

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ  
DEPARTAMENTO ACADÊMICO DE ELETRÔNICA  
DEPARTAMENTO ACADÊMICO DE INFORMÁTICA  
CURSO DE ENGENHARIA DE COMPUTAÇÃO

EVERTON SANTOS BARRETO JUNIOR

**BUSKAKI EMPRESAS - FERRAMENTA PARA BUSCA DE DADOS  
ABERTOS DE EMPRESAS CURITIBANAS**

TRABALHO DE CONCLUSÃO DE CURSO

CURITIBA

2022

**EVERTON SANTOS BARRETO JUNIOR**

**BUSKAKI EMPRESAS - FERRAMENTA PARA BUSCA DE DADOS  
ABERTOS DE EMPRESAS CURITIBANAS**

Trabalho de Conclusão de Curso do curso de Curso de Engenharia de Computação do Departamento Acadêmico de Eletrônica e Departamento Acadêmico de Informática da Universidade Tecnológica Federal do Paraná como requisito parcial para obtenção do grau de Engenheiro de Computação

Orientadora: Profa. Dra. Nádia Puchalski Kozievitch

**CURITIBA**

**2022**

**Everton Santos Barreto Junior**

**Buskaki Empresas: Ferramenta para busca de dados abertos de empresas  
curitibanas**

Trabalho de Conclusão de Curso de Graduação  
apresentado como requisito para obtenção do título  
de Bacharel em Engenharia de Computação da  
Universidade Tecnológica Federal do Paraná  
(UTFPR).

Data de aprovação: 22/junho/2021

---

Nádia Puchalski Kozievitch  
Doutorado em Ciência da Computação  
Universidade Tecnológica Federal do Paraná

---

Marina Esther Groshaus  
Doutorado em Ciência da Computação  
Universidade Tecnológica Federal do Paraná

---

Rita Cristina Galarraga Berardi  
Doutorado em Informática  
Universidade Tecnológica Federal do Paraná

**Curitiba**

**2021**

## **AGRADECIMENTOS**

Dedico meus sinceros agradecimentos, primeiramente, à minha orientadora, Profa. Nádia Puchalski Kozievitch. Muito obrigado pelos seus ensinamentos, sua dedicação e seu apoio nessa jornada que caminhamos juntos.

Agradeço à UTFPR, na forma de sua Instituição e docentes, pela excelência na formação e pelas oportunidades concedidas. Nesta universidade, devo um obrigado especial aos fantásticos professores, alguns com quem tive o privilégio de ter aula, enquanto outros tive a sorte de conhecer pelos corredores, que sempre me apoiaram e acreditaram em mim.

Principalmente, acima de qualquer gratidão que eu possa expressar, agradeço à minha família e amigos pela presença em minha vida. Aos meus pais, Jane e Everton, pela vida, pelo amor e pelos ensinamentos que me deram com tanto carinho em todos esses anos. À minha companheira, Andréia, por me dar os abraços e o conforto que tanto precisei em momentos de desespero. Às minhas filhas de quatro patas, Morgana e Kiara, que me forneceram carinho e conforto em vários momentos difíceis. À todos os meus grandes amigos, com quem partilhei tantos momentos importantes de minha vida. Amo vocês!

## RESUMO

BARRETO JUNIOR, Everton Santos. BUSKAKI EMPRESAS - FERRAMENTA PARA BUSCA DE DADOS ABERTOS DE EMPRESAS CURITIBANAS. 82 f. Trabalho de Conclusão de Curso – Curso de Engenharia de Computação, Universidade Tecnológica Federal do Paraná. Curitiba, 2022.

A busca de dados de Pessoa Jurídica é um processo relevante para diversas entidades, seja para consultar uma fonte confiável de dados sobre potenciais parceiros de negócios, para estudos relacionados a desenvolvimento urbano ou para simplesmente coletar informações sobre determinada empresa. Porém, a maioria das ferramentas de busca existentes não disponibilizam a consulta pelo nome da empresa, exigindo que o usuário tenha conhecimento do Cadastro Nacional de Pessoa Jurídica (CNPJ) da empresa. Apesar de serem dados abertos, não há uma plataforma gratuita e de fácil acesso para consultá-los. Neste contexto, este projeto tem como objetivo modelar e construir uma aplicação web que considere essa questão permitindo que os usuários possam consultar dados de empresas curitibanas utilizando outros campos além do CNPJ. A aplicação tira proveitos de conceitos de banco de dados, sistemas de informação geográfica (SIG), cidades inteligentes e APIs. A ferramenta utiliza dados abertos e aplica algoritmos de comparação textual e *entity matching* para aumentar a abrangência dos resultados da busca.

**Palavras-chave:** banco de dados, SIG, cidades inteligentes, algoritmos de comparação textual, Curitiba, CNPJ, *entity matching*

## ABSTRACT

BARRETO JUNIOR, Everton Santos. Buskaki Empresas - A Search Tool for Curitiba's companies open data. 82 f. Trabalho de Conclusão de Curso – Curso de Engenharia de Computação, Universidade Tecnológica Federal do Paraná. Curitiba, 2022.

The search for Legal Entity data is a relevant process for several entities, whether to consult a reliable source of data about potential business partners, for studies related to urban development or to simply collect information about a specific company. However, most of the existing tools do not provide search by the name of the company, requiring the user to be aware of its National Register of Legal Entities (Cadastro Nacional da Pessoa Jurídica in Portuguese – CNPJ). Despite being open data, there is no free and easily accessible platform for consulting them. In this context, this project aims to model and build a web application that considers this issue, allowing users to consult data from Curitiba companies using fields other than CNPJ. The application takes advantage of database, geographic information systems (GIS), smart cities and APIs concepts. The tool uses open data and applies textual comparison and entity matching algorithms to increase the range of search results.

**Keywords:** databases, GIS, string matching, smart city, Curitiba, CNPJ, entity matching

## LISTA DE FIGURAS

FIGURA 1	– Diagrama de áreas de abrangência do projeto .....	12
FIGURA 2	– Interface de busca da JUCESP (A) & Interface de busca do Portal REDE-SIM (B) .....	29
FIGURA 3	– Busca do app CNPJ (A) & Busca do app Consultar CNPJ (B) .....	29
FIGURA 4	– Etapas da metodologia proposta .....	31
FIGURA 5	– Diagrama de casos de uso .....	34
FIGURA 6	– Esquema do banco de dados .....	36
FIGURA 7	– Arquitetura da aplicação .....	38
FIGURA 8	– Trecho do <i>model</i> EmpresaReceitaCuritiba .....	41
FIGURA 9	– Trecho da <i>view</i> resultado_busca_por_nome .....	42
FIGURA 10	– Tela de resultado da busca por nome .....	43
FIGURA 11	– Trecho do <i>controller</i> EmpresaReceitaCuritibaController .....	44
FIGURA 12	– As 10 grafias mais recorrentes para a Alameda Dom Pedro II .....	49
FIGURA 13	– Comparação de serviços de <i>geocoding</i> .....	50
FIGURA 14	– Tela de login .....	67
FIGURA 15	– Tela de cadastro .....	68
FIGURA 16	– Formulário para busca por nome .....	68
FIGURA 17	– Formulário para busca avançada .....	69
FIGURA 18	– Formulário para busca por rua .....	69
FIGURA 19	– Formulário para busca por bairro .....	70
FIGURA 20	– Formulário para busca por CNPJ .....	70
FIGURA 21	– Tela de resultados da busca .....	71
FIGURA 22	– Tela de informações da empresa .....	71
FIGURA 23	– Tela Meu perfil .....	72
FIGURA 24	– Tela Histórico de buscas .....	72

## LISTA DE TABELAS

TABELA 1	– Diferentes referências para um mesmo endereço na base de CNPJ da Receita Federal .....	18
TABELA 2	– Código dos grupos fonéticos do soundex .....	21
TABELA 3	– Código dos grupos fonéticos do soundex-br .....	22
TABELA 4	– Empresas curitibanas em maior número .....	37
TABELA 5	– Bairros de Curitiba com mais empresas .....	37
TABELA 6	– Análise de algoritmos de comparação textual - Etapa 2: Alameda Augusto Stellfeld .....	46
TABELA 7	– Análise de algoritmos de comparação textual - Etapa 3: Alameda Augusto Stellfeld .....	47
TABELA 8	– Análise de algoritmos de comparação textual - Resultado .....	47



## SUMÁRIO

<b>1 INTRODUÇÃO</b>	<b>10</b>
1.1 OBJETIVOS	13
1.1.1 Objetivos Específicos	13
1.2 CONTRIBUIÇÕES DO TRABALHO	13
1.3 ESTRUTURA DO DOCUMENTO	14
<b>2 FUNDAMENTAÇÃO TEÓRICA E TECNOLOGIAS UTILIZADAS</b>	<b>15</b>
2.1 CIDADES INTELIGENTES	15
2.1.1 Dados abertos	16
2.2 BANCO DE DADOS	17
2.2.1 <i>Entity Matching</i>	18
2.2.1.1 Distância de Levenshtein	20
2.2.1.2 Soundex	21
2.2.1.3 Metaphone	23
2.3 GEOPROCESSAMENTO	23
2.3.1 Definições de Geoprocessamento	24
2.3.2 Aplicações do Geoprocessamento	24
2.3.3 Geocodificação	25
2.3.3.1 Softwares de Geocodificação	26
2.4 <i>APPLICATION PROGRAMMING INTERFACES</i>	27
2.5 BUSKAKI EMPRESAS	28
2.5.1 Aplicações semelhantes	28
<b>3 METODOLOGIA</b>	<b>31</b>
3.1 REVISÕES BIBLIOGRÁFICAS	31
3.2 ANÁLISE E MODELAGEM DO SISTEMA	31
3.3 DESENVOLVIMENTO E AVALIAÇÃO	32
3.4 CONCLUSÃO	32
<b>4 MODELAGEM DO SISTEMA</b>	<b>33</b>
4.1 DIAGRAMAÇÃO	33
4.1.1 Casos de uso	33
4.1.1.1 Busca Avançada	33
4.1.1.2 visualizar informações de uma empresa	34
4.1.1.3 Acessar histórico	35
4.1.2 Esquema dos dados	35
<b>5 DESENVOLVIMENTO</b>	<b>38</b>
5.1 ARQUITETURA DA APLICAÇÃO	38
5.2 TECNOLOGIAS UTILIZADAS	38
5.3 OBTENÇÃO DOS DADOS	40
5.4 DESCRIÇÃO DO DESENVOLVIMENTO DA ESTRUTURA DO PROJETO	40
5.5 DESCRIÇÃO DO DESENVOLVIMENTO DOS ALGORITMOS E CONSULTAS	
SQL	44
5.6 ANÁLISE DOS ALGORITMOS DE COMPARAÇÃO TEXTUAL	46

5.7	LIMITAÇÕES E DESAFIOS ENCONTRADOS .....	47
<b>6</b>	<b>CONCLUSÃO .....</b>	<b>51</b>
	<b>REFERÊNCIAS .....</b>	<b>53</b>
	<b>Apêndice A – ALGORITMO SOUNDEX BR .....</b>	<b>57</b>
	<b>Apêndice B – CASOS DE USO .....</b>	<b>59</b>
B.1	BUSCAR EMPRESA POR NOME .....	59
B.2	BUSCA AVANÇADA .....	59
B.3	VISUALIZAR PERFIL .....	60
B.4	ACESSAR HISTÓRICO .....	60
B.5	VISUALIZAR INFORMAÇÕES DE UMA EMPRESA .....	60
B.6	CRIAR CONTA .....	60
B.7	LOGIN .....	61
B.8	LOG OUT .....	61
B.9	BUSCAR EMPRESA POR RUA .....	61
B.10	BUSCAR EMPRESA POR CNPJ .....	62
B.11	BUSCAR EMPRESA POR BAIRRO .....	62
	<b>Apêndice C – ESQUEMA DE DADOS .....</b>	<b>63</b>
C.1	TABELA DE EMPRESAS .....	63
C.2	TABELA DE ALVARÁS .....	65
C.3	TABELA DE USUÁRIOS .....	66
C.4	TABELA DE HISTÓRICO DE BUSCA .....	66
	<b>Apêndice D – TELAS DA APLICAÇÃO .....</b>	<b>67</b>
D.1	PÁGINA INICIAL - LOGIN .....	67
D.2	criação de cadastro .....	68
D.3	FORMULÁRIO DE BUSCA POR NOME .....	68
D.4	FORMULÁRIO DE BUSCA AVANÇADA .....	69
D.5	FORMULÁRIO DE BUSCA POR RUA .....	69
D.6	FORMULÁRIO DE BUSCA POR BAIRRO .....	70
D.7	FORMULÁRIO DE BUSCA POR CNPJ .....	70
D.8	RESULTADOS DA BUSCA .....	71
D.9	INFORMAÇÕES DA EMPRESA .....	71
D.10	MEU PERFIL .....	72
D.11	HISTÓRICO DE BUSCA .....	72
	<b>Apêndice E – ALGORITMOS DE BUSCA .....</b>	<b>73</b>
E.1	BUSCA POR CNPJ .....	73
E.2	BUSCA POR AVANÇADA .....	74
E.3	BUSCA POR BAIRRO .....	75
E.4	BUSCA POR RUA .....	76
	<b>Apêndice F – TABELAS DA ANÁLISE DOS ALGORITMOS DE COMPARAÇÃO</b>	
	<b>TEXTUAL .....</b>	<b>77</b>
F.1	ETAPA DE LISTAGEM .....	78
F.2	ETAPA FINAL .....	81

## 1 INTRODUÇÃO

Segundo Lemos [1], nos anos 90, o debate entre as novas tecnologias de informação e comunicação (TIC) e o espaço urbano estava prestes a cunhar o termo “cidades digitais”. O objetivo era implantar nesse espaço urbano uma infraestrutura digital eficiente, com intuito de fomentar processos inovadores nas estruturas de governo, nas empresas e no comércio. A intenção era reaquecer o ambiente público, ampliar os laços comunitários e a participação política. Nessa proposta, a inclusão social se daria pela democratização do acesso a equipamentos tecnológicos e redes digitais. Em relação à esfera política, os projetos visavam criar ferramentas de *e-government*, para promover a transparência por meio de ouvidorias, com a intenção de conferir maior poder de influência e de decisão ao cidadão.

Essas iniciativas continuam a ser implementadas e aperfeiçoadas, mas o crescimento da Internet trouxe mudanças substanciais a esse cenário. Houve uma ampliação considerável do acesso à Internet nas várias camadas da sociedade brasileira e mundial. Também assistimos a um acelerado e agressivo desenvolvimento de sistemas de geolocalização, acesso, consumo e distribuição de informação. Isso se deu principalmente com o advento da computação em nuvem (impulsionada pela popularização de *smartphones* e *tablets*), da expansão de formas de tratamento de dados — o chamado *Big Data* — e da comunicação entre os objetos — nomeada de “Internet das Coisas”(IoT), a qual é viabilizada por sensores e etiquetas de radiofrequência instalados nos mais diversos artefatos, dotando-os de capacidade infocomunicacional em rede.

Saber trabalhar com grandes quantidades de dados procedentes de diversas localidades e com diferentes formatos é uma das habilidades mais desejadas na última década [2]. Isso ocorre devido ao crescimento exponencial dos dados gerados pela sociedade e à necessidade de minerar informações obtidas por meio da análise das conexões semânticas entre conceitos e relações presentes nestes dados.

As cidades precisam ser pensadas de forma abrangente, inovadora e urgente, com planos efetivos e verossímeis, utilizando os avanços das TIC em seu favor. Weiss et al. [3] aponta que organizações governamentais e não governamentais, firmas de consultoria especializadas

e os principais provedores globais de tecnologias têm se debruçado sobre as questões que envolvem a manutenção das condições e funcionalidades dos espaços e infraestruturas urbanas, buscando desenvolver soluções que atendam de forma plena às crescentes demandas de todos os atores que atuam nas cidades, apoiando-se sobre as tecnologias da informação e comunicação como a principal ferramenta para a realização das expectativas e objetivos da dinâmica global contemporânea.

Capital do estado do Paraná, também situada na região sul do país, tem sua história e desenvolvimento marcados por fortes influências européias, resultantes de intensa migração de alemães, italianos, poloneses e ucranianos. Curitiba, que é uma cidade planejada, é o centro econômico do estado do Paraná e apresenta o quarto maior PIB do país (IBGE, 2012). Desde a década de 70, se tornou um modelo mundial de transporte, urbanização e respeito ao meio ambiente, sendo considerada uma das 10 cidades mais inteligentes do mundo. Nos últimos anos, mais precisamente a partir da década de 1980, o poder público vem realizando investimentos no planejamento e execução de modernizações nos sistemas de infraestrutura da cidade, com especial atenção às questões de mobilidade.

O Nota Paraná<sup>1</sup> é um programa do governo do Paraná de combate à sonegação fiscal, que consiste em uma estratégia em que se devolve uma parte do imposto pago por empresas na venda de produtos ou serviços. O consumidor pode cadastrar seu CPF na nota fiscal para receber esse retorno financeiro ou doar a nota sem o CPF cadastrado a entidades sociais depositando-a em urnas localizadas nesses estabelecimentos. Essas são recolhidas pelas entidades, que recebem a contribuição monetária para cada nota que cadastram no sistema do Nota Paraná.

A Associação de Amigas da Mama <sup>2</sup> (AAMA) é uma ONG curitibana destinada a apoiar e ajudar mulheres diagnosticadas com câncer de mama. A AAMA presta serviços sócio assistenciais gratuitos de consultoria jurídica, atendimento psicológico, artesanato e meditação. Para arcar com as diversas despesas de infraestrutura e serviços prestados, a instituição arrecada dinheiro através de doações. Para receber as doações de notas fiscais, a entidade precisa inserir dados da empresa emissora da NFC-e no sistema Nota Paraná, que pode ser um problema caso a nota fiscal não contenha todas as informações requeridas pelo sistema. Alguns voluntários da AAMA dedicam dias de seu trabalho a esse processo e em muitos casos não é possível para eles obterem todas as informações necessárias, o que causa a perda de todas as doações recebidas destes estabelecimentos.

Baseando-se no contexto apresentado, a aplicação Buskaki Empresas propõe a arqui-

---

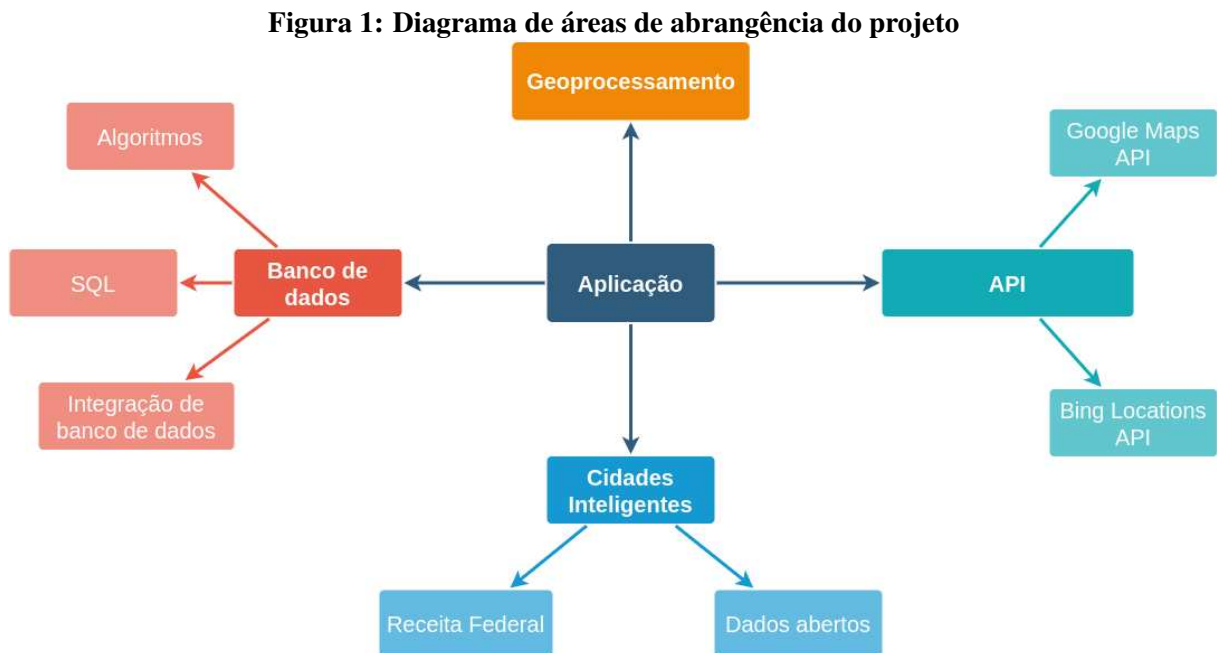
<sup>1</sup><http://www.notaparana.pr.gov.br/>

<sup>2</sup><https://www.amigasdamama.org.br/>

tetura, prototipagem e o desenvolvimento de uma aplicação web para busca de dados abertos de empresas curitibanas.

Nesta aplicação, são implementadas buscas por quatro parâmetros distintos sendo necessário apenas acesso à internet para utilizá-la. A aplicação também faz uso de informações geográficas para mostrar ao usuário a localização das empresas encontradas, facilitando a identificação do registro desejado. Os dados utilizados foram estruturados a partir de dados abertos disponibilizados pela Receita Federal<sup>3</sup>.

