostos - e os imóveis cadastrados na plataforma, que funcionam como hospedagens e estão praticamente desregulamentados no Brasil. Nesse aspecto, a disponibilização de dados sobre a atuação das plataformas permitiria investir, por exemplo, se a intensificação da concorrência pode levar ou acelerar a adoção de tecnologia e redução do número de funcionários na indústria hoteleira tradicional, o que resultaria em contratos de trabalho mais precários.

Portanto, partindo de uma abordagem multidisciplinar que envolve Turismo e Ciência da Computação, este trabalho busca promover transparência sobre a atuação das plataformas envolvidas ao propor a implementação de um sistema web que permita a realização de web scraping e técnicas de ciência de dados para extrair e analisar de forma automatizadas os dados de interesse tanto do Airbnb quanto do Booking.

Caberia a este instrumental desempenhar um papel crucial na compreensão e análise do mercado imobiliário turístico ao fornecer uma visão abrangente e detalhada dos imóveis que competem diretamente com o segmento de alojamento tradicional, permitindo uma compreensão profunda de como esses imóveis têm impactado significativamente esse segmento. Essas informações são vitais não só para a tomada de decisões empresariais, mas também para a formulação e implementação de políticas públicas eficazes nesse setor.

Esse conhecimento é valioso tanto para os gestores e proprietários de hotéis, permitindo uma melhor alocação de recursos humanos e financeiros, quanto para os formuladores de políticas públicas, que podem utilizar essas informações para promover uma distribuição mais equitativa de trabalho e renda na indústria hoteleira.

Em resumo, o projeto desenvolvido neste trabalho possui um grande valor. Ele oferece uma visão realista do mercado imobiliário no setor de alojamento tradicional, permitindo uma compreensão aprofundada dos impactos e tendências. Além disso, fornece dados valiosos para a formulação de políticas públicas mais eficazes, auxilia na análise da relação trabalho e capital na hotelaria e possibilita uma avaliação crítica da atuação das plataformas de hospedagem no processo de gentrificação.

1.2 Objetivo

O principal propósito deste projeto é criar uma ferramenta viável para terceiros utilizarem na avaliação do uso de acomodações de curta duração em uma cidade específica, a fim de levantar dados que contribuam para subsidiar o trabalho do Estado e pesquisadores no sentido de implementar políticas públicas que possam se beneficiar do georreferenciamento das acomodações.

Objetivos Específicos

Buscando compreender quais são as contradições alavancadas pelas plataformas Airbnb e Booking no sentido social, econômico, espacial e legislativo, especialmente quando analisados de forma concorrencial às ofertas de hospedagens do setor tradicional de hotelaria, este trabalho tem com objetivos específicos:

1. Definir estratégias para coletar dados das plataformas Airbnb e Booking;
2. Possibilitar a visualização de informações referentes aos dados coletados;
3. Disponibilizar a ferramenta desenvolvida para uso generalizado de terceiros.

1.3 Organização do Trabalho

Este documento está organizado da seguinte forma:

Capítulo 1: Introdução, onde é realizada uma abertura ao tema do trabalho, localizando-o cultural e tecnologicamente de forma a compreender melhor sua justificativa e seus objetivos.

Capítulo 2: Revisão Bibliográfica, onde é apresentado o embasamento teórico e os principais trabalhos relacionados ao presente estudo.

Capítulo 3: Desenvolvimento, onde os métodos aplicados para a implementação do sistema proposto são apresentados de maneira mais aprofundada e precisa.

Capítulo 4: Estudo de Caso, onde são apresentados os produtos da implementação aplicados em potenciais usos de usuário.

Capítulo 5: Conclusão, onde as análises e potenciais trabalhos futuros são apresentados.

2 Revisão Bibliográfica

Neste capítulo, são descritos os trabalhos e os conceitos fundamentais para o entendimento do presente trabalho.

2.1 Fundamentação Teórica

Nesta seção, é apresentado o suporte teórico necessário para o entendimento e desenvolvimento deste trabalho.

2.1.1 Uberização

Uberização é um termo que surgiu a partir do sucesso da empresa de transporte Uber e se refere a um modelo de negócio baseado em plataformas digitais, que proporciona serviços sob demanda por meio de aplicativos. Essa tendência é caracterizada pela desintermediação, uma vez que os usuários podem se conectar diretamente aos prestadores de serviços, sem a necessidade de intermediários tradicionais como empresas ou agências.

O Airbnb é um exemplo de empresa que se aproveitou deste modelo. Ela facilita a oferta e reserva de acomodações por meio de um aplicativo ou site, conectando diretamente os proprietários de imóveis e quartos aos interessados em alugá-los. Com a plataforma, pessoas com espaço disponível em suas casas podem se tornar anfitriões e lucrar com a locação para viajantes.

Assim como o Uber, o Airbnb contribuiu para a uberização ao criar uma conexão mais direta entre oferta e demanda, permitindo que indivíduos comuns empreendam no setor de hospedagem de forma mais independente e lucrativa. Ao disponibilizar a plataforma, a empresa também oferece recursos adicionais, como sistemas de segurança, avaliações e pagamento facilitado (SLEE, 2019b).

Empresas como Uber e Airbnb se promovem se atrelando ao conceito à chamada economia de compartilhamento. Nesse modelo, pessoas disponibilizam seus ativos, como carros, casas ou habilidades, para outros usuários, cobrando uma taxa pelo uso. O objetivo central da economia de compartilhamento é maximizar a utilização dos recursos existentes e reduzir o desperdício.

Porém, autores como o próprio Slee colocam em dúvida a ideia de colaboração das plataformas digitais, considerando, por exemplo, a identificação do real papel social dessas empresas. É comum na literatura o questionamento sobre a contradição em torno do nome “economia de compartilhamento” (PORTES, 2019).

De tal forma, tem-se atrelado tais empresas à chamada “economia de plataforma”, termo que não tem a intenção de atribuir papel social ou empresarial tradicional à temática. Tal modelo

envolve o uso de tecnologia e plataformas digitais para facilitar a conexão entre produtores e consumidores, permitindo a troca de bens, serviços e informações. Uma plataforma digital atua como intermediário, facilitando e coordenando transações entre usuários, fornecendo as ferramentas necessárias para viabilizar negócios.

Diferenciando ambos os modelos de economia, a economia de plataforma é um conceito mais amplo que envolve o uso de tecnologia para conectar diferentes partes do mercado, enquanto a economia de compartilhamento se concentra no compartilhamento de recursos ociosos. No entanto, muitas vezes esses dois conceitos se sobrepõem e podem ocorrer simultaneamente dentro de uma mesma empresa ou plataforma.

2.1.2 Web Scraping

Web scraping (também conhecida como raspagem ou coleta de dados web) é definido como um processo de mineração de dados automatizada, buscando extrair uma quantidade elevada de informações relevantes de um site para posterior análise e eventuais conclusões, a partir destes dados coletados (MITCHELL, 2019).

O *web scraping* pode ser realizado por diversas técnicas. No escopo desse trabalho, o scrap foi realizado a partir do consumo de APIs externas, próprias para o fornecimento facilitado desses dados, e do uso de ferramentas de automatização de navegadores, que simulam a atividade de um usuário em uma determinada página através do *browser*, por meio da biblioteca Selenium, por exemplo. Tais ferramentas são descritas a seguir.

2.1.2.1 Airbnb Data Collection

Implementado por Slee em Python, esse algoritmo funciona fazendo uso de bibliotecas específicas para coleta dos dados do Airbnb e a manipulação destes.

Destaca-se nessa etapa o uso de uma API própria do Airbnb. Essa API é responsável por retornar listas de acomodações disponíveis em um determinado local via uma requisição com protocolos de comunicação (por exemplo, HTTP). A requisição faz uso de uma chave pública para acessar acomodações com metadados como preço, localização, número de hóspedes, número de quartos, tipo de tarifa entre outros parâmetros.

Nesse caso, a chamada da API retorna um objeto JSON que corresponde a uma lista de acomodações contidas em uma página de resultados específica de pesquisa. Para realizar uma requisição a tal API, são importantes dados como sua chave de acesso, encontrada no site do Airbnb, e os *proxies* utilizados para cada requisição.

Sobre parâmetros específicos requisitados pela API, pode-se destacar o tipo de pesquisa, a quantidade de itens retornados por páginas e as coordenadas em si da busca (que, no caso, correspondem às quatro coordenadas retornadas por uma Caixa Delimitadora).

Além disso, também é interessante que se definam critérios como intervalos de tempo entre uma requisição e outra e a quantidade máxima de tentativas de se realizar determinada requisição, justamente com o intuito de evitar que a plataforma bloqueie as chamadas.

Além do uso desta API, o *Airbnb Data Collection* também faz uso da biblioteca *geopy* para geolocalização, que permitiu captar as coordenadas de uma região especificada pelo nome ou então realizar o caminho contrário, identificando a região através das coordenadas oferecidas. As sucessivas buscas automatizadas do algoritmo continuam ocorrendo enquanto novas acomodações forem encontradas e a caixa delimitadora puder ser fracionada em partes menores da caixa delimitadora original (SLEE, 2019a). O andamento das pesquisas pode variar entre curtos espaços de tempo, no caso de áreas menores e/ou com poucos resultados, ou espaços maiores no caso de áreas maiores e/ou com maiores resultados e o acesso á eles se faz a medida que ocorre a execução de métodos internos voltados à a aquisição, tratamento e inserção desses dados, não necessariamente imediatamente após seu início.

Para o bom funcionamento dessa busca, é preciso definir algumas configurações para a Pesquisa. A Figura 2.1 representa algumas dessas configurações necessárias, assim como a explicação fornecida por Slee para cada uma delas.

2.1.2.2 Selenium

A biblioteca Selenium¹ permite simular e automatizar a atividade em um navegador como um usuário real o usaria. Assim, é possível lidar com eventos em Javascript (por exemplo, cliques em um botão, envio de um formulário, carregamento de uma página) e encontrar os elementos pertinentes através de estratégias de localização.

Tais estratégias geralmente envolvem identificar atributos, sintaxes ou expressões específicas dentro do código-fonte da página. Assim, um elemento pode ser encontrado por sua(s) classe(s), âncoras ou demais *tags*, ou pela hierarquia de posicionamento dos elementos. Para realizar tal localização, é comum o uso de XPath², linguagem usada para encontrar elementos dentro da hierarquia de nodos de um arquivo XML ou HTML. As expressões definidas pelo XPath, então, percorrem estes nodos a fim de identificar ou localizar os elementos que satisfaçam às suas condições e expressão de caminho, e extrair informações a partir deles (por exemplo, seus textos ou atributos de *tag*).

Para realizar a automação de ações web com o Selenium, é necessário importar a biblioteca *webdriver* e o analisador sintático HTML do *lxml*. É então preciso iniciar o *webdriver* do Selenium para posteriormente acessar a página desejada e obter o código HTML da mesma.

Depois disso, deve-se utilizar o XPath para localizar elementos na página e, com tais elementos localizados, podem-se realizar as ações desejadas.

¹ <https://robotframework.org/SeleniumLibrary/SeleniumLibrary.html>

² <https://www.w3.org/TR/xpath/>

```
[SURVEY]
# -----
# These don't often need to be changed
# -----

fill_max_room_count = 50000

# -----
# For the special case of doing a global sample of Airbnb listings, room
# values are chosen at random for a range with this as the maximum.
# For other searches, this value is not needed.
# -----

room_id_upper_bound = 20000000

# -----
# Maximum number of pages to loop over in search,
# for a given area, room_type, and number of guests
# search_max_pages = 10
# -----

search_max_pages = 20

# -----
# Maximum number of guests in search
# If set to 1 (recommended), searches will not include a "guests" parameter
# -----

search_max_guests = 1

# -----
# Maximum zoom level for bounding box search
# For search_max_guests = 1 or for larger areas, set to 8
# For search_max_guests > 1 and a city (as opposed to larger area),
# 6 may be sufficient
# For search_do_loop_over_prices = 0, increase further (eg 12)
# -----
```

Figura 2.1 – Arquivo de configuração de variáveis de Pesquisa

Por fim, é importante finalizar o *webdriver* para garantir a correta execução do processo de automação. O algoritmo representado na Figura 2.2 exemplifica esse processo, instanciando um *webdriver*, localizando um elemento, realizando a ação de clicar sobre ele, e posteriormente o encerrando. A sintaxe do caminho XPath é formulada substituindo *tag*, "atributo" e "valor" pelos respectivos elementos a serem encontrados.

```
from selenium import webdriver
from lxml import html

driver = webdriver.Chrome(executable_path=r'C:\caminho.exe')
url = 'https://www.exemplo.com'
driver.get(url)
elemento = driver.find_element(BY.XPATH, '//tag[@atributo="valor"]')
elemento.click()
driver.quit()
```

Figura 2.2 – Modelo de algoritmo simples para scrap via selenium

O algoritmo acima exhibe um caso onde os elementos são encontrados por seu XPath, mas também poderiam ser encontrados por outros métodos como id ou nome da classe atribuída a

tag ou seletor do arquivo de estilos.

2.1.3 Agrupamento

A ferramenta desenvolvida realiza também agrupamento de acomodações similares. Para isso, são usadas técnicas de agrupamento de dados, baseadas em aprendizado de máquina não supervisionado. Aprendizado de máquina não supervisionado é uma categoria de aprendizado de máquina, em que é possível inferir modelos realizando o treinamento com dados não rotulados. Assim, não há categorias prévias para *fit* dos dados, elas são identificadas e se apresentam computacionalmente através dos dados preparados e treinados. Métodos não supervisionados costumam ser aplicados em áreas como estoque de mercado, segmentação de marketing e processamento de linguagem natural (ARTASANCHEZ; JOSHI, 2020).

A grande maioria dos algoritmos de aprendizado analisa pontos de dados e, a partir da generalização das similaridades do conjunto, é capaz de identificar grupos distintos e naturais que segmentam os dados e tornam possível obter *insights* até então não imaginados.

Agrupamento (ou ainda, "clusterização"), aqui, se refere a uma técnica de análise que agrupa dados, via aprendizado de máquina não supervisionado, com o intuito de gerar n grupos a partir da semelhança ou dissimilaridade entre os elementos, de forma a facilitar a interpretação e visualização dos resultados (SAXENA et al., 2017). Assim, os elementos de um grupo devem ser semelhantes entre si e, em conjunto, possuem maior grau de diferença dos elementos que não pertencem ao mesmo grupo que eles. Cada um destes grupos pode ser determinado como *cluster*.

Neste trabalho, três tipos de agrupamento são abordados: agrupamento por particionamento, agrupamento por densidade e agrupamento hierárquico. O agrupamento por particionamento divide os pontos de dados em k grupos. Cada grupo é um conjunto exclusivo e não é compartilhado por outros pontos. A clusterização por particionamento é eficaz quando o número de grupos é conhecido e os grupos têm uma forma circular ou esférica (SØRENSEN, 1948).

Um dos algoritmos mais populares para agrupamento por particionamento é o chamado *k-means*. Capaz de processar grandes conjuntos de dados, esse é um método comumente utilizado na análise e mineração de dados, e tem diversas aplicações em campos como marketing, biologia e ciência de dados.