Para a implementação de tal aplicação, as áreas do conhecimento que são objetos de estudo, ilustradas na Figura 1, possuem um papel importante no desenvolvimento deste protótipo, sendo descritas brevemente a seguir.



Fonte: Autoria própria

- **Application Programming Interface (API):** são interfaces suportadas por linguagens de programação que permite que um componente de software acesse programaticamente outro componente [4];
- **Banco de Dados:** coleção de dados relacionados entre si que representam fatos com um significado implícito [5];

<sup>3</sup><https://receita.economia.gov.br/>

- **Cidades Inteligentes:** uma maneira eficiente de utilizar tecnologias de informação e comunicação para melhorar os serviços da cidade [6];
- **Geoprocessamento:** conjunto de conhecimentos matemáticos e tecnologias computacionais para tratamento e criação de informações georreferenciadas [7].

## 1.1 OBJETIVOS

O objetivo geral deste projeto é desenvolver uma ferramenta para busca de informações de empresas curitubanas que disponibilize outras formas de consulta além do CNPJ, auxiliando o processo de cadastro no programa de auxílio a entidades sociais do Nota Paraná como a ONG Amigas da Mama. Para isto a aplicação será baseada em banco de dados, sistemas de informação geográfica (SIG) e cidades inteligentes. O projeto utilizará dados abertos fornecidos pela Receita Federal<sup>4</sup> e aplicará algoritmos de comparação textual para aumentar a abrangência dos resultados da busca.

### 1.1.1 OBJETIVOS ESPECÍFICOS

- Realizar revisão teórica dos conceitos de algoritmos de comparação de texto, banco de dados e frameworks de desenvolvimento web, juntamente da revisão das áreas que possuam ligação à solução do problema e análise dos dados abertos das empresas do Paraná;
- Analisar e comparar as principais aplicações semelhantes disponíveis, apresentando vantagens e desvantagens;
- Modelar e desenvolver uma solução de coleta de dados de Pessoa Jurídica, focada na experiência do usuário, listando as principais dificuldades encontradas durante o desenvolvimento;
- Apresentar protótipo criado a um grupo de voluntários para avaliação do funcionamento da aplicação;
- Analisar e apresentar os resultados obtidos.

## 1.2 CONTRIBUIÇÕES DO TRABALHO

Como resultado temos a prototipagem de uma aplicação que:

---

<sup>4</sup><https://receita.economia.gov.br/>

- Disponibilize ao usuário quatro formas de realizar a consulta dos dados de uma empresa curitibana: por nome, por CNPJ, por bairro e por rua, cujos protótipos das telas podem ser consultados no apêndice D;
- Retorne ao usuário resultados próximos aos dados de entrada fornecida para busca;
- Realize consultas que permitam uso de filtros por localização;
- Utilize uma abordagem visual, através de mapas, que facilite a identificação do registro desejado;
- Tenha uma interface simples e de fácil uso para usuários de qualquer nível de conhecimento técnico.
- Crie um documento que relata o desenvolvimento e testes iniciais do protótipo.

### 1.3 ESTRUTURA DO DOCUMENTO

Este documento está dividido em 6 capítulos, incluindo esta Introdução. O capítulo 2 aborda a base teórica e as tecnologias utilizadas no desenvolvimento da ferramenta. O Capítulo 3 apresenta a metodologia proposta para o desenvolvimento deste projeto. O Capítulo 4 discute o resultado de todas as análises realizadas e ilustra o levantamento de requisitos, a modelagem do sistema proposto e do banco de dados. O capítulo 5 descreve o problema, o desenvolvimento da ferramenta propriamente dita, com detalhes de implementação e aborda os resultados obtidos com suas devidas análises. Por fim, o capítulo 6 apresenta as considerações finais do trabalho.

## 2 FUNDAMENTAÇÃO TEÓRICA E TECNOLOGIAS UTILIZADAS

Esse capítulo aborda os conceitos teóricos sobre os principais conceitos abordados pelo problema. A seção 2.1 aborda os conceitos e definições de Cidades Inteligentes e seus temas. A seção 2.2 apresenta a definição de banco de dados, conceitos e algoritmos para limpeza e integração de dados. A seguir, em 2.3 é abordado o tema Geoprocessamento. Em 2.4 são aprofundados os tópicos das APIs Google Maps e Bing Locations e por fim, em 2.5, são apresentadas as plataformas de desenvolvimento utilizadas para implementação do projeto, assim como algumas aplicações existentes com finalidades semelhantes.

### 2.1 CIDADES INTELIGENTES

Atualmente, as grandes cidades são protagonistas da economia global e, consequentemente, apresentam diversas características da globalização, sejam elas positivas, como a concentração de fluxo monetário, a concentração e expansão dos serviços e do setor manufatureiro ou também negativas, como a precariedade ou mesmo a falta de alguns serviços básicos, como educação, saúde e mobilidade [8].

Nesse sentido, a internet e os seus serviços vêm tomando grande importância quando se trata de inovação nos diversos tópicos de gestão pública. As cidades inteligentes buscam envolver a tecnologia e os serviços direcionados ao usuário, para que esse seja parte primordial no avanço da pesquisa e da inovação aplicada [9]. As cidades inteligentes têm seu crescimento econômico baseado numa governança pública participativa que investe em infraestrutura de comunicação e transporte juntamente ao uso adequado dos recursos naturais.

BATTY et al.[10] define as bases de conhecimento de uma cidade inteligente como sistemas que deverão permitir a aquisição de dados de fontes múltiplas e distribuídas, incluindo a participação online, o gerenciamento e integração desses dados numa base de dados coerente, bem como a extração de informações relevantes. Caso os dados forem abertos, a tomada de decisão pelos gestores da cidade pode obter uma perspectiva mais realista à medida que as informações tornam-se acessíveis ao público [11].



### 2.1.1 DADOS ABERTOS

Dados são definidos como abertos se não existem restrições de acesso a qualquer pessoa, a qual pode utilizá-los, modificá-los e compartilhá-los com qualquer intuito, desde que conserve-se a abertura e origem desses dados [12]. Ao disponibilizar abertamente estes dados, os interessados podem utilizar as informações de forma a misturá-las e combiná-las, agregando mais valor [13].

Os princípios dos Dados Governamentais Abertos(DGA) também conhecidos como “8 *Open Government Data Principles*”[14] ou “*Sebastopol Principles*”, afirmam que os dados devem ser:

- **Completos:** Todos os dados públicos são disponibilizados. Dados são informações eletronicamente gravadas, incluindo, mas não se limitando a, documentos, bancos de dados, transcrições e gravações audiovisuais. Dados públicos são dados que não estão sujeitos a limitações válidas de privacidade, segurança ou controle de acesso, reguladas por estatutos.
- **Primários:** Os dados são publicados na forma coletada na fonte, com a mais fina granularidade possível, e não de forma agregada ou transformada.
- **Atuais:** Os dados são disponibilizados o quanto rapidamente seja necessário para preservar o seu valor.
- **Acessíveis:** Os dados são disponibilizados para o público mais amplo possível e para os propósitos mais variados possíveis.
- **Processáveis por máquina:** Os dados são razoavelmente estruturados para possibilitar o seu processamento automatizado.
- **Acesso não discriminatório:** Os dados estão disponíveis a todos, sem que seja necessária identificação ou registro.
- **Formatos não proprietários:** Os dados estão disponíveis em um formato sobre o qual nenhum ente tenha controle exclusivo.
- **Livres de licenças:** Os dados não estão sujeitos a regulações de direitos autorais, marcas, patentes ou segredo industrial. Restrições razoáveis de privacidade, segurança e controle de acesso podem ser permitidas na forma regulada por estatutos.

Os dados abertos são, no Brasil, pautados por esses oito princípios e também por três leis. As três leis foram propostas por David Eaves [15], para contemplar o significado de Dados Abertos Governamentais:

- Se o dado não pode ser encontrado e indexado na web, ele não existe;
- Se não estiver aberto e em formato compreensível por máquina, ele não pode ser reaproveitado;
- Se algum dispositivo legal não permitir sua reaplicação, ele não é útil.

Além desses pontos, é importante ressaltar a presença das características de dados abertos presentes em legislação no Brasil, como na Lei Complementar 131/2009 (Lei da Transparência)<sup>1</sup>, que determina a disponibilização em tempo real de informações financeiras e orçamentárias da União, dos Estados, do Distrito Federal e dos Municípios, e na Lei 12.527/2011 (Lei de Acesso à Informação)<sup>2</sup>, que prevê a disponibilização de informações públicas a qualquer pessoa, seja ela física ou jurídica, sem necessidade de justificar o acesso aos dados. No caso da cidade de Curitiba, sua prefeitura disponibilizou a base de Dados Abertos<sup>3</sup>, onde existem registros de reclamações, alvarás, e de outros programas e recursos competentes à gestão municipal.

Para o desenvolvimento deste projeto, foram utilizados dados provenientes da Receita Federal do Brasil<sup>4</sup> e da Prefeitura Municipal de Curitiba<sup>5</sup>, para montar as bases de dados de CNPJ de empresas brasileiras e alvarás de funcionamento de empresas curitibanas, respectivamente.

## 2.2 BANCO DE DADOS

Banco de dados é uma coleção de dados persistentes utilizada pelos sistemas de aplicação de determinada empresa [16]. O termo banco de dados se refere a um conjunto de informações relacionadas a diversos temas e classificadas de várias maneiras, que compartilham entre si algum tipo de vínculo ou relação, que é utilizada para organizar e classificar dados. Com o surgimento das cidades inteligentes, os bancos de dados passaram a ter um impacto profundo

---

<sup>1</sup><http://www.leidatransparencia.cnm.org.br/>

<sup>2</sup><http://www.acessoainformacao.gov.br/assuntos/conheca-seu-direito/a-lei-de-acesso-a-informacao>

<sup>3</sup><http://www.curitiba.pr.gov.br/dadosabertos/>

<sup>4</sup><https://www.gov.br/receitafederal/pt-br>

<sup>5</sup><https://www.curitiba.pr.gov.br/>

no planejamento das cidades. Um bom tratamento dos dados armazenados em um banco de dados pode levar ao aperfeiçoamento do planejamento urbano.

*Entity matching* (também conhecido como Identificação de duplicatas ou Resolução de Entidades) é uma tarefa crucial para integração e limpeza de dados [17, 18, 19]. Definido como a tarefa de identificar entidades (objetos, instâncias de dados) que se referem à mesma entidade no mundo real. As entidades a serem resolvidas podem residir em fontes de dados distribuídas, tipicamente heterogêneas, ou em uma única fonte de dados, por exemplo, em um banco de dados ou no armazenamento de um mecanismo de busca.

### 2.2.1 ENTITY MATCHING

O processo de *Entity matching* é particularmente desafiador para entidades que são muito heterogêneas e de qualidade de dados limitada em relação à integridade e consistência de seus atributos. A Tabela 1 ilustra alguns das diferentes maneiras de se referir a um mesmo endereço na base de CNPJs da Receita Federal. Podemos assumir que esses dados cadastrais são inseridos manualmente por funcionários e podem conter vários problemas de qualidade, como erros de digitação, abreviações e diferentes classificações de Tipo do Logradouro.

<b>Tipo do Logradouro</b>	<b>Logradouro</b>
ALAMEDA	DOUTOR MURICY
RUA	DR MURICI
RUA	DR MURICY
ALAMEDA	DR MURICY
ALAMEDA	DR. MURICY

**Tabela 1: Diferentes referências para um mesmo endereço na base de CNPJ da Receita Federal**

O problema de *entity matching* foi definido originalmente por Newcombe et al. [20] e foi formalizado por Fellegi e Sunter [21] 10 anos depois. Desde então o problema foi considerado em diversos aspectos e por diferentes comunidades, incluindo as comunidades de pesquisa em Inteligência Artificial, pesquisa em Banco de Dados e na indústria. Numerosas abordagens vêm sendo propostas para solucionar esse problema, especialmente em casos de dados bem estruturados.

Essa técnica pode ser aplicada em diferentes áreas, como em gerenciamento de banco de dados relacional, comércio eletrônico, vinculação de registros de censos, detecção de *spam*, entre outros [22]. Por exemplo, no contexto de referências bibliográficas de publicações essa

técnica pode ser aplicada para resolução de nomes em listas de coautores. Isso atua como uma etapa importante de processos como a análise das relações de coautoria através de publicações [23], assim como permite melhor precisão e qualidade em métricas obtidas a partir da base de dados.

Devido à grande variedade de fontes de dados e de entidades para fazer o *match*, não existe um “melhor algoritmo de *entity matching*”. Uma mesma abordagem de *match* pode ter alta variação de performance para diferentes formatos de dados e domínios de aplicação. Por exemplo, já foi provado que não existe um melhor algoritmo de *string matching* universal [24, 25], ao invés disso, é usualmente mais benéfico combinar vários métodos para melhorar a assertividade da comparação, por exemplo, para considerar a similaridade de diversos atributos ou considerar o relacionamento entre as entidades. Para grandes bases de dados, também é necessário aplicar estratégias para reduzir o espaço de busca para o *entity matching* a fim de atingir um tempo de execução satisfatório.

Para Garg e Singla[26] o problema da comparação de strings consiste em comparar duas strings, uma é o texto  $T [1..n]$ , a string principal fornecida, e a outra é o padrão  $P [1..m]$ , para ser combinada com a principal, dado  $m \leq n$ . A correspondência de strings é usada de forma variada em aplicações, como por exemplo num esquema de banco de dados ou num sistemas de rede. Existem duas técnicas principais de correspondência de strings, uma é a correspondência exata e a outra é a correspondência aproximada.

Dada a natureza do problema relacionado a este projeto, foram implementados e analisados os algoritmos de comparação textual aproximados. Para efetuar os filtros de busca propostos, apesar da eficiência do operador LIKE, do operador de igualdade e demais operadores lógicos em consultas SQL, eles são limitados em bases de dados onde ocorreram erros de digitação ou em buscas fonéticas. Mesmo que os dados estejam consistentes e as informações cadastradas corretamente, a falha humana pode ocorrer no momento em que o usuário digita a informação que deseja buscar [27]. No momento em que um grande volume de dados é armazenado, a Lógica Fuzzy auxilia no reconhecimento de padrões para que estes dados se tornem informações úteis aos usuários [28].

A correspondência aproximada de strings é uma tarefa importante em muitas aplicações de processamento de dados incluindo correspondência estatística, pesquisa de texto, classificação de texto, verificação ortográfica e genômica. Podemos observar um exemplo em nosso cotidiano de aplicação deste tema ao buscar produtos em sites de compras online. O fator mais considerável na comparação aproximada de strings é a habilidade de quantificar a similaridade em dois textos, e a seguir nos aprofundaremos nos algoritmos utilizados no desenvolvimento

deste projeto.

### 2.2.1.1 DISTÂNCIA DE LEVENSHTTEIN

O algoritmo de Levenshtein[29] funciona calculando a distância de uma palavra a outra de forma que o grau de similaridade de dois documentos com base no peso da distância possa ser determinado. A seguir, um exemplo de pseudo código para calcular a distância de Levenshtein entre duas strings:

---

**Algoritmo 1** Cálculo da distância de Levenshtein entre duas strings

---

**Input:** String: str1, String: str2

**Output:** Int: distanciaLevenshtein

```

1: tabela ← Int[str1.tamanho, str2.tamanho]
2: X, Y, custo ← 0
3: for X de 0 até str1.tamanho − 1 do
4:   tabela[X, 0] ← X
5: for Y de 0 até str2.tamanho − 1 do
6:   tabela[0, Y] ← Y
7: for X de 0 até str1.tamanho − 1 do
8:   for Y de 0 até str2.tamanho − 1 do
9:     if str1[X] = str2[Y] then
10:       custo ← 0
11:     else
12:       custo ← 1
13:     tabela[X, Y] ← menor(
        tabela[X-1, Y] + 1, //Deleção
        tabela[X, Y-1] + 1, //Inserção
        tabela[X-1, Y-1] + custo //Substituição
    )
return tabela[str1.tamanho − 1, str2.tamanho − 1]

```

---

A comparação de sequências de caracteres é utilizada em diferentes áreas. Kruskal [30] dá uma visão geral sobre essas aplicações. Primeiramente, Kruskal menciona a aplicação à biologia molecular, onde a comparação de sequências é usada para a comparação de macromoléculas. Um exemplo mais relacionado à nossa pesquisa é a aplicação da comparação de sequências para contornar erros de digitação. A comparação de sequência também é usada para

reconhecimento de fala e locutor, para a comparação de arquivos de computador e para controle de erros de códigos que são transferidos por rádio ou telégrafo. Levenshtein apresentou a primeira definição de um algoritmo para cálculo de distância de strings que é apropriado para lidar com inserções e deleções de caracteres.

Na teoria da informação, existem duas métricas populares para avaliar strings em semelhança de forma, a distância de Hamming e a distância de Levenshtein. A distância de Hamming conta o número mínimo de substituições necessárias para editar uma string na outra. A distância de Levenshtein também leva em consideração inserções e deleções. Assim, a distância de Levenshtein produzirá distâncias menores ou iguais à distância de Hamming. Devido à essa propriedade, a distância de Levenshtein se comporta melhor com casos de abreviações do que a distância de Hamming.

#### 2.2.1.2 SOUNDEX

O termo “Soundex” cobre várias variações de um algoritmo desenvolvido pela primeira vez e patenteado por Russell e Odel em 1918[31]. A maioria das versões do Soundex converte uma palavra em um código que consiste na primeira letra da palavra, seguida por três (ou mais em alguns casos) dígitos. Os dígitos são atribuídos de acordo com um pré-determinado agrupamento de consoantes, onde os grupos consonantais compartilham características fonéticas (isto é, soar de forma semelhante). Este é o conceito-chave por trás do Soundex: uma constante relação entre letras e som deve assegurar que palavras com sons semelhantes sejam atribuídas com o mesmo código.

O algoritmo padrão do Soundex, para a fonética do inglês, define os seguintes grupos:

<b>Letra</b>	<b>Código</b>
B,F,P,V	1
C,G,J,K,Q,S,X,Z	2
D,T	3
L	4
M,N	5
R	6
H, Y, W	(omitido)
A,E, I, O, U	(omitido)

**Tabela 2: Código dos grupos fonéticos do soundex**

Nesse algoritmo canônico, a letra mais à esquerda é sempre mantida e todas as vogais não iniciais, bem como H, Y e W não iniciais, são omitidas. Apenas um dígito é usado para letras consecutivas que resultam no mesmo código (por exemplo, CK = 2, não 22); códigos com mais de três dígitos são truncados para os três dígitos mais à esquerda; e os códigos para palavras com menos de três consoantes são preenchidos com zeros (por exemplo, PEEL = P400). O grande número de caracteres na categoria 2 resultam de relações sobrepostas entre consoantes. A letra “C”, por exemplo, está relacionada a “S” (que por sua vez está relacionado a “Z”) e “K” (que está relacionado a “G” e “Q”).

O Soundex permite que palavras com pronúncias semelhantes, mas grafias distintas, sejam obtidos em uma única consulta. Por exemplo: PATER e PAIDER (Código Soundex P360); SOMERS , SUMMERS e SOMMARS (Código S562) e GARDNER, GARDINER e GARTNER (Código G635).

Variações do algoritmo original foram surgindo à medida que as limitações foram aparecendo. Essas variações introduzem técnicas como separar os grupos de consoantes em grupos menores e fortemente relacionados, aumentar o tamanho máximo do código permitindo mais de 3 dígitos e codificar as letras iniciais.

A Tabela 2 foi baseada na classificação dos fonemas da língua inglesa. Para adaptar o Soundex para o português brasileiro, fez-se necessário mudar o valor da tabela de códigos baseado na classificação fonética língua portuguesa, segundo a adaptação proposta por Ruberto e Antoniazzi [27]. A Tabela 3 apresenta a proposta para adaptação.

<b>Letra(s)</b>	<b>Código</b>
A, E, I, O, U, H, W, Y	0
P, B, M	1
F, V	2
T, D, N	3
L, R	4
S, Z	5
J, DI, GI, TI, CH, LH, NH	6
K, C, G, Q	7
X	8

**Tabela 3: Código dos grupos fonéticos do soundex-br**

O algoritmo implementado em PL/pgSQL está ilustrado no Apêndice A.

### 2.2.1.3 METAPHONE

O metaphone foi criada em 1990 por Lawrence Philips [32] como uma alternativa para resolver as deficiências do Soundex. Mais tarde, 10 anos depois, o autor lançou outra versão, chamada Double Metaphone [33]. É chamado de “Double” porque pode retornar um primário e um código secundário para uma string; isso é responsável por alguns casos ambíguos, bem como para múltiplas variantes de sobrenomes com ancestralidade comum. Double Metaphone tenta contabilizar para uma miríade de irregularidades em inglês derivadas de outras línguas como alemão, grego, francês, italiano, chinês e outras origens. Assim, ele usa um conjunto de regras muito mais complexo para codificação do que seu antecessor; por exemplo, ele codifica aproximadamente 100 contextos diferentes do uso da letra C sozinha.

Em 2009, Lawrence lançou a terceira versão do metaphone que atinge precisão de 99% das palavras em inglês. Esta série de algoritmos de metaphone são adequados para a maioria das palavras e esses algoritmos são a base para muitos verificadores ortográficos e dicionários [34].

Por se tratar de um algoritmo fonético, mesmo com as melhorias feitas por Lawrence nas versões subsequentes, os algoritmos da família Metaphone não conseguem agrupar eficientemente palavras de outros idiomas. Dessa forma, Jordão e Rosa [35], definiram uma adaptação do algoritmo metaphone para o português brasileiro, Metaphone-pt\_BR. A essência da definição publicada pelos autores estava em definir grupos ou contextos gramaticais que representem os fonemas da língua brasileira não compartilhados com o inglês. A implementação do Metaphone-pt\_BR foi disponibilizada pelos autores no SourceForge<sup>6</sup> sob a licença GPL e pode ser usada como uma extensão instalável em alguns SGBDs.

## 2.3 GEOPROCESSAMENTO

O tratamento e criação de informações geográficas por meio de tecnologias computacionais é uma parte essencial para o artefato proposto no presente trabalho e neste capítulo serão abordados os conceitos ligados à criação e armazenamento de coordenadas geográficas. Para armazenar coordenadas, mostrar localizações das empresas, ruas e endereços, conhecimento nesta área se faz necessário.

---

<sup>6</sup><http://sourceforge.net/projects/metaphoneptbr/>



### 2.3.1 DEFINIÇÕES DE GEOPROCESSAMENTO

Geoprocessamento corresponde a um conjunto de conhecimentos matemáticos e tecnologias computacionais para tratamento e criação de informações georreferenciadas, baseando-se nas áreas de cartografia, análise de recursos naturais, transporte, comunicações, energia e planejamento urbano [7, 36].

São inúmeras as definições na literatura e algumas delas comentadas por *Braga et al.* [37] caracterizam geoprocessamento como: uma disciplina que através de técnicas matemáticas e computacionais trata informações geográficas; um processamento digital de informações espaciais georreferenciadas; uma ferramenta que permite investigar de forma sistemática propriedades, relações e entidades representados em uma base de dados georreferenciada, transformando dados em informações destinadas à apoiar tomadas de decisões.

Para Rocha [7] o geoprocessamento precisa ser visto de forma mais ampla, como uma tecnologia transdisciplinar que integra softwares, equipamentos, processos, entidades, dados, metodologias e pessoas através da localização e processamento de dados geográficos. Rocha comenta que dentro do campo do geoprocessamento, para coletar, analisar, tratar e apresentar informações associadas à mapas digitais são utilizadas as ferramentas denominadas Sistemas de Informações Geográficas (SIGs).

### 2.3.2 APLICAÇÕES DO GEOPROCESSAMENTO

O geoprocessamento vêm sendo utilizado para fins comerciais, acadêmicos e governamentais sempre visando a integração de dados espaciais e não espaciais em seus projetos [36]. Alguns exemplos de suas aplicações são:

- **Manejo e conservação de recursos naturais:** Sob forma de estudos de impacto ambiental, modelagem de águas subterrâneas e caminhamento dos contaminantes, estudos de migrações e habitats das faunas, pesquisas de potencial mineral, etc. Ex: Geoprocessamento como ferramenta para a análise do uso da terra em bacias hidrográficas [38].
- **Gestão das explorações agrícolas:** Envolvendo a gestão de cultivo do campo, manejo de irrigação, avaliação do potencial agrícola, etc. Ex: Desenvolvimento de sistemas de geoprocessamento e tecnologia móvel aplicados à agricultura de precisão [39].
- **Planejamento de área urbana:** Abordando o planejamento dos transportes, desenvolvimento de planos de evacuação, localização de acidentes, seleção de itinerários de via-

gens. Ex: Abordagem sobre as técnicas de geoprocessamento aplicadas ao planejamento e gestão urbana[40].

- **Gestão das instalações:** Avaliando a localização de cabos e tubulações para o planejamento e manutenção das instalações. Ex: Utilização do Geoprocessamento na rede de comunicação de dados, voz e imagem da Prefeitura de Belo Horizonte[41].
- **Administração pública:** Servindo para gestão de cadastros, avaliação predial/territorial, gestão da qualidade das águas, planos de organização, etc. Ex: Acidentes nas rodovias brasileiras nos últimos 10 anos: uma análise com dados abertos[42].
- **Comércio:** Abordando análises de estruturas de mercado, planejamento de desenvolvimento, análises de concorrência e das tendências de mercado entre outros. Ex: Uso de geoprocessamento no mercado imobiliário[43].
- **Saúde pública:** Sob forma de epidemiologias, análise da distribuição e evolução de doenças, distribuição dos serviços sociais sanitários, planos de emergência, etc. Ex: Situação atual do geoprocessamento e da análise de dados espaciais em saúde no Brasil [44].

Dado as características do presente trabalho, compreende-se um alinhamento ao conceito de planejamento de administração pública.

### 2.3.3 GEOCODIFICAÇÃO

A geocodificação, de forma breve, permite a conversão de endereços escritos em coordenadas geográficas [45]. Por endereços, entende-se aqueles que são utilizados para descrever localizações físicas de forma comumente interpretada pelas pessoas, como um endereço residencial com nome de uma rua, número, e até mesmo cidade, estado e país. Além disso, tem-se a geocodificação reversa, que trata do processo contrário: a conversão de coordenadas para endereços escritos por extenso.

A geocodificação está presente em diversas aplicações cotidianas, como em aplicativos web e mobile quando em algum momento precisam receber um endereço por extenso ou posição de lugares ou pessoas no mapa, e por vezes esse tipo de tecnologia pode passar despercebido[46]. Para o presente trabalho, tanto a geocodificação quanto a geocodificação reversa são instrumentos esperados como parte do desenvolvimento de uma solução computacional, uma vez que será necessário converter posições físicas de entrada de usuários e tomar decisões com base em diferentes entradas.

Como a área de geocodificação possui uma grande complexidade e é extensa por si só, para o escopo aqui definido serão estudadas alternativas já presentes na comunidade. De maneira geral, podemos analisar a utilização da geocodificação de duas maneiras: como um serviço que pode ser requerido externamente, ou como um pedaço de software que pode ser incorporado nos próprios servidores da aplicação em que o tipo de tecnologia for necessária.

### 2.3.3.1 SOFTWARES DE GEOCODIFICAÇÃO

Existem alguns serviços de geocodificação já disponíveis via web, e em sua maioria funcionam através de aplicações REST (*Representational State Transfer*). Aqui serão listadas soluções que podem ser incorporadas na própria aplicação:

- **Google Maps Geocoding API<sup>7</sup>**: É o geocodificador padrão utilizado pelos serviços do Google Maps. Seu código é fechado e conta com uma API própria para lidar com requisições REST, retornando as coordenadas ou endereços em formato JSON. Não possui limite diário, porém tem um limite de 50 requisições por segundo. Ao extrapolar este limite cobra-se US\$5,00 para 1000 solicitações adicionais.
- **Nominatim<sup>8</sup>**: É um dos geocodificadores inclusos na interface web padrão do *Open Street Map*. É open source e possui uma API REST porém não apresentou boa assertividade nos testes realizados.
- **Bing Maps Locations API<sup>9</sup>**: Semelhante à API do Google, a Bing Maps Locations API é o geocodificador do Bing Maps. É necessário requisitar uma *api key* para utilizar o serviço, porém a Bing oferece uma chave gratuita para projetos educacionais com limitação de 50.000 requisições por dia.
- **Mapquest<sup>10</sup>**: Trata-se de uma API que conta com diversas funções, incluindo geocodificação, cálculo de rotas e exibição de mapas. Também utiliza o Open Street Maps como base e em alguns testes rápidos mostraram problemas para a região de Curitiba, como retornar pontos muito distantes do local real ou até não fornecer um resultado.

Dado os testes realizados, a geocodificação do Google Maps se mostrou a mais adequada pois conseguiu uma precisão razoável para os endereços testados, além de compreender (fazer o *parsing*) sem nenhum problema algumas inversões na ordem de escrita dos

<sup>7</sup><https://developers.google.com/maps/documentation/geocoding/overview>

<sup>8</sup><https://nominatim.org/>

<sup>9</sup><https://docs.microsoft.com/en-us/bingmaps/rest-services/locations/>

<sup>10</sup><https://www.mapquest.com/>

endereços, porém devido à característica da aplicação, por ser uma página web pública e a Google Maps Geocoding API ser um serviço *pay as you go* que cobra uma taxa por cada requisição feita ao serviço, foi optado por utilizar a API Bing Maps Locations, que também demonstrou uma boa assertividade nos testes e é gratuita para projetos educacionais sem fins lucrativos.

## 2.4 APPLICATION PROGRAMMING INTERFACES

*Application Programming Interfaces* (API) são interfaces que permitem que um componente de software acesse outro componente ou informação presente em outro software [4]. Essas interfaces consistem em um conjunto de normas que possibilita a comunicação entre plataformas através de uma série de padrões e protocolos.

As APIs estão presentes na maioria dos aplicativos do cotidiano. No WhatsApp<sup>11</sup>, por exemplo, podemos perceber a integração da lista de contatos salva no dispositivo com os contatos do aplicativo. A comodidade que o uso de APIs resulta popularizou o conceito, causando o surgimento de interfaces para diversas necessidades, podendo-se citar as que permitem a manipulação de mapas e outros componentes gráficos, possibilitando a criação de aplicações de mobilidade urbana.

A Google Maps JavaScript API é uma biblioteca da Google<sup>12</sup> que permite adicionar mapas em aplicações *web* e em dispositivos móveis. Entre as opções disponíveis estão mapas de dados fotográficos de satélite, mapas vetoriais, bem como o mapa híbrido, que agrega características de mapas de ambos os tipos. Google Maps JavaScript API também é responsável por possibilitar a adição de marcadores (do inglês, *markers*), que são responsáveis por indicar localizações únicas no mapa. É possível também fazer uso de uma funcionalidade da API responsável por agrupar marcadores próximos gerando um *cluster*, facilitando a navegação no mapa para encontrar marcadores específicos. O uso dessa API do Google Maps possui um pequeno custo por utilização de 7.00 dólares a cada 1000 mapas gerados, porém há uma cota mensal gratuita de \$200 possibilitando a geração de mais de 28.000 mapas sem custo.

Outra API utilizada no projeto foi a Bing Maps Locations API<sup>13</sup>, conforme comentado na seção 2.3.3.1 é uma interface de geocodificação que possui licenças gratuitas para softwares educacionais sem fins lucrativos e demonstrou boa assertividade na geocodificação dos endereços testados.

---

<sup>11</sup><https://www.whatsapp.com/>

<sup>12</sup><https://developers.google.com/maps/documentation/javascript/overview>

<sup>13</sup><https://docs.microsoft.com/en-us/bingmaps/rest-services/locations/>

## 2.5 BUSKAKI EMPRESAS

O projeto foi desenvolvido utilizando a linguagem de programação Ruby [47] utilizando a IDE da JetBrains [48], RubyMine [49]. Para desenvolvimento foi utilizado o framework Rails [50] e as gems *Geocoder* [51], *Net SSH Gateway* [52] e *GMaps4Rails* [53] para geocodificação de endereços, tunelamento SSH e criação de mapas dinâmicos com a API do google maps.

O Ruby on Rails[50] foi escolhido para desenvolver o projeto devida à sua facilidade e rapidez para implementação de sistemas web. Além disso, a gem *Geocoder*[51] permite que a geocodificação seja parcialmente independente do serviço utilizado. Por fim, a gem *Gmaps4Rails*[53] facilita a criação de um mapa dinâmico para interação com o usuário, utilizando a API do Google Maps.

### 2.5.1 APLICAÇÕES SEMELHANTES

O processo de coleta de informações de CNPJ é também relevante para algumas áreas de pesquisa, pois a existência de alvarás pode ser diretamente relacionada a uma representação de conceitos primordiais associados a processos de crescimento urbano [54]. Estudar o desenvolvimento de uma área comercial oferece possíveis indicadores que ajudariam a identificar o impacto econômico e social em seu contexto local [55]. Existem algumas ferramentas para consulta de dados de pessoa jurídica, desde as mais simples e gratuitas às ferramentas pagas, para uma consulta com maior quantidade de informações das empresas e seus sócios. As ferramentas gratuitas podem ser separadas em dois grupos: sites e aplicativos. Dentre os sites, destacam-se o Portal REDESIM <sup>14</sup> e o site da Junta Comercial do Estado de São Paulo <sup>15</sup> (JUCESP), cujas interfaces de busca são ilustradas na Figura 2, enquanto os aplicativos que se destacam são o CNPJ<sup>16</sup>, disponibilizado pela Receita Federal, e o Consulta CNPJ <sup>17</sup>, apresentados na Figura 3.

<sup>14</sup><https://consultacnpj.redesim.gov.br/>

<sup>15</sup><https://www.jucesponline.sp.gov.br/pesquisa.aspx?IDProduto=7>

<sup>16</sup><https://play.google.com/store/apps/details?id=br.gov.fazenda.pessoajuridica>

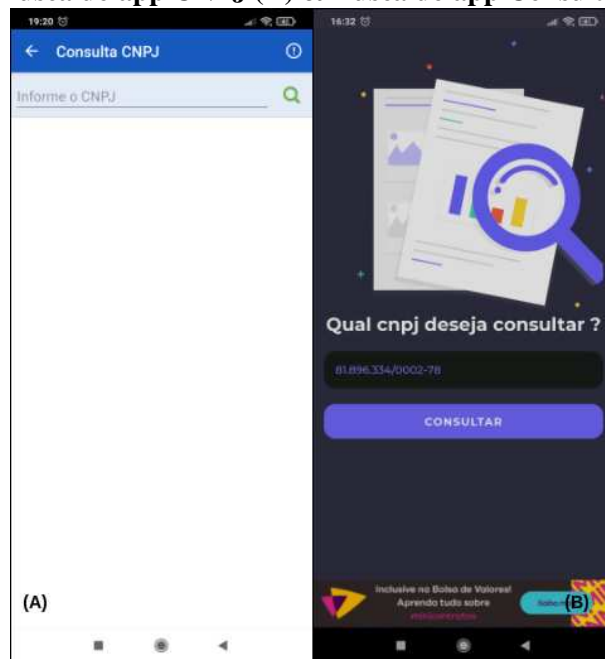
<sup>17</sup><https://play.google.com/store/apps/details?id=com.consultacnpj>

**Figura 2: Interface de busca da JUCESP (A) & Interface de busca do Portal REDESIM (B)**



Fonte: Autoria própria

**Figura 3: Busca do app CNPJ (A) & Busca do app Consultar CNPJ (B)**



Fonte: Autoria própria

Existem também algumas ferramentas pagas para realização desta consulta, como a API do Serpro e a consulta do Serasa Experian, ambas cobram por cada consulta realizada. Dentre as ferramentas gratuitas citadas anteriormente, apenas duas possibilitam que a consulta seja feita de outra forma além do CNPJ: o Portal REDESIM<sup>18</sup> e o site da JUCESP<sup>19</sup> suportam

<sup>18</sup><https://consultacnpj.redesim.gov.br/>

<sup>19</sup><http://www.institucional.jucesp.sp.gov.br/>

consultas pelo nome da empresa. Para a utilização da ferramenta do Portal é necessário um certo nível de conhecimento técnico: é necessário criar um cadastro, lidar com captchas, saber diferenciar Razão Social de Nome Fantasia e o único filtro de localização é por unidade da federação. Quanto a consulta da JUCESP, apesar de possuir registros de empresas de todo país, seu maior acervo são das empresas do estado de São Paulo, não contemplando a maioria das empresas de outros estados. Nenhuma dessas ferramentas aplica um algoritmo robusto para realizar a pesquisa, sendo capaz apenas de retornar os registros com um match do texto inserido.

Os grandes diferenciais da ferramenta proposta neste trabalho são a oferta de opções de buscas por outros campos além do CNPJ, a utilização de algoritmos para aumentar a abrangência da busca e a abordagem visual, plotando a localização dos resultados em mapas dinâmicos. Esses diferenciais foram pensados para melhorar usabilidade do sistema e a experiência do usuário de qualquer nível de conhecimento técnico.

### 3 METODOLOGIA

Este capítulo apresenta o método proposto para obter os objetivos apresentados na Seção 1.1. Este método é dividido em quatro etapas, ilustradas na Figura 4, que serão descritas a seguir.



**Fonte: Autoria própria**

#### 3.1 REVISÕES BIBLIOGRÁFICAS

Na etapa de revisões bibliográficas ocorreu um estudo sobre os principais conceitos que englobam o projeto Buskaki Empresas: APIs, banco de dados, cidades inteligentes e geoprocessamento. Para este estudo, foram analisados artigos científicos e trabalhos que abordassem de alguma maneira os conceitos citados, juntamente com os dados abertos da Receita Federal e da prefeitura de Curitiba relacionados às empresas do município. O resultado de todo estudo realizado se encontra no Capítulo 2.

#### 3.2 ANÁLISE E MODELAGEM DO SISTEMA

Durante a análise do problema, foram elencadas e avaliadas as possíveis soluções propostas.

Na modelagem, os resultados obtidos anteriormente foram usados como base para projetar o funcionamento do sistema, sendo dividida nas seguintes partes:

- Modelagem da base de dados;



- Modelagem da aplicação;
- Protótipo inicial das telas.

### 3.3 DESENVOLVIMENTO E AVALIAÇÃO

Após o idealização do sistema necessário para a criação do protótipo, iniciou-se a etapa de desenvolvimento, composta das seguintes etapas:

- Integração com a base de dados e criação das tabelas da aplicação;
- Desenvolvimento dos algoritmos e scripts *Structured Query Language* (SQL) para realizar as buscas de empresas;
- Desenvolvimento da interface da página web;
- Integração com a API do Google Maps;
- Implementação do algoritmo SoundexBr;
- Integração com a biblioteca PostgreSQL desenvolvida por JORDÃO e ROSA (2012) [35];
- Análise dos algoritmos de comparação textual.

### 3.4 CONCLUSÃO

Na etapa de conclusão ocorreu a análise dos resultados após o desenvolvimento. Também foram apontados possíveis trabalhos futuros, que poderão ser realizados como forma de aprimoramento e extensão da aplicação.

## 4 MODELAGEM DO SISTEMA

Neste capítulo serão apresentados os diagramas e modelos desenvolvidos para entender e modelar o sistema. Inicialmente, serão apresentados os diagramas de caso de uso e a definição dos casos de teste. Na sequência será abordado o diagrama do esquema de dados e o protótipo das telas da aplicação

### 4.1 DIAGRAMAÇÃO

Para auxiliar o desenvolvimento do protótipo, foram gerados diagramas de caso de uso e do esquema dos dados, que serão apresentados nesta seção.

#### 4.1.1 CASOS DE USO

A Figura 5 ilustra os casos de uso da aplicação, sendo que os principais casos estão descritos a seguir. O restante dos casos de uso está presente no Apêndice B.

##### 4.1.1.1 BUSCA AVANÇADA

**Descrição:** O usuário tenta realizar uma busca por determinada empresa, inserindo até quatro parâmetros: nome, rua, bairro e cep.

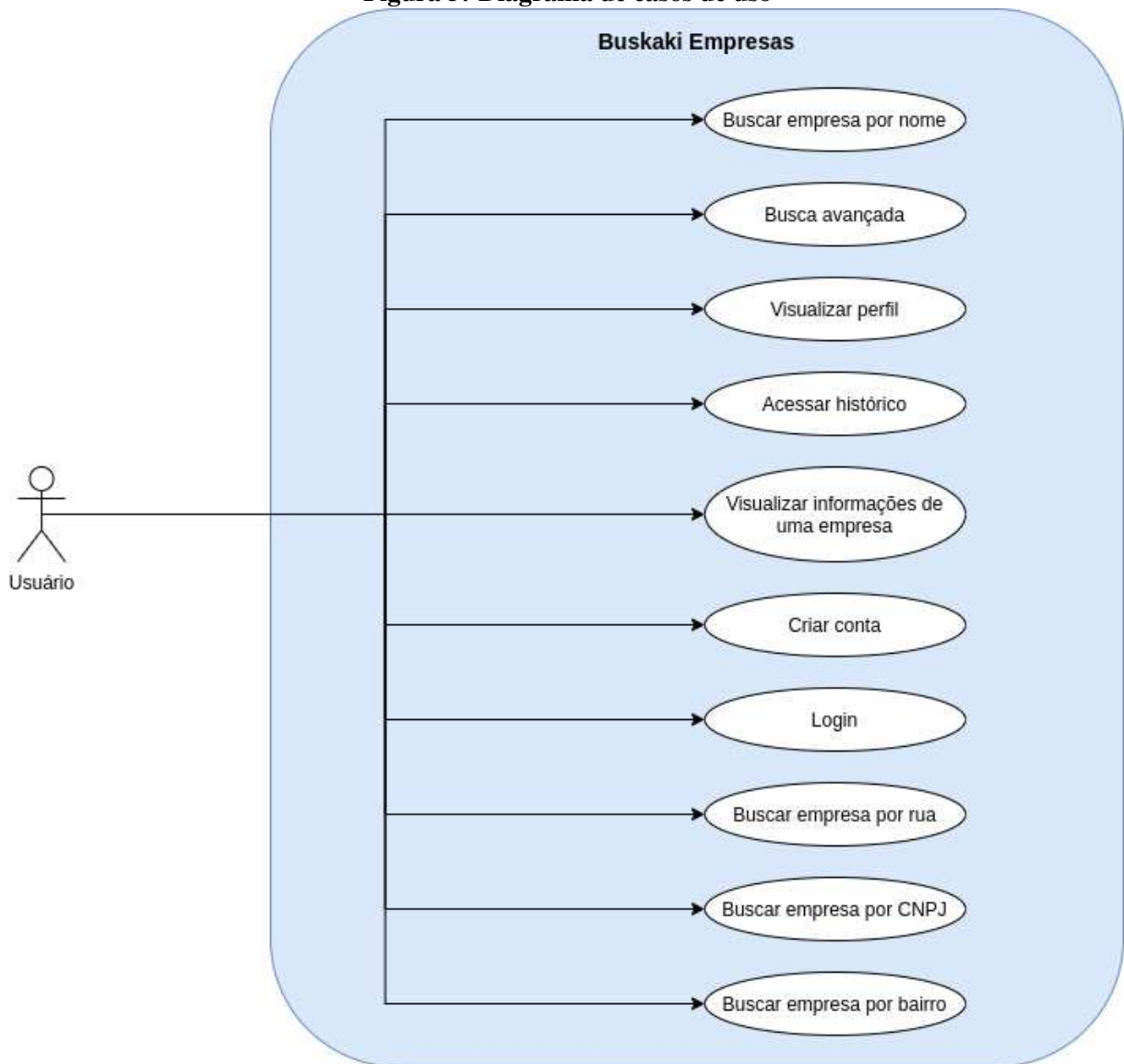
**Pré-Condições:** O usuário deve estar logado.