O algoritmo *k-means* começa a atribuir aleatoriamente k centroides no conjunto de dados. Então, se calcula a distância euclidiana, ou outra medida de distância numérica, entre cada ponto e os centroides designados, e se atribui cada ponto ao grupo do centroide mais próximo. Portanto, se recalcula os centroides para que estejam no centro de cada um dos aglomerados formados. Este processo se repete até que os centroides não mudem significativamente, ou se cumpra um determinado número de iterações (HUANG, 1998).

A precisão do algoritmo e sua eficácia dependerão do valor de k escolhido e a definição desse valor de *clusters* deve ser baseada na interpretação do usuário a cerca do conjunto de dados

e na homogeneidade dos grupos. Além disso, o resultado do algoritmo pode variar dependendo do subconjunto de dados utilizado, pelo que pode ser necessário realizar várias iterações em conjunto com validações cruzadas para garantir que o resultado seja robusto.

Resumindo, *k-means* é um algoritmo de agrupamento não supervisionado por particionamento que utiliza a distância euclidiana para atribuir cada ponto a um centroide, e se utiliza para agrupar dados em *clusters* com base em sua similitude.

Entretanto, ao se valer do cálculo da distância, o algoritmo *k-means* se limita a trabalhar apenas com dados numéricos. Tal limitação restringe seu uso em diversas aplicações em que o conjunto de dados possui valores categóricos e quando não há uma abordagem para converter estes dados em campos numéricos que mostre bons resultados (por exemplo, por aumentar a dimensionalidade do conjunto).

Teorizado por Huang, *k-modes* surge como uma variação do *k-means*, no qual as dissimilaridades substituem as distâncias, mensurando o número de incompatibilidades entre dois objetos para determinar sua semelhança. Em vez de calcular a média dos dados, essa técnica utiliza modos, que são vetores de elementos que minimizam as dissimilaridades entre o vetor e cada objeto do conjunto de dados. O número de modos é igual ao número de *clusters* necessários, tendo em vista que eles exercem a função de centroides.

O método de clusterização hierárquica, por sua vez, cria uma estrutura hierárquica de *clusters*, onde *clusters* menores são agrupados para formar *clusters* maiores. Essa hierarquia pode ser representada em forma de árvore ou dendrograma. A clusterização hierárquica pode ser aglomerativa, onde cada ponto começa como um *cluster* individual e é gradualmente agrupado, ou divisiva, onde todos os pontos começam como um único *cluster* e são divididos sucessivamente. Não há um algoritmo específico para a clusterização hierárquica, já que diferentes abordagens podem ser usadas. Uma técnica comum é medir a dissimilaridade entre pontos de dados e usar isso para agrupá-los. A clusterização hierárquica é útil quando a estrutura dos dados é desconhecida e quando se deseja visualizar a relação entre os *clusters* em diferentes níveis de granularidade.

Um exemplo de método de agrupamento hierárquico é o *BIRCH* (ZHANG; RAMAKRISHNAN; LIVNY, 1997). Este algoritmo consiste em três fases principais: construção da árvore CF (*Clustering Feature*), construção dos *subclusters* e refinamento dos *clusters*.

Na primeira fase, é construída uma árvore CF que contém informações resumidas sobre conjuntos de pontos de dados. A árvore é atualizada iterativamente, adicionando novos pontos de dados. Na segunda fase, o algoritmo constrói *subclusters* a partir dos nós externos da árvore CF. Cada nó externo representa um *cluster* potencial, que consiste em um conjunto de pontos de dados. Os *subclusters* são construídos para agrupar os pontos de dados com base em sua proximidade em relação à árvore CF. Os pontos de dados são atribuídos a um *subcluster* existente ou um novo *subcluster* é criado para acomodá-los. Na terceira fase, ocorre um refinamento dos *clusters*. Isso envolve combinar os *subclusters* para formar *clusters* maiores e mais estáveis. Isso é

feito calculando os centroides dos *subclusters* e identificando pontos próximos a esses centroides para atualizar a estrutura de cada *cluster*.

Uma das principais vantagens do algoritmo *BIRCH* é sua eficiência computacional. Isso significa que ele é capaz de lidar com grandes conjuntos de dados de forma eficiente, já que utiliza estruturas de dados compactas para representar as informações dos *clusters*. Além disso, o algoritmo possui escalabilidade, sendo capaz de trabalhar tanto com um grande número de objetos quanto com um grande número de atributos. Ele realiza o agrupamento em tempo linear, em relação ao tamanho do conjunto de dados. Outra vantagem do algoritmo é a sua capacidade de detectar e lidar com *outliers* presentes nos dados. Isso evita que valores atípicos influenciem negativamente o resultado do agrupamento.

No entanto, o algoritmo *BIRCH* também apresenta algumas desvantagens. Uma delas é a sua sensibilidade aos parâmetros escolhidos. O desempenho do algoritmo pode variar dependendo dos valores escolhidos para parâmetros como o limiar de semelhança e o fator de compactação. A escolha inadequada desses parâmetros pode comprometer a qualidade do agrupamento. Outra limitação do algoritmo diz respeito a sua capacidade de identificar e representar corretamente *clusters* com formas não-esféricas. Isso ocorre porque o *BIRCH* pressupõe que os *clusters* sejam aproximadamente esféricos e isotrópicos, o que pode afetar sua capacidade de identificar corretamente *clusters* com formas mais complexas. Por fim, também é importante destacar que a ordem de inserção dos dados pode afetar o resultado do agrupamento. Diferentes ordenações dos mesmos dados podem levar a resultados distintos.

Por fim, o agrupamento por densidade identifica *clusters* com base na densidade dos pontos de dados. Ele considera que os pontos em áreas densas são parte de um *cluster*, enquanto os pontos menos densos estão fora dele. A clusterização por densidade é útil quando os *clusters* têm formas e tamanhos irregulares e quando a quantidade de ruído (dados que não pertencem a nenhum *cluster* específico) é alta (HAN; KAMBER; PEI, 2001).

Um exemplo de algoritmo de clusterização por densidade é o *DBSCAN*, amplamente utilizado em mineração de dados e análise de dados espaciais, para realizar tarefas de agrupamento. Esse algoritmo é baseado na densidade dos pontos de dados para agrupá-los de forma eficiente (ESTER et al., 1996).

O algoritmo começa com a seleção de pontos de dados, que possuem o número mínimo de pontos vizinhos em uma determinada distância, chamada raio. Caso haja pontos suficientes dentro do raio, esses pontos são considerados parte de um *cluster*. Em seguida, o algoritmo repete esse processo para os pontos vizinhos recém-encontrados, formando assim um novo *cluster*. Esse processo é repetido até que todos os pontos de dados tenham sido visitados.

Dentre suas vantagens, pode-se destacar sua capacidade de identificar automaticamente o número de *clusters* presentes nos dados. Diferentemente de outros algoritmos de agrupamento, como o *K-means* (e seus derivados), o *DBSCAN* não exige que o número de *clusters* seja

especificado previamente. Ele é capaz de detectar automaticamente o número de *clusters* com base na densidade dos dados.

Além disso, essa é uma abordagem robusta em relação à presença de ruídos e outliers nos dados. Ele é capaz de lidar com pontos de dados que não pertencem a nenhum *cluster* específico, ignorando-os ou considerando-os como clusters separados, dependendo das configurações do algoritmo. Outra vantagem é a sua flexibilidade em relação às formas e tamanhos de clusters. Ele é capaz de detectar clusters com formas arbitrárias, como clusters de forma irregular ou clusters densos e alongados.

Porém, há algumas desvantagens e limitações a serem consideradas ao utilizá-lo. Uma delas é a sensibilidade aos parâmetros. O desempenho do *DBSCAN* depende da configuração correta dos parâmetros, como o raio e o número mínimo de pontos. Se esses parâmetros não forem ajustados adequadamente, os resultados da clusterização podem não ser satisfatórios.

Além disso, o *DBSCAN* pode apresentar um baixo desempenho em conjuntos de dados de alta dimensionalidade. Isso ocorre porque a densidade dos pontos pode se tornar menos significativa em espaços de alta dimensionalidade. Outra limitação deste algoritmo é sua dificuldade em encontrar clusters de densidades diferentes em um mesmo conjunto de dados, especialmente se a diferença de densidade não for acentuada.

Em resumo, o algoritmo *DBSCAN* é uma técnica de agrupamento flexível e robusta, especialmente adequada para conjuntos de dados com densidade variada e presença de ruídos. No entanto, é necessário ajustar corretamente os parâmetros e considerar suas limitações em relação a conjuntos de dados de alta dimensionalidade e clusters de densidades diferentes.

2.1.4 Back-end e Front-end

Nesta seção, são apresentados os conceitos de back-end (a parte "escondida" do sistema, que atua com o processamento dos dados) e front-end (a parte "visível", por onde o usuário consegue realizar sua intenção) e as ferramentas e *frameworks* FastAPI e Next.js, voltadas, respectivamente, para cada uma dessas especialidades de código.

2.1.4.1 Back-end

O back-end é a parte do sistema ou aplicação que fica "atrás" da interface de usuário, lidando com as funcionalidades e lógica de negócio. O back-end é responsável por se comunicar com um SGBD, processar os dados disponibilizados por ele e gerar respostas que serão enviadas ao front-end. Também é o back-end o responsável por definir as regras de negócio que serão implementadas, bem como as validações necessárias para garantir a segurança e integridade dos dados.

Para seu desenvolvimento, foi usado o *framework* FastAPI³, voltado para desenvolvimento

³ <https://fastapi.tiangolo.com/>

de APIs em Python. O framework é chamado de "rápido" devido ao seu suporte à tipagem estática e geração automática e eficiente de código, o que torna a aplicação muito mais rápida em comparação com outros *frameworks* como Flask e Django.

Apesar de ser recente e não possuir tanta documentação quanto os outros dois *frameworks*, o FastAPI se destaca por seu desempenho e pelo suporte à tipagem estática via Pydantic, que permite definir os tipos de dados esperados nas entradas e saídas da API.

A tipagem via Pydantic é realizada a partir de um *Base Model*, classe que fornece recursos para criar modelos de dados ou estendê-los de forma a definir seus atributos e métodos. Ao usar os *Models*⁴, é possível definir os atributos do modelo de dados, especificar tipos de dados, validar os dados de entrada e definir a obrigatoriedade ou não de alguns campos.

O FastAPI usa os recursos de validação fornecidos pelo Pydantic para validar os dados de entrada antes de processá-los e retorna mensagens de erro caso os dados não atendam aos critérios estabelecidos. Isso garante que seu código seja mais resiliente a erros e evita que dados inválidos sejam processados. Por meio de tal integração, o *framework* também consegue converter automaticamente os dados de entrada no tipo de dado esperado definido no *Base Model*, o que é reduz a necessidade de realizar conversões manualmente e simplifica o código.

Outra funcionalidade do FastAPI é a de *Background Tasks*. *Background Tasks* são tarefas assíncronas executadas em segundo plano, de forma assíncrona em um processo separado, sem a necessidade de aguardar sua conclusão antes de retornar a resposta para o cliente. Elas são úteis para melhorar a capacidade de resposta e desempenho da aplicação quando se deseja executar tarefas demoradas em segundo plano. Isso permite que a resposta da requisição em si seja enviada rapidamente para o cliente, sem esperar pela conclusão dessa tarefa. Outra vantagem é que outras requisições podem ser atendidas enquanto tais tarefas ocorrem em segundo plano.

Para que o front-end acesse tais dados disponibilizados pelo back-end, foi configurada a política de CORS (sigla para "*Cross-Origin Resource Sharing*"), que permite ou bloqueia solicitações de recursos de diferentes origens, impedindo que um site ou aplicação execute ações maliciosas em outro. Este é um conceito importante em APIs, já que algumas solicitações podem ser bloqueadas por motivos de segurança.

2.1.4.2 Front-end

Front-end refere-se à parte visual e interativa de um aplicativo, site ou sistema. É a interface com a qual os usuários interagem diretamente. O front-end é responsável por criar uma experiência de usuário agradável e intuitiva, garantindo que o usuário consiga navegar e interagir com o aplicativo ou site de forma eficiente e satisfatória. Além disso, o front-end também envolve a criação de layouts responsivos, que se adaptem a diferentes tamanhos de tela, como smartphones e tablets.

⁴ <https://docs.pydantic.dev/latest/usage/models/>

Assim, ele é responsável por criar a parte visual e interativa de um aplicativo ou site, proporcionando uma experiência positiva e facilitando a interação dos usuários com a plataforma.

Em seu desenvolvimento, são utilizadas linguagens de programação como HTML, CSS e JavaScript. O HTML é responsável pela estruturação e organização do conteúdo, enquanto o CSS é usado para estilizar e formatar a aparência visual do aplicativo ou site. O JavaScript é uma linguagem de programação que permite adicionar interatividade e dinamismo à página.

Para o desenvolvimento do sistema proposto, foi usado o Next.js⁵, que é uma estrutura de aplicativo da web de código aberto lançada em 2016. É baseado em React.js e é especialmente projetado para desenvolver aplicativos de página única com facilidade e eficiência. É uma das estruturas mais populares para o desenvolvimento front-end de aplicativos da web e é amplamente utilizada em todo o mundo.

Next.js tem várias vantagens e recursos úteis. Um dos principais recursos é que ele usa renderização do lado do servidor (SSR) e renderização do lado do cliente (CSR), dependendo do contexto do usuário, o que torna os aplicativos mais rápidos e eficientes. Além disso, os arquivos em cache permitem que as páginas sejam carregadas instantaneamente.

Outra vantagem do Next.js é a sua capacidade de criar aplicativos híbridos, que podem ser executados tanto no servidor quanto no cliente. Isso permite que os aplicativos sejam mais escaláveis e os desenvolvedores aprimorem a usabilidade do aplicativo. Além disso, o Next.js tem muitos plugins e componentes pré-escritos disponíveis na comunidade React, permitindo que os desenvolvedores integrem seus aplicativos com eficiência a outros aplicativos, serviços e sistemas.

Para exibir as acomodações geograficamente posicionadas, uma alternativa é a biblioteca Leaflet. Esta é uma solução em código aberto desenvolvida em JavaScript para a criação de mapas interativos em páginas web. Com a sua utilização, é possível disponibilizar recursos personalizáveis, como camadas, controles de zoom e indicadores, para aprimorar a experiência do usuário. Além disso, o Leaflet oferece suporte para múltiplos provedores de mapas, incluindo o *OpenStreetMap*.

Para simplificar ainda mais o processo de integração do Leaflet com o Next.js, é possível a utilização da biblioteca React-Leaflet e, por fim, é importante lembrar que o Leaflet é frequentemente utilizado em conjunto com outras ferramentas de desenvolvimento web para aprimorar e incrementar a visualização de mapas em aplicações geoespaciais, tornando essa solução uma das mais populares e flexíveis em termos de desenvolvimento de mapas interativos.

Já a Mapbox é uma API que fornece ferramentas e serviços para desenvolver aplicativos com recursos de mapas personalizados. Para usos mais comuns, uma chave de acesso única é tudo o necessário para a sua utilização dentro de uma aplicação, ainda que seja possível especificar dados como o tipo de estilizações ou mesmo criar uma. Em conjunto com o Leaflet, é possível

⁵ <https://nextjs.org/>

integrar a Mapbox para oferecer estilos personalizados em seus mapas e torná-los ainda mais únicos e exclusivos.

Vercel⁶ é uma plataforma para hospedagem de aplicativos em nuvem que permite que os desenvolvedores implantem seus aplicativos com facilidade e rapidez. As configurações gerais de uma aplicação Next.js permitem que a porta seja redirecionada automaticamente e já preparada para suportar o SSR do framework.