**Fluxo Básico:** O usuário preenche os campos do formulário de busca e envia a requisição ao servidor. Então ele é redirecionado para a tela com o resultado da busca.

**Fluxo Alternativo:** Nenhum.

**Pós-Condições:** Mostrar ao usuário uma lista paginada com os resultados adequados aos parâmetros informados, assim como a posição geográfica dos resultados desta página num mapa dinâmico.

**Figura 5: Diagrama de casos de uso**



**Fonte: Autoria própria**

#### 4.1.1.2 VISUALIZAR INFORMAÇÕES DE UMA EMPRESA

**Descrição:** O usuário tenta acessar mais informações de uma das empresas resultantes da busca.

**Pré-Condições:** O usuário deve estar logado e ter realizado uma busca.

**Fluxo Básico:** O usuário fez uma busca e foi redirecionado para a tela com os resultados da busca, então ele clica no botão “Exibir” de uma das empresas resultantes e é redirecionado para a tela com informações daquela empresa.

**Fluxo Alternativo:** Nenhum.

**Pós-Condições:** Mostrar ao usuário uma tela com os dados abertos disponíveis da empresa

selecionada, assim como um mapa com a localização da mesma.

#### 4.1.1.3 ACESSAR HISTÓRICO

**Descrição:** O usuário tenta acessar seu histórico de buscas.

**Pré-Condições:** O usuário deve estar logado e ter realizado alguma busca.

**Fluxo Básico:** O usuário clica no menu “Histórico”.

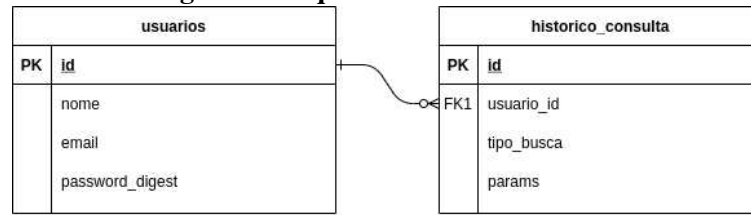
**Fluxo Alternativo:** Nenhum.

**Pós-Condições:** Mostrar ao usuário uma lista paginada com o seu histórico de buscas, ordenado da mais recente à mais antiga.

#### 4.1.2 ESQUEMA DOS DADOS

Para auxiliar o desenvolvimento da aplicação foi criado um Diagrama que representa como estão dispostas as tabelas no banco de dados, representado na Figura 6, onde estão contidos todos os dados e inter-relacionamentos necessários para o funcionamento do protótipo. As descrições das tabelas, juntamente com as consultas de criação, se encontram no Apêndice C.

**Figura 6: Esquema do banco de dados**



alvaras	
PK	id
	nome_empresa
	data_inicio_atividade
	numero_do_alvara
	nome_da_empresa
	data_emissao
	data_expiracao
	atividade_principal
	atividade_secundaria1
	atividade_secundaria2
	endereco
	numero
	unidade
	andar
	complemento
	bairro
	cep
	cidade
	estado
	latitude
	longitude
	localidade
	atividade_principal_agregada
	tipo_modificado

empresas_receita_curitiba	
PK	idempresas
	cnnpj
	identificadormatrizfilial
	nome_empresa
	nome_fantasia
	situacao_cadastral
	data_situacao_cadastral
	motivo_situacao_cadastral
	nome_cidade_exterior
	codigo_pais
	nome_pais
	codigo_natureza_juridica
	data_inicio_atividade
	cnae_fiscal
	descricao_tipo_logradouro
	logradouro
	numero
	complemento
	bairro
	cep
	uf
	codigo_municipal
	municipio
	ddd_telefone
	telefone
	email_contribuinte
	qualificacao_responsavel
	capital_social
	porte_empresa
	opcao_simples
	data_opcao_simples
	data_exclusao_simples
	opcao_mei
	situacao_especial
	data_situacao_especial
	geom
	latitude
	longitude

A tabela *empresas\_receita\_curitiba* possui 585.624 registros de empresas curitibanas, fazendo uso desses dados podemos obter algumas informações interessantes sobre a cidade, como por exemplo: a Tabela 4 que apresenta as empresas em maior número e a Tabela 5 com os bairros com mais empresas registradas.

<b>Empresas em maior número</b>	
BANCO BRADESCO S.A.	126
CAIXA ECONOMICA FEDERAL	123
FARMACIA E DROGARIA NISSEI S.A.	116
MITRA DA ARQUIDIOCESE DE CURITIBA	114
IGREJA DO EVANGELHO QUADRANGULAR	108

**Tabela 4: Empresas curitibanas em maior número**

<b>Bairros com mais empresas</b>	
CENTRO	81508
BOQUEIRAO	26506
CIDADE INDUSTRIAL	22456
AGUA VERDE	20510
SITIO CERCADO	19252

**Tabela 5: Bairros de Curitiba com mais empresas**

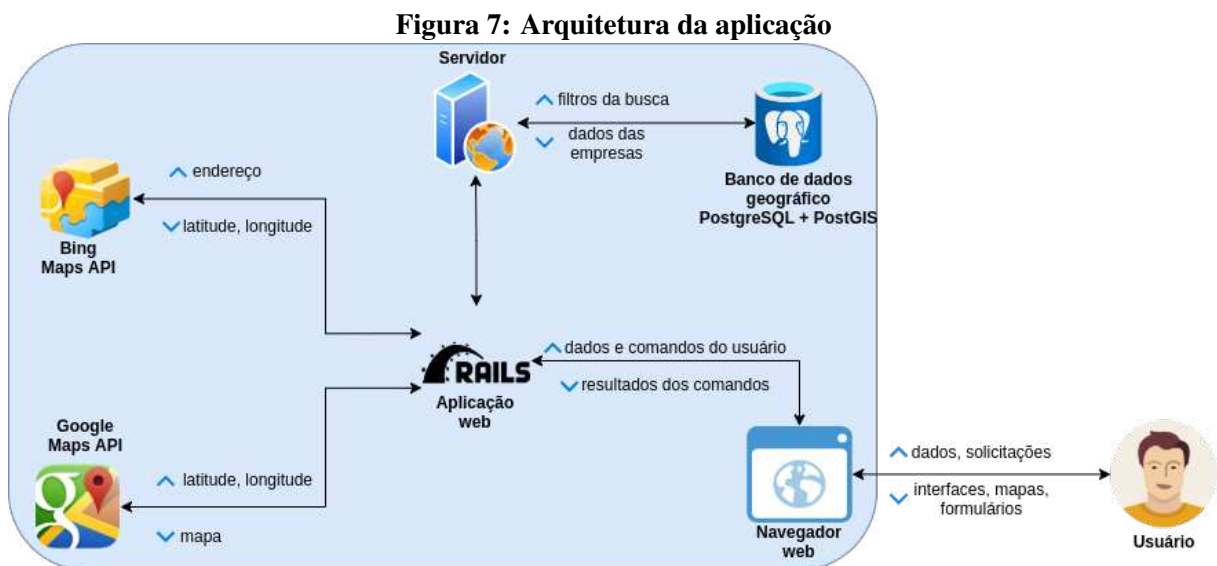
A partir das tabelas acima, podemos extrair informações que correspondem às expectativas: as empresas em maior quantidade na cidade, considerando filiais, agências, sucursais ou qualquer entidade registrada sob o mesmo nome comercial, são bancos, farmácias e igrejas e o centro da cidade é o bairro com maior quantidade de empresas registradas.

## 5 DESENVOLVIMENTO

Esta seção explicará o processo de desenvolvimento do protótipo, as tecnologias usadas e abordagens utilizadas para implementar o sistema.

### 5.1 ARQUITETURA DA APLICAÇÃO

A Figura 7 apresenta a arquitetura da aplicação. Utilizando um dispositivo móvel ou computador, o usuário realiza uma interação com a página web, que se conecta ao servidor e retorna os dados solicitados, vindos do banco de dados.



Fonte: Autoria própria

### 5.2 TECNOLOGIAS UTILIZADAS

Do lado do servidor da aplicação foi utilizado o web server Puma versão 4.3.7. O software utiliza 10 tecnologias diferentes:

- Cascading Style Sheets (CSS) e HyperText Markup Language (HTML): Para a exibição da página web em si, dos formulários e de elementos textuais.
- JavaScript: para a comunicação com o Google Maps<sup>1</sup> e envio dos objetos JSON (JavaScript Object Notation) para o servidor do Google Maps API e exibição dos mapas no HTML.
- Linguagem de programação Ruby<sup>2</sup> versão 2.2.3 .
- Framework da linguagem de programação Ruby, Ruby on Rails<sup>3</sup> 5.0.0.1, que orienta o desenvolvimento no padrão Model View Controller criando um web service RESTful para a aplicação, gerando as páginas de HTML de forma automática e atuando como ObjectRelational-Mapper (Mapeador objeto-relacional, facilitando a manipulação do banco de dados). Esta tecnologia realiza a comunicação intermediária entre o usuário e: banco de dados PostgreSQL e PostGIS, APIs que auxiliam as envio de dados do SGBD para o JavaScript, autenticação, menus e validação dos formulários.
- PostgreSQL<sup>4</sup> versão 9.6.13 x64: que se trata de um SGBD objeto-relacional *open source* e é utilizado para o armazenamento dos dados das empresas, dos usuários e do histórico de consultas.
- PostGIS<sup>5</sup> versão 2.3.1: extensão do PostgreSQL que o transforma num banco de dados geográfico.
- Ruby on Rails GMaps4Rails<sup>6</sup> gem com Google Maps Javascript API: para exibição dos mapas vindos do servidor do Google Maps, com funcionalidades de clusterização, utilizados para facilitar o reconhecimento da empresa na busca.
- Ruby on Rails Geocoder<sup>7</sup> Gem com Bing Geocoding API: para transformação de um endereço ou CEP em uma coordenada geográfica expressa em latitude e longitude no sistema de coordenadas Web Mercator (WGS 84 Web Mercator ou WGS 84/PseudoMercator) (BATTERSBY et al., 2014).
- Ruby on Rails ActiveRecord PostGIS Adapter<sup>8</sup> gem: para permitir que o ORM (ObjectRelational-Mapper) do Ruby on Rails consiga lidar com objetos espaciais.

---

<sup>1</sup><https://developers.google.com/maps?hl=pt-br>

<sup>2</sup><https://www.ruby-lang.org/pt/>

<sup>3</sup><https://rubyonrails.org/>

<sup>4</sup><https://www.postgresql.org/>

<sup>5</sup><https://postgis.net/>

<sup>6</sup><https://rubygems.org/gems/gmaps4rails/versions/2.1.2?locale=pt-BR>

<sup>7</sup><https://rubygems.org/gems/geocoder>

<sup>8</sup><https://rubygems.org/gems/activerecord-postgis-adapter>



- Ruby on Rails Net SSH Gateway<sup>9</sup> gem: para criar o tunel ssh entre a aplicação e o banco de dados, necessário pela política de segurança do banco.

A aplicação utilizará a base de dados do Centro de Computação Científica e Software Livre (C3SL)<sup>10</sup>. A utilização do servidor do C3SL deu-se devido à pré-disponibilidade dos dados de empresas na base e à altíssima capacidade de processamento do servidor, que tem a seguinte configuração: AMD EPYC 7401 24-Core 48-threads x64 2.8GHz, utilizando como sistema gerenciador de banco dados o PostgreSQL v9.6.13 com extensão PostGIS v2.3.

Devido à necessidade de disponibilizar o protótipo para testes com usuários, o sistema foi hospedado numa plataforma nuvem, Heroku<sup>11</sup>, dessa forma a aplicação fica disponível online podendo ser acessada de qualquer dispositivo com internet. A aplicação foi hospedada num container de categoria *free*, que possui 512MB de RAM dedicada.

### 5.3 OBTENÇÃO DOS DADOS

Através do site da Receita Federal<sup>12</sup>, é possível fazer o download dos arquivos que contêm os dados abertos das empresas do país, atualizados semestralmente. Esses arquivos estão formatados de acordo com um layout específico disponibilizado no site e com isso foi feita a importação desses dados para um tabela na base de dados. A tabela utilizada no desenvolvimento do protótipo já existia no banco de dados do C3SL antes do início deste trabalho de conclusão de curso devido ao trabalho de Bichibichi et al [55].

### 5.4 DESCRIÇÃO DO DESENVOLVIMENTO DA ESTRUTURA DO PROJETO

A aplicação, desenvolvida com as tecnologias apresentadas na seção 5.2, foi implementada com a arquitetura *Model-View-Controller* (MVC). A arquitetura MVC é um padrão de projeto de software formulado na década de 70 focado no reuso de código e na separação de responsabilidades em três camadas: camada de modelo, camada de controle e camada de visualização.

A camada de modelo consiste na parte lógica da aplicação, que gerencia o comportamento dos dados através de regras de negócios, lógica e funções. É a camada que permite

<sup>9</sup><https://rubygems.org/gems/net-ssh-gateway>

<sup>10</sup><https://www.c3sl.ufpr.br/>

<sup>11</sup><https://www.heroku.com/>

<sup>12</sup><https://receita.economia.gov.br/orientacao/tributaria/cadastros/cadastro-nacional-de-pessoas-juridicas-cnpj/dados-publicos-cnpj>

o acesso para os dados do banco serem coletados, gravados e exibidos. Neste projeto foi criado um modelo correspondente a cada tabela do banco, concentrando a lógica de negócio, como a construção da *query* de busca ou a formatação do endereço da empresa utilizado na geocodificação. A Figura 8 contém um trecho do *model* EmpresaReceitaCuritiba.

**Figura 8: Trecho do *model* EmpresaReceitaCuritiba**

```
def self.filtrar_busca_avancada(nome, bairro, rua, cep)
  nome = ApplicationController.helpers.remove_acentos(nome)
  bairro = ApplicationController.helpers.remove_acentos(bairro)
  rua = ApplicationController.helpers.remove_acentos(rua)
  query = ''
  query += "(nomeempresarial LIKE '%#{nome}%' OR nomefantasia LIKE '%#{nome}%')" unless nome.empty?
  query += "#{query.empty? ? '' : 'AND'} bairro = '#{bairro}'" unless bairro.empty?
  query += "#{query.empty? ? '' : 'AND'} logradouro LIKE '%#{rua}%' " unless rua.empty?
  query += "#{query.empty? ? '' : 'AND'} cep = '#{cep}'" unless cep.empty?
  query += "#{query.empty? ? '' : 'AND'} situacaocadastral != '08'"

  EmpresasReceitaCuritiba.where(query)
end

def endereco_completo_google
  "BRASIL, #{municipio}, #{descricaootipologradouro} #{logradouro} #{numero}, #{bairro}"
end
```

**Fonte: Autoria própria**

A camada de visualização representa a saída dos dados. Numa aplicação web, normalmente uma *view* representa uma página web, pois é nela que está implementado o código que constrói a página visualizada pelo usuário final. As *views* do Buskaki Empresas possuem código HTML, CSS e JavaScript, necessários para construir, estilizar e manipular a visualização disponibilizada. A Figura 9 mostra parte do código da *view* dos resultados de uma busca por nome: a parte superior é responsável por construir o corpo da tabela de resultados, enquanto o script na parte inferior serve para colocar os marcadores da localização das empresas no mapa. A Figura 10 é o resultado de uma busca, renderizada pelo código da *view*.

Figura 9: Trecho da *view* resultado\_busca\_por\_nome

```

<tr>
  <td><%= empresas_receita_curitiba.cnpj %></td>
  <td><%= empresas_receita_curitiba.nomeempresarial %></td>
  <td><%= empresas_receita_curitiba.nomefantasia %></td>
  <td><%= formatar_endereco(empresas_receita_curitiba) %></td>
  <td><%= empresas_receita_curitiba.bairro %></td>
  <td><%= empresas_receita_curitiba.cep %></td>
  <td><%= link_to "Exibir", get_by_cnpj_path(:cnpj => empresas_receita_curitiba.cnpj) %></td>
</tr>
<% end %>
</tbody>
</table>
<%= pagy_bootstrap_nav(@pagy) %>

<script type="text/javascript">
  handler = Gmaps.build('Google', {markers:
    {clusterer: { gridSize: 60, maxZoom: 20}}});
  handler.buildMap({ provider: {maxZoom: 20}, internal: {id: 'map'}}, function(){
    let markers_json_list = [];
    <% @empresas_receita_curitibas.each do |empresa| %>
      markers_json_list.push({
        "lat": <%= empresa.lat %>,
        "lng": <%= empresa.lng %>,
        "picture": {
          "width": 32,
          "height": 32
        },
        "infowindow": "<%= empresa.cnpj %> - <%= empresa.nomefantasia %>"
      })
    <% end %>
    markers = handler.addMarkers(markers_json_list);
    handler.bounds.extendWith(markers);
    handler.fitMapToBounds();
  });
</script>

```


Fonte: Autoria própria

**Figura 10: Tela de resultado da busca por nome**

**Buskaki Empresas**

Log out

[Buscar por nome](#)  
 [Buscar por CNPJ](#)  
 [Listar empresas por bairro](#)  
 [Listar empresas por rua](#)  
 [Meu perfil](#)  
 [Histórico](#)



< Anterior 1 Próximo >

Cnpj	Nome Empresarial	Nome Fantasia	Endereço	Bairro	Cep	
75111823000193	SUPERMERCADO CARNAVALE LTDA	TOP SUPERMERCADO	RUA VIRGINIA DALLABONA 101	NOVA ORLEANS	82310390	Exibir
81896334000278	CLARION COMERCIO DE GAS LTDA	TOP SUPER	RUA GASTAO LUIZ CRULS 351	BAIRRO ALTO	82840180	Exibir

**Fonte: Autoria própria**

Todas as telas da aplicação podem ser consultadas no Apêndice D.

A camada de controle é responsável por fazer a mediação da entrada e saída, comandando a *view* e o *model* para serem alterados de forma apropriada conforme o usuário solicitou através do mouse e teclado. Neste projeto foi criado um *controller* para cada *model*, responsável por salvar o histórico de buscas, aplicar a lógica de validação e normalização dos dados de entrada, e também utilizar a API de geocodificação para buscar as coordenadas geográficas dos resultados da busca que não possuem estes dados na base. A Figura 11 ilustra um trecho do código do *controller* EmpresaReceitaCuritibaController.

**Figura 11: Trecho do controller EmpresaReceitaCuritibaController**

```

def georreferenciar_empresa_api_bing(empresa)
  results = Geocoder.search(empresa.endereco_completo_bing)
  endereco_georreferenciado = results.first&.data
  if results.length > 1
    filtrar_endereco_por_cep = lambda { |result| empresa.cep.include? result.data["address"]["postalCode"].to_s }
    endereco_georreferenciado = results.select(&filtrar_endereco_por_cep).first&.data
    if endereco_georreferenciado.nil?
      filtrar_endereco_por_numero = lambda { |result| result.data["name"].include? empresa.numero }
      endereco_georreferenciado = results.select(&filtrar_endereco_por_numero).first&.data ? results.select(&filtrar_endereco_por_numero).first&.data : results.first.data
    end
  end
  unless results.blank?
    lat, lng = endereco_georreferenciado["point"]["coordinates"]
    empresa.lat = lat
    empresa.lng = lng
  end
end

def salvar_historico_busca
  search_params = params.select {|k,v| [%w[cep], 'nome', 'endereco', 'descricao', 'tipo', 'logradouro', 'logradouro', 'numero', 'complemento', 'bairro', 'cep', 'endereco', 'nome'].include?(k)}
  tipos_buscas = {
    'filtrar_por_nome' => 'Por nome',
    'filtrar_por_bairro' => 'Por bairro',
    'filtrar_por_rua' => 'Por rua',
    'get_by_cep' => 'Por CEP',
    'busca_avancada' => 'Busca avançada'
  }
  current_user.historico_buscas.create(:params => search_params, :tipo_busca => tipos_buscas[params[:action]])
end

```

**Fonte: Autoria própria**

O intuito de ter um mapa dinâmico na aplicação, como o ilustrado na Figura 10, é para auxiliar o usuário a distinguir a empresa desejada entre os resultados da busca.