2.2 Trabalhos Relacionados

Buscando encontrar padrões nas formas como as características das acomodações do Airbnb e os atributos dos anfitriões influenciam as avaliações dos hóspedes e quais os principais tópicos das mesmas, [Rodrigues \(2021\)](#) atuou com um conjunto de dados contendo quase 4 milhões de avaliações das cidades de Milão, Lisboa, Amsterdã, Toronto, São Francisco e Sydney para realizar o processamento do texto e investigar as normas sociais e de mercado das avaliações, bem como os sentimentos dos convidados e a distribuição dos temas. Essa abordagem permite que tanto anfitriões quanto pesquisadores entendam o quanto as opiniões dos avaliadores são importantes e tem o potencial de ditar a qualidade dos serviços prestados.

Ao analisar os dados das acomodações de mais de 52 mil anúncios do Airbnb em Nova York, referentes ao mês de novembro de 2018, [Gupta \(2019\)](#) fornece alguns insights interessantes a cerca de critérios interessantes de análise sobre tal conjunto de dados. Aqui, ele não apenas aponta o crescimento exponencial da quantidade de anúncios ofertados no período de 10 anos, mas também possibilita a imaginação de uma interface interativa onde os anúncios podem ser filtrados a partir de diversos parâmetros ditados pelo usuário (por exemplo, valores de classificação média mínima, capacidade de acomodação ou tipo de propriedade).

Tentando identificar como fatores de etnia, renda e instrução impactam o Airbnb, [Cansoy e Schor \(2016\)](#) realizam a análise de dados coletados da plataforma combinados com informações disponíveis no *US Census* nas mesmas regiões. Realizada a análise, os autores teorizam que anfitriões racializados costumam ter anúncios com preços e classificações menores, gerando menos renda para o Airbnb, ao contrário dos anfitriões com maiores instrução e acesso a capital, cujos preços e classificações são mais altos e representam uma parte maior da receita da plataformas.

Corroborando a justificativa dessa pesquisa, [Cox e Slee \(2016\)](#) discorrem sobre a "defesa de transparência" defendida pelo Airbnb e como tal ação pode ter sido uma estratégia para esconder a maior fonte de sua receita (advinda justamente dos anfitriões de melhores condições socioeconômicas) e manter sua boa imagem para o público geral e não refletir uma mudança real de paradigma.

Buscando compreender como o Airbnb se faz instrumento de desigualdade no sistema

⁶ <https://vercel.com/>

do capital, Slee (2020) propõe um script para coletar informações sobre os anúncios presentes na plataforma Airbnb. Como explicado pelo autor, tal script, intitulado *Airbnb Data Collection* (melhor descrito no tópico 2.1.2.1) realiza uma coleta automatizada de dados do site do Airbnb para uma cidade específica em uma data específica ou próximo a ela. A presente pesquisa estende Slee ao realizar não apenas o *web scraping* para as acomodações desta plataforma, mas também para o Booking, fazendo com que se torne possível estabelecer uma análise comparativa entre ambas. A extração dos dados do Booking é também embasada no trabalho de Oswaldo Alcala, que propõe um *web scraping* do Booking (ALCALA, 2019).

No script desenvolvido por Alcala, o *web scraping* é realizado a partir do WebDriver disponibilizado pela biblioteca Selenium (melhor descrita no tópico 2.1.2.2), que acessa o domínio das acomodações do Booking e coleta os dados de nome, localização, comodidades populares e classificação média presentes nestes links. Este trabalho estende a pesquisa de Alcala ao expandir os tipos de dados coletados.

3 Desenvolvimento

Buscando cumprir os objetivos do presente trabalho, posterior à Revisão Bibliográfica (vide Capítulo 2), no desenvolvimento desse trabalho, foi implementado um web scraping para extrair, mediante a execução de consultas sucessivas e automatizadas de ofertas do Airbnb e do Booking referentes às hospedagens em uma cidade arbitrária. Posteriormente, desenvolveu-se um sistema web para interação do usuário com essa coleta, de forma que todos os parâmetros da busca poderiam ser definidos diretamente por tal usuário. Tal processo será descrito com maior detalhamento nas seções a seguir.

3.1 Concepção de Software

É comum que o processo de desenvolvimento de um software seja comprometido devido a objetivos concorrentes das áreas envolvidas (de vendas, de design, de desenvolvimento, por exemplo), o que pode ocasionar em eventual descumprimento de prazo, desenvolvimento desestruturado e baixa qualidade do produto final (GONÇALVES et al., 2016). Visando evitar tal cenário, é necessária uma abordagem de engenharia.

A engenharia de software, então, "engloba processos, métodos e ferramentas que possibilitam a construção de sistemas complexos baseados em computador dentro do prazo e com qualidade", propondo atividades estruturais que envolvem comunicação, planejamento, modelo, construção e entrega do software (PRESSMAN; MAXIM, 2021).

Sendo assim, dentro deste campo de engenharia, a concepção de software é o processo de criação e planejamento de um software, envolvendo desde a ideia inicial até a definição dos requisitos e funcionalidades necessárias para sua construção. É nessa etapa que se determina a arquitetura e as tecnologias a serem utilizadas, além da elaboração do modelo de negócio e estratégias de marketing.

A etapa de concepção é essencial para o sucesso do software, uma vez que é nela que forma-se a base para todas as etapas seguintes do desenvolvimento. É importante que haja um entendimento claro do problema a ser resolvido e das expectativas dos usuários e potenciais usuários, além de uma visão clara dos recursos e limitações técnicas disponíveis.

Abaixo, será descrita a concepção de software da plataforma proposta.

3.1.1 Objetivo

Esta concepção especifica o DashAcomodações: um sistema web para acessar os dados provenientes das plataformas Airbnb e Booking, sobre acomodações disponibilizadas nas mesmas.

3.1.2 Missão e Descrição do Produto

O DashAcomodações é um sistema web com o objetivo de coletar informações públicas sobre anúncios disponibilizados por Airbnb e/ou Booking com a missão de auxiliar pesquisadores.

3.1.3 Escopo Positivo e Negativo

3.1.3.1 Positivo

O DashAcomodações permite que o usuário solicite uma nova pesquisa por acomodações a determinada região de acordo com alguns parâmetros de interesse. Os dados coletados são disponibilizados apenas publicamente via API/navegação no Airbnb e via navegação no Booking.

3.1.3.2 Negativo

O DashAcomodações não permite que o usuário acesse ou altere qualquer informação ou estrutura no banco de dados que não seja via os métodos próprios para tal. Após coletados, os dados das acomodações não são excluídos. Os dados das pesquisas são limitadas pelos parâmetros da API e da navegação via navegador. Não foram executados testes automatizados e a segurança também pode ser melhorada. O sistema não é voltado para agendar viagens. Os dados retornados pelo sistema podem não corresponder 100% aos dados reais. As pesquisas buscam apenas por anúncios futuros e a disponibilização deles no DashAcomodações pode demorar algum tempo, de acordo com o fluxo da pesquisa.

3.1.4 Descrição dos Atores

O DashAcomodações possui três tipos de usuários: Visitante, Pesquisador e Administrador, todos descritos na Tabela 3.1.

Tabela 3.1 – Descrição dos Atores do DashAcomodações

Ator	Descrição
Usuário Visitante	Qualquer pessoa física ou jurídica que acesse o sistema apenas para visualizar informações públicas
Usuário Pesquisador	Qualquer pessoa física ou jurídica que faça uso do sistema a fim de visualizar ou solicitar dados, com foco em pesquisadores e agentes desenvolvedores de políticas públicas
Usuário Administrador	Pessoa (ou grupo de pessoas) específica responsável pela manutenção do sistema

A hierarquia entre os atores está representada na Figura 3.1. Tal hierarquia indica que um Usuário Pesquisador pode executar tanto os seus serviços quanto os serviços associados ao Usuário Visitante. Um Usuário Administrador, por sua vez, pode executar tanto os seus serviços quanto os serviços associados ao Usuário Pesquisador.

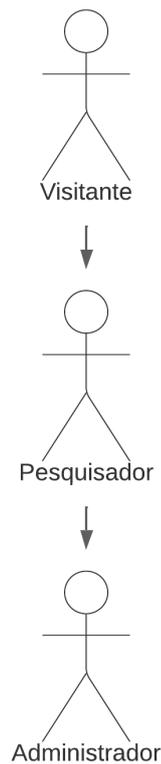


Figura 3.1 – Hierarquia entre Atores do DashAcomodações

3.1.5 Diagrama de Caso de Uso

O Diagrama de Caso de Uso na Figura 3.2 indica como os usuários do DashAcomodações se relacionam com os serviços oferecidos pelo sistema que, por sua vez, estão descritos na tabela 3.2

Tabela 3.2 – Descrição dos Serviços oferecidos pelo DashAcomodações

Serviço	Descrição
Serviço de Autenticação	permite ao usuário se conectar ao sistema ou gerenciar os dados relacionados a este acesso ao sistema
Serviço de Navegação	permite ao usuário visualizar os dados coletados dos anúncios do Airbnb e/ou Booking cujas pesquisas foram solicitadas por ele
Serviço de Solicitação de Pesquisa	permite ao usuário gerenciar o status de uma pesquisa solicitada por ele (a ser realizada, a ser continuada ou atualmente pausada)
Serviço de Gestão de Usuários	possibilita ao Administrador gerenciar os níveis de permissão de outros usuários (Administradores ou Visitantes) ao sistema
Serviço de Gestão do Sistema	possibilita ao Administrador manter a integridade do sistema através de manutenção e/ou melhorias

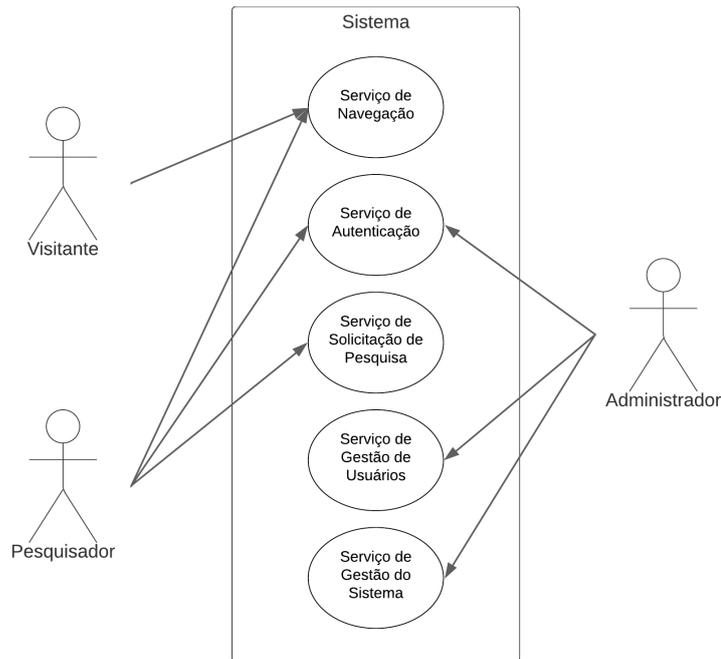


Figura 3.2 – Diagrama de Caso de Uso do DashAcomodações

3.1.6 Backlog

3.1.6.1 Serviço de Autenticação

- Solicitar cadastro no sistema
- Realizar login no sistema
- Recuperar senha esquecida
- Alterar dados cadastrais

3.1.6.2 Serviço de Navegação

- Visualizar listagem de pesquisas solicitadas
- Visualizar informações de anúncios coletados em uma pesquisa específica

3.1.6.3 Serviço de Solicitação de Pesquisa

- Solicitar nova pesquisa
- Continuar pesquisa
- Parar pesquisa

3.1.6.4 Serviço de Gestão de Usuários

- Aceitar solicitação de cadastro
- Remover usuário
- Alterar nível de permissão do usuário

3.1.6.5 Serviço de Gestão do Sistema

- Realizar backup
- Atualizar Sistema
- Restaurar sistema

3.1.7 Estórias de Usuário

3.1.7.1 Serviço de Autenticação

1. Solicitar cadastro no sistema

- Como um Usuário Pesquisador
- Com o objetivo de conseguir permissão para acessar o sistema
- Eu gostaria de solicitar o meu cadastro no sistema a um Usuário Administrador

2. Realizar login no sistema

- Como um Usuário Cadastrado
- Com o objetivo de acessar o sistema
- Eu gostaria de realizar login

3. Recuperar Senha Esquecida

- Como um Usuário Cadastrado
- Com o objetivo de redefinir a minha senha
- Eu gostaria de recuperar a senha esquecida

4. Alterar dados cadastrais

- Como um Usuário Cadastrado
- Com o objetivo de atualizar os meus dados de acesso
- Eu gostaria de atualizar os dados cadastrais

3.1.7.2 Serviço de Navegação

1. Visualizar listagem de pesquisas públicas
 - Como um Visitante
 - Com o objetivo de encontrar informações gerais sobre as pesquisas solicitadas por outros usuários
 - Eu gostaria de visualizar a listagem das pesquisas solicitadas
2. Visualizar listagem de pesquisas solicitadas
 - Como um Pesquisador
 - Com o objetivo de encontrar informações gerais sobre as pesquisas solicitadas por mim
 - Eu gostaria de visualizar a listagem das pesquisas solicitadas
3. Visualizar informações de anúncios coletados em uma pesquisa específica
 - Como um Visitante
 - Com o objetivo de encontrar informações sobre as acomodações coletadas nas pesquisas solicitadas
 - Eu gostaria de visualizar informações de anúncios coletados em uma pesquisa específica por vez

3.1.7.3 Serviço de Solicitação de Pesquisa

1. Solicitar nova pesquisa
 - Como um Pesquisador
 - Com o objetivo de coletar novas informações sobre acomodações
 - Eu gostaria de solicitar uma nova pesquisa
2. Continuar pesquisa
 - Como um Pesquisador
 - Com o objetivo de voltar a executar uma pesquisa interrompida por status de erro ou conclusão
 - Eu gostaria de continuar uma pesquisa previamente solicitada
3. Parar pesquisa
 - Como um Pesquisador

- Com o objetivo de interromper intencionalmente a execução de uma pesquisa em andamento
- Eu gostaria de parar uma pesquisa previamente solicitada

3.1.7.4 Serviço de Gestão de Usuário

1. Aceitar solicitação de cadastro

- Como um Usuário Administrador
- Com o objetivo de permitir que um novo usuário acesse as páginas privadas do sistema
- Eu gostaria de aceitar a solicitação de cadastro

2. Remover usuário

- Como um Usuário Administrador
- Com o objetivo de impedir que um usuário acesse as páginas privadas do sistema
- Eu gostaria de remover esse usuário

3. Alterar nível de permissão do usuário

- Como um Usuário Administrador
- Com o objetivo de promover um usuário Pesquisador a Administrador ou vice-versa
- Eu gostaria de alterar o nível de permissão desse usuário

3.1.7.5 Serviço de Gestão de Sistema

1. Realizar backup

- Como um Usuário Administrador
- Com o objetivo de criar uma cópia de segurança dos arquivos do estado atual do sistema
- Eu gostaria de fazer um backup dos dados do sistema

2. Atualizar sistema

- Como um Usuário Administrador
- Com o objetivo de implementar novas funcionalidades ou realizar manutenção no sistema
- Eu gostaria de atualizar sistema

3. Restaurar sistema

- Como um Usuário Administrador
- Com o objetivo de voltar a versão do sistema para uma versão previamente publicada
- Eu gostaria de restaurar o sistema

3.1.8 Cenários de Teste

3.1.8.1 Serviço de Autenticação

1. Solicitar cadastro no sistema

- DADO que eu sou Visitante E tenho a intenção de solicitar meu cadastro no sistema, QUANDO eu clicar em "Quero me Cadastrar" na página de Login, ENTÃO serei redirecionado para a página de Solicitação de Cadastro;
- DADO que eu preenchi os valores pedidos E eles são válidos, QUANDO eu clicar em "Solicitar Cadastro", ENTÃO uma mensagem de sucesso será exibida. CASO CONTRÁRIO, uma mensagem de erro será informada;

2. Realizar login no sistema

- DADO que eu sou um Usuário Cadastrado E tenho a intenção de me logar no sistema, QUANDO eu preencher meus dados de acesso na página de login E esses dados forem válidos, ENTÃO serei autenticado e uma mensagem de sucesso será exibida. CASO CONTRÁRIO, uma mensagem de erro será informada;

3. Recuperar senha esquecida

- DADO que eu sou um Usuário Cadastrado E não estou logado E tenho a intenção de redefinir minha senha, QUANDO eu clicar em "Recuperar Senha" na página de login, ENTÃO serei redirecionado para a página de recuperação de senha;
- DADO que estou na página de Recuperar Senha, QUANDO eu preencher os valores pedidos E eles forem válidos, ENTÃO uma mensagem de sucesso será exibida. CASO CONTRÁRIO, uma mensagem de erro será informada;
- DADO que eu sou um Usuário Cadastrado E estou logado E tenho a intenção de redefinir minha senha E estou na tela de "Trocar Senha", QUANDO eu preencher os valores pedidos E eles forem válidos, ENTÃO uma mensagem de sucesso será exibida. CASO CONTRÁRIO, uma mensagem de erro será informada;

4. Alterar dados cadastrais

- DADO que eu sou um Usuário Cadastrado E estou logado E estou na página de Atualizar DADOS, QUANDO eu preencher os valores pedidos E eles forem válidos, ENTÃO uma mensagem de sucesso será exibida. CASO CONTRÁRIO, uma mensagem de erro será informada;

3.1.8.2 Serviço de Navegação

1. Visualizar listagem de pesquisas solicitadas

- DADO que eu sou um Visitante E estou na página inicial, QUANDO eu abrir a página, ENTÃO será exibida uma tabela listando todas as pesquisas públicas
- DADO que eu sou um Pesquisador E estou logado E estou na página de Minhas Pesquisas, QUANDO eu abrir a página, ENTÃO será exibida uma tabela listando todas as pesquisas que já solicitei

2. Visualizar informações de anúncios coletados em uma pesquisa específica

- DADO que eu sou um Visitante (ou sou um Pesquisador E estou logado) E estou na página de Minhas Pesquisas, QUANDO eu clicar no botão de "Detalhes" em uma determinada linha da tabela, ENTÃO serei redirecionado para a página de detalhes da pesquisa correspondente
- DADO que eu sou um Visitante (ou sou um Pesquisador E estou logado) E estou na página de Detalhes da Pesquisa, QUANDO eu interagir com os filtros, ENTÃO os dados serão recarregados de acordo com os filtros aplicados. CASO CONTRÁRIO, uma mensagem de erro será informada;
- DADO que cliquei no botão de "Baixar dados filtrados", ENTÃO os dados coletados na pesquisa correspondente serão baixados com os filtros correntemente selecionados em um arquivo de valores separado por vírgula;

3.1.8.3 Serviço de Solicitação de Pesquisa

1. Solicitar nova pesquisa

- DADO que sou um Pesquisador E estou na página de Minhas Pesquisas, QUANDO eu clicar no botão de "Iniciar nova pesquisa", ENTÃO serei redirecionado para a página de Nova Pesquisa;
- DADO que sou um Pesquisador E estou na página de Nova Pesquisa e preenchi os valores pedidos E eles são válidos, ENTÃO uma nova pesquisa será iniciada. CASO CONTRÁRIO, uma mensagem de erro será informada;

2. Continuar pesquisa

- DADO que sou um Pesquisador E estou na página de Minhas Pesquisas, QUANDO eu clicar no botão de "Tentar novamente" em uma determinada linha da tabela, SE essa pesquisa foi interrompida devido a um status de erro, ENTÃO a pesquisa correspondente será continuada a partir do próximo status possível. CASO CONTRÁRIO, uma mensagem de erro será informada;

- DADO que sou um Pesquisador E estou na página de Minhas Pesquisas, QUANDO eu clicar no botão de "Tentar novamente" em uma determinada linha da tabela, SE essa pesquisa está concluída, ENTÃO a pesquisa correspondente será reiniciada. CASO CONTRÁRIO, uma mensagem de erro será informada;

3. Parar pesquisa

- DADO que sou um Pesquisador E estou na página de Minhas Pesquisas, QUANDO eu clicar no botão de "Interromper pesquisa" em uma determinada linha da tabela, ENTÃO a pesquisa correspondente será pausada até que se deseje continuá-la. CASO CONTRÁRIO, uma mensagem de erro será informada;

3.1.8.4 Serviço de Gestão de Usuários

1. Aceitar solicitação de cadastro

- DADO que sou um Administrador E estou na página de Minhas Pesquisas, QUANDO eu clicar no botão de "Permitir acesso" em uma determinada linha da tabela, ENTÃO o usuário correspondente passará a ter acesso às páginas privadas do sistema pertinentes ao seu nível de permissão. CASO CONTRÁRIO, uma mensagem de erro será informada;

2. Remover usuário

- DADO que sou um Administrador E estou na página de Minhas Pesquisas, QUANDO eu clicar no botão de "Remover usuário" em uma determinada linha da tabela, ENTÃO o usuário correspondente será excluído do sistema. CASO CONTRÁRIO, uma mensagem de erro será informada;

3. Alterar nível de permissão do usuário

- DADO que sou um Administrador E estou na página de Lista de Usuários, QUANDO eu clicar no botão de "Alterar permissão" em uma determinada linha da tabela, ENTÃO a permissão do usuário correspondente será trocada de Pesquisador para Administrador caso a permissão atual seja de Pesquisador ou de Administrador para Visitante caso a permissão atual seja de Administrador. CASO CONTRÁRIO, uma mensagem de erro será informada;

3.1.8.5 Serviço de Gestão do Sistema

1. Realizar backup

- DADO que sou um Administrador E tenho acesso ao código-fonte do sistema e as credenciais de acesso ao servidor, CONSEGUIREI realizar backup do sistema

2. Atualizar sistema

- DADO que sou um Administrador E tenho acesso ao novo código-fonte do sistema e as credenciais de acesso ao servidor QUANDO eu subir essa nova versão do código-fonte, CONSEGUIREI atualizar a versão do sistema

3. Restaurar sistema

- DADO que sou um Administrador E tenho acesso a uma versão anterior do código-fonte do sistema e as credenciais de acesso ao servidor QUANDO eu subir essa versão do código-fonte, CONSEGUIREI retornar a versão do sistema para uma versão mais antiga

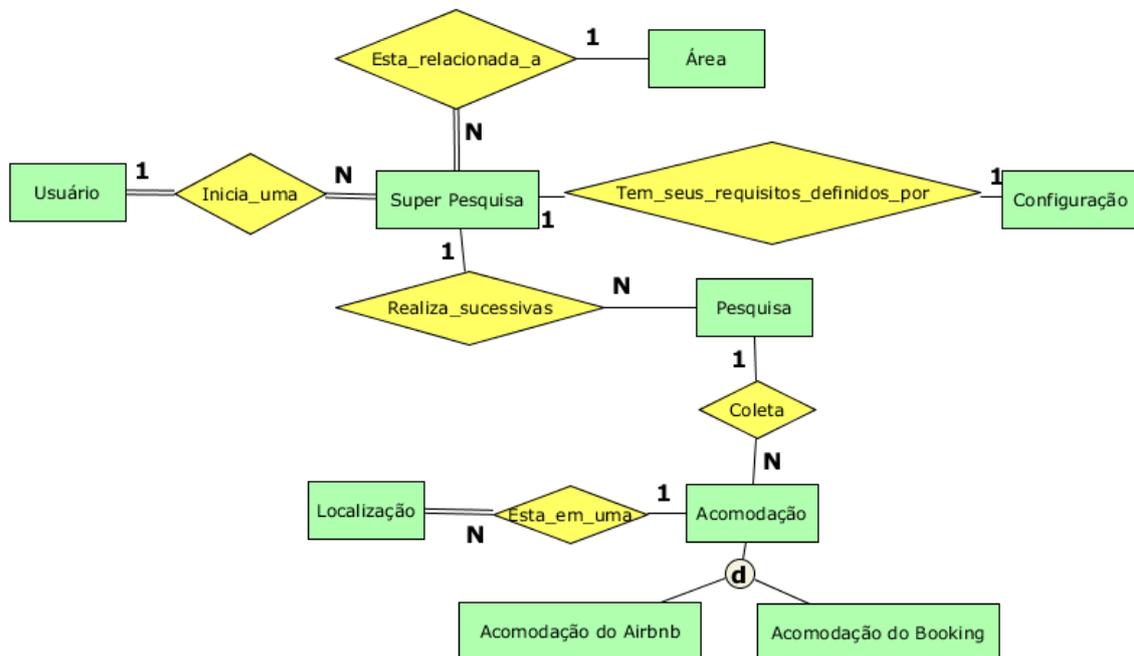
3.2 Modelagem do Banco de Dados

Neste momento do trabalho, tem-se a seguinte situação a modelar: uma pessoa X, pesquisadora que atua no estudo da ação de determinada plataforma em uma cidade específica, buscando entender melhor como Airbnb e/ou Booking atuam em sua área de pesquisa, vê a necessidade de encontrar os dados desses anúncios de forma estruturada e facilitada a partir de um determinado site. Essa pessoa, então, deve informar os parâmetros da pesquisa ao site para que ocorra a execução da busca através dos anúncios da plataforma a partir dos critérios especificados. Os anúncios coletados apresentam dados de acomodações que se diferem entre si devido a forma de apresentação da plataforma de origem (Airbnb/Booking), ainda que possuam várias características comuns entre si, como nome, preço e classificação média. Tais acomodações representam estabelecimentos que existem no mundo físico real, ou seja, possuem um endereço associado a eles, e é de grande relevância para esta pessoa que esta localização também seja informada pelo site.

Assim sendo, na Figura 3.3, é representado um diagrama relacional da estruturação do banco de dados do sistema, onde podem ser visualizadas as relações utilizadas para armazenar os dados coletados e as formas como elas se relacionam entre si e as outras entidades formuladas. Sobre tal diagrama, cabe dizer que uma Acomodação do Airbnb se trata de um imóvel disponibilizado no site, enquanto, de forma similar, uma Acomodação do Booking se trata de um imóvel encontrado no site de mesmo nome. Ambos os tipos de Acomodações são extraídos a partir de uma Pesquisa que pode ou não estar atrelada a uma série de pesquisas sucessivas que caracteriza uma Super Pesquisa. Cada Pesquisa ocorre dentro de uma Área de Pesquisa e cada Área de Pesquisa corresponde a uma Localização. Cada Acomodação identificada possui uma localização geoespacial que pode ser identificada a partir de suas coordenadas.

A Figura 3.4 representa a conversão do Diagrama ER em um modelo relacional do minimundo descrito. Os atributos (representados na Figura com o nome de variável referido no código implementado) descritos para cada relação são descritos no Apêndice A.

Figura 3.3 – Diagrama ER para o DashAcomodações



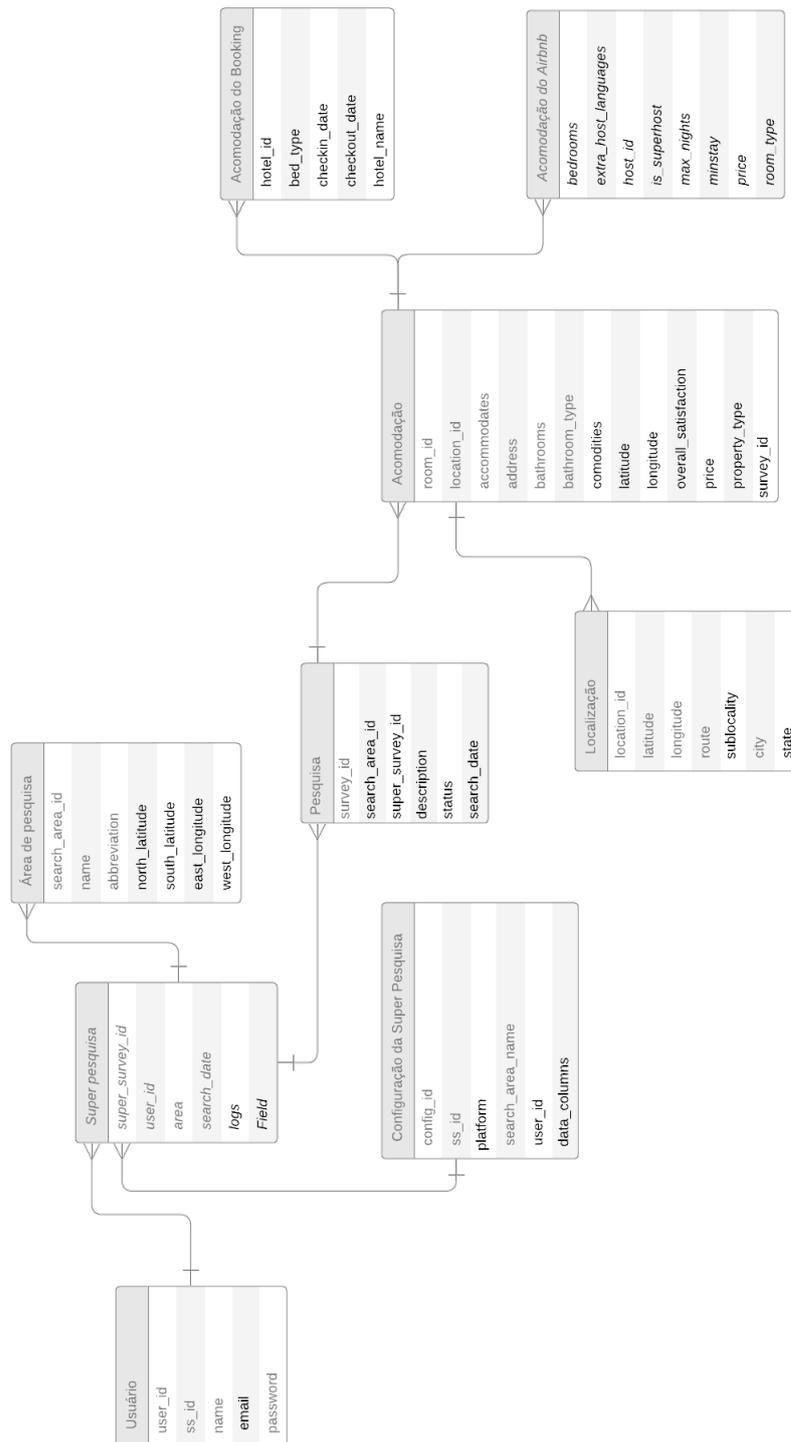
3.3 Extração dos dados

O início do processo de extração dos dados ocorre através da utilização da biblioteca geopy, que é responsável por gerar a Caixa Delimitadora da cidade informada. Esta Caixa Delimitadora consiste nas coordenadas de latitude norte e sul, e longitude leste e oeste, que circunscrevem o território físico da cidade. Uma vez que a Caixa Delimitadora foi obtida, um novo processo de pesquisa é criado, tendo como objetivo principal a coleta de dados dentro dessa área específica. A partir daí, tem-se dois casos: a pesquisa pode prosseguir para coletas acomodações do Airbnb e/ou acomodações do Booking (caso ambos os tipos de acomodação devam ser coletados, primeiro se coleta as acomodações do Airbnb e, em seguida, as acomodações do Booking). Cada um dos casos será melhor descrito nas duas sessões seguintes.