O processamento da geocodificação das empresas é feito através da *gem* Geocoder. Esta *gem* possui diversas funcionalidades de geocodificação direta ou inversa e integração com 17 APIs de serviços de geolocalização global<sup>13</sup>. Entre os serviços disponíveis, foi escolhido o Bing por ser um serviço gratuito, com suporte a uma licença para projetos estudantis e por ter demonstrado boa precisão nos testes realizados.

O código fonte do protótipo está disponível no GitHub<sup>14</sup> e também foi gravado um tutorial da aplicação, disponível no Youtube<sup>15</sup>.

## 5.5 DESCRIÇÃO DO DESENVOLVIMENTO DOS ALGORITMOS E CONSULTAS SQL

Nesta subseção objetiva-se demonstrar as formas com que as ferramentas mencionadas anteriormente foram utilizadas para produzir o protótipo. A explicação segue uma ordem cronológica do seu uso, desde o cadastro até a visualização dos dados.

Todos os usuários precisam realizar um cadastro para acessar as funcionalidades de busca, para que os dados de histórico de consultas fiquem salvos em suas respectivas contas.

<sup>13</sup><https://github.com/alexreisner/geocoder>

<sup>14</sup>[https://github.com/kakivis/buskaki\\_empresas\\_cwb\\_public](https://github.com/kakivis/buskaki_empresas_cwb_public)

<sup>15</sup><https://youtu.be/mOdAQpmZZtM>

Tendo uma conta e autenticando-a dentro do sistema, a primeira ação esperada do usuário é realizar algum tipo de consulta. Ao informar os parâmetros da busca e submeter o formulário, a seguinte lógica é executada no servidor:

---

**Algoritmo 2** Busca por nome da empresa

---

**Input:** nome da empresa a ser pesquisada

**Output:** lista de empresas compatíveis

```

1: salvar_historico_buscas(nome)
2: nome ← remover_acentos(nome)
3: nome ← para_letra_maiuscula(nome)
4: lista_empresas ← executa_query_busca(nome)
5: for empresa em lista_empresas do
6:   if empresa.latitude é nulo OU empresa.longitude é nulo then
7:     endereco_empresa ← formatar_endereco_para_api_endereços(empresa)
8:     resultados_geocoder ← consulta_api_geocodificacao(endereco_empresa)
9:     endereco_georreferenciado ← encontrar_por_cep(resultados_geocoder, empresa.cep)
10:    empresa.latitude ← endereco_georreferenciado.latitude
11:    empresa.longitude ← endereco_georreferenciado.longitude
12:    atualizar_empresa(empresa)
return lista_empresas

```

---

A query de busca mencionada na linha 4 é:

```

SELECT *
FROM empresas_receita_curitiba
WHERE (nomeempresarial LIKE '%#{nome}%' -- utiliza-se o
↪ operador like para casos de match parcial
OR nomefantasia LIKE '%#{nome}%' -- a busca é feita pelo
↪ nome empresarial ou pelo nome fantasia
OR levenshtein(nomeempresarial, '#{nome}') < 4 -- algoritmo
↪ de levenshtein para contornar erros de digitação
OR levenshtein(nomefantasia, '#{nome}') < 4)
AND situacaocadastral != '08'; -- retorna apenas empresas
↪ com situação ATIVA

```

A escolha do algoritmo de Levenshtein para aumentar a abrangência da consulta foi feita a partir de uma análise na base de dados, que será aprofundada na seção 5.6. Vale salientar

que a adição desse algoritmo não causou grande impacto de performance na *query*, a consulta acima sem o algoritmo é executada em torno de 0.5s enquanto a consulta com o algoritmo dura em média 1.4s. As demais opções de busca possuem lógicas semelhantes à anterior, que estão disponibilizadas no apêndice E.

## 5.6 ANÁLISE DOS ALGORITMOS DE COMPARAÇÃO TEXTUAL

Com intuito de aumentar a abrangência da busca, possibilitando contornar possíveis erros de digitação ou abreviações, foi decidido utilizar estratégias de comparação textual aproximada. Para isso, comparamos a performance de três algoritmos com dados da tabela de empresas: Distância de Levenshtein[29], Soundex[31] adaptado ao português brasileiro, cuja implementação está disponível no apêndice A, e Metaphone-pt\_BR[35].

O processo de análise foi dividido em 4 etapas. A primeira etapa consistiu em uma análise sobre os dados da tabela, para identificar incidências de erros de digitação ou falta de padronização nos cadastros. Como resultado, foi identificado que os campos de endereço possuem vários erros de digitação, uso inconsistente de abreviações ou falta de padronização na classificação para identificar a mesma entidade.

A segunda etapa consistiu em listar as variações de escrita com maior ocorrência dentre esses campos. Para isso, listamos as 20 ruas com maior quantidade de empresas registradas e para cada um desses endereços foram encontradas as 5 variações mais ocorrentes na base. Para a Alameda Augusto Stellfeld, foram listados os resultados da Tabela 6:

Ocorrências	Endereço
539	ALAMEDA AUGUSTO STELLFELD
229	RUA AUGUSTO STELLFELD
193	RUA AUGUSTO STELFELD
19	ALAMEDA AUGUSTO STELFELD
11	RUA AUGUSTO STELFELLD

**Tabela 6: Análise de algoritmos de comparação textual - Etapa 2: Alameda Augusto Stellfeld**

A etapa seguinte foi identificar, dentre as variações elencadas anteriormente, quais poderiam ser consideradas a forma normalizada para referenciar o endereço corretamente. A estratégia escolhida foi definir a variação mais utilizada como a forma “correta”, que seria utilizada para a comparação e análise dos algoritmos. Para a mesma rua da segunda etapa, foi inferida a análise descrita na Tabela 7.

<b>Endereço</b>	<b>Endereço normalizado</b>
ALAMEDA AUGUSTO STELLFELD	ALAMEDA AUGUSTO STELLFELD
RUA AUGUSTO STELLFELD	ALAMEDA AUGUSTO STELLFELD
RUA AUGUSTO STELFELD	ALAMEDA AUGUSTO STELLFELD
ALAMEDA AUGUSTO STELFELD	ALAMEDA AUGUSTO STELLFELD
RUA AUGUSTO STELFELLD	ALAMEDA AUGUSTO STELLFELD

**Tabela 7: Análise de algoritmos de comparação textual - Etapa 3: Alameda Augusto Stellfeld**

A quarta e última etapa da análise consistiu em aplicar os três algoritmos escolhidos, comparando o registro com o endereço normalizado, e analisar suas eficácias ao assimilar se o endereço do registro e o endereço normalizado correspondem à mesma entidade. A Tabela 8 representa o resultado dessa análise.

<b>Algoritmo</b>	<b>Taxa de acerto(%)</b>
Levenshtein(distância máx 4)	45,74
SoundexBr	32,98
Metaphone-pt_Br	32,98

**Tabela 8: Análise de algoritmos de comparação textual - Resultado**

Deste resultado, é possível concluir-se que: os algoritmos fonéticos apresentaram a mesma performance e o algoritmo de Levenshtein foi o que se mostrou mais efetivo para interpretar as variações contidas na base de dados. Visto que os algoritmos fonéticos foram desenvolvidos para identificar palavras que soam de forma similar, eles não são tão assertivos para lidar com falta de padronização na classificação dos endereços (ex. Rua ↔ Alameda) e com abreviações (ex. Marechal ↔ Mal.), enquanto o algoritmo de Levenshtein, utilizando o valor 4 como parâmetro de distância máximo entre as *strings*, foi capaz de assimilar a correspondência entre registros com poucos erros de digitação e algumas abreviações, como pode ser observado no apêndice F.

A partir dessas conclusões, as consultas realizadas pela aplicação foram todas adaptadas para utilizar o algoritmo de Levenshtein a fim de aumentar a abrangência dos resultados da busca.

## 5.7 LIMITAÇÕES E DESAFIOS ENCONTRADOS

No escopo deste protótipo, tivemos dois grandes desafios.



O primeiro desafio foi aumentar a abrangência da busca sem afetar muito sua assertividade, ou seja, era necessário que o algoritmo de busca fosse capaz de retornar resultados aproximados dos dados inseridos, a ponto de contornar possíveis erros de digitação ou padronização, mas sem considerar como um *match* registros muito destoantes dos parâmetros informados. Para isso foram realizados diversos testes de mesa com algoritmo de Levenshtein [29], para encontrar um valor de distância máximo a ser considerado como igualdade sem que fosse adicionado ao resultado registros muito diferentes do *input*.

O segundo desafio consistiu em diminuir o uso de quota dos serviços de geocodificação através do aproveitamento de dados já existentes em uma outra entidade na base de dados, a fim de diminuir o tempo de resposta da busca e auxiliar a não atingir o limite de uso dos serviços de geocodificação. Para isso, foi implementada uma *query SQL* para relacionar os dados das empresas disponibilizados pela receita federal, com os dados de alvarás disponibilizados pela prefeitura de Curitiba.

A tabela “alvara” possui 155760 registros, enquanto a tabela “empresas\_receita\_curitiba”, 585624 registros. Após analisar os atributos de ambas as entidades e levando em consideração o contexto histórico dos dados das empresas, a única forma de garantir que a latitude e longitude da empresa seja a mesma do alvará seria fazendo o *match* entre as tabelas através do endereço.

Devido a falta de padronização quanto ao uso de abreviações nos dados de endereços fornecidos pela Receita Federal, foi observado um número razoável de registros em que não foi possível fazer o *match*. A Figura 12 ilustra algumas diferentes grafias para o mesmo endereço. A falta de normalização de dados é um problema que atrapalha os consumidores de dados abertos. O processo de *entity matching* é dificultado quando os dados não possuem um padrão. Um procedimento simples e recomendado para que a RFB normalize seus dados de endereços seria seguir o padrão proposto pelos Correios<sup>16</sup>, empresa pública federal responsável pela execução do sistema de envio e entrega de correspondências no Brasil: não utilizar abreviações e consultar o endereço nos correios para utilizar a classificação correta e evitar erros de digitação.

---

<sup>16</sup><https://www.correios.com.br/>

**Figura 12: As 10 grafias mais recorrentes para a Alameda Dom Pedro II**

	logradouro_completo	ocorrencias
1	ALAMEDA DOM PEDRO II	330
2	RUA DOM PEDRO II	48
3	ALAMEDA D PEDRO II	47
4	RUA D PEDRO II	29
5	ALAMEDA D. PEDRO II	10
6	RUA ALAMEDA D PEDRO II	5
7	RUA D. PEDRO II	5
8	ALAMEDA D.PEDRO II	4
9	AL D. PEDRO II	4
10	RUA ALAMEDA DOM PEDRO II	4

**Fonte: Autoria própria**

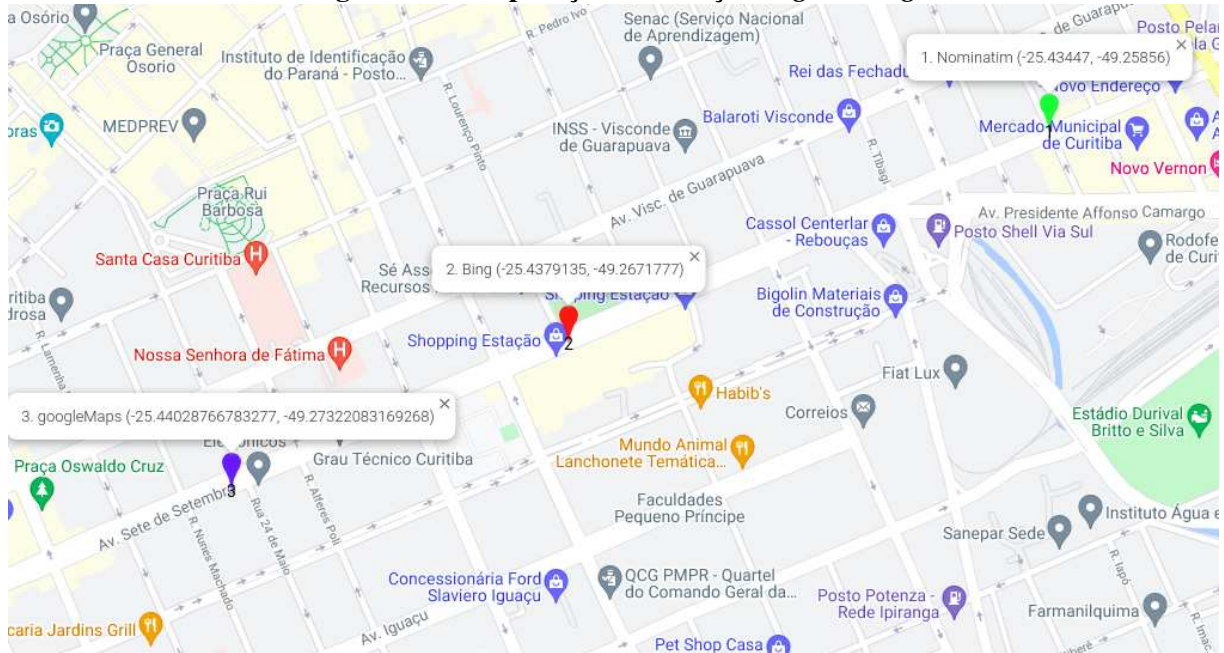
Com a execução do script mencionado anteriormente, foi possível popular os dados geoespaciais de cerca de 22% da tabela `empresas_receita_curitiba`.

Outro desafio encontrado na implementação foi a escolha do serviço de geocodificação. Muitos dos serviços gratuitos demonstraram uma precisão ruim durante os testes, como por exemplo o Nominatim<sup>17</sup> que demonstrava erros de até 2km em certos casos. Na Figura 13 podemos observar a diferença do resultado da geocodificação de três serviços – Bing, Nominatim e GoogleMaps – para o mesmo endereço: Avenida Sete de Setembro 3561, Centro, Curitiba - PR. Como o Buskaki Empresas é uma aplicação *web* sem restrição de uso, utilizar um serviço *pay-as-you-go*, como a GoogleMaps Geocoding API<sup>18</sup>, é um risco, pois quando é excedida a quota grátis do serviço o uso da aplicação pode gerar um alto custo inesperado.

<sup>17</sup><https://wiki.openstreetmap.org/wiki/Nominatim>

<sup>18</sup><https://developers.google.com/maps/documentation/geocoding/overview>

**Figura 13: Comparação de serviços de geocoding**



**Fonte: Autoria própria**

Segundo Davis e Fonseca [56], para que a geocodificação tenha maior assertividade, antes de realizar a consulta em um serviço o endereço precisa ser formatado, facilitando o *match* das informações. Cada API retorna os resultados em um formato diferente, então para cada serviço testado também era necessário adequar o *parsing* da resposta da API.

As principais limitações do protótipo apresentado são:

- Não foi implementada uma automação para atualizar os dados das empresas na base de dados;
- Devido à pandemia do COVID-19, não foi possível realizar testes em campo com aplicação de questionário com o público alvo principal ao qual a aplicação foi destinada: os voluntários da AAMA;
- O protótipo está hospedado na plataforma Heroku<sup>19</sup>, utilizando um *container* gratuito, que possui 512MB de RAM dedicada, o que afeta o tempo de resposta da aplicação.
- Não foram adicionados testes unitários nos métodos implementados, uma boa prática para garantir que alterações ou desenvolvimento de novas funcionalidades não afetem negativamente o funcionamento da aplicação.

## 6 CONCLUSÃO

Este trabalho utilizou conceitos de banco de dados, API, cidades inteligentes, dados abertos e geoprocessamento de forma prática para tentar solucionar um problema do mundo real através de diversas tecnologias dentro da área. O consumo de dados abertos pode ser problemático quando os dados não estão padronizados.

Neste contexto, o projeto Buskaki Empresas surgiu da ideia de desenvolver uma ferramenta para consultar os dados abertos de empresas curitibanas. Ele oferece serviços básicos, porém essenciais, e utiliza os dados abertos da Receita Federal e informações fornecidas pela prefeitura do município. Apesar de existirem aplicações com propostas semelhantes, em geral, estas estão sob responsabilidade de empresas privadas – fazendo com que o acesso às informações coletadas esteja sujeito à cobrança – ou não aplicam nenhum recurso que facilite a busca de informações por usuários leigos. O trabalho e o protótipo trazem uma possível solução para este problema.

Para auxiliar a limitação do escopo e a escolha das tecnologias que seriam utilizadas, foi realizada uma revisão bibliográfica sobre os principais conceitos que sustentam o projeto, verificando o funcionamento de aplicações que os usam. A análise e modelagem resultaram no projeto de uma aplicação web que é uma ferramenta que permite a consulta e visualização das informações de empresas curitibanas, com o auxílio de um mapa dinâmico para facilitar o reconhecimento do registro desejado.

O protótipo já foi disponibilizado para os voluntários da Associação Amigas da Mama (AAMA), que elogiaram a usabilidade da ferramenta e já começaram a incluir seu uso em suas rotinas.

Visando aperfeiçoar a experiência na utilização da aplicação Buskaki Empresas, é possível sugerir a implementação de uma nova funcionalidade responsável por atualizar automaticamente a base de dados através de um *crawler*, um software desenvolvido para realizar uma varredura na internet de maneira sistemática em busca de informações relevantes para sua função, que identifique que a Receita Federal disponibilizou uma nova versão dos dados e

atualize a base de acordo com eles. Outras possíveis melhorias seriam: aperfeiçoar a funcionalidade de histórico com um link direto para o resultado de cada consulta feita anteriormente, adicionar a implementação de um dicionário para lidar com o *match* de abreviações e permitir a atualização manual de endereços.

## REFERÊNCIAS

- [1] A. Lemos, “Cidades inteligentes,” *GV EXECUTIVO*, vol. 12, no. 2, pp. 46–49, 2013.
- [2] S. Isotani and I. I. Bittencourt, *Dados Abertos Conectados: Em busca da Web do Conhecimento*. Novatec Editora, 2015.
- [3] M. C. Weiss, R. C. Bernardes, and F. L. Consoni, “Cidades inteligentes: casos e perspectivas para as cidades brasileiras.,” *Revista tecnológica da Fatec americana*, vol. 5, no. 1, pp. 01–13, 2017.
- [4] C. R. de Souza, D. Redmiles, L.-T. Cheng, D. Millen, and J. Patterson, “Sometimes you need to see through walls: a field study of application programming interfaces,” in *Proceedings of the 2004 ACM conference on Computer supported cooperative work*, pp. 63–71, 2004.
- [5] R. Elmasri, S. B. Navathe, and M. G. Pinheiro, *Sistemas de banco de dados*. São Paulo: Pearson Addison Wesley, 2005.
- [6] K. Gama, A. Alvaro, and E. Peixoto, “Em direção a um modelo de maturidade tecnológica para cidades inteligentes,” in *Anais do VIII Simpósio Brasileiro de Sistemas de Informação*, pp. 513–518, SBC, 2012.
- [7] C. H. B. Rocha, *Geoprocessamento: tecnologia transdisciplinar*. Ed. do autor, 2002.
- [8] M. C. Weiss, R. C. Bernardes, and F. L. Consoni, “Cidades inteligentes como nova prática para o gerenciamento dos serviços e infraestruturas urbanas: a experiência da cidade de porto alegre,” *urbe. Revista Brasileira de Gestão Urbana*, vol. 7, no. 3, pp. 310–324, 2015.
- [9] H. Schaffers, N. Komninos, M. Pallot, B. Trousse, M. Nilsson, and A. Oliveira, “Smart cities and the future internet: Towards cooperation frameworks for open innovation,” in *The future internet assembly*, pp. 431–446, Springer, Berlin, Heidelberg, 2011.
- [10] M. Batty, K. W. Axhausen, F. Giannotti, A. Pozdnoukhov, A. Bazzani, M. Wachowicz, G. Ouzounis, and Y. Portugali, “Smart cities of the future,” *The European Physical Journal Special Topics*, vol. 214, no. 1, pp. 481–518, 2012.
- [11] G. Percivall, C. Rösndorf, S. Liang, D. McKenzie, and L. McKee, “Ogc smart cities spatial information framework,” *OGC White Paper*, 2015.
- [12] C. C. Dutra and K. M. G. Lopes, “Dados abertos: Uma forma inovadora de transparência.,” in *Anais do VI Congresso CONSAD de Gestão Pública, Brasília*, 2013.
- [13] V. Diniz, “Como conseguir dados governamentais abertos,” in *Congresso consad de gestão pública, III, Brasília*, 2010.
- [14] B. Ubaldi, “Open government data: Towards empirical analysis of open government data initiatives,” *OECD Working Papers on Public Governance*, n. 22, 2013.