Cabe dizer também que o início do processo descrito nos parágrafos anteriores e a continuação do mesmo para o caso de ambas as plataformas se repete no caso das buscas por áreas menores dentro de uma área maior (por exemplo, na busca por bairros dentro de uma cidade ou na busca por ruas dentro de um bairro), com a única diferença sendo a definição da área a ser observada. As ruas e bairros de determinada cidade são identificadas via geopy a partir das coordenadas de latitude e longitude de acomodações localizadas na primeira etapa do processo de Pesquisa.

Também é importante dizer que, a cada etapa percorrida, o status da pesquisa é alterado no banco de dados, mas a ocorrência de um estado de erro não interrompe o fluxo da pesquisa (neste caso, a observação é adicionada aos logs e a pesquisa é movida para a próxima etapa possível).

Figura 3.4 – Diagrama Relacional para o DashAcomodações



A Tabela 3.3 exibe alguns dos status de pesquisa mapeados, com seus códigos identificadores, mensagens associadas e o que indicam.

Como pode ser observado pelos exemplos presentes na tabela, os códigos negativos (menores que 0) representam mensagens de erro e os códigos positivos representam mensagens de sucesso relacionados à etapa associada.

Tabela 3.3 – Exemplos de status possíveis para a Super Pesquisa

Código do Status	Mensagem	Significado
-1	Falha ao configurar pesquisa	Indica que houve um erro ao cadastrar a Super Pesquisa ou a Configuração associada a ela no banco de dados
1	Configuração concluída	Indica que a Configuração de Pesquisa foi inserida com sucesso no banco de dados
3	Iniciando busca por nome da Área de Pesquisa	Indica que a próxima etapa a ser executada será a busca pela Área de Pesquisa em si
-42	Falha ao buscar por bairro nome do bairro	Indica que houve algum erro ao realizar a coleta das acomodações do bairro apresentado
200	Pesquisa por nome da Área de pesquisa finalizada	Indica que todas as etapas da pesquisa já foram perpassadas e sua execução foi finalizada

3.3.1 Extração dos dados dos anúncios do Airbnb

A extração de dados do Airbnb foi baseado no Airbnb Data Collection de Slee. Com o objetivo de obter acesso a todos os anúncios de estadias disponibilizados para uma dada busca, foram utilizadas estratégias para "driblar" a API do Airbnb, que, como se teorizou, apresenta um número limite de anúncios independentemente da cidade consultada ou do real valor total. Dentre as estratégias adotadas, destacam-se: realização de múltiplas requisições, definindo a cidade alvo e data de estadia, em intervalos de tempos pré-definidos; utilização de vários *proxies* para disparar as requisições de interesse; realização de requisições utilizando coordenadas da Caixa Delimitadora e, por fim, a combinação de todos os resultados obtidos a partir das requisições realizadas.

Posterior à busca recursiva, com o intuito de coletar dados que não são fornecidos pela API do Airbnb, com o apoio da biblioteca Selenium, se instancia um *webdriver* que acessa a página de cada uma das acomodações encontradas durante a primeira etapa e, para cada uma dessas páginas, busca elementos como as comodidades ou o tipo de banheiro ofertado. Os registros das acomodações que foram encontrados na etapa anterior são atualizados com estes novos valores identificados.

uma localização específica delimitada por uma caixa delimitadora e cada acomodação do Airbnb pode possuir várias avaliações postadas na plataforma.

3.3.2 Extração dos dados dos anúncios do Booking

Para realizar a extração dos dados dos anúncios do Booking, baseado em Alcalá, com o auxílio da biblioteca *selenium*, se instancia um *webdriver* com uma url no formato

"https://www.booking.com/searchresults.pt-br.html?ss={nome da cidade}&ssne={nome da cidade}&ssne_untouched={nome da cidade}&checkin={data de checkin}&checkout={data de checkout}", onde o nome da cidade é informado sem o acréscimo do nome do estado informado na área de pesquisa e as datas de checkin e checkout informam o que seus nomes indicam. Tais datas correspondem, por padrão, a 15º e 16º imediatamente após a data de iniciação da Pesquisa.

A partir da primeira página de resultados da pesquisa por acomodações, encontra os links correspondentes às demais páginas da pesquisa para acessá-las em etapa posterior.

Para cada página de resultados, se acessa página por página das acomodações coletando as informações de interesse via XPath, id do elemento ou seletor CSS. Por exemplo, a Figura 3.5 exibe a implementação de uma função para encontrar o tipo de propriedade da acomodação. O "driver" usado como parâmetro da função se refere ao webdriver correspondente à página da acomodação como um todo, onde deve ser localizado o elemento cujo data-testid corresponde a "property-type-badge".

Figura 3.5 – Função implementada para encontrar o texto correspondente ao Tipo de Propriedade de uma Acomodação do Booking

```
def find_property_type(self, driver):
    try:
        element = driver.find_element(
            By.XPATH, '//*[@data-testid="property-type-badge"]')
        self.property_type = element.text
    except selenium.common.exceptions.NoSuchElementException:
        raise
```

Vale dizer que, para cada página encontrada para um tipo de propriedade mais de uma tupla é identificada e inserida no banco de dados. Diferente do Airbnb onde cada página correspondente a uma Acomodação, cada Acomodação do Booking se refere a um dos quartos disponíveis dentro de um estabelecimento maior. Tal lógica se deve ao fato de ser mais comum que hotéis e pousadas, por exemplo, sejam alugados por conjuntos distintos de turistas não relacionados entre si, de forma que 2 quartos diferentes poderiam ser alugados por indivíduos completamente desconhecidos (em outras palavras, 2 grupos de turistas). O mesmo não acontece no Airbnb, onde cada acomodação (uma casa, por exemplo), costuma agrupar um único grupo. De forma análoga, se existe mais de um quarto disponível, mais de um grupo de turistas poderia ser alocado. Assim sendo, cada quarto é visto neste sistema como uma Acomodação que pode ser destinada a um grupo de turistas.

A Figura 3.6 exemplifica um caso de inserção. Nesse exemplo, para um determinado hotel que disponibiliza o "Quarto Triplo com Banheiro Privativo", seriam identificadas no mínimo 3 tuplas. Assim, as tuplas de identificadores 7946809021 e 7946809022 representam cada um dos quartos de 600 reais, enquanto a tupla 7946809011 representa 1 quarto de 570 reais (a quantidade deste segundo quarto foi oculta do exemplo, por isso foi escrito "no mínimo" 3 tuplas). O último

dígito nos códigos identificadores (1, 1 e 2, respectivamente) representam o índice de inserção do quarto.

Figura 3.6 – Exemplo de Identificação de Acomodações do Booking

Categoria de quarto	Acomoda:	Preço de hoje	Suas escolhas	Selecionar nº de quartos
<p>Quarto Triplo com Banheiro Privativo</p> <p>Restam só 2 quartos no nosso site</p> <p>1 cama de solteiro e 1 cama de casal</p> <p>Ar-condicionado Banheiro no quarto TV de tela plana Frigobar</p> <p>WiFi Gratuito</p> <p>Produtos de higiene pessoal gratuitos Chuveiro Vaso sanitário Toalhas Roupa de cama Tomada perto da cama TV Micro-ondas Ventilador Roupa de cama e banho (custo adicional) Serviço de despertar/despertador Canais a cabo Serviço de despertar Andares superiores acessíveis somente por escada Papel higiênico A unidade é totalmente acessível para cadeira de rodas Álcool gel</p>	<p>1</p>	<p>R\$ 570</p> <p>Impostos e taxas incluídos</p>	<p>Café da manhã excepcional incluído</p> <p>Não reembolsável</p> <p>Genius - desconto disponível</p>	<p>0</p>
	<p>2</p>	<p>R\$ 600</p> <p>Impostos e taxas incluídos</p>	<p>Café da manhã excepcional incluído</p> <p>Cancelamento grátis antes de 25 de agosto de 2023</p> <p>Genius - desconto disponível</p>	<p>0</p> <p>1 (R\$ 600)</p> <p>2 (R\$ 1.200)</p>



	room_id bigint	room_name character varying (255)	price double precision
1	7946809011	Quarto Triplo com Banheiro Privativo	570
2	7946809021	Quarto Triplo com Banheiro Privativo	600
3	7946809022	Quarto Triplo com Banheiro Privativo	600

Tendo sido coletadas essas informações e as inserido no banco de dados, a próxima página de resultados é acessada e assim por diante até o fim do processo.

3.4 Agrupamento

Posterior à extração dos dados, ocorreu a etapa de agrupamento destes. A preparação de dados é um passo fundamental antes de os aplicar algoritmos de agrupamento previamente citados, K-modes, DBSCAN e BIRCH. Esses algoritmos geralmente esperam dados numéricos como entrada, o que pode ser um desafio quando temos informações de diferentes tipos, como dados textuais e numéricos misturados. A preparação cuidadosa dos dados é fundamental para garantir resultados confiáveis e úteis na análise de agrupamento e ajuda a manter a integridade dos dados e evitar problemas durante o processo de agrupamento.

Inicialmente, a partir de uma cópia dos dados coletados, foi necessário lidar com valores nulos ou faltantes nos dados antes de aplicar o agrupamento. A abordagem aplicada foi preencher

esses valores com os valores médios da coluna correspondente para casos numéricos. Os valores textuais nulos foram preenchidos com a *string* "NULL" para indicar que estes valores faltantes.

Tendo feito isso, buscou-se converter os dados mistos em dados numéricos adequados para o agrupamento. Com isso, pode-se aproveitar as funcionalidades desses algoritmos e obter agrupamentos significativos a partir dos dados.

Para realizar essa conversão, é necessário realizar algumas etapas de preparação. Primeiro, é importante identificar todas as colunas que contêm informações textuais. Em seguida, é preciso substituir esses valores por números, atribuindo um valor numérico único para cada categoria. Por exemplo, para dados de bairros que incluam os valores "Barra", "Bauxita", "Centro", podem ser atribuídos os valores 1, 2 e 3 a cada uma delas, respectivamente. Tal etapa foi realizada com a classe *LabelEncoder* do *scikit-learn*.

Posterior ao *LabelEncoder*, foi usado o preprocessor *StandardScaler* para normalizar e padronizar os dados que, nesse momento, são todos numéricos dentro do escopo do agrupamento.

A partir dessa preparação, os algoritmos de agrupamento podem ser aplicados. Destaca-se, aqui, que o agrupamento não é um processo obrigatório dentro do sistema, mas sim uma etapa escolhida (ou não) pelo usuário. No caso de ocorrer a clusterização, um novo atributo é atribuído a cada Acomodação a ser retornada nos dados originais: um valor inteiro representando o seu *cluster*. Baseando-se na implementação destes algoritmos pela biblioteca *scikit-learn* (ou na biblioteca específica *kmodes*, no caso do método de mesmo nome), a Tabela 3.4 apresenta os parâmetros passíveis de personalização pelos usuários do sistema, assim como os valores padrão atribuídos a cada um deles, caso não haja sobreposição explícita.

Tabela 3.4 – Parâmetros para agrupamento

Nome	Algoritmo(s)	Significado	Valor padrão
<code>branching_factor</code>	BIRCH	Quantidade máxima de elementos em um grupo	50
<code>threshold</code>	BIRCH	Distância limite entre os grupos gerados	0.5
<code>n_clusters</code>	BIRCH e K-Modes	Quantidade de grupos a serem formados	3
<code>eps</code>	DBScan	Quantidade de épocas	3
<code>min_samples</code>	DBScan	Quantidade mínima de amostras	2
<code>init</code>	K-Modes	Método de inicialização	Huang
<code>n_init</code>	K-Modes	Quantidade de tentativas	3

3.5 Implementação do back-end e front-end

Nesta sessão, são descritas as organizações dos repositórios do back-end e do front-end do DashAcomodações. Tal organização foi pensada tendo como critérios a facilidade de

desenvolvimento e manutenção do projeto. Ao modularizar o código de tal forma, é possível ter uma visão mais clara das diferentes partes do sistema. Além disso, essa divisão permite que as funcionalidades sejam implementadas de forma mais modular, permitindo reutilização de código e facilitando possíveis atualizações e melhorias.

3.5.1 Back-end

O repositório back-end foi organizado de forma a facilitar a gestão e o desenvolvimento do sistema, sendo dividido em diferentes pastas:

A pasta "config" contém os arquivos de descrição e configuração do banco de dados e outras APIs utilizadas pelo sistema. Nessa pasta, é possível encontrar arquivos com informações sobre a conexão com o banco de dados, autenticação de APIs externas, entre outros. A Figura 3.7 exibe algumas das configurações necessárias para se conectar o back-end com o banco de dados PostgreSQL, por exemplo.

Figura 3.7 – Arquivo de configuração de credenciais do banco de dados

```
[DATABASE]
# -----
# The database is a PostgreSQL database, and these are the standard
# connection parameters.
# -----
# The host is the machine name for the connection
# -----

db_host =

# -----
# The port number (PostgreSQL default port is 5432)
# -----

db_port =

# -----
# A PostgreSQL database server may host multiple databases.
# Specify the database you are using here.
# -----

db_name =

# -----
# Database user, which must have sufficient permissions for
# whatever operations you are carrying out
# -----

db_user =

# -----
```

A pasta "controllers" está organizada de acordo com os serviços descritos em 3.1.6. Isso permite que as funcionalidades sejam implementadas de forma mais estruturada. Nesse caso, há controllers específicas para os serviços de Autenticação, Navegação, Pesquisa e Usuário. Cada uma dessas controllers implementa os métodos correspondentes às funcionalidades relacionadas a esses serviços. Por exemplo, na controller de Autenticação, relacionada ao Serviço de mesmo nome, estão os métodos de login e cadastro.

Outro exemplo é a presença de uma função para alterar o nível de permissão de um Visitante ou Administrador dentro da controller de Usuário, como está especificado no Serviço de Gestão do Usuário. A Figura 3.8 exibe a implementação de tal função.

Figura 3.8 – Método implementado para alterar permissão do usuário

```
def change_permission(data: ChangePermissionModel):
    try:
        user_id = update_command(ab_config,
                                sql_script="""UPDATE users set permission = %s
                                             where user_id = %s returning user_id""",
                                params=(
                                    (data.permission, data.user_id)),
                                initial_message="Atualizando permissão do usuario...",
                                failure_message="Falha ao atualizar permissão do usuário")

        if (user_id):
            response = {
                "object": user_id,
                "message": "Dados retornados com sucesso!",
                "success": True
            }
            return response
    except:
        return {"message": "Falha ao alterar permissão do usuário", "success": False}
```

As controllers são diretamente acessadas pelas rotas da aplicação FastAPI, como representa a Figura 3.9.