- [15] D. Eaves, “The three laws of open government data,” *Eaves. ca*, vol. 30, no. 8, 2009.
- [16] C. J. Date, *Introdução a sistemas de bancos de dados*. Elsevier Brasil, 2004.
- [17] W. W. Cohen, H. Kautz, and D. McAllester, “Hardening soft information sources,” in *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 255–259, 2000.
- [18] M. A. Hernández and S. J. Stolfo, “The merge/purge problem for large databases,” *ACM Sigmod Record*, vol. 24, no. 2, pp. 127–138, 1995.
- [19] E. Rahm and H. H. Do, “Data cleaning: Problems and current approaches,” *IEEE Data Eng. Bull.*, vol. 23, no. 4, pp. 3–13, 2000.
- [20] H. B. Newcombe, J. M. Kennedy, S. Axford, and A. P. James, “Automatic linkage of vital records,” *Science*, vol. 130, no. 3381, pp. 954–959, 1959.
- [21] I. P. Fellegi and A. B. Sunter, “A theory for record linkage,” *Journal of the American Statistical Association*, vol. 64, no. 328, pp. 1183–1210, 1969.
- [22] H. Wang, *Innovative Techniques and Applications of Entity Resolution*. IGI Global, 2014.
- [23] L. A. Digiampietri, J. P. Mena-Chalco, G. S. Silva, L. B. Oliveira, A. P. Malheiros, and D. Meira, “Dinâmica das relações de coautoria nos programas de pós-graduação em computação no brasil,” in *Brazilian Workshop on social Network Analysis and Mining (BRASNAM)*, SBC, 2012, Curitiba. Anais [...]. Porto Alegre: Sociedade Brasileira de Computação, 2012 .
- [24] S. Guha, N. Koudas, A. Marathe, and D. Srivastava, “Merging the results of approximate match operations,” in *Proceedings of the Thirtieth international conference on Very large data bases-Volume 30*, pp. 636–647, 2004.
- [25] S. Sarawagi and A. Kirpal, “Efficient set joins on similarity predicates,” in *Proceedings of the 2004 ACM SIGMOD international conference on Management of data*, pp. 743–754, 2004.
- [26] N. Singla and D. Garg, “String matching algorithms and their applicability in various applications,” *International journal of soft computing and engineering*, vol. 1, no. 6, pp. 218–222, 2012.
- [27] D. L. V. G. Ruberto and R. L. Antoniazzi, “Análise e comparação de algoritmos de similaridade e distância entre strings adaptados ao português brasileiro,” in *Escola Regional de Banco de Dados (ERBD)*, 13., 2017, Passo Fundo. Anais [...]. Porto Alegre: Sociedade Brasileira de Computação, 2017 . ISSN 2595-413X.
- [28] S.-M. Chen, S.-H. Cheng, and T.-C. Lan, “A novel similarity measure between intuitionistic fuzzy sets based on the centroid points of transformed fuzzy numbers with applications to pattern recognition,” *Information Sciences*, vol. 343, pp. 15–40, 2016.
- [29] V. Levenshtein, “Levenshtein distance.” <https://xlinux.nist.gov/dads//HTML/Levenshtein.html>, 1965.

- [30] J. B. Kruskal, “An overview of sequence comparison: Time warps, string edits, and macromolecules,” *SIAM review*, vol. 25, no. 2, pp. 201–237, 1983.
- [31] R. Russell and M. Odell, “Soundex patent 01 261 167,” 1918.
- [32] L. Philips, “Hanging on the metaphone,” *Computer Language*, vol. 7, no. 12, pp. 39–43, 1990.
- [33] L. Philips, “The double metaphone search algorithm,” *C/C++ users journal*, vol. 18, no. 6, pp. 38–43, 2000.
- [34] V. P. Parmar and C. Kumbharana, “Study existing various phonetic algorithms and designing and development of a working model for the new developed algorithm and comparison by implementing it with existing algorithm (s),” *International Journal of Computer Applications*, vol. 98, no. 19, pp. 45–49, 2014.
- [35] C. C. Jordão and J. L. G. Rosa, “Metaphone-pt.br: The phonetic importance on search and correction of textual information,” in *International Conference on Intelligent Text Processing and Computational Linguistics*, pp. 297–305, Springer, 2012.
- [36] E. Hamada and R. R. do Valle Gonçalves, “Introdução ao geoprocessamento: princípios básicos e aplicação,” Embrapa Meio Ambiente, 2007.
- [37] J. O. N. Braga, L. A. d. Costa, A. L. Guimarães, and J. C. R. Tello, “O uso do geoprocessamento no diagnóstico dos roteiros de coleta de lixo da cidade de manaus,” *Engenharia Sanitária e Ambiental*, vol. 13, no. 4, pp. 387–394, 2008.
- [38] R. C. Criado and E. L. Piroli, “Geoprocessamento como ferramenta para a análise do uso da terra em bacias hidrográficas,” *Revista Geonorte*, vol. 3, no. 6, pp. 1010–1021, 2012.
- [39] D. Boemo, *Desenvolvimento de sistemas de geoprocessamento e tecnologia móvel aplicados à agricultura de precisão*. PhD thesis, Universidade Federal de Santa Maria, 2011.
- [40] F. C. Farina, “Abordagem sobre as técnicas de geoprocessamento aplicadas ao planejamento e gestão urbana,” *Cadernos EBAPE. br*, vol. 4, no. 4, pp. 01–13, 2006.
- [41] F. A. C. Rodrigues, “Utilização do geoprocessamento na rede de comunicação de dados, voz e imagem da prefeitura de belo horizonte,” Monografia (Aperfeiçoamento/Especialização em Especialização em Geoprocessamento) - Instituto de Geociências - UFMG. Belo Horizonte, p. 51. 2010.
- [42] N. Kozievitch, R. Berardi, and M. Kageyama, “Acidentes nas rodovias brasileiras nos últimos 10 anos: uma análise com dados abertos,” in *Anais da XV Escola Regional de Banco de Dados*, pp. 71–80, SBC, 2019.
- [43] L. P. Carminato, S. Ciarlariello, and S. V. Denser Pamboukian Orientador-sergio, “Uso de geoprocessamento no mercado imobiliário,” *Revista Mackenzie De Engenharia E Computação*, vol. 18, no. 1, 2018.
- [44] C. d. C. Barcellos and W. M. Ramalho, “Situação atual do geoprocessamento e da análise de dados espaciais em saúde no brasil,” pp. 221–230, 2002.



- [45] M. J. Hutchinson, *Developing an agent-based framework for intelligent geocoding*. PhD thesis, Curtin University, 2010.
- [46] D. Martins, C. A. Davis Jr, and F. T. Fonseca, “Geocodificação de endereços urbanos com indicação de qualidade,” *Proceedings XIII GEOINFO*, pp. 36–41, 2012.
- [47] “Ruby.” <https://www.ruby-lang.org/pt/>, 2021.
- [48] “Jetbrains.” <https://www.jetbrains.com/>, 2018.
- [49] JetBrains, “Rubymine.” <https://www.jetbrains.com/ruby/>, 2021.
- [50] “Rails.” <https://rubyonrails.org/>, 2021.
- [51] “Geocoder.” <https://rubygems.org/gems/geocoder>, 2021.
- [52] “Net ssh gateway.” <https://rubygems.org/gems/net-ssh-gateway>, 2021.
- [53] “Gmaps4rails.” <https://rubygems.org/gems/gmaps4rails>, 2021.
- [54] F. A. M. Kono, “Um modelo de representação computacional baseado em conceitos de crescimento urbano associados a alvarás e primitivas em banco de dados espacial,” Master’s thesis, Universidade Tecnológica Federal do Paraná, 2016.
- [55] Y. S. Bichibichi, N. P. Kozievitch, and R. A. Carvalho, “Análise de evolução de emissão de alvarás próximos a dois shoppings em curitiba,” in *Escola Regional de Banco de Dados (ERBD), 14.*, 2018, Rio Grande. Anais [...]. Porto Alegre: Sociedade Brasileira de Computação, 2018 . ISSN 2595-413X.
- [56] C. A. Davis and F. T. Fonseca, “Assessing the certainty of locations produced by an address geocoding system,” *Geoinformatica*, vol. 11, no. 1, pp. 103–129, 2007.

## APÊNDICE A - ALGORITMO SOUNDEX BR

```

CREATE OR REPLACE FUNCTION soundex_br(input text) RETURNS text
    IMMUTABLE STRICT COST 500 LANGUAGE plpgsql
AS $$
DECLARE
    soundex text = '';
    char text;
    symbol text;
    last_symbol text = '';
    pos int = 1;
BEGIN
    input = replace(input, 'DI', 'J');
    input = replace(input, 'GI', 'J');
    input = replace(input, 'TI', 'J');
    input = replace(input, 'DI', 'J');
    input = replace(input, 'CH', 'J');
    input = replace(input, 'LH', 'J');
    input = replace(input, 'NH', 'J');
    WHILE length(soundex) < 6
        LOOP
            char = upper(substr(input, pos, 1));
            pos = pos + 1;
            CASE char
                WHEN '' THEN -- End of input string
                    IF soundex = '' THEN
                        RETURN '';
                    ELSE
                        RETURN rpad(soundex, 4, '0');
                    END IF;
            END CASE;
        END LOOP;
END $$

```

```

    WHEN 'P', 'M', 'B' THEN symbol = '1';
    WHEN 'F', 'V' THEN symbol = '2';
    WHEN 'T', 'D', 'N' THEN symbol = '3';
    WHEN 'L', 'R' THEN symbol = '4';
    WHEN 'S', 'Z' THEN symbol = '5';
    WHEN 'J' THEN symbol = '6';
    WHEN 'K', 'C', 'G', 'Q' THEN symbol = '7';
    WHEN 'X' THEN symbol = '8';
    ELSE
        symbol = '';
    END CASE;

    IF soundex = '' THEN
        IF char ~>=~ 'A' AND char ~<=~ 'Z' THEN
            soundex = char;
            last_symbol = symbol;
        END IF;
    ELSIF last_symbol != symbol THEN
        soundex = soundex || symbol;
        last_symbol = symbol;
    END IF;
END LOOP;
RETURN soundex;
END;
$$;

```

## APÊNDICE B – CASOS DE USO

Este capítulo apresenta os casos de uso do projeto Buskaki Empresas, ilustrados na Figura 5.

### B.1 BUSCAR EMPRESA POR NOME

**Descrição:** O usuário tenta realizar uma busca por determinada empresa, inserindo o nome da mesma.

**Pré-Condições:** O usuário deve estar logado.

**Fluxo Básico:** O usuário preenche o campo de busca e envia a requisição ao servidor. Então ele é redirecionado para a tela de resultado da busca.

**Fluxo Alternativo:** Nenhum.

**Pós-Condições:** Mostrar ao usuário uma lista paginada de empresas com nome semelhante ao informado, assim como a posição geográfica dos resultados desta página num mapa dinâmico.

### B.2 BUSCA AVANÇADA

**Descrição:** O usuário tenta realizar uma busca por determinada empresa, combinando os parâmetros do formulário para restringir a busca.

**Pré-Condições:** O usuário deve estar logado.

**Fluxo Básico:** O usuário preenche os campos de busca e envia a requisição ao servidor. Então ele é redirecionado para a tela de resultado da busca.

**Fluxo Alternativo:** Nenhum.

**Pós-Condições:** Mostrar ao usuário uma lista paginada de empresas que se enquadram nos parâmetros fornecidos, assim como a posição geográfica dos resultados desta página num mapa dinâmico.

### B.3 VISUALIZAR PERFIL

**Descrição:** O usuário clica no menu para visualizar seu perfil.

**Pré-Condições:** O usuário deve estar logado.

**Fluxo Básico:** O usuário é redirecionado para tela Meu perfil.

**Fluxo Alternativo:** Nenhum.

**Pós-Condições:** Mostrar informações do cadastro do usuário.

### B.4 ACESSAR HISTÓRICO

**Descrição:** O usuário clica no menu para visualizar seu histórico de busca.

**Pré-Condições:** O usuário deve estar logado e ter realizado alguma busca.

**Fluxo Básico:** O usuário é redirecionado para tela Histórico de buscas.

**Fluxo Alternativo:** Nenhum.

**Pós-Condições:** Mostrar tabela com histórico das consultas realizadas pelo usuário.

### B.5 VISUALIZAR INFORMAÇÕES DE UMA EMPRESA

**Descrição:** O usuário está na tela de resultados de uma busca e clica em Exibir.

**Pré-Condições:** O usuário deve estar logado, realizado uma busca por nome, rua, bairro ou avançada.

**Fluxo Básico:** O usuário é redirecionado para tela com informações da empresa em questão.

**Fluxo Alternativo:** Nenhum.

**Pós-Condições:** Mostrar todos os dados da empresa contidos na base de dados, assim como ilustrar sua geolocalização num mapa.

### B.6 CRIAR CONTA

**Descrição:** O usuário tenta criar uma nova conta, inserindo: nome completo (obrigatório), e-mail (obrigatório), senha (obrigatório) e confirmação de senha (obrigatório).

**Pré-Condições:** O usuário não pode estar logado.

**Fluxo Básico:** O usuário preenche os dados requeridos e os envia ao servidor. Então ele é redirecionado para a tela de login com a informação de um novo usuário foi cadastrado.

**Fluxo Alternativo:** Se as senhas não coincidirem, o e-mail já estiver em uso ou algum dos campos não estiver preenchido, o usuário é informado da situação e nenhum dado é salvo no

banco.

**Pós-Condições:** Redirecionar o usuário para tela de login com o cadastro efetuado.

## B.7 LOGIN

**Descrição:** O usuário tenta realizar login, inserindo: e-mail (obrigatório) e senha (obrigatório).

**Pré-Condições:** O usuário não pode estar logado.

**Fluxo Básico:** O usuário preenche os dados requeridos e os envia ao servidor. Então ele é redirecionado para a tela de Buscar empresa por nome.

**Fluxo Alternativo:** Se a senha não estiver correta, é mostrada uma mensagem de erro para o usuário e permanece na mesma tela.

**Pós-Condições:** Redirecionar o usuário para tela de Buscar empresa por nome, com o login efetuado.

## B.8 LOG OUT

**Descrição:** O usuário clica em Logout no canto superior esquerdo.

**Pré-Condições:** O usuário deve estar logado.

**Fluxo Básico:** O usuário é redirecionado para tela de login.

**Fluxo Alternativo:** Nenhum.

**Pós-Condições:** Usuário não deve conseguir acessar a área autenticada sem realizar o login novamente.

## B.9 BUSCAR EMPRESA POR RUA

**Descrição:** O usuário tenta realizar uma busca por empresas de uma determinada rua.

**Pré-Condições:** O usuário deve estar logado.

**Fluxo Básico:** O usuário preenche o campo de busca e envia a requisição ao servidor. Então ele é redirecionado para a tela de resultado da busca.

**Fluxo Alternativo:** Nenhum.

**Pós-Condições:** Mostrar ao usuário uma lista paginada de empresas situadas na rua informada, assim como a posição geográfica dos resultados desta página num mapa dinâmico.

## B.10 BUSCAR EMPRESA POR CNPJ

**Descrição:** O usuário tenta realizar uma busca por determinada empresa, inserindo o CNPJ da mesma.

**Pré-Condições:** O usuário deve estar logado.

**Fluxo Básico:** O usuário preenche o campo de busca e envia a requisição ao servidor. Então ele é redirecionado para a tela de informações da empresa.

**Fluxo Alternativo:** Caso não seja encontrada nenhuma empresa com o CNPJ informado, o usuário permanece na mesma página e é mostrado um aviso na parte superior da tela.

**Pós-Condições:** Mostrar todos os dados da empresa contidos na base de dados, assim como ilustrar sua geolocalização num mapa.

## B.11 BUSCAR EMPRESA POR BAIRRO

**Descrição:** O usuário tenta realizar uma busca por empresas de um determinado bairro de Curitiba.

**Pré-Condições:** O usuário deve estar logado.

**Fluxo Básico:** O usuário preenche o campo de busca e envia a requisição ao servidor. Então ele é redirecionado para a tela de resultado da busca.

**Fluxo Alternativo:** Nenhum.

**Pós-Condições:** Mostrar ao usuário uma lista paginada de empresas situadas no bairro informado, assim como a posição geográfica dos resultados desta página num mapa dinâmico.

## APÊNDICE C – ESQUEMA DE DADOS

Este capítulo apresenta as descrições e script de criação das tabelas utilizadas pela aplicação, ilustrados na Figura 6.

### C.1 TABELA DE EMPRESAS

---

```

CREATE TABLE empresas_receita_curitiba
(
  idempresas      SERIAL PRIMARY KEY, -- Chave primária da tabela
  cnpj            VARCHAR(14) DEFAULT NULL::CHARACTER VARYING, -- cnpj da empresa
  identificadormatrizfilial VARCHAR(1) DEFAULT NULL::CHARACTER VARYING, --
  ↪ Matriz ou Filial
  nomeempresarial VARCHAR(150) DEFAULT NULL::CHARACTER VARYING, -- Razão
  ↪ social da empresa
  nomefantasia    VARCHAR(55) DEFAULT NULL::CHARACTER VARYING, -- Nome
  ↪ comercial da empresa
  situacaocadastral VARCHAR(2) DEFAULT NULL::CHARACTER VARYING, -- NULA,
  ↪ ATIVA, SUSPENSA, INAPTA, BAIXADA
  datasituacaocadastral VARCHAR(8) DEFAULT NULL::CHARACTER VARYING, -- Data
  ↪ do evento da situação cadastral
  motivosituacaocadastral VARCHAR(2) DEFAULT NULL::CHARACTER VARYING, --
  ↪ Código do motivo da situação cadastral
  nomecidadeexterior VARCHAR(55) DEFAULT NULL::CHARACTER VARYING, -- Nome
  ↪ da cidade no exterior
  codigopais      VARCHAR(3) DEFAULT NULL::CHARACTER VARYING, -- Código do
  ↪ país
  nomepais        VARCHAR(70) DEFAULT NULL::CHARACTER VARYING, -- Nome do país
  codigonaturezajuridica VARCHAR(4) DEFAULT NULL::CHARACTER VARYING, --
  ↪ Código da natureza jurídica da empresa segundo tabela do IBGE
  datainicioatividade VARCHAR(8) DEFAULT NULL::CHARACTER VARYING, -- Data
  ↪ de abertura
  cnaefiscal      VARCHAR(7) DEFAULT NULL::CHARACTER VARYING, -- Código
  ↪ descritor da atividade principal da empresa
  descricao tipologradouro VARCHAR(20) DEFAULT NULL::CHARACTER VARYING, --
  ↪ Descrição do logradouro (Rua, Avenida, Praça, etc)
  logradouro      VARCHAR(60) DEFAULT NULL::CHARACTER VARYING, -- Nome do
  ↪ logradouro onde se localiza o estabelecimento

```



```

numero      VARCHAR(6) DEFAULT NULL::CHARACTER VARYING, -- Número do
↳ logradouro
complemento  VARCHAR(156) DEFAULT NULL::CHARACTER VARYING, --
↳ Complemento para o endereço
bairro      VARCHAR(50) DEFAULT NULL::CHARACTER VARYING, -- Bairro onde se
↳ localiza o estabelecimento
cep         VARCHAR(8) DEFAULT NULL::CHARACTER VARYING, -- Código de endereço
↳ postal referente ao logradouro
uf          VARCHAR(2) DEFAULT NULL::CHARACTER VARYING, -- Sigla da unidade da
↳ federação onde se encontra o estabelecimento
codigomunicipal VARCHAR(4) DEFAULT NULL::CHARACTER VARYING, -- Código do
↳ município de jurisdição
municipio   VARCHAR(50) DEFAULT NULL::CHARACTER VARYING, -- Nome do
↳ município
dddtelefone VARCHAR(4) DEFAULT NULL::CHARACTER VARYING,
telefone    VARCHAR(8) DEFAULT NULL::CHARACTER VARYING,
emailcontribuinte VARCHAR(115) DEFAULT NULL::CHARACTER VARYING, --
↳ e-mail de contato da empresa
qualificacaoresponsavel VARCHAR(2) DEFAULT NULL::CHARACTER VARYING, --
↳ Qualificação da pessoa física responsável pela empresa
capitalsocial VARCHAR(14) DEFAULT NULL::CHARACTER VARYING, -- Valor do
↳ capital social aplicado na abertura da empresa
porteempresa VARCHAR(2) DEFAULT NULL::CHARACTER VARYING, --
↳ MICROEMPRESA, EMPRESA DE PEQUENO PORTE, OUTROS
opcaosimples VARCHAR(1) DEFAULT NULL::CHARACTER VARYING, -- Optante
↳ pelo regime tributário Simples Nacional
dataopcaosimples VARCHAR(8) DEFAULT NULL::CHARACTER VARYING, -- Data de
↳ opção pelo regime Simples Nacional
dataexclusaosimples VARCHAR(8) DEFAULT NULL::CHARACTER VARYING, -- Data
↳ de exclusão do Simples Nacional
opcaomei    VARCHAR(1) DEFAULT NULL::CHARACTER VARYING, -- Optante pelo
↳ regime tributário Micro Empresário Individual
situacaoespecial VARCHAR(23) DEFAULT NULL::CHARACTER VARYING, --
↳ Situação especial da empresa
datasituacaoespecial VARCHAR(8) DEFAULT NULL::CHARACTER VARYING, -- Data
↳ da situação especial da empresa
geom        geometry(Point, 4326), -- Informação geoespacial gerada a partir
↳ dos dados de latitude e longitude
latitude    DOUBLE PRECISION, -- latitude da coordenada geográfica onde
↳ está localizada a empresa
longitude   DOUBLE PRECISION -- longitude da coordenada geográfica onde
↳ está localizada a empresa
);

```

---

## C.2 TABELA DE ALVARÁS

---

```

CREATE TABLE alvara
(
  id                SERIAL PRIMARY KEY -- chave primária da
    ↳ tabela,
  nome_empresarial  VARCHAR(100) -- razão social da empresa,
  data_inicio_atividade DATE -- data de abertura da empresa,
  numero_do_alvara  INTEGER -- número do alvará emitido,
  nome_da_empresa   VARCHAR(100) -- nome fantasia da empresa,
  data_emissao      DATE -- data de emissão do alvará,
  data_expiracao    DATE -- data de expiração do alvará,
  atividade_principal VARCHAR(200) -- atividade principal da
    ↳ empresa,
  atividade_secundaria1 VARCHAR(200) -- atividade secundária da
    ↳ empresa,
  atividade_secundaria2 VARCHAR(200) -- outra atividade secundária
    ↳ da empresa,
  endereco          VARCHAR(255) -- endereço da empresa (tipo
    ↳ do logradouro + logradouro),
  numero            VARCHAR(15) -- número do logradouro,
  unidate           VARCHAR(15) -- identificação do
    ↳ apartamento ou loja em que a empresa se encontra,
  andar             VARCHAR(15) -- número do andar em que a
    ↳ empresa se encontra,
  complemento       VARCHAR(200) -- complemento do endereço,
  bairro            VARCHAR(50) -- bairro em que se localiza o
    ↳ estabelecimento,
  cep               VARCHAR(8) -- código de endereço postal
    ↳ referente ao logradouro,
  cidade            VARCHAR(50) -- nome do município da
    ↳ empresa,
  estado            VARCHAR(50) -- sigla da unidade da
    ↳ federação onde se encontra o estabelecimento,
  latitude          DOUBLE PRECISION -- latitude da coordenada
    ↳ geográfica onde está localizada a empresa,
  longitude         DOUBLE PRECISION -- longitude da
    ↳ coordenada geográfica onde está localizada a empresa,
  localidade        geometry -- informação geoespacial gerada
    ↳ a partir dos dados de latitude e longitude,
  atividade_principal_agregada VARCHAR(100) -- classificação do "tipo de
    ↳ atividade" exercida pela empresa (ex: lanchonete, banco...)
  tipo_modificado   VARCHAR(255) -- sub-classificação da
    ↳ atividade principal agregada (ex: Pizzaria, Bar, Churrascaria...)
);

```

---

### C.3 TABELA DE USUÁRIOS

---

```
CREATE TABLE usuarios
(
  id          BIGSERIAL    NOT NULL
    CONSTRAINT usuarios_pkey
      PRIMARY KEY -- chave primária da tabela,
  nome        VARCHAR     -- nome do usuário,
  email       VARCHAR     -- email de cadastro,
  password_digest VARCHAR -- hash do password do usuário
);
```

---

### C.4 TABELA DE HISTÓRICO DE BUSCA

---

```
CREATE TABLE historico_buscas
(
  id          BIGSERIAL    NOT NULL
    CONSTRAINT historico_buscas_pkey
      PRIMARY KEY -- chave primária da tabela,
  usuario_id BIGINT       NOT NULL
    CONSTRAINT fk_rails_56c282eb47
      REFERENCES usuarios -- chave estrangeira para tabela usuarios,
  tipo_busca VARCHAR     -- tipo da busca executada pelo usuário,
  params      VARCHAR     -- parâmetros da busca executada
);
```

---

## APÊNDICE D – TELAS DA APLICAÇÃO

Neste capítulo constam todas as telas desenvolvidas para a aplicação web.

### D.1 PÁGINA INICIAL - LOGIN

**Figura 14: Tela de login**

A imagem mostra a interface de usuário para o sistema "Buskaki Empresas". No topo, o nome "Buskaki Empresas" está centralizado, e um botão "Login" está no canto superior direito. Abaixo, há um formulário centralizado com o título "Login". O formulário contém dois campos de entrada: "Email" e "Senha". Abaixo dos campos, há um botão "Entrar" e um link "Nao possuiu conta? Crie sua conta agora!".

**Fonte: Autoria própria**

## D.2 CRIAÇÃO DE CADASTRO

**Figura 15: Tela de cadastro**

The screenshot shows the registration page for 'Buskaki Empresas'. At the top right, there is a 'Login' button. The main heading is 'Buskaki Empresas'. Below this, there is a registration form with the following fields: 'Nome', 'Email', 'Senha', and 'Confirmacao de senha'. A 'Cadastrar' button is located at the bottom of the form.

**Fonte: Autoria própria**

## D.3 FORMULÁRIO DE BUSCA POR NOME

**Figura 16: Formulário para busca por nome**

The screenshot shows the search page for 'Buskaki Empresas'. At the top right, there is a 'Log out' button. The main heading is 'Buskaki Empresas'. Below this, there is a navigation menu with links: 'Buscar por nome', 'Buscar por CNPJ', 'Listar empresas por bairro', 'Listar empresas por rua', 'Meu perfil', and 'Histórico'. The main content area contains a search form with the label 'Busca por nome', a 'Nome:' label, an input field, and a 'Buscar' button. Below the search form, there is a link for 'Buscar avançada'.

**Fonte: Autoria própria**

## D.4 FORMULÁRIO DE BUSCA AVANÇADA

**Figura 17: Formulário para busca avançada**

The screenshot shows the 'Busca avançada' form within the 'Buskaki Empresas' application. The header includes the title 'Buskaki Empresas' and a 'Log out' button. Below the header, there are navigation links: 'Buscar por nome', 'Buscar por CNPJ', 'Listar empresas por bairro', 'Listar empresas por rua', 'Meu perfil', and 'Histórico'. The main form area is titled 'Busca avançada' and contains the following fields: 'Nome:', 'Bairro:', 'Endereço:', and 'CEP:'. Each field has a corresponding text input box. A 'Buscar' button is located at the bottom right of the form.

Fonte: Autoria própria

## D.5 FORMULÁRIO DE BUSCA POR RUA

**Figura 18: Formulário para busca por rua**

The screenshot shows the 'Busca por rua' form within the 'Buskaki Empresas' application. The header is identical to Figure 17, including the title 'Buskaki Empresas' and the 'Log out' button. The navigation links are also the same. The main form area is titled 'Busca por rua' and contains a single field: 'Endereço:'. This field has a corresponding text input box. A 'Buscar' button is located at the bottom right of the form.

Fonte: Autoria própria

## D.6 FORMULÁRIO DE BUSCA POR BAIRRO

**Figura 19: Formulário para busca por bairro**

The screenshot shows the 'Buskaki Empresas' website interface. At the top, there is a navigation bar with the following links: 'Buscar por nome', 'Buscar por CNPJ', 'Listar empresas por bairro', 'Listar empresas por rua', 'Meu perfil', and 'Histórico'. On the far right of the navigation bar is a 'Log out' button. Below the navigation bar, the main content area is titled 'Busca por bairro'. Under this title, there is a label 'Bairro:' followed by a text input field and a 'Buscar' button.

**Fonte: Autoria própria**

## D.7 FORMULÁRIO DE BUSCA POR CNPJ

**Figura 20: Formulário para busca por CNPJ**

The screenshot shows the 'Buskaki Empresas' website interface. At the top, there is a navigation bar with the following links: 'Buscar por nome', 'Buscar por CNPJ', 'Listar empresas por bairro', 'Listar empresas por rua', 'Meu perfil', and 'Histórico'. On the far right of the navigation bar is a 'Log out' button. Below the navigation bar, the main content area is titled 'Busca por CNPJ'. Under this title, there is a label 'CNPJ:' followed by a text input field and a 'Buscar' button.

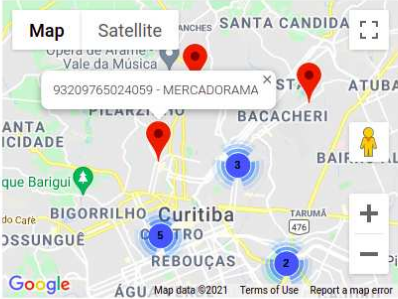
**Fonte: Autoria própria**

## D.8 RESULTADOS DA BUSCA

**Figura 21: Tela de resultados da busca**

### Buskaki Empresas

Buscar por nome
Buscar por CNPJ
Listar empresas por bairro
Listar empresas por rua
Meu perfil
Histórico
Log out



< Anterior
1
Próximo >

Cnpj	Nome Empresarial	Nome Fantasia	Endereço	Bairro	Cep	
93209765012638	WMS SUPERMERCADOS DO BRASIL LTDA.	MERCADORAMA - FILIAL CABRAL	RUA BOM JESUS 159	JUVEVE	80035010	Exibir
93209765012395	WMS SUPERMERCADOS DO BRASIL LTDA.	MERCADORAMA - FILIAL RUA MATEUS LEME	RUA MATEUS LEME 4556	SAO LOURENCO	82200000	Exibir

Fonte: Autoria própria

## D.9 INFORMAÇÕES DA EMPRESA

**Figura 22: Tela de informações da empresa**

### Buskaki Empresas

Buscar por nome
Buscar por CNPJ
Listar empresas por bairro
Listar empresas por rua
Meu perfil
Histórico
Log out

#### MERCADORAMA - FILIAL CABRAL

**CNPJ:** 93209765012638

**Matriz ou filial:** 2

**Nome empresarial:** WMS SUPERMERCADOS DO BRASIL LTDA.

**Nome fantasia:** MERCADORAMA - FILIAL CABRAL

**Situação cadastral:** 02

**Motivo da situação cadastral:** 00

**Código da natureza jurídica:** 2062

**Início das atividades:** 19990707

**CNAE Fiscal:** 4711302

**Endereço:** RUA BOM JESUS 159


**Bairro:** JUVEVE

**Cep:** 80035010

**UF:** PR

**Código municipal:** 7535

**Município:** CURITIBA



Fonte: Autoria própria



## D.10 MEU PERFIL

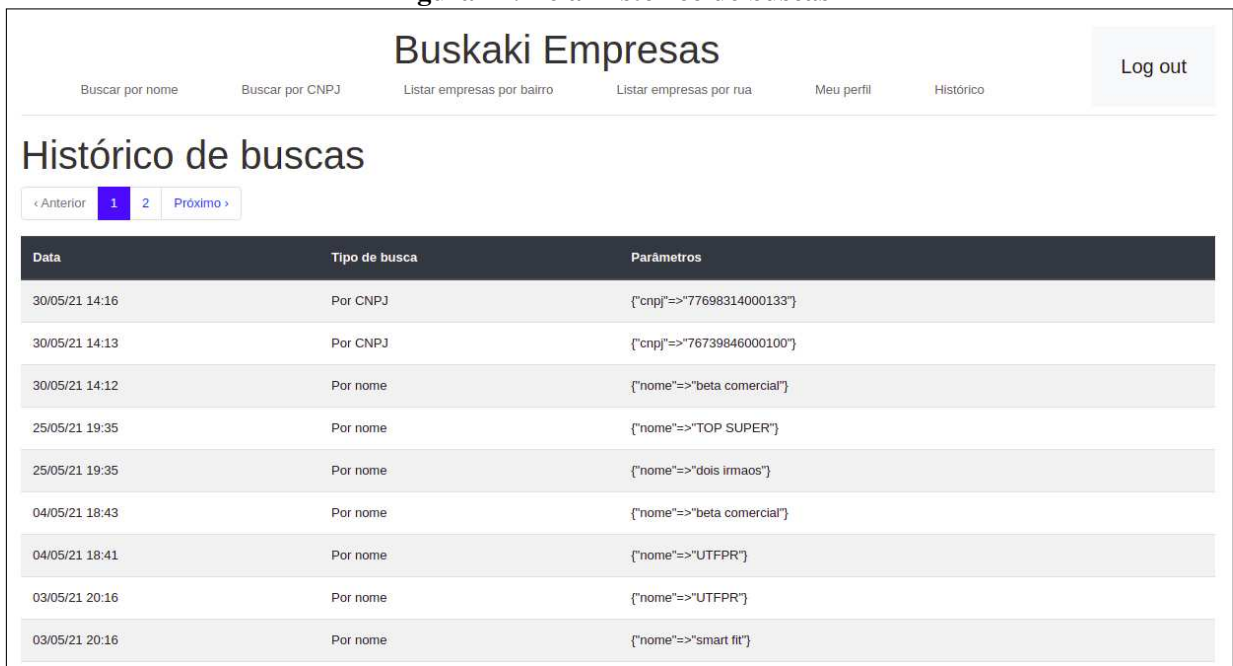
Figura 23: Tela Meu perfil



Fonte: Autoria própria

## D.11 HISTÓRICO DE BUSCA

Figura 24: Tela Histórico de buscas



Fonte: Autoria própria

## APÊNDICE E – ALGORITMOS DE BUSCA

Este capítulo descreve os algoritmos das buscas não citadas no Seção 5.5

### E.1 BUSCA POR CNPJ

---

#### Algoritmo 3 Busca por CNPJ

---

**Input:** CNPJ da empresa

**Output:** Informações da empresa

```

1: salvar_historico_buscas(cnpj)
2: cnpj ← remover_ao_numerico(cnpj)
3: empresa ← buscar_empresa_por_cnpj(nome)
4: if empresa.latitude é nulo OU empresa.longitude é nulo then
5:   endereco_empresa ← formatar_endereco_para_api_endereços(empresa)
6:   resultados_geocoder ← consulta_api_geocodificacao(endereco_empresa)
7:   endereco_georreferenciado ← encontrar_por_cep(resultados_geocoder, empresa.cep)
8: empresa.latitude ← endereco_georreferenciado.latitude
9: empresa.longitude ← endereco_georreferenciado.longitude
10: atualizar_empresa(empresa) return empresa

```

---

A query `buscar_empresa_por_cnpj` é:

```

SELECT *
FROM empresas_receita_curitiba
WHERE (cnpj = '#{cnpj}');

```

## E.2 BUSCA POR AVANÇADA

**Algoritmo 4** Busca por avançada**Input:** nome, bairro, endereco, cep**Output:** lista de empresas compatíveis

---

```

1: salvar_historico_buscas(nome, bairro, endereco, cep)
2: nome ← remover_acentos(nome)
3: nome ← para_letra_maiuscula(nome)
4: bairro ← remover_acentos(bairro)
5: bairro ← para_letra_maiuscula(bairro)
6: endereco ← remover_acentos(endereco)
7: endereco ← para_letra_maiuscula(endereco)
8: lista_empresas ← executa_query_busca_avancada(nome, bairro, endereco, cep)
9: for empresa em lista_empresas do
10:   if empresa.latitude é nulo OU empresa.longitude é nulo then
11:     endereco_empresa ← formatar_endereco_para_api_endereços(empresa)
12:     resultados_geocoder ← consulta_api_geocodificacao(endereco_empresa)
13:     endereco_georreferenciado ← encontrar_por_cep(resultados_geocoder, empresa.cep)
14:     empresa.latitude ← endereco_georreferenciado.latitude
15:     empresa.longitude ← endereco_georreferenciado.longitude
16:   atualizar_empresa(empresa)
return lista_empresas

```

---

A query `executa_query_busca_avancada` é:

```

SELECT *
FROM empresas_receita_curitiba
WHERE ((nomeempresarial LIKE '%#{nome}%'
  OR nomefantasia LIKE '%#{nome}%'
  OR levenshtein(nomeempresarial, '#{nome}') < 4
  OR levenshtein(nomefantasia, '#{nome}') < 4)
AND (CONCAT(descricaotipologradouro, ' ', logradouro) LIKE
  ↪ '%#{rua}%'
  OR levenshtein(CONCAT(descricaotipologradouro, ' ',
  ↪ logradouro), '#{rua}') < 4)
AND (bairro LIKE '%#{bairro}%'

```

```

    OR levenshtein(bairro, '#{bairro}') < 4)
  AND cep = '#{cep}'
  AND situacaocadastral != '08'; -- retorna apenas empresas com
  ↪ situação ATIVA;

```

### E.3 BUSCA POR BAIRRO

---

**Algoritmo 5** Listar empresas por bairro

---

**Input:** bairro

**Output:** lista de empresas daquele bairro

```

1: salvar_historico_buscas(bairro)
2: bairro ← remover_acentos(bairro)
3: bairro ← para_letra_maiuscula(bairro)
4: lista_empresas ← buscar_empresas_por_bairro(bairro)
5: for empresa em lista_empresas do
6:   if empresa.latitude é nulo OU empresa.longitude é nulo then
7:     endereco_empresa ← formatar_endereco_para_api_endereços(empresa)
8:     resultados_geocoder ← consulta_api_geocodificacao(endereco_empresa)
9:     endereco_georreferenciado ← encontrar_por_cep(resultados_geocoder, empresa.cep)
10:    empresa.latitude ← endereco_georreferenciado.latitude
11:    empresa.longitude ← endereco_georreferenciado.longitude
12:    atualizar_empresa(empresa)
return lista_empresas

```

---

A query buscar\_empresas\_por\_bairro é:

```

SELECT *
FROM empresas_receita_curitiba e
WHERE (bairro LIKE '%#{bairro}%')
    OR levenshtein(bairro, '#{bairro}') < 4)
  AND situacaocadastral != '08'; -- retorna apenas empresas com
  ↪ situação ATIVA;

```

## E.4 BUSCA POR RUA

**Algoritmo 6** Listar empresas por rua**Input:** rua**Output:** lista de empresas registradas na rua informada

---

```

1: salvar_historico_buscas(rua)
2: rua ← remover_acentos(rua)
3: rua ← para_letra_maiuscula(rua)
4: lista_empresas ← buscar_empresas_por_rua(rua)
5: for empresa em lista_empresas do
6:   if empresa.latitude é nulo OU empresa.longitude é nulo then
7:     endereco_empresa ← formatar_endereco_para_api_endereços(empresa)
8:     resultados_geocoder ← consulta_api_geocodificacao(endereco_empresa)
9:     endereco_georreferenciado ← encontrar_por_cep(resultados_geocoder, empresa.cep)
10:    empresa.latitude ← endereco_georreferenciado.latitude
11:    empresa.longitude ← endereco_georreferenciado.longitude
12:    atualizar_empresa(empresa)
return lista_empresas

```

---

A query buscar\_empresas\_por\_rua é:

```

SELECT *
FROM empresas_receita_curitiba e
WHERE (CONCAT(descricao_tipologradouro, ' ', logradouro) LIKE
↪ '%#{rua}%')
  OR levenshtein(CONCAT(descricao_tipologradouro, ' ',
↪ logradouro), '#{rua}') < 4)
AND situacaocadastral != '08'; -- retorna apenas empresas com
↪ situação ATIVA;

```

## **APÊNDICE F – TABELAS DA ANÁLISE DOS ALGORITMOS DE COMPARAÇÃO TEXTUAL**

Neste capítulo, é disponibilizado as tabelas utilizadas na análise dos algoritmos de comparação textual, citadas na Seção 5.6. Na etapa de listagem, foram listadas as 20 ruas com mais empresas e 5 variações de escrita do mesmo logradouro, encontradas na base. Na etapa final da análise e comparação foram executados os 3 algoritmos para cada par (endereço, endereço normalizado) e calculada a assertividade total.