Figura 3.9 – Acesso da aplicação a método implementado na controller de Autenticação

```
@app.post('/users/change_permission')
async def change_permission(data: ChangePermissionModel):
    return users.change_permission(data)
```

A tipagem observada na Figura 3.9 é definida na pasta "models", onde estão localizadas as tipagens dos tipos de entrada possíveis para as requisições realizadas pelo front-end. A figura 3.10 exibe o tipo de dados necessários e aceitos ao solicitar ao back-end a alteração do nível de permissão de um usuário.

Figura 3.10 – Tipagem para dados da requisição do método para alterar nível de permissão do usuário

```
from pydantic import BaseModel

...
class ChangePermissionModel(BaseModel):
    permission: str
    user_id: str
```

A pasta "utils" define métodos com diferentes funcionalidades que podem ser úteis em diferentes momentos da implementação do back-end. Essas funções são compartilhadas entre as diferentes partes do sistema e, portanto, é interessante que se promova o reuso delas. Alguns exemplos de funções úteis presentes nessa pasta são a clusterização de dados e a exportação de dados para formatos específicos.

As funcionalidades descritas sobre a extração de dados estão definidas no diretório `scrap`. Os métodos para a inicialização de uma pesquisa são executadas no back-end via Background Tasks, como exibe a Figura 3.11. Isso permite que as pesquisas sejam executadas como uma ação de fundo do sistema, sem bloquear demais requisições.

Figura 3.11 – Tipagem para dados da requisição do método para alterar nível de permissão do usuário

```
def start(data: StartModel, background_tasks: BackgroundTasks):
    try:
        ss_id = search.initialize_search(config=ab_config,
                                       platform=data.platform,
                                       search_area_name=data.city,
                                       user_id=data.user_id,
                                       columns=data.columns,
                                       clusterization_method=data.clusterization_method,
                                       aggregation_method=data.aggregation_method,
                                       start_date=data.start_date,
                                       finish_date=data.finish_date,
                                       include_locality_search=data.include_locality_search,
                                       include_route_search=data.include_route_search)

        background_tasks.add_task(search.full_process,
                                  platform=data.platform,
                                  search_area_name=data.city,
                                  user_id=data.user_id,
                                  columns=data.columns,
                                  start_date=data.start_date,
                                  finish_date=data.finish_date,
                                  include_locality_search=(data.include_locality_search == 'true'),
                                  include_route_search=(data.include_route_search == 'true'),
                                  super_survey_id=ss_id)

        response = {
            "object": {"super_survey_id": ss_id},
            "message": "Pesquisa cadastrada com sucesso!",
            "success": True
        }

        return response
    except:
        return {"message": "Falha ao iniciar pesquisa", "success": False}
```

3.5.2 Front-end

No front-end do projeto, o diretório de pages contém os componentes responsáveis por renderizar as páginas completas do site (para fins de exemplificação, a Figura 3.12 exibe a Tela Inicial do DashAcomodações).

Geralmente, um arquivo dentro deste diretório consiste em um arquivo que importa um componente previamente implementado na pasta `componentes`, composta por elementos gráficos menores. Dentro desta pasta, existem subpastas chamadas de `"ui"` e `"local"`.

A subpasta `"ui"` contém elementos básicos de interface de usuário (daí vem o seu nome) que podem ser facilmente reutilizados em diferentes módulos sem depender de outros componentes. Por exemplo, botões e inputs (como é o caso do input de Área de Texto na Figura 3.13) são exemplos de elementos presentes em `"ui"`.

Figura 3.12 – Tela inicial do DashAcomodações

DashAcomodações
Login ou Cadastro

DashAcomodações

Partindo de uma abordagem multidisciplinar que envolve Turismo e Ciência da Computação, este trabalho promove transparência sobre a atuação das plataformas envolvidas ao propor a implementação de um web scraping e técnicas de ciência de dados para extrair e analisar de forma automatizadas os dados de interesse tanto do Airbnb quanto do Booking.



A importância do projeto



Considerando a complexidade das contradições envolvendo a presença do Airbnb em determinadas comunidades (em muitas delas, onde já havia a presença e atuação de empresas hoteleiras tradicionais), é imprescindível que os órgãos responsáveis pela regulação do mercado imobiliário realizem uma análise quantitativa sobre a dimensão e influência desse serviço nessas localidades específicas. Entretanto, a aquisição e manipulação de dados referentes a esse fenômeno pode ser extremamente desafiador e requer a aplicação de esforços adicionais e efetivos por parte das entidades gerenciadoras.

Ao navegar por esse site, você pode ver ou solicitar dados sobre acomodações ofertadas por ambas as plataformas em determinado espaço de tempo. Abaixo, por exemplo, são listadas as cidades cujos dados já foram coletados previamente. Dê uma olhada!

Cidade	Quantidade de Pesquisas	Última atualização em	
Airuoca, Minas Gerais	9	22/Jul/2023	i
Ouro Preto, Minas Gerais	54	03/Jul/2023	i
Aiuroca, MG	3	27/Jun/2023	i
Oliveira, Minas Gerais	4	27/Jun/2023	i
Airbnb	1	02/Jun/2023	i
Mariana, Minas Gerais	16	02/Jun/2023	i
Tiradentes, Minas Gerais	1	02/Jun/2023	i
Airuoca, Minas Gerais	12	26/May/2023	i



Por Victor Martins

Já a subpasta "local" é composta por componentes criados especificamente para a construção deste sistema e não são facilmente reutilizáveis em outros projetos. Alguns exemplos de elementos presentes em "local" são cabeçalhos e rodapés, compostos a partir de elementos "ui".

Na subpasta "structures" estão presentes os componentes que fazem referência a estruturas completas de telas, comumente com integração com a API do back-end. Esses componentes são mais complexos e são formados a partir de componentes locais ou de UI.

Como pode-se notar nesta figura, arquivos de estilização são importados para ornamentação dos componentes definidos. Tais arquivos de estilização são definidos na própria pasta do

Figura 3.13 – Implementação do componente de input por Área de Texto

```
import React from 'react';
import ClassJoin from '~/utils/ClassJoin/ClassJoin';
import { TextAreaProps } from './textArea.interface';
import style from './textArea.module.scss';

const TextArea = React.forwardRef<HTMLTextAreaElement, TextAreaProps>(({ error, type, ...rest }, ref) => {
  return (
    <textarea
      className={ClassJoin([style.textArea, error && style.error])}
      ref={ref}
      style={{ resize: 'vertical' }}
      rows={5}
      {...rest}
    />
  );
});

export default TextArea;
```

componente (assim como sua interface), enquanto estilizações que afetam todo o projeto, estão definidas na pasta `assets`, que contém duas subpastas, "imagens" e "arquivos de estilo globais". A pasta "arquivos de estilo globais" possui arquivos que definem estilos globais, como paletas de cores, padrões de tipografia e resolução de tela para tornar o design responsivo. Já a pasta "imagens" é destinada ao armazenamento de todas as imagens utilizadas no sistema.

No sistema geral, a estilização segue uma abordagem de CSS modular, onde os estilos são escritos de forma isolada para cada componente ou módulo do aplicativo.

Outro elemento que se pode notar é a importação da interface `TextAreaProps`, que define a tipagem para o componente `TextArea`. Como o front-end foi desenvolvido em TypeScript, a pasta `types` (onde tal tipagem está localizada) foi criada com o intuito de listar os tipos de dados criados para a implementação do código.

A pasta `config` armazena informações importantes para o funcionamento do sistema, como links de acesso à API do back-end, tanto para o ambiente de desenvolvimento quanto para produção. Também é possível encontrar o nome dos métodos disponibilizados e mensagens padrão para respostas de requisição que resultarem em erro.

Tanto back-end quanto front-end armazenam configurações próprias que interferem na execução do sistema e de suas pesquisas como um todo. O que diferencia ambos os repositórios de configurações é que, no back-end, então estabelecidas os ajustes mais técnicos da programação, como as já citadas credenciais de banco de dados e servidor e chaves de API. Já no front-end, as configurações estão mais focadas em rotas acessadas e mensagens de resposta padrão.

O diretório `routes` se refere aos diferentes tipos de acesso que existem no DashAcomodações, sendo dividido entre páginas públicas (que podem ser acessadas por qualquer usuário, independentemente de autenticação), páginas privadas (que só podem ser acessadas por usuários autenticados) e páginas de login (que permitem a autenticação quando o usuário ainda não está logado).

O diretório `services` contém funções que tratam o envio e recebimento de dados do

back-end. Essas funções são responsáveis por realizar a comunicação com a API e processar as informações.

A pasta de `utils` é destinado a funções que podem ser úteis em diferentes momentos da implementação do front-end. São exemplos de funções úteis aquelas que realizam conversões de um tipo de valor para outro ou aplicam máscaras sobre esses valores.

informações de interesse que pudessem ser extraídas a partir deles.

4 Estudo de Caso

Nesta seção, é apresentado um estudo de caso para o uso do DashAcomodações em um cenário que engloba grande parte de suas funcionalidades.

Neste cenário, um (futuro) Usuário Pesquisador deseja se cadastrar no sistema. Para isso, depende da aprovação de um Usuário Administrador. Tendo se cadastrado, o Usuário deseja pesquisar a presença de acomodações presentes na cidade Ouro Preto e, posterior à coleta, agrupar os dados para identificar possíveis tipos de anúncios do Airbnb distintos dentro da cidade.

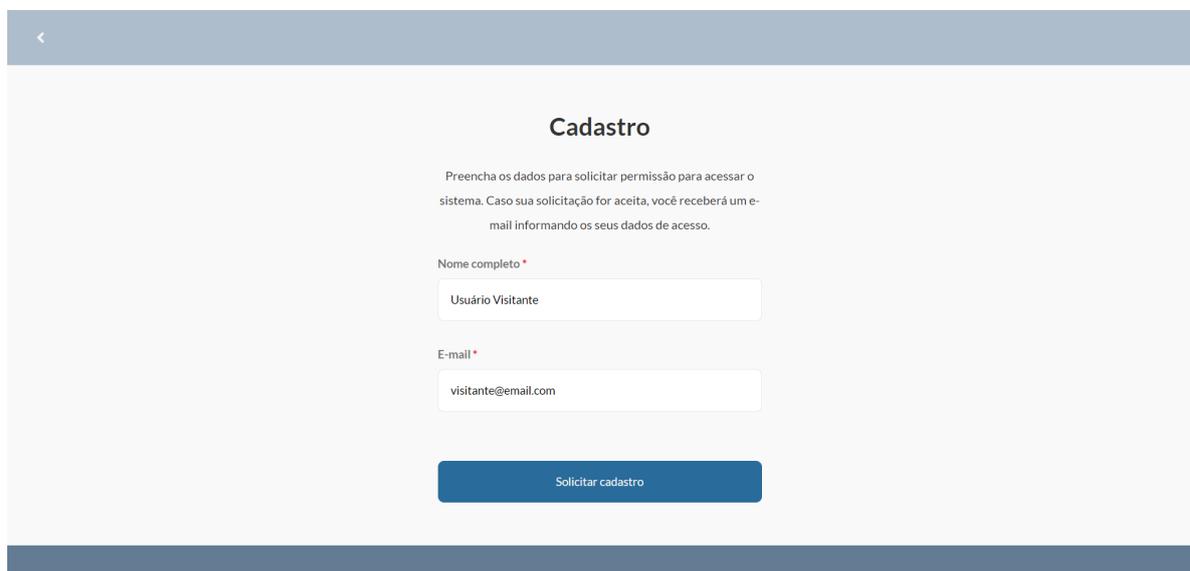
Para realizar esta tarefa, o Usuário deve acessar a Tela Inicial (representada na Figura 3.12) e clicar no botão localizado no cabeçalho da página para realizar seu cadastro. Dessa forma, ele é redirecionado para a página de Login, onde pode selecionar a opção para, ao invés de se logar, se cadastrar. O elemento que ele deve clicar está ilustrado na Figura 4.1.

Figura 4.1 – Link para acesso da página de cadastro



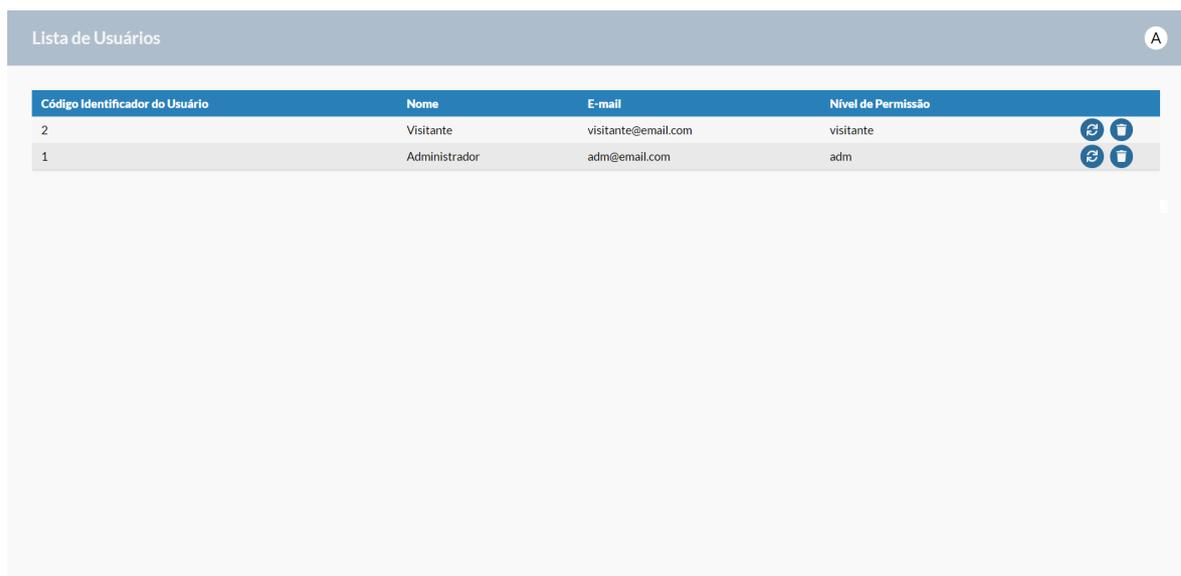
Acessando a Tela de Cadastro, o usuário deve informar as suas informações de acesso para realizar tal operação, como representa a Figura 4.2.

Figura 4.2 – Tela para cadastro

A imagem mostra a tela de cadastro de um sistema. No topo, há uma barra cinza com um ícone de seta para trás. O título "Cadastro" está centralizado. Abaixo dele, há um texto explicativo: "Preencha os dados para solicitar permissão para acessar o sistema. Caso sua solicitação for aceita, você receberá um e-mail informando os seus dados de acesso." Seguem dois campos de entrada: "Nome completo *" com o texto "Usuário Visitante" e "E-mail *" com o texto "visitante@email.com". No final, há um botão azul "Solicitar cadastro".

Tendo solicitado o cadastro, um Administrador, ao acessar a tela de "Lista de Usuários" (Figura 4.3), consegue gerenciar os usuários do sistema, incluindo a aceitação ou remoção definitiva de cadastros. Assim, o administrador pode aceitar a solicitação do usuário de forma permitir seu cadastro. Após aceitar a solicitar e enviar, via e-mail, a senha automática gerada para usuário, o usuário Pesquisador pode se logar. O Pesquisador pode alterar sua senha posteriormente.

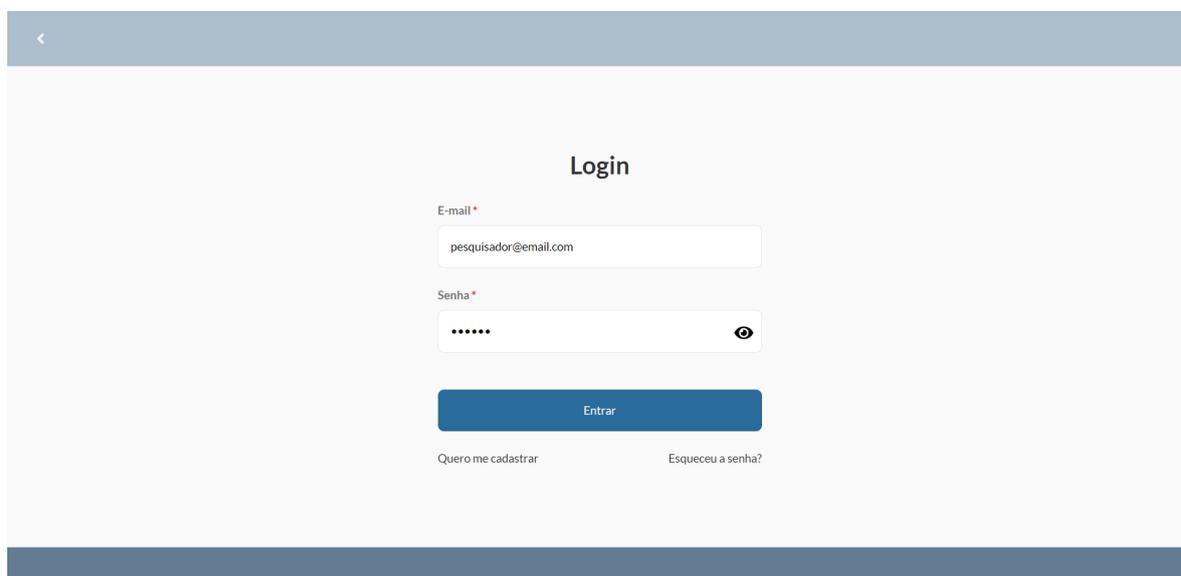
Figura 4.3 – Tela exibindo listagem com usuários do DashAcomodações



Código Identificador do Usuário	Nome	E-mail	Nível de Permissão
2	Visitante	visitante@email.com	visitante
1	Administrador	adm@email.com	adm

A partir disso, voltando ao Usuário Pesquisador, uma possibilidade para continuar seu percurso é acessar a Tela Inicial, representada na Figura 3.12. A partir dela, o usuário consegue se logar no sistema ao clicar no botão localizado no cabeçalho da página e ser redirecionado para a tela de Login e preencher as suas credenciais, como representa a Figura 4.4.

Figura 4.4 – Tela de Login



Login

E-mail *
pesquisador@email.com

Senha *
.....

Entrar

[Quero me cadastrar](#) [Esqueceu a senha?](#)

Após ter se logado, novamente através das opções dispostas no cabeçalho da página inicial (dessa vez representados pela Figura 4.5, o usuário consegue acessar a tela de "Novas Pesquisas", onde consegue preencher os dados do formulário de acordo com os critérios que são de interesse (nesse caso, anúncios do Airbnb dentro da cidade de Ouro Preto).

Figura 4.5 – Menu exibido após autenticação do usuário logado

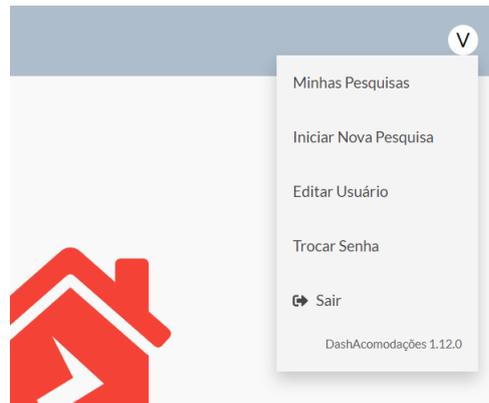
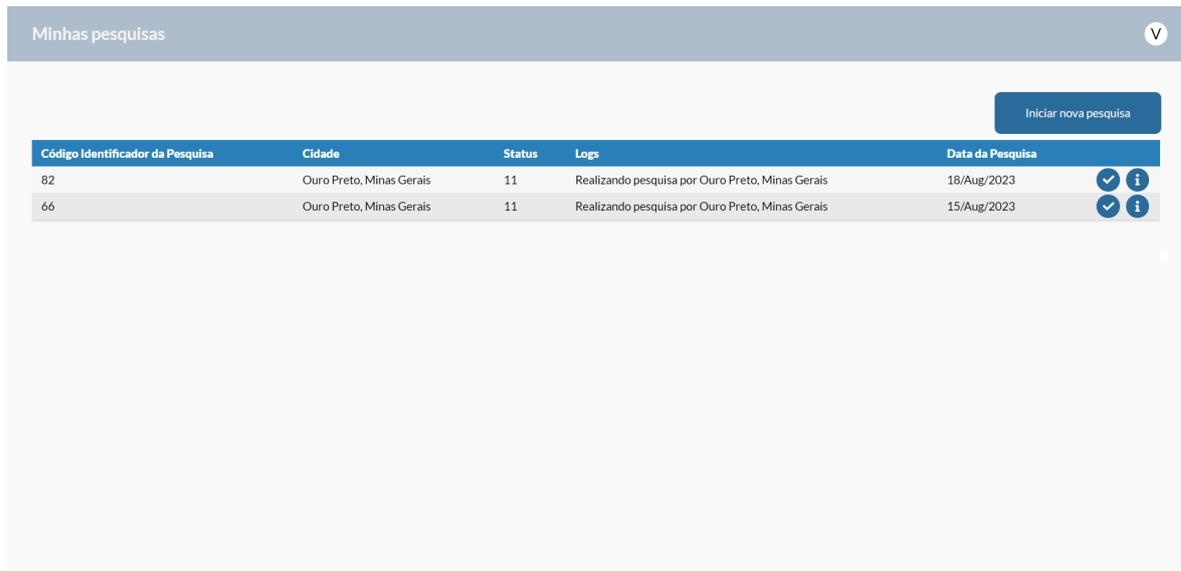


Figura 4.6 – Tela exibindo formulário para inserção dos critérios de configuração de uma nova pesquisa

Tendo sido retornada uma mensagem de sucesso, o usuário deve acessar a página de Minhas Pesquisas, também pelo menu representado na Figura 4.5. Nesta página (Figura 4.7), o usuário tem acesso a uma lista com todas as pesquisas que já solicitadas por ele. Aqui, ele tem controle sobre o fluxo da pesquisa, podendo interagir com os botões para reiniciar uma busca que tenha sido interrompida por erro ou finalizar uma pesquisa em andamento.

Outra opção é clicar em um dos botões da tabela e ser redirecionado para a tela de "Detalhes", onde podem ser visualizadas informações desejadas por ele sobre as acomodações coletadas até então pela pesquisa, assim como interagir com alguns filtros com o potencial de

Figura 4.7 – Tela exibindo listagem com todas as pesquisas solicitadas pelo usuário



Código Identificador da Pesquisa	Cidade	Status	Logs	Data da Pesquisa	
82	Ouro Preto, Minas Gerais	11	Realizando pesquisa por Ouro Preto, Minas Gerais	18/Aug/2023	✓ i
66	Ouro Preto, Minas Gerais	11	Realizando pesquisa por Ouro Preto, Minas Gerais	15/Aug/2023	✓ i

realizar análises sobre os dados. A Figura 4.8 exibe esta página. Como desejado pelo usuário, os filtros da tela permitem que a data seja agrupada em 3 grupos distintos via algoritmos K-Derivados, onde cada cor dos pontos no mapa indica o pertencimento a um grupo distinto.

Assim, é concluído, com êxito, o cenário descrito.

Figura 4.8 – Tela exibindo dados sobre acomodações coletadas por uma pesquisa específica

< Detalhes da pesquisa V

Ver detalhes da execução
Baixar dados filtrados
Visualizar anúncios por origem

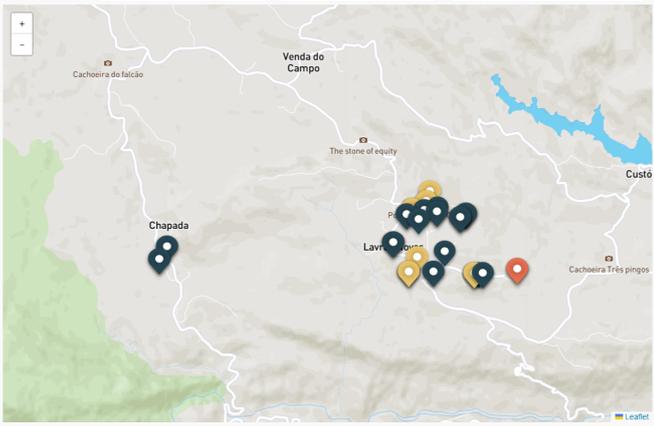
Quantidade De Banheiros

Código Identificar da Acomodação

Código Identificador do Anfitrião (do Hotel para Booking)

Preço

Limpar filtros
Filtrar



Gráfico

Selecione abaixo as colunas para relacionar dados em um gráfico de barras. Se estiver com dúvidas sobre o que selecionar, clique aqui para obter mais informações

Informação a se obter *

Contagem
 Média
 Maior valor
 Menor valor

Campo categórico *

Tipo De Quarto
 Tipo De Propriedade
 Código Identificador da Localização
 Cidade
 Bairro
 Rua

Campo numérico *

Nenhum
 Estadia Mínima
 Quantidade De Reviews
 Quantidade De Pessoas Acomodadas
 Quantidade De Quartos
 Classificação Média
 Quantidade De Banheiros
 Preço

Gerar gráfico

Dados filtrados

Tipo De Quarto	Estadia Mínima	Tipo De Propriedade	Quantidade De Reviews	Quantidade De Pessoas Acomodadas	Quantidade De Quartos	Código Identificador da Localização	Classificação Média	Cidade	Bairro	Rua	Comodidades	E Super Anfitrião	Idiomas Do Anfitrião	Tipo De De Banheiro
Entire home/apt	2	Entire cottage	34	10	4	826	4.74	Mariana	Liberdade	AQU-434		0	{en,es}	5 baths
Entire home/apt	2	Entire rental unit	91	8	3	177	4.96	Mariana	Liberdade	Rua E		1	{}	2 baths
Entire home/apt	2	Entire home	160	8	4	89	4.87	Mariana	Liberdade	Rua G		1	{}	4 baths
Entire home/apt	2	Entire loft	41	2	1	912	4.51	Ouro Branco	Luzia Augusta da Silva			0	{pt}	1 bath
Entire home/apt	2	Tiny home	28	3	1	228	4.93	Ouro Branco	Siderurgia	Rua Rio Grandense		0	{en,pt}	1 bath
Entire home/apt	2	Entire cottage	76	2	1	15	4.95	Ouro Branco		Rua Padre Marcelo Braga		1	{fr,pt}	1 bath
Entire home/apt	2	Entire cottage	125	2	0	33	4.98	Ouro Branco		Rua Padre Marcelo Braga		1	{fr,pt}	1 bath
Entire home/apt	2	Entire cottage	37	2	0	432	5	Ouro Branco		Rua Padre Marcelo Braga		1	{fr,pt}	1 bath
Entire home/apt	2	Entire cottage	33	2	0	455	5	Ouro Branco		Rua Padre Marcelo Braga		1	{fr,pt}	1 bath
Entire home/apt	2	Entire cottage	76	2	1	484	4.95	Ouro Branco		Rua Padre Marcelo Braga		1	{fr,pt}	1 bath

5 Conclusão

Nos últimos anos, há um avanço significativo das economias de compartilhamento e plataformas. Esses modelos econômicos têm transformado a forma como as pessoas consomem (e pensam sobre o consumo de) produtos e serviços. As economias de compartilhamento são baseadas no princípio de que é mais eficiente utilizar recursos existentes de forma colaborativa, em vez de possuir tudo individualmente. Essa ideia tem ganhado popularidade com o surgimento de plataformas digitais que se promovem indicando uma intenção de facilitar o compartilhamento de bens, como veículos, imóveis e equipamentos.

Plataformas como o Airbnb permitem que pessoas aluguem quartos ou apartamentos de proprietários locais, proporcionando aos viajantes uma experiência mais autêntica e aos anfitriões uma fonte de renda extra. Aos consumidores, essa abordagem tem se mostrado uma alternativa viável aos hotéis tradicionais, permitindo que mais pessoas tenham acesso a hospedagens a preços mais acessíveis.

Apesar do sucesso do Airbnb, tanto financeiramente quanto em termos de seu modelo de negócios, essa plataforma também enfrenta críticas e oposição de várias entidades relacionadas ao setor imobiliário e hoteleiro. Além disso, o Booking também enfrenta desafios e contradições, especialmente durante a pandemia, quando as vendas diretas dos hotéis aumentaram. As redes hoteleiras veem essa venda direta como uma forma de ter maior independência na definição de preços e evitar comissões consideradas abusivas pelas OTAs, como o Booking. Entretanto, apesar de já existirem discussões que pontuem tais tópicos, na realidade brasileira ainda é difícil (ou até mesmo impossível, em casos de inexistência) acessar dados que sejam capazes de validar as teorias apontadas.

Diante disso, se viu como relevante o desenvolvimento de um sistema web que disponibiliza publicamente um grande volume de dados do Airbnb e Booking colabora diretamente com pesquisadores e curiosos que buscam acessar e analisar essas informações. Ao oferecer uma plataforma que coleta, organiza e estrutura esses dados de maneira eficiente, o sistema permite que diferentes análises e técnicas matemáticas sejam aplicadas. Essa disponibilidade aumenta consideravelmente a acessibilidade e a usabilidade dos dados, proporcionando insights valiosos sobre o comportamento dos usuários, as tendências do mercado e as preferências dos viajantes.

Uma das principais vantagens do DashAcomodações é a facilidade com que os usuários podem navegar e extrair informações valiosas de um grande volume de dados. Anteriormente, era um desafio encontrar e analisar essas informações, demandando tempo e recursos significativos. No entanto, com essa nova plataforma, os usuários podem ter acesso rápido e seguro a uma vasta gama de dados, facilitando a elaboração de estudos e análises mais aprofundadas.

Outra grande contribuição deste trabalho é a sua capacidade de fornecer dados de forma

organizada e estruturada. Ao agrupar e categorizar essas informações, o sistema permite que os pesquisadores utilizem diferentes técnicas matemáticas e algoritmos de agrupamento para identificar padrões e tendências. Essa análise mais aprofundada pode levar a novas descobertas e auxiliar na tomada de decisões estratégicas de diferentes setores, como turismo, marketing e economia.

Além disso, ao disponibilizar os dados do Airbnb e Booking publicamente, o sistema promove a transparência e o acesso democrático à informação. Isso é essencial para a comunidade acadêmica, empresas e órgãos governamentais, pois permite que todos tenham acesso a informações relevantes para a tomada de decisões e a realização de estudos mais abrangentes.