Ocorrências	Tipo do logradouro	Logradouro	Bairros	Logradouro normalizado
250	ALAMEDA	DR MURICY	{CENTRO,CENTRO> SALA 5,SAO FRANCISCO}	DOUTOR MURICY
122	ALAMEDA	DR. MURICY	{CENTRO,SAO FRANCISCO}	DOUTOR MURICY
381	RUA	ENGENHEIRO REBOUCAS	{AGUA VERDE,BAIRRO REBOUCAS,CAPANEMA,CAPANEMA,CENTRO,CENTRO,IGUACU,JARDIM BOTANICA,JARDIM BOTANICO,JARDIM BOTANIO,JD BOTANICO,LIBERDADE,LINDOIA,NOSSA SRA DA LUZ,PRADO VELHO,REBOUCAS,REBOUCAS,SANTA TEREZINHA,STA TEREZINHA,VILA CAPANEMA,VILA PINTO}	ENGENHEIRO REBOUCAS
309	RUA	ENGENHEIROS REBOUCAS	{AGUA VERDE,CAPANEMA,CENTRO,JARDIM BOTANICO,JD BOTANICO,REBOUCAS,VILA CAPANEMA}	ENGENHEIRO REBOUCAS
39	RUA	ENG REBOUCAS	{AGUA VERDE,CAPANEMA,CENTRO,IGUACU,JARDIM BOTANICO,JD BOTANICO,LEAO JUNIOR,REBOUCAS}	ENGENHEIRO REBOUCAS
10	RUA	ENG. REBOUCAS	{JD BOTANICO,REBOUCAS}	ENGENHEIRO REBOUCAS
7	RUA	ENG.REBOUCAS	{CENTRO,REBOUCAS}	ENGENHEIRO REBOUCAS
5238	RUA	MARECHAL DEODORO	{ALDO DA XV,ALTO DA DA RUA XV,ALTO DA GLORIA,ALTO DA QUINZE,ALTO DA RUA QUINZE DE NOVEMBRO,ALTO DA RUA XV,ALTO DA RUA XV,ALTO DA RU XV,ALTO DA XV,ALTO DA XV,ALTO DA XV DE NOVEMBRO,ALTO DO RUA XV,ALTO RUA XV,ALTO XV,BAIRRO CENTRO,CENRO,CENTRO,CENTRO,CENTRO COMERCIAL,CENTRO COM L ITALIA,CJ 1605,CONJ 2308,CRISTO REI,CURITIBA,CURITIBA CENTRO,GENERAL CARNEIRO,ITUPAVA,SALGADINHO,SANTO AMARO,SEDE,SHOPPING ITALIA,ZONA URBANA}	MARECHAL DEODORO
757	RUA	MAL DEODORO	{ALTO DA QUINZE,ALTO DA RUA XV,ALTO DA XV,ALTO XV,B ALTO DA XV,CENTRO,CONJ 511,CRISTO REI}	MARECHAL DEODORO
193	AVENIDA	MARECHAL DEODORO	{ALTO DA RUA XV,ALTO DA XV,BAIRRO ALTO DA XV,CENTRO,CENTRO0,CENTRO/CURITIBA,CRISTO REI,SEDE}	MARECHAL DEODORO
120	RUA	MAL DEODORO	{ALTO DA RUA XV,ALTO DA RUA XV,ALTO DA XV,CENTRO,CRISTO REI,SEDE}	MARECHAL DEODORO
36	AVENIDA	MAL DEODORO	{ALTO DA XV,CENTRO,SEDE}	MARECHAL DEODORO
3756	AVENIDA	MARECHAL FLORIANO PEIXOTO	{-81730000,AGUA VERDE,ALTO BOQUEIRAO,BAIRRO PAROLIN,BOQUEIRAO,BOQUEIRAO,BOQUEIRO,BOQUIERAO,BOUEIRAO,BOQUEIRAO,B. PAROLIN,CARMO,CENTRO,CURITIBA,HAUER,HAUER LOJA 01,PAROLIM,PAROLIN,PAROLIN,PRADO VELHO,PEROLIM,PRADO VELHO,REBOUAS,REBOUCAS,SHOPPING CIDADE,VILLA HAUER,VILA FANNY,VILA HAUER,VILA PALORIN,VILA PAROLIM,VILA PAROLIN,VL HAUER,VL HAUER}	MARECHAL FLORIANO PEIXOTO
1895	RUA	MAL FLORIANO PEIXOTO	{ALTO BOQUEIRAO,ASILO,AZILO,BAIRRO BOQUEIRAO,BOQUEIRAO,BOQUEIEAO,BOQUEIRAO,BOQUEIRAO,CURITIBA,BOQUEIRO,BOUQUEIRAO,CARMO,CARMO,BOQUEIRAO,CENTRO,CENTRO/REBOUCAS,CURITIBA,HAUER,HAUER,IGUACU,LIBERDADE,NA SA DA LUZ,NOSSA SRA DA LUZ,NOSS SEN DA LUZ,N S DA LUIZ,N S DA LUZ,N S LUZ,N SRA DA LUZ,PAROLIM,PAROLIN,PRADO VELHO,REBOUCAS,SEDE,UBERABA,V HAUER,VILA FANI,VILA GUAIRA,VILA HAAVER,VILA HAUER,VILAHAUER,VILA HAUSER,VILA HAVER,VILA PAROLIM,VILA PAROLIN,VILA HAUER,VL HAUER,VL HAUER,VL HAUER,VL PAROLIN}	MARECHAL FLORIANO PEIXOTO
1067	AVENIDA	MAL FLORIANO PEIXOTO	{AGUA VERDE,ALTO BOQUEIRAO,BAIRRO HAUER,BOQUEIRAO,BOQUERAO,BOQUEIRAO,BOQUIRAO,CARMO,CENTRO,CENTRO CURITIBA,ENG REBOUCAS,HAUER,IGUACU,N S DA LUZ,PAROLIM,PAROLIN,PCA CARLOS GOMES,PRADO VELHO,REBOUCAS,REBOUCAS/CURITIBA,SEDE,VILA FANNY,VILA HAUER,VILA HAUR,VILA PALORIN,VILA PAROLIM,VILA PAROLIN,VL HAUER,VL HAUER,VL PAROLIN,VL PAROLIN}	MARECHAL FLORIANO PEIXOTO
485	RUA	MARECHAL FLORIANO PEIXOTO	{ALTO BOQUEIRAO,BAIRRO HAUER,BOQUEIRAO,BOQUEIRO,CENMITRO,CENTRO,CENTRO/REBOUCAS,CONJ 204,HAUER,PAROLIM,PAROLIN,PRADO VELHO,REBOUCAS,VILA HAUER,VILA HAUER,VILA PAROLIN,VL HAUER}	MARECHAL FLORIANO PEIXOTO
270	AVENIDA	MAL FLORIANO PEIXOTO	{BAIRRO REBOUCAS,BOQUEIRAO,CARMO,CENTRO,HAUER,N S DO CARMO,PAROLIM,PAROLIN,PCA N SRA DO CARMO,REBOUCAS,SEDE,VILA HAUER,VL HAUER,VL PAROLIN}	MARECHAL FLORIANO PEIXOTO
450	RUA	MARTIM AFONSO	{BATEL,BIGORILHO,BIGORRILHO,BIGORROLHO,CAMPINA DO SIQUEIRA,CENTRO,CHAMPAGNAT,CHAMPAGNAT,CURITIBA,JARDIM CHAMPAGNAT,J CHAMPAGNAT,MERCES,MOSSUNGUE,SAO FRANCISCO}	MARTIM AFONSO
241	RUA	MARTIN AFONSO	{ALTO SAO FRANCISCO,AP1001,BIGORRILHO,BATEL,BIGORILHO,BIGORRILHO,BIGORRILHO,CAMPAGNAT,CAMP DO SIQUEIRA,CAMPINA DE SIQUEIRA,CAMPINA DO SIQUEIRA,CAMPINA SIQUEIRA,CENTRO,CETRO,CHAMPAGNAT,CHAMPAGNATH,CHAMPANGANT,CHAMPANHAT,C.SIQUEIRA,MERCES,SAO FRANCISCO}	MARTIM AFONSO
4	RUA	MARTINS AFONSO	{BIGORRILHO,CENTRO}	MARTIM AFONSO
2	AVENIDA	MARTIN AFONSO	{BIGORRILHO,MERCES}	MARTIM AFONSO
623	RUA	NICOLA PELLANDA	{BOQUEIRAO,PINHEIRINHO,SITIO CERCADO,TATUQUARA,UMBARA,XAXIM}	NICOLA PELLANDA
362	RUA	NICOLA PELLANDA	{PINHEIRINHO,TATUQUARA,UMBARA,VILA SANTO ANTONIO}	NICOLA PELLANDA
3	RUA	NICOLA PELLANDA	{PINHEIRINHO,TATUQUARA,UMBARA}	NICOLA PELLANDA
757	AVENIDA	PRESIDENTE AFFONSO CAMARGO	{CAJURU,CAPANEMA,Capao da Imbuia,CAPAO DA IMBUIA,CAPAO IMBUIA,CAPAO RASO,CENTRO,CRISTO REI,CURITIBA,ESTACAO RODOFERROVIARIA DE CURITIBA,JARDIM BOTANICO,JARDIM BOTANICO,REBOUCAS,SANTA FELICIDADE}	PRESIDENTE AFFONSO CAMARGO
184	AVENIDA	PRES AFONSO CAMARGO	{ALTO CAJURU,CAJURU,CAPANEMA,CAPANEMA,CAPAO DA IMBUIA,CENTRO,CRISTO REI,JARDIM DAS AMERICAS,LIBERDADE,REBOUCAS,RODOFERROVIARIA}	PRESIDENTE AFFONSO CAMARGO
106	AVENIDA	PRESIDENTE AFONSO CAMARGO	{BAIRRO CRISTO REI,CAJURU,CAPANEMA,CAPAO DA IMBUIA,CAPAO DO IMBUIA,CENTRO,CRISTO REI,JARDIM BOTANICO,REBOUCAS}	PRESIDENTE AFFONSO CAMARGO
62	AVENIDA	PRES AFFONSO CAMARGO	{CAJURU,CAPANEMA,CAPAO DA IMBUIA,CAPAO DE IMBUIA,CAPAO IMBUIA,CENTRO,CRISTO REI,REBOUCAS,VL MORGENAUI}	PRESIDENTE AFFONSO CAMARGO
1026	AVENIDA	PRESIDENTE GETULIO VARGAS	{AGIA VERDE,AGUA V ERDE,AGUA VERDE,BAIRRO AGUA VERDE,CENTRO,IGUASSU,REBOUAS,REBOUCAS,REBOUCAS,SANTA QUITERIA,VILA ISABEL,VILA IZABEL}	PRESIDENTE GETULIO VARGAS
382	AVENIDA	GETULIO VARGAS	{AGUA VERDE,AGUA VERDES,ATLETICO,BATEL,CENTRO,IGUACU,REBOUCAS,SANTA CRUZ,SANTA TERESINHA,SANTA TEREZINHA,STA TEREZINHA,VILA ISABEL,VILA IZABEL,VL ISABEL,VL IZABEL,VL IZABEL}	PRESIDENTE GETULIO VARGAS
190	AVENIDA	PRES GETULIO VARGAS	{AGUA VERDE,AGUAVERDE,BAIRRO REBOUCAS,BATEL,CAPAO DA AMORA,CENTRO,IGUACU,IGUA U,REBOUCAS,STA QUITERIA,VILA IZABEL,VL ISABEL,VL IZABEL}	PRESIDENTE GETULIO VARGAS
41	AVENIDA	PRES. GETULIO VARGAS	{AGUA VERDE,REBOUCAS,SEDE,VILA ISABEL,VILA IZABEL,VL IZABEL}	PRESIDENTE GETULIO VARGAS
35	AVENIDA	PRES.GETULIO VARGAS	{AGUA VERDE,CENTRO,IGUASSU,REBOUCAS,VILA IZABEL}	PRESIDENTE GETULIO VARGAS
343	ALAMEDA	PRUDENTE DE MORAES	{CENTRO,CENTRO / MERCES,CENTRO/MERCES,MERCES,SAO FRANCISCO}	PRUDENTE DE MORAES
141	RUA	PRUDENTE DE MORAIS	{BIGORRILHO,CENTRO,MERCES,SALDANHA,SAO FRANCISCO}	PRUDENTE DE MORAES
87	RUA	PRUDENTE DE MORAES	{BAIRRO MERCES,BATEL,BIGORRILHO,CENTRO,MERCES,SAO FRANCISCO}	PRUDENTE DE MORAES
50	ALAMEDA	PRUDENTE DE MORAIS	{BIGORRILHO,CENTRO,MERCES,SAO FRANCISCO}	PRUDENTE DE MORAES





## F.2 ETAPA FINAL

Endereço (Tipo Logradouro + Logradouro)	Endereço normalizado	Levenshtein (<=4)	Soundex (PT-BR)	Metaphone (PT-BR)
RUA 24 DE MAIO	RUA 24 DE MAIO	OK	OK	OK
RUA VINTE E QUATRO DE MAIO	RUA 24 DE MAIO	NOK	NOK	NOK
RUA VINTE QUATRO DE MAIO	RUA 24 DE MAIO	NOK	NOK	NOK
RUA VINTE E QUATRO DE MAIO,	RUA 24 DE MAIO	NOK	NOK	NOK
RUA RUA 24 DE MAIO	RUA 24 DE MAIO	OK	NOK	NOK
ALAMEDA AUGUSTO STELLFELD	ALAMEDA AUGUSTO STELLFELD	OK	OK	OK
RUA AUGUSTO STELLFELD	ALAMEDA AUGUSTO STELLFELD	NOK	NOK	NOK
RUA AUGUSTO STELFELD	ALAMEDA AUGUSTO STELLFELD	NOK	NOK	NOK
ALAMEDA AUGUSTO STELFELD	ALAMEDA AUGUSTO STELLFELD	OK	OK	OK
RUA AUGUSTO STELFELD	ALAMEDA AUGUSTO STELLFELD	NOK	NOK	NOK
RUA BARAO DO CERRO AZUL	RUA BARAO DO CERRO AZUL	OK	OK	OK
RUA BARAO DO SERRO AZUL	RUA BARAO DO CERRO AZUL	OK	OK	OK
RUA BARAO CERRO AZUL	RUA BARAO DO CERRO AZUL	OK	NOK	NOK
AVENIDA BARAO DO CERRO AZUL	RUA BARAO DO CERRO AZUL	NOK	NOK	NOK
RUA BARAO SERRO AZUL	RUA BARAO DO CERRO AZUL	OK	NOK	NOK
RUA BARTOLOMEU LOURENCO DE GUSMAO	RUA BARTOLOMEU LOURENCO DE GUSMAO	OK	OK	OK
RUA BARTOLOMEU L DE GUSMAO	RUA BARTOLOMEU LOURENCO DE GUSMAO	NOK	NOK	NOK
RUA BARTOLOMEU L GUSMAO	RUA BARTOLOMEU LOURENCO DE GUSMAO	NOK	NOK	NOK
RUA BARTOLOMEU L.DE GUSMAO	RUA BARTOLOMEU LOURENCO DE GUSMAO	NOK	NOK	NOK
RUA BARTOLOMEU L. DE GUSMAO	RUA BARTOLOMEU LOURENCO DE GUSMAO	NOK	NOK	NOK
RUA BOM JESUS DE IGUAPE	RUA BOM JESUS DE IGUAPE	OK	OK	OK
RUA BOM JESUS DO IGUAPE	RUA BOM JESUS DE IGUAPE	OK	OK	OK
RUA BOM JESUS IGUAPE	RUA BOM JESUS DE IGUAPE	OK	NOK	NOK
RUA RUA BOM JESUS DE IGUAPE	RUA BOM JESUS DE IGUAPE	OK	NOK	NOK
RUA BOM JESUS DE IGUAPE	RUA BOM JESUS DE IGUAPE	OK	OK	OK
RODOVIA BR 116	RODOVIA BR 116	OK	OK	OK
RODOVIA BR CENTO E DEZESSEIS	RODOVIA BR 116	NOK	NOK	NOK
RODOVIA BR 116 CEASA	RODOVIA BR 116	NOK	NOK	NOK
RODOVIA BR 116 KM 10	RODOVIA BR 116	NOK	NOK	NOK
RODOVIA BR CENTO E DESESEIS	RODOVIA BR 116	NOK	NOK	NOK
RUA DESEMBARGADOR WESTPHALEN	RUA DESEMBARGADOR WESTPHALEN	OK	OK	OK
RUA DES WESTPHALEN	RUA DESEMBARGADOR WESTPHALEN	NOK	NOK	NOK
RUA DEZ WESTPHALEN	RUA DESEMBARGADOR WESTPHALEN	NOK	NOK	NOK
RUA DES. WESTPHALEN	RUA DESEMBARGADOR WESTPHALEN	NOK	NOK	NOK
RUA DES.WESTPHALEN	RUA DESEMBARGADOR WESTPHALEN	NOK	NOK	NOK
RUA DESEMB WESTPHALEN	RUA DESEMBARGADOR WESTPHALEN	NOK	NOK	NOK
ALAMEDA DOM PEDRO II	ALAMEDA DOM PEDRO II	OK	OK	OK
RUA DOM PEDRO II	ALAMEDA DOM PEDRO II	NOK	NOK	NOK
ALAMEDA D PEDRO II	ALAMEDA DOM PEDRO II	OK	NOK	NOK
RUA D PEDRO II	ALAMEDA DOM PEDRO II	NOK	NOK	NOK
ALAMEDA D. PEDRO II	ALAMEDA DOM PEDRO II	OK	NOK	NOK
ALAMEDA DOUTOR MURICY	ALAMEDA DOUTOR MURICY	OK	OK	OK
RUA DR MURICI	ALAMEDA DOUTOR MURICY	NOK	NOK	NOK
RUA DR MURICY	ALAMEDA DOUTOR MURICY	NOK	NOK	NOK
ALAMEDA DR MURICY	ALAMEDA DOUTOR MURICY	OK	NOK	NOK
ALAMEDA DR. MURICY	ALAMEDA DOUTOR MURICY	NOK	NOK	NOK
RUA ENGENHEIRO REBOUCAS	RUA ENGENHEIRO REBOUCAS	OK	OK	OK
RUA ENGENHEIROS REBOUCAS	RUA ENGENHEIRO REBOUCAS	OK	NOK	NOK
RUA ENG REBOUCAS	RUA ENGENHEIRO REBOUCAS	NOK	NOK	NOK
RUA ENG. REBOUCAS	RUA ENGENHEIRO REBOUCAS	NOK	NOK	NOK
RUA ENG.REBOUCAS	RUA ENGENHEIRO REBOUCAS	NOK	NOK	NOK
RUA MARECHAL DEODORO	RUA MARECHAL DEODORO	OK	OK	OK
RUA MAL DEODORO	RUA MARECHAL DEODORO	NOK	NOK	NOK
AVENIDA MARECHAL DEODORO	RUA MARECHAL DEODORO	NOK	NOK	NOK
RUA MAL. DEODORO	RUA MARECHAL DEODORO	NOK	NOK	NOK
AVENIDA MAL DEODORO	RUA MARECHAL DEODORO	NOK	NOK	NOK
AVENIDA MARECHAL FLORIANO PEIXOTO	AVENIDA MARECHAL FLORIANO PEIXOTO	OK	OK	OK
RUA MAL FLORIANO PEIXOTO	AVENIDA MARECHAL FLORIANO PEIXOTO	NOK	NOK	NOK

Endereço (Tipo Logradouro + Logradouro)	Endereço normalizado	Levenshtein (<=4)	Soundex (PT-BR)	Metaphone (PT-BR)
AVENIDA MAL FLORIANO PEIXOTO	AVENIDA MARECHAL FLORIANO PEIXOTO	NOK	NOK	NOK
RUA MARECHAL FLORIANO PEIXOTO	AVENIDA MARECHAL FLORIANO PEIXOTO	NOK	NOK	NOK
AVENIDA MAL. FLORIANO PEIXOTO	AVENIDA MARECHAL FLORIANO PEIXOTO	NOK	NOK	NOK
RUA MARTIM AFONSO	RUA MARTIM AFONSO	OK	OK	OK
RUA MARTIN AFONSO	RUA MARTIM AFONSO	OK	OK	OK
RUA MARTINS AFONSO	RUA MARTIM AFONSO	OK	NOK	NOK
AVENIDA MARTIN AFONSO	RUA MARTIM AFONSO	NOK	NOK	NOK
RUA NICOLA PELLANDA	RUA NICOLA PELLANDA	OK	OK	OK
RUA NICOLA PELANDA	RUA NICOLA PELLANDA	OK	OK	OK
RUA NICOLA PELANDA	RUA NICOLA PELLANDA	OK	OK	OK
AVENIDA PRESIDENTE AFFONSO CAMARGO	AVENIDA PRESIDENTE AFFONSO CAMARGO	OK	OK	OK
AVENIDA PRES AFONSO CAMARGO	AVENIDA PRESIDENTE AFFONSO CAMARGO	NOK	NOK	NOK
AVENIDA PRESIDENTE AFONSO CAMARGO	AVENIDA PRESIDENTE AFFONSO CAMARGO	OK	OK	OK
AVENIDA PRES AFFONSO CAMARGO	AVENIDA PRESIDENTE AFFONSO CAMARGO	NOK	NOK	NOK
AVENIDA PRESIDENTE GETULIO VARGAS	AVENIDA PRESIDENTE GETULIO VARGAS	OK	OK	OK
AVENIDA GETULIO VARGAS	AVENIDA PRESIDENTE GETULIO VARGAS	NOK	NOK	NOK
AVENIDA PRES GETULIO VARGAS	AVENIDA PRESIDENTE GETULIO VARGAS	NOK	NOK	NOK
AVENIDA PRES. GETULIO VARGAS	AVENIDA PRESIDENTE GETULIO VARGAS	NOK	NOK	NOK
AVENIDA PRES.GETULIO VARGAS	AVENIDA PRESIDENTE GETULIO VARGAS	NOK	NOK	NOK
ALAMEDA PRUDENTE DE MORAES	ALAMEDA PRUDENTE DE MORAES	OK	OK	OK
RUA PRUDENTE DE MORAIS	ALAMEDA PRUDENTE DE MORAES	NOK	NOK	NOK
RUA PRUDENTE DE MORAES	ALAMEDA PRUDENTE DE MORAES	NOK	NOK	NOK
ALAMEDA PRUDENTE DE MORAIS	ALAMEDA PRUDENTE DE MORAES	OK	OK	OK
AVENIDA SETE DE SETEMBRO	AVENIDA SETE DE SETEMBRO	OK	OK	OK
AVENIDA 7 DE SETEMBRO	AVENIDA SETE DE SETEMBRO	OK	NOK	NOK
RUA SETE DE SETEMBRO	AVENIDA SETE DE SETEMBRO	NOK	NOK	NOK
RUA 7 DE SETEMBRO	AVENIDA SETE DE SETEMBRO	NOK	NOK	NOK
AVENIDA SETE DE SETEMBRO	AVENIDA SETE DE SETEMBRO	OK	OK	OK
AVENIDA WINSTON CHURCHILL	AVENIDA WINSTON CHURCHILL	OK	OK	OK
AVENIDA WINSTON CHURCHIL	AVENIDA WINSTON CHURCHILL	OK	OK	OK
RUA WINSTON CHURCHILL	AVENIDA WINSTON CHURCHILL	NOK	NOK	NOK
RUA WINSTON CHURCHIL	AVENIDA WINSTON CHURCHILL	NOK	NOK	NOK
RUA XV DE NOVENBRO	RUA XV DE NOVENBRO	OK	OK	OK
RUA QUINZE DE NOVENBRO	RUA XV DE NOVENBRO	NOK	NOK	NOK
RUA 15 DE NOVENBRO	RUA XV DE NOVENBRO	OK	NOK	NOK
AVENIDA XV DE NOVENBRO	RUA XV DE NOVENBRO	NOK	NOK	NOK
	<b>Percentual de acerto</b>	45,74468085	32,9787234	32,9787234