Em conclusão, o sistema web que disponibiliza publicamente um grande volume de dados do Airbnb e Booking é um avanço significativo para a pesquisa e a análise dessas informações. Ao facilitar o acesso, aumentar a organização e permitir o uso de técnicas matemáticas e algorítmicas, ele contribui para a compreensão mais aprofundada do comportamento dos usuários e das tendências do mercado. Essa plataforma traz benefícios ao impulsionar o desenvolvimento de estudos mais abrangentes e ajudar a implementação de políticas públicas mais embasadas.

Trabalhos Futuros

Como sugestões para trabalhos futuros, é altamente recomendável que sejam direcionados esforços adicionais para aperfeiçoar e aprimorar a segurança dos dados e a infraestrutura do sistema. É fundamental que essas melhorias estejam concentradas na escalabilidade para lidar com um grande número de usuários simultâneos, além de garantir a capacidade máxima de armazenamento do banco de dados, dentre outros aspectos relevantes que devem ser considerados.

Para garantir a eficiência e a qualidade dos dados fornecidos pelo sistema, seria interessante realizar verificações mais minuciosas da acurácia dos dados. Essa averiguação deverá ser realizada de forma a avaliar o quão próximos os dados fornecidos estão dos valores reais. É imprescindível levar em consideração tanto a quantidade de dados quanto os atributos individuais e verificar a sua precisão.

Desta forma, é interessante investir em tecnologias e metodologias capazes de garantir a proteção e confiabilidade das informações armazenadas no sistema. Um sistema seguro e confiável é crucial para a integridade e o sucesso de qualquer organização nos dias atuais.

Além disso, é importante assegurar a capacidade da infraestrutura do sistema para lidar com o crescimento do número de usuários e a quantidade de dados gerados. A capacidade máxima de armazenamento do banco de dados deve ser constantemente revista e atualizada para atender às demandas crescentes.

Referências

AIRBNB. *Saiba mais sobre o Airbnb*. 2018. Disponível em: <news.airbnb.com/br/saiba-mais-sobre-o-airbnb/>.

ALCALA, O. *Selenium: Web Scraping Booking.com Accommodations*. 2019. Disponível em: <<https://python.gotrained.com/selenium-scraping-booking-com/>>.

ARTASANCHEZ, A.; JOSHI, P. *Artificial Intelligence with Python: Your complete guide to building intelligent*. [S.l.]: Packt Publishing, 2020. volume.

BOKEH. *Bokeh 2.1.1 Documentation*. 2020. Disponível em: <<https://docs.bokeh.org/en/2.1.1/#>>.

BOOKING.COM. *Termos de Serviço ao Cliente*. 2022. Disponível em: <<https://secure.booking.com/content/terms.pt-br.html?aid=356980&label=gog235jc-1BCBQoggJCC2hvd193ZV93b3JrSDNYA2ggiAEBmAETuAEXyAEM2AEB6AEBiAIBqAIDuAId=1c98f1e986c950170f33d468427f3730>>.

CANSOY, M.; SCHOR, J. *Who Gets to Share in the “ Sharing Economy ” : Understanding the Patterns of Participation and Exchange in Airbnb*. 2016. Disponível em: <http://www.bc.edu/content/dam/files/schools/cas_sites/sociology/pdf/SharingEconomy.pdf>.

CARVALHO, M. de. *Construindo o saber: técnicas de metodologia científica*. [S.l.]: Papirus Editora, 1989. ISBN 9788530800710.

COMPANIESMARKETCAP.COM. *Market capitalization of Airbnb (ABNB)*. 2023. Disponível em: <<https://companiesmarketcap.com/airbnb/marketcap/>>.

COX, M.; SLEE, T. *How Airbnb’s data hid the facts in New York City*. 2016. Disponível em: <<http://insideairbnb.com/reports/how-airbnbs-data-hid-the-facts-in-new-york-city.pdf>>.

ESTER, M.; KRIEGEL, H.-P.; SANDER, J.; XU, X. *A density-based algorithm for discovering clusters in large spatial databases with noise*. 1996. Disponível em: <<https://cdn.aaii.org/KDD/1996/KDD96-037.pdf>>.

GONÇALVES, R. F.; GAVAA, V. L.; FLEURY, A. L.; PESSÔA, M. S. de P.; SPINOLA, M. de M. *Uma abordagem sistêmica do processo de produção em engenharia web, na fase de concepção*. 2016. Disponível em: <<https://www.scielo.br/j/prod/a/t7yKxQh88x3wwq6cdw5rh8w/?lang=pt&format=pdf>>.

GUPTA, S. *Airbnb Rental Listings Dataset Mining*. 2019. Disponível em: <<https://towardsdatascience.com/airbnb-rental-listings-dataset-mining-f972ed08ddec>>.

HAN, J.; KAMBER, M.; PEI, J. *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers. 2001. Disponível em: <<http://myweb.sabanciuniv.edu/rdehkharghani/files/2016/02/The-Morgan-Kaufmann-Series-in-Data-Management-Systems-Jiawei-Han-Micheline-Kamber-Jian-Pei-Data-Mining-Concepts-and-Techniques-3rd-Edition-Morgan-Kaufmann-2011.pdf>>.

HUANG, Z. *Extensions to the k-Means Algorithm for Clustering Large Data Sets with Categorical Values*. 1998. Disponível em: <<https://typeset.io/papers/extensions-to-the-k-means-algorithm-for-clustering-large-1w7dxjag2>>.

- MIRANDA, G. *Lisboa deve limitar oferta de Airbnb e de outros aluguéis turísticos*. 2018. Disponível em: <<https://www1.folha.uol.com.br/mundo/2018/07/lisboa-vai-limitara-oferta-de-airbnb-e-de-outros-alugueis-turisticos.shtml>>.
- MITCHELL, R. *Web Scraping com Python: Coletando mais dados da web moderna*. 2019. Disponível em: <https://books.google.com.br/books?id=95qODwAAQBAJ&dq=web+scraping+o+que+\%C3\%A9&lr=&hl=pt-BR&source=gbs_navlinks_s>.
- MORAES, D. *Web scraping: saiba o que é e como você pode usar a seu favor!* 2020. Disponível em: <<https://rockcontent.com/br/blog/web-scraping/>>.
- PANDAS. *User Guide - pandas 1.2.3 documentation*. 2020. Disponível em: <https://pandas.pydata.org/docs/user_guide/index.html#user-guide>.
- PLATFORM, G. M. *Overviews | Geocoding API*. 2020. Disponível em: <<https://developers.google.com/maps/documentation/geocoding/overview>>.
- PORTES, J. H. *ECONOMIA DE PLATAFORMA: Um Estudo Sobre Redes de Colaboração na Nova Configuração do Trabalho*. 2019. Disponível em: <https://www.pucsp.br/sites/default/files/download/efisus/Economia_de_Plataforma.pdf>.
- POSTGRESQL. *PostgreSQL: Documentation*. 2020. Disponível em: <<https://www.postgresql.org/docs/>>.
- PRESSMAN, R. S.; MAXIM, B. R. *Engenharia de Software - 9ª Edição*. 2021. Disponível em: <https://books.google.com.br/books?id=FSE3EAAAQBAJ&dq=concep\%C3\%A7\%C3\%A3o+de+software+pressman&lr=&hl=pt-BR&source=gbs_navlinks_s>.
- RAMPAZZO, L. *Metodologia científica*. [S.l.]: Edições Loyola, 2005. ISBN 9788515024988.
- REDIS. *Redis*. 2020. Disponível em: <<https://redis.io/>>.
- RODRIGUES, A. C. G. F. A. *Text Mining of Airbnb Reviews*. 2021. Disponível em: <<https://run.unl.pt/bitstream/10362/127908/1/TGI0454.pdf>>.
- SAXENA, A.; PRASAD, M.; GUPTA, A.; BHARILL, N.; PATEL, O. P.; TIWARI, A.; JOO, E. M.; WEIPING, D.; CHIN-TENG, L. *A review of clustering techniques and developments*. 2017. Disponível em: <<https://www.sciencedirect.com/science/article/abs/pii/S0925231217311815?via\%3Dihub>>.
- SLEE, T. *Airbnb Data Collection: Methodology and Accuracy*. 2019. Disponível em: <<https://tomslee.net/airbnb-data-collection-methodology-and-accuracy>>.
- SLEE, T. *Uberização: a Nova Onda do Trabalho Precarizado*. [S.l.]: Editora Elefante, 2019. ISBN 8593115071.
- SLEE, T. *Airbnb web site scraper*. 2020. Disponível em: <<https://github.com/tomslee/airbnb-data-collection>>.
- STREITFELD, D. *Airbnb Listings Mostly Illegal, New York State Contends*. 2014. Disponível em: <<https://www.nytimes.com/2014/10/16/business/airbnb-listings-mostly-illegal-state-contends.html>>.

SØRENSEN, T. *A Method of Establishing Groups of Equal Amplitude in Plant Sociology Based on Similarity of Species and Its Application to Analyses of the Vegetation on Danish Commons*. 1948. Disponível em: <https://www.royalacademy.dk/Publications/High/295_S%C3%B8rensen,%20Thorvald.pdf>.

TOURINEWS, R. *¿'Adiós' a Booking?: Los hoteles españoles reducen su dependencia de las OTA*. 2022. Disponível em: <https://www.tourinews.es/resumen-de-prensa/notas-de-prensa-empresas-turismo/espana-hoteles-reducen-dependencia-booking-expedia-ota_4469701_102.html>.

ZHANG, T.; RAMAKRISHNAN, R.; LIVNY, M. *BIRCH: A New Data Clustering Algorithm and Its Applications*. 1997. Disponível em: <<https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=ce9eb450a6598075b66eb78b6131353e56823167>>.

Apêndices

APÊNDICE A – Descrição dos atributos da Entidades do Sistema

Tabela A.1 – Descrição dos atributos da entidade Acomodação do Airbnb

Atributo	Semântica	Tipo de Dado
room_id	identificador da acomodação (chave primária)	int(11)
location_id	identificador da Localização associada à acomodação (chave secundária)	int(11)
survey_id	Pesquisa em qual a acomodação foi encontrada (chave secundária)	double precision(11)
accommodates	quantidade de hóspedes suportados pela acomodação	int(11)
address	endereço escrito da acomodação indicado pelo site	varchar(255)
bathroom_type	tipo de banheiro da acomodação	double precision(11)
bathrooms	quantidade de banheiros da acomodação	int(11)
bedrooms	quantidade de camas da acomodação	int(11)
comodities	comodidades da acomodação	varchar(1055)
extra_host_languages	idiomas dominados pelo anfitrião, além do português	double precision(11)
host_id	identificador do anfitrião da acomodação	int(11)
is_superhost	indica se é um "superanfitrião" ou não	boolean
latitude	coordenada de latitude do endereço	numeric(30,6)
longitude	coordenada de longitude do endereço	numeric(30,6)
max_nights	quantidade máxima de dias de estadia da acomodação	double precision(11)
minstay	quantidade mínima de dias de estadia da acomodação	double precision(11)
name	nome da acomodação	varchar(255)
overall_satisfaction	classificação média das avaliações da acomodação	double precision(11)
price	preço da acomodação	double precision(11)
property_type	tipo de propriedade da acomodação	double precision(11)
reviews	quantidade de avaliações da acomodação	int(11)
room_type	tipo da acomodação	varchar(255)

Tabela A.2 – Descrição dos atributos da entidade Acomodação do Booking

Atributo	Semântica	Tipo de Dado
room_id	identificador do quarto (chave primária)	int(11)
location_id	identificador da Localização associada à acomodação (chave secundária)	int(11)
survey_id	pesquisa em qual a acomodação foi encontrada (chave secundária)	double precision(11)
hotel_id	identificador do quarto (chave primária)	int(11)
accommodates	quantidade de hóspedes suportados pelo quarto	int(11)
address	endereço escrito do hotel indicado pelo site	varchar(255)
bed_type	quantidade e tipos de camas do quarto	varchar(255)
comodities	comodidades do quarto	varchar(1055)
hotel_name	nome do hotel ao qual o quarto pertence	varchar(255)
latitude	coordenada de latitude do hotel	numeric(30,6)
longitude	coordenada de longitude do hotel	numeric(30,6)
overall_satisfaction	classificação média das avaliações do hotel	double precision(11)
price	preço do quarto	double precision(11)
property_type	tipo de propriedade do hotel	varchar(255)
reviews	quantidade de avaliações do hotel	int(11)
room_name	nome do quarto	varchar(255)
checkin_date	data inicial da hospedagem coletada	int(11)
checkout_date	data final da hospedagem coletada	int(11)

Tabela A.3 – Descrição dos atributos da entidade Área de Pesquisa

Atributo	Semântica	Tipo de Dado
search_area_id	identificador da área de pesquisa (chave primária)	int(11)
name	nome da área de pesquisa	varchar(255)
abbreviation	nome da área de pesquisa sem caracteres especiais	varchar(255)
north_latitude	latitude norte da caixa delimitadora	numeric(30,6)
south_latitude	latitude sul da caixa delimitadora	numeric(30,6)
east_longitude	longitude leste da caixa delimitadora	numeric(30,6)
west_longitude	longitude oeste da caixa delimitadora	numeric(30,6)

Tabela A.4 – Descrição dos atributos da entidade Configuração de Pesquisa

Atributo	Semântica	Tipo de Dado
config_id	identificador da configuração (chave primária)	varchar(255)
ss_id	identificador da super pesquisa associada a configuração (chave secundária)	int(11)
user_id	código identificar do usuário que solicitou a Super Pesquisa (chave secundária)	varchar(30)
data_columns	lista de nomes das colunas a serem coletadas na super pesquisa	varchar(30)
platform	área a ser buscada na super pesquisa	varchar(255)
search_area_name	nome da área a ser buscada na super pesquisa	varchar(255)

Tabela A.5 – Descrição dos atributos da entidade Localização

Atributo	Semântica	Tipo de Dado
location_id	identificador do endereço (chave primária)	int(11)
latitude	coordenada de latitude do endereço	numeric(30,6)
longitude	coordenada de longitude do endereço	numeric(30,6)
route	rua do endereço	varchar(255)
sublocality	bairro do endereço	varchar(255)
city	cidade do endereço	varchar(255)
country	país do endereço	varchar(255)

Tabela A.6 – Descrição dos atributos da entidade Pesquisa

Atributo	Semântica	Tipo de Dado
survey_id	identificador da pesquisa (chave primária)	int(11)
search_area_id	identificador da área de pesquisa associada (chave secundária)	int(11)
super_survey_id	identificador da super pesquisa associada (chave secundária)	int(11)
description	descrição opcional da pesquisa	varchar(255)
status	status da pesquisa (aberta, em andamento, concluída)	smallint
search_date	data da pesquisa	date

Tabela A.7 – Descrição dos atributos da entidade Super Pesquisa

Atributo	Semântica	Tipo de Dado
super_survey_id	identificador da super pesquisa (chave primária)	int(11)
user_id	código identificador do usuário que solicitou a super pesquisa	integer
area	nome da área associada à super pesquisa	varchar(255)
date	data da super pesquisa	date
logs	status atual da pesquisa	character varying(100)
all_logs	texto contendo todos os status da super pesquisa, incluindo antigos e atual	character varying(1000)

Tabela A.8 – Descrição dos atributos da entidade Usuário

Atributo	Semântica	Tipo de Dado
email	e-mail do usuário (chave primária)	varchar(255)
name	nome do usuário	varchar(255)
password	senha do usuário	varchar(255)
permission	tipo de permissão do usuário	varchar(30